

文章编号: 1001- 2486(2008) 03- 0059- 06

# 适于 PCT 编码的低内存通用树状多带滤波器组实现<sup>\*</sup>

张雄明<sup>1</sup>, 卢焕章<sup>1</sup>, 成礼智<sup>2</sup>

(1. 国防科技大学 ATR 实验室, 湖南长沙 410073; 2. 国防科技大学理学院, 湖南长沙 410073)

**摘要:** 基于常用的多带滤波器组的紧支撑性及 FIFO (First-in First-out) 缓存技术提出了一种具有低内存需求的通用树状多带滤波器组 (Tree-structured Filter Bank, TSFB) 的实现方法 SBFB。该方法具有两大特点: (1) 生成与全局变换法相同的子带系数, 但是内存需求大大减小且仅与图像宽度及采用的 TSFB 相关; (2) 在 FIFO 缓存中直接生成父子树 (Parent-children Tree, PCT)。基于 PCT 的编码器可直接对位于缓存中的 PCT 进行编码, 而无须在 SBFB 与编码器之间引入中间缓存。通过分析 TSFB 的各分解层中的数据流, 给出了样本点与子带系数之间的时序关系, 并且使用该关系从数学上严格证明了 SBFB 的正确性。

**关键词:** 树状多带滤波器组; 低内存实现; 父子树结构

中图分类号: TP391 文献标识码: A

## Low-memory Implementation of Generic Tree-structured Filter Banks for PCT-based Image Coders

ZHANG Xiong-ming<sup>1</sup>, LU Huan-zhang<sup>1</sup>, CHENG Li-zhi<sup>2</sup>

(1. ATR Lab, National Univ. of Defense Technology, Changsha 410073, China;

2. College of Science, National Univ. of Defense Technology, Changsha 410073, China)

**Abstract:** Based on the compact support property of popular multi-channel filter banks, a low-memory implementation of generic tree-structured filter banks (TSFBs), called the stripe-based tree-structured filter bank (SBFB), is presented by utilizing the first-in first-out (FIFO) buffer technique. The SBFB has two advantages: (1) It produces exactly the same sub-band coefficients as the conventional global implementation of TSFB does, while the memory budget is unrelated to the image height and only dependent on the image width and the TSFB adopted. As a result, the memory budget of SBFB is only a fraction of that of the global implementation; (2) It generates integral parent-children trees (PCTs), which are completely contained in its buffers. The SBFB lends itself to PCT-based coders in that no intermediate buffering is needed between the SBFB and the coder. By analyzing the dataflow of different levels of TSFB, the relation between the samples and the sub-band coefficients is also attained, by which the correctness of the SBFB is strictly proved.

**Key words:** tree-structured filter bank; low memory implementation; parent-children tree

图像压缩一般包括变换与编码两个部分。当前, 大部分图像压缩方法使用的变换可通过树状多带滤波器组 (Tree-structured Filter Bank, TSFB) 实现。例如, 离散小波变换 (Discrete Wavelet Transform, DWT) 就是一类特殊的 TSFB, 其所有分解层使用同一个具有正则性的 2 带滤波器组<sup>[1]</sup>。Tran 等<sup>[2]</sup> 引入了一个 4 层 TSFB (第 1 层为一个 8 带滤波器组, 其余分解层均为 DWT 2 带滤波器组); 对于细节丰富的图像, 该 TSFB 的性能优于 DWT。一般而言, 图像的 TSFB 分解需要缓存整幅图像, 其内存需求至少与图像的像素数目相当, 因此在资源受限的硬件平台上很难实现高分辨率图像的分解。TSFB 的高内存需求已成为阻碍基于 TSFB 的图像压缩方法普及的一个瓶颈。解决该瓶颈最简单的方案是: 将原始图像分割成较小的不相交的子块, 每个子块单独进行分解。该方案可以归结到单个子块的处理上来, 其内存需求与子块像素数目量级相当。但是, 相对于全局变换, 该方案在块边界处生成的子带系数存在失真。对于图像压缩而言, 该失真表现为低比特率下明显的分块效应 (Blocking Artifacts)。此外, 由于各子块的处理是独立

\* 收稿日期: 2007- 12- 01

基金项目: 国家自然科学基金资助项目 (6057, 3027); 国家 863 高技术研究发展计划资助项目 (2007AA801305)

作者简介: 张雄明 (1978-), 男, 博士生。

的, 编码特定子块时可用的信息相对较少, 因此压缩效率有所降低。

利用常用的多带滤波器组的紧支撑性及 FIFO (First-in First-out) 缓存技术给出了一种具有低内存需求的 TSFB 实现技术 SBFB (Stripe-based TSFB); 该技术生成与全局 TSFB 分解完全相同的子带系数且内存需求仅与图像宽度及所使用的 TSFB 有关, 而与图像高度无关。SBFB 的基本输出单元为基于 TSFB 的图像压缩中应用广泛的数据结构 PCT 结构 (Parent-children Tree, PCT<sup>[3]</sup>), SBFB 与以 PCT 为基本编码实体的图像编码器结合得非常紧密, 变换引擎与编码器之间无需中间缓存。

## 1 符号约定及系统假设

在实际应用中, 数据采集通常是序列进行的。例如, 桌面型扫描仪是以图像行为单位采集数据的<sup>[4]</sup>。当图纸从上至下穿过扫描仪时, 传感器阵列依次产生各个图像行, 因此, 假设系统以 1 样本/单位时间 (对图像而言, 则为 1 行/单位时间) 的恒定速率接收数据。为了描述样本序列  $x[n]$  与子带系数之间的时序关系, 引入时间索引  $t$ ;  $t$  为非负整数且在  $t = n$  时, 系统接收到样本点  $x[n]$ , 但是没有接收到样本点  $x[n+1]$ 。为确定起见, 只考虑如图 1 中所示类型的 TSFB, 其每个中间分解层只有编号为 0 的子带被分解。下文中,  $L, N, M_i$  分别表示样本点总数, TSFB 的分解层数及分解层  $i$  所使用的多带滤波器组的子带数 ( $i = 0, \dots, N-1$ ); 假设  $L$  足够大且能被  $\prod_{i=0}^{N-1} M_i$  整除。考虑到图像压缩的背景, 限定 TSFB 中所有的多带滤波器组均为非冗余变换 (Nonexpansive Transform<sup>[5]</sup>)。  $y_k^{(i)}[n]$  表示分解层  $i$  中第  $k$  路子带对应的子带系数序列 (见图 1) ( $i = 0, \dots, N-1$ ;  $k = 0, \dots, M_i-1$ ); 当  $i \neq N-1$  时,  $y_0^{(i)}[n]$  被继续分解, 故称其为中间低频子带系数。考虑到多带滤波器组在接收一定数量的样本点后才开始生成子带系数, 定义  $M$  带滤波器组的分析时延为其可以生成第 1 组  $M$  个连续的子带系数的最小时间索引。用  $S_i$  表示分解层  $i$  中的  $M_i$  带滤波器组的分析时延, 则  $S_i \geq M_i - 1$  对于  $i = 0, \dots, N-1$  成立; 否则, 该  $M_i$  带滤波器组使用少于  $M_i$  个样本点生成了  $M_i$  个子带系数, 这些子带系数间一定存在冗余, 这与非冗余变换的假设矛盾。

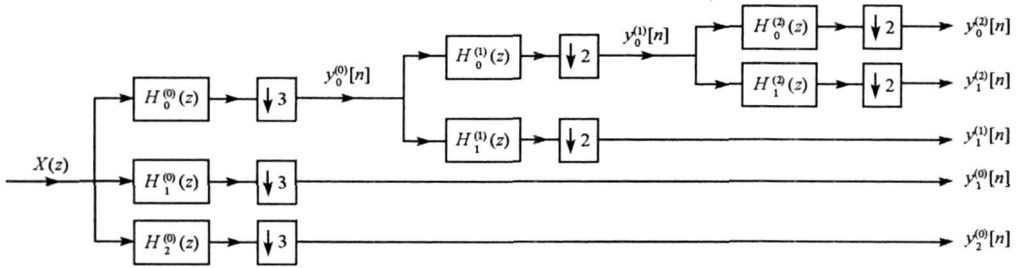


图 1 一个 3 层 TSFB 及其相应的子带系数序列的例子, 其中, 分解层 0 为 3 带滤波器组, 分解层 1、2 均为 2 带滤波器组

Fig. 1 An example of three level TSFB with the first level being a 3-channel filter bank and the rest two being 2-channel filter banks

## 2 PCT 结构及系统时延

狭义的 PCT 结构定义为 DWT 分解后与某个特定空间位置相关的变换系数组成的集合<sup>[3,6-7]</sup>。实际上, 一般的 TSFB 分解产生的子带系数也具有父子依赖关系, 因此可将与特定图像区域相关的 TSFB 子带系数组成的集合称为广义的 PCT 结构。如无特别声明, 下文中的 PCT 均指其广义的版本。对于 1 维信号而言, 一个 PCT 结构包含  $\prod_{i=0}^{N-1} M_i$  个子带系数, 其中来自分解层  $i$  的每个高频子带的子带系数有  $\prod_{j=i+1}^{N-1} M_j$  个 ( $i = 0, \dots, N-1$ )。对宽为  $W$  的图像而言, SBFB 生成大小为  $W \left( \prod_{i=0}^{N-1} M_i \right)$  的 PCT, 其中, 分解层  $i$  的每个高频子带均有  $\prod_{j=i+1}^{N-1} M_j$  行宽度为  $W \left( \prod_{j=0}^i M_j \right)^{-1}$  的子带系数。该 PCT 对应于原图像中  $\prod_{i=0}^{N-1} M_i$  个相邻的图像行组成的图像条带 (Image Stripe)。例如, 对于大小为  $24 \times 24$  的图像进行图 1 所示的 3 层 TSFB 分解, 图 2 给出

了一个 PCT 结构及其在全局分解所得的子带系数中的对应关系。变换引擎的系统时延定义为其生成第 1 个 PCT 时的时间索引。

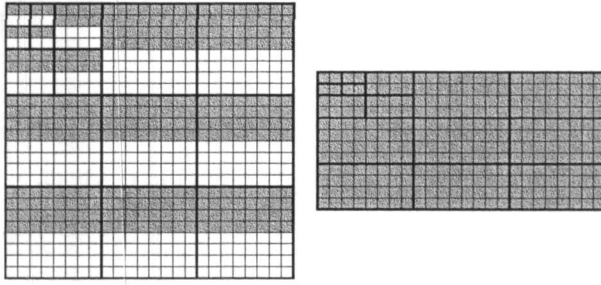


图 2 大小为  $24 \times 24$  的图像的(如图 1 所示的)3 层 TSFB 分解所得的子带系数及其第 1 个 PCT 在全局子带系数中的位置示意图

Fig. 2 Subband representation of a  $24 \times 24$  image and its first PCT for the TSFB given in Fig. 1

### 3 SBF 算法描述

本节主要考虑 1 维信号的 TSFB 分解, 2 维情景可类似处理。在设计 SBF 时, 主要遵循以下 3 个原则: (1) 为了最大限度地减小系统时延, 应尽可能早地产生子带系数; (2) 在对子带系数进行编码时, 其读写应尽可能地安排到系统已经分配的缓存里, 以期望最大限度地减小系统缓存; (3) 子带系数的读取应尽可能地易于操作, 最好对其进行访问时无需复杂的索引。

按功能划分, SBF 使用了 3 类 FIFO 的缓存: (1) 滤波缓存  $F_{\text{buf}}^{(i)}$  ( $i = 0, \dots, N - 1$ ); (2) 同步缓存  $\text{buf}_k^{(i)}$  ( $i = 0, \dots, N - 1; k = 1, \dots, M_i - 1$ ); (3) DC(直流)缓存  $\text{buf}^{(\text{dc})}$ 。滤波缓存  $F_{\text{buf}}^{(i)}$  接收分解层  $i - 1$  传入的中间低频子带系数, 对其进行滤波以产生分解层  $i$  的子带系数。同步缓存  $\text{buf}_k^{(i)}$  接收分解层  $i$  生成的第  $k$  路高频子带系数。DC 缓存  $\text{buf}^{(\text{dc})}$  接收系统生成的 DC 子带系数(即最低频子带系数  $y_0^{(N-1)}[n]$ )。同步缓存应足以容纳系统在生成第 1 个 DC 子带系数之前传输至该缓存中的子带系数。每个中间分解层接收前一个分解层传来的数据, 对其进行处理并将生成的中间低频子带系数传送到下一个分解层。图 3 给出了图 1 所示的 3 层 TSFB 分解的 SBF 的实现流程图, 其中所有缓存及其对应的子带序列也已标明。

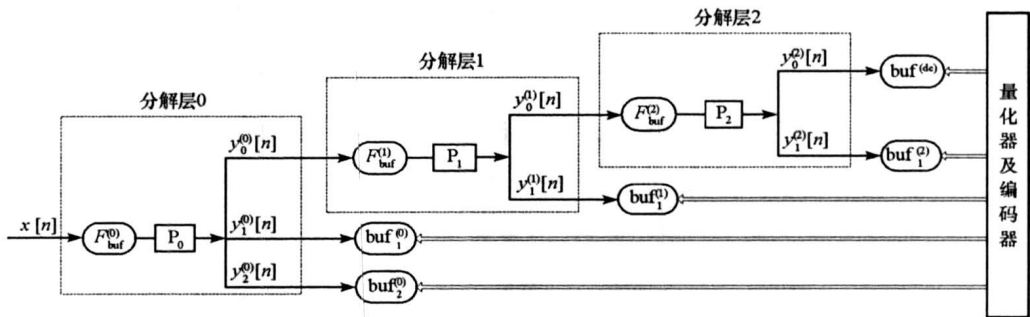


图 3 一个 3 层 SBF 实现流程图, 其中虚线框对应于分解层, 符号  $P_i$  为滤波的组件 ( $i = 0, 1, 2$ )

Fig. 3 The scheme diagram of a three-level SBF, where the dashed boxes denote the decomposition levels

#### 3.1 时间索引与子带系数之间的关系

当分解层  $i$  接收到  $S_{i+1}$  个输入数据时, 生成第 1 组  $M_i$  个子带系数; 自此时刻起, 该分解层每接收  $M_i$  个输入数据, 便生成  $M_i$  个子带系数 ( $0 \leq i \leq N - 1$ )。假设数据处理不会引起时间索引的延迟, 则在 SBF 中, 时间索引  $t$  与子带系数之间的时序关系可总结为定理 1。该定理可通过分析分解层之间的数据流予以证明。定理 1 将被应用于子带系数的辨别过程。

定理1 对于非负整数  $n$ , 当且仅当其可以写成  $\sum_{r=0}^i S_r \prod_{j=0}^{r-1} M_j + m \prod_{j=0}^i M_j$ , 其中,  $0 \leq i \leq N-1$ ,  $m$  为非负整数时, 分解层  $i$  在  $t = n$  时生成一组子带系数  $\{y_k^{(i)}[m] \mid k = 0, \dots, M_i - 1\}$ 。

### 3.2 缓存分析

记  $T_n = \sum_{i=0}^{N-1} S_i \prod_{j=0}^{i-1} M_j + n \prod_{i=0}^{N-1} M_i$ , 其中,  $n$  为任意非负整数。由定理1知, SBFB在  $t = T_n$  时生成索引为  $n$  的DC子带系数(即  $y_0^{(N-1)}[n]$ )。在生成  $y_0^{(N-1)}[0]$  之前, SBFB无法提供PCT。因此从  $t = 0$  到  $t = T_0$ , 分解层  $i$  产生的第  $k$  路高频子带系数被传送到同步缓存  $\text{buf}_k^{(i)}$  中。基于定理1, 这些系数对应的时间索引可以写成  $t = \sum_{r=0}^i S_r \prod_{j=0}^{r-1} M_j + m \prod_{j=0}^i M_j$ , 且  $0 \leq t \leq T_0$ 。该条件等价于  $0 \leq m \leq \sum_{r=i+1}^{N-1} S_r \prod_{j=i+1}^{r-1} M_j$ 。因此可考虑设置同步缓存  $\text{buf}_k^{(i)}$  的大小为

$$\text{size}(\text{buf}_k^{(i)}) = \sum_{r=i+1}^{N-1} S_r \prod_{j=i+1}^{r-1} M_j + 1 \quad (1)$$

其中,  $0 \leq i \leq N-1, k = 1, \dots, M_i - 1$ 。实际上, 由定理1可知, 在  $t = T_n$  之前(包括  $T_n$ ), 对应于分解层  $i$  产生其第  $k$  路高频子带系数的时间索引可以表示为  $\sum_{r=0}^i S_r \prod_{j=0}^{r-1} M_j + m \prod_{j=0}^i M_j$ , 并且满足

$$\sum_{r=0}^i S_r \prod_{j=0}^{r-1} M_j + m \prod_{j=0}^i M_j \leq T_n \quad (2)$$

式(2)可简化为  $m \leq u_k^{(i)}$ , 其中,  $u_k^{(i)} = \sum_{r=i+1}^{N-1} S_r \prod_{j=i+1}^{r-1} M_j + n \prod_{j=i+1}^{N-1} M_j$ 。再考虑到FIFO的同步缓存  $\text{buf}_k^{(i)}$  总是保存其最新接收的  $\text{size}(\text{buf}_k^{(i)})$  个高频子带系数, 可知在  $t = T_n$  时, 位于  $\text{buf}_k^{(i)}$  头部的高频子带系数的索引为

$$m_k^{(i)} = u_k^{(i)} - \text{size}(\text{buf}_k^{(i)}) + 1 = n \prod_{j=i+1}^{N-1} M_j \quad (3)$$

其中,  $0 \leq i \leq N-1, k = 1, \dots, M_i - 1$ 。另一方面, 考虑到当  $0 \leq i \leq N-1$ , 有  $S_i \geq M_{i-1}$ (参看第1节), 于是

$$\text{size}(\text{buf}_k^{(i)}) \geq \sum_{r=i+1}^{N-1} (M_r - 1) \prod_{j=i+1}^{r-1} M_j + 1 = \prod_{j=i+1}^{N-1} M_j$$

总是成立。据此, 并利用式(3), 可以证明下述关于SBFB生成PCT的定理2。

定理2 从  $t = T_0$  起(包括  $T_0$ ), SBFB每接收  $\prod_{i=0}^{N-1} M_i$  个样本点, 就生成一个PCT, 并且构成该PCT的子带系数位于各同步缓存的头部及DC缓存中。进一步地, 在  $t = T_n$  时, SBFB生成索引为  $n$  的PCT。

由式(1)知, 1维SBFB分解所需的同步缓存总量为

$$B_{\text{Synch}}^{(1D)} = \sum_{i=0}^{N-1} \sum_{k=1}^{M_i-1} \text{size}(\text{buf}_k^{(i)}) = \sum_{i=0}^{N-1} S_i \prod_{j=0}^{i-1} M_j - \sum_{i=0}^{N-1} S_i + \sum_{i=0}^{N-1} M_i - N \quad (4)$$

考虑到SBFB在  $t \in [0, T_0]$  时只产生一个DC系数, 故  $\text{size}(\text{buf}^{(\text{dc})}) = 1$ , 因此1维SBFB分解所需的缓存总量为

$$B_{\text{Total}}^{(1D)} = B_{\text{Synch}}^{(1D)} + \sum_{i=0}^{N-1} \text{size}(F_{\text{buf}}^{(i)}) + 1$$

其中, 滤波缓存  $F_{\text{buf}}^{(i)}$  的大小取决于多带滤波器组的实现方法, 将在具体的TSFB实现中给出。

### 3.3 二维SBFB

考虑对宽为  $W$  的图像进行2维TSFB分解, 其中  $W$  可被  $\prod_{i=0}^{N-1} M_i$  整除。此时, 分解层  $i$  有  $M_i^2 - 1$  个高

频子带, 且其第  $k$  路子带对应的同步缓存为  $\text{size}(\text{buf}_k^{(i)})$  行宽度为  $W(\prod_{r=0}^i M_r)^{-1}$  的系数行 ( $i = 0, \dots, N-1$ )。因此同步缓存总量为

$$\begin{aligned} B_{\text{Synch}}^{(2D)} &= \sum_{i=0}^{N-1} W(\prod_{r=0}^i M_r)^{-1} \times (M_i^2 - 1) \times \text{size}(\text{buf}_k^{(i)}) \\ &= W \sum_{i=0}^{N-1} S_i \left[ \prod_{r=0}^{i-1} M_r - \left( \prod_{r=0}^{i-1} M_r \right)^{-1} \right] + W \sum_{i=0}^{N-1} (M_i^2 - 1) \left( \prod_{r=0}^i M_r \right)^{-1} \end{aligned} \quad (5)$$

另一方面, 分解层  $i$  的滤波缓存为  $\text{size}(F_{\text{buf}}^{(i)})$  行宽度为  $W(\prod_{r=0}^{i-1} M_r)^{-1}$  的存储单元, 故滤波缓存总量为

$$B_{\text{Filtering}}^{(2D)} = \sum_{i=0}^{N-1} W \left( \prod_{r=0}^{i-1} M_r \right)^{-1} \times \text{size}(F_{\text{buf}}^{(i)}) \quad (6)$$

再考虑到 DC 缓存只需存储一行宽度为  $W(\prod_{r=0}^{N-1} M_r)^{-1}$  的 DC 变换系数, 可知 2 维 SBWT 所需的缓存总量为

$$B_{\text{Total}}^{(2D)} = B_{\text{Synch}}^{(2D)} + B_{\text{Filtering}}^{(2D)} + W \left( \prod_{r=0}^{N-1} M_r \right)^{-1} \quad (7)$$

#### 4 SFBF 的例子及内存比较

本节将 SFBF 应用于两类特殊的 TSFB: (1)  $N$  层 DWT; (2) 广义双正交叠式变换 (General Lapped Biorthogonal Transform, GLBT) 与 DWT 的混合变换<sup>[2]</sup>。该类 TSFB 有  $(N-2)$  个分解层, 其中第 1 层为 8 带的 GLBT, 其余  $(N-3)$  层均为 2 带 DWT 滤波器组,  $N \geq 4$ 。在这两类 TSFB 中, DWT 采用著名的 CDF 9/7 小波滤波器组<sup>[12]</sup>, 叠式变换采用  $8 \times 16$  GLBT<sup>[2]</sup>。Tran 等<sup>[2]</sup>指出, 频带均匀划分的  $2^k$  带 GLBT 与  $k$  层 2 带 DWT 之间具有相似性: 当结合基于 PCT 的编码器使用时,  $k$  层相邻的 2 带 DWT 滤波器组可用一个  $2^k$  带 GLBT 代替。对于细节丰富 (即含大量高频分量) 的图像, 这种替换通常能得更高的编码效率<sup>[2]</sup>。据此, 可知 8 带 GLBT +  $(N-3)$  层 2 带 DWT 所得的混合变换模拟  $N$  层 DWT 并可直接应用在基于 PCT 的图像编码中。

对于 DWT, 本节也将 SFBF 与两种常用的 DWT 低内存实现方法——LBWT<sup>[8]</sup> 及 Memory-constrained WT<sup>[10]</sup> 进行比较。比较将从缓存总量、系统时延两个方面进行。为公平起见, DWT 均采用提升结构实现<sup>[11]</sup>, 其分析时延与滤波缓存分别为 4, 5。  $8 \times 16$  GLBT 基于格形分解实现时, 其分析时延及滤波缓存均为  $11$ <sup>[13]</sup>。通过式 (5) ~ (7) 可计算出上述两类 TSFB 的内存需求及系统时延, 如表 1 所示。对于  $N$  层 DWT, 该表也列出了 LBWT 及 Memory-constrained WT 的内存需求与系统时延。由表 1 可知, SFBF 相对于 LBWT 及 Memory-constrained DWT 缓存分别减小了  $2^N + 2^{N-3} - 3, 2^N + 2^{N-4} - 4$  行。当分解层数  $N=5$  时, 相对于 LBWT 或者 Memory-constrained WT, SFBF 的系统时延减小了 31 行, 内存消耗分别减小了 18.4% 与 17.9%。

表 1 TSFB 的内存需求及系统时延 (单位: 行)

Tab. 1 Memory requirements and system delays for some typical TSFBs (units: image line)

实现方法	内存需求	系统时延
使用 SFBF 实现的 DWT	$4(2^N - 2^{N-3}) + 1$	$4(2^N - 1)$
$8 \times 16$ GLBT + $(N-3)$ 层 DWT	$4(2^N - 2^{N-3}) - 12.5$	$4 \times 2^N - 21$
LBWT <sup>[8]</sup>	$5 \times 2^N - 2 - 2 \times 2^{N-3}$	$5(2^N - 1)$
Memory-constrained DWT <sup>[10]</sup>	$5 \times 2^N - 3$	$5(2^N - 1)$

## 5 结论

基于常用的多带滤波器组的紧支撑性及 FIFO 缓存技术给出了一种具有低内存需求的 TSFB 实现技术——SBFB。SBWT 的基本数据输出单元为 PCT 结构,可直接应用于基于 PCT 的图像编码算法中。SBFB 的内存需求只与所选用的 TSFB 有关,而与信号的长度或高度无关,非常适合于信号的实时压缩。对于一类特殊的 TSFB——DWT,当使用图像压缩中常用的 CDF 9/7 滤波器组及分解 5 层时,相对于 LBWT 或者 Memory-constrained WT, SBFB 的系统时延减小了 31 行,内存消耗分别减小了 18.4% 与 17.9%。

## 参考文献:

- [1] Taulman D S, Marcellin M W. JPEG2000: Image Compression Fundamentals, Standards and Practice[M]. Boston: Kluwer, 2002.
- [2] Tran T D, Nguyen T Q. A Progressive Transmission Image Coder Using Linear Phase Uniform Filterbanks as Block Transforms[J]. IEEE Trans. Image Processing, 1999, 8(11): 1493- 1507.
- [3] Andreopoulos Y, et al. A Wavelet-tree Image Coding System with Efficient Memory Utilization[C]//Proc. IEEE Int. Conf. ASSP, Salt Lake, Utah, 2001, 3: 1709- 1712.
- [4] Vihel M, Saber E, Trussell H J. Color Image Generation and Display Technologies: An Overview of Methods, Devices, and Research[J]. IEEE Signal Processing Mag., 2005, 22(1): 23- 33.
- [5] Brislawn C M. Classification of Nonexpansive Symmetric Extension Transforms for Multirate Filter Banks[J]. Journal of Applied Computational Harmonic Analysis, 1996, 3: 337- 357.
- [6] Shapiro J M. Embedded Image Coding Using Zerotree of Wavelet Coefficients[J]. IEEE Trans. Signal Processing, 1993, 41(12): 3445- 3462.
- [7] Said A, Pearlman W A. A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees[J]. IEEE Trans. Circuits Syst. Video Technol., 1996, 6(3): 243- 250.
- [8] Chrysafis C, Ortega A. Line-based, Reduced Memory, Wavelet Image Compression[J]. IEEE Trans. Image Processing, 2000, 9(3): 378- 389.
- [9] Xie G, Shen H. Highly Scalable, Low-complexity Image Coding Using Zeroblocks of Wavelet Coefficients[J]. IEEE Trans. Circuits Syst. Video Technol., 2005, 15(6): 762- 770.
- [10] Xu J, Xiong Z, Li S, et al. Memory-constrained 3-D Wavelet Transform for Video Coding without Boundary Effects[J]. IEEE Trans. Circuits Syst. Video Technol., 2002, 12(9): 812- 818.
- [11] Daubechies I, Swelders W. Factoring Wavelet Transforms into Lifting Steps[J]. Journal of Fourier Anal. Appl., 1998, 4(3): 245- 267.
- [12] Cohen A, Daubechies I, Feauveau J. Biorthogonal Bases of Compactly Supported Wavelets[J]. Comm. Pure and Appl. Math., 1992, 45: 485- 560.
- [13] Zhang X, Cheng L, Lu H, et al. A Unified Framework for Low Memory and On-the-fly Implementation of Generic Tree-structured M-channel QMF Banks[J]. Signal Processing, (accepted).