

武汉理工大学毕业设计（论文）

有损压缩方法研究

学院（系）： 计算机科学与技术学院

专业班级： 软件工程软件 ZY1402 班

学生姓名： 叶哲宇

指导教师： 钟 珞

学位论文原创性声明

本人郑重声明: 所提交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外, 本论文不包括任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名:

年 月 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保障、使用学位论文的规定, 同意学校保留并向有关学位论文管理部门或机构送交论文的复印件和电子版, 允许论文被查阅和借阅。本人授权省级优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索, 可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

1、保密 , 在 年解密后适用本授权书

2、不保密

(请在以上相应方框内打“√”)

作者签名:

年 月 日

导师签名:

年 月 日

摘要

离散余弦变换 (DCT) 一直是图像技术中最主要的压缩算法, 因其算法简单、计算量小等特点在实际应用中获得了巨大成功。本文完成了 DCT 有损编码技术与行程无损编码技术的有效结合, 在保证一定程度视觉质量的条件下实现了高压缩比的图像压缩。并在传统 JPEG 中 DCT 编码算法的基础上提出了基于阈值分割的改进算法。

分形编码利用图像自相似性, 在局部迭代函数系统中遍历每个值域块, 搜索与其均方误差 (MSE) 最低的定义域块, 生成仿射变换码本完成编码, 并通过拼贴定理实现迭代解码。但分形编码因其编码步骤冗长、计算复杂度高等缺点在实际应用缺乏竞争力。

本文将离散余弦变换以及分形编码结合到一起, 提出了一种混合编码方法。在图像分割后, 对数值差异不大的平坦子块进行 DCT 变换, 对缩小后的值域块池进行分形编码有效缩小了搜索时间。同时对图像重分割, 对二次变化后的子块对恢复图像进行补偿以消除块效应。实验结果表明, 本文提出的改进算法在编码时间上对分形编码有了一定程度的改善, 在图像恢复质量上也得到了提升。

关键词: 图像压缩编码; 离散余弦变换; 阈值分割; 分形; 迭代函数系统

Abstract

Discrete cosine transform (DCT) has always been the most important compression algorithm in photographic compressing and coding method. It has got great success in application fields, because of its simple algorithm schematic and low computational complexity. This thesis has completed the effective combination of DCT Coding (Lossy Coding) and Run-Length Coding (Lossless Coding), realize the restoration of image ,with high compression rate, under a certain extent of visual quality. Above the traditional DCP coding algorithm in JPEG, an improved arithmetic based on threshold segmentation is proposed.

Fractal coding uses image self-similarity to traverse each range block in a local iterated function system to search for the domain block with the lowest mean squared error (MSE). Generate an affine transform codebook to complete the coding and through the collage theorem completes iterative decoding. However, fractal coding is not a competitive image compression technique in practical applications due to its long coding time and high computational complexity.

This thesis proposed a hybrid coding method, combined with discrete cosine transform and fractal coding. After image segmentation, DCT transform is performed on simple blocks with little difference in numerical values, through which reduces the search time of domain block to shorten the encoding time of fractal. At the same time, re-segment the original image and use the sub-block after second segmentation compensates the restored image to eliminate the block effect. The experimental results show that the improved algorithm proposed in thesis paper decrease the time of encoding to a certain extent compared basic fractal coding, and also performed significantly improvement of image restoration quality.

Key Words: Image Compression Encoding; Discrete Cosine Transform; Threshold Segmentation; Fractal; Iterative Function System.

目 录

摘要	I
Abstract	II
第 1 章 绪论	1
1.1 图像压缩研究背景及意义	1
1.2 图像压缩技术的研究现状	1
1.2.1 图像压缩技术概述	1
1.2.2 图像压缩技术分类	2
1.3 论文研究内容及安排	4
第 2 章 图像编码基础	5
2.1 图像表示与编码质量评价	5
2.1.1 静态图像的数字表示	5
2.1.2 编码质量评价	7
2.2 信息论基础与熵编码	8
2.2.1 离散信源的熵表示	9
2.2.2 Huffman 编码	10
2.2.3 行程编码	12
第 3 章 基于离散余弦变换理论的图像编码方法	13
3.1 离散余弦变换理论	13
3.1.1 一维离散余弦变换	13
3.1.2 二维离散余弦变换	14
3.2 基于 DCT 变换的图像压缩	15
3.2.1 基本算法原理	15
3.2.2 算法实现步骤	16
第 4 章 基于分形的图像编码方法	21
4.1 分形图像压缩基本理论	21
4.1.1 度量空间与仿射变换	21
4.1.2 压缩映射与不动点定理	21
4.1.3 迭代函数系统	22
4.1.4 拼贴定理	23
4.2 基本分形编码方法	23
4.2.1 基本算法原理	24
4.2.2 算法实现步骤	27

4.3 分形与离散余弦变换的混合编码	28
第5章 实验结果与分析	31
5.1 实验设置	31
5.2 DCT 编码算法分析	31
5.3 分形编码算法分析	33
5.4 混合编码算法分析	35
第6章 总结与展望	39
6.1 总结	39
6.2 展望	40
参考文献	41
附录	43
致谢	44

第 1 章 绪论

1.1 图像压缩研究背景及意义

进入信息爆炸时代后数字信息化高速发展，人们的日常生活中无处不充斥着大量的信息包括文字、图像、音频、视频、音频。随信息技术高歌猛进式的快速发展，人们对于手机摄像拍照等功能的使用也日益频繁，图像的数字化传输在诸如视频会议、气象云图、医疗成像、遥感图像等数据传输中扮演着非常重要的作用。这些图像文件可能非常大，处理这些大量的数据会占据大量的时间、空间和带宽，我们必须高效地存储、检索、分析和处理数字信息，以便投入实际使用。

未压缩的图像数据通常会占据大量的存储空间。CD-ROM 光盘是一种典型的数据存储介质，可以容纳大约 650Mb 的数据。虽然这足以存储文本文件，一个众所周知的例子是《大英百科全书》的全部文本可以存储在一张光盘中，且仍有空余空间。但它远远不能满足存储数百个彩色图像，视频剪辑和声音轨道这样的多媒体应用需求。再以 RAW 图像为例，计算分辨率为 640×480 的图片大小。10 种颜色中的每一种都由 8 位表示，那么对于每个像素它需要 24 位。图像中的像素总数为 $640 \times 480 = 307200$ 像素，因此图像的大小即为 307200×3 个字节 = 921600 个字节。在 25 帧/秒的帧速率为下，每秒 RAW 视频流占据 $640 \times 480 \times 3 \times 25 = 23040000$ 字节（22MB），为此我们需要 105600MB（ $22\text{MB} \times 60 \times 80$ ）作为 80 分钟视频的存储空间。然而 80 分钟相同分辨率的 VCD 格式视频仅需要 700MB。基于这些数据展现的微观细节，我们可以以牺牲一定计算复杂度为代价，实现约为 150:1 的高压缩^[1]。

正是基于这样的环境背景以及迫在眉睫的应用需求，近三十年来图像压缩技术成为了一个炙手可热的研究领域，并在多个领域得到了广泛应用，取得了卓越的经济效应^[2]。

1.2 图像压缩技术的研究现状

1.2.1 图像压缩技术概述

图像压缩是图像处理众多技术中的一个，在很多领域都有应用，并在图片的高效传输和存储中起着重要的作用。图像的压缩与原始的二进制数据压缩有着显著不同，图像具有一定的统计特性（信源特性），使用不同设计原理的编码器，根据其独特的统计特性进行因地制宜地编码，相较于使用单一通用的编码原理可以有效提升编码效率以及表现出更好的压缩效果。图像压缩的核心原理是消除信号源的冗余，即去除下列三个不必要的基础数据：

1. 编码冗余

2. 相邻像素之间的空间冗余及相关性, 即像素间冗余
3. 被人类视觉系统所忽略的心理视觉冗余

通过上述三个重要方法, 去除图像数据中对图像质量影响不大的冗余信息, 以存储或传输大小尽可能少的样本数据, 并从样本数据中重构能被人类视觉感知的良好图像, 以达到可靠、优质、经济的要求^[3]。

图像压缩的核心是如何对数字化表示后的图像进行有效压缩, 在允许图像有一定失真的基础准则下, 获得尽可能小的数据。一般图像压缩过程不单包括图像压缩的编码, 也包括对图像恢复的解码过程, 在这一过程中根据有损、无损压缩技术的分类, 获得部分或完全的恢复, 从而复原表达原始图像所需的数据量^[2]。压缩系统接受输入的信源 X 进行编码后生成数据量更少的压缩信息, 经过传输、存储经由解压缩算法重构成信宿 Y, 其原理框图见图 1.1。

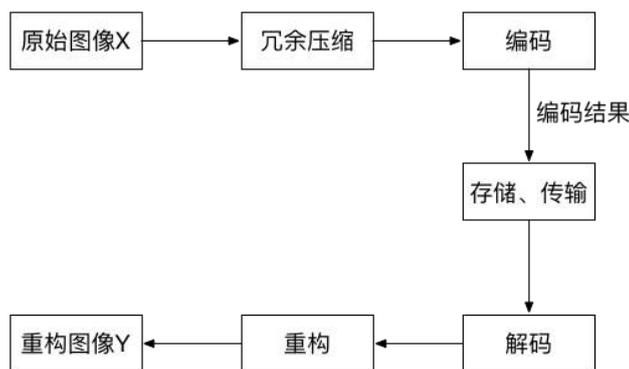


图 1.1 图像压缩编码与解码的原理框图

1.2.2 图像压缩技术分类

目前有多种图像压缩技术以减少其存储大小以及使用更小的空间。按压缩前后图像是否存在信息的丢失, 分为无损压缩和有损压缩。无损压缩利用信息论理论缩减数据大小, 依据数据的统计冗余进行压缩, 从压缩数据中完美重构原始数据而不产生任何数据损失, 是一种可逆的压缩方式。而另一种不针对原始图像数据进行精准重建的图像压缩技术被认为是有损压缩, 其结合人类视觉感知以及图像固有特性, 在压缩的过程中使用不精确的近似值逼近原始图像并丢弃部分冗余数据^[4]。还原图像数据时只能通过压缩数据进行近似重构, 而无法将原始图像完美准确地恢复。

如表 1.1 所示的一个有关无损与有损压缩的示例。在无损压缩中, 将每一个符号运行的次数编码为具体的数值有效地将编码缩短, 并且可以反解码得到原始的数据流; 无损压缩中数据被量化为 10 的整倍数, 有效地简化了原始数据但这种转变显然不可逆转。

表 1.1 无损与有损压缩的简单示例

	源数据	压缩数据
无损压缩	aaaaabbbbccccddddeeeeffffaaaaabb0	a5b4c2d4e4f4a5bb0
有损压缩	0.123, 1.2345, 21.2165, 21.999, 12.123,20,20,10	0,0,20,20,10

无损图像压缩在多个领域得到广泛而深远的应用，尤其在图像归档（如法律或医疗记录的存储）传真传输。但是今天的大多数场合下都使用有损图像压缩，因为其远高于无损压缩的压缩比可以帮助我们节约更多的带宽和信道资源。目前互联网上通用的图像格式包括 JPG 以及动态图像 GIF，通常将各种有损与无损压缩算法链接在一起，以实现压缩效率的最大化。

近来在信息传输中得到广泛应用的图像无损压缩编码算法主要有：

行程编码 (Run-length Encoding)，是一种原理简便的压缩线性数据的方法^[5]。这种技术将连续重复出现的序列替换为被称为行程 (runs) 的较短符号。一个灰度压缩图像的行程编码由序列 (lr, Nr) 表示其中 lr 是像素的强度， Nr 指连续像素的数量。行程编码与其他一些编码方法混合使用，是现在比较主流的行程编码应用方式，较传统的行程编码实现更好的压缩效果。

Huffman 编码, 由 D.A.Huffman 提出^[6]。Huffman 编码将图像中的每一个像素当作一个符号, 高频率出现的相同符号更频繁地分配一个较小的比特数, 而出现得不那么频繁的符号指定一个相对大的比特数。Huffman 编码是一种前缀代码, 即任何一个符号的二进制编码都不是另一个符号编码的前缀。大多数图像编码标准使用有损压缩技术在前期压缩阶段, 并用 Huffman 编码进行最后的熵编码。

LZW 编码,LZW(Lempel-Ziv-Welch) 最早在 1977 年由两位以色列教授提出, Welch 于 1985 年在此基础扩展延伸, 故由三个人名字共同命名^[7]。LZW 是一种字典编码, 可以是静态亦或者是动态。在静态字典编码中, 字典序列在编码和解码过程固定; 而在动态字典编码中, 字典信息时刻在变化。LZW 在计算机行业中得到广泛应用, 并在 UNIX 操作系统上作为压缩命令的主要编码。

区域编码 (Area Coding), 是基于行程编码的增强算法, 能很好地体现二维图像的特征。区域编码相对于其他无损压缩技术是一个重大的进展, 它试图在矩形区域内找到相同的特点以实现压缩。

应用领域主要使用的有损压缩编码算法有：

变换编码 (Transform Coding)^[8], 通常将像素分割为 $N \times N$ 图像细分为更小的 $R \times R$ 块并且对每个图像子块执行单一变换。在变化编码中, DFT 和 DCT 变换是用来改变原始图像各个像素的频域系数 (变换系数) 从而实现原始图像的去相关冗余。变换编码的一

大优点就是，仅通过保存少量的变换系数就能完成对信号能量的重新分配。

矢量量化 (Vector Quantization)，是一种在图像、语音信号处理领域普遍应用的数据压缩方法^[9]。其基本思想是构造一种固定大小的向量字典，称为码向量，一个码向量通常是一个像素值块的值。将需要压缩的图像划分为非重叠块（矢量），并由可进一步熵编码的索引序列表示。

分形压缩 (Fractal Compression)，最早由 Barnsley 于 1985 年引入^[10]。分形压缩基于使用迭代函数系统的分形变换，根据特定的图像及同一副图像的一部分与其他部分的相似程度进行反射变换与迭代压缩。其分形参数包括分形维数、颜色分离、边缘检测和光谱。

预测编码 (Predictive Coding)，是另一种符合声音、图像数据需求的压缩方法^[11]。预测编码根据离散信号之间存在的一定关联性，利用当前以及过往像素值预测下一个数据信号进行，仅对首字符以及预测误差进行编码。在图像编码中常用的预测编码是差分脉冲编码调制方法 (Different Pulse Code Modulation, DPCM)^[12]。

1.3 论文研究内容及安排

本文基于大量国内外图像压缩技术的有关文献，完成了对 DCT 变换图像编码以及分形图像编码的理论研究以及算法实现，在二者各自特点的基础上提出了一种混合算法。论文内容安排如下：

第一章主要介绍了图像压缩的研究背景、基本原理，并以几种常见的图像编码方式对有损、无损压缩技术之间的差异进行了解释说明。

第二章以图像编码的基础展开，用理论知识描述了图像数字化表示的过程，给出了编码质量评价的几个重要指标参数。以香农的信息论为基础，对信息与熵的概念做了明确的阐述，并介绍了 Huuffman 编码、行程编码这两种主要的熵编码方式。

第三章介绍了离散余弦变换 (DCT) 的理论概念，对其基本原理以及算法步骤进行完整阐述，在基本算法的基础上提出了两种基于阈值二分的改进算法。

第四章首先介绍了分形的基本理论，包括拼贴定理与函数迭代系统。道明了基本分形编码的算法原理，提出了一种基于 DCT 补偿的混合编码算法。

第五章完成了对前文提出算法的 MATLAB 程序实现，通过 PSNR、SSIM 以及编解码用时三个性能参数的比对，完成了对 DCT 变换编码、分形编码的优缺点对比的同时，验证了本文所提出的几个改进算法的可行性。

第六章对本文的工作进行总结概括，点出了本文实验的不足之处，就图像压缩的研究方向表达了自己的微薄看法，勾绘了一副宏伟的蓝图。

第 2 章 图像编码基础

2.1 图像表示与编码质量评价

2.1.1 静态图像的数字化表示

黑白二维图像是数字领域最基本的表现形式，下面以它为例简单描述图像数字化的表示过程。用一个二维连续函数 $f(x, y)$ 来表示这个黑白二维静止图像， $f(x, y)$ 表示在 (x, y) 处的像素亮度值（或灰度值），如图 2.1 所示。

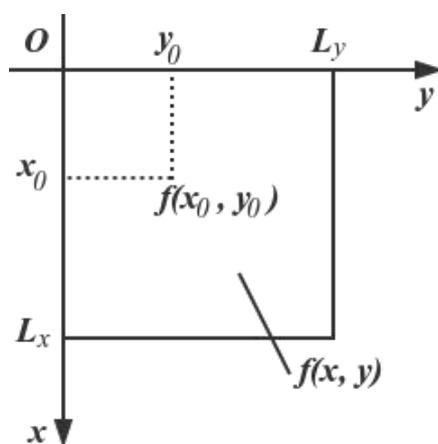


图 2.1 黑白二维图像的数字形式

图像的灰度值通常是一个非负的有界值，它总小于某常数 k

$$0 \leq f(x, y) < k \quad (2.1)$$

此外这个图像可以认为是矩形的，即满足

$$\left. \begin{aligned} 0 \leq x < L_x \\ 0 \leq y < L_y \end{aligned} \right\} \quad (2.2)$$

其中 L_x 和 L_y 分别表示该二维静态图像的高和宽 (图 2.1)。

对二维图像 $f(x, y)$ 在空间域上取样后的连续值 $f(i, j)$ 值 $f(i, j)$ 量化为离散值 $f_q(i, j)$ 的过程，即为图像数字化。图像的每个取样点 ($f(i, j)$ 中的每一个单元) 为一个像素，我们通常所说的图像的分辨率即与其在空间上包含的像素数量正相关。

假设量化后的 $f_q(i, j)$ 共有 $(M * N)$ 个数值，它们组成一个量化图像矩阵 F_q ,

$$F_q = \begin{bmatrix} f_q(0,0) & f_q(0,1) & \cdots & f_q(0,M-1) \\ f_q(1,0) & f_q(1,1) & \cdots & f_q(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f_q(N-1,0) & f_q(N-1,1) & \cdots & f_q(N-1,M-1) \end{bmatrix} \quad (2.3)$$

可以从下图看到一个很清晰地将灰度图像 Lenna 数字化的实例

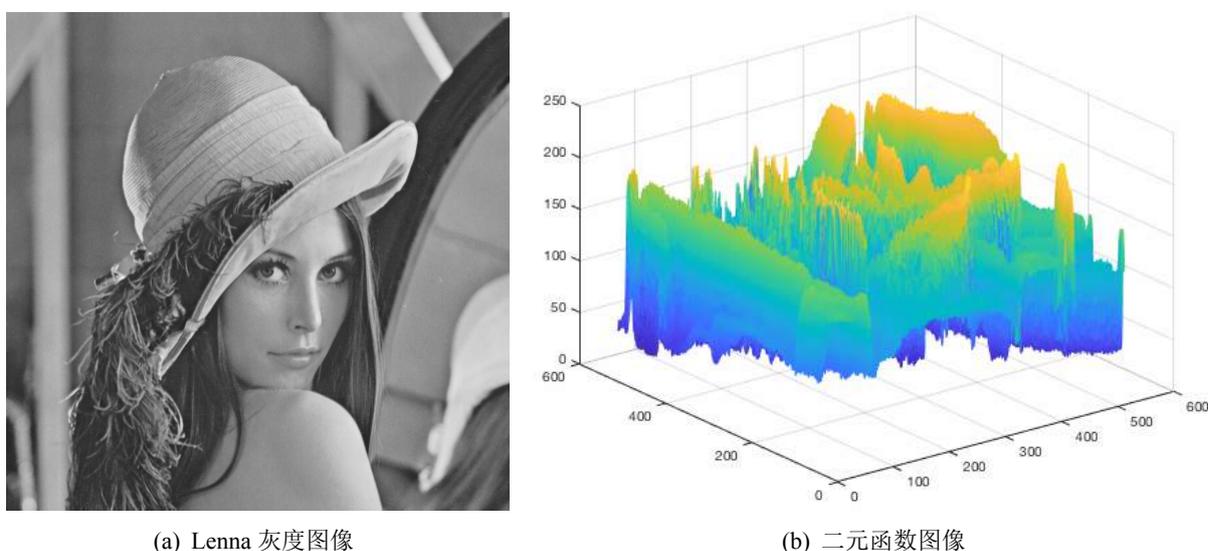


图 2.2 灰度图像的数字化

不同于灰度图像，单一的数据集合 $\{f(i, j), 0 \leq i < M, 0 \leq j < N\}$ 只能表示彩色图像的某一个分量，无法完整地表示一幅彩色图像^[2]。

彩色图像分量组合方式主要有 RGB 和 YUV 这两种。RGB(Red, Green, Blue) 根据亮度和色度内容对颜色进行描述, 在许多应用领域中能更有效地处理和传输彩色信号。通过改变红、绿、蓝三种颜色中的每种颜色的强度，可以复制眼睛可感知的任何颜色。对于计算机应用而言，每个灯光的强度通常被分成 256 个离散水平，编号为 0 到 255。通过这三种彩色光的不同组合，可以演绎超过 1600 万种颜色，从黑色 (RGB=[0,0,0]，即根本没有光) 到亮白色 (RGB=[255,255,255])^[13]，下式 (2.4) 可以更好地解释这一点：

$$C = Rr + Gg + Bb \quad (2.4)$$

其中 r, g, b 为红, 绿, 蓝的单位向量。

YUV 则是另一种颜色编码系统。它将人类感知考虑在内，减少了色度分量的带宽与相关性。在传输编码变换错误或压缩成像上比 RGB 系统更容易被人类感知屏蔽，因而更适用于图像压缩中色彩系统的应用。在 YUV 格式中，Y 表示彩色图像的亮度分量，U,V 则为用来表示色差值的色度分量^{[2][14]}。

可以通过下式实现由 RGB 分量到成 YUV 分量的正向转换

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.5)$$

由 YUV 分量转换成 RGB 分量的转换关系则为

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.42 \\ 1 & -0.34414 & -0.71414 \\ 1 & 1.1772 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \quad (2.6)$$

2.1.2 编码质量评价

图像在数字化处理过程中会遇到不同程度的失真，无论是在采集、处理、压缩，还是在存储、传输和重构中，都可能致使图像质量在视觉上的下降。由于图像最终要由人类观看，评价视觉图像质量的唯一“正确”方法是通过人的主观评估，这类方法将人类视觉系统 (Human Visual System, HVS) 的特性纳入知觉测量当中。然而在实践中，主观评估费时费力，劳神伤财，这就要用到基于数学定义的客观质量评价，如日前被广泛使用的均方误差 (Mean Squared Error, MSE)、峰值信噪比 (Peak Signal-to-Noise Ratio, PSNR)、均方根误差 (Root Mean Squared Error, RMSE)、平均绝对误差 (Mean Absolute Error, MAE) 和信噪比 (Signal-to-Noise Ratio, SNR)^[15]。以及由德州大学奥斯丁分校的图像和视频工程实验室 (Laboratory for Image and Video Engineering) 提出的结构相似性 (Structural Similarity Index, SSIM)^[16]。本文在实验结果与分析这一章，将对压缩算法的质量评价将采用如下客观质量评价指标。

(1) 压缩比 (Compression Ratio, CR)

$$C = \frac{m}{n} \quad (2.7)$$

CR 用来表示压缩前图像所需的数据量与压缩后图像所需的数据量之比， m, n 分别为

原图像、恢复图像的比特数。

(2) 均方误差 (MSE)

$$MSE = \frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M (f(i, j) - f'(i, j))^2 \quad (2.8)$$

其中 $A = f(i, j)(i = 1, 2, \dots, N; j = 1, 2, \dots, M)$ 为设原始的二维图像, $A' = f'(i, j)(i = 1, 2, \dots, N; j = 1, 2, \dots, M)$ 为解压后的图像数据。MSE 用来表示原始图像和重建图像之间的失真差异。

(3) 峰值信噪比 (PSNR)

$$PSNR = 10 \lg \frac{f_{max}^2 \cdot MN}{MSE} \quad (2.9)$$

其中 f_{max} 表示图像最大灰度值。PSNR 以 db 为单位，是衡量图像保真度的使用最普遍的评鉴图像画质的客观量测依据。

(4) 结构相似性 (SSIM)

$$SSIM(f(x, y), f'(x, y)) = \frac{(2\mu_x\mu_y + c1)(\sigma_{xy} + c2)}{\mu_x^2\mu_y^2 + c1)(\sigma_x^2 + \sigma_y^2 + c2)} \quad (2.10)$$

其中 μ_x 是 x 的平均值, μ_y 是 y 的的平均值, σ_x 是 x 的方差, σ_y 是 y 的方差, σ_{xy} 是 x 和 y 的协方差。 $c1 = (k1L)^2, c2 = (k2L)^2$, 是用来维持稳定的常数。 L 是像素值的取值范围, $k1 = 0.01, k2 = 0.03$ 。SSIM 作为一种衡量两幅图像相似度的新指标, 用简单的 0 至 1 的取值将图像差异性数值化, 值为 1 时表示二者完全相同。

2.2 信息论基础与熵编码

信息熵被定义为由随机数据源产生的平均信息量, 其概念是由香农 (Claude Shannon) 在其 1948 年的论文《通信的数学理论》中提出的^[17]。他将信息的概念进行简化, 以 bit (比特) 作为量纲衡量, 在信息、不确定性、熵、混沌这些概念之间建立了联系。

信息熵的度量是该数据值概率函数的负对数, 即当数据源具有较低的概率值 (即发生低概率事件时) 时, 该数据源较高概率事件携带更多的信息。以这种方式定义的事件信息量变成了一个随机变量, 其期望值就是信息熵。基于熵代表无序与不确定性这一概念, 信息论结合了统计热力学中熵的概念 (布朗运动), 建立了对信息的有效量度方法。

2.2.1 离散信源的熵表示

在许多应用场景中存在一些可能的相互排斥的结果用符号 s_1, s_2, \dots, s_n 表示。例如，在图像处理中我们可能有 256 个符号，其中 s_1, s_2, \dots, s_{256} 表示特定像素的灰度值。对于音频信号， s_i 可以表示振幅信号样本的电平。在文本应用中， s_i 可能表示各种字母，数字，标点符号和其他特殊字符。上述应用场景下，往往会有许多单一符号组成的“流”或者称为“源”。例如，一幅 512×512 像素的图像有超过 260,000 个符号，一秒钟的数字音频有 44,100 个符号，而在文本文档中，我们通常会面临有数千个符号。如果有人发送来诸如上述那些符号串的信息源，我们会收到一些不确定性符号序列。一种有效而常用的模拟方法是假设数据按照某种概率分配随机出现的，每个符号 s_1, s_2, \dots, s_n 发生的概率为 $P(s_1), P(s_2), \dots, P(s_n)$ 可以明确 $0 \leq P(s_i) \leq 1, (1 \leq i \leq n)$ ，另外我们假设 s_1, s_2, \dots, s_n 是唯一可能正在使用的符号，假设均出自符号集 S ，则有

$$P(s_1) + P(s_2) + \dots + P(s_n) = \sum_{i=1}^N P(s_i) = 1 \quad (2.11)$$

一个数据值包含的信息量（可以称为自信息量）与其出现概率的对数负相关，可直观地定义符号 s_n 的自信息量为

$$I(s_i) = \log_2 \frac{1}{P(s_i)} = -\log_2 P(s_i) \quad (2.12)$$

构成的离散信源的每个符号的平均自信息量 $H(S)$ ，则被称为信源的熵 (Entropy)。

$$H(S) = \sum_{i=1}^N P(s_i) I(s_i) = -\sum_{i=1}^N P(s_i) \log_2 P(s_i) \quad (2.13)$$

上述以 2 为底的信息熵的度量单位为 bit，以自然对数 e 为底的熵度量单位被称为 nat，以 10 为底则可以用 ban 来表示，上述三者中 bit 为最常用的一个。信息源熵的重要性在于其操作意义涉及编码来源。而由于 H 所代表的平均数来自信息源的每个符号的信息，意味着每个符号至少需要使用 H 个比特来表示具有唯一可解码的源码。这就引发了有关于信源理论的思考，有关信源理论有如下结论：

1. 编码一组信源符号时，码字的平均码长 $L \geq H(A)$ 。 L 的计算公式如下：

$$L = \sum_{i=1}^N P(s_i) l(s_i) \quad (2.14)$$

2. 如果所有 $I(s_i)$ 为整数, 则令 $l(s_i) = I(s_i)$, 可以使平均码长等于熵。
3. 等概率分布的信源符号具有更大的熵, 需要更长的平均码字, 可以证明

$$H(S) = - \sum_{i=1}^N P(s_i) \log_2 P(s_i) \leq \log_2^N = \sum_{i=1}^N \frac{1}{N} \log_2 N \quad (2.15)$$

基于信息熵的概念, 可以获悉的是传送信源信息理论上只需要具有信源极限熵 H 大小的信息, 编码所要解决的问题就是使得信源熵在实际通信系统中达到理论要求的最小信息率。而基于信源理论, 可以根据每个符号的自信息量差异分配不同码长的码字, 对于自信息量大的分配长码字, 自信息量小的分配短码字, 从而将平均码长降低至与信息熵接近的程度, 这就是一种常见而具有优越性的编码——变长编码, 也就是熵编码。两种最常用的熵编码技术是哈夫曼 (Huffman) 编码和算术编码^[6]。本文压缩算法在熵压缩阶段中用到的熵编码为行程编码 (Run-Length Encoding)^[5]。

2.2.2 Huffman 编码

Huffman 编码, 由 David A. Huffman 在其 1952 年的论文中提出^[6]。Huffman 编码被认为是计算机科学和信息论中, 用以无损数据压缩的前缀码最优编码方式。其基本思想是为输入字符分配长度不等的编码, 分配代码的长度取决相应字符的出现频率, 从而减少整个文件的数据量完成压缩。

Huffman 编码通过使用频率排序的前缀二叉树来对符号进行编码, 描述编码的前缀二叉树确保了每个特定符号的编码与任何其他符号编码的前缀都不相同, 以防止解码中歧义的产生。为了确定符号的二进制编码赋值, 使树的叶子与符号一一对应, 且其编码赋值将从树的根遍历到叶的路径。为了确保字符频率与编码长度负相关, 生成 Huffman 树的完整步骤如下:

1. 将每个符号值以及其出现频率记为一个叶子节点, 且所有叶子节点的集合为 A 。
2. 找到概率最小的两个节点 a_{T1}, a_{T2} 并将它们组合成单个节点, a_{T1}, a_{T2} 作为这个节点的左右子节点, 且其频率为左右节点 a_{T1}, a_{T2} 之和。
3. 将 a_c 放入 A 并将 a_m, a_n 删除, 重新进行频率排序
4. 重复上述步骤 2,3, 直到叶子节点集合 A 中仅有一个节点且其频率为 1, 这个节点即为 Huffman 树的根节点, 这颗树即为所求 Huffman 最佳编码树。

下表 2.1 为 8 个符号组成的信源数据, 并按概率排序后计算出各个符号的自信息量, 后两列分别表示 Huffman 编码后的编码长度和码字。

表 2.1 Huffman 实例

符号	P_k	自信息量	码长	码字
a_1	0.35	1.5146	1	0
a_2	0.30	1.7370	2	10
a_3	0.17	2.5564	3	110
a_4	0.10	3.3219	4	1110
a_5	0.04	4.6439	5	11110
a_6	0.02	5.6439	6	111110
a_7	0.01	6.6439	7	1111110
a_8	0.01	6.6439	7	1111111

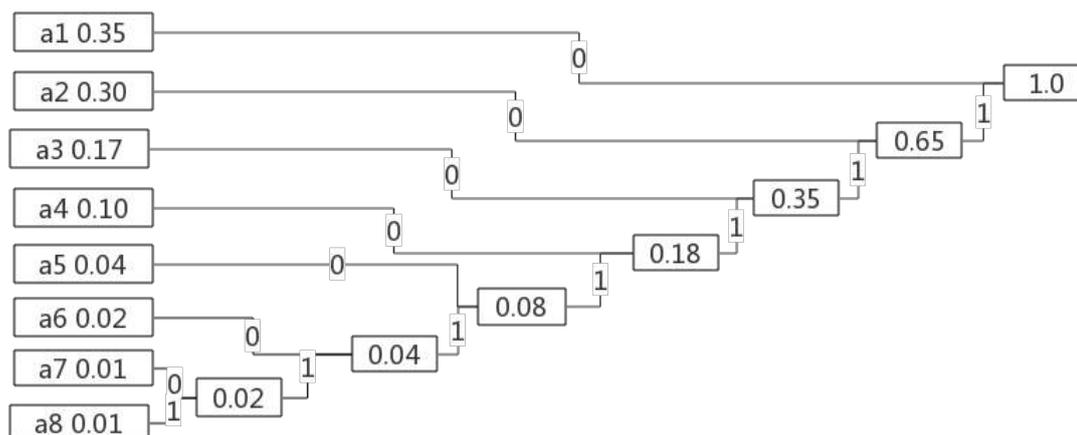


图 2.3 Huffman 编码过程实例

上图 2.2 为 Huffman 编码树, 从图中可以直观地看到符号概率与码长的关系。根据信息论理论, 我们可以计算出信息熵 $H(A) = 2.2495$, 平均码长 $L = 2.32$, 编码效率 (信息熵与平均码长之比) 为 $\rho = 96.96\%$ 。

另外值得注意的是, 对于每个不同的文件, 利用 Huffman 编码对其进行压缩时都需要对其生成一一对应的编码表, 也就是说输入信息 X 的编码表无法作用于另一个信源数据信号 Y, 这就意味着需要将 Huffman 编码表在数据传输过程中与压缩数据一同传送。

2.2.3 行程编码

行程编码, 又称游程编码 (Run-Length Encdeing, RLE), 一种算法简便的无损数据压缩形式^[5]。行程编码将信源数据符号重复出现次数（即在许多连续数据元素中出现相同数据值的序列）存储为单个数据值并进行计数而非传统的连续排列。这对于包含许多此列连续数据排列的信源可以大幅度减少数据冗余，从而完成数据的压缩，例如单一图像，包括图标，线条图和动画以及那些含有大区域目标和背景的图像。试想一个白底黑字的图片文件，空白区域会出现很多白色像素，文本中会出现很多黑色像素细小区域。用一根假设的扫描线读取图像数据，其中 B 代表黑色像素，W 代表白色像素，编码如下所示：

WWWWWWWWWWWWBWWWWWWWWWWBBBWWWWWWWW

以上 37 个字节的的数据若采用行程编码, 可以将其渲染如下:W12BW12B3W8, 只用 11 个码字表示，如下图所示。

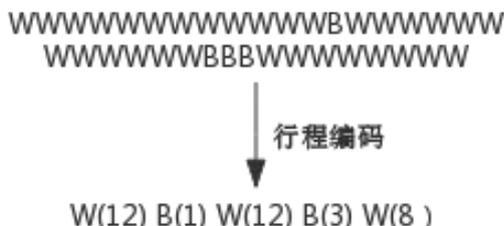


图 2.4 行程编码示例

上图 2.4 中, 括号内数字表示符号的连续重复次数, 称之为行程 (Run-Lnegth)。由上图的例子可以清楚的看到，行程编码在行程长度大于 3 时可以获得不错的压缩效果。但是行程编码对于符号连续重复率不高的信源，压缩效果往往差强人意；在某些极端情况下, 反而会使文件的数据量变大。

第 3 章 基于离散余弦变换理论的图像编码方法

3.1 离散余弦变换理论

离散余弦变换 (Discrete Cosine Transform, DCT) 将宏块上的数据扩展为较少数量的重要值，因其良好的能量压缩和解相关性，常被应用于对信号和图像进行有损数据压缩，在图像有损压缩处理标准 JPEG (Joint Photographic Experts Group)，以及广泛使用的有损视频处理标准 MPEG (Moving Picture Experts Group) 中都以核心编码算法地位出现^[18]。

在基于 DCT 的图像压缩编码中，一个图像被分割成几个小方块（通常是 8×8 ）并需要在执行 DCT 系数运算后对系数矩阵进行量化，以实现高压缩比^{[19] [20]}。

3.1.1 一维离散余弦变换

设 $x(n) = 0, 1, 2, \dots, N - 1$ 为一维实数信号，其 DCT 变换为：

$$Y(k) = c(k) \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \cos \frac{2n+1}{2N} k\pi \quad (3.1)$$

$$\text{其中 } k = 0, 1, 2, \dots, N - 1, c(k) = \begin{cases} 1/\sqrt{2}, & k = 0; \\ 1, & k = 1, 2, \dots, N - 1 \end{cases}$$

一维信号 $x(n)$ 的余弦反变换 (IDCT) 公式为：

$$x(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} c(k) Y(k) \cos \frac{2n+1}{2N} k\pi \quad (3.2)$$

$$\text{其中 } c(n) = \begin{cases} 1/\sqrt{2}, & n = 0 \\ 1, & n = 1, 2, \dots, N - 1 \end{cases}$$

当公式 (3.1) 中的 $k = 0$ 时， $Y(0) = \frac{1}{\sqrt{2}} \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \cos \frac{2N+1}{2N} 0\pi = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} x(n)$ ，而变换后的第一个系数 $Y(0)$ 相当于整个信号的平均，频率为 0，称之为直流系数，而其他系数 $Y(1), Y(2), \dots, Y(N - 1)$ 称为交流系数。

下面将信号 $x(n), Y(k)$ 写成矩阵的形式：

$$Y = \begin{bmatrix} Y(0) \\ Y(1) \\ \vdots \\ Y(N-1) \end{bmatrix} = \sqrt{\frac{2}{N}} \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \cdots & \frac{\sqrt{2}}{2} \\ \cos \frac{1\pi}{2N} & \cos \frac{3\pi}{2N} & \cdots & \cos \frac{(2N-1)\pi}{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \cos \frac{1(N-1)\pi}{2N} & \cos \frac{3(N-1)\pi}{2N} & \cdots & \cos \frac{(2N-1)(N-1)\pi}{2N} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix} \quad (3.3)$$

将 (3.3) 记为：

$$Y = Ax \quad (3.4)$$

$$\text{其中 } A = \sqrt{\frac{2}{N}} \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \cdots & \frac{\sqrt{2}}{2} \\ \cos \frac{1\pi}{2N} & \cos \frac{3\pi}{2N} & \cdots & \cos \frac{(2N-1)\pi}{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \cos \frac{1(N-1)\pi}{2N} & \cos \frac{3(N-1)\pi}{2N} & \cdots & \cos \frac{(2N-1)(N-1)\pi}{2N} \end{bmatrix}$$

同理，式 (3.2) 可表示为：

$$x = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix} = \sqrt{\frac{2}{N}} \begin{bmatrix} \frac{\sqrt{2}}{2} & \cos \frac{\pi}{2N} & \cdots & \cos \frac{1(N-1)\pi}{2N} \\ \frac{\sqrt{2}}{2} & \cos \frac{3\pi}{2N} & \cdots & \cos \frac{3(N-1)\pi}{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\sqrt{2}}{2} & \cos \frac{(2N-1)\pi}{2N} & \cdots & \cos \frac{(2N-1)(N-1)\pi}{2N} \end{bmatrix} \begin{bmatrix} Y(0) \\ Y(1) \\ \vdots \\ Y(N-1) \end{bmatrix} \quad (3.5)$$

同样，可将 (3.5) 化简为

$$x = TY \quad (3.6)$$

$$\text{其中 } T = \sqrt{\frac{2}{N}} \begin{bmatrix} \frac{\sqrt{2}}{2} & \cos \frac{\pi}{2N} & \cdots & \cos \frac{1(N-1)\pi}{2N} \\ \frac{\sqrt{2}}{2} & \cos \frac{3\pi}{2N} & \cdots & \cos \frac{3(N-1)\pi}{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\sqrt{2}}{2} & \cos \frac{(2N-1)\pi}{2N} & \cdots & \cos \frac{(2N-1)(N-1)\pi}{2N} \end{bmatrix}$$

可以看到在 (3.4) 和 (3.6) 中，矩阵 A 是矩阵 T 的转置，即 $T^T = A$ ，并且有 $T^{-1} = A^T$, $A^T A = E$ (E 为单位矩阵)。这可以证明 A 是一个正交矩阵，即离散余弦变换为正交变换。

3.1.2 二维离散余弦变换

离散余弦变换和傅立叶变换关系密切，当傅立叶级数展开式是一个实偶函数时，那么对这个只包含余弦项的傅立叶级数离散化就可以得到余弦变换，离散余弦变换也因此得

名。下面介绍二维 DCT 的解析式。

设 $f(k, l)$ 是二维离散函数, $k, l = 0, 1, 2, \dots, N - 1$, 进行二维离散余弦变换后:

$$F(u, v) = c(u)c(v) \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} f(k, l) \cos \frac{(2k+1)u\pi}{2N} \cos \frac{(2l+1)v\pi}{2N} \quad (3.7)$$

其反变换函数为:

$$f(k, l) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} c(u)c(v) F(u, v) \cos \frac{(2k+1)u\pi}{2N} \cos \frac{(2l+1)v\pi}{2N} \quad (3.8)$$

$$\text{其中, } k, l = 0, 1, 2, \dots, N - 1, c(u), c(v) = \begin{cases} 1/\sqrt{N}, & u, v = 0 \\ \sqrt{\frac{2}{N}}, & u, v = 1, 2, \dots, N - 1 \end{cases}$$

在二维矩阵形式下的 DCT 变换因为其可分性与对称性, 可以由行列的两个一维变换分别计算得到, DCT 写成矩阵形式如 (3.9)

$$A_{DCT} = T \times A \times T' \quad (3.9)$$

其中 T 为式 (3.4) 中的 A , T' 为 T 的转置矩阵, A_{DCT} 即为元素 $a_{i,j} = F(i, j)$ 的 $N \times N$ 矩阵。并可以通过式 (3.10) 实现反离散余弦变换 (IDCT)

$$A_{IDCT} = T' \times A_{DCT} \times T \quad (3.10)$$

3.2 基于 DCT 变换的图像压缩

3.2.1 基本算法原理

首先对原始图像进行图像分割, 将原始图像切分为 8×8 的图像子块, 在这些子块上计算 DCT。对于切割后的像素矩阵 A 实现同式 (3.9) 的 DCT 变换, 当 $u = v = 0$ 时 $A_{DCT}(u, v)$ 的值是 DC 系数, 直流分量 (Direct Current), 也是 64 个图像采样的平均值, 其余为 AC 系数, 交流分量 (Alternating Current)。

使用离散余弦变换对一个图像数据进行压缩的过程, 主要经由变换器、量化器、编码器这三个主要组件。对 RGB 色彩系统的彩色图像进行 DCT 变换时需要分别对 R, G, B 三个维度分别做系数变换, 而不是对灰度图像变换压缩时仅需要对灰度值进行 DCT 变换。

本文采用的 DCT 图像编码与解码算法的流程图如下:



(a) 编码流程图



(b) 解码流程图

图 3.1 DCT 算法编码与解码流程图

3.2.2 算法实现步骤

DCT 编码是一种对称的数据压缩方式，即编码与解码的原理、工作量大致相同。这里只对编码过程作详细介绍，解码过程几乎是其逆过程。

首先对图像进行预处理，主要包括将 RGB 色彩系统转化为 YUV 色彩系统，需要注意的是根据第 1 章 (2.5) 的转换公式得到 Y,U,V 取值在 $[-255, 255]$ 之间，为了便于计算以及满足以满足离散余弦函数对于偶函数的要求，需要将其转化到 $[-128,127]$ 范围内。同时需要在需要解码中进行反变换，以将各像素点还原到原来的 RGB 颜色空间中。值得注意的是，在 RGB-YUV 转换过程中 YUV 的值往往是实数而非整数，这需要在量化过程中对其进行取整。

其次对图像进行子块分割 (8×8)，对每个图像子块应用二维 DCT 变换， 8×8 的 DCT 矩阵如 T 所示：

$$T = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{bmatrix}$$

将 DCT 变换后的图像数据输入量化器完成对数据的量化处理以及数据选择。现以测

试图片 baboon.png 中 Y 分量第 (1:8,1:8) 像素的 8×8 矩阵为例介绍阈值处理以及量化过程, 原始数据如下:

$$Y = \begin{bmatrix} 72.59 & 27.92 & 24.40 & 44.41 & 68.72 & 45.15 & 45.15 & 16.67 \\ 58.16 & 50.53 & 19.81 & 33.71 & 45.21 & 27.22 & 27.22 & 22.78 \\ 38.29 & 56.99 & 23.06 & 23.23 & 48.97 & 29.95 & 29.95 & 23.59 \\ 35.49 & 67.94 & 47.65 & 24.04 & 41.31 & 59.96 & 59.96 & 23.36 \\ 43.32 & 60.80 & 65.93 & 24.87 & 38.00 & 43.69 & 43.69 & 25.02 \\ 40.81 & 37.96 & 79.41 & 30.29 & 36.06 & 47.81 & 47.81 & 23.09 \\ 21.58 & 25.74 & 66.86 & 64.90 & 35.08 & 61.06 & 61.06 & 30.94 \\ 16.59 & 25.51 & 55.63 & 76.70 & 43.45 & 59.58 & 59.58 & 29.28 \end{bmatrix}$$

经过 DCT 变换后的 Y 分量 dY

$$dY = \begin{bmatrix} 317.60 & 40.70 & -37.47 & 17.11 & -17.48 & -13.27 & 17.77 & -0.35 \\ -11.01 & 16.55 & 38.74 & 47.99 & 26.05 & -15.10 & -38.75 & 15.66 \\ 1.50 & -15.63 & -28.43 & 8.04 & 41.92 & 28.21 & 1.84 & -12.00 \\ 10.20 & 13.06 & 3.62 & 4.05 & 0.04 & 12.95 & 21.72 & 20.41 \\ 14.07 & -0.88 & -8.60 & 9.50 & -2.28 & -1.60 & 1.84 & -3.93 \\ 4.10 & -8.14 & -13.17 & 5.64 & -1.31 & -1.92 & 10.88 & -7.06 \\ -5.37 & 2.38 & -2.97 & 3.90 & 4.97 & -4.06 & 2.61 & 10.62 \\ -0.02 & -5.22 & 0.40 & -2.03 & 4.63 & 3.81 & -9.11 & 3.76 \end{bmatrix}$$



(a) 原始图像



(b) 恢复图像

图 3.2 DCT 变换的 AC 信号验证

可以看到离散余弦变换的能量集中于矩阵的左上角, 对于能量占比较小的 AC 高频分

量，我们在这里可以适当舍弃以实现高压缩比，同时应该将小数取整以减少空间存储压力。图 3.2 是对 Lenna.png 图像做 DCT 变换后仅保留 AC 直流低频信号后获得的恢复图像，可以看到经过 IDCT 后能较完整的还原图像，能还原出大致的图形、轮廓，但是在细节上缺乏表现力。经计算该恢复图像结构相似度 SSIM 仅为 0.5953，这就意味着我们不能完全舍弃 AC 系数，需要对其有所保留的量化。

对于 AC 分量进行有选择地舍弃的过程也就是对变换后的 DCT 系数量化的过程，这个过程也就是信息损失的根源^[21]。在压缩算法中通常使用均匀量化器，具体的量化定义为：对变换后 8×8 子块中的 DCT 系数除以其量化步长，然后四舍五入。量化表达式为：

$$Q(u, v) = IntegerRound(F(u, v)/S(u, v)) \quad (3.11)$$

其中， $Q(u, v)$ 为量化的系数幅度； $S(u, v)$ 为量化步长，它是量化表的元素^[22]。以及通过以下公式进行反量化

$$F(u, v) = Q(u, v) \times S(u, v) \quad (3.12)$$

通过上文有关 YUV 色彩系统的介绍，可以知道亮度分量（Y）相比色度分量（U,V）对人眼而言更重要，因此我们对 Y 分量采用量化步长较小的量化表，对 U,V 采用量化步长较大的量化表，弱化对 U,V 分量的数据存储，这里采用 JPEG 标准的量化表。

可以看到经由量化后的矩阵 qY 在数据复杂度上已经有了很大幅度地改善。

$$qY = \begin{bmatrix} 20 & 4 & -4 & 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 3 & 3 & 1 & 0 & -1 & 0 \\ 0 & -1 & -2 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

除了量化我们还可以通过阈值处理，有效地选择数据从而降低冗余，这一步包括在量化器内。JPEG 算法中对于每个单一的 8×8 子块只保留左上角的 10 个系数而将其他 54 个系数直接置零，本文也对此进行了算法实现并记为算法一。

表 3.1 亮度 Y 量化表

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

表 3.2 色度 UV 量化表

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	56	99	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

在量化阶段我们对传统的 JPEG 量化器做出一定的改进与优化。对 DCT 变化后的矩阵不进行量化处理，而直接进行阈值分割处理，用下面的算法可以自动地得到阈值 T ：

1. 用矩阵中最大值与最小值的平均值作为 T 的初值。
2. 用 T 分割剩余矩阵。这样做会生成两组变换系数：所有绝对值大于 T 的系数 P_1 ，而 P_2 由所有绝对值小于或等于 T 的系数组成。
3. 对区域 G1 和 G2 中的所有变换系数计算平均值 μ_1, μ_2 ，并计算新的门限值： $T = (\mu_1 + \mu_2)/2$
4. 重复步骤 2,3，直到逐次迭代所得的 μ_1, μ_2 的均值与 T 的差小于预先定义的参数 T_0

对于 DCT 变换后的 8×8 矩阵中所有 AC 系数进行以阈值 T 为界限的阈值选择，选取 $T_0 = 0.03$ 为步骤 4 中设定的参数，绝对值大于阈值 T 的保持原有数值，绝对值小于 T 的记为 0，此记为算法二。

在算法二的基础上改变阈值选择的范围为保留左上角 10 个数，对剩下 54 个数进行上述二分阈值处理，此记为算法三。

经过量化器后所有无关紧要的高频值大多会被忽略，矩阵 qV 给出 63 个高频值大多为零。这将产生大量的冗余，我们可以通过有效地表示零，在熵编码阶段实现高压压缩。以此目的，本文在熵压缩部分采用了 Zig-Zag 扫描方法并对扫描后的线性阵列进行行程编码。

将量化后每个 8×8 矩阵中的低频 DC 信号单独存储，对 AC 信号采用 Zig-Zag 扫描将零值和非零值从矩阵数据中有效划分开来。量化值以 Z 字形的方式重新排序，通过取位置 $(0,1), (1,0), (2,0), (1,1), (1,2), (0,3) \dots (7,7)$ 的顺序来生成线性阵列，该线性阵列的顺序也在侧面体现了变换系数矩阵中能量由高到低的变化。它在下图中解释。

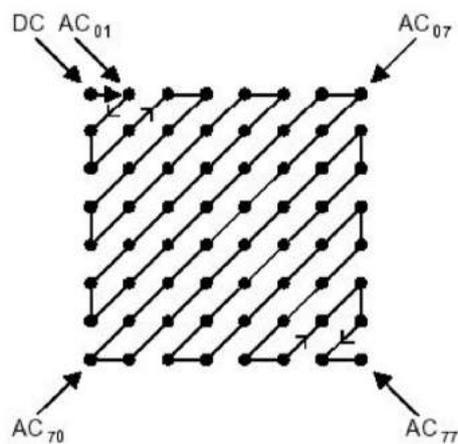


图 3.3 Zig-Zag 扫描

将上述量化矩阵进行阈值选择仅保留右上角十个数后。对 AC 系数可通过 ZigZag 扫描转换为 1×63 的单行线性阵列： $\{4, -1, 0, 1, -4, 1, 3, -1, 1, 0, \dots, 0\}$ 考虑到阵列末尾的 $\underbrace{44}_{44\text{个}0}$ 个 0，结合行程长度编码对于高频重复出现信号的由良好压缩效果的特点，对 AC 系数进行熵编码可以大大减少编码长度。

行程长度编码在第 2 章有过详细描述，这里不再赘述。

第 4 章 基于分形的图像编码方法

4.1 分形图像压缩基本理论

基于迭代函数系统 (Iterated Function System, IFS) 和拼贴定理的分形编码, 用一个简单的规则来描述分形几何对象之间的自相似性以及无限的细节, 将无限的复杂性编码成一个简单的公式。1985 年 Barnsley 提出“迭代函数系统”打开了分形图像编码的大门^[10], 1990 年 A.E.Jacquin 提出了全自动的分形图像压缩方法^[23], 以局部仿射变换取代全局仿射变换, 奠定了分形图像编码的基石。分形编码依据图像本身各子块之间的相似性, 存储相应的系数以完成编码; 在解码过程中, 将变换系数作用于任何大小与原始图像相同的图像, 在有限次迭代内就可以得到高质量的恢复图像。

4.1.1 度量空间与仿射变换

在对图像进行解码重构后, 往往需要与原图像进行差异性比对, 这就需把图像放在以数学理论为基础的度量空间上。

定义 4.1. (度量空间) 设 X, d 是一个非空集合, 其中的元素叫做点, 连同一个距离函数或者度量的映射 $d: X \times X \rightarrow R$ 。它把 X 中的每一对点 (x, y) 指派到一个非负的实数 $d(x, y)$, 且对于任何 $x, y, z \in X$, 必满足下述三条公理

1. (正定性) $d(x, y) \geq 0$, 且 $d(x, y) = 0$ 当且仅当 $x = y$ 。
2. (对称性) $d(x, y) = d(y, x)$
3. (三角不等式) $d(x, z) \leq d(x, y) + d(y, z)$

定义 4.2. 仿射变换 w (affine transform) 在二维空间中是 $R^2 \rightarrow R^2$ 空间的线性变换, 可写成如下形式:

$$w \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} ax + by + e \\ cx + dy + f \end{bmatrix} \quad (4.1)$$

4.1.2 压缩映射与不动点定理

定义 4.3. ^[24] 设 $w: X \rightarrow X$ 是 (X, d) 完备度量空间上的一个映射, 如果存在一个常数 $0 < c < 1$, 使

$$d(w(x), w(y)) \leq c \cdot d(x, y) \quad \forall x, y \in X \quad (4.2)$$

则称 w 为 (X, d) 上的压缩映射, c 为映射 w 的压缩因子。

定理 4.1. 设 T 是 (X, d) 完备度量空间上的一个映射，压缩比为 c ， $H(x)$ 是 X 的非空紧子集的全体，那么

$$T : H(X) \rightarrow H(X) \quad (4.3)$$

是分形空间 $(H(x), d_H)$ 的压缩映射，且压缩比为 c 。

定理 4.2.（不动点定理）^[25,26] 设 $T : X \rightarrow X$ 是 (X, d) 完备度量空间上的一个压缩映射，且压缩因子 $0 < c < 1$ ，则 W 存在唯一的不动点 B ，对于所有的 $B \in X$ ，满足

$$B = \lim_{n \rightarrow \infty} T^n B = \lim_{n \rightarrow \infty} T(T^{n-1} B) \quad (4.4)$$

4.1.3 迭代函数系统

定义 4.4.（迭代函数系统）^[10] 完备的度量空间 (X, d) 以及 n 个压缩映射 $w : X \rightarrow X$ （其因子分别为 c_1, c_2, \dots, c_n ）一起，组成一个迭代函数，简称 *IFS*，记作 $\{X : w_1, w_2, \dots, w_n\}$ ， $c = \max(c_1, c_2, \dots, c_n)$ 称为 *IFS* 的因子，则压缩变换 w 定义为

$$w(B) = \bigcup_{i=1}^n w_i(B) \quad \forall B \in X \quad (4.5)$$

由不动点定理可知，存在唯一的不动点 $B \in X$ 满足 $B = w(B)$ ，即

$$B = \lim_{n \rightarrow \infty} w^n(B) \quad (4.6)$$

则称 x 为 $\{w_i, i = 1, 2, 3, \dots, N\}$ 的吸引子，这个吸引子是一个分形。

将分形理论知识应用于图像处理，不难发现这样一个弊病：压缩变换 w_i 作用于图像 I 的整体去逼近一个个分散的局部往往过于理想化。为此 *Jacquin* 提出的局部迭代函数系统（*Local Iteration Function System, LIFS*）能很好地解决这个问题。它取 X 的一部分 $D_i \subseteq X$ 作为一个单独的度量空间，再此内部寻找压缩变换 w_i 取逼近 D_i 。通过这样的优化改进，使得分形图像压缩具有现实的应用前景，而不仅仅停留于理论与人造图像。

定义 4.5.（局部迭代函数系统）^[23] 设 (X, d) 是完备度量空间， $D_i \subseteq X$ （其中 $i = 1, 2, \dots, N$ ），*LIFS* 就是下列压缩映射集：

$$\{u_i : D_i \rightarrow X, (i = 1, 2, \dots, N)\} \quad (4.7)$$

4.1.4 拼贴定理

定理 4.3.（拼贴定理）^[25,26] 设 $T: X \rightarrow X$ 是完备度量空间 (X, d) 上的压缩映射，且压缩因子 $0 \leq c < 1$ ，则对于所有的 $x \in X$ ， T 的不动点 x_w 满足于：

$$d(x, x_w) \leq \frac{1}{1-c} d(x, T(x)) \quad (4.8)$$

通过拼贴定理我们可以知道，如果能就原始图像 I 找到一个迭代函数系统 W ，使得 I 和 $W(I)$ 之间的差异比较小，即使迭代函数系统 W 的不动点接近待编码图像，那么以 W 吸引子 A 作为重构的图像与原图相比误差很小。且根据式（4.5）可以明确整个图像 $W(I)$ 由 N 幅复制变形而又相似子图 $w_i(I)$ 拼贴而成，失真率取决于 $d(I, W(I))$ 大小^[27]。

4.2 基本分形编码方法

分形是由自然产生的精致结构，隐藏在我们周围的明显景象中，从一个细小的蕨类植物到规模庞大的宇宙结构。但是这种局部与整体成自相似性的完美图案很难在自然图像中找到，图像局部间的相似性却往往不难发觉，比如下图 4.1：



(a) Lenna 局部自相似性

(b) Building 局部自相似性

图 4.1 局部-局部相似性

基于这一点，将图像进行子块划分，依据块和块之间的相似性实现对图像的压缩，是算法的基本思想。需要注意的是，不同于 DCT 变换这类对称的压缩方法，编码和解码过程花费大致相同的时间或工作量。分形编码是一种典型的非对称图像压缩方法，此类方法在图像编码过程中完成对图像处理的大部分工作，从而创建一个可以快速解码的输出文件。

4.2.1 基本算法原理

我们将 $I_{N \times N}$ 划分为 $N_r \times N_r$ 个互不重叠的值域块 (Range Block) $\{R_i, i = 1, 2, 3, \dots, n\}$, 尺寸为 $B \times B$, $N_r = N/B$, 称之为值域块池, 使得

$$I_{N \times N} = \bigcup_{i=1}^{N_r \times N_r} R_i, R_i \cap R_j = \emptyset, i \neq j \quad (4.9)$$

再将图像 $I_{N \times N}$ 划分为尺寸较大的允许相互重叠的定义域块 (Domain Block), 用 $\{D_j\}$ 表示, 尺寸通常为 $2B \times 2B$, 我们将其称之为定义域块池。设相邻两个 D 块之间的移位为 δ 个像素点, 称 δ 为步长, 可以对垂直以及水平分量的步长进行区分并分别表示为 δ_h 和 δ_v , 通常取 $\delta_h = \delta_v = \delta$ 。那么图像 I 共有 $N_d = (\frac{N-2B}{\delta} + 1)^2$ 个 D 块。

对于任一个 R_i , 寻求仿射变换 w_i 使得变换后的 R_i 近似于某个 D_j , 即有 $D_j = w_i(R_i)$, 如图:

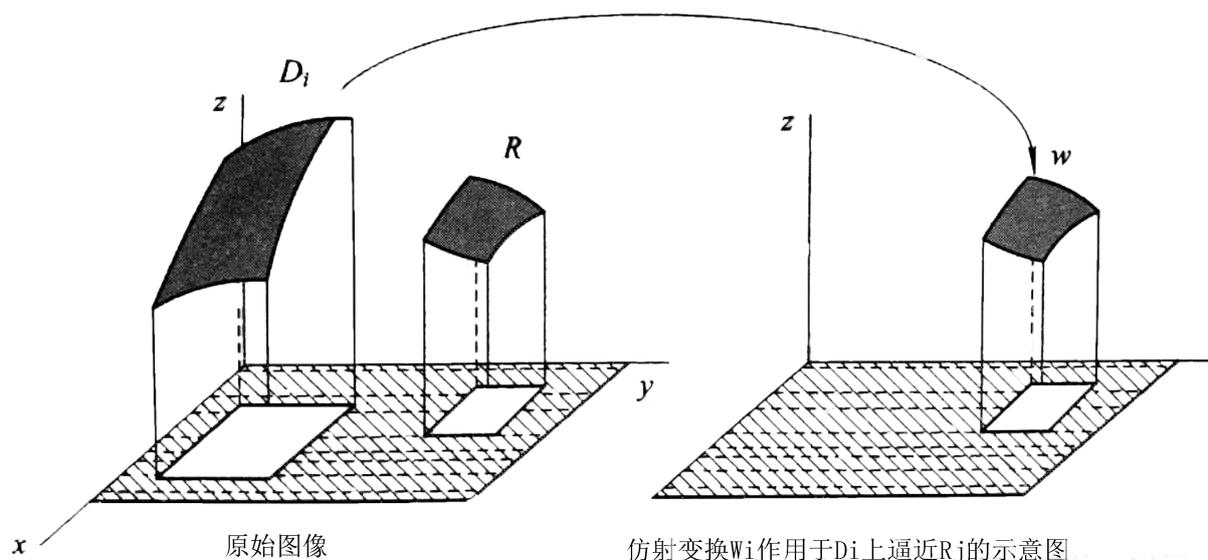


图 4.2 局部仿射变换示意图

由变换 $\lambda_i \gamma_i$ 复合而成的 w_i 形成了所有局部变换库 $\Omega = \{w_i, i = 1, 2, \dots, N_r^2\}$, 这些变换即为分形变换:

$$\forall I \in R^{N \times N}, W(I) = \sum_{i=0}^{N_r^2} (W(i)) | R_i = \sum_{i=0}^{N_r^2} w_i(I | D_{m(i)}) \quad (4.10)$$

上面式子的涵义是图像 I 中的每个 R 块用与之吻合的 D 块 $D_{m(i)}$ 在局部变换 w_i 下的变换

结果 $w_i(I | D)$ 替代从而重新拼贴出新图像 $W(I)$ 。

D_i 通过 γ_i 收缩、平移变换到 R_i 如上图 4.2 所示。常用收缩方式有四邻域像素均匀采样和欠采样的方法，具体如下图所示：

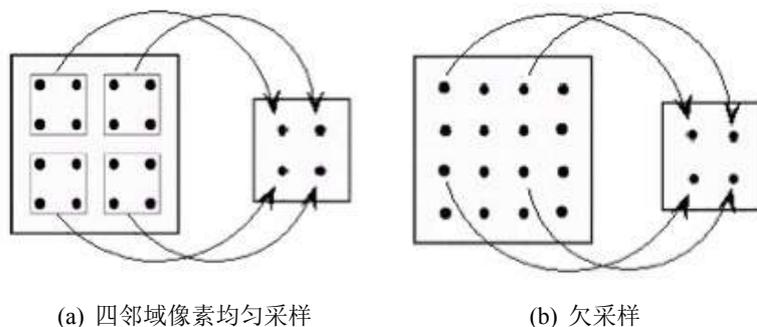


图 4.3 γ_i 收缩变换方式

λ_i 主要完成了对 D_i 的灰度级调整及像素等距变换，其变换方式如下：

$$\gamma_i \begin{bmatrix} x \\ y \\ z(x,y) \end{bmatrix} = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z(x,y) \end{bmatrix} + \begin{bmatrix} e \\ f \\ \sigma_i \end{bmatrix} \quad (4.11)$$

其中 $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ 为八种像素等距变换的变换矩阵， s_i, σ_i 分别表示亮度调整和亮度偏移，分别如图 4.4、图 4.5 所示。

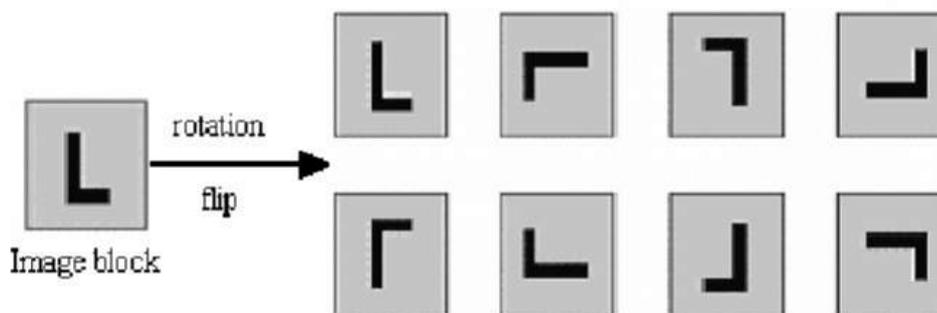


图 4.4 八种等距变换

通过上述对 λ_i, γ_i 的详细介绍，可将映射 w_i 表示为一般形式

$$w_i(D_{m(i)}) = \lambda_i(\gamma(D_{m(i)})) = s_i \cdot t_k(\gamma_i(D_{m(i)})) + \sigma_i \quad (4.12)$$

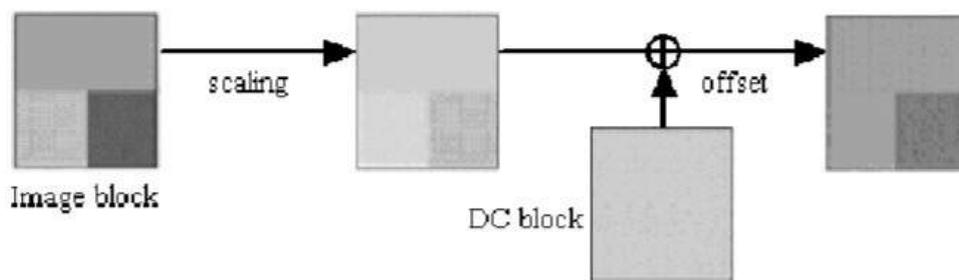


图 4.5 灰度级调整与偏移

其中 t_k 为等距变换矩阵，具体为恒等变换、对 y 轴对称变换、对 x 轴对称变换、90° 度旋转、180° 度旋转、270° 度旋转、对 45° 度线对称变换、对 135° 度线对称变换。

图 4.6 可以更好地解释式 (4.12) 中 w_i 的变换过程。图中 D_i 块经空域压缩到与 R_i 块一样的大小，然后经过等距变换、亮度调整、亮度偏移三种变换的组合（变换次序可以改变），得到的满足相似性匹配的 R_i 块。

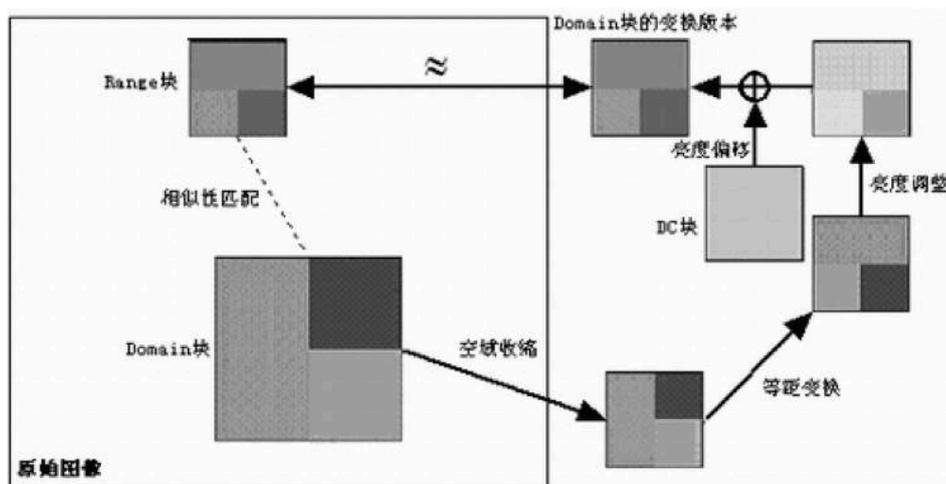


图 4.6 局部变换 w_i 作用示意图

通过上文对拼贴定理的描述，寻求即极小化拼贴误差 $d(I, W(I))$ ，即寻找 $D_{m(i)} \in \{D_j\}$ 和块变换 $w_i \in W$ ，使得 $R'_i = w(D_{m(i)})$ 与 R_i 在选定的失真度量 d 下尽可能接近成为了算法实现阶段的主要目标，可以用下式表示：

$$d(R_i, w(D_{m(i)})) = \min\{d(R_i, w(D)) \mid w \in W, \forall D\} \quad (4.13)$$

将定义域池中的 $D_{m(i)}$ 及最优局部变换 w_i 分配给值域池中每一个 R_i 的过程，我们称

之为码本生成，也就是分形编码的过程。但是这个步骤是一个计算复杂度非常高的非线性优化问题，分形编码的绝大部分计算时间都消耗在了这一部分。

分形解码较为简单，利用基于压缩变换不动点定理的迭代算法能在有限地迭代次数下完成对从任意图像到原始图像的解码过程。

4.2.2 算法实现步骤

依据上述有关分形的基本理论知识，设计出了以图 4.7 为原理流程的算法步骤，主要分为图像分割、码本构成、等距变换、分形编码这几个阶段。

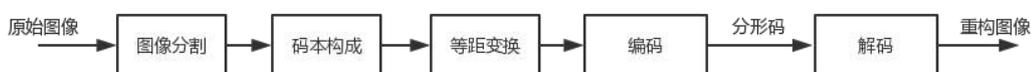


图 4.7 分形编码原理框图

图像分割：图像进行子块分割，将原图像像素分割成互不重叠的尺寸为 $B \times B$ 的 R 块以及 $2B \times 2B$ 大小的可重叠子块 D 块，水平相邻的两个 D 块之间的步长为 δ_h ，垂直相邻的两个 D 块之间的步长为 δ_v 。

码本构成：将四邻域均匀采样应用于所有的 D 块 $D_j(j = 1, 2, 3, \dots, N)$ 获得 D'_j ：

$$d'_{k,l} = (d_{2k,2l} + d_{2k+1,2l} + d_{2k,2l+1} + d_{2k+1,2l+1})/4 \quad (4.14)$$

这些变换后大小为 $B \times B$ 的 D'_j 构成了码本池 $\Omega = \{D'_j : j = 1, 2, 3, \dots, N\}$ 。

等距变换：对所有的 D'_j 进行八种等距变换 t_k 以增加搜索匹配度，八种等距变换的函数表示以及变换矩阵如下：

恒等变换	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$f(x, y)$
对 y 轴对称变换	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	$f(B - 1 - x, y)$
对 x 轴对称变换	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$f(x, B - 1 - y)$
180° 度旋转	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$	$f(R - 1 - x, R - 1 - y)$

对 45° 度线对称变换	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	$f(y, x)$
270° 度旋转	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$	$f(y, R - 1 - x)$
90° 度旋转	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$f(R - 1 - y, x)$
对 135° 度线对称变换	$\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$	$f(R - 1 - y, R - 1 - x)$

编码阶段, 遍历 $R_i (i = 1, 2, \dots, N_r \times N_r)$, 采取照某种特定方法 (如穷举暴力搜索) 在码本 Ω 中寻找满足要求的 $D_{m(i)}$, 使得 $D_{m(i)}$ 进行仿射变换后的 $D'_{m(i)}$ 与 R_i 之间的差异 $MSE(R_i, D'_{m(i)})$ 是最小的, 即求解下面的最小化问题:

$$\min_j \left\{ \min_{s, \sigma \in R, |s| < 1} \|R_i - t_k(s \cdot D_j + \sigma)\|^2 \right\} \quad (4.15)$$

其中 s, σ 分别表示亮度调整和亮度偏移, t_k 为等距变换序号。可以求得最小化问题的解为四元组 $\{t_k, m(i), s_i, \sigma_i\}$, 也就是 R_i 的分形码。

解码是相对简单的迭代过程, 基于不动点定理, 对于与原始图像大小相同的任意初始图像 μ_0 在预设的迭代次数 N 下, 可用下式生成压缩变换 T 的不动点图像 μ_{fix} :

$$T^N \mu_0 \approx \lim_{n \rightarrow \infty} T^n \mu_0 = \mu_{fix} \quad (4.16)$$

在码本中提取 R_i 的分形码后, 需依据原始图像的大小定义 X 区和 Y 区, 分别用来保存迭代后的重构图像、生成码本。在 Y 区码本中找到与 X 区 R_i 对应的 $D_{m(i)}$, 根据拼贴定理将 X 区的图像复制到 Y 区一次次迭代, 产生尺度更精细的细节, 进而得到最后的重构图像。

4.3 分形与离散余弦变换的混合编码

基于前文介绍的离散余弦函数与分形的图像压缩编码, 不难发现二者在算法实现上有明显差异。离散余弦变换的优点是算法步骤简单直观, 编码、解码时间较短; 分形编码的优点是恢复图像质量较好, 与分辨率无关, 缺点是压缩步骤冗长、编码计算量大。基于此, 本文结合了二者各自优点, 提出了一种基于 DCT 补偿的混合编码算法。

由离散余弦变换是正交变换的特性可以证明, 将元素全是 k 的 $N \times N$ 矩阵代入式 (3.9)

经过 DCT 变换后，可以得到一个 DC 系数为 $N \cdot k$ ，AC 系数全是 0 的矩阵。这样的子块在仅保留 DC 系数的情况下，可以做到完全的不失真，但是在实际的图像处理过程中往往很难找到这样变化单一的图像子块。为此我们选定一个评判标准来衡量图像的变化幅度，参考了两幅图像保真度的重要指标峰值信噪比（PSNR），对一个图像子块采用如下衡量公式：

$$P = 10 \lg \frac{f_{max}^2 \cdot M^2 N^2}{\sum_{i=1}^N \sum_{j=1}^M (f(i, j) - f'(0, 0))^2} \quad (4.17)$$

即以子块自身的与其第一个系数做差异性比对，称这个数据为“自信噪比”。

对于图像分割后的所有子块进行自信噪比计算，并设定阈值 T_p ，对于所有 $P \geq T_p$ 的图像子块记为“平坦子块”，对于所有 $P < T_p$ 的图像子块记为“非平坦子块”。对于平坦子块做 DCT 变换后，仅保留 DC 系数，就能存储其大部分信息；并把所有非平坦子块放入值域池寻找与之吻合的定义域块 $D_{m(i)}$ 。

但因为这两个算法都在图像预处理阶段都采用了子块分割，基于图像子块的编码运算使得子块与子块之间的相关性在编码过程中被忽视了，这就导致恼人的“块效应”。可以在下图 4.8 清楚看到 Lenna 图像在基本 DCT 编码（仅保留 DC 系数）中产生的块效应。



图 4.8 Lenna 图像块效应

可以从图 4.9 解释块效应产生的原因，像素点 A, B 被分割在同以子块内，在编码过程中保留了之间的相关性；而像素点 A, C 因图像分割而产生逻辑上的远离，它们之间的相关性被当成冗余而舍弃，这就导致了 A, C 之间出现了块效应，尽管 A, C 的距离甚至小于 A, B 。

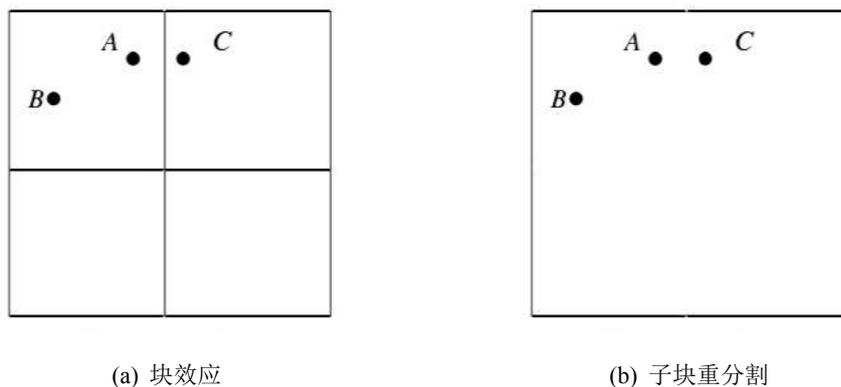


图 4.9 块效应与子块重分割

为消除因图像分割产生的块效应,我们对图像进行了重分割,将图像二次分割为 16×16 的子块进行 DCT 变换,并取变换矩阵左上角 36 个系数进行量化保存。用 16 图像块去“补偿”相邻 8×8 子块之间因图像分割丢失的边界连续性,进而降低方块效应。

算法实现主要由两步骤构成,对 8×8 和 16×16 子块分别记为 $R(i, j), D(i, j)$,并将 $(R(i, j) + D(i, j))/2$ 作为重构图像在 (i, j) 处的像素值,构成恢复图像。原理框图如下:

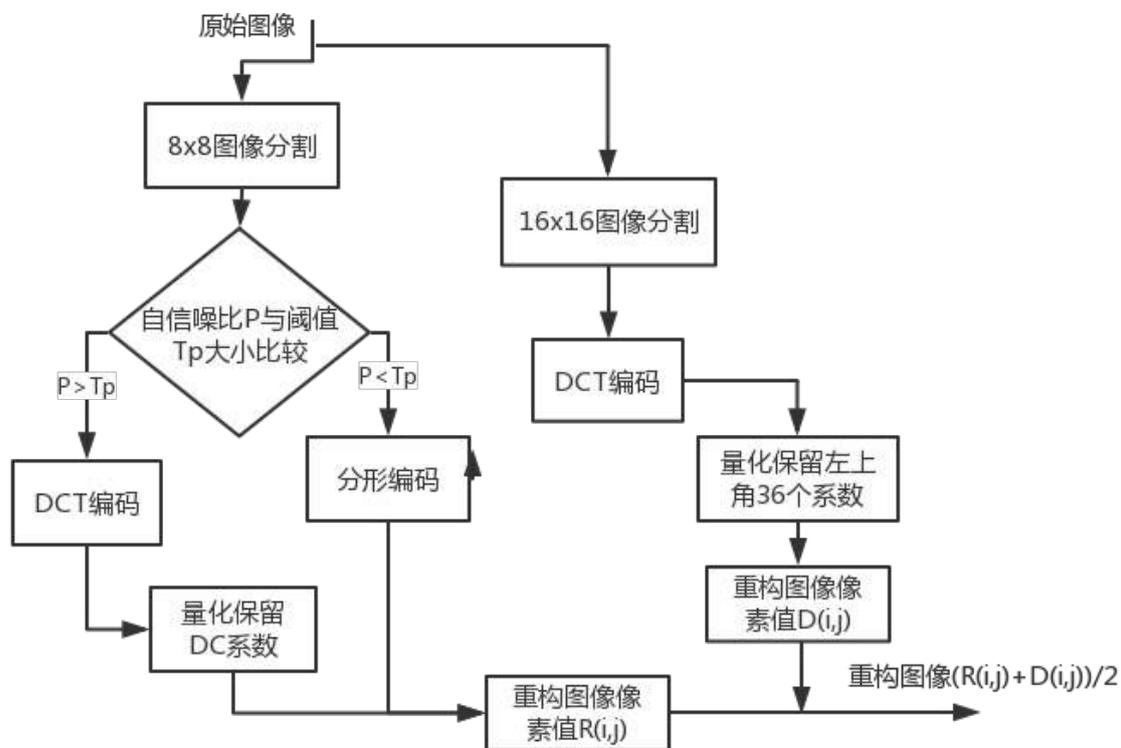


图 4.10 混合编码算法原理框图

第 5 章 实验结果与分析

5.1 实验设置

本文设计的所有算法的实验设备均为 MacBook Air,6 GHz Intel Core i5, 并利用 Matlab 工具对图像进行仿真测试。Matlab 工具提供了许多用于图像处理的函数^[28], 如 blkproc、zeros、reshape, 对于图像质量性能的计算也比较方便。测试的性能参数采用用时、峰值信噪比 (PSNR)、结构相似性 (SSIM)、压缩比 (Rate) 四个方面对上述算法进行比较。

测试图片选取选取 512×512 的标准 Lenna 测试图像作为实验的算法验证测试图片, 选取 Koadk 标准测试集的 24 张广泛用于比较图像压缩技术的 768×512 大小的 PNG 图片作为算法性能测试图片^[29]。

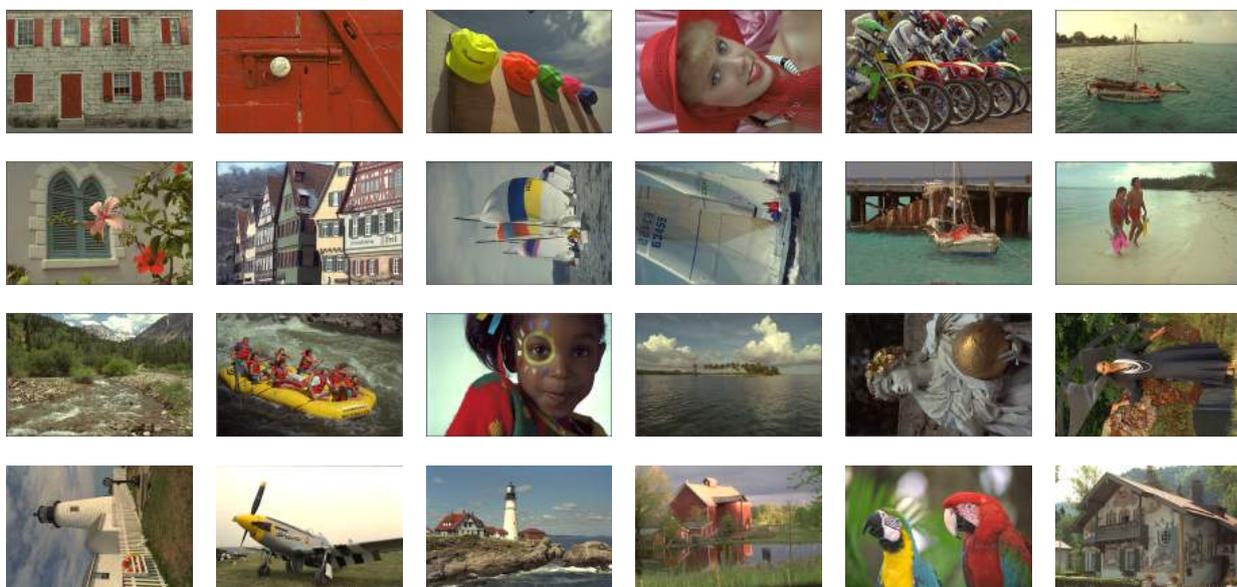


图 5.1 Koadk 标准测试集

编码后生成的压缩文件以 mat 形式存储, 在另一个函数加载获取其中的图像压缩数据以实现图像的恢复, 并在此过程中分别计算编码与解码用时, 分析图像性能。

5.2 DCT 编码算法分析

基于 DCT 变换的实验首先选取 512×512 的标准 Lenna 测试图像作为算法验证图片, 图像分割子块的大小为 8×8 。

实验结果如下图所示, 其中图 5.1(a) 为 JPEG 压缩的实现效果,(c)(d) 为基于阈值的分割改进算法的实现效果。



图 5.2 Lenna 图像的算法实现压缩效果对比

从图像的主观评价上来看，图像的恢复质量都不错。尽管 DCT 算法二拥有较高的压缩比，但在细节的表现上明显不如其他两种算法。可以通过下表显示的 PSNR、SSIM 数据更加直观地看到图像质量差异。

表 5.1 Lenna 图像的算法实现效果对比

算法	压缩比	编码用时 (s)	解码用时 (s)	PSNR(dB)	SSIM
DCT 算法一	2.01	9.14	4.56	28.2445	0.7951
DCT 算法二	8.81	5.58	1.47	23.9275	0.6106
DCT 算法三	1.40	4.81	2.81	29.9229	0.8690

下图 6.2 中表示了基于 DCT 算法一的原始图像、恢复图像、经熵压缩后获得的压缩数据的文件大小。可以发现尽管恢复图像的文件大小仍然有 246Kb，约占原始图像 474Kb 的一半。但是我们可以仅凭借 45Kb 的压缩数据 Encodefile.mat 而完成对原始图像有损压缩后的图像恢复，从而实现 10.5 的压缩比。数据传输过程中也仅需传输压缩数据以及压缩算法规则，从而实现缓解带宽、信道压力的目的。

Name	Size
Lenna.png	474 KB
restore.png	246 KB
EncodeFile.mat	45 KB

图 5.3 压缩数据文件大小

通过 Lenna 图像完成了对算法的验证性测试，证明 DCT 算法可以完成对图像有效地压缩。现通过对 Koadal 标准测试集的 24 张 PNG 图片的 DCT 变换压缩，计算平均值从而宏观比较算法时间以及压缩效果。

表 5.2 Koadk 图像集的实验数据

算法	编码用时 (s)	解码用时 (s)	PSNR(dB)	SSIM
DCT 算法一	25.95	12.19	26.4337	0.7811
DCT 算法二	18.18	10.97	23.1299	0.5996
DCT 算法三	10.50	6.13	27.0563	0.8314

从表 5.2 的实验数据可以看到算法二的 PSNR 较其他二者较低，算法三的峰值信噪比最高，结构相似性的情况与峰值信噪比类似。就用时而言，改进的 DCT 算法二、三使用阈值分割替代传统的量化表量化方式，相较于 DCT 算法一大幅缩短了编码、解码时间。

可以得出结果：改进的算法三在提高了图像恢复质量的同时，缩短了算法用时，较其他两个算法表现出了理想的压缩效果。下文也将采用 DCT 算法三与分形编码、混合编码进行算法性能比对，并称之为 DCT 改进编码。

5.3 分形编码算法分析

因为分形压缩编码时间过长，这部分的实验仅选取 512×512 标准 Lenna.png 图像为基本分形算法的迭代测试图像，而不取用 Koadak 测试集。

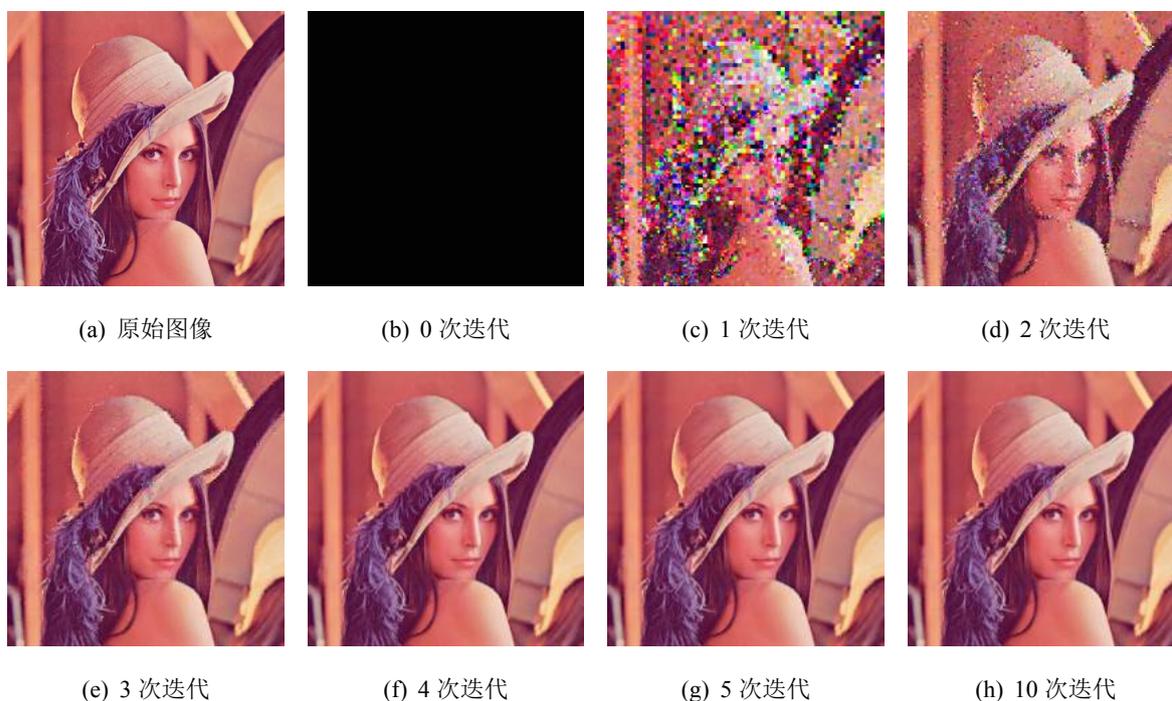


图 5.4 经分形迭代的 Lenna 恢复图像

实验测试结果如上图 5.4 所示。上述实验中 R 块大小为 8×8 ， D 块大小为 16×16 ，并

且取步长 $\delta_h = \delta_v = \delta = 16$ 。测试的性能参数是峰值信噪比 (PSNR)、结构相似性 (SSIM)、编码时间。

从图 5.3 可以看出，经由一幅与原始图像大小一样的任意图像（这里选取 512×512 全黑图像）可以完成对其的迭代恢复。第 1 次迭代恢复图像能看出图像轮廓但干扰噪点密集且繁多；第 2 次迭代恢复图像已经完成了整体轮廓的恢复，产生干扰的像素点也变少。第 3 次迭代以后的恢复图像在视觉感受上与原始图像的差异已经不大。根据上表 5.3 的 PSNR 数据绘制线性图 5.4，可以更加直观地发现图像的峰值信噪比随迭代次数增长渐渐趋近于某个值，第 5 次迭代后 PSNR 几近于 30.72dB，优于 DCT 算法三的 29.14dB，所以从节约耗时以及计算资源的情况考虑，可以第 5 次迭代恢复图像作为最后的收敛图像。

表 5.3 Lenna 图的实验数据记录

迭代次数	1	2	3	4	5	8	10
PSNR(dB)	12.1451	17.6567	24.3269	29.7560	30.6845	30.7181	30.7181
SSIM	0.1706	0.3527	0.6460	0.8127	0.8330	0.8337	0.8337
解码时间 (s)	0.1863	0.2553	0.4314	0.7225	0.9370	1.5177	1.8894

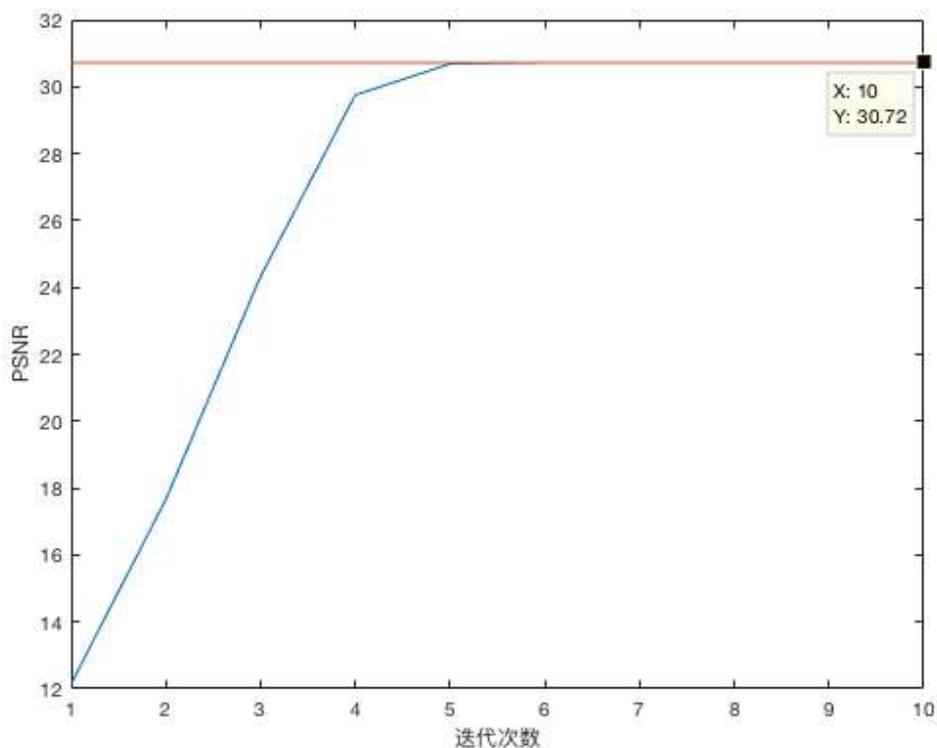


图 5.5 PSNR 与迭代次数关系图

尽管上表 5.3 中显示的解码时间较短，但是该算法的编码极其耗时，完成对 Lenna.png

测试图像的分形压缩编码共耗时 663.85s, 相较于 DCT 算法耗时过久, 约为其 170 倍。也正是因为编码时间过长、编码过程繁琐的原因, 基本分形算法在日常的图片压缩中并不常用。

5.4 混合编码算法分析

混合编码算法的实验首先选取 512×512 的标准 Lenna 测试图像作为算法验证图片, 图像分割子块, R 块和 D 块的大小分别为 8×8 和 16×16 , 取步长 $\delta_h = \delta_v = \delta = 16$, 图像重分割子块大小 16×16 。测试的性能参数是峰值信噪比 (PSNR)、结构相似性 (SSIM)、编解码时间, 选取不同的自信噪比阈值 T_p 进行实验测算, 解码迭代次数为 10 次, 实验结果如下。

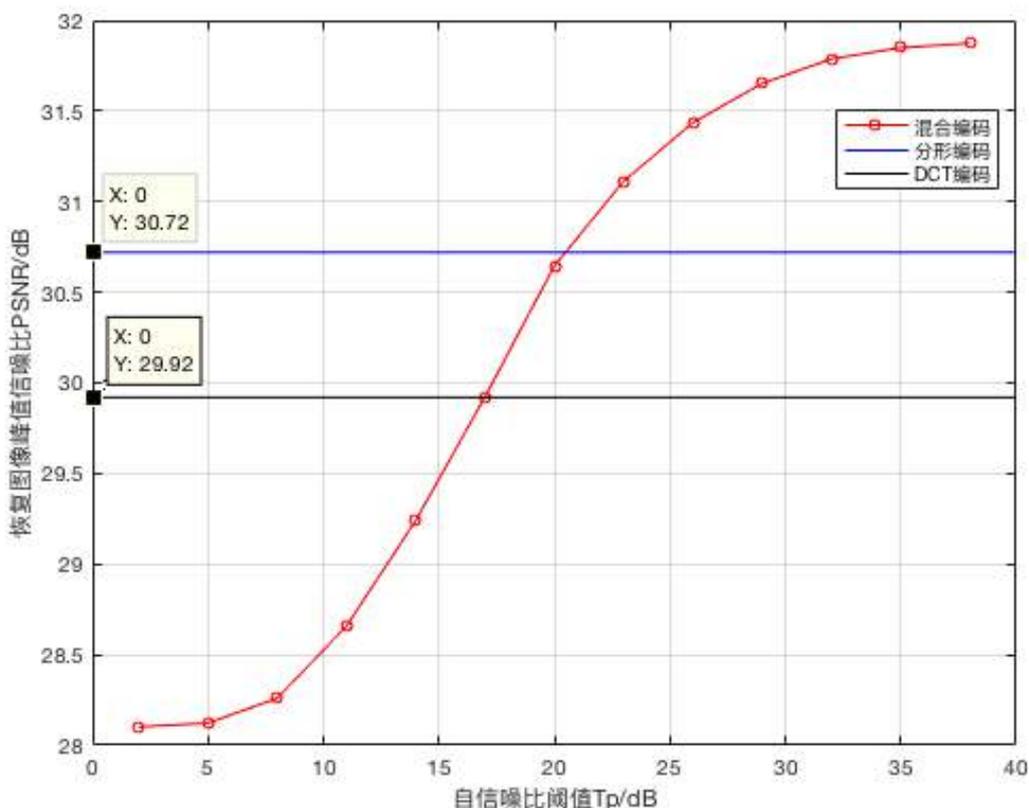


图 5.6 PSNR 与 T_p 关系图

可以在图 5.6 看到阈值 T_p 与 PSNR 的关系变换曲线, 其中在 $T_p = 17dB, T_p = 21dB$ 时分别超过了 DCT 编码 (DCT 算法三) 的 29.92dB 和分形编码的 30.72dB, 证明该算法在图像恢复质量上的提升。

下表 5.4 记录了各项图像质量评价的数据, 图 5.7 显示了 T_p 与编码时间的关系。

表 5.4 混合编码实验数据

阈值 T_p	2	11	17	20	23	29	35
PSNR(dB)	28.1021	28.6581	29.9234	30.6455	31.1126	31.6530	31.8491
SSIM	0.7754	0.7827	0.8026	0.8160	0.8267	0.8437	0.8543
编码时间 (s)	0.56	3.14	25.14	42.26	66.62	145.24	237.96
解码时间 (s)	0.96	0.75	0.79	0.86	0.97	1.14	1.19

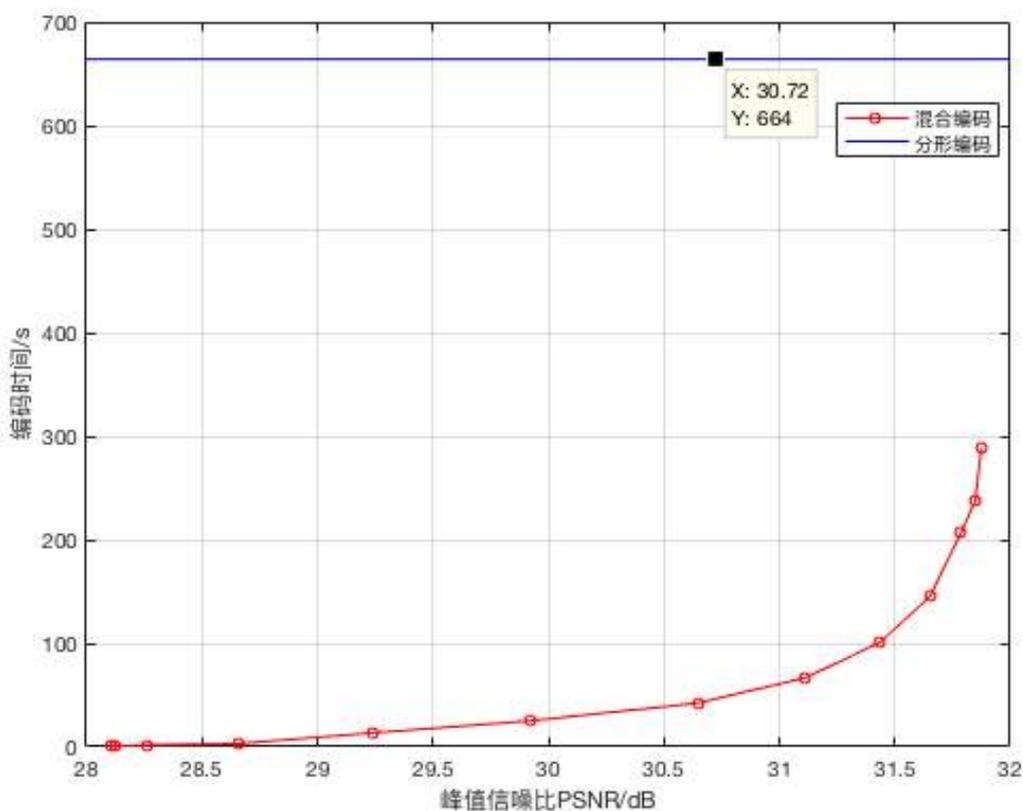


图 5.7 混合编码与基本分形编码时间对比

可以明确，混合编码保留了分形编码快速解压缩的特性，能在较短时间内完成对图像的恢复。同时编码时间较基本分形编码算法大幅度改善这一显著特点，但较 DCT 算法，编码时间仍需要克服的另一大难题。

下图 5.8 为 $T_p = 2, 17, 23, 35dB$ 下的 Lenna 恢复图像，可以看到除了 $T_p = 2dB$ 以外的恢复图像中，混合编码都很好消除了图像子块之间的块效应，从视觉感觉上已经看不到明显的质量差异，很好地还原了色彩、图形等所有图像特点，在细节表达上也足够充分。



图 5.8 混合编码的 Lenna 恢复图像

选取 $T_p = 21$ 对 Koadk 标准测试集的 24 张 PNG 图片的进行混合编码算法压缩实验，计算平均值与 DCT 算法进行比较，实验结果如下：

表 5.5 Koadk 图像集的混合编码实验

算法	编码用时 (s)	解码用时 (s)	PSNR(dB)
DCT 改进编码	10.50	6.13	27.0563
混合编码	75.69	1.57	29.6400

下图 5.9 显示了 Koadk 测试图集中的两张具有典型图片的压缩效果对比, 第一行为混合编码效果图, 第二行为 DCT 编码效果图, 第三行为基本分形编码效果图。

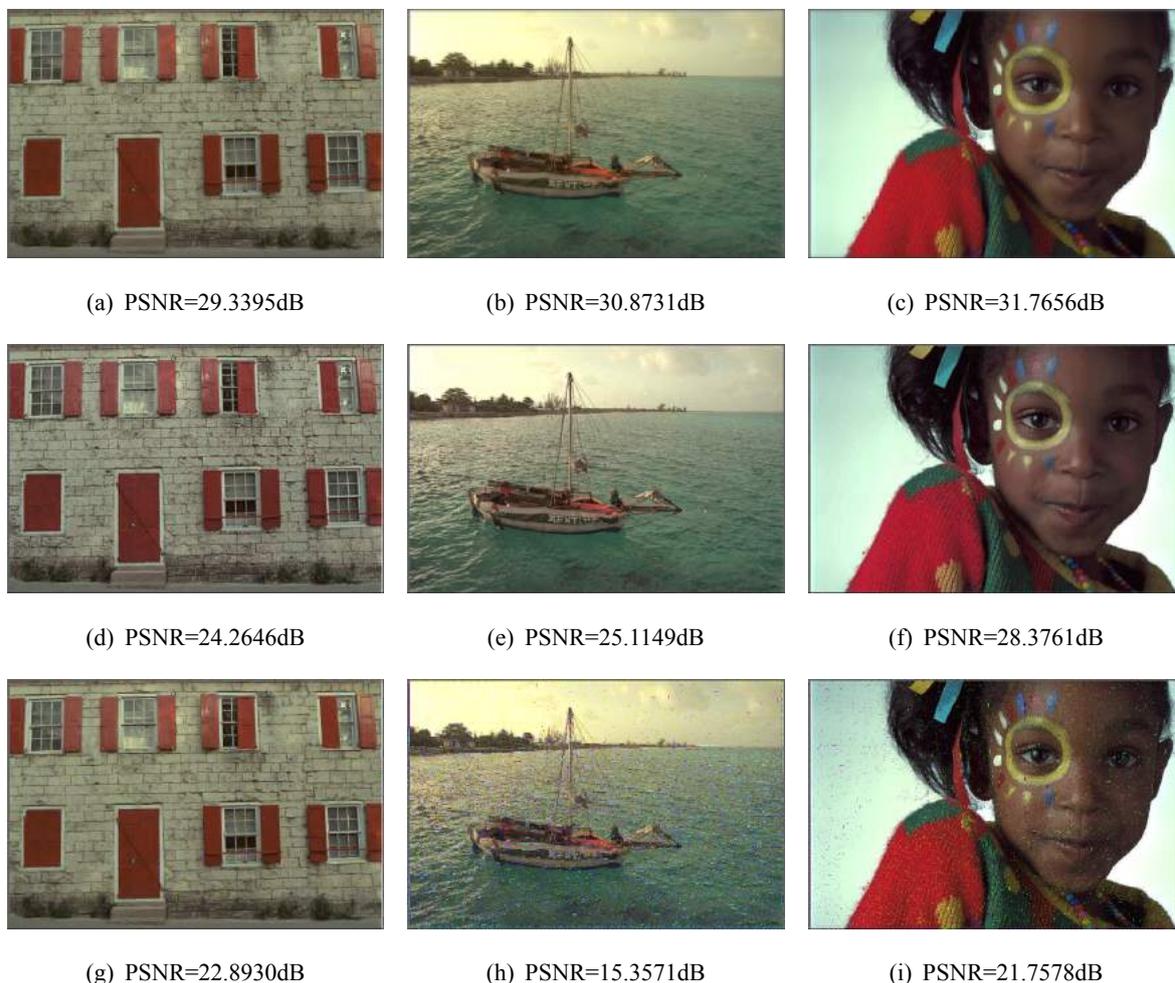


图 5.9 不同编码效果对比

综合上述图表数据中不难发现, 尽管混合编码其在解码用时与图像恢复质量 (PSNR、SSIM) 上都占据优势。还原图像在视觉感官上较另外两种编码也十分明显, 上述基本分形编码的还原图像中甚至出现了细小的肉眼可见的蓝绿色噪点, 极大影响了视觉感官。但是改进算法在编码时间上仍然大大逊色于 DCT 改进算法。

混合编码算法的提出在一定程度上是 DCT 编码与分形编码各自优点的结合, 去除块效应并提高图像恢复质量的同时较分形编码缩短了编解码时间。但其在编码时间上的弊端, 导致其在应用层面仍不如 DCT 算法。

第 6 章 总结与展望

6.1 总结

图像压缩编码技术是一种非常重要的技术，在近年来得到了各个方面科研力量的关注与支持，并在现实应用中取得了良好成效。本文先介绍了图像压缩编码的发展历程，列举了几个常见的有损、无损压缩编码方法，以信息论为基础阐述了图像的数字化表示与图像压缩编码的理论基础。

离散余弦变换（DCT）编码在互联网中得到了广泛应用，是 JPEG 图像编码标准的重要基石。用少量的数据表示原图像，具有算法简单和易于实现等优点。本文以 DCT 变换的数学理论为起点，详细描述了 DCT 图像编码的算法流程，并结合行程编码，实现了较好的图像压缩比与恢复质量。同时在传统 JPEG 算法的基础上提出了基于阈值分割的量化器改进算法，完成了对 DCT 变换编码的优化。

分形图像压缩方法具解压缩速度快、失真率小、运算速度与分辨率无关等特点，适用于需要快速访问高质量图像的应用。但分形编码还是有其局限性，其冗长的压缩步骤、过长的编码时间导致它将不能用与需要实时传输的压缩图像应用中。本文阐述了分形编码的基本理论知识，包括拼贴定理与迭代函数系统，对编码的算法步骤进行了详细描述并完成了对基本分形编码算法的迭代实验，

在此基础上，本文针对分形编码耗时过长的缺点，结合 DCT 变换的滤波特性与能量保持和紧缩特性，提出了一种混合编码算法。计算图像子块的自信噪比判断图像平坦程度有选择地进行分形或 DCT 编码，有效地将分形编码的编码耗时缩减至十分之一，并利用图像重分割减少了块效应的出现。实验结果表明，本文的提出的混合编码方法结合了传统分形编码和离散余弦变换编码各自的优点，较好地做到了提升图像恢复质量（PSNR）的效果，但在编码时间上仍不敌本文提出的基于阈值分割的 DCT 改进算法。

6.2 展望

在 DCT 编码部分中完成了 AC 系数的行程编码有效去除了冗余，原本在算法设计中加入了对 DC 系数的差分脉冲编码调制 (DPCM) 编码方式并完成了实现，但是在实际的实验测试中 DPCM 并没有有效地提升压缩比，故而将这部分删去。希望在日后的工作与研究中完善预测编码与 DCT 编码的有效结合。

分形编码以及混合编码的实现一定程度上能提升图像恢复质量，但是其编码时间过长仍然是个诟病，仍然无法撼动 DCT 编码与 JPEG 标准在图像压缩领域的地位。编码所花费的时间主要是最佳定义域块的搜索时间，本文提出的缩小值域块的方式具有一定局限性，在未来的分形图像研究上可以根据以下几点展开

1. 采取其他图像分割方式，而不是单一的块状固定分割。对定义域块和值域块进行特征分类，在各自特征类内搜索满足仿射变换的码本。
2. 缩小定义域块的搜索范围，即在领域内搜索而非全局搜索。
3. 与小波变换、矢量量化、遗传算法等其他编码的有效结合。

除此之外可以拓宽思路，尝试其他领域与图像压缩技术的有效结合。例如利用深度学习之卷积神经网络 (CNN) 进行图像压缩。

参考文献

- [1] 韩冉. 多/高光谱图像有损压缩技术研究 [D]. 西安: 西安电子科技大学, 2014
- [2] 万建邦. 图像有损压缩方法研究 [D]. 北京: 北京化工大学, 2005
- [3] Vartika Singh, OP Singh, GR Mishra. A Brief Introduction on Image Compression Techniques and Standards[J]. International Journal of Technology and ResearchAdvances Volume of. 2013
- [4] 王亚男, 张敬申, 冯杰. 数字图像压缩技术综述 [J]. 科教导刊 (电子版). 2014, (13):134–134
- [5] Ali H Hussein Al-nuaimi, Shyamaa Shakir Al-juboori, RJ Mohammed. Image Compression Using Proposed Enhanced Run Length Encoding Algorithm[J]. arXiv preprint arXiv:180400589. 2018
- [6] David A Huffman. A method for the construction of minimum-redundancy codes[J]. Proceedings of the IRE. 1952, 40(9):1098–1101
- [7] Jacob Ziv, Abraham Lempel. A universal algorithm for sequential data compression[J]. IEEE Transactions on information theory. 1977, 23(3):337–343
- [8] 郑伟, 崔跃利, 王芳. 基于小波变换的图像压缩编码研究综述 [J]. 通信技术. 2008, 41(2):83–85
- [9] 徐全元, 邵玉斌, 鲁莹. 基本矢量量化器的性能研究 [J]. 昆明理工大学学报: 理工版. 2004, 29(6):75–78
- [10] Michael F Barnsley, Stephen Demko. Iterated function systems and the global construction of fractals[J]. Proc R Soc Lond A. 1985, 399(1817):243–275
- [11] 徐飞. 浅析图像压缩编码方法 [J]. 电脑知识与技术. 2010, (23):6584–6586
- [12] 马锐. 基于分类的图像压缩算法研究 [D]. 西安: 西安科技大学, 2009
- [13] LP Hurd, MF Barnsley. Fractal image compression[J]. AK Peters, Wellesley. 1992
- [14] Wei-Yi Wei. An introduction to image compression[J]. National Taiwan University, Taipei, Taiwan, ROC. 2008
- [15] 罗迅. 静态图像有损压缩技术的研究 [D]. 武汉: 武汉理工大学, 2003
- [16] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity[J]. IEEE transactions on image processing. 2004, 13(4):600–612
- [17] Claude Elwood Shannon. A mathematical theory of communication[J]. ACM SIGMOBILE Mobile Computing and Communications Review. 2001, 5(1):3–55
- [18] Nuno Roma, Leonel Sousa. Efficient hybrid DCT-domain algorithm for video spatial downscaling[J]. EURASIP Journal on Advances in Signal Processing. 2007, 2007(1):057291

- [19] William B Pennebaker, Joan L Mitchell. JPEG: Still image data compression standard[M]. Springer Science & Business Media, 1992
- [20] Majid Rabbani, Paul W Jones. Digital image compression techniques[M], vol. 7. Bellingham, Washington USA: SPIE Press, 1991
- [21] 赵珊, 赵倩. 基于 DCT 系数的 JPEG 图像检索算法 [J]. 计算机工程. 2010, 36(19):190–192
- [22] C He, SX Yang, X Xu. Fast fractal image compression based on one-norm of normalised block[J]. Electronics Letters. 2004, 40(17):1052–1053
- [23] Arnaud E Jacquin. Image coding based on a fractal theory of iterated contractive image transformations[J]. IEEE transactions on Image Processing. 1992, 1(1):18–30
- [24] 张济忠. 分形 [M]. 北京: 清华大学出版社, 1995
- [25] 张石生. 不动点理论及其应用 [M]. 重庆: 重庆出版社, 1984
- [26] 夏道行, 吴卓人, 严绍宗等. 实变函数论与泛函分析 (上册· 第二版修订本)[M]. 北京: 高等教育出版社, 2010
- [27] 何传江. 分形图像编码技术的算法研究 [D]. 重庆: 重庆大学, 2004
- [28] Inc MathWorks. MATLAB: the language of technical computing. Desktop tools and development environment, version 7[M], vol. 9. MathWorks, 2005
- [29] Rich Franzen. Kodak lossless true color image suite[J]. source: <http://r0k.us/graphics/kodak>. 1999, 4

致谢

首先我要感谢我的导师钟珞教授，从论文的组织结构、行文逻辑、算法设计、实验设置与开展，钟珞教授都尽心尽力地在每一个环节对我提出了有效地指导。没有他的帮助，本文的成文将不会这么顺利。

感谢母校武汉理工大学尤其是计算机科学与技术学院所有的老师们。寒窗四年，是他们孜孜不倦的教诲鞭策了一个懵懵懂懂的少年，是他们在学术上对我提供了足够的支持与帮助扶持了一个刚刚迈入象牙塔的攀登者。这里要特别感谢钟欣老师，从最初在数据结构课堂上的结识，到后来在毕业设计阶段对我无微不至的关怀以及细心帮助。他的细致入微的教学风格和对课题的热情对我留下了深刻印象，也是正是数据结构这门课，打开了我对计算机这个领域的求知之门。

论文的顺利完成不仅仅需要学术上的支持，我还有许多人要感谢，感谢他们对我的包容与理解。这里要特别感谢相处了四年的三个室友金周、贾纪贝、罗文智，四年来他们给予了我从生活到学习各方面的无私帮助、互相扶持，也正是他们让我在这个遥远的异乡感到了家的温暖。

最重要的是，没有我的家人这一切都是空谈。感谢父母的养育之恩，以及在我求学的四年期间一直对我坚定不移的支持，我永远感激不尽。本文作为你们无条件的爱与鼓励的证明。

从四年前从千里之外的家乡来到这个陌生的城市，有过感动也有过感慨。白驹过隙，眨眼就到了毕业的时季；思绪万千，不舍得道出一声别离。在这里，将“莫愁前路无知己，天下谁人不知君”，送给即将分别的老师、同学，也送给即将远赴重洋在另一个国度继续求学之路的我自己。未来的路并不可知，只是希望我能继续保持这颗求知的心，保持对计算机技术的执著热爱，保持对科研的不尽探索，不忘初心，方得始终。

最后，再次对关心、帮助我的老师和同学表示衷心的感谢！