Subject: Variable Pixel Spacing for Images in IDL
Posted by robseigel on Fri, 29 Nov 2013 13:54:51 GMT
View Forum Message <> Reply to Message

Hello,

I have a 2D array that is an x-z vertical plane, where the vertical axis is stretched from ~ 25 meter spacing between rows at the bottom (index = 0) to 100 meter spacing at the top. I am trying to plot these data as an image with the y-axis "stretched" appropriately:

```
yaxis = zcoords  ; Vertical axis [12.4,37.67,63.44...16400,16500,16600]
xaxis = xcoords  ; Horizontal axis [0,100,200...50900,51000,51100]
p = image(rgbData,xaxis,yaxis,/buffer, $
        axis_style=2)
```

But, the above code does stretch the image properly. Using contour stretches the data fine, but I would prefer to plot these data as an image. Does anyone know how I can plot these data as an image so that the data points match the locations specified by yaxis?

Thanks,
Rob

Subject: Re: Variable Pixel Spacing for Images in IDL
Posted by David Fanning on Fri, 29 Nov 2013 14:23:29 GMT
View Forum Message <> Reply to Message

Robert Seigel writes:

> I have a 2D array that is an x-z vertical plane, where the vertical axis is stretched from ~ 25 meter spacing between rows at the bottom (index = 0) to 100 meter spacing at the top. I am trying to plot these data as an image with the y-axis "stretched" appropriately:
>
> yaxis = zcoords ; Vertical axis [12.4,37.67,63.44...16400,16500,16600]
> xaxis = xcoords ; Horizontal axis [0,100,200...50900,51000,51100]
> p = image(rgbData,xaxis,yaxis,/buffer, $
>   axis_style=2)
>
> But, the above code does stretch the image properly. Using contour stretches the data fine, but I would prefer to plot these data as an image. Does anyone know how I can plot these data as an image so that the data points match the locations specified by yaxis?

I would try PG_PlotImage by Paolo Grigis. You can find it in the "public" directory of the IDL Coyote Library, among other places. It is a direct graphics solution, however.

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: Variable Pixel Spacing for Images in IDL
Posted by lecacheux.alain on Fri, 29 Nov 2013 14:24:55 GMT
View Forum Message <> Reply to Message

Le vendredi 29 novembre 2013 14:54:51 UTC+1, Robert Seigel a écrit :
> Hello,
>
>
>
> I have a 2D array that is an x-z vertical plane, where the vertical axis is stretched from ~ 25 meter spacing between rows at the bottom (index = 0) to 100 meter spacing at the top. I am trying to plot these data as an image with the y-axis "stretched" appropriately:
>
>
>
> yaxis = zcoords ; Vertical axis [12.4,37.67,63.44...16400,16500,16600]
>
> xaxis = xcoords ; Horizontal axis [0,100,200...50900,51000,51100]
>
> p = image(rgbData,xaxis,yaxis,/buffer, $
>
> axis_style=2)
>
>
>
> But, the above code does stretch the image properly. Using contour stretches the data fine, but I would prefer to plot these data as an image. Does anyone know how I can plot these data as an image so that the data points match the locations specified by yaxis?
>
>
>
> Thanks,
>
> Rob

rgbData is a 2D array, xaxis and yaxis are 1D vectors.

if you can use IDL 8.2.3, following the IMAGE documentation:
"If Data, X, and Y are two-dimensional arrays with the same number of elements, then the X and

Y coordinates will be tested to determine if the points are regularly or irregularly spaced. [...], if the data are irregularly spaced, then IDL will automatically grid the data so that the points lie on a regular grid".
Then you should succeed with:
p = image(rgbdata, rebin(xaxis,nx,ny), rebin(reform(yaxis,1,ny),nx,ny), ...)

if you cannot, you have to do interpolation by yourself. For instance:
regYaxis = (yaxis[-1] - yaxis[0])*findgen(ny)/(ny -1)
regData = interpolate(rgbData, xaxis, interpol(findgen(ny), yaxis, regYaxis), /GRID)
p = image(regData, xaxis, regYaxis, ...)

alx.

Subject: Re: Variable Pixel Spacing for Images in IDL
Posted by robseigel on Fri, 29 Nov 2013 16:34:17 GMT
View Forum Message <> Reply to Message

On Friday, November 29, 2013 8:54:51 AM UTC-5, Robert Seigel wrote:
> Hello,
>
>
>
> I have a 2D array that is an x-z vertical plane, where the vertical axis is stretched from ~ 25 meter spacing between rows at the bottom (index = 0) to 100 meter spacing at the top. I am trying to plot these data as an image with the y-axis "stretched" appropriately:
>
>
>
> yaxis = zcoords ; Vertical axis [12.4,37.67,63.44...16400,16500,16600]
>
> xaxis = xcoords ; Horizontal axis [0,100,200...50900,51000,51100]
>
> p = image(rgbData,xaxis,yaxis,/buffer, $
>
> axis_style=2)
>
>
>
> But, the above code does stretch the image properly. Using contour stretches the data fine, but I would prefer to plot these data as an image. Does anyone know how I can plot these data as an image so that the data points match the locations specified by yaxis?
>
>
>
> Thanks,
>
> Rob

Thank you for the replies.

Alx,

As I have tried in the past, I am unable to use two-dimensional arrays for X and Y in the IMAGE function. Using your [Alex] example:

```
IDL>                    p =  image(rgbData,rebin(xaxis,xcount,zcount),rebin(reform(yaxis,
1,zcount),xcount,zcount),/buffer, $
IDL>                              axis_style=2)
% IMAGE: X must be a vector.
```

I am not sure why IMAGE does not accept X and Y as 2d arrays. However, your second suggestion worked well with one slight modification to the interpolate call [indgen(xcount) rather than xaxis]:

```
regYaxis = (zcoords[-1] - zcoords[0])*findgen(zcount)/(zcount -1)
data = interpolate(data, indgen(xcount), interpol(findgen(zcount), zcoords, regYaxis), /GRID)
```

But, I cannot interpolate these data because they are flags and interpolation between them results in incorrect classification at many locations. The values in the array are one of [-4,-3,-2,-1,0,1,2,3,4], so e.g. when a 4 is next to a 0 the interpolation often creates a false classification.

David,

This routine is exactly what I was looking for!

Cheers,
Rob

Subject: Re: Variable Pixel Spacing for Images in IDL
Posted by David Fanning on Fri, 29 Nov 2013 16:43:21 GMT
View Forum Message <> Reply to Message

Robert Seigel writes:

> Thank you for the replies.
>
> Alx,
>
> As I have tried in the past, I am unable to use two-dimensional arrays for X and Y in the IMAGE function. Using your [Alex] example:
>
> IDL>   p = image(rgbData,rebin(xaxis,xcount,zcount),rebin(reform(yaxis,
1,zcount),xcount,zcount),/buffer, $

> IDL>   axis_style=2)
> % IMAGE: X must be a vector.
>
> I am not sure why IMAGE does not accept X and Y as 2d arrays. However, your second suggestion worked well with one slight modification to the interpolate call [indgen(xcount) rather than xaxis]:
>
> regYaxis = (zcoords[-1] - zcoords[0])*findgen(zcount)/(zcount -1)
> data = interpolate(data, indgen(xcount), interpol(findgen(zcount), zcoords, regYaxis), /GRID)
>
> But, I cannot interpolate these data because they are flags and interpolation between them results in incorrect classification at many locations. The values in the array are one of [-4,-3,-2,-1,0,1,2,3,4], so e.g. when a 4 is next to a 0 the interpolation often creates a false classification.
>
> David,
>
> This routine is exactly what I was looking for!

Oh, damn! Another blow struck for direct graphics. We may never be able
to get rid of these things. ;-)

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: Variable Pixel Spacing for Images in IDL
Posted by lecacheux.alain on Fri, 29 Nov 2013 17:14:37 GMT
View Forum Message <> Reply to Message

Le vendredi 29 novembre 2013 17:43:21 UTC+1, David Fanning a écrit :
> Robert Seigel writes:
>
>
>
>> Thank you for the replies.
>
>>
>
>> Alx,
>
>>

>
>> As I have tried in the past, I am unable to use two-dimensional arrays for X and Y in the IMAGE function. Using your [Alex] example:
>
>>
>
>> IDL> p = image(rgbData,rebin(xaxis,xcount,zcount),rebin(reform(yaxis, 1,zcount),xcount,zcount),/buffer, $
>
>> IDL> axis_style=2)
>
>> % IMAGE: X must be a vector.
>
>>
>
>> I am not sure why IMAGE does not accept X and Y as 2d arrays. However, your second suggestion worked well with one slight modification to the interpolate call [indgen(xcount) rather than xaxis]:
>
>>
>
>> regYaxis = (zcoords[-1] - zcoords[0])*findgen(zcount)/(zcount -1)
>
>> data = interpolate(data, indgen(xcount), interpol(findgen(zcount), zcoords, regYaxis), /GRID)
>
>>
>
>> But, I cannot interpolate these data because they are flags and interpolation between them results in incorrect classification at many locations. The values in the array are one of [-4,-3,-2,-1,0,1,2,3,4], so e.g. when a 4 is next to a 0 the interpolation often creates a false classification.
>
>>
>
>> David,
>
>>
>
>> This routine is exactly what I was looking for!
>
>
>
> Oh, damn! Another blow struck for direct graphics. We may never be able
>
> to get rid of these things. ;-)
>
>
>

> 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.")

The IMAGE function does work with X and Y as 2D arrays. Try:

im = image(dist(200,100), rebin(findgen(200),200,100), rebin((findgen(1,100)),200,100), AXIS_STYLE=2)

It works in this case, because re-gridding is not needed and then not used.
But the automatic re-gridding is far from optimal (due to present limitations in IDL's internal gridding software). It is usually very slow and can even crash your IDL session.
Hope that this will be fixed in IDL 8.3 !
alx.

Subject: Re: Variable Pixel Spacing for Images in IDL
Posted by David Fanning on Fri, 29 Nov 2013 17:22:24 GMT
View Forum Message <> Reply to Message

alx writes:

> The IMAGE function does work with X and Y as 2D arrays. Try:
>
> im = image(dist(200,100), rebin(findgen(200),200,100), rebin((findgen(1,100)),200,100), AXIS_STYLE=2)
>
> It works in this case, because re-gridding is not needed and then not used.
> But the automatic re-gridding is far from optimal (due to present limitations in IDL's internal gridding software). It is usually very slow and can even crash your IDL session.
> Hope that this will be fixed in IDL 8.3 !

Oh. Bummer! :-(

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: Variable Pixel Spacing for Images in IDL
Posted by David Fanning on Fri, 29 Nov 2013 17:24:09 GMT
View Forum Message <> Reply to Message

David Fanning writes:

>> It works in this case, because re-gridding is not needed and then not used.
>> But the automatic re-gridding is far from optimal (due to present limitations in IDL's internal gridding software). It is usually very slow and can even crash your IDL session.
>> Hope that this will be fixed in IDL 8.3 !
>
> Oh. Bummer! :-(

Reminds me of the play, Waiting for Godot. :-)

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: Variable Pixel Spacing for Images in IDL
Posted by David Fanning on Fri, 29 Nov 2013 17:40:07 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> Reminds me of the play, Waiting for Godot. :-)

Sorry. I'm putting off doing important work this morning and my productivity quota is already though the roof after solving that map problem, so I'm just fooling around. But, it occurs to me that we can resolve the controversy of whether the refurbished iTools graphics system introduced in IDL 8.0 should be called "new graphics" or

"function graphics", by using this alliterative alternative: Godot
Graphics. It has the advantage of being catchy *and* descriptive. ;-)

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: Variable Pixel Spacing for Images in IDL
Posted by lecacheux.alain on Fri, 29 Nov 2013 18:14:38 GMT
View Forum Message <> Reply to Message

Le vendredi 29 novembre 2013 18:40:07 UTC+1, David Fanning a écrit :
> David Fanning writes:
>
>
>
>> Reminds me of the play, Waiting for Godot. :-)
>
>
>
> Sorry. I'm putting off doing important work this morning and my
>
> productivity quota is already though the roof after solving that map
>
> problem, so I'm just fooling around. But, it occurs to me that we can
>
> resolve the controversy of whether the refurbished iTools graphics
>
> system introduced in IDL 8.0 should be called "new graphics" or
>
> "function graphics", by using this alliterative alternative: Godot
>
> Graphics. It has the advantage of being catchy *and* descriptive. ;-)
>
>
>
> 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.")

In this case, I am Vladimir, you are Estragon !
alx.

Subject: Re: Variable Pixel Spacing for Images in IDL
Posted by David Fanning on Fri, 29 Nov 2013 18:16:37 GMT
View Forum Message <> Reply to Message

alx writes:

> In this case, I am Vladimir, you are Estragon !

Haha!

Dave
--
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.")
