数字图像处理

代数运算与几何变换

代数运算

- 代数运算定义
  - 代数运算是指两幅或多幅输入图像进行点对点的加、减、乘或除计算而得到输出图像的运算。

代数运算

- 代数运算用途
  - 减法运算可以用来减去背景，运动检测，进行梯度幅度运算
  - 加法运算可以用来降低图像中的随机噪音（前提是图像中的其他部分必须是不动的）。加法运算可以达到二次或多次曝光的效果。
  - 乘法运算通常用来进行掩模运算
  - 除法运算可以用来归一化

乘法和除法用的比较少，但在某些应用上很重要

减法

- 减法的表示
  - $g(x,y) = f(x,y) - h(x,y)$
  - 通过计算两幅图像所有对应像素点的差来表示两幅图像的差异

高阶位平面带有图像大量的可见信息

低阶位平面分布着图像一些细节信息

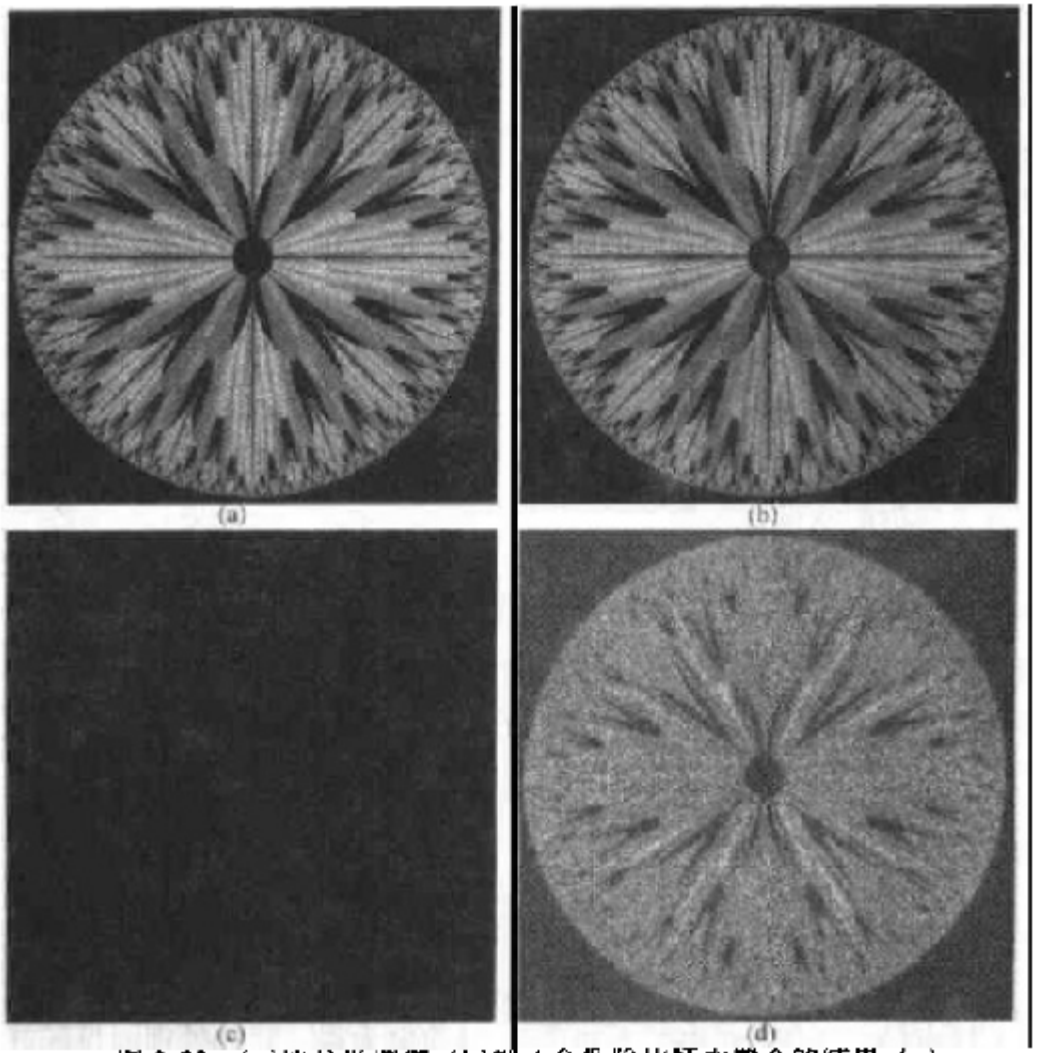


图 3.28 (a)原分形图像,(b)把 4 个低阶比特面置 0 的结果,(c)(a)和(b)间的差别,(d)直方图均衡后的差值图像(原图像由 Swarthmore 学院的 Melissa D. Binde 先生提供)

=

-

用减法处理来移去图像中静止的部分。

加法

- 噪声分类
  - 加性噪声：加性噪声和图像信号强度不相关。

$$g(x,y) = f(x,y) + n(x,y)$$

  - 乘性噪声：乘性噪声和图像信号是相关的。

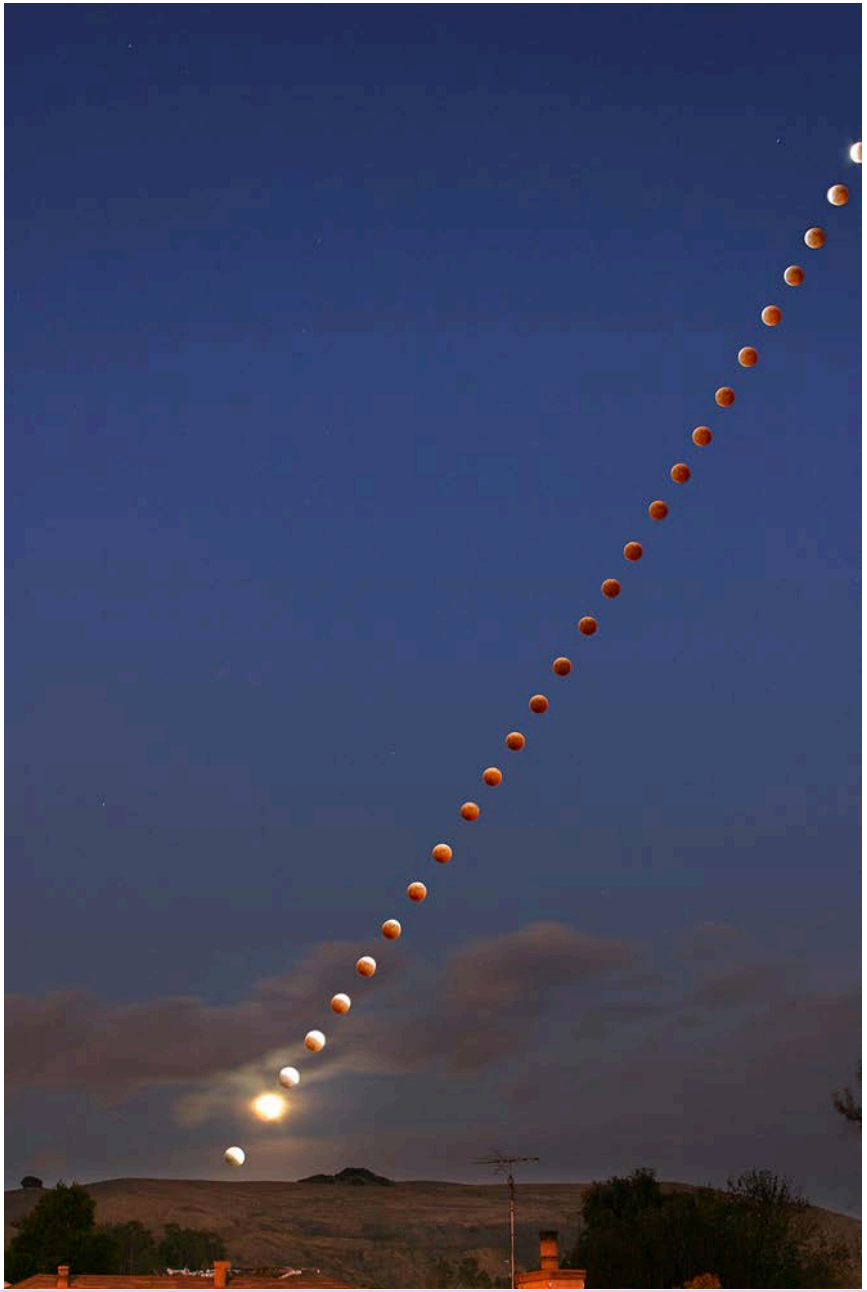
$$g(x,y) = f(x,y) + f(x,y) \times n(x,y)$$

  - 椒盐(Salt and Pepper)噪声：黑图像上的白点，白图像上的黑点。
  - 量化噪声：是由量化过程引起的，解决的最好方法是最佳量化。

+

=

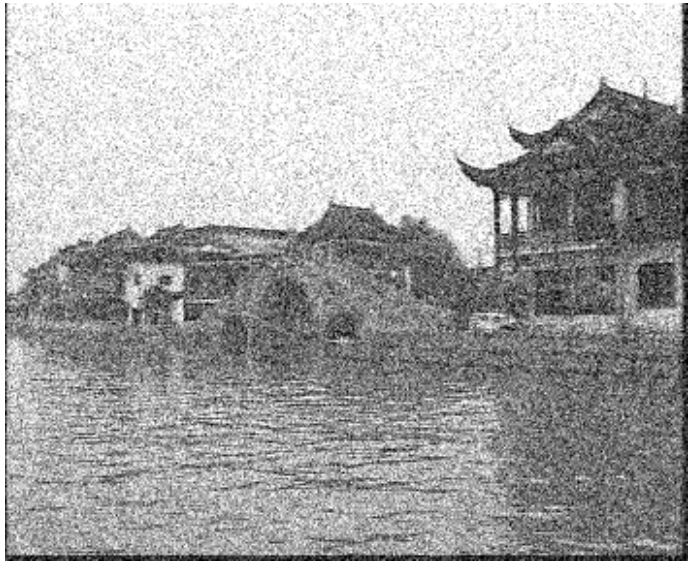
多次曝光



加法

- 例子

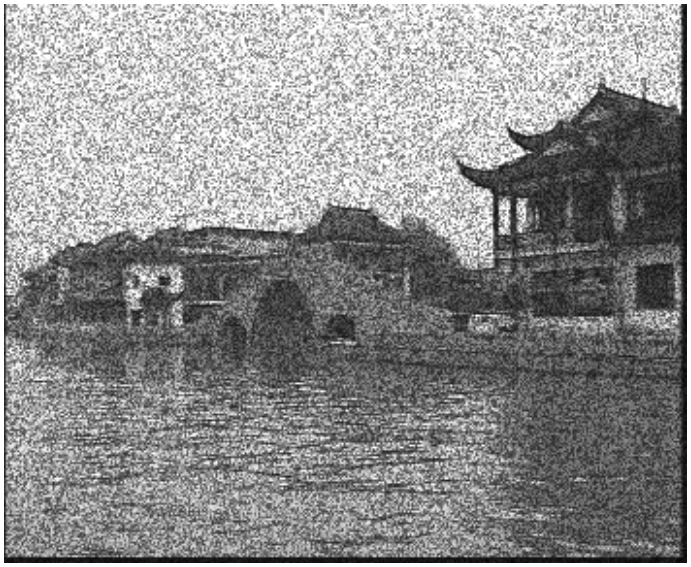
上海朱家角风光



有加性噪声的朱家角风光

加法

- 例子



有乘性噪声的朱家角风光

有椒盐噪声的朱家角风光

加法

- 例子

原图像

椒盐噪声

高斯噪声

加法运算可以用来降低图像中的随机加性噪音

加法应用

- Lenna的8个随机加性噪音图片

噪声图像1

噪声图像2

噪声图像3

噪声图像4

噪声图像5

噪声图像6

噪声图像7

噪声图像8

加法应用

- 降噪后的图像

原始图像

降噪后图像

几何变换

- 图像的几何变换是指使用户将原始图像按照需要产生大小、形状和位置变化的图像。
- 图像的几何变换有二维变换和三维变换以及由三维向二维平面投影变换等。
- 从变换的性质分， 图像的几何变换有位置(平移、镜像、旋转）、形状变换（比例缩放）和复合变换等。
- 从易到难，主要工具： 线性代数。

平移变换

- 图像平移变换

$$\begin{cases} x = x_0 + \Delta x \\ y = y_0 + \Delta y \end{cases} \qquad \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

$$\begin{cases} x_0 = x - \Delta x \\ y_0 = y - \Delta y \end{cases} \qquad \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta x \\ 0 & 1 & -\Delta y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

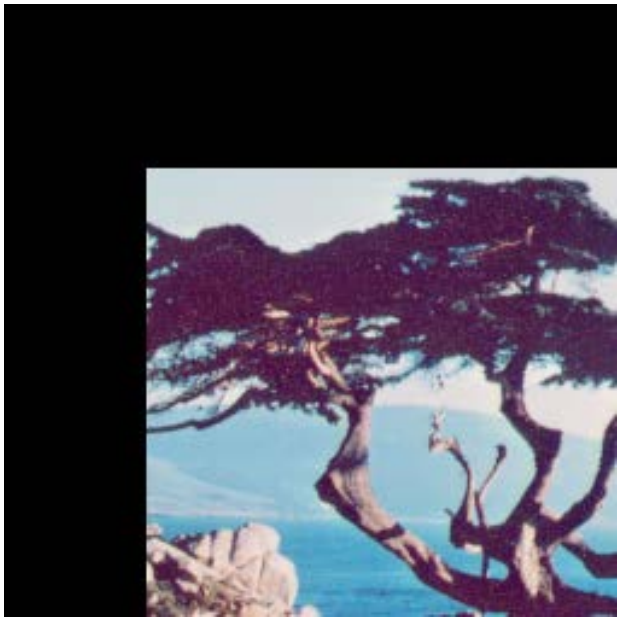
平移变换

- 平移后图像上的每一点都可以在原图像中找到对应的点。对于不在原图像中的点，可以直接将它的像素值统一设置为0或者255（对于灰度图就是黑色或白色）。

$$F=\begin{bmatrix} f_{11} & f_{12} & \cdots f_{1n-1} & f_{1n} \\ f_{21} & f_{22} & \cdots f_{2n-1} & f_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ f_{n1} & f_{n2} & \cdots f_{nn-1} & f_{nn} \end{bmatrix} \qquad G=\begin{bmatrix} 0 & 0 & 0 \cdots & 0 \\ 0 & f_{11} & f_{12} & \cdots f_{1n-1} \\ 0 & f_{21} & f_{22} & \cdots f_{2n-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & f_{n1} & f_{n2} & \cdots f_{nn-1} \end{bmatrix}$$

$$H=\begin{bmatrix} 255 & 255 & 255 & \cdots & 255 \\ 255 & f_{11} & f_{12} & \cdots f_{1n-1} \\ 255 & f_{21} & f_{22} & \cdots f_{2n-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 255 & f_{n1} & f_{n2} & \cdots f_{nn-1} \end{bmatrix}$$





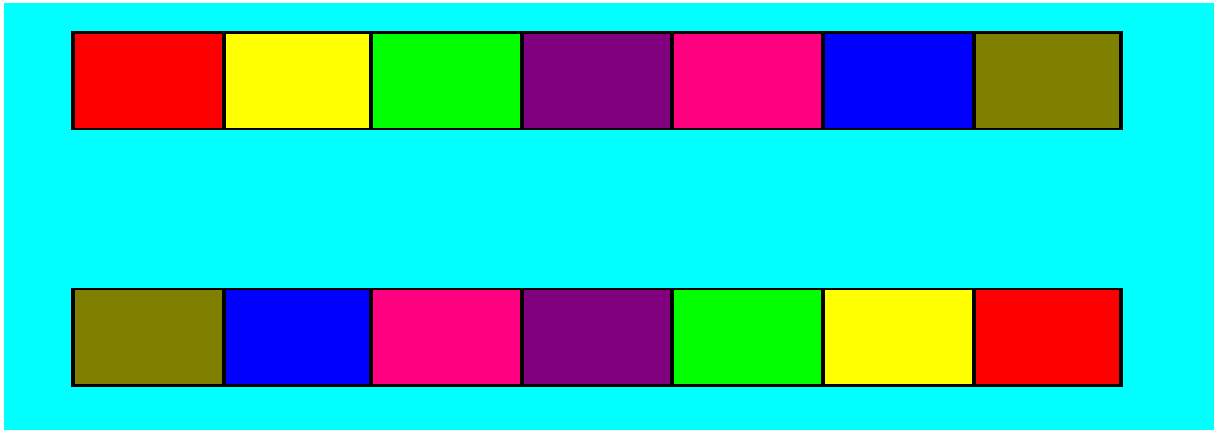
镜像变换

- 图像的镜像变换不改变图像的形状。图像的镜像变换分为三种：水平镜像，垂直镜像和对角镜像

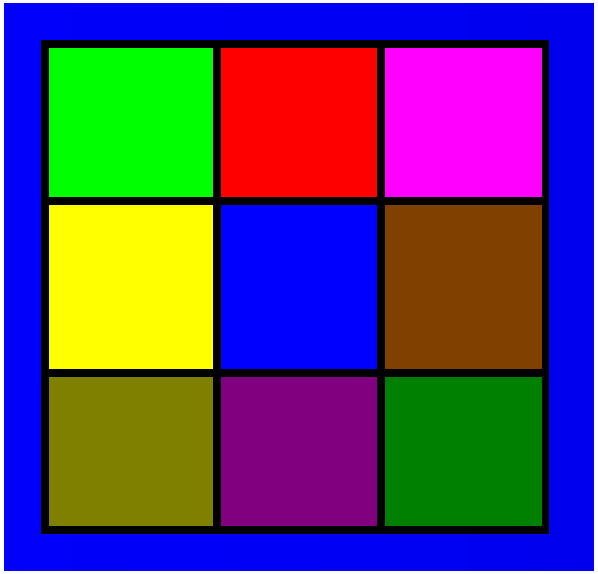
镜像变换

- 图像水平镜像
  - 图像的水平镜像操作是将图像左半部分和右半部分以图像垂直中轴线为中心进行镜像对换。
  - 设图像的大小为$M \times N$，水平镜像可按式计算

$$\begin{cases} i' = i \\ j' = N - j + 1 \end{cases}$$



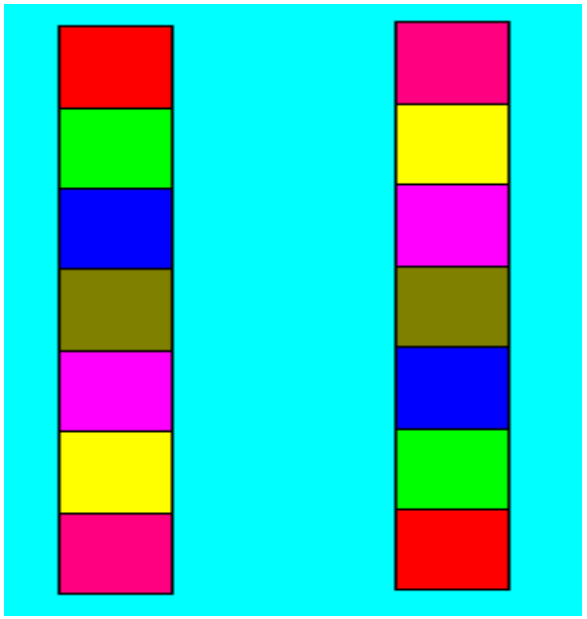
$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} & f_{15} \\ f_{21} & f_{22} & f_{23} & f_{24} & f_{25} \\ f_{31} & f_{32} & f_{33} & f_{34} & f_{35} \\ f_{41} & f_{42} & f_{43} & f_{44} & f_{45} \\ f_{51} & f_{52} & f_{53} & f_{54} & f_{55} \end{bmatrix} \quad F = \begin{bmatrix} f_{15} & f_{14} & f_{13} & f_{12} & f_{11} \\ f_{25} & f_{24} & f_{23} & f_{22} & f_{21} \\ f_{35} & f_{34} & f_{33} & f_{32} & f_{31} \\ f_{45} & f_{44} & f_{43} & f_{42} & f_{41} \\ f_{55} & f_{54} & f_{53} & f_{52} & f_{51} \end{bmatrix}$$



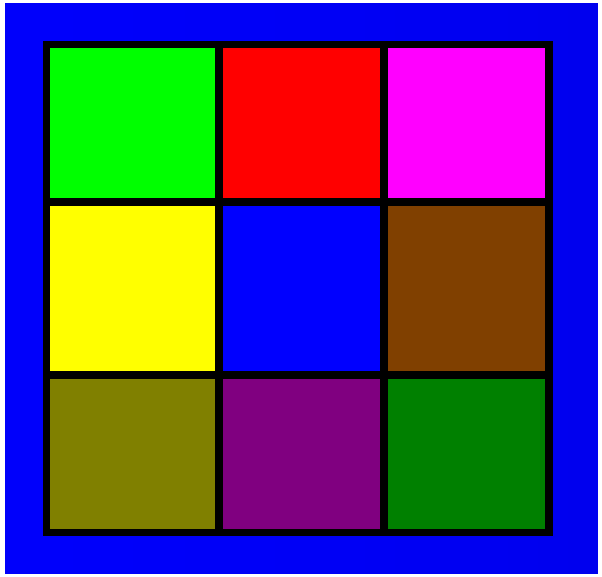
镜像变换

- 图像垂直镜像
  - 图像的垂直镜像操作是将图像上半部分和下半部分以图像水平中轴线为中心进行镜像对换。
  - 设图像的大小为$M \times N$，垂直镜象可按式计算

$$\begin{cases} i' = M - i + 1 \\ j' = j \end{cases}$$



$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} & f_{15} \\ f_{21} & f_{22} & f_{23} & f_{24} & f_{25} \\ f_{31} & f_{32} & f_{33} & f_{34} & f_{35} \\ f_{41} & f_{42} & f_{43} & f_{44} & f_{45} \\ f_{51} & f_{52} & f_{53} & f_{54} & f_{55} \end{bmatrix}$$



$$H = \begin{bmatrix} f_{51} & f_{52} & f_{53} & f_{54} & f_{55} \\ f_{41} & f_{42} & f_{43} & f_{44} & f_{45} \\ f_{31} & f_{32} & f_{33} & f_{34} & f_{35} \\ f_{21} & f_{22} & f_{23} & f_{24} & f_{25} \\ f_{11} & f_{12} & f_{13} & f_{14} & f_{15} \end{bmatrix}$$

镜像变换

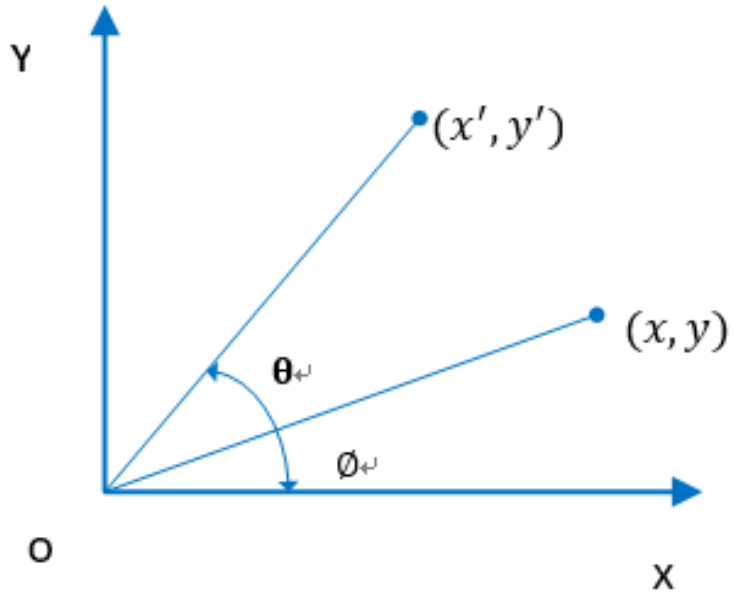
- 图像对角镜像
  - 图像的对角镜像操作是将图像以图像水平中轴线和垂直中轴线的交点为中心进行镜像对换。
  - 相当于将图像先后进行水平镜像和垂直镜像。
  - 设图像的大小为$M \times N$，对角镜像可按式计算

$$\begin{cases} i' = M - i + 1 \\ j' = N - j + 1 \end{cases}$$

$$F=\begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} & f_{15} \\ f_{21} & f_{22} & f_{23} & f_{24} & f_{25} \\ f_{31} & f_{32} & f_{33} & f_{34} & f_{35} \\ f_{41} & f_{42} & f_{43} & f_{44} & f_{45} \\ f_{51} & f_{52} & f_{53} & f_{54} & f_{55} \end{bmatrix} \quad H=\begin{bmatrix} f_{55} & f_{54} & f_{53} & f_{52} & f_{51} \\ f_{45} & f_{44} & f_{43} & f_{42} & f_{41} \\ f_{35} & f_{34} & f_{33} & f_{32} & f_{31} \\ f_{25} & f_{24} & f_{23} & f_{22} & f_{21} \\ f_{15} & f_{14} & f_{13} & f_{12} & f_{11} \end{bmatrix}$$

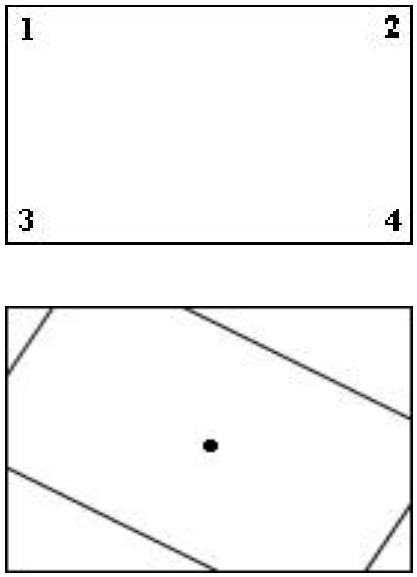
旋转变换

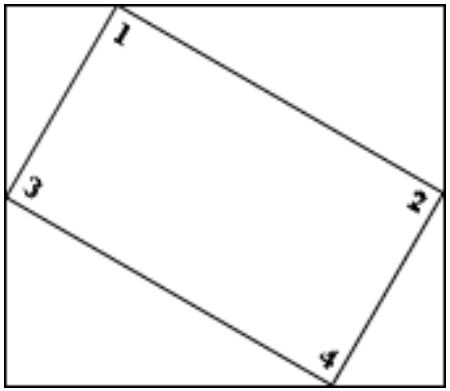
- 图像旋转变换
  - 旋转(rotation)中心：通常的做法是以图像的中心为圆心旋转，将图像上的所有像素都旋转一个相同的角度。



旋转变换

- 图像旋转变换
  - 图像的旋转变换是图像的位置变换，但旋转后，图像的大小一般会改变。和图像平移一样， 在图像旋转变换中既可以把转出显示区域的图像截去，旋转后也可以扩大图像范围以显示所有的图像。





旋转变换

- 设旋转前(x_0, y_0)的坐标分别为$x_0 = r\cos b$；$y_0 = r\sin b$，当旋转a角度后：旋转后的坐标(x_1, y_1)的坐标分别为

$$\begin{cases} x_1 = r\cos(b-a) = r\cos b\ \cos a + r\sin b\sin a = x_0\cos a + y_0\sin a \\ y_1 = r\sin(b-a) = r\sin b\ \cos a - r\cos b\sin a = -x_0\sin a + y_0\cos a \end{cases}$$

写成矩阵表达式为

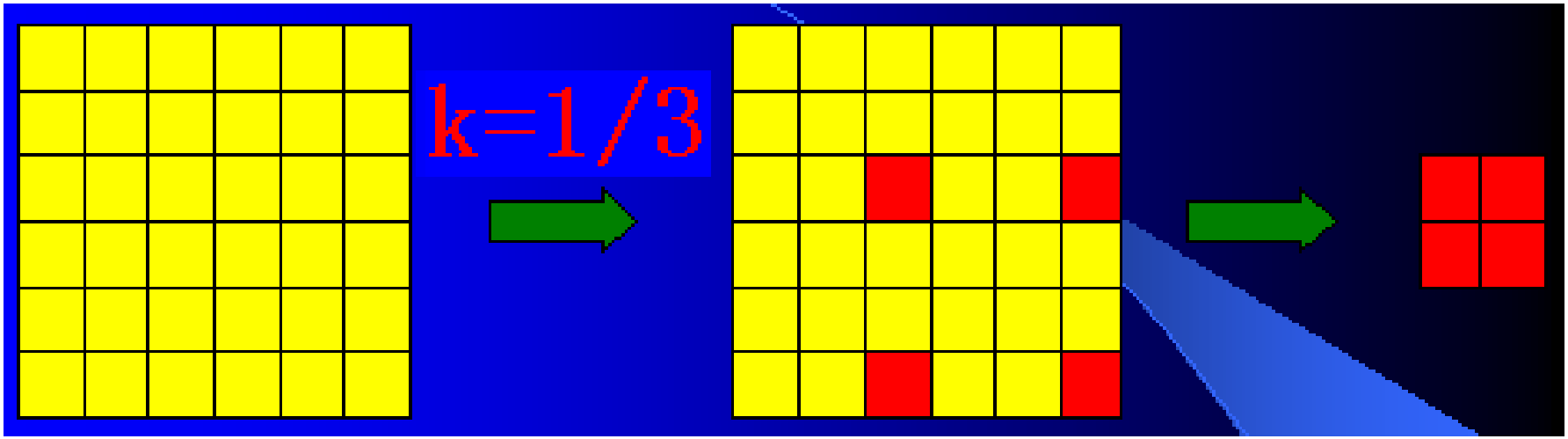
$$[x_1\ y_1\ 1] = [x_0\ y_0\ 1]\begin{bmatrix} \cos a & -\sin a & 0 \\ \sin a & \cos a & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

比例缩放变换

- 图像比例缩放变换
  - 图像比例缩放是指将给定的图像在x轴方向按比例缩放f_x倍， 在y轴方向按比例缩放f_y倍，从而获得一幅新的图像。

$$\begin{cases} X = f_x \cdot X_0 \\ Y = f_y \cdot Y_0 \end{cases}$$

比例伸缩变换



比例伸缩变换

- 图像的比例缩小变换
  - 从数码技术的角度来说，图像的缩小是将通过减少像素个数来实现的。
  - 因此，需要根据所期望缩小的尺寸数据，从原图像中选择合适的像素点，使图像缩小之后可以尽可能保持原有图像的概貌特征不丢失。

比例缩小变换

- 基于等间隔采样的图像缩小方法
  - 这种图像缩小方法的设计思想是，通过对画面像素的均匀采样来保持所选择到的像素仍旧可以保持像素的概貌特征。

比例缩小变换

- 基于等间隔采样的图像缩小方法
  - 该方法的具体实现步骤为：
    - 设原图为$F(i,j)$，大小为：$M \times N$ $(i = 1,2,\dots,M;\ j =$

$$\Delta i = 1/k_1, \quad \Delta j = 1/k_2$$

$$g(i,j) = f(\Delta i \cdot i, \Delta j \cdot j)$$

等间隔采样

- 设原图像为

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} & f_{15} & f_{16} \\ f_{21} & f_{22} & f_{23} & f_{24} & f_{25} & f_{26} \\ f_{31} & f_{32} & f_{33} & f_{34} & f_{35} & f_{36} \\ f_{41} & f_{42} & f_{43} & f_{44} & f_{45} & f_{46} \end{bmatrix}$$

等间隔采样

- 图像矩阵的大小为4×6，将其进行缩小，缩小的倍数为$k_1 = 0.7$，$k_2 = 0.6$，则缩小图像的大小为3×4，计算得

$$\Delta i = 1/k_1 = 1.4 \quad ,\Delta j = 1/k_2 = 1.7$$

$$G = \begin{bmatrix} f_{12} & f_{13} & f_{15} & f_{16} \\ f_{32} & f_{33} & f_{35} & f_{36} \\ f_{42} & f_{43} & f_{45} & f_{46} \end{bmatrix}$$

局部均值缩小变换

- 基于局部均值的图像缩小方法
  - 从前面的缩小算法可以看到，算法的实现非常简单，但是采用上面的方法对没有被选取到的点的信息就无法反映在缩小后的图像中。为了解决这个问题，可以采用基于局部均值的方法来实现图像的缩小。

利用$g(i，j)=F(i，j)$的均值,求出缩小的图像

局部均值缩小变换

- 基于局部均值的图像缩小方法
  - 该方法的具体实现步骤如下：
    - 计算采样间隔，得到：

$$\Delta i = 1/k_1 \quad ,\Delta j = 1/k_2$$

    - 求出采样点所包含的原图像的子块：

$$F^{(i,j)} = \begin{bmatrix} f_{\Delta i\cdot(i-1)+1,\Delta j\cdot(j-1)+1} & \cdots & f_{\Delta i\cdot(i-1)+1,\Delta j\cdot j} \\ \vdots & & \vdots \\ f_{\Delta i\cdot i,\Delta j\cdot(j-1)+1} & \cdots & f_{\Delta i\cdot i,\Delta j\cdot j} \end{bmatrix}$$

    - 利用$g(i,j) = F(i,j)$的均值，求出缩小的图像

局部均值缩小变换

- 设原图像为

$$F=\begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} & f_{15} & f_{16} \\ f_{21} & f_{22} & f_{23} & f_{24} & f_{25} & f_{26} \\ f_{31} & f_{32} & f_{33} & f_{34} & f_{35} & f_{36} \\ f_{41} & f_{42} & f_{43} & f_{44} & f_{45} & f_{46} \end{bmatrix}$$

局部均值缩小变换

- 大小为4×6，将其进行缩小，缩小的倍数为$k_1 = 0.7$，$k_2 = 0.6$，则缩小图像的大小为3×4，计算得

$$\Delta i = 1/k_1 = 1.4 \quad ,\Delta j = 1/k_2 = 1.7$$

- 将图像F分块为

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} & f_{15} & f_{16} \\ f_{21} & f_{22} & f_{23} & f_{24} & f_{25} & f_{26} \\ f_{31} & f_{32} & f_{33} & f_{34} & f_{35} & f_{36} \\ f_{41} & f_{42} & f_{43} & f_{44} & f_{45} & f_{46} \end{bmatrix}$$

局部均值缩小变换

- 计算局部（块内）均值

$$G=\begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \end{bmatrix}$$

$$g_{21}=\frac{1}{4}\times(f_{21}+f_{22}+f_{31}+f_{32})$$

缩小变换

- 若图像为

$$F=\begin{bmatrix}31 & 35 & 39 & 13 & 17 & 21\\ 32 & 36 & 10 & 14 & 18 & 22\\ 33 & 37 & 11 & 15 & 19 & 23\\ 34 & 38 & 12 & 16 & 20 & 24\end{bmatrix}$$

按照上例缩小的比例，采用等间隔采样和采用局部均值采样得到的缩小图像分别为

$$G=\begin{bmatrix}35 & 39 & 17 & 21\\ 37 & 11 & 19 & 23\\ 38 & 12 & 20 & 24\end{bmatrix}$$

等间隔采样

$$G=\begin{bmatrix}33 & 39 & 15 & 21\\ 35 & 11 & 17 & 23\\ 36 & 12 & 18 & 24\end{bmatrix}$$

局部均值采样

比例放大变换

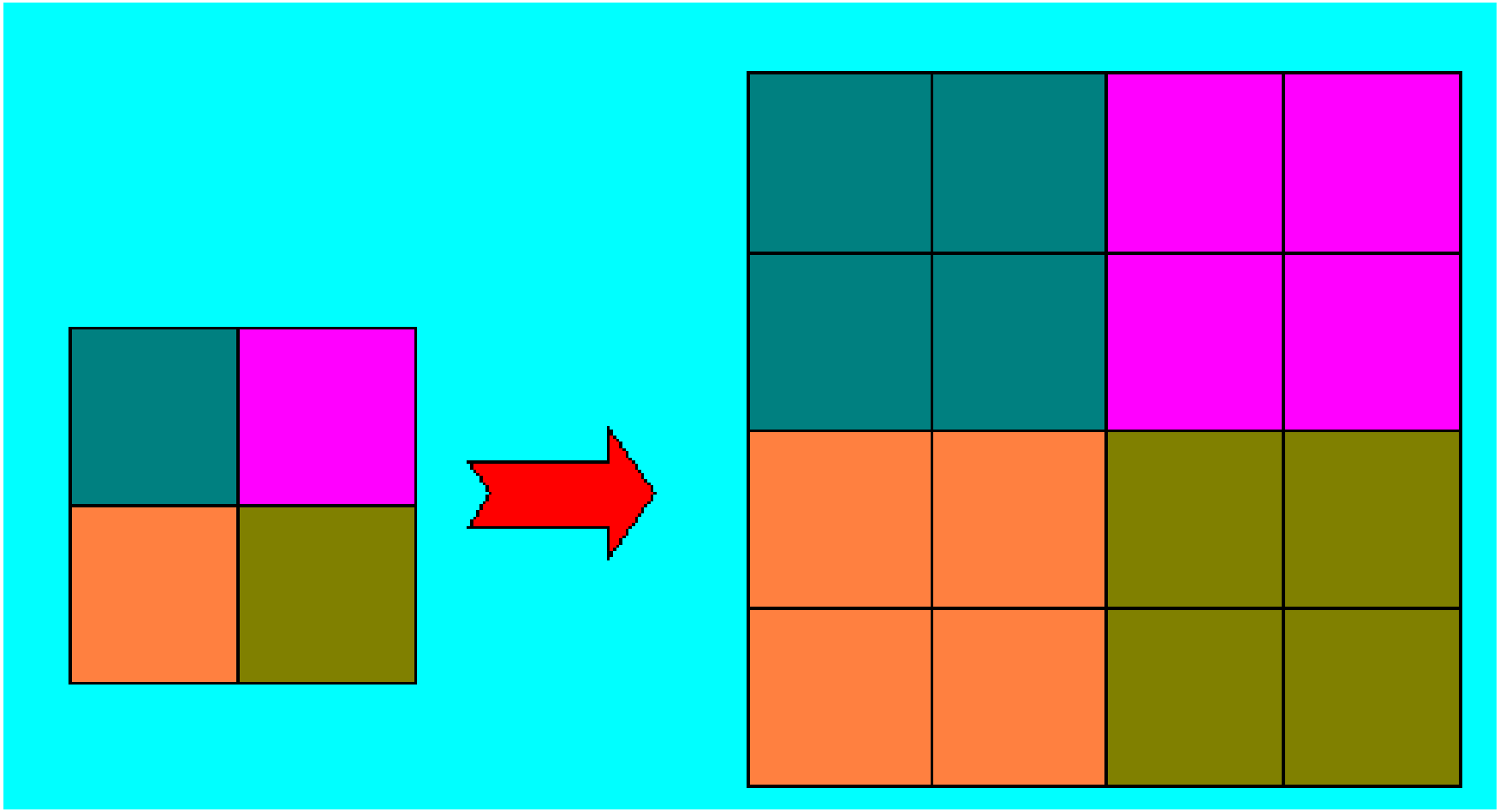
- 图像的比例放大变换
  - 图像在缩小操作中，是在现有的信息里如何挑选所需要的有用信息。而在图像的放大操作中，则需要对尺寸放大后所多出来的空格填入适当的像素值，这涉及像素信息的估计，一般比图像的缩小要难一些。
  - 由于图像的相邻像素之间的相关性很强，可以利用这个相关性来实现图像的放大。与图像缩小相同，按比例放大不会引起图像的畸变，而不按比例放大则会产生图像的畸变，图像放大一般采用最近邻域法和线性插值法。

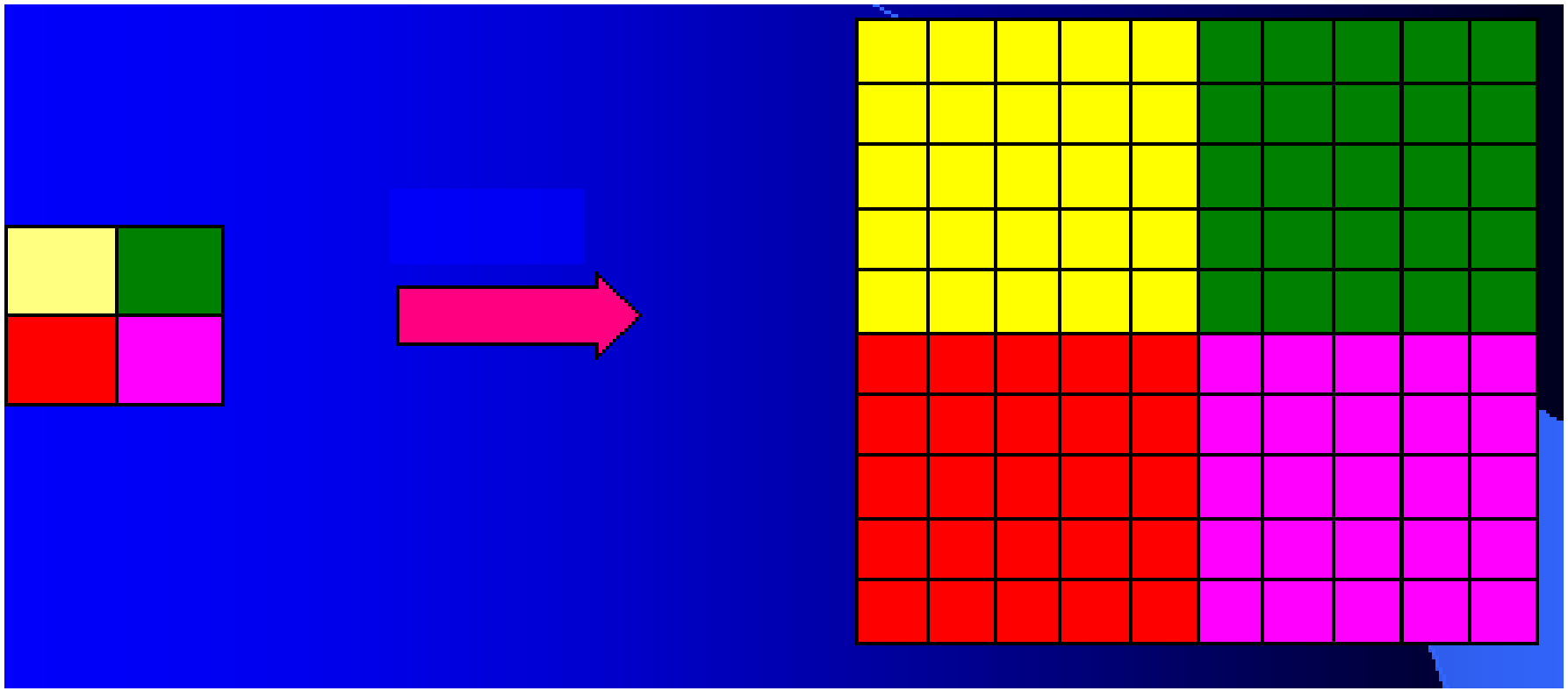
比例放大变换

- 最近邻域法
  - 按比例将原图像放大k倍时，如果按照最近邻域法则需要将一个像素值添在新图像的$k \times k$的子块中

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$$

$$G = \begin{bmatrix} f_{11} & f_{11} & f_{11} & f_{12} & f_{12} & f_{12} & f_{13} & f_{13} & f_{13} \\ f_{11} & f_{11} & f_{11} & f_{12} & f_{12} & f_{12} & f_{13} & f_{13} & f_{13} \\ f_{11} & f_{11} & f_{11} & f_{12} & f_{12} & f_{12} & f_{13} & f_{13} & f_{13} \\ f_{21} & f_{21} & f_{21} & f_{22} & f_{22} & f_{22} & f_{23} & f_{23} & f_{23} \\ f_{21} & f_{21} & f_{21} & f_{22} & f_{22} & f_{22} & f_{23} & f_{23} & f_{23} \\ f_{21} & f_{21} & f_{21} & f_{22} & f_{22} & f_{22} & f_{23} & f_{23} & f_{23} \\ f_{31} & f_{31} & f_{31} & f_{32} & f_{32} & f_{32} & f_{33} & f_{33} & f_{33} \\ f_{31} & f_{31} & f_{31} & f_{32} & f_{32} & f_{32} & f_{33} & f_{33} & f_{33} \\ f_{31} & f_{31} & f_{31} & f_{32} & f_{32} & f_{32} & f_{33} & f_{33} & f_{33} \end{bmatrix}$$

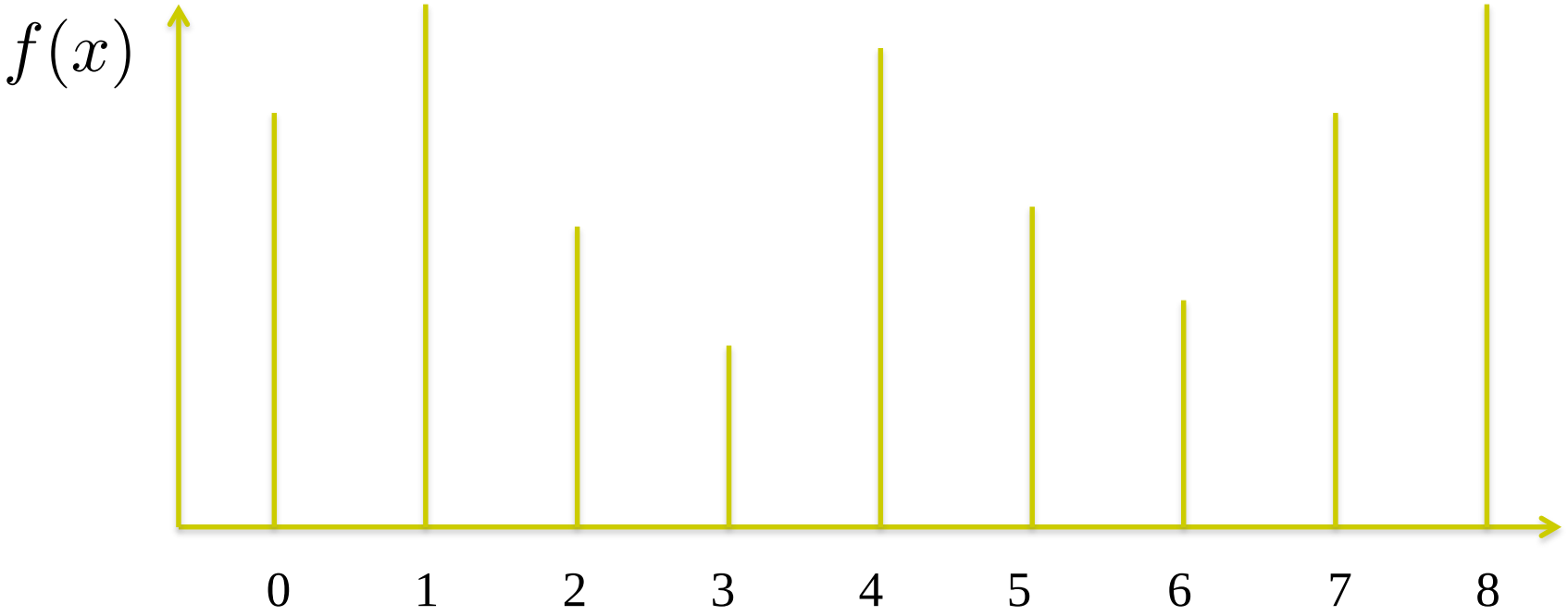




按最近邻域法放大五倍的图像

插值

$$f(0) = 10$$
$$f(1) = 12$$
$$f(0.5) = ?$$



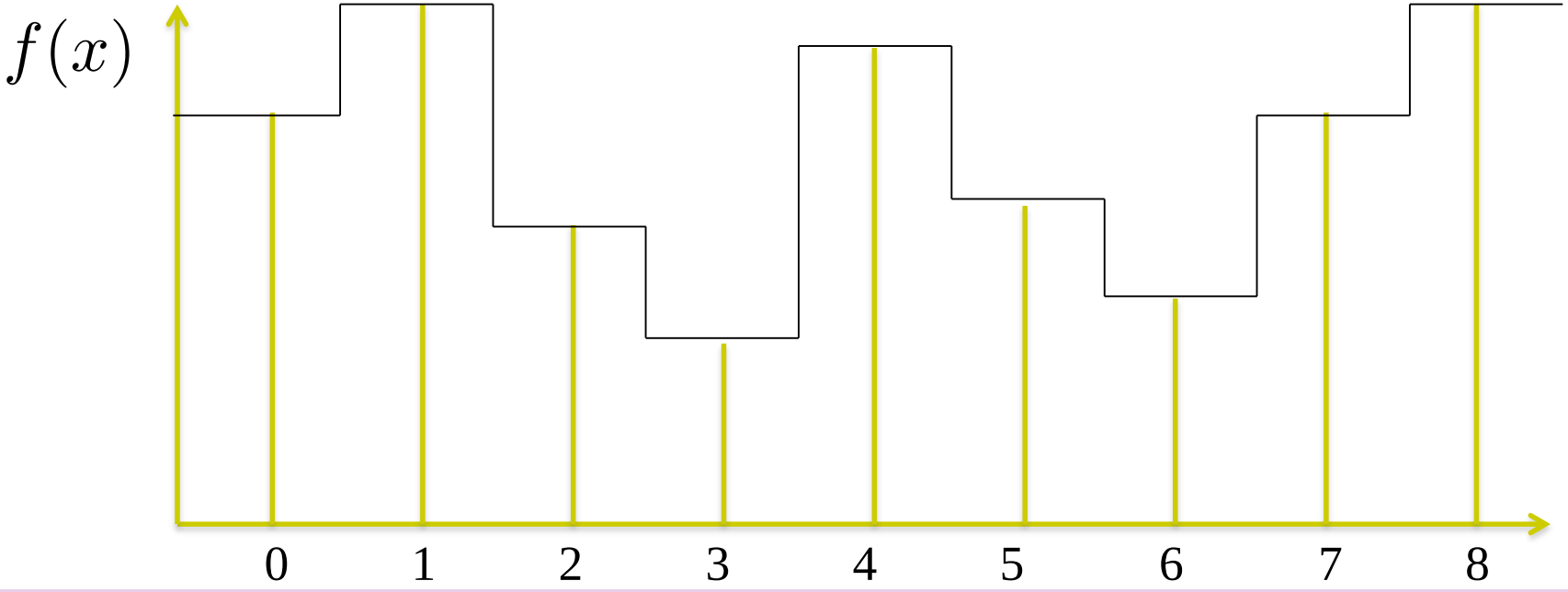
插值

- 最近邻插值

$$f(0) = 10$$

$$f(1) = 12$$

$$f(0.4) = f(0), f(0.6) = f(1), f(0.5) = \ldots$$



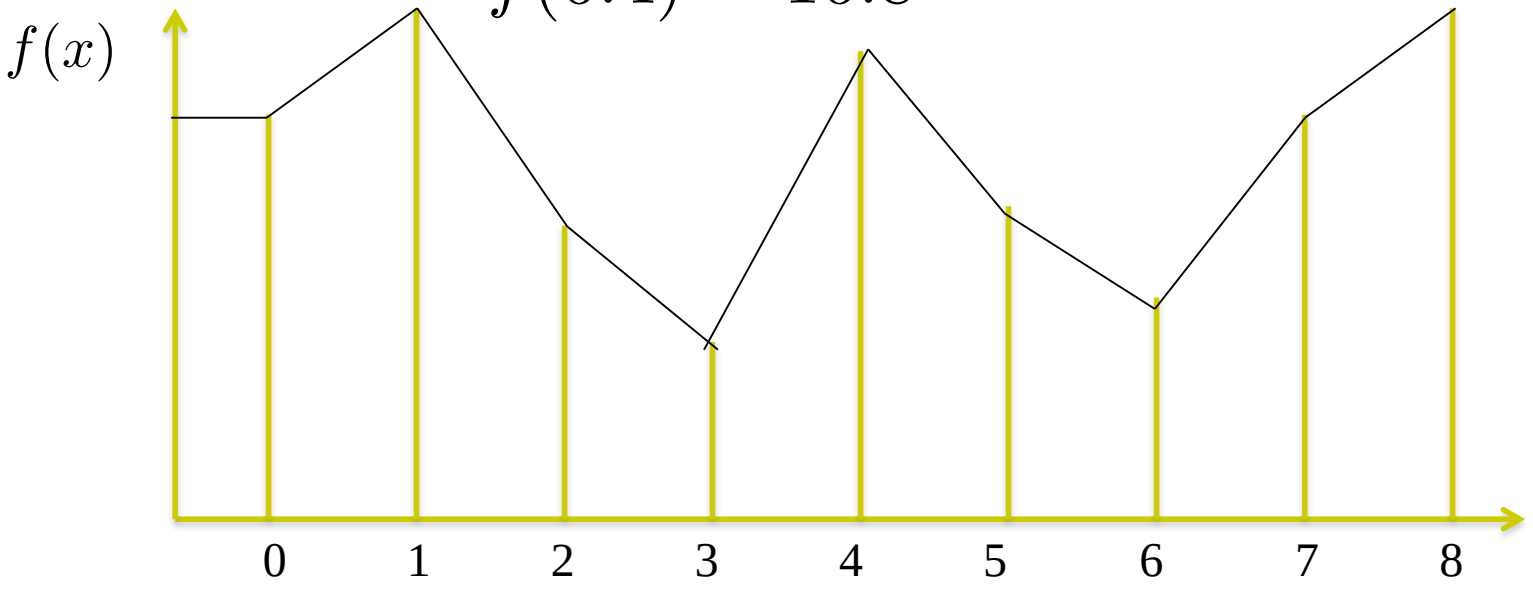
插值

- 线性插值

线性插值　$f(0) = 10, f(1) = 12$

解出[0,1]：　$f(x) = 10 + x * 2$

$f(0.4) = 10.8$



灰度级插值

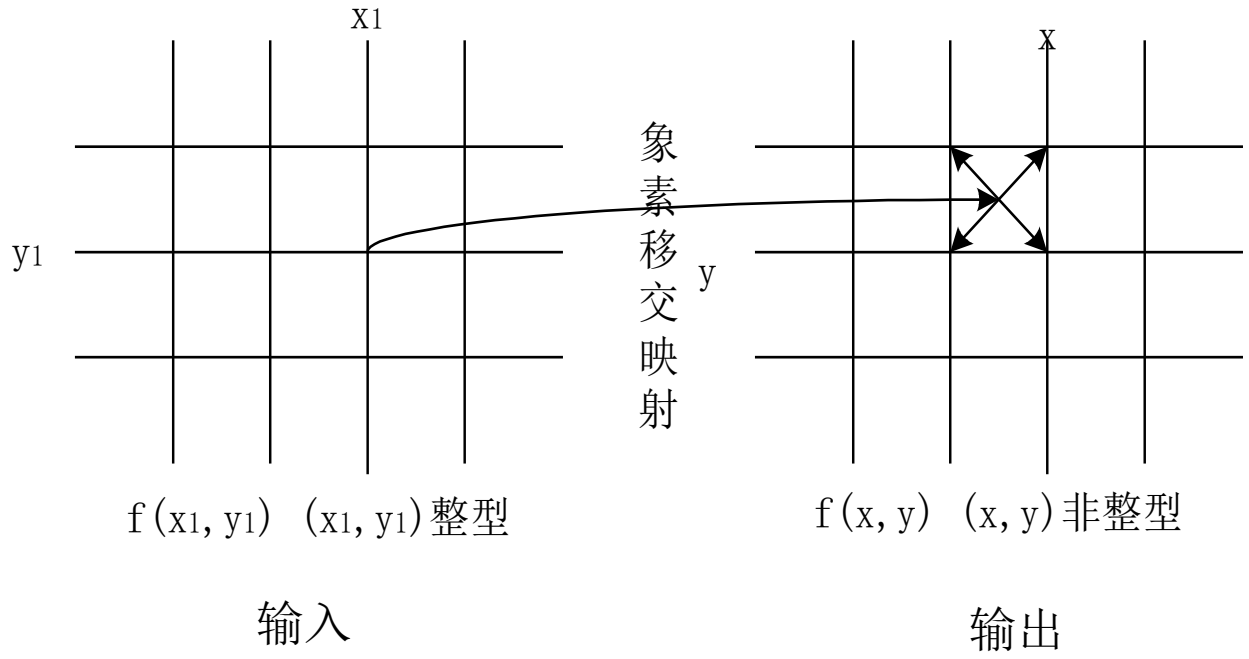
- 假设有一副大小为64×64的灰度图像A，现在将图像A 放大到256×256，令其为图像B。
  - 根据简单的几何关系，可以知道B图像中(x, y)处的像素值应该对应着A图像中的$(x/4, y/4)$处的像素值，即
$$B(x, y) = A(x/4, y/4)$$
  - 对于B中的(4,4),(4,8),(4,16)…(256,256)这些位置，可以计算出其在A中的位置，从而可以得到灰度值。
  - 对于B中的(1,1),(1,2),(1,3)…等坐标点而言，计算后它们在A中对应的坐标不再是整数。例如对于B中的坐标点(1,1)，其在A中的对应坐标就变成了(0.25,0.25)。对于数字图像来说，小数坐标没有意义。

灰度级插值

- 向前映射法

通过输入图像像素位置，计算输出图像对应像素位置；

将该位置像素的灰度值按某种方式分配到输出图像相邻四个像素.

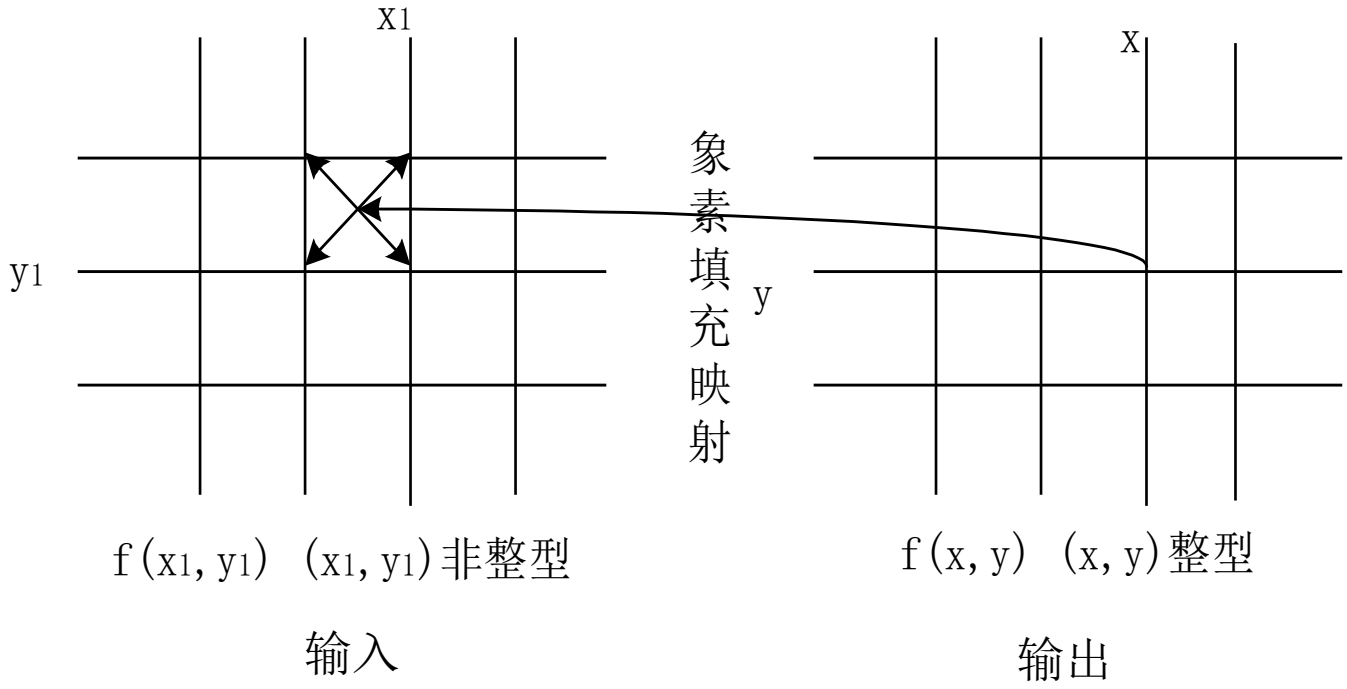


灰度级插值

- 向后映射法

通过输出图像像素位置, 计算输入图像对应像素位置;

根据输入图像相邻四个像素的灰度值计算该位置像素的灰度值.



灰度级插值

- 两种映射方法的对比
  - 对于向前映射：

    每个输出图像的灰度要经过多次运算；
  - 对于向后映射：

    每个输出图像的灰度只要经过一次运算。

实际应用中，更经常采用向后映射法。

其中，根据四个相邻像素灰度值计算某位置的像素灰度值即为灰度级插值。

灰度级插值

- 双线性插值
  - 问题描述：单位正方形顶点已知，求正方形内任一点的$f(x,y)$值。

灰度级插值

- 基本过程
  - Step 1: 定义双线性方程$f(x,y)=ax+by+cxy+d.$
  - Step 2: 代入已知四点的值

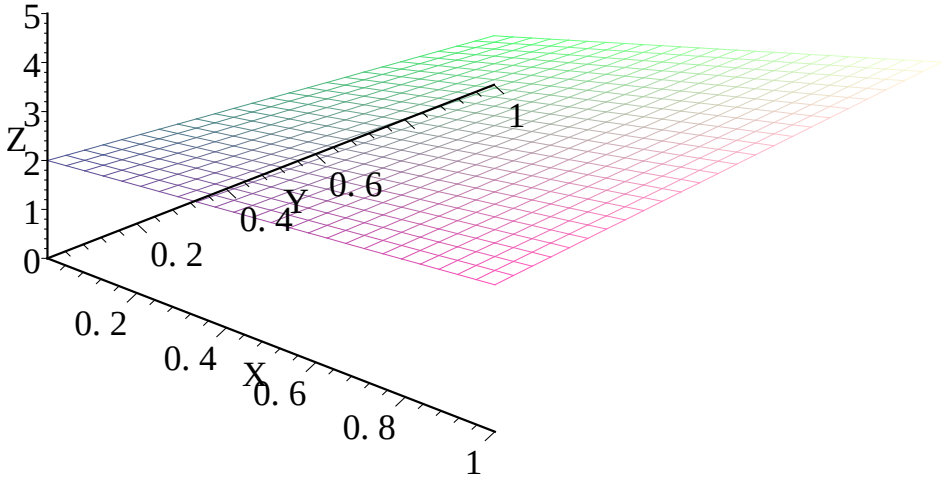
$$\begin{cases} f(0,0)=d \\ f(1,0)=a+d \\ f(0,1)=b+d \\ f(1,1)=a+b+c+d \end{cases}$$

  - Step3: 根据插值公式计算

$$f(x,y)=\left[f(1,0)-f(0,0)\right]x+\left[f(0,1)-f(0,0)\right]y+\left[f(1,1)+f(0,0)-f(1,0)-f(0,1)\right]xy+f(0,0)$$

灰度级插值

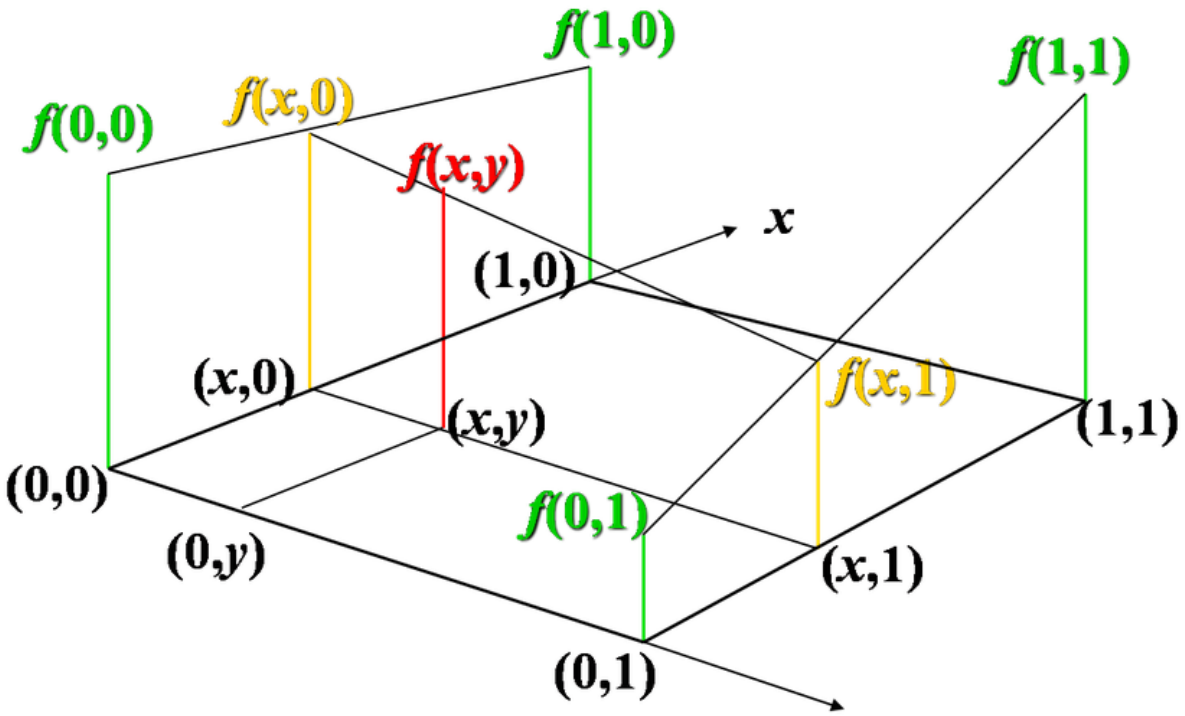
- 假设 $f(0,0)=2, f(1,0)=3, f(0,1)=1, f(1,1)=4$
- 则 $f(x,y)=x-y+2xy+2$



$$f(x,0) = f(0,0) + x\left[f(1,0) - f(0,0)\right]$$

$$f(x,1) = f(0,1) + x\left[f(1,1) - f(0,1)\right]$$

$$f(x,y) = f(x,0) + y\left[f(x,1) - f(x,0)\right]$$



灰度级插值

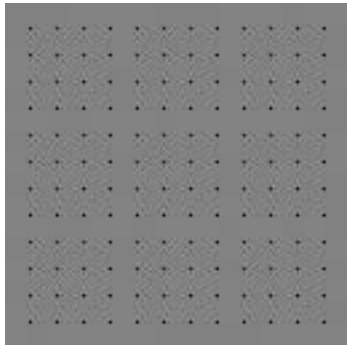
- 双线性插值能够避免采用最近领域插值方式时可能存在的图像模糊、块状失真等问题。

采用最近邻插值放大1.5倍

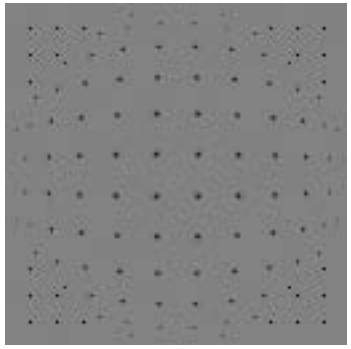
采用双线性插值放大1.5倍

空间变换

- 多项式卷绕
  - 多项式的项数与控制点数相同，解线性方程组，得系数后矩阵求逆。



测试靶



对应的鱼眼图像

空间变换

- 多项式卷绕的基本原理
  - 对于一个有10个点的测试靶，可以设计一个二维三阶函数拟合。

$$p(x,y)=c_0+c_1x+c_2y+c_3xy+c_4x^2$$
$$+c_5y^2+c_6x^2y+c_7xy^2+c_8x^3+c_9y^3$$
$$\begin{bmatrix} p(x_1,y_1) \\ p(x_2,y_2) \\ \vdots \\ p(x_{10},y_{10}) \end{bmatrix}=\begin{bmatrix} 1 & x_1 & \cdots & {y_1}^3 \\ 1 & x_2 & \cdots & {y_2}^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{10} & \cdots & {y_{10}}^3 \end{bmatrix}\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_9 \end{bmatrix}$$

空间变换

变形后的老虎

校正后的老虎

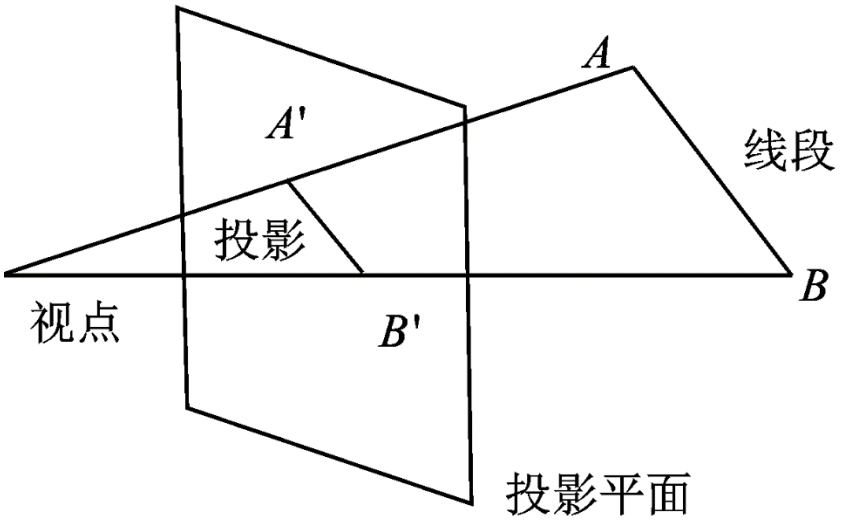
三维几何变换的投影变换简介

投影变换

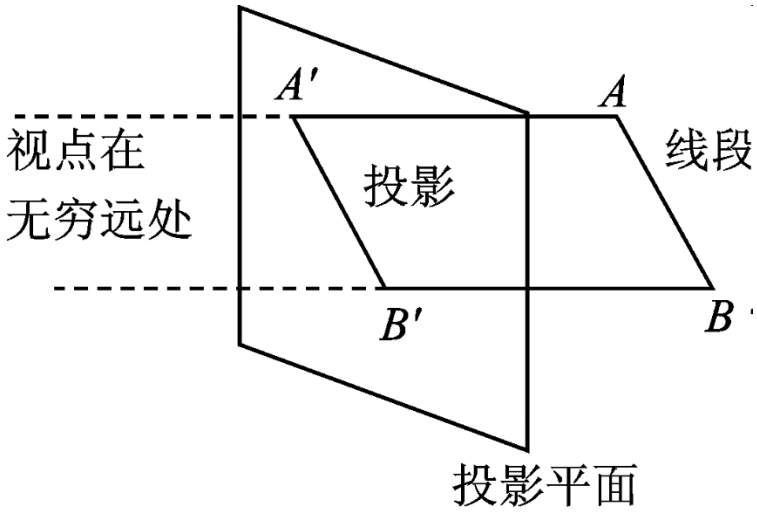
- 在一幅二维图像上显示三维图形的对象形状，实际上是完成了一次三维信息到二维平面上的投影过程。把三维物体或对象转变为二维图形表示的过程称为投影变换。
- 根据视点（投影中心）与投影平面之间距离的不同，投影可分为平行投影和透视投影。

投影变换

- 根据视点（投影中心）与投影平面之间距离的不同，投影可分为平行投影和透视投影。



(a) 透视投影



(b) 平行投影

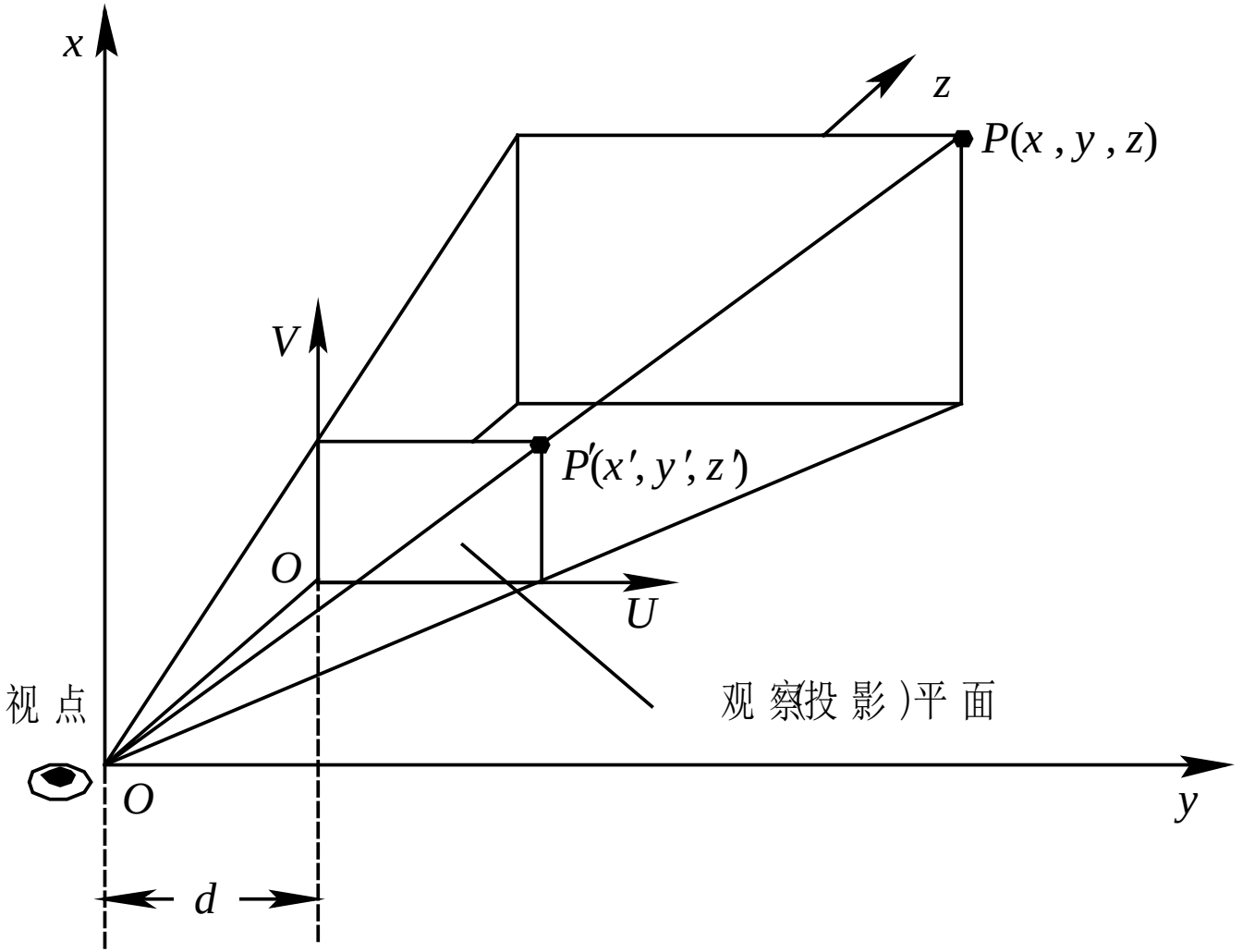
透视投影

把三维物体或对象转变为二维图形表示的过程称为投影变换。

根据视点（投影中心）与投影平面之间距离的不同，投影可分为平行投影和透视投影。

平行投影的视点与投影平面之间的距离为无穷大。

透视投影（变换），该距离是有限的。这个距离决定着透视投影的特性——透视缩小效应， 即三维物体或对象透视投影的大小与形体到视点的距离成反比。

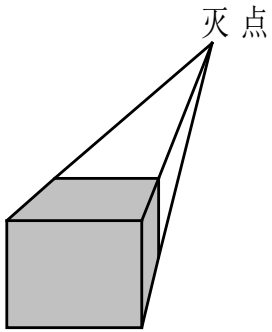

x
z
P(x , y , z)
V
P′(x′, y′, z′)
O
U
视 点
观 察(投 影)平 面
O
y
d

透视投影

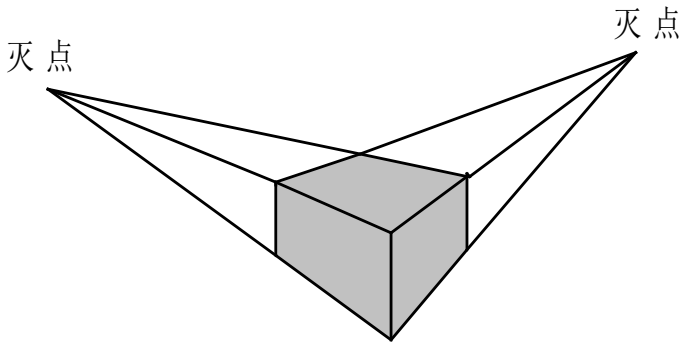
- 对于透视投影，一束平行于投影面的平行线的投影可保持平行，而不平行于投影面的平行线的投影会聚集到一个点，这个点称为灭点。
- 灭点可以看作是无限远处的一点在投影面上的投影。透视投影的灭点可以有无限多个，不同方向的平行线在投影面上就能形成不同的灭点，坐标轴方向的平行线在投影面上形成的灭点又称作主灭点。
- 可分为一点透视、二点透视和三点透视。

透视投影

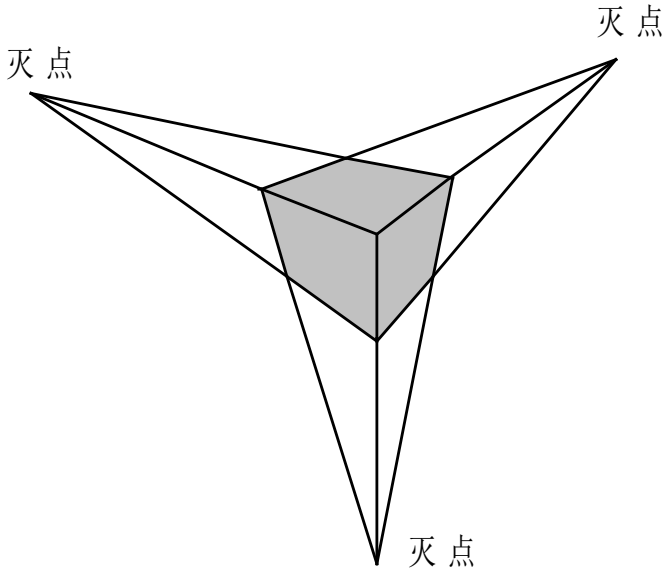
- 分为一点透视、二点透视和三点透视。



(a)



(b)



(c)