

## [12] 发明专利说明书

专利号 ZL 200610061321.8

[51] Int. Cl.

H04N 7/26 (2006.01)

H04N 1/41 (2006.01)

[45] 授权公告日 2009 年 12 月 23 日

[11] 授权公告号 CN 100574434C

[22] 申请日 2006.6.23

[21] 申请号 200610061321.8

[73] 专利权人 深圳安凯微电子技术有限公司

地址 518057 广东省深圳市高新技术产业园南区深圳清华大学研究院 B608

[72] 发明人 倪武学 李小明

[56] 参考文献

CN1520185A 2004.8.11

CN1441555A 2003.9.10

US2005/0276501A1 2005.12.15

基于 FPGA 的 JPEG 图像压缩芯片设计.  
贺军. 电子科技大学硕士学位论文. 2003

审查员 李萍

[74] 专利代理机构 中原信达知识产权代理有限责任公司  
代理人 樊卫民

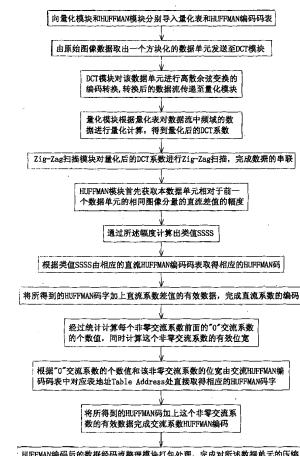
权利要求书 3 页 说明书 18 页 附图 3 页

[54] 发明名称

一种图像压缩/解压缩方法和系统

[57] 摘要

一种涉及计算的图像压缩/解压缩方法和系统，该系统包括 DCT 模块、量化模块、Zig - Zag 扫描模块、HUFFMAN 模块、码流整理模块依次相连，其特征在于：还包括存储器和主控模块，主控模块控制各工作模块的工作状态，对原始图像数据完成方块化处理或提取压缩数据块，该主控模块根据处理进程对存储器中的表项进行相应的调用，向量化模块或 HUFFMAN 模块加载信息，压缩方法包括：A) 导入量化表和 HUFFMAN 编码码表，B) 取出数据单元发送至 DCT 模块，C) DCT 模块对进行离散余弦变换，D) 量化模块进行量化计算，得到量化系数，E) Zig - Zag 扫描模块对系数进行 Zig - Zag 扫描，F) 对系数进行相应的 HUFFMAN 编码，G) HUFFMAN 编码后的数据打包处理，完成压缩，本发明实时性高且成本低。



1. 一种图像压缩/解压缩系统，包括 DCT 模块、量化模块、Zig-Zag 扫描模块、HUFFMAN 模块、码流整理模块并依次相连，其中，
  - 所述的 DCT 模块对数据进行离散余弦变换及反变换；
  - 所述的量化模块完成数据的量化或反量化计算；
  - 所述的 Zig-Zag 扫描模块完成数据的串联或解串联；
  - 所述的 HUFFMAN 模块完成数据的 HUFFMAN 编码或解码；
  - 所述的码流整理模块完成数据的打包或分包处理；其特征在于：还包括存储器和主控模块，其中，
  - 所述的存储器中保存有关表项；
  - 所述的主控模块控制 DCT 模块、量化模块、Zig-Zag 扫描模块、HUFFMAN 模块、码流整理模块的工作状态，对原始图像数据完成方块化处理发送至 DCT 模块，或提取压缩数据块至码流整理模块，该主控模块根据处理进程对存储器中的表项进行相应的调用，向量化模块或 HUFFMAN 模块加载信息；
  - 所述的存储器中的表项包括量化表和 HUFFMAN 编解码表；
  - 所述的主控模块调用所述量化表，加载至量化模块；
  - 所述的主控模块调用所述 HUFFMAN 编解码表，并按照代码长度，且对于每个长度的所有码字根据大小顺序对 HUFFMAN 编解码表进行排序，排序后的 HUFFMAN 编解码表的表项对应于按顺序的地址入口，加载至 HUFFMAN 模块，解码时同时向 HUFFMAN 模块加载解压缩过程中用到的从 HUFFMAN 编解码表中提取出来的码字特征信息；
  - 所述的码字特征信息包括每个长度 HUFFMAN 码的开始地址、开始码字、以及表示每个长度的码字是否有效的标志。

2. 根据权利要求 1 所述的图像压缩/解压缩系统，其特征在于：

所述的 HUFFMAN 编解码表包括直流 HUFFMAN 编解码表和交流 HUFFMAN 编解码表，其中，

所述的直流 HUFFMAN 编解码表包括：类值、码字长度、码字；类值为表项访问入口；

所述的交流 HUFFMAN 编解码表包括：零游长/类值、码字长度、码字；零游长/类值为表项访问入口。

3. 一种图像解压缩方法，其特征在于：所述解压缩步骤为：

- a、主控模块调用 HUFFMAN 编解码表，并按照代码长度，且对于每个长度的所有码字根据大小顺序对 HUFFMAN 编解码表进行排序，排序后的 HUFFMAN 编解码表的表项对应于按顺序的地址入口，加载至 HUFFMAN 模块，且将量化表导入量化模块；
- b、主控模块从 HUFFMAN 编解码表提取码字特征信息，且向 HUFFMAN 模块加载该信息，所述的码字特征信息包括每个长度 HUFFMAN 码的开始地址、开始码字、以及表示每个长度的码字是否有效的标志；
- c、主控模块提取压缩数据包至码流整理模块；
- d、码流整理模块对压缩数据包进行分包处理，将分出的数据码流发送至 HUFFMAN 模块；
- e、HUFFMAN 模块根据码字特征信息确定 HUFFMAN 码的长度以及在 HUFFMAN 编解码表中的位置；
- f、以所取得的长度、位置信息为检索入口，从 HUFFMAN 编解码表中取出相应的类值或零游长/类值；
- g、通过类值得到数据码流中系数的有效位宽，从码流中取出相应个位的码流，作为系数的有效数据；如果是交流系数的解码，零游长表示该系数前面所具有“0”系数的个数，这样就完成了

- 
- 一段码流的 huffman 解码；后续码流的解码过程相同；
- h、Zig-Zag 扫描模块对解码后的数据流进行反 Zig-Zag 扫描，完成数据流的解串联；
- i、量化模块根据量化表对数据流进行反量化操作，产生频域数据；
- j、DCT 模块将数据流的频域数据转化为重构图像的时域数据，完成解压缩。
4. 根据权利要求 3 所述的图像解压缩方法，其特征在于：所述的步骤 e 包括如下步骤：
- e1、HUFFMAN 模块读取一组数据码流，根据码字特征信息，通过与每个长度的开始码字进行比较，确定该数据码流中 HUFFMAN 码的码字长度；
- e2、通过码字长度从数据码流中取出 HUFFMAN 码字，把取出的码字与相应长度的开始码字相比较，计算出在 HUFFMAN 编解码表中该码字相对于开始码字的相对位置；
- e3、通过开始码字在 HUFFMAN 编解码表中的位置和计算所得到的相对位置算出该码字在 HUFFMAN 码表中的绝对位置。

## 一种图像压缩/解压缩方法和系统

### 技术领域

本发明涉及计算，尤其涉及一种图像压缩/解压缩方法和系统。

### 背景技术

JPEG (Joint Photographic Experts Group) 是 90 年代兴起的采用数字方法对静止图片进行压缩的国际标准，压缩比可高达十倍左右，随着数字设备拍照功能在手机和手持设备上的普及，符合 JPEG 国际静止图片压缩-解压缩标准的装置或电路已成为这些设备必须的组成部分。

在进行压缩时，需要将原始的图像数据从 RGB Color domain 转换到 YUV 或 YCbCr domain。整个 JPEG 压缩标准定义在 YUV domain 而非 RGB domain 里。定义在 YUV domain 里是利用人类的视觉特性的缘故，Y 成为亮度 (Luminance)，而 U、V 则成为色度 (Chrominance)，因为人类眼球中感光的细胞与感色的细胞在数目上有相当程度的差别（人眼对亮度的敏感度要远大于对色度的敏感度），因此感受力上的差距，使我们能对 U、V 信号做再次采样，可以更进一步地提高压缩比。

JPEG 编码压缩主要流程一般如下：

#### I、 图片方块化：

在进行压缩之前，将原始的图像数据切割成 8X8 的方块，切成 8X8 方块的原因是对于绝大部分图片，8X8 方块内的 64 个像素表示

的信息是密切相关的，这样通过后续的过程处理过的数据具有更高的压缩度。

## II、 离散余弦变换：

离散余弦变换 (Discrete Cosine Transform) 简称 DCT，是一种转换编码的技术，它能将时域中的数据转换到频域，转换到频域后的数据有一种特性，就是能量（表达原始图像的信息）会集中在 8x8 的方块左上角的位置，越往右下角所代表的就越是高频的数据。

如果两个 k 位像素 A 和 B 密切相关，那末我们可以用 (k+1) 位表示它们的和 (A+B)，用很小的位数表示它们的差 (A-B)。这样总的表示位减少了。

这一概念的推广就是 DCT 变换：

$$S_{uv} = \frac{1}{4} C_u C_v \sum_x \sum_y s_{xy} \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right]$$

这里，  $C_u, C_v = \begin{cases} 1/\sqrt{2} & u = 0, v = 0 \\ 1 & otherwise \end{cases}$

密切相关 8 X 8 图像方块对应的 64 个变换参数

$\{S_{uv}, \quad u=0, 1, 2, \dots, 7; \quad v=0, 1, 2, \dots, 7\}$  表示该方块数据的特性。

如果该方块内的像素比较平稳，那末  $S_{uv}$  的低频分量较重，而高频分量较轻，这样总的表示位减少了。在一般的图像数据里，低频的数据占据着较大的能量比例，而高频数据所占据的能量比例则较低，加上人类视觉系统的低通滤波特性，DCT 转换近乎完美地将这些高低频的信息依次分类在 8x8 的方块里。

DCT 变换是完全可逆变换，其反变换为：

$$s_{xy} = \frac{1}{4} \sum_u \sum_v C_u C_v S_{uv} \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right]$$

### III、量化：

做完 DCT 转换后，将需要将频域的数据经量化计算，量化计算是经特定的量化表做除法运算。量化表的设计是基于人类视觉系统的特性的，因此在低频的值小，而在高频的值大，这使得高频数据在经过除法量化后变为零得几率大增。

JPEG 提供了一套标准的量化矩阵  $\{Q_{uv}, u=0, 1, 2, \dots, 7; v=0, 1, 2, \dots, 7\}$ 。

$Q_{uv}$  矩阵也可以由用户根据本图像特点进行优化，在图像质量和产生的码流长度之间进行平衡。

必须注意的是，因为量化过程采取了除法取整的过程，如下式所示：

$$Q_{uv} \times \text{round}\left[\frac{S_{uv}}{Q_{uv}}\right] \neq S_{uv}$$

$S_{uv}$  为做完 DCT 后  $8 \times 8$  方块内的值； $Q_{uv}$  为  $8 \times 8$  的量化表。

这样就会产生失真，这也是 JPEG 唯一会产生失真的步骤。

虽然量化会产生失真，却也是能提高压缩比的关键所在。

JPEG 提供的标准量化矩阵如下，一般情况 U, V 分量精度可以差一些（这在一定程度上也是因为人眼对色度的敏感度要差一些）：

例如，适用于 Y 分量的  $\{Q_{uv}\}$  矩阵如下表 1 所示：

|    |    |    |    |     |     |     |     |
|----|----|----|----|-----|-----|-----|-----|
| 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1

又如，适用于 U, V 分量的 {Quv} 矩阵如下表 1 所示：

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

表 2

#### IV、Zig-Zag 扫描：

经过量化后的数据再通过特定的数据串联法则串联在一起，称为 Zig-Zag Scan，以求最终能以最少的位数来表示原始数据信息。

例如，Zig-Zag 扫描的顺序如下表 3 所示：

|    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| 0  | 1  | 5  | 6  | 14 | 15 | 27 | 28 |
| 2  | 4  | 7  | 13 | 16 | 26 | 29 | 42 |
| 3  | 8  | 12 | 17 | 25 | 30 | 41 | 43 |
| 9  | 11 | 18 | 24 | 31 | 40 | 44 | 53 |
| 10 | 19 | 23 | 32 | 39 | 45 | 52 | 54 |
| 20 | 22 | 33 | 38 | 46 | 51 | 55 | 60 |
| 21 | 34 | 37 | 47 | 50 | 56 | 59 | 61 |
| 35 | 36 | 48 | 49 | 57 | 58 | 62 | 63 |

表 3

经过 Zig-Zag 扫描后的数据有个特点就是，低频分量在前，高频分量在后，换句话说就是非零系数基本集中在前面，零系数基

---

本集中在后面，这样更有利于提高后面的编码压缩率。

## V、 HUFFMAN 编码（熵编码）：

经过 Zig-Zag 扫描串联后的数据进入一种特殊的编码，称为哈夫曼编码（HUFFMAN）或者熵编码（Entropy Coding），它是可变字长编码（VLC）的一种。它完全依据字符出现的概率来构造码表，使得编码后的平均字长最短。

频域上的数据分为两种，一种在左上角 00 位置，称为直流 DC 系数，其余的位置皆为交流 AC 系数。

HUFFMAN 编码并不对 DC 或 AC 直接编码。因为对 DC 系数而言，每个方块都存在着类似的 DC 值，这些类似的 DC 值（一般称之为冗余 Redundancy），如果都拿来重复编码，将会降低编码效率，因此实际上是将目前的 DC 值减去前一个 8x8 方块的 DC 值，以此差值做编码。

而对于 AC 而言，由于高低频数据分布的特性使得串联数据中有许多零项，这些零项也会在编码时被充分利用，即，交流系数的编码是对一个非零系数和它前面所有的零仅做一次编码，而不是对每个交流系数都做一次编码，这样可以大大提高压缩效率。对哈夫曼压缩效率贡献最大的也就是这种对 AC 系数的编码方式。

以上所述为数据压缩的主要流程，解压缩则为上述流程的逆过程。

在现有技术中，实现上述控制流程主要采用以下两种方式：

- a) 纯软件方式：对于 200MHz 的嵌入式微处理器每分钟才能处理一幅 4 百万像素的图像，不能满足实时性的要求。
- b) 纯硬件方式：现有的纯硬件方式缺乏嵌入式微处理器软件的灵活

性，过分强调每个周期进行一个 HUFFMAN 解码，造成硬件开销很大，成本过高。

### 发明内容

本发明的目的在于提供一种实时性高且成本低的图像压缩/解压缩方法和系统，以克服现有技术中纯软件方式实时性差，纯硬件方式成本高的问题。

本发明所采用的图像压缩/解压缩系统，包括 DCT 模块、量化模块、Zig-Zag 扫描模块、HUFFMAN 模块、码流整理模块依次相连，其中，

所述的 DCT 模块对数据进行离散余弦变换 (Discrete Cosine Transform) 及反变换；

所述的量化模块完成数据的量化或反量化计算；

所述的 Zig-Zag 扫描模块完成数据的串联或解串联；

所述的 HUFFMAN 模块完成数据的 HUFFMAN 编码或解码；

所述的码流整理模块完成数据的打包或分包处理；

其特征在于：还包括存储器和主控模块，其中，

所述的存储器中保存有关表项；

所述的主控模块控制 DCT 模块、量化模块、Zig-Zag 扫描模块、HUFFMAN 模块、码流整理模块的工作状态，对原始图像数据完成方块化处理发送至 DCT 模块，或提取压缩数据块至码流整理模块，该主控模块根据处理进程对存储器中的表项进行相应的调用，向量化模块或 HUFFMAN 模块加载信息。

所述的存储器中的表项包括量化表和 HUFFMAN 编解码表；

所述的主控模块调用所述量化表，加载至量化模块；

所述的主控模块调用所述 HUFFMAN 编解码表，并对其进行排序，加

---

载至 HUFFMAN 模块，解码时向 HUFFMAN 模块加载解压缩过程中用到的从 HUFFMAN 码表中提取出来的码字特征信息；

所述的码字特征信息包括每个长度 HUFFMAN 码的开始地址、开始码字、以及表示每个长度的码字是否有效的标志。

所述的 HUFFMAN 编解码表包括直流 HUFFMAN 编解码表和交流 HUFFMAN 编解码表，其中，

所述的直流 HUFFMAN 编解码表包括：类值 SSSS、码字长度 Code length、码字 Code word；类值 SSSS 为表项访问入口；

所述的交流 HUFFMAN 编解码表包括：零游长 RRRR/类值 SSSS、表址 Table Address、码字长度 Code length、码字 Code word；零游长 RRRR/类值 SSSS 为表项访问入口；

在解压缩中，主控模块按照码字长度 Code length，且对于每个长度的所有码字根据大小顺序对 HUFFMAN 解码码表排序，排序后的 HUFFMAN 解码码表的表项对应于按顺序的地址入口。

本发明中的图像压缩方法采用如下步骤：

- A、 主控模块向量化模块和 HUFFMAN 模块分别导入量化表和 HUFFMAN 编码码表；
- B、 主控模块由原始图像数据取出一个方块化的数据单元发送至 DCT 模块；
- C、 DCT 模块对该数据单元进行离散余弦变换的编码转换，转换后的数据流传递至量化模块；
- D、 量化模块根据量化表对数据流中频域的数据进行量化计算，得到量化后的 DCT 系数；
- E、 Zig-Zag 扫描模块对量化后的 DCT 系数进行 Zig-Zag 扫描，完成数

据的串联。

- F、 HUFFMAN 模块根据 HUFFMAN 编码码表对经过 Zig-Zag 扫描的系数进行相应的 HUFFMAN 编码。
- G、 HUFFMAN 编码后的数据经码流整理模块打包处理，完成对所述数据单元的压缩。

所述的步骤 F 包括如下步骤：

- F1、 HUFFMAN 模块首先获取本数据单元相对于前一个数据单元的相同图像分量的直流差值的幅度 magnitude；
- F2、 通过所述幅度 magnitude 计算出类值 SSSS；
- F3、 根据类值 SSSS 由相应的直流 HUFFMAN 编码码表取得相应的 HUFFMAN 码；
- F4、 将所得到的 HUFFMAN 码字加上直流系数差值的有效数据，完成直流系数的编码；
- F5、 经过统计计算每个非零交流系数前面的“0”交流系数的个数，同时计算这个非零交流系数的有效位宽；
- F6、 根据“0”交流系数的个数值和该非零交流系数的位宽由交流 HUFFMAN 编码码表中对应表地址 Table Address 处直接取得相应的 HUFFMAN 码字；
- F7、 将所得到的 HUFFMAN 码加上这个非零交流系数的有效数据完成交流系数 HUFFMAN 编码。

所述的步骤 F4 中，所述有效数据的位宽即类值 SSSS 的大小。

本发明中的图像解压缩方法采用如下步骤：

- a、 主控模块调用 HUFFMAN 解码码表，并对其进行排序，加载至 HUFFMAN

- 模块，且将量化表导入量化模块；
- b、主控模块从 HUFFMAN 表提取码字特征信息，且向 HUFFMAN 模块加载该信息，所述的码字特征信息包括每个长度 HUFFMAN 码的开始地址、开始码字、以及表示每个长度的码字是否有效的标志；
  - c、主控模块提取压缩数据包至码流整理模块；
  - d、码流整理模块对压缩数据包进行分包处理，将分出的数据码流发送至 HUFFMAN 模块；
  - e、HUFFMAN 模块根据码字特征信息确定 HUFFMAN 码的长度以及在 HUFFMAN 编解码表中的位置；
  - f、以所取得的长度、位置信息为检索入口，从 HUFFMAN 解码码表中取出相应的类值 SSSS 或零游长 RRRR/类值 SSSS；
  - g、通过类值 SSSS 得到数据码流中系数的有效位宽，从码流中取出相应个 bit 的码流，作为系数的有效数据；如果是交流系数的解码，零游长表示该系数前面所具有“0”系数的个数，这样就完成了一段码流的 huffman 解码；后续码流的解码过程相同；
  - h、Zig-Zag 扫描模块对解码后的数据流进行反 Zig-Zag 扫描，完成数据流的解串联；
  - i、量化模块根据量化表对数据流进行反量化操作，产生频域数据；
  - j、DCT 模块将数据流的频域数据转化为重构图像的时域数据，完成解压缩。

所述的步骤 a 中：所述的主控模块按照码字长度 Code length，且对于每个长度的所有码字根据大小顺序对 HUFFMAN 解码码表排序，排序后的 HUFFMAN 解码码表的表项对应于按顺序的地址入口。

所述的步骤 e 包括如下步骤：

- 
- e1、HUFFMAN 模块读取一组数据码流，根据码字特征信息，通过与每个长度的开始码字进行比较，确定该数据码流中 HUFFMAN 码的码字长度；
  - e2、通过码字长度从数据码流中取出 HUFFMAN 码字，把取出的码字与相应长度的开始码字相比较，计算出在 HUFFMAN 码表中该码字相对于开始码字的相对位置；
  - e3、通过开始码字在 HUFFMAN 表中的位置和计算所得到的相对位置算出该码字在 HUFFMAN 码表中的绝对位置。

本发明的有益效果为：在本发明中，采用主控模块控制各个功能模块（DCT 模块、量化模块、Zig-Zag 扫描模块、HUFFMAN 模块、码流整理模块、存储器等），利用主控模块对存储器中保存的表项（量化表和 HUFFMAN 编解码表）操作，使得压缩和解压缩共享一套表项，并可根据实际需要通过主控模块对表项内容进行修改、排序等，本发明的这种硬件与软控制相结合的方式，处理速度可以得到一定程度的保证，满足了处理实时性要求，而且，避免了纯硬件方式中对于操作周期的限制，可以利用主控模块即时控制硬件状态，使得硬件开销大大减少，降低成本，同时也降低了系统设计难度，因此，本发明实时性高且成本低，克服了现有技术中纯软件方式实时性差，纯硬件方式成本高的问题。

在本发明的压缩处理中，根据零游长 RRRR 和类值 SSSS 由交流 HUFFMAN 编解码表中对应表地址 Table Address 处直接取得相应的 HUFFMAN 码，避免了现有技术中通过对零游长 RRRR 和类值 SSSS 进一步译码，取得表地址 Table Address 的操作，使得本发明表项操作简便、直接性强，运行速度快，进一步提高了本发明的实用性。

在本发明的解压缩处理中，HUFFMAN 模块根据码字特征信息确定数据码流中 HUFFMAN 码的码字长度，再将取出的 HUFFMAN 码字与相应长度

的开始码字相比较，计算出在 HUFFMAN 码表中该码字相对于开始码字的相对位置，然后以开始码字在 HUFFMAN 表中的位置和计算所得到的相对位置算出该码字在 HUFFMAN 码表中的绝对位置，这种 HUFFMAN 码的确定方法操作简便，对于类值 SSSS 或零游长 RRRR/类值 SSSS 的获取，实际上通过了一种二次查表方式：

第一次：把取出的码字与相应长度的开始码字相比较。

第二次：以所取得的长度、位置信息为检索入口，从 HUFFMAN 解码码表中取出相应的类值 SSSS 或零游长 RRRR/类值 SSSS。

这种二次查表方式的检索量小，速度快，进一步提高了本发明的实用性。

#### 附图说明

图 1 为本发明基本结构示意图；

图 2 为本发明中数据压缩控制流程示意图；

图 3 为本发明中数据解压缩控制流程示意图。

#### 具体实施方式

下面根据附图和实施例对本发明作进一步详细说明：

根据图 1，本发明包括 DCT 模块 1、量化模块 2、Zig-Zag 扫描模块 3、HUFFMAN 模块 4、码流整理模块 5 并依次相连，以及存储器 6 和主控模块 7。

如图 1 所示，DCT 模块 1 对数据进行离散余弦变换转换及反变换，量化模块 2 完成数据的量化或反量化计算，Zig-Zag 扫描模块 3 完成数据的串联或解串联，HUFFMAN 模块 4 完成数据的 HUFFMAN 编码或解码，所述的码流整理模块 5 完成数据的打包或分包处理；

如图 1 所示，存储器 6 中保存有关表项，其中的表项包括量化表 QT0-DC、QT0-AC、QT1-DC、QT1-AC，和 HUFFMAN 编解码表，HUFFMAN 编解码表中包括直流 HUFFMAN 编解码表 HT0-DC、HT1-DC，和交流 HUFFMAN 编解码表 HT0-AC、HT1-AC。

如下表 4 所示，直流 HUFFMAN 编解码表包括：

类值 SSSS、码字长度 Code length、码字 Code word，其中，类值 SSSS 为表项访问入口。

| 表项访问入口                | 表 体                   |                   |
|-----------------------|-----------------------|-------------------|
| 类值 Category<br>(SSSS) | 码字长度<br>(Code length) | 码字<br>(Code word) |
| 0                     |                       |                   |
| 1                     |                       |                   |
| 2                     |                       |                   |
| 3                     |                       |                   |
| 4                     |                       |                   |
| 5                     |                       |                   |
| 6                     |                       |                   |
| 7                     |                       |                   |
| 8                     |                       |                   |
| 9                     |                       |                   |
| 10                    |                       |                   |
| 11                    |                       |                   |

表 4

如下表 5 所示，交流 HUFFMAN 编解码表包括：

零游长 RRRR/类值 SSSS、表地址 Table Address、码字长度 Code length、码字 Code word。

零游长 RRRR/类值 SSSS 为表项访问入口，表地址 Table Address 通过零游长 RRRR/类值 SSSS 直接取得，零游长 RRRR 是指非零交流系数前面的“0”交流系数的个数。

| 表项访问入口                             |                        | 表 体                   |                   |
|------------------------------------|------------------------|-----------------------|-------------------|
| 零游长/类值<br>Run/Size<br>(RRRR, SSSS) | 表地址<br>(Table Address) | 码字长度<br>(Code length) | 码字<br>(Code word) |
| 0/0                                | 0x00                   |                       |                   |
| 0/1                                | 0x01                   |                       |                   |
| 0/2                                | 0x02                   |                       |                   |
| .                                  | .                      |                       |                   |
| .                                  | .                      |                       |                   |
| 1/1                                | 0x11                   |                       |                   |
| 1/2                                | 0x12                   |                       |                   |
| 1/3                                | 0x13                   |                       |                   |
| .                                  | .                      |                       |                   |
| .                                  | .                      |                       |                   |
| .                                  | .                      |                       |                   |
| .                                  | .                      |                       |                   |
| .                                  | .                      |                       |                   |
| F/0                                | 0xf0                   |                       |                   |
| F/1                                | 0xf1                   |                       |                   |
| F/2                                | 0xf2                   |                       |                   |
| .                                  | .                      |                       |                   |
| .                                  | .                      |                       |                   |

表 5

如图 1 所示，主控模块 7 控制 DCT 模块 1、量化模块 2、Zig-Zag 扫描模块 3、HUFFMAN 模块 4、码流整理模块 5 的工作状态，对原始图像数据完成方块化处理发送至 DCT 模块 1，或提取压缩数据块至码流整理模块 5，该主控模块 7 根据处理进程对存储器 6 中的表项进行相应的调用，向量化模块 2 或 HUFFMAN 模块 4 加载信息。

在本发明的压缩或解压缩中，主控模块 7 调用所述量化表 QT0-DC、

---

QT0-AC、QT1-DC、QT1-AC，加载至量化模块2。

所述的主控模块7调用所述HUFFMAN编解码表，并对其进行排序，加载至HUFFMAN模块4，解码时向HUFFMAN模块4加载解压缩过程中用到的从HUFFMAN码表中提取出来的码字特征信息，码字特征信息包括每个长度HUFFMAN码的开始地址、开始码字、以及表示每个长度的码字是否有效的标志等。

在解压缩中，主控模块7按照码字长度Code length，且对于每个长度的所有码字根据大小顺序对HUFFMAN编解码表排序，排序后的HUFFMAN编解码表的表项对应于按顺序的地址入口。

如图2所示，本发明的具体压缩控制流程如下：

- 1、 主控模块7向量化模块2和HUFFMAN模块4分别导入量化表和HUFFMAN编码码表。
- 2、 主控模块7由原始图像数据取出一个方块化的8X8数据单元发送至DCT模块1。
- 3、 DCT模块1对该数据单元进行离散余弦变换的编码转换，转换后的数据流传递至量化模块2。
- 4、 量化模块2根据量化表QT0-DC、QT0-AC、QT1-DC、QT1-AC对数据流中频域的数据进行量化计算，得到量化后的DCT系数。
- 5、 Zig-Zag扫描模块3对量化后的DCT系数进行Zig-Zag扫描，完成数据的串联。
- 6、 HUFFMAN模块4首先获取本数据单元相对于前一个数据单元的相同图像分量的直流差值的幅度magnitude。
- 7、 HUFFMAN模块4通过所述幅度magnitude计算出类值SSSS；

在本发明中，通过下式转换得到相应的类值 SSSS（下式中的“MAG”即为幅度 magnitude）：

```

if MAG[DIFF][10] = 0; SSSS = 11; ELSE
if MAG[DIFF][9] = 0; SSSS = 10; ELSE
if MAG[DIFF][8] = 0; SSSS = 9; ELSE
if MAG[DIFF][7] = 0; SSSS = 8; ELSE
if MAG[DIFF][6] = 0; SSSS = 7; ELSE
if MAG[DIFF][5] = 0; SSSS = 6; ELSE
if MAG[DIFF][4] = 0; SSSS = 5; ELSE
if MAG[DIFF][3] = 0; SSSS = 4; ELSE
if MAG[DIFF][2] = 0; SSSS = 3; ELSE
if MAG[DIFF][1] = 0; SSSS = 2; ELSE
if MAG[DIFF][0] = 0; SSSS = 1; ELSE
SSSS = 11;

```

- 8、HUFFMAN 模块 4 根据类值 SSSS 由相应的直流 HUFFMAN 编码码表 HT0-DC、HT1-DC（如表 4）取得相应的 HUFFMAN 码字，其中编码表 HT0-DC 用于 Y 分量直流系数的编码，HT1-DC 用于 U、V 分量直流系数的编码。
- 9、HUFFMAN 模块 4 将所得到的 HUFFMAN 码加上直流系数差值的有效数据，有效数据的位宽即类值 SSSS 的大小，完成直流系数的编码。
- 10、HUFFMAN 模块 4 经过统计计算每个非零交流系数前面的“0”交流系数的个数，同时计算这个非零交流系数的位宽。
- 11、根据“0”交流系数的个数值（即零游长 RRRR）和位宽（即类值 SSSS）由交流 HUFFMAN 编码码表 HT0-AC、HT1-AC（如表 5）中对应表地址 Table Address 处直接取得相应的 HUFFMAN 码字。
- 12、将所得到的 HUFFMAN 码加上这个非零交流系数的有效数据完成交

流系数 HUFFMAN 编码，通过上述步骤 6-步骤 12，HUFFMAN 模块 4 根据 HUFFMAN 编码码表对经过 Zig-Zag 扫描所产生的 64 个系数进行相应的 HUFFMAN 编码，步骤 6-步骤 9 完成第一个系数的直流系数编码，步骤 10-步骤 12 完成后面 63 个系数的交流系数编码。

- 13、HUFFMAN 编码后的数据经码流整理模块 5 打包处理，完成对该数据单元的压缩。

如图 3 所示，本发明的具体解压缩控制流程如下：

- 1) 主控模块 7 调用 HUFFMAN 解码码表，并对其进行排序，加载至 HUFFMAN 模块 4，且将量化表 QT0-DC、QT0-AC、QT1-DC、QT1-AC 导入量化模块 2，在对 HUFFMAN 解码码表的排序中，主控模块 7 按照码字长度 Code length，且对于每个长度的所有码字根据大小顺序对 HUFFMAN 解码码表排序，排序后的 HUFFMAN 解码码表的表项对应于按顺序的地址入口。

直流 HUFFMAN 解码码表排序如下表 6 所示：

| 存储访问入口                |                           | 存储体                           |
|-----------------------|---------------------------|-------------------------------|
| 码字长度<br>(Code Length) | 码字地址入口<br>(Code Position) | 类值 Category ([3:0])<br>(SSSS) |
|                       | 0                         | 0000                          |
|                       | 1                         | 0001                          |
|                       | 2                         | 0010                          |
|                       | 3                         | 0011                          |
|                       | 4                         | 0100                          |
|                       | 5                         | 0101                          |
|                       | 6                         | 0110                          |
|                       | 7                         | 0111                          |
|                       | 8                         | 1000                          |
|                       | 9                         | 1001                          |
|                       | 10                        | 1010                          |
|                       | 11                        | 1011                          |

表 6

交流 HUFFMAN 解码码表排序如下表 7 所示：

| 存储访问入口                |                           | 存储体                      |                           |
|-----------------------|---------------------------|--------------------------|---------------------------|
| 码字长度<br>(Code Length) | 码字地址入口<br>(Code Position) | 零游长 (RRRR)<br>Run([7:4]) | 类值 (SSSS)<br>Size ([3:0]) |
| 2                     | 0                         |                          |                           |
| 2                     | 1                         |                          |                           |
| .                     |                           |                          |                           |
| .                     |                           |                          |                           |
| 3                     |                           |                          |                           |
| 3                     |                           |                          |                           |
| .                     |                           |                          |                           |
| .                     |                           |                          |                           |
| 16                    |                           |                          |                           |
| 16                    |                           |                          |                           |
| .                     |                           |                          |                           |
| .                     |                           |                          |                           |
|                       |                           |                          |                           |

表 7

- 2) 主控模块 7 从 HUFFMAN 表提取码字特征信息，且向 HUFFMAN 模块 4 加载该信息，码字特征信息包括每个长度 HUFFMAN 码的开始地址、开始码字、以及表示每个长度的码字是否有效的标志；
- 3) 主控模块 7 提取压缩数据包至码流整理模块 5。
- 4) 码流整理模块 5 对压缩数据包进行分包处理，将分出的数据码流发送至 HUFFMAN 模块 4。
- 5) HUFFMAN 模块 4 读取其中一组数据码流，根据码字特征信息，通过与每个长度的开始码字进行比较，确定该数据码流中 HUFFMAN 码的码字长度。

- 
- 6) 通过码字长度从数据码流中取出 HUFFMAN 码字, 把取出的码字与相应长度的开始码字相比较, 计算出在 HUFFMAN 码表中该码字相对于开始码字的相对位置。
  - 7) 通过开始码字在 HUFFMAN 表中的位置和计算所得到的相对位置算出该码字在 HUFFMAN 码表中的绝对位置, 通过上述步骤 5) - 步骤 7), HUFFMAN 模块 4 根据配置信息和提取信号确定 HUFFMAN 码的长度以及在 HUFFMAN 编解码表中的位置。
  - 8) HUFFMAN 模块 4 以所取得的长度、位置信息为检索入口, 从 HUFFMAN 解码码表中取出相应的类值 SSSS 或零游长 RRRR/类值 SSSS, 对于直流信号, 由表 6 中取出相应的类值 SSSS, 对于交流信号, 由表 7 中取出相应的零游长 RRRR/类值 SSSS。
  - 9) HUFFMAN 模块 4 通过类值 SSSS 或位宽 SSSS 得到数据码流中系数的有效位宽, 从码流中取出相应个 bit 的码流, 作为系数的有效数据, 对于交流信号, 零游长 RRRR 表示该系数前面所具有“0”系数的个数, 对数据码流完成 HUFFMAN 解码。
  - 10) Zig-Zag 扫描模块 3 对解码后的数据流进行反 Zig-Zag 扫描, 完成数据流的解串联。
  - 11) 量化模块 2 根据量化表对数据流进行反量化操作, 产生频域数据。
  - 12) DCT 模块 1 将数据流的频域数据转化为重构图像的时域数据, 完成解压缩。

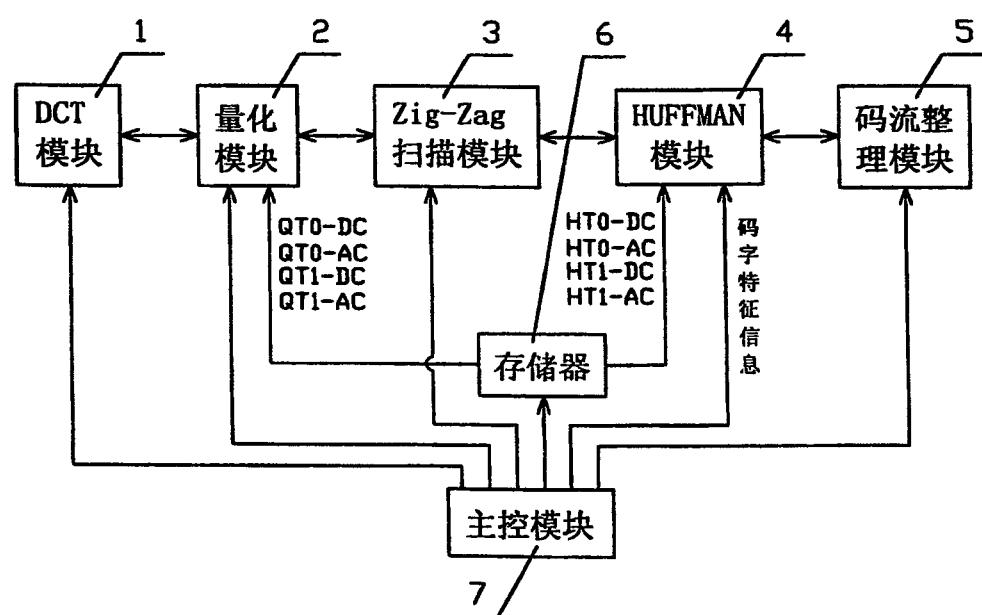


图1

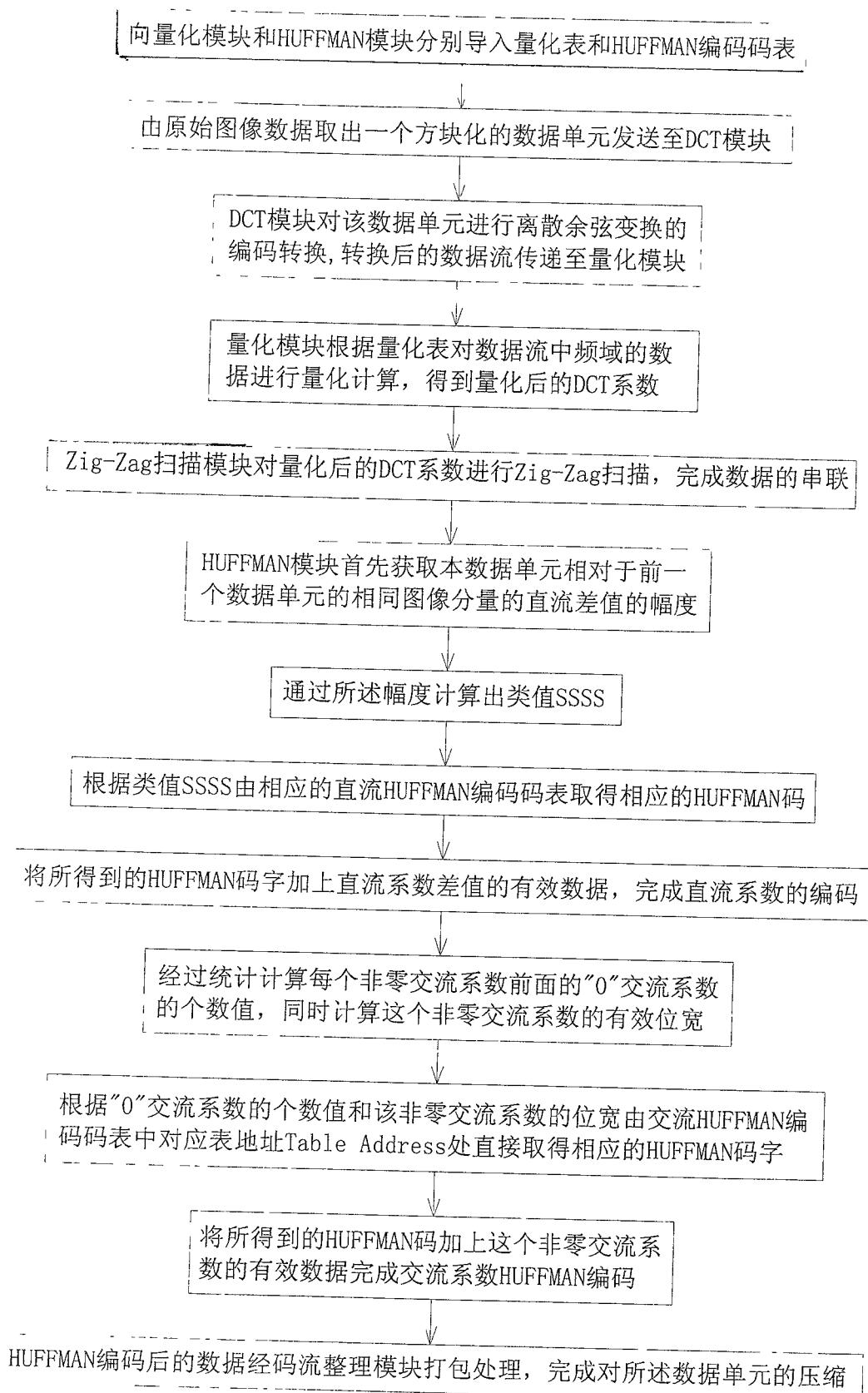


图2

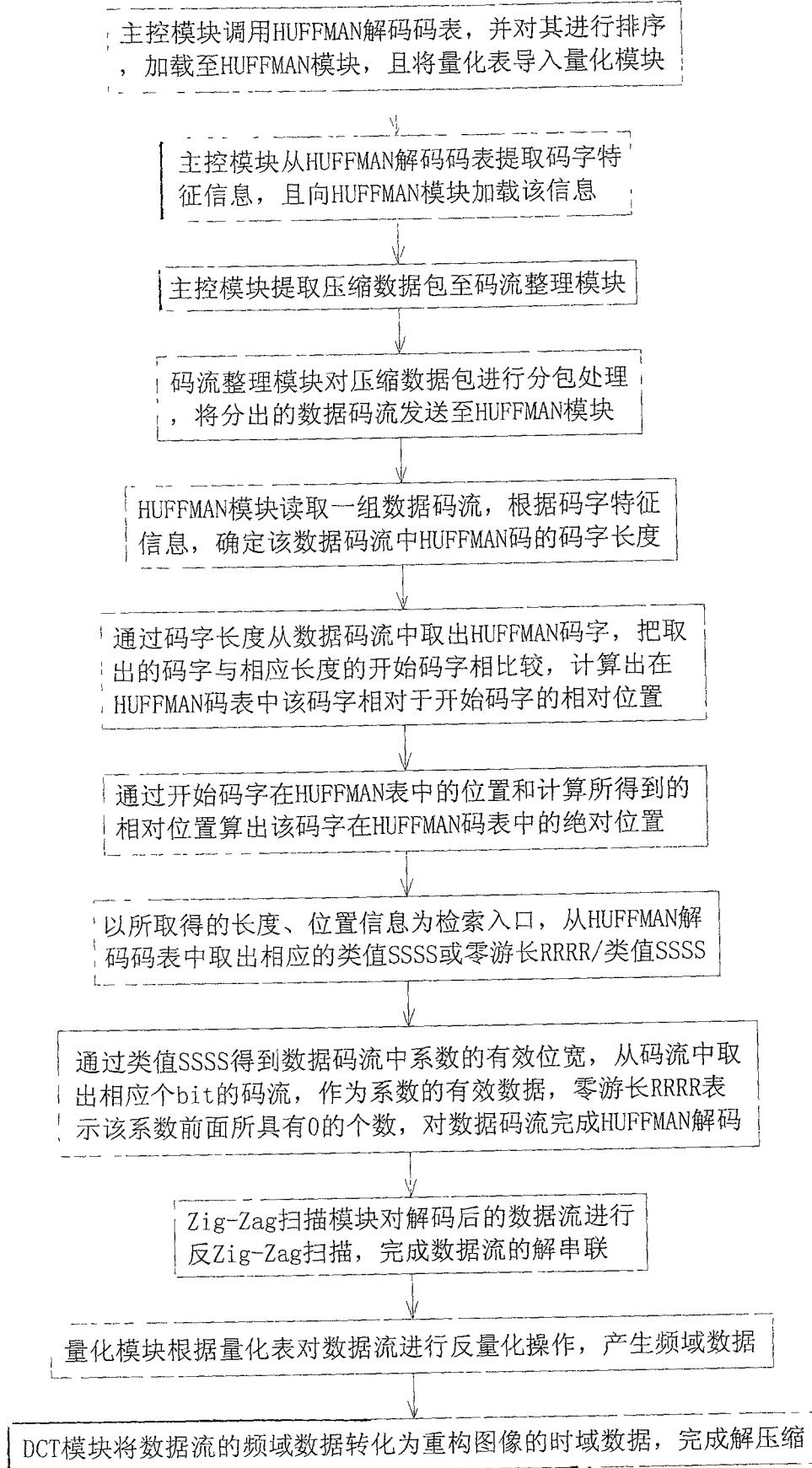


图3