夏显茁 15331322

1. Exercises

1.1 Storage:

1. $128 = 2^7$, So there should be 7 bit planes in this image.
2. 7^{th} panel is the most visually significant one.
3. $\frac{1024*2048*128}{8} = 33554432 = 2^{25}\ Byte$.

1.2 Adjacency

i. 4-path does not exist. Because None of the elements in

ii. V = {1, 2, 3} locates in q's 4's area.

iii. The shortest length of 8-path is 4.

iv. The shortest length of m-path is 5.

1.3 Logical Operations

i. $\boldsymbol{A} \cap \boldsymbol{B} \cap \boldsymbol{C}$

ii. $(\boldsymbol{A} \cap \boldsymbol{B}) \cup (\boldsymbol{A} \cap \boldsymbol{C}) \cup (\boldsymbol{B} \cap \boldsymbol{C})$

iii. $(\boldsymbol{B} - (\boldsymbol{A} \cup \boldsymbol{C})) \cup (\boldsymbol{A} \cap \boldsymbol{C} - \boldsymbol{A} \cap \boldsymbol{B} \cap \boldsymbol{C})$

2. Programming Tasks

Original Image

2.2 Scaling

1. Down-scale to 192 × 128 (width: 192, height: 128), 96 × 64, 48 × 32, 24 × 16 and 12 × 8:

192×128

96×64

48×32

24×16

12×8

2. Down-scale to 300×200:

300×200

3. Up-scale to 450×300:

450×300

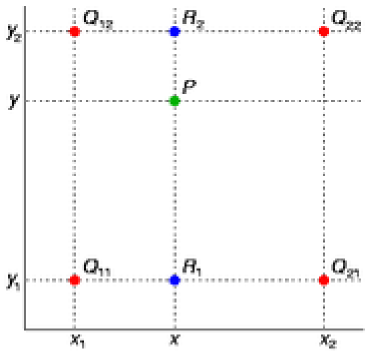
4. Scale to 500 $\times$200:

500×200

5. Detailedly discuss how you implement the scaling operation:

   为了让人看懂，还是用中文。

   a) 双线性插值原理：



图中：红色点为原图像的点；绿色点为带插值点

首先在 x 方向进行插值，得到R_1和R_2两点:

$$f(R_1) \approx \frac{x_2-x}{x_2-x_1} f(Q_{11}) + \frac{x-x_1}{x_2-x_1} f(Q_{21})$$

$$f(R_2) \approx \frac{x_2-x}{x_2-x_1} f(Q_{12}) + \frac{x-x_1}{x_2-x_1} f(Q_{22})$$

然后在 y 方向进行插值，得到 P 点:

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2)$$

所以，所求的结果$f(x,y)$:

$$f(P) \approx \frac{(x_2 - x)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} f(Q_{11}) + \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} f(Q_{21}) + \frac{(x_2 - x)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} f(Q_{12}) + \frac{(x - x_1)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} f(Q_{22})$$

在实际使用中，x_1和x_2，y_1和y_2均相差为 1

即$(x_2 - x_1)(y_2 - y_1) = 1$，所以原式可以化简为

$$f(P) \approx (x_2 - x)(y_2 - y) f(Q_{11}) + (x - x_1)(y_2 - y) f(Q_{21}) + (x_2 - x)(y - y_1) f(Q_{12}) + (x - x_1)(y - y_1) f(Q_{22})$$

为减少变量便于计算，使用

u来代替$(x_1 - x)$——u实际为 P 点横坐标的小数部分

v来代替$(y_1 - x)$——u实际为 P 点纵坐标的小数部分

所以，得到的最终公式为：

$$f(P) \approx (1 - u)(1 - v) f(Q_{11}) + u(1 - v) f(Q_{21}) + (1 - u) v f(Q_{12}) + u v f(Q_{22})$$

b) 实现

i. 由于在 scaling 的过程中，会发生待插值点出现在边界的情况。为了使用通用的方法来处理所有带插值点，在插值前，为原矩阵添加外边界。详见代码 extendBorder(img)->NewImg

ii. 通过 scaling 前后的宽高对应比例，来寻找带插值点。详见代码 scale(img, size)->NewImg

2.3 Quantization

1. Reduce grey level resolution to 128, 32, 8, 4 and 2 levels:

128 grey level resolution

32 grey level resolution

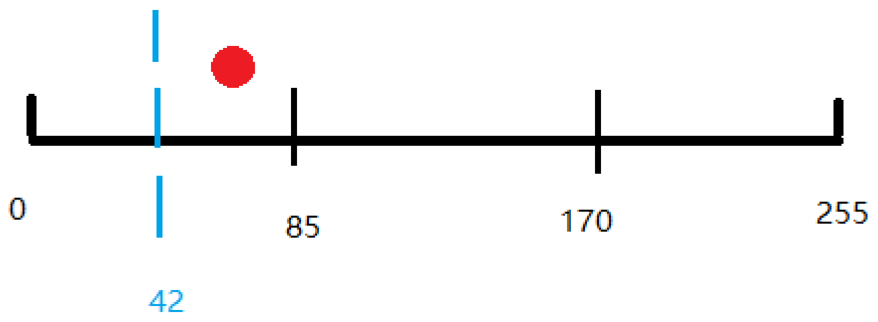
8 grey level resolution

4 grey level resolution

2 grey level resolution

2. Detailedly discuss how you implement the quantization operation

原理



图为降低到 4 个灰度级时，算法的解释用图

图中，我们将 256 个灰度级别降低为 4 个灰度级别，

[0, 255]的区间划分为 3 个区间(0, 85), (85, 170)和(170, 255)，

红点为灰度值落在(0,85)间的某一个灰度值。

对于落在相应区间的灰度值，采取就近取值的原则。

例如图上的红点，距离 85 比距离 0 要近，

所以在量化时该灰度值会被统一为 85 而不是 0。

实现：

将区间按照所给灰度级划分成相应的份数（可能会不均匀，

区间长度进行向下取整）。判断灰度值距离哪个区间端点较近，

采用作差取绝对的方法进行判断。

详见代码 quantize(img, level) ->NewImg