Inside DVD Video

wikibook

Contents

Articles

References

Article Licenses

Inside DVD-Video

This book is intended to be a complete collection of all the technical information that is publicly known about DVD-Video discs. The official DVD-Video specifications are only available to those willing to sign non-disclosure agreements and pay hefty licence fees; nevertheless, a lot of dedicated people have managed to reverse-engineer nearly all of the format, to the point where it has become possible to create Free Software authoring tools like DVDAuthor [1] and playback tools like Ogle [2] and VLC [3].

Target Audience

This book is aimed at a technical audience; anybody working on software for authoring or playing DVD-Video, or otherwise messing about with DVD-Video, who needs to have a detailed understanding of the data layout.

Terminology

The terminology and abbreviations used follow what seems to be standard when talking about DVD-Video. Thus, a “chapter” is referred to as a “Part of Title” and abbreviated “PTT” (though it is explained that this happens to mean a “chapter”). There are also various data structure field names that seem to be standard.

References

Documents

(Don’t do verbatim copying of these.)

- A useful overview of DVD-Video data structures is available at mpucoder.com [4]. Subsets of this (differing in some details) are also available at MPlayerHQ [5] and SourceForge [6].
- Bunch of old draft MPEG-related specs here [7].
- Publishing in the Age of DVD [8]

Source Code

- Ogle [2] is a now-moribund effort at a DVD-Video player, which is still useful as a reference for some subtle points, and also for testing authoring output.
- DVDAuthor [1] is an open-source authoring tool which has accumulated some useful information in its source code.
- FFmpeg [9] knows all the details of generating DVD-Video-compliant MPEG files.

Table of Contents

References

[1] http://dvdauthor.sourceforge.net/
[2] http://www.dtek.chalmers.se/groups/dvd/
[3] http://www.videolan.org/
[4] http://www.mpucoder.com/DVD/
[5] http://dvdnav.mplayerhq.hu/dvdinfo/
[6] http://dvd.sourceforge.net/dvdinfo/
[7] http://www.ece.cmu.edu/~ece796/documents/
[8] http://dvdauthor.com/tech01.htm
[9] http://ffmpeg.org/

Overview

This book is about the specific use of DVD discs to hold DVD-Video content. Information on the lower-level structure of a DVD disc can be found elsewhere; here we will concentrate only on the directory- and file-level structures necessary for DVD-Video.

The *titles* (content) on a DVD-Video disc are stored in MPEG format. In addition to this, a disc will typically have one or more *menu* screens with *buttons* marked on them, linked to each other and to the titles. You can navigate around the buttons in a menu by using the up, down, left and right arrow buttons on your player remote, and *select* a menu button by pressing the button marked “OK” or “Enter” on the remote, normally located in the middle of the arrow buttons.

There will usually be a “main menu” which automatically comes up when you put the disc in the player (after initial company logos, copyright warnings and other guff that the distributors insist on inflicting on us), and that you can return to with the “Top Menu” button on the remote. Other remote buttons labelled “Menu”, “Subtitle”, “Audio”, “Angle” or “Chapter” may bring up other special menus, depending on the disc.

Disc Structure Versus Title Structure

One thing that will strike you about the DVD-Video data structures is that they tend to offer a variety of ways of doing what amounts to the same thing. For example, how menus and titles are grouped on disc is one thing, whereas how they appear to the user (linked to each other by the buttons in the menus) can be very different.

Subpictures

DVD-Video also allows for subtitles. These are called *subpictures*, because they are represented, not as text, but as bitmaps overlaid on the video. They can thus represent arbitrary fonts, languages, writing systems etc. And they don’t even have to be text. However they are stored as two bits per pixel, thus allowing for only four different colours (transparent areas where the video shows through also count as a colour).

Subpictures are also used to implement highlighting of buttons in menus. In this situation, their display is normally “forced” (i.e. unconditional), so the user doesn’t have to turn them on. In fact it doesn’t make sense for the user *not* to see the buttons on the menus.

Interaction Commands

Menu buttons can do more than just bring up a title or another menu; there is a whole *virtual machine language* that is understood by an interpreter built into the DVD player, that can implement quite complicated interactive behaviours.

Evil Stuff

Region Coding

Commercial DVD-Video discs are commonly *region-coded*, while the consumer players are *region-locked* to only play discs coded with the same region. For example, North America is Region 1, Europe is Region 2, etc. This was to allow the distributors to release discs first into their most important markets, while taking their time over making them available elsewhere, without the worry that somebody might ship the discs to the latter markets and bypass the "official" distributors in the process.

Nevertheless, region-unlocked players are commonly available in many parts of the world, and in some places it is actually *illegal* to prevent customers from unlocking their players.

Copy Protection

Commercial DVD-Video discs are commonly encrypted with the *Content-Scrambling System* (CSS). The idea was that makers of player hardware and software would only be given access to the encryption keys if they signed an agreement to ensure that the decrypted video could not be easily recorded or copied by other devices or software.

Unfortunately, the CSS encryption was badly designed, and many people who did not sign such agreements were soon able to break it. Regretfully there are still legal restrictions in some jurisdictions on the distribution of CSS-decryption software, but all the good Free Software DVD-Video players are able to use the decryption library once you find it and install it (wink, wink).

CSS-encrypted discs have a special area containing decryption keys, which is only accessible via a special handshake between the DVD drive and the player software, details of which are not publicly known. Writable DVD blanks that ordinary consumers can buy do not have provision for this special area (and in any case off-the-shelf DVD writers cannot write it). So at this time Free Software cannot create such discs.

Prohibited User Operations

Remember I mentioned above the stuff that commonly starts playing on commercial discs as soon as you put them in the player? What makes them doubly annoying is that you often cannot skip or fast-forward over them.

This is done by specifying that certain "user operations" (skip, fast-forward, pause/still etc) are *prohibited* over certain areas of the disc.

Directory Structure

If you look at a DVD-Video disc on a PC, you will see a common directory structure: at the top level, a directory called `VIDEO_TS`, and probably also one called `AUDIO_TS`. The latter will have nothing in it; it is part of the structure of DVD-Audio discs, which never seemed to become very popular, details of which are not publicly-known.

Inside the `VIDEO_TS` directory, you will see a bunch of files with names ending in `.IFO`, `.BUP` and `.VOB`. The (typically large) `.VOB` files contain the actual MPEG-format content, while the (much smaller) `.IFO` ("info") files contain information about the menu and title structure. The `.BUP` ("backup") files are copies of the corresponding `.IFO` files with the same name, meant to be kept on a physically separate area of the disc, to provide some robustness in the event of physical damage.

Menus and titles in DVD-Video are grouped into *domains*. There can be one or more *titleset domains* (VTS), and there will also always be a *video manager* (VMG) domain. Differences between the two:

- Titlesets can contain both menus and titles, while the VMG can only contain menus.
- VM jump commands in the VMG can directly address menus and titles in any titleset, while those in titlesets can only address menus and titles within the same titleset or the VMG.

The VMG is kept in the files `VIDEO_TS.IFO`/`BUP`/`VOB`, while titleset *nn* (*nn* ∈ 01 .. 99) is kept in the files `VTS_`*nn*`_`*m*`.IFO`/`BUP`/`VOB. The *m* part of the name is used to break up long titles to keep each VOB file from becoming too large (it is always 0 for the IFO and BUP files, and for the VOB file containing the menus, if any). If the VMG has no menus or titles, then the file `VIDEO_TS.VOB` will be absent.

Example

Example listing of the contents of the `VIDEO_TS` directory:

```
-rw-r--r-- 1 ldo users      30720 Feb 21  2010 VIDEO_TS.BUP
-rw-r--r-- 1 ldo users      30720 Feb 21  2010 VIDEO_TS.IFO
-rw-r--r-- 1 ldo users     366592 Feb 21  2010 VIDEO_TS.VOB
-rw-r--r-- 1 ldo users      18432 Feb 21  2010 VTS_01_0.BUP
-rw-r--r-- 1 ldo users      18432 Feb 21  2010 VTS_01_0.IFO
-rw-r--r-- 1 ldo users     366592 Feb 21  2010 VTS_01_0.VOB
-rw-r--r-- 1 ldo users     366592 Feb 21  2010 VTS_01_1.VOB
-rw-r--r-- 1 ldo users      92160 Feb 21  2010 VTS_02_0.BUP
-rw-r--r-- 1 ldo users      92160 Feb 21  2010 VTS_02_0.IFO
-rw-r--r-- 1 ldo users  477292544 Feb 21  2010 VTS_02_0.VOB
-rw-r--r-- 1 ldo users 1073553408 Feb 21  2010 VTS_02_1.VOB
-rw-r--r-- 1 ldo users   71847936 Feb 21  2010 VTS_02_2.VOB
```

Here you can see the VMG and two titlesets (`VTS_01_`*x*`.xxx` and `VTS_02_`*x*`.xxx`). Each titleset has some menu footage (`VTS_`*nn*`_0.VOB`), while the title footage in the second titleset is too big for one VOB file, so it is split into two.

Physical Layout

Notwithstanding that a DVD has a full filesystem on it, there are still constraints on how files are located in the physical sectors on the disc. This allows players to be simpleminded and be able to play discs without having to decode fully-arbitrary filesystem structures.

First of all, all files must be contiguous. (This is guaranteed by the ISO9660 filesystem.)

Next, their contents must appear in a specific order, with files belonging to the same domain together: VIDEO_TS.*xxx* first, then (where present) VTS_01_0.IFO, VTS_01_*n*.VOB, VTS_01_0.BUP, VTS_02_0.IFO, VTS_02_*n*.VOB, VTS_02_0.BUP ... etc with the *n* in ascending order. Thus within each domain, the files are ordered thus: first the .IFO, then all the .VOB ones, then the .BUP.

It should be clear from this that the contents of the VIDEO_TS.IFO file will be the physically earliest file data on the disc.

Also, even though there may be multiple VOB files in a titleset, their contents will be physically contiguous and in sequence on the disc. That way, the player can simply determine where the first sector of the first VOB file in a titleset is located, and find the rest of the movie from there, without further reference to the filesystem.

(Ordering information taken from the cdrkit [1] source code, source file `genisoimage/udf.c`)

Can a .VOB File Be Played on Its Own?

The question is often asked: can the .VOB files be simply copied out of a DVD-Video disc and played on their own? The answer is “mostly yes”.

A .VOB file is an MPEG file with some additional information. The video is in MPEG format, while the audio might be in MPEG-standard or DVD-Video-specific format. Most video players these days should be able to cope with these just fine.

But there is additional information that doesn’t make sense when the .VOB file is taken on its own out of the containing DVD-Video structure: the menus and other interaction features won’t work, and the colour information for the subtitles comes from the IFO file, so these cannot be displayed in the right colours. But they should still be legible if the player assigns some reasonable default colours (or lets the user choose them). Also the identification of which audio track and/or subtitle track to choose for which language comes from the IFO file, so the player will have to let the user try them all to choose a suitable one.

So, in short, if you just want to treat the .VOB file as an MPEG file and ignore the DVD-Video-specific features, it should mostly work.

References

[1] http://www.cdrkit.org/

MPEG Format

MPEG (Motion Picture Experts Group) is the name of a family of format specifications for digital video. DVD-Video is primarily based around MPEG-2, though it also allows MPEG-1 video. This book will not go into all the details of MPEG; it will cover just enough to make sense of how it is used in DVD-Video.

Streams

Information in an MPEG file is divided up into *streams*: the video images form one stream, while the audio soundtrack is kept in another stream. DVD-Video allows for multiple audio streams, which can be used for example for soundtracks in different languages (only one audio stream may be playing at one time). Each stream is identified by a number, of which different ranges are allotted to different stream types. MPEG also allows for two *private streams*, the format of which is not further defined in MPEG; DVD-Video uses these for various purposes:

- additional audio formats not defined by MPEG (AC-3, DTS, LPCM) (private stream 1)
- subpictures (private stream 1)
- menu buttons, time-display information (*Presentation Control Information* (PCI), private stream 2)
- navigation information for trick-play and multi-angle modes (*Data Search Information* (DSI), private stream 2)

Video Formats

DVD-Video discs come in two main flavours, corresponding to the main formats used around the world for analog video: NTSC and PAL. NTSC was the first colour TV system, developed in the USA, and used in North America, Japan and a few other places. PAL was a later German development, used in Europe and much of the rest of the world. (There is also a French-developed broadcast format called SECAM, but the format for recorded media is identical to PAL.)

In the NTSC format, each video frame is 720*480 pixels, displayed at a rate of 29.97 frames per second. In the PAL format, each frame is 720*576 pixels, and the display rate is 25fps. Frames may be displayed at an *aspect ratio* of 4:3 for narrowscreen footage, or 16:9 for widescreen footage. Note that there is no difference in the numbers of pixels per frame for narrowscreen versus widescreen; the image is simply stretched out for widescreen (this is called *anamorphic* widescreen).

The allowed resolutions in DVD-Video are

- for NTSC: 720*480, 704*480, 352*480 or 352*240
- for PAL: 720*576, 704*576, 352*576 or 352*288

Interlacing

Unfortunately, DVD-Video has to carry over the *interlacing* feature from broadcast TV. That means each video frame is split into two *fields*, one consisting of the odd-numbered scan lines, the other containing the even-numbered ones, which are displayed one after the other.

Packets and Headers

The contents of each stream are divided into *packets*, which are *multiplexed*—a packet for one stream is followed by that for another stream which is to be presented at close to the same time—to allow a player to read and decode the file sequentially. You will come across the term *Packetized Elementary Stream* (PES) for this representation. In particular, each packet begins with a *header* containing an identifying code indicating the type of packet, followed by a two-byte field indicating the length of the packet contents.

There are also additional “headers”, with different identifying codes, used to specify various additional information:

- a *system header* gives information about the number of streams in the movie file. There needs to be at least one of these, at the start of the file.
- a *PACK header* gives information about the data rate needed to decode the movie file, as well as giving a high-precision clock reference (in units of a 27MHz clock). The presence of one of these indicates the start of a "PACK", which basically consists of the header plus all following PES packets until the next PACK header. DVD-Video requires that each pack be 2048 bytes in size.

GOP, I-Frame, B-Frame, P-Frame

Most video codecs rely heavily on interframe as well as intraframe compression to reduce data sizes. An *I-Frame* is a frame of video compressed by itself, without looking at other frames. The encoding scheme used is similar to JPEG compression. However, subsequent frames are quite likely to look similar (think of the common case of something or someone moving against a still background); therefore, instead of compressing them on their own as additional I-Frames, it makes sense to encode them as *P-Frames* which are differences from the preceding *reference* frame (which can be an I-Frame or a P-Frame) or as *B-Frames* which are differences from both preceding and following I- or P-frames.

The drawback with this is, if you try to start playback from some arbitrary point that is not at the beginning of the file, the player has to seek backwards until it hits an I-frame before it can start sensibly decoding the video. Thus, using fewer I-frames improves compression, at the expense of quick random access into the video stream. The DVD-Video specification requires at least one I-frame every 36 fields for NTSC or every 30 fields for PAL (i.e. at least every 0.6 seconds).

The sequence of frames starting from an I-frame until the last frame before the next I-frame (in other words, containing all the frames depending in some way on the starting I-frame) is called a *Group of Pictures* (GOP).

VOBU, Cell, Program, PGC

As previously mentioned, DVD-Video expects to see a PACK header every 2048 bytes. The contents of the first pack must be a PCI and a DSI packet; this is called a "NAV PACK". Following this will be other PACKs containing video, audio or subpicture stream packets as appropriate, in no particular order, but the NAV information must be first, and the video packets should contain one or more complete GOPs. The NAV and following packs (up to the next NAV PACK) make up a *Video Object Unit* (VOBU), and is the smallest unit that the decoder can work with. The duration of a VOBU must be from 0.4 to 1.0 seconds.

DVD-Video defines additional levels of grouping beyond this; one or more VOBUs make up a *cell*; one or more cells make up a *program*, and one or more programs make up a *program chain* (PGC). Particular programs can each be identified in the file structure as a *Part of Title* (PTT), otherwise known to ordinary people as a *chapter*. An actual title on the disc can correspond to a single PGC, or sometimes to multiple PGCs.

Additional significance of the groupings is as follows:

- a cell is the smallest unit that can be targeted by a jump.
- a cell can have a single VM instruction attached to be executed when the cell finishes playing.
- skipping with the Next/Prev buttons on the player remote is done in units of programs, not necessarily chapters.
- only PTTs (chapters) can be the target of a jump from outside the containing PGC; jumps to cells and (non-chapter) programs can only happen from within the containing PGC.
- a PGC can have a sequence of VM commands to be executed prior to playing, and another to be executed at the end. The PGC also defines the 16-entry colour table from which subpictures within the PGC can choose 4 at a time to display.

There is a special "first-play" PGC (FPC) in the VMG which is automatically entered when the disc is put into the player; this has no MPEG data, but it is the commands here that are responsible for jumping to the initial menu (if

any), playing all the preamble sequences etc.

Menus vs Titles

Menus and *titles* are very similar; both are PGCs, and both can have interactive buttons. However, only titles can have chapters, and only titles can be split across multiple VOB files (which imposes a maximum length on the duration of a menu). Also, only titles can have multiple audio and subpicture tracks for different languages, because a menu is already part of a language-specific menu group.

Special Menus

Certain menus can be marked as *entries* that can be directly invoked via dedicated buttons on the player remote.

In the VMG, one menu can be marked as the *title* entry, which means it can be brought up by the “Top Menu” button on the remote.

In a titleset, one menu each can be marked as *root*, *PTT* (chapter), *audio*, *subtitle* and *angle*; the root entry is brought up by the “Menu” button, the others by correspondingly-titled buttons on the remote.

IFO Files

IFO (“information”) files impose a structure on the raw MPEG data. The VMG has an IFO file, and there is also one in each VTS. The two kinds of IFO files (VMG vs VTS) have a common initial structure, but differ in later parts.

(see diagram here [1] for now)

Here are kept various tables of pointers to find the titles, chapters and so on. Here also the *attributes* of the video, audio and subpicture streams are summarized. Some of this (e.g. audio format) is redundant and could be obtained by scanning the actual MPEG data (though presumably having it here is more convenient for player initialization), while other info, particularly the language-code assignments, is important for letting the user configure the playback of the titles. This information does not include the actual stream IDs; these are to be found within each PGC (below).

Program Chains (PGCs)

(see here [2] for diagram for now)

Each PGC has an entry in an IFO file. The PGC’s VM commands and colour table are kept here, along with pointers to find the corresponding MPEG data in the VOB files.

Note that the PGC colour table defines colours in Y,Cr,Cb format with no transparency component. Instead, transparency (which DVD-Video refers to as *contrast*) is specified when the colour is referenced in the subpicture stream, or in a menu button definition.

Here will also be found *stream control* information, 8 entries for audio streams and 32 entries for subpicture streams. These correspond in order to the audio and subpicture attributes entries in the earlier part of the IFO. They define the particular stream numbers corresponding to those streams within the MPEG data for this PGC. Note that each subpicture stream control entry has room for 4 different stream numbers, being the streams to use for narrowscreen footage on a narrowscreen TV, widescreen footage on a widescreen TV, or widescreen footage on a narrowscreen TV in letterbox or pan-and-scan modes.

The ability to have different stream IDs for corresponding streams in different PGCs is presumably to allow easier authoring of a disc, by being able to combine preexisting material from different sources, without having to reprocess them all to assign consistent stream IDs. The “SCR discontinuity” flag that each cell has looks like part of this purpose as well: it warns the player about discontinuities in clock timestamps, saving the authoring system the

bother of reprocessing the MPEG data to assign consistent timestamps.

Cells and Programs Within the PGC

The cell position information table explicitly lists all the cells that make up this PGC. These cells do not have to occur in sequence in the MPEG data, and the same cells can be referenced multiple times. This allows for the same material to be presented in multiple different ways, without the need for complicated programming of explicit cell-level sequencing using the interaction machine.

The program map entry table defines the groupings of cells into programs within the PGC. Each entry specifies a starting index into the cell position information table, and specifies the inclusion of all cells listed from that point in the table up to prior to the start of the next program.

Menu Groupings

Both the VMG and VTS IFO headers include sections listing all the menus. These are grouped by *language units*, each identified by ISO639 language code. Within each language unit there can be one or more menus, and each menu can consist of one or more PGCs. However, it's not clear that these language units serve anything more than an informational purpose, since all selection of menus is done by explicit execution of VM instructions.

Title Map

The VMG IFO includes an additional section that assigns a sequential index to every title in every titleset. This is used for cross-domain jump instructions in the VM: instead of being able to say "jump titleset *m* title *n*", you have to say "jump title *i*", where *i* is an index into this table that lists *m* and *n*.

VOB ID

Each cell is identified by a 16-bit "VOB ID" concatenated with an 8-bit "cell ID" within that VOB. This "VOB" doesn't correspond to the segmentation into multiple .VOB files in the output DVD-Video file structure; it could correspond to input MPEG source files used to author the structure, with each input file being assigned an incremental VOB ID starting from 1. Thus, if an input source MPEG file is smaller than 524272 sectors, then the next input source file will be concatenated onto the same output .VOB file, but the VOB ID will increase.

But does this really matter? The player doesn't care how the input source files were divided up.

VMGM_C_ADT, VTSM_C_ADT, VTS_C_ADT

These *xxx*_C_ADT tables are indexed by VOBU number, and return information about which cell the VOBU belongs to. These tables define the cells.

VMGM_VOBU_ADMAP, VTSM_VOBU_ADMAP, VTS_VOBU_ADMAP

These *xxx*_VOBU_ADMAP tables are indexed by VOBU number, and return the starting sector, relative to the start of the containing VOB, of the VOBU.

Where the VOBs Are

Both the *xxx*_C_ADT and *xxx*_VOBU_ADMAP tables above locate things (cells, VOBUs) by sector number *within the containing VOB*. But there's no table of where the VOBs themselves start. (Remember, VOBs have nothing to do with the division into .VOB files in the DVD-Video directory structure.) So how do you find the VOBs?

It seems to me the player has to deduce these, as follows:

- The first VOB begins at the start of the first .VOB file. From this, you can determine the start of each cell and VOBU within the first VOB.
- By examining the *xxx*_C_ADT tables, you can deduce the highest cell number within each VOB. This gives you the highest sector number within the first VOB.
- Therefore, the next sector following must be the start of the second VOB. Redoing from the previous step on this VOB lets you work out its highest sector number.

And so on in sequence, until you have all the VOBs mapped out.

References

[1] http://www.mpucoder.com/DVD/ifo.html
[2] http://www.mpucoder.com/DVD/pgc.html

Subpicture Streams

Subpictures are used to display subtitles as well as menu buttons, overlaid on the video. These are stored in an MPEG stream (private stream 1) as a sequence of *Sub-Picture Unit*s (SPUs). SPUs may be broken across multiple MPEG stream packets.

Each SPU contains an image bitmap with 2 bits per pixel, split into odd and even fields, stored using a run-length-encoded (RLE) compression format. This is accompanied by a sequence of commands that control the actual display of the bitmap, including the choice of which four colours from the PGC colour table to use. There is also a time delay that can be specified before executing the commands.

(see here [1] for more details for now)

A complete SPU cannot exceed 65535 bytes, because the length header field is only 16 bits. However, an uncompressed full-screen subpicture image can be as large as 103680 bytes (= 720 * 576 / 4) for PAL, or 84600 bytes (= 720 * 480 / 4) for NTSC, and RLE compression performs poorly when there are only short runs of identical pixels. Any image which cannot be RLE-compressed to fit within the limit is simply not allowed.

Subpictures may be turned off or on under user control, or their display may be *forced* so they are always visible. User-controllable subpicture display is useful for showing subtitles, while forced subpictures are useful for buttons in menus—you don't want the user wondering where the buttons have gone!

At most one subpicture stream can be visible at a time, overlaid on the video image, which will show through wherever the subpicture pixels are not fully opaque.

Widescreen Considerations

One subtlety to keep in mind with widescreen video is that in this case *subpictures are scaled to fit the screen, not the video*. Specifically, if widescreen video is being shown "letterboxed" on a narrowscreen TV, the subpicture display area will still include the parts occupied by the black bars above and below the video image. As they say, "this is a feature, not a bug"; presumably, the point is to allow you to use the black area to show subtitles, leaving the video image untouched. The PGC header allows you to specify up to four alternative subpicture streams to be substituted as appropriate, depending on the following situations:

- narrowscreen footage ("normal") being viewed on all TVs
- widescreen footage being viewed on a widescreen TV
- widescreen footage being letterboxed on a narrowscreen TV
- widescreen footage being shown in "pan-and-scan" mode on a narrowscreen TV

(Though of course narrowscreen video will only use the first one, and widescreen will not use the first one.)

References

[1] http://www.mpucoder.com/DVD/spu.html

NAV PACKs

NAV (navigation) PACKs contain one PCI (Presentation Control Information) packet and one DSI (Data Search Information) packet. I'm not sure if the order of the two is important, but every VOBU must begin with a NAV PACK.

PCI Packet

The PCI packet defines buttons. Here is also hidden the flag that turns on Macrovision APS copy-protection for the video in this VOBU.

(see here [1] for more details for now)

Buttons

Buttons are the way for the user to interact with the DVD-Video content. They can appear in both menus and titles. Their appearance and behaviour is defined by the currently-visible subpicture stream, in conjunction with information from the last-seen PCI packet. Since all this information is coming from the VOB file which is being continuously read by the player, buttons can be *dynamic*: they can appear and disappear, change their appearance, and move around over time.

Each button belongs to a button group, of which there may be up to three. Button groups fulfil two conflicting purposes: on one hand they allow different colour schemes (choice of highlighted/selected colours), on the other hand they offer alternative layouts when widescreen footage is being shown on a narrowscreen set in letterbox or pan-and-scan modes versus on a widescreen set. This is why, even though there is room for 36 button entries, a widescreen menu can be limited to as few as 12 buttons, if you provide three alternative button layouts for widescreen footage.

A button can be in one of 3 states: the *normal* state is that defined by the subpicture stream. One button will always be in the *highlighted* state: this is indicated by the subpicture pixels within the button's rectangle having their colours (including transparency) taken from the relevant table entry in the PCI, overriding what the subpicture stream specifies. As the user uses the up/down/left/right keys on the remote, the highlighted state moves from button to button, according to the corresponding spatial relationship information as defined in the PCI tables.

Just to be clear: switching between normal, highlighted and selected states is done purely by changing the colours (and transparencies) assigned to the pixel values; the pixel values do not change from those specified in the subpicture stream.

Then, when the user presses the "OK" or "Enter" button, the highlighted button is (supposed to be) briefly shown in the "selected" state, and then the interaction command associated with that button is executed. Some players don't seem to bother to show the button in the selected state.

Buttons can also be defined as "auto-action" buttons; this means the associated command is executed immediately the button becomes highlighted, without waiting for the "OK" press. This is particularly handy for defining menus that require more buttons than can fit on one screen; the "next screen" and "previous screen" buttons can be auto-action ones, saving the number of keypresses the user needs to switch between screens.

A common technique is for the "normal" state, as defined by the subpicture stream, to show no visible (opaque) subpicture pixels at all. Instead, it is up to the video layer to define what the user sees as the "normal" appearance of the buttons. This is because the video layer allows for more elaborate, full-colour images, while subpictures are

limited to a maximum of four colours. Of course, the “highlighted” and “selected” embellishments are still limited to the four-colour maximum.

DSI Packet

The DSI packet has links to preceding and following VOBUs at various predefined time offsets from 0.5 seconds up to 120 seconds. This helps the player to implement its trick-play (fast forward/reverse) modes. It also defines how VOBUs corresponding to different camera view angles are interleaved.

(see here [2] for more details for now or See also nav_notes [3] for some C slightly different pseudo code).

References

[1] http://www.mpucoder.com/DVD/pci_pkt.html
[2] http://www.mpucoder.com/DVD/dsi_pkt.html
[3] http://dvd.sourceforge.net/nav_notes

Interaction Machine

DVD-Video specifies a low-level machine-language instruction set that is used to define interactions with the user. Instruction sequences are invoked when the user selects an on-screen button, and may also be triggered at various other points, such as the end of a cell or a PGC, or when the disc is inserted into the player.

The instruction set is commonly referred to as the “VM” (Virtual Machine) language, but the implication of calling it “virtual” is that there cannot be *real* hardware that directly implements this instruction set. But why not? That’s why I am here using the term “interaction machine”, which makes it clearer what the purpose of the instruction set is.

(Note that I’m avoiding using the term “program” for sequences of instructions in this instruction set, because that already has a completely different meaning in DVD-Video.)

Each instruction is 8 bytes long. This sounds like a lot, because each instruction can do several things at once: for example, perform a comparison, set a register, jump to some location.

(For now: instruction set summary [1] and details [2])

There are 16 *general-purpose parameter registers* (GPRMs), each of which can hold an unsigned 16-bit integer. Their use is entirely up to the instruction-sequence writer. They are the only data storage available to instruction sequences. There are no indirect or indexed addressing modes available.

There are also various *system parameter registers* (SPRMs), with predefined meanings to the player. There are special instructions for accessing these, and some of them are read-only.

(SPRMs are listed here [3] for now)

Interaction instructions can be found in the following places:

- a cell can end with a single instruction to be executed when the cell has finished playing.
- a button specifies a single instruction to be executed when the button is selected.
- the pre- and post-sections of a PGC can each contain a sequence of instructions, to be executed before the PGC starts playing and after it finishes, respectively.
- the FPC contains a sequence of instructions which automatically start executing when the disc is inserted into the player.

The single-instruction limitation on buttons is not a big issue in practice; it’s easy enough for the single instruction to be “jump to location *n* in post-section of PGC”, to execute a longer sequence of instructions beginning at that location *n*.

Most operations can be executed conditionally: the condition is based on comparing either two GPRMs, or a GPRM and a literal value from the instruction, according to a choice of comparison operators.

Besides being conditional, instructions can perform the following sorts of operations:

- transfer of control within an instruction sequence, or terminate an instruction sequence
- *link* (transfer to a cell, program, PTT or PGC within the same domain), optionally specifying a button to show as highlighted
- *jump* or *call*—transfer to a PGC or PTT in another domain. Call differs from jump in that the current (or an alternative specified) cell location is saved as the address to return to on execution of a resume instruction. Only a single such cell location can be saved; executing a new call overwrites the return address from the previous call. Not all combinations [4] are allowed
- set a GPRM to either a literal value, the value of another GPRM, or the value of an operation involving a GPRM and a literal value
- set specified SPRMs to specified values (might be literal or from a GPRM, depending on the instruction). Only certain SPRMs can be set this way
- combination of a set-GPRM and a link (conditionals not allowed)
- combination of a conditional, set-GPRM and link, in 3 variations: 1) the set is unconditionally done, then the conditional test is used to decide if the link should be done 2) the conditional test is used to decide if the set followed by the link should be done 3) the conditional test is used to decide if the set should be done; in either case, the link is always done

Permitted Jumps/Calls

Here is a table of the permitted transfers between domains, and the names of the instructions to use.

from\to	FPC	VMG	VTSM	VTST
FPC		JumpSS	JumpSS	JumpTT
VMG	JumpSS		JumpSS	JumpTT, RSM
VTSM	JumpSS	JumpSS		JumpVTS_TT, JumpVTS_PTT, RSM
VTST	CallSS	CallSS	CallSS	JumpVTS_TT, JumpVTS_PTT

Note that, from a VTS (menu or title), it is not permitted to jump directly into another VTS. You have to set up indirect jumps via the VMG. Note also that, from a title, you have to “call” rather than “jump” into a menu; this is to allow the menu to use RSM to resume playback of the title.

References

[1] http://dvdnav.mplayerhq.hu/dvdinfo/vmi-sum.html
[2] http://dvdnav.mplayerhq.hu/dvdinfo/vmi.html
[3] http://dvdnav.mplayerhq.hu/dvdinfo/sprm.html
[4] http://dvdnav.mplayerhq.hu/dvdinfo/vmi-jmp.html

Limits

Summary of Specification Limits

The following limits are imposed by the DVD-Video spec.

Each title can have one video stream, up to 8 audio streams, and up to 32 subpicture streams.

There can be no more than 99 titlesets, no more than 99 menus in the VMG or a titleset, and no more than 99 titles in a titleset.

Each title may be made up of up to 999 PGCs. Each PGC may consist of up to 255 programs. The pre- and post-sections of a PGC put together can contain no more than 128 VM instructions.

Since there is only one VOB file (`VIDEO_TS.VOB`) in the VMG, the total amount of video in the VMG menus must fit into 1073709056 bytes (524272 sectors of 2kiB each). In each titleset, all the menu video must fit in the first VOB (`VTS_`*nn*`_0.VOB`), so is limited to the same amount.

Up to 9 different angles are allowed for multi-angle.

There can be up to 36 buttons on-screen at once for narrowscreen footage, but only 12 at once for widescreen, because of the need to provide up to 3 different versions of the subpicture stream, one for use on widescreen TVs and the other two on narrowscreen.

Glossary

Aspect Ratio

: a somewhat ambiguous term, sometimes used as a synonym for pixel density. In this book, I will take it to mean the relative dimensions of the video image as viewed on a TV screen, measured in units of distance, not pixels. Thus, narrowscreen video has an aspect ratio of 4:3, while widescreen video has an aspect ratio of 16:9.

BCD

: short for Binary-Coded Decimal. BCD numbers are used for the playback time field in the PGC, and for the cell elapsed time in each PCI packet. Presumably this format is used, not for calculation purposes by the player, but for easy conversion to a human-readable form in the player's display.

B-Frame

: a video frame which is encoded as the difference between two other reference frames, one temporally preceding and one temporally following.

Button

: a defined rectangular area on the screen during the display of a menu or title (yes, titles, not just menus, may have buttons). May be selectively *highlighted* by the user pressing the up/down/left/right arrow keys on the player remote to navigate around the available buttons; when the OK/Enter button is pressed, the highlighted button (depending on the player) may be briefly shown in the *selected* state, after which the VM instruction attached to the button is executed. A button may also be an *auto-action* button, which means it is selected (and the instruction executed) by the mere act of highlighting it.

Cell

: a unit consisting of one or more VOBUs, possibly with a single VM instruction attached to be executed when it finishes playing. The smallest unit that can be targeted by a VM jump instruction.

Domain

one of the major divisions in the DVD-Video disc structure; either a VMG or a VTS.

DSI

Data Search Information. A packet that contains information needed for scanning/stepping forward and backward in trick-play modes.

DTS

Decoder Timestamp. This value needs to be a fraction of a second in advance of the PTS in order to give the decoder machinery time to process the packet. I'm not sure why this needs to be present at all; since the necessary delay would presumably vary from one decoder design to another, so it should be up to the decoder implementation to deal with any necessary timing compensation, this information has no business being in the MPEG stream.

DVD

an optical disc format which was a development from the earlier Compact Disc format. Originally supposed to stand for "Digital Video Disc", then "Digital Versatile Disc", then finally it was decreed that it stood for nothing at all, so that it could be trademarked.

DVD-Video

a specification for using DVDs to hold video and movies together with options for interactive menus for accessing them; the subject of this book.

Field

half of a frame (either all the odd-numbered scanlines, or all the even-numbered ones), for interlacing.

FPC

First-Play PGC. A special PGC in the VMG that gets control when the disc is inserted in the player. Would contain instructions to bring up the main menu, but this is often preceded by distributor logos, copyright warnings, and sometimes even advertisements that can't be skipped.

Frame

a video frame is a single still video image. A succession of frames displayed one after another at a sufficiently high rate gives the illusion of a moving image. Cinema movies normally use a rate of 24 frames per second, which is adequate for most, but not all, motion. However, flashing 24 images per second onto a screen, separated by darkness, leads to very discernible and annoying flicker. So each frame is actually projected *twice* in succession, giving 48 flashes per second, which most people don't notice. Broadcast TV solved the flicker problem in a different way, by using interlacing.

GOP

Group Of Pictures. An I-Frame, together with all dependent P- and B-Frames. In DVD-Video, there must be one I-Frame every 18 frames (36 fields) for NTSC, and one every 15 frames (30 fields) for PAL.

Header

a unit of information in an MPEG file. Begins with 2 bytes of 0, then one byte containing 1, then another byte containing an ID code which indicates what (if any) further information might be following. Some codes indicate the presence of a packet.

I-Frame

a video frame which is compressed in a self-contained way, and can be decoded without reference to any other frame. See also P-Frame and B-Frame.

Interlacing

a bandwidth-saving hack that was originally introduced for analog broadcast TV, and for some reason kept for digital TV. Instead of showing each frame in its entirety, and using the double-projection trick to mask the

flicker, each frame is split into two fields, which are shown in succession. Undoing the interlacing, by combining pairs of fields into frames, is not so simple, because video cameras record the fields at different times, so any movement of the objects in the picture can lead to strange artifacts.

Letterbox

a way of displaying widescreen footage on a narrowscreen TV, by shrinking the image to fit the width of the screen, and padding out the unused parts of the screen above and below the picture with a black background. Compare Pan & Scan.

MPEG

stands for **M**otion **P**icture **E**xperts **G**roup. The name for a family of standards to do with digital video, of which the only ones relevant to DVD-Video are the earliest two: MPEG-1 and MPEG-2.

Narrowscreen

the original 4:3 aspect ratio for broadcast TV, adopted from early cinema movies.

NAV PACK

a PACK that contains PCI and DSI packets—navigation information. There must be one of these at the start of every VOBU.

NTSC

National **T**elevision **S**ystem **C**ommittee. The original analog broadcast colour TV specification developed in the USA, and one of the two most widely used in the world, the other being PAL. Defines a frame rate of 29.97 fps, and digital video frame dimensions of 720x480.

Overscan

a limitation of early TV sets, due to their inability to accurately match up the boundaries of the visible picture with the boundaries of the set's picture display area, meant that a few percent of the image was lost all round the edges. Even with today's pixel-accurate LCD and plasma TVs, the convention has carried over of throwing away a few percent of the video image around the edges, because broadcasters often allow weird rubbish to creep into this area, like flashing timecode dots. For this reason, when composing a video image, or laying out a DVD-Video menu, or such, remember not to put anything important too close to the edges of the picture, or it might get chopped off when viewed on a TV.

PACK/Pack

(I'm not sure whether it's capitalized or not) a part of an MPEG file, beginning with a PACK header and including the following packets up to but not including the next PACK header. In DVD-Video, each PACK must be 2048 bytes in size (the DVD sector size).

Packet

a unit of information in an MPEG file. Begins with 2 bytes of 0, then one byte containing 1, then another byte containing an ID code which indicates the packet type, followed by two bytes containing the length of the following packet data.

PAL

Phase **A**alternation **L**ine. An analog broadcast colour TV specification developed in Germany, and one of the two most widely used in the world, the other being NTSC. Defines a frame rate of 25 fps, and digital video frame dimensions of 720x576.

Pan & Scan

a way of displaying widescreen footage on a narrowscreen TV, by chopping off the sides of the picture to fit. Instead of always chopping equal amounts on both sides, it is possible to define a "picture origin" in the MPEG video stream, which moves to track the area of greatest importance in the image. Compare Letterbox.

PCI

Presentation Control Information. A packet that contains information about buttons.

P-Frame

a video frame which is encoded as the difference from a preceding I-Frame. This reduces the size of the video data greatly, the downside being that if you randomly skip around in the video stream, playback can only start properly from an I-Frame.

PGC

Program Chain. A unit consisting of one or more programs. A menu or title is made up of one or more PGCs. A title PGC may have PTTs defined in it.

Pixel

a sample of an image, defining intensity and colour at a single point. Contrary to common misconception, a pixel is *not* a little square; its ideal shape is a dimensionless point.

Pixel Density

the number of pixels per unit distance in a displayed image. For our purposes, the important issue is whether the pixel density is *uniform* (equal both horizontally and vertically) or not. Computer displays (all the ones worth using, anyway) have uniform pixel densities, and computer image-manipulation software commonly assumes it is dealing with images with uniform pixel densities. Digital video in NTSC and PAL formats, however, does *not* have uniform pixel densities.

Private Stream

the MPEG format reserves two stream IDs for purposes not further defined in MPEG. DVD-Video uses these as follows:

* Private Stream 1: for additional non-MPEG audio formats, as well as subpicture streams

* Private Stream 2: for PCI and DSI packets.

Program

A unit consisting of one or more cells. The smallest unit that can be directly addressed for navigation purposes. A program may also be marked as a PTT (chapter point).

PTS

Presentation Timestamp. In order to ensure that playback is properly synchronized, a minimum proportion of the MPEG packets must have PTS values attached to them. These are a monotonically-increasing 33-bit counter in units of a 90kHz clock. (90000 just happens to be easily divisible by the common frame rates like 24, 25, 30.)

PTT

Part of Title. A program that can be directly addressed via a jump VM instruction from outside the containing PGC. The DVD-Video term for what ordinary people refer to as a *chapter*.

Reference Frame

an I-Frame or P-Frame.

Resolution

an ambiguous term, which can refer to either the total number of pixels or to the pixel density. In this book, I will use it to refer to the total numbers of pixels across and down in the following form: *width*x*height*, e.g. 720x480 for full-resolution NTSC images, 720x576 for full-resolution PAL.

SCR

System Clock Reference. An additional timestamp found in PACK headers, with additional resolution so its units are a 27MHz clock rather than 90kHz.

SECAM

Séquentiel Couleur A Mémoire. An analog broadcast colour TV specification developed in France. In terms of recorded media compatibility (DVD-Video as well as the older VHS tapes), it can be lumped in with PAL.

Stream

one of the concurrent information channels in an MPEG movie file. Video is a stream, and audio is another stream. DVD-Video also defines other streams used for various purposes (navigation, subtitles).

Subpicture

a separate image stream defined by DVD-Video which can be overlaid on top of the video. Can represent arbitrary graphics, except it is limited to two bits per pixel (unlike the video, which can be in full colour). Used to display subtitles and menu button highlights; because it is graphics, and not text, it can represent arbitrary languages and writing systems, use any fonts, etc. Can have transparent areas where the video shows through.

Uops

User Operations. Classification of the operations the user can perform (e.g. pause, skip) into different categories that can be selectively prohibited in various places. You see these prohibitions in action every time a disc shows you an ad or copyright warning that can't be skipped.

VM

Virtual Machine. A term with many uses in computing, but in DVD-Video it specifically refers to the machine language defined for implementing button operations and other interactive functions.

VMG

Video Manager. The domain in the DVD-Video disc structure that gives access to all the others. There must be exactly one of these. Can only contain menus, not titles.

VMGM

Video Manager Menu. A menu in the VMG.

VOBU

Video Object Unit. A part of a DVD-Video MPEG file beginning with a NAV PACK and followed by PACKs containing data for the other streams (video, audio, subpicture) as appropriate. The video packets must make up one or more complete GOPs.

VTS

Video Titleset. A domain in the DVD-Video disc structure. There can be one or more of these. Can contain menus, titles, or both.

VTSM

Video Titleset Menu. A menu in a VTS.

Widescreen

in DVD-Video and current-generation digital TVs, an image aspect ratio of 16:9. Most modern cinema movies are filmed in a much wider ratio than this, but 16:9 was presumably considered not to be too jarring a step up from the older 4:3.

Article Sources and Contributors

Inside DVD-Video *Source*: http://en.wikibooks.org/w/index.php?oldid=2350069 *Contributors*: Adrignola, Ldo, Pi zero

Overview *Source*: http://en.wikibooks.org/w/index.php?oldid=2249203 *Contributors*: Adrignola, Ldo

Directory Structure *Source*: http://en.wikibooks.org/w/index.php?oldid=2250669 *Contributors*: Adrignola, Ldo

MPEG Format *Source*: http://en.wikibooks.org/w/index.php?oldid=2340852 *Contributors*: Adrignola, Ldo, 1 anonymous edits

IFO Files *Source*: http://en.wikibooks.org/w/index.php?oldid=2292982 *Contributors*: Ldo

Subpicture Streams *Source*: http://en.wikibooks.org/w/index.php?oldid=2250672 *Contributors*: Ldo

NAV PACKs *Source*: http://en.wikibooks.org/w/index.php?oldid=2249216 *Contributors*: Ldo, Panic2k4, 2 anonymous edits

Interaction Machine *Source*: http://en.wikibooks.org/w/index.php?oldid=2336710 *Contributors*: Ldo

Limits *Source*: http://en.wikibooks.org/w/index.php?oldid=1966560 *Contributors*: Adrignola, Ldo

Glossary *Source*: http://en.wikibooks.org/w/index.php?oldid=2072453 *Contributors*: Ldo

License