

On-Line Computer Graphics Notes

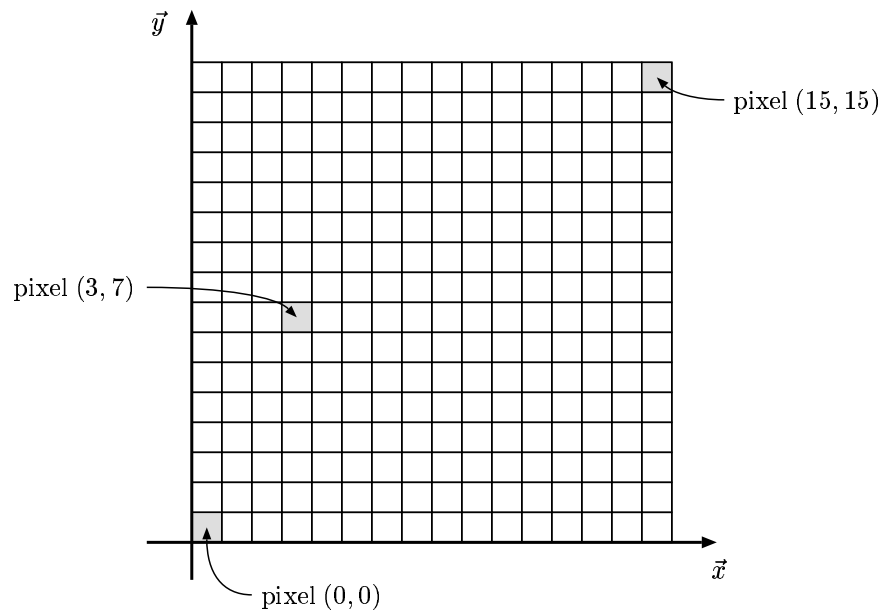
DEVICE SPACE

Kenneth I. Joy
Visualization and Graphics Research Group
Department of Computer Science
University of California, Davis

A model is reduced to a picture by systematic transformation through several coordinate systems. The lowest-level coordinate system included in this reduction is called *Device Space* – and it is closest to the hardware coordinate systems of the device itself. In these notes, we describe the device-space coordinate system.

Device Space

Device space is usually defined to be the $n \times m$ array of pixels that represent the area of the screen. A coordinate system is imposed on this space by labeling the lower-left-hand corner of the array as $(0,0)$, with each pixel having unit length and width. This is illustrated in the figure below.



Three of the pixels are shown in the figure. The lower-left corner of the (i, j) th pixel is the coordinate

(i, j) , the lower-right corner is $(i + 1, j)$, the upper-left corner is $(i, j + 1)$, and the upper-right corner is $(i + 1, j + 1)$.

An arbitrary point (x, y) in image space is converted to coordinates in device space by converting the (x, y) coordinate to

$$\left(\frac{n(x + 1)}{2}, \frac{m(y + 1)}{2}, z \right)$$

where it is assumed that the pixel array is $n \times m$.

To map any device-space point (x, y) into its corresponding pixel is accomplished by taking the floor of each of the coordinates ($\lfloor x \rfloor, \lfloor y \rfloor$). Thus the pixels in device space are all indexed by referring to their lower-left-hand corner.

Device Space with Depth Information

We can extend device space to three dimensions by adding the z coordinate of image space. It is primarily used to obtain the device-space coordinate system for use in the rendering algorithms (e.g., scanline, z -buffer, painters). Device-space coordinates (x, y, z) have bounds $0 \leq x \leq n, 0 \leq y \leq m$, with z arbitrary – where n and m are numbers of columns and rows of the pixel array respectively. The point (x, y, z) in image space is converted to coordinates in extended device space by converting the (x, y) coordinate from as above, and adding the z coordinate. This gives

$$(x, y, z) \longrightarrow \left(\frac{n(x + 1)}{2}, \frac{m(y + 1)}{2}, z \right)$$

where it is assumed that the pixel array is $n \times m$.