

## 时间与空间局部性强弱的量化方法

侯 昉, 黄承慧, 陆寄远

(广东金融学院 计算机科学与技术系, 广东 广州 510521)

**摘 要:** 为了解决虚拟内存空间的管理问题, 基于现有工作集管理算法的参数, 提出了时间与空间局部性强弱量化描述的定义、性质以及可行计算方法. 该方法时间复杂度小, 量化结果反映了程序运行期间工作集的时间与空间局部性的强弱变化.

**关键词:** 工作集; 驻留集; 时间局部性; 空间局部性; 量化方法

**中图分类号:** TP311

**文献标识码:** A

**文章编号:** 1000-7180(2014)06-0048-05

## Quantitative Approaches to Temporal and Spatial Locality Strength

HOU Fang, HUANG Cheng-hui, LU Ji-yuan

(Department of Computer Science and Technology, Guangdong University of Finance, Guangzhou 510521, China)

**Abstract:** In order to manage virtual memory of applications. The locality principle is a fundamental for performance optimization methods such as caching and prefetching. The strength of locality is described in a qualitative way as strong or weak at present. To evaluate the locality strength in a more accurate way, this paper proposes quantitative definitions and approaches of temporal and spatial locality strength respectively. The time complexity of this approach method is acceptable, and the quantitative locality strengths represent the behavior of running program precisely.

**Key words:** working set; resident set; temporal locality; spatial locality; quantitative approach

### 1 引言

局部性原理是研究人员在对程序运行期间对指令和代码访问地址的分布情况进行统计和分析时, 发现的在时间和空间尺度上出现的集中在某一个区间的现象<sup>[1]</sup>. 一般而言, 存在着两种局部性: 时间局部性和空间局部性. 时间局部性是指程序访问的地址序列呈现出周期性的特点, 在某一个时隙  $t$  内访问的地址, 在下一个时隙  $t+1$  内有较大的可能被继续访问; 空间局部性是指程序访问的地址呈现出线性递增的特点, 存放在地址  $a$  的指令或者数据被访问后, 所访问的下一条指令或者数据的地址有较大可能是  $a+1$ .

局部性原理是目前被广泛使用的缓存和预取技术的基础. 时间局部性是较早被利用的局部性现

象<sup>[2]</sup>. 将当前访问的指令或者数据放在高速存储器件中, 以提高再次访问的速度. 近年来, 多核硬件结构<sup>[3]</sup>和并行软件结构<sup>[4]</sup>下的时间局部性现象是研究的重点. 而对于空间局部性的利用则主要表现为在缓存中引入预取手段<sup>[5]</sup>, 将与当前访问指令和数据地址相邻的信息在尚未收到读写请求时就先行读入高速器件, 当读取实际发生时, 提高首次访问的速度. 随着研究的深入, 目前研究人员提出了一些同时考虑时间和空间局部性的缓存管理算法<sup>[6]</sup>.

对于局部性强弱的度量, Vanichpun<sup>[7]</sup>提出了一种比较重复引用次数 (Comparing strength of locality of reference) 的方法, 这里的重复应用次数实际是指的对相同地址块的重复访问的次数. 这种方法存在的问题主要是只能静态地比较固定的页面集合之间的局部性强弱.

**收稿日期:** 2013-07-30; **修回日期:** 2013-09-23

**基金项目:** 广东省自然科学基金 (S2012040007847)

围绕着局部性原理,研究人员设计了大量的缓存和预取管理算法,但是目前对于局部性的强弱,还局限在使用局部性较强或者较弱这样的定性的描述上,缺乏一个量化的描述方法. 本文将对时间和空间局部性的强弱量化方法进行研究,分别对时间局部性和空间局部性提出量化指标及其计算方法.

## 2 时间局部性强弱的量化

时间局部性描述的是在相邻的一个或者若干个时间间隙内,某些地址被重复周期性访问的现象. 在目前的操作系统中,指令和数据在外存储器与主存储器之间的交换是以页面为单位进行的. 一般的 32 位系统默认的页面大小为 4 kB, Intel 的 IA-64 体系结构支持最大的页面大小为 256 MB. 对操作系统的虚拟内存管理模块 MMU 而言,在一个时隙内,只要该页面中的一个地址被访问,则认为该页面被访问. 因此,本文以下的讨论都以页面为单位进行.

除此之外,工作集(Working set)与驻留集(Resident set)也是内存管理中的两个重要概念. 工作集值指的是程序在一次运行(Working)期间所访问的所有页面的集合. 而工作集的页面是在 MMU 的调度下分批次地读入主存储器,驻留(Resident)在主内存中的页面中的指令被执行,页面中的数据被访问.

### 2.1 时间局部性的定量描述

以工作集和驻留集为基础,将采样时隙  $t(0)$ ,  $t(2)$ , ...,  $t(i)$  的驻留集的分别表示为  $S_{t(0)}$ ,  $S_{t(1)}$ , ...,  $S_{t(i)}$ .

定义  $IRS_i$  (Intersection of Resident Sets) 表示时刻  $i$  的驻留集与时刻  $i+1$  的驻留集的交集.  $IRS_i$  可以表示为

$$IRS_0 = S_{t(0)} \cap S_{t(1)}$$

$$IRS_1 = S_{t(1)} \cap S_{t(2)}$$

...

$$IRS_i = S_{t(i)} \cap S_{t(i+1)}$$

由于时间局部性的作用,总会有若干页面存在于相邻的采样时隙的驻留集内,有时候甚至是跨越若干个采样时隙存在于驻留集内. 直观地看,这样的页面越多,也就意味着周期性重复被访问的页面越多,时间局部性越强.

因此,利用  $IRS_i$ , 本文提出时间局部性强弱的量化指标:  $TLS_i$  (Temporal Locality Strength).  $TLS_i$  的定义如下:

记  $|IRS_i| = \text{number of pages in } IRS_i$

$|S_{t(i)}| = \text{number pages in } S_{t(i)}$

$$TLS_0 = \frac{|IRS_0|}{|S_{t(0)}|}$$

$$TLS_1 = \frac{|IRS_1|}{|S_{t(1)}|}$$

...

$$TLS_i = \frac{|IRS_i|}{|S_{t(i)}|}.$$

### 2.2 时间局部性强度 TLS 的性质

TLS 满足以下性质:

(1)  $TLS_i$  的值分布在  $0 \sim 1$  之间.

(2)  $TLS_i$  的值是  $IRS_i$  所包含页面数量的增函数.

这两个性质的证明如下:

命题 1: 对于任意的  $i$ ,  $0 \leq TLS_i \leq 1$ .

证明:

$$\because TLS_i = \frac{|IRS_i|}{|S_{t(i)}|},$$

$$IRS_i = S_{t(i-1)} \cap S_{t(i)}$$

$$\Rightarrow \varnothing \subseteq IRS_i \subseteq S_{t(i)}$$

$$\Rightarrow 0 \leq |IRS_i| \leq |S_{t(i)}|$$

$$\Rightarrow 0 \leq \frac{|IRS_i|}{|S_{t(i)}|} \leq 1$$

$$\Rightarrow 0 \leq TLS_i \leq 1$$

命题得证.

命题 2:  $TLS_i$  的值是  $IRS_i$  所包含页面数量的增函数.

证明:

对于相同的  $S_i$ , 与  $S_{i+1}$ , 若观察到 2 个不同的交集:  $IRS_i$  与  $IRS'_i$

$$TLS_i = \frac{|IRS_i|}{|S_{t(i)}|}, TLS'_i = \frac{|IRS'_i|}{|S_{t(i)}|}$$

$$\text{若 } |IRS_i| \geq |IRS'_i|,$$

$$\text{则 } \frac{|IRS_i|}{|S_{t(i)}|} \geq \frac{|IRS'_i|}{|S_{t(i)}|},$$

$$\text{即: } TLS_i \geq TLS'_i$$

命题得证.

通过以上 TLS 两个性质的证明, 本文的研究发现, 如果相邻时隙内同时存在的页面较少, 则 TLS 的值会接近于 0; 而如果同时存在的页面较多, 则 TLS 的值接近于 1. 以上两个得到证明的性质说明:

(1) TLS 的值分布在一个比较小的范围内;

(2) 同时, 当有较多的页面在相邻两个采样时隙内存在时, 它的值较大, 反之则较小.

如果在相邻的采样时隙中, 发现程序所访问的

页面有很高的重合比例,那么就认为这个程序在这一时刻有较强的时间局部性. 基于时间局部性设计的缓存管理算法 LRU (Least Recently Used) 及其变体往往能取得较好地缓存加速效果. 反之,则时间局部性较弱.

而上述两个性质与我们对时间局部性定性的较强或者较强的描述相一致. 因此,把 TLS 作为量化表征指标是与我们对时间局部性强弱的认识是相容的.

### 2.3 时间局部性强度 TLS 的计算

以上本文对时间局部性 TLS 的定义和性质进行了描述. 如果直接使用 TLS 的定义来进行计算,则需要依次检查两个采样时间周期内某一个页面是否同时存在,若  $n$  为驻留集的页面数量,则这种求交集的操作的时间复杂度是  $O(n^2)$ . 这样的复杂度和时间开销显然不能适应实际需要.

本文的研究发现,利用现在广泛使用的工作集缺页率管理算法 (Page Fault Frequency, PFF) 和 LRU 页面替换管理算法,可以从现有参数中得到时间局部性强度 TLS 的简便计算方法.

PFF 算法的思想是对于当前驻留集设置一个缺页次数阈值(这个阈值实际上是缺页率的倒数),如果在某一个采样周期内发生的缺页次数超过给定阈值,则在下一个采样周期开始前将驻留集扩大. 而在采样周期内,如果发生缺页,需要将所需页面从外存换入驻留集. 换入时,LRU 算法会检测页面的使用位 (use bit),来确定将哪个页面替换出去. use bit 记录了页面在当前采样周期内是否被访问过的信息. 为了提高缓存命中率,页面换入时,会将与之相邻但是还未实际发生访问的 PL (Prefetch Length) 个页面也读入驻留集中,PL 称为预取深度.

利用 PFF 算法和 LRU 算法的数据结构中的参数,可以得到时间局部性强度 TLS 的近似计算方法. 所需要的参数包括 PFF 算法中的缺页发生次数  $k$  以及欲取深度 PL, LRU 算法中的当前驻留集页面总数  $n$  和所有页面 use bit 位的和  $\text{Sig}_{ub}$ .

根据 PFF 和 LRU 算法,在每一次采样周期开始时,所有页面的 use bit 位均置为 0. 如果在本周期内发生了对某一特定页面的访问,则将其对应的 use bit 位置为 1. 显然,在本次采样结束时,所有 use bit 位为 1 的页面,就包括了自从上一次采样周期以来存在于驻留集的页面,也就是所求的时刻  $i$  与时刻  $i+1$  的驻留集的交集. 此外,当缺页发生时,被预取的页面在置入驻留集时,其 use bit 位也是被置为

1 的. 当发生  $k$  次缺页时,共有  $k + k * PL$  个页面被读入,其中  $k * PL$  个页面为预取页面.

因此可以得到

$$S_{\text{use bit}=1} = (S_{t(i)} \cap S_{t(i+1)}) \cup S_{\text{prefetched}}.$$

如果记  $\text{Sig}_{ub}$  为集合  $S_{\text{use bit}=1}$  中的页面数量

则我们可以得到

$$\text{Sig}_{ub} \approx |S_{t(i)} \cap S_{t(i+1)}| + |S_{\text{prefetched}}|,$$

实际上,就是

$$|IRS_i| = \text{Sig}_{ub} - |S_{\text{prefetched}}| = \text{Sig}_{ub} - k * PL,$$

有

$$\begin{aligned} \text{TLS}_i &= \frac{\text{当前周期对前一周期页面的再次访问}}{\text{当前周期页面数量}} \\ &= \frac{\text{number of pages in } IRS_i}{\text{number of pages in } S_{t(i)}} \\ &= \frac{|IRS_i|}{|S_{t(i)}|} \\ &= \frac{\sum_{i=1}^n \text{use bit of Page } i - \text{预取页面数量}}{n} \\ &= \frac{\text{Sig}_{ub} - k * PL}{n} \end{aligned}$$

于是得到了利用现有参数计算的时间局部性量化强度 TLS:

$$\text{TLS}_i = \frac{\text{Sig}_{ub} - k * PL}{n} \quad (1)$$

考虑到在每次采样周期结束时,LRU 算法将要遍历驻留集中的所有页面,将其 use bit 位重置为 0. 对  $\text{Sig}_{ub}$  的计算可以在重置同时完成,因此其增加的时间复杂度是  $O(1)$  的.

## 3 空间局部性强弱的量化

空间局部性描述的是程序运行期间,访问的指令和数据地址是簇集在相邻空间的现象. 这种在相邻地址空间集中程度的强弱,一直以来也缺乏合适的量化指标和量化手段. 本文的研究发现,利用当前缓存算法的预取策略,可以得到空间局部性强弱的量化指标以及相应的可行量化手段.

### 3.1 空间局部性的定量描述

若在  $t(1)$  时刻的当前驻留集为  $S_{t(1)}$ ,则在当前采样周期内,驻留集  $S_{t(1)}$  可以划分为两个子集:本次采样周期内被进程访问过的页面集合  $S_{t(1)\text{visited}}$  和在本次采样周期内没有被进程访问过的页面集合  $S_{t(1)\text{unvisited}}$ :

$$S_{t(0)} = S_{t(0)\text{visited}} \cup S_{t(0)\text{unvisited}}$$

$$S_{t(1)} = S_{t(1)\text{visited}} \cup S_{t(1)\text{unvisited}}$$

...

$$S_{l(i)} = S_{l(i) \text{ visited}} \cup S_{l(i) \text{ unvisited}}$$

由于缓存预取机制的存在,如果在采样周期内发生了缺页现象,则 MMU 会将所缺页面调入当前驻留集,同时,根据预设好的预取深度 PL 将与所缺页面相邻的 PL 个页面也读入驻留集中,这些页面的 use bit 使用位也是被标记为 1 的.当对这些预取的页面的实际访问发生时,因为这些页面已经被先期读入驻留集,则不会发生缺页,避免了需要较大时间代价的缺页中断响应操作.

基于预取工作机制,考察访问过的页面集合  $S_{l(1) \text{ visited}}$ ,它可以进一步分解为两个子集:发生过实际存取的  $S_{l(1) \text{ accessed}}$ ,预取的页面  $S_{l(1) \text{ prefetched}}$ .

$$S_{l(0)} = S_{l(0) \text{ accessed}} \cup S_{l(0) \text{ prefetched}} \cup S_{l(0) \text{ unvisited}}$$

$$S_{l(2)} = S_{l(2) \text{ accessed}} \cup S_{l(2) \text{ prefetched}} \cup S_{l(2) \text{ unvisited}}$$

...

$$S_{l(i)} = S_{l(i) \text{ accessed}} \cup S_{l(i) \text{ iprefetched}} \cup S_{l(i) \text{ unvisited}}$$

显然,因为空间局部性的存在,本周预取的页面在下一周期有很大可能被访问.或者说:本周预取的页面  $S_{l(i) \text{ prefetched}}$  就是空间局部性原理所期望的下一个采样时隙内的访问页面集合  $S_{l(i+1) \text{ visited}}$ .因此定义  $ISS_i$  表示在某一时刻前一采样周期预取页面集合与下一周期的实际访问页面集合的交集:

$$ISS_0 = S_{l(0) \text{ prefetched}} \cap S_{l(1) \text{ accessed}}$$

$$ISS_1 = S_{l(1) \text{ prefetched}} \cap S_{l(2) \text{ accessed}}$$

...

$$ISS_i = S_{l(i) \text{ prefetched}} \cap S_{l(i+1) \text{ accessed}}$$

用  $SLS_k$  (Spatial Locality Strength) 表示第  $k$  个周期的驻留集页面的空间局部性强度,将其定义如下:

记  $|ISS_k| = \text{number of pages in } ISS_k$

$$SLS_0 = \frac{|ISS_0|}{|S_{l(0) \text{ prefetched}}|}$$

$$SLS_1 = \frac{|ISS_1|}{|S_{l(1) \text{ prefetched}}|}$$

...

$$SLS_i = \frac{|ISS_i|}{|S_{l(i) \text{ prefetched}}|}$$

### 3.2 空间局部性强度 SLS 的性质

对于本文提出的空间时间局部性强度量化指标 SLS,可以证明其具备以下性质:

(1)  $SLS_i$  的值分布在  $0 \sim 1$  之间.

(2)  $SLS_i$  的值是预取页面集合与后一周实际读取页面集合交集大小的增函数.

性质证明如下:

命题 3:在采样周期  $i$ ,  $SLS_i$  的值分布在  $0 \sim 1$  之间.

证明:

$$SLS_i = \frac{|ISS_i|}{|S_{l(i) \text{ prefetched}}|}$$

$$ISS_i = S_{l(i) \text{ prefetched}} \cap S_{l(i+1) \text{ accessed}}$$

$$\Rightarrow \varphi \subseteq ISS_i \subseteq S_{l(i) \text{ prefetched}}$$

$$\Rightarrow 0 \leq \text{number of elements in } ISS_i \leq \text{number of elements in } S_{l(i) \text{ prefetched}}$$

$$\Rightarrow 0 \leq \frac{|ISS_i|}{|S_{l(i) \text{ prefetched}}|} \leq 1$$

$$\Rightarrow 0 \leq SLS_i \leq 1$$

命题得证.

命题 4: SLS 在整个观察时域内,是前一周预取页面集合与后一周实际读取页面集合交集大小的增函数.

证明:

若观察到两个不同的交集:  $ISS_i, ISS'_i$

$$SLS_i = \frac{|ISS_i|}{|S_{l(i) \text{ prefetched}}|}$$

$$= \frac{|S_{l(i) \text{ prefetched}} \cap S_{l(i+1) \text{ accessed}}|}{|S_{l(i) \text{ prefetched}}|}$$

$$SLS'_i = \frac{|ISS'_i|}{|S_{l(i) \text{ prefetched}}|}$$

$$= \frac{|S'_{l(i) \text{ prefetched}} \cap S'_{l(i+1) \text{ accessed}}|}{|S_{l(i) \text{ prefetched}}|}$$

$$\text{若 } ISS_{i-1} \geq ISS'_{i-1}$$

$$\text{则 } SLS_{i-1} \geq SLS'_{i-1}$$

命题得证.

由以上命题得到 SLS 的特性:

(1) SLS 分布在  $0 \sim 1$  之间.

(2) SLS 是上一周期 prefetched 子集与下一周期 accessed 子集交集大小的增函数.

### 3.3 空间局部性强度 SLS 的计算

设计并实现预取机制的目的在于:将与当前访问页面相邻的若干个页面在一次缺页中断的服务程序中完成读取.当将来发生对这些页面的实际访问请求时(因为空间局部性的存在,这种情况发生的概念比较高),因为这些页面已经存在于驻留集中,不再需要执行很大代价的缺页服务程序,因此程序的性能得到了提高.

以上空间局部性强度 SLS 的定义与计算同样需要对前后两个采样周期内的页面集合进行求交集的计算,具备  $O(n^2)$  的时间复杂度.本小节讨论对其进行一个可行的代替计算.

在最理想的状况下,因为时间与空间局部性的存在,下一个采样周期内所有需要访问的页面都已经在上一个周期内被读入了内存.也就是说,上一周期的驻留集可以覆盖下一周期的工作集.而在真实情况下,正是因为空间局部性不能够完美地覆盖下一个周期内的工作集,因此会发生缺页现象.如果空间局部性越强,则覆盖的页面越多,发生的缺页次数越少;相反,如果空间局部性越弱,则不能覆盖的页面越少,发生的缺页次数越多.因此,可以将缺页发生的次数或者首次缺页发生的时间作为计算空间局部性 SLS 的变量.表达式为

$$SLS_i = \frac{t_i}{C} \quad (2)$$

式中,  $t_i$  表示在第  $i$  个周期内发生首次缺页时,距本次采样开始经过的时长;  $C$  表示采样周期的时间长度.

如果上一周期的驻留集 100% 覆盖了当前周期的工作集,则一直不会发生缺页,时间变量  $t_i$  会一直

增长到等于  $C$  为止,计算得到  $SLS = 1$ ,此时我们认为具备最强的空间局部性;另一种极端情况下,上一个周期的驻留集与当前工作集完全没有交集,则第一次页面访问就会发生缺页,此时  $t_i \approx 0$ ,相应地,  $SLS = 0$ . 其他的情况均处于这两种情况之间,因此,式(2)的计算结果与我们对 SLS 的定义及其具备的性质是一致的.

对 SLS 的计算使用的表达式只涉及对两个已知量进行一次除法运算,因此其增加的时间复杂度也是  $O(1)$  的.

#### 4 实验

为了考察以上指标是否能够反映局部性强弱,对程序运行期间的地址访问情况及其相关指标进行了计算.实验对程序运行访问的地址进行了记录,并使用本文提出的量化指标计算方法对局部性强弱进行了计算.结果如图 1 所示.

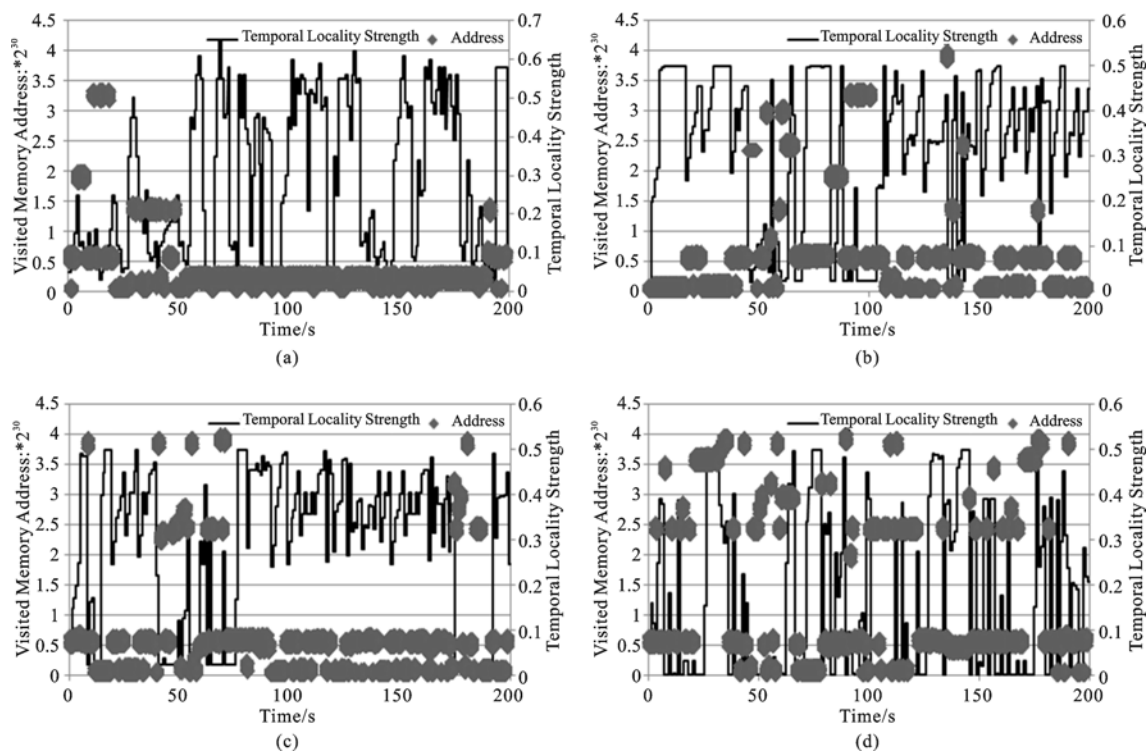


图 1 访存地址与局部性强弱

图 1(a), (b), (c) 和 (d) 中的左纵轴是虚拟地址空间,其范围为  $0 \sim 2^{32}$  字节地址;右纵轴为计算出的局部性强度量化值,其范围为  $0 \sim 1$ . 从图中可以看出,当程序访问的地址集中在一个较小的访问内时,计算得到的量化局部性强弱值较大(相对接近于 1),而当访问地址变化剧烈时,局部性量化值较小(相对接近于 0).

#### 5 结束语

本文的研究提出了时间与空间局部性的量化描述指标的定义、性质及利用现有工作集管理和页面替换算法参数进行局部性强弱量化指标计算的方法.方法简单、直接,复杂度较低.

(下转第 56 页)

大小分别为(3,7)、(5,9)、(5,11)、(7,13)、(7,15)和(9,17)进行实验,其中(3,7)表示‘窗口 1’大小为 3

×3 和‘窗口 2’的大小为 7×7. 实验结果表明,本文方法去噪效果较其他两种方法有大幅提高.

表 1 SMF,NDBA 和 TSDA 方法处理不同噪声密度(70%,80%,90%)图像 lena 的三种数据指标值

| Noise Density | MSE    |        |        | PSNR   |        |        | IEF   |       |       |
|---------------|--------|--------|--------|--------|--------|--------|-------|-------|-------|
|               | SMF    | NDBA   | TSDA   | SMF    | NDBA   | TSDA   | SMF   | NDBA  | TSDA  |
| 70%           | 31.101 | 18.931 | 17.902 | 21.040 | 26.005 | 26.563 | 2.868 | 4.712 | 4.982 |
| 80%           | 46.611 | 28.551 | 22.393 | 16.994 | 21.896 | 24.325 | 2.187 | 3.571 | 4.553 |
| 90%           | 78.646 | 60.710 | 27.668 | 11.763 | 14.352 | 22.210 | 1.454 | 1.884 | 4.134 |

## 5 结束语

提出了一种去除高密度椒盐噪声的 TSDA 方法. 实验结果表明,TSDA 方法在保持图像边缘的同时,其去除高密度椒盐噪声的能力优于其他两种方法,TSDA 方法较之 SMF 方法和 NDBA 方法在检验指标 MSE、PSNR 和 IEF 上都有较大改进.

### 参考文献:

- [1] Astola J, Kuosmanen P. Fundamentals of nonlinear digital filtering[M]. Florida:Boca Raton, CRC, 1997.
- [2] Pomalaza-Racz C A, Macgille C D. An adaptive non-linear edge preserving filter [J]. IEEE Trans. Acoustics, Speech, Signal Process, 1984 ( ASSP-32 ): 571-576.
- [3] Chan R H, Chung W H, Nikolova M. Salt and pepper

noise removal by median type noise detectors and detail preserving regularization[J]. IEEE Transactions on Image Processing, 2005, 14(10):1479-1485.

- [4] Zhang S, Karim M A. A new impulse detector for switching median filters [J]. IEEE Signal Process, Letters, 2002, 9 (11): 360-363.
- [5] Kenny K V T, Nor A M. Noise adaptive fuzzy switching median filter for salt and pepper noise reduction[J]. IEEE Signal Processing Letters, 2010, 17(3):281-284.
- [6] Srinivasan K S, Ebenezer D. A new fast and efficient decision-based algorithm for removal of high-density impulse noises [J]. IEEE Signal Processing Letters, 2007, 14(3):189-192.

### 作者简介:

王际朝 男,(1979-),博士研究生,讲师. 研究方向为图像处理、数值计算等.

(上接第 52 页)

### 参考文献:

- [1] Denning P J. The locality principle[J]. Communication of. ACM, 2005, 48(7):19-24.
- [2] Jeffrey R Spirn, Peter J Denning. Experiments with program locality[C] // Proceedings of the Fall Joint Computer Conference. Anaheim, California: ACM, 1972(part D):611-621.
- [3] Jaleel A, Borch E, Bhandaru M, et al. Achieving non-Inclusive cache performance with inclusive caches: temporal locality aware (TLA) cache management policies[C] // Proceedings of Annual IEEE/ACM International Symposium on Micro Architecture (MICRO 2010 ). Los Alamitos, CA: IEEE Computer Society, 2010:151-162.
- [4] Tran Ngoc Minh, Lex Wolters. Modeling parallel system workloads with temporal locality[C] // Proceedings of International Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP 2009). Heidelberg, Berlin: Springer, 2009: 101-115.
- [5] Torrellas J, Lam H S, Hennessy J L. False sharing

and spatial locality in multiprocessor caches[J]. IEEE Transactions on Computers, 1994, 43(6):651-663.

- [6] Xiaoning Ding, Song Jiang, Feng Chen. A buffer cache management scheme exploiting both temporal and spatial localities[J]. ACM Transaction on Storage, 2007, 3(2):5.
- [7] Vanichpun Somparing. Strength of locality of reference - popularity majorization and some folk theorems[C] // Proceedings of Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004). Toronto, Canada, IEEE Computer Society, 2004(2):838-849.

### 作者简介:

侯 昉 男,(1975-),博士,讲师. 研究方向为计算机体系结构、计算机存储系统.

黄承慧 女,(1976-),博士,讲师. 研究方向为文本挖掘、信息检索.

陆寄远 男,(1977-),博士,系统分析师. 研究方向为计算机图形图像.