图像几何变换不改变图像的像素值，在图像平面上进行像素变换。适当的几何变换可以最大程度地消除由于成像角度、透视关系乃至镜头自身原因所造成的几何失真所产生的负面影响。几何变换常常作为图像处理应用的预处理步骤，是

图像归一化的核心工作之一。

一个几何变换需要两部分运算：

 空间变换：包括平移、缩放、旋转和正平行投影等，需要用它来表示输出图像与输入图像之间的像素映射关系。

 灰度插值算法：按照这种变换关系进行计算，输出图像的像素可能被映射到输入图像的非整数坐标上。

图像几何变换在变换过程中会建立一种原图像像素与变换后图像像素之间的映射关系，通过这种关系，能够从一方的像素计算出另一方的像素的坐标位置。

通常将图像坐标映射到输出的过程称作向前映射，反之，将输出图像映射到输入的过程称作向后映射。

向后映射在实践中使用较多，原因是能够避免使用向前映

射中出现映射不完全和映射重叠的问题。

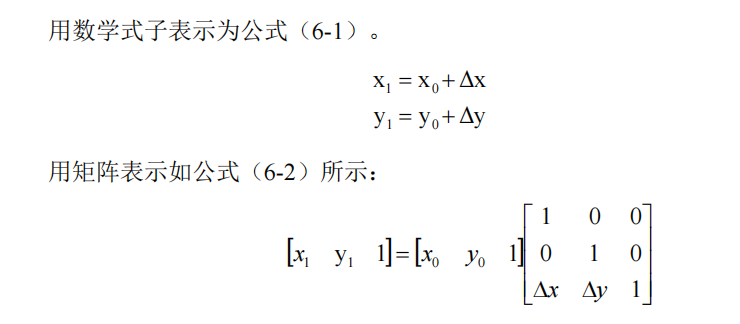
对于数字图像而言，像素的坐标是离散型非负整数，但是在进行变换的过程中有可能产生浮点坐标值。这在图像处理中是一个无效的坐标。为了解决这个问题需要用到插值算法。常见算法如下：

 最近邻插值

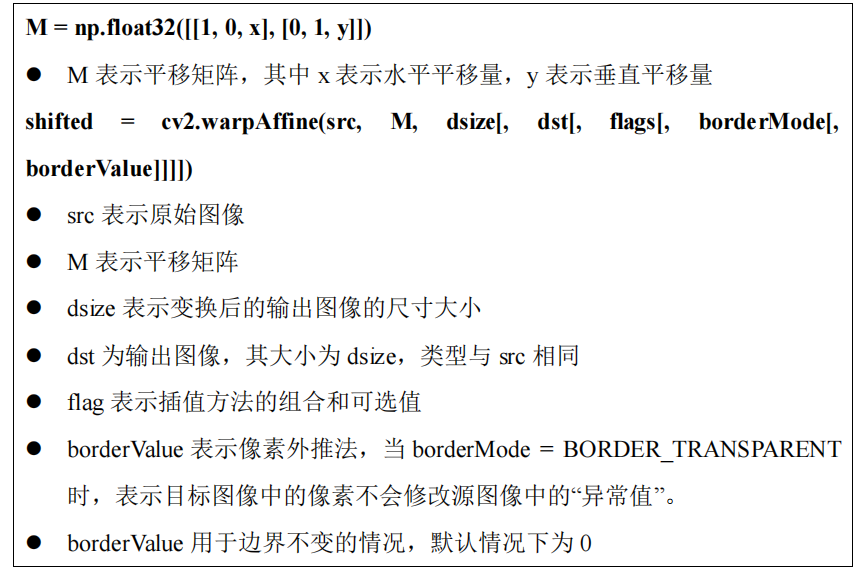
 双线性插值

 双立方插值

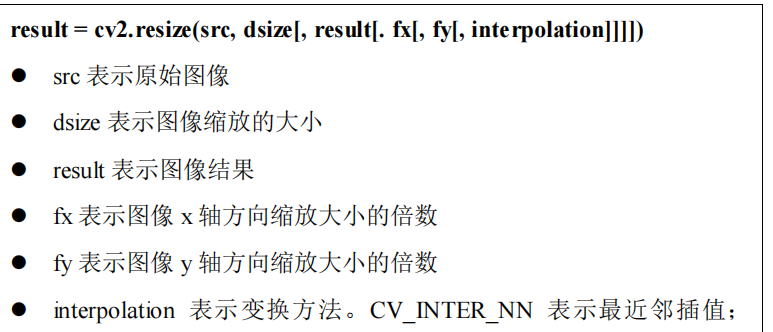
图像平移是将图像中的所有像素点按照给定的平移量进行水平或垂直方向上的移动。假设原始像素的位置坐标为（x0，y0），经过平移量（△ x，△ y）后，坐标变为（x1, y1）

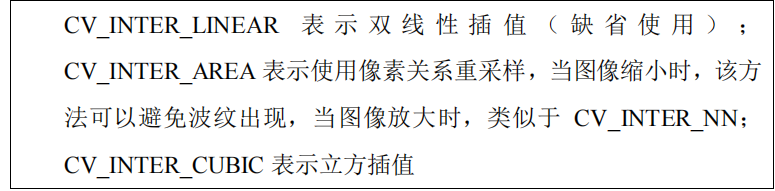


矩阵称为平移变换矩阵或因子，△ x 和△ y 称为平移量。图像平移，首先定义平移矩阵 M，再调用 warpAffine()函数实现平移



图像缩放（image scaling）是指对数字图像的大小进行调整的过程。在 Python中，图像缩放主要调用 resize()函数实现



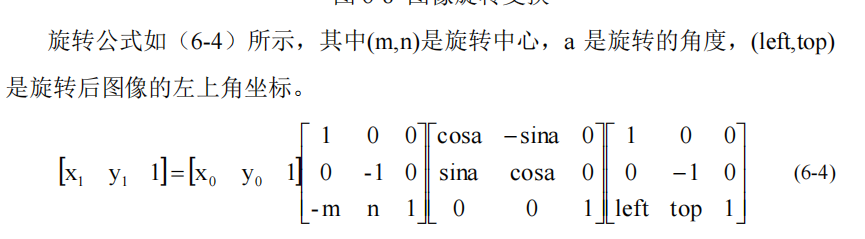


常见的图像缩放两种方式如下所示，第一种方式是将原图像设置为(180, 160) 像素大小，第二种方式是将原始图像缩小为 0.5 倍。

 result = cv2.resize(src, (160,180)) #(col,row)

 result = cv2.resize(src, None, fx=0.5, fy=0.5)

图像旋转是指图像以某一点为中心旋转一定的角度，形成一幅新的图像的过 程。图像旋转变换会有一个旋转中心，这个旋转中心一般为图像的中心，旋转之 后图像的大小一般会发生改变。



图像旋转变换主要调用 getRotationMatrix2D()函数和 warpAffine()函数实现，绕图像的中心旋转

