

# 单片机中实现数字图像压缩的算法研究

王 晓, 杨维剑

(四川理工学院计算机学院, 四川 自贡 643000)

**摘 要:** 文章主要论述了在单片机中对数字图像压缩算法的硬件设计和如何在单片机实现对数字图像压缩成 JPEG 格式算法的具体实现方法, 给出了在图像压缩算法中离散余弦变换 (DCT) 函数、Huffman 编码函数等关键函数, 并对其算法加以改进, 对其复杂性进行了可靠的分析。

**关键词:** 图像压缩; DCT; Huffman; 数字图像; JPEG

**中图分类号:** TP393.02

**文献标识码:** A

## 引 言

目前实现图像压缩的方法很多, 但是主要集中在 DSP 和 PC 机中进行, 而在单片机中实现数字图像压缩, 则主要存在内存容量和计算速度两大问题。随着单片机技术和存储技术的发展, 在单片机中实现数字图像压缩成为可能。由于数字图像的数据量很大, 远距离成像、传输等就必须对其进行压缩编码。利用单片机实现数字图像压缩, 提供高可靠性、高性价比的应用系统, 不仅在军事侦察、抢险救灾、气象探测等领域有着广泛应用, 而且在工业检测领域 (如外观在线检测、特殊图形检测等)、高温作业等领域有着更加广泛的应用。

## 1 硬件设计与系统实现

为了解决单片机中内存容量问题, 设计两片具有 8M 位的存储容量芯片 L4V8M440 该芯片是由 AVER-LOGIC 推出的一款性价比较高的视频帧存储芯片, 目前市场上 FIFO 的存储容量均比较小, 包括该公司前几年推出的具有 3M 位容量的 AL422 系列 FIFO 芯片, 也逐渐无法适应目前高分辨率视频技术的发展。L4V8M440 沿袭了 AL422B 的设计思想, 从功能、容量和速度上比 AL422 均有显著提高<sup>[1]</sup>。L4V8M440 具有 8M 位存储容量, 可以满足 1024 × 768 以上高分辨率视频技术的需要, 且存储速度可达 80MHz, 具有异步读写时钟, 用户可控的读写使能, 输入输出使能及读写复位输入端口, 可以通过 PC 读写其内部寄存器控制读写初始地址, 可以

方便进行总线扩展。一片用于扇入, 一片用于扇出。其硬件逻辑框图如图 1 所示。



图 1 硬件逻辑框图

为了解决速度问题, 我们选取了 C8051F120 作为 CPU, Cygnal 公司推出高性能 8051 微控制器 C8051F120 系列产品, CPU 吞吐量达 100M PS, 具备两个周期 16 × 16 MAC, 12 位 ADC 和 DAC, 包括 128KB 闪存、8448KB RAM<sup>[2]</sup>。

系统实现的关键是如何将图像芯片中的数据信息存储到图像扇入存储芯片, 以及经过压缩后形成 JPEG 图像格式转存到图像扇入存储芯片。在本设计中图像芯片使用 3620/OV3610 (OV3620/OV3610 是 Omni Vision Technologies, inc 2005 年推出的高度集成、高分辨率 (1024 × 768)、隔行逐行扫描、CMOS 的彩色黑白的图像采集芯片)<sup>[3]</sup>。根据图像芯片的行时序图和场时序, 以及 L4V8M440 的读写时序, 利用 OV3620/3610 的输出信号 PCLK、HREF、VSYNC; L4V8M440 的 WCK、WE、RCK、RE 和 OE 等信号; 单片机的 P1.1、P2.7、ALE、WR 和 RD 等信号之间的逻辑关系为:  $WCK = PCLK$ ;  $RCK = ALE + \overline{WR} + \overline{RD}$ ;  $INT0 = VSYNC$ ;  $WE = \overline{HREF} + VSYNC + P1.1$ ;  $RE/OE = RD + P2.7$ ; 从而实现了图像数据的扇入与扇出存储。

## 2 图像压缩

图像压缩的算法很多。我们选取比较流行的 JPEG 压缩标准进行压缩。基本 JPEG 编码器和解码器 (基于 DCT) 的结构如图 2 所示。其算法的基本思路是: DCT 函数通过把数据从空间域变换到频率域, 从而去除数据的冗余度; 量化器用加权函数来产生对人眼优化的量化 DCT 系数, 同时熵编码器将量化 DCT 系数的熵最小化<sup>[4]</sup>。

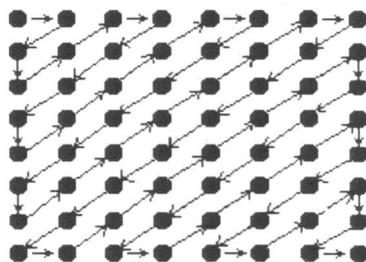


图 4 “Z”行排列

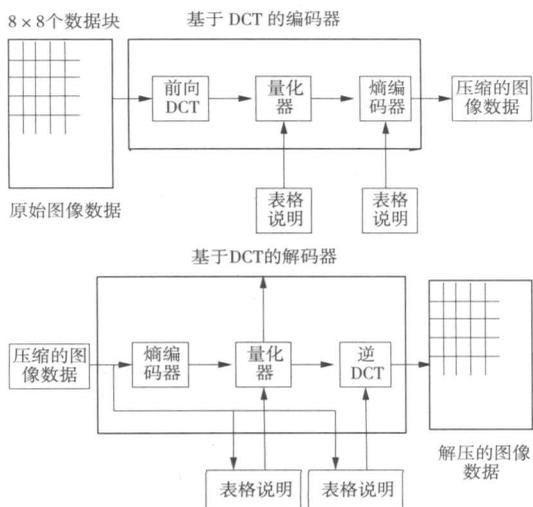


图 2 基于 JPEG 编码器和解码器 (基于 DCT) 的结构

基本 JPEG 的编码方法为顺序编码。其步骤是: 首先, 将图像分为  $8 \times 8$  的像素块, 根据从左到右, 从上到下的光栅扫描方式进行排序 (如图 3 所示)。DCT 对  $8 \times 8$  的像素块进行计算, 再对 64 个 DCT 系数用均匀量化器进行量化, 均匀量化表是根据心理听觉的实验得出的。这种均匀的标量量化表可以作为 JPEG 标准的一部分, 但不是必须的。将 DCT 系数量化后, 块中的系数再根据“Z”形扫描方式 (如图 4 所示) 排序, 得到的比特流用行顺序编码生成中间的符号序列, 然后这些符号经过 Huffman 编码用于传输或存储<sup>[5]</sup>。

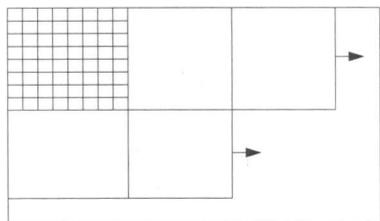


图 3 将图像分成  $8 \times 8$  的像素块

由此可以得出, 基于 DCT 编码的 JPEG 压缩算法可由如下几个步骤实现:

- (1) 颜色模式转换及采样;
- (2) 正向离散余弦变换 (FDCT);
- (3) 量化 (Quantization);

- (4) Z 字型编码;
- (5) 使用差分脉冲编码调制 (DPCM) 对直流系数 (DC) 进行编码;
- (6) 使用行程长度编码 (RLE) 对交流系数 (AC) 进行编码;
- (7) 熵编码 (Entropy Coding)。

在整个过程中, 最为关键的编程技术是离散余弦变换 (DCT) 和 Huffman 编码。

### 2.1 离散余弦变换 (DCT)

离散余弦变换 (Discrete Cosine Transform, 简称 DCT) 是一种与傅立叶变换紧密相关的数学运算。由于彩色图像由像素组成, 这些像素具有 RGB 彩色值。每个像素都带有  $x, y$  坐标, 对每种原色使用  $8 \times 8$  或者  $16 \times 16$  矩阵。在灰度图像中像素具有灰度值, 它的  $x, y$  坐标由灰色的幅度组成。为了在 JPEG 中压缩灰度图像, 每个像素被翻译成亮度或灰度值。为了压缩 RGB 彩色图像, 这项工作必须进行三遍, 因为 JPEG 分别地处理每个颜色成分, R (Red 红) 成分第一个被压缩, 然后是 G (Green 绿) 成分, 最后是 B (Blue 蓝) 成分。而一个  $8 \times 8$  矩阵的 64 个值, 每个值都带有各自的  $x, y$  坐标, 这样就有了一个像素的三维表示法, 称作空间表达式或空间域。通过 DCT 变换, 空间表达式就转化为频谱表达式后频率域。从而达到了数据压缩的目的<sup>[6]</sup>。

其二维 DCT 正变换的公式为 (采用  $8 \times 8$  的块):

$$F(u, v) = \frac{1}{4} C(u) C(v) \left[ \sum_{i=0}^7 \sum_{j=0}^7 f(i, j) \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} \right] \quad (1)$$

DCT 的逆变换采用公式:

$$f(i, j) = \frac{1}{4} c(u) c(v) \left[ \sum_{i=0}^7 \sum_{j=0}^7 F(u, v) \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} \right] \quad (2)$$

在 (1) 式、(2) 式中:

$$C(u), C(v) = 1/\sqrt{2} \text{ (当 } u, v = 0 \text{)}$$

$$C(u), C(v) = 1 \text{ (其它情况)}$$

$f(i, j)$  表示位于  $(i, j)$  坐标处的像素函数 (如亮度等)。这样经过正变换所得的结果就为 DCT 系数, 其中的  $F(0, 0)$  称为直流系数 (DC), 而其他的则称为交流系数

(AC)。二维的离散余弦变换具有可分离性,所以其正变换和逆变换均可以将二维变换分解成一系列一维变换(行、列)进行计算。即:

$$F(u, v) = \frac{1}{2}C(u) \left[ \sum_{i=0}^7 G(i, v) \cos \frac{(2i+1)u\pi}{16} \right] \quad (3)$$

和

$$G(i, v) = \frac{1}{2}C(v) \left[ \sum_{j=0}^7 f(i, j) \cos \frac{(2j+1)v\pi}{16} \right] \quad (4)$$

在(3)式、(4)式中,最为关键是用 C51 汇编求余弦函数。传统求法是用 C51 求解或是用浮点数的方法求解。而我们采用定点小数求解余弦函数可大大加快计算速度。利用浮点数求解在主频 12MHZ 时约需 2.2 ms, 但用定点小数求解只需 250 $\mu$ s, 二者相差十倍。如用本文提出的 CPU 来处理,用定点小数求解只需 2.5 $\mu$ s 左右。我们采用的求解公式为:

$$\cos x = \sin \left( \frac{\pi}{2} - x \right)$$

和

$$\sin \frac{\pi}{2} x \approx x + (0.57036 - 0.6425 - 0.07185x^2) \times x^2$$

用三字节的定点小数加以实现,其精度完全达到 DCT 算法的要求。

## 2.2 Huffman 编码

Huffman 编码是一种比较经典的算法,是 1952 年由 Huffman 提出的一种编码方法。这种编码方法的主要思想是根据源数据符号出现的概率进行编码,出现概率越高的符号,分配以越短的编码,反之,分配以较长的编码,从而达到用尽可能少的编码符号表示数据源。该算法的步骤如下:

- (1) 概率统计(对一副图像做灰度信号统计),得到  $n$  个不同概率的信息符号;
- (2) 将信源符号按概率递减顺序排序;
- (3) 把两个最小概率相加作为新符号的概率,并按(2)重排;
- (4) 重复(1)、(2),直到概率为 1;
- (5) 在每次合并信源时,将合并的信源分别赋“0”和“1”;
- (6) 寻找从每一信源符号到概率为 1 处的路径,记录

路径上的“1”和“0”;

(7) 写出每一符号的“1”,“0”序列(从树跟到信源符号节点)。

可见在计算 Huffman 编码时,需要对原始图像扫描两遍:第一遍扫描要精确地统计出原始图像中的每个灰度值出现的概率;第二遍扫描是建立 Huffman 树并进行编码,进行数据压缩,形成 JPEG 图像数据。

因此,我们在硬件设计中设计了两片 FIFO 芯片 L4V8M440 在具体实现的过程中,根据 FIFO 芯片的特点,我们每次从 L4V8M440 读入一定量的图像数据,经过 DCT 处理后存放在 C8051F120 的内部,同时对其进行统计,当 DCT 完成第一遍后,统计也就完成。第二遍时对其建立 Huffman 树并进行编码,进行数据压缩,形成 JPEG 图像数据存放在另一片 L4V8M440 中,从而完成了对整个图像数据的压缩和编码。

## 3 结束语

用以上的原理和方法,在钻井平台井下倾角测试系统中得到了很好的应用,并实现了图像数据很好的压缩效果。具有很高的推广价值。为 CNG 储气井检测技术与系统研发中的数字成像、存储和压缩提供了可靠的技术支持。

## 参考文献:

- [1] AverLogic Technologies Inc. AL4V8M440 Data Sheet [EB/OL]. 2004 [http //www. averLogic. com](http://www.averLogic.com).
- [2] Silicon Laboratories Inc. C8051F120/1/2/3/4/5/6/7 C8051F130/1/2/3 Data Sheet [EB/OL]. 2005 [http // www. silabs. com](http://www.silabs.com).
- [3] OMNIVISION TECHNOLOGIES, Inc. OV3610 Color CMOS QXGA (3.1M Pixel) CAMERACHIPIM [EB/OL]. 2004-3-18 [http //www. ovt. com](http://www.ovt.com).
- [4] 阴躲芬. 浅谈数字图像压缩编码技术 [J]. 科技广场, 2008 (1): 138-139
- [5] 朱明轩. 数字图像压缩中的域变换技术 [J]. 电脑知识与技术, 2008 34 1745-1746
- [6] 粟塔山. 最优化计算原理与算法程序设计 [M]. 长沙: 国防科技大学出版社, 2001

## Research on Realizing the Algorithm for Digital Image in Single Chip Microcomputer

WANG Xia, YANG Wei-jian

(School of Computer Science, Sichuan University of Science & Engineering, Zigong 643000, China)

**Abstract** The hardware design for digital image compression algorithm in the SCM, and how to make the digital image compress to the format of JPEG are discussed in the paper. Some crucial functions such as DCT and Huffman is introduced as well. At last, some improvement for their algorithms are given. We also do some research on the complexity of these algorithms.

**Keywords** picture compress, DCT, Huffman, numerical picture, JPEG