数图第一次作业

数图作业
郭柱明 15331094

1

1.1

If we consider an N-bit gray image as being composed of N 1-bit planes, with plane 1 containing
the lowest-order bit of all pixels in the image and plane N all the highest-order bits, then given
a 1024 * 2048, 128-level gray-scale image:

1. How many bit planes are there for this image?

   答：需要7位，因为2^7 = 128

2. Which panel is the most visually signicant one?

   答：第N层，就是最高位组成的层对视觉影响最大

1. How many bytes are required for storing this image? (Don't consider image headers and compression.)

   答：1024 * 2048 * 7 = 14680064

1.2

Figure 1 is a 5 * 5 image. Let V = f1; 2; 3g be the set of pixels used to dene adjacency. Please
report the lengths of the shortest 4□, 8□, and m□path between p and q. If a particular path
does not exist, explain why.

3	4	1	2	0
0	1	0	4	2(q)
2	2	3	1	4
(p)2	0	4	2	1
1	2	0	3	4

Figure 1: *Adjacency.*

lengths of the shortest 4-path

答：不存在，因为q的上，左，下分别是0，4，4,无法形成一条从p到q的只包含{1, 2, 3}的4-path

lengths of the shortest 8-path

答：存在，如下图，长度为4

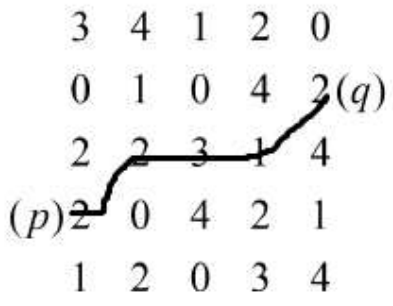


Figure 1: *Adjacency.*

lengths of the shortest m-path

答：存在，如下图，长度为5

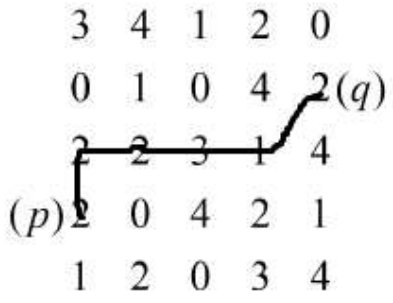
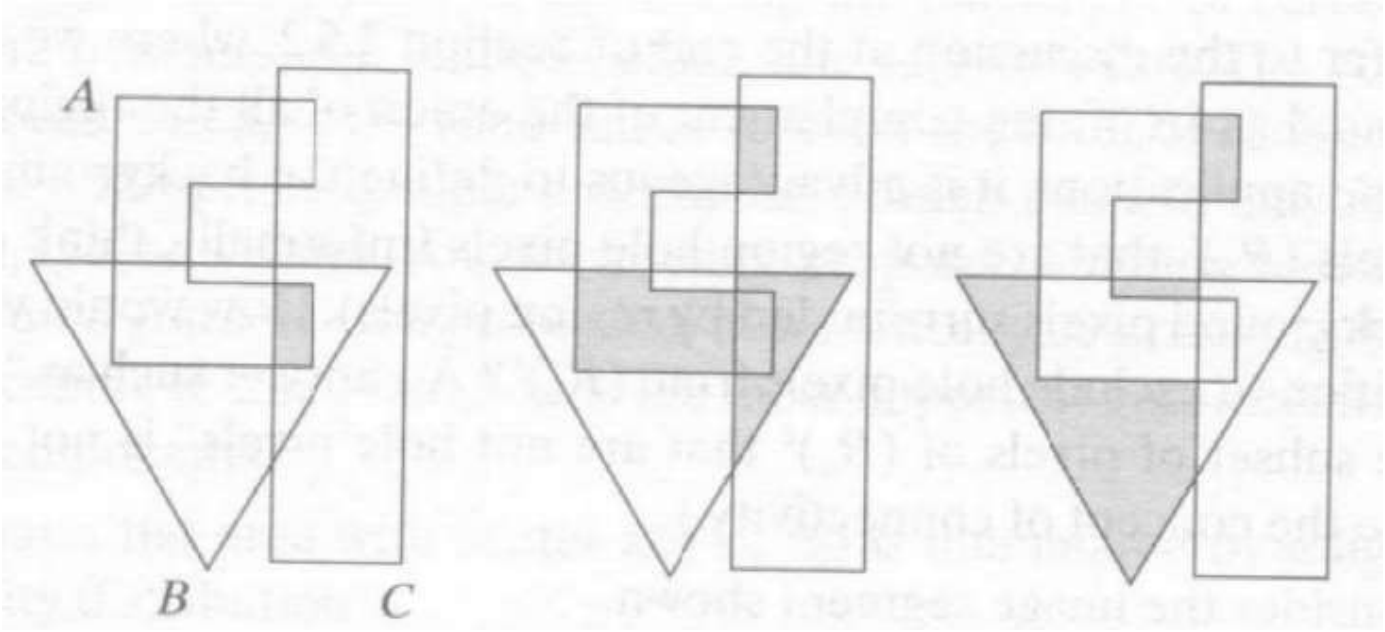


Figure 1: *Adjacency.*

1.3

Figure 2 are three di□erent results of applying logical operations on sets A;B and C. For each of
the result, please write down one logical expression for generating the shaded area. That is, you
need to write down three expressions in total.



答：
图1：

$$A \bigcap B \bigcap C$$

图2：

$$(A \bigcap B) \bigcup (B \bigcap C) \bigcup (A \bigcap C)$$

图3：

$$(B - (A \bigcup C)) \bigcup (A \bigcap C))$$

2

2.1

答：下图是我的输入图片

2.2

1. Down-scale to 192 * 128 (width: 192, height: 128), 96 * 64, 48 * 32, 24 * 16 and 12 * 8, then manually paste your results on the report.

   答：192 * 128如下：

   96 * 64如下：

   48 * 32如下：

   12 * 8如下：

1. Down-scale to 300 * 200, then paste your result.

   答：

1. Up-scale to 450 * 300, then paste your result.

答：

1. Scale to 500 * 200, then paste your result.

答：

1. Detailedly discuss how you implement the scaling operation, i.e., the \scale" function, in less than 2 pages. Please focus on the algorithm part. If you have found interesting

phenomenons
in your scaling results, analyses and discussions on them are strongly welcomed and may bring you bonuses. But please don't widely copy/paste your codes in the report, since your codes are also submitted.

答：

1. 首先先用python的PIL中的Image类打开原始图片，生成灰度值数组，
2. 再申明定义一个尺寸为目标尺寸全为0的目标矩阵，遍历目标矩阵中的每一个点，使用公式
$srcX = dstX * (srcWidth/dstWidth), srcY = dstY * (srcHeight/dstHeight)$
即：

```
1. dst[dst_x, dst_y] = (1 - u) * (1 - v) * q1 + (1 - u) * v * q3 + u * (1 - v) * q2 + u * v * q4
```

求出目标数组中的点对应原始数组中的点的横坐标和纵坐标，然后使用**双线性插值法**来求这个原始数组中点的值，最后根据目标数组生成目标图片

2.3

1.

答：各个灰度范围的结果如下：
128如图：

32如图：

8如图：

4如图：

1. 答：

    1. 首先先用python的PIL中的Image类打开原始图片，生成灰度值数组，
    2. 再跟据参数中的level值生成一个灰度值列表，例如当level是4的时候，gray_list为
    $$[0, 63, 127, 191, 255]$$
    3. 遍历原始数组中的每一个点，那他的灰度值去跟gray_list的比较，求得差值绝对值最小的那个灰度，则目标数组中对应的那一点的灰度值即为此灰度值
    4. 由目标数组生成目标图片，保存。