
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 your 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 = y > .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
```
