

用做出良好的选择。

这里的另一个因素是图7-20a中的磁盘存储管理更加复杂,因为存放一帧需要找到大小合适的一连串连续的磁盘块。理想情况下,这一连串磁盘块不应该跨越一个磁道的边界,但是通过磁头偏斜,这一损失并不严重。然而,跨越一个柱面的边界则应该避免。这些要求意味着,磁盘的自由存储空间必须组织成变长孔洞的列表,而不是简单的块列表或者位图。与此相对照,块列表或者位图都可以用在图7-20b中。

在上述所有情况下,还要说明的是,只要可能应该把一部电影的所有块或者帧放置在一个狭窄的范围之内,比如说几个柱面。这样的存放方式意味着寻道可以更快,从而留下更多的时间用于其他(非实时)活动,或者可以支持更多的视频流。这种受约束的存放可以通过将磁盘划分成柱面组来实现,每个组保持单独的空闲块列表或位图。如果使用孔洞,可能存在于一个1KB孔洞的列表、一个2KB孔洞的列表、一个3KB到4KB孔洞的列表、一个5KB到8KB孔洞的列表等。以这种方法在一个给定的柱面组中找到一个给定大小的孔洞是十分容易的。

这两种方法之间的另一个区别是缓冲。对于小块方法,每次读操作正好读取一帧。因此,采用简单的双缓冲策略就工作得相当好:一个缓冲区用于回放当前帧,另一个用于提取下一帧。如果使用固定大小的缓冲区,则每个缓冲区必须足够大以装得下最大可能的I帧。另一方面,如果针对每一帧从一个池中分配不同的缓冲区,并且当帧在被读入之前其大小未知,那么对于P帧和B帧就可以选择一个较小的缓冲区。

使用大磁盘块时,因为每一块包含多个帧,并且在每一块的尽头还可能包含帧的片段(取决于选定前面提到的是哪种选择),因而需要更加复杂的策略。如果显示或传输帧时要求它们是连续的,那么它们就必须被复制,但是复制是一个代价高昂的操作,应该尽可能避免。如果连续性是不必要的,那么跨越块边界的帧可以分两次送出到网络上或者送出到显示设备上。

双缓冲也可以用于大磁盘块,但是使用两个大磁盘块会浪费内存。解决浪费内存问题的一种方法是使用比为网络或显示器提供数据的磁盘块(每个数据流)稍大一些的循环传输缓冲区。当缓冲区的内容低于某个阈值时,从磁盘读入一个新的大磁盘块,将其内容复制到传输缓冲区,并且将大磁盘块缓冲区返还给通用池。循环缓冲区大小的选取必须使得在它达到阈值时,还有空间能够容纳另一个完整的磁盘块。因为传输缓冲区可能要环绕,所以磁盘读操作不能直接达到传输缓冲区。这里复制和内存的使用量相互之间存在着权衡。

在比较这两种方法时,还有另一个因素就是磁盘性能。使用大磁盘块时磁盘可以以全速运转,这经常是主要关心的事情。作为单独的单位读入小的P帧和B帧效率是比较低的。此外,将大磁盘块分解在多个驱动器上(下面将讨论)是可能的,而将单独的帧分解在多个驱动器上是不可能的。

图7-20a的小块组织有时称为恒定时间长度(constant time length),因为索引中的每个指针代表着相同的播放时间毫秒数。相反,图7-20b的组织有时称为恒定数据长度(constant data length),因为数据块的大小相同。

两种文件组织间的另一个区别是,如果帧的类型存储在图7-20a的索引中,那么有可能通过仅仅显示I帧实现快进。然而,根据I帧出现在数据流中的频度,人们可能会察觉到播放的速率太快或太慢。在任何情况下,以图7-20b的组织,这样的快进都是不可能的。实际上连续地读文件以选出希望的帧需要大量的磁盘I/O。

第二种方法是使用一个特殊的文件给人以10倍速度快进的感觉,而这个特殊的文件是以正常速度播放的。这个文件可以用与其他文件相同的方法构造,可以使用帧索引也可以使用块索引。打开一个文件的时候,如果需要,系统必须能够找到快进文件。如果用户按下快进按钮,系统必须立即找到并且打开快进文件,然后跳到文件中正确的地方。系统所知道的是当前所在帧的帧号,但是它所需要的是能够在快进文件中定位到相应的帧。如果系统当前所在的帧号是4816,并且知道快进文件是10倍速,那么它必须在快进文件中定位到第482帧并且从那里开始播放。

如果使用了帧索引,那么定位一个特定的帧是十分容易的,只要检索帧索引即可。如果使用的是块索引,那么每个索引项中需要有额外的信息以识别哪一帧在哪一块中,并且必须对块索引执行二分搜索。快倒的工作方式与快进相类似。

7.7.3 近似视频点播的文件存放

到目前为止我们已经了解了视频点播的文件存放策略。对于近似视频点播,采用不同的文件存放策