

用查表法快速实现二维 8 × 8 离散余弦逆变换的研究

纪秀花^{1,2}, 张彩明^{1,2}, 韩慧健²

(1. 山东大学计算机科学与技术学院, 山东济南 250061; 2. 山东经济学院计算机科学与技术学院, 山东济南 250014)

摘要: 提出一种基于查表法的二维 8 × 8 离散余弦逆变换(2D 8 × 8 IDCT)的快速算法,其查找表 LUT(Look-Up Table)结构的设计是基于二维 8 × 8 DCT 的基本图像. 利用两种技术减小查找表长度: 利用基本图像的对称特性; 通过对离散余弦正变换(DCT)和量化过程的分析,推导出每个量化后 DCT 系数的取值范围,使得查找表只有 10.9746K 项数据,若量化矩阵具有对称性 $q(u, v) = q(v, u)$, LUT 的长度还可减少近半. 新算法利用查表法消除 IDCT 中乘法运算,并利用图像数据的特点和基本图像的对称特性大大减少加法次数,提高了计算速度. 以多幅标准图像为样本数据进行实验,结果表明:新算法实现 2D 8 × 8 IDCT 运算平均只需加法 182 次. 与当前运算量最小的 Feig 快速算法做比较,新算法避免了乘法,所需加法次数也降低了约 15%.

关键词: 图像压缩; 离散余弦逆变换; 基本图像; 查表法

中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2008) 04-0639-07

A Fast 2D 8 × 8 IDCT Algorithm Based on Look-Up Table

Ji Xiu-hua^{1,2}, ZHANG Cai-ming^{1,2}, HAN Hui-jian²

(1. School of Computer Science and Technology, Shandong University, Jinan, Shandong 250061, China;

2. School of Computer Science and Technology, Shandong Economic University, Jinan, Shandong 250014, China)

Abstract: A new fast two dimension 8 × 8 Inverse Discrete Cosine Transform (IDCT) algorithm based on Look-Up Table (LUT) is presented, in which, the LUT structure is based on the DCT basic images. The LUT size is decreased by using two techniques. One is to use the symmetries of the basic images; another is to elicit the range of each quantized DCT coefficient. As a result, the LUT only has 10.9746K data. If the quantization matrix has the symmetry $q(u, v) = q(v, u)$, the LUT size can be decreased half nearly again. The new algorithm avoids multiplications by using LUT and decreases additions greatly by utilizing the characteristics of actual images and the symmetries of the basic images. The experimental results indicate that it only needs 182 additions averagely and decreases about 15 percent of additions besides eliminating multiplications compared with Feig's algorithm that needs least arithmetic operations nowadays.

Key words: image compression; inverse discrete cosine transform; basic image; look-up table

1 引言

二维离散余弦变换(DCT)和离散余弦逆变换(IDCT)在许多图像和视频数据压缩应用中起着关键的作用,是现有多种多媒体压缩标准(例如:JPEG、H.26X和MPEG)的重要组成部分.随着VLSI技术的进步,视频通信终端变得越来越小,而且具有可移动性,例如视频手机和掌上电脑.由于这些终端没有特殊硬件支持,它们对高速且占存储少的数据压缩算法有更强烈的要求. DCT/IDCT的运算量很大,在MPEG1解码过程中计算IDCT的时间约占总时间的38%,因而在一些高速或实时场合能否快速实现DCT就成为一个关键因素.为减

少运算次数,缩短运算时间,人们作了不懈的努力,提出了多种快速算法,如文献[1~6],其中最有影响的见表1.为了提高速度,还出现了一些近似计算算法^[7~9].但处理过程中数学运算仍然相当复杂.在集成电路实现时乘法占用较大芯片面积和功耗.在可移动产品应用蓬勃发展的今天,如何在保持高性能的情况下减小芯片面积和功耗,无疑是设计的关键,这要求设计者对芯片的重要部件进行各方面的优化.因此很多被提出的DCT实现方法都强调减少乘法器数量.因为DCT中的余弦系数为定值,所以一些中间结果可以预先计算存于存储器中.文献[10~14]提出利用查表法计算DCT/IDCT.由于存储器芯片愈来愈便宜,查表法愈来愈得到重视.文献

收稿日期:2007-01-08;修回日期:2007-04-26

基金项目:国家自然科学基金(No. 60573114, No. 60573181);山东自然科学基金(No. Y2005G15)

[10, 11]是将变换所需的乘积项结果预先计算并存成一张表,通过查表避免乘法运算;文献[12~14]的基本思想是通过存储器查表、移位和累加器代替了乘法器.文献[10~14]算法虽然避免了乘法,但比表1中的快速算法存在更多的加法运算.文献[10, 11]算法占用存储量较大(约100KB),文献[12~14]算法虽然占用存储量小但引入许多移位运算.目前,Feig算法是当前运算量最小、最有影响的准确计算DCT的快速算法.

表1 8×8点IDCT运算所需运算次数

算法	乘法	加法	总数
Chen[4]	256(208)	416(464)	672(672)
Lee[3]	192	464	656
Loeffler[2]	176	464	640
Feig[1]	94	454	548
Feig Scaled DCT[1]	54	462	516

本文提出一种基于基本图像的查表法实现2D 8×8 IDCT准确计算的快速算法.该算法不仅消除了IDCT中所需的乘法运算,而且,它只针对非零的DCT系数进行逆变换计算,这样可充分利用图像数据特点大大减少加法次数,运算速度得到很大提高.另外,反量化过程也很容易嵌入此算法中并且不增加额外运算量.鉴于在图像、视频数据压缩中,一般8×8子块的DCT系数中,只有约10%的数据为非零值(特殊情况为7~16个)^[15],所以通过实验看出:在实际图像解压时,此算法是很有效的.为了减少查找表所需的存储容量,对DCT和量化过程进行分析,给出了每个量化后DCT系数的取值范围,使得查找表的长度大大降低.另外,通过巧妙地设计查找表LUT的结构,使整个计算过程中查表时间很小.本文首先介绍基于基本图像表示的IDCT的表达式及基本图像的对称性情况,然后说明查找表LUT的结构设计以及进行查找LUT的寻址方式,进而叙述了本文IDCT的快速算法思想.

2 DCT及其基本图像的性质

本算法的基本思想是利用基于基本图像表示的IDCT的表达式,将所需乘积项结果预先计算存成一张查找表LUT.根据基本图像的对称性情况及每个量化后DCT系数的取值范围减少LUT的大小,并且充分利用实际图像数据的特点和基本图像的对称性极大地减少加法次数,提高运算速度.下面将讨论基于基本图像的IDCT表示式及基本图像对称性情况.

2.1 用基本图像表示IDCT

二维8×8 DCT变换定义为:

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{m=0}^7 \sum_{n=0}^7 (f(m, n) \cos \frac{(2m+1)u}{16} \cos \frac{(2n+1)v}{16}) \quad (1)$$

二维8×8 IDCT变换定义为:

$$f(m, n) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 (C(u) \cdot C(v) \cdot F(u, v) \cos \frac{(2m+1)u}{16} \cos \frac{(2n+1)v}{16}) \quad (2)$$

其中: $m, n, u, v = 0, 1, \dots, 7$

$$C(s) = \begin{cases} 1/\sqrt{2} & s=0 \\ 1 & \text{其他} \end{cases}$$

式(1)和(2)也可用矩阵形式表示:

二维8×8 DCT为: $F = G \cdot f \cdot G^T$

二维8×8 IDCT为: $f = G^T \cdot F \cdot G$

$$G = \begin{bmatrix} \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \dots & \frac{1}{2\sqrt{2}} \\ \frac{1}{2} \cos \frac{1}{16} & \frac{1}{2} \cos \frac{3}{16} & \dots & \frac{1}{2} \cos \frac{15}{16} \\ \dots & \dots & \dots & \dots \\ \frac{1}{2} \cos \frac{7}{16} & \frac{1}{2} \cos \frac{21}{16} & \dots & \frac{1}{2} \cos \frac{105}{16} \end{bmatrix} = \begin{bmatrix} G_0 \\ G_1 \\ \dots \\ G_7 \end{bmatrix}$$

G_u 分别为矩阵G中各行所组成的矢量,二维8×8 IDCT还可表示为:

$$f = \sum_{u=0}^7 \sum_{v=0}^7 (F(u, v) \times G_u^T \times G_v) = \sum_{u=0}^7 \sum_{v=0}^7 (F(u, v) \times T_{uv}) \quad (3)$$

其中: $T_{uv} = G_u^T \times G_v$ ($u, v = 0, 1, \dots, 7$)被称为系数 $F(u, v)$ 对应的基本图像.设 $t_{uv}(m, n)$ 为 T_{uv} 矩阵的元素,8×8 DCT正变换又可写为:

$$F(u, v) = \sum_{m=0}^7 \sum_{n=0}^7 (f(m, n) \cdot t_{uv}(m, n)) = f \cdot T_{uv} \quad (4)$$

为了提高运算效率,可把反量化过程加入到IDCT变换计算中去.可得:

$$f = \sum_{u=0}^7 \sum_{v=0}^7 [D(u, v) \times T_{uv}] \quad (5)$$

其中 $T_{uv} = T_{uv} \cdot q(u, v)$ ($q(u, v)$ 为量化步长), $D(u, v)$ 为量化后的DCT系数:

$$D(u, v) = \text{int}[(F(u, v) / q(u, v)) + 0.5]$$

2.2 基本图像对称性情况

基本图像具有对称性^[15].为说明对称性情况,可将式(1)~(5)中的 u, v 表示为:

$$u = 2^{h_u} \times L_u, \quad v = 2^{h_v} \times L_v$$

其中 h_u, h_v 取非负整数, L_u, L_v 取值为0或奇数.当 $u, v = 1, 2, \dots, 7$ 时, $h_u, h_v = 0, 1, 2$, 而当 u 或 $v = 0$ 时, h_u, h_v 可取任意有限值,此时记 h_u 和 h_v 为 \cdot .这样 h_u 和 h_v 各有0, 1, 2和 \cdot 四种取值.

性质1 具有相同 h_u, h_v 的 $F(u, v)$ 系数所对应的基本图像 T_{uv} 有相同的对称性^[15].

性质2 根据性质1,可将具有相同 h_u, h_v 的基本

图像分为一类,则二维 8 × 8 DCT 的 64 个基本图像可分为 16 种对称性情况^[15]。

性质 3 定义每个基本图像中的左上角矩阵块(大小为 $W_u \times W_v$, 其中 $W_u = \frac{4}{2^{h_u}}$, $W_v = \frac{4}{2^{h_v}}$)为最小对称单元,其余部分可由此单元做行列奇/偶对称来形成^[15]。(说明:由于当 u 或 $v=0$ 时, $W_u=1$ 或 $W_v=1$,则在此认为 $h_u=h_v=2$)

二维 8 × 8 DCT 的 64 个基本图像的对称性情况见文献[15]附录 2。以 T_{64} 为例,它的 $u=6, v=4$, 可得 $h_u=1, h_v=2$, 则最小对称单元大小为 (2×1) , T_{64} 可由 A、B 组成的最小对称单元(图 1 虚框内)依次做行列对称延拓形成,如图 1 所示,其中:

$$A = \frac{1}{4} \cos \left\{ \begin{matrix} 3 \\ 8 \end{matrix} \right\} \cos \left\{ \begin{matrix} - \\ 4 \end{matrix} \right\}$$

$$B = \frac{1}{4} \cos \left\{ \begin{matrix} 9 \\ 8 \end{matrix} \right\} \cos \left\{ \begin{matrix} - \\ 4 \end{matrix} \right\}$$

根据性质 1 可知基本图像 T_{24} 具有与图 1 类似的对称性。

性质 4 T_{uv} 等于 T_{vu} 的转置,即 $T_{uv} = T_{vu}^T$ 。(根据基本图像定义式,很容易证明此性质。在此略。)

因为 $T_{uv} = T_{vu} \cdot q(u, v)$, 所以量化后 DCT 系数 $D(u, v)$ 对应的矩阵 T_{uv} 的对称性与 T_{vu} 完全一样。利用式(5)计算 DCT 时,可利用上述性质减少运算量和存储量。



图 1 第 10 类 T_{64} 的对称性情况

3 查找表算法

为消除乘法运算,本文采用查表法实现二维 8 × 8 IDCT 计算,本文算法的简单流程图如图 2 所示,它是采用式(5)计算 IDCT。算法分为两步:

(1) 根据 $D(u, v)$ ($u, v=0, 1, \dots, 7$) 的值,查询 LUT 表得 M_{uv} 矩阵。 M_{uv} 为乘积项 $D(u, v) \times T_{uv}$ 矩阵中的一

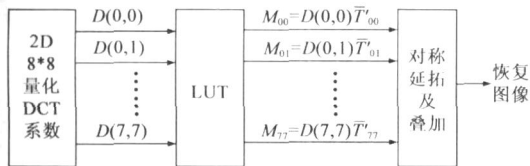


图 2 本文查找表算法简单流程图

部分,即:

$$M_{uv} = D(u, v) \times \bar{T}_{uv}$$

其中 \bar{T}_{uv} 为 T_{uv} 的最小对称单元部分, $D(u, v) \times T_{uv}$ 的其他部分则根据 M_{uv} 进行相应的对称延拓来得到。

(2) 根据对称性对 M_{uv} 做相应的延拓后,进行叠加运算得到恢复的原图像 f 。

下面将从查找表的结构设计、寻址方式和对称延拓及叠加运算三方面说明本文算法的具体过程。

3.1 查找表 LUT 的结构设计

首先,我们分析量化后各 DCT 系数的取值范围,以减少查找表 LUT 的长度,然后讨论如何设计查找表 LUT 的结构,以减少对 LUT 的寻址查找次数。

3.1.1 量化后各 DCT 系数的取值范围分析

由于基本图像的取值是固定的,而原图像 f 的每个象素的取值范围是 $0 \sim 255$, 所以可求得式(4)的最大值和最小值,即各 DCT 系数的取值范围。根据式(4)可知,当原图像 f 中对应 T_{uv} 矩阵所有正元素值位置处的像素值都为 255, 对应 T_{uv} 矩阵中负元素值位置处的像素值都为 0 时, DCT 系数 $F(u, v)$ 取最大值 $F_{\max}(u, v)$ 。例如,当 $u=3, v=2$ 时:

$$F_{\max}(3, 2) = f \cdot T_{32}$$

$$= \begin{bmatrix} 255 & 255 & 0 & 0 & 0 & 0 & 255 & 255 \\ 0 & 0 & 255 & 255 & 255 & 255 & 0 & 0 \\ 0 & 0 & 255 & 255 & 255 & 255 & 0 & 0 \\ 0 & 0 & 255 & 255 & 255 & 255 & 0 & 0 \\ 255 & 255 & 0 & 0 & 0 & 0 & 255 & 255 \\ 255 & 255 & 0 & 0 & 0 & 0 & 255 & 255 \\ 255 & 255 & 0 & 0 & 0 & 0 & 255 & 255 \\ 0 & 0 & 255 & 255 & 255 & 255 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0.19 & 0.08 & -0.08 & -0.19 & -0.19 & -0.08 & 0.08 & 0.19 \\ -0.05 & -0.02 & 0.02 & 0.05 & 0.05 & 0.02 & -0.02 & -0.05 \\ -0.23 & -0.09 & 0.09 & 0.23 & 0.23 & 0.09 & -0.09 & -0.23 \\ -0.13 & -0.05 & 0.05 & 0.13 & 0.13 & 0.05 & -0.05 & -0.13 \\ 0.13 & 0.05 & -0.05 & -0.13 & -0.13 & -0.05 & 0.05 & 0.13 \\ 0.23 & 0.09 & -0.09 & -0.23 & -0.23 & -0.09 & 0.09 & 0.23 \\ 0.05 & 0.02 & -0.02 & -0.05 & -0.05 & -0.02 & 0.02 & 0.05 \\ -0.19 & -0.08 & 0.08 & 0.19 & 0.19 & 0.08 & -0.08 & -0.19 \end{bmatrix}$$

$= 853.9$

根据基本图像的对称性可推得,除了 $F_{\min}(0, 0) = 0$, 其他 DCT 系数 $F(u, v)$ 最小取值 $F_{\min}(u, v) = -F_{\max}(u, v)$ 。因此,对于任意 8 × 8 图像子块,计算的 $F(3, 2)$ 值一定处于 $-853.9 \sim 853.9$ 范围内。考虑量化后,每个 DCT 系数需要除以其相应的量化步长然后取整数部分,这样,若量化步长 $q(3, 2) = 22$, 则量化后的 DCT 系数 $D(3, 2)$ 值一定处于 $-39 \sim 39$ 范围内。若采用 JPEG 标准缺省的亮度量化矩阵,量化后的所有 DCT 系数的最大绝对值矩阵为 D_{\max} 。

$$D_{\max} = \begin{bmatrix} 128 & 84 & 94 & 58 & 43 & 23 & 18 & 15 \\ 77 & 70 & 61 & 44 & 36 & 14 & 14 & 15 \\ 67 & 66 & 54 & 36 & 24 & 15 & 13 & 15 \\ 66 & 49 & 39 & 29 & 18 & 10 & 11 & 14 \\ 57 & 42 & 25 & 17 & 15 & 8 & 9 & 12 \\ 39 & 24 & 16 & 13 & 11 & 8 & 8 & 9 \\ 19 & 13 & 11 & 10 & 9 & 7 & 7 & 8 \\ 13 & 9 & 9 & 9 & 8 & 8 & 8 & 8 \end{bmatrix}$$

D_{\max} 中各元素值是相应位置的量化后 DCT 系数的最大绝对值, 即: 对于任意 8×8 图像子块, 计算的量化后的 DCT 值 $D(u, v)$ 一定处于 $-D_{\max}(u, v) \sim D_{\max}(u, v)$ 范围内.

3.1.2 查找表 LUT 的结构

若按式(5)进行 IDCT, 需计算乘积项矩阵 $D(u, v) \cdot T_{uv}$. 由各基本图像的对称性看出, 对 $D(u, v) \cdot T_{uv}$ 的计算可以先计算出相应的最小对称单元部分, 其他部分则根据对称性做相应的延拓来得到. 因为每个 T_{uv} 的各元素都为常数, 量化后的 DCT 系数 $D(u, v)$ 为整数且其绝对值的范围为 $0 \sim D_{\max}(u, v)$, 所以可将所有这些乘积值存入一个表 LUT 中. 当计算乘积项矩阵 $D(u, v) \cdot T_{uv}$ 时, 用查表操作代替乘法运算.

LUT 的构造如图 3 所示, LUT 由 $LUT_{00}, LUT_{01}, \dots, LUT_{77}$ 共 64 个子表块组成, 其中每个子表 LUT_{uv} 对应一个 DCT 量化系数 $D(u, v)$ ($u, v = 0, 1, \dots, 7$). 为了在查表时避免引入额外的运算操作, 提高查表速度, 每一个子表 LUT_{uv} 只存入 T_{uv} 最小对称单元中的各元素与所有整数值 $1 \sim D_{\max}(u, v)$ 的乘积值. 子表块的具体结构见图 3. $D(u, v)$ 系数对应的 T_{uv} 中最小对称单元为:

$$\begin{bmatrix} t_{uv}(0,0) & t_{uv}(0,1) & \dots & t_{uv}(0,M) \\ t_{uv}(1,0) & t_{uv}(1,1) & \dots & t_{uv}(1,M) \\ \dots & \dots & \ddots & \dots \\ t_{uv}(N,0) & t_{uv}(N,1) & \dots & t_{uv}(N,M) \end{bmatrix}$$

其中, $N = \frac{4}{2^{h_u}} - 1, M = \frac{4}{2^{h_v}} - 1$

如此构造 LUT 的优点在于: 当计算乘积项 $D(u, v) \cdot T_{uv}$ 时, 只需找到 $t_{uv}(0,0) \times D(u, v)$ 项的地址(地址的计算将在 3.2 节中讨论), 相应最小对称单元位置的其他数据项向下依次取出即可, 乘积项矩阵的其他部分则根据对称性做相应的延拓来得到.

综上所述, 每个子表 LUT_{uv} 包含 $(M+1) \times (N+1) \times D_{\max}(u, v)$ 个数据项, 可算得 LUT 中数据项的数目为:

$$\sum_{u=0}^7 \sum_{v=0}^7 (M+1) \times (N+1) \times D_{\max}(u, v) = 11238$$

10.9746K

而且每计算一个 DCT 系数的 $D(u, v) \cdot T_{uv}$ 只需查找 LUT 一次.

要说明的是: 如果量化矩阵具有对称性 $q(u, v) = q(v, u)$, 根据 2.2 节中的性质 4 可知: LUT_{uv} 和 LUT_{vu} 可共用一块子表, 所以查找表 LUT 中的数据项数目还可减少近半.

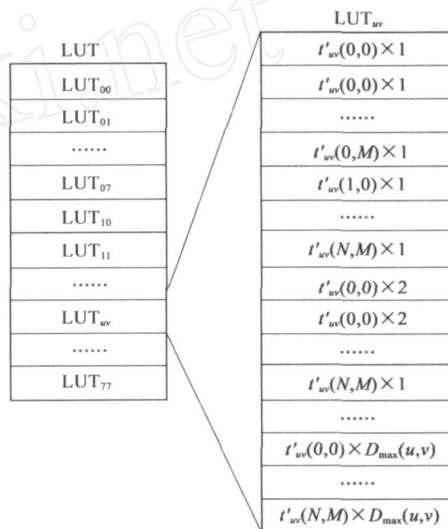


图 3 LUT 结构图

3.1.3 LUT 中数据项位数分析

从整个算法流程看, 本文算法是 DCT 的精确计算. 但由于在查找表中, 每个数据项的比特位数是有限的, 所以运算结果的精度会受到影响. 为了在保证精度要求情况下, 尽量减少 LUT 中数据项的字长, 本节将讨论 LUT 中数据项的最大取值情况, 以确定其数据项整数部分所需的位数. 其数据项的小数部分将根据实际的精度要求确定所需位数.

在查找表中, 每个 LUT_{uv} 中数据项值为: $t_{uv}(m, n) \times D(u, v)$, 则每个子表 LUT_{uv} 中所有数据项的最大绝对值为 $l_{\max}(u, v)$:

$$l_{\max}(u, v) = \max\{abs(t_{uv}(m, n)) \mid m, n = 0, 1, \dots, 7\} \times D_{\max}(u, v)$$

其中 $\max\{abs(t_{uv}(m, n)) \mid m, n = 0, 1, \dots, 7\}$ 为 T_{uv} 最小对称单元中元素的最大绝对值. 若采用 JPEG 标准缺省的亮度量化矩阵, 则每个 LUT_{uv} 中数据项的最大绝对值整数部分构成的矩阵 L_{\max} 为:

$$L_{\max} = \begin{bmatrix} 255 & 160 & 153 & 160 & 129 & 159 & 149 & 158 \\ 160 & 202 & 193 & 201 & 162 & 195 & 190 & 198 \\ 153 & 194 & 184 & 195 & 156 & 193 & 191 & 190 \\ 160 & 200 & 194 & 202 & 159 & 209 & 199 & 208 \\ 128 & 160 & 151 & 165 & 127 & 151 & 151 & 160 \\ 162 & 202 & 199 & 200 & 154 & 200 & 204 & 199 \\ 152 & 188 & 183 & 197 & 151 & 191 & 179 & 183 \\ 162 & 199 & 193 & 212 & 155 & 192 & 186 & 190 \end{bmatrix}$$

因此,在查找表中,每个数据项整数部分的位数设计为 9bit(包括 1 bit 符号位)即可,小数部分将根据实际精度的要求确定所需位数.

3.2 寻址方式

当计算乘积项 $D(u, v) \cdot T_{uv}$ 时,需要找到 $t_{uv}(0, 0) \times D(u, v)$ 项的地址,然后查表得到矩阵 $M_{uv} = D(u, v) \times \bar{T}_{uv}$. 具体寻址的实现方法为:事先将(LUT₀₀、LUT₀₁、...和 LUT₇₇子表块首地址-k)共 64 个地址放在 LUTaddr 表中,则 $t_{uv}(0, 0) \times D(u, v)$ 项的地址为: $LUTaddr(u, v) + |D(u, v)| \times k$ (k 为 T_{uv} 的最小对称单元中的元素数目). 由于 k 为 2^i ($i=0, 1, 2, 3$ 或 4), 所以 $D(u, v) \times k$ 的计算可用平移来实现. 这样,计算 $t_{uv}(0, 0) \times D(u, v)$ 项的地址只需整数加法和平移各一次.

为了消除寻址中的移位运算,可将上述方法进行改进:将所有 LUT_{uv} 子表块的 ($t_{uv}(0, 0) \times Val$ 项的地址值顺序放入 LUTaddr1 地址表中,其中 $Val \in [1, D_{max}(u, v)]$,再将 LUTaddr1 表中存有 LUT_{uv} 子表块首地址的单元地址减 1 后放入 LUTaddr0 地址表中. 这样,当计算 $t_{uv}(0, 0) \times D(u, v)$ 项的地址时,可通过两次查地址表来完成:首先在 LUTaddr0 地址表中得到一地址 $addr0$,令 $addr1 = addr0 + |D(u, v)|$,然后利用地址 $addr1$ 在 LUTaddr1 表中得到 $t_{uv}(0, 0) \times D(u, v)$ 项的地址 $addr$. 所以,每计算一个非零 DCT 系数的乘积项 $D(u, v) \cdot T_{uv}$ 只需整数加法一次. 因为 LUT 只有 10.9746K 个数据项,其中每个 LUT_{uv} 子表块需要有 $D_{max}(u, v)$ 个地址要存入 LUTaddr1 表中,有 1 个地址要存入 LUTaddr0 表中,所以 LUTaddr1 表长度为 1790(1.75K) 个单元,则每个存储单元的字长为 14bit, LUTaddr0 地址表共有 64 个存储单元,每个存储单元的字长为 11bit.

3.3 对称延拓及叠加处理

通过查表得到 M_{uv} 矩阵后,根据基本图像的对称性逐级对称延拓并进行累加运算,最后得到恢复图像 $f^{[15]}$. 为了减少加法运算量,可按图 4 进行对称延拓及累加运算. 在图 4 中,我们首先按照具有对称性相同的原理将乘积项分为 20 组,分别累加;接着又按照对称性相同或接近的原理把累加和矩阵按图示顺序逐级叠加,最后结果就是逆变换后的原图像矩阵. 图 4 中 $\begin{pmatrix} + \\ (m \times n) \end{pmatrix}$

符号意指:所有输入矩阵需按对称性延拓成 $(m \times n)$ 大小后再进行叠加. 例如: $(M_{02} + M_{06}) = [a_1 \quad b_1]$, 根据对称性延拓为 (2×2) 矩阵: $\begin{bmatrix} a_1 & b_1 \\ a_1 & b_1 \end{bmatrix}$, $(M_{42} + M_{46}) = [a_2 \quad b_2]$, 根据对称性延拓为 (2×2) 矩阵: $\begin{bmatrix} a_2 & b_2 \\ -a_2 & -b_2 \end{bmatrix}$, 则 $X_3 = \begin{bmatrix} a_1 & b_1 \\ a_1 & b_1 \end{bmatrix} + \begin{bmatrix} a_2 & b_2 \\ -a_2 & -b_2 \end{bmatrix}$. 在

累加过程中,一般是按照图中所示矩阵大小 $(m \times n)$ 将延拓后的矩阵中每个元素进行叠加,但在 $(M_{22} + M_{66})$ 或 $(M_{26} + M_{62})$ 的计算过程中,虽然矩阵大小都是 (2×2) ,但 $2 \times 2 = 4$ 个元素中只存在三个不同的绝对值,且位置分布一样,所以只需三次加法;同理计算 $(M_{11} + M_{33} + M_{55} + M_{77})$ 、 $(M_{31} + M_{57} + M_{15} + M_{73})$ 、 $(M_{35} + M_{17} + M_{53} + M_{71})$ 或 $(M_{13} + M_{37} + M_{51} + M_{75})$ 时只需 10 次加法. 根据基本图像对称性得:计算 X_1, X_2, \dots, X_9 分别需要加法:8、8、8、10、168、32、56、32、56 次;进一步计算 W_1, W_2, \dots, W_7 分别又需要加法:8、8、16、32、16、16、32 次;最后一步求得 f 需要加法 64 次. 所以,所有乘积项 M_{uv} 对称延拓及累加共需加法 570 次.

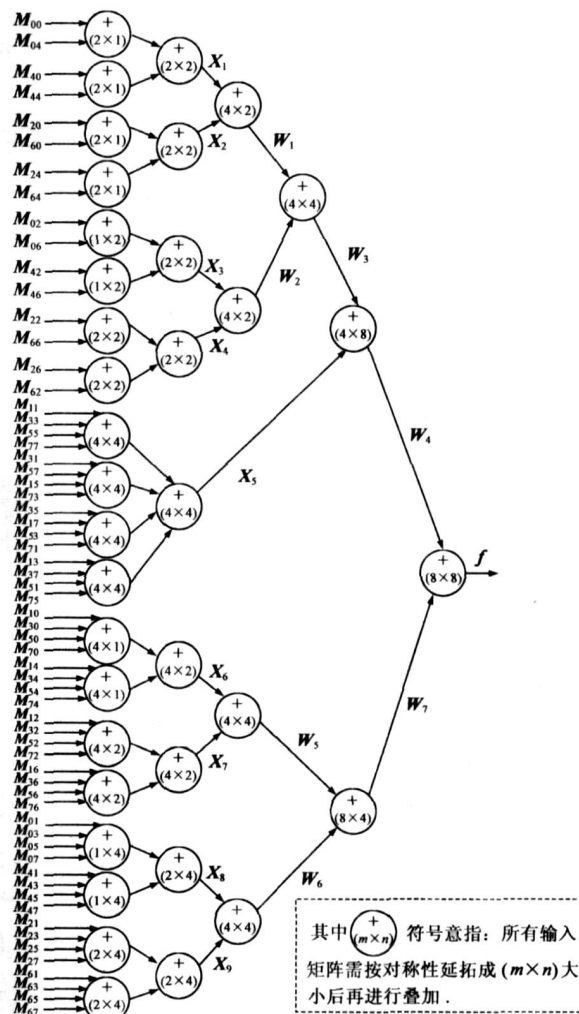


图 4 对称延拓及叠加处理过程

实际图像解码计算时,在 8×8 量化后 DCT 系数矩阵中,绝大多数元素 $D(u, v)$ 为零值. 可利用此特点忽略许多 $D(u, v) \cdot T_{uv}$ 的计算及相应的叠加运算.

综合以上讨论,利用查表法计算完整的二维 8×8 IDCT,只需要 $570 + 64 = 634$ 次加法. 在实际图像解码计算中,只让非零的 DCT 系数参与计算,这样可大大减少

运算量. 另外, 从图 2 和图 4 直观看出, 本算法的计算过程具有很强的并行性.

4 实验及结果比较

本节用几个标准图像(见表 2)对新算法与 Feig 算法、文献[15]算法(简称算法 15)进行比较. 标准图像中包含灰度均衡的 Lena 和 Peppers 等图像, 以及细节变化大的 Baboon 图像. 对 Lena 等 5 幅(512 × 512)灰度图以及 Baboon(256 × 256)彩色图像的 Y 图像采用了 JPEG 标准缺省的亮度量化表进行量化. 从表中数据可看出: 标准图像中平均每 8 × 8 子块量化后的非零值个数情况及各算法采用软件“短路”技术后应用到标准图像上的运算量情况. 对比两算法的数据统计结果表明了本文算法在实际应用中的有效性.



图 5 标准图像

表 2 三算法应用到标准图像中每 8 × 8 子块所需平均运算量比较

标准图像	平均非零值个数	Feig Scaled IDCT		算法 15		新算法
		平均乘法	平均加法	平均乘法	平均加法	
Peppers	7.06	27.18	175	9.61	147	149
Elaine	7.39	30.63	197	8.94	161	163
Airplane	7.64	28.02	169	11.21	143	144
Women1	8.64	33.15	201	11.35	169	171
Lena	9.36	34.93	206	9.19	175	178
baboon	19.64	62.30	314	26.14	290	287

对表 2 中 Feig 算法与新算法的平均运算量进行比较, 结果表明: Feig 算法的平均乘法次数为 36.035, 平均加法次数为 210.33; 本文算法在消除了乘法运算的同时, 平均加法次数为 182, 加法次数降低了约 15%. 另外, 在编程过程中发现: 当采用软件“短路”技术时, 新算法仅需判断各 DCT 量化系数是否为非零值, 而 Feig 算法在整个计算过程中判断是否为 0 值的次数比新算法要多得多. 对表 2 中新算法与算法 15 相比: 消除了所有乘法, 加法次数基本一样. 所以运算速度比算法 15 有大幅度的提高. 由于已有的查表算法与表 1 中 Feig 等一般快速算法相比, 虽然消除了乘法运算, 但增加了更多的加法运算, 所以也就说明了本文算法比已有查表算法的加法次数少很多.

新算法使用查找表消除了乘法运算, 同时查找所需时间很小. 查找表是以每个基本图像的最小对称单元为一组存放数据, 所以每计算一个 DCT 量化系数对应的 $D(u, v) \cdot T_w$ 只需查找 LUT 一次. 这样, 计算一个 2D 8 × 8 子块的 IDCT 的查找次数等于该子块的非零系数个数(见表 2). 对表 2 中 6 幅标准图像来讲, 平均查找次数小于 10 次. 每次查找所需的 1 次整数加法也已记入总运算量中, 所以查找时间很小.

5 总结

本文利用查表法并基于基本图像的概念, 提出了一种准确计算二维 8 × 8 IDCT 的算法, 旨在消除乘法运算的同时, 加法运算量也比已有的快速算法的加法运算量小, 而且通过对量化后 DCT 系数的分析求得其取值范围, 大大减少了查找表的长度. 本文算法与当前最有影响的快速算法相比, 在消除了乘法运算的同时, 加法运算量也比这些快速算法小; 和一般查表法相比: (1) 平均加法运算量很小; (2) 查找表占用的存储量较小, 查表次数很少; (3) 计算过程具有很强的并行性. 在 VLSI 集成电路设计中, 此算法不仅有利于芯片面积和功耗的减小, 而且运算速度也大大提高. 对于那些没有特殊硬件支持的可移动终端, 若采用此算法进行终端间的图像、视频实时传输时, 在处理速度及存储容量方面可获得更高的性能. 在实际应用中, 对传输质量要求不高的情形, 本算法的运算量比实验结果更少. 因此新算法为提高图像压缩传输的效率提供了新的思路.

参考文献:

- [1] Feig E, Winograd S. Fast algorithms for the discrete cosine transform[J]. IEEE Transactions on Signal Processing, 1992, 40(9): 2174 - 2193.
- [2] Leoffler C, Ligtenberg A, Moschytz G S. Practical fast ID DCT algorithms with 11 multiplications[A]. In: Proceedings of IEEE International Conference Acoustics, Speech and Signal Processing (ICASSP) [C]. Glasgow, 1989, 2: 988 - 991.
- [3] Lee B G. A new algorithm to compute the discrete cosine transform[J]. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1984, 32(6): 1243 - 1245.
- [4] Chen W H, et al. A fast computational algorithm for the discrete cosine transform[J]. IEEE Transactions on Communications, 1977, 25(9): 1004 - 1009.
- [5] Huang Yuh-Ming, Wu Ja-Ling. A refined fast 2-D discrete cosine transform algorithm[J]. IEEE Transactions on Signal Processing, 1999, 47(3): 904 - 907.
- [6] Trac D Tran. The binDCT: fast multiplierless approximation of the DCT[J]. IEEE Signal Processing Letters, 2000, 7(6): 141 - 144.

- [7] Liang Jie. Fast multiplierless approximations of the DCT with the lifting scheme[J]. IEEE Transactions on Signal Processing, 2001, 49(12): 3032 - 3044.
- [8] Cui Xiang Biao, Johnson Martin. Evolution of the discrete cosine transform using genetic programming[J]. Research Letters in the Information and Mathematical Sciences, 2002, 3(1): 117 - 125.
- [9] Henning R, Chakrabarti C. A quality/energy tradeoff approach for IDCT computation in MPEG-2 video decoding[A]. In Proc IEEE Signal Processing Systems (SiPS '2000) [C]. Lafayette, 2000. 90 - 99.
- [10] 黄贤武, 王加俊, 李家华. 数字图像处理与压缩编码技术[M]. 成都: 电子科技大学出版社, 2000. 298 - 305.
- [11] Kutka, R. Fast computation of DCT by statistic adapted look-up tables [A]. In Proc IEEE ICME '2002 [C]. Lausanne, Switzerland, 2002. 781 - 784.
- [12] Uramoto S, Inoue Y, Takeda J, Takabatake A, Terane H, Yoshimoto M. A 100MHz 2-D discrete cosine transform core processor[J]. IEEE Journal of Solid-State Circuits, 1992, 27(4): 492 - 499.
- [13] M Matsu et al. A 200 MHz 13mm² 2-D DCT macrocell using sense-amplifying pipeline flip-flop scheme[J]. IEEE Journal of Solid-State Circuits, 1994, 29(12): 1482 - 1490.
- [14] Yu Sungwook, Earl E Swartlander. DCT implementation with distributed arithmetic [J]. IEEE Transactions on Computers, 2001, 50(9): 985 - 991.
- [15] 纪秀花, 张彩明. 适于图像解压的离散余弦逆变换快速算法[J]. 计算机学报, 2005, 28(12): 2079 - 2088.
Ji Xiuhua, Zhang Caiming. A fast IDCT algorithm for image decompression [J]. Chinese journal of Computer, 2005, 28(12): 2079 - 2088. (in Chinese)

作者简介:



纪秀花 女, 1964年10月生于湖北孝感, 1985年、1988年在山东大学电子系获理学学士、理学硕士学位, 现为山东大学博士研究生, 山东经济学院教授. 主要从事计算机图像处理、视频压缩等方面的研究工作.
E-mail: jane@sdu.edu.cn



张彩明 男, 1955年11月生于山东威海, 1982年、1984年在山东大学获工学学士、工学硕士学位, 1994年在日本东京工业大学大学院获工学博士学位, 1998年7月至1999年12月在美国 Kentucky 大学做博士后. 现为山东大学教授, 博士生导师, 山东经济学院计算机科学与技术学院院长. 主要从事 CAD、CG、科学计算可视化和医学图像处理等方面的研究工作.