Subject: Computer Graphics Question
Posted by David Fanning on Wed, 04 Aug 2004 23:05:21 GMT
View Forum Message <> Reply to Message

Folks,

Here is a question you can probably answer if you were paying attention in your computer graphics course.

Several questions this week have prodded me into writing my own annotation tool for my Catalyst Library. In just two days, I've gotten unbelievably far. I can now grab text anywhere in the window and drag it anywhere else, change all the text properties via a property sheet, group text lines, etc., etc. To be honest, I've never seen the like in IDL outside of iTools. (It's a quixotic quest to build iTool like functionality in direct graphics, I admit, but it has one HUGE advantage: I can understand the code!!)

I'm even able to rotate the text at some arbitrary angle ... sorta. There is my problem. Each text line has a "box" associated with it, which describes the location of the text in a window (normalized coordinates). The window can then "select" a text line (to move, for example) by asking the text object "Is this point location inside your box." (The box also allows me to write my text with a background color.)

Well and good as long as the window has just as many pixels in the X direction as in the Y direction.

My rotation code looks like this. I am rotating the four corners of the box after calculating and translating the middle of the box to the origin:

```
    ; Translate to origin and rotate about the Z axis.
    T3D, /Reset, $
      Translate=[-midx, -midy, 0], $
  Rotate=[0, 0, self.orientation], $
      Scale=[1/ratio, 1*ratio, 0], $
      Matrix=ctm
    p1 = Transpose([x1, y1, 0, 1])
    p2 = Transpose([x1, y2, 0, 1])
    p3 = Transpose([x2, y2, 0, 1])
    p4 = Transpose([x2, y1, 0, 1])
```

```
v1 = ctm ## p1
v2 = ctm ## p2
v3 = ctm ## p3
v4 = ctm ## p4

; Translate back to where you found the box.
T3D, /Reset, Translate=[midx, midy, 0], Matrix=ctm
v1 = ctm ## v1
v2 = ctm ## v2
v3 = ctm ## v3
v4 = ctm ## v4
```

In this case "ratio" is the ratio of !D.Y_Size/!D.X_Size.
The code works perfectly for 0 and 90 degree rotations.
It does not work so well for a, say, 45 degree rotation,
unless the ratio is 1 (the window has the same X and Y size).

What I am looking for is the way to set the scale factor
for rotation when the window is NOT square. I've thought
that the ratio needs to be adjusted for the amount of
X and Y in the rotation. (You rotate a little and the ratio
is mostly influenced by X. You rotate a lot and the ratio
is mostly influenced by Y.) I've tried various Sine and Cosine
functions. Any maybe I just haven't hit on the right combination
yet. But it seems to me this must be a common problem.
(I know in examples the windows are *always* square, but
surely that's not true in the real world!)

Anyone have an idea?

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/