

**数字图像处理实验报告**

实验二

题 目：熟悉DFT和DCT变换的基本原理

学 院 计算机学院

专 业 计算机科学与技术

学 号 14S003021

学 生 赵中祥

任 课 教 师 姚鸿勋(教授)

哈尔滨工业大学计算机科学与技术学院

2014.10.15

说明

实验报告一般包含以下几个部分：实验内容，实验目的，实验设计、算法和流程，实验结果，结论，参考文献。

注意：不要把所有代码都粘贴到实验报告中，如确有必要，请粘贴少量关键代码即可，源代码和可执行程序单独提交！

Notes: Experimental report usually includes following sections: experiment content, experiment purpose, experiment design、algorithm and procedure， experimental results, conclusion, reference.

Do not copy and paste all source code into the report, if necessary, please paste some key codes. At the same time, please submit the total source code and the executive program.

实验二 报告

1. 实验内容（contents）
   1. 熟悉图像的颜色空间
   2. 熟悉图像变换的思想
   3. 熟练掌握DCT变换
      1. 编程实现8\*8块DCT变换,然后分别显示，当使用64个DCT系数里面的一些系数进行重构，比较其质量；同时熟悉采用FFT来实现DCT变换的思想，并显示2维8\*8DCT变换的基函数图像
   4. 预习图像直方图的统计和显示
   5. 有余力的同学
      1. DCT变换的一个小应用-估计经历过JPEG压缩的BMP图像的压缩历史
2. 实验目的（purposes）

(1)补充课堂内容

(2)熟悉编程语言

(3)加深算法理解

(4)锻炼编程能力

(5)熟练掌握基本的图像处理操作及其基本的应用情况

1. 实验设计、算法和流程(Design, algorithm and procedure)
2. 在上次实验的基础上已经对BMP图像清晰掌握，然后对上次实验的SpaceConversion类进行扩展即可完成RGB 到YUV ，YIQ， HSI ，HSV ，XYZ等颜色空间的映射。

SpaceConversion：

void ToHSI(RGB \*img,HSI &img1,int h,int w,BITMAPINFOHEADER

infoheader); //RGB 到HSI

void ToYUV(RGB \*img,YUV &img2,int h,int ,BITMAPINFOHEADER

infoheader); //RGB 到YUV

void ToYIQ(RGB \*img,YIQ &img3,int h,int w,BITMAPINFOHEADER infoheader); //RGB 到YIQ

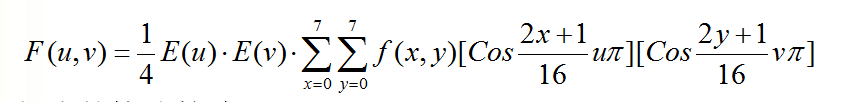
void ToHSV(RGB \*img,HSV &img4,int h,int ,BITMAPINFOHEADER infoheader); //RGB 到HSV

void ToXYZ(RGB \*img,XYZ &img5,int h,int ,BITMAPINFOHEADER infoheader); //RGB 到XYZ

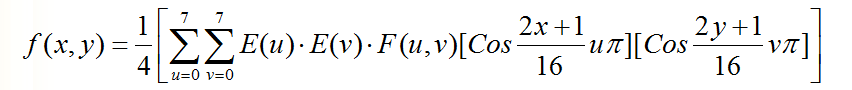
void HSItoRGB(RGB &img,HSI img1); //HIS 到RGB

(b)

8\*8DCT变换的原理(DCT):



8\*8DCT逆变换的原理：



这里我们模拟JPG图像压缩过程

同样将这个变化封装在一个类change中，定义函数如下：

void DCT(BYTE a[BLOCK][BLOCK],double b[BLOCK][BLOCK]);

//定义DCT正变化函数

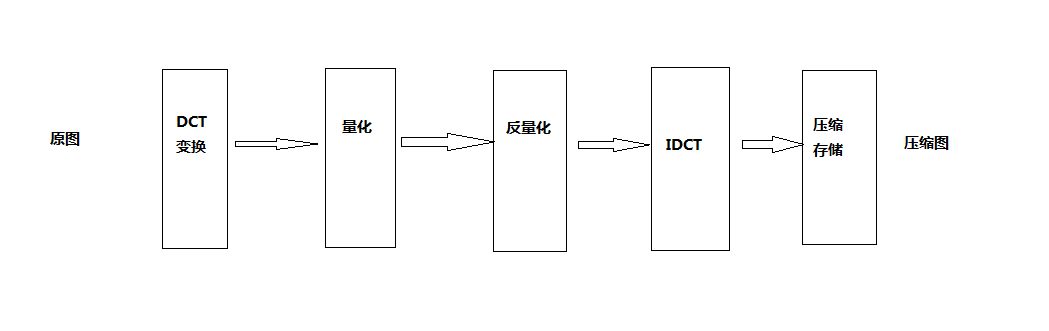
void IDCT(double a[BLOCK][BLOCK],BYTE b[BLOCK][BLOCK]);

//定义DCT你变化函数

void LH(double a[BLOCK][BLOCK],int b[BLOCK][BLOCK],const int c[BLOCK][BLOCK]); //定义量化过程

void LR(int a[BLOCK][BLOCK],double b[BLOCK][BLOCK],const int c[BLOCK][BLOCK]); //定义反量化过程

我们这里还是用bmp图像进行模拟，通过模拟理解其原理及过程：



(c)设计：

main()

{

1. 读取原图信息
2. 针对图像RGB分量进行DCT变化
3. RGB分量量化
4. RGB反量化
5. IDCT变换
6. 压缩存储图像，并生成图像

}

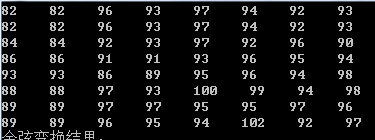
1. 实验结果(results)

1）实例lena

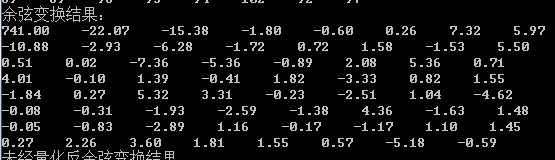


这里仅截取了第一个8\*8模块来观测整个过程：

(a)读取的R原值



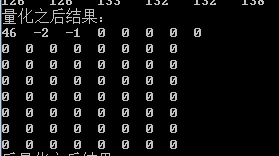
(b)DCT变换后：



(c)未量化直接进行IDCT变换：

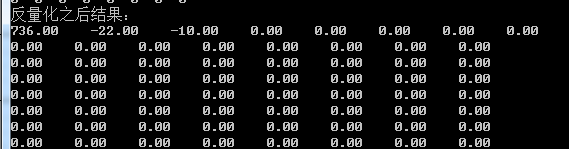


(d)量化后：

****

(e)反量化后：

反量化后得到一个稀疏矩阵，压缩就是在这一步进行的。

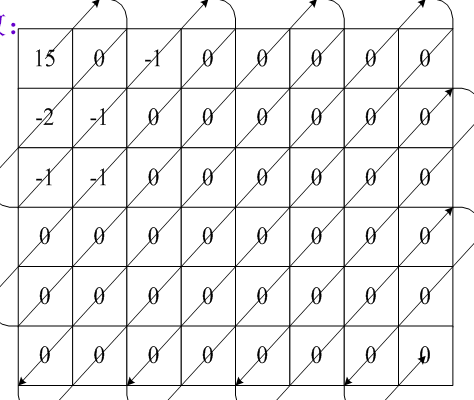


(f)量化后反余弦变换结果：



1.理论上读取的RGB值在DCT变换后和IDCT变换后没有损失信息，应该保持不变。比较(a)和(c)基本完全一样，可能中间的double型转换为整型丢失了一些信息。

2.在(e)因为这是一个系数矩阵，所以在这里可以进行压缩存储。



按照类似上图的方式进行扫描：

对于量化后的结果我们存储的时候，只需要记录两个不为0的数据中间有多少个0。

头信息：46，之后(-2,0)在之后(-1,3).

所以对于单纯存储这样一个8\*8数据来说，这样只需要存储5个数。压缩比为大概为13倍。还是很可观的。

3.那么经过量化反量化到底损失了多少信息，比较(c)和(f)发现数据的损失还是比较小的。所以这种压缩方式是可行的。

4)这是经过上述变换流程处理后的Lena图像：



和原图对比，实际信息损失很少，因此网络上采用jpg格式传输图像，可以很大程度上节省带宽。

实例2：

原图：

  
经DCT处理的图：



对比得出信息保存还是很完整的。

1. 结论(conclusion)
2. 本实验没有实际生成JPG图像，而是利用DCP压缩图像的原理来进行模拟一个过程。并对比原图和生成的图片，得出结论，DCP在压缩图像处理中信息保存完整度比较好。
3. DCP变换是一种特殊的傅里叶变换，通过复制使原信号为偶函数，（所以余弦函数里面的式子分母为2N）这样我们傅里叶变换时候只需要在实部进行计算。标准化后(即规范为离散图像变换通式)有利于我们针对量化矩阵对图像进行压缩。

如果不经过Uniform Even DCT处理，直接进行量化，得到的实例2的图像如下：



信息丢失严重。

1. 傅里叶变换中有快速傅里叶变换，同样我们可以在经过扩展的

2N个离散点进行快速傅里叶变换。变换之后利用下试：



只取其实部。

1. 参考文献(reference)

[1]R.C.Gonzalez Addison-Wesley, Digital Image Processing[M],北京,电子工业出版社,2007:367-395.

***Digital Image Processing***

**Mark sheet of experimentation**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Student No. | Name | Score of Part 1  (code) | Score of Part 2  (content) | Score of Part 3  (result) | Score of Part 4  （report） | Total Score |
| 1 |  |  |  |  |  |  |  |

Signature: Date: