

文章编号:1000-582X(2005)12-0069-04

基于不完全扫描法的彩色图像压缩方法*

朱冰莲,王文涛,吕 信,张文明

(重庆大学 通信工程学院,重庆 400030)

摘 要:针对彩色图像的特性,对零树算法在具体算法中的应用进行了改进,得出一种新的算法——不完全扫描法,并将该算法应用到彩色图像压缩中。首先将彩色图像从 RGB 空间变换到亮度、色度 YUV 空间进行压缩编/解码。通过理论分析和仿真实验表明:在保证恢复图像质量的前提下,提高了压缩比和编/解码速度。

关键词:零树算法;不完全扫描法;彩色图像
中图分类号:TN919.81

文献标识码:A

数字图像通常要求巨大的比特数,这给图像的传输和存储带来了很大的困难,因此图像压缩技术应运而生。为了在保证图像质量的前提下提高压缩比,人们一直努力探索更有效的压缩方法。对于灰度图像的压缩方法已经进行了深入的研究,对于彩色图像压缩方法的研究则不多见。虽然把灰度图像的编码方法直接用于彩色图像编码是可行的,但不一定是最好的^[1]。彩色图像有其本身所固有的特性,红绿蓝(RGB)三基色图像有一定的相关性和相似性,因此如何利用这种相关性来进一步提高压缩比是彩色图像压缩研究的重点。

文中首先把(RGB)彩色图像转换到 YUV 空间表示,在 YUV 空间进行小波变换,然后用改进的零树编码方法——不完全扫描法对小波系数进行编码。解码时先得到 YUV 空间的估计值,再把 YUV 空间的估计值转换到 RGB 空间,得到 RGB 空间的估计图像。实验仿真证明,该方法提高了编/解码速度,在相同压缩比下,提高了图像复现效果。

1 彩色图像从 RGB 空间到 YUV 空间的变换

彩色图像由三基色图像组成。它们是红色 R、绿色 G 和蓝色 B。RGB 三基色空间对彩色图像进行编码压缩,虽然也有很好的效果,但它并没有利用彩色图像固有的特性,所以应寻求比这种方法更好的编码方法,以更加有效地对彩色图像进行压缩。像数据从 RGB 空间到 YUV 空间的转换关系为^[2]:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

从 YUV 空间到 RGB 空间的转换关系为:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & -0.001 & 1.402 \\ 1.000 & -0.344 & -0.714 \\ 1.000 & 1.772 & 0.001 \end{bmatrix} \times \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

Y 表示图像中像素点的亮度分量,U、V 分别表示像素点的色度分量,这样表示对图像压缩是非常有利的。首先,图像从 RGB 变换到 YUV 后,能量主要集中于亮度分量 Y。Y 分量幅值的分布范围是[0,255],而 U、V 空间分量的能量较小,变化范围是[0,64],远小于亮度分量的变化范围,因此 U、V 分量的小波变换系数幅值也较小,用零树编码方法进行编码时所需的码较少。由此可知当达到同等精度的估计图像时,在 YUV 空间编码可以达到比在 RGB 空间编码更高的压缩比,而且对 U、V 分量编码时的计算量也大大减少。因此,虽然在 YUV 空间进行编码增加了从 RGB 空间到 YUV 空间转换的计算量,但编码时的计算量减小,所以总的计算量并没有增加。另一方面,人眼对像素点的亮度分辨力较强,而对像素点的色度分辨力较弱。因此在 YUV 空间对图像进行编码时,可以提高对 Y 分量的编码精度,而降低对 U、V 分量的编码精度,这样在不影响视觉效果的前提下可以进一步提高图像的压缩比。由以上分析可见,在 YUV 空间利用小波变换和零树编码方法进行彩色图像的压缩应该比在 RGB 三基色空间进行压缩的效率更高,图像复现效果更好。

* 收稿日期:2005-07-29

作者简介:朱冰莲(1959-),女,四川富顺人,重庆大学副教授,主要从事信号与信息处理研究。

2 零树编码算法

图像压缩编码后,用于存储图像信息的比特流通常被划分为存储非零系数的比特流和存储非零系数位置的比特流 2 部分. 一般情况下,用来表示非零系数位置的开销要远大于用来表示非零系数的开销,所以,提高非零系数位置的存储效率必将提高图像的压缩效果. 文献[3]中由 ShaPiro 提出的 EZW 编码算法正是一种描述图像经小波变换后非零系数位置的有效方法,其理论依据是利用了相同方向不同尺度子带之间具有的相关性. 下面就以图像 3 级小波分解为例,简要说明零树结构及 EZW 编码算法工作原理. 其中: L 表示低频, H 表示高频,下标 1,2,3 表示一级、二级、三级分解,称粗尺度上的小波系数为双亲,称邻近粗尺度的较细尺度相应位置上的所有小波系数为子女,称粗尺度之后的较细尺度相应位置上的所有小波系数为后代.

2.1 零树结构

如图 1 所示,一幅经过小波变换的图像按其频带从低到高形成一个树状结构,树根是最低频子带的结点,它有 3 个子女,分别位于次低频子带的相应位置;其余子带(最高频子带除外)的结点都有 4 个子女位于高一级的子带的相应位置(由于高频子带分辨率增加,所以一个低频子带结点对应 4 个高频子带结点),这样图 1 所示的 3 级小波分解便形成了深度为 4 的树. 当然,最高频子带和 LL 子带中的小波系数因无子女而不能构造树结构.

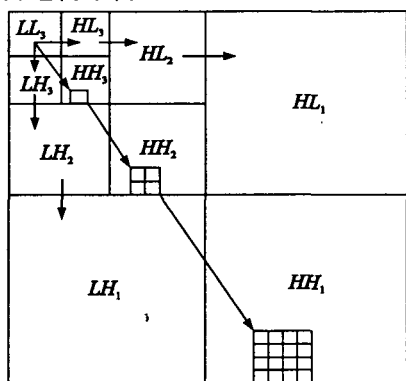


图 1 零树结构示意图

一个小波系数 x , 对于一个给定的阈值 T , 如果 $|x| < T$, 则称小波系数 x 是不重要的, 否则称小波系数 x 是重要的. 如果一个小波系数在一个粗的尺度上关于给定的阈值 T 是不重要的, 之后在较细尺度上同样空间位置中的所有小波系数也关于阈值 T 也是不重要的, 则称小波系数形成了一个零树结构. 阈值 T 将小波系数分成了 4 种情况: 正重要系数 P 表示该系数绝对值大于等于门限且为正; 负重要系数 N , 表示该系数绝对值大于等于门限且为负; 孤立零 I 表示该系数不重要但它的后代中有重要系数; 零树根 Z 表示该系

数不重要且它的所有后代不重要.

2.2 零树编码 EZW 算法工作原理

零树编码算法是以零树结构为理论基础, 通过逐次逼近量化完成嵌入编码的. 而所谓逐次逼近量化就是要通过逐次使用阈值序列 $T[0], T[1], T[2], \dots, T[n-1]$ 来决定重要系数. 其中 $T[n] = (T[n-1])/2$, 且初始阈值 $T[0]$ 的选择要满足条件: 对于所有小波系数 $x[i]$, 有 $|x[i]|_{\max} / 2 < T[0] < |x[i]|_{\max}$, $|x[i]|_{\max}$ 是绝对值最大的小波系数. 逐次逼近量化过程可以描述为:

```
T = T0;
do{
  dominant_Pass(image);
  refinement_Pass(image);
  symbol_encode();
  T = T/2;
}while(T > T[n-1]);
```

整个逐次逼近量化过程包括主扫描 (dominant_pass)、精细扫描 (refinement_pass) 和符号编码 (symbol_code) 3 个子过程. 其中主扫描过程完成以下 3 项任务:

- 1) 搜索扫描整个图像并为每个小波系数分配符号, 以创建用于指示小波系数相对于当前阈值是否重要的位平面, 该位平面被称做重要图.
- 2) 将所有重要小波系数抽取出来(不带符号) 并安放在被称为附属表的一维数组中.
- 3) 在图像中重要小波系数所处位置填入 0.

精细扫描过程总的任务是输出附属表中所有小波系数的下一个最重要位.

符号编码过程主要负责利用自适应算术编码算法完成对主扫描和精细扫描所产生符号的编码.

在整个逐次逼近量化过程中, 不断等分当前阈值, 重复主扫描、精细扫描和符号编码 3 个过程, 直到满足目标比特率需要. 利用该零树结构, 只用很少的编码位就可以对大量的不重要系数进行编码, 而留下更多的编码位来编码那些重要的系数, 这也是零树编码之所以高效率的原因.

3 不完全扫描法——改进的零树编码算法

然而, 仔细分析 EZW 算法工作原理, 不难看出零树编码算法运算复杂度比较高, 当用于彩色图像压缩时这一缺陷尤为突出. 彩色图像由 R、G、B 三基色图像组成. 在 RGB 空间对彩色图像进行编码压缩, 要对 R、G、B 3 个基色图像分别进行逐次逼近量化, 重复主扫描、精细扫描和符号编码 3 个子过程. 这势必要花费大量的时间, 而且随着压缩倍数的提高, 恢复图像边缘有较明显的模糊, 视觉效果不好.

针对 EZW 算法在彩色图像压缩中存在的以上不

足,文中结合彩色图像固有的特性和人眼的视觉特性,在图像编码中利用人眼的视觉掩盖效应,可以在主观感觉图像失真相同的条件下,允许更大的实际失真^[4].因此提出了一种改进的嵌入零树小波图像编码算法——不完全扫描法,进一步提高 EZW 算法工作效率.

鉴于在 YUV 空间中图像能量主要集中于亮度分量 Y,而 U、V 分量能量较少;同时考虑到人眼对像素点的亮度分辨力较强,而对像素点的色度分辨力较弱,因此首先将彩色图像从 RGB 空间变换到 YUV 空间,对彩色图像在 YUV 空间的小波变换系数进行编码,对 Y 分量进行主扫描、精细扫描和符号编码;而对 U、V 信号只进行主扫描和符号编码,省略了精细扫描.经过改进后,大大降低了运算复杂度,提高了运算速度,在保证图像质量的前提下,提高了压缩比,更有利于硬件实现.具体改进算法如下:

对于 Y 信号:

```
T = T[0];
do{
    dominant_Pass(image);
    refinement_Pass(image);
    symbol_encode();
    T = T/2;
    }while(T > T[n-1]);
```

对于 U、V 信号:

```
T = T[0];
do{
    dominant_Pass(image);
    symbol_encode();
    T = T/2;
    }while(T > T[n-1]);
```

其中主扫描(Dominant_pass)的过程为:

```
do{
    按指定的扫描顺序取出一个系数  $x[i]$ ;
    如果  $x[i] > T$ , 则输出 P,  $x[i] = x[i] - 3/2 T$ ;
    如果  $x[i] < -T$ , 则输出 N,  $x[i] = x[i] + 3/2 T$ ;
    如果  $|x[i]| < T$ , 且该系数后代中有重要系数, 则输出 I;
    如果  $|x[i]| < T$ , 且该系数后代中没有重要系数, 则输出 Z;
    }while(还有系数没被扫描);
```

文中在主扫描中对重要小波系数进行了移位操作,进一步减少了存储小波系数的比特数,减少了内存的开销.

精细扫描(refinement_pass)对重要系数进行细化.对于阈值 $T[i]$ 来说,重要系数的所在区间为

$\{T[i], 2T[i]\}$,若重要系数位于 $\{T[i], 3/2T[i]\}$,则用符号“0”表示,否则用符号“1”表示.这样做的目的是为了减小非零系数值的变化范围.编码时主扫描和精细扫描 2 个过程交替进行.然后对 2 个过程输出的码流进行熵编码,文中采用算术编码^[5].

解码时先将系数初始化为零,然后把熵编码的结果转换成由 P、N、Z、I 组成的码流,对于 P,则系数恢复成系数的值加上 $3/2 T$,对于 N,则系数恢复成系数的值减去 $3/2 T$.最后一个系数确定后,将阈值减半,重复上面的过程.直到阈值为 $T[n-1]$ (文中取 $T[n-1] = 1$).

4 实验结果

文中用 Lena 标准彩色测试图像作为实验图像.采用 5/3 双正交小波基进行 4 层二维整数小波变换,分别在 RGB 三基色空间用零树编码对小波系数编码,和在 YUV 空间用不完全扫描法对小波系数编码压缩,最后对码流进行算术编码.实验结果如下.(注:实验在 Windows98 操作系统、P III 555 MHz、Visual C++ 6.0 编译环境下进行.)

从表 1 可看出,当用 2 种方法所得压缩比基本相同时.文中算法进行编码的恢复图像峰值信噪比高于在 EZW 算法.

表 1 2 种算法的峰值信噪比对照 (PSNR/dB)

Bit/bpp	EZW (RGB 空间)	文中算法 (YUV 空间)
0.25	28.303 8	31.216 1
0.5	30.225 0	34.272 6
1.0	32.757 4	35.070 4

从表 2 可看出,文中算法在编码速度上比 EZW 算法有所提高.压缩比越大,优势越明显.

表 2 2 种算法的编码时间对照 s

Bit/bpp	EZW (RGB 空间)	文中算法 (YUV 空间)
0.25	4.480 0	3.720 0
0.5	5.527 5	5.218 0
1.0	6.862 2	6.664 0

从表 3 可看出,文中算法在解码速度上比 EZW 算法有了明显的提高.且由表 1 和表 2 可以看出,文中的解码速度较之编码速度快了很多,比原 EZW 算法有了明显的提高.

表 3 2 种算法的解码时间对照 s

Bit/bpp	EZW (RGB 空间)	文中算法 (YUV 空间)
0.25	4.274 0	1.764 0
0.5	5.280 0	2.080 0
1.0	6.573 3	2.084 0

比较图 2 和图 3 可见,EZW 算法的恢复图像边缘有较明显的模糊,视觉效果不太好;采用文中算法的恢复图像视觉效果较好,这是因为进行图像压缩时,是以损失人眼视觉不太敏感的色度信号为代价来提高压缩

比的,亮度信号的损失不太大,因此恢复图像的视觉效果和原图像比较基本上没有区别.



图2 EZW 算法 0.25 bpp(PSNR = 28.303 8)



图3 文中算法 0.25 bpp(PSNR = 31.216 1)



图4 Lena 图像原图

5 结 论

针对彩色图像的固有特性,对零树编码算法进行了改进,提出了不完全扫描法,并应用到了彩色图像中.在 YUV 空间对彩色图像进行压缩编码,以损失人眼不太敏感的色度信号为代价来提高压缩比,可以达到较高的压缩比,对视觉效果无明显影响,比原算法的压缩效率高.并且提高了编/解码速度.仿真实验也验证了这一结论.

参考文献:

- [1] 黄贤武,李家骅.基于 HVS 彩色图像的扩展零树小波编码方法的研究[J].数据采集与处理,2001,16(2):215-219.
- [2] 杨春玲,余英林.基于小波变换的彩色图像编码方法[J].华南理工大学学报(自然科学版),2000,28(9):87-92.
- [3] SHAPIRO J M. Embedded Image Coding Using Zerotrees of Wavelet Coefficients[J]. IEEE Transactions on Signal Processing, 1993, 41(12):3 445 -3 462.
- [4] 刘榴娣,刘明奇,党长民.实用数字图像处理[M].北京:北京理工大学出版社,2001.
- [5] WITTEN I, NEAL R, CLEARY J. Arithmetic Coding for Data Compression [J]. Commun. ACM, 1987, 30: 520-540.
- [6] 张益贞,刘滔.Visual C++实现 MPEG/JPEG 编解码技术[M].北京:人民邮电出版社,2002.

Color Image Compression Method Based on Incomplete Scanning Method

ZHU Bing-lian, WANG Wen-tao, LU Ji, ZHANG Wen-ming

(College of Communication Engineering, Chongqing University, Chongqing 400030, China)

Abstract: In view of color image characteristic, the paper improves the zero algorithm in the application of concrete algorithm, draws a new algorithm-incomplete scanning method, which is applied to color image compression. The authors convert color image from the RGB space into luminance-chrominance YUV space, then compress the converted image to encode/decode. Theoretical analysis and simulation experiment indicate that the compression ratio and encoding/decoding speed are increased, at the same time the quality of restored image is guaranteed.

Key words: EZW; incomplete scanning method; color image