

堆滤波并行流水线实现结构

胡梦佑* 陈钧量

马宗龙

(中山大学无线电电子学系, 广州 510275) (重庆通信学院, 重庆 630035)

摘要 根据信号的广义阈值分解特性, 获得了调整阈值电平的修正二叉树搜索算法, 并提出一种堆滤波的并行流水线式实现结构.

关键词 堆滤波, 修正二叉树, 并行流水线结构

分类号 TN911. 7

线性滤波能有效地抑制加性高斯噪声, 但不能有效地去除脉冲噪声和乘性噪声. 因而, 出现了各种非线性滤波方法, 如中值滤波、排序滤波、数学形态滤波和堆滤波^[1]等. 在既有加性或乘性噪声, 且有高斯或非高斯噪声环境下, 要求有效地滤除这些噪声, 又出现了各种线性-非线性混合滤波方法. Harber 等提出一种基于 $5\ \mu\text{m}$ 工艺的 VLSI 实现方案^[2], 但其芯片面积随输入信号比特数呈指数增长; 基于滤波函数的最小最大表示, Fitch 给出一种芯片面积随输入信号比特数呈线性增长的 VLSI 实现结构^[3]; Chen 给出了一种基于比特串的实现方案^[4], 但该方案是一种串行实现结构. 最近, 作者提出了信号的广义阈值分解算法^[5,6], 并给出了相应的神经网络实现方案. 本文根据信号的广义阈值分解特性和参考 Chen 的方案^[4], 提出一种堆滤波的并行流水线式实现结构.

1 广义阈值分解

中值滤波、排序滤波、数学形态滤波、堆滤波及各种线性-非线性混合滤波都有阈值分解 (Threshold decomposition) 和堆 (Stacking) 特性^[1], 它们的滤波函数都可用正布尔函数 (PBF, Positive Boolean Function) 表示.

1.1 阈值分解算法

非均匀量化可改善小信号电平的量化性能. 非均匀量化一般先对输入信号 x 进行一

收稿日期: 1992-06-27

* 1991 届硕士研究生, 现在重庆通信学院

可逆的非线性变换 T ，再将变换后的输出 y 进行均匀量化，即

$$y = T(x) \tag{1}$$

相应地，在恢复端作另一可逆的非线性变换 G ，并且满足

$$G = T^{-1} \tag{2}$$

通常， T 的形状取决于信号幅度的统计分布 $F(x)$ ；语音信号幅度一般服从拉普拉斯分布^[7]。这时，常选 T 为对数式函数如图 1。

若将 $y \in [-E, E]$ 划分成 $N = 2^k$ 个等间距区段，并将各区段长记为 $\Delta y_1, \Delta y_2, \dots, \Delta y_n$ ，显然存在

$$\Delta y = \Delta y_1 = \Delta y_2 \dots = \Delta y_n = 2E/N \tag{3}$$

相应地， $x \in [-E, E]$ 被划分成 N 个不等间距区段，并将各区段长记为 $\Delta x_1, \Delta x_2, \dots, \Delta x_n$ ， x 和 y 坐标轴上的 $(N + 1)$ 个坐标分点分别记为 $x_i, y_i (i = 0, 1, \dots, N)$ 。显然存在下列迭代式

$$\begin{aligned} x_i &= x_{i-1} + \Delta x_i, \\ x_0 &= -E, \quad i = 1, 2, \dots, N \end{aligned} \tag{4}$$

利用式 (4) 可得

$$x_i = \overline{\Delta x}^T \overline{B}_i - E \tag{5}$$

式中， $\overline{B}_i = (b_N, \dots, b_j, \dots, b_1)^T$

$$\overline{\Delta x} = (\Delta x_N, \Delta x_{N-1}, \dots, \Delta x_1)^T$$

其中 $b_j = \begin{cases} 1, & j \leq i \\ 0, & j > i \end{cases}$

式 (5) 即为广义阈值分解算法。

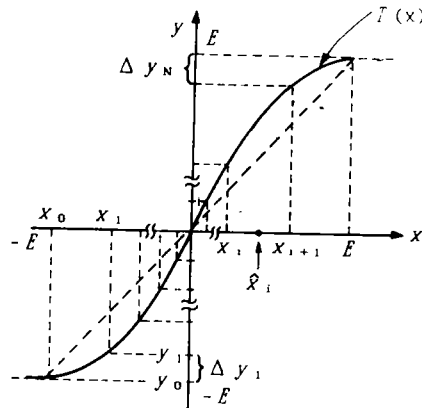


图 1 非线性变换特性

Fig. 1 Nonlinear transform

1. 2 变换函数 T 的形成

若已知 x 的分布函数 $F(x)$ ，则最佳量化应同时满足下列等式^[7]

$$\hat{x}_i = \frac{\int_{\Delta x_i} x dF(x)}{\int_{\Delta x_i} dF(x)}, \quad x_i = \frac{1}{2}(\hat{x}_{i-1} + \hat{x}_i) \tag{6}$$

式中 \hat{x}_i 和 x_i 分别为量化电平值和量化区间边界点。形如 $\int_{\Delta x} G(x) dT(x)$ 的积分称为 Stieltjes 积分。

若未知 x 的分布函数 $F(x)$ ，则可由 Floyd-Max 算法^[7]，通过训练序列进行训练来确定 $\hat{x}_i (i = 0, 1, \dots, N)$ 。

2 实现原理

2.1 串行实现方案

图 2 给出堆滤波的串行实现方案^[4]. 其工作过程是: 首先给定第一级阈值电平 L_1 , 并与 $x_{i(n)}$ ($i=1, 2, \dots, 2M+1$) 相比较, 若 $x_{i(n)}$ 为高, 则相应的 $b_{i(n)}$ 输出为 1; 否则输出为 0. 由各 $b_{i(n)}$ 值及 PBF 的型式, PBF 输出 h_1 . 由 h_1 调整阈值电平为 L_2 , 并再与 $x_{i(n)}$ 比较后输出各 $b_{i_2(n)}$, 同时经 PBF 输出 h_2 . 由 (h_1, h_2) 得到 L_3 , 同理可得 h_3 . 如此循环, 最后得到 h_k . 这时, 输出 (h_1, h_2, \dots, h_k) 为输出的二进制表示.

2.2 修正的二叉树搜索算法

藉广义阈值分解, 可以得到 $(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N)$. 由于 $N = 2^k$, 我们可将 \hat{x}_i 的下标 i 用 k 位二进制数表示. 这样有

$$Y(i)_2 = \hat{x}_i \tag{7}$$

式中符号 $(i)_2$ 为整数 i 的二进制表示.

若以 $\{Y(i)_2\}_1^N$ 为树叶, 可以构成一个 $(k+1)$ 层的二叉树. 图 3 给出 $N=8$ 的修正二叉树.

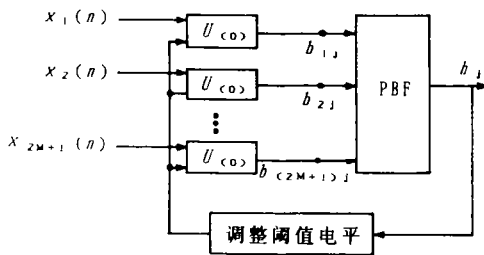


图 2 堆滤波串行实现方案

Fig. 2 A serial realization of stack filtering

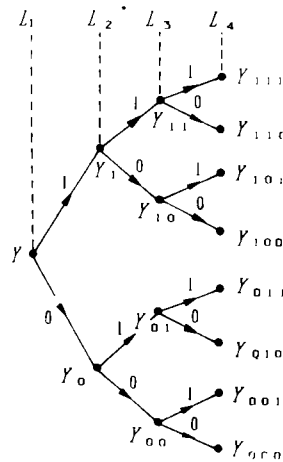


图 3 $N=8$ 时修正二叉树

Fig. 3 Modified binary-tree for $N=8$

图 3 中, $Y_{i_1, \dots, i_k} = \frac{1}{2}(Y_{i_1, \dots, i_k, 1} + Y_{i_1, \dots, i_k, 0})$, 树枝所标值为相应的 h_j 值. 由图可见, 第 i 级阈值电平 L_i 的可能取值为 2^{i-1} 个, 还可注意到第 i 级 Y 的下标正好对应于前 $i-1$ 级的 h 值. 即

$$L_i = Y_{h_1, h_2, \dots, h_{i-1}} \tag{8}$$

可见, 在给定二叉树条件下, L_i 取值完全取决于前 $i-1$ 级 h 输出序列 $(h_1, h_2, \dots, h_{i-1})$ 构成

的逻辑地址所对应的存贮单元的内容. 因此, 可用查表法完成 L_i 的计算.

3 并行流水线实现结构

基于上述实现原理, 我们给出窗宽为 $(2M + 1)$ 堆滤波器计算 h_i 的硬件框图见图 4. 图中

$$x_1(n) = x(n + 1), x_2(n) = x(n), \dots, x_{2M+1}(n) = x(n + 1 - 2M) \quad (9)$$

式中 $x(n)$ 为一维采样输入数据.

以 $M = 1$ 为例, 图 5 给出了整个并行流水线实现结构, 图中, S_i 代表图 4 电路, E 为单位延时算子. 并对边界处数据作如下处理

$$\chi_{(i)} = \chi_{(1)} \quad i < 0 \quad (10)$$

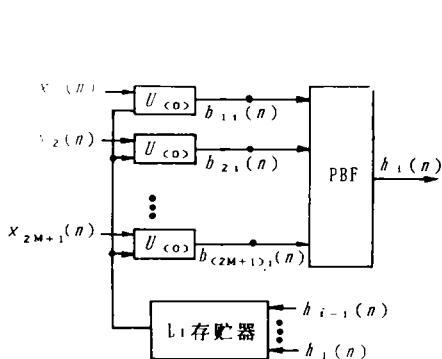


图 4 单个 h 的形成电路

Fig. 4 Block scheme of single h

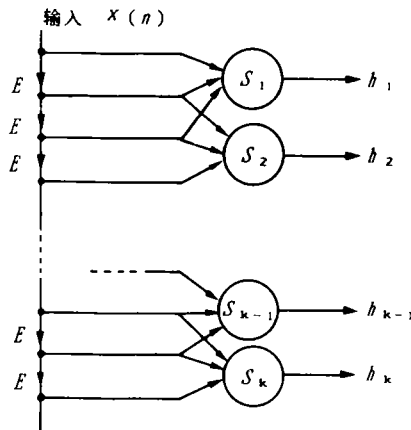


图 5 堆滤波的并行流水线实现结构

Fig. 5 A parallel pipelined architecture of stack filtering

为了研究输出 h 的构成, 下面列表 1 形式 ($k = 8$).

表 1 堆滤波器输出 h 的构成 ($k = 8$)

Tab. 1 A construction of stack filter output for $k = 8$

采样时刻	n	$n + 1$	$n + 2$	\dots	$n + 7$	$n + 8$
h_1	$h_1(n - 1)$	$h_1(n)$	$h_1(n + 1)$	\dots	$h_1(n + 6)$	$h_1(n + 7)$
h_2	$h_2(n - 2)$	$h_2(n - 1)$	$h_2(n)$	\dots	$h_2(n + 5)$	$h_2(n + 6)$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
h_7	$h_7(n - 7)$	$h_7(n - 6)$	$h_7(n - 5)$	\dots	$h_7(n)$	$h_7(n + 1)$
h_8	$h_8(n - 8)$	$h_8(n - 7)$	$h_8(n - 6)$	\dots	$h_8(n - 1)$	$h_8(n)$

由表可见,输出 $H(n) = \{h_1(n), h_2(n), \dots, h_k(n)\}$ 有 k 个采样的延时,但每个采样时刻均有一个输出,即采样时刻 n 的输出为 $H(n-k)$. 堆滤波输出 $S = Y_{h_1, h_2, \dots, h_k}$ 也可用查表法实现.

参 考 文 献

- 1 Wendt P D, Coyle E J, Gallagher N C. IEEE Trans ASSP, 1986, 34(4):898~911
- 2 Harber R G, Bass S C, Neudeck G W. Proc ICASSP '85, 1985, 1396~1399
- 3 Fitch J P. IEEE Trans CAS, 1987, 34(5):553~559
- 4 Chen K. IEEE Trans CAS, 1989, 36(6): 785~794
- 5 胡梦佑,陈钧量,李迎文. 中山大学学报论丛(物理学论文集),1992(2):30~34
- 6 胡梦佑,陈钧量. 军事通信技术,1992(4):39~42
- 7 胡征,杨有为. 矢量量化原理与应用. 西安:西安电子科技大学出版社. 1988

A Parallel Pipelined Architecture of Stack Filtering

*Hu Mengyou * Chen Junliang Ma Zhonglong*

Abstract Based on the generalized threshold decomposition property, a modified binary-tree search algorithm used to adjust the threshold levels is obtained. A parallel pipelined architecture of stack filtering is proposed.

Keywords stack filtering, modified binary-tree, parallel pipelined architecture

* Department of Radio and Electronics, Zhongshan University, Guangzhou 510275