Subject: Re: Advice on making a plot
Posted by David Fanning on Mon, 17 Nov 2003 19:25:16 GMT
View Forum Message <> Reply to Message

Lawrence Bleau writes:

> Hello, I'm running IDL 5.2 on OpenVMS 7.1-2. And no, I cannot upgrade, as
> IDL isn't made for VMS any more :-( This is about design advice, though,
> not a problem.
>
> I've been tasked with writing code to create a plot of values. I'm given
> a 2-dimensional array of values, which are bins. The bins are the same
> size in each dimension, but different in the x- and y- dimensions; i.e.,
> each bin is a rectangle when viewed in the x,y space.
>
> I'm familiar with the plot command, and have an idea how to go about
> writing IDL code to shade individual rectangles different colors in
> specific locations on the plot field:
>  for i=0,n-1
>    for j=0,m-1
>      translate x(i), y(i) to plot coordinates
>      calculate corners of rectangle
>      shade rectangle appropriate color
>    endfor
>  endfor
> One caveat: the final bin in each coordinate (along right and top edges) is
> larger than the other bins, so its rectangle will be of a different size.
>
> I'm thinking, though, that this operation is a bit inefficient and might be
> better accomplished by treating the array of values as an image and using
> some image display commands. How would this be done, though, with
> non-square pixels? That's just the first problem. The second is the
> different sized rectangles along the right and top sides. There may be
> other problems to address which would make an image approach unwieldy.
>
> Let me know your thoughts. Thanks.

This shouldn't be too hard, although slow if you
have a lot of points. Then, of course, there is the VAX... :-)

Here is an example:

```
PRO Example
Device, Decomposed=0
x = [0.0, 1, 2, 3, 4, 6]
y= [0.0, 1, 2, 3, 4, 6]
Plot, x, y, /NoData
FOR j=0,N_Elements(x)-2 DO BEGIN
```

```
    FOR k=0,N_Elements(x)-2 DO BEGIN
      TVLCT, Fix(Randomu(seed,1)*255), Fix(Randomu(seed,1)*255), $
        Fix(Randomu(seed,1)*255), 255
      Polyfill, [x[j], x[j],   x[j+1], x[j+1], x[j]], $
            [y[k], y[k+1], y[k+1], y[k],   y[k]], $
            Color=255
    ENDFOR
  ENDFOR
  END
```

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

---

## Subject: Re: Advice on making a plot
Posted by bleau on Tue, 18 Nov 2003 20:18:49 GMT
View Forum Message <> Reply to Message

In article <MPG.1a22ce8b188d84eb989740@news.frii.com>, David  Fanning
<david@dfanning.com> writes:
[snip original post]


> This shouldn't be too hard, although slow if you
> have a lot of points. Then, of course, there is the VAX... :-)
>
> Here is an example:
>
>  PRO Example
>  Device, Decomposed=0
>  x = [0.0, 1, 2, 3, 4, 6]
>  y=  [0.0, 1, 2, 3, 4, 6]
>  Plot, x, y, /NoData
>  FOR j=0,N_Elements(x)-2 DO BEGIN
>    FOR k=0,N_Elements(x)-2 DO BEGIN
>      TVLCT, Fix(Randomu(seed,1)*255), Fix(Randomu(seed,1)*255), $
>        Fix(Randomu(seed,1)*255), 255
>      Polyfill, [x[j], x[j],   x[j+1], x[j+1], x[j]], $
>            [y[k], y[k+1], y[k+1], y[k],   y[k]], $
>            Color=255
>    ENDFOR

```
> ENDFOR
> END
>
> Cheers,
>
> David
```

Thanks, David.  This is about what I do now.  I was sort of hoping for
something along the lines of

```
  create large array A (82x22 or bigger if need be)
  fill array A with values based on some special formula
  do one TVxxx call with A to display it all at once
```

The catch, of course, is in step 2.  If each element of A is displayed as
one pixel, then one won't be able to see the colored regions.  Since the
size of each rectangular region is L/22 wide by L/82 high, where L the the
size of the side of the square plot region, we'd need A to be 1804x1804 (22*82),
and each region with A would be 82 wide by 22 high, and all the same value.

That's sort of what I was thinking.  Of course, this could be optimized a bit
by removing the common factor of 2.  While this approach minimizes the number
of plotting or drawing calls, it wastes temporary storage space.

I'm starting to think this other approach isn't much better, and is in fact
worse, that the double FOR loop with repeated calls to polyfill.  Unless
there's a way to dramatically improve its design.  Thoughts?

Lawrence Bleau
University of Maryland
Physics Dept., Space Physics Group
301-405-6223
bleau@umtof.umd.edu

Subject: Re: Advice on making a plot
Posted by David Fanning on Tue, 18 Nov 2003 21:53:49 GMT
View Forum Message <> Reply to Message

Lawrence Bleau writes:

```
> Thanks, David.  This is about what I do now.  I was sort of hoping for
> something along the lines of
>
>   create large array A (82x22 or bigger if need be)
>   fill array A with values based on some special formula
>   do one TVxxx call with A to display it all at once
>
```

> The catch, of course, is in step 2.  If each element of A is displayed as
> one pixel, then one won't be able to see the colored regions.  Since the
> size of each rectangular region is L/22 wide by L/82 high, where L the the
> size of the side of the square plot region, we'd need A to be 1804x1804 (22*82),
> and each region with A would be 82 wide by 22 high, and all the same value.

I'd do it small (fast), then blow it up to the size
you need. Fortunately, someone has already written
CONGRID for us. :-)

> That's sort of what I was thinking.  Of course, this could be optimized a bit
> by removing the common factor of 2.  While this approach minimizes the number
> of plotting or drawing calls, it wastes temporary storage space.
>
> I'm starting to think this other approach isn't much better, and is in fact
> worse, that the double FOR loop with repeated calls to polyfill.  Unless
> there's a way to dramatically improve its design.  Thoughts?

Well, I wouldn't let anyone see you do it! I'd do
the drawing in a pixmap and slam that over to the
display window when I was done. It won't take less
time, but it will look professional. :-)

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155