

基于队列式缓存结构的视频图像存储算法

郝 伟^{1,2} 苏秀琴¹ 杨小君^{1,2} 李 哲^{1,2} 吴慧莲³

(1 中国科学院西安光学精密机械研究所, 西安 710068)

(2 中国科学院研究生院, 北京 100039)

(3 河北工程学院理学院, 河北邯郸 056038)

摘 要 针对视频图像数据实时存储过程中普遍存在的丢帧问题, 通过深入分析典型的存储算法, 找出实时存储过程中视频图像数据丢失的原因; 计算机在连续存储相同大小的文件时, 存储速度在一定的范围内波动. 本文采用队列式缓存结构有效地消除了存储速度波动所带来的影响, 保证了在数据流速小于磁盘平均存储速度的前提下, 高速视频图像数据的实时存储, 解决了丢帧问题. 实践验证了该算法的可行性与可靠性.

关键词 视频存储; 视频图像; 队列式缓存; 多线程; RAID 0

中图分类号 TN317.4 **文献标识码** A

0 引言

随着科学技术的发展, 计算机外部设备不断增加, 而且这些设备能够在短时间内产生大量的数据, 需要计算机实时存储到磁盘上, 以用于事后分析. 计算机技术发展突飞猛进, 磁盘存储速度和容量不断提高, 但是仍然不能满足人们的需求. 如何把这些设备产生的高速数据流实时地存储到计算机磁盘中已成为人们研究的一个重要课题. 本文主要针对视频图像实时存储进行了研究. 首先通过采用典型的乒乓式采集存储结构发现存储过程中丢帧现象极为普遍, 原因是计算机在连续存储同样大小的数据流文件过程中, 其存储速度在一定的范围内波动, 从而造成了实时数据的丢失, 不能保证高速数据流的实时存储. 因此对这种存储结构进行了改进, 采用更为可靠的队列式缓存结构, 有效地解决了视频图像实时存储过程中的丢帧问题, 保证了采集来的高速数据流实时存储的稳定性^[1~6].

1 典型存储算法分析

视频图像采集与存储系统中普遍采用双缓存页技术(即乒乓式缓存结构), 原理如图 1. 图中 InBuf 用来存储当前帧图像数据, Buffer1 和 Buffer2 是两页缓存器, 其中一页用来接收采集数据流, 称为缓存页, 另一页用来将数据记录到硬盘上, 称为记录页. 缓存页写满数据后发出换页请求信号, 记录页将数据记录到磁盘后则发出换页允许信号. 换页请求信号和换页允许信号同时有效时产生换页信号, 它触发换页逻辑将两页缓存器相互交换, 原记录页成为缓存页, 原缓存页成为记录页. 开始采集时, 两页缓

存器均空, 换页请求信号无效, 换页允许信号有效. 当采集数据流写满缓存页后, 换页请求信号变为有效, 两页缓存器相互交换, 交换后换页请求信号和换页允许信号均变为无效. 接着, 记录页向磁盘写数据, 同时缓存页接收采集数据流. 当记录页的数据记录完后发出换页允许信号, 然后重复上述过程直到采集过程结束. 显然, 要保证数据不丢失, 换页允许信号一定要比换页请求信号先有效, 即存储的速度一定要大于采集的速度.

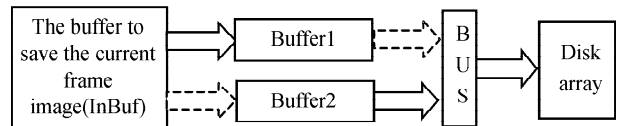


图 1 双缓存页技术原理

Fig. 1 The principle map of two RAM page technique

通常视频采集与存储系统采用多线程来实现, 主要由采集线程和存储线程组成. 采集线程为主线程主要是将采集来的视频图像数据存入计算机内存当中; 存储线程主要是将采集线程存入内存中的图像数据存入磁盘中. 在采用乒乓式缓存结构的情况下两个线程的时序关系如图 2.

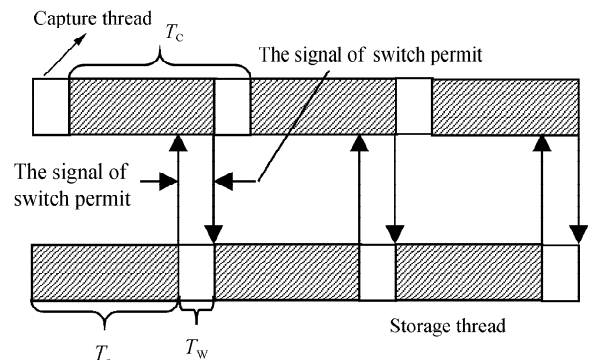


图 2 乒乓结构时序关系

Fig. 2 The sequence of the ping-pong structure

要保证每次的换页允许信号比换页请求信号先

有效,也就意味着存储相同大小的视频图像所用的最大时间要小于最小的采集时间. 通常视频图像的采集时间是固定的,而存储时间不固定,有一定的波动,主要原因就在于计算机的磁盘有机械运动. 对于相同大小的文件,由于寻道时间的不同造成存储时间的差异. 所以说采用乒乓式存储结构的情况下,要想使采集来的视频图像实时并且不丢帧的存储到计算机磁盘中,就必须保证磁盘存储每帧图像所用的最大时间 t_{max} 小于采集周期 T_C (即 $t_{max} < T_C$). 而 t_{max} 除了主要依赖于存储系统的硬件性能外还带有很大的随机性,在实际应用过程中很难保证 $t_{max} < T_C$,所以采用这种存储结构很难避免丢帧问题^[4,5],见图 3.

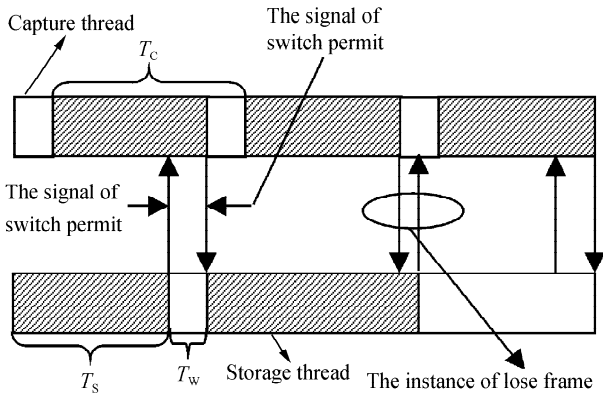


图 3 丢帧时序关系
Fig. 3 The sequence of the lose frame

2 磁盘存储试验及结果分析

基于对乒乓式存储结构的分析,主要针对视频采集与存储系统的磁盘存储速度波动问题,磁盘存

储模拟试验如下:在一定的硬件基础上进行多次试验,将一幅图像读入内存然后重复的写入计算机磁盘 n (n 较大)次,记录了如下数据:

- ① 最小存储时间 t_{min} ,
- ② 最大存储时间 t_{max} ,
- ③ 总的存储时间 t_{count} , 然后计算平均存储时间: $t_{even} = \frac{t_{count}}{n}$.

试验结果如下: $t_{max} > 5t_{min}$, $t_{max} > 3t_{even}$, 由此可见采用乒乓式存储结构要保证 $t_{max} < T_C$ 比较困难,而且不能保证系统的稳定可靠,还造成硬件系统资源的浪费. 所以必须设计新的缓存结构以消除磁盘存储速度的波动带来的影响,保证视频图像实时不丢帧的存储到计算机磁盘中.

3 队列式缓存结构存储算法设计

3.1 队列式缓存结构

基于以上分析和试验来看,采用乒乓式缓存结构必须保证 $t_{max} < T_C$ 才能保证采集来的图像信息实时地存入计算机磁盘中,由于 t_{max} 带有很大的随机性,系统不能够稳定可靠的工作,因此希望改进缓存结构,使得在保证 $t_{even} < T_C$ 的情况下就能够使采集来的图像实时存入磁盘. 通过对乒乓式缓存结构的分析发现在不丢帧的情况下 $T_w > 0$,也就是说存储线程有等待时间,这无疑是一种浪费,所以把 $T_s < T_C$ 情况下的等待时间 T_w 充分利用,在充分的内存资源中开发较大的缓存,改变线程间的同步控制,建立一个像蓄水池一样的缓存结构,从而使记录线程能够不丢帧地完成记录任务. 在这里采用队列(先进先出)式缓存结构,如图 4.

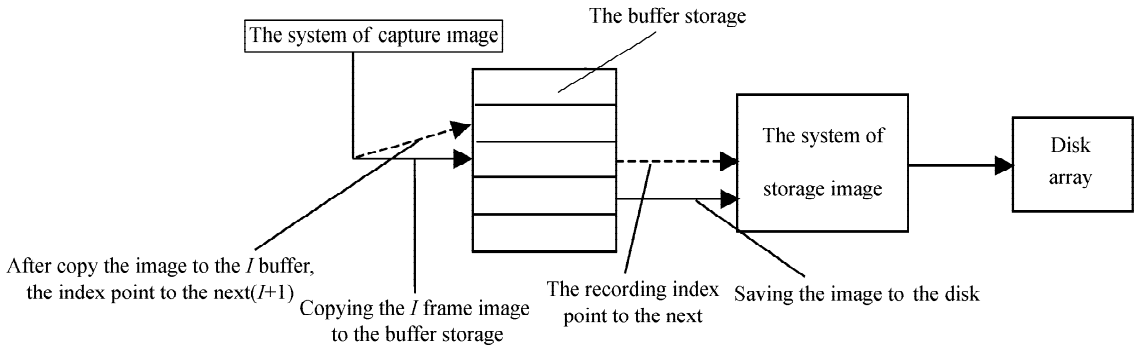


图 4 队列式缓存结构
Fig. 4 Sketch map of the queue buffer

缓存区总数 n 的大小根据 $t_{max} > 3t_{even}$ 设定为一个大于 3 的数即可,而且 n 越大系统越可靠.

3.2 存储算法设计及 VC 实现

如图 4,图像采集系统将采集来的图像存入队列式缓存结构的每一帧缓存中. 当第 I 帧图像数据拷贝到缓存后,主线程将指向队列式缓存结构的指

针 $RecordIndex_i$ 加 1, 然后对队列式缓存结构缓存图像总数 ($StreamBufMax$) 取余数, 同时标记当前缓存中未能存入磁盘的图像帧数 $StreamCount$ 加 1,

$$RecordIndex_i = RecordIndex_i \% StreamBufMax ;$$

$$StreamCount ++ ;$$

即当指针 $RecordIndex_i$ 指向队列尾时,采完当

前帧数据后将指针 RecordIndexi 指向队列头, 循环使用数组缓存来实现队列式缓存结构. 记录线程指针(Indexi)变化和主线程相同. 采用队列式缓存结构的程序控制如下:

1) 主线程开始记录后往缓存中的第 1 帧送数据, 这时标志缓存中未能存入磁盘的图像帧数的公共变量 StreamCount 为 0, 记录线程处于等待事件(m_hRecordEvent)状态;

2) 当主线程存完当前帧后, 将指针变化如下: RecordIndexi = StreamBufMax (RecordIndexi + 1)%; 并锁住互斥信号量(Hmutex)将标志缓存中未能存入磁盘的图像帧数的公共变量 StreamCount 加 1; 再释放互斥信号量(Hmutex);

```
::WaitForSingleObject(HMutex, TimeOut);
```

```
StreamCount++;
```

```
::ReleaseMutex(HMutex);
```

然后发送事件: SetEvent(m_hRecordEvent); 当下一帧数据到来后存入 RecordIndexi 所指向的缓存中;

3) 这时记录线程接收到记录事件(m_hRecordEvent), 结束等待进入记录程序, 将该帧数据存入磁盘中, 然后将其在队列式缓存结构取数据帧的指针操作如下:

```
Indexi = (Indexi + 1) % StreamBufMax;
```

并锁住互斥信号量(Hmutex)将标志缓存中未能存入磁盘的图像帧数的公共变量 StreamCount 减 1; 再释放互斥信号量(Hmutex);

```
::WaitForSingleObject(HMutex, TimeOut);
```

```
StreamCount--;
```

```
::ReleaseMutex(HMutex);
```

这时判断 StreamCount 是否为零, 如果是则结束循环, 并重置记录事件:

```
ResetEvent(m_hRecordEvent);
```

进入事件等待状态; 如果 StreamCount 不为零, 则表示主线程已将下一帧数据送到了队列式结构缓存中, 则记录线程继续记录缓存中的数据.

4) 这里还要说明的是记录线程是由主线程来控制的, 当记录完毕后则将公共变量 EndThread 赋值为真, 记录线程将队列式缓存中的数据全部存入磁盘后就自动结束线程.

按照以上控制进行存储, 当某一帧写入磁盘的时间大于采集周期 T_c 的情况下, 采集来的下一帧图像保存入缓存中不会造成数据丢失, 记录线程等到写入磁盘的时间小于采集周期 T_c 的时候利用原来乒乓式缓存结构的等待时间继续将缓存中的图像写入磁盘, 从而保证了在 $t_{even} < T_c$ 的前提下, 系统

能够将采集来的图像数据实时的存入计算机磁盘中, 使系统能够稳定可靠的工作. 有关事件和互斥信号量等机制请参考文献[1~3].

3.3 算法实际应用及效果

此存储算法已经多次应用于工程项目中的视频存储系统. 通过多次试验, 对实际使用缓存数的记录与分析发现: 视频存储系统由于计算机内部因素丢帧的概率几乎为零, 即用该种算法能够保证把采集进来的所有图像实时存储到计算机磁盘中. 该算法已经随同项目交付用户方使用, 曾多次参加任务, 未发现丢帧现象, 由此可见该算法是稳定可靠的.

4 结论

视频记录系统的记录机制是采用如上所述的事件触发机制和队列式缓存结构来完成实时记录工作的. 实践证明, 该种机制使记录数据流连续起来, 将记录线程的等待时间 T_w 充分利用, 能够不丢帧地完成视频记录工作, 将采集来的图像数据存到计算机磁盘中, 而且该种机制稳定可靠不会造成数据丢失和死机. 总之只要保证磁盘的平均存储时间小于采集周期 T_c , 就能够不丢帧的将采集来的数据记录到磁盘中. 经过采用队列式缓存结构, 在硬件一定的前提下充分利用系统资源, 从而解决了视频存储系统中普遍存在的丢帧问题, 保证了系统稳定可靠地工作.

参考文献

- 王险峰, 刘宝宏. Windows 环境下多线程编程原理与应用. 北京: 清华大学出版社, 2002. 137~335
Wang X F, Liu B H. The programme theory and application of multithreading in Windows. Beijing: Tsinghua University Press, 2002. 137~335
- 任哲. MFC windows 应用程序设计. 北京: 清华大学出版社, 2004. 205~224
Ren Z. MFC windows the design of application program. Beijing: Tsinghua University Press, 2004. 205~227
- Jon Bares and Tim Tompkins 著, 石祥生 译. Visual C++ 使用指南. 北京: 电子工业出版社, 1999. 431~458
Jon Bares and Tim Tompkins. Visual C++ use guide. Beijing: Publishing House of Electronics Industry, 1999. 431~458
- 田雁, 曹剑中, 杨小军. 间歇式高速电视 CCD 视频图像实时存储系统. 光子学报, 2002, 12(12): 1549~1552
Tian Y, Cao J Z, Yang X J. Acta Photonica Sinica, 2002, 12(12): 1549~1552
- 达选福, 张伯珩, 边川平. 高速 CCD 图像数据存储技术. 光子学报, 2003, 11(11): 1393~1395
Da X F, Zhang B H, Bian C P. Acta Photonica Sinica, 2003, 11(11): 1393~1395
- 李清军. 基于硬盘的视频实时存储方法的研究. 光学精密

A Storage Arithmetic of Video Image Based on Queue Buffer Structure

Hao Wei^{1,2}, Su Xiuqin¹, Yang Xiaojun^{1,2}, Li Zhe^{1,2}, Wu Huilian³

¹ Xi'an Institute of Optics and Precision Mechanics of CAS, Xi'an 710068

² Graduate School of CAS (The Chinese Academy of Sciences), Beijing 100039

³ The Science School, Hebei Engineering College, Handan 056038

Received date: 2005-06-20

Abstract Aimed at the ubiquitous problem of losing frame in the course of video image real-time storing, the typical storage arithmetic has been analyzed thoroughly in this paper. Then, the cause of losing video image in the process of real-time storing was found. When the same size files were saved continuously to the disk of the computer, the storage rate was fluctuant in stated range. Through adopting the queue buffer structure, the influence of the storage rate fluctuating was avoided effectively; consequently the real-time storage of high-speed video image was ensured in the condition that the rate of data flow is less than the mean rate of storage. Finally the problem of losing frame was solved. By experiment, the feasibility and reliability of the arithmetic was proved.

Keywords Video Record; Video Image; The queue buffer; Multithreading; RAID 0



Hao Wei was born in 1979 and graduated from Shenyang Institute of Aeronautic Engineering in 2001 with his Bachelor of Application of computer, then acquired Master degree of Engineering in 2004 at Xi'an Institute of Optics and Precision Mechanics of CAS. Since 2004, he has been pursuing his Doctor. degree at The Graduate School of The Chinese Academy of Sciences in Communication & Information System. Now his research jobs are Image Processing and High speed video signal processing.