Subject: 2D array as colour dot image
Posted by markjamie on Mon, 14 Jan 2013 20:22:37 GMT
View Forum Message <> Reply to Message

Hi

Think I'm having a slow day as I'm sure the answer to this will be obvious...

I'm trying to use the new / function graphics in IDL8 to plot a 2D array as a colour scaled dotted plot, where each dot represents the value of an element in the array. In essence, the plots I'm trying to create are similar to those featured throughout this presentation http://www.insinume2012.com/downloads/pdf/Session_2/2.4._Zha ng.pdf (see slides 5, 6, 7, 9, 10, 11, 12, 13).

I've done similar things in the past with direct graphics using shade_surf and automatically rotating the surface so the view is "top down". I've tried this with surface in function graphics and while it works, when the number of points gets large it becomes very slow...

Is there any way to do it in function graphics with a 2D image plot?

Given an image is just an array of pixels it seems like it would be obvious.....like I say, think I'm just having a slow day!

Cheers

Subject: 2D array as colour dot image
Posted by markjamie on Mon, 14 Jan 2013 22:44:03 GMT
View Forum Message <> Reply to Message

Thinking about it, this is actually a 3D array problem just viewed in two dimensions with colour as the third dimension.

Subject: Re: 2D array as colour dot image
Posted by cgguido on Mon, 14 Jan 2013 23:01:45 GMT
View Forum Message <> Reply to Message

On Monday, January 14, 2013 4:44:03 PM UTC-6, mark...@gmail.com wrote:
> Thinking about it, this is actually a 3D array problem just viewed in two dimensions with colour as the third dimension.

This runs using Coyote Graphics and DG, but the principle I would guess is the same. First take a 2D histogram to bin you data. Each histogram count is then associated to a colour. Finally, the data that contributed to each bin are plotted with that colour.

This code is messy and not optimized, but I need it rarely...

```
PRO plotc,  x0,  y0,  ctable = ctable, every = every, buffer = buffer, $
        overplot = overplot, quickie = quickie, xlog = xlog, ylog = ylog, $
        silent = silent, _extra = eee,  bin = bin, histogram = h2

compile_opt idl2

backgrnd = !P

IF ~keyword_set(ctable) THEN ctable = 33
IF n_elements(x0) EQ 1 THEN BEGIN

  IF ~keyword_set(overplot) THEN cgplot, $
  x, y, $
  /noda, back=!p.background, $
  color=!p.color, chars=1.5,  $
  _extra = eee

  cgplot, x0, y0,  /ov, col=cgcolor('red'), xlog = xlog, ylog = ylog, _extra=eee

ENDIF

IF  n_elements(x0) GT 1 THEN BEGIN
  IF keyword_set(every) THEN BEGIN
    message,  "Every is not working right!"
    n = n_elements(x0)
    w = indgen(floor(1.*n/every))*every < n-1
    nw = n_elements(w)
    x = x0[w]
    y = y0[w]
    message, /inf, "Plotting "+n2s(100.*nw/n )+"% of the points. That's "+n2s(nw)+" points"
  ENDIF ELSE BEGIN
    x = x0
    y = y0
  ENDELSE

  IF ~keyword_set(quickie) THEN BEGIN
    IF ~keyword_set(silent) THEN ginfo,  "Binning data the nice way..."
    h2=sshist_2d(x,y, re=ri1, cost=co,  outbin = bin)
  ENDIF
  IF keyword_set(quickie) THEN BEGIN
    IF ~keyword_set(silent) THEN ginfo,  "Binning data the quick way..."
    co = 0
    deltax = stddev(x)
    deltay = stddev(y)
    IF ~keyword_set(bin) THEN bin = .5*[ deltax/alog10(n_elements(x)) > 1,
deltay/alog10(n_elements(y)) > 1]
    h2=hist_nd([reform(x)##1,reform(y)##1], re=ri1,bin)
```

```
   ENDIF
   IF ~keyword_set(silent) THEN ginfo,  "    ... Done Binning."
   xmin = min(X) &  ymin = min(y)
   h2size = size(h2, /dimen)
   col = h2[ floor((x-xmin)/bin[0]) + floor((y-ymin)/bin[1])*h2size[0] ]


   cgloadct, ctable

   IF keyword_set(buffer) THEN BEGIN
      set_plot, 'z'
      device, z_buff=0, set_res=[!D.X_SIZE,!D.Y_SIZE]
   ENDIF

   col = bytscl(col)

   IF ~keyword_set(overplot) THEN cgplot, $
   x, y, /noda, back=!p.background, $
   color=!p.color, chars=1.5, xlog = xlog, ylog = ylog,  _extra = eee

;   IF keyword_set(xlog) AND xmin EQ 0 THEN x = x > .1
;   IF keyword_set(ylog) AND ymin EQ 0 THEN y = x > .1

   ;;;;;;-----------;;;;; device,  decomposed = 0

   cmin = min(fix(col),  max = cmax)

;colstr = string(round(1.*col))
   ;;;;;;-----------;;;;; plots, x, y, col=(indgen(255))[histobin(ri1, col)], _extra=eee ;sysm=.1
 ; plots, x, y, col=cgcolor(colstr), _extra=eee ;sysm=.1

   for c=cmin, cmax do BEGIN
      w=where(col EQ c)
      if w[0] ne -1 THEN BEGIN
         cgplot, x[w], y[w],  /ov, col=c, _extra=eee ;sysm=.1
         ;plots, x[w], y[w],  col=c, _extra=eee ;sysm=.1
      ENDIF
   ENDFOR

ENDIF

IF keyword_set(buffer) THEN BEGIN
   a=tvrd(/tr)
   set_plot, 'x'
   tv, 255b-a, /tr
ENDIF

!P = backgrnd
```

```
cgloadct

RETURN
END
```

Subject: Re: 2D array as colour dot image
Posted by chris_torrence@NOSPAM on Tue, 15 Jan 2013 18:06:38 GMT
View Forum Message <> Reply to Message

Hi Mark,

How about something like this, using New Graphics?

```
F1 = RANDOMN(seed, 10000000)
F2 = f1 + RANDOMN(seed, 10000000)
Result = HIST_2D(F1, F2, MIN1=-2, MAX1=2.001d, $
  MIN2=-2, MAX2=2.001d, BIN1=0.05d, BIN2=0.05d)
; HIST_2D seems to include a column & row of all zeroes. Remove it.
Result = Result[0:-2, 0:-2]
x = -2 + 0.05d*FINDGEN(80)
y = x
print, min(result, max=mx), mx

i = IMAGE(result, x, y, RGB_TABLE=74, AXIS_STYLE=1, $
  XTITLE='X data', YTITLE='Y data', POSITION=[0.1,0.1,0.8,0.84], $
  TITLE='Ran1 vs (Ran1+Ran2)')
!null = COLORBAR(TARGET=i, /ORIENTATION, $
  TITLE='Magnitude', TEXTPOS=1, $
  POSITION=[0.82,0.1,0.87,0.84])
c = CONTOUR(result, x, y, /OVERPLOT)

i.Save, 'scatter.png', RESOLUTION=96, BORDER=10
```

Obviously, you would need to change the HIST_2D inputs, and I just hardcoded the X and Y variables.

This should create a scatter plot, colored by "magnitude", which here is just the number of samples in each bin in the 2D histogram. Then I added a colorbar on the right side, and a contour plot with labels.

Note that this example requires IDL 8.2.1.

Regarding the speed difference between direct and new graphics: for direct graphics the surface is just "burned" into the window, so it only has to draw it once. For new graphics, the image (or surface) is a dynamic object, which can be manipulated after creation.

Hope this helps!

-Chris
ExelisVIS
p.s. here's a screenshot of the resulting graphic:
http://www.flickr.com/photos/79705059@N06/8383412553/

Subject: Re: 2D array as colour dot image
Posted by markjamie on Tue, 15 Jan 2013 23:35:59 GMT
View Forum Message <> Reply to Message

Hi Chris

That's perfect! Just what I needed. IDL 8.2.1 is ok - out of interest which part of the code is 8.2.1 specific?

The 2d histogram part is ok - I've already done that bit just couldn't work out the image plot. Speaking of hist_2d I've used this function a few times and when plotting the results with direct graphics I've sometimes seen repeated "stripes" across the colour mapped shade_surf plot. Have you seen this before? Apologies, I don't have an example image to show you what I mean.

M

Subject: Re: 2D array as colour dot image
Posted by markjamie on Tue, 15 Jan 2013 23:56:54 GMT
View Forum Message <> Reply to Message

Looking at your code again, I guess it's the Brewer colour table which makes it 8.2.1 specific?

M

Subject: Re: 2D array as colour dot image
Posted by markjamie on Wed, 16 Jan 2013 00:06:02 GMT
View Forum Message <> Reply to Message

Thanks for your input Gianguido Cianci

Subject: Re: 2D array as colour dot image
Posted by chris_torrence@NOSPAM on Wed, 16 Jan 2013 04:28:07 GMT
View Forum Message <> Reply to Message

On Tuesday, January 15, 2013 4:56:54 PM UTC-7, mark...@gmail.com wrote:
> Looking at your code again, I guess it's the Brewer colour table which makes it 8.2.1 specific?

>
>
>
> M

Yes, it's the Brewer color table. And there were also some changes made to the Colorbar function, so I can't guarantee that it would look exactly the same in an earlier IDL version.

Glad that the code was a help!

Can I ask a favor? If you get something working that looks nice, could you post either the code or a screenshot? It's always great to see what people are trying to accomplish with IDL's graphics.

Thanks!

-Chris
p.s. regarding the "stripes" in the hist_2d, I noticed that I had to remove that last column & row of all zeroes. I haven't studied the code in the lib directory, but I would bet it's some sort of artifact of the histogram function. We probably need to take a good hard look at hist_2d, and bring it into the 21st century...

Subject: Re: 2D array as colour dot image
Posted by David Fanning on Wed, 16 Jan 2013 15:11:13 GMT

View Forum Message <> Reply to Message

Chris Torrence writes:

> Yes, it's the Brewer color table. And there were also some changes made to the Colorbar function, so I can't guarantee that it would look exactly the same in an earlier IDL version.

I just point out that the only thing "new" about Brewer color tables and
a working Colorbar function is that they were first introduced into IDL
proper in IDL 8.2.1. I've been using Brewer color tables and a working
colorbar program in IDL since 2008 and 1996, respectively. There is
nothing special about this plot that would cause it to have to be done
in IDL 8.2.1. I think it can be done in any version of IDL you happen to
have lying around. :-)

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: 2D array as colour dot image
Posted by markjamie on Fri, 25 Jan 2013 00:15:27 GMT
View Forum Message <> Reply to Message

So I've got an image of my array working quite well, the down side is that often my array is sparsely populated and contains a lot of zeros.

In my code I've been using colour table 39 as it provides fairly good colour variation. The bad thing about using this CT is that the bottom colour is black, which given many of my array elements are zero, means that the majority of my image is black.

I tried replacing all the zeros in my array with. !values.d_nan into the hope that they're not plotted, but they still get drawn as the bottom colour (black).

How can I plot my array using only values above zero, meaning the majority of the image is transparent / the same colour as the background (white), helping the non-zero elements to stand out?

Subject: Re: 2D array as colour dot image
Posted by David Fanning on Fri, 25 Jan 2013 00:25:39 GMT
View Forum Message <> Reply to Message

markjamie@gmail.com writes:

> So I've got an image of my array working quite well, the down side is that often my array is sparsely populated and contains a lot of zeros.
>
> In my code I've been using colour table 39 as it provides fairly good colour variation. The bad thing about using this CT is that the bottom colour is black, which given many of my array elements are zero, means that the majority of my image is black.
>
> I tried replacing all the zeros in my array with. !values.d_nan into the hope that they're not plotted, but they still get drawn as the bottom colour (black).
>
> How can I plot my array using only values above zero, meaning the majority of the image is transparent / the same colour as the background (white), helping the non-zero elements to stand out?

You should have a look at the last example in the new Coyote Graphics
Plot Gallery. In the Density Plot of Two Variables with Contours
Overlaid example, the values that are zero are displayed in a light gray
color. You can make it any color you like.

  http://www.idlcoyote.com/gallery/

This is exactly the situation you are facing.

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.")