

从窗口到视口的映射

高中时我们学过垂直坐标系和极坐标系，同一个点在不同的坐标系里的坐标值是不一样的，比如垂直坐标系里的(1, 1)这个点，对应的极坐标却是(1.414, 45)。这两个坐标值可以通过一定的计算公式相互转换。我们把这里的垂直坐标系称为“逻辑坐标系”，其坐标值只是一个数值，是没有物理单位的，每一个逻辑单位可以代表 1mm，也可以代表 1m，还可以代表 1 个像素或者 1/12 英寸。这个跟具体的物理设备有关。

这里我们再定义一个新的坐标系，其 x 轴方向、y 轴方向、x 轴刻度和 y 轴刻度都和逻辑坐标系相同。我们把这个坐标系称为“设备坐标系”。如图 1 所示：

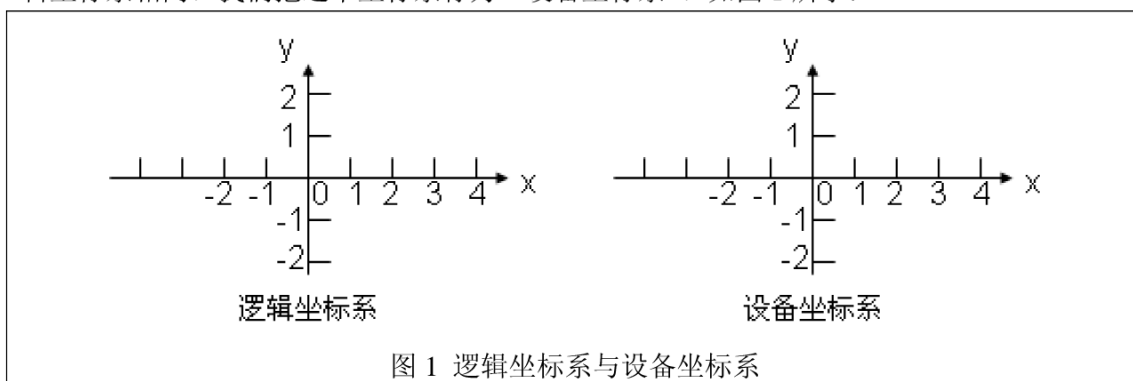


图 1 逻辑坐标系与设备坐标系

这样，同一个坐标值在不同的设备坐标系里，可以对应不同的点，它们之间存在一种映射关系。比如逻辑坐标系中的 A 点(1,1)，可以映射到设备坐标系的 A'(1,1)。

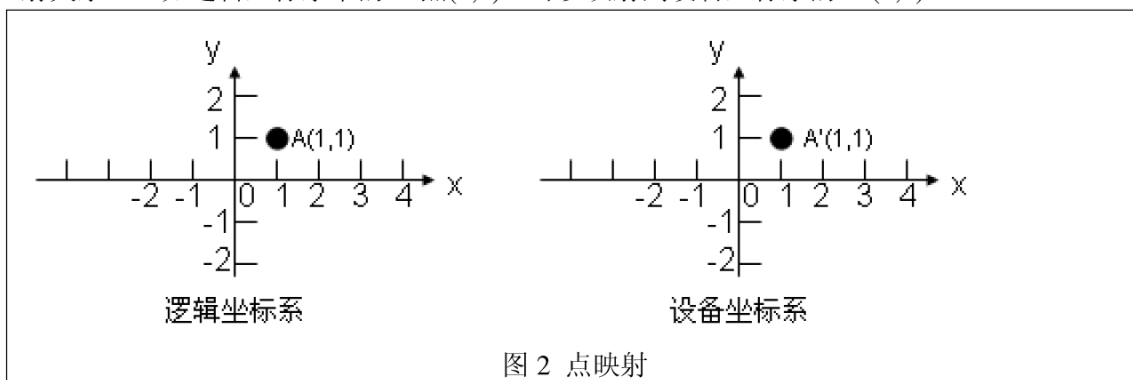


图 2 点映射

同理，逻辑坐标系里的一个矩形 ABCD 也可以映射到设备坐标系里的矩形 A'B'C'D'。

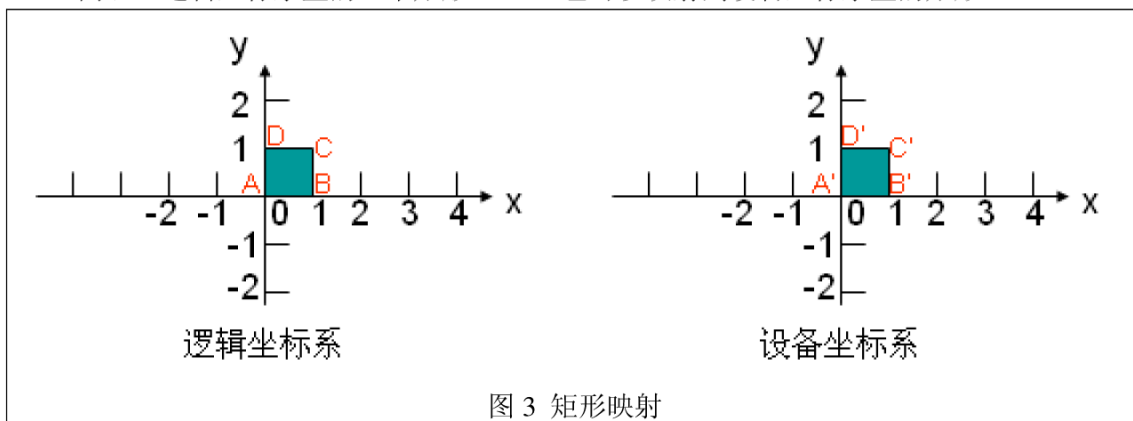
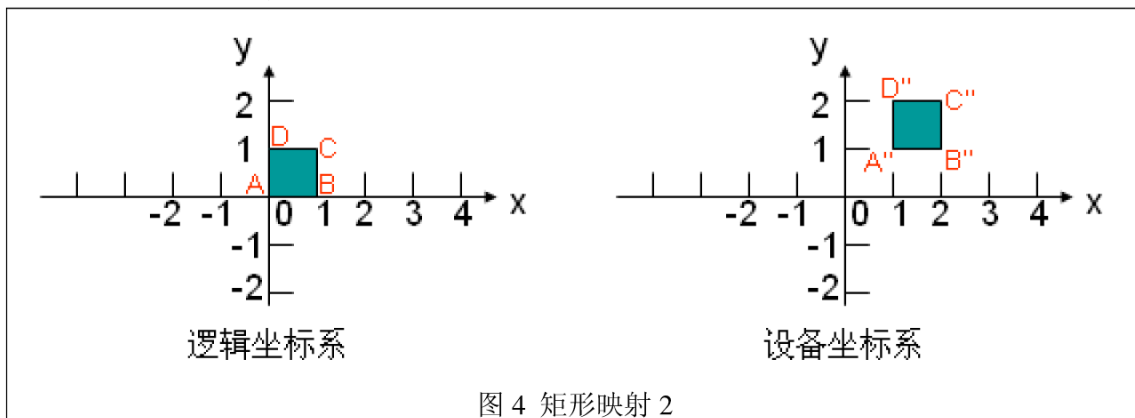


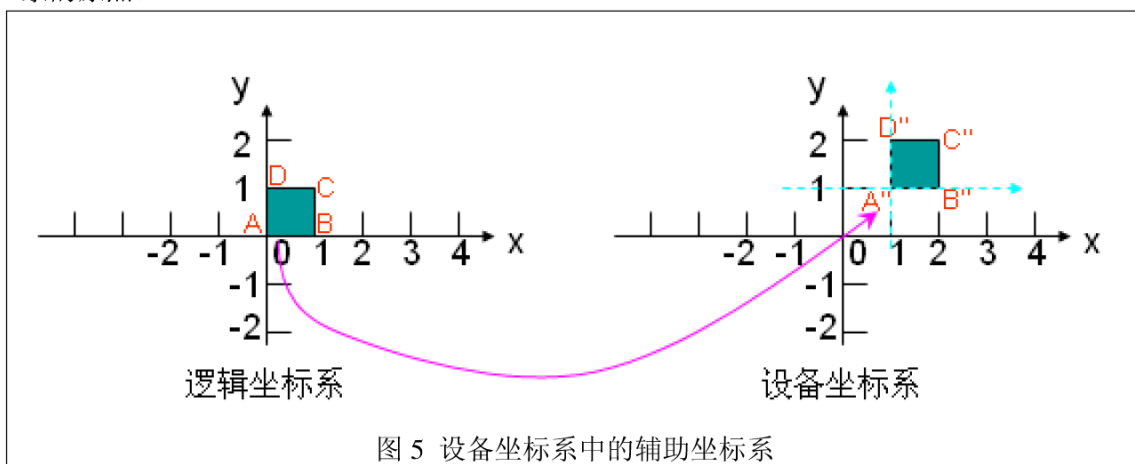
图 3 矩形映射

这种映射都很简单。现在，我想把逻辑坐标系里的矩形 ABCD 映射到设备坐标系里的

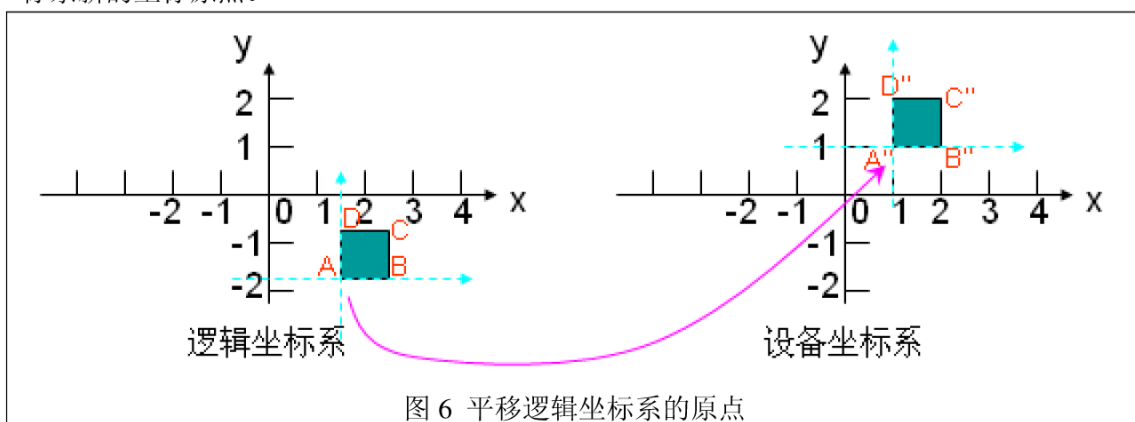
A”B”C”D”，如图 4 所示，可不可以呢？



当然可以，也很简单，只需要把设备坐标系里的映射图形平移一下就可以了。怎么平移呢？只需要把设备坐标系里的原点平移到现在的 A”位置。现在，我们以 A”为原点，再作一个辅助坐标系，如图 5 所示。这时，逻辑坐标系里的原点就映射到设备坐标系里的辅助坐标系的原点。



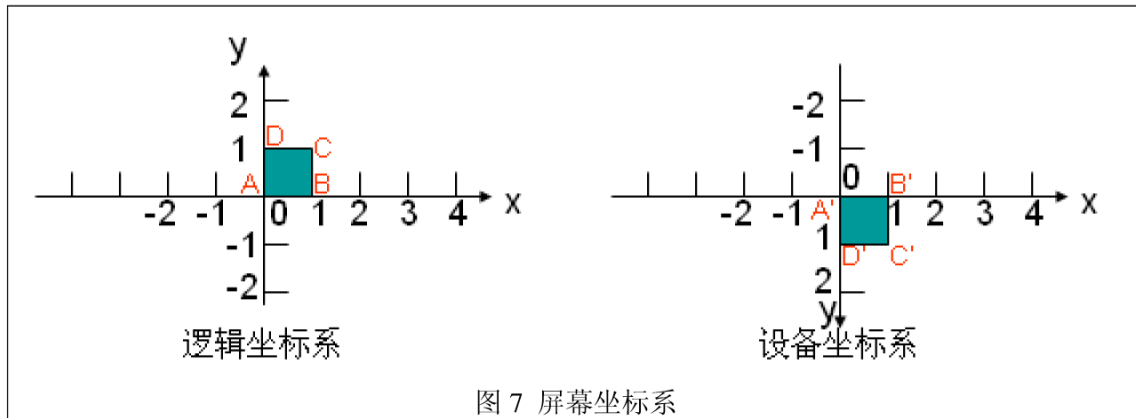
同理，我们也可以平移逻辑坐标系里的原点，在逻辑坐标系里也作一个辅助坐标系。平移之后的映射效果如图 6 所示。可见，在平移之后，逻辑坐标系新的坐标原点映射到设备坐标系新的坐标原点。



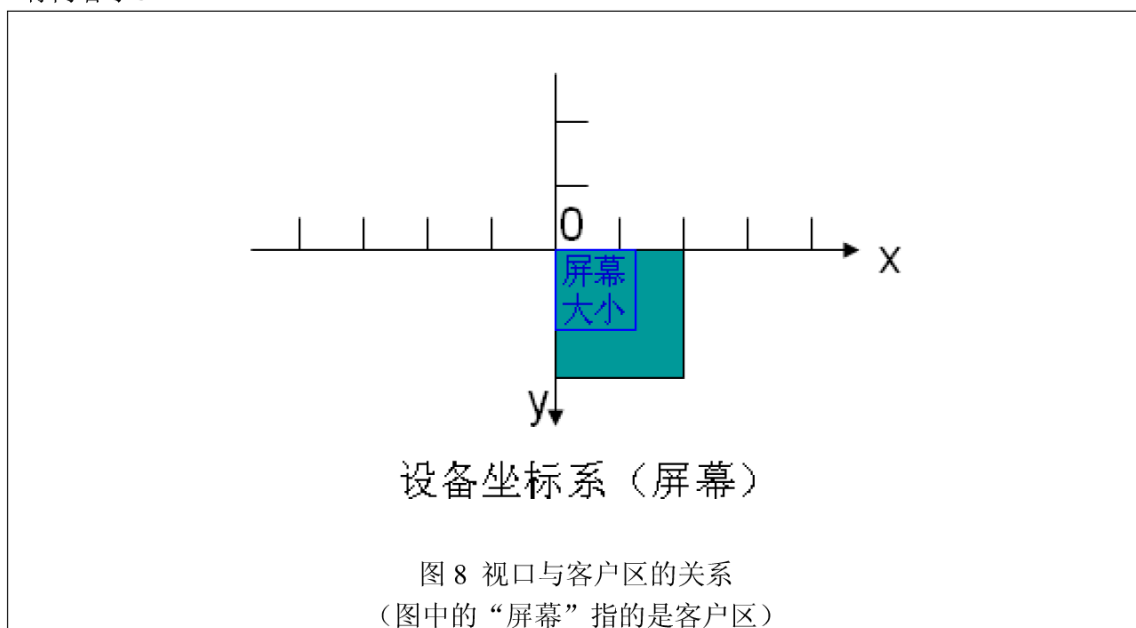
现在，我们把逻辑坐标系里的辅助坐标系称为“窗口”，把设备坐标系里的辅助坐标系称为“视口”。把平移逻辑坐标系原点的操作称为 `SetWindowOrg()`，把平移设备坐标系原点的操作称为 `SetViewportOrg()`。这两个函数的参数都是以各自原来的坐标系为基准的。比如，

图中平移窗口原点应该调用 `SetWindowOrg (1.5, - 1.8)`，而平移视口原点应该是 `SetViewportOrg (1, 1)`。平移之后，原坐标系的零点并不变，改变的只是窗口或者视口的原点（相当于辅助坐标系的零点）。

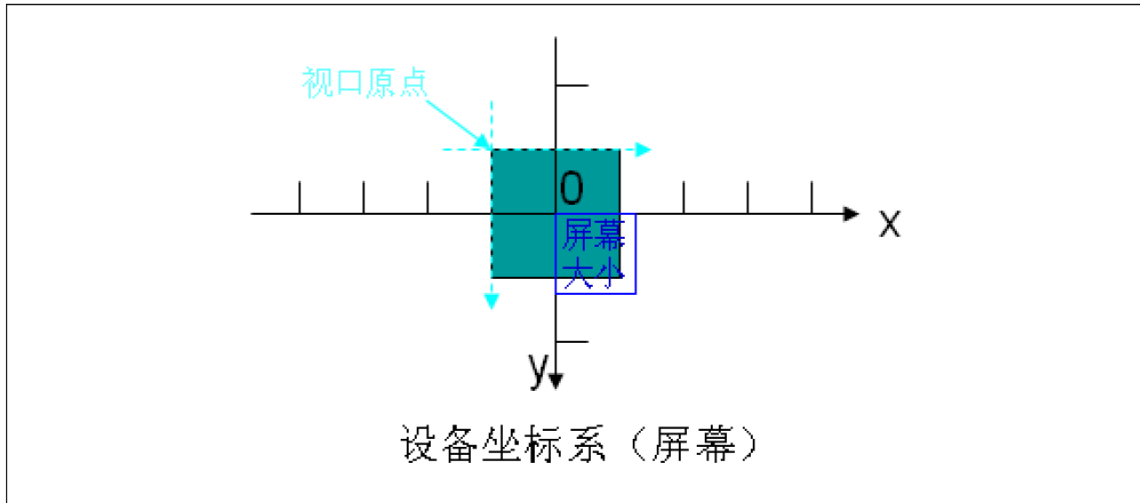
所有的映射都是从窗口映射到视口。上述的映射方式很简单。下面介绍一种新的坐标系，其 x 轴方向依然向右，但 y 轴方向却是向下，每一个逻辑单位代表一个像素。这就是“屏幕坐标系”，我们把从逻辑坐标系映射到屏幕坐标系的映射方式称为“MM_TEXT”。



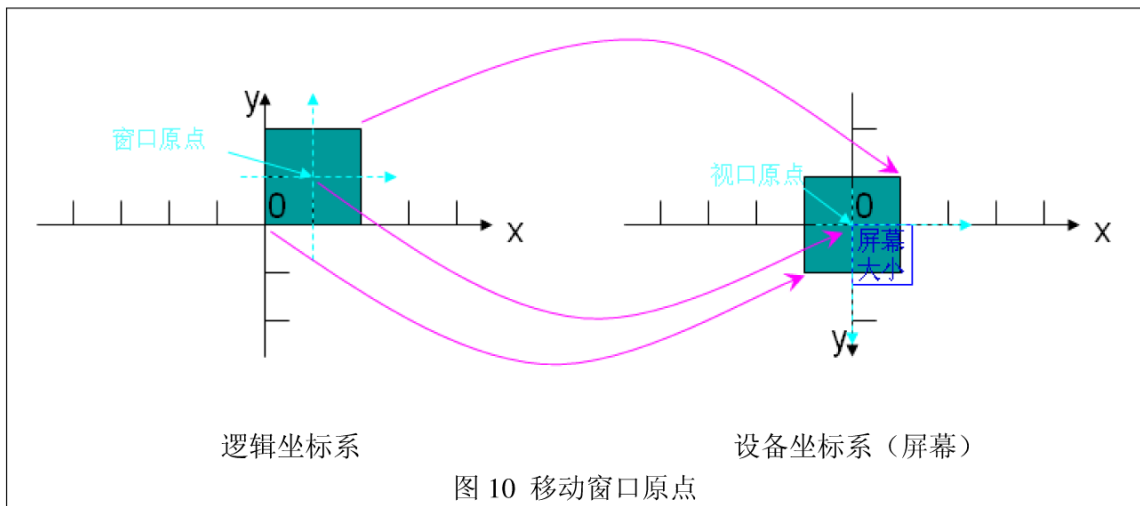
我们知道，逻辑坐标系是无穷大的，想在哪儿绘图就可以在哪儿绘图，但是显示设备（比如屏幕，在应用程序中是客户区）大小却是有限的。现在，我要在逻辑坐标系里画一个 2000×2000 大小的矩形，映射到屏幕上就是一个 2000 像素 \times 2000 像素的一个矩形。但是，应用程序的客户区并没有那么大，只能显示矩形的一部分。显示哪一部分呢？在 Windows 程序中，客户区的左上角始终对应设备坐标系的零点，显示区域是第一象限，这个是不能改变的。这时，客户区只能显示矩形的左上角部分。怎样才能显示矩形的其他部分呢？只需要把矩形移动一下。这就好比用放大镜看报纸，放大镜不动，只移动报纸就可以看到报纸上的所有内容了。



那怎么样移动矩形呢？移动视口的原点就可以了。默认情况下，视口原点和设备坐标系的原点是重合的，把视口的原点移到 $(-1000, -1000)$ 的位置，就可以看到矩形的右下角部分了。



当然，移动窗口的原点也是可以的。因为在逻辑坐标系里作图时，图形的逻辑位置不变，当窗口的原点移动时，图形与窗口原点的相对位置也就发生了改变，映射后视口中的图形与视口原点的相对位置也会相应的发生改变。如图 10，把窗口原点移动到(1000,1000)后，矩形的中心会映射到视口的零点，即客户区的左上角。



刚才我们在逻辑坐标系里画了一个 2000*2000 大小的矩形，显示在屏幕上的矩形长 2000 像素，宽也是 2000 像素。这样，在不同大小的显示器上，由于像素间距不同，所以实际大小也不同。并且，有的显示器像素的水平间距和垂直间距也不同，这时画的本来是个正方形，但显示出来却变成了长方形！这时，可以使用 `MM_LOMETRIC` 映射模式，保证所画的每一个逻辑单位长度对应 0.1mm。在不同的显示器上显示时，它会根据自己的点距来调整。