数字图像处理

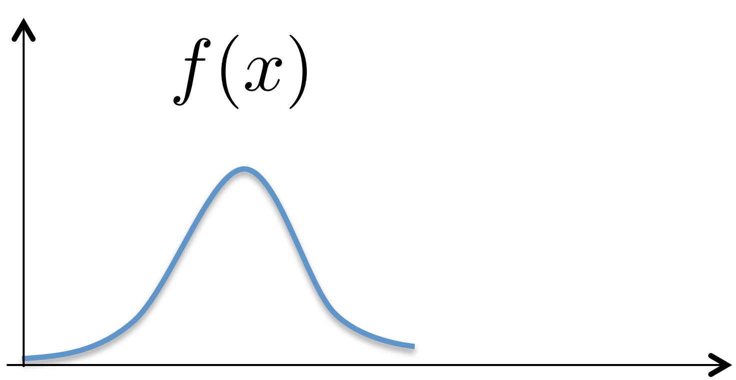
图像插值与几何运算

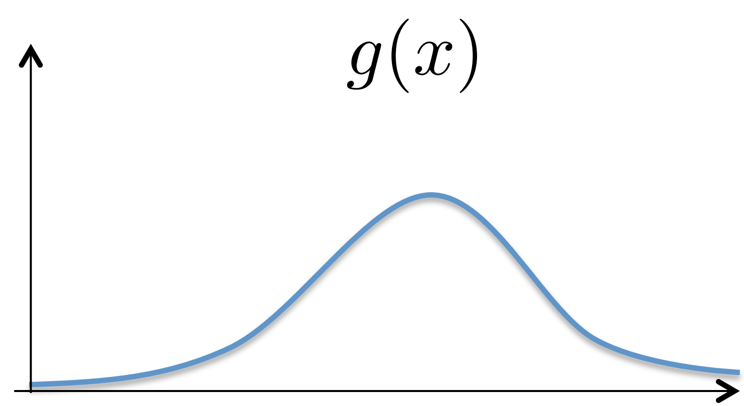
1 引言

函数图像的放大缩小

放大一倍：

$$g(x) = f(\frac{1}{2}x) \qquad g(2x) = f(x)$$





1 引言

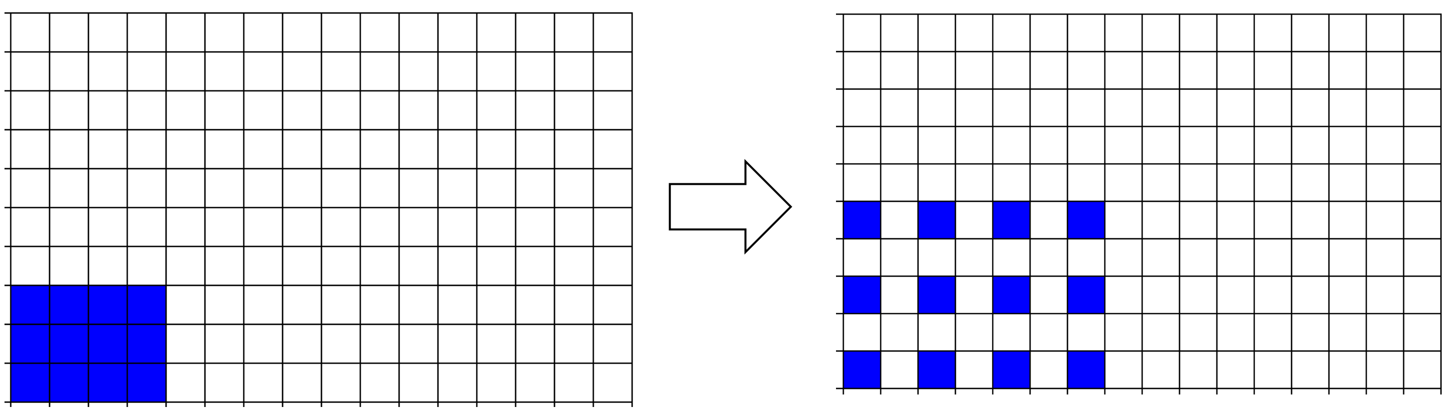
图像的放大缩小

$$g[2x, 2y] = f[x, y]$$

1 引言

图像的放大缩小

$$g[2x, 2y] = f[x, y]$$



1 引言

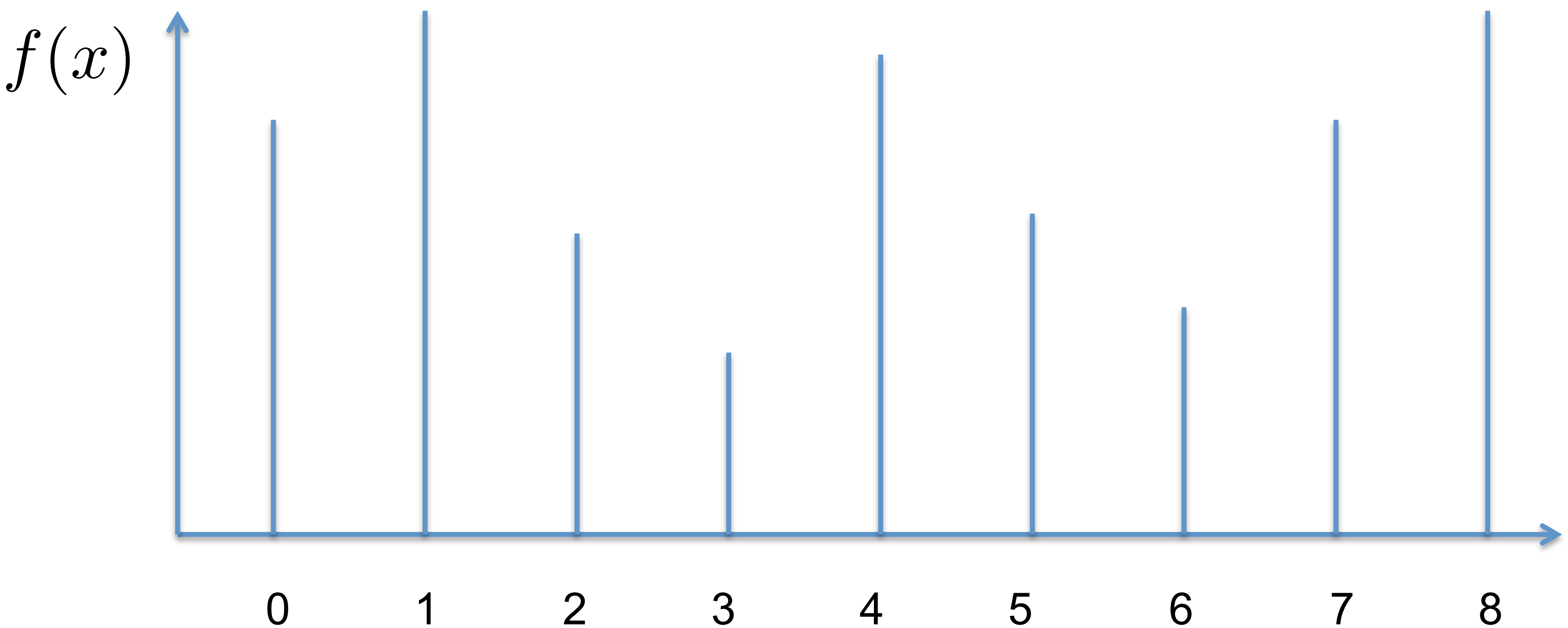
Lenna及变形图像

插值

$$f(0) = 10$$

$$f(1) = 12$$

$$f(0.5) = ?$$



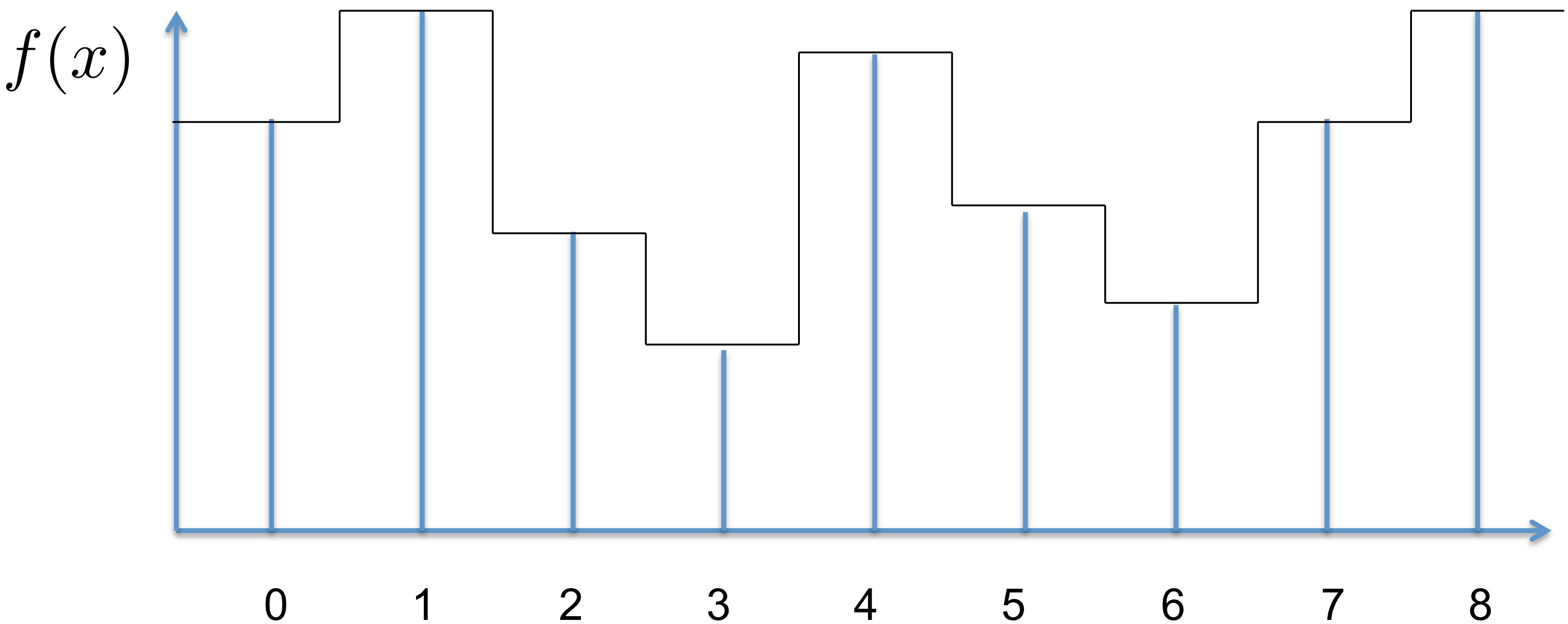
插值

最近邻插值

$$f(0) = 10$$

$$f(1) = 12$$

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

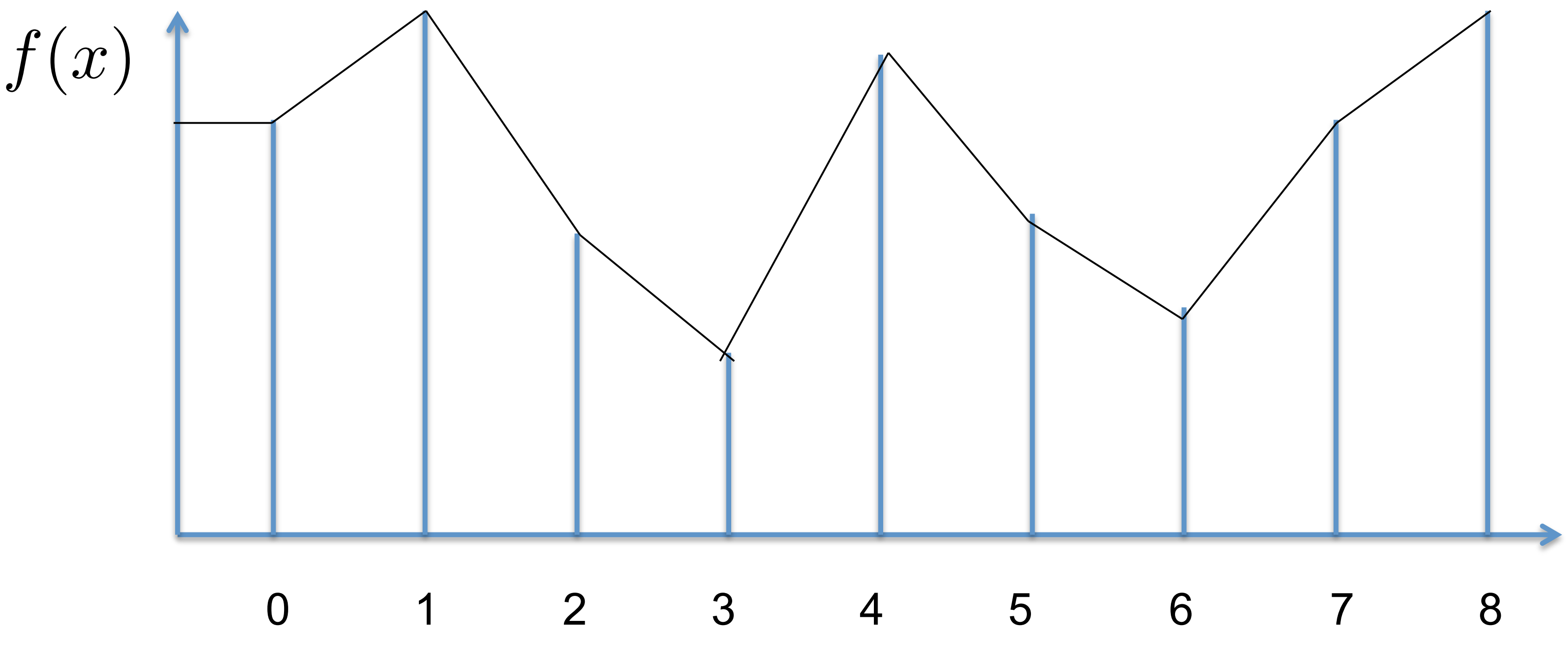


插值

线性插值 $f(0) = 10, f(1) = 12$

解出[0,1]: $f(x) = 10 + x * 2$

$$f(0.4) = 10.8$$

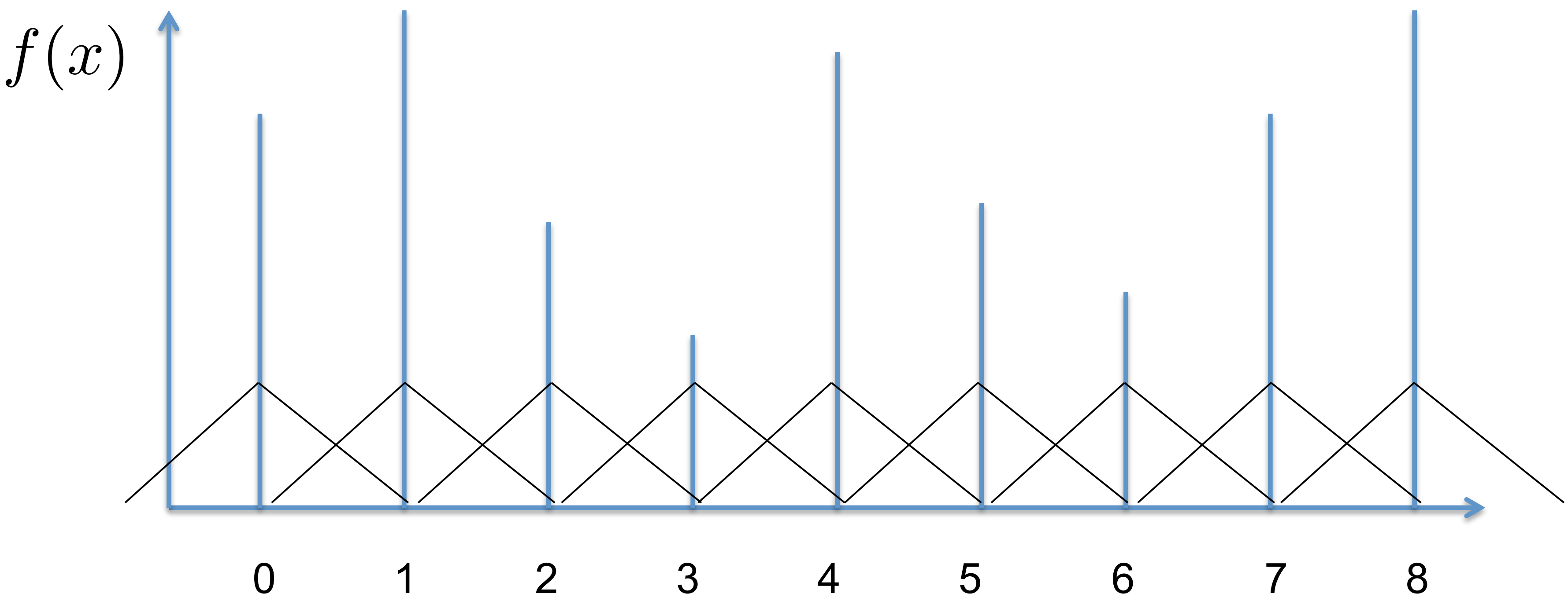


插值

线性插值 $f(0) = 10, f(1) = 12$

另一种写法：$f(x) = (1 - x) * f(0) + (x - 0) * f(1)$

$$f(0.4) = 0.6 * 10 + 0.4 * 12 = 6 + 4.8 = 10.8$$

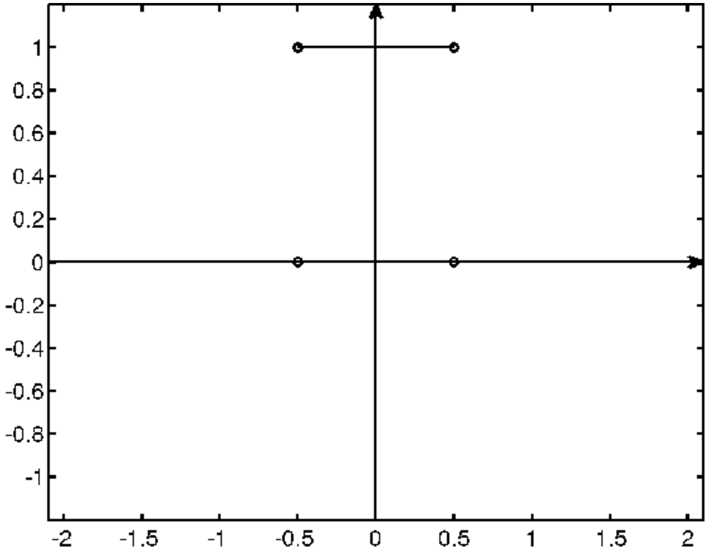


插值

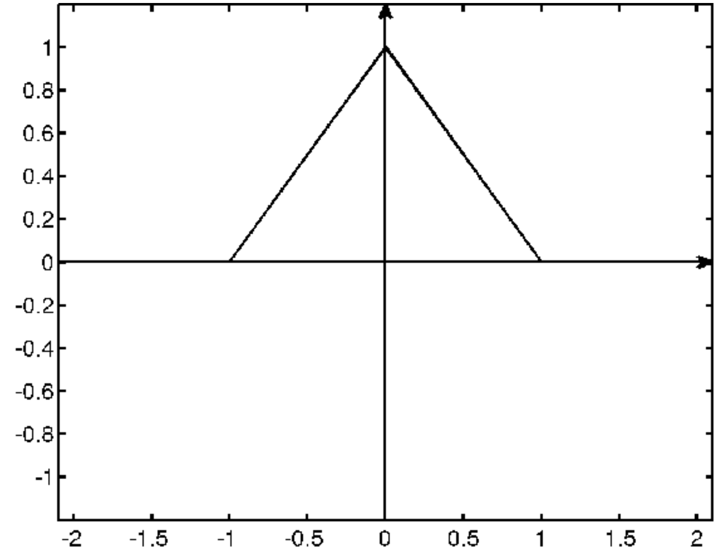
B样条

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$



p=0



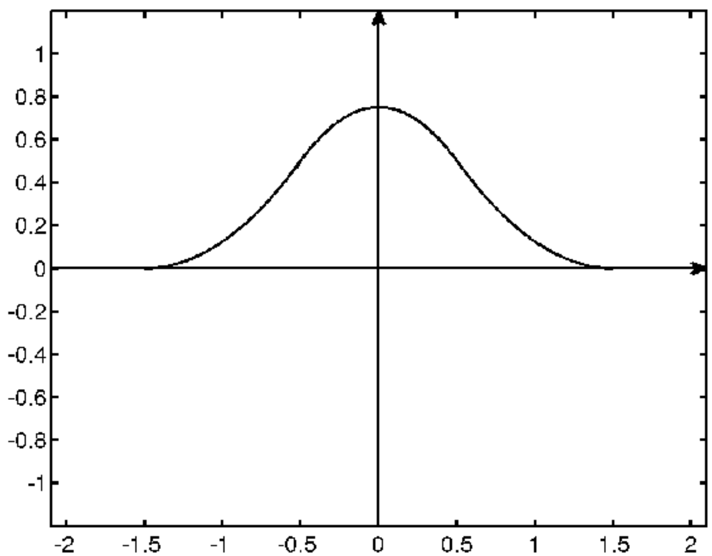
p=1

插值

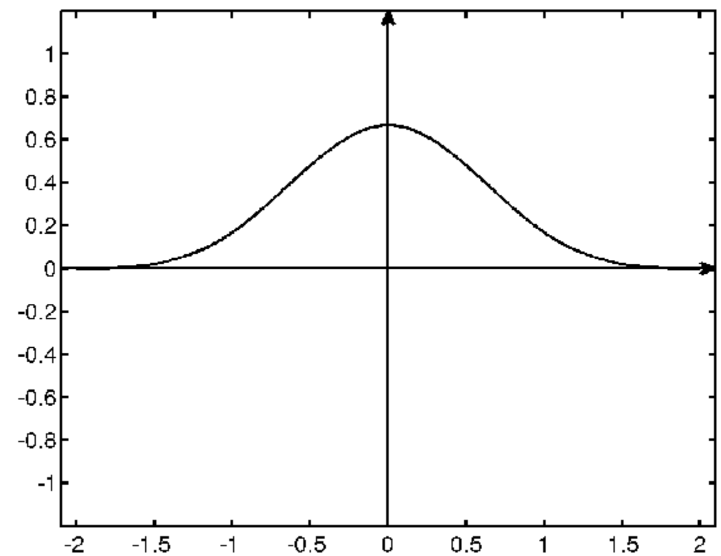
B样条

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$



p=2



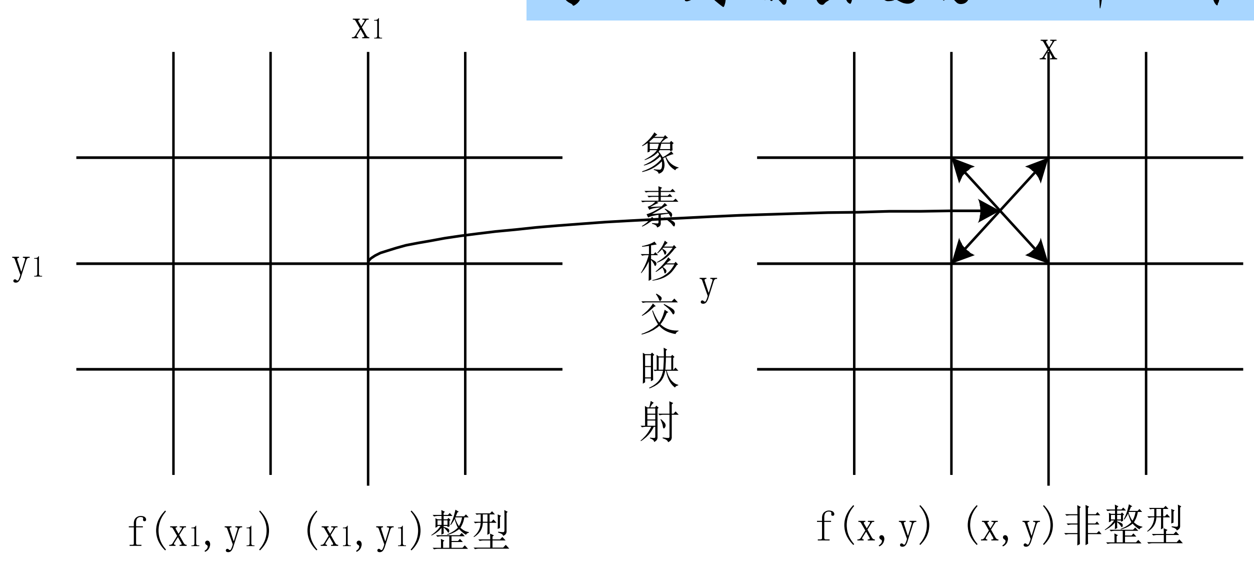
p=3

2 灰度级插值

- 1）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。

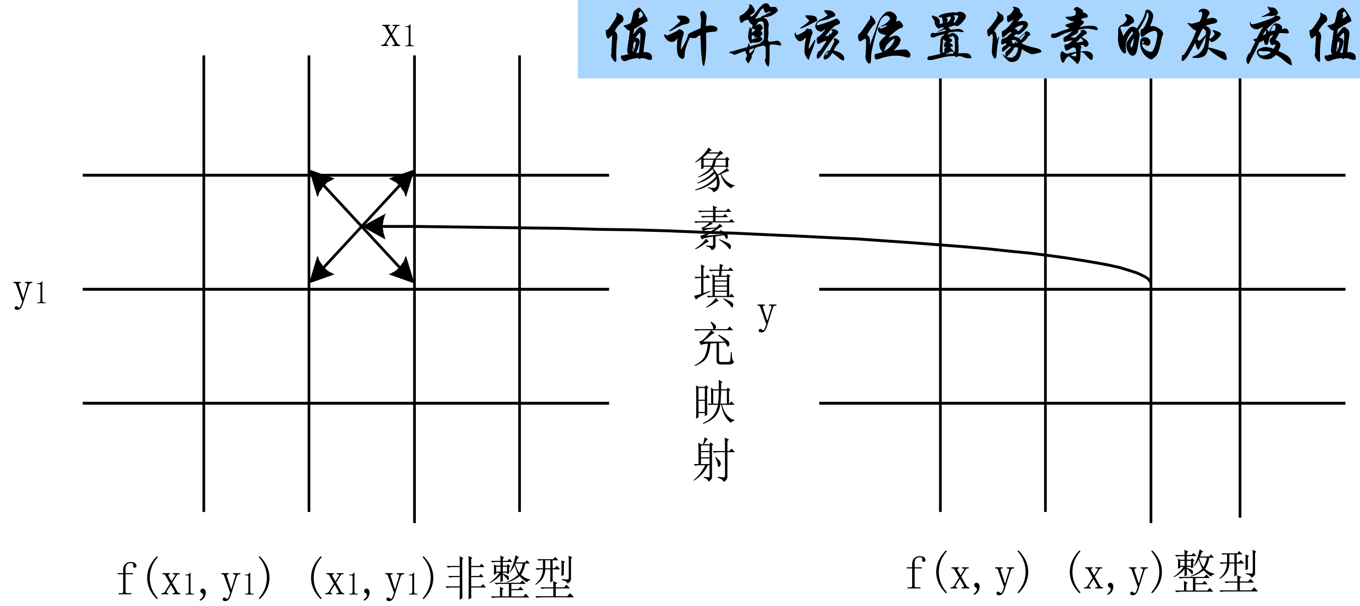


2 灰度级插值

- 2）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。



2 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。

2 灰度级插值

两种常用方法

- 3）最近邻插值
  - 向后映射时，输出图像的灰度等于离它所映射位置最近的输入图像的灰度值。
- 4）双线性插值
  - 四点确定一个平面函数，属于过约束问题；
  - 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。

2 灰度级插值

$setp1:$定义双线性方程$f\left(x,y\right)=ax+by+cxy+d$

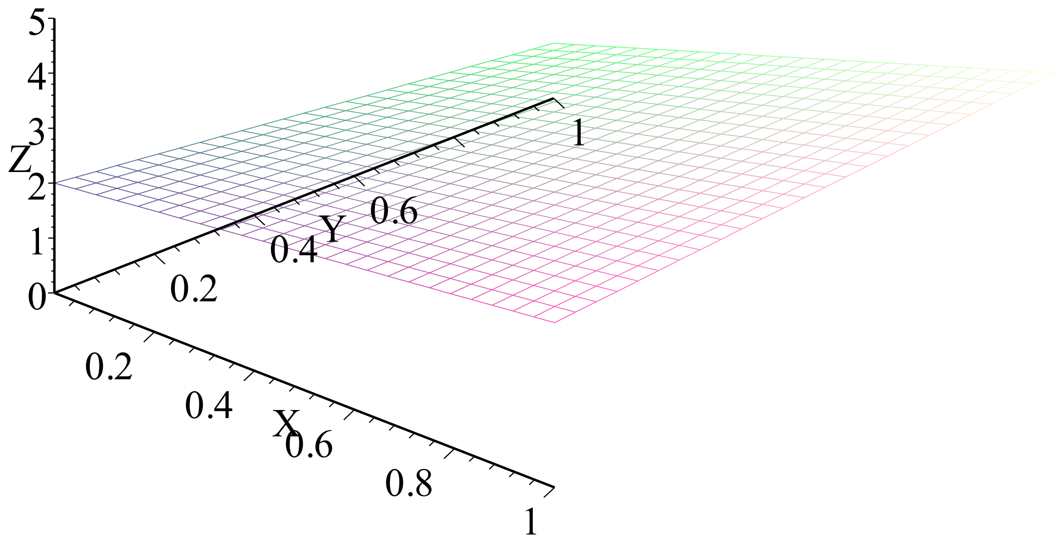
$step2:$代入已知四点的值

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

$$step3: f\left(x,y\right)=\left[f\left(1,0\right)-f\left(0,0\right)\right]x+\left[f\left(0,1\right)-f\left(0,0\right)\right]y$$
$$+\left[f\left(1,1\right)+f\left(0,0\right)-f\left(1,0\right)-f\left(0,1\right)\right]xy+f\left(0,0\right)$$

2 灰度级插值

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



2 灰度级插值

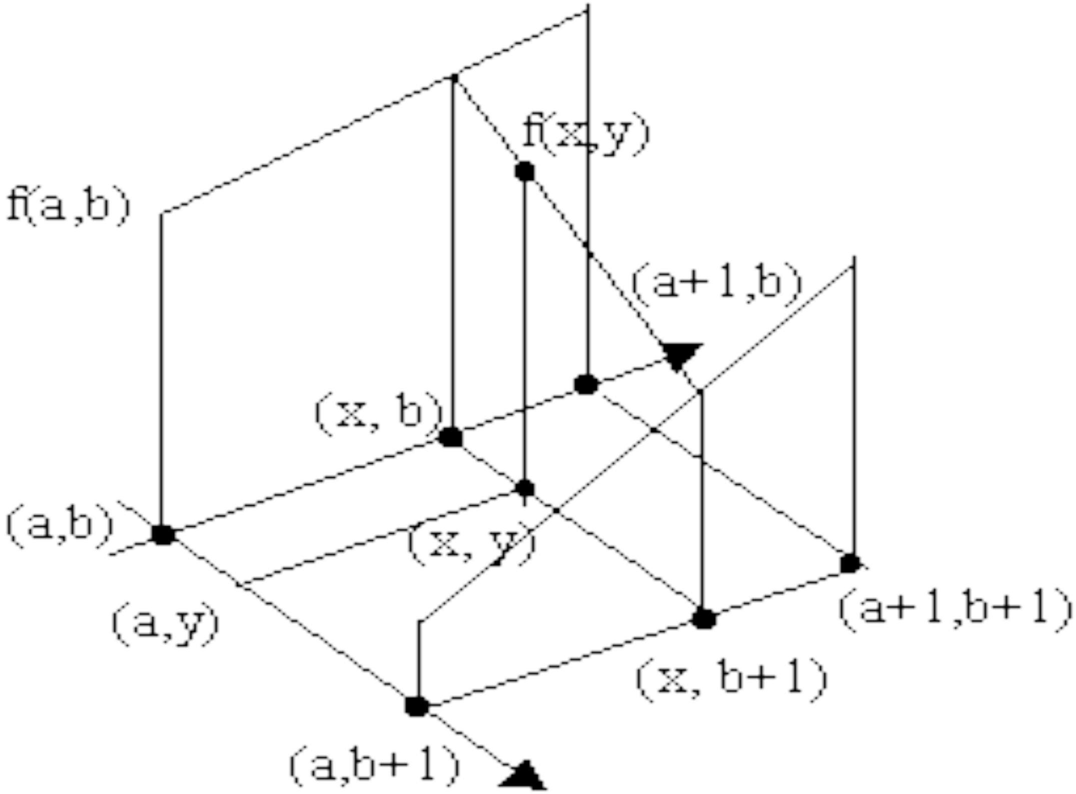


图 1. 点（x, y）灰度值的确定

2 灰度级插值

- 双线性插值的第二种算法

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

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

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

2 灰度级插值

- 用最近邻插值和双线性插值的方法分别将老虎放大1.5倍。

2 灰度级插值

采用最近邻插值放大1.5倍

采用双线性插值放大1.5倍

2 灰度级插值

- 5）比例变换中对应图像的确定
  - 假设输出图像的宽度为W，高度为H;
  - 输入图像的宽度为w高度为h，要将输入图像的尺度拉伸或压缩变换至输出图像的尺度;
  - 按照线性插值的方法，将输入图像的宽度方向分为W等份，高度方向分为H等份;
  - 那么输出图像中任意一点（x，y）的灰度值就应该由输入图像中四点（a，b）、（a+1，b）、（a，b+1）和（a+1，b+1）的灰度值来确定。其中a和b的值分别为:

$$a = \left[x \cdot \frac{w}{W} \right] \quad b = \left[y \cdot \frac{h}{H} \right]$$

3 空间变换

- 1）简单变换
- 2）多项式卷绕和几何校正
- 3）控制栅格插值和图像卷绕

3 空间变换

- 1）简单变换
  - 问题描述：图像的平移、放缩和旋转。
  - 解题思路：从易到难。工具：线性代数中的齐次坐标。

$step1$：图象的平移

$$a(x,y)=x+x_0$$

$$b(x,y)=y+y_0$$

$$\begin{bmatrix} a(x,y) \\ b(x,y) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

3 空间变换

$step2:$图象在x方向放大c倍，y方向放大d倍。

$$a(x,y)=cx \quad b(x,y)=dy$$

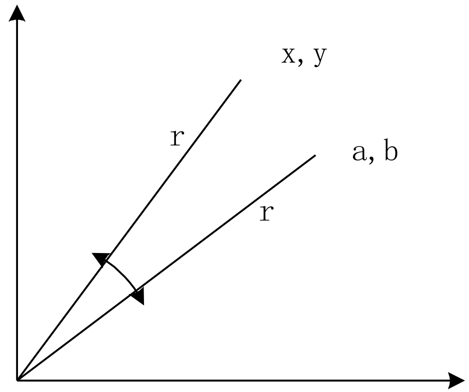
$$\begin{bmatrix} a(x,y) \\ b(x,y) \\ 1 \end{bmatrix} = \begin{bmatrix} c & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

3 空间变换

$step3$：图象饶原点顺时针旋转θ角。

$$a(x,y)=x\cos(\theta)\text{-ysin}(\theta) \quad b(x,y)=x\sin(\theta)\text{+ycos}(\theta)$$

$$\begin{bmatrix} a(x,y) \\ b(x,y) \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



3 空间变换

$$a = r\cos(\alpha - \theta) = r\cos\alpha\cos\theta + r\sin\alpha\sin\theta = x\cos\theta + y\sin\theta$$

$$b = r\sin(\alpha - \theta) = r\sin\alpha\cos\theta - r\cos\alpha\sin\theta = y\cos\theta - x\sin\theta$$

$$= -x\sin\theta + y\cos\theta$$

$$\sin(\alpha - \theta) = \sin\alpha\cos\theta - \cos\alpha\sin\theta$$

$$\cos(\alpha - \theta) = \cos\alpha\cos\theta + \sin\alpha\sin\theta$$

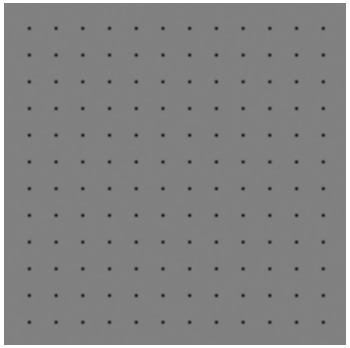
3 空间变换

- 复合变换：绕点(x_0,y_0)旋转θ角。

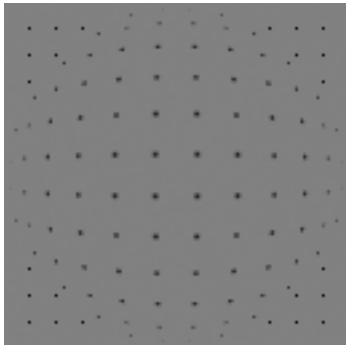
$$\begin{bmatrix} a(x,y) \\ b(x,y) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

3 空间变换

- 2）多项式卷绕和几何校正
  - 几何校正（calibration）:由于镜头的几何变形，使用矩形栅格校准。
  - 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

3 空间变换

变形后的老虎

校正后的老虎

3 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。

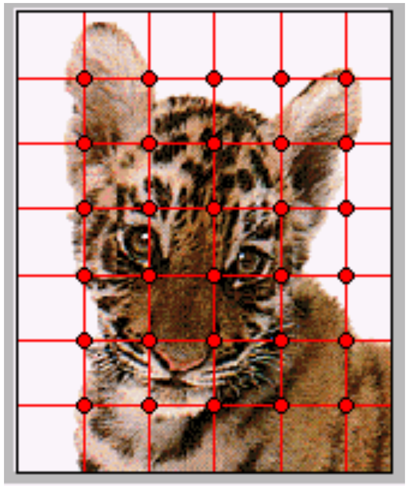
$$p(x,y)=c_0+c_1x+c_2y+c_3xy+c_4x^2$$
$$+c_5y^2+c_6x^2y+c_7xy^2+c_8x^3+c_9y^3$$

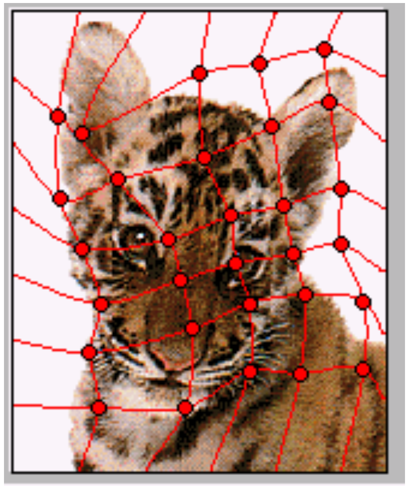
$$\begin{bmatrix} p(x_1,y_1) \\ p(x_2,y_2) \\ \vdots \\ p(x_{10},y_{10}) \end{bmatrix} = \begin{bmatrix} 1 & x_1 & \cdots & y_1{}^3 \\ 1 & x_2 & \cdots & y_2{}^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{10} & \cdots & y_{10}{}^3 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_9 \end{bmatrix}$$

3 空间变换

- 3）控制栅格插值和图像卷绕
  - 控制栅格插值：将图像分成小块进行卷绕变换。
  - 常用双线性插值。
  - 又是双线性插值，区别? (请思考)

3 空间变换





3 空间变换

3 空间变换

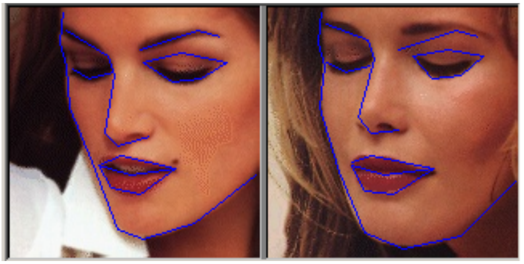
$Method1:$双线性空间变换的表达式

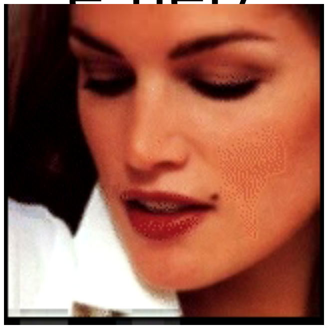
$$G(x,y)=F(ax+by+cxy+d,ex+fy+gxy+h)$$

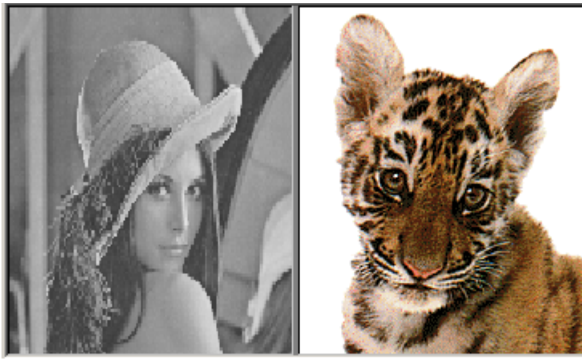
解4个点8个方程式可得。

4 图像变形

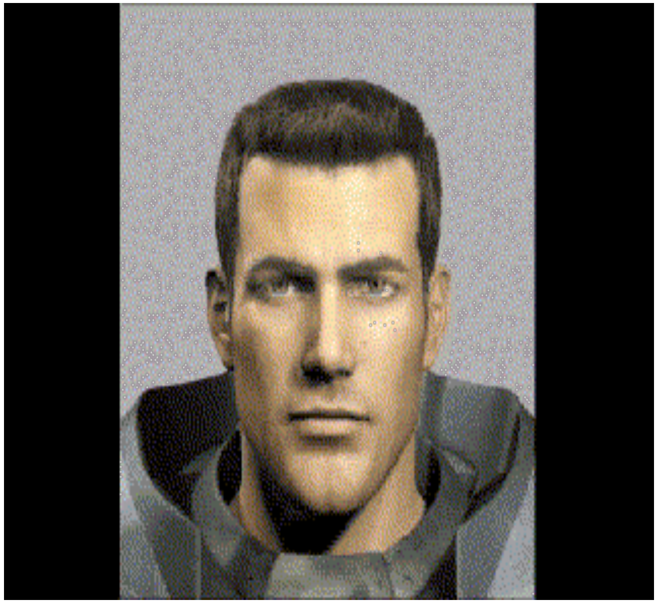
http://
www.morpheussoftwar
e.net/





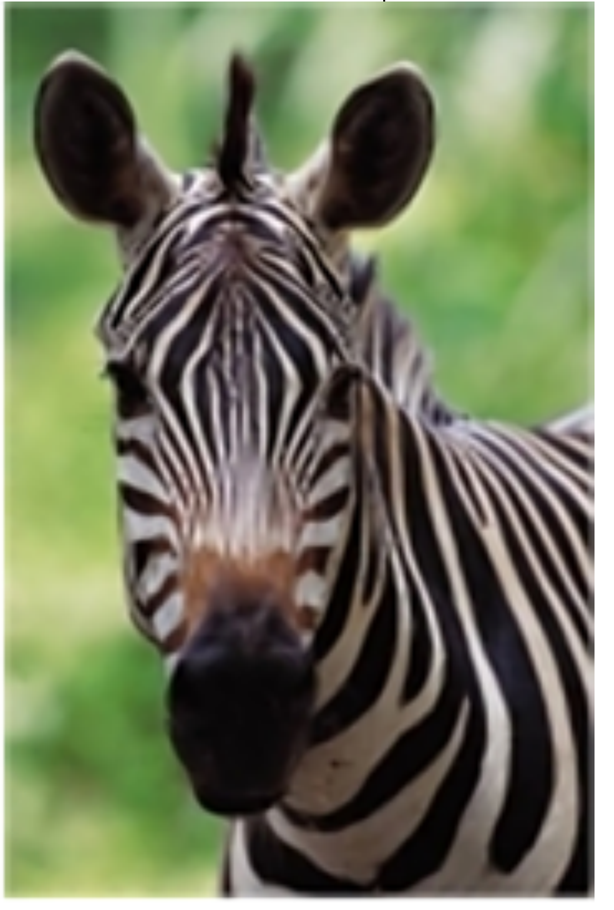


4 图像变形



5 要点总结

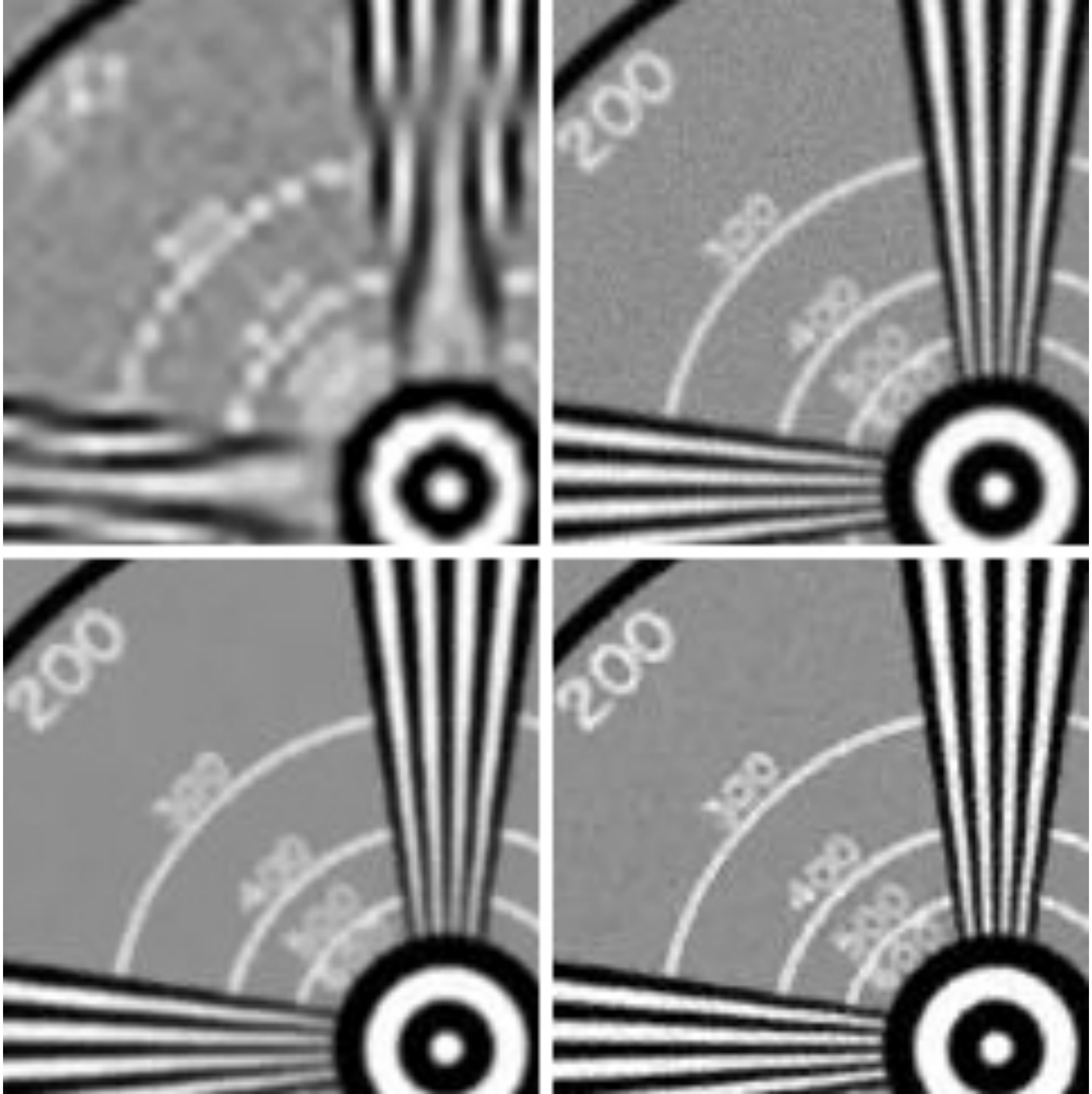
- 1）几何运算包括空间变换和灰度级插值两步；
- 2）灰度级插值有向前映射和向后映射两种；
- 3）灰度级插值有包括最近邻插值和双线性插值两类；
- 4）几何运算可用在几何校正、图像卷绕以及图像变形等应用中。



Bilinear Interpolation **Bicubic Interpolation** **Dir8 algorithm**

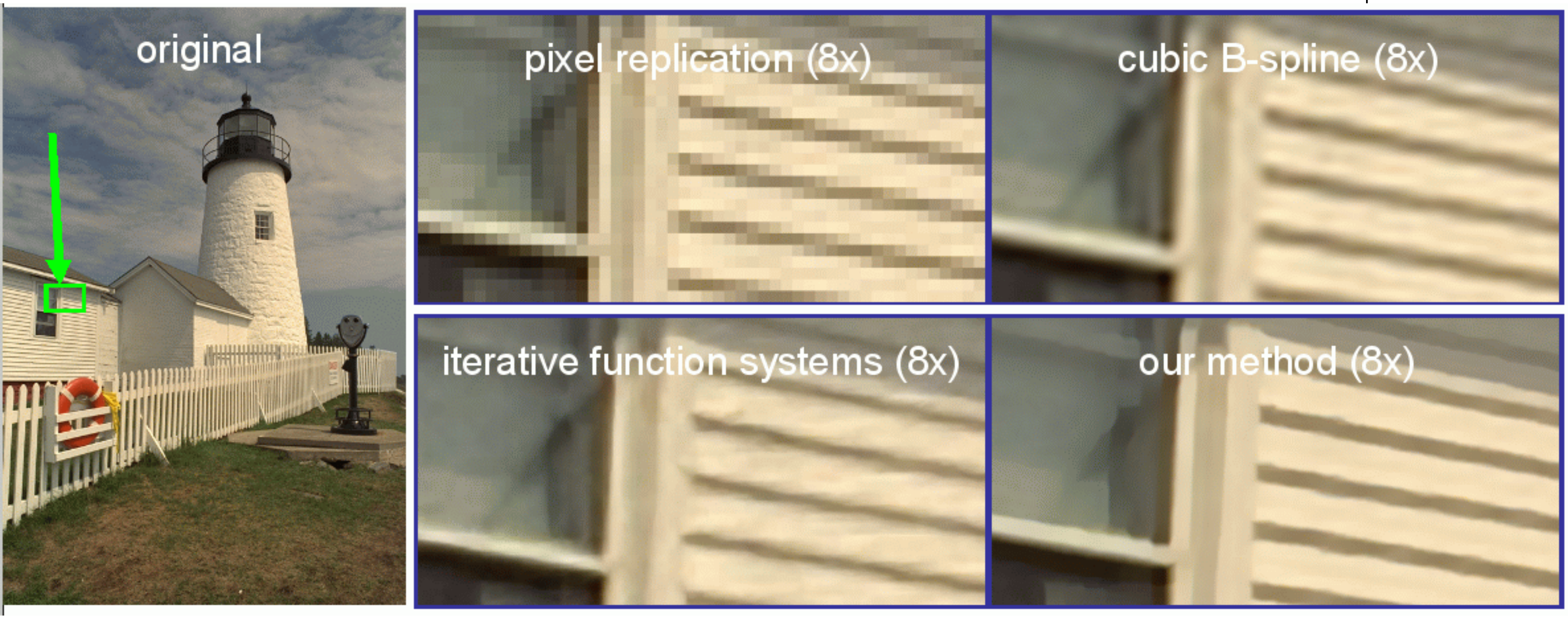
Comparison with conventional interpolation algorithms

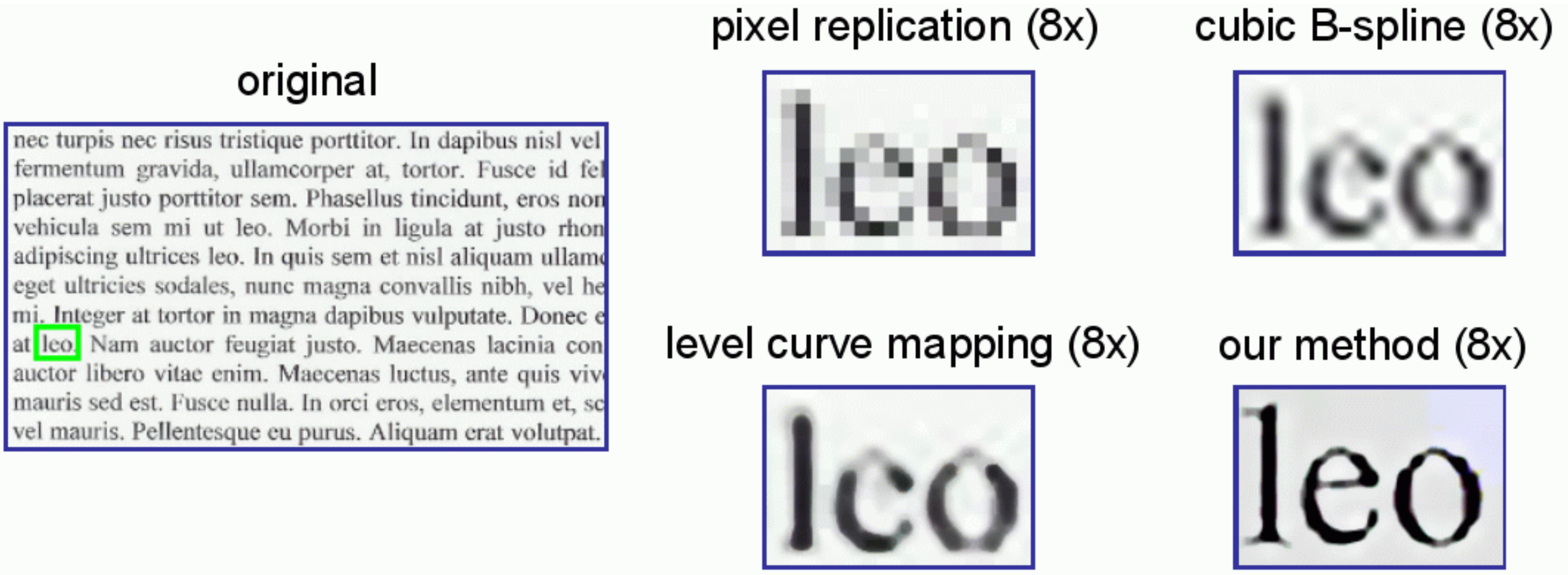
(Original image: 100x150 pixel s, Output image: 400x600 pixels)

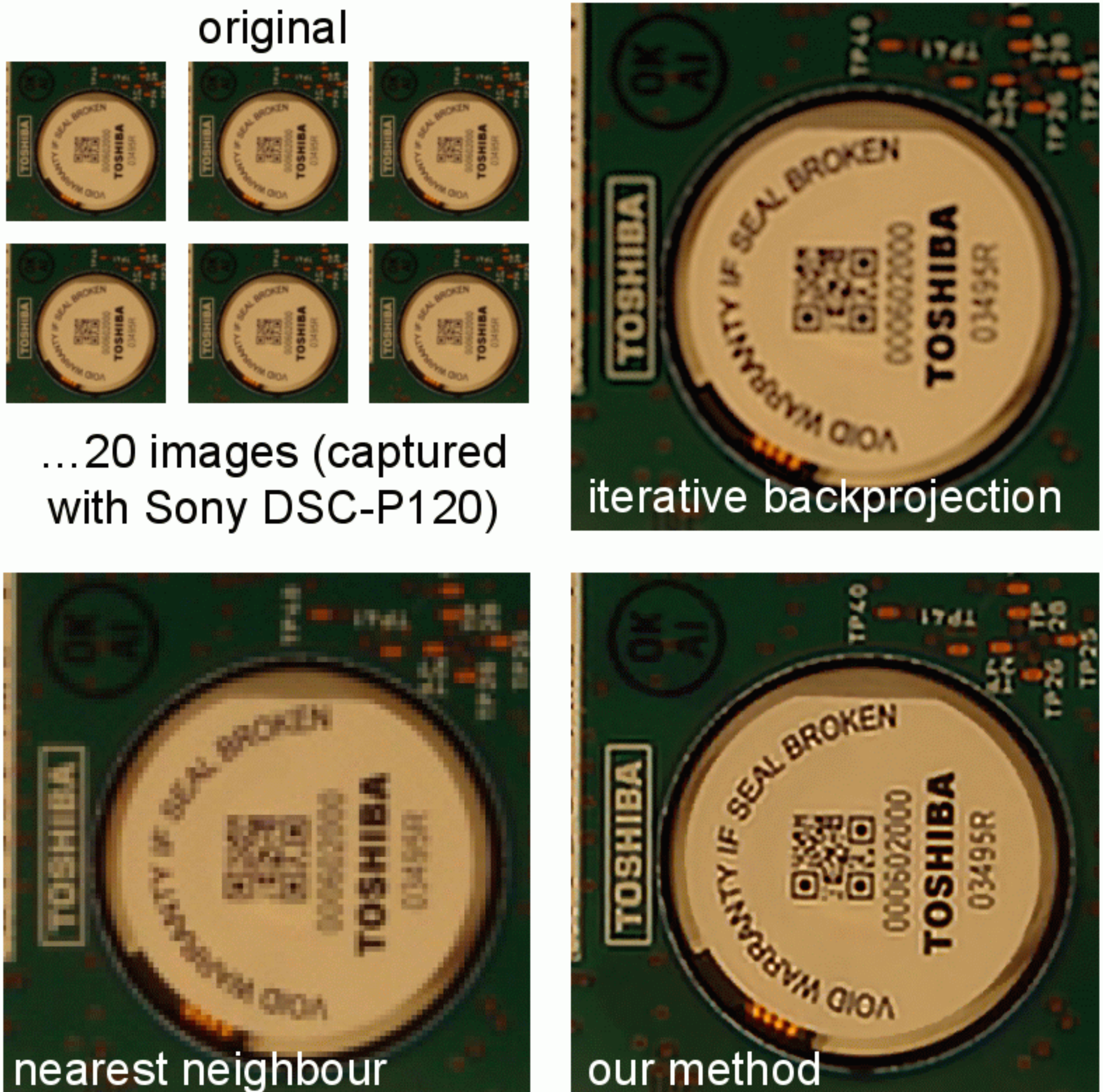


200
400

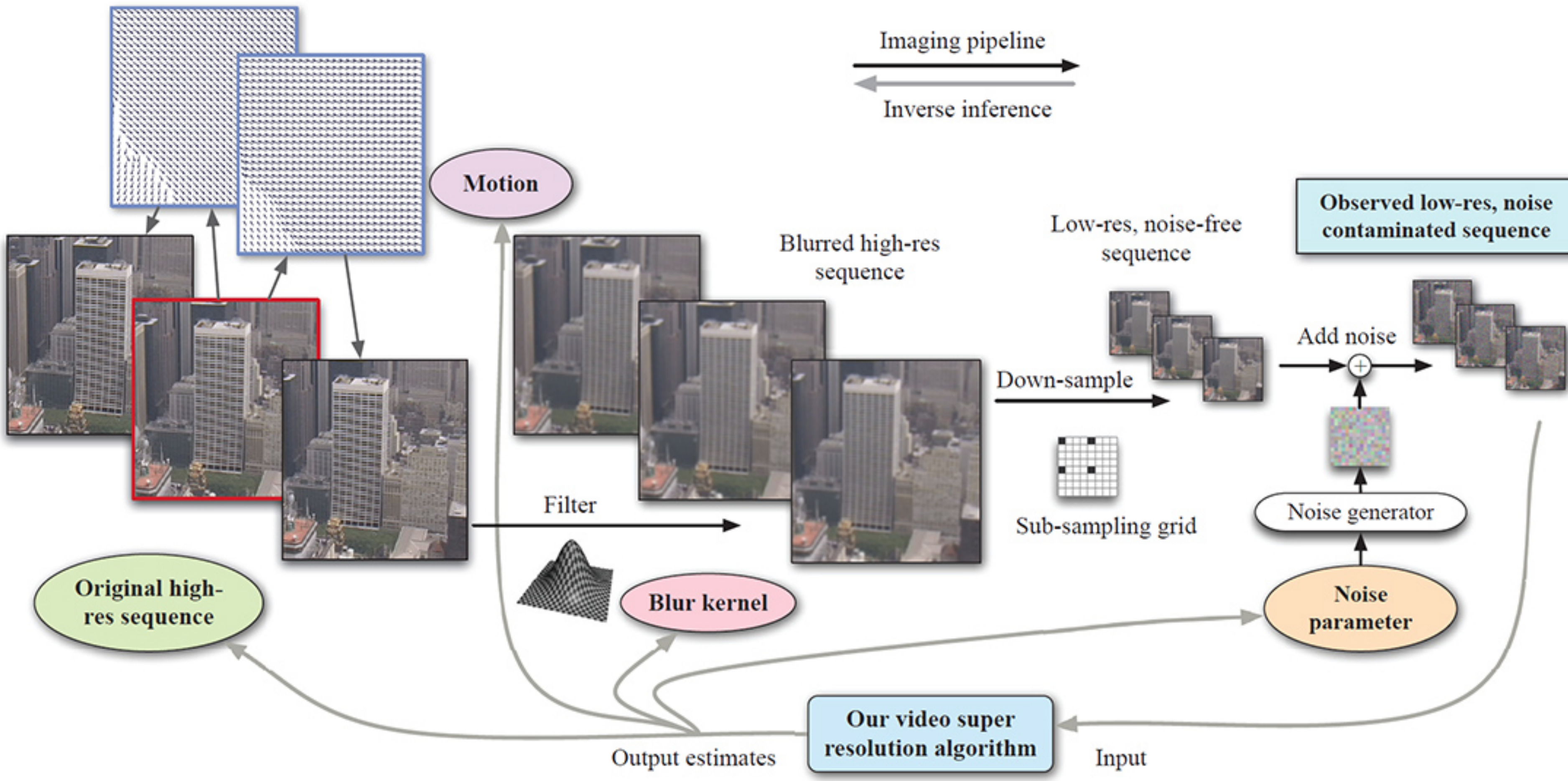

COLOR MAKES THE DIFFERENCE

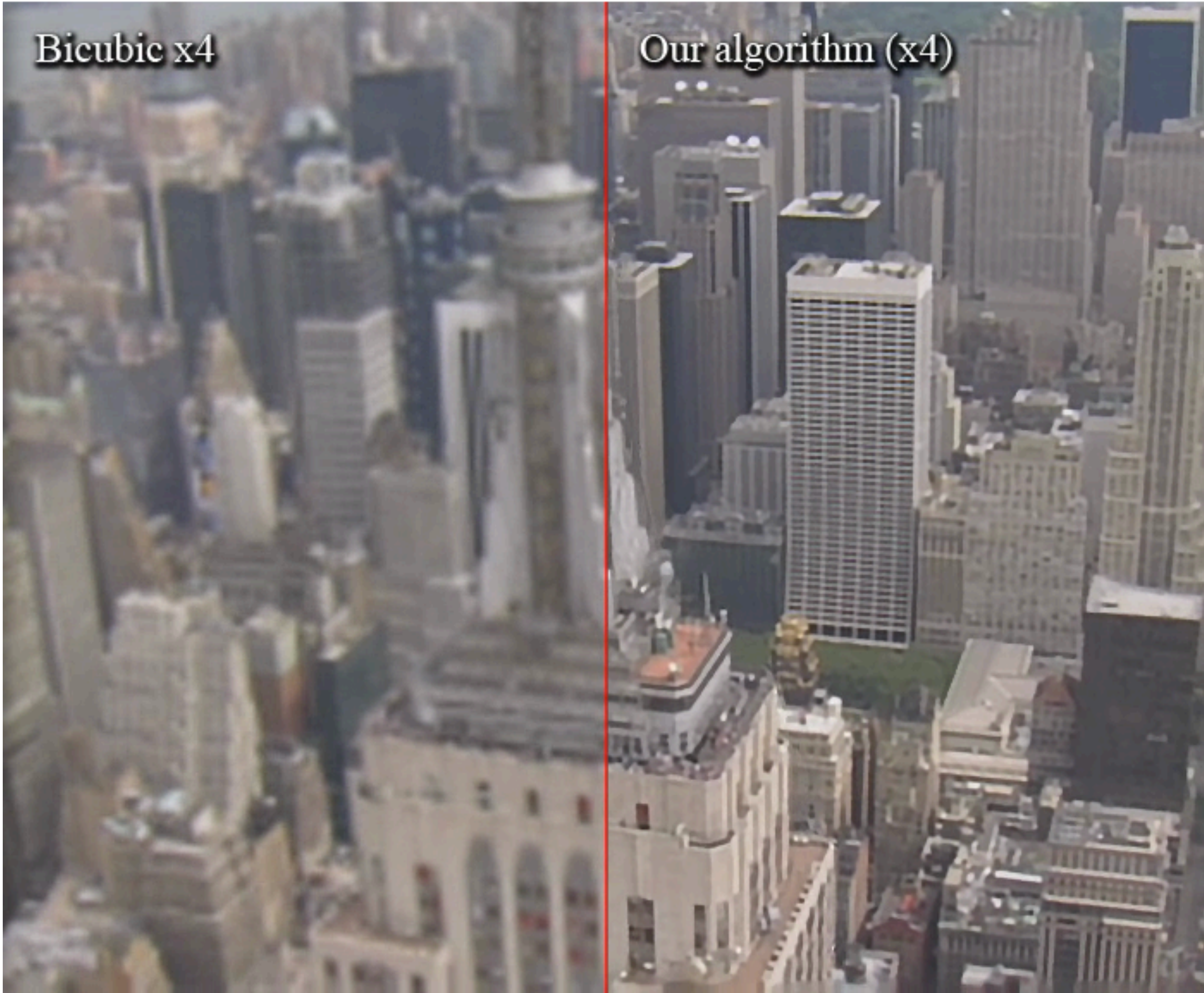

original
pixel replication (8x)
cubic B-spline (8x)
iterative function systems (8x)
our method (8x)


original
nec turpis nec risus tristique porttitor. In dapibus nisl vel
fermentum gravida, ullamcorper at, tortor. Fusce id fel
placerat justo porttitor sem. Phasellus tincidunt, eros non
vehicula sem mi ut leo. Morbi in ligula at justo rhon
adipiscing ultrices leo. In quis sem et nisl aliquam ullam
eget ultricies sodales, nunc magna convallis nibh, vel he
mi. Integer at tortor in magna dapibus vulputate. Donec e
at leo. Nam auctor feugiat justo. Maecenas lacinia con
auctor libero vitae enim. Maecenas luctus, ante quis viv
mauris sed est. Fusce nulla. In orci eros, elementum et, sc
vel mauris. Pellentesque eu purus. Aliquam erat volutpat.
pixel replication (8x)
cubic B-spline (8x)
level curve mapping (8x)
our method (8x)


original
…20 images (captured with Sony DSC-P120)
iterative backprojection
nearest neighbour
our method

Liu & Sun. A Bayesian Approach to Adaptive Video Super Resolution. CVPR'11




Bicubic x4
Our algorithm (x4)

思考题

- 为什么图像缩放需要使用插值
- 向前插值和向后插值相比有何缺点
- 理解双线性插值的计算
- 理解靶点变形为何要进行两次插值