

# 1 Exercises

---

## 1.1 storage

---

### answer 1:

There are totally 7 bit planes in this image.

### answer 2:

There plane 7 is the most visually significant one.

### answer 3:

At least  $7 \cdot (2^{18})$  bytes are required for storing this image.

## 1.2 Adjacency

---

The 4-path between q and p dose not exist because the pixels in  $N_4(q)$  do not belong to the set of  $V=\{1,2,3\}$ .

The lengths of shortest 8-path is 4 and of m-path is 5.

## 1.3

---

Subfig 1:

$$A \cap B \cap C$$

Subfig 2:

$$(A \cap B) \cup (A \cap C) \cup (B \cap C)$$

Subfig 3:

$$(B \cup (A \cap C)) - (A \cap B) - (B \cap C)$$

# 2 Programming Tasks

---

## 2.2 Scaling

## answer 1-4:

Please look up the attached file for the scaled results.

## answer 5:

在实现缩放的过程中，采用了双线性插值的缩放算法。双线性插值的基本数学模型可以归纳为：若图像为灰度图像，那么目标图像(i,j)像素所对应的原图像(x, y)点的像素值有以下公式：

$$f(x, y) = b1 + b2 * x + b3 * y + b4 * xy \text{ (式1)}$$

其中，系数b1-b4由(x,y)点周围像素求得 所以根据该数学模型，我在函数中进行了如下几步操作：

(1) 计算要计算出目标图像中(i,j)像素对应在原图像的像素(x,y)的位置，其中通过如下公式求出：

$$x = i * m / a \text{ (式2)}$$

$$y = j * n / b \text{ (式3)}$$

(2) 再根据以下四个公式求出与之最相邻四个像素点的位(x0,y0),(x0,y1),(x1,y0),(x1,y1)

$$x_0 = \lfloor x \rfloor \text{ (式4)}$$

$$y_0 = \lfloor y \rfloor \text{ (式5)}$$

$$x_1 = \min(x_0 + 1, \text{srcwidth} - 1) \text{ (式6)}$$

$$y_1 = \min(y_0 + 1, \text{srcheight} - 1) \text{ (式7)}$$

(3) 根据公式1的恒等变换成两点式方程形式，并用两点式解法分两步求解像素值

### 一些有趣发现

1、一开始图像缩放结果与常用函数库缩放结果不同，经查阅相关资料，第一步过程中采用公式

$$x = (i + 0.5) * m / a - 0.5$$

$$y = (j + 0.5) * n / b - 0.5$$

分别代替函数

$$x = i * m / a$$

$$y = j * n / b$$

该公式替换使原图像和目标图像是以几何中心作为原点对应,可以更充分利用每个像素，这是opencv和Matlab中的双线性插值函数的常用方法。

2、同时在放大时候，发现了一个有趣的现象，放大输出的图像在最右侧和最下方分别出现了一个像素点宽度或高度的黑线，经分析发现，是因为最右（下）侧目标图像像素对应的原图像像素四个像素

中只存在两个像素，并因为公式6和公式7限制，计算出来四个像素坐标为两组相同坐标，在计算过程中相互抵消，导致最终计算结果为零。

解决办法：在算法设计过程中加入判断，如计算的目标像素为右侧和下放的边缘像素点，则双线性插值降维成只进行一个维度的线性计算，避免产生0值。

## 2.3 Quantization

---

### answer 1:

Please look up the attached file for the quantized results.

### answer 2:

量化函数主要采用了线性量化的算法，主要目标是寻找量化函数：

$$y = [(kx + b) + 0.5]$$

再将量化后结果进行映射，方便图像显示即可。因为采取正相关量化，所以可以通过输入的level值获得(0,0)和(255,level-1)两点解出系数k,b值，再通过映射表完成显示像素值的映射即可。

