# A recording-failure DVD playback scheme with packet reading

Chun-Yi Tsai[1], Shann-Chiuen Wu[2] and Tzao-Lin Lee[1]
*Department of Computer Science and Information Engineering,*
*National Taiwan University[1], Taipei, Taiwan, R.O.C*
*Department of Accounting Information,*
*Tzu Chi College of Technology[2], Hualien, Taiwan, R.O.C*
*d92922016@ntu.edu.tw, chiuen@gmail.com, tl_lee@csie.ntu.edu.tw*

### *Abstract*

*DVD VR(Video Recording) is a logic format adopted by most home consumer DVD recorder machines to record real-time TV and multimedia content on DVD+R/RW or DVD-R/RW blank discs. However, DVD VR format can only be recognized and played by the DVD recorder itself. To achieve forward compatibility and playability, the recorder provides a function called disc finalizing process which closes every tracks and sessions of a DVD VR disc and changes its logic format from DVD VR to DVD-Video to make it playable on generic DVD-Video compatible players. Unfortunately, disc finalizing process provided by DVD recorder is very unstable and unreliable due to various uncertain reasons. This unreliable finalizing process often and easily produces failure discs which are neither in correct DVD VR format nor in DVD-Video format. Thus, these discs become unrecognized, and can no longer be played any more by any DVD players or recorders. This paper proposes an extension of software solution with packet reading which is capable of reading and playing non-finalized, finalizing-failure or other recording-failure DVD VR discs.*

## 1. Introduction

Current multimedia applications and consumer products, such as media centers, digital homes, personal video recording systems and other related audio/video applications are becoming more and more popular. All these applications have a common and important issue that they all need to provide the recording function to record multimedia content on certain type of storage media. DVD+R[2,7] and DVD-R[3,6] are the important optical storage media and physical format for recording applications in recent years benefited by their high volumes, low prices, and continuous promotion of high reading and writing speeds. DVD VR(Video Recording) format is a logical format adopted by most consumer DVD recorders to record real-time[13] multimedia content on DVD+R/RW or DVD-R/RW blank discs. DVD+R and DVD-R are the “write-once”[14] media that the storage area which is once written by application data cannot be erasable, while DVD+RW and DVD-RW discs can be. DVD VR format can be explicitly categorized as DVD+VR[7] and DVD-VR[6] as specified by different organizations, however there are quite a few similarities for these two from applications’ point of view. So the term DVD VR will be used in the remaining part of this paper to represent both DVD+VR and DVD-VR. The DVD VR format allows the recording of audio/video contents on the blank space of the disc in an incremental way. When the remaining space of a DVD VR disc is nearly depleted, users or the recording process have to stop recording and apply disc finalization afterwards. When a successful finalization process is completed, the logical format of the disc is transformed from the DVD VR format into the DVD-Video[12] compatible format which can then be recognized and played by nearly all kinds of recorders and players. Otherwise the non-finalized DVD VR discs can only be played by the consumer DVD recorder itself. Unfortunately, the disc finalization processes provided by the consumer DVD recorders are very unreliable possibly due to various reasons such as low quality disc media, inappropriate hardware design with inefficient heat-dissipation, instability of system software or other application bugs. Under any of the above-mentioned situations, the DVD+R or DVD-R discs become unrecognizable. Since they are write-once media that cannot be erasable, it means that they are damaged and no longer to be usable.

Another problem is the lack of free and open sources support for DVD VR format on PC DVD recorders today. So far the free and open sources of DVD software players or device drivers that we can find under Linux and Win32 systems only support for DVD-Video format and are incapable of reading and playing non-finalized DVD VR discs, not to mention those finalizing-failure DVD VR discs. This can be simply realized because DVD VR format is not yet a free and open specification today. In order to understand some features of DVD VR format, we use the black-box approaches to do experimenting on finalized, non-finalized, and finalizing-failure DVD VR discs to observe and understand the major differences among them. By experimenting with self-developed sector-inspecting tool, the implementation of a software player for directly reading and playing the multimedia content within non-finalized, finalizing-failure, or other recording-failure DVD VR discs without any drivers become feasible.

## 2. Observation

In order to add the DVD VR support and to implement this extension feature on some existing software player, it is necessary to understand the main difference between a non-finalized DVD VR disc and a finalized DVD VR disc. No specification details will be given here but only the difference sufficient for developments and implementations will be illustrated. A disc in DVD VR format can be transformed into DVD-Video compatible format after doing a proper disc finalization process. A non-finalized DVD VR disc can be recorded sequentially with multimedia content in MPEG-2 encoding format before doing the disc finalization. The disc finalization process recalculates the offset of several different multimedia content segments in this DVD VR disc, records the final file indexing information on certain sectors in the head of disc which is near the center ring of the disc, and at the same time writes disc closing information to the proper position of the corresponding sectors in the end of disc which is close to the edge of the disc.

To realize what a disc finalization process does, a DVD VR disc with only one short movie recorded using a home consumer recorder machine is observed for the difference before and after disc finalization. As illustrated in Figure 1, the movie file VTS_01_1.vob can be extracted from both the non-finalized and the finalized DVD VR discs. It can be firstly observed that the actual movie content itself is in the segment which is marked as ‘D’. Also, there is an appending segment marked as ‘E’ being recorded to both discs along with D segment. Hence, the file VTS_01_1.vob which can be copied from the finalized disc later will contains both D and E segments. It does not matter whether the VTS_01_1.vob contains E, both of them can be played normally by way of file playing. In addition, the starting address of VTS_01_1.vob in non-finalized disc ‘s1’ and that in finalized disc marked as ‘s4’ are the same. Similarly, ‘s2’ equals ‘s5’, and ‘s3’ equals ‘s6’.

The second observation is that, a major difference between non-finalized and finalized DVD VR disc is a reserved segment in the head of non-finalized DVD VR disc. This segment contains a section which is fully-filled with characters '0xAA'. After the disc finalization is successfully done, the file allocation and indexing information should have been written on this segment which is marked as 'C' in figure 1. Also the segment marked as 'F' in figure 1 is written at the same time. The successful recording of segment 'C' and segment 'F' make a DVD VR disc to look like a DVD-Video disc. Hence, it can be played by most of the DVD recorders and players.

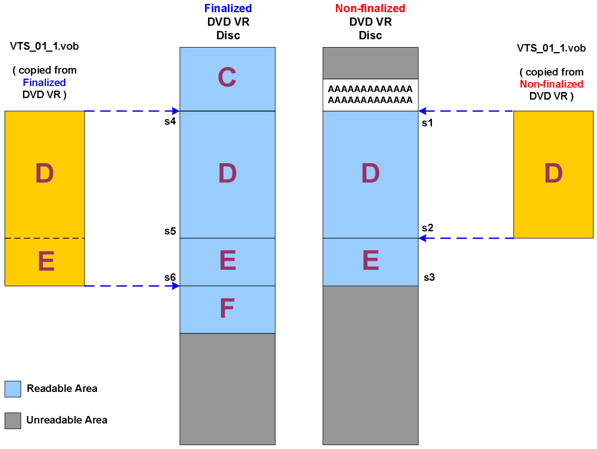

**Figure 1. Comparison of finalized and non-finalized DVD VR discs**

It implies that as long as there does not exist any serious recording errors contained in the storage area of movie content, i.e., the D segment, the movie content could be extracted for reading and playing simply by ignoring and skipping the formal stage of processing the information reading from the C and F segments which may contain various kinds of recording errors from the disc finalization process. Under most circumstances when the DVD finalization process fails, various recording errors on some logical sectors in the head of a disc would appear according to long term observations. In fact, the integrity of the actual multimedia content of the failure discs remains intact in most of the cases. It means that the multimedia contents are often good to be able to read and can be played normally by skipping the file indexing and other related processes that require the integrity of the information stored on some logical sectors in the head of disc.

## 3. Implementation

MPlayer[1] is one of the most popular multimedia software player available and widely-used to play video and audio contents. It is open sourced and is suitable for research and development. Certainly it is capable of playing DVD discs in DVD-Video format, however, as well as the same limitation that most of other players have, it does not support DVD discs in DVD VR format and is unable to play those discs. Since a non-finalized or finalizing-failure DVD VR disc is not in the DVD-Video compatible format, the default MPlayer cannot recognize and play such a disc. However, the starting address at which the multimedia content begins can be determined from the above observations. Thus the area of multimedia content would be readable and can be treated as a large file. Since MPlayer is capable of recognizing, reading and playing multimedia files, the additional support of DVD VR can be added into the source codes [9] which are responsible for initializing and recognizing files as the following:

```
stream_info_t stream_info_file = {
  "File",
  "file",
  open_f,
  { "file", "dvdvr", "", NULL}, //codes added;
  &stream_opts,
  1 // Urls are an option string
};
```

Running “mplayer” in a command line followed up with the argument string “dvdvr://”, mplayer will deal with the input disc in the way of file processing by the new DVD VR processing codes which is newly added into the original open_f() function in the source.

MPlayer uses standard POSIX lseek(), read(), and write() functions in default to do disc seeking, reading, and writing actions respectively to access a DVD-Video compatible disc or a finalized DVD VR disc. Unfortunately, these POSIX functions will return incorrect data values if they are accessing non-finalized or finalizing-failure DVD VR discs according to the actual experiments. It is probably that POSIX functions need to have the support of file indexing information from the head of discs furnished by device drivers in which case the DVD-Video can provide but DVD VR cannot. However, there exists a set of very useful programming APIs called MMC packet commands [4,5] providing functions for accessing SCSI devices, CD/DVD recorders and players or other related devices. The key advantage of MMC packet command is that no matter in what kind of format a DVD disc is, it can correctly read any readable logical sector of the disc for a given legal sector number.

Therefore, in order to make DVD VR discs accessible in MPlayer, MMC packet commands porting needs to be done. MMC Packet commands are written in C++ language with its major data structures declared in class syntax applied in the source of dvd+rw-tools[8] which is a backend tool containing the functions of DVD reading, writing, and disc media information browsing, etc. MPlayer version 1.0pre8 is chosen for development. It implements DVD seeking, reading, and writing functions in pointer function style. Each pointer function calls the corresponding POSIX function as mentioned earlier. Sectors reading is the primary IO action for a DVD playback, so READ(10) is the most frequently used function among MMC packet commands in this implementation. After successfully porting packet command support to MPlayer, the pointer functions can be rewritten by replacing the POSIX functions with the functions of calling MMC packet commands. This major modification looks like the following:

```
if (strcmp(stream->disc_type, “dvdvr”) == 0)
{
stream->fill_buffer = dvdvr_fill_buffer;
 ……
}
else
{
stream->fill_buffer = fill_buffer;
 ……
}
```

If MPlayer discovers the stream->disc_type is in “dvdvr“ format, then the function pointer of sectors reading of the stream, “stream->fill_buffer”, will be assigned to a newly implemented function dvdvr_fill_buffer() which calls the MMC packet command READ(10) to do sectors reading as the following codes, otherwise the function pointer keeps pointing to the original function fill_buffer() which calls the POSIX read(). The reading function fill_buffer() implemented in the original MPlayer for playing DVD is reading the multimedia content in a sector-by-sector fashion from the disc. Thus, the extended function dvdvr_fill_buffer() should follow the same way of sector reading as to be illustrated below. The sector number can be obtained from current position/address of a stream divided by the size of a logical sector, 2048 Bytes. Using the sector number the data format of a packet command READ(10) is filled with appropriate values containing from the least significant byte to the most significant byte by calling set_pkt_cmd() function. Finally the run_pkt_cmd() is called to actually execute the action of sector reading.

```
static int dvdvr_fill_buffer(stream_t *s, char* buffer, int max_len)
{
unsigned long sec_num = s->pos/sec_size; //sec_size:2KB
……
set_pkt_cmd(0, 0x28); //Packet Command READ(10)
set_pkt_cmd(2, (sec_num >> 24)&0xFF);
set_pkt_cmd(3, (sec_num >> 16)&0xFF);
set_pkt_cmd(4, (sec_num >> 8)&0xFF);
set_pkt_cmd(5, (sec_num) & 0xFF);
set_pkt_cmd(8, 1);  //reading a single sector
set_pkt_cmd(9, 0);
……
run_pkt_cmd(s->fd, READ, buffer, max_len);
……
}
```

When the DVD seeking and reading pointer functions augmented with MMC packet commands are tested to have worked normally, MPlayer is now capable to read consecutive data by reading each logical sector for each given legal sector number. If MPlayer starts reading from the first logical sector that the actual multimedia content begins, then it can read the content directly and play it immediately, and thus successfully achieve the goal of playing non-finalized, finalizing-failure, or other recording-failure DVD VR discs.

## 4. Result

### 4.1 Playback

Since MPlayer is a highly portable open source program, it can be built and run on many heterogeneous operating systems. In addition, MMC packet commands porting are successfully accomplished to make non-finalized or finalizing-failure DVD VR discs playable by the modified MPlayer on both Linux and Win32 systems. As shown in Figure 2, the modified MPlayer can successfully play both non-finalized and finalizing-failure DVD VR discs. The testbed platform for experimenting is a PC desktop equipped with a single CPU, AMD Athlon64 3000+ 1.8GHz, a DDR RAM module 1GB, and a DVD recorder with at most 16X reading speed running Ubuntu Linux with kernel 2.6.16. The playback is very smooth and all playback functions via user interface control such as three different speeds of forward and rewind control, sound control, full-screen, and other functions which are provided in the original source program can still be used normally in the new modified MPlayer.

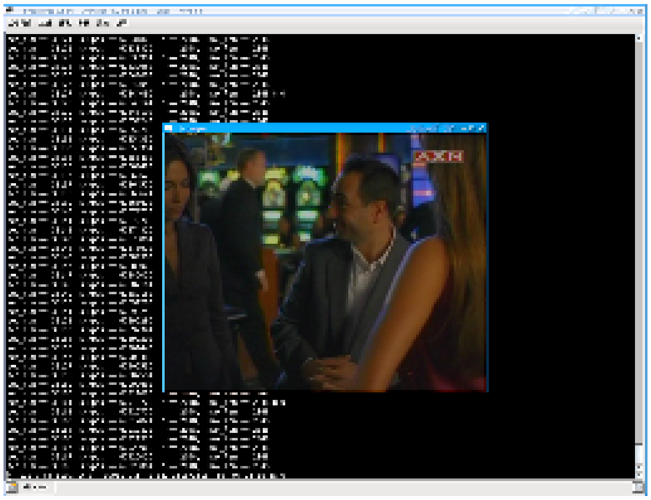

**Figure 2. Playback of a non-finalized, finalizing-failure or recording-failure DVD VR disc**

## 4.2. Performance evaluation

The performance evaluation which one may concern about is the elapsed time of reading a single logical sector of DVD VR discs. As mentioned earlier, the POSIX read() function is often applied on reading data from finalized DVD VR discs while the MMC packet command READ(10) adopted is capable of reading data from non-finalized or finalizing-failure DVD VR discs. Two methods can be used to measure the elapsed time of both the POSIX read() and MMC packet command READ(10). The first one is to insert the instrumentation directly in the code to get the starting time and the ending time of the reading action, thus the elapsed reading time can be obtained by substracting the beginning time from the ending time. The second one is to measure the reading time using strace[10] which is a handy tool for profiling and performance analyzing under unix-based systems. As shown in Figure 3, time required for reading each logical sector in the first 1000 samples by MMC packet command READ(10) is mainly located on the range from 400 to 800 microseconds which is measured by either instrumentation or strace. The time required for reading a logical sector by POSIX read() shown in Figure 4 is only about 6 to 11 microseconds by similar experimenting ways.

The POSIX read() needs less time than MMC packet command READ(10) does because only finalized DVD VR discs or DVD-Video compatible discs, i.e. the DVD discs with correct UDF file format[11], can be supported by POSIX read(). In fact, the reading action will directly call system call read() in kernel space and it is the reason why the reading time under such circumstances is relatively less. However, there is no direct support for those discs with non-finalized DVD VR format in most of OS kernels so far. That is why we have to port MMC packet command to MPlayer to enable the DVD VR reading support. The MMC packet command READ(10) can do the job of reading logical sectors of DVD discs with any DVD format including DVD VR. However it takes more time than POSIX read() does because it actually calls the SCSI library via SG_IO ioctl() instead of calling the system call read(). Although experimenting results show that MMC packet command READ(10) needs more time for reading logical sectors than POSIX read() does, actual running of the modified MPlayer when playing non-finalized or finalizing-failure DVD VR discs is very smooth. For example, reading a DVD VR disc recorded in a NTSC video with 720x480 resolution and 2130 Kbps bit rate of MPEG-2 format, the necessary number of sectors-to-read per second is about 130 to 140 sectors. In other words, reading a sector should be accomplished in 7 to 8 milliseconds. Hence, MMC packet command READ(10) can easily meet the requirement and is capable of playing DVD VR discs very smoothly under the experimenting hardware platform.

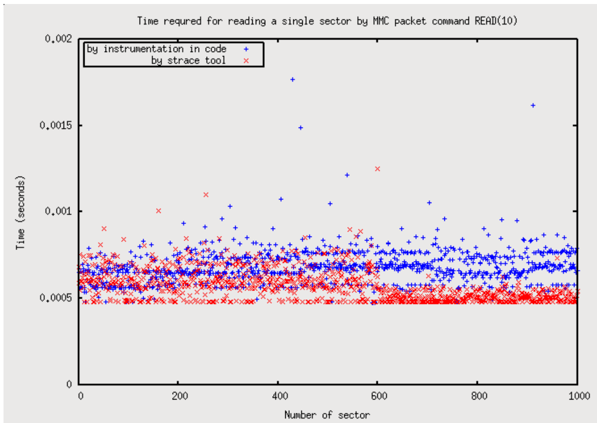

**Figure 3. Time required for reading a single sector by MMC packet command READ(10)**

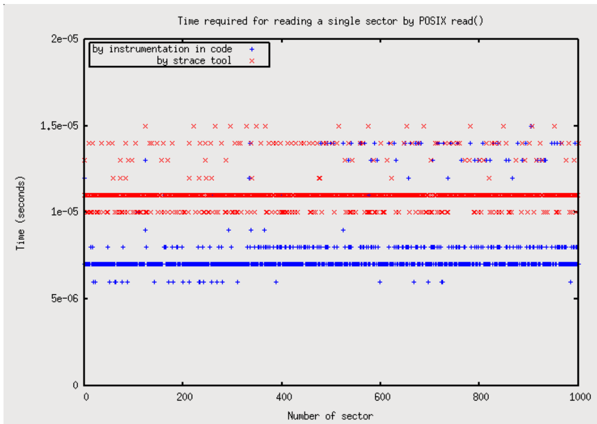

**Figure 4. Time required for reading a single sector by POSIX read()**

## 5. Conclusion

The demands for DVD or better quality recording are eminent. As the digital multimedia contents are produced more rapidly, the issues of the reliability and the quality of consumer DVD recorders are becoming more and more important. However, according to the practical testing for some consumer DVD recorder products, there exists still high probability of producing many finalizing-failure and other recording-failure DVD VR discs. Once these discs are produced, it means that they are damaged, no longer usable and cannot be played by any DVD recorders or players unless the new functionality proposed here can be deployed. It is not environmentally sound to junk so many DVD VR discs just because of the disc finalization or some other recording errors. When the applications of the 1080i streaming recording draw near, the next generation optical media, for instance, the HD-DVD and Blu-ray discs might become pervasive in the near future. This problem will become more significant because they can store far more data than the current DVD discs can. If unexpected recording errors still happen so often, it will not only make user frustrated but also the time and money wasted.

The proposed MPlayer implementation by incorporating the MMC packet commands can make non-finalized, finalizing-failure, and even other recording-failure DVD VR discs playable both on Linux and Win32 environment without any specific drivers support. Although it takes much more time for reading sectors by MMC packet command than by POSIX read(), the performance of MMC packet command is proved efficient enough to play DVD VR discs by experiments. Most importantly, it can save lots of efforts, time and money for the users, and significantly promotes the reusability of the otherwise would be useless discs.

## 6. References

[1] MPlayer web site, http://www.mplayerhq.hu, Aug. 2008.

[2] ECMA Standard, Data Interchange on 120 mm and 80 mm Optical Disk using +R Format - Capacity: 4,7 and 1,46 Gbytes per Side, ECMA-349, 3rd Edition, Dec. 2005.

[3] ECMA Standard, 80 mm (1,23 Gbytes per side) and 120 mm (3,95 Gbytes per side) DVD-Recordable Disk (DVD-R), ECMA-279, Dec. 1998.

[4] International Committee for Information Technology Standards (INCITS),Multi-Media Commands - 6 (MMC-6), Nov. 2006.

[5] SFF Committee, Mt. Fuji Commands for Multimedia Devices Version 7 (INF8090i v7), Aug. 2006.

[6] The DVD Forum web site, http://www.dvdforum.com, Aug. 2008.

[7] The DVD+RW Alliance web site, http://www.dvdrw.com, Aug. 2008.

[8] dvd+rw-tools web site, http://fy.chalmers.se/~appro/linux/DVD+RW/, Aug. 2008.

[9] ddd web site, http://www.gnu.org/software/ddd/, Aug. 2008.

[10] strace web site, http://sourceforge.net/projects/strace/, Aug. 2008.

[11] Universal Disk Format web site, http://www.osta.org/specs/index.htm, Aug. 2008.

[12] Jim Taylor, DVD-Video: Multimedia for the Masses, IEEE Multimedia, July-Sep. 1999.

[13] Hiroyuki Marumori, Hiroshi Chiba, Susumu Yoshida, A Real Time Recording System for Camcorders on the DVD Video Format, IEEE Trans. on Consumer Electronics, Vol. 49, No. 4, Nov. 2003.

[14] G. Stan and Hans Spruit, DVD+R - A Write-Once Optical Recording System for Video and Data Applications, International Conference on Consumer Electronics (ICCE), June 2002.