

基于图像纹理的混合图像压缩算法

屠思红, 白恩健

东华大学信息科学与技术学院, 上海

收稿日期: 2024年6月9日; 录用日期: 2024年6月29日; 发布日期: 2024年7月11日

摘要

针对经典图像压缩编码中跨纹理边缘图像块编码性能差、存在非稀疏表示等问题, 提出一种基于图像纹理的混合图像压缩编码方案。在图傅里叶变换中, 利用图像块具有不同纹理特征的特点对图拉普拉斯矩阵进行构造并实现图像块纹理特征的自适应选择, 以提高图像块的压缩编码性能; 再通过率失真优化实现离散余弦变换和图傅里叶变换的最优变换选择。实验结果表明, 在相同比特率下, 混合图像压缩算法的峰值信噪比比单纯的离散余弦变换平均高出1.08 dB, 并且在相同的量化步长下具有更好的图像质量。

关键词

图像压缩, 图傅里叶变换, 纹理相关表示, 离散余弦变换

Hybrid Image Compression Algorithm Based on Image Texture

Sihong Tu, Enjian Bai

School of Information Science & Technology, Donghua University, Shanghai

Received: Jun. 9th, 2024; accepted: Jun. 29th, 2024; published: Jul. 11th, 2024

Abstract

A hybrid image compression scheme based on image texture is proposed to address the problems of poor coding performance of image blocks across texture edges and the existence of non-sparse representation in classical image compression coding. Firstly, the image blocks with different texture features are utilized in the graph Fourier transform to construct the graph Laplace matrix and realize the adaptive selection of texture features of image blocks to improve the compression coding performance of image blocks. Secondly, the optimal transform selection is achieved by rate-distortion optimization in discrete cosine transform and graph Fourier transform. Experi-

mental results show that the hybrid image compression algorithm outperforms a simple discrete cosine transform by an average of 1.08 dB in peak signal-to-noise ratio at the same bit rate, and it has better image quality at the same quantization step.

Keywords

Image Compression, Graph Fourier Transform, Texture Correlation Representation, DCT

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

图像压缩作为计算机应用方向的重要研究内容, 如何能使用较少的比特数以获得高分辨率和较好品质的图像始终是不懈研究的重要方向。离散余弦变换(discrete cosine transform, DCT)作为一种经典的图像压缩变换方案, 被认为是一种准最佳变换[1]。但 DCT 对于降低谱的相关性方面不如 K-L 变换(Karhunen-Loeve Transform, KLT)有效, 其假设相邻像素之间的相关性趋于 1, 但如跨纹理边界的像素相关性明显小于 1, 这就导致 DCT 压缩在明显纹理边界的像素块上的稀疏性变换存在问题, 而这将导致对应像素块较差的编码性能。

随着图信号处理技术的不断发展, 为了实现更加高效的变换, 提出了图傅里叶变换(graph Fourier transform, GFT) [2]。通过图的方式来表示图像, 其中图像的像素表示为图的节点, 图像的像素相关性表示为边缘权重, 图的连通性被编码在图拉普拉斯函数中。Pavez 等[3]提出了图模板变换, 它通过利用已知的由图模板表示的信号的先验信息来近似 KLT。Hu 等[4]提出了一种基于多分辨率图傅里叶变换的编码方案, 通过考虑不同像素块信号变换系数的稀疏性以及变换后分布的紧凑性, 自适应的确定最佳的图傅里叶变换。Fracastoro 等[5]提出了一种新的图学习算法, 对图结构进行设计来优化整体的率失真性能, 通过最小化其图傅里叶系数的稀疏性, 并将图描述的代价引入到优化问题, 定义图像信号的变换。Uruma 等[6]提出了一种基于图变换的彩色图像编码算法, 使用 JPEG 编码对亮度图像进行编码, 将色度图像转换为图频谱图像进行压缩, 并通过存储称为代表性像素的多个彩色像素来压缩彩色图像。Gnutti 等[7]提出了一组基于对称的图傅里叶变换编码方案, 其变换集很好地表示了自然图像。Yan 等[8]提出在笛卡尔积图上定义多维图信号, 将二维图变换扩展到多维图分数傅里叶变换。

GFT 能提高图像的紧凑表示, 为能量信号集中在低频提供了可能性[9]。其关键在于构造 GFT 的拉普拉斯矩阵[10] [11]。上述的方案没有明确如何定义一个图, 并在图变换系数的稀疏性与图变换编码成本之间进行平衡。目前定义边缘权重的常用方法是通过高斯核加权函数[12], 但对图像而言节点之间的距离是不变的, 通过高斯核函数定义的像素相关性往往只能代表一类像素纹理块。

本文利用不同的边缘权重定义函数代表不同的图像纹理块, 提出基于图像纹理和图傅里叶变换的混合图像压缩算法。通过研究不同图像块纹理的变化, 将不同纹理图像的相关表示为不同的图拉普拉斯矩阵, 这将会比单纯的高斯核函数所定义的图拉普拉斯矩阵在描述图像方面更加精准, 并对图像纹理在低复杂度下进行自适应的图拉普拉斯矩阵选择。与现有的独立变换图像编码方法不同, 本文从 DCT 或基于纹理相关的 GFT (texture based graph Fourier transform, TGFT)中选择最优变换[13]。实验结果表明, 该变换方法在稀疏表示方面优于 DCT。

2. 图像纹理相关的图傅里叶变换

2.1. 图傅里叶变换

图可以表示为 $\mathcal{G} = \{v, \varepsilon, \mathbf{W}\}$, 其中 v 表示有限的顶点集, 并且 $|v| = N$ 。 ε 表示边集, $\mathbf{W} \in \mathbf{R}^{N \times N}$ 表示加权邻接矩阵。本文只考虑无向图, 并且默认权重值为非负值, 即 $W_{i,j} = W_{j,i} \geq 0$ 。图拉普拉斯矩阵可以表示为 $\mathbf{L} = \mathbf{D} - \mathbf{W}$, 其中度矩阵 $\mathbf{D}$ 是对角矩阵, 其元素定义为

$$D_{i,j} = \begin{cases} \sum_{j=1}^N W_{i,j}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

由于 $\mathbf{L}$ 是一个实对称矩阵, 它存在完备的正交特征向量集合 $\{u_i\} (i=1, \dots, N)$, 以及一组实非负特征值 $\lambda_i (i=1, \dots, N)$ 满足 $\mathbf{L}u_i = \lambda_i u_i$ 。所以矩阵 $\mathbf{L}$ 可以表示为 $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, 其中 $\mathbf{U} \in \mathbf{R}^{N \times N}$ 是 $\mathbf{L}$ 的特征向量矩阵, 对应特征向量按列排序, $\mathbf{\Lambda} \in \mathbf{R}^{N \times N}$ 是对角特征值矩阵, 特征值按递增顺序排序。

图信号 $f \in \mathbf{R}^N$ 是在图 $\mathcal{G}$ 的节点上定义的实值函数。利用拉普拉斯矩阵的特征向量定义信号 f 的 GFT [14] [15] 为

$$\hat{f} = \mathbf{U}^T f \quad (2)$$

逆图傅里叶变换(inverse graph Fourier transform, IGFT)为

$$f = \mathbf{U}\hat{f} \quad (3)$$

本文的算法通过对图像像素块进行 GFT 来消除图像信号相邻像素之间的相关性, 从而让图像的能量更加集中在低频。图 1 为对图像像素 8×8 分块并纵向排列后进行一维图变换得到的谱图图像示例。从图 1 可以看到背景部分即纹理平滑区域能量分布较少, 而纹理丰富部分能量分布较多。通过对不同纹理的图像块进行底层图结构分类构建, 使得对应的图拉普拉斯矩阵能够更准确地模拟图像信号之间的相关性。

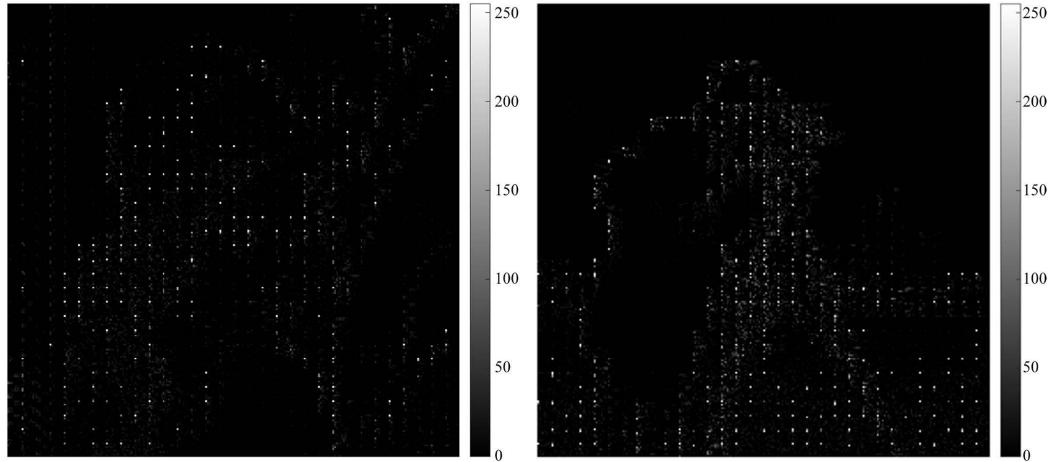


Figure 1. The effect of GFT

图 1. 图变换的效果

2.2. 图结构构建

为了有效地压缩图信号, 考虑四种加权邻接矩阵的构建方式, 使得图信号由图拉普拉斯矩阵的四种特征向量表示, 精确的构建图拉普拉斯矩阵对提高图像压缩性能起到关键性作用。

2.2.1. 距离加权图

考虑到一些图像块颜色差异和距离具有较强的相关性, 距离越远颜色差距越大。对于这类图像纹理块使用距离加权图去近似图像像素之间的相关性。图像的边缘权重通过阈值高斯核加权函数定义为

$$W_{i,j} = \begin{cases} \exp\left(-\frac{[dist(i,j)]^2}{\theta^2}\right), & dist(i,j) \leq \kappa \\ 0, & otherwise \end{cases} \quad (4)$$

其中 $dist(i,j)$ 表示顶点 i 和顶点 j 之间的距离, θ 是高斯参数, 定义 $\theta = 0.15 \max_{i,j} |dist(i,j)|$ [5], κ 是预定义的距离阈值。

2.2.2. 距离 - 颜色加权图

通过定义距离 - 颜色加权图将距离和色彩值相结合, 并将其应用在复杂纹理图像块中。增加色彩预测值的能为图信号构建提供更多的信息量, 对像素点之间的建模提供更高的准确性。图的距离颜色关联的边缘权重被定义为

$$W_{i,j} = \begin{cases} \exp\left(-\frac{[dist(i,j)]^2}{\theta^2}\right) \exp\left(-\frac{[pred(i,j)]^2}{\delta^2}\right), & dist(i,j) \leq \kappa \text{ and } pred(i,j) \leq \tau \\ 0, & otherwise \end{cases} \quad (5)$$

其中 $pred(i,j)$ 表示顶点 i 和顶点 j 之间的预测色差, 通过相邻图像块与边缘图进行编码预测得到。 δ 表示预测色差的中位数绝对偏差, κ 是预定义的距离阈值。

2.2.3. 边缘划分图

对于平滑纹理图像块, 通过对图像边缘进行划分的方式, 来判断图像全局平滑还是局部平滑。对于局部平滑图像块, 根据其图像边缘进行权重值的改变。对于非同一边缘内部的像素点, 对其边缘权重值赋 0, 同一边缘内部顶点进行四连通连接操作。图 2 是一个 2×2 像素块构造邻接矩阵的例子, 根据边缘轮廓划分不同像素值顶点, 并连接相似像素值顶点来构建图形, 并给出对应的权重矩阵。

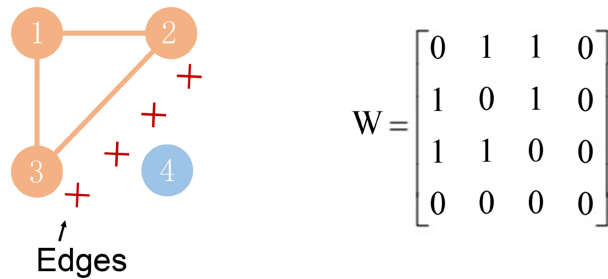


Figure 2. Constructing adjacency matrix from a 2×2 pixel block

图 2. 2×2 像素块构造邻接矩阵

2.2.4. 连通图

全局平滑的图像块意味着图像块内部具有相似的像素值, 将图顶点之间的权重值设置为 1 可以表示顶点之间得到强相关性。但若对图顶点进行全连接则会增加图像块的计算复杂度, 为了使计算复杂度可控, 本文提出四种方案对图顶点之间进行连通。分别是顶点最多两个度的 Hilbert、Zigzag 连接以及四连通和八连通方案, 如图 3 所示。最后通过实验确定使用四连通图作为全局平滑的图像块在图上的矩阵构造方式。

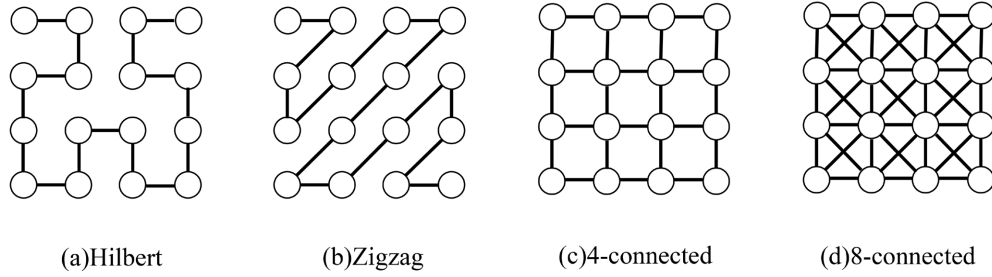


Figure 3. Global smoothing image block construction
图 3. 全局平滑图像块构造

2.3. 自适应选择方案

利用设定的判断阈值对图像块进行筛选, 根据图像块的颜色相关性度量、纹理复杂度度量及相似性度量结果与判断阈值进行比较, 使得图像块可以对应于合适的图构造方案。

2.3.1. 距离颜色相关性度量

如果对于一个图像块, 其颜色和距离存在很强的相关性, 那么可以通过距离来设置其权重值, 在避免图形表示的额外开销的同时可以建模像素点之间颜色的相关性。本文通过 Pearson 相关系数[16]去衡量距离与颜色相关性程度

$$\rho(\mathbf{D}, \mathbf{C}) = \frac{1}{M-1} \sum_{i=1}^M \left(\frac{D_i - \mu_D}{\sigma_D} \right) \left(\frac{C_i - \mu_C}{\sigma_C} \right) \quad (6)$$

其中 $\mathbf{D}$, $\mathbf{C}$ 分别是距离变化矩阵和颜色变化矩阵, M 为连接的边的数量。 $\mu_D(\mu_C)$ 以及 $\sigma_D(\sigma_C)$ 分别表示对应矩阵的平均值和标准差。

2.3.2. 纹理复杂度度量

当图像块的距离与颜色相关度不强时, 可以从图像最直观的颜色特征来建模图像块内部的颜色相似度。通过结合图像颜色种类和空间分布的特征复杂度的方式求解图像块颜色特征的复杂度[17], 并将图像块分为复杂纹理的图像块与平滑纹理图像块。给定一个一般集合, 计算其复杂度的公式为[18]

$$C = - \sum_{i=1}^k n_i \log \left(\frac{n_i}{N} \right) \quad (7)$$

其中 C 表示对应图像块特征的复杂程度, k 表示集合中不同特征值的个数, n_i 和 N 分别表示每种特征纹理对应的个体个数和个体总数。对于图像颜色特征复杂度可以表示为

$$C_t = C_s + \mu \sum_{i=1}^k \frac{C_i n_i}{N} \quad (8)$$

其中 μ 是权重因子, C_s 表示图像块的颜色种类复杂度, C_i 和 n_i 分别表示第 i 种颜色空间的分布复杂度和像素个数, C_t 表示结合颜色种类复杂度和颜色空间分布复杂度的图像块总复杂度。

2.3.3. 相似性度量

对平滑纹理图像块进行细化, 划分成全局平滑和局部平滑, 并利用不同的图结构进行图拉普拉斯矩阵构建, 以此提高图像表示的准确性。具体来说, 相似性度量利用边缘划分矩阵统计图像块内部的光滑区域的数量, 对于光滑子区域存在两个及两个以上的图像块, 将其视作局部平滑图像块, 否则为全局平滑图像块, 并使用对应的图结构进行构建。

3. 纹理相关的图变换图像压缩方案

3.1. 编码方案

图 4 给出了基于图像纹理的混合图像压缩编码框架。对图像的编码操作分为三个步骤。

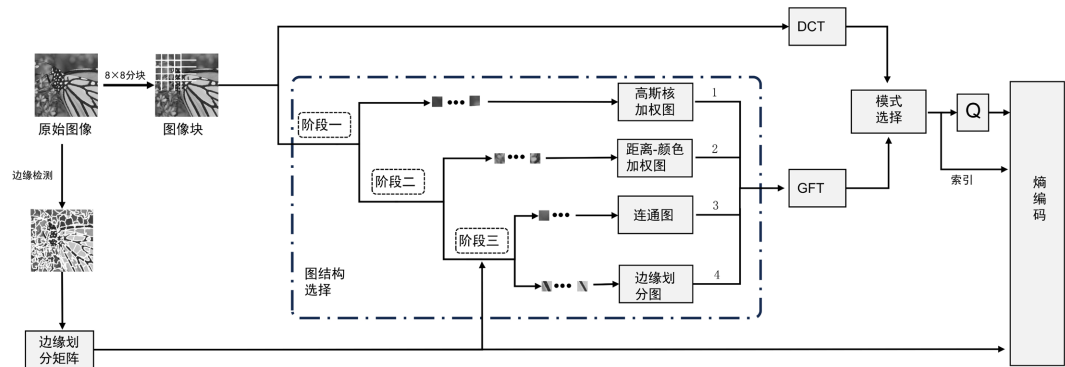


Figure 4. Image compression coding framework
图 4. 图像压缩编码框架

首先，在编码器端利用图像梯度的硬阈值检测图像的明显边界得到图像对应的边缘划分矩阵。

其次，使用两种变换对图像块进行变换。一是对图像块进行 DCT 变换；二是对其进行图傅里叶变换，具体是先进行分选操作，将每个图像块进行距离颜色相关性度量(阶段一)，颜色与距离相关性高的图像，其图拉普拉斯矩阵按照高斯核加权图的构造方式进行构建。反之进行纹理复杂度计算(阶段二)后继续划分成复杂纹理图像块与简单纹理图像块，针对复杂纹理图像块，其图拉普拉斯矩阵为距离颜色加权矩阵。而针对简单纹理图像块，根据阶段二的图像检测结构中是否存在明显边界划分成全局平滑图像块和局部平滑图像块并分别进行连通图和边缘划分图的搭建(阶段三)。对所有的图像块进行对应图结构构建后对其进行相应的图傅里叶变换。

第三，在从两个候选变换中选择每个图像块对应的最优变换后，对最终选定的变换系数进行量化和熵编码，并标识所选变换的变换索引进行熵编码，以便在解码器端执行适当的反变换。

为了清楚整个图像压缩编码流程并加以实现，根据该算法的编码框架，使用伪代码对其整体流程加以阐述，如表 1 所示。

Table 1. Image compression coding algorithm pseudo-code
表 1. 图像压缩编码算法伪代码

算法输入：图像 I ，参数 α, β	
算法输出：边缘划分矩阵 E ，索引 ind ，谱信号 $\hat{f}$	
1.	$E \leftarrow \text{Edge_Division}(I)$ //生成边缘划分矩阵
2.	$block_size \leftarrow 8$
3.	$(block_m, block_n) \leftarrow \text{How_blocks}(I, block_size)$
4.	for $i \leftarrow 1$ to $block_m$
5.	for $j \leftarrow 1$ to $block_n$
6.	$P' \leftarrow P((i-1) \times block_size + 1: i \times block_size, (j-1) \times block_size + 1: j \times block_size)$
7.	$(RDcost_dct, ind, \hat{f}_{dct}) \leftarrow \text{DCT}(P')$ //离散余弦变换
8.	$stage1 \leftarrow \text{Pearson_corr}(P')$ //相关系数
9.	$stage2 \leftarrow \text{Compl_eval}(P')$ //复杂度评估
10.	if $stage1 > \alpha$ then
11.	$(x, ind_gft) \leftarrow \text{Gaussian_graph}(P', block_size)$ //高斯图

续表

```

12.   else if stage2 >  $\beta$  then
13.      $(x, ind\_gft) \leftarrow Dis\_color\_graph(P', block\_size, I)$  //距离颜色图
14.   else if  $E(i, j) = 0$  then
15.      $(x, ind\_gft) \leftarrow Connected\_graph(P', block\_size)$  //连通图
16.   else
17.      $(x, ind\_gft) \leftarrow Edge\_graph(P', block\_size, E)$  //边缘划分图
18.   end if
19.    $(RDcost\_gft, \hat{f}\_gft) \leftarrow GFT(x)$  //图变换
20.    $(\hat{f}, ind) \leftarrow \min(RDcost\_gft, RDcost\_dct)$  //选择最优变换对应的谱信号和索引
21. end for
22. end for
23.  $\hat{f} \leftarrow quant(\hat{f})$  //量化

```

3.2. 解码方案

图 5 给出了基于图像纹理的混合图像压缩解码框架。

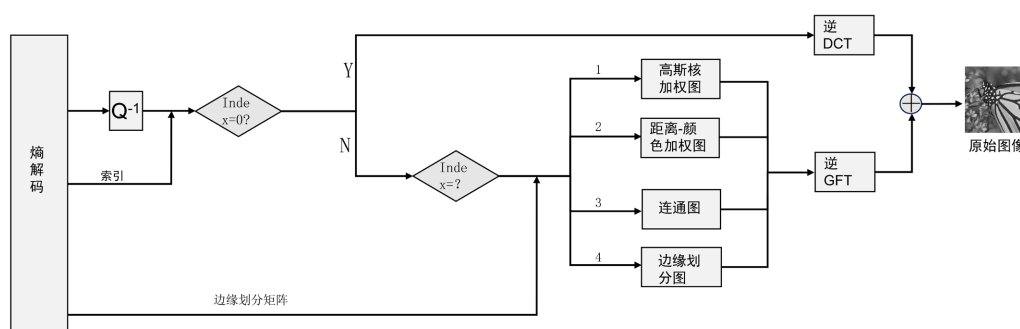


Figure 5. Image compression decoding framework

图 5. 图像压缩解码框架

在解码器端首先对图像块进行逆量化, 然后根据索引进行选择变换的判断, 对进行 DCT 变换的图像块进行逆 DCT 变换, 相应的对 GFT 变换具体使用的图结构进行判断后选择相应的变换系数进行逆图傅里叶变换。其中针对边缘划分图的逆变换需要根据得到的边缘划分情况进行变换系数的计算后再进行逆图傅里叶变换。最后对所得结果进行图像重建。

解压缩过程是压缩过程的逆过程, 根据算法的解码框架, 使用伪代码对其整体流程加以阐述, 如表 2 所示。

Table 2. Image compression decoding algorithm pseudo-code

表 2. 图像压缩解码算法伪代码

算法输入: 边缘划分矩阵 E , 索引 ind , 谱信号 $\hat{f}$

算法输出: 重建图像 I'

```

1.  $\hat{f} \leftarrow iquant(\hat{f})$  //反量化
2.  $block\_size \leftarrow 8$ 
3.  $(block\_m, block\_n) \leftarrow How\_blocks(I, block\_size)$ 
4. for  $i \leftarrow 1$  to  $block\_m$ 
5.   for  $j \leftarrow 1$  to  $block\_n$ 
6.      $\hat{f}' \leftarrow \hat{f}((i-1) \times block\_size + 1 : i \times block\_size, (j-1) \times block\_size + 1 : j \times block\_size)$ 
7.      $ind' \leftarrow ind(i, j)$ 
8.     if  $ind' = 1$  then

```

续表

```

9.       $x \leftarrow \text{Gaussian\_graph}(block\_size)$ //高斯图
10.    elseif  $ind'=2$  then
11.       $x \leftarrow \text{Dis\_color\_graph}(f, block\_size, i, j)$ //距离-颜色图
12.    elseif  $ind'=3$  then
13.       $x \leftarrow \text{Connected\_graph}(block\_size)$ //连通图
14.    elseif  $ind'=4$  then
15.       $x \leftarrow \text{Edge\_graph}(block\_size, E)$ //边缘划分图
16.    end if
17.    if  $ind'=0$  then
18.       $f\_block \leftarrow \text{IDCT}(\hat{f})$ //逆离散余弦变换
19.       $\hat{f} \leftarrow \text{Built\_matrix}(f\_block)$ //当前块放入矩阵
20.    else
21.       $f\_block \leftarrow \text{IGFT}(\hat{f}, x)$ //逆图傅里叶变换
22.       $\hat{f} \leftarrow \text{Built\_matrix}(f\_block)$ //当前块放入矩阵
23.    endif
24.  endfor
25. endfor
26.  $I' \leftarrow \text{Rebuild}(f)$ //重建图像

```

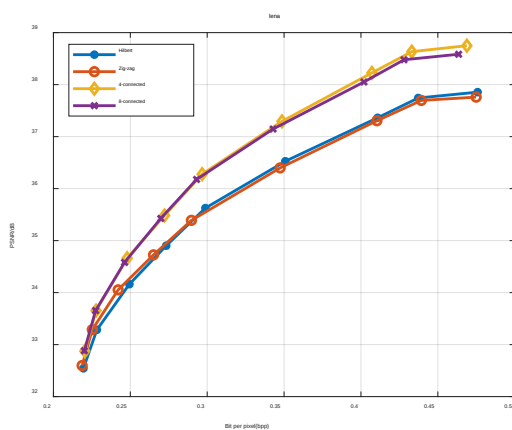
4. 实验结果

算法在 Intel Core i5-12500H CPU @2.5 GHz 和 16 GB 内存的 PC 上进行了测试并在 MATLAB 中实现。实验选择 10 幅经典灰度图像(Lena, Boat, Peppers, Cameraman, Couple, Barbara, Man, Hill, Mandri, Parrot)作为测试图像, 均为 256×256 的图像。

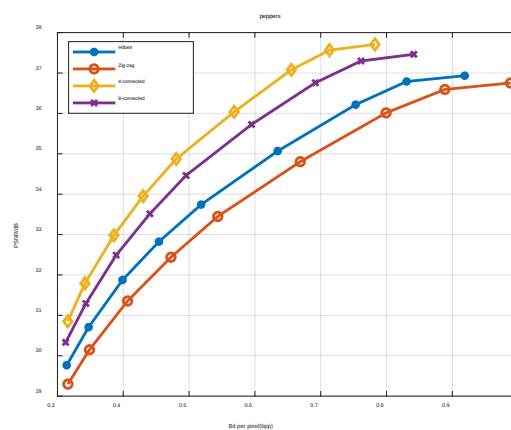
4.1. 参数设置

在全局平滑的图像块中, 图拉普拉斯矩阵选择为四连通图, 图 6 给出了图 3 中不同图构造方式的率失真曲线比较, 可以看到四连通图具有较好的性能。

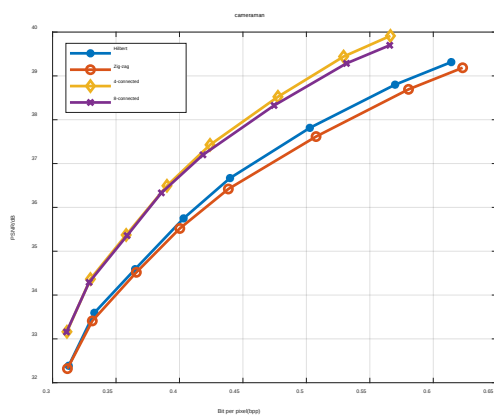
图像分选操作可以通过是否存在明显边界对全局平滑图像和局部平滑图像进行直接判断。距离颜色相关性度量以及纹理复杂度的判断阈值利用对测试图像的图像块进行随机采样得到的图像进行估计。首先得到样本图像块的率失真模型, 计算每个图像块在给出的四种图结构下的最小速率, 最后设置最大系数及最小熵作为对应阈值。由表 3 可见共进行 10 次随机采样并获得对应图像的阈值参数, 对其进行平均取值, 最后阈值确定为 0.50 和 0.67。纹理复杂度度量中设置为 1, 表示图像颜色种类复杂度及空间分布复杂度与图像本身的复杂度呈正相关。



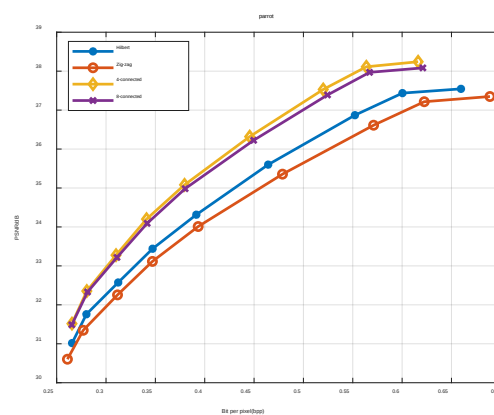
(a)



(b)



(c)



(d)

Figure 6. Comparison of RD performance for different construction schemes for global smooth blocks

图 6. 全局平滑块不同构造方案的 RD 性能比较

Table 3. Threshold parameter selection

表 3. 阈值参数选择

采样次数	阶段一	阶段二	采样次数	阶段一	阶段二
1	0.452	0.675	6	0.48	0.681
2	0.457	0.686	7	0.549	0.675
3	0.555	0.682	8	0.458	0.696
4	0.534	0.668	9	0.534	0.668
5	0.48	0.697	10	0.454	0.651



(a) 高斯加权图



(b) 距离 - 颜色加权图



(c) 边缘划分图



(d) 连通图

Figure 7. Block classification of the image Lena

图 7. Lena 图中不同图构造块分类

图 7 显示了一张图像内部不同的图像块使用不同的图结构的块分类结果, 可以明显看出对不同纹理图像块进行了一定的分类, 全局平滑的图像块多位于(d)中, 具有复杂纹理的图像块多位于(b)中。为了验证所提出的图构造方法具有一定的优势, 本方案与使用高斯核函数[12]的图变换方案和使用 KLT 的经典编码方案进行了比较, 其中 KLT 是对每个 8×8 图像块单独计算变换结果。三种方案的量化和熵编码采用了相同的方法。

4.2. 性能比较与分析

Table 4. Bjontegaard average gain in PSNR for natural image compared with DCT
表 4. 图像中 PSNR 的 Bjontegaard 平均增益与 DCT 对比

图像	本文算法	KLT	Gaussian graph
Peppers	0.38	0.35	0.14
Parrot	1.20	0.51	0.93
Lena	0.57	0.24	0.52
Boat	1.24	0.74	0.95
Cameraman	0.93	0.68	0.63
Barbara	1.85	1.43	1.21
Couple	0.75	0.65	0.43
Hill	0.58	0.38	0.42
Man	0.80	0.77	0.49
Mandri	2.50	2.05	2.13

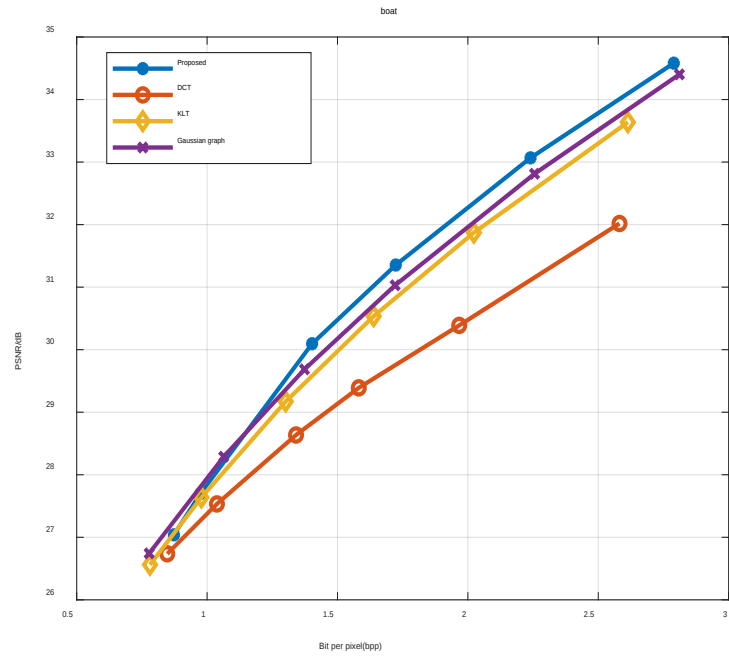


Figure 8. Rate distortion curve of image Boat
图 8. 图像 Boat 的率失真曲线

表4是与单纯的DCT方案相比,采用 Bjontegaard 度量[19]评估的峰值信噪比(peak signal-to-noise ratio, PSNR)平均增益的性能结果。相较于 DCT 方案,本文方案、KLT 方案和 Gaussian 图方案的平均增益分别为 1.08 dB、0.3 dB 和 0.3 dB。图8展示了图像 Boat 的率失真曲线,由图8可以看出本文方案所带来的性能优势随着比特率的降低而逐渐减弱,这主要是由于为表示图结构的标志,局部与全局平滑图像块的分类等额外编码存在一定的开销。随着不断增加的量化步长,额外开销所占的比例在整个比特流中越来越大,从而不断削弱了本方案所带来的性能优势。

图9展示了图像 peppers 的直观视觉比较。与其他方案相比,本文的方案具有最清晰边界和最干净表面的图像。与 DCT 相比,本文的方案在图像边缘的处理更加清晰。本文利用标量量化的方式比较在相同量化步长下本方案与 DCT 的压缩比及峰值信噪比。由表5可知当量化步长为 40 时,本方案的压缩比高于 DCT,但在图像质量即峰值信噪比和结构相似性(structural similarity index, SSIM)上相较于 DCT 具有更好的效果。图像编码及解码执行时间如表6所示,本文方案在编码时长上高于其他方案,这是因为在编码器端进行实时特征分解,并且最优变换的选择增加了编码执行时间。但在解码器端本文方案的性能与 Gaussian 方案相近。

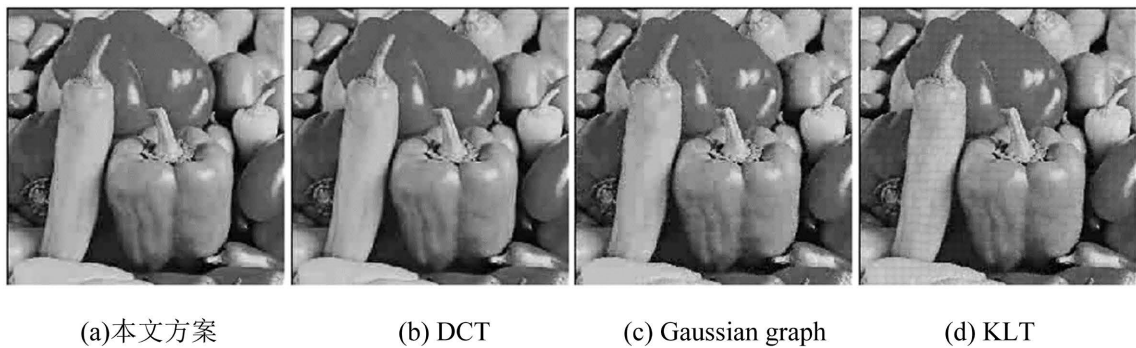


Figure 9. The subjective quality comparison among different compression schemes (0.9 bpp)

图9. 不同压缩方案的主观质量比较(0.9 bpp)

Table 5. Comparison of compression ratio and image quality with the same quantization step (40)

表5. 相同量化步长(40)下压缩比与图像质量比较

图像	本方案			DCT		
	压缩比	PSNR/dB	SSIM	压缩比	PSNR/dB	SSIM
Parrot	1.10	30.62	0.867	0.73	27.63	0.817
Peppers	1.02	31.33	0.871	0.69	28.09	0.804
Lena	0.68	31.50	0.848	0.47	29.02	0.786
Boat	1.38	30.11	0.811	0.83	26.41	0.740
Cameraman	0.99	30.68	0.873	0.69	27.88	0.829
Barbara	1.34	28.66	0.870	0.99	26.03	0.798
Couple	1.14	28.87	0.829	0.82	26.27	0.749
Hill	0.99	29.13	0.796	0.74	27.10	0.734
Man	1.11	28.77	0.809	0.81	26.40	0.731
Mandri	1.66	27.07	0.818	1.19	24.11	0.716

Table 6. Execution time for encoding and decoding images
表 6. 编码和解码图像的执行时间

方案	编码时长/s	解码时长/s
本文算法	14.989	1.364
DCT	1.341	1.136
KLT	1.867	1.174
Gaussian	2.502	1.317

5. 总结

本文提出了一种基于图像纹理的图傅里叶变换编码方案,用于自然图像的压缩。与 DCT 等变换所不同的是,本文的方案针对不同纹理特征的图像块定义不同的图拉普拉斯矩阵,并且自适应地选择不同的图模式。通过将 TGFT 与 DCT 结合,在图像块的变换选择上可以实现在 DCT 和 TGFT 之间最优选择。在自然图像上的测试结果表明所提出的变换方案优于经典的 DCT 和其他的图变换方案,实现了更加高效的图像数据压缩。

参考文献

- [1] Ahmed, N., Natarajan, T. and Rao, K.R. (1974) Discrete Cosine Transform. *IEEE Transactions on Computers*, **23**, 90-93. <https://doi.org/10.1109/t-c.1974.223784>
- [2] Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A. and Vandergheynst, P. (2013) The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains. *IEEE Signal Processing Magazine*, **30**, 83-98. <https://doi.org/10.1109/msp.2012.2235192>
- [3] Pavez, E., Egilmez, H.E., Wang, Y. and Ortega, A. (2015) GTT: Graph Template Transforms with Applications to Image Coding. 2015 *Picture Coding Symposium (PCS)*, Cairns, 31 May-3 June 2015, 199-203. <https://doi.org/10.1109/pcs.2015.7170075>
- [4] Hu, W., Cheung, G., Ortega, A. and Au, O.C. (2015) Multiresolution Graph Fourier Transform for Compression of Piecewise Smooth Images. *IEEE Transactions on Image Processing*, **24**, 419-433. <https://doi.org/10.1109/tip.2014.2378055>
- [5] Fracastoro, G., Thanou, D. and Frossard, P. (2020) Graph Transform Optimization with Application to Image Compression. *IEEE Transactions on Image Processing*, **29**, 419-432. <https://doi.org/10.1109/tip.2019.2932853>
- [6] Uruma, K., Konishi, K., Takahashi, T. and Furukawa, T. (2019) Colorization-Based Image Coding Using Graph Fourier Transform. *Signal Processing: Image Communication*, **74**, 266-279. <https://doi.org/10.1016/j.image.2018.12.011>
- [7] Gnutti, A., Guerrini, F., Leonardi, R. and Ortega, A. (2021) Symmetry-Based Graph Fourier Transforms: Are They Optimal for Image Compression? 2021 *IEEE International Conference on Image Processing (ICIP)*, Anchorage, 19-22 September 2021, 1594-1598. <https://doi.org/10.1109/icip42928.2021.9506636>
- [8] Yan, F. and Li, B. (2022) Multi-Dimensional Graph Fractional Fourier Transform and Its Application to Data Compression. *Digital Signal Processing*, **129**, 103683. <https://doi.org/10.1016/j.dsp.2022.103683>
- [9] Fracastoro, G., Thanou, D. and Frossard, P. (2016) Graph Transform Learning for Image Compression. 2016 *Picture Coding Symposium (PCS)*, Nuremberg, 4-7 December 2016, 1-5. <https://doi.org/10.1109/pcs.2016.7906368>
- [10] Shao, Y., Zhang, Q., Li, G., Li, Z. and Li, L. (2018) Hybrid Point Cloud Attribute Compression Using Slice-Based Layered Structure and Block-Based Intra Prediction. *Proceedings of the 26th ACM International Conference on Multimedia*, Seoul, 22-26 October 2018, 1199-1207. <https://doi.org/10.1145/3240508.3240696>
- [11] Gu, S., Hou, J., Zeng, H., Yuan, H. and Ma, K. (2020) 3D Point Cloud Attribute Compression Using Geometry-Guided Sparse Representation. *IEEE Transactions on Image Processing*, **29**, 796-808. <https://doi.org/10.1109/tip.2019.2936738>
- [12] Grady, L.J. and Polimeni, J.R. (2010) Discrete Calculus: Applied Analysis on Graphs for Computational Science. Springer, London. <https://doi.org/10.1007/978-1-84996-290-2>
- [13] Song, F., Li, G., Gao, W. and Li, T.H. (2022) Rate-Distortion Optimized Graph for Point Cloud Attribute Coding.

-
- IEEE Signal Processing Letters*, **29**, 922-926. <https://doi.org/10.1109/lsp.2022.3161868>
- [14] Hammond, D.K., Vandergheynst, P. and Gribonval, R. (2011) Wavelets on Graphs via Spectral Graph Theory. *Applied and Computational Harmonic Analysis*, **30**, 129-150. <https://doi.org/10.1016/j.acha.2010.04.005>
- [15] Li, L., Zhao, Y. and Wang, S. (2023) Astronomical Image Coding Based on Graph Fourier Transform. *Proceedings of the 12th International Conference on Image and Graphics*, Nanjing, 22-24 September 2023, 311-322. https://doi.org/10.1007/978-3-031-46311-2_26
- [16] Song, F., Li, G., Yang, X., Gao, W. and Li, T.H. (2022) Fine-Grained Correlation Representation for Graph-Based Point Cloud Attribute Compression. 2022 *IEEE International Conference on Multimedia and Expo (ICME)*, Taipei, 18-22 July 2022, 1-6. <https://doi.org/10.1109/icme52920.2022.9859998>
- [17] Durmus, D. (2020) Spatial Frequency and the Performance of Image-Based Visual Complexity Metrics. *IEEE Access*, **8**, 100111-100119. <https://doi.org/10.1109/access.2020.2998292>
- [18] Zhou, B., Xu, S. and Yang, X.-X. (2015) Computing the Color Complexity of Images. 2015 *12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, Zhangjiajie, 15-17 August 2015, 1898-1902. <https://doi.org/10.1109/fskd.2015.7382237>
- [19] Bjontegaard, G. (2001) Calculation of Average PSNR Differences between RD-Curves. *Proceedings of the ITU-T Video Coding Experts Group*, 2-4.