文章编号:1000-582x(1999)05-0013-04

一种中值滤波的快速算法*

靳 斌[1], 郭永彩[1], 杨冠玲[2], 何振江[2], 谢利利[1]
(1. 重庆大学 光电信息工程学院, 重庆 400044; 2. 华南师范大学 应用物理系, 广州 510631)

摘 要: 给出了一种中值滤波的快速算法, 该算法利用了两次中值滤波的窗口内数据的相关性。在中值滤波过程中, 除了第一个中值要用传统排序算法求得, 以后的中值都是通过把新进入窗口元素在前一次排好的序列进行对分查找和内插操作求得。为了便于窗口移动, 设计了一种数据结构, 可以快速用新移入数据覆盖移出数据, 大大减小计数量, 还给出了窗口按"之"字形路线移动的 2 维中值滤波方法。

关键词: 排序; 中值滤波; 图象处理 快速算法 信号处理
中图分类号: TP 273 **文献标识码**: A

在粉尘速度场测量中, 粉尘信息被各种随机噪声干扰(如 CCD 散粒噪声、测试场的热噪声), 所以有必要进行局部平滑处理。平均值滤波是一种可以减小随机噪声影响的常用算法, 但是也会使粉尘图象的边缘模糊化, 为以后处理带来困难。中值滤波是窗口内数据由大到小排列, 取序列中间的值作为均值。它即可以克服噪声影响, 也保护了原始信号的细节信息, 所以中值滤波是粉尘速度场测量中不可或缺的滤波算法。

中值滤波的主要运算是对窗口内数据排序。文献[1]提出把相邻两次的中值滤波合并为一次进行, 只需一次排序, 从而把总的排序次数减少一半。文献[2]是通过对有序列元素的快速查找内插实现中值滤波, 速度比文献[1]有所提高。

笔者提出的中值滤波算法, 是基于前一次排好得到的有序序列, 将新进入窗口元素通过设计的一种数据结构来找到新元素在窗口中的起始位置, 然后通过对分查找和内插, 调整窗口中元素的顺序, 找到新元素在窗口中的新位置, 使窗口成为有序序列, 实现中值滤波, 其效率比文献[2]有较大提高。

1 算法原理及实现

如图 1, 一般的中值滤波是将滤波窗口 $w(0), w(1), \cdots, w(2N)$ (N 为自然数) 输入原始信号, 进行排序, 找到中值, 其比较次数为 $(M-2N)(2N+1)N$. 若设 $M=512$, $N=10$, 则比

* 收稿日期: 1998-10-07
基金项目: 广东省自然科学基金资助项目(970306); 广东省高教厅自然科学重点科研项目资助基金(粤高教科[1997]号)
作者简介: 靳斌(1969-), 男, 重庆人, 重庆大学博士生。从事光电技术及系统领域研究。

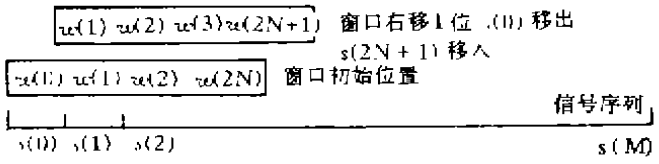

图 1 窗口移动示意图

较次数为(512 − 2 × 10)(2 × 10 + 1)10 = 103 320 次，另外还有相当多的交换运算，可见计算量较大。上述中值滤波的缺点是：窗口每移动一步，更换一个元素就要重新更换窗口数据，重新排序，浪费较大计算量，要想利用原窗口的有序序列就要能找到新元素应放到窗口有序序列的哪一位置，针对该情况设计的窗口数据结构为：

```
struct {
        float Element;
        unsigned char Sequence;
      }w[2N + 1];
unsigned char Map[2N + 1];
```

初始时 Map[2N + 1] = {0, 1, 2, ⋯, 2N}

w[i].Sequence 是进入窗口的顺序，w[i].Sqence = j%(2N+1)，j等于原始信号下标，w[i].Element 是原始信号的内容，Map 是一个映射，将窗口数据顺序映射到窗口数据的下标 Map[w[i].Sequence] = i，这样随着窗口沿信号滑动，新原始数据就可以通过下标 j，i = map[j%2N + 1] 找到应存在窗口中的那一元素 w[i].

同时考虑 w 是一个有序序列，插入窗口的新元素与其左右元素对比就可以知道它该向窗口的哪一方向调整，因此可在部分窗口上采用对分查找算法[3]，如图 2 所示。若新元素的窗口中存放的初始位置为 Position，经对分查找算法得到新元素的位置为 k，当 k < Position 将 w 序列中第 k+1 至第 Position-1 序号元素顺序右移一个序号填充，另将新元素存入 w(k+1)，当 k ≥ Position 将 w 序列中第 Position + 1 至第 k 序号元素顺序左移一个序号填序，另将新元素存入 w(k)，由此调整好的新窗口序列的 w(N) 即为窗口中值，其操作如下。不断移动窗口就可实现对原信号进行中值滤波。

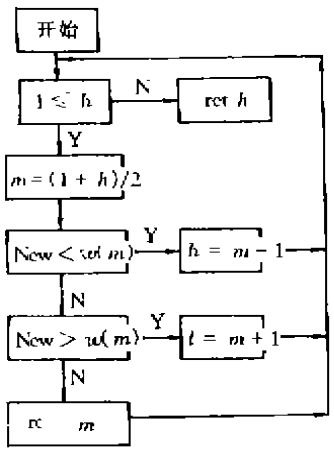

图 2 对分查找算法

文献[2] 中找进入窗口元素的位置用对分查找算法(序列长 2N+1) 在平均情况下比较次数为 $\log_2(n+1)-1$ 次，而本文采用映射方式。文献[2] 找新进入窗口元素的调整后的位置用对分查找算法(序列长 2N+1) 在平均情况下比较次数为 $\log_2(2N+1)+1$ 次；而本文也用对分算法序列长 $n < 2N+1$(n 为部分窗口的长度，$n = 1, 2, 3, \cdots, 2N$)，比较次数 $\log_2(2n+1)+1$ 次，因此可以认为本文算法将比文献[2] 算法快 1 倍以上。

```
当 k < Position
    for (ib = Position, ib < k, ib ++)
    { w[ib].Element = w[ib + 1].Element;
      w[ib].Sequence = w[ib + 1].
      Sequence;
      Rule[w[ib].
      Squence] = ib; }

    w[k].Element = New_ Element;
    w[k].Sequence = j%(2N + 1);
    Map[j%(2N + 1)] = k;
```

```
当 k ≥ Position
    for (ib = Position, ib > k, ib --)
    { w[ib + 1].Element = w[ib].Element;
      w[ib + 1].Sequence = w[ib].
      Sequence;
      Map[w[ib].Squence] = ib + 1; }

    w[k + 1].Element = New_ Element;
    w[k + 1].Sequence = j%(2N + 1);
    Map[j%(2N + 1)] = k + 1;
```

但是本文算法在窗口数据排序过程中不仅要对信号数据进行赋值操作还要对进入窗口的顺序(Sequence)和映射组做赋值操作，其计算略大于文献[2]，而且 Sequence 和 Map 是 Unsigned Char 型数据，也限制了窗口的长度(< 255)。

总之，本文提出的中值滤波的算法的快速特性在于：为了求一个中值，在窗口序列只要做 1 次对分查找算法，元素比较次数 $\log_2(n+1)$，$n < 2N+1$，另外还要至多 n 个元素的顺序移位。无论元素的比较次数和交换次数，都比文献[1、2]的算法少。

3　实验对比

表 1 对传统中值滤波算法、文献[1]算法及文献[2]算法进行了实验对比，其中传统算法用 C.A.R.Hoare 发明的快速排序算法，运算量 $n\log_2 n$，模拟的原始数据是随机浮点数，一个时钟单位为 55 ms，所用计算机为 386。

表 2 是对传统中值滤波算法和本文算法进行了实验对比，模拟的原始数据为随机浮点数，一个时钟单位为 10 ms，所用计算机为 486(频率 66 MHz)。

表 1　几种算法对比

原始数据/个	2048			8192			12288		
窗口长度/个	传统	文 1	文 2	传统	文 1	文 2	传统	文 1	文 2
11	18	11	5	75	40	20	113	60	30
15	28	15	5	112	58	21	168	88	32
25	53	28	7	217	110	28	325	166	42
31	70	36	7	284	144	30	426	216	45

表 2　与传统算法对比

原始数据/个	2048		8192		12288	
窗口长度/个	传统	本文	传统	本文	传统	本文
11	11	2	56	6	82	11
15	22	6	93	6	132	11
25	55	6	219	11	335	15
31	83	6	324	16	489	25

从表1可以看出传统算法与文[2]算法的比值分别为:10,9.47,9.47 从表2可以看出传统算法与本文算法的比值分别为:17.83,20.25,19.56. 可见本文算法比文[2]算法快1倍,这与前面的分析相符。

4 小结

本文提出的中值滤波算法不仅可以用于1维信号处理,还可用于2维图象处理,2维中值滤波与1维中值滤波的主要区别在于滤波窗口不同。例如2维中值滤波用3×3的窗口,窗口每移动一次就有3个新元素进入窗口,因此对每1个进入窗口的新元素都须要按本文方法对窗口进行1次排序,即3·3窗口每移动1步须要做3次排序才有一个中值输出(这如同1维中值滤波中窗口移动3步选取1个中值)。另外当窗口移动到图象右端,可将窗口向下移动一步,再将窗口由右向左移动,如图3所示。通过用这种方法对粉尘图象进行中值滤波处理获得满意效果。

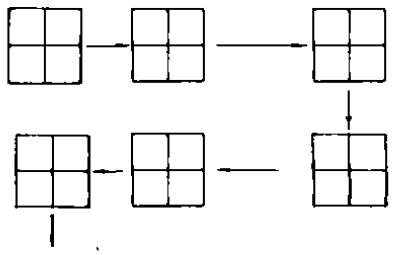

图3 2维窗口移动示意图

参 考 文 献

[1] 吴小培. 一种中值滤波的快速算法[J]. 数据采集与处理, 1995, 10(2): 151~154.
[2] 叶小东, 朱兆达. 中值滤波的快速算法[J]. 信号处理, 1997, 13(3): 227~230.
[3] 潘道才, 陈一华. 数据结构[M]. 成都: 电子科技大学出版社, 1994.
[4] 田捷. 实用图象分析与处理技术[M]. 北京: 电子工业出版社, 1995.

A Fast Median Filtering Algorithm

JIN Bin[1], *GOU Yongcai*[1], *YANG Guanling*[2], *HE Zhenjiang*[2], *XIE Lili*[1]
(1. College of Optoelectronic Engineering, Chongqing University, Chongqing, 400044, China; 2. Department of Applied Physics, Southern China Normal University, Guangdong Guangzhou, 510631)

ABSTRACT: A fast median filtering algorithm basing on the coherence of data in adjacent windows is presented. During median filtering, only the first median value is found by conventional squencing algorithm, the other median values are obtained by bisecting search method and quickinserting new element in sequenced window. For convenience of moving filter window along signal data, a data structure is desinged, which can make the new input element to cover output element and greatly reduce the quantity of computation. A 2-dimensional fast median filtering method by moving filter window along snack path is also introduced.

KEYWORDS: sequencing; median filtering; image processing

(责任编辑 张小强)