

图像压缩算法消除数据冗余问题分析

肖卫华, 毕笃彦, 王禹

(空军工程大学 工程学院, 陕西 西安 710038)

摘要:对当前几种典型图像变换及编码算法在消除图像冗余的效果方面进行了比较分析。通过对原始图像变换和拼贴,构造了重复、对称、旋转、插值4种形式的冗余图像,然后用几种典型的变换算法和压缩算法对图像进行处理,实验结果显示当前几种图像变换及编码算法均无法同时消除本文提到的4种形式的冗余。本文的结果为进一步改进图像压缩算法提供了一个有益的参考。

关键词:图像压缩;图像冗余;小波;边缘波;分形图像压缩

中图分类号: TP391 **文献标识码:** A **文章编号:** 1009-3516(2007)06-0064-04

主流的图像压缩算法通常分为3个部分:变换、量化和熵编码。对于图像变换,除了DCT变换和小波变换外,还有SVD^[1](奇异值分解),KL变换等非主流变换方法,近几年来,很多学者提出了超越小波的图像变换方法^[2],如脊波(Ridgelet)^[3]、曲波(Curvelet)^[4]、边缘波(Contourlet)^[5]等,变换的目的就是为了消除图像数据之间的相关性。

量化也从最初的标量均匀量化发展到非均匀量化,矢量量化及针对小波系数的零树量化,基于人眼视觉特性的量化等方法。如JPEG2000采用的EBCOT算法,其量化方法之一就是使用带中央“死区”的均匀量化器。而EZW算法^[6]和SPIHT算法^[7]是零树量化的经典之作,把扫描和量化有机地结合起来,利用了不同尺度下小波系数的相似性来进行量化编码,其是集成了算术编码算法的SPIHT算法在编码效率不亚于JPEG2000,另外不少学者也提出了小波域矢量量化编码的算法,以及针对人眼视觉特性的非均匀量化算法,相比于标量量化在编码性能上也有一定的提高,但缺点是计算量比较大。

熵编码也经历了从Huffman编码到算术编码、自适应算术编码及基于上下文的自适应算术编码等,熵编码可对量化后得数据进一步的压缩。图像压缩算法除主流的基于变换域编码的算法外,还有基于不动点理论的分形图像压缩方法^[8]等等。这些算法都极大地推动了图像压缩技术的发展。但很多图像不仅仅存在相邻像素间的相关性,还存在其它形式的数据冗余。如在自然界中,很多物体如人脸、汽车、飞机等的图像都有一定的对称性,再如有关建筑物的图像,存在高度的重复性。因此,本文构造了具有重复、对称、旋转、插值这4种形式冗余的图像,并对比分析了当前图像变换算法和压缩算法在消除图像冗余方面的性能,以寻找进一步提高算法压缩性能的途径。

1 实验及分析

1.1 实验条件

为了评价分析不同图像压缩算法在消除图像冗余方面的效果,本文构造了具有重复、对称、旋转、插值这4种形式的冗余信息的图像,如图1、图2、图3、图4所示,并分别命名为Lena_Rep, Lena_Sym, lena_Ro和Lena_I,尺寸为512×512。Lena原始图像尺寸为512×512,并命名为Lena_O,将Lena_O下采样后得到的图像

收稿日期:2007-01-23

基金项目:国家“863”计划项目(2006AA701121)

作者简介:肖卫华(1977-),男,湖北随州人,博士生,主要从事图像压缩,视频压缩,VLSI设计研究;

毕笃彦(1962-),男,陕西扶风人,教授,博士生导师,主要从事数据压缩,图像处理,模式识别等研究。

命名为 Lena_D,其尺寸为 256 × 256。4 副冗余图像均由 Lena_D 通过不同的方式拼贴构造而来。



图 1 重复冗余图像

图 2 对称冗余图像

图 3 插值冗余图像

图 4 旋转冗余

显然,如果用 kolmogorov 复杂度来度量,以上 4 种图像的复杂度相对于 Lena_D 只是略微有所增加,因此,如果图像压缩算法能够消除所有的冗余,在保证相同的恢复质量的条件下,这 4 种冗余图像压缩后的数据量应和 Lena_D 基本相同。在下面的分析中,首先比较几种图像变换对图像非线性逼近的能力,即图像经过变换后,保留相同个数的最显著系数,比较重建图像的质量。然后再比较几种典型的图像编码算法的压缩效果,以比较整个编码过程对数据冗余的消除效果。

1.2 图像变换及结果分析

1) DCT 变换。表 1 所示为 DCT 变换的实验结果。其中 N 为保留的变换系数个数即保留 N 个绝对值最大的系数,将其系数量化为 D ,然后再进行图像重建。在 N 相同的情况下,Lena_Rep 和 Lena_Ro 相比于 Lena_D 有明显的恶化,其它 3 幅图片都有和 Lena_D 大致相同的重建质量。可见 DCT 变换仅能够较好地消除镜像对称和插值形式的冗余。

表 1 使用 DCT 进行非线性估计的结果

dB

N	Lena_D	Lena_O	Lena_Rep	Lena_Sym	Lena_I	Lena_Ro
2 000	25.94	25.94	24.10	25.94	25.73	23.45
5 000	28.40	28.32	26.25	28.40	27.79	25.77
8 000	29.98	29.77	27.50	29.98	28.94	27.06

2) Wavelet 变换。表 2 所示为在 N 取不同值条件下 Wavelet 变换的重建结果。在这里采用 CDF9/7 提升小波对图像进行 5 级分解,边界扩展采用对称扩展。在 N 相同的情况下,Lena_Rep, Lena_Sym, Lena_Ro 的重建质量明显低于 Lena_D,而 Lena_I 的重建质量和 Lena_D 非常接近,在 $N=8\ 000$ 时 Lena_Rep 和 Lena_Sym 的重建质量只相当于 $N=2\ 000$ 时 Lena_D 的重建质量。由此可见,Wavelet 变换仅能消除插值形式的冗余,但对于重复、对称、旋转形式的冗余均不能有效消除。

表 2 使用 Wavelet 进行非线性估计的结果

dB

N	Lena_D	Lena_O	Lena_Rep	Lena_Sym	Lena_I	Lena_Ro
2 000	27.94	27.63	23.24	23.56	27.18	24.98
5 000	32.14	31.08	26.01	26.27	29.91	28.26
8 000	34.86	33.06	27.72	27.93	31.50	30.31

3) SVD 分解。本实验采用 SVD 算法对原图像进行分解,然后选取 N 个奇异值和相应的奇异矢量进行图像重建,实验结果见表 3。由此可见,当 N 相同时,Lena_Rep Lena_Sym Lena_I 有完全相同的重建质量,只有 Lena_Ro 有明显的下降。考虑到冗余图像经过奇异值分解后奇异矢量的维数变为 Lena_D 的 1 倍,相当于系数的个数是 Lena_D 的 1 倍,因此,SVD 分解对图像冗余只有部分的消除能力,并不能完全消除。

表 3 使用 SVD 进行非线性估计的结果

dB

N	Lena_D	Lena_O	Lena_Rep	Lena_Sym	Lena_I	Lena_Ro
30	27.255 795	27.232 436	27.255 795	27.255 795	27.255 795	24.315 671
60	31.707 194	31.429 360	31.707 194	31.707 194	31.707 194	28.194 145
100	36.921 872	35.665 439	36.921 872	36.921 872	36.921 872	31.688 040

4) Contourlet 变换。本实验采用有冗余的 Contourlet 变换,实现步骤如下:先对图像进行 4 级 LP 分解,然后对每一级图像进行方向滤波,其中第一级 16 个方向,第二级 8 个方向,第三级 4 个方向,第四级进行小波分解。然后保留 N 个绝对值最大的系数进行图像重建,重建结果见表 4。在 N 相同的情况下,Lena_Rep Lena_Sym Lena_Ro 重建质量明显低于 Lena_D 的重建质量,而 Lena_I 的重建质量也略低于 Lena_D 的重建质

量, $N = 8\ 000$ 时, Lena_Rep, Lena_Sym, Lena_Ro 的恢复质量只相当于 $N = 2\ 000$ 时 Lena_D 的图像质量。因此, Contourlet 变换只是对于插值形式的冗余有一定的消除效果,但不能消除重复,对称和旋转形式的冗余。

表4 使用 Contourlet 进行非线性估计的结果

N	Lena_D	Lena_O	Lena_Rep	Lena_Sym	Lena_I	Lena_Ro
2 000	25.473 504	24.758 730	21.780 474	22.273 582	24.524 748	21.606 461
4 000	27.582 635	26.816 566	23.639 668	24.114 524	26.293 634	23.453 147
8 000	29.951 967	28.759 076	25.478 197	25.985 614	27.980 427	25.418 437

2 图像压缩结果及分析

测试了几种有损压缩算法,限于篇幅,本文只列出几种原理差别很大的几种代表性的算法, JPEG 作为将图像分块后逐块压缩的一个代表,选用 SPIHT 算法作为小波图像压缩的一个代表。同时,分形图像压缩^[8]作为一种风格完全不同的编码方式,这里也比较了其压缩结果。

1) JPEG 编码。不同质量级别的 JPEG 图像压缩所需的字节数见表 5,为了获得相同的图像质量, Lena_Rep 和 Lena_Sym 所需的字节数大致是 Lena_D 的 4 倍, Lena_I 和 Lena_Ro 所需的字节数大致是 Lena_D 的 3 倍。由此可见, JPEG 压缩算法不能有效去除以上 4 种形式的数据冗余。

表5 JPEG 压缩质量

质量	Lena_D	Lena_O	Lena_Rep	Lena_Sym	Lena_I	Lena_Ro
0	6 631	17 842	24 660	24 579	18 413	18 356
40	9 104	25 193	34 348	34 283	26 844	24 762
100	44 573	154 909	175 416	175 406	138 920	108 802

2) SPIHT 编码。采用 CDF9/7 小波 5 级分解后再进行 SPIHT 编码和算术编码的结果如表 6,比较了不同 bpp 下各图像的重建质量,其中 bpp 表示比特/像素。结果显示,在相同 bpp 的情况下, Lena_Rep, Lena_Sym 与 Lena_D 的重建质量大致相同, Lena_Ro 由于存在大部分的空白区域,导致重建质量略高一些,只有 Lena_I 在 bpp 为 0.05 时的重建质量接近于 Lena_D 在 bpp 为 0.2 时的质量,说明了目前基于小波的压缩算法只能比较有效的去除插值形式的冗余,而不能去除重复、对称及旋转形式的冗余。

表6 SPIHT 压缩结果 bpp - PSNR

bpp	Lena_D	Lena_O	Lena_Rep	Lena_Sym	Lena_I	Lena_Ro
0.05	23.33	27.55	23.73	23.94	27.13	25.36
0.2	27.74	33.17	27.99	28.15	31.58	30.66
0.5	32.07	37.25	32.34	32.49	35.57	36.25

3) 分形图像编码。选用不同参数进行分形图像压缩后得到的 PSNR 和文件尺寸见表 7,可以看出, Lena_D 压缩到 16 k 时获得的 PSNR,高于 Lena_Rep, Lena_Sym 压缩到 54 kB 时的 PSNR,接近于 Lena_I 压缩到 31 kB 时的 PSNR,略高于 Lena_Ro 压缩到 19 kB 时的 PSNR,说明分形图像编码算法对于旋转、插值形式的冗余有一定的消除作用,但却不能有效消除重复、对称两种形式的冗余。

表7 分形压缩结果

参数	Lena_D	Lena_O	Lena_Rep	Lena_Sym	Lena_I	Lena_Ro	
$T = 2$	PSNR/dB	35.387 646	35.519 530	30.064 229	30.204 253	33.222 408	32.511 833
	size/kB	45	51	54	54	49	31
$T = 4$	PSNR/dB	34.305 512	34.870 738	29.821 091	30.077 564	32.738 811	32.387 896
	size/kB	28	31	41	41	31k	25
$T = 8$	PSNR/dB	32.161 532	32.707 226	29.226 743	29.455 035	31.161 381	31.838 559
	size/kB	16	18	30	30	19	19

4) 无损图像编码。选用不同无损图像编码算法后得到的文件尺寸见表 8,可以看出,在 JPEG2000 无损压缩算法下,只有 Lena_Ro 的尺寸大致为 Lena_D 的两倍,其它几种冗余图像的尺寸都接近于 Lena_D 的 4 倍。在 JPEG 无损压缩算法下,相对来说就只有 Lena_I 得到的结果能稍微好些,尺寸为 119 kB。WinRar 压缩结果仅仅体现了对插值冗余有较好的效果。但就总体而言,这几种无损压缩算法均不能很好的消除本文列举的 4 种形式的冗余。

表8 无损压缩算法结果比较

编码器	Lena_D	Lena_O	Lena_Rep	Lena_Sym	Lena_I	Lena_Ro
JPEG2000	41	139	158	158	133	96
JPEG - LS	47	166	185	185	119	233
WinRAR	48	165	176	186	73	117

3 结论

通过以上的实验结果可以发现,在图像变换算法中,DCT对于对称形式的数据冗余消除效果较好,小波对于插值形式的数据冗余效果较好,而SVD对对称、插值、重复三种冗余都有相同的冗余消除效果。在图像编码算法中,只有基于小波的算法能较有效地消除插值形式的冗余。但总的来说,当前的几种图像压缩算法均不能同时消除图像中存在的重复、对称、旋转形式的数据冗余,即没有充分考虑自然图像的冗余,若能够将这些形式的数据冗余考虑进去,将会大大提高压缩性能。对于对称、重复、旋转三种形式的冗余,帧内预测是一个解决途径,但如何在整幅图像范围内寻找匹配区域将是一个难题,而利用小波分解的多尺度特点,在小波域由粗略到精细的去寻找匹配区域,或许是一种解决途径。

参考文献:

- [1] Waldemar P. Image compression Using Singular Value Decomposition[D]. Norway:Norwegian University of Science and Technology, 1997.
- [2] 焦李成,谭山. 图像的多尺度几何分析:回顾和展望[J]. 电子学报, 2003,31(12A):1975-1981.
- [3] Candes E J. Monoscale Ridgelets for the Representation of Images with Edges[R]. USA:Department of Statistics, Stanford University, 1999.
- [4] Candes J, Donoho D L. Curvelets[R]. USA:Department of Statistics, Stanford University, 1999.
- [5] Do M N, Vetterli M. Contourlets[A]. Stoeckler J, Welland G V. Beyond Wavelets[C]. [S. L.]: Academic Press, 2002.
- [6] Shapiro J M. Embedded Image Coding Using Zerotrees of Wavelet Coefficients[J]. IEEE Trans SP, 1993, 41(12):3445-3462.
- [7] Said A, Pearlman W. A new Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees[J]. IEEE Trans on Circuit and System for Video Technology, 1996, 6(3):243-250.
- [8] Fischer Y. Fractal Image Compression with Quadrees[A]. Fischer Y. Fractal Image Compression Theory and Application[C]. New York:Springer Verlay Press, 1995:55-77.

(编辑:姚树峰)

Analysis of the Redundancy Elimination Effect for Image Compression Algorithms

XIAO Wei-hua, BI Du-yan, WANG Yu

(The Engineering Institute, Air Force Engineering University, Xi'an 710038, China)

Abstract: Several image transforming and coding algorithms in eliminating the redundancy of image are compared. Four images with high redundancies which are referred to as repetition, symmetry, rotation and interpolation are generated by transforming and tiling the original image, then processed by using transforming and coding algorithm. The result shows that at present no algorithm can be used to eliminate the redundancies referred to as repetition, symmetry, rotation and interpolation simultaneously. This paper implies a direction to enhance the performance of image coding.

Key words: image compression; image redundancy; wavelet; contour-let; fractal image compression