

Part 1

1.1 Storage (3 * 3 = 9 Points)

1. How many bit planes are there for this image?
2. Which panel is the most visually significant one?
3. How many bytes are required for storing this image? (Don't consider image headers and compression.)

Answer:

1. There're 7 planes for this image.
2. The plane 7 is the most visually significant one.
3. $1024 * 2048 * 7 / 8 = 1835008$ bytes

1.2 Adjacency (3 * 3 = 9 Points)

Answer:

1. There's no 4-path between p and q, because there is no value which is from $V=\{1,2,3\}$ in the 4-neighbour of the q.
2. The length of the 8-path is 4.
3. The length of the m-path is 5.

1.3 Logical Operations (3 * 3 = 9 Points)

Answer:

1. $A \cap B \cap C$
2. $(A \cap C) \cup (A \cap B) \cup (B \cap C)$
3. $[B \cap (A \cap B)^c \cap (B \cap C)^c] \cup [(A \cap C) \cap B^c]$

Part 2

2.2 Scaling (45 Points)

Answer:

1.
 $192 * 128:$



96*64



48*32



24*16



12*8



2.
300*200



3.
450*300



4.
500*200



5. 在完成 scale 函数时，使用的是双线性插值的算法。

对于给定的目标图像的长宽比例，首先用源图像的长宽(length,width)分别除以目标图像的长宽(src_length,src_width)，得出长比(length_rate)跟宽比(width_rate)。然后对于目标图像的每个像素点的位置(x,y)乘以刚才得到的长比宽比得到对应于原图的浮点坐标(i+u, j+v)

(ps: 这里的 i, j 为坐标的整数部分，而 u、v 则为小数部分，范围是[0,1)，
 $u = \text{double}(x * \text{length_rate}) - \text{uint8}(x * \text{length_rate})$, v 同理可得。)

然后根据双线性插值公式

$$f(i+u, j+v) = (1-u)(1-v)f(i, j) + (1-u)v f(i, j+1) + u(1-v)f(i+1, j) + uv f(i+1, j+1)$$

可得到浮点坐标 f(i+u, j+v) 的像素值，也就是目标图像位置(x,y)的像素值。

过程要注意到对于边界时 f(i, j+1)、f(i+1, j)、f(i+1, j+1) 出现越界，此时对于边界做以下处理：

```
if i >= src_col;
    i = src_col - 1;
end
if j >= src_row;
    j = src_row - 1;
end
```

2.3 Quantization (28 Points)

Answer:

1.

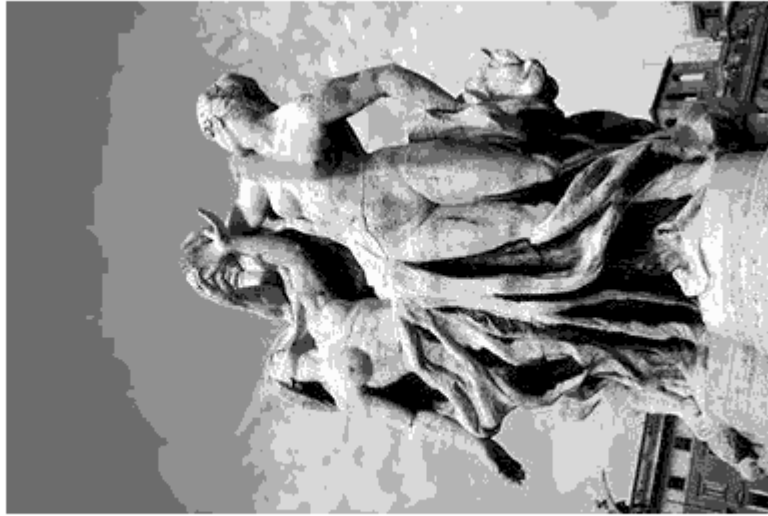
128-level



32-level



8-level



4-level



2-level



2. 这里我采取的是均匀量化，即将 0-255 划分为 n 个均等的区间，根于原图像各像素点处于哪个区间决定目标图像该位置像素值对应的值。

这里以 4level 为例：

原图像	目标图像
0-63	0
64-127	85
128-191	170
192-255	255

以下是算法的步骤：

- ①对于给定的目标图像的 level n，首先用 $256/n$ 得出每个区间的长度 range。
- ②由题目例子我认为映射区间的值从 0 开始，因此对于不同的 level，区间映射的值的间隔为 $range_length = 255/(n-1)$ 。
- ③遍历原图所有像素点灰度值，用 $f(x,y)/range$ 判断该点处于哪个区间。
- ④最后根据不同区间给目标图像的像素点赋值 $F(x,y) = f(x,y)/range * range_length$ 。