

1.1 Storage

1. 7
2. plane 7
3. $1024 * 2048 * 7 / 8 = 1835008$ bytes

1.2 Adjacency

根据 4 邻接的定义，从 p 点到 q 点之间不存在 4 通路；存在 8 通路，最短长度为 4。路径为：2-2-3-1-2；存在 m 通路，最短长度为 5。路径为：2-2-2-3-1-2。

1.3 Logical Operations

1. $A \cap B \cap C$
2. $(A \cap B) \cup (B \cap C) \cup (A \cap C) - 2(A \cap B \cap C)$
3. $(A \cap C) \cup (B - A \cap B) - B \cap C$

2.2 Scaling

1.

缩放后图像：192*128



缩放后图像：96*64



缩放后图像：48*32



缩放后图像：24*16



缩放后图像：12*8



2.

缩放后图像：300*200



3.

缩放后图像：450*300



4.

缩放后图像：500*200



5. 算法说明：

对图片的缩放是通过双线性差值法实现的。具体算法如下：对于一个目的像素，设置坐标通过反向变换得到的浮点坐标为 $(i+u, j+v)$ (其中 i 、 j 均为浮点坐标的整数部分， u 、 v 为浮点坐标的小数部分，是取值 $[0,1)$ 区间的浮点数)，则这个像素得值 $f(i+u, j+v)$ 可由原图像中坐标为 (i, j) 、 $(i+1, j)$ 、 $(i, j+1)$ 、 $(i+1, j+1)$ 所对应的周围四个像素的值决定，即：

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

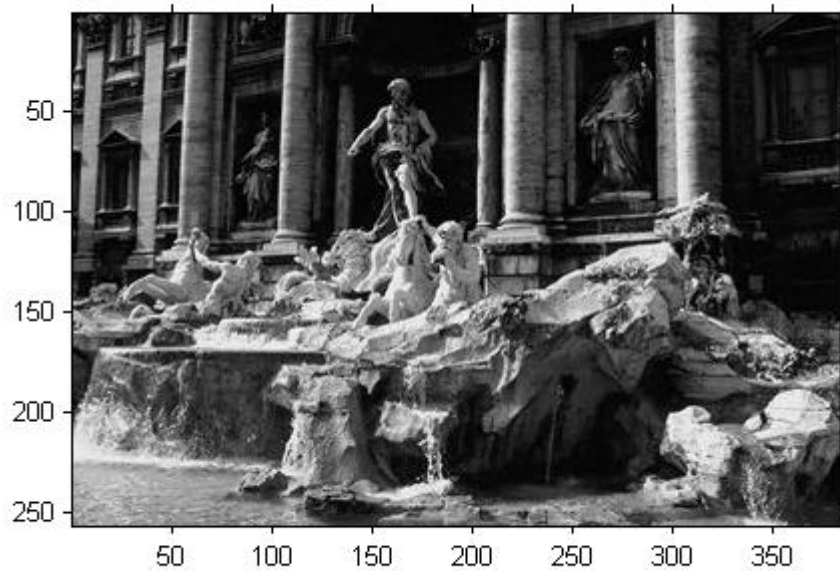
其中 $f(i, j)$ 表示源图像 (i, j) 处的像素值。

通过对目标图像的每个像素点进行如上运算估值即可。对于边界像素点，我是通过拓宽了原图像对应矩阵来避免越界。

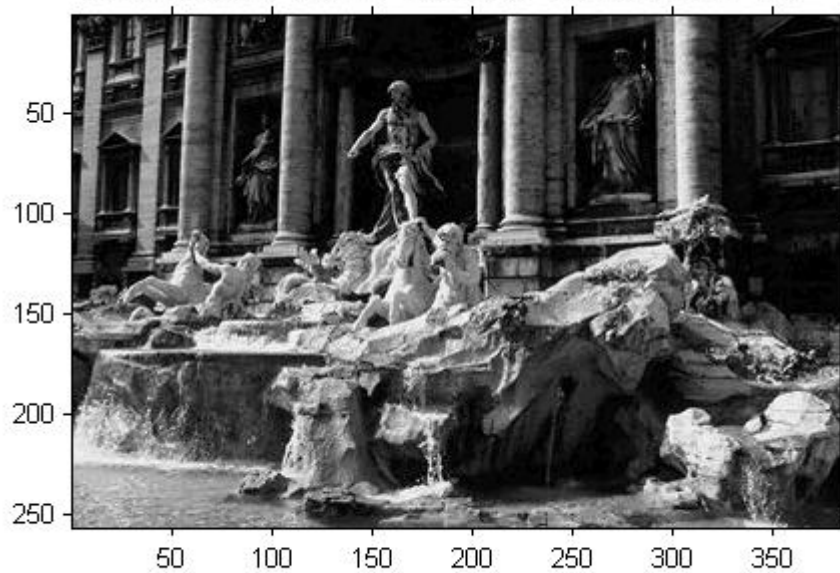
2.3 Quantization

1.

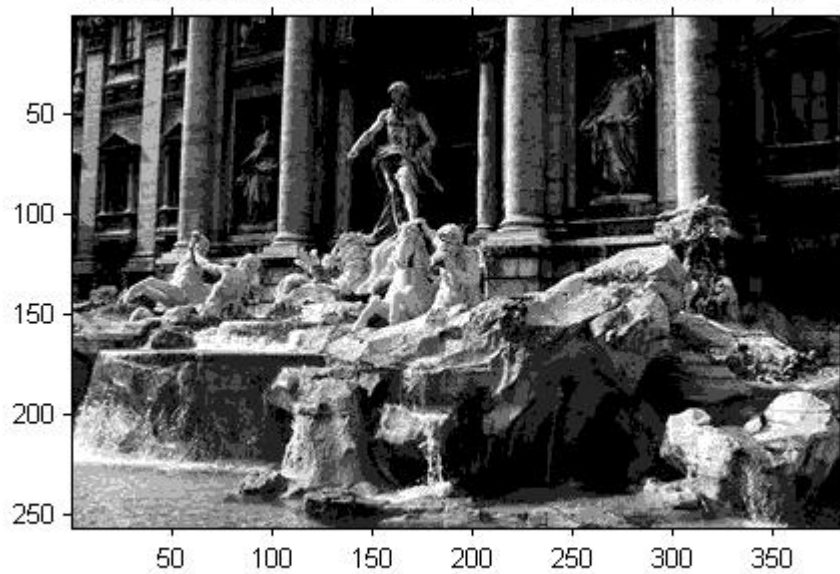
量化后的图像（大小：384*256*，灰度分辨率：128）



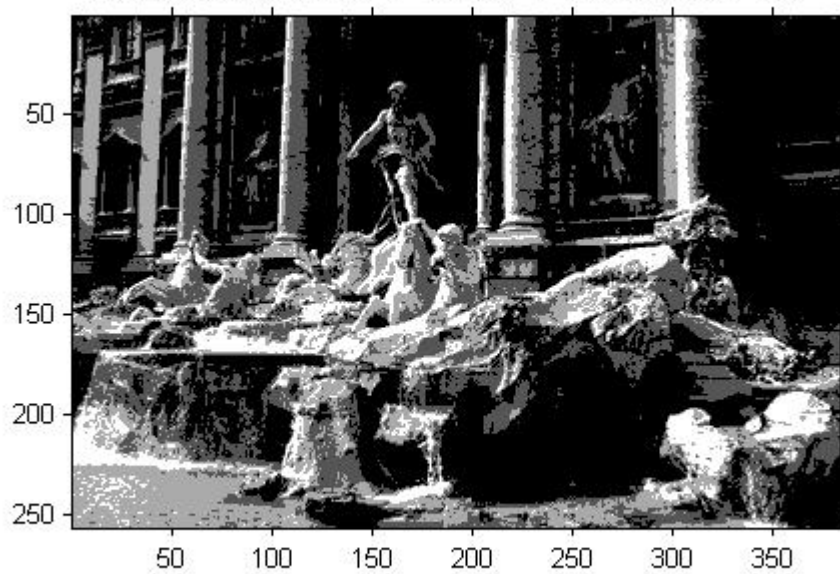
量化后的图像（大小：384*256*，灰度分辨率：32）

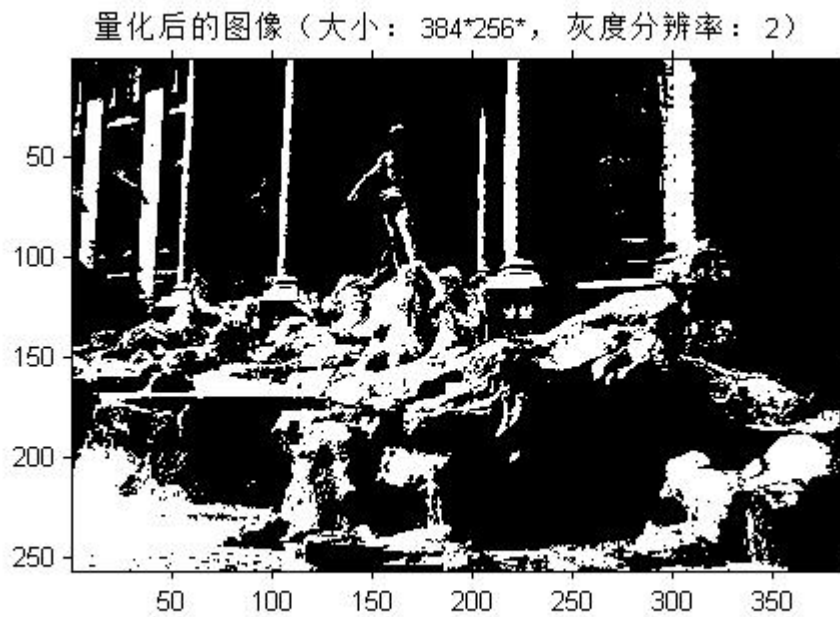


量化后的图像（大小：384*256*，灰度分辨率：8）



量化后的图像（大小：384*256*，灰度分辨率：4）





2. 算法说明：

量化算法很简单，就是通过将原图像 256 个灰度值等分区域地映射到要求下降的灰度分辨率上的某个值上。比如要将灰度分辨率降为 4，那么就将原图像灰度值为 0—63 范围内的像素点的灰度值映射为 0，在 64—127 范围内的像素点的灰度值映射为 85，在 128—191 内的映射为 170，在 192—255 范围内的映射为 255。以此类推。

