2D Local & Global Coordinates
Introduction to Computer Graphics
Arizona State University

Dianne Hansford

January 30, 2005

1 Motivation

Commonly in computer graphics, we design geometry in a special coordinate system, one that makes the construction easy, and then we move this geometry to a more general environment. Font design is a good 2D example of this. A designer will create a character of a font within a unit square (local coordinates), and then this character must be mapped to an arbitrary place and size on the printed page (global coordinates).

2 Local to Global and Back

As illustrated in Figure 1, we have a local coordinate box defined by lower-left corner $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and an upper-right corner $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ that we want to map to the global coordinate (target) box defined by lower-left corner $\mathbf{g}_{min}$ and an upper-right corner $\mathbf{g}_{max}$. Therefore, if we are *given* a point $\mathbf{l}$ in local coordinates, how do we *find* a point $\mathbf{g}$ in global coordinates?

We formulate an expression for the $x-$ and $y-$coordinates separately and

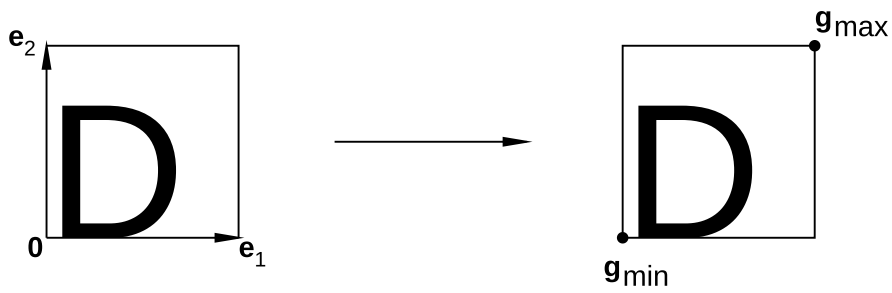

Figure 1: Local and global coordinates: notation

notice that we would like to *preserve ratios*. This leads to

$$\frac{l_x - 0}{1 - 0} = \frac{g_x - g_{minx}}{g_{maxx} - g_{minx}}$$
$$\frac{l_y - 0}{1 - 0} = \frac{g_y - g_{miny}}{g_{maxy} - g_{miny}}$$

Solving for the unknown $\mathbf{g}$ leads to

$$\begin{aligned} g_x &= g_{minx} + l_x(g_{maxx} - g_{minx}) \\ &= g_{minx} + l_x\Delta_x \end{aligned} \tag{1}$$

$$\begin{aligned} g_y &= g_{miny} + l_y(g_{maxy} - g_{miny}) \\ &= g_{miny} + l_y\Delta_y. \end{aligned} \tag{2}$$

We can re-write these equations in the form

$$\begin{aligned} g_x &= (1 - l_x)g_{minx} + l_x g_{maxx} \\ g_y &= (1 - l_y)g_{miny} + l_x g_{maxy}, \end{aligned}$$

which is linear interpolation.

Notice that Δ_x and Δ_y effect the shape of the geometry in the target box.

- If $\Delta_x > 1$ then the geometry is stretched in x.

- If $0 < \Delta_x < 1$ then the geometry is squished in x.
- If $\Delta_x = 1$ then there is no change.
- If $\Delta_x < 0$ then the geometry flips.

The ratio of Δ_x/Δ_y defines the *aspect ratio* of the target box. The aspect ratio of the local system is one; If the target box has an aspect ratio different from one, then the object will be distorted.

If we are given a point in the global system, how do we find a point in the local system? Easy – simply solve (1) and (2) for l_x and l_y, respectively, namely

$$l_x = \frac{g_x - g_{minx}}{\Delta_x}$$
$$l_y = \frac{g_y - g_{miny}}{\Delta_y}$$

3 More General Coordinate Transformations

Often times we will want to map an arbitrary local coordinate box to a global target box. Suppose the local box is defined by the corners $\mathbf{l}_{min}$ and $\mathbf{l}_{max}$, and we wish to map to the global target box with corners $\mathbf{g}_{min}$ and $\mathbf{g}_{max}$, as illustrated in Figure 2.

Given $\mathbf{l}$, find $\mathbf{g}$: Set up the ratios that will be preserved.

$$\frac{l_x - l_{minx}}{l_{maxx} - l_{minx}} = \frac{g_x - g_{minx}}{g_{maxx} - g_{minx}}$$
$$\frac{l_y - l_{miny}}{l_{maxy} - l_{miny}} = \frac{g_y - g_{miny}}{g_{maxy} - g_{miny}}$$

And solve for $\mathbf{g}$:

$$g_x = g_{minx} + [\frac{l_x - l_{minx}}{l_{maxx} - l_{minx}}](g_{maxx} - g_{minx})$$
$$g_y = g_{miny} + [\frac{l_y - l_{miny}}{l_{maxy} - l_{miny}}](g_{maxy} - g_{miny}).$$

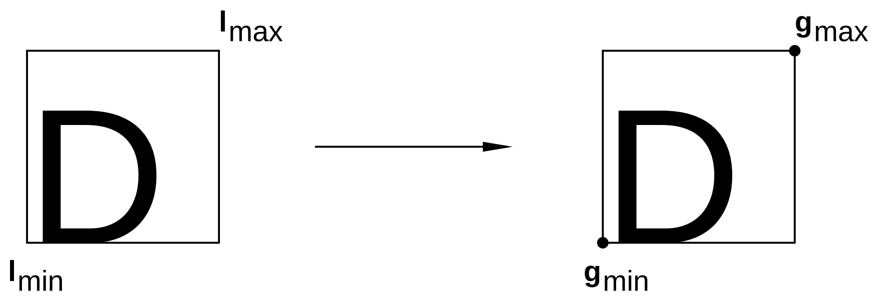

Figure 2: Local and global coordinates: notation

Notice the $[l_{max}, l_{min}] \rightarrow [0, 1]$ mapping embedded in this equation!

Now it is easy to find **l** given **g**. Try it for yourself.

4 World to Window Transformations

It is a common practice to define a *minmax box* around the geometry that you want to display. Another name for this box is a *world coordinate system.* This minmax box is then the input to a command such as OpenGL's gluOrtho2D(), which in turn, maps this minmax box to the window. Let's suppose that our minmax box is a square of width four, centered at the origin. In other words, the lower-left corner is at $\begin{bmatrix} -2 \\ -2 \end{bmatrix}$. The OpenGL command takes the form

$$\text{gluOrtho2D}(-2, 2, -2, 2).$$

As long as the window remains square, the aspect ratio of the world system and the window are the same, and your geometry is not distorted. However, if the window is allowed to be reshaped, and if we want our geometry to be displayed without distortion, then we need to compensate for such changes in aspect ratio.

Suppose the world system has width w_0 and height h_0, and therefore, aspect ratio w_0/h_0. In our example here, the aspect ratio of the world system is one. Suppose the window has been reshaped to have width w_1 and h_1, and further, suppose that $w_1 < h_1$.

We need to stretch the minmax box to match the shape of the window's shape. In other words, we need

$$\frac{w_0}{h_0} = \frac{w_1}{h_1}.$$

We will leave w_0 as it is, and stretch the height:

$$h_0 = w_0 \frac{h_1}{w_1}.$$

Because our minmax box is centered at the origin, we simply center this new height around the origin, and the new OpenGL command becomes

$$\text{gluOrtho2D}(-2, 2, -2\frac{h_1}{w_1}, 2\frac{h_1}{w_1}).$$

Our initial assumption was that $w_1 < h_1$, therefore $h_1/w_1 > 1$ and we have achieved the stretching we needed.

Now determine the gluOrtho() parameters if $w_1 > h_1$ for yourself!

For more information, examples, and exercises, see **The Geometry Toolbox for Graphics and Modeling**.