

# 离散 Haar 变换的快速算法设计\*

卢力

王能超

(武汉城市建设学院电气与计算机工程系, 武汉 430074) (华中理工大学并行计算研究所, 武汉 430074)

**摘要** 运用二分技术设计出了离散 Haar 变换的一组快速算法。

**关键词** Haar 函数, Haar 矩阵, 离散 Haar 变换, 二分技术, 快速算法

## 1 引言

Haar 函数系是荷兰数学家 A. Haar 于 1910 年提出的一组完备的正交函数系<sup>[1,2]</sup>。所谓 Haar 函数系是指下列函数系:

$$\begin{aligned}
 h(0,t) &= 1, \quad 0 \leq t < 1; \\
 h(1,t) &= \begin{cases} 1, & 0 \leq t < 1/2, \\ -1, & 1/2 \leq t < 1; \end{cases} \quad (1) \\
 h(2^k + j,t) &= h(1, 2^k t - j), \\
 k &= 1, 2, \dots; j = 0, 1, 2, \dots, 2^k - 1
 \end{aligned}$$

由于 Haar 函数系的前两个函数为全局函数(在单位区间 $[0,1]$ 上非零),而其它函数均为局部函数(在单位区间的部分上非零),这种全局/局部结构应用于图像处理的边缘检测、轮廓提取以及图像编码等方面是非常有用的。

设  $N = 2^n (n = 1, 2, 3, \dots)$ , 按 Haar 函数的定义, 将前  $N$  个 Haar 函数在二分区间

$\left\{ \left[ \frac{j}{N}, \frac{j+1}{N} \right] \right\}_{j=0}^{N-1}$  上的函数值依矩阵方式进行排列

得到  $N$  阶 Haar 矩阵, 记为  $H_N = [H_N(j, k)]$ 。

显然, Haar 矩阵具有如下的递推性:

$$H_1 = [1], \quad H_{2N} = \begin{bmatrix} H_N \otimes (1 & 1) \\ I_N \otimes (1 & -1) \end{bmatrix} (N > 1).$$

其中  $\otimes$  为 Kronecher 乘积运算符号。

设  $[x(k)]$  为输入数据,  $[X(j)]$  为  $[x(k)]$  的 Haar 变换系数, 则称

$$[X(j)]^T = H_N [x(k)]^T, \quad j = 0, 1, 2, \dots, N-1 \quad (2)$$

$$[x(k)]^T = H_N^T P_N [X(j)]^T, \quad k = 0, 1, 2, \dots, N-1 \quad (3)$$

分别为离散 Haar 变换和逆离散 Haar 变换<sup>[1]</sup>。其中  $H_N^T P_N = H_N^{-1}$ , 且

$$P_N = \text{Diag}[2^{-n}, 2^{-n}, 2^{-n+1}, 2^{-n+1}, 2^{-n+2}, 2^{-n+2}, \dots, 2^{-1}, \dots, 2^{-1}]$$

如  $P_8 = \text{Diag}[2^{-3}, 2^{-3}, 2^{-2}, 2^{-2}, 2^{-1}, 2^{-1}, 2^{-1}, 2^{-1}]$  将(2)改写为

$$X(j) = \sum_{k=0}^{N-1} x(k) H_N(j, k) \quad (4)$$

从而由(4)式可知, 要完成  $N$  点的 DHT 运算, 需要做  $N \log_2 N$  次加减操作。

本文将运用文献[4,5]中所提出的二分技术设计离散 Haar 变换(4)的几种快速算法。对应的逆变换的快速算法易得(限于篇幅, 本文不加讨论)。

## 2 离散 Haar 变换的快速算法设计

### 2.1 奇偶二分法

**命题 1** 当  $j, k = 0, 1, \dots, N/2 - 1$  时,  $H_N$  具有下列性质:

$$\begin{aligned}
 H_N(j, 2k) &= H_N(j, 2k+1) = H_{N/2}(j, k), \\
 H_N(N/2 + j, 2j) &= \\
 &= -H_N(N/2 + j, 2j+1) = 1
 \end{aligned}$$

\* 国家 863 高技术资助项目(863-306-ZD-01-8)

收稿日期: 1997-06-02

按照文献[4,5]所提出的奇偶二分法,将(4)式拆成两个子和式:

$$X(j) = \sum_{k=0}^{N/2-1} x(2k)H_N(j,2k) + \sum_{k=0}^{N/2-1} x(2k+1)H_N(j,2k+1)$$

$j = 0, 1, 2, \dots, N-1$

根据命题1,当  $j = 0, 1, 2, \dots, N/2-1$  时有

$$X(j) = \sum_{k=0}^{N/2-1} [x(2k) + x(2k+1)]H_{N/2}(j,k)$$

$$X(N/2+j) = \sum_{k=0}^{N/2-1} [x(2k) - x(2k+1)]H_{N/2}(N/2+j,k) = x(2j) - x(2j+1)$$

若令

$$\begin{cases} x_1(k) = x(2k) + x(2k+1) \\ x_1(N/2+k) = x(2k) - x(2k+1) \end{cases} \quad k = 0, 1, 2, \dots, N/2-1 \quad (5)$$

则有

$$\begin{cases} X(j) = \sum_{k=0}^{N/2-1} x_1(k)H_{N/2}(j,k) \\ X(N/2+j) = x_1(N/2+j) \end{cases} \quad j = 0, 1, 2, \dots, N/2-1 \quad (6)$$

于是奇偶二分一次,将1个  $N$  点的 DHT 变成1个  $N/2$  点的 DHT,并同时得出  $N/2$  个变换结果。

继续二分直到一点变换为止。这样对  $N$  个数据的 DHT,经过  $\log_2 N$  步奇偶二分,便得变换的结果。此算法为 Andrews 算法<sup>[1,3]</sup>(奇偶到对半)。按此算法,要完成  $N$  点的 DHT 运算,仅需要做  $2(N-1)$  次加减操作。

Andrews 算法的流程图如图1所示 ( $N=8$ )。

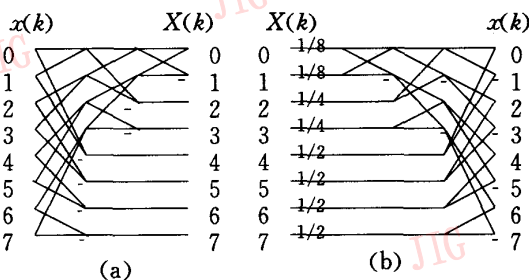


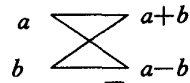
图1 算法1的计算流程图(Andrews算法)

( $N=8$ ), (a) Haar 变换, (b) 逆 Haar 变换。

Fig. 1 Flow diagram for algorithm 1 (Andrews algorithm)

( $N=8$ ), (a) HT, (b) IHT

所采用的蝶形计算格式为:



Andrews 算法为非即位运算。对此算法进行改造,改造过程如下。

将(5)式变为:

$$\begin{cases} x_1(2k) = x(2k) + x(2k+1) \\ x_1(2k+1) = x(2k) - x(2k+1) \end{cases} \quad k = 0, 1, 2, \dots, N/2-1 \quad (7)$$

则(6)式变为

$$\begin{cases} X(j) = \sum_{k=0}^{N/2-1} x_1(2k)H_{N/2}(j,k) \\ X(N/2+j) = x_1(2j+1) \end{cases} \quad j = 0, 1, 2, \dots, N/2-1 \quad (8)$$

同样经过二分一次后,将原来1个  $N$  点的 DHT 变为1个  $N/2$  点的 DHT,并得到  $N/2$  个变换结果。

继续二分直到一点变换为止。此算法<sup>[2]</sup>为即位运算算法,但最后一步要进行调序。如果将这一步的调序分配到奇偶二分的每一步中去,即进行  $\log_2 N$  步调序,便得到另外一种算法称之为改进的 Andrews 算法(奇偶到奇偶)。

改进的 Andrews 算法的流程图如图2所示 ( $N=8$ )。所采用的蝶形计算格式同前。

### 2.2 对半二分法

方法1 将  $H_N$  的  $2k$  列变换到左边,  $2k+1$  列变换到右边,  $k = 0, 1, \dots, N/2-1$ , 得到一新的矩阵记为  $\tilde{H}_N$ 。

显然  $\tilde{H}_N = H_N I_N$ , 其中  $I_N$  为  $N$  阶单位矩阵,且  $I_N I_N^T = I_N$ , 即  $I_N^{-1} = I_N^T$

命题2 当  $j, k = 0, 1, \dots, N/2-1$  时,  $\tilde{H}_N$  具有下列性质:

$$\begin{aligned} \tilde{H}_N(j,k) &= \tilde{H}_N(j, N/2+k) = H_{N/2}(j,k), \\ \tilde{H}_N(N/2+j,j) &= \\ &= -\tilde{H}_N(N/2+j, N/2+j) = 1 \end{aligned}$$

从而,若令

$$[\tilde{x}(k)] = [x(k)]I_N,$$

则(4)式变为

$$X(j) = \sum_{k=0}^{N-1} x(k)H_N(j,k) = \sum_{k=0}^{N-1} \tilde{x}(k)\tilde{H}_N(j,k) \quad (9)$$

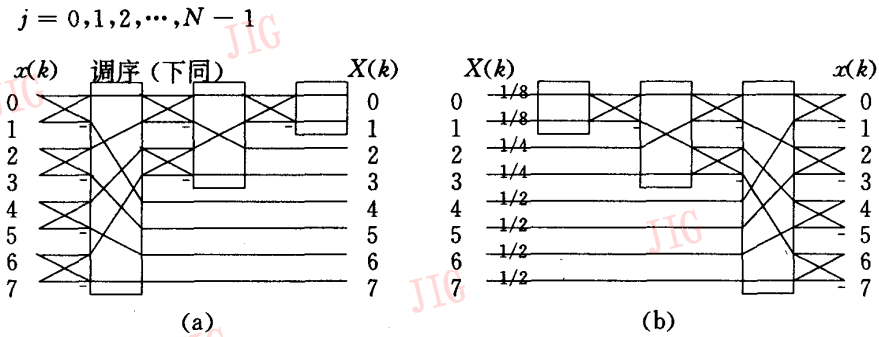


图 2 算法 2 的计算流程图(改进的 Andrews 算法)( $N = 8$ ), (a) Haar 变换, (b) 逆 Haar 变换  
Fig. 2 Flow diagram for algorithm 2(improved Andrews algorithm)( $N = 8$ ), (a) HT, (b) IHT

按照文献[4,5]所提出的对半二分法, 将(9)式拆成两个子和式:

$$X(j) = \sum_{k=0}^{N-1} \tilde{x}(k) \tilde{H}_N(j, k) = \sum_{k=0}^{N/2-1} \tilde{x}(k) \tilde{H}_N(j, k) + \sum_{k=0}^{N/2-1} \tilde{x}(N/2 + k) \tilde{H}_N(j, N/2 + k)$$

$j = 0, 1, 2, \dots, N - 1$

由命题 2 知, 当  $j = 0, 1, 2, \dots, N/2 - 1$  时有

$$X(j) = \sum_{k=0}^{N/2-1} [\tilde{x}(k) + \tilde{x}(N/2 + k)] \tilde{H}_N(j, k) = \sum_{k=0}^{N/2-1} [\tilde{x}(k) + \tilde{x}(N/2 + k)] H_{N/2}(j, k),$$

$$X(N/2 + j) = \sum_{k=0}^{N/2-1} [\tilde{x}(k) - \tilde{x}(N/2 + k)] \tilde{H}_N(N/2 + j, k) = \tilde{x}(j) - \tilde{x}(N/2 + j)$$

若令

$$\begin{cases} x_1(k) = \tilde{x}(k) + \tilde{x}(N/2 + k) \\ x_1(N/2 + k) = \tilde{x}(k) - \tilde{x}(N/2 + k) \end{cases}$$

$k = 0, 1, 2, \dots, N/2 - 1$

则有

$$\begin{cases} X(j) = \sum_{k=0}^{N/2-1} x_1(k) H_{N/2}(j, k), \\ X(N/2 + j) = x_1(N/2 + j) \end{cases}$$

$j = 0, 1, 2, \dots, N/2 - 1$

于是对半二分一次, 将 1 个  $N$  点的 DHT 变成 1 个  $N/2$  点的 DHT, 并同时得出  $N/2$  个变换结果。

继续二分直到一点变换为止。对  $N$  个数据的 DHT, 经过  $\log_2 N$  步对半二分, 便得变换的结果。此算法是即位运算算法(对半到对半)。按此算法, 要完成  $N$  点的 DHT 运算, 需要做  $2(N - 1)$  次加减操作及  $\log_2 N$  步调序。

这种算法的流程图如图 3 所示 ( $N = 8$ )。

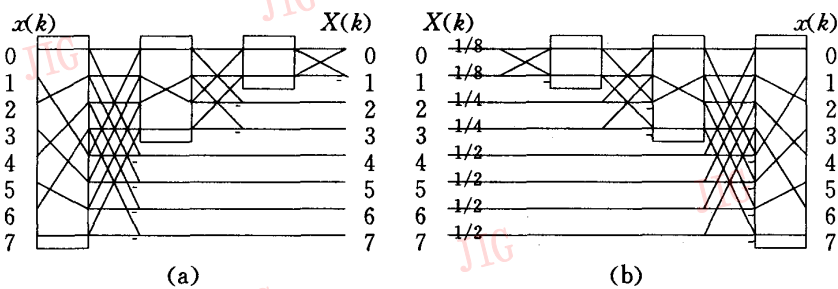


图 3 算法 3 的计算流程图( $N = 8$ ), (a) Haar 变换, (b) 逆 Haar 变换。

Fig. 3 Flow diagram for algorithm 3( $N = 8$ ), (a) HT, (b) IHT.

所采用的蝶形计算格式同前。

方法 2 将  $H_N$  的列序号进行二进制逆序调序,

得到一新的矩阵记为  $\hat{H}_N$ 。

显然  $\hat{H}_N = H_N \hat{I}_N$ , 其中  $\hat{I}_N$  为  $N$  阶单位矩阵, 且

$\hat{I}_N \hat{I}_N^T = I_N$ , 即  $\hat{I}_N^{-1} = \hat{I}_N^T$ .

**命题 3** 当  $j, k = 0, 1, \dots, N/2 - 1$  时,  $\hat{H}_N$  具有下列性质:

$$\hat{H}_N(j, k) = \hat{H}_N(j, N/2 + k) = \hat{H}_{N/2}(j, k),$$

$$\hat{H}_N(N/2 + j, \langle j \rangle) =$$

$$-\hat{H}_N(N/2 + j, N/2 + \langle j \rangle) = \hat{I}_{N/2}(j, \langle j \rangle)$$

其中  $\langle j \rangle$  为  $j$  的二进制逆序。

从而, 若设

$$[\hat{x}(k)] = [x(k)] \hat{I}_N,$$

则(4)式变为

$$X(j) = \sum_{k=0}^{N-1} x(k) H_N(j, k) = \sum_{k=0}^{N-1} \hat{x}(k) \hat{H}_N(j, k) \quad (10)$$

$$j = 0, 1, 2, \dots, N - 1$$

按照文献[4,5]所提出的对半二分法, 将(10)式拆成两个子和式:

$$X(j) = \sum_{k=0}^{N-1} \hat{x}(k) \hat{H}_N(j, k) = \sum_{k=0}^{N/2-1} \hat{x}(k) \hat{H}_N(j, k) + \sum_{k=0}^{N/2-1} \hat{x}(N/2 + k) \hat{H}_N(j, N/2 + k)$$

$$j = 0, 1, 2, \dots, N - 1.$$

由命题 3 知, 当  $j = 0, 1, 2, \dots, N/2 - 1$  时有

$$X(j) = \sum_{k=0}^{N/2-1} [\hat{x}(k) + \hat{x}(N/2 + k)] \hat{H}_N(j, k) =$$

$$\sum_{k=0}^{N/2-1} [\hat{x}(k) + \hat{x}(N/2 + k)] \hat{H}_{N/2}(j, k),$$

$$X(N/2 + j) =$$

$$\sum_{k=0}^{N/2-1} [\hat{x}(k) - \hat{x}(N/2 + k)] \hat{H}_{N/2}(N/2 + j, k) =$$

$$\hat{x}(\langle j \rangle) - \hat{x}(N/2 + \langle j \rangle)$$

若令

$$\begin{cases} \hat{x}_1(k) = \hat{x}(k) + \hat{x}(N/2 + k) \\ \hat{x}_1(N/2 + k) = \hat{x}(k) - \hat{x}(N/2 + k) \end{cases}$$

$$k = 0, 1, 2, \dots, N/2 - 1$$

则有

$$\begin{cases} X(j) = \sum_{k=0}^{N/2-1} \hat{x}_1(k) \hat{H}_{N/2}(j, k), \\ X(N/2 + j) = \hat{x}_1(N/2 + \langle j \rangle). \end{cases}$$

$$j = 0, 1, 2, \dots, N/2 - 1$$

于是对半二分一次, 将 1 个  $N$  点的 DHT 变成 1 个  $N/2$  点的 DHT, 并同时得出  $N/2$  个变换结果。

继续二分直到一点变换为止。对  $N$  个数据的 DHT, 经过  $\log_2 N$  步对半二分, 便得变换的结果。此算法为 Tukey-Cooley 算法<sup>[3]</sup>(对半到对半)。按此算法, 要完成  $N$  点的 DHT 运算, 需要做  $2(N-1)$  次加减运算及  $\log_2 N$  步调序。

这种算法的流程图如图 4 所示 ( $N = 8$ )。

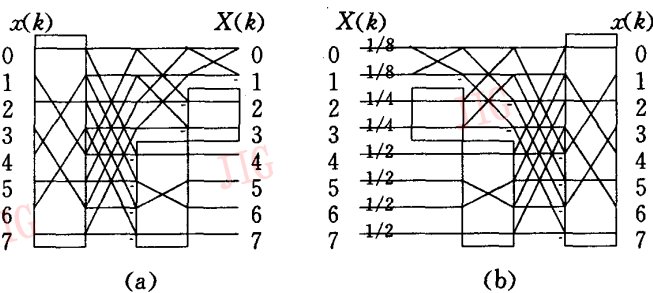


图 4 算法 4 的计算流程图(Tukey-cooley 算法)( $N = 8$ ), (a) Haar 变换, (b) 逆 Haar 变换。

Fig. 4 Flow diagram for algorithm 4(Tukey-Cooley type algorithm)( $N = 8$ ), (a) HT, (b) IHT.

所采用的蝶形计算格式同前。

### 参考文献

- Elliott D F, Rao K R. Fast Transforms; Algorithms, Analyses, Applications. New York: Academic Press. 1982, 399~405.
- Rao K R, ed. Discrete Transforms and Their Applications. New

York; Van Nostrand Reinhold Company. 1985, 234~238.

- Rao K R, Narasimham M A, Revuluri K. A Family of Discrete Haar Transform. Comput. & Elect. Engng. 1975, 2: 367~388.
- 王能超. 同步并行算法设计的二分技术. 中国科学(A 辑). 1995, 2(2): 207~211.
- 王能超. 同步并行算法设计. 北京: 科学出版社. 1996.



卢力, 武汉城市建设学院电气与计算机工程系讲师。1986年毕业于华中师范大学数学系获理学学士学位。1997年在华中理工大学数学系获得计算数学硕士学位。主要研究方向为并行计算和图像信号处理等。



王能超, 教授, 博士生导师, 现任华中理工大学并行计算研究所所长, 中国图象图形学会理事, 全国计算机数学研究会理事长, 湖北省计算数学学会理事长。主要研究方向: 并行计算和演化数学方法。

## Fast Algorithms Design for Discrete Haar Transform

Lu Li

(Department of electrical engineering & computer engineering, WUCI, Wuhan 430074)

Wang Nengchao

(Parallel Computation Research Institute, HUST, Wuhan 430074)

**Abstract** Some fast algorithms for discrete Haar transform are designed based on the Dichotom Technique.

**Keywords** Haar function, Haar matrix, Discrete Haar transform, Dichotom technique, Fast algorithm

### VTEL 美国视讯公司桌面会议电视产品—SmartStation 被评为年度荣誉会议电视产品 —荣获 CTI 杂志第一年度奖

VTEL 美国视讯公司的 SmartStation 桌面视讯系统被 CTI 杂志评为 1997 年年度荣誉会议电视产品。

据 CTI 杂志称:“我们第一年度产品奖是授予在其领域表现杰出的高科技产品。我们相信这些产品会为您在工作的领域中选择最佳计算机视讯集成产品提供第一手资料。”

VTEL 副总裁兼个人及工作组系统总经理 Bob Romano 说“能够得到 CTI 杂志的认可, 又一次证明了 SmartStation 桌面视讯系统独具创新的平台设计, 可提供业界最佳的图像质量、数据共享集成, 并针对客户当前和将来的网络具有强大的兼容性。”

最近, SmartStation 还接受了其它两家杂志的荣誉称号。在有 12 家流行桌面视讯产品面对面的评比中, SmartStation 被《通讯新闻》月刊 (Communications News) 杂志评为最佳桌面视讯产品。在德国汉诺威举办的 CeBIT'97 上, 《会议电视观察》杂志 (Videoconferencing Insight) 授予 SmartStation 最佳新技术奖。

SmartStation 于 1997 年 4 月问世。它将微软公司 NetMeeting™ 2.0 与数据共享、38kbps 图象质量以及 PCI 即插即用特征相融合, 成为高品质桌面视讯市场的领导产品。