文章编号:1001 - 9081(2005)04 - 0772 - 03

适用于嵌入式系统的二维 DCT 算法

魏本杰[1,2],刘明业[2],章晓莉[1]
(1. 北京电子科技学院 计算机系,北京 100070; 2. 北京理工大学 计算机系,北京 100081)
(wei9100@besti.edu.cn)

摘 要:提出一种简单、快速的二维 DCT 算法,即将二维 DCT 的变换系数和图像的像素值事先乘好,形成查找表,在实现直接二维 DCT 变换时,把表调入内存,乘积的结果就能通过寻址的方式得到而无需乘法器,速度很快,非常适合嵌入式系统的设计。在设计查找表时,优化了其结构,减小了存储空间。实验结果表明了算法的有效性。

关键词:二维 DCT 算法;变换系数;查找表;地址映射
中图分类号: TP391.41 **文献标识码**:A

2-D DCT algorithm for embedded system

WEI Ben-jie[1,2], LIU Ming-ye[2], ZHANG Xiao-li[1]
(1. *Department of Computer Science, Institute of Beijing Electronic Science and Technology, Beijing* 100070, *China;*
2. *Department of Computer Science, Beijing Institute of Technology, Beijing* 100081, *China*)

Abstract: A simple and fast 2-D DCT algorithm was presented. Firstly the transform coefficient of 2-D DCT was multiplied by the value of pixels, which formed a lookup table. When realizing the direct 2-D DCT, it put the table into memory, and got the product value by addressing without multipliers. This algorithm can work with high speed, so it is specially suitable for the design of embedding system. The structure was optimized and the memory space was reduced. Experiment results show that the algorithm is effective.

Key words: 2-D DCT algorithm; transform coefficients; lookup table; address mapping

0 引言

传统的二维 DCT(离散余弦变换)算法在实现时,一般采用以下两种办法:1)利用二维 DCT 算法具有可分离的特点,将其分成两个一维 DCT 来实现;2)直接利用二维 DCT 的计算公式来实现。其他方式都是在这两种方式的基础上产生的,本文针对嵌入式系统的特点,即在浮点数乘法器资源短缺的情况下,使用尽可能少的乘法,甚至不需要乘法就可以实现二维 DCT 快速计算。在算法设计中我们采取了空间换取时间的策略,因为现在嵌入式系统的内存可达到 128M 甚至更多且成本越来越便宜,足以满足需要。算法采用查表的方式,计算过程中所需结果都提前算好并置入内存中,只有必要的系数加法运算,查表法的运算时间与变换对象的个数成正比,快于其他的方法[4]。

1 算法描述

1.1 二维 DCT 算法[3]

对于一个大小为 $N \times N$ 的图像块来说,图像样值 $x(i,j)(i,j = 0,1,\cdots,N-1)$ 的二维 DCT 定义为:

$$y(K,L) = \frac{2c(K)c(L)}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}x(i,j) \times \cos\frac{(2i+1)K\pi}{2N}\cos\frac{(2j+1)L\pi}{2N} \quad (1)$$

其中:

$u,v = 0,1,2,\cdots,N-1$

$$c(K),c(L) = \begin{cases}\sqrt{\frac{1}{2}} & u,v = 0\\ 1 & u,v = 1,2,3,\cdots,N-1\end{cases}$$

按照通常的 DCT 变换,本文取 $N = 8$,从以上公式可以看出,每个变换后的数据由 64 项组成和式,每一项中由三部分组成:第一部分是图像数据 $x(i,j)$,取值范围在[0,255];第二、三部分都是 DCT 系数,可将其视为整体。式(1)中变化的是 $x(i,j)$,DCT 系数对每次变换来说是常量。一个 0 ~ 255 之间的数和一个常数相乘,有 256 种可能,若将可能值作成一张表,就可以利用查表法避免乘法运算。

查找表的数据个数有:$64 \times 64 \times 256 = 1M$ 个,若每个数据使用 16 位的浮点数表示,则需要总的表格存储器的容量为 2M Byte。

1.2 二维 DCT 算法系数表的建立

由于系数表占有较大的内存,需要利用算法的特点降低系数表中的数据,由公式(1)可知,计算 $Y(K,L)$ 的 64 项和式中,每一项的第二部分为:

$$\cos\frac{(2i+1)K\pi}{2N}\cos\frac{(2j+1)L\pi}{2N} \quad (2)$$

用 $C0$ 表示 $\cos(0)$,$C1$ 表示 $\cos\frac{\pi}{16}$,$C2$ 表示 $\cos\frac{2\pi}{16}$,$\cdots$,$C15$ 表示 $\cos\frac{15\pi}{16}$,则可知计算 $Y(K,L)$ 时需计算的式(2)的系数值如下:

(1) 若 $K = 0,L = 0$,上式为 $C0 \times C0 = 1$;

(2) 若 $K = 0,L = 1$ 时,则需计算 8 个系数:$C1,C3,C5,C7,C9,$

收稿日期:2004 - 09 - 26;**修订日期**:2004 - 11 - 24

作者简介:魏本杰(1973 -),男,讲师,博士研究生,主要研究方向:多媒体信号处理、嵌入式、ASIC 设计; 刘明业(1934 -),男,辽宁营口人,教授,主要研究方向:专用集成电路设计、数字系统自动综合理论; 章晓莉(1964 -),女,江西抚州人,副教授,主要研究方向:数字图像处理.

$C11, C13, C15$;

…

(5) $K = 0, L = 7$ 时,则需计算 8 个系数:$C7, C11, C3, C15, C1, C13, C5, C9$;

(6) 若 $K = 1, L = 0$ 时,则需计算 8 个系数:$C1, C3, C5, C7, C9, C11, C13, C15$;

(7) 若 $K = 1, L = 1$ 时,则需计算 64 个系数:

$C1 \times C1$, $C3 \times C1$, $C5 \times C1$, $C7 \times C1$, $C9 \times C1$, $C11 \times C1$, $C13 \times C1$, $C15 \times C1$;

…

$C1 \times C15, C3 \times C15, C5 \times C15, C7 \times C15, C9 \times C15, C11 \times C15, C13 \times C15, C15 \times C15$;

按照这样的规律,逐个计算从 $K = 2, L = 0$ 直到 $K = 6, L = 7$,最后:

(64) 若 $K = 7, L = 7$ 时,则需计算 64 个系数:

$C7 \times C7, C11 \times C7, C5 \times C7, C7 \times C7, C9 \times C7, C11 \times C7, C13 \times C7, C15 \times C7$;

…

$C7 \times C15, C11 \times C15, C5 \times C15, C7 \times C15, C9 \times C15, C11 \times C15, C13 \times C15, C15 \times C15$;

列出表格如表 1 所示。

表 1　二维 DCT 的算法系数表

	C0	C1	…	C7	C8	C9	…	C15
C0	C0*C0	C0*C1	…	C0*C7	0	C0*C9	…	C0*CC15
C1	C1*C0	C1*C1	…	C1*C7	0	C1*C9	…	C1*C15
…		…			0	…		
C6	C6*C0	C6*C1	…	C6*C7	0	C6*C9	…	C6*C15
C7	C7*C0	C7*C1	…	C7*C7	0	C7*C9	…	C7*C15
C8	0	0	…	0	0	0	…	0
C9	C9*C0	C9*C1	…	C9*C7	0	C9*C9	…	C9*C15
C10	C10*C0	C10*C1	…	C10*C7	0	C10*C9	…	C10*C15
…		…			0	…		
C14	C14*C0	C14*C1	…	C14*C7	0	C14*C9	…	C14*C15
C15	C15*C0	C15*C1	…	C15*C7	0	C15*C9	…	C15*C15

从表 1 可以看出,表中的值具有重复性和对称性,深色阴影将表的下方矩阵分成四块(如表中四块浅色阴影部分所示),它们刚好呈对角对称,而左上方和右下方的浅色阴影数据块内部也对称。如能去掉矩阵中重复值,那么就可以大量的缩小表的数据容量。在设计系数表时,首先求出 Ci, Cj 的值,用数组 $CosValue[16][16]$ 表示,然后,通过实际计算可去掉 184 个多余系数值,生成 72 个有用的系数值,系数表的大小变为 $72 \times 256 \times 4 = 72$KB,用这 72 个三角系数再乘上常数 $C(K)$、$C(L)$,即建立一个供查询的系数表。

还需以下两个步骤:1) 对每个 DCT 系数,乘上常数 1/4 及图像数据的可能值 0 ~ 255,至于 $C(K) * C(L)$ 添加工作可放在 DCT 变换中完成;2) 建立数据文件,存放 DCT 系数表,系数表的大小是 72KB,系数表的结构定义为:设有用系数各自以表 1 中的先后顺序进行编排,每个系数分别乘上 0 ~ 255 设为一组,共有 72 组 72×256 个。

1.3　建立地址映射表

系数表建立后,要确立查找系数的地址映射方法。在计算 $Y(K,L)$ 时,其和式共有 64 项,当 K, L 确定后,和式中的余弦项也就确定了。换言之,$x(i,j)$ 和哪个系数相乘是由 K, L 来进行确定的,利用它们之间的这种联系可建立一种映射关系,从而形成一个系数的地址映射表(如使用数组 $FpAddr[8][8][64]$ 表示),并利用该表在 DCT 变换中来查询相应的系数。

数组 $FpAddr$ 的结构如表 2 所示。

表 2　系数的地址映射表

$K = 0, L = 0$	0	1	…	63
$K = 0, L = 1$	0	1	…	63
…				
$K = 7, L = 7$	0	1	…	63

表 2 中共有 $64 \times 64 = 4$k 项,每一项为整数类型,占 2B,总表长一共为 8KB。每 64 项对应一组 (K,L) 值,从 0 ~ 63 顺序编号,第 0 项的内容为 $x(0,0)$ 要乘的三角系数在系数表中所属一组的起始位置,第 1 项的内容为 $x(0,1)$ 要乘的三角系数在系数表中所属一组的起始位置 …… 第 63 项的内容为 $x(7,7)$ 要乘的三角系数在系数表中所属一组的起始位置。数组 $FpAddr[K][L][64]$ 中记录的是地址映射结果,写入存储区中,就可以建立起查表法所需的数据。

3　查表法实现 DCT 变换

以一幅图像变换测试中普遍采用的 256 × 256 灰度图像 Lena 为例进行运算,如图 1 所示。



图 1　原始图像

在进行二维 DCT 变换时,当计算变换值 $Y(K,L)$ 时,首先读入系数表($FpCoes[72][256]$)和地址映射表($FpAddr[8][8][64]$),然后由 K, L 值确定 $x(i,j)$ 对应的系数在数组 $FpAddr[8][8][64]$ 中的位置,依次取出后面的 64 个值,根据这些值找出该系数在数组 $FpCoes[72][256]$ 中的起始位置,加上 $x(i,j)$ 值,将这些取出的数依次相加,就完成了 DCT 变换中某一系数的计算。上述计算过程可用下面的式子(3)表示:

$$\begin{aligned} Y(K,L) = {} & FpCoes[(FpAddr[K][L][0])][x(0,0)] + \\ & FpCoes[(FpAddr[K][L][1])][x(0,1)] + \\ & FpCoes[(FpAddr[K][L][2])][x(0,2)] + \\ & \cdots + \\ & FpCoes[(FpAddr[K][L][63])][x(7,7)] \end{aligned} \tag{3}$$

具体实现过程中,用数组运算(在嵌入式系统中对应为存储区),即数组 $FpAddr[8][8][64]$ 存放地址映射表,$FpCoes[72][256]$ 对应于某个图像块的像素值与数组 $CosValue[16][16]$ 中相应元素的乘积值。其中,数组 $FpImage[32][32][64]$ 代表一幅 256 × 256 的 Lena 图像的 1024 个数据块(8 × 8)值,DCT 变换后的最终结果存放在数组 $FpFdct[32][32][64]$ 中。

3　实验结果及分析

3.1　查表法计算二维 DCT 的结果

对设计好的算法采用 C 代码进行描述,在测试时使用查表法对图 1 进行了二维的 DCT 变换,共划分为个 1024 个 8 ×

8 数据块进行计算，以验证结果的正确性及执行的速度。由于篇幅有限，此处只给出其中的 1 个 8×8 矩阵块（如表 3 所示）的处理结果（包括传统方式[5]计算二维 DCT 的结果）。

表 3 原始的图像块(8×8)

133	132	130	129	129	128	129	129
133	132	130	129	129	128	129	129
130	128	128	128	129	128	129	129
128	127	127	127	126	126	127	129
129	128	127	126	126	127	127	128
129	129	128	127	127	127	128	128
130	130	129	127	127	127	128	132
128	128	127	127	127	128	131	136

表 4 两种不同实现方式的结果

传统方式实现的标准结果							
1028.38	1.57	7.68	-1.42	1.62	-0.8	-0.07	0.44
3.09	5.48	-2.37	2.56	-0.17	1	-0.79	0.67
7.03	-0.57	2.2	-1.02	0.39	-0.81	-0.55	0.05
-0.22	4.38	-0.62	0.93	-1.39	-0.18	-0.25	-0.03
-1.88	-2.2	1.05	-0.71	0.37	0.1	0.24	-0.67
-0.81	0.37	-0.27	-0.89	0.46	-0.16	0.36	0.12
-0.34	-1.86	-1.05	0.79	-0.03	0.61	-0.45	0.36
0.41	-0.04	0.13	0.76	0.06	-0.29	0.48	0.25
2D-DCT 的查表法结果							
1028.38	1.57	7.68	-1.42	1.62	-0.8	-0.07	0.44
3.09	5.48	-2.37	2.56	-0.17	1	-0.79	0.67
7.03	-0.57	2.2	-1.02	0.39	-0.81	-0.55	0.05
-0.22	4.38	-0.62	0.93	-1.39	-0.18	-0.25	-0.03
-1.88	-2.2	1.05	-0.71	0.37	0.1	0.24	-0.67
-0.81	0.37	-0.27	-0.89	0.46	-0.16	0.36	0.12
-0.34	-1.86	-1.05	0.79	-0.03	0.61	-0.45	0.36
0.41	-0.04	0.13	0.76	0.06	-0.29	0.48	0.25

输入如表3的原始数据（Lena图像数据中的一个8×8图像块），输出结果如表 4 所示。

从以上的运算结果可以看出，采用查表法实现的结果和常规方式完全一致，这是因为我们设计的数据表本身就是通过常规方法事先生成的，故应该和原标准结果相同。

3.2 查表法二维 DCT 的执行速度

在计算运算量的基础上，将本算法所需乘法和加法运算次数与常规二维 DCT 算法了进行比较：

常规算法：385 次乘法；257 次加法；

查表法：无乘法；65 次加法；

实际测试表明，查表法计算 DCT 的速度比传统方式快一倍左右。

4 结语

本文提出了适合于嵌入式系统的二维 DCT 的查表实现算法。首先建立一个二维 DCT 算法的系数表，对其进行了简化，去除重复值；接着对系数表进行了地址映射，形成地址映射表；最后给出了比较的结果。实验表明本文给出的查表法实现的二维 DCT 算法具有速度快，结果准确，无需乘法器，便于 VLSI 设计和硬件实现等优点，因此适合应用在快速视频编码处理的场合。

参考文献：

[1] ISO - IEC / JTC 1 / SC 2 / WG 1 0 Joint Photographic Experts Group, CD10918-1. Digital compression and coding of continuous-tone still images[Z]. 1991.

[2] AHMED N, NATARAYAN T, RAO KR. Discrete cosine transform[J]. IEEE Transactions Computer, 1974, C-23(1): 90 - 93.

[3] 马小虎，张明敏，严华明. 多媒体数据压缩标准及实现[M]. 北京：清华大学出版社，1996.

[4] 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.

[5] 何斌，马天宇，王运坚，等. Visual C++ 数字图像处理[M]. 北京：人民邮电出版社，2001.

（上接第 771 页）

声图像去噪，不再一一赘述，这都充分说明了 H-Curve 准则应用的广泛性，且和现在大多数方法去噪图像相比能产生较好的视觉性。

4 结语

本文采用了二树复数小波变换，在基于 H-Curve 准则选取更合理阈值的基础上进行图像去噪，通过典型的实验可以看出：本文采用的方法在去噪能力，取得的视觉效果和确定阈值的广泛性方面都优于目前多数方法。

虽然 H-Curve 准则在理论上有些方面还存在着一些缺点[7]，但它在实际中已经表现出应用的广泛性和实用性。复数小波比实数小波在平移不变性、更好的方向性和精确的相空间信息等方面具有更多的优点，故它在一些方面的应用如视频消噪[5]中是优于实数小波的。不同类型的小波变换在解决具体问题时还要灵活运用，另外对于阈值的确定很有技巧性，且是去噪中的一个重点，还有待进一步深入研究。

参考文献：

[1] 谢杰成，张大力，徐文立. 小波图像去噪综述[J]. 中国图象图形学报，2002，7(3)：209 - 217.

[2] SHENSA MJ. Discrete wavelet transform: wedding the A Trous and Mallat algorithm[J]. IEEE Transactions on Signal Processing, 1992, 40: 2464 - 82.

[3] KINGSBURY NG. The Dual - tree Complex Wavelet Transform: A New Technique for Shift Invariance and Directional Filters[A]. IEEE Digital Signal Processing Workshop, DSP 98[C]. Bryce Canyon, 1998. 86 - 89.

[4] KINGSBURY NG. Complex wavelets for shift invariant analysis and filtering of signals[J]. Applied and Computational Harmonic Analysis, 2002, 10(3): 234 - 253.

[5] SELESNICK IW, LI KY. Video denosing using 2D and 3D dual-tree complex wavelet transforms[EB/OL]. http://telin.rug.ac.be/~sanja/Papers/VideoDenSPIE03.pdf, 2004.

[6] SELESNICK IW. Hilbert transform pairs of wavelet bases[J]. IEEE Transactions on Signal Processing, 2001, 8(6): 170 - 173.

[7] MONTEFUSCO LB, PAP S. A Parameter Selection method for wavelet shrinkage denoising BIT Numerical Nathematics[J]. Bit Numerical Mathematics, 2003, 43(3): 611 - 626.

[8] CHAMBOLLE A, DEVORE RA, LEE N, *et al*. Nonlinear wavelet image processing: Variational problem, compression, and noise removal through wavelet shrinkage[A]. IEEE Transactions on Image Process[C]. 1998. 319 - 335.

[9] VOGEL CR, OMAN ME. Iterative methods for total variation denoising[J]. SLAM Journal of Scientific Computing, 1996, 17(1): 227 - 238.