

文章编号 1004-924X(2013)03-0799-08

# 基于多维矢量矩阵的离散余弦变换快速算法

桑爱军, 吴 杨, 崔海廷\*, 陈贺新

(吉林大学 通信工程学院, 吉林 长春 130022)

**摘要:** 为了提高彩色视频流压缩变换的速度, 缩短变换时间, 将基于多维矢量离散余弦变换(DCT)正交矩阵的视频流压缩算法同现有的快速 DCT 算法相结合, 提出了基于多维矢量矩阵 DCT 的快速算法。首先, 对一种实用性较强的一维快速 DCT 算法进行部分校正与补充, 并将两种极具代表性的二维快速 DCT 算法, 即行列分解法和线性组合法拓展到了多维层面。然后, 结合多维矢量矩阵理论, 推导出了两种 DCT 快速算法及其蝶形示意图, 并分析了算法的乘法运算次数。最后, 将这两种基于多维矢量矩阵的 DCT 快速算法应用在标准视频库中测试视频的压缩编码中, 同时做了对比实验。实验结果表明: 两种算法在 DCT 时所需的时间均小于 0.25 s, 满足视频实时性的要求。该算法在保证重建视频质量的同时降低了运算时间, 并将 DCT 快速算法从低维扩展到了多维, 可以处理更多维数的视频。

**关键词:** 彩色视频压缩编码; 多维矢量矩阵; 快速离散余弦变换; 正交变换

**中图分类号:** TN919.81 **文献标识码:** A **doi:** 10.3788/OPE.20132103.0799

## Fast DCT algorithms based on multi-dimensional vector matrix

SANG Ai-jun, WU Yang, CUI Hai-ting\*, CHEN He-xin

(College of Communication Engineering, Jilin University, Changchun 130022, China)

\* Corresponding author, E-mail: cui\_hai\_ting@126.com

**Abstract:** With the aim to improve the speed and reduce the time of color video stream compression transform, this paper proposed a fast Discrete Cosine Transform (DCT) algorithm based on multi-dimensional vector matrix. The algorithm combined a fast DCT based on multi-dimensional vector orthogonal matrix transform algorithm with a existing fast DCT algorithm for the first time. Firstly, it corrected and supplemented the fast 1D-DCT algorithm with strong practicality, and extended two most representative 2D-DCT fast algorithms, the rank decomposition method and the linear combination method, to the multi-dimensional field. Then, two DCT fast algorithms and their butterfly diagrams were deduced by combining multi-dimensional vector matrix theory, and the times of multiplications were analyzed. Finally, two methods were applied to test of the video in the standard video library meanwhile comparative experiments were performed. The experimental results show that the required time of the two methods is both less 0.25 s in DCT and these meet the real-time requirements of video. The proposed two methods ensure the reconstructed video quality, and not only reduce the computation time significantly, but also expand the DCT fast algorithm from a low-dimension to the multi-dimension.

**Key words:** color video compression coding; multi-dimensional vector matrix; fast Discrete Cosine Transform(DCT); orthogonal transform

## 1 引言

随着数字技术和网络技术的飞速发展,视频图像作为多媒体信息的重要载体,已经成为当前信息科学研究的热点,其中视频图像压缩<sup>[1]</sup>技术由于可解决巨大数据量的存储和传输问题而更为引人注目。变换编码能对均匀分布的视频图像数据进行适当的变换,把大部分能量只集中到少数几个变换系数上,极大降低了视频图像的相关性,是视频图像压缩编码<sup>[2]</sup>领域中的一项关键技术。K-L 变换是均方误差准则下的最优变换,但实现起来非常困难,因此,在现行视频编码标准中多采用离散余弦变换算法(DCT)。这是由于 DCT 与视频图像的统计特性相吻合,在压缩性能和误差上与最佳变换 K-L 变换相当,计算复杂度适中,又具有可分离特性、算法快速等特点。

为了减少 DCT 所需的时间,提高视频编码的效率,近年来国内外学者对快速离散余弦变换算法进行了大量的研究。Wu 等<sup>[3]</sup>提出了一种二维快速 DCT 算法,该算法可以通过一个简单的程序设计应用到逻辑图上。吴红文等<sup>[4]</sup>提出了一种新的二维 DCT 快速算法,即利用三角函数公式将  $N \times N$  的二维 DCT 分解成  $N$  个一维 DCT 进行运算,给出了  $8 \times 8$  的算法流程图,并与其他二维 DCT 快速算法在计算量和计算结构上进行比较,具有明显的计算精度。罗小明等<sup>[5]</sup>介绍了一种用一维 DCT 线性组合计算二维 DCT 的快速算法,将原始数据预分组时改模  $N$  为模  $2N$ ,实现了一次性正确分组,推导出计算线性组合的具体公式并讨论了它的计算速度。Zheng 等<sup>[6]</sup>提出了一种基于 JPEG 的快速 DCT 算法,用移位和加法代替传统 JPEG 编码中的乘法,降低了变换的复杂性。Yu 和 Sun 等<sup>[7-8]</sup>对基于并行运算的快速一维及二维 DCT 算法作了研究,显著提高了运算速度。然而大量算法的提出和改进并没有很好地去除视频帧之间的冗余,因此必须寻找一种更为有效的快速算法。

多维矢量矩阵理论<sup>[9]</sup>可以把多维数据统一在一个模型中,在消除视频空间及帧间冗余的同时还能进一步提高压缩比与峰值信噪比。因此,本文将多维矢量矩阵理论同现有的快速 DCT 算法

相结合,提出了基于多维矢量矩阵的 DCT 快速算法,在保证视频流实时性的前提下,将 DCT 的快速算法由低维拓展到多维。

## 2 多维矢量矩阵及其 DCT 操作算子

### 2.1 多维矢量矩阵的定义

定义 1:  $F$  上的  $M \times N$  数据排列  $a_{i_1 i_2}$  称为二维矩阵,其全体组成的集合记为  $M_{M \times N}$ ,  $F$  上的  $I_1 \times I_2 \times I_n$  多维数据排列表  $a_{i_1 i_2 \dots i_n}$  称为多维矩阵,其全体组成的集合记为  $M_{I_1 \times I_2 \times \dots \times I_n}$ 。

定义 2: 如果将多维矩阵的维数分成两组,分别用 2 个矢量表示,比如  $M_{K_1 \times K_2 \times \dots \times K_r}$  表示成  $M_{(I_1 \times I_2 \times \dots \times I_m) \times (J_1 \times J_2 \times \dots \times J_n)}$ , 记为  $M_{IJ}$ , 其中  $I, J$  为矢量,  $I = (I_1, I_2, \dots, I_m)$ ,  $J = (J_1, J_2, \dots, J_n)$ , 则称多维矩阵  $M$  为维数按照矢量  $I, J$  划分的多维矢量矩阵,简称多维矢量矩阵。显然一个多维矩阵可以表示成很多种的多维矢量矩阵,而一个多维矢量矩阵只对应唯一的多维矩阵。

### 2.2 4D 矢量矩阵 DCT 操作算子

以 4D 矢量矩阵为例。定义 4D 矢量矩阵正交变换、正交逆变换为:

$$F_U = C_U f_U C_U^T, \quad (1)$$

$$f_U = C_U^T F_U C_U, \quad (2)$$

其中:  $f_U$  是 4D 矢量待变换矩阵,  $F_U$  是 4D 矢量变换系数矩阵,  $C_U$  和  $C_U^T$  是 4D 矢量矩阵 DCT 操作算子。

构造的 4D 矢量矩阵 DCT 操作算子应满足两种性质:一是对称性( $C = C^*$ ),二是正交性( $C^T = C^{-1}$ , 即  $C_U C_U^T = E$ ),三是当矢量退化成本量时,就是通常的二维 DCT 变换矩阵。形式如下:

$$C_U = (c_{k_1 k_2 n_1 n_2}), I = (N_1, N_2), J = (N_1, N_2).$$

其中:

$$c_{k_1 k_2 n_1 n_2} = \left( \frac{2^2}{N_1 N_2} \right)^{\frac{1}{2}} c(k_1) c(k_2) \cos \frac{(2n_1 + 1)k_1 \pi}{2N_1} \cdot \cos \frac{(2n_2 + 1)k_2 \pi}{2N_2},$$

$$c(k_i) = \begin{cases} \frac{1}{\sqrt{2}} & k_i = 0 \\ 1 & k_i = \text{others} \end{cases}, \begin{cases} k_i = 0, 1, \dots, N_i - 1, \\ n_i = 0, 1, \dots, N_i - 1. \end{cases}$$

基于多维矢量矩阵理论,定义  $C$  的转置矩阵:

$$C_U^T = C_U = (c_{n_1 n_2 k_1 k_2}).$$

### 3 多维矢量矩阵 DCT 的快速算法

#### 3.1 1D-DCT 快速算法

首先对文献[10]剖析并对得出的实用性较强的一维快速算法进行了部分的校正与补充。简单来说,利用蝶形图进行优化的重要思想就是巧妙地利用矩阵分解的方法,依照矩阵各自特点提取矩阵中相关联部分,并且运用定义和暂存中间变量,进而达到减少运算次数的目的。图 1 及表 1 分别给出了 1D 8 阶 DCT 快速算法蝶形图及符号定义表。

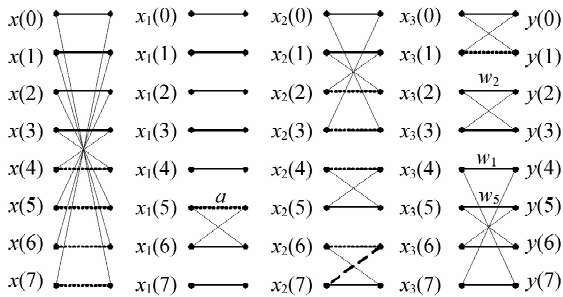


图 1 一维 8 点 DCT 快速算法蝶形示意图

Fig. 1 Butterfly diagram of 1D DCT fast algorithm

表 1 蝶形示意图符号定义

Tab. 1 Symbol definition of butterfly diagram

编号	符号	表达式	理论计算次数
1		$y_0 = x_0 + x_1$	加法: 2
		$y_1 = x_0 - x_1$	乘法: 0
2		$y_0 = a \times (x_0 + x_1)$	加法: 2
		$y_1 = a \times (x_0 - x_1)$	乘法: 2
3		$y_0 = x_0 - x_1$	加法: 2
		$y_1 = x_0 + x_1$	乘法: 0
4		$y_0 = x_1 - x_0$	加法: 2
		$y_1 = x_0 + x_1$	乘法: 0
5		$y_0 = -w_1 x_0 + w_8 - i x_1$	加法: 2
		$y_1 = w_1 x_1 + w_8 - i x_0$	乘法: 4

#### 3.2 2D-DCT 快速算法

2D-DCT 公式可以转化成以下公式(3)的形式,再根据 2D-DCT 可分离性,将公式(3)分解成

(4)和(5),由此便产生了行列分解算法的思想。

$$Y(k, l) = \sqrt{\frac{4}{MN}} \sum_{m=0}^{M-1} \left\{ \alpha(l) \left[ \sum_{n=0}^{N-1} \alpha(k) x(n, m) \times \cos \frac{\pi(2n+1)k}{2N} \right] \times \cos \frac{\pi(2m+1)l}{2M} \right\}, \quad (3)$$

$$z(k, m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} \alpha(k) \times x(n, m) \times \cos \frac{\pi(2n+1)k}{2N}, \quad (4)$$

$$Y(k, l) = \sqrt{\frac{2}{M}} \sum_{m=0}^{M-1} \alpha(l) \times z(k, m) \times \cos \frac{\pi(2m+1)l}{2M}. \quad (5)$$

以输入为  $8 \times 8$  的矩阵数据为例,给出其 2D 矩阵的快速 DCT 的蝶形图,如图 2 所示。

2D-DCT 快速变换中的线性组合<sup>[11]</sup>算法思想是一种非常巧妙地、非递归算法思想。它利用三角函数中积化和差公式、有规律的移位换序以及额外附加的实数加法,用  $N$  组  $N$  点的 1D-DCT 以及  $\log_2 N + 1$  级蝶形对运算最终完成对  $N \times N$  点 2D-DCT 的计算,其中  $\log_2 N + 1$  级蝶形对运算是包括加减法。

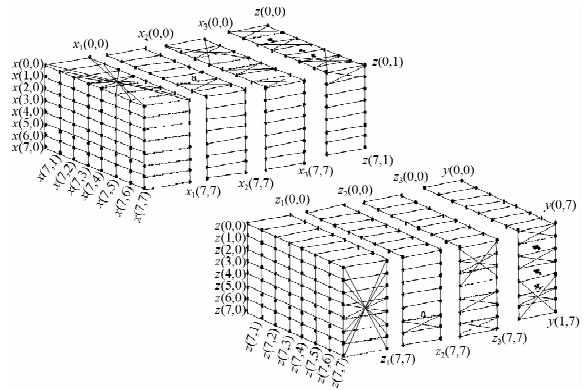


图 2 对  $8 \times 8$  源数据矩阵进行快速 2D-DCT

Fig. 2 Source data matrix for fast 2D-DCT

对于一个给定的输入序列  $f_{i,j}, 0 \leq i \leq N-1, 0 \leq j \leq N-1$ , 忽略系数因子,其 2D-DCT 的非标准形式如下所示 ( $0 \leq m \leq N-1, 0 \leq n \leq N-1$ ):

$$Y_{m,n} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f_{i,j} \cos \frac{(2i+1)m\pi}{2N} \cos \frac{(2j+1)n\pi}{2N}. \quad (6)$$

根据三角函数中积化和差公式  $\cos a \cdot \cos b = \frac{1}{2} [\cos(a+b) + \cos(a-b)]$ , 可将(6)变形为:  $Y_{m,n} = (A_{m,n} + B_{m,n}) / 2$ , 其中:

$$A_{m,n} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{((4j+3)+i)N,j} \cos \frac{(4j+3)((4i+1)m+n) - 4Nq_{ij}m}{2N} \pi = \sum_{i=0}^{N-1} \left[ \sum_{j=0}^{N-1} X_{((4j+3)+i)N,j} \cos \frac{(4j+3)((4i+1)m+n)\pi}{2N} \right],$$

$$B_{m,n} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{((4j+3)+i)N,j} \cos \frac{(4j+3)((4i+1)m-n) - 4Nq_{ij}m}{2N} \pi = \sum_{i=0}^{N-1} \left[ \sum_{j=0}^{N-1} X_{((4j+3)+i)N,j} \cos \frac{(4j+3)((4i+1)m-n)\pi}{2N} \right].$$

这就意味着完成对  $N \times N$  点 2D-DCT 的计算, 只需用  $N$  组  $N$  点的 1D-DCT 计算量与一些额外附加的实数加法即可。二维 DCT 快速变换之线性组合法的蝶形示意图见图 3 所示。

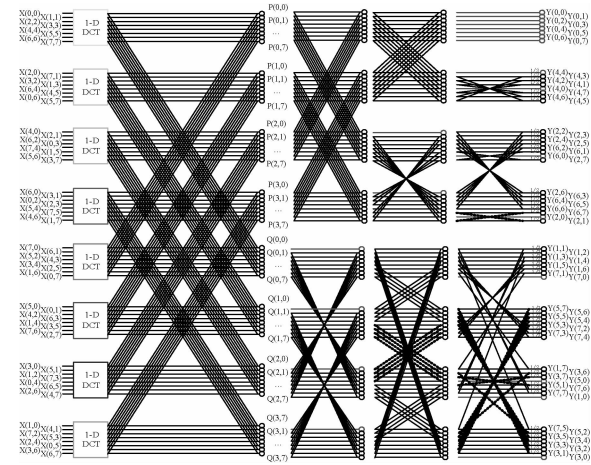


图 3 对  $8 \times 8$  数据进行线性组合的蝶形示意图

Fig. 3 Butterfly diagram of  $8 \times 8$  data by linear combination method

### 3.3 多维矢量矩阵 DCT 的快速算法

在多维矢量矩阵理论的基础上, 给出方法一: 对  $Y, U, V$  3 个分量分别进行  $8 \times 8 \times 8$  子数据块快速 DCT 运算, 即:

$$P_{8 \times 8 \times 8} = C_{8 \times 8 \times 8 \times 8} A_{8 \times 8 \times 8} C_{8 \times 8}^T. \quad (7)$$

1. 设多维矢量矩阵  $P'_{8 \times 8 \times 8} = C_{8 \times 8 \times 8 \times 8} A_{8 \times 8 \times 8}$ , 令  $C_{1 \times J} = C_{8 \times 8 \times 8 \times 8}$ ,  $A_{J \times K} = A_{8 \times 8 \times 8}$ . 依据多维矢量矩阵的乘法规则, 对  $J$  分量进行行列分解法的快速 2D-DCT.

2. 令  $P_{8 \times 8 \times 8} = P'_{8 \times 8 \times 8} C_{8 \times 8}^T$ ,  $P'_{1 \times J} = P'_{8 \times 8 \times 8}$ ,  $C_{J \times K}^T = C_{8 \times 8}^T$ , 依据多维矢量矩阵乘法规则, 对  $J$  分量进行快速 1D-DCT.

图 4 给出方法一的蝶形示意图(彩图见期刊网刊电子版)。红色代表源数据及多维矩阵中的行方向的数据经过行方向快速 1D-DCT 获得中间多维

矩阵的数据; 绿色代表经过行变换的中间数据及多维矩阵中的列方向的数据经过列方向快速 1D-DCT 获得中间多维矩阵的数据; 蓝色代表经过列变换的中间数据及多维矩阵中的帧方向的数据经过帧方向快速 1D-DCT 获得最终多维矩阵的数据, 即最终结果。

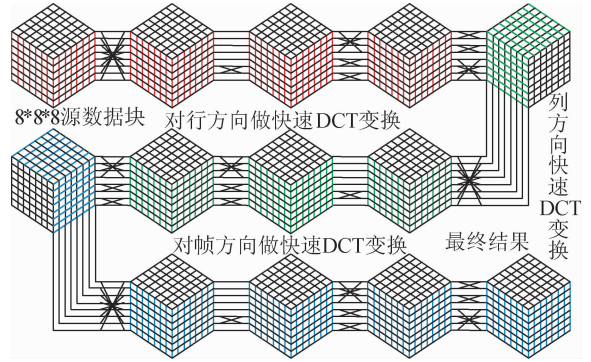


图 4 方法一: 视频  $8 \times 8 \times 8$  子数据块蝶形示意图

Fig. 4 Method one: Butterfly diagram of video  $8 \times 8 \times 8$  data

方法二: 对  $Y, U, V$  3 个分量分别进行  $8 \times 8 \times 8$  子数据块快速 DCT 运算, 即:

$$P_{8 \times 8 \times 8} = C_{8 \times 8 \times 8 \times 8} A_{8 \times 8 \times 8} C_{8 \times 8}^T. \quad (8)$$

1. 设多维矢量矩阵  $P'_{8 \times 8 \times 8} = C_{8 \times 8 \times 8 \times 8} A_{8 \times 8 \times 8}$ , 令  $C_{1 \times J} = C_{8 \times 8 \times 8 \times 8}$ ,  $A_{J \times K} = A_{8 \times 8 \times 8}$ . 依据多维矢量矩阵的乘法规则, 对  $J$  分量进行线性组合法的快速 2D-DCT.

2. 令  $P_{8 \times 8 \times 8} = P'_{8 \times 8 \times 8} C_{8 \times 8}^T$ ,  $P'_{1 \times J} = P'_{8 \times 8 \times 8}$ ,  $C_{J \times K}^T = C_{8 \times 8}^T$ , 依据多维矢量矩阵的乘法规则, 对  $J$  分量进行快速 1D-DCT.

图 5 给出方法二的蝶形示意图(彩图见期刊网刊电子版)。首先对  $8 \times 8 \times 8$  原始数据进行分组取点, 分组取点的方法是每一帧内分成八组, 每一组分别用黄色、浅绿、浅蓝、紫色、蓝色、红色、深蓝、绿色这 8 种标准色予以区别, 每组内共有 8 个数据。之后将分组取点后的数据进行有序排列组合, 属于

同一组的数据排列在一起,将每排列好的一组八点数据先进行快速 1D-DCT,将得到的中间多维矩阵数据按线性组合法蝶形示意图所示进行加运算,将再次得到的中间多维矢量数据再次重新组合排列,最后再对每帧之间属于同一组同一位置上的数据进行快速 1D-DCT,最终得到的多维矩阵数据即为最终结果。

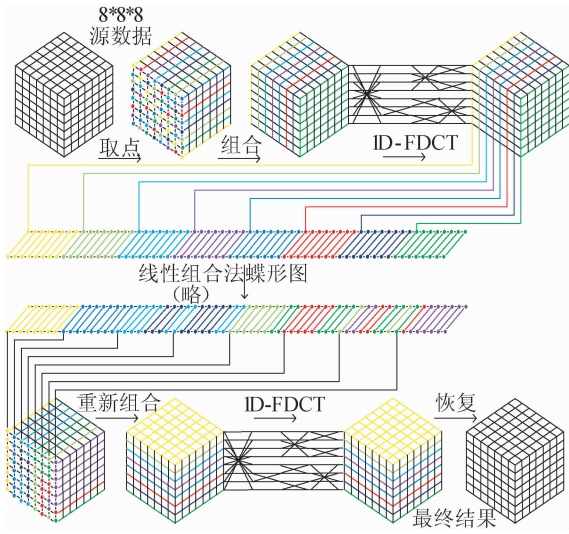


图 5 方法二:视频 8×8×8 子数据块蝶形示意图

Fig. 5 Method two: Butterfly diagram of video 8×8×8 data

众所周知,在计算机计算的过程中,乘法次数明显会影响计算的速度。以  $M=3$  维,  $N=8$  阶的数据为例,分别对传统公式法 DCT、多维矢量矩阵法 DCT、多维矢量矩阵快速 DCT 的方法一和方法二的乘法次数进行了对比,如表 2 所示。

表 2 三维分块数据的 DCT 运算量对比

Tab. 2 Contrast of data operations of 3D DCT

M 维 N 阶的数据	乘法运算次数	乘法次数
传统公式法 DCT	$N^{2M}$	262 144
多维矢量矩阵法 DCT	$N^M(N^{M-1} + N)$	36 864
多维矢量矩阵快速 DCT(方法一)	$MN^{M-1}[(N \log_2 N)/2 - 1]$	2 112
多维矢量矩阵快速 DCT(方法二)	$NM-1(N \log_2 N - 1)$	1 408

从表 2 可以看出,本文提出算法的两组方法,均明显降低了乘法的次数,换句话说,如果将本文

提出的两组方法应用到数字视频压缩编码系统中,将会在很大程度上提高运行速度,降低运行时间。在下一节的实验中将给出本文两组方法之间的对比,以及与传统方法相比的优势。

### 4 实验结果

在测试实验中,选取的是实验室标准视频库中 3 种不同格式大小的源彩色视频数据作为测试视频,它们的格式大小分别是  $352 \times 288 \times 8$ 、 $512 \times 384 \times 8$ 、 $704 \times 576 \times 8$ 。

考虑到邓琳琳<sup>[12]</sup>论文中算法优于 JPEG,因此表 3 至表 5 分别给出了在非均匀量化条件下,基于多维矢量矩阵的 DCT 快速变换算法(方法一)与邓琳琳的基于多维矢量矩阵 DCT 算法对 3 种格式大小的源彩色测试视频数据进行对比实验的结果。表格中的 DCT 与 IDCT 所给出的时间均为两部分时间之和,即:一部分是变换所用时间、另一部分是非均匀量化所用时间。

表 3 352×288×8 测试视频数据的对比试验结果

Tab. 3 Results of contrast test for video 352×288×8 data

		t/s	
8×8×8 分块	0.015 0	8×8×8 分块	0.015 0
多维矢量矩阵 DCT	0.885 3	多维矢量矩阵的 DCT 快速变换算法(方法一)	0.062 0
多维矢量矩阵 IDCT	0.867 5	多维矢量矩阵的 IDCT 快速变换算法(方法一)	0.055 0
8×8×8 块重组	0.016 0	8×8×8 块重组	0.016 0
总时间	1.783 8	总时间	0.148 0

表 4 512×384×8 测试视频数据的对比试验结果

Tab. 4 Results of contrast test for video 352×288×8 data

		t/s	
8×8×8 分块	0.023 0	8×8×8 分块	0.023 0
多维矢量矩阵 DCT	1.727 0	多维矢量矩阵的 DCT 快速变换算法(方法一)	0.109 0
多维矢量矩阵 IDCT	1.687 3	多维矢量矩阵的 IDCT 快速变换算法(方法一)	0.101 5
8×8×8 块重组	0.023 5	8×8×8 块重组	0.023 5
总时间	3.460 8	总时间	0.257 0

表 5 704×576×8 测试视频数据的对比试验结果

Tab. 5 Results of contrast test for video 704×576×8 data

		t/s	
8×8×8 分块	0.046 0	8×8×8 分块	0.046 0
多维矢量矩阵 DCT	3.532 0	多维矢量矩阵的 DCT 快速变换算法(方法一)	0.221 0
多维矢量矩阵 IDCT	3.461 0	多维矢量矩阵的 IDCT 快速变换算法(方法一)	0.195 5
8×8×8 块重组	0.047 0	8×8×8 块重组	0.047 0
总时间	7.086 0	总时间	0.509 5

图 6 和图 7 分别给出了原始视频和非均匀量化后的重构视频。

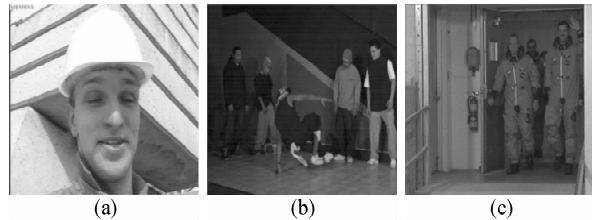


图 6 原始视频  
Fig. 6 Original videos

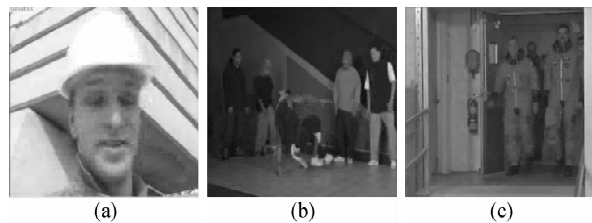


图 7 重构视频  
Fig. 7 Reconstructed videos

从重构视频和图表结果可以看出,对 3 种不同格式的测试视频数据进行多维矢量矩阵的 DCT 快速变换算法在保证与邓琳琳的基于多维矢量矩阵 DCT 有着相同重建视频质量的同时,其执行时间有了明显的降低,有效地提高了运算速度,并且实验结果显示 DCT 所用时间均小于 0.25 s,完全满足视频流实时的要求。因此,基于多维矢量矩阵的快速 DCT 算法(方法一)具有一定的可行性,当以速度、时间为评价指标时,此种算法的优越性更加突出。

图 8 和图 9 分别给出了基于多维矢量矩阵的

快速 DCT 算法思想中方法一和方法二在无量化条件下与均匀量化条件下的 DCT 所需时间的比较结果。图中测试的视频大小中 1 代表的是 352×288×8,2 代表的是 512×384×8,3 代表的是 704×576×8。

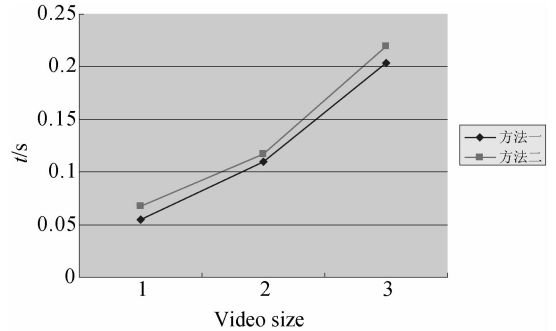


图 8 两种方法在无量化条件下的时间比较  
Fig. 8 Time comparison of two method without quantization

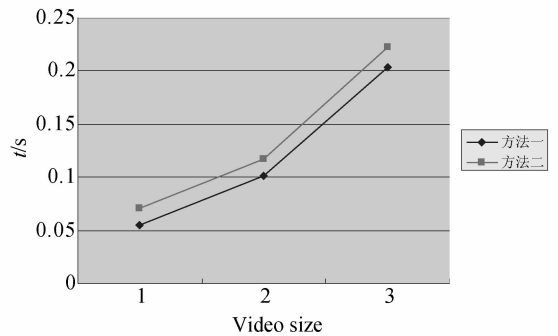


图 9 两种方法在均匀量化条件下时间比较  
Fig. 9 Time comparison of two method with uniform quantization

从图 8 和图 9 可以看出基于多维矢量矩阵的快速 DCT 算法的方法一、二在无量化条件下和均匀量化条件下 DCT 时间非常接近。虽然方法二中所需理论乘法运算量比方法一中所需理论乘法运算量减少近 1/3,但是其需要大量的相对于计算机来说是“无序”的取点分组,并且在用 VC++6.0 软件编程实验时,需按照推导出的蝶形示意图逐个“对号入座”。对计算机而言,部分程序毫无规律可言,所编写的程序非常繁复,这些都使得最后程序运行的时间结果并不像理论上那么具有一定的优越性。但如果将方法二采用硬件方式加以实现,必将取得更好的效果。

## 5 结 论

本文提出了基于多维矢量矩阵的 DCT 快速算法的两组方法。在一维快速 DCT 算法蝶形图的基础上,通过对其进行部分校正和补充,然后结合具有把多维数据表示在同一个多维模型中的优点的多维

矢量矩阵理论,给出了本文算法的两组方法对应的蝶形图,从而将 DCT 快速算法由低维扩展到多维,以便处理更多维数的视频,比如立体视频、多视角视频等。实验结果表明:本文提出算法的两种方法运行所用的时间均小于 0.25 s。本文提出的算法在满足视频流实时性的前提下,将 DCT 的快速算法从低维扩展到了多维的层面。

## 参考文献:

- [1] 汤毅,辛勤,李刚,等. 基于内容的高光谱图像无损压缩[J]. 光学精密工程,2012,20(3):668-674.  
TANG Y, XIN Q, LI G, et al.. Lossless compression of hyperspectral images based on contents [J]. *Opt. Precision Eng.*, 2012, 20(3): 668-674. (in Chinese)
- [2] 黄爱爱,陈耀武. 可伸缩视频编码中基于部分解码的率失真优化[J]. 光学精密工程,2010,19(9):2170-2176.  
HUANG A A, CHEN Y W. Rate distortion optimization based on partial decoding for scalable video coding [J]. *Opt. Precision Eng.*, 2010, 19(9): 2170-2176. (in Chinese)
- [3] WU H R, PAOLONI F J. A two-dimensional fast cosine transform algorithm based on Hou's approach [J]. *IEEE Trans Signal Processing*, 1991, 39(2):544-546.
- [4] 吴红文,李久贤,夏良正. 一种新二维离散余弦变换快速算法[J]. 东南大学学报,1996,26(2):7-15.  
WU H W, LI J X, XIA L ZH. A new 2D discrete cosine transform fast algorithm [J]. *Journal of Southeast University*, 1996, 26(2): 7-15. (in Chinese)
- [5] 罗小明,王能忠. 一种二维 DCT 快速算法及其改进[J]. 西南师范大学学报:自然科学版,1997,22(3):271-277.  
LUO X M, WANG N ZH. A 2D DCT fast algorithm and its improvement[J]. *Journal of Southwest China Normal University: Natural Science*, 1997, 22(3): 271-277.
- [6] ZHENG W, LIU Y CH. Research in a fast DCT algorithm based on JPEG [C]. *International Conference on Consumer Electronics, Communications and Networks (CECNet)*, 2011:551-553.
- [7] YU G, SUN W B, ZHOU M Y, et al.. Research on fast 1-D DCT algorithm based on parallel computer [C]. *International Conference on Computer Science and Electronics Engineering (ICCSEE)*, 2012: 192-195.
- [8] SUN W B, CHEN H X, SUN W B. Research on fast 2-D DCT algorithm based on parallel computer [C]. *International Conference on Networks Computer and Information Security (NCIS)*, 2011: 401-404.
- [9] SANG A J, CHEN M SH, CHEN H X, et al.. Multi-dimensional vector matrix theory and its application in color image coding [J]. *Imaging Science Journal*, 2010, 6(58):171-176.
- [10] 管海兵,宣国荣. 快速 DCT 算法的蝶形图[J]. 微型电脑应用,1999,2:24-26.  
GUAN H B, XUAN G R. Butterfly diagram of fast DCT algorithm [J]. *Microcomputer Applications*, 1999, 2: 24-26. (in Chinese)
- [11] HUANG Y H, WU J M. A Refined Fast 2-D Discrete Cosine Transform Algorithm. *IEEE Transactions on Signal Processing*, 1999, 47(3):904-907.
- [12] 邓琳琳. 基于视觉特性的彩色视频流压缩编码算法的研究[D]. 长春:吉林大学,2009.  
DENG L L. *Research on compression coding algorithm for video based on human visual system* [D]. Changchun: Jilin University, 2009. (in Chinese)

## 作者简介:



桑爱军(1973—),女,山东莱州人,博士,教授,2002年于吉林大学获得博士学位。主要从事多维信号处理,多维视频流编解码方面的研究。E-mail: sangaj@jlu.edu.cn



吴 杨(1987—),女,吉林长春人,在读研究生,2010年于长春理工大学获得学士学位。主要从事彩色视频流编解码方面的研究。E-mail: hahahehe.1987@yahoo.com.cn



崔海廷(1987—),女,吉林长春人,在读研究生,2010年于长春理工大学获得学士学位。主要从事多视角视频编解码方面的研究。E-mail: cui\_hai\_ting@126.com



陈贺新(1949—),男,吉林大安人,博士,教授,博士生导师,1990年于吉林大学获得博士学位。主要从事多维信号处理,图像与视频编解码,视频通信,多媒体数据库技术方面的研究。E-mail: chenhx@jlu.edu.cn

(版权所有 未经许可 不得转载)