

Notes for Lab 1 (Image scaling)

1. Digital Image:

A gray-level digital image is defined as a function mapping from $Z^* \times Z^*$ to Z^* , where Z^* indicates non-negative integer values. For example a digital image with size = $m \times n$ can be represented as: $G(x,y) \in Z^*$. The value of $G(x,y)$ indicates the gray level of the pixel (x,y) , where x and y indicate the horizontal and vertical positions of the pixel in the image. Of course x can only take an integer value between 0 and $(m-1)$ and similarly y can only take an integer value between 0 and $(n-1)$.

2. Digital Image Scaling:

Given a digital image $G(x,y)$ with size = $m \times n$ ($x = 0..m-1$ and $y = 0..n-1$), scaling means changing its size from $m \times n$ to $m' \times n'$ and obtaining a new image $G'(u,v)$, where $u = 0..m'-1$ and $v = 0..n'-1$.

We set $q_w = m'/m$ as the width scaling ratio and $q_h = n'/n$ as the height scaling ratio of the original image G and the target image G' . If q_w or q_h is bigger than 1, it means the original image stretches in horizontal or vertical direction. If q_w or q_h is smaller than 1, the original image shrinks in horizontal or vertical direction.

The gray value of each pixel in the target image G' is decided based on the corresponding values in the original image G by using the following equation:

$$G'(u,v) = G(x,y), \text{ where } x = \frac{u(m-1)}{m \cdot q_w - 1} \text{ and } y = \frac{v(n-1)}{n \cdot q_h - 1} \quad (1)$$

Obviously in equation (1), it is possible that x and y are real values. For example: x can be 0.33 when $m = 2$, $u = 1$ and $q_w = 2$, and y can be 1.25 when $n=3$, $v = 5$ and $q_h = 3$. This example means the gray value of the pixel $(1,5)$ in the target image G' takes the value of the pixel $(0.33,1.25)$ in the original image G . But in image G , the value of $G(x,y)$ is defined only if x and y are both non-negative integers. Then the problem becomes: when x and y are not integer values, how to guess the value of $G(x,y)$ based on those existing values in G . Image interpolation methods, such as the nearest neighbour method, the bi-linear method and the bi-cubic methods, can be used to solve this problem.

3. Image Interpolation Methods

Nearest Neighbour Method

When x or y are not integers, we set $G(x,y) = G(x',y')$, where $x' = \text{ground}(x+0.5)$ and $y' = \text{ground}(y+0.5)$. The *ground* operation converts a real value into an integer value simply by deleting its fraction part. For example $\text{ground}(1.2)=1$ and $\text{ground}(2.7)=2$.

In the above example, the value of the pixel $(0.33, 1.25)$ is set to $G(0,1)$, because $ground(0.33+0.5) = 0$ and $ground(1.25+0.5)=1$.

Bi-linear Method

If x is not an integer, we set $x' = ground(x)$ and $f_x = x - x'$. Actually x' is the integer part of x and f_x is the fraction part of x . Similarly we set $y' = ground(y)$ and $f_y = y - y'$ if y is not an integer. Then the value of $G(x,y)$ is determined as:

$$\begin{aligned} G(x,y) &= (1-f_y)G(x,y') + f_yG(x,y'+1), \text{ where} \\ G(x,y') &= (1-f_x)G(x',y') + f_xG(x'+1,y') \text{ and} \\ G(x,y'+1) &= (1-f_x)G(x',y'+1) + f_xG(x'+1,y'+1) \end{aligned}$$

According to above equations, the value of $G(x,y)$ is determined based on four neighbourhood integer pixels (x',y') , $(x'+1,y')$, $(x',y'+1)$ and $(x'+1,y'+1)$.

Bi-cubic Method

In this method, the value of $G(x,y)$ is decided by 16 neighbourhood pixels located at

$$\begin{array}{cccc} (x'-1,y'-1), & (x'-1,y'), & (x'-1,y'+1), & (x'-1,y'+2), \\ (x',y'-1), & (x',y'), & (x',y'+1), & (x',y'+2), \\ (x'+1,y'-1), & (x'+1,y'), & (x'+1,y'+1), & (x'+1,y'+2), \\ (x'+2,y'-1), & (x'+2,y'), & (x'+2,y'+1), & (x'+2,y'+2) \end{array}$$

$$G(x,y) = (1/6)(R_1G(x,y'-1) + R_2G(x,y') + R_3G(x,y'+1) + R_4G(x,y'+2)),$$

$$G(x,y'-1) = (1/6)(S_1G(x'-1,y'-1) + S_2G(x',y'-1) + S_3G(x'+1,y'-1) + S_4G(x'+2,y'-1)),$$

$$G(x,y') = (1/6)(S_1G(x'-1,y') + S_2G(x',y') + S_3G(x'+1,y') + S_4G(x'+2,y')),$$

$$G(x,y'+1) = (1/6)(S_1G(x'-1,y'+1) + S_2G(x',y'+1) + S_3G(x'+1,y'+1) + S_4G(x'+2,y'+1)),$$

$$G(x,y'+2) = (1/6)(S_1G(x'-1,y'+2) + S_2G(x',y'+2) + S_3G(x'+1,y'+2) + S_4G(x'+2,y'+2)),$$

$$R_1 = (3+f_y)^3 - 4(2+f_y)^3 + 6(1+f_y)^3 - 4(f_y)^3,$$

$$R_2 = (2+f_y)^3 - 4(1+f_y)^3 + 6(f_y)^3,$$

$$R_3 = (1+f_y)^3 - 4(f_y)^3,$$

$$R_4 = (f_y)^3,$$

$$S_1 = (3+f_x)^3 - 4(2+f_x)^3 + 6(1+f_x)^3 - 4(f_x)^3,$$

$$S_2 = (2+f_x)^3 - 4(1+f_x)^3 + 6(f_x)^3,$$

$$S_3 = (1+f_x)^3 - 4(f_x)^3,$$

$$S_4 = (f_x)^3$$

4. Reference

http://en.wikipedia.org/wiki/Bicubic_interpolation

http://en.wikipedia.org/wiki/Bilinear_interpolation

ftp://131.104.49.49/cis4720/lab_1/interpolation.pdf

ftp://131.104.49.49/cis4720/lab_1/Survey_Interpolation.pdf