**图像压缩与重建**

——《数字图像处理》第6次作业

116032910031

陈晓

**1 函数定义**

首先定义计算过程中需要多次使用的3个函数，功能分别为打印矩阵，四舍五入矩阵值，以及移位矩阵值。

**1.1打印矩阵**

//打印矩阵

void printMat(Mat src)

{

for (int i = 0; i < src.rows; i++) {

for (int j = 0; j < src.cols; j++) {

cout << src.at<double>(i, j) << " ";

}

cout << endl;

}

cout << endl;

}

**1.2 四舍五入矩阵值**

//对矩阵值进行四舍五入

void roundMat(Mat src)

{

for (int i = 0; i < src.rows; i++) {

for (int j = 0; j < src.cols; j++) {

src.at<double>(i, j) = round(src.at<double>(i, j));

}

}

}

**1.3 移位矩阵值**

//对矩阵值进行移位

//c：移位值

void shiftMat(Mat src, int c)

{

for (int i = 0; i < src.rows; i++) {

for (int j = 0; j < src.cols; j++) {

src.at<double>(i, j) += c;

}

}

}

**2 图像压缩**

对8×8矩阵进行JPEG标准的压缩。

**2.1 定义矩阵**

定义JPEG标准阵列和原始矩阵。代码如下：

//JPEG 标准阵列

Mat Z = (Mat\_<double>(8, 8) <<

16, 11, 10, 16, 24, 40, 51, 61,

12, 12, 14, 19, 26, 58, 60, 55,

14, 13, 16, 24, 40, 57, 69, 56,

14, 17, 22, 29, 51, 87, 80, 62,

18, 22, 37, 56, 68, 109, 103, 77,

24, 35, 55, 64, 81, 104, 113, 92,

49, 64, 78, 87, 103, 121, 120, 101,

72, 92, 95, 98, 112, 100, 103, 99);

//原始矩阵

Mat src = (Mat\_<double>(8, 8) <<

52, 55, 61, 66, 70, 61, 64, 73,

63, 59, 66, 90, 109, 85, 69, 72,

62, 59, 68, 113, 144, 104, 66, 73,

63, 58, 71, 122, 154, 106, 70, 69,

67, 61, 68, 104, 126, 88, 68, 70,

79, 65, 60, 70, 77, 68, 58, 75,

85, 71, 64, 59, 55, 61, 65, 83,

87, 79, 69, 68, 65, 76, 78, 94);

cout << "原始矩阵：" << endl;

printMat(src);

输出原始矩阵如图1所示。

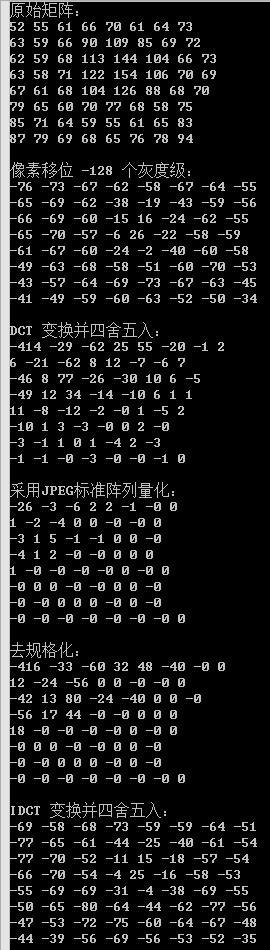


图1. 原始矩阵

**2.2 原始矩阵移位**

对原始矩阵移位 -128 个灰度级。代码如下：

src.copyTo(T);

shiftMat(T, -128);

输出移位后的矩阵如图2所示。

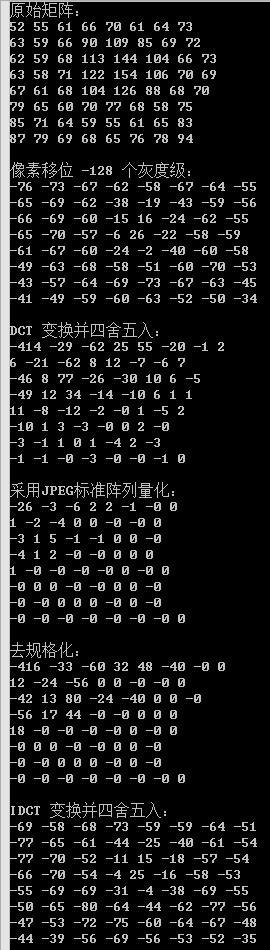


图2. 移位-128个灰度级

**2.3 DCT变换**

对移位后的矩阵进行DCT变换并四舍五入。代码如下：

dct(T, T);

roundMat(T);

输出变换后的矩阵如图3所示。

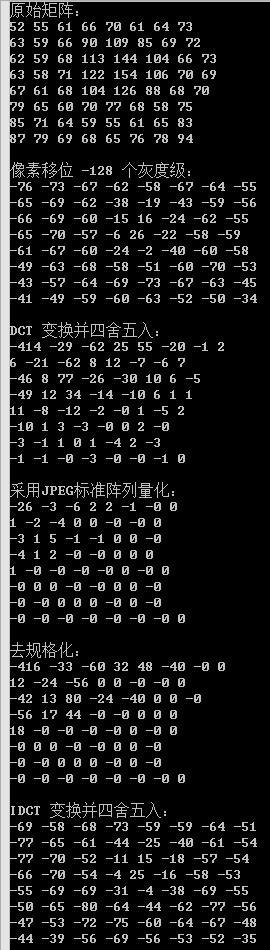


图3. DCT 变换

**2.4 量化**

采用 JPEG 标准阵列对变换矩阵进行量化：

代码如下：

divide(T, Z, T);

roundMat(T);

输出量化后的矩阵如图4所示。

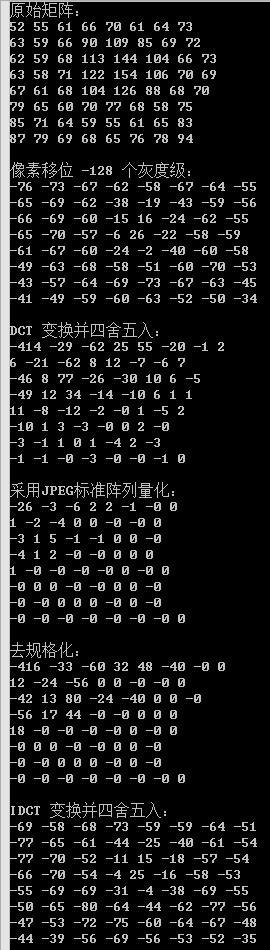


图4. 量化矩阵

**3 图像重建**

对矩阵进行 JPEG 标准的重建。

**3.1 去规格化**

对量化矩阵进行去规格化：

代码如下：

multiply(T, Z, T);

输出去规格化后的矩阵如图5所示。

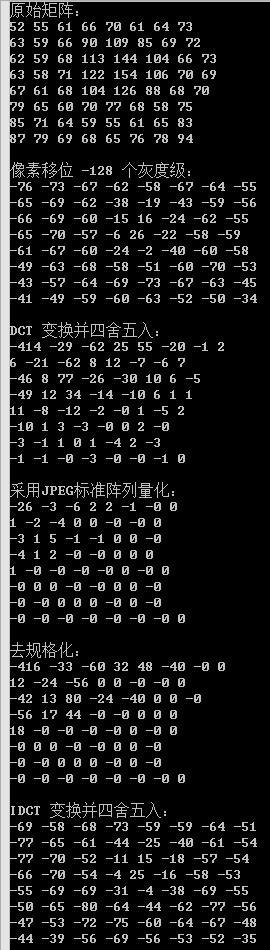


图5. 去规格化

**3.2 IDCT 变换**

对矩阵进行 IDCT 变换并四舍五入。代码如下：

idct(T, T);

roundMat(T);

输出 IDCT 变换后的矩阵如图6所示。

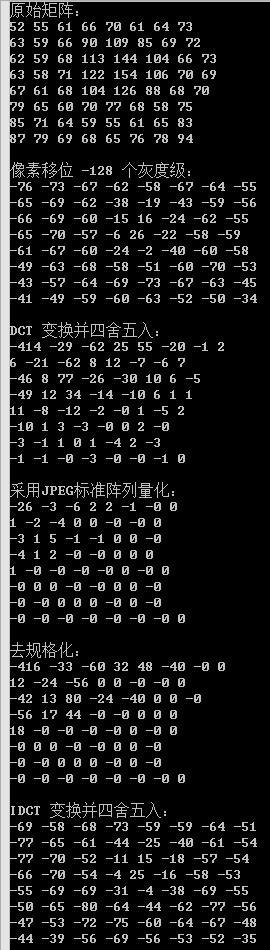


图6. IDCT 变换

**3.3 变换矩阵移位**

对IDCT变换后的矩阵移位 +128 个灰度级，得到重建矩阵。代码如下：

shiftMat(T, 128);

输出重建矩阵如图7所示。

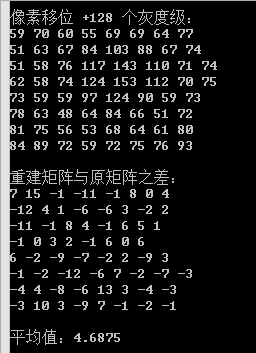


图7. 移位+128个灰度级

**4 统计误差**

将重建矩阵和原矩阵作差并统计误差绝对值的平均值。代码如下：

diff = T - src;

double m = mean(abs(diff))[0];

输出误差矩阵和误差绝对值的平均值如图8所示。

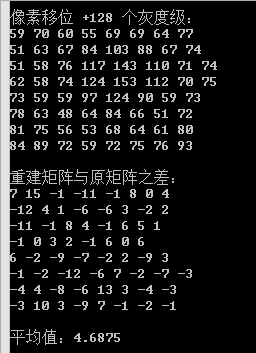


图8. 误差矩阵