

MPlayer

[名称](#)[概要](#)[描述](#)[交互式控制](#)[用法](#)[配置文件](#)[配置集](#)[通用选项](#)[播放器选项 \(仅适用于MPLAYER\)](#)[分离器/媒体流选项](#)[OSD/字幕选项](#)[音频输出选项 \(仅适用于MPLAYER\)](#)[音频输出驱动 \(仅适用于MPLAYER\)](#)[视频输出选项 \(仅适用于MPLAYER\)](#)[视频输出驱动 \(仅适用于MPLAYER\)](#)[解码/过滤器选项](#)[音频过滤器](#)[视频滤镜](#)[通用编码选项 \(仅用于MENCODER\)](#)[特定编解码器专用的编码选项 \(仅用于MENCODER\)](#)[环境变量](#)[相关文件](#)[MPLAYER的使用范例](#)[MENCODER的使用范例](#)[程序错误](#)[程序的作者](#)

名称

mplayer — 电影播放器

mencoder — 电影编解码器

概要

mplayer [选项] [文件|URL|播放列表|-]

mplayer [选项] 文件1 [指定选项] [文件2] [指定选项]

mplayer [选项] {文件和选项组} [组指定选项]

mplayer [dvd|dvdnav]://[标题|[开头标题]-末尾标题] [选项]

mplayer vcd://轨迹[/设备]

mplayer tv://[频道][input_id] [选项]

mplayer radio://[频道|频率][capture] [选项]

mplayer pvr:// [选项]

mplayer dvb://[卡号@]频道 [选项]

mplayer mf://[文件掩码@列表文件] [-mf 选项] [选项]

mplayer [cdda|cddb]://[轨迹[-末轨迹][:速度][设备] [选项]

```
mplayer cue://文件[:轨迹] [选项]
mplayer [file|mms[t]|http|http_proxy|rt[s]|ftp|udp|unsv|smb]:// [用户名:密码@]URL[:端口] [选项]
mplayer sdp://文件 [选项]
mplayer mpst://主机[:端口]/URL [选项]
mplayer tivo://主机/[list|l|list|fsid] [选项]
gmplayer [选项] [-skin 皮肤]
mencoder [选项] 文件 [文件|URL|-] [-o 文件 | file://文件 | smb://[用户名:密码@]主机/文件路径]
mencoder [选项] 文件1 [指定选项] [文件2] [指定选项]
```

描述

MPlayer 是一个 LINUX 下的电影播放器 (也能运行在许多其它的平台和 CPU 架构, 请参阅文档)。通过许多原生的和可执行的编解码器的支持, 它能播放绝大部分的 MPEG/VOB, AVI, ASF/WMA/WMV, RM, QT/MOV/MP4, Ogg/OGM, MKV, VIVO, FLI, NuppelVideo, yuv4mpeg, FILM 和 RoQ 文件。 你还能观看 VCD, SVCD, DVD, 3ivx, DivX 3/4/5, 甚至 WMV 电影。

MPlayer 广泛地支持音视频输出驱动。 它不仅可以使用 X11, Xv, DGA, OpenGL, SVGAlib, fbdev, AALib, libcaca, DirectFB, Quartz, Mac OS X CoreVideo, 也能使用 GGI, SDL (及它们的所有驱动), 所有 VESA 兼容显卡上的 VESA (甚至不需要 X11), 某些低级的 显卡相关的驱动 (如 Matrox, 3dfx 及 ATI) 和一些硬件 MPEG 解码器卡, 比如 Siemens DVB, Hauppauge PVR (IVTV), DXR2 和 DXR3/Hollywood+。 它们中绝大多数支持软件或硬件缩放, 所以你可以享受全屏电影。

MPlayer 具有 OSD(屏上显示) 功能显示状态信息, 有 抗锯齿带阴影的漂亮大字幕和键盘控制的可视反馈。 支持的字体包括欧洲语种/ ISO8859-1,2 (匈牙利语, 英语, 捷克语等等), 西里尔语和韩语, 可以播放 12 种格式的字幕文件 (MicroDVD, SubRip, OGM, SubViewer, Sami, VPlayer, RT, SSA, AQTitle, JACOSub, PJS 及我们自己的: MPsub) 和 DVD 字幕 (SPU 流, VOBsub 及隐藏式 CC 字幕)。

mencoder (MPlayer 的电影编解码器) 是一个简单的电影编解码器, 其设计为将 MPlayer 可播放的电影(见上所述)编码成其他的 MPlayer 可播放格式(见下面)。 其编码成的 MPEG-4 (DivX/Xvid), 是 libavcodec 编解码器的一种和 使用 1, 2 或 3-passes(通) 的 PCM/MP3/VBRMP3 音频。 此外还具有流复制的能力, 拥有强大的(过滤器)插件系统(裁剪, 扩展, 反转, 后处理, 旋转, 缩放, 降噪声, RGB/YUV 转换)等。

gmplayer 是具有图形用户界面的 MPlayer。其选项与 MPlayer 相同。

本手册页最后有用法示例, 能让你迅速上手。

也请参阅 HTML 文档!

交互式控制

MPlayer 有个完全可配置的, 命令驱动的控制层, 让你通过键盘, 鼠标, 操纵杆或遥控器(需使用 LIRC)来控制 MPlayer。 参见 `-input` 选项定制控制层的方法。

键盘控制

```
<- 和 ->          后退/ 快进 10 秒。
up 和 down        快进/ 后退 1 分。
pgup 和 pgdown    快进/ 后退 10 分。
```

[和]	减少/增加当前回放速度 10%。
{ 和 }	减半/加倍当前回放速度。
backspace	重置正常回放速度。
< 和 >	在播放列表中后退/ 快进。
ENTER	在播放列表中快进, 甚至越过末尾。
HOME 和 END	下一个/ 前一个父列表中的播放树项。
INS 和 DEL (仅用于 ASX 播放列表)	下一个/ 前一个供选择的源。
p / SPACE	暂停 (再按取消暂停)。
.	步进。 按一下暂停回放, 每按一下就播放一帧(直到按其它键取消步进)。
q / ESC	停止播放和退出。
+ 和 -	调整音频延迟增加/减少 0.1 秒。
/ 和 *	降低/ 提高音量。
9 和 0	降低/ 提高音量。
(和)	调整有利用于左/ 右声道的声音平衡。
m	静音。
_ (仅用于MPEG-TS和libav库格式)	循环可用的像轨。
# (仅用于DVD、MPEG、Matroska、AVI和libav库格式)	循环可用的音轨。
TAB (仅用于MPEG-TS)	循环可用的节目。
f	切换全屏 (也请参见 -fs)。
T	切换顶置 (也请参见 -ontop)。
w 和 e	降低/ 提高摇移范围。
o	循环 OSD 状态: 无 / 搜索条 / 搜索条 + 计时器 / 搜索条 + 计时器 + 总时间。
d	循环丢帧状态: 无 / 跳过显示 / 跳过解码。 (请参见 -framedrop 和 -hardframedrop)。
v	切换显示字幕。

j
循环可用的字幕。

y 和 g
在字幕列表中步进/步退。

F
切换显示 "锁定的字幕"。

a
切换对齐字幕: 上 / 中 / 下。

x 和 z
调整字幕延迟增加/减少 0.1 秒。

r 和 t
上/下调整字幕位置。

i (仅用于 `-edlout` 模式)
设置 EDL 的开头或末尾并写入指定的文件。

s (仅用于 `-vf 截屏`)
截屏。

S (仅用于 `-vf 截屏`)
开始/停止截屏。

l
用 OSD 显示文件名。

! 和 @
寻到前一章节/后一章节的开始处。

D (仅用于 `-vo xvmc`, `-vf yadif`, `-vf kerndeint`)
激活/撤销去交错。

(以下的按键仅适用于硬件加速视频输出(xv, (x)vidix, (x)mga, 等), 软件均衡器(`-vf eq` or `-vf eq2`), 或者色调过滤器(`-vf hue`)).)

1 和 2
调整对比度。

3 和 4
调整亮度。

5 和 6
调整色调。

7 和 8
调整饱和度。

(以下的按键仅适用于 quartz or macosx 视频输出驱动。)

command + 0
半尺寸。

command + 1
原始尺寸。

command + 2
双倍尺寸。

command + f
切换全屏 (也请参见 `-fs`)。

command + [和 command +]
设置窗口的 alpha。

(以下的按键仅适用于 sdl 视频输出驱动。)

c
切换可用的全屏模式。

n

原始尺寸。
(以下的按键仅适用于多媒体键盘。)

PAUSE
暂停。

STOP
停止并退出。

PREVIOUS 和 NEXT
后退/ 快进 1 分。
(以下的按键仅适用于已编译支持 GUI 并优先于以前的定义。)

ENTER
开始播放。

ESC
停止播放。

l
加载文件。

t
加载字幕。

c
打开皮肤浏览器。

p
打开播放列表。

r
打开首选项。
(以下的按键仅适用于已编译支持 TV 或 DVB 的输入并优先于以前的定义。)

h 和 k
选择上一个/ 下一个频道。

n
改变制式。

u
改变频道列表。
(以下的按键仅适用于已编译支持 dvdnav: 用于导航菜单)

keypad 8
选择上。

keypad 2
选择下。

keypad 4
选择左。

keypad 6
选择右。

keypad 5
返回主菜单。

keypad 7
返回最近的菜单(先后次序是: 章节->标题->根)。

keypad ENTER
确定。
(以下按键只有当编译时启用对于电视字幕的支持 (teletext) 时才有效: 这些按键用于控制电 视字幕。)

X
打开/
关闭电视字幕。

Q和W

切换至下一个 / 上一个电视字幕页面。

鼠标控制

button 3 和 button 4

后退 / 快进 1 分。

button 5 和 button 6

降低 / 提高音量。

操纵杆

left 和 right

后退 / 快进 10 秒。

up 和 down

快进 / 后退 1 分。

button 1

暂停。

button 2

循环 OSD 状态: 无 / 搜索条 / 搜索条 + 计时器 / 搜索条 + 计时器 + 总时间。

button 3 和 button 4

降低 / 提高音量。

用法

每个 "flag" 选项有个相反的 "noflag" 选项对应, 如: 选项 `-fs` 相反的是 `-nofs`。

如果某选项标为: (仅用于 XXX), 就仅适用于与选项 XXX 联合使用或者已编译进 XXX。

注意: 子选项解析器(例如用于 `-ao pcm` 子选项)支持 一种特殊的字符串转义(string-escaping)意欲和外部的 GUI 一起使用。

其格式如下:

`%n%string_of_length_n`

示例:

`mplayer -ao pcm:file=%10%C:test.wav test.avi`

或在脚本中:

`mplayer -ao pcm:file=%'expr length "$NAME"'%"$NAME" test.avi`

配置文件

你可以把所有选项放在配置文件中, 每次 MPlayer/MEncoder 运行时都会读取它们。系统范围的配置文件 "mplayer.conf" 位于你的配置目录 (例如: `/etc/mplayer` 或者 `/usr/local/etc/ mplayer`)中, 用户指定的配置文件是 "`~/.mplayer/config`"。MEncoder 的配置文件是 "mencoder.conf" 位于你的配置目录 (例如: `/etc/mplayer` 或者 `/usr/local/etc/ mplayer`)中, 用户指定的配置文件是 "`~/.mplayer/mencoder.conf`"。用户指定的选项优先于系统范围的选项, 而命令行选项优先于以上两者。配置文件的语法是 "option=<值>", 任何 "#" 后面的都认为是注释。启用没有参数值的选项可以把参数设为 "yes" 或者 "1" 或者 "true", 而如果要停用就把它们设置为 "no" 或者 "0" 或者 "false"。甚至子选项也可以如此设定选项。

你也可以为特定文件写配置文件。如果你希望一个特定文件 "movie.avi" 有个配置文件, 就新建文件, 名为 "movie.avi.conf", 写上该特定文件的选项并放在目录 `~/.mplayer/` 下。你也可以把配置文件放在与播放该特定文件时该特定文件相同的目录之下, 只要你(要么在命令行或者在全局配置文件

中)给定 `-use-filedir-conf` 选项。

MPLAYER 配置文件示例:

```
# 默认使用 Matrox 驱动。
vo=xmga
# 我喜欢在看片子的时候练习倒立。
flip=yes
# 解码/编码多个 PNG 文件,
# 开始于 mf://文件掩码
mf=type=png:fps=25
# 怪异的负片图像好酷。
vf=eq2=1.0:-0.8
```

MENCODER 配置文件示例:

```
# 产生 MEncoder 输出到默认的文件名。
o=encoded.avi
# 下面四行使 mencoder tv:// 立刻开始捕捉。
oac=pcm=yes
ovc=lavc=yes
lavcopts=vcodec=mjpeg
tv=driver=v4l2:input=1:width=768:height=576:device=/dev/video0:audiorate=48000
# 较复杂的默认编码选项设置
lavcopts=vcodec=mpeg4:autoaspect=1
lameopts=aq=2:vbr=4
ovc=lavc=1
oac=lavc=1
passlogfile=pass1stats.log
noautoexpand=1
subfont-autoscale=3
subfont-osd-scale=6
subfont-text-scale=4
subalign=2
subpos=96
spuaa=20
```

配置集

为了方便操作不同的配置文件,可以在配置文件中定义配置集。配置集以放在方括号中的名称开始,如: "[my-profile]"。所有跟随的选项成为配置集的一部分。描述(通过 `-profile help` 显示)定义于 `profile-desc` 选项。要结束该配置集,使用别的名称开始即可 或使用配置集名称 "default" 继续正常的选项。

MENCODER 配置集示例:

```
[mpeg4]
profile-desc="MPEG4 encoding"
ovc=lavc=yes
lavcopts=vcodec=mpeg4:vbitrate=1200
```

```
[mpeg4-hq]
profile-desc="HQ MPEG4 encoding"
profile=mpeg4
lavcopts=mbd=2:trell=yes:v4mv=yes
```

通用选项

- `-codecs-file <文件名>` (也请参见 `-afm`, `-ac`, `-vfm`, `-vc`)
重载标准的搜索路径并使用指定的文件代替内置的 `codecs.conf`。
- `-include <配置文件>`
指定配置文件在默认的配置文件的后面解析。
- `-list-options`
打印所有可用的选项。
- `-msgcharset <字符集>`
转换控制台消息为指定的字符集(默认是自动侦测: `autodetect`)。文字的字符集编码用选项 `--charset` 指定配置。设置为 `"noconv"` 以停用转换 (为了如: `iconv` 问题)。
注意: 选项在命令行解析完成后生效。环境变量 `MPLAYER_CHARSET` 能帮你去除开头几行混乱的输出。
- `-msglevel <all=<级别>:<模块>=<级别>:...>`
直接对各个模块控制冗赘度。此 `"all"` 模块改变所有模块(虽然没有在命令行中显式指定)的冗赘级别。查看所有模块的列表请参见于 `"-msglevel help"`。
注意: 消息是在命令行解析之前打印的, 仅受到能应用于所有模块的环境变量 `MPLAYER_VERBOSE` 的控制。
可用级别:

完全沉默	-1
仅致命消息	0
错误消息	1
警告消息	2
简短提示	3
信息消息	4
状态消息 (隐藏用 <code>-quiet</code>)	5
冗赘消息	6
调试级别 2	7
调试级别 3	8
调试级别 4	9
- `-quiet`

使得控制台消息少输出; 特别地, 阻止状态行 (即 A: 0.7 V: 0.6 A-V: 0.068 ...) 的显示。
对慢机器或者不能正确处理回车符(即 \r)的旧机器特别有用。

—priority <优先> (仅适用于 Windows)

根据 Windows 下可用的预定义优先级设置 MPlayer 的进程优先级。 可能的 <优先> 值为:

idle|belownormal|normal|abovenormal|high|realtime

警告: 使用 realtime(实时) 优先级会引起系统锁住。

—profile <配置轮廓1,配置轮廓2,...>

使用给定的配置轮廓, —profile help 显示定义的配置轮廓列表。

—really-quiet (也请参见 —quiet)

显示比 —quiet 更少的输出和状态信息。 同时禁止图形界面错误消息对话框。

—show-profile <配置轮廓>

显示配置轮廓的描述和内容。

—use-filedir-conf

在文件播放时的相同目录之下寻找该特定文件的配置文件。

警告: 播放未经信任的媒体可能有危险。

—v

提高冗赘级别, 对命令行中发现的各个 —v 提高一级。

播放器选项 (仅适用于MPLAYER)

—autoq <质量> (和 —vf [s]pp 一起使用)

依据可用的 CPU 空闲时间动态改变后期处理的级别。 你指定的数字可能是已使用中的最高级别。通常你可以使用有点高的级别数字。为此你必须使用不带参数的 —vf [s]pp 选项。

—autosync <因子>

基于音频延迟的检测逐步调整 A/V 同步。 指定 —autosync 0, 也就是默认值, 将导致帧计时完全基于音频延迟的检测。 指定 —autosync 1 也是一样, 但会微妙地改变所使用的 A/V 修正算法。对那些视频帧率不均匀, 但用 —nosound 可以正常播放的电影, 设置大于 1 的整数值一般会有所帮助。这个值越大, 计时方法越接近于 —nosound。对于某些音频延迟检测实现不完善的声卡驱动, 试试用 —autosync 30 来平滑此问题。使用这个值时, 如果出现 A/V 同步大偏移, 只需要 1, 2 秒时间就可以搞定。对于所有的声卡驱动, 对突然的 A/V 偏移, 在反应时间上有延迟, 这应该是打开此选项的唯一副作用。

—benchmark

在终端打印一些回放后的 CPU 使用率和丢帧数的统计信息。与 —nosound 和 —vo null 联合使用可用来仅仅评测视频编解码器。

注意: 使用此参数仅仅回放视频时, MPlayer 也将忽略帧持续期(frame duration) (你可以认为那是帧率)。

—colorkey <号码>

改变 colorkey 为你所选的 RGB 值。0x000000 是黑色而 0xffffffff 是白色。仅被视频输出驱动 cvidix, fbdev, svg, vesa, winvidix, xmga, xv, xvid, xover, xv (参见 —vo xv:ck), xvmc (参见 —vo xv:ck) 和 directx 支持。

—nocolorkey

停用 colorkeying。仅被视频输出驱动 cvidix, fbdev, svg, vesa, winvidix, xmga, xv, xvid, xover, xv (参见 —vo xv:ck), xvmc (参见 —vo xv:ck) 和 directx 支持。

—correct-pts (试验)

切换 MPlayer 到试验模式, 其中视频帧的时戳计算不同, 且支持视频过滤器添加新的帧或修改已有的时戳。可以看到更精确的时戳, 例如通过 —ass 选项修改播放字幕与镜头同步。没有 —correct-pts, 字幕同步典型地会被某些帧关闭。此选项不能与某些分路器

和编解码器一起正常工作。

-crash-debug (调试代码)

一旦崩溃或 SIGTRAP 自动联上 gdb 调试程序。此支持必须已编译进配置 `--enable-crash-debug`。

-doubleclick-time

以毫秒为单位的时间值，用以判定两次连续的按键动作是否为一次双击动作（默认值：300）。设为0则让你的窗体系统判断什么样动作的是双击（仅当 `-vo directx` 时）。注意：你会体验到不同的效果，这取决于你绑定了 `MOUSE_BTN0_DBL` 还是 `MOUSE_BTN0-MOUSE_BTN0_DBL`。

-edlout <文件名>

新建文件并写入 EDL(编辑决定列表) 记录。回放时，用户击 "i" 键标记跳过块的开始结束处。这提供了用户一个以后能精调 EDL (记录)项的开始点。详情请参见 <http://www.mplayerhq.hu/DOCS/HTML/en/edl.html>。

-enqueue (仅用于 GUI)

将命令行中给定的文件排入播放列表但不立刻播放。

-fixed-vo

为多个文件锁定一个固定的视频系统 (对所有的文件(反)初始化仅一次)。这样，对所有文件只打开一个窗口。目前 `fixed-vo` 兼容的驱动有: `gl`, `gl2`, `mga`, `svga`, `x11`, `xmga`, `xv`, `xvidix` 和 `dfbmga`。

-framedrop (也请参见 `-hardframedrop`)

跳过显示一些帧从而在慢的机器上保持 A/V 同步。视频滤镜不会应用到这些帧上。对于 B 帧 甚至解码也完全跳过。

-(no)gui

启用或停用 GUI 接口(默认取决于可执行文件名)。仅能作为命令行第一个参数。不能作为选项写入配置文件。

-h, -help, --help

显示简短的选项概要。

-hardframedrop

更高强度的丢帧(破坏解码)。导致图像失真!

-identify

是 `-msglevel identify=4` 的简式。使用易于分析的格式显示文件参数。也打印更详细的字幕信息和音轨语言和 ID。在某些情形下，你使用 `-msglevel identify=6` 能得到更多的信息。例如，对于 DVD 能列出每个标题的时长和碟片 ID 号。`wrapper`(封套) 脚本 `TOOLS/midentify` 能抑制 MPlayer 的其它输出 并(希望能) `shellessapes`(脚本转义) 文件名。

-idle (也请参见 `-slave`)

当没有文件播放时，让 MPlayer 空闲等待而不是退出。一般在 `slave` 模式时有用，能通过输入命令控制 MPlayer。

-input <命令>

此选项可以用来配置输入系统的某些部分。路径相对于 `~/.mplayer/`。
注意: 自动重复功能目前只有游戏操纵杆支持。

可用的命令有:

`conf=<文件名>`

指定输入配置文件而不是默认的 `~/.mplayer/input.conf`。如果没有给出完全路径名，将假设是 `~/.mplayer/` <文件名>。

`ar-delay`

在开始自动重复一个键之前等待多少毫秒 (0 代表停用)。

`ar-rate`

自动重复数是每秒重复多少次按键。

- keylist
列出所有可以被绑定于命令的键。
- cmdlist
列出所有可以被绑定于键的命令。
- js-dev
指定使用的操纵杆设备(默认是: /dev/input/js0)。
- file=<文件名>
从给定文件中读取命令。一般结合 FIFO 时有用。
注意: 当给定文件是 FIFO 时 MPlayer 打开两个端头, 这样你就可以做几次 'echo "seek 10" > mp_pipe' 而管道仍保持有效。
- key-fifo-size <2-65000>
指定 FIFO 大小供缓冲按键事件 (默认值: 7)。n 大小的 FIFO 能容纳 (n-1) 缓冲事件。如果太小会丢失一些事件 (导致 "鼠标键卡住" 或类似效果)。如果太大, MPlayer 处理缓冲的事件时可能看上去挂住了。为得到同以前一样的行为效果而引入此选项, 对 Linux 设置为 2 或对 Windows 为 1024。
- lircconf <文件名> (仅适用于 LIRC)
为 LIRC 指定配置文件(默认是: ~/.lircrc)。
- list-properties
打印可用的属性列表。
- loop <数>
重复回放电影 <数> 遍。0 表示重复不断。
- menu (仅用于 OSD 菜单)
打开 OSD 菜单支持。
- menu-cfg <文件名> (仅用于 OSD 菜单)
使用另一种的 menu.conf。
- menu-chroot <路径> (仅用于 OSD 菜单)
将文件选择菜单的根目录更改为一个特定位置。
- 示例:
-menu-chroot=/home
将限制文件选择菜单查找 /home 以及其下的目录 (即不可能访问 /, 但可以访问 /home/user_name)。
- menu-keepdir (仅用于 OSD 菜单)
文件浏览器从最近浏览位置开始浏览, 而不是当前目录。
- menu-root <参数值> (仅适用于 OSD 菜单)
指定主菜单。
- menu-startup (仅适用于 OSD 菜单)
在 MPlayer 启动时显示主菜单。
- mouse-movements
允许 MPlayer 接受指针事件来自视频输出驱动(目前仅支持 X11 衍生型)。必须选择 DVD 菜单中的按钮。
- noconsolecontrols
阻止 MPlayer 读取标准输入的键盘事件。读取标准输入的数据时有用。命令行中发现 - 会自动启用。会遇到不得不手动设置的情形, 比如你打开 /dev/stdin (或者你系统中等价的设备), 在播放列表中使用 stdin 或者意欲以后通过 slave 命令的 loadfile 或 loadlist 从 stdin 读取。
- nojoystick
关闭支持操纵杆。
- nolirc
关闭支持 LIRC。

- nomouseinput**
停用鼠标键 按/放 输入 (mplayerxp 的上下文菜单依赖此选项)。
- rtc** (仅适用 RTC)
打开使用 Linux 的 RTC (实时钟 - /dev/ rtc) 作为计时机制。唤醒过程每 1/1024 秒核查当前时间。除非现代 Linux 内核配置为桌面应用, 因为当使用正常的定时睡眠时, 已唤醒相似精度的过程。
- playing-msg** <字符串>
开始回放前打印出字符串。支持以下的扩展:
 - `$(NAME)`
扩展为属性 NAME 的值。
 - `$(NAME:TEXT)`
仅当属性 NAME 可用时扩展 TEXT(文字)。
- playlist** <文件名>
根据播放列表播放文件(ASX, Winamp, SMIL, 或者 one-file-per-line(每行一个文件) 格式)。
注意: 此选项作为条目仅应用于播放列表的元素以后, 会找到选项。
FIXME: 需要彻底理清并建立文档。
- rtc-device** <设备>
为 RTC 计时使用指定的设备。
- shuffle**
以随机顺序播放文件。
- skin** <名> (仅用于 GUI)
从默认皮肤目录 /usr/local/share/mplayer/skins/ 或 ~/.mplayer/ skins/ 下的, 作为参数的给定目录中装载皮肤。

示例:
 - `-skin fittyfene`
尝试 /usr/local/share/ mplayer/skins/fittyfene 然后是 ~/.mplayer/skins /fittyfene。
- slave** (也请参见 `-input`)
切换到 slave 模式, MPlayer 为其他程序在后台运行。MPlayer 从标准输入读取以新行 (\n) 分隔开的命令行, 而不再截获键盘事件。
注意: 请参见 `-input cmdlist` 的命令列表和 DOCS/tech/slave.txt 中的描述。
- softsleep**
通过不断核查当前时间来测定帧的时间, 而不是请求内核在适当的时刻唤醒 MPlayer。对于内核计时器精度不高且不能使用 RTC 时有用, 代价是较高的 CPU 消耗。
- ssstep** <数>
每帧后跳过 <数> 秒。影片的正常帧率仍被保留, 所以回放加速了。既然 MPlayer 仅能搜寻到下一个 keyframe, 加速过程可能不精准。

分离器/媒体流选项

- a52drc** <级别>
选择 AC3 音频流的动态范围压缩(DRC)级别。<级别> 是 0 到 1 的浮点数, 其中 0 表示无压缩, (默认值) 1 表示全压缩 (使得响亮的通路更安静或相反)。此选项仅当 AC3 流包含所需范围的压缩信息时显示效果。
- aid** <标识> (也请参见 `-alang`)
选择音频频道 (MPEG: 0-31, AVI/OGM: 1-99, ASF/RM: 0-127, VOB(AC3): 128-159, VOB(LPCM): 160-191, MPEG-TS 17-8190)。MPlayer 在 verbose (-v) 模

式下会打印可用的音频标识。当播放 MPEG-TS 流时, MPlayer/MEncoder 用第一个程序播放已选择的(如果有的话)音频流。

-alang <语言代号[,语言代号,...]> (也请参见 **-aid**)

指定使用音频语言的优先级列表。不同的容器格式使用不同的语言代号。DVDs 使用 ISO 639-1 的两字符语言代号, Matroska、MPEG-TS 和 NUT 使用 ISO 639-2 的三字符语言代号, 而 OGM 使用任意格式标识符。MPlayer 在 verbose (**-v**) 模式下会打印可用的语言。

示例:

```
mplayer dvd://1 -alang hu,en
```

选择 DVD 上的匈牙利语音轨播放而英语在没有匈牙利语时备用。

```
mplayer -alang jpn example.mkv
```

播放日语 Matroska 文件。

-audio-demuxer <[+]名称> (仅用于 **-audiofile**)

锁定用于 **-audiofile** 的音频分路器类型。在名称前使用 "+" 锁定类型, 跳过某些核查! 按照 **-audio-demuxer help** 打印出来的指定分路器类型。为了向后兼容, 也接受定义在 libmpdemux/demuxer.h 的分路器标识。 **-audio-demuxer audio** 或 **-audio-demuxer 17** 锁定 MP3。

-audiofile <文件名>

在看电影时播放外部音频文件(WAV, MP3 或 Ogg Vorbis)。

-audiofile-cache <千字节>

对 **-audiofile** 的文件流启用缓存, 使用指定大小的内存。

-reuse-socket (仅适用于 udp://)

允许 SOCKET 在关闭后立即被其它进程重新利用。

-bandwidth <参数值> (仅适用于网络)

指定网络流的最大带宽(用于服务器可以以不同带宽传送内容的情况)。当你以慢速连接观看现场流媒体时有用。对于 Real RTSP 流, 也用来设置最大的传送带宽 以容许较快的缓存填充和流转储。

-cache <千字节>

此选项指定用多少内存(以 kBytes 为单位)作为播放文件或 URL 的预缓冲。对速度慢的媒体特别有用。

-nocache

关闭缓冲。

-cache-min <百分数>

当缓冲填充到总数的 <百分数> 时开始回放。

-cache-seek-min <百分数>

如果查寻到的位置处于从当前位置算起的缓存大小的 <百分数> 之内, MPlayer 等待填充缓存到此位置而不是执行流查寻(默认是: 50)。

-cdda <选项1:选项2> (仅适用于 CDDA)

此选项用来调整 MPlayer 的 CD 音频读取特性。

可用选项有:

speed=<参数值>

设定 CD 转速。

paranoia=<0-2>

设定谨慎级别。0 以外的值可能会打断除了第一个音轨的任何回放。

0: 关闭检测 (默认)

1: 只进行重叠检测

2: 完全数据修正和校检

generic-dev=<参数值>

使用指定的通用 SCSI 设备。

sector-size=<参数值>

单位读取量。

overlap=<参数值>

锁定校检时的最小重叠搜索为 <参数值> 个扇区。

toc-bias

假定 TOC 报告中第1音轨的起始偏移量为 LBA 0。有些东芝光驱需要这样来修正音轨边界。

toc-offset=<参数值>

报告音轨时在扇区数上再加上 <参数值> 个扇区。可能是负数。

(no)skip

(不)接受不完整的数据重建。

-cdrom-device <设备路径>

指定 CD-ROM 设备 (默认是: /dev/ cdrom)。

-channels <数字> (也请参见 -af channels)

请求播放的声道数 (默认是: 2)。MPlayer 要求解码器把音频解码到指定数量的声道。然后由解码器来实现这个要求。此选项通常只有在播放带 AC3 音频(比如 DVD)的视频时才显得重要。在那时默认使用 liba52 解码并把音频正确地混合到需要的声道数。为了直接控制输出声道数独立于解码声道数, 使用声道过滤器。

注意: 此选项可以被解码器(仅适用于 AC3), 滤镜镜(环绕立体声), 和音频 输出驱动(至少 OSS 可以)接受。

可用选项有:

立体声

2

环绕立体声

4

6

5.1

立体声

-chapter <场景标识>[-<末尾场景标识>] (仅适用于 dvd:// 和 dvdnav://)

指定从哪个场景开始播放。也可以指定在哪个场景结束播放 (默认值为: 1)。

-cookies (仅适用于网络)

HTTP 请求时发送 cookies。

-cookies-file <文件名> (仅适用于网络)

从 <文件名> 中读取 HTTP cookies (默认是: ~/.mozilla/ 和 ~/.netscape/) 而跳过从默认位置读取。假定文件是 Netscape 格式。

-delay <数>

音频延迟数秒 (正或负的浮点数)

负值延迟音频, 而正值延迟视频。注意这与MEncoder的-audio-delay选项正好相反。

注意: 当和 MEncoder 一起用时, 并不保证能与 -ovc copy 一起工作正常; 请使用 -audio-delay。

-ignore-start

忽略 AVI 文件中指定的流开始时间。对于 MPlayer, 用 -audio-delay 选项编码的文件废除了流延迟。在编码时期, 此选项防止 MEncoder 转送原流的开始时间到新的文件; 但是 -audio-delay 选项并不受影响。注意 MEncoder 有时会调整流开始时间以补偿预期的解码延迟, 所以没有测试之前不要使用此选项进行编码。

-demuxer <[+]类型名>

锁定分路器类型。在名称前使用 "+" 锁定类型, 跳过某些核查! 按照 -demuxer help 打印出来的指定分路器类型。为了向后兼容, 也接受定义在 libmpdemux/demuxer.h 的分路器标识。

- dumpaudio (仅用于 MPlayer)
转储原始的音频压缩流到 ./stream.dump (用于 MPEG/AC3, 对大多数其它情形, 转储的文件不可播放)。如果你在命令行中指定多于其中 -dumpaudio, -dumpvideo, -dumpstream 的一个, 只有最后一个生效。
- dumpfile <文件名> (仅用于 MPlayer)
指定 MPlayer 应该转储到的文件。应该与 -dumpaudio / -dumpvideo / -dumpstream 一起使用。
- dumpstream (仅用于 MPlayer)
转储原始流到 ./stream.dump。从 DVD 或网络上抓取的时候有用。如果你在命令行中指定多于其中 -dumpaudio, -dumpvideo, -dumpstream 的一个, 只有最后一个生效。
- dumpvideo (仅用于 MPlayer)
转储原始流到 ./stream.dump (不是十分有用)。如果你在命令行中指定多于其中 -dumpaudio, -dumpvideo, -dumpstream 的一个, 只有最后一个生效。
- dvbin <选项> (仅用于 DVB)
为了替代默认的值, 把下面的参数传递给 DVB 输入模块:
 - card=<1-4>
指定使用卡号 1-4 (默认是: 1)。
 - file=<文件名>
指示 MPlayer 从频道中读取 <文件名>。默认是 ~/.mplayer /channels.conf.{sat,ter,cbl,atsc} (基于卡的类型) 或者最后求助 ~/.mplayer/channels.conf。
 - timeout=<1-30>
放弃试图调谐的最大等候秒数(默认是: 30)。
- dvd-device <设备路径> (仅用于 DVD)
指定 DVD 设备(默认是: /dev/dvd)。你也可以指定目录, 其中包含以前直接从 DVD 拷贝 (比如用 vobcopy)的文件。首先注意使用 -dumpstream 通常是更好的拷贝 DVD 的方法 (请参见示例)。
- dvd-speed <以KB/s为单位的因数或速度值> (仅适用于DVD)
尝试限制DVD的传输速度 (默认值: 0, 代表不作更改)。DVD的基础速率约为 1350KB/s, 所以一台8倍速的驱动器能以最高10800KB/s的速度读取 数据。较低的速度使驱动器工作得更安静, 对于观看DVD片来说, 2700KB/s应该是足够安静且快速了。MPlayer在关闭时将驱动器的速度重置至默认值。数值如果小于100则表示 1350KB/s的倍数, 例如 -dvd-speed 8表示选择10800KB/s。
注意: 你需要用DVD设备的写权限以更改传输速度。
- dvddangle <视角标识> (仅用于 DVD)
有些 DVD 碟片中的场景可以从多个视角观看。这里你可以告诉 MPlayer 使用哪个视角 (默认是: 1)。
- edl <文件名>
回放时启用编辑决定列表(EDL)。根据给定文件的条目视频被跳过, 而音频或静音或不静音。如何使用详情请参见 <http://www.mplayerhq.hu/DOCS/HTML/en/edl.html>。
- endpos <[[hh:]mm:]ss[.ms]|size[b|kb|mb]> (也请参见 -ss 和 -sb)
停止在给定的时间或字节位置。
注意: 字节位置仅为了 MEncoder 而启用但不精确, 因为它只会停在帧的边缘。当与 -ss 选项组合使用时, -endpos 的时间会向前移动 -ss 指定的秒数。

示例:

- endpos 56
56 秒后停止。
- endpos 01:10:00
1 小时 10 分后停止。

`-ss 10 -endpos 56`
1 分 6 秒后停止。
`-endpos 100mb`
只编码 100 MB。

`-forceidx`

强制索引重建。对索引受损的文件(A/V 不同步等等)有用。这使得不可能搜索的文件启用搜索。你能使用 MEncoder 永久性地修复索引(请参见文档)。

注意: 此选项仅对原本支持搜索的媒体有效(即不能是: 标准输入, 管道, 等)。

`-fps <浮点数>`

重载帧速率。如果原先的值是错误的或缺少的有用。

`-frames <参数值>`

只播放/ 转换前<参数值>帧, 然后退出。

`-hr-mp3- seek (仅用于 MP3)`

高精度 MP3 搜索。播放外部 MP3 文件时启用, 因为我们需要搜索到非常精确的位置来保持 A/V 同步。这种方法在后退搜索时尤其慢, 因为需要绕回到开头来寻找准确的帧位置。

`-idx (也请参见 -forceidx)`

在没有找到索引的情况下重建文件的索引, 从而允许搜索。对于损坏的/ 不完整的下载, 或制作低劣的文件有用。

注意: 此选项仅对原本支持搜索的媒体有效(即不能是: 标准输入, 管道, 等)。

`-noidx`

跳过重建索引文件的步骤。MEncoder在此选项下跳过写入索引的步骤。

`-ipv4-only-proxy (仅适用于网络)`

对于 IPv6 地址跳过代理服务器。仍然用于 IPv4 连接。

`-loadidx <索引文件>`

从 `-saveidx` 保存的此文件中, 读取视频索引数据。索引用于搜索, 过载其它任何包含在 AVI 本身的索引。MPlayer 不阻止你装载产生于不同 AVI 的索引文件, 但是的确会导致不利的结果。

注意: 既然 MPlayer 现在支持 OpenDML, 此选项已废弃。

`-mc <每帧秒数>`

每帧的最大 A-V 同步修正(以秒为单位)。

`-mf <选项1:选项2:...>`

用来从多个 PNG 或 JPEG 文件中解码。

可用选项有:

`w=<参数值>`

输入文件的宽度(默认是: 自动检测)

`h=<参数值>`

输入文件的高度(默认是: 自动检测)

`fps=<参数值>`

输出的帧率(默认是: 25)

`type=<参数值>`

输入文件的类型(可用类型是: jpeg, png, tga, sgi)

`-ni (仅用于 AVI)`

锁定使用非交错的 AVI 分析器(修复某些质量差的 AVI 文件的播放)。

`-nobps (仅用于 AVI)`

不使用平均比特率值来维持 A-V 同步。对某些文件头损坏的 AVI 文件有帮助。

`-noextbased`

停用基于后缀名的分路器选择机制。默认情况下, 当文件类型(分路器)无法可靠检测时(文件没有头部或者不够可靠), 使用文件后缀名来选择分路器。基于内容的分路器总是

可作为备用的。

—passwd <密码> (也请参见—user 选项) (仅适用于网络)

设置 http 认证密码。

—prefer-ipv4 (仅适用于网络)

使用 IPv4 网络连接。IPv6 自动地作为备用。

—prefer-ipv6 (仅适用于 IPv6 网络)

使用 IPv6 网络连接。IPv4 自动地作为备用。

—psprobe <字节所在位置>

当播放MPEG-PS流时，此选项让你设定你希望MPlayer查看多少字节以确认所要使用的视频编/解码器。在播放含有H.264流的EVO文件时需要此选项。

—pvr <选项1:选项2:...> (仅适用于 PVR)

此选项调谐 PVR 捕捉模块的各种编码属性。不得与基于硬件 MPEG 编码器 V4L2 驱动支持的卡一起使用。Hauppauge WinTV PVR-150/250/350/500 和所有基于 IVTV 的卡是众所周知的 PVR 捕捉卡。小心只有 Linux 2.6.18 及以上的内核能通过 V4L2 层处理 MPEG 流。对于 MPEG 流硬件捕捉卡并使用 MPlayer/MEncoder 来观看电影的，使用 "pvr://" 作为 URL 地址。

可用选项有：

aspect=<0-3>

指定宽高比：

0: 1:1

1: 4:3 (默认)

2: 16:9

3: 2.21:1

arate=<32000-48000>

指定编码音频频率(encoding audio rate) (默认是: 48000 Hz, 可用的有: 32000, 44100 和 48000 Hz)。

alayer=<1-3>

指定编码 MPEG 音频层 (默认是: 2)。

abitrage=<32-448>

指定音频编码比特率(以 kbps 为单位) (默认是: 384)。

amode=<参数值>

指定音频编码模式。可用的预设值是 "stereo", "joint_stereo", "dual" 和 "mono" (默认是: stereo)。

vbitrate=<参数值>

指定平均编码视频比特率(以 Mbps 为单位) (默认是: 6)。

vmode=<参数值>

指定视频编码模式：

vbr: Variable BitRate (可变比特率) (默认)

cbr: Constant BitRate (常量比特率)

vpeak=<参数值>

指定编码视频比特率峰值(以 Mbps 为单位) (只对 VBR 编码有用, 默认是: 9.6)。

fmt=<参数值>

选择进行编码的 MPEG 格式：

ps: MPEG-2 Program Stream (MPEG-2 程序流) (默认)

ts: MPEG-2 Transport Stream (MPEG-2 传输流)

mpeg1: MPEG-1 System Stream (MPEG-1 系统流)

vcd: Video CD compatible stream (VCD 可兼容流)

svcd: Super Video CD compatible stream (SVCD 可兼容流)

dvd: DVD compatible stream (DVD 可兼容流)

-radio <选项1:选项2:...> (仅适用于收音)

这些选项集用以设置收音捕捉模块的各种参数。要用MPlayer收听电台使用‘radio://<频率>’ (如果不提供频道选项) 或 ‘radio://<频道号>’ (如果提供频道选项) 作为URL。你可以通过以‘-v’选项运行MPlayer查看允许的频率范围。若要启动捕捉子系统, 则使用‘radio://<频率或频道>/capture’。如果不给出关键字capture, 则你可以只使用输入线缆收听电台。由于同步问题, 不推荐使用capture来收听电台, 这会使效果不佳。

可用选项有:

device=<值>

所使用的收音设备 (默认值: Linux下为/dev/radio0, 而*BSD下为/dev/tuner0)。

driver=<值>

所使用的收音驱动 (默认值: 如果有则用v412, 否则为v41)。当前支持v41和v412驱动。

volume=<0..100>

收音设备的音量 (默认为100)

freq_min=<值> (仅适用于*BSD)

允许的最小频率 (默认值: 87.50)

freq_max=<value> (仅适用于*BSD)

允许的最大频率 (默认值: 108.00)

channels=<频率>-<名称>,<频率>-<名称>,...

设置频道列表。在名称中用“_”代替空格 (或使用引号)。频道名字就会以OSD方式显示出来, 而从命令radio_step_channel和radio_set_channel 对于遥控会很有用 (参见LIRC)。如果给出, 则URL中的数字将作为频道列表中的频道位置编号。

示例: radio://1, radio://104.4, radio_set_channel 1

adevice=<值> (需要激活收音捕捉功能)

作为所捕捉声音来源的设备的名字 没有这样的名字将禁用捕捉功能, 即便URL中出现capture关键字。如果设备名中含有一个‘=’, 则模块将使用ALSA来捕捉, 否则用OSS。

arate=<参数值> (需要激活收音捕捉功能)

以每秒采样数为单位的频率值 (默认值: 44100)。

注意: 当使用音频捕捉时, 也将-rawaudio rate=<数值>选项设置为arate中一样的值。如果你存在声音播放速度的问题 (播放太快), 尝试以不同的速率值 (例如4800, 44100, 32000……) 播放。

achannels=<参数值> (需要激活收音捕捉功能)

捕捉声音的声道数。

-rawaudio <选项1:选项2:...>

此选项可让你播放原始音频文件。你也不得不使用 -demuxer rawaudio。也可以用来播放不是 44kHz 16-位 立体声的音频 CD。播放原始 AC3 流请使用 -rawaudio format=0x2000 -demuxer rawaudio。

可用选项有:

channels=<参数值>

声道数

rate=<参数值>

每秒采样率

samplesize=<参数值>

样本大小(以字节为单位)

bitrate=<参数值>
原始音频文件的比特率
format=<参数值>
16-进制的 fourcc

—rawvideo <选项 1: 选项 2:...>

此选项可让你播放原始视频文件。 你也不得不使用 —demuxer rawvideo。

可用的选项有:

fps=<参数值>
每秒帧(速)率 (默认是: 25.0)
sqcif|qcif|cif|4cif|pal|ntsc
设置标准的图像大小
w=<参数值>
图像宽(以像素为单位)
h=<参数值>
图像高(以像素为单位)
i420|yv12|yuy2|y8
设置色彩空间
format=<参数值>
16-进制或字符串常量形式的色彩空间值(fourcc) 使用—rawvideo
format=help以获取可用字符串的列表。
size=<参数值>
帧大小(以字节为单位)

示例:

mplayer foreman.qcif -demuxer rawvideo -rawvideo qcif
播放著名的 "foreman" 视频样本。

mplayer sample-720x576.yuv -demuxer rawvideo -rawvideo w=720:h=576
播放原始 YUV 样本。

—rtsp-port

与 "rtsp://" 的 URL 地址一起用来锁定客户端的端口号。 如果你在路由器后面, 想把 RTSP 流从服务器转到指定客户端, 此选项可能有用。

—rtsp-destination

与 "rtsp://" 的 URL 地址一起用来强制目标绑定 IP 地址。 对某些 RTSP 服务器不能把 RTP 包送到正确的接口, 此选项可能有用。 如果连接 RTSP 服务器失败, 使用 —v 看看 MPlayer 试图绑定到 哪个 IP 地址, 然后试试用它强制分配到你的计算机。

—rtsp-stream-over-tcp (仅适用于 LIVE555)

与 "rtsp://" 的 URL 地址一起用来指定让 RTP 和 RTCP 进入包流过 TCP (使用与 RTSP 相同的 TCP 连接)。 如果你的破 internet 连接不传递 UDP 进入包, 此选项可能有用 (参见 <http://www.live555.com/mplayer/>)。

—saveidx <文件名>

强制重建索引并转储到 <文件名>。 目前只对 AVI 文件有效。

注意: 既然 MPlayer 现在支持 OpenDML, 此选项已废弃。

—sb <比特位置> (也请参见 —ss)

搜索到比特位置。 回放开始部分是垃圾的 CD-ROM 镜像或 VOB 文件时有用。

—speed <0.01–100>

根据给定的参数因子, 减慢或加快回放速度。 并不保证能与 —oac copy 一起工作正常。

—srate <Hz>

选择输出的取样率(当然声卡对此有限制)。 如果选择的样本频率与当前的媒体有差异, 会在音频过滤层中插入 音频过滤器 resample(重采样) 或 lavcresample 来补偿此差异。 重采样的类型由 —af-adv 选项控制。 默认的是快速重采样, 可能会引起失真。

`-ss <时间>` (也请参见 `-sb`)
搜索到给定的时间位置。

示例:

`-ss 56`
搜索到 56 秒处。

`-ss 01:10:00`
搜索到 1 小时 10 分钟处。

`-tskeepbroken`
告诉 MPlayer 不要丢弃流中报告损坏的 TS 包。有时需要播放篡改了的 MPEG-TS 文件。

`-tsprobe <字节位置>`
播放 MPEG-TS 流时, 此选项让你指定流中多少字节给 MPlayer 搜索 你所要的音频和视频的标识。

`-tsprog <1-65534>`
播放 MPEG-TS 流时, 你可以用此选项指定你想运行的程序(如果知道的话)。能和 `-vid` 及 `-aid` 一起使用。

`-tv <选项1:选项2:...>` (仅用于 TV/PVR)
此选项会调谐 TV 捕捉模块的各种属性。为了用 MPlayer 收看 TV, 使用 "tv://" 或 "tv://<频道号>" 甚至用 "tv://<频道名>" (参见下面的频道选项) 作为 URL 地址。

可用选项有:

`noaudio`
没有声音

`automute=<0-255>` (仅适用于v4l和v4l2)
如果设备报告的信号强度小于这个值, 那么音频和视频将静音。大多数情况下`automute=100`就足够用了。默认值为0 (禁用`automute`)。

`driver=<参数值>`
参见`-tv driver=help`以获取内置TV输入驱动力的列表。可用参数: `dummy`, `v4l`, `bsdbt848`

`device=<参数值>`
指定 TV 设备 (默认是: `/dev/video0`)。注意: 对于`bsdbt848`驱动, 你可以同时给出`bktr`和调谐设备的名称, 使用逗号分隔两者, 其中调谐器在`bktr`之后 (例如: `-tv device=/dev/bktr1,/dev/tuner1`)。

`input=<参数值>`
指定输入 (默认是: 0 (TV), 参见控制台输出的可用输入列表)。

`freq=<参数值>`
指定调谐器的频率(例如: 511.250)。与频道参数不兼容。

`outfmt=<参数值>`
指定调谐器的输出格式, 用 V4L 驱动支持的预设置(`yv12`, `rgb32`, `rgb24`, `rgb16`, `rgb15`, `uyvy`, `yuy2`, `i420`) 或者一个以 16-进制值给定的武断的格式。试试用 `outfmt=help` 察看可用的所有格式的列表。

`width=<参数值>`
输出窗口的宽度

`height=<参数值>`
输出窗口的高度

`fps=<参数值>`
捕捉视频的帧速率(帧每秒)

`buffersize=<参数值>`
捕捉缓冲区的最大尺寸(以兆为单位) (默认值: 动态)

norm=<参数值>
bsdbt848 和 v4l 可用的参数: PAL, SECAM, NTSC。对于 v4l2, 参见控制台输出的所有可用制式列表, 也请参看下面的 normid 选项。

normid=<参数值> (仅用于 v4l2)
设置 TV 制式 为给定的数字标识。TV 制式取决于捕捉卡。参见控制台输出的可用 TV 制式列表。

channel=<参数值>
设置调谐器到 <参数值> 频道。

chanlist=<参数值>
可用参数: europe-east, europe-west, us-bcast, us-cable, 等等

channels=<频道>-<名称>,<频道>-<名称>,...
设置频道的名称。 注意: 如果<频道>是一个大于1000的整数, 它将被看作是频率 (以kHz为单位) 而不是频率表中的 频道名。
名称中使用 _ 代替空格(或使用引号 ;-))。 OSD 会显示频道名称, slave 命令 tv_step_channel, tv_set_channel 及 tv_last_channel 会有用于遥控(参见 LIRC)。 与频率参数不兼容。
注意: 频道号就将是 "频道" 列表上从1开始计数的位置。
示例: tv://1, tv://TV1, tv_set_channel 1, tv_set_channel TV1

[brightness|contrast|hue|saturation]=<- 100-100>
设置显卡的色彩均衡器。

audiorate=<参数值>
设置音频捕捉比特率。

forceaudio
即使 v4l 报告没有音频源也捕捉音频。

alsa
从 ALSA 捕捉。

amode=<0-3>
选择音频模式:
0: 单声道
1: 立体声
2: 语言 1
3: 语言 2

forcechan=<1-2>
默认情况下, 音频声道数计数由电视卡查询音频模式自动决定。 此选项允许锁定立体声/ 单声道记录而不管 amode 选项和 v4l 返回的参数。 在电视卡不能报告正确的当前音频模式的时候可以用此选项解决麻烦。

adevice=<参数值>
设置音频设备。 <参数值> 必须是 /dev/xxx 用于 OSS 而硬件标识用于 ALSA。 硬件标识用于 ALSA 时, 你务必要用 "." 代替 ":"。

audioid=<参数值>
选择捕捉卡的音频输出, 如果它有不止一个的话。

[volume|bass|treble|balance]=<0-65535> (v4l1)
[volume|bass|treble|balance]=<0-100> (v4l2)
这些选项用来设置视频捕捉卡上的混音器参数。 如果你的卡没有混音器, 它们将没有效果。 对于 v4l2 50 根据驱动器的报告影射到控制器上的默认值。

immediatemode=<布尔值>
参数值为 0 表示同时捕捉和缓冲音频和视频 (MEncoder 的默认值)。 参数值为 1 (MPlayer 的默认值)表示只捕捉视频 而让音频通过环路电缆由电视

卡输入声卡。

mjpeg

使用硬件 MJPEG 压缩(如果芯片支持的话)。使用此选项时, 你不需要指定输出窗口的宽和高, 因为 MPlayer 会根据 decimation 参数(见下面)自动确定。

decimation=<1|2|4>

选择硬件 MJPEG 压缩的图像尺寸:

1: 全尺寸 704x576 PAL 704x480 NTSC

2: 中等尺寸 352x288 PAL 352x240 NTSC

4: 小尺寸 176x144 PAL 176x120 NTSC

quality=<0-100>

选择 JPEG 压缩的质量(全尺寸推荐值 < 60)。

tdevice=<值> (默认值: none)

指定电视字幕设备 (示例: /dev/ vbi0) 。

tformat=<格式> (默认值: 0)

指定电视字幕的显示格式:

0: 不透明

1: 透明

2: 不透明且反色

3: 透明且反色

tpage=<100-899> (默认值: 100)

指定初始电视字幕的页面号。

-user <用户名> (也请参见 -passwd) (仅用于网络)

指定 HTTP 认证的用户名。

-user-agent <字符串>

使用 <字符串> 作为 HTTP 流的用户代理。

-vid <标识>

选择视频频道 (MPG: 0-15, ASF: 0-255, MPEG-TS: 17-8190)。当播放 MPEG-TS 流时, MPlayer/MEncoder 用第一个程序 播放已选择的(如果有的话)视频流。

-vivo <子选项> (调试代码)

锁定 VIVO 分路器的音频参数(用于调试)。

OSD/字幕选项

注意: 也请参见 -vf expand。

-ass (仅适用于 FreeType)

打开 SSA/ASS 字幕提供。通过此选项, libass 将用于 SSA/ASS 外部字幕和 Matroska 轨迹。你可能也想使用 -embeddedfonts。

-ass-border-color <参数值>

为文字字幕设置边框 (轮廓) 颜色。颜色格式是 RRGGBBAA。

-ass-bottom-margin <参数值>

在帧底部添加黑条。SSA/ASS 提供器能在那里放置字幕 (与 -ass-use-margins 组合用)。

-ass-color <参数值>

为文字字幕设置颜色。颜色格式是 RRGGBBAA。

-ass-font-scale <参数值>

为 SSA/ASS 提供器中的字体设置比例系数。

-ass-force-style <[Style.]Param=参数值[,...]>

重载一些风格参数。

示例:

- ass-force-style FontName=Arial,Default.Bold=1
- ass-hinting <类型>
设置微调类型。
<类型>
 - 0 : 无微调。
 - 1 : FreeType自动微调器, 轻量模式。
 - 2: FreeType自动微调器, 普通模式。
 - 3: 字体内置微调。
 - 0-3 + 4 : 相同, 但微调只有当OSD以全屏方式显示而因此不会伸缩时才会使用。
 - 默认值为7 (当OSD不伸缩时使用内置微调器, 否则无微调)。
- ass-line-spacing <参数值>
为 SSA/ASS 提供器设置行间隔。
- ass-styles <文件名>
装载指定文件中找到的所有 SSA/ASS 风格并在提供文字字幕时使用。 文件的语法完全像是 SSA/ASS 中的一节 [V4 Styles] / [V4+ Styles]。
- ass-top-margin <参数值>
在帧顶部添加黑条。 SSA/ASS 提供器能在那里放置字幕 (与 –ass-use-margins 组合用)。
- ass-use-margins
启用放置字幕(如果有的话)在黑框中。
- dumpjacosub (仅用于 MPlayer)
转换给定的(由 –sub 选项指定的)字幕成为基于时间的 JACOsub 字幕格式。 在当前的目录中创建 dumpsub.js 文件。
- dumpmicrodvds (仅用于 MPlayer)
转换给定的(由 –sub 选项指定的)字幕成为 MicroDVD 字幕格式。 在当前的目录中创建 dumpsub.sub 文件。
- dumpmpsub (仅用于 MPlayer)
转换给定的(由 –sub 选项指定的)字幕成为 MPlayer 的字幕格式, MPsub。 在当前的目录中创建 dumpsub.mpsub 文件。
- dumpsami (仅用于 MPlayer)
转换给定的(由 –sub 选项指定的)字幕成为基于时间的 SAMI 字幕格式。 在当前的目录中创建 dumpsub.smi 文件。
- dumpsrtsub (仅用于 MPlayer)
转换给定的(由 –sub 选项指定的)字幕成为基于时间的 SubViewer (SRT) 字幕格式。 在当前的目录中创建 dumpsub.srt 文件。
注意: 有些破硬件播放器对 Unix 行结尾的 SRT 字幕会哽住。 如果你够彩碰到这样的盒子, 把你的字幕文件到 unix2dos 或类似的程序过一过, 替换 Unix 行结尾为 DOS/Windows 行结尾。
- dumpsub (仅用于 MPlayer) (BETA 代码)
从 VOB 流中转储字幕子流。 也请参见 –dump*sub 和 –vobsubout* 选项。
- embeddedfonts (仅用于 FreeType)
允许抽取 Matroska 内嵌字体 (默认值: 禁止)。 这些字体能用于 SSA/ASS 字幕渲染 (–ass 选项)。 字体文件创建在 ~/.mplayer/ fonts 目录中。
注意: 在 FontConfig 2.4.2 或更新的版本里, 内嵌字体直接从内存中打开而这个选项默认打开。
- ffactor <数字>
对字体的 alphasmap 重新采样。 可设为:

3

-slang <语言代号[,语言代号,...]> (也请参见 **-sid**)
 指定使用字幕语言的优先级列表。不同的容器格式使用不同的语言代号。DVDs 使用 ISO 639-1 的两字符语言代号, Matroska 使用 ISO 639-2 的三字符语言代号, 而 OGM 使用自由格式标识符。MPlayer 在 verbose (**-v**) 模式下会打印可用的语言。

示例:

```
mplayer dvd://1 -slang hu,en
```

选择 DVD 上的匈牙利语字幕轨迹而英语在没有匈牙利语时备用。

```
mplayer -slang jpn example.mkv
```

播放日语字幕的 Matroska 文件。

-spuaa <模式>

设置 DVD/VOBsub 的反锯齿/ 缩放模式。甚至在原始的与缩放的帧尺寸已匹配时, 还可能 在 <模式> 上增加 16 以强制进行缩放, 比如使用高斯柔化来平滑字幕。可用模式有:

不缩放(最快, 很丑)

0

近似缩放(好像坏了?)

1

完全缩放(慢)

2

二次线性缩放(默认值, 快速而且效果不坏)

3

使用软件缩放的高斯柔化(看起来很好)

4

-spualign <-1-2>

指定 SPU (DVD/VOBsub) 字幕该如何对齐。

原始位置。

-1

顶部对齐 (原始行为, 默认)。

0

中央对齐。

1

底部对齐。

2

-spugauss <0.0-3.0>

高斯模糊的可变参数用于 **-spuaa** 4。越高表示越柔化 (默认是: 1.0)。

-sub <字幕文件1,字幕文件2,...>

使用/ 显示指定的字幕文件。同时只能显示一个文件。

-sub-bg-alpha <0-255>

指定字幕和 OSD 背景的 alpha 通道值。值越大代表越透明。但 0 表示完全透明。

-sub-bg-color <0-255>

指定字幕和 OSD 背景的颜色值。目前字幕是灰度图像所以这个值相当于颜色的亮度。255 代表白色 0 代表黑色。

-sub-demuxer <[+]名称> (仅用于 **-subfile**) (BETA 代码)

锁定 **-subfile** 的字幕分路器类型。在名称前使用 "+" 锁定类型, 跳过某些核查! 按照 **-demuxer help** 打印出来的指定分路器类型。为了向后兼容, 也接受定义在 **subreader.h** 的分路器标识。

-sub-fuzziness <模式>

搜寻字幕时调整匹配模糊度:

精确匹配

0

装载所有包含电影名称的字幕。	1
装载当前目录的所有字幕。	2
<code>-sub-no-text-pp</code> 停用载入字幕后的任何形式的文字后期处理。 用于调试目的。	
<code>-subalign <0-2></code> 根据 <code>-subpos</code> 给定的高度, 指定该对齐哪个字幕边缘。	
顶部对齐 (原始行为)。	0
中央对齐。	1
底部对齐 (默认)。	2
<code>-subcc</code> 显示 DVD 的隐藏式(CC)字幕。 它们 不 是 VOB 字幕, 它们是为听力有障碍的人准备的特殊的 ASCII 字幕, 编码在大多数 DVD 地区码为 1 的 VOB 用户数据流中。 CC 字幕到目前为止还没有在其它地区码的 DVD 中发现。	
<code>-subcp <编码页></code> (仅用于 <code>iconv</code>) 如果你的系统支持 <code>iconv(3)</code> , 你可以用此选项来指定字幕的编码页。	
示例: <code>-subcp latin2</code> <code>-subcp cp1250</code>	
<code>-subcp enca:<语言>:<备用编码页></code> (仅用于 ENCA) 你可以使用两字符语言代号指定语言, 让 ENCA 自动检测编码页。 如果不能确定, 随便输入点什么然后察看 mplayer 的 <code>-v output</code> 中可用的语言。 备用编码页指定自动检测失败时使用的编码页。	
示例: <code>-subcp enca:cs:latin2</code> 猜测编码页, 假定字幕是捷克语, 如果自动检测失败 latin 2 作备用。 <code>-subcp enca:pl:cp1250</code> 猜测波兰语的编码页, cp1250 作备用。	
<code>-subdelay <数></code> 字幕延迟 <code><数></code> 秒。可以是负数。	
<code>-subfile <文件名></code> (BETA 代码) 目前没有用。 与 <code>-audiofile</code> 一样, 但用于字幕流(OggDS?)。	
<code>-subfont <文件名></code> (仅用于FreeType字体) 设置字幕字体。 如果没有给出 <code>-subfont</code> , 则使用 <code>-font</code> 。	
<code>-subfont-autoscale <0-3></code> (仅用于 FreeType) 设置自动缩放模式。 注意: 0 表示文字和 OSD 的尺度为字体高度(以点 point 为单位的)。	
可用模式有:	
不自动缩放	0
高度成比例	1
	2

宽度成比例

对角线成比例 (默认值)

- `-subfont-blur <0-8>` (仅用于 FreeType)
设置字体柔化半径(默认是: 2)。
- `-subfont-encoding <参数值>` (仅用于 FreeType)
设置字幕编码。当设为 "unicode" 时, 字体文件中的所有字模都会被渲染 并使用 unicode 编码(默认是: unicode)。
- `-subfont-osd-scale <0-100>` (仅用于 FreeType)
设置 OSD 元素的自动缩放系数(默认是: 6)。
- `-subfont-outline <0-8>` (仅用于 FreeType)
设置字体边框的宽度(默认是: 2)。
- `-subfont-text-scale <0-100>` (仅用于 FreeType)
设置字幕文本的自动缩放系数(屏幕尺寸的百分比) (默认是: 5)。
- `-subfps <速率>`
指定字幕文件的帧率(默认是: 电影的帧率)。
注意: 仅用于基于帧的字幕文件, 即 MicroDVD 格式。
- `-subpos <0-100>` (用于 `-vf expand`)
指定字幕在屏幕上显示的位置。 参数值是以屏幕高度的百分比为单位的字幕垂直位置。
- `-subwidth <10-100>`
指定字幕在屏幕上显示的最大宽度。 对于电视输出(TV-out)有用。 参数值是以屏幕宽度的百分比为单位的字幕宽度。
- `-noterm-osd`
当没有可用的视频输出时, 停止在控制台上显示 OSD 信息。
- `-term-osd-esc <转义序列>`
指定, 在控制台上显示 OSD 信息之前, 使用的转义序列。 转义序列应该把指针移到 OSD 的行首然后清理(默认是: `^[A\r^[K`)。
- `-unicode`
告诉 MPlayer 把字幕文件作为 unicode 来处理。
- `-utf8`
告诉 MPlayer 把字幕文件作为 UTF-8 来处理。
- `-vobsub <无后缀名的 VOBsub 文件名>`
指定用于字幕显示的 VOBsub 文件。 不得不是无后缀名的完整路径名, 即没有 ".idx", ".ifo" 及 ".sub"。
- `-vobsubid <0-31>`
指定 VOBsub 字幕标识。

音频输出选项 (仅适用于MPLAYER)

- `-abs <参数值>` (仅用于 `-ao oss`) (废弃)
重载音频驱动/ 声卡的缓冲区大小检测。
- `-format <格式>` (也请参见音频过滤器一节中的格式)
选择样本格式用于从音频过滤器层输出到声卡。 下面音频过滤器一节中的格式描述列出了可采用的<格式>值。
- `-mixer <设备>`
使用不是默认的 `/dev/mixer` 混音设备。 对于 ALSA 来说是混音名称。
- `-mixer-channel <混音线>[,混音索引]` (仅用于 `-ao oss` 和 `-ao alsa`)
此选项告诉 MPlayer 使用不同的通道控制音量而不是默认的 PCM。 用于 OSS 的选项包括 vol, pcm, line。 有关选项的完全列表请查找 `/usr/include/linux/soundcard.h` 中的

SOUND_DEVICE_NAMES。对于 ALSA (如 alsamixer displays), 你可以使用的名称, 如 Master, Line, PCM。

注意: ALSA 混音通道名称后面跟的数字必须用 <名称,数字> 格式指定, 即 alsamixer 中通道标为 "PCM 1" 的必须转换为 PCM,1。

`-softvol`

锁定使用软件混音, 而不用声卡混音。

`-softvol-max <10.0-10000.0>`

设置最大扩音级别(以百分比为单位) (默认是: 110)。值 200 允许你调整的音量最高达到目前级别的两倍。对于值低于 100 的, 初始音量(为 100%) 会高于最大值, OSD 不能正确显示。

`-volstep <0-100>`

设置混音音量调整的步进单位为整个范围的百分比 (默认是: 3)。

音频输出驱动 (仅适用于MPLAYER)

音频输出驱动是各种音频输出设备的接口。语法是:

`-ao <驱动1[:子选项1[=参数值]:...],驱动2,...[,]>`

指定使用音频输出驱动的优先级列表。

如果列表尾有 ", " MPlayer 使用没有列出的驱动作为后备。子选项是可选的, 大多数省略。

注意: 已编译进的音频输出驱动参见于 `-ao help` 的列表。

示例:

`-ao alsa,oss,`

尝试使用 ALSA 驱动, 然后是 OSS 驱动, 再后是其他的。

`-ao alsa:noblock:device=hw=0.3`

设置 noblock-模式和 device-名称为第一个卡, 第四个设备。

可用的音频输出驱动有:

alsa

ALSA 0.9/1.x 音频输出驱动

`noblock`

设置 noblock-模式。

`device=<设备>`

设置设备名称。在 ALSA 设备名称中替换 ", " 为 ".", ":" 为 "="。对于通过 S/PDIF 输出的 hwac3, 使用 "iec958" 或 "spdif" 设备, 除非你真的知道如何正确设置。

alsa5

ALSA 0.5 音频输出驱动

oss

OSS 音频输出驱动

`<dsp-device>`

设置音频输出设备 (默认是: /dev/dsp)。

`<mixer-device>`

设置音频混音设备 (默认是: /dev/mixer)。

`<mixer-channel>`

设置音频混音通道 (默认是: pcm)。

sdl (仅用于 SDL)

高级平台独立的 SDL(简单 Directmedia 层) 库音频输出驱动

`<driver>`

显式选择 SDL 音频输出 (默认是: 让 SDL 选)。

arts
音频输出通过 aRts daemon

esd
音频输出通过 ESD daemon
 <server>
 显式选择 ESD 服务器 (默认是: localhost)。

jack
音频输出通过 JACK(Jack 音频连接工具)
 port=<名>
 连接给定端口名 (默认是: 物理端口)。
 name=<客户端名>
 传递客户端名给 JACK (默认是: MPlayer [<PID>])。 如果你想自动建立某些连接时有用。
 (no)estimate
 估计音频延迟, 假定让视频回放更流畅 (默认是: 启用)。

nas
 音频输出通过 NAS

macosx (仅用于 Mac OS X)
 原生 Mac OS X 音频输出驱动

openal
试验性的, 未完成的 (会低端混音成单声道) OpenAL 音频输出驱动

sgi (仅用于 SGI)
 原生 SGI 音频输出驱动
 <输出设备名>
 显式选择音频输出设备/ 接口 (默认是: system-wide default)。 例如:
 "Analog Out" 或 "Digital Out"。

sun (仅用于 Sun)
 原生 Sun 音频输出驱动
 <device>
 显式选择音频设备 (默认是: /dev/audio)。

win32 (仅用于 Windows)
 原生 Windows waveout 音频输出驱动

dsound (仅用于 Windows)
 DirectX DirectSound 音频输出驱动
 device=<devicenum>
 设置设备号。 用 -v 播放文件能显示可用设备列表。

dxr2 (也请参见 -dxr2) (仅用于 DXR2)
 Creative DXR2 专用的输出驱动

ivtv (仅用于 IVTV)
 IVTV 专用的 MPEG 音频输出驱动。 只能与 -ac hwmpa 一起用。

v4l2 (需要Linux 2.6.22+的内核)
 带有硬件MPEG解码器的V4L2卡的音频输出驱动。

mpegpes (仅用于 DVB)
 DVB 卡音频输出驱动输出到 MPEG-PES 文件如果没安装 DVB 卡。
 card=<1-4>
 指定 DVB 卡如果有多于一个卡。
 file=<文件名>
 输出文件名

null

	不产生音频输出但维持视频回放速度。 基准测试用 <code>-nosound</code> 。
pcm	原始 PCM/wave 文件记录器的音频输出 (no)waveheader 包括或不包括声波头 (默认是: 包括)。 如果不包括, 会产生原始的 PCM。 file=<文件名> 写声音到 <文件名> 而不是默认的 audiodump.wav。 如果没指定声波头, 默认是 audiodump.pcm。 fast 尝试比实时更快地转储。 确保输出不被截掉 (通常有 "缓冲中视频包太多" 消息)。 你得到消息 "你的系统太"慢"了, 播放不了!" 是正常的。
plugin	音频输出驱动插件

视频输出选项 (仅适用于MPLAYER)

<code>-adapter <参数值></code>	设置显卡接受图像。 用此选项带 <code>-v</code> 能得到可用卡的列表。 目前只能与 <code>directx</code> 视频输出驱动一起工作。
<code>-bpp <深度></code>	重载自动检测的色彩深度。 只被 <code>fbdev</code> , <code>dga</code> , <code>svga</code> , <code>vesa</code> 视频输出驱动支持。
<code>-border</code>	使用窗口边框和装饰来播放电影。 既然默认是启用, 使用 <code>-noborder</code> 停用标准的窗口装饰。 被 <code>directx</code> 视频输出驱动支持。
<code>-brightness <-100-100></code>	调整视频信号的亮度 (默认是: 0)。 不被所有的视频输出驱动支持。
<code>-contrast <-100-100></code>	调整视频信号的对比度 (默认是: 0)。 不被所有的视频输出驱动支持。
<code>-display <名> (仅用于 X11)</code>	指定你希望使用 X server 的主机名和显示号。
示例:	<code>-display xtest.localdomain:0</code>
<code>-dr</code>	打开直接渲染功能(还不被所有的编解码器视频输出支持) 警告: 可能导致 OSD/字幕 损坏!
<code>-dxr2 <选项1:选项2:...></code>	此选项用来控制 <code>dxr2</code> 视频输出驱动。 ar-mode=<参数值> 宽高比模式 (0 = 普通, 1 = pan-and-scan, 2 = letterbox (默认)) iec958-encoded 设置 iec958 输出模式为 encoded。 iec958-decoded 设置 iec958 输出模式为 decoded (默认)。 macrovision=<参数值> macrovision 模式 (0 = off (默认), 1 = agc, 2 = agc 2 colorstripe, 3 = agc 4 colorstripe) mute 静音输出

unmute
非静音输出

ucode=<参数值>
microcode 的路径

TV output

75ire
启用 7.5 IRE 输出模式

no75ire
停用 7.5 IRE 输出模式 (默认)

bw
黑白电视输出

color
彩色电视输出 (默认)

interlaced
交错电视输出 (默认)

nointerlaced
停用交错电视输出

norm=<参数值>
TV 制式 (ntsc (默认), pal, pal60, palm, paln, palnc)

square-pixel
设置像素模式为 square

ccir601-pixel
设置像素模式为 ccir601

overlay

cr-left=<0-500>
设置左裁减值 (默认是: 50)。

cr-right=<0-500>
设置右裁减值 (默认是: 300)。

cr-top=<0-500>
设置上裁减值 (默认是: 0)。

cr-bottom=<0-500>
设置下裁减值 (默认是: 0)。

ck-[r|g|b]=<0-255>
设置叠加 color-key 的红(r), 绿(g)和蓝(b)增益。

ck-[r|g|b]min=<0-255>
各自 color key 的最小值

ck-[r|g|b]max=<0-255>
各自 color key 最大值

ignore-cache
忽略缓存的叠加设置。

update-cache
更新缓存的叠加设置。

ol-osd
启用叠加屏上显示(OSD)。

nool-osd
停用屏上显示(OSD) (默认)。

ol[h|w|x|y]-cor=<-20-20>
调整叠加尺寸(h,w) 和位置(x,y), 万一跟窗口匹配不够完美 (默认是: 0)。

overlay

激活叠加 (默认)。

nooverlay

激活电视输出。

overlay-ratio=<1-2500>

调整叠加 (默认是: 1000)。

-fbmode <模式名> (仅用于 -vo fbdev)

改变视频模式为 /etc/fb.modes 中标记为 <模式名> 的模式。

注意: VESA 帧缓冲不支持改变模式。

-fbmodeconfig <文件名> (仅用于 -vo fbdev)

重载帧缓冲模式的配置文件 (默认是: /etc/fb.modes)。

-fs (也请参见 -zoom)

全屏幕回放 (电影居中, 四周填充黑色条边)。 还不被所有的视频输出驱动支持。

-fsmode-dontuse <0-31> (废弃, 使用 -fs 选项)

如果你仍然有全屏问题试试此选项。

-fstype <类型1,类型2,...> (仅用于 X11)

指定使用全屏模式的优先级列表。 你可以用前缀 "-" 取消模式。 如果你遇到像全屏窗口被其他窗口遮住问题, 试试不同的顺序。

注意: 参见 -fstype help 列出的全部可用模式的列表。

可用的类型有:

above

如果可用的话, 使用 _NETWM_STATE_ABOVE 提示。

below

如果可用的话, 使用 _NETWM_STATE_BELOW 提示。

fullscreen

如果可用的话, 使用 _NETWM_STATE_FULLSCREEN 提示。

layer

与默认层一起使用 _WIN_LAYER 提示。

layer=<0...15>

与给定的层号一起使用 _WIN_LAYER 提示。

netwm

锁定 NETWM 风格。

none

不设置全屏窗口层。

stays_on_top

如果可用的话, 使用 _NETWM_STATE_STAYS_ON_TOP 提示。

示例:

layer,stays_on_top,above,fullscreen

如果指定了不正确的或不支持的模式, 默认的顺序作为备用而使用。

-fullscreen

修复 OpenBox 1.x 的全屏开关。

-geometry x[%][:y[%]] or [WxH][+x+y]

调整屏幕输出的初始位置。 x 和 y 代表从屏幕左上角到显示图像左上角的距离(以像素为单位)。 不过如果在参数值后有百分号记号, 参数值就变为该方向上的屏幕尺寸的百分数。 它也支持标准 X11 的 -geometry 选项格式。 如果使用 -wid 选项指定了外部窗口, 那么 x 和 y 坐标是相对于窗口而不是屏幕的左上角。

注意: 此选项仅被 x11, xmg, xv, xvmc, xvidix, gl, gl2, directx 和 tdfxfb 视频输出驱动支持。

示例:

50:40

把窗口放在 x=50, y=40 处。

50%:50%

把窗口放在屏幕中央。

100%

把窗口放在屏幕右边中央。

100%:100%

把窗口放在屏幕右下角。

`-guiwid <窗口标识>` (也请参见 `-wid`) (仅用于 GUI)

告诉 GUI 也使用 X11 窗口并把自己钉在视频窗口的下方, 这对于将一个 mini-GUI 嵌入到浏览器时(比如 MPlayer 插件)有用。

`-hue <-100-100>`

调视频信号的色调(默认是: 0)。你可以通过此选项得到负片效果的图像。还不被所有视频输出驱动支持。

`-monitor-dotclock <范围[,范围,...]>` (仅用于 `-vo fbdev` 和 `vesa`)

指定监视器的 dotclock 或 pixelclock 范围。

`-monitor-hfreq <范围[,范围,...]>` (仅用于 `-vo fbdev` 和 `vesa`)

指定监视器的水平频率范围。

`-monitor-vfreq <范围[,范围,...]>` (仅用于 `-vo fbdev` 和 `vesa`)

指定监视器的垂直频率范围。

`-monitoraspect <宽高比>` (也请参见 `-aspect`)

设置监视器或电视屏幕的宽高比。值 0 将禁用先前的设置(例如, 配置文件中的设置)。如果激活了, 此选项将屏蔽 `-monitorpixelaspect` 中的设置。

示例:

`-monitoraspect 4:3` 或者 `1.3333`

`-monitoraspect 16:9` 或者 `1.7777`

`-monitorpixelaspect <宽高比>` (也请参见 `-aspect`)

设置监视器或电视屏幕的单像素宽高比(默认是停用的)。值 1 表示正方形像素((几乎?) 对所有的 LCD 是正确的)。

`-nodouble`

停用双缓冲, 多数用于调试目的。双缓冲通过在内存里储存两帧来解决闪烁问题, 在显示一帧的同时解码另一帧。会负面影响 OSD, 但常常能去除 OSD 闪烁。

`-nogradpointer`

视频模式改变(`-vm`)后, 不截获鼠标焦点。对多输出头设置有用。

`-nokeepaspect`

缩放窗口时不保持窗口的宽高比。只能与 `x11`, `xv`, `xmga`, `xvidix`, `directx` 视频输出驱动一起工作。而且在 X11 下你的窗口管理器必须尊重窗口宽高比的提示。

`-ontop`

使得此播放器窗口位于其他窗口之上。视频输出驱动使用 X11 的, 除了 SDL, 还有使用 `directx`, `macosx`, `quartz`, `ggi` 和 `gl2` 的都支持此功能。

`-panscan <0.0-1.0>`

启用 pan-and-scan 功能(裁减如 16:9 电影的边适应 4:3 显示不带黑边)。范围参数值用来控制裁减多少图像。只能与 `xv`, `xmga`, `mga`, `gl`, `gl2`, `quartz`, `macosx` 和 `xvidix` 的视频输出驱动一起工作。

注意: 也允许-1和0间的值, 但这些值处于高度试验阶段, 从而可能导致崩溃或更糟。使用后果自负!

`-panscanrange <-19.0-99.0>` (试验)

改变 pan-and-scan 功能的范围参数值(默认是: 1)。正值表示倍扩默认范围。负值表示你可以放大到因子 `-panscanrange+1` 所允许的范围。例如 `-panscanrange -3` 允许缩

- 放因子达到 4。此特性属实验性, 不要报告相关错误除非你在用 `using -vo gl`。
- `-refreshrate <Hz>`
设置监视器的刷新率(以 Hz 为单位)。目前仅被 `-vo directx` 与 `-vm` 的组合支持。
- `-rootwin`
在根窗口(桌面背景)中播放电影。虽然桌面背景图像可能盖住电影窗口。只能与 `x11`, `xv`, `xmga`, `xvidix`, `quartz`, `macosx` 和 `directx` 的视频输出驱动一起工作。
- `-saturation <-100-100>`
调整视频信号的饱和度(默认值: 0)。你可以通过此选项获得灰度输出。还不被所有的视频输出驱动支持。
- `-screenh <像素>`
指定垂直分辨率, 如果视频输出驱动不知道屏幕分辨率, 如 `fbdev`, `x11` 和 `TVout`。
- `-screenw <pixels>`
指定水平分辨率, 如果视频输出驱动不知道屏幕分辨率, 如 `fbdev`, `x11` 和 `TVout`。
- `-stop-xscreensaver` (仅用于 `X11`)
启动时关闭 `xscreensaver`, 退出时再打开它。
- `-vm`
尝试改变不同的视频模式。视频输出驱动支持的有 `dga`, `x11`, `xv`, `sdl` 和 `directx`。如果用于 `directx` 视频输出驱动 `-screenw`, `-screenh`, `-bpp` 和 `-refreshrate` 选项可用来设置新的显示模式。
- `-vsync`
启用 VBI 用于 `vesa`, `dfbmga` 和 `svga` 视频输出驱动。
- `-wid <窗口标识>` (也请参见 `-guiwid`) (仅用于 `X11`, `OpenGL` 和 `DirectX`)
告诉 MPlayer 附着于一个已有的窗口。把 MPlayer 嵌入浏览器(比如 `plugger` 扩展)时有用。
- `-xineramascreeen <-2-...>` (仅用于 `X11`)
在 Xinerama 配置中(就是一个单一桌面展开在多个显示器上), 此选项告诉 MPlayer 把电影显示在哪个屏幕上。值 `-2` 表示全屏幕展开到整个虚拟的显示屏幕上(在这种情况下将完全忽略 Xinerama 的信息), `-1` 表示在目前窗口所在的显示屏上全屏幕。通过 `-geometry` 选项设置的初始位置是相对于指定屏幕的。通常只能与 `"-fstype -fullscreen"` 或 `"-fstype none"` 一起工作。
- `-zrbw` (仅用于 `-vo zr`)
黑白显示。用于优化性能, 此选项可以与 `"-lavdopts gray"` 组合使用。
- `-zrcrop <[宽]x[高]+[x 偏移]+[y 偏移]>` (仅用于 `-vo zr`)
选择显示输入图像的一部分, 使用多个这样的选项就启动了 cinerama 模式。在 cinerama 模式下电影分布在多个电视(或投影仪)上, 用于创建一个较大的图像。在第 `n` 个 `-zrcrop` 后面的选项应用于第 `n` 个 MJPEG 解码卡, 每一个编码卡至少需要有一个 `-zrcrop` 选项加上一个 `-zrdev` 选项。请察看 `-zrhel` 的输出和文档 `Zr` 一节中的示例。
- `-zrdev <设备>` (仅用于 `-vo zr`)
指定你的 MJPEG 编码卡使用的设备文件名, 默认情况下 此 `zr` 视频输出驱动使用它所能找到的第一个 `v4l` 设备。
- `-zrfd` (仅用于 `-vo zr`)
强制简化取样: 简化取样由 `-zrhdec` 和 `-zrvdec` 指定, 一般只有在硬件缩放能把图像伸展到原始尺寸时才使用。
- `-zrhdec <1|2|4>` (仅用于 `-vo zr`)
水平简化取样: 要求驱动只发送输入图像的每 2 或 4 行/ 像素到 MJPEG 编码卡, 并使用 MJPEG 卡的缩放器把图像回复到原始尺寸。
- `-zrhel` (仅用于 `-vo zr`)
显示一个列表包括所有 `-zr*` 的选项, 它们的默认值和使用 cinerama 模式的例子
- `-zrnorm <制式>` (仅用于 `-vo zr`)

- 指定电视制式为 PAL 或 NTSC (默认是: 不改变)。
- `-zrquality <1-20>` (仅用于 `-vo zr`)
从数值 1(最好) 到 20(最差) 代表 JPEG 编码质量。
- `-zrvdec <1|2|4>` (仅用于 `-vo zr`)
垂直简化取样: 要求驱动只发送输入图像的每 2 或 4 行/ 像素到 MJPEG 编码卡, 并使用 MJPEG 卡的缩放器把图像回复到原始尺寸。
- `-zrxdooff <x 显示偏移量>` (仅用于 `-vo zr`)
如果电影屏幕比电视的小, 此选项指定 相对于电视屏幕左上角的 x 偏移量 (默认是: 中央位置)。
- `-zrydooff <y 显示偏移量>` (仅用于 `-vo zr`)
如果电影屏幕比电视的小, 此选项指定 相对于电视屏幕左上角的 y 偏移量 (默认是: 中央位置)。

视频输出驱动 (仅适用于MPLAYER)

视频输出驱动是各种视频输出设备的接口。语法是:

`-vo <驱动1[:子选项1[=参数值]:...],驱动2,...[,]>`

指定使用视频输出驱动的优先级列表。

如果列表尾有 ", " MPlayer 使用没有列出的驱动作为后备。子选项是可选的, 大多数省略。

注意: 已编译进的视频输出驱动参见于 `-vo help` 的列表。

示例:

```
-vo ximga,xv,
    尝试使用 Matrox X11 驱动, 然后是 Xv 驱动, 再后是其他的。
-vo directx:noaccel
    使用加速特性关闭的 DirectX 驱动。
```

可用的视频输出驱动有:

xv (仅用于 X11)

使用 XFree86 4.x 的 XVideo 扩展来启用硬件加速回放。如果你不能使用硬件指定的驱动, 这或许是最好的选项。有关用了什么 colorkey 以及如何绘制的信息, 请运行 MPlayer 的 `-v` 选项并注意开头有标签 `[xv common]` 的行。

`port=<数值>`

选择指定的 XVideo 端口。

`ck=<cur|use|set>`

选择用了 colorkey 的源 (默认是: cur)。

cur

默认采用 Xv 中目前设置的 colorkey。

use

使用但不从 MPlayer (用 `-colorkey` 选项来修改)来设置 colorkey。

set

同 use

一样但设置已提供的 colorkey。

`ck-method=<man|bg|auto>`

设置 colorkey 绘制方法 (默认是: man)。

man

手动绘制 colorkey (减少某些情况下的闪烁)。

bg设置 colorkey 作为窗口背景。

auto

让 Xv 绘制 colorkey。

x11 (仅用于 X11)

每当 X11 出场时, 共享内存的视频输出驱动虽没有硬件加速也能工作。

xover (仅用于 X11)

添加 X11 支持所有基于视频输出驱动的重叠。 目前仅被 tdfx_vid 支持。

<vo_驱动>

选择此驱动作为源使用重叠在 X11 上面。

xvmc (仅用于 X11 的 -vc ffmpeg12mc)

视频输出驱动使用 XFree86 4.x 的 XvMC(X 视频运动补偿) 扩展 以加速 MPEG-1/2 和 VCR2 的解码。

port=<数值>

选择指定的 XVideo 端口。

(no)benchmark

停用图像显示。 对于适当的驱动基准测试是必要的, 因为只会改变监视器 retrace 的图像缓冲 (nVidia)。 默认是不停用图像显示 (nobenchmark)。

(no)bobdeint

非常简单的去交错。 可能没有 -vf tfields=1 好, 但是是 xvmc 仅有的去交错 (默认是: nobobdeint)。

(no)queue

显示帧排队, 以允许视频硬件更多的并行工作。 可能增加微小的 (不令人注意的) 常量去 A/V 同步(desync) (默认是: noqueue)。

(no)sleep

当等待完成渲染时, 使用睡眠功能(不推荐 Linux 使用) (默认是: nosleep)。

ck=cur|use|set

同 -vo xv:ck 一样 (参见 -vo xv)。

ck-method=man|bg|auto

同 -vo xv:ck-method 一样 (参见 -vo xv)。

dga (仅用于 X11)

通过 XFree86 Direct Graphics Access 扩展播放视频。 考虑废弃。

sdl (仅用于 SDL)

高级平台独立的 SDL(简单 Directmedia 层) 库视频输出驱动。 既然 SDL 使用自己的 X11 层, MPlayer X11 选项对 SDL 没有影响。

driver=<驱动>

显式选择 SDL 驱动。

(no)forcexv

通过 sdl 视频输出驱动使用 XVideo (默认是: forcexv)。

(no)hwaccel

使用硬件加速的缩放 (默认是: hwaccel)。

vidix

VIDIX (VIdeo Interface for *nIX) 是不同显卡的视频加速特性的接口。 非常快速的卡视频输出驱动支持它。

<子设备>

明确表示选择使用哪个 VIDIX 子设备驱动。 可用的子设备有 cyberblade, mach64, mga_crtc2, mga, nvidia, pm2, pm3, radeon, rage128, sis 和 unichrome。

xvidix (仅用于 X11)

VIDIX 的 X11 前台

<子设备>

同 vidix 一样

cvidix

通用的平台独立的 VIDIX 前台, 甚至能用 nVidia 卡运行在文本控制台。

<子设备>

同 vidix 一样

winvidix (仅用于 Windows)

VIDIX 的 Windows 前台

<子设备>

同 vidix 一样

directx (仅用于 Windows)

使用 DirectX 接口的视频输出驱动。

noaccel

关闭硬件加速。 如果显示有问题, 尝试此选项。

quartz (仅用于 Mac OS X)

Mac OS X Quartz 视频输出驱动。 在某些情况下, 锁定打包 YUV 输出格式, 如: -vf format=yuy2, 可能更有效。

device_id=<数值>

选择全屏显示设备。

fs_res=<宽>:<高>

指定全屏分辨率 (对慢系统有用)。

macosx (Mac OS X 10.4 或 10.3.9 的 QuickTime 7)

Mac OS X CoreVideo 视频输出驱动

device_id=<数值>

选择全屏显示设备。

fbdev (仅用于 Linux)

使用内核帧缓冲播放视频。

<设备>

显式选择 fbdev 设备名 (如 /dev/fb0), 或者如果设备名以 "vidix" 开始 选择 VIDIX 子设备名 (如: sis 驱动的 "vidixsis_vid")。

fbdev2 (仅用于 Linux)

使用内核帧缓冲播放视频, 另一种实现。

<设备>

显式选择 fbdev 设备名 (默认是: /dev/fb0)。

vesa

非常通用的视频输出驱动, 应该能用于任何的 VESA VBE 2.0 兼容卡。

(no)dga

打开或关闭DGA模式 (默认为: 打开) 。

neotv_pal

激活 NeoMagic 电视输出并设置成 PAL 制式。

neotv_ntsc

激活 NeoMagic 电视输出并设置成 NTSC 制式。

vidix

使用 VIDIX 驱动

lvo:

激活 Linux 视频重叠在 VESA 模式上面。

svga

使用 SVGA 库播放视频。

<视频模式>

指定使用的视频模式。 此模式以 <宽>x<高>x<颜色> 格式给出, 如 640x480x16M 或图形模式数字, 如 84。

bbosd

绘制 OSD 到(较慢)影片下方的黑条。

native

只使用原生的绘制功能。 避免直接渲染, OSD 和硬件加速。

retrace

强制帧打开垂直 retrace。 只与 `-double` 合用有用。 同 `-vsync` 选项的效果一样。

sq

尝试选择方像素视频模式。

vidix

使用 VIDIX 的 svga。

gl

OpenGL 视频输出驱动, 简版。 视频尺寸一定小于 OpenGL 实现的最大纹理尺寸。 瞄准能与甚至最基本的 OpenGL 实现一起工作, 但也利用较新的扩展, 允许支持更多的色彩空间和直接渲染。 既然能对高清晰度提供 大 的提速, 请使用 `-dr` 如果你的 OpenGL 实现能一同工作。 此代码执行极少的检测, 所以如果某个特性不工作, 可能是因为你的卡或 OpenGL 实现不支持, 尽管连任何错误消息都没给你。 使用 `glxinfo` 或类似的工具以显示支持的 OpenGL 扩展。

(no)scaled-osd

当窗口大小改变时, 改变 OSD 的行为方式 (默认是: 停用)。 启用时, 行为就像其他的视频输出驱动, 对固定大小的字体更好。 停用时似乎对 FreeType 更有利并使用全屏模式的边框。 无法在使用 ass 字幕的情况下正确地工作 (参见 `-ass`)。 作为替代性方法, 你可以通过 `-vf ass` 在不用 OpenGL 的情况下渲染这种字幕。

osdcolor=<0xRRGGBB>

OSD 颜色 (默认是: 0xffffffff, 对应白色)。

rectangle=<0,1,2>

选择矩形纹理保存视频内存的方法, 但常常会更慢 (默认是: 0)。

0: 使用 power-of-two 纹理 (默认)。

1: 使用 GL_ARB_texture_rectangle 扩展。

2: 使用 GL_ARB_texture_non_power_of_two 扩展。 在某些情况下仅软件支持所以非常慢。

swapinterval=<n>

两缓冲交换的最小间隔, 对已显示的帧计时 (默认是: 1)。 1 等价于启用 VSYNC, 0 等价于停用 VSYNC。 系统对小于 0 的值忽略保持自己的默认值。 此限制帧率到 (水平刷新率 / n)。 需要 GLX_SGI_swap_control 的支持。 某些(大多数/所有?)实现仅工作于全屏模式。

yuv=<n>

选择 YUV 到 RGB 转换的类型。

0: 使用软件转换 (默认)。 兼容于所有的 OpenGL 版本。 提供亮度, 对比度和饱和度控制。

1: 使用寄存器组合。 这里使用 nVidia 专用的扩展 (GL_NV_register_combiners)。 至少需要三个纹理单位。 提供饱和度和色调的控制。 此方法快速但不精确。

2: 使用片断程序。 需要 GL_ARB_fragment_program 扩展和至少三个纹理单位。 提供亮度, 对比度, 饱和度和色调控制。

3: 使用片断程序, 用到 POW 指令。 需要 GL_ARB_fragment_program 扩展和至少三个纹理单位。 提供亮度, 对比度, 饱和度, 色调和 gamma 控制。 对红, 绿和蓝色, Gamma 也能被独立设置。 方法 4 通常更快。

4: 使用片断程序及附加的查询。 需要 GL_ARB_fragment_program 扩展

和至少四个纹理单位。提供亮度, 对比度, 饱和度, 色调和 gamma 控制。对红, 绿和蓝色, Gamma 也能被独立设置。

5: 使用 ATI-specific method (供较旧的卡)。这里使用 ATI 专用的扩展 (GL_ATI_fragment_shader - 不是 GL_ARB_fragment_shader!)。至少需要三个纹理单位。提供饱和度和色调的控制。此方法快速但不精确。

6: 使用 3D 纹理通过查询做转换。需要 GL_ARB_fragment_program 扩展和至少四个纹理单位。既然使用一个纹理和边框像素, 在某些(所有?) ATI 卡上 (软件模拟) 极其慢。提供亮度, 对比度, 饱和度, 色调和 gamma 控制。对红, 绿和蓝色, Gamma 也能被独立设置。速度比其他方法更多地依赖于 GPU 内存带宽。

lscale=<n>

选择缩放功能供调节发光性。仅对 yuv 模式 2, 3, 4 和 6 有效。

0: 使用简单线性过滤 (默认)。

1: 使用双立体过滤 (较好的质量)。需要一个额外的纹理单位。至少在全屏模式, 较旧的卡不能处理色度。

2: 在水平方向使用立方体过滤, 在垂直方向使用线性过滤。比起方式 1 适用于稍多的显卡。

cscale=<n>

选择缩放功能供调节色度。详情参见 lscale。

customprog=<文件名>

从 <文件名> 装载定制的片断程序。示例参见 TOOLS/edgedect.fp。

customtex=<文件名>

从 <文件名> 装载定制的 "gamma ramp" 纹理。能用于与 yuv=4 或 customprog 选项的组合。

(no)customtlin

如果启用(默认), 使用 GL_LINEAR 插值, 否则使用 GL_NEAREST 用于 customtex 纹理。

(no)customtrect

如果启用, 使用 texture_rectangle 用于 customtex 纹理。默认是停用。

正常情况下没有理由使用以下选项, 它们大多是为了测试而存在的。

(no)glfinish

在交换缓存前调用 glFinish()。较慢但在某些情况下输出更精确 (默认: 禁用)。

(no)manyfmts

允许支持更多的色彩格式 (RGB和BGR) (默认: 允许)。要求 OpenGL 版本 >= 1.2。

slice-height=<0-...>

所复制的单片纹理中包含多少行内容 (默认值: 0)。0 代表整幅图像。

注意: 如果使用了 YUV 色彩空间 (参见 yuv 子选项), 将使用特别的规则:

如果解码器采用分片渲染方式 (参见 -noslices), 该设置没有任何效果, 所采用的分片大小由解码器提供。

如果解码器不采用分片渲染方式, 则默认值为 16。

(no)osd

允许或禁止通过 OpenGL 支持 OSD 渲染 (默认: 允许)。此选项用于测试; 而禁用 OSD 应使用 -osdlevel 0。

(no)aspect

允许或禁止支持保持比例缩放和全景缩放方式 (默认: 允许)。禁用的话可能提高速度。

OpenGL 视频输出驱动, 第二代。 支持 OSD 和大于最大纹理尺寸的视频。

(no)glfinish

同 gl 一样 (默认是: 启用)

yuv=<n>

选择 YUV 到 RGB 转换的类型。 如果设置不同于 0 的其它值 OSD 被停用, 而亮度, 对比度和 gamma 的设置只能是通过全局的 X 服务器的设置才有效。 其它的值的意思同 -vo gl 一样。

null

不产生视频输出。 对基准测试有用。

aa

ASCII art 视频输出驱动工作于文字控制台。 执行 *mplayer -vo aa:help*, 你能得到可用子选项的列表和解释。

caca

彩色 ASCII art 视频输出驱动工作于文字控制台。

bl

使用 Blinkenlights UDP 协议回放视频。 此驱动高度特定于硬件。

<子设备>

显式选择 Blinkenlights 子设备。 这多少有点像 arcade:host=localhost:2323 或 hdl:file=文件名1,file=文件名2。 你必须指定一个子设备。

ggi

GGI 图形系统视频输出驱动

<驱动>

显式选择 GGI 驱动。 替换任何的 "," 为 "."。

directfb

使用 DirectFB 库播放视频。

(no)input

使用 DirectFB 替代 MPlayer 键盘码 (默认是: 启用)。

buffermode=single|double|triple

如果想避免撕裂问题, 双倍和三倍缓冲是最好的结果。 三倍缓冲比双倍缓冲更有效, 因为它等待垂直 retrace 时不限制 MPlayer。 单一缓冲应避免 (默认是: 单一)。

fieldparity=top|bottom

为交错帧控制输出次序 (默认是: 停用)。 有效值 top = 先顶域, bottom = 先底域。 此选项对逐行(像大多数 MPEG)影片不影响。 如果观看交错影片时遇到撕裂问题或动作不顺畅, 需启用此选项。

layer=N

锁定回放的层标识 N (默认是: -1 - 自动)。

dfbopts=<list>

指定 DirectFB 参数列表。

dfbmga

Matrox G400/G450/G550 专用的视频输出驱动 使用 DirectFB 库以利用特别的硬件特性。 启用 CRTC2 (第二头), 独立于第一头显示视频。

(no)input

同 directfb 一样 (默认是: 停用)

buffermode=single|double|triple

同 directfb 一样(默认是: triple)

fieldparity=top|bottom

同 directfb 一样

- (no)bes
启用 Matrox BES(后台缩放) (默认是: 停用)。 因为图像插值过程在硬件里完成, 结果是非常好的速度和输出质量。但仅工作于主头。
- (no)spic
利用 Matrox 子图像层显示 OSD (默认是: 启用)。
- (no)crtc2
用第二头打开电视输出 (默认是: 启用)。 输出质量出奇的好, 因为这是交错图像对每个奇/ 偶域有恰当的同步。
- tvnorm=pal|ntsc|auto
设置 Matrox 卡的电视制式不需要改变 /etc/directfbrc (默认是: 停用)。 有效的制式是 pal = PAL, ntsc = NTSC。 特别的制式是 auto (auto-仅仅使用 PAL/NTSC), 因为是通过察看影片的帧率决定使用什么制式。
- mga (仅用于 Linux)
Matrox 专用的视频输出驱动通过内核模式利用 Gxxx 卡的 YUV 后台缩放器。 如果有 Matrox 卡, 这是最快的选项。
<设备名>
显式选择 Matrox 设备名 (默认是: /dev/mga_vid)。
- xmga (仅用于 Linux, X11)
mga 视频输出驱动, 运行在 X11 窗口。
<设备名>
显式选择 Matrox 设备名 (默认是: /dev/mga_vid)。
- s3fb (仅适用于Linux) (另参见 -vf yuv2 和 -dr)
S3 Virge专用的视频输出驱动。 此驱动支持显卡的YUV转换与缩放、双重缓冲和直接渲染的功能。 使用-vf yuy2以采用硬件加速下的YUV2渲染, 这种方式在此款显卡下比YV12快很多。
<device>
明确表示选择使用哪个fbdev设备名 (默认值: /dev/ fb0) 。
- 3dfx (仅用于 Linux)
3dfx 专用的视频输出驱动, 此驱动在 X11 上直接使用硬件。 只支持 16 bpp。
- tdfxfb (仅用于 Linux)
此驱动使用 tdfxfb 帧缓冲驱动以在3dfx卡上的使用YUV加速播放影片。
<设备名>
显式选择 fbdev 设备名 (默认是: /dev/fb0)。
- tdfx_vid (仅用于 Linux)
3dfx 专用的视频输出驱动, 此驱动与tdfx_vid内核模块协同工作。
<设备名>
显式选择设备名 (默认是: /dev/tdfx_vid)。
- dxr2 (也请参见 -dxr2) (仅用于 DXR2)
Creative DXR2 专用的视频输出驱动。
<vo_driver>
输出视频子驱动用作重叠 (x11, xv)。
- dxr3 (仅用于 DXR3)
Sigma Designs em8300 MPEG 解码器芯片 (Creative DXR3, Sigma Designs Hollywood Plus) 专用的视频输出驱动。 也请参见 lavc 视频过滤器。
overlay
激活重叠代替电视输出。
prebuf
打开预缓冲。
sync

将打开新的 sync-engine。

norm=<制式>

指定电视制式。

0: 不改变目前制式 (默认)。

1: 使用 PAL/NTSC 自动调节。

2: 使用 PAL/PAL-60 自动调节。

3: PAL

4: PAL-60

5: NTSC

<0-3>

如果有多于一个 em8300 卡, 指定设备号。

ivtv (IVTV 仅适用于)

Conexant CX23415 (iCompression iTVC15) 或 Conexant CX23416 (iCompression iTVC16)

MPEG 解码器芯片 (Hauppauge WinTV PVR-150/250/350/500) 专用的视频输出驱动供电视输出。 也请参见 lavc 视频过滤器。

device

显式选择 MPEG 解码器设备名 (默认是: /dev/video16)。

output

显式选择电视输出用于视频信号。

v4l2 (需要Linux 2.6.22+的内核)

带有内建硬件MPEG解码器的V4L2标准扩展卡的视频输出驱动。 另参见lavc视频滤镜。

device

明确选定所使用的MPEG解码设备的名称 (默认值: /dev/video16) 。

output

明确选定用于视频信号的电视输出输出口。

mpegpes (仅用于 DVB)

视频输出驱动用于 DVB 卡写输出到一个 MPEG-PES 文件, 如果没有安装 DVB 卡。

card=<1-4>

如果有多于一个 DVB 输出卡, 指定设备号 (仅适用于 V3 API, 比如 1.x.y 驱动系列)。

<文件名>

输出文件名 (默认是: ./grab.mpg)

zr (也请参见 -zr* 和 -zrhelph)

一些 MJPEG 捕捉/ 回放卡的视频输出驱动。

zr2 (也请参见 zrmjpeg 视频 过滤器)

一些 MJPEG 捕捉/ 回放卡的视频输出驱动, 第二代。

dev=<设备>

指定视频设备。

norm=<PAL|NTSC|SECAM|auto>

指定视频制式 (默认是: auto)。

(no)prebuf

激活或撤销预缓冲, 还没被支持。

md5sum

计算每帧的 MD5 和并写入文件。 支持 RGB24 和 YV12 色彩空间。 对调试有用。

outfile=<参数值>

指定输出文件名 (默认是: ./md5sums)。

yuv4mpeg

变换视频流为无压缩 YUV 4:2:0 图像序列并储存到文件 (默认是: ./stream.yuv)。 格式与 mjpegtools 所用的一样, 所以如果想要用 mjpegtools 套装软件处理视频, 这就有用。

它支持 YV12, RGB (24 bpp) 和 BGR (24 bpp) 格式。能与 `-fixed-vo` 选项组合来连接具有相同维数和 fps 值的文件。

`interlaced`

作为交错帧写输出, 先顶域。

`interlaced_bf`

作为交错帧写输出, 先底域。

`file=<文件名>`

写输出到 <文件名> 代替默认的 `stream.yuv`。

注意: 如果你不指定任何选项, 输出的是逐行的(即没有交错的)。

gif89a

输出每帧到当前目录的单个动画 GIF 文件。仅支持 24 bpp 的 RGB 格式并转换输出为 256 色。

`<fps>`

指定浮点数帧率 (默认是: 5.0)。

`<output>`

指定输出文件名 (默认是: `./out.gif`)。

注意: 你必须在指定文件名之前指定帧率, 否则帧率成为文件名的一部分。

示例:

```
mplayer video.nut -vo gif89a:fps=15.0:output=test.gif
```

jpeg

输出每帧到当前目录的 JPEG 文件。每帧以前面 0 填充引导的帧号作为文件名。

`[no]progressive`

指定标准的或逐行的 JPEG (默认是: `noprogressive`)。

`[no]baseline`

指定是否使用基线 (默认是: `baseline`)。

`optimize=<0-100>`

优化因子 (默认是: 100)

`smooth=<0-100>`

平滑因子 (默认是: 0)

`quality=<0-100>`

质量因子 (默认是: 75)

`outdir=<dirname>`

指定保存 JPEG 文件的目录 (默认是: `./`)。

`subdirs=<前缀>`

创建编号的子目录保存文件, 用指定的前缀而不是用当前的目录。

`maxfiles=<参数值>`

用于在每个子目录保存文件的最大编号。必须大于或等于 1 (默认是: 1000)。

png

输出每帧到当前目录的 PNM 文件。每帧以前面 0 填充引导的帧号作为文件名。它在 `raw` 和 `ASCII` 模式下支持 PPM, PGM 和 PGMYUV 文件。也请参见 `png(5)`, `ppm(5)` 和 `pgm(5)`。

`ppm`

写 PPM 文件 (默认)。

`pgm`

写 PGM 文件。

`pgmyuv`

写 PGMYUV 文件。PGMYUV 像 PGM, 但也包含 U 和 V 平面, 添加在图像的底部。

raw
在 raw 模式下写 PNM 文件 (默认)。

ascii
在 ASCII 模式下写 PNM 文件。

outdir=<目录名>
指定目录保存 PNM 文件 (默认是: ./)。

subdirs=<前缀>
创建编号的子目录保存文件, 用指定的前缀而不是用当前的目录。

maxfiles=<参数值>
用于在每个子目录保存文件的最大编号。 必须大于或等于 1 (默认是: 1000)。

png
输出每帧到当前目录的 PNG 文件。 每帧以前面 0 填充引导的帧号作为文件名。 支持 24 bpp 的 RGB 和 BGR 格式。
z=<0-9>
指定压缩级别。 0 没压缩, 9 最强压缩。

tga
输出每帧到当前目录的 Targa 文件。 每帧以前面 0 填充引导的帧号作为文件名。 此视频输出驱动的目的是拥有一个简单无损失的图像绘制器且不需要任何外部库。 支持 15, 24 和 32 bpp 的 BGR[A] 色彩格式。 你能锁定一个特殊的格式带视频格式过滤器。

示例:
mplayer video.nut -vf format=bgr15 -vo tga

解码/过滤器选项

-ac <[-|+]编解码器 1,[-|+]编解码器 2,...[,]>
按照在 codecs.conf 中音频编解码器的名称, 指定使用它们的优先级列表。 编解码器名称前使用 "-" 表示忽略它。 编解码器名称前使用 "+" 表示锁定它, 这有可能会崩溃! 如果列表尾有 ",", MPlayer 使用没有列出的编解码器作为后备。
注意: 可用的编解码器其完整列表请参见 -ac help。

示例:
-ac mp3acm
锁定 I3codeca.acm MP3 编解码器。
-ac mad,
先尝试 libmad, 其它作为后备。
-ac hwac3,a52,
先尝试硬件 AC3 通透, 然后是软件 AC3, 最后是其它。
-ac hwdts,
先尝试硬件 DTS 通透, 然后是其它。
-ac ffmpeg3,
跳过 FFmpeg 的 MP3 编解码器。

-af-adv <force=(0-7):list=(filters)> (也请参见 -af)
指定高级音频过滤器选项:
force=<0-7>
锁定音频过滤器的插入为下面其中之一:
0: 使用完全自动的过滤器插入。
1: 精度优化 (默认)。

- 2: 速度优化。 警告: 某些音频过滤器特性可能默默地消失, 且音质可能下降。
- 3: 使用无自动的过滤器插入且无优化。 警告: 使用此设定可能使 MPlayer 崩溃。
- 4: 根据 0 以上的, 使用自动的过滤器插入, 但尽可能使用浮点处理。
- 5: 根据 1 以上的, 使用自动的过滤器插入, 但尽可能使用浮点处理。
- 6: 根据 2 以上的, 使用自动的过滤器插入, 但尽可能使用浮点处理。
- 7: 根据 3 以上的, 使用自动的过滤器插入, 但尽可能使用浮点处理。

list=<过滤器列表>

同 -af 一样。

-afm <驱动1,驱动2,...>

按照在 codecs.conf 中音频编解码器族的名称, 指定使用它们的优先级列表。 如果给定的编解码器族没有一个能工作, 使用默认的后备编解码器族。 如果列表尾有 ", " MPlayer 使用没有列出的编解码器作为后备。

注意: 可用的编解码器族其完整列表请参见 -afm help。

示例:

-afm ffmpeg

先尝试 FFmpeg 的 libavcodec 编解码器。

-afm acm,dshow

先尝试 Win32 编解码器。

-aspect <高宽比> (也请参见 -zoom)

重载影片的高宽比, 万一播放中的文件的高宽比信息不正确或缺少。

示例:

-aspect 4:3 或 -aspect 1.3333

-aspect 16:9 或 -aspect 1.7777

-noaspect

停用影片高宽比的自动补偿。

-field-dominance <-1-1>

设置隔行扫描内容的第一个域。 对于那些使帧速加倍的去隔行扫描器来说那很有用: -vf tfields=1, -vf yadif=1 和 -vo xvmc:bobdeint。

-1

自动 (默认值): 如果解码器不导出适当的信息的话, 其将设回为0 (上半域)。

0

上半域为第一个域

1

下半域为第一个域

-flip

上下翻转图像。

-lavdopts <选项1:选项2:...> (调试代码)

指定 libavcodec 解码参数。 用逗号分隔多重选项。

示例:

-lavdopts gray:skiploopfilter=all:skipframe=nonref

可用的选项有:

bitexact

对所有的解码步骤只用 bit-exact 算法 (用于编解码测试)。

bug=<参数值>

手工绕过编码器错误:

- 0: 无
- 1: 自动检测错误 (默认)
- 2 (msmpeg4v3): 老式 lavc 生成的 msmpeg4v3 文件 (不自动检测)
- 4 (mpeg4): Xvid 交错错误 (如果 fourcc==XVIX, 会自动检测)
- 8 (mpeg4): UMP4 (如果 fourcc==UMP4, 会自动检测)
- 16 (mpeg4): 填补错误 (自动检测)
- 32 (mpeg4): 非法 vlc 错误 (每个 fourcc 都自动检测)
- 64 (mpeg4): Xvid 或 DivX qpel 错误 (每个 fourcc/ 版本都自动检测)
- 128 (mpeg4): 旧标准的 qpel (每个 fourcc/ 版本都自动检测)
- 256 (mpeg4): 另一个 qpel 错误 (每个 fourcc/ 版本都自动检测)
- 512 (mpeg4): direct-qpel-blocksize 错误 (每个 fourcc/ 版本都自动检测)
- 1024 (mpeg4): 边沿填补错误 (每个 fourcc/ 版本都自动检测)

debug=<参数值>

显示调试信息。

- 0: 停用
- 1: 图像信息
- 2: 率控制
- 4: 比特流
- 8: 宏块(MB)类型
- 16: 每块的量化参数(QP)
- 32: 运动矢量
- 0x0040: 运动矢量可视化 (使用 -noslices)
- 0x0080: 宏块(MB)跳略
- 0x0100: 启动码
- 0x0200: PTS
- 0x0400: 错误恢复力(resilience)
- 0x0800: 内存管理控制操作(H.264)
- 0x1000: 错误
- 0x2000: 可视化量化参数(QP), QP 值越低着色越绿。
- 0x4000: 可视化块型。

ec=<参数值>

设置错误隐藏策略。

- 1: 对受损的宏块使用强去块(deblock)过滤器。
- 2: 往返式运动矢量(MV)搜寻 (慢)
- 3: 全部 (默认)

er=<参数值>

设置错误恢复力策略。

- 0: 停用
- 1: 谨慎的 (应该能用于破编码器。)
- 2: 正常的 (默认) (用于正常的编码器。)
- 3: 挑衅的 (更多检查, 但是连有效的比特流也可能导致问题。)
- 4: 非常挑衅的

fast (仅用于MPEG-2、MPEG-4和H.264)

启用与标准不符的优化, 并且可能导致一些潜在的问题, 诸如简化反量化步聚, 想当然地使用默认的量化矩阵, 想当然地采用YUV 4:2:0, 以及跳过一些检测受损比特流的检查步骤。

gray

只灰度图像解码 (比彩色解码快一点)

idct=<0-99> (参见 -lavcopts)

为了最佳的解码品质, 使用与编码和解码时相同的 IDCT 算法。不过这可能会牺牲一些精确性。

lowres=<数值>[,<宽度>]

以较低的分辨率解码。低分辨率解码还不被所有的边界码器支持, 且常常导致难看的图像。这不是有错误, 只是一个副作用, 全分辨率时不解码。

0: 停用

1: 1/2 分辨率

2: 1/4 分辨率

3: 1/8 分辨率

如果 <宽度> 指定了, 就使用了低分辨率解码, 只要视频的宽度宽过(is major than)或等于 <宽度>。

sb=<数值> (仅用于 MPEG-2)

跳过宏块底部的制定行数。

st=<数值> (仅用于 MPEG-2)

跳过宏块顶部的制定行数。

skiploopfilter=<跳略值>

在 H.264 解码期间跳过循环过滤器(亦称为 "去块")。既然被过滤的帧被当作解码依赖帧的引用, 在品质上 这比不进行去块步骤, 如不对 MPEG-2 视频去块, 效果更差。但是至少对高比特流的 HDTV 提供了大的提速且不损失视觉品质。

<跳略值> 可以是以下的值之一:

none: 从不跳过。

default: 跳过无用的处理步骤 (如: AVI 中的 0 大小的包)。

nonref: 跳过没被引用的帧 (即, 不用于解码其它的帧, 错误不能 "兴旺")。

bidir: 跳过 B-帧。

nonkey: 跳过所有的帧除了 keyframes。

all: 跳过所有的帧。

skipidct=<跳略值> (仅用于 MPEG1/2)

跳过 IDCT 步骤。在几乎所有的情况下此步骤使得品质降级许多 (可用的跳略值请参见 skiploopfilter)。

skipframe=<跳略值>

完全跳过解码帧。大的提速, 但是有痉挛的动作或有时糟糕的图像 (可用的跳略值请参见 skiploopfilter)。

threads=<1-8> (仅用于MPEG-1/2)

用于解码的线程数 (默认是: 1)

vismv=<数值>

可视化运动矢量。

0: 停用

1: 可视化 P-帧的前瞻运动矢量。

2: 可视化 B-帧的前瞻运动矢量。

4: 可视化 B-帧的后瞻运动矢量。

vstats

打印一些统计量并保存到 ./vstats_*.log。

-noslices

停用 16-像素高的片/条方式绘制视频, 而是一次绘制整个帧。可能更快或更慢, 取决于显卡和可用的缓存。它只对 libmpeg2 和 libavcodec 编解码器有效。

-nosound

不播放/ 编码声音。 对基准测试有用。

-novideo

不播放/ 编码视频。 但在许多情况下不工作, 请使用 **-vc null -vo null** 替代。

-pp <品质> (也请参见 **-vf pp**)

设置 DLL 的后期处理级别。 仅可以用于有内部后期处理例程的 Win32 DirectShow DLL。 此选项不再对 **-vf pp** 有用。 **-pp** 参数值的有效范围依编解码器不同而不同, 大部分为 0-6, 其中 0=禁用 6=最慢/ 最好。

-pphelp (也请参见 **-vf pp**)

显示可用的后期处理滤镜及其用法的简介。

-ssf <模式>

指定软件缩放参数。

示例:

-vf scale -ssf lgb=3.0

lgb=<0-100>

高斯柔化过滤器 (亮度)

cgb=<0-100>

高斯柔化过滤器 (色度)

ls=<-100-100>

锐化过滤器 (亮度)

cs=<-100-100>

锐化过滤器 (色度)

chs=<h>

水平色度偏移

cvs=<v>

垂直色度偏移

-stereo <模式>

选择 MP2/MP3 立体声输出的类型。

立体声

0

左声道

1

右声道

2

-sws <软件缩放类型> (也请参见 **-vf scale** 和 **-zoom**)

指定软件缩放算法用于 **-zoom** 选项。 这对缺少硬件加速, 如 x11 的视频输出驱动不影响。

可用选项有:

快速二次线性

0

快速二次线性

1

二次立方 (品质好) (默认)

2

实验中

3

最短距离 (品质差)

4

5 area

6	亮度二次立方/色度二次线性
7	高斯
8	sincR
9	lanczos

10

自然二次立方样条曲线

注意: 有些 `-sws` 选项可调。视频缩放过滤器的描述有更多的信息。

`-vc <[-|+]编解码器 1,[-|+]编解码器 2,...[,]>`

按照在 `codecs.conf` 中视频编解码器的名称, 指定使用它们的优先级列表。编解码器名称前使用 "-" 表示忽略它。编解码器名称前使用 "+" 表示锁定它, 这有可能会崩溃! 如果列表尾有 "," MPlayer 使用没有列出的编解码器作为后备。

注意: 可用的编解码器其完整列表请参见 `-vc help`。

示例:

`-vc divx`

锁定 Win32/VfW DivX 编解码器, 无后备。

`-vc -divxds,-divx,`

跳过 Win32 DivX 编解码器。

`-vc ffmpeg12,mpeg12,`

先尝试 libavcodec 的 MPEG-1/2 编解码器, 其次是 libmpeg2, 然后是其它的。

`-vfm <驱动1,驱动1,...>`

按照在 `codecs.conf` 中视频编解码器族的名称, 指定使用它们的优先级列表。如果给定的编解码器族没有一个能工作, 使用默认的后备编解码器族。如果列表尾有 "," MPlayer 使用没有列出的编解码器作为后备。

注意: 可用的编解码器族其完整列表请参见 `-vfm help`。

示例:

`-vfm ffmpeg,dshow,vfw`

先尝试 libavcodec, 其次是 Directshow, 然后 VfW 编解码器, 最后如果它们都不工作使用其它后备的。

`-vfm xanim`

先尝试 XAnim 编解码器。

`-x <x>` (也请参见 `-zoom`) (仅用于 MPlayer)

缩放图像到宽度 `<x>` (如果软件/硬件缩放可用)。停用高宽比计算。

`-xvidopts <选项1:选项2:...>`

指定使用Xvid解码时的附加参数。

注意: 既然 libavcodec 快过 Xvid 你可能想使用 libavcodec 后期处理过滤器 (`-vf pp`) 和解码器 (`-vfm ffmpeg`) 来替换。

Xvid 的内部后期处理过滤器有:

`deblock-chroma`

色度去块(deblock)过滤器

`deblock-luma`

亮度去块(deblock)过滤器

`dering-luma`

亮度去环(deringing)过滤器

`dering-chroma`

色度去环(deringing)过滤器

`filmeffect`
在视频中添加人工的胶片纹理。可能提高了视觉品质, 虽然降低了真实的品质。

渲染方法:

`dr2`
激活直接渲染的方法 2。

`nodr2`
撤销直接渲染的方法 2。

`-xy <参数值>` (也请参见 `-zoom`)
参数值<=8
按因子 <参数值> 缩放图像。
参数值>8
设置图像宽度为参数值, 并计算图像高度以保持高宽比。

`-y <y>` (也请参见 `-zoom`) (仅用于 MPlayer)
缩放图像到高度 <y> (如果软件/ 硬件缩放可用)。 停用高宽比计算。

`-zoom`
在可能的情况下允许使用软件缩放。 不支持硬件缩放的输出驱动(像 x11, fbdev), 因为性能缘故 而被 MPlayer 默认停用缩放, 但现在允许软件缩放。

音频过滤器

音频过滤器允许你修改音频流及其属性。 语法是:

`-af <过滤器 1[=参数 1:参数 2:...], 过滤器 2,...>`
建立音频过滤器链。

注意: 请参见 `-af help`, 获得可用的音频过滤器其完整列表。

可用的过滤器有:

`resample[=srate[:sloppy[:type]]]`

改变音频流的采样(速/频)率。 可用于声卡只有一个固定的频率或者要卡住的旧卡最高只能达 44.1kHz。 此过滤器在必要时被自动启用。 它只支持 16-位 native-endian(低位在前) 格式的整数值和浮点数作为输入。

注意: 对于 MEncoder, 你也需要使用 `-srate <srate>`。

`<srate>`

输出的采样率(以 Hz 为单位)。 参数的有效范围是从 8000 到 192000。 如果输入和输出的采样率相同, 或者如果此参数被省略, 此过滤器被自动卸载。 正常情况下高的采样率能改善音频品质, 尤其是与其它过滤器结合时。

`<sloppy>`

允许(1) 或不允许(0) 输出采样率与给定的频率 `<srate>` 稍微不同 (默认是: 1)。 可用于启动回放时极端慢的情形。

`<type>`

选择使用何种重采样方式。

0: 线性插值 (快速, 品质低 尤其是 upsampling 时)

1: 多相过滤器组(filterbank)并且整数处理

2: 多相过滤器组(filterbank)并且浮点数处理 (慢, 品质最佳)

示例:

`mplayer -af resample=44100:0:0`

设置重采样过滤器的输出采样率为 44100Hz, 使用此不允许微调的输出采样率 缩放和线性插值。

lavcresample[=srate[:length[:linear[:count[:cutoff]]]]]

改变音频流采样率为整数 <srate> (以 Hz 为单位)。它只支持 16位 native-endian(低位在前) 格式。

注意: 对于 MEncoder, 你也需要使用 `-srate <srate>`。

<srate>

输出的采样率

<length>

有关较低采样率的过滤器长度 (默认是: 16)

<linear>

如果值是 1, 过滤器将在多相进入处之间线性插值

<count>

多相进入处数量的 log2 (... , 10->1024, 11->2048, 12->4096, ...) (默认是: 10->1024)

<cutoff>

截频 (0.0-1.0), 默认设置取决于过滤器长度

sweep[=speed]

产生正弦 sweep。

<0.0-1.0>

delta 的正弦, 使用非常低的值听此扫听(sweep)。

sinesuppress[=freq:decay]

删除一个指定频率的正弦。对去除低品质音频设备上 50/60Hz 的噪音时有用。可能只在单声道输入上可工作。

<freq>

应删除的频率 (以 Hz 为单位) (默认是: 50)

<decay>

控制适应性 (一个较大的值会使得过滤器适应幅度和相位的变化较迅速, 而一个较小的值会使得此适应过程较慢) (默认是: 0.0001)。合理的值约为 0.001。

hrtf[=flag]

头部相关的转变功能: 为头戴式耳机转换多声道音频成双声道输出, 但保留声音的空间感。

Flag 含义

m 后场声道矩阵解码

s 双声道矩阵解码

0 无矩阵解码 (默认)

equalizer=[g1:g2:g3:...:g10]

10-八度波段图形均衡器, 其实现使用了 10 IIR 带通过滤器。这意味着不管回放的是何种音频都行。10 波段的中心频率各是:

段号 频率

0 31.25 Hz

1 62.50 Hz

2 125.00 Hz

3 250.00 Hz

4 500.00 Hz

5 1.00 kHz

6 2.00 kHz

7 4.00 kHz

8 8.00 kHz

9 16.00 kHz

如果此声音播放的采样率低于频率波段的中心频率, 此波段就被停用。此过滤器的一个已知错误是, 最高波段的特性没有完全对称, 如果采样率接近其波段的中心频率时。此问题可被绕过。办法是 upsampling 其声音: 在此过滤器之前使用重采样过滤器。

<g1>:<g2>:<g3>:...:<g10>

浮点数代表各频率波段的增益分贝(dB) (-12-12)

示例:

```
mplayer -af equalizer=11:11:10:5:0:-12:0:5:12:12 media.avi
```

放大声音的高端和底端的频率区段, 但几乎完全取消 1kHz 附近的声音。

channels=nch[:nr:from1:to1:from2:to2:from3:to3:...]

能用于添加, 去除, 路由和拷贝音频声道。如果仅给定 <nch>, 默认是用路由(跳接), 其所做工作如下: 如果输出声道数大于输入声道数, 插入空的声道 (除了混音单声道成立体声时, 在两个输出声道中重复此单声道)。如果输出声道数小于输入声道数, 截掉多余的声道。

<nch>

输出声道数 (1-6)

<nr>

路由数 (1-6)

<from1:to1:from2:to2:from3:to3:...>

0 到 5 对定义各声道的路由。

示例:

```
mplayer -af channels=4:4:0:1:1:0:2:2:3:3 media.avi
```

改成 4 声道并设置 4 对路由: 交换声道 0 和声道 1, 保留声道 2 和 3 未动。会观察到如果回放媒体含有两个声道, 声道 2 和 3 无声, 但声道 0 和 1 仍要交换。

```
mplayer -af channels=6:4:0:0:0:1:0:2:0:3 media.avi
```

改成 6 声道并设置 4 对路由: 拷贝声道 0 到声道 0 至 3。声道 4 和 5 无声。

```
mplayer -af channels=6:6:0:4:1:0:2:1:3:2:4:3:5:5 media.avi
```

应该能使 6 声道 ffdca (DTS) 输出正确地与 ALSA 协同工作。

format[=format] (也请参见 -format)

在不同的采样格式之间转换。自动被启用, 当声卡或其他的过滤器需要时。

<format>

设置所需的格式。一般的形式是 "sbe", 其中 "s" 指示符号 (要么 "s" 指带符号, 要么 "u" 指不带符号), "b" 指示每个样本的位数 (16, 24 或 32) 而 "e" 指示 endian 格式 ("le" 指 little-endian, "be" 指 big-endian 或者 "ne" 指 MPlayer 运行其上的计算机所用的 endian 格式)。有效值(尤其)有: "s16le", "u32be" 和 "u24ne"。例外但也有有效的指定格式有: u8, s8, floatle, floatbe, floatne, mulaw, alaw, mpeg2, ac3 和 imaadpcm。

volume[=v[:sc]]

实现软件的音量控制。请谨慎使用此过滤器, 因为会降低信号至声音的噪声率。在大多数情况下设置 PCM 音量至最大是最佳的, 省去过滤器, 而用混音器的主音量控制来控制输出到音响的水平。万一你的声卡有数码的 PCM 混音器而不是模拟的, 且听到失真, 请使用 MASTER(主) 混音器。如果计算机联有外接扩音器(这几乎总是常有的事), 调节主音量和扩音器的音量旋钮能减少噪声水平, 直到背景的嘶嘶声消失。

此过滤器有第二个功能: 测量总体的最高声音水平并通过 MPlayer 显示出来。此估计的音量能用来设置 MEncoder 的音量以使用最大动态范围。

注意: 此过滤器不能重返进入, 所以对每个音频流只能启用一次。

<v>

设置所有声道中流所需的增益分贝(dB): 范围从 -200dB 至 +60dB, 其中

-200dB 完全静音而 +60dB 等于增益 1000 (默认是: 0)。

<sc>

打开 (1) 或关闭 (0) 软剪切。软-剪切 能使得声音更平滑流畅, 如果使用了非常高的音量水平。启用此选项, 如果扩音器的动态范围非常低。

警告: 此功能产生失真, 是应该最后才考虑使用的手段。

示例:

```
mplayer -af volume=10.1:0 media.avi
```

放大声音 10.1dB, 且声音水平太高时使用 硬-剪切。

pan=n[:L00:L01:L02:...L10:L11:L12:...Ln0:Ln1:Ln2:...]

武断混音。基本上是音量和声道过滤器的组合能用于混合许多声道至仅仅几个声道, 如: 立体声到单声道或者改变环绕声系统中心扬声器的 "宽度"。此过滤器很难用, 且需要一些修补才能获得所要的结果。此过滤器的选项数取决于输出声道数。如何用此过滤器把 6-声道文件混音成双声道的实例, 能在示例一节中将近最后的地方找到。

<n>

输出声道数 (1-6)

<Lij>

输入声道 i 混音成输出声道 j 的份量 (0-1)。所以基本上你先要有 n 个数表示要作用到第一个输入声道, 然后再有 n 个数要作用到第二个输入声道, 如此以往。如果你不对某些输入声道指定任何数, 就认为是 0。

示例:

```
mplayer -af pan=1:0.5:0.5 media.avi
```

立体声混音成单声道。

```
mplayer -af pan=3:1:0:0.5:0:1:0.5 media.avi
```

给定 3 声道输出, 保留声道 0 和 1 未动, 但混合声道 0 和 1 成输出声道 2 (被送到比如超低音声道)。

sub[=fc:ch]

添加一个超低音(subwoofer)声道到音频流。用于创建超低音声道的音频数据声道 0 和声道 1 的平均声音。四阶 Butterworth(巴特沃斯) 过滤器低通过滤声音的结果是默认截频 60Hz 且在音频流中添加了一个独立的声道。

警告: 停用此过滤器, 当播放 DVDs 带杜比(Dolby Digital 5.1)声音时, 否则此过滤器分解声音到超低音声道。

<fc>

低通过滤器的截频(以 Hz 单位) (20Hz 至 300Hz) (默认是: 60)。为了最佳结果, 尝试设置截频尽可能低。这能改善立体声或环绕音效的体验。

<ch>

决定插入超音声道音频的声道号。声道号在 0 和 5 之间 (默认是: 5)。观察到声道数会自动增加到 <ch>, 如果必要的话。

示例:

```
mplayer -af sub=100:4 -channels 5 media.avi
```

添加超低音声道到输出声道 4 且其中截频为 100Hz。

center

从前方声道中创建中心声道。可能目前品质较低, 因为还没有实现用高通滤波器进行适当抽取, 而是用平均和半折其前方声道替代。

<ch>

决定插入中心声道的声道号。声道号在 0 和 5 之间 (默认是: 5)。观察到声道数会自动增加到 <ch>, 如果必要的话。

surround[=delay]

矩阵编码的环绕音效解码器, 像杜比环绕音效(Dolby Surround)。许多双声道音频文件事实上含有矩阵编码的环绕音效。需要声卡支持至少 4 声道。

<delay>

后方扬声器的延迟时间(以 ms 为单位) (0 至 1000) (默认是: 20)。此延迟按如下设置: 如果 d1 是聆听点到前方扬声器的距离, d2 是聆听点到后方扬声器的距离, 那么延迟该 设为 15ms 如果 d1 <= d2 或设为 15 + 5*(d1-d2) 如果 d1 > d2。

示例:

```
mplayer -af surround=15 -channels 4 media.avi
```

添加环绕音效解码, 后方扬声器声音为 15ms 的延迟。

delay[=ch1:ch2:...]

延迟声音到扬声器, 这样可以让不同声道的声音同时到达聆听点。这只对多于两个扬声器时有用。

ch1,ch2,...

应该加给每个声道的延迟时间(以 ms 为单位) (0 至 1000 之间的浮点数)。

不同声道所需的延迟时间计算如下:

1.

测量不同的扬声器到聆听点的距离(以 meters(米) 为单位), (对于 5.1 的系统) 得到距离 s1 至 s5。没有补偿超低音的点 (因为你听不出差异)。

2.

取其中最大距离与各个 s1 至 s5 的差, 即 $s[i] = \max(s) - s[i]; i = 1...5$ 。

3.

计算所需的各个延迟时间(以 ms 为单位)为 $d[i] = 1000*s[i]/342; i = 1...5$ 。

示例:

```
mplayer -af delay=10.5:10.5:0:0:7:0 media.avi
```

延迟左前方和右前方声道各 10.5ms, 两个后方声道和超低音声道 0ms, 而中心声道 7ms。

export[=mmaped_file[:nsamples]]

使用内存映射 (mmap()) 导出进入的信号到其它处理过程。内存映射的区域含有头部:

```
int nch                /*声道数*/
int size               /*缓冲大小*/
unsigned long long counter /*用于保持同步, 更新每次
                        新导出的数据。*/
```

其余是 (无间隔方式) 有效载荷的 16-位数据。

<mmaped_file>

映射数据到文件 (默认是: ~/.mplayer/mplayer-af_export)

<nsamples>

每个声道的样本数 (默认是: 512)

示例:

```
mplayer -af export=/tmp/mplayer-af_export:1024 media.avi
```

导出每个声道的 1024 样本到 "/tmp/mplayer-af_export"。

extrastereo[=mul]

(线性地)增加左右声道的差异以添加某些 "现场" 回放效果。

<mul>

设置差异系数 (默认是: 2.5)。0.0 表示单声道 (两个声道的平均值), 1.0 表示声音未改变, 而 -1.0 表示交换了左右声道。

volnorm[=method:target]

没有失真的最大音量。

<method>

设置被用的方式。

- 1: 使用单样本平滑差异, 通过对旧的样本进行标准加权平均 (默认)。
- 2: 使用几个样本平滑差异, 通过对旧的样本进行标准加权平均。

<target>

为此样本类型设置目标幅度为最大值的分数 (默认是: 0.25)。

ladspa=file:label[:controls...]

装载 LADSPA (Linux Audio Developer's Simple Plugin API) 插件。此过滤器可重返进入, 所以多个 LADSPA 插件可同时使用。

<file>

指定 LADSPA 插件的库文件。如果设置了 LADSPA_PATH, 从中搜寻指定的文件。如果没设置, 你一定要提供一个完整说明的路径名。

<label>

指定库中的过滤器。有些库仅含一个过滤器, 但有些含许多。进入这里的 "help", 会列出指定库中所有可用的过滤器, 从而消除使用 LADSPA SDK 的 "listplugins"。

<controls>

控制 0 或多个浮点数值, 其决定装载的插件的行为(例如: 延迟, 阈值和增益)。在冗赘模式(添加 -v 到 MPlayer 命令行)下, 所有可用的控制及其有效范围可打印。从而消除使用 LADSPA SDK 的 "analyseplugin"。

comp

压缩/扩张过滤器, 话筒输入有用。防止人为的非常高的声音和提高非常低的声音的音量。此过滤器未经测试, 甚至可能不稳定。

gate

噪声门过滤器类似于 comp(压缩/扩张) 音频过滤器。此过滤器未经测试, 甚至可能不稳定。

karaoke

简单声音去除过滤器利用这样的事实: 人的声音通常录成单声道, 之后 "中心(center)" 混音进最后的音频流。小心此过滤器会把信号变成单声道。对 2-声道音轨工作得很好; 在非 2-声道立体声上别费劲去尝试。

视频滤镜

视频滤镜让你修改视频流及其属性。其句法是:

-vf <滤镜1[=参数1:参数2:...], 滤镜2, ...>

设置一连串的视频滤镜。

许多参数是可选的并且如果省略的话则设为默认值。要明确表示使用默认值, 则将参数设为'-1'。

参数 w:h 表示以像素为单位宽 x 高 h, x:y 表示从较大的图像的左上角开始数 x:y 的位置。

注意: 要获取可用视频滤镜列表, 参见 -vf help。

视频滤镜以列表的形式维护。有一些命令用于维护滤镜列表。

-vf-add <滤镜1[, 滤镜2, ...]>

在滤镜列表尾部加上参数中所给的滤镜。

-vf-pre <滤镜1[, 滤镜2, ...]>

在滤镜列表头部加上参数中所给的滤镜。

-vf-del <索引号1[, 索引号2, ...]>

删除位于指定索引号上的滤镜。索引号起始为0, 负数表示从列表尾部开始寻址 (-1表示最后一个)。

-vf-clr

完全清空滤镜列表。 Completely empties the filter list.

如果滤镜支持，你可以通过参数的名字访问参数。

—vf <滤镜>=help

打印特定滤镜的参数名及参数的取值范围。

—vf <滤镜=已命名参数1=参数值1[:已命名参数2=参数值2:...]>

将一个已命名参数设置为指定参数值。使用 on 和 off 或 yes 和 no 来设置标志位参数。

可用插件有:

crop[=w:h:x:y]

切割出图像的指定部分并丢弃其余部分。对去掉宽银幕电影的黑边很有用。

<w>,<h>

切割出来部分的宽和高，默认值为原始的宽度和高度。

<x>,<y>

切割出来部分的位置，默认值是中央。

cropdetect[=0-255]

计算必要的切割参数并把推荐值显示在标准输出上。

<limit>

门限值，可设范围为无（0）至所有（255）（默认值：24）

<round>

指定宽:高值为该值的倍数（默认值：16）。偏移值将被自动调整以使视频位于屏幕中心。使用 2 以只得到偶数尺寸（4:2:2 的视频需要此选项）。16 对于大多数编码器的编码来说是很适合的。

rectangle[=w:h:x:y]

该插件响应 input.conf 中的两参数指令‘change_rectangle’。

<w>,<h>

宽度和高度（默认值：-1，在边缘可见的条件下宽度最大的可能值）。

<x>,<y>

左上角的位置（默认值：-1，屏幕最左上角）

expand[=w:h:x:y:o:a:r]

将影片的分辨率扩展（而不是缩放）至指定的值并将未缩放的原点移至x,y坐标。可以用于把字幕/ OSD 放置在该滤镜生成的黑边中。

<w>,<h>

扩展后的宽、高（默认值：原始的宽、高）。w和h的负数值将被视作为相对于原始大小的增量。

示例：

expand=0:-50:0:0

在图片的底部添加50像素的边。

<x>,<y>

原始图像在扩展后图像中的位置（默认：位于中心）

<o>

渲染OSD/ 字幕

0：禁用（默认值）

1：启用

<a>

扩展时适应某个宽高比而不是某个分辨率（默认值：0）。

示例：

expand=800:::4/3

除非源视频有更高的分辨率，否则扩展至800x600，当出现前一种情况时，其将扩展直至填满一个4/3宽

高比的区域。

<r>

舍入数值以使宽和高值能被<r>整除（默认值：1）。

flip（另参见 -flip）

翻转图像使其上下颠倒。

mirror

使图像相对于Y轴反转。

rotate[=<0-7>]

以90度为单位旋转图像并有选择地将其翻转。对于4至7间的值，只有当影片的几何形状为肖像式（译注：宽度比高度小）而不是风景画式（译注：宽度比高度大）时才执行旋转操作。

0

顺时针旋转90度并翻转（默认值）。

1

顺时针旋转90度。

2

逆时针旋转90度。

3

逆时针旋转90度并翻转。

scale[=w:h[:ilaced[:chr_drop[:par[:par2[:presize[:noup[:arnd]]]]]]]]

使用软件缩放器（速度慢）缩放图像并执行YUV<->RGB色彩空间转换（另参见 -sws）。

<w>,<h>

缩放后的宽/高值（默认值：原始的宽/高值）

注意：若使用了 -zoom，并且底层的滤镜（包括libvo）无法执行缩放，则其默认设置为 d_宽/ d_高！

0：缩放后的d_宽/ d_高

-1：原始的宽/高

-2：使用另一个尺寸的值和缩放前的宽高比计算w/h。

-3：使用另一个尺寸的值和原始的宽高比计算w/h。

-(n+8)：与以上的-n相似，但舍入尺寸值至最接近的16的倍数。

<ilaced>

触发隔行扫描视频的缩放方式。

0：关闭（默认）

1：打开

<chr_drop>

色度信息的跳过方式 chroma skipping

0：使用所有可用输入扫描行上的色度信息。

1：只使用每隔2个输入扫描行上的色度信息。

2：只使用每隔4个输入扫描行上的色度信息。

3：只使用每隔8个输入扫描行上的色度信息。

<par>[:<par2>]（另参见 -sws）

根据 -sws 中所选的缩放器类型设置缩放参数。

-sws 2（双立方插值）：B（雾化）和C（色斑）

0.00:0.60 默认值

0.00:0.75 VirtualDub中的“精确双立方插值”

0.00:0.50 Catmull-Rom插值

0.33:0.33 Mitchell-Netravali插值

1.00:0.00 立方式B-插值

- sws 7 (高斯模糊) : 锐化度 (0 (柔和) - 100 (锐利))
- sws 9 (LANCZOS迭代) : 过滤长度 (1-10)

<resize>

缩放至预设尺寸。

- qntsc : 352x240 (NTSC四分之一大小屏幕)
- qpal : 352x288 (PAL四分之一大小屏幕)
- ntsc : 720x480 (标准NTSC)
- pal : 720x576 (标准PAL)
- sntsc : 640x480 (正方形像素NTSC)
- spal : 768x576 (正方形像素PAL)

<noup>

不允许放大时超出原始尺寸。

- 0 : 允许放大时超出 (默认)。
- 1: 若有一个尺寸超出其原始值时则不允许。
- 2: 若有两个尺寸均超出其原始值时则不允许。

<arnd>

垂直缩放器使用精确数值舍入, 这可能比默认的舍入方式快速或缓慢。

- 0 : 禁止精确数值舍入 (默认)
- 1 : 允许精确数值舍入 (默认)

dsize[=aspect|w:h:aspect-method:r]

更改滤镜链上任意位置要使用的显示大小/ 宽高比。 宽高比可以是分数 (4/3) 或浮点数 (1.33)。 或者, 你可以指定所希望显示的宽和高的实际大小。 注意这个滤镜自己不 做任意缩放操作; 它只影响到后面的 (硬件或软件) 缩放器在修正 宽高比时将采取什么样的行为。

<w>,<h>

新显示模式的宽度和高度。 也可以是以下这些特殊值:

- 0 : 原来显示模式的宽度和高度
- 1 : 原始视频的宽度和高度 (默认)
- 2 : 使用另一个尺寸的值和原来显示模式的宽高比计算w/h。
- 3 : 使用另一个尺寸的值和原始视频宽高比计算w/h。

示例:

dsize=800:-2

当视频宽高比为4/3时指定显示分辨率为800x600, 或当视频宽高比为16/9时指定为800x450。

<aspect-method>

根据原来的宽高比修改宽度和高度。

- 1 : 忽略原有的宽高比。 (默认)
- 0 : 在<w>和<h>作为最大的分辨率的情况下保持显示模式的宽高比。
- 1 : 在<w>和<h>作为最小的分辨率的情况下保持显示模式的宽高比。
- 2 : 在<w>和<h>作为最大的分辨率的情况下保持视频的宽高比。
- 3 : 在<w>和<h>作为最小的分辨率的情况下保持视频的宽高比。

示例:

dsize=800:600:0

指定显示分辨率小于等于800x600, 以保持宽高比。

<r>

将宽度和高度值舍入至可被<r>整除 (默认值: 1)。

yuy2

强制执行YV12/ I420/ 422P至YUY2的软件转换。 对于那些对YV12支持较差但对YUY2支持较好的显卡很有用。

yvu9
强制执行YVU9至YV12色彩空间的软件转换。 不赞成使用，因为使用软件色彩调整器更好。

yuvdsp
将YUV色彩值限制在CCIR 601定义的范围内而不做实际的色彩转换。

rgb2bgr[=swap]
RGB 24/32 <-> BGR 24/32色彩空间转换。
swap
同样是执行R <-> B转换。

palette
使用palette（调色板）进行RGB/BGR 8 -> 15/16/24/32bpp色彩空间转换。

format[=fourcc]
限制下一个滤镜使用的色彩空间而不做任何转换。 与色彩调整滤镜共同作用完成实际转换。
注意： 要获取可用格式列表参见format=fmt=help。
<fourcc>
格式名称如rgb15、bgr24、yv12（默认值：yuy2）

noformat[=fourcc]
限制下一个滤镜使用的色彩空间而不做任何转换。 与format滤镜不同，这个滤镜能处理任意色彩空间 除了 你自己定义的。
注意： 要获取可用格式列表参见noformat=fmt=help。
<fourcc>
格式名称如rgb15、bgr24、yv12（默认值：yv12）

pp[=filter1[:option1[:option2...]]/[-]filter2...]（另参见 -pphelp）
启用指定序列的后期处理子滤镜。 子滤镜必须用'/'分隔并且可以通过在其名字前加以 '-'禁用。 每个子滤镜和一些选项有一个短名和一个长名，两者可以互换使用。例如 dr/dering 是一样的。 一些共用的选项在所有的子滤镜中使用以决定这些子滤镜的作用范围：
a/autoq
如果CPU太慢则自动关闭子滤镜。
c/chrom
同时执行色度过滤（默认值）。
y/nochrom
只执行亮度过滤（不使用色度信息）。
n/noluma
只执行色度过滤（不使用亮度信息）。
注意： -pphelp列出了可用子滤镜的列表。

可用的子滤镜有
hb/hdeblock[:difference[:flatness]]
水平方向上的反块效应（deblocking）滤镜
<difference>：差别系数，值越高意味着反块效应效果越强（默认值：32）。
<flatness>：平坦度阈值，值越低意味着反块效应效果越强（默认值：39）。
vb/vdeblock[:difference[:flatness]]
垂直方向上的反块效应（deblocking）滤镜
<difference>：差别系数，值越高意味着反块效应效果越强（默认值：32）。
<flatness>：平坦度阈值，值越低意味着反块效应效果越强（默认

值：39)。

ha/hadeblock[:difference[:flatness]]

精确的水平方向上的反块效应 (deblocking) 滤镜

<difference>：差别系数，值越高意味着反块效应效果越强 (默认值：32)。

<flatness>：平坦度阈值，值越低意味着反块效应效果越强 (默认值：39)。

va/vadeblock[:difference[:flatness]]

精确的垂直方向上的反块效应 (deblocking) 滤镜

<difference>：差别系数，值越高意味着反块效应效果越强 (默认值：32)。

<flatness>：平坦度阈值，值越低意味着反块效应效果越强 (默认值：39)。

水平反块效应滤镜和垂直反块效应滤镜共用同一个差别系数和平坦度阈值，所以不能在水平和垂直方向上设置不同阈值。

h1/x1hdeblock

试验性的水平方向上的反块效应 (deblocking) 滤镜

v1/x1vdeblock

试验性的垂直方向上的反块效应 (deblocking) 滤镜

dr/dering

去色斑 (deringing) 滤镜

tn/tmpnoise[:threshold1[:threshold2[:threshold3]]]

瞬时噪声去除器

<threshold1>：更大规模->更强的过滤

<threshold2>：更大规模->更强的过滤

<threshold3>：更大规模->更强的过滤

al/autolevels[:f/fullyrange]

自动更正亮度/对比度

f/fullyrange：增大亮度范围至 (0-255)。

lb/linblenddeint

线性混合反隔行扫描滤镜，通过使用 (1 2 1) 过滤器过滤所有的输入扫描行以解除所给区块的隔行扫描。

li/linipoldeint

线性插值反隔行扫描滤镜，通过线性插值算法每隔两个扫描行执行插值以解除所给区块的隔行扫描。

ci/cubicpoldeint

立方插值反隔行扫描滤镜，通过立方插值算法每隔两个扫描行执行插值以解除所给区块的隔行扫描。

md/mediandeint

中位数反隔行扫描滤镜，通过每隔两个扫描行应用中位数过滤器以解除所给区块的隔行扫描。

fd/ffmpegdeint

FFmpeg反隔行扫描滤镜，通过每隔两个扫描行使用 (-1 4 2 4 -1) 过滤器以解除所给区块的隔行扫描。

l5/lowpass5

垂直方向上应用的FIR低用反隔行扫描滤镜，通过对所有输入扫描行使用 (-1 2 6 2 -1) 过滤器以解除所给区块的隔行扫描。

fq/forceQuant[:quantizer]

将输入进来的量化表替换为你指定的常数量化表。

<quantizer> : 所使用的量化表

de/default

默认pp过滤器组合 (hb:a,vb:a,dr:a)

fa/fast

快速pp过滤器组合 (h1:a,v1:a,dr:a)

ac

高质量pp过滤器组合 (ha:a:128:7,va:a,dr:a)

示例 :

-vf pp=hb/vb/dr/al

水平和垂直方向上反块效应、去色斑以及自动亮度/ 对比度调整

-vf pp=de/-al

使用默认的过滤器组合而不使用亮度/ 对比度调整

-vf pp=default/tmpnoise:1:2:3

启用默认的过滤器组合以及瞬时噪声去除器。

-vf pp=hb:y/vb:a

水平方向反块效应只针对亮度信息，并且根据可用CPU时间资源打开或关闭垂直方向的反块效应过滤。

spp[=quality[:qp[:mode]]]

简单后期处理滤镜，它在多次（或者——在质量等级为6时——在所有的）变换中压缩和解压图像并使结果平均化。

<quality>

0-6（默认值：3）

<qp>

强制使用的量化参数（默认值：0，使用来自视频的量化参数）。

<mode>

0：刚性阈值（默认值）

1：柔性阈值（去色斑效果好，但较模糊）

4：与0相似，但同时使用B帧的量化参数（可能导致闪烁）

5：与1相似，但同时使用B帧的量化参数（可能导致闪烁）

uspp[=quality[:qp]]

极简单且缓慢的后期处理滤镜，它在多次（或者——在质量等级为8时——在所有的）变换中压缩和解压图像并使结果平均化。其与spp行为的差异在于uspp在所有情况下使用libavcodec Snow编码及解码，而spp仅使用类似于MJPEG的简化的内部8x8DCT变换。

<quality>

0-8（默认值：3）

<qp>

强制使用的量化参数（默认值：0，使用来自视频的量化参数）。

fspp[=quality[:qp[:strength[:bframes]]]]

简单后期处理滤镜的一个快速版本

<quality>

4-5（与spp中的等价；默认值：4）

<qp>

强制使用的量化参数（默认值：0，使用来自视频的量化参数）。

<-15-32>

过滤强度，值越低意味着加入更多细节但同时也带来更多损伤，而值越高越使图像平滑但同时也越模糊（默认值：0 - PSNR优化）。

<bframes>

0：不使用B帧的量化参数（默认值）

1：也使用B帧的量化参数（可能导致闪烁）

pp7[=qp[:mode]]

spp滤波的变种，与使用7顶点DCT的spp=6相似，在这种情况下只有中央的采样在IDCT后使用。

<qp>

强制使用的量化参数（默认值：0，使用来自视频的量化参数）。

<mode>

0：刚性阈值

1：柔性阈值（去色斑效果较好，但较模糊）

2：中性阈值（默认值，效果好）

qp=equation

量化参数（QP）变换滤镜

<equation>

诸如“2+2*sin(PI*qp)”之类的方程式

geq=equation

通用方程式变换滤镜

<equation>

某种方程式，例如用以水平翻转图像的‘p(W-XY)’。你可以使用空白分隔符使方程式更容易读懂。有一些常量可以在方程式中使用：

PI：圆周率派

E：自然常数e

X / Y：当前采样中的坐标

W / H：图像的宽度和高度

SW / SH：当前滤镜处理过后的平面的宽/高度量尺度，例如用于YUV 4:2:0的1,1和 0.5,0.5。

p(x,y)：返回当前平面x/y位置上像素的值。

test

生成各种测试中的测试用图样。

rgbttest

生成一个RGB测试图样，适用于对比RGB和BGR的场合。你会看到由上而下的红、黄、蓝三色条纹。

lavc[=quality:fps]

使用libavcodec的YV12至MPEG-1的快速软件转换，用于与DVB/ DXR3/IVTV/ V4L2一同使用。

<quality>

1–31：固定的量化尺度

32–：以kbits为单位的固定比特率

<fps>

强制指定的输出fps值（浮点值）（默认值：0，根据图像高度自动检测）

dvbscale[=aspect]

设置用于DVB卡的优化缩放比例，用硬件度量x轴而用软件计算y轴比率以保持宽高比。只有当与expand和scale滤镜一同使用时才有用。

<aspect>

控制宽高比例，计算方式为DVB_HEIGHT*ASPECTRATIO（默认

值：576*4/3=768），对于16:9的电视将其设置为576*(16/9)=1024。

示例：

–vf dvbscale,scale=1:0,expand=1:576:-1:-1:1,lavc

FIXME：解释一下这个是用来干什么的。

noise[=luma[u][t|a][h][p]:chroma[u][t|a][h][p]]

添加噪声效果。

<0-100>

亮度噪声

<0-100>

色度噪声

u

均匀噪声（否则就是高斯分布的）

t

瞬时噪声（噪声样式在不同帧上是不同的）

a

平均化瞬时噪声（更平滑，但相较而言渲染得很慢）

h

高质量（稍显好看，渲染稍慢）

p

具有（半）规则样式的混合随机噪声

denoise3d[=luma_spatial:chroma_spatial:luma_tmp:chroma_tmp]

该滤镜目标在于消除图像上的噪声，产生平滑的图像，并且使静态图像真正静止（这能提高可压缩性。）。

<luma_spatial>

空间亮度强度（默认值：4）

<chroma_spatial>

空间色度强度（默认值：3）

<luma_tmp>

亮度瞬时强度（默认值：6）

<chroma_tmp>

色度瞬时强度（默认值：luma_tmp*chroma_spatial/luma_spatial）

hqdn3d[=luma_spatial:chroma_spatial:luma_tmp:chroma_tmp]

denoise3d滤镜的高精确/ 质量版本。 参数与用法与denoise3d相同。

eq[=brightness:contrast]（已被淘汰）

就像硬件均衡器一样的可以交互式控制的软件均衡器，用于那些不支持硬件级 亮度和对比度控制的显卡/ 驱动。 可能对于MEncoder很有用，不管是用来修改抓取得不好的影片，还是稍稍降低对比度以隐藏画质损伤以及在低比特率下保持质量

<-100-100>

初始亮度

<-100-100>

初始对比度

eq2[=gamma:contrast:brightness:saturation:rg:gg:bg:weight]

另一种软件均衡器，其使用对照表（运行缓慢），除了简单的亮度和对比度 调整外还允许伽玛修正。 注意当所有的伽玛值为1.0时，它与-vf eq使用同样的MMX优化代码。 参数值是以浮点值形式给出的。

<0.1-10>

初始伽玛值（默认值：1.0）

<-2-2>

初始对比度，其中负数值将导致图像呈现负片效果（默认 值：1.0）

<-1-1>

初始亮度（默认值：0.0）

<0-3>

初始饱和度（默认值：1.0）

<0.1-10>

红色成份的伽玛值（默认值：1.0）

<0.1-10>

绿色成份的伽玛值（默认值：1.0）

<0.1-10>

蓝色成份的伽玛值（默认值：1.0）

<0-1>

权值参数可以用来消减速高伽玛值在图像明亮区域的效果，比方说防止这些区域过度增益而变得仅仅是纯白色。值为0.0则使得在所有情况下执行伽玛值修正，而值为1.0则让伽玛值始终保持其修饰力度（默认值：1.0）。

hue[=hue:saturation]

就像硬件均衡器一样的可以交互控制的软件均衡器，，用于那些不支持硬件级色调和饱和度控制的显卡/驱动。

<-180-180>

初始色调（默认值：0.0）

<-100-100>

初始饱和度，其中负数值将导致负色度（默认值：1.0）

halfpack[=f]

将按平面存放的YUV 4:2:0格式转换成半高度打包的4:2:2格式，对于亮度值降低质量采样但保留所有色度采样值。适用于当硬件收缩图像的质量不高或无法提供该功能时，将信号输出至一个低分辨率显示设备的场合。也可以用来作为一种占用CPU低的仅针对亮度信号的基本反隔行扫描滤镜。

<f>

默认情况下，半打包过程在降低质量采样时在每对扫描行间取平均值。任何不为0或1的值都是指定采用默认的（取平均值的）行为。

0：在降低质量采样时只使用偶数扫描行的信息。

0：在降低质量采样时只使用奇数扫描行的信息。

ilpack[=mode]

当隔行扫描视频以YUV 4:2:0格式存放时，由于色度信道在垂直方向的降低质量采样的关系，色度信号在隔行扫描中没有恰当地排列。该滤镜将按平面存放的4:2:0的数据打包进YUY2（4:2:2）格式时将色度信号恰当地排列至应处的位置，以便在任何扫描行中，亮度和色度数据来源于同一个数据域时。

<mode>

选择采样的模式。

0：最接近邻居采样模式，快速但不准确

1：线性插值（默认方式）

harddup

只适用于MEncoder。如果在编码中使用了hardup，则它将强制使重复的帧编码至输出中。这将占用稍多的空间，但对于输出至MPEG文件或当你要在编码后分离再合并视频流时却有必要。应被放置于滤镜链的最后或接近于最后，除非你有充足的理由不这样做。

softskip

只适用于MEncoder。softskip将编码中跳过（丢弃）帧的步骤从滤镜链之前移至滤镜链中的某个位置。这使那些需要读取所有帧的滤镜（反电视图像模式、反瞬时噪声等等）能正常工作。应被放置在那些需要读取所有帧的滤镜之后，而在这些滤镜后的任何大量消耗CPU的滤镜之前。

decimate[=max:hi:lo:frac]

丢弃那些与先前的帧差别不大的帧，以降低帧率。该滤镜的主要用途是为了在极低的比特率下编码（例如在拨号接入的调制解调器线路上传输流媒体），但在理论上它可以用于修复那些不准确地去掉电视模式过的影片）。

`<max>`
设置最大可以连续丢弃的帧的个数（当该值为正数时），或被丢弃 帧间的最小间隔（当该值为负数时）。

`<hi>,<lo>,<frac>`
当没有一个8x8的区域的差别值大于阈值<hi>，并且不超过<frac>个部分（1表示 整幅画面）的差别值大于阈值<lo>时，这个帧就可以被丢弃。数值<hi>和<lo>表示8x8像素区块中实际存在的不同值的像素个数，于是阈值64 对应于每个像素都有一个不同的值，或这种情形以不同的形式在整个区块中分布 的形式。

`dint[=sense:level]`
这个丢弃反隔行扫描（dint）滤镜侦测出并丢弃一连串隔行扫描视频的帧中的 第一个帧。

`<0.0–1.0>`
相邻像素间的相对差别值（默认值：0.1）

`<0.0–1.0>`
图像中的哪个部分必须被侦测以判定是否为隔行扫描以便丢弃帧（默认值：0.15）。

`lavcdeint`（已被淘汰）
FFmpeg中的反隔行扫描滤镜，与`-vf pp=fd`相同

`kerndeint[=thresh[:map[:order[:sharp[:twoway]]]]]`
Donald Graft的自适应内核级反隔行扫描滤镜。当超过配置中的一个阈值时，解除视频中的隔行扫描部分。

`<0–255>`
阈值（默认值：10）

`<map>`
0：忽略超出阈值的像素（默认方式）。
1：将超出阈值的像素输出为白色。

`<order>`
0：不对场次序做更改（默认方式）。
1：交换场的次序。

`<sharp>`
0：禁用额外的锐化处理（默认方式）。
1：启用额外的锐化处理

`<twoway>`
0：禁用双向锐化处理（默认方式）。
1：启用双向锐化处理。

`unsharp[=l|cWxH:amount[:l|cWxH:amount]]`
反锐化修饰 / 高斯模糊

`l`
将效果应用于亮度部分。

`c`
将效果应用于色度部分。

`<width>x<height>`
矩阵的宽度和高度，两者都是奇数尺寸（最小值 = 3x3，最大值 = 13x11 或11x13，通常在3x3至7x7之间）

`amount`
图像锐力度/ 模糊度的相对增量（合理的值应为-1.5–1.5）。
 <0：模糊
 >0：锐力

swapuv

交换U和V两个平面的位置。

ii[=d|i][s][:[d|i][s]]

对扫描行执行（反）交错操作。该滤镜的目标是提供处理隔行扫描图像的前半个扫描场而不对其进行反隔行扫描操作的功能。你可以在不破化隔行扫描结构的情况下过滤你的隔行扫描DVD视频，然后让其在电视上播放。（在后期处理滤镜的作用下）反隔行扫描（通过柔化、平均化之类的手段）永久性地去除了隔行扫描结构，而解交错操作将帧分离成2个扫描场（所谓的半幅画面），因此你可以分别地处理（过滤）这些扫描场然后重新将它们交错起来。

d

反交错操作（将一个扫描场置于另一个上方）

i

交错操作

s

交换扫描场的位置（交换奇偶扫描线的位置）

fil[=i|d]

对扫描行执行（反）交错操作。与ii滤镜十分相像，但相较之下运行得很快，其主要的缺点是它不是总能工作。尤其是当它与其它滤镜一起使用时，它可能随机产生一些渲染错误的图像，所以如果它能工作则暗自庆幸吧，但如果它在你的滤镜组合中不能工作也不要抱怨。

d

对扫描场执行反交错操作，将它们一个个紧挨着放在一起。

i

对扫描场再次执行交错操作（产生与fil=d相反的效果）。

field[=n]

使用大跨度计算方式从一幅隔行扫描图像中抽取单个扫描场，以避免浪费CPU时间。可选参数n指定抽取的是偶数场还是奇数场（取决于n是偶数还是奇数）。

detc[=var1=value1:var2=value2:...]

尝试逆转‘电视图像模式’处理操作以还原一个电影帧率下的纯净、非隔行扫描的视频流。这是第一个也是最基本的一个添加到MPlayer/ MEncoder中的反电视图像模式滤镜。其工作方式是锁定电视图像模式中的3:2画面特征，然后尽可能长地跟踪这个特征。这使它适合完全电视图像模式化的来源，即使在有一定噪声的情况下亦是如此，但不适用于复杂的后期电视图像模式的图像的编辑。这个滤镜的开发已不再继续，因为ivtc、pullup、和filmdint在大多数应用中效果更好。以下参数（参见下面的句法解释）可用于控制detc的行为：

<dr>

设置丢弃帧的模式。

0：不丢弃帧以保证输出帧率恒定（默认方式）。

1：当最近5个帧没有丢弃或不是电视图像模式时则一定丢弃一个帧。

2：总保证输入输出帧率比正好为5:4。

注意：模式1和2用于MEncoder。

<am>

分析模式。

0：固定特征，初始帧代号由<fr>中所指定。

1：积极搜寻电视图像模式特征（默认值）

<fr>

设定帧序列中初始帧的代号。0-2是三个纯逐行扫描的帧；3和4是两个隔行扫描的帧。默认值-1表示‘初始帧不在电视图像模式的序列中’。这里指定的代号是影片开始前假想的那个前一帧的类型。

<t0>, <t1>, <t2>, <t3>

用于某些模式下的阈值。

ivtc[=1]

实验性的‘无状态’逆转电视图像模式滤镜。 ivtc并不像detc滤镜那样尝试锁定一个特征，而是对于每一帧独立地决策。 这对于那些在应用电视图像模式后经历较多编辑的来源相对而言具有很好的效果，但相应地它相对不能容忍有噪声的输入，比如电视上捕捉来的视频。 可选参数 (ivtc=1) 对应于detc滤镜的dr=1选项，并且应当用于MEncoder中而不能用于MPlayer。 与detc一样，你必须在使用MEncoder时指定正确的输出帧率

(-ofps24000/1001)。 ivtc的进一步开发已停止，因为pullup和filmdint滤镜显得相对精确得多。

pullup[=jl:jr:jt:jb:sb:mp]

第三代反pulldown (逆转电视图像模式) 滤镜，能够处理混合型硬性电视图像模式的，24000/1001帧率逐行扫描的，以及30000/1001帧率逐行扫描的内容。 pullup滤镜的设计原则是通过在决策中利用将来的内容环境，以求得比detc或 ivtc更稳定。 与ivtc相似，pullup是无状态的，这源于它不锁定扫描场以识别要匹配的内容然后重塑逐行扫描的帧。 它仍处于开发之中，但人们相信它是相当精确的。

jl, jr, jt, and jb

这些选项用于设置图像的左边、右边、上边和下边各有多少“垃圾信息”要忽略。 左/ 右两边是以8像素为一个单元，而上/ 下两边是以2个扫描行为一个单元。 默认是每边各取8个像素。

sb (strict breaks)

将该选项设置为1将减小pullup产生偶发性不匹配帧的机率，但它也可能导致高动态的帧序列中有太多的帧被丢弃。 相反地，将它设置为-1将使pullup能更容易地匹配扫描场。 这能帮助处理那些在扫描场间存在一些模糊特性的视频，但也可能导致输出中存在隔行扫描的帧。

mp (metric plane)

这个选项可以设置成1或2以使用色度平面而不是亮度平面来进行pullup的计算。 这能提高处理清晰来源时的精确度，但更可能的是降低精确度，尤其是存在色度噪声 (彩虹效应) 或灰度视频的场合。 将mp设置成色度平面的主要目的是降低CPU负荷，并使pullup在慢速机器上可用。

注意：在编码中一定要在pullup后跟上softskip滤镜以保证pullup能读取所有帧。 由于编解码器/ 滤镜层在设计上的局限，不这样做会导致输出不正确，并且往往程序会崩溃。

filmdint[=options]

逆转电视图像模式滤镜，与上面的pullup滤镜相似。 它被设计成用于处理任意下拉特征，包括混合型软性和硬性的电视图像模式，以及提供对于那些会慢于或快于其在电视上的原始帧率的影片的有限支持。 只有亮度平面用以寻找帧分开的位置。 如果一个扫描场没有匹配上特征，则就使用简单的线性估计方式解除其隔行扫描结构。 如果来源是MPEG-2格式，那么这个滤镜必须是第一个滤镜以便能够访问MPEG-2解码器设置的扫描场标志。 根据来源MPEG的不同，你可能不需采纳这个建议，只要你不看到许多“下半场先扫描”的警告。 不设选项时，它只是做普通的逆转电视图像模式处理，并且应当与 mencoder -fps 30000/1001 -ofps 24000/1001一同使用。 当这个滤镜用于mplayer时，它将导致播放时帧速率不均衡，但在通常情况下比使用pp=lb或不进行解除隔行扫描操作要好。 多个选项可以使用./分隔开指定。

crop=<w>:<h>:<x>:<y>

就像crop滤镜一样，但更快而且能用于混合型硬性和软性电视图像模式的内容，以及y不用4的倍数的情形。 如果x或y的设置将需要从色度平面中切除部分像素时，切除区域将延伸。 这通常意味着x和y必须是偶数。

io=<ifps>:<ofps>

对于每ifps个输入帧，滤镜将输出ofps个帧。 ifps/ ofps的比率应当与

—fps/—ofps的比率相匹配。 这可以用于过滤那些在电视上以不同原始帧率的帧率播放的影片。

luma_only=<n>

如果n为非零值，色度平面就原封不动地输出。 这适用于YV12采样的电视，这种方式丢弃其中一个色度扫描场。

mmx2=<n>

在x86系统上，如果n=1，则使用MMX2优化的函数，如果n=2，则使用3DNow! 优化的函数，否则使用普通的C语句。 如果不指定这个选项，则将自动侦测MMX2和3DNow!，使用这个选项重写自动侦测 的结果。

fast=<n>

n值较大时能提高滤镜的运行速度但牺牲了精确性。 默认值是n=3。 如果n是奇数，MPEG的REPEAT_FIRST_FIELD标志位已标记的帧后紧跟的一帧将被 认作是逐行扫描的，于是滤镜将不在处理软性电视图像模式的MPEG-2内容上花任 何时间了。 如果MMX2或3DNow!可用，那么这就是该标志位的唯一作用。 在没有MMX2和3DNow!的情况下，如果n=0或1，那么将使用与n=2或3的时候同样的 计算方式。 如果n=2或3，用于寻找帧间断的亮度阶数值就从256降低为128，这使滤镜运行更 快而不损失太多的精确性。 如果n=4或5，将使用一个更快的，但相较之下很不精确的测量标准来寻找帧间 断，这样做会增加将垂直方向上细节很多的图像误测为隔行扫描内容的可能性。

verbose=<n>

如果n是非零值，则打印出用于每个帧的具体测量标准。 适用于程序调试。

dint_thres=<n>

解除隔行扫描的阈值。 用于解除未匹配任何特征的帧的隔行扫描结构的过程中。 阈值大意味着较少的帧经过解除隔行扫描处理，要完全关闭解除隔行扫描处理则 使用n=256。 默认值为n=8。

comb_thres=<n>

比较上半扫描场和下半扫描场的阈值。 默认值为128。

diff_thres=<n>

用于侦测扫描场的瞬时变化的阈值。 默认值为128。

sad_thres=<n>

合计绝对差的阈值，默认为64。

softpulldown

这个滤镜只能用于MEncoder并且依赖于用于软性3:2下拉（软性电视图像模式）的MPEG-2标志位。 如果你想对半软性电视模式的影片使用ivtc或detc滤镜，那么将该滤镜插入到它们之前 能使他们更稳定。

divtc[=options]

逆转隔行扫描视频的电视图像模式。 如果3:2下拉电视图像模式的视频丢失了其中的一个扫描场，或在解除隔行扫描时使用了 保留一个扫描场而插值计算另一个的方法，那么输出的是一个晃动的视频，其中的每四 个帧后有一个是重复的帧。 该滤镜目的是找到并丢弃这些重复的帧，并还原原来的影片帧率。 在使用这个滤镜时，你必须设将—ofps指定为输入视频文件帧率的4/5 并将softskip放在滤镜链中位于其后的地方，以保证divtc能读取所有 的帧。 有两种不同的模式可选： 一阶段模式是默认的模式，并且直接就能使用，但缺点是什么电 视图像模式下相位的变化（丢帧或编辑错误）将导致暂时的画面 抖动直至滤镜重新恢复同步。 二阶段模式通过事先分析整个视频来避免这种问题，所以它能事 先知道相位的变化并能在准确的位置恢复同步。 这两个阶段 不 对应于第一阶段和第二阶段的编码处理过程。 你必须在真正编码而释放出处理后的视频之前，使用divtc的第一 阶段配置运行额外的一个阶段。 使用—nosound —ovc raw —o /dev

/null来避免在这一阶段浪费 CPU 的运算能力。 你可以在divtc后加上诸如crop=2:2:0:0之类的东西以运行 得更快。 然后使用divtc第二阶段配置以进行真正的编码。 如果你使用多阶段编码器编码，那么在所有阶段均要 使用divtc第二阶段的配置。 相关选项有：

pass=1|2

使用二阶段模式。

file=<filename>

设置第二阶段使用的日志的文件名（默认值：“framediff.log”）。

threshold=<value>

设置滤镜认为某个特征是电视图像模式特征时，该特征所必须达到的最小强度（默认 值：0.5）。 这用于避免从视频中很暗或很静止的部分里错误地识别出一些特征。

window=<numframes>

设置搜寻特征时需要察看多少个最近读到的帧（默认值：30）。 较长的察看范围能增加特征搜寻的可靠性，但较短的察看范围能改善对于电视图像模式中相位变化的反应时间。 该选项只对一阶段模式有作用。 当前，二阶段模式使用固定的察看范围，该范围包含了过去和将来输入的帧。

phase=0|1|2|3|4

设置一阶段模式中电视图像模式的初始相位（默认值：0）。 二阶段模式能读取将来输入的帧，所以它能够在一开始就使用正确的相位值，但 一阶段模式只能靠猜测。 当它找到正常的相位时它能跟上这个相位，但该选项能用来开始时可能出现的抖动。 二阶段模式的第一阶段也使用该选项，所以如果你保存第一阶段的输出内容，你就会得 到持续的相位值跟踪结果。

degghost=<value>

设置去鬼影处理的阈值（0–255用于一阶段模式，-255–255用于二阶段模式，默认0）。 如果为非零值，则使用去鬼影模式。 该选项用于那些通过将扫描场混合在一起而不是丢弃其中一个场的方式解除 隔行扫描的视频。 去鬼影处理会增强混合后帧中的压缩损伤，所以该参数值作为一个阈值，用以 将那些在去鬼影过程中与前一帧的相差值小于一个特定值的像素排除在外。 如果使用了二阶段模式，可以用负数值使滤镜在第二阶段开始时分析整个视频，以 确定它是否需要去鬼影处理，然后要么选择一个零值，要么选择该值的绝对值作 为去鬼影的参数。 应将该选项用于第二阶段，这与用于第一阶段没有差别

phase=[t|b|p|a|u|T|B|A|U][:v]

将隔行扫描的视频延迟一个扫描场的时间，以改变扫描场的次序。 其目的是修复那些在录像带传输至电脑视频的捕捉过程中使用了相反的扫描场次 序的PAL影片。 选项有：

t

捕捉时扫描场次序为上半场先输入，传输时为下半场先输入。 滤镜将延迟下半场。

b

捕捉时扫描场次序为下半场先输入，传输时为上半场先输入。 滤镜将延迟上半场。

p

捕捉和传输时使用同样的扫描场次序。 该模式只是为了在其它选项的使用说明中引用而存在的，但如果你真的选了它，那么滤 镜会按照您的旨意不做任何事情。;-)

a

捕捉时的次序将自动通过扫描场的标志位来判断，传输时的与捕捉时的相反。 滤镜根据扫描场的标志位为每一个帧在t和b两个模式中选择一个。 如果没有任何扫描场的信息，那么该模式就与u一样了。

u

捕捉时的次序未知或不断变化，传输时的与捕捉时的相反。 滤镜通过分析图像并选择能在扫描场间产生最佳匹配的那个候选模式来给每一个帧 在t和b两个模式中选择

一个。

T

捕捉时为上半场先输入，传输时未知或不断变化。滤镜通过分析图像在t和p模式中选择一个。

B

捕捉时为下半场先输入，传输时未知或不断变化。滤镜通过分析图像在b和p模式中选择一个。

A

捕捉时的次序由扫描场标志位判断，传输时的未知或不断变化。滤镜通过扫描场标志位和图像分析从t、b和p中选择一个模式。如果没有任何扫描场信息，则该模式与U一样。这是默认的模式。

U

捕捉时和传输时的次序均未知或不断变化。滤镜只通过图像分析以从t、b和p中选择一个模式。

V

细节信息处理。打印每个帧所选用的模式以及在t、b和p三种模式下的扫描场间的均方差值。

telecine[=start]

应用3:2'电视图像模式'处理以使帧率增加20%。该选项极可能无法用于MPlayer，但它可以用'mencoder -fps 30000/1001 -ofps 30000/1001 -vf telecine'形式使用。其中的两个fps选项都是必需的！（如果它们不正确，就无法A/V同步。）可选的start参数告诉滤镜从电视图像模式特征中的哪里开始执行（0-3）。

tinterlace[=mode]

瞬时扫描场隔行扫描化——将一对对帧合并为一个个隔行扫描的帧，使帧率减半。偶数帧移至上半扫描场，奇数帧移至下半扫描场。该滤镜可用于充分反转（模式0下）tfields滤镜的效果。可用模式有：

0

将奇数帧移至上半场，偶数帧移至下半场，以在半速帧率下产生一个完全高度的帧。

1

只输出奇数帧，丢弃偶数帧；帧的高度不变。

2

只输出偶数帧，丢弃奇数帧；帧的高度不变。

3

将每个帧扩展至完全高度，但每两个扫描行间插入黑色的行；帧速不变。

4

将偶数帧的偶数扫描行与奇数帧的奇数扫描行交叉在一起。帧高度不变，帧速减半。

tfields[=mode[:field_dominance]]

瞬时扫描场分离——将扫描场分离成帧，输出帧率加倍。就像telecine滤镜，tfields只在MEncoder中能正常工作，并且只在-fps 和-ofps设置成所需的（加倍的）帧率的情况下！

<mode>

0：保持扫描场不变（会产生跳帧/闪烁）。

1：插值恢复缺少的扫描行。（所用的算法可能不太好。）

2：使用线性插值法以1/4像素精度转换扫描场（不产生跳帧）。

4：使用4tap滤镜以1/4像素精度转换扫描场（较高质量）（默认方式）。

<field_dominance>（不推荐使用）

-1：自动（默认值）只有当解码器输出适当的信息并且在滤镜链中tfields之前没有其它的滤镜丢弃这些信息时才能工作，否则该值设回为0（上半

扫描场先输入)。

0：上半扫描场先输入

1：下半扫描场先输入

注意：该选项相当可能在以后的版里中去除。使用`-field-dominance`代替它。

`yadif=[mode[:field_dominance]]`

又一个解除隔行扫描的滤镜

<mode>

0：每存在一帧输出一帧。

1：每存在一个扫描场输出一帧。

2：与0相似但跳过空间隔行扫描检查。

3：与1相似但跳过空间隔行扫描检查。

<field_dominance> (不推荐使用)

与`tfields`运行方式相似。

注意：该选项相当可能在以后的版本中去除。使用`-field-dominance`替代它。

`mcdeint=[mode[:parity[:qp]]]`

包含运动补偿的解除隔行扫描滤镜。它要求每帧有一个扫描场作为输入并且必须与`tfields=1`或`yadif=1/3`或与之等价的滤镜一起使用。

<mode>

0：快速

1：中等

2：慢速，迭代式的运动估计

3：更慢，与模式2外加参照多个帧的方式相似

<parity>

0或1用于选择使用哪个扫描场（注意：目前还不能自动侦测！）。

<qp>

较高的值能产生较平滑的运动矢量场，但单个矢量得到优化的较少。

`boxblur=radius:power[:radius:power]`

盒状模糊

<radius>

模糊过滤的强度

<power>

应用过滤的数量

`sab=radius:pf:colorDiff[:radius:pf:colorDiff]`

自适合形状的模糊

<radius>

模糊过滤的强度（~0.1–4.0）（值越大越慢）

<pf>

预过滤强度（~0.1–2.0）

<colorDiff>

像素间会被认同的最大差值（~0.1–100.0）

`smartblur=radius:strength:threshold[:radius:strength:threshold]`

智能模糊

<radius>

模糊过滤的强度（~0.1–5.0）（值越大越慢）

<strength>

模糊化（0.0–1.0）或锐利化（-1.0–0.0）

<threshold>

过滤全部区域 (0) , 过滤平坦区域 (0-30) 或过滤边缘 (-30-0)

perspective=x0:y0:x1:y1:x2:y2:x3:y3:t
修正没有摄制得垂直于屏幕的影片视角。
<x0>,<y0>,...
左上角、右上角、左下角、右下角的坐标
<t>
线性 (0) 或立方式 (1) 重采样

2xsai
使用2x缩放及插值算法缩放并平滑图像。

1bpp
YUV/BGR 8/15/16/ 32转换中的1bpp映射位图

down3drigt[=lines]
重设立体图像的位置及大小。 将两个立体扫描场都抽取出来并将其紧靠着摆放, 以将它们缩放至保持原有 影片宽高比的大小。
<lines>
要从图像的中间选取的扫描行的数量 (默认值: 12)

bmovl=hidden:opaque:fifo
这个位图覆盖滤镜从FIFO管道中读取位图并将它们显示在影片的上方, 以支持某些对于图像的变换。 另参见TOOLS/bmovl-test.c以获取一个小型的bmovl测试程序。
<hidden>
设置'hidden'标志位的默认值 (0=可见, 1=不可见)。
<opaque>
设置'opaque'标志位的默认值 (0=透明, 1=不透明)。
<fifo>
FIFO管道 (连接'mplayer -vf bmovl'和主控程序的命名管道) 的路径/ 文件名

FIFO管道中使用的命令有:

RGBA32 width height xpos ypos alpha clear
其后出现的是width*height*4字节大小的原始RGBA32数据。

ABGR32 width height xpos ypos alpha clear
其后出现的是width*height*4字节大小的原始ABGR32数据。

RGB24 width height xpos ypos alpha clear
其后出现的是width*height*3字节大小的原始RGB24数据。

BGR24 width height xpos ypos alpha clear
其后出现的是width*height*3字节大小的原始BGR24数据。

ALPHA width height xpos ypos alpha
更改指定区域的阿尔法透明度。

CLEAR width height xpos ypos
清空某个区域。

OPAQUE
禁用一切阿尔法透明。 要再次启用则发送"ALPHA 0 0 0 0 0"

HIDE
隐藏位图。

SHOW
显示位图。

参数有:

<width>, <height>
图像/区域的大小

<xpos>, <ypos>

从x/y位置开始位图混合

<alpha>

设置阿尔法差值。 如果你将该值设为-255，你就可以发送一系列的阿尔法命令将该区域设置为-225、-200、-175等等以得到一个不错的渐渐出现的效果！;)

0：保持原来的值

255：使所有的位图不透明。

-255：使所有的位图透明。

<clear>

在位图混合前清空帧缓冲。

0：在原来的图像上进行位图混合，于是你不必每当屏幕中的一小部分更新时就发送1.8MB 的RGB32数据。

1：清空图像

framestep=l|[i]step

仅仅每隔n个帧或只对每个内部参照帧（关键帧）进行渲染。

如果你使用I（大写）作为参数调用这个滤镜，那么 只有 关键帧才渲染。对于DVD来说它通常意味着每15/12个帧中才有一帧（IBBPBBPBBPBBPBB），对于 AVI来说它意味着每当场景切换时或每隔keyint值（参见-lavcopts keyint= value） 所指定数量的帧过后才有一帧。

当找到关键帧时，将打印一个'!'字符串以及紧接着的一个换行符，以结束当前 MPlayer/ MEncoder在屏幕上输出的那一行，因为这段信息中包含了关键帧的时间值（以秒计）以及帧编号（你可以利用这一信息切分AVI。）。

如果你使用一个数值参数'step'调用这个滤镜，那么只有每隔'step'个帧过后 才有一个帧得到渲染。

如果你在数值前加上一个'i'（小写），那么将输出'!'（就像I参数一样）

如果你只给出i，那么不对帧做任何处理，只打印!。

tile=xtiles:ytiles:output:start:delta

将一系列图像拼成单个大图像。 如果你省略了一个参数或使用一个小于0的值，则将使用默认值。 你也可以在你认为满意的情况下停止指定参数（... -vf tile=10:5 ...）。 将缩放滤镜放在tile之前很可能是个好主意:-)

这些参数有：

<xtiles>

x轴方向拼贴的图像数（默认值：5）

<ytiles>

y轴方向拼贴的图像数（默认值：5）

<output>

当'output'个帧到达时，渲染拼贴的图像，这里'output'应当是一个小于 xtile * ytile的数。 拼贴图像中缺失的部分保留为空白。 比如，你可以每50帧写入一个8 * 7的拼贴图像，这样就在25fps下每2秒产生一幅 图像。

<start>

以像素为单位的外边框的厚度（默认值：2）

<delta>

以像素为单位的内边框的厚度（默认值：4）

delogo[=x:y:w:h:t]

通过根据周围的像素进行简单的插值来去除电视台的台标。 只需设置一个能盖住台标的

长方形区域然后看着它消失就行了（有时更难看 的画面会出现——你能得到效果是说不准的）。

<x>,<y>

台标的左上角

<w>,<h>

被清理的长方形区域的宽度和高度

<t>

长方形区域的模糊边缘的宽度（增加w和h的大小）。当设为-1时，屏幕上将绘出一个绿色的长方形以方便寻找合适的x、y、w、h参数值。

remove-logo=/path/to/logo_bitmap_file_name.pgm

去除电视台台标，使用PGM或PPM图像文件来判断哪些像素组成了台标。图像文件的宽度和高度必须与所处理的视频流的宽高相匹配。使用过滤图像以及一个循环模糊算法去除台标。

/path/to/logo_bitmap_file_name.pgm

过滤图像的[路径] + 文件名。

zrmjpeg[=options]

与zr2视频输出设备一同使用的软件YV12至MJPEG编码器。

maxheight=<h>|maxwidth=<w>

这些选项zr采集卡能处理的最大宽度和高度（MPlayer的滤镜层当前 还不能查询这些信息）。

{dc10+,dc10,buz,lm133}-{PAL|NTSC}

使用这些选项将maxwidth和maxheight自动设置为zr卡/混合模式所知的值。比如，有效的选项有：dc10-PAL和buz-NTSC（默认值：dc10+PAL）

color|bw

选择彩色或黑白的编码。黑白编码较快。彩色编码为默认方式。

hdec={1,2,4}

水平方向抽取采样因数1、2或4。

vdec={1,2,4}

垂直方向抽取采样因数1、2或4。

quality=1-20

设置JPEG压缩的质量[最好] 1 - 20 [非常差]。

fd|nofd

默认设置下，只有当Zoran采集卡的硬件支持MJPEG图像放大至其原有大小时才进行抽取采样。选项fd命令滤镜总是进行所请求的抽取采样操作（效果很糟）。

screenshot

允许使用可以与按键绑定的被动模式下的命令以获取影片的截屏。参见被动模式的说明文档以及交互式控制一节以获取详细信息。命名为'shotNNNN.png'的文件将保存在工作目录下，所用的编号为第一个可以用的编号——不会覆盖任何文件。该滤镜在不用时不占用资源，并且支持任意色彩空间，所以将其添加进配置文件中很安全。

ass

将SSA/ASS字幕的渲染移至滤镜链中的任意一个位置。只适用于有-ass选项的时候。

示例：

-vf ass,screenshot

将SSA/ASS的渲染移至screenshot滤镜之前。这样做后的截屏内容将包含字幕。

blackframe[=amount:threshold]

侦测出（几乎）完全黑色的帧。可适用于侦测影片章节的切换或广告。输出行由所侦测出帧的编号、黑色程度的百分比、帧的类型和最近遇到的关键帧的编号组成。

<amount>

数值低于阈值的像素的百分比率（默认值：98）。

<threshold>

决定像素值低于多少就被认作是黑色的阈值（默认值：32）。

通用编码选项（仅用于MENCODER）

—audio-delay <any floating-point number>

通过在文件头部中设置延迟信息域来延迟音频或视频。（默认值：0.0）。该选项不在编码过程中延迟音频或视频流，但播放器会读出延迟信息域并做相应的补偿处理。正数值延迟音频，而负数值延迟视频。注意该选项正好与—delay选项相反。例如，如果一段视频在—delay 0.2下正确播放，那么你可以在MEncoder中使用 —audio-delay -0.2修复视频的不同步问题。

当前，该选项只能用于默认的流合并器（—of avi）。如果你使用了另外的流合并器，那么你必须使用—delay来代替该选项。

—audio-density <1–50>

每秒钟处理的音频数据块数量（默认值为2，用于0.5s长的音频数据块）。

注意：只用于CBR，VBR方式忽略该选项，因为它将每个数据包分别放入一个数据块中。

—audio-preload <0.0–2.0>

设置音频缓冲的时音间隔（默认值：0.5s）。

—fafmttag <format>

可用于改写输出文件中的音频格式标签。

示例：

—fafmttag 0x55

将使输出文件包含0x55（mp3）的信息作为音频格式标签。

—ffourcc <fourcc>

可用于改写输出文件中的视频fourcc值。

示例：

—ffourcc div3

将使输出文件包含‘div3’的信息作为视频fourcc值。

—force-avi-aspect <0.2–3.0>

代替存储在AVI OpenDML vprp头部中的宽高比值。该选项可用于在使用‘—ovc copy’时更改宽高比值。

—frameno-file <filename>（不推荐使用）

指定包含有帧序号对应关系表的音频文件的文件名，该对应关系表是在特殊的三阶段编码模式的第一阶段（只处理音频）中生成的。

注意：使用该模式很可能会让你得到A-V不同步的结果。不要使用它。它只是为了向后兼容而保留的，并且相当可能在以后的版本中去除。

—hr-edl- seek

在跳过处理区域时使用一种更精确但相对很慢的方式。不是定位标记为需跳过的处理区域，而是解码所有的帧并且只有编码需要的帧。该选项使从非关键帧的边界开始编码成为可能。

注意：不保证能与‘—ovc copy’一同使用。

—info <option1:option2:...>（只用于AVI）

指定生成的AVI文件的头部信息内容。

可用选项有：

help
显示本部分描述。
name=<value>
作品的标题
artist=<value>
作品的作者
genre=<value>
原作的类别
subject=<value>
作品的内容
copyright=<value>
版权信息
srcform=<value>
数字化资料的原始格式
comment=<value>
关于作品的总体性注释

—noautoexpand

不要自动在MEncoder滤镜链中插入扩展滤镜。适用于将字幕内嵌入影片时控制在滤镜链中的哪一个点上渲染字幕。

—noencodedups

不要尝试将重复的帧重复地编码；必须以输出零字节帧的方式表明有重复帧。除非加载了能进行重复帧编码的滤镜或编码器，否则无论如何将写入零字节帧。当前这样的滤镜只有hardup。

—noodml (仅用于—of avi的情况下)

对于大于1GB的AVI文件不写入OpenDML索引。

—noskip

不要跳过任何帧。

—o <filename>

输出至所给的文件名。

如果你需要有一个默认的输出文件名，你可以将此选项放在MEncoder的配置文件中。

—oac <codec name>

使用所给的音频编解码器编码（没有默认值）。

注意：使用—oac help来获取可用音频编解码器的列表。

示例：

—oac copy
不编码，直接复制音频流
—oac pcm
编码为未压缩的PCM格式。
—oac mp3lame
编码为MP3格式（使用LAME）。
—oac lavc
使用libavcodec编解码器编码。

—of <format> (BETA测试版代码！)

编码为指定的容器格式（默认值：AVI）。

注意：使用—of help以获取可用的容器格式的列表。

示例：

- `-of avi`
编码为AVI格式。
- `-of mpeg`
编码为MPEG格式（另参见`-mpegopts`）。
- `-of lavf`
使用libavformat流合并器编码（另参见`-lavfopts`）。
- `-of rawvideo`
原始视频流（不合并——只含一个视频流）
- `-of rawaudio`
原始音频流（不合并——只含一个音频流）
- `-ofps <fps>`
为输出文件指定一个每秒帧数（fps）值，该值可以与原来视频中的不同。必须为可变fps（ASF、一些MOV）以及逐行扫描（30000/1001 fps电视 图像模式的MPEG）文件设置该值。
- `-ovc <codec name>`
使用给出的视频编解码器编码（没有默认值）。
注意：使用`-ovc help`来获取可用视频编解码器的列表。

示例：
 - `-ovc copy`
不编码，直接复制视频流
 - `-ovc raw`
编码为任意的未压缩格式（使用`'-vf format'`选择具体格式）。
 - `-ovc lavc`
使用libavcodec编解码器编码。
- `-passlogfile <filename>`
在二阶段编码模式中将第一阶段的信息导出至<filename>，而不是默认的divx2pass.log。
- `-skiplimit <value>`
指定编码了一帧之后最多可以跳过的帧的数量（`-noskiplimit`表示无限 制）。
- `-vobsubout <basename>`
指定输出的.idx和.sub文件的主文件名。该选项使字幕不在编码后影片得到渲染，而是将其转而输出至VOBsub的字幕文件。
- `-vobsuboutid <langid>`
为字幕指定双字母的语言代码。该选项改写了从DVD或.ifo文件中读取的语言代码。
- `-vobsuboutindex <index>`
指定输出文件中字幕的索引号（默认值：0）。

特定编解码器专用的编码选项（仅用于MENCODER）

你可以使用以下的句法指定特定编解码器专用的参数：

`-<codec>opts <option1[=value],option2,...>`

这里<codec>可以是：lavc、xvidenc、lame、toolame、twolame、nuv、xvfw、faac、x264enc、mpeg、lavf。

`lame (-lameopts)`

`help`

获取帮助信息

`vbr=<0-4>`

可变比特率方式

	0	cbr	
	1	mt	
	2	rh (默认值)	
	3	abr	
	4	mtrh	
abr			
		平均比特率	
cbr			
		恒定比特率 同时强制在后面指定的ABR预设模式中使用CBR模式。	
br=<0-1024>			
		以kbps计量的比特率 (仅用于CBR和ABR)	
q=<0-9>			
		质量 (0 - 最好, 9 - 最差) (仅用于VBR)	
aq=<0-9>			
		算法质量 (0 - 最好/最慢, 9 - 最差/最快)	
ratio=<1-100>			
		压缩比	
vol=<0-10>			
		音频输入的增益	
mode=<0-3>			
		(默认方式: 自动判断)	
			0
		立体声	
			1
		共享式立体声	
			2
		双声道	
			3
		单声道	
padding=<0-2>			
			0
		无填充	
			1
		所情况下均填充	
	2	调整	
fast			
		打开后面所指定的VBR预设模式中的较快编码方式。 这将造成质量稍差而比特率稍高。	
highpassfreq=<freq>			
		设置高通滤波的频率值, 单位为Hz。 低于所指定值的频率将被滤除。 值-1将禁用过	
		滤, 值0将让LAME自动选择一个值。	
lowpassfreq=<freq>			
		设置低通过滤的频率值, 单位为Hz。 高于所指定值的频率将被滤除。 值-1将禁用过	
		滤, 值0将让LAME自动选择一个值。	
preset=<value>			
		预设值	
		help	
		打印预设设置的其它选项与信息。	
		medium	
		VBR编码, 质量不错, 比特率范围为150-180 kbps	

standard

VBR编码，高质量，比特率范围为170–210 kbps

extreme

VBR编码，超高质量，比特率范围为200–240 kbps

insane

CBR编码，最高的预设质量，比较率为320bps

<8–320>

ABR编码，平均比特率为所给的kbps值

示例：

fast:preset=standard

适用大多数人以大多数音乐类型，质量已经相当高了

cbr:preset=192

使用192kbps的ABR预设模式，强制使用固定比特率编码。

preset=172

使用ABR预设模式、172kbps的平均比特率编码。

preset=extreme

用于那些有极好听力和极好设备的人

toolame和twolame（分别对应 -toolameopts 和 -twolameopts）

br=<32–384>

在CBR模式下该参数指明比特率值，单位为kbps，而在VBR模式下则是每个音频帧所允许的最 小比特率。VBR模式不能在该值低于112时工作。

vbr=<-50–50>（仅用于VBR）

比特率可变范围；如果为负值，则编码器使平均比特率向所限制最低值靠拢，如果为正值，则向最高值靠拢。设为0时使用CBR模式（默认值）。

maxvbr=<32–384>（仅用于VBR）

每个音频帧所允许的最大比特率，单位为kbps

mode=<stereo | jstereo | mono | dual>

（默认值：单声道音频使用mono，否则为stereo）

psy=<-1–4>

心理声学模型（默认值：2）

errprot=<0 | 1>

使用错误保护功能。

debug=<0–10>

调试等级

faac（-faacopts）

br=<bitrate>

平均比特率，单位为kbps（与quality选项不能同时使用）

quality=<1–1000>

质量模式，值越高效果越好（与br选项不能同时使用）

object=<1–4>

目标类型的复杂度

MAIN（默认值）

2 LOW

3 SSR

1

4

LTP（运行极慢）

mpeg=<2|4>

MPEG版本（默认值：4）

tns

启用瞬时噪声整形（TNS）功能。

cutoff=<0-采样频率/2>

截断频率（默认值：采样频率/2）

raw

将比特流保存为原始负载，而额外的数据保存在容器文件的头部中（默认值：0，与ADTS相一致）。如果没有明确的需要就不要设置这个标志位，否则你以后将不能重新将音频流合并进容器文件了。

lavc (-lavcopts)

有许多libavcodec（简称lavc）选项只是简要说明。要了解详情请阅读源代码。

示例：

 vcodec=msmpeg4:vbitrate=1800:vhq:keyint=250

acodec=<value>

 音频编解码器（默认值：mp2）

 mp2

 MPEG Layer 2

 mp3

 MPEG Layer 3

 ac3

 AC3

 adpcm_ima_wav

 IMA自适应PCM（每个采样占用4比特，4:1的压缩率）

 sonic

 试验性的有损/无损编解码器

abitrage=<value>

 音频比特率，单位为kbps（默认值：224）

atag=<value>

 使用所指定的Windows音频格式标签（例如atag=0x55）。

bit_exact

 只使用精确到比特的算法（除了I)DCT）。此外，精确到比特的方式禁用了许多的优化模式，因而应当只用于回归测试。这种测试要求即使编码器的版本变了，文件的二进制内容也要相同。该选项同时禁用MPEG-4流中的用户数据头。除非你确切知道你要做什么，否则不要使用这个选项。

threads=<1-8>

 最大能使用的线程数（默认值：1）。可能对运动预测有一定负面影响。

vcodec=<value>

 使用所指定的编解码器（默认值：mpeg4）。

 mjpeg

 动态JPEG

 ljpeg

 无损JPEG

 h261

 H.261

 h263

 H.263

 h263p

 H.263+

 mpeg4

	MPEG-4 (DivX 4/5)	
msmpeg4	DivX 3	
msmpeg4v2	微软MPEG4v2	
wmv1	Windows媒体视频，版本1（也就是WMV7）	
wmv2	Windows媒体视频，版本2（也就是WMV8）	
rv10	一个老旧的RealVideo编解码器	
mpeg1video	MPEG-1视频	
mpeg2video	MPEG-2视频	
huffyuv	HuffYUV	
ffvhuff	使用YV12格式的非标准压缩率提升20%的HuffYUV	
asv1	华硕视频v1	
asv2	华硕视频v2	
ffv1	FFmpeg的无损视频编解码器	
flv	用于Flash视频的Sorenson H.263	
dvvideo	Sony数字视频	
svq1	苹果Sorenson视频1	
snow (另参见：vstrict)	FFmpeg实验性的基于小波变换的编解码器	
vqmin=<1-31>	最低量化参数（第1/2阶段编码）	1
	不推荐（文件大很多，质量差别不大，而且有奇怪的副作用：msmpeg4、h263的质量会很差，码率控制会混乱从而质量下降，而且一些解码器将不能解码）。	2
	推荐用于普通mpeg4/ mpeg1video编码（默认值）。	3
	推荐用于h263(p)/ msmpeg4。建议采用3而不是2的原因是2可能造成溢出。（这在h263(p)中会在将来通过改变每个宏块的量化参数而解决，而msmpeg4中无法修复，因为它不支持那种方式。）	
lmin=<0.01-255.0>	用于码率控制的最小帧级别的拉格朗日乘子（默认值：2.0）。Lancz很少会使用低于vqmin值的量化参数。降低lmin会使lancz更容易在处理某些帧选用更低的量化参数，但不会低于vqmin值。类似的，提高lmin将使lancz更不可能选用低的量化参数，即使vqmin参数允许使用这些量化参数。你很可能想将lmin设为大致与vqmin相等。当使用自适应量	

化模式时，改变lmin/lmax可能影响不大；参见mblmin/mblmax。

lmax=<0.01-255.0>
用于码率控制的最大拉格朗日乘子（默认值：31.0）

mblmin=<0.01-255.0>
用于码率控制的最小宏块级别的拉格朗日乘子（默认值：2.0）该参数影响诸如qprd、lumi_mask等自适应量化选项。

mblmax=<0.01-255.0>
用于码率控制的最大宏块级别的拉格朗日乘子（默认值：31.0）

vqscale=<0-31>
恒量化参数/ 恒定质量编码（选用固定量化参数模式）。值越小意味着质量越好，但文件越大（默认值：-1）。在使用snow编解码器的情况下，值0表示无损编码。由于其它编解码器不支持这个值，vqscale=0会有未经定义的效果。不推荐用1（参见vqmin获取详情）。

vqmax=<1-31>
最大量化参数（第1/2阶段编码），10-31应该是一个合理的范围（默认值：31）。

mbqmin=<1-31>
过时参数，应使用vqmin

mbqmax=<1-31>
过时参数，应使用vqmax

vqdiff=<1-31>
连续的I或P帧之间最大的量化参数差别（第1/2阶段编码）（默认值：3）

vmax_b_frames=<0-4>
非B帧之间存在的最大B帧数目：

0	无B帧（默认值）	0
0-2	对于MPEG-4来说是合理的范围	

vme=<0-5>
运动预测方式。可用方式有：

0	none（质量很低）
1	full（运行慢，当前代码未维护且已禁用）
2	log（质量低，当前代码未维护且已禁用）
3	phods（质量低，当前代码未维护且已禁用）
4	EPZS：size=1的菱形模板，size可以通过*dia选项调整（默认值）
5	X1（实验性选项，当前是EPZS的一个别名）
8	iter（递归重叠的区块，仅用于snow）

注意：目前0-3无视所花费比特的多少，所以质量可能比较低。

me_range=<0-9999>
运动预测的搜索范围（默认值：0（不限制））

mbd=<0-2>（另参见*cmp、qpel）
宏块决策算法（高质量模式），将每个宏块在所有的模式下编码然后选择最佳的模式。该模式运行很慢，但能使质量提高，文件大小降低。当mbd设为1或2时，比较宏块过程

中将不考虑mbcmp的值。然而当某个比较设置（precmp、subcmp、cmp或mbcmp）为非零值时，将会使用一种更慢但更 好的半像素运动搜索模式，而不管mbd设为何值。如果设置了qpel参数，则将肯定使用四分之一像素运动搜索模式。

	使用mbcmp提供的比较函数（默认方式）。	0
	选用需要比特最少的宏块模式（=vhq）。	1
	选用码率失真最优的宏块模式。	2
vhq	与mbd=1相同，出于兼容性考虑而保留了下来。	
v4mv	允许每个宏块有4个运动矢量（质量较好些）。如果在mbd>0时使用效果更好。	
obmc	重叠块的运动补偿（H.263+）	
loop	循环过滤（H.263+） 注意，该选项有问题	
inter_threshold <-1000-1000>	当前没有任何作用。	
keyint=<0-300>	关键帧的最大间隔，单位为帧数（默认值：250或者说25fps的影片中每10秒一个关键帧。这个数值是MPEG-4中推荐使用的默认值）。大多数编解码器需要定期使用关键帧以限制不匹配错误聚积过多。关键帧也在播放定位时需要，因为只能定位于关键帧——但关键帧比其它帧需要更多空间，所以这里的数值设得越大意味着文件越小，但定位越不精确。0与1相同，这将使每个帧都成为关键帧。不推荐大于300的值，因为质量可能很差，这取决于解码器、编码器和运气。通常MPEG-1/2使用小于等于30的值。	
sc_threshold=<-1000000000-1000000000>	用于场景切换侦测的阈值。当libavcodec侦测到场景切换时，它将插入一个关键帧。你可以通过这个选项指定侦测模式的敏感度。-1000000000表示每个帧下都会侦测出一个场景切换，1000000000表示侦测不出任何场景切换（默认值：0）。	
sc_factor=<any positive integer>	导致使用较高量化参数的帧更容易引发场景切换的侦测，从而使libavcodec使用一个帧（默认值：1）。1-16是一个合理的范围。2至6之间的值能提高峰值信噪比（PSNR）（最高大约为0.04dB）并且让帧在高动态场景中 放置于更合适的位置 高于6的值可能提高一点点PSNR（大约比sc_factor=6多0.01dB），而视觉质量下降却相当可观。	
vb_strategy=<0-2>（只用于第一阶段编码）	选用I/P/B帧的策略：	
	总是使用最多的B帧（默认值）。	0
	在高动态场景中避免使用B帧。参观b_sensitivity选项以调试这一策略。	1
	调整使用B帧的个数以使质量最好（运行缓慢）。你可以要通过调整brd_scale选项以减少该选项对于运行速度的负面作用。	2
b_sensitivity=<任何大于0的整数>	调整vb_strategy=1时运动侦测的敏感程度从而避免使用B帧（默认值：40）。敏感程度越低将导致越多的B帧。使用更多的B帧通常能提升峰值信噪比，但太多的B帧可能导致在高动态场景中质量下降。除非运动程度极其高，否则b_sensitivity设为低于默认的值	

是很安全的；10在大多数情况 下是一个理想的值。

`brd_scale=<0-10>`

减小动态B帧决策中帧的尺寸（默认值：0）。 `brd_scale`每增加一，帧的宽高将减小一半，这样就使运行速度提升至四倍。 尺寸减小后帧的宽和高都必须是偶数，所以 `brd_scale=1`要求原来的宽高尺寸是四的倍数， `brd_scale=2`要求原来的宽高尺寸是八的倍数，依次类推。 换句话说，原来帧的宽高尺寸都必须能被 $2^{(brd_scale+1)}$ 整除。

`bidir_refine=<0-4>`

优化用于双向宏块的两个运动矢量，而不是重复使用向前和向后搜索所找到的矢量。 该选项只对B帧有效。

0

禁用（默认值）。

1-4

使用更广泛的搜索模式（值越大运行越慢）。

`vpass=<1-3>`

启用内置的二阶段（或更多阶段）编码模式，只有当你想使用二阶段（或更多阶段）编码模式时才指定这个值。

1

第一阶段编码（另参见turbo）

2

第二阶段编码

3

第N阶段编码（N阶段编码模式中第二阶段以及之后各阶段的编码）

以下展示了其如何工作，以及如何使用：

第一阶段编码（`vpass=1`）输出统计文件。 你可能希望关闭一些大量消耗CPU的选项，就像“turbo”模式所做的那样。

在二阶段编码模式中，第二阶段（`vpass=2`）读取统计文件的内容并依据该内容作出码率控制的决策。

在N阶段模式中，第二阶段（`vpass=3`——这不是本文档的印刷错误）同时做两件事：首先读取统计信息，然后重写这些信息。如果存在不得不中断MEncoder的可能，那么你可能希望在 做这一步前备份

你可以反复执行与此相同的编码阶段以优化编码。 每个紧接着的编码阶段将使用上一个阶段的统计信息并进一步优化。 最后一个编码阶段可以包含任意大量消耗CPU的编码选项。

如果你想用二阶段编码模式，那么先使用`vpass=1`，然后再用`vpass=2`。

如果你想用三阶段或更多阶段的编码模式，那么在第一阶段使用`vpass=1`，接着用`vpass=3`，接着不断用`vpass=3`直至你对编码结果满意了。

huffyuv：

第一阶段

保存统计信息。

第二阶段

使用基于第一阶段统计信息的最优哈夫曼表编码。

turbo（仅用于二阶段编码）

使用更快的算法并禁用大量占用CPU的选项以使第一阶段编码的运行大幅加快。 该选项可能稍稍降低全局峰值信噪比（大约为0.01dB）并且改变单个帧的类型以及稍稍改变该帧的峰值信噪比（最多至0.03dB）。

`aspect=<x/y>`

在文件内部保存影片的宽高比，就像MPEG文件一样。 比使用缩放功能效果好多了，因为质量不下降。 只有MPlayer能正确播放这些文件，其它播放器会以错误的宽高比显

示。宽高比参数可以以比率或浮点数字的形式给出。

示例：

aspect=16/9或aspect=1.78

autoaspect

与aspect选项相同，但是通过综合考量滤镜链上的各种调整（crop/ expand/scale/等等）自动计算宽高比。不会对性能产生负面影响，所以你可以放心地让该选项保持为打开的状态。

vbitrate=<value>

指定比特率（第1/2阶段编码）（默认值：800）。

警告：1kbit = 1000 bits

4–16000

（单位为kbit）

16001–24000000

（单位为bit）

vratetol=<value>

文件大小控制中允许出入的数量，单位为kbit。1000–100000是一个合理的范围。

（警告：1kbit = 1000 bits）（默认值：8000）

注意：vratetol在第二阶段编码中不应设得太大，否则当设置了vrc_(min|max)rate时，可能会产生问题。

vrc_maxrate=<value>

最大比特率，单位为kbit/ sec（第1/2阶段编码）（默认值：0，无限制）

vrc_minrate=<value>

最小比特率，单位为kbit/ sec（第1/2阶段编码）（默认值：0，无限制）

vrc_buf_size=<value>

缓冲大小，单位为kbit（第1/2阶段编码）。在MPEG-1/2中该选项同时设置了vbv缓冲的大小，对于VCD请使用327，对于SVCD使用917，而DVD使用1835。

vrc_buf_aggressivity

当前没有什么作用

vrc_strategy

码率控制方式。注意当vrc_strategy不设为0时，一些影响码率控制的选项将不起任何作用。

0

使用内置的lavc码率控制方式（默认值）。

1

使用Xvid的码率控制方式（仅供测试用；要求MEncoder编译时加入对Xvid 1.1或更高版本的支持）。

vb_qfactor=<- 31.0–31.0>

B帧和非B帧间的量化参数因数（第1/2阶段编码）（默认值1.25）

vi_qfactor=<- 31.0–31.0>

I帧和非I帧间的量化参数因数（第1/2阶段编码）（默认值0.8）

vb_qoffset=<- 31.0–31.0>

B帧和非B帧间的量化参数偏移量（第1/2阶段编码）（默认值1.25）

vi_qoffset=<- 31.0–31.0>

（第1/2阶段编码）（默认值0.0）

如果v{b|i}_qfactor > 0

I/B帧量化参数 = P帧量化参数 * v{b|i}_qfactor + v{b|i}_qoffset

否则

使用常规的码率控制方式（不限于于下一个P帧的量化参数）而设置 q= -q *

v{b|i}_qfactor + v{b|i}_qoffset

提示：要执行恒定量化参数的编码而对I/P和B帧应用不同的量化参数，你可以使用：
 $lmin = \langle I/P \text{帧量化参数} \rangle : lmax = \langle I/P \text{帧量化参数} \rangle : vb_qfactor = \langle B \text{帧量化参数} / I/P \text{帧量化参数} \rangle$ 。

$vqblur = \langle 0.0 - 1.0 \rangle$ （第一阶段编码）

量化参数模糊系数（默认值：0.5），值越大将使量化参数在时间上分布得越平均（变化得越慢）。

0.0

禁用量化参数模糊功能。

1.0

使量化参数在所有先前出现的帧中平均化。

$vqblur = \langle 0.0 - 99.0 \rangle$ （第二阶段编码）

量化参数高斯模糊（默认值：0.5），值越大将使量化参数在时间上分布得越平均（变化得越慢）。

$vqcomp = \langle 0.0 - 1.0 \rangle$

量化参数压缩， vrc_eq 选项依赖于该选项（第1/2阶段编码）（默认值：0.5）。例如，假设使用了默认的码率控制方程式，如果 $vqcomp = 1.0$ ，那么码率控制过程中将分配给每个帧足够的比特数，以使它们在同一个量化参数下编码。如果 $vqcomp = 0.0$ ，那么码率控制中将分配给每个帧相同的比特数，也就是严格的CBR模式。注意：以上都是极端性的设置，应当从不使用。在这两个极端值之间的某个值能使感观质量达到最优。

$vrc_eq = \langle \text{equation} \rangle$

主码率控制方程式（第1/2阶段编码）

1

恒定比特率

tex

恒定质量

$1 + (\text{tex} / \text{avgTex} - 1) * qComp$

近似于以前码率控制代码所用的方程式

tex^{qComp}

式中 $qcomp$ 为0.5或某个类似设置（默认值）

中置运算符：

+, -, *, /, ^

变量：

tex

纹理复杂度

iTex, pTex

内部、非内部纹理复杂度

avgTex

平均纹理复杂度

avgITex

I帧中的平均内部纹理复杂度

avgPITex

P帧中的平均内部纹理复杂度

avgPPTex

P帧中的平均非内部纹理复杂度

avgBPTex

B帧中的平均非内部纹理复杂度

mv

用于运动矢量的比特数

fCode

	以log2对数值计量的最大运动矢量长度	
iCount	内部宏块数/宏块数	
var	空间复杂度	
mcVar	时间复杂度	
qComp	命令行输入的qcomp值	
isl, isP, isB	若图像类型为I/P/B则为1，否则为0。	
Pi,E	参见你最喜欢的数学书。	
函数：		
max(a,b),min(a,b)	最大值/最小值	
gt(a,b)	若a>b则为1，否则为0	
lt(a,b)	若a<b则为1，否则为0	
eq(a,b)	若a==b则为1，否则为0	
sin, cos, tan, sinh, cosh, tanh, exp, log, abs		
vrc_override=<options>	用户指定的针于影片特定部分（尾声、演职员表……）的质量值（第1/2阶段编码）。 选项有<start-frame>、<end-frame>、<quality>[/<start-frame>, <end-frame>, <quality>[/...]]：	
	quality (2–31)	
	量化参数	
	quality (-500–0)	
	质量修正的百分比程度	
vrc_init_cplx=<0–1000>	初始复杂度（第一阶段编码）	
vrc_init_occupancy=<0.0–1.0>	初始缓冲占用量，为vrc_buf_size值的一个分数（默认值：0.9）	
vqsquish=<0 1>	指定如何使用量化参数保持在qmin和qmax之间（第1/2阶段编码）。	0
	使用截断方式。	1
	使用一种不错的微分函数（默认值）。	
vlelim=<-1000–1000>	设置亮度的单一系数消减阈值。 设为负数值则将同时考量直流系数（在以quant=1编码时应当至少设为-4或更低）：	0
	禁用（默认值）	-4
	JVT推荐值	
vcelim=<-1000–1000>		

设置色度的单一系数消减阈值。 设为负数值则将同时考量直流系数（在以quant=1编码时应当至少设为-4或更低）：

	禁用（默认值）	0
	JVT推荐值	7
vstrict=<-2 -1 0 1>	严格符合编码标准	
	0 禁用	1
		只有当你想让MPEG-4标准解码器读取编码输出结果时才推荐使用
		-1
	允许使用libavcodec专用的扩展功能（默认值）。	-2
vdpart	启用实验性的编解码器和功能，这些可能在将来版本的MPlayer中无法播放（用于snow）。	
	数据分隔。 每个视频数据包增加2字节内容，从而改善在不可靠信道上传输时（例如，在因特网上传输 流媒体）的容错能力。 每个视频数据包将编码为3个独立的部分：	
	1. MVs	
	运动信息	
	2. DC coefficients（直流系数）	
	低分辨率画面	
	3. AC coefficients（交流系数）	
	细节	
	MV和DC最为重要，丢失这两个部分似乎远比丢失AC部分和1. & 2.部分严重得多。（MV & DC）远比3.部分（AC）小得多，这意味着传输错误更多出现在AC部分而不是MV & DC部分。 所以，画面在数据分割后将比未分割时看上去更好，因为未分割时传输错误将同等地损坏 AC/DC/ MV三个部分。	
vpsize=<0-10000>	（另参见vbpart）	
	视频数据包的大小，用于改善容错能力。	
	0	
	禁用（默认值）	
	100-1000	
	较好的选择	
ss		
	用于H.263+的分片结构模式	
gray		
	仅灰度的编码（较快）	
vfdct=<0-10>		

DCT算法

自动选用一个良好的算法（默认值）。

快速整数式

精确整数式

- | | |
|---|---------|
| 3 | MMX |
| 4 | mlib |
| 5 | Altivec |
| 6 | 浮点AAN |

idct=<0-99>

IDCT算法

注意：据我们所知，所有这些IDCT算法均通过IEEE1180测试。

自动选用一个良好的算法（默认值）。

JPEG标准整数式

简单式

- | | |
|----|--|
| 3 | 简单MMX |
| 4 | libmpeg2mmx（不精确，不要用于keyint >100模式下的编码） |
| 5 | ps2 |
| 6 | mllib |
| 7 | arm |
| 8 | Altivec |
| 9 | sh4 |
| 10 | simplearm |
| 11 | H.264 |
| 12 | VP3 |
| 13 | IPP |
| 14 | xvidmmx |
| 15 | CAVS |
| 16 | simplearmv5te |
| 17 | simplearmv6 |

lumi_mask=<0.0-1.0>

亮度屏蔽是一种‘心理感官性’的设置，一般认为其所利用的事实是人眼倾向于在画面中很亮的部分里只注意到较少的细节。亮度屏蔽模式比中性的模式更大幅度地压缩明亮的区域，所以它能省下不少比特将其用于其它的帧中，从而提高整体的主观感官质量，但同时可能降低了峰值信噪比。

警告：小心，值过大可能导致灾难性的后果。

警告：值设得大可能在某些显示器上看起来效果好，但可能在其它显示器上就显得难看。

0.0

禁用（默认值）

0.0-0.3

合理的范围

dark_mask=<0.0-1.0>

暗度屏蔽是一种‘心理感官性’的设置，一般认为其所利用的事实是人眼倾向于在画面中

很暗的部分里只注意到较少的细节。暗度屏蔽模式比中性的模式更大幅度地压缩明亮的区域，所以它能省下不少比特将其用于其它的帧中，从而提高整体的主观感官质量，但同时可能降低了峰值信噪比。

警告：小心，值过大可能导致灾难性的后果。

警告：值设得大可能在某些显示器上看起来效果好，但可能在其它显示器/电视/液晶屏上就显得难看。

0.0

禁用（默认值）

0.0–0.3

合理的范围

tcplx_mask=<0.0–1.0>

时间复杂度屏蔽（默认值：0.0（禁用））。设想一下一个场景中一只鸟飞过整个场景；tcplx_mask将提高描绘鸟的宏块的量化参数（从而降低这些宏块的质量），而人眼通常没有时间去看清鸟的所有细节。注意如果被屏蔽的对象停止运动（例如鸟儿落地了），那么很有可能在短时间内这个对象会显得很难看，直至编码器注意到这个对象不动了而需要更好的块来描绘。节省下来的比特将用于视频中的其它部分，只要tcplx_mask经过精心选取，这将提高主观感官质量。

scplx_mask=<0.0–1.0>

空间复杂度屏蔽 如果没有解块效应滤镜用于解码过程，那么较大的值有助于消除块效应，但这可能不是一个好办法。

设想一下一个含有草丛（这通常有很大的空间复杂度）、蓝天和一间房子的场景；

scplx_mask将提高描绘草丛的宏块的量化参数，从而降低其质量，由此将更多的比特花在天 空和房子上。

提示：要充分切除所有的影片黑边，因为它们使宏块的质量下降（这同样适用于不用scplx_mask的情形）。

0.0

禁用（默认值）

0.0–0.5

合理的范围

注意：这个设置与使用能更大幅度压缩高频信号的自定义量化矩阵不具有相同效果，因为即使直流信号改变时scplx_mask也不会降低P块的质量。scplx_mask的效果可能也不如那种方式好。

p_mask=<0.0–1.0>（另参见vi_qfactor）

降低帧间块的质量。这等价于提高内部块的质量，因为码率控制器仍在整个视频序列上以相同的平均码率分配了数据资源（默认值：0.0（禁用））。p_mask=1.0使分配到每个内部块的比特数加倍。

border_mask=<0.0–1.0>

用于MPEG风格编码器的画面边缘处理模式。画面边缘处理模式提高距离画面帧边缘不到1/5个帧宽度/高度的宏块的量化参数，因为这些宏块通常对于显示效果较为不重要。

naq

使自适应量化模式正态化（实验性选项）。当使用自适应量化模式（*_mask）时，平均每个宏块的量化参数大小可能不再与所需要的帧级别的量化参数相一致。Naq将尝试调整每个宏块的量化参数以保证平均值为适当的值。

ildct

使用隔行扫描的DCT。

ilme

使用隔行扫描的运动估计模式（与qpel不能同时使用）。

alt

使用另一种扫描表。		
top=<- 1 -1>		-1
自动判断		0
下半扫描场先进入		1
上半扫描场先进入		
format=<value>		
YV12	默认值	
444P	用于ffv1	
422P	用于HuffYUV、无损JPEG、dv和ffv1	
411P	用于无损JPEG、dv和ffv1	
YVU9	用于无损JPEG、ffv1和svq1	
BGR32	用于无损JPEG和ffv1	
pred	(用于HuffYUV)	
左预测	1	平面/
梯度预测		2
中位数预测		
pred	(用于无损JPEG)	
左预测		0
上预测		1
左上预测		2
梯度预测	3	平面/
平均数预测		6
coder	(用于ffv1)	
vlc编码 (Golomb-Rice)		0
算术编码 (CABAC)		1
context	(用于ffv1)	

	小型上下文环境模式	0
	大型上下文环境模式 (用于ffvhuff)	1
	预先确定的哈夫曼表 (内置的或是二阶编码中产生的)	0
	自适应哈夫曼表	1
qpel	使用四分之一像素的运动补偿模式 (与ilme不能同时使用)。 提示: 该选项似乎只对高比特率的编码有用。	
mbcmp=<0-2000>	设置用于宏块决策的比较函数, 只有当mbd=0时才有作用。	
	0 (SAD)	绝对值差的合, 运行快 (默认值)
	1 (SSE)	误差的平方合
	2 (SATD)	Hadamard变换后绝对值差的合
	3 (DCT)	DCT变换后绝对值差的合
	4 (PSNR)	量化误差的平方合 (避免使用, 质量差)
	5 (BIT)	块中所需的比特的数量
	6 (RD)	码率失真最优, 运行慢
	7 (ZERO)	0
	8 (VSAD)	垂直方向绝对值差的合
	9 (VSSE)	垂直方向绝对值差的平方合
	10 (NSSE)	保留噪声的差值的平方合
	11 (W53)	5/3小波, 只用于snow中
	12 (W97)	9/7小波, 只用于snow中
	+256	同时使用色度信息, 当前无法在B帧中 (正确) 运行。
ildctcmp=<0-2000>	设置隔行扫描DCT决策中使用的比较函数 (参见mbcmp以获取可用比较函数信息)。	
precmp=<0-2000>	设置运动估计预处理阶段所使用的比较函数 (参见mbcmp以获取可用比较函数信息) (默认值: 0)。	
cmp=<0-2000>	设置整像素运动估计中所使用的比较函数 (参见mbcmp以获取可用比较函数信息) (默	

认值：0)。	
subcmp=<0-2000>	设置亚像素运动估计中所使用的比较函数（参见mbcmp以获取可用比较函数信息）（默认值：0）。
skipcmp=<0-2000>	FIXME：为这个选项写说明文档。
nssew=<0-1000000>	该设置控制NSSE的权重，其中较大的权重将导致较多的噪声。0 NSSE与SSE是一样的如果你希望在你编码后的视频中保留一些噪声，而不是在编码时将他们滤除，那么你会发现这个设置很有用（默认值：8）。
predia=<-99-6>	运动估计预处理阶段中使用菱形的类型和大小
dia=<-99-6>	运动估计中使用的菱形的类型和大小。运动搜索是一个迭代式的过程。使用一个较小的菱形并不会限制搜索中只找较小的运动矢量。而是说在找到最合适的运动矢量前搜索更倾向于停止，尤其是在噪声介入的情形下。较大的菱形将允许对于最佳运动矢量的搜索更加广泛，从而搜索更加慢但得到的结果质量更加好。 大型的常规菱形比自适应形状的菱形质量更好。 自适应形状的菱形在速度和质量之间有较好的折中。 注意：常规菱形和自适应形状菱形在大小概念上的含义是不同的。
	-3
大小为3的自适应形状（快速）菱形	
	-2
大小为2的自适应形状（快速）菱形	
	-1
非均匀的多重六边形搜索（运行慢）	
	1
size=1的常规菱形（默认值）=EPZS类型的菱形	
0	
000	
0	
	2
size=2的常规菱形	
0	
000	
00000	
000	
0	
trell	使用格子因子搜索的量化模式。该选项将为每个8x8块找到最优的编码方式。使用格子因子搜索的量化方式就是一种在峰值信噪比对比码率意义下最优的量化模式（假设IDCT不造成任何数值舍入错误，这种情形显然不在讨论之列。）。这种量化模式就是找一个具有最小错误数和lambda*bits值的块。
lambda	与量化参数（QP）相关的常数
bits	编码这个块所需的比特数量。
error	量化错误数的平方和

cbp	码率失真最优的编码后块的特征。 将选取使削减值与 $\lambda \cdot \text{rate}$ 值的合最小化的编码后块的特征。 该选项只能与格子因子量化模式一起使用。	
mv0	尝行在运动矢量= $\langle 0,0 \rangle$ 的条件下编码每个宏块并选取最好的一个。 当mbd=0时该选项没有作用。	
mv0_threshold=<任意非负整数>	当周围的运动矢量为 $\langle 0,0 \rangle$ ，而当前块的运动估计评分小于mv0_threshold时，将使用 $\langle 0,0 \rangle$ 作为运动矢量，而跳过进一步的运动估计步骤（默认值：256）。 将mv0_threshold降低至0可以使峰值信噪比有稍许（0.01dB）增加，并且可能使编码后的视频看上去稍微好一些；将mv0_threshold设为高于320时将导致峰值信噪比和视觉质量大大降 减。 较高的值使编码加快一点点（通常低于1%，这取决于所使用的其它选项）。	
	注意：该选项不要求启用mv0选项。	
qprd（仅用于mbd=2）	对于每个宏块中给定 λ 值的码率失真最优的量化参数（QP）	
last_pred=<0-99>	来自前一帧的运动预测器的数量	0
	（默认值）	a
	将使用 $2a+1 \times 2a+1$ 宏块数平方个来自前一帧的运动矢量预测器。	
preme=<0-2>	运动估计的预处理阶段	1
	0 禁用	
	总是使用	只在帧后使用（默认值）
subq=<1-8>	亚像素优化质量（用于qpel）（默认值：8（高质量））	2
	注意：该选项对速度有极大影响。	
refs=<1-8>	运动补偿中所考虑的参照帧的数量（仅用于Snow）（默认值：1）	
psnr	编码后打印整个视频的PSNR（峰值信噪比），并将每个帧的PSNR存放在一个名字诸如'psnr_hhmmss.log'的文件中。 返回的值单位为dB（分贝），值越高越好。	
mpeg_quant	使用MPEG的量化器而不是H.263的。	
aic	启用MPEG-4的AC预测模式，或是H.263+的高级帧内预测模式。 该选项将很有限地提升质量（大约0.02dB的PSNR），同时很有限地减慢编码速度（大约为 1%）。	
	注意：vqmin应当设为8或更大以使用H.263+ AIC。	

aiv
用于H.263+的另一种帧间vlc模式

umv
无限制的运动矢量 (MV) (仅用于H.263+) 允许编码任意长的MV。

ibias=<-256-256>
帧内量化器偏向系数 (256等价于1.0, MPEG类型的量化器的默认值: 96, H.263类型的量化器的默认值: 0)
注意: H.263 MMX量化器无法处理正值的偏向系数 (应设置vfdct=1或2), MPEG MMX量化器无法处理负值的偏向系数 (应设置vfdct=1或2)。

pbias=<-256-256>
帧间量化器偏向系数 (256等价于1.0, MPEG类型的量化器的默认值: 0, H.263类型的量化器的默认值: -64)
注意: H.263 MMX量化器无法处理正值的偏向系数 (应设置vfdct=1或2), MPEG MMX量化器无法处理负值的偏向系数 (应设置vfdct=1或2)。
提示: 偏向系数的值越大 (-32 - -16而不是-64) 看上去越能提升PSNR。

nr=<0-100000>
噪声削减模式, 0表示禁用。0-600对于典型的视频内容来说是一个有用的范围, 但是你可能希望将其调高一点以用于噪点十分多的视频内容 (默认值: 0)。考虑到该选项对于编码速度影响小, 你可能想更倾向于使用这个选项, 而不是用诸如 denoise3d或 hqdn3d之类的视频滤镜来过滤噪声。

qns=<0-3>
量化器噪声整形模式。该选项不是选择在PSNR上最接近源视频的量化模式, 而是选择能使噪声 (通常是色斑) 会被图像中相近频率内容所屏蔽的量化模式。较大的值运行速度较慢, 但可能不能产生较好的编码质量。该选项可以并且应当与格子因子量化模式一起使用, 在这种情况下格子因子量化模式 (对于恒定权值是最优的) 将作为迭代搜索的起始操作。

0

禁用 (默认值)

1

仅降低系数的绝对值。

2

仅改变最后一个非零系数+1之前的系数。

3

尝试所有可能操作。

inter_matrix=<逗号分隔的矩阵>
使用自定义的帧间矩阵。该选项需要一个用逗号分隔的64个整数组成的字符串。

intra_matrix=<comma separated matrix>
使用自定义的帧内矩阵。该选项需要一个用逗号分隔的64个整数组成的字符串。

vqmod_amp
实验性的量化器调整模式

vqmod_freq
实验性的量化器调整模式

dc
以比特为单位的帧内DC精度 (默认值: 8)。如果你指定了vcodec=mpeg2video, 那么这个值可以是8、9、10或11。

cgop (另参见sc_threshold)
关闭所有的GOP。当前该选项只有当场景切换侦测功能禁用时才能工作 (sc_threshold=1000000000)。

vglobal=<0-3>

控制写入全局视频头部信息的方式。

0

编解码器决在哪里写入全局头部信息（默认值）。

1

只在extradata（额外数据）域中写入全局头部信息（.mp4/MOV/NUT需要这样做）。

2

只在关键帧之前写入全局头部信息。

3

结合1和2两种方式。

aglobal=<0-3>

与vglobal相同，用于音频头部信息。

level=<value>

设置编解码器环境等级。使用31或41以在Playstation 3上播放视频。

skip_exp=<0-1000000>

FIXME：为这个选项写说明文档。

skip_factor=<0-1000000>

FIXME：为这个选项写说明文档。

skip_threshold=<0-1000000>

FIXME：为这个选项写说明文档。

nuv (-nuvopts)

Nuppel视频基于RTJPEG和LZO。默认方式下，帧首先以RTJPEG方式编码，然后再用LZO压缩，但可以禁用这两个阶段的一个或两个。于是，你可以实际上输出的是原始的i420、LZO压缩的i420、RTJPEG或默认情况下的经LZO压缩的RTJPEG。

注意：nuvrec文件包含了一些关于用于大多数常见TV编码的设置的建议和示例。

c=<0-20>

色度阈值（默认值：1）

l=<0-20>

亮度阈值（默认值：1）

lzo

启用LZO压缩（默认值）。

no lzo

禁用LZO压缩。

q=<3-255>

质量等级（默认值：255）

raw

禁用RTJPEG编码。

rtjpeg

启用RTJPEG编码。（默认值）。

xvidenc (-xvidencopts)

有三个模式可以使用：恒定比特率模式（CBR）、固定量化器模式和二阶段编码模式。

pass=<1|2>

在二阶段编码模式中指定哪一个阶段。

turbo（仅用于二阶段编码模式）

通过使用更快的算法并禁用大量消耗CPU的选项，以加速第一阶段编码。该选项可能稍许降低全局的PSNR，并且改变单个帧的类型，以及稍许提升这个帧的PSNR。

bitrate=<value>（用于CBR或二阶段编码模式）

设置所使用的比特率。若值小于16000，则单位为千比特/秒；若值大于16000，则单位为bits/second。如果<value>为负数，那么Xvid将把这个值的绝对值作为视频的目标

大小（单位为 kBytes），并自动计算出相应的比特率（默认值：687kbts/s）。

`fixed_quant=<1-31>`

切换至固定量化器模式，并指定所使用的量化器。

`zones=<zone0>[/<zone1>[/...]]`（CBR或二阶段编码模式）

用户指定的针对影片特定部分（片尾、演职员表、……）的质量值。每个zone的格式是<start-frame>,<mode>,<value>，其中<mode>可以是

q

更改恒定量化器，在这种模式下value=<2.0-31.0>，代表量化参数值。

w

更改码率控制权值，在这种模式下value=<0.01-2.00>，代表单位为%的质量修正值。

示例：

`zones=90000,q,20`

将90000开始的所有帧以恒定量化参数20编码。

`zones=0,w,0.1/10001,w,1.0/90000,q,20`

将0-10000帧以10%的比特率编码，将90000直至结尾的帧以20的恒定量化参数编码。注意需要有第二个zone以限定第一个zone，因为没有它直至89999的所有帧都会以10%的比特率编码。

`me_quality=<0-6>`

该选项控制运动估计子系统。值越高，运动估计就越精确（默认值：6）。运动估计越精确，就能节省越多的比特。提高精度是以牺牲CPU时间为代价的，所以如果你需要实时编码，那么就降低这个设置。

(no)qpel

MPEG-4默认在其运动搜索中采用半像素的精度。标准中提出了一种允许编码器采用四分之一像素精度的模式。该选项通常导致图像较为锐化。不幸的是，该选项对于比特率有很大影响，并且有时候这种较高的比特率的占用阻碍其在固定比特率下给予图像较好质量。最好在启用和禁用这个选项的情况下测试一下，以看看是否值得激活该选项。

(no)gmc

启用全局运动补偿模式，该模式使Xvid生成特殊的帧（GMC帧），这种帧相当适合带有摇摄/推近/旋转等镜头的画面。使用该选项是否能节省比特是高度取决于源视频素材。

(no)trellis

格子因子量化模式是一种自适应的量化方式，该方式通过修改量化后的系数，以使这些系数能被熵编码器更大幅度地压缩，从而节省比特。其对质量的提升是不错的，如果VHQ对你来说占用了太多CPU，那么这个设置可能是相比VHQ而言以较少代价节省一些比特（并且提升固定比特率下画面质量）的一种替代方法（默认值：打开）。

(no)cartoon

如果你编码的帧序列是动画/卡通的话，那么就激活这个选项。这个选项修改Xvid内部的一些阈值，以使Xvid在针对外观平面化的卡通片的帧类型和运动矢量的决策中有较好判断。

(no)chroma_me

通常的运动估计算法仅使用亮度信息寻找最佳的运动矢量。然而对于一些视频素材，使用色度平面能有助于找到更好的矢量。该设置开启在运动估计中对于色度平面的使用（默认值：开启）。

(no)chroma_opt

启用色度优化器的一个预过滤器。这个预过滤器将对色彩信息做一些额外的特殊处理，以使图形边缘的阶梯性变化效果降至最低。这个预过滤器会以牺牲编码速度为代价以提升质量。这自然会降低PSNR，因为相对于原始画面的数学上的偏移将增大，但是主观感觉上图像的质量将得到提升。由于其依据色彩信息而工作，因而在以灰度模式编

码时你可能想把这个选项关掉。

(no)hq_ac

对于来自邻近块的内部帧，激活对于AC系数的高质量预测模式（默认值：开启）。

vhq=<0-4>

运动搜索算法基于一种对于通常色彩域的搜索，并且尝试找到一个能将参照帧和编码所得帧 的差别降至最小的运动矢量。 激活该选项时，Xvid将同时利用频率域（DCT）以搜索到一个既将空间上的差别降至最小， 又将块的编码长度降至最短的运动矢量。 从运行最快至运行最慢的设置有：

0

关闭

1

根据
模式
决定
(帧
间/
帧内
宏
块)
(默
认
值)
2

限制性搜索

3

中性搜索

4

宽泛性搜索

(no)lumi_mask

自适应的量化模式使宏块的量化器能在每个帧中有所不同。 这是一种‘心理感官性’的设置，一般认为它所利用的事实是人眼倾向于在画面中十分亮的 和十分暗的部分只注意到较少的细节。 相对于中性的区域，该选项对于这些区域压缩幅度更大，这将节省一些比特而将其用在其它 的帧中，从而提升整体的主观视觉质量但可能降低了PSNR。

(no)grayscale

使Xvid丢弃色度平面信息，从而使编码得到的视频只是灰度模式的。 注意该选项并不加快编码速度，而只是在编码的最后阶段阻止写入色度数据。

(no)interlacing

编码隔行扫描视频内容的扫描场。 打开该选项以用于隔行扫描内容。

注意： 如果你要改变视频的尺寸，那么你需要一个能识别隔行扫描内容的缩放器，你可以使用 `-vf scale=<width>:<height>:1`来激活这种缩放器。

min_iquant=<0-31>

最小I帧量化参数（默认值：2）

max_iquant=<0-31>

最大I帧量化参数（默认值：31）

min_pquant=<0-31>

最小P帧量化参数（默认值：2）

max_pquant=<0-31>

最大P帧量化参数（默认值：31）

min_bquant=<0-31>

最小B帧量化参数（默认值：2）

max_bquant=<0-31>

最大B帧量化参数（默认值：31）

min_key_interval=<value>（仅用于二阶段编码模式）

关键帧间的最小间隔（默认值：0）

max_key_interval=<value>

关键帧间的最大间隔（默认值：10*fps）

quant_type=<h263|mpeg>

设置所使用的量化器类型。对于高码率，你会发现MPEG量化模式保留了更多的细节。对于低码率，H.263的平滑特性会为了你带来较少的宏噪声。当使用自定义的量化矩阵时，必须使用MPEG量化模式。

quant_intra_matrix=<filename>

载入一个自定义的帧内量化矩阵文件。你可以使用xvid64conf的矩阵编码器以构建这样的文件。

quant_inter_matrix=<filename>

载入一个自定义的帧间量化矩阵文件。你可以使用xvid64conf的矩阵编码器以构建这样的文件。

keyframe_boost=<0-1000>（仅用于二阶段编码模式）

将一些比特从其它帧类型的空间储备中转移至内部帧中，从而提高关键帧的质量。该数量是增加的百分比量，所以值为10将给你的关键帧带来比通常情况下多10%的比特量（默认值：0）。

kfthreshold=<value>（仅用于二阶段编码模式）

与kfreduction一同使用。定义一个最小距离，低于这个距离你就认为两个帧应当视作是连续的，从而根据kfreduction的设置处理这个情况（默认值：10）。

kfreduction=<0-100>（仅用于二阶段编码模式）

以上两个设置可用于调整那些你认为过于接近（一个序列中）头个帧的关键帧的大小。kfthreshold设置了在哪个范围内的关键帧要被削减，而kfreduction决定了这些关键帧被削减比特率的量。最后一个帧将以通常的方式处理（默认值：30）。

max_bframes=<0-4>

放置于I/P帧间的B帧的最大数量（默认值：2）。

bquant_ratio=<0-1000>

B帧与非B帧间的量化参数比，150=1.50（默认值：150）

bquant_offset=<-1000-1000>

B帧与非B帧间的量化参数差值，100=1.00（默认值：100）

bf_threshold=<-255-255>

该选项让你指定使用B帧的优先级。这个值越高，B帧就越可能被使用（默认值：0）。不要忘了B帧通常量化参数比较高，从而过分产生B帧可能导致视觉质量变差。

(no)closed_gop

该选项告诉Xvid封闭每个GOP（Group Of Pictures——由两个帧分隔出来的画面组），这使GOP彼此之间相互独立。这就是意味着GOP的最后一个帧要么是P帧，要么是N帧，而不是B帧。通常开启这个选项是比较好的（默认值：开启）。

(no)packed

该选项意在解决编码至诸如AVI之类的不能处理乱序帧序列的容器格式时，所发生的帧顺序问题。实际应用中，大多数解码器（无论是软件的还是硬件的）都能够自己处理帧的顺序，因而当此选项开启时可能反而引起混乱，所以你可以放心地让这个选项保留为关闭状态，除非你确实知道你在做什么。

警告：该选项会产生非法的比特流，从而不能被DivX/ libavcodec/ Xvid以外的ISO-MPEG-4的解码器解码。

警告：该选项同时会在文件中存入一个虚假的DivX版本号，所以某些解码器中有问题的自动侦测系统可能会搞不清楚。

frame_drop_ratio=<0-100> (仅用于max_bframes=0时)

该设置控制可变帧率视频流的创建。该设置的值指定了一个阈值，如果后一个帧相对于前一个帧的差别低于或等于这个阈值，那么将跳过对于一个帧的编码（一个所谓的n-vop将放置于视频流中）。在播放时，当遇到一个n-vop时，将显示前一个帧。

警告：滥用这个设置可能导致视频跳跃，所以使用该选项后果自负！

rc_reaction_delay_factor=<value>

该参数控制CBR码率控制在对于比特率变后作出反应并且对其作出补偿，以使比特率在一个平均化范围内的帧中保持恒定前，所等待的帧的数量。

rc_averaging_period=<value>

真正的CBR是很难达到的。由于视频素材的不同，比特率可能是变化而难以预测的。所以Xvid采用了一种平均化周期，在这个周期中它保证比特数量为给定的一个值（减去一个较小的可变值）。这个设置所表达的是Xvid使“多少数量的帧”的比特率平均化从而达到CBR。

rc_buffer=<value>

码率控制缓冲的大小

curve_compression_high=<0-100>

该设置让Xvid从高比特率的场景中拿出一定百分比的比特，而将这些比特还给比特预留储备中。你也可以使用这个设置，如果你的一个视频有太多的比特分配给了高比特率的场景，以致于（较）低比特率的场景变得看上去很糟糕（默认值：0）。

curve_compression_low=<0-100>

该设置让Xvid将一定百分比的额外的比特分给低比特率的场景，而从整个视频剪辑中拿去一部分比特。如果你的一些低比特率场景仍然有马赛克，那么这个设置可能很好用（默认值：0）。

overflow_control_strength=<0-100>

在二阶段编码模式的第一阶段，计算出了一个经过缩放的比特率曲线。这个所期望曲线与编码中得到的曲线之间的差值称为溢出。显然，二阶段编码的码率控制器尝试弥补这个溢出量，将这个差值派分到后续帧中。该设置控制了每次有一个新帧时，有多少溢出量分配至其上。较低的值允许使用较迟缓的溢出控制，较大的码率喷发将较慢地得到补偿（可能导致小型视频剪辑中缺乏精度）。较大的值将使比特再分配过程中的变化更具突发性，如果你设得太高则可能太突然，以致产生损伤（默认值：5）。

注意：该设置对于质量极具影响，小心使用！

max_overflow_improvement=<0-100>

在帧所用比特的分配过程中，溢出控制可能增加帧的大小。该参数溢出控制所允许相对理想分配曲线所增加帧大小的最大的百分比（默认值：5）。

max_overflow_degradation=<0-100>

在帧所用比特的分配过程中，溢出控制可能减少帧的大小。该参数溢出控制所允许相对理想分配曲线所减小帧大小的最大的百分比（默认值：5）。

container_frame_overhead=<0...>

指定每帧的平均开销，单位为字节。大多数时候用户表达的是他们针对视频的目标比特率，而不关心视频容器文件的开销。这种较小的但（往往）恒定的开销可以导致目标文件大小超过预期值。Xvid让用户设置容器产生的平均每帧的开销数量（只给出每帧的平均值）。0具有特殊的含义，这个值让Xvid采用自己的默认值（默认值：24——AVI的平均开销）。

profile=<profile_name>

根据简单配置集（Simple Profile）、高级简单配置集（Advanced Simple Profile）以及DivX配置集（DivX Profile）限制选项的使用和VBV（短时间内的峰值比特率）。所产生的视频应该能够在符合这些配置集标准的独立播放器上播放。

unrestricted

没有限制（默认值）

sp0	第0级的简单配置集
sp1	第1级的简单配置集
sp2	第2级的简单配置集
sp3	第3级的简单配置集
asp0	第0级的高级简单配置集
asp1	第1级的高级简单配置集
asp2	第2级的高级简单配置集
asp3	第3级的高级简单配置集
asp4	第4级的高级简单配置集
asp5	第5级的高级简单配置集
dxnhandheld	DXN手持式设备配置集
dxnportntsc	DXN便携NTSC制式设备配置集
dxnportpal	DXN便携PAL制式设备配置集
dxnhtntsc	DXN家庭影院NTSC制式设备配置集
dxnhtpal	DXN家庭影院PAL制式设备配置集
dxnhdtv	DXN高清电视设备配置集

注意：这些配置集应当与适当的`-ffourcc`选项一起使用。通常DX50是可以采用的，因为某些播放器不能识别Xvid但大多数能识别DivX。

`par=<mode>`

指定像素宽高比 (Pixel Aspect Ratio) 模式 (不要与DAR——屏幕宽高比——混淆在一起)。PAR是单个像素的宽度和高度的比例。所以两者的关系如下： $DAR = PAR * (宽度/高度)$ 。

MPEG-4定义了5种像素宽高比以及一种泛用宽高比，这为指定一种专用的像素宽高比留下了余地。可以指定5种标准模式：

vga11	这是通常用于PC视频内容的PAR。像素是正方形单元。
pal43	PAL标准的4:3的PAR。像素是长方形的。
pal169	与前面同理
ntsc43	与前面同理
ntsc169	

与前面同理（不要忘了给出精确的比率）

ext

允许你使用par_width和par_height指定你自己的像素宽高比。

注意：通常，设置aspect和autoaspect选项就已经足够了。

par_width=<1-255>（仅用于par=ext时）

指定自定像素宽高比的宽度。

par_height=<1-255>（仅用于par=ext时）

指定自定像素宽高比的高度。

aspect=<x/y | f（浮点值）>

将影片的宽高比存放于文件内部，就像MPEG文件一样。与重新缩放的效果好很多，因为质量并不下降。MPlayer与其它一些播放器能正确播放这些文件，除此之外的播放器会以错误的宽高比显示这些文件。宽高比参数可以以比率或是浮点数的形式给出。

(no)autoaspect

与aspect选项相同，但是是自动计算宽高比，在此过程中考虑了滤镜链中所进行的所有调整（crop/expand/scale/等等）。

psnr

编码之后打印出整个视频的PSNR（峰值信噪比），并将逐帧计算的PSNR保存在当前目录中名字诸如'psnr_hhmmss.log'的一个文件里。返回值的单位是dB（分贝），值越高越好。

debug

将逐帧计算的统计信息保存在./xvid.dbg中。（这个文件不是二阶段编码模式中的码率控制文件。）

以下选项只存在于Xvid 1.1.x中。

bvhq=<0|1>

该设置允许通过使用一种针对码率失真优化的运算，来选取用于编码中使用的B帧的候选运动矢量，对于P帧这种方式是通过vhq选项实现的。该方式产生的B帧看上去较好，而同时也几乎不影响性能（默认值：1）。

以下选项只存在于1.2.x版的Xvid中。

threads=<0-n>

创建n个线程用以运行运动估计任务（默认值：0）。最多可以使用的线程的数量为画面的高度除以16所得的值。

x264enc（-x264encopts）

bitrate=<value>

设置所采用的平均比特率，单位为千比特/秒（默认值：关闭）。由于局部比特率会变化，因而这个平均值对于十分短的视频来说可以不精确（参见ratetol）。可以通过将此设置与vbv_maxrate一起使用来实现恒定的比特率，代价是质量严重下降。

qp=<0-51>

该选项选择的是用于P帧的量化器。I与B帧的话分别是该值加上ip_factor与pb_factor后得到的值。20-40是一个有用的范围。较低的值产生较好的精确，但导致比特率较高。0代表无损。注意H.264的量化模式与MPEG-1/2/4的工作方式不同：H.264的量化参数是基于对数尺度的。两者之间的映射关系大至是 $H264QP = 12 + 6 * \log_2(MPEGQP)$ 。例如，MPEG的QP=2与H.264的QP=18是等价的。

crf=<1.0-50.0>

启用恒定质量模式，并选择质量值。该质量值的尺度与QP的相类似。就像基于比特率的模式一样，该模式允许每个帧根据帧的复杂度使用不同的QP。

pass=<1-3>

启用2或3阶段编码模式。推荐总是以2或3阶段编码模式编码，因为该模式使比特的分配更佳，从而提升整体质量。

第一阶段

2

(二阶段编码模式中的) 第二阶段

3

第N阶段 (三阶段编码模式的第二和第三阶段)

以下介绍的是其如何工作，以及使用这个参数：

第一阶段 (pass=1) 收集视频上的统计信息，并将这些信息写入一个文件中。除了那些默认为打开的选项，你可能想关闭一些消耗CPU的选项。

在二阶段编码模式，第二阶段 (pass=2) 读取统计信息文件，并基于这个文件进行码率控制 决策。

在三阶段编码模式中，第二阶段 (pass=3——这是不打印错误) 同时做两件事：其首先读取 统计信息，然后重写这些统计信息。你可以使用所有的编码选项，除了那些十分消耗CPU的选项。

第三阶段 (pass=3) 与第二阶段相同，只是这一阶段基于第二阶段产生的统计信息工作。你可以使用所有的编码选项，包括消耗CPU的那些选项。

第一阶段可以使用平均比特率模式，或是使用恒量化器模式。推荐使用ABR，因为该模式不需求猜测所要使用的量化器。后续的阶段是ABR模式的，并且必须指定比特率。

turbo=<0-2>

快速第一阶段模式。在二阶段或更多阶段编码的第一阶段中，可以通过禁用一些对于最终阶段输出质量影响甚 小，甚至没有影响的选项，以提升运行速度。

0

禁用 (默认值)

1

减小subq、frameref的值，并且禁用一些宏块间划分分析的模式。

2

减小subq、frameref至1，使用菱形运动估计搜索，并且禁用所有划分分析模式。

第1级可以最多使第一阶段的速度加快至2倍，而与完全质量的第一阶段模式相比，对于最终阶段的全局PSNR没有任何改变。

第2级可以最多使第一阶段的速度加快至4倍，而与完全质量的第一阶段模式相比，对于最终阶段的全局PSNR有大约+/- 0.05dB的改变。

keyint=<value>

设置IDR帧间的最大间隔 (默认值：250)。较大的值节省比特，从而提高质量，代价是降低播放中的定位精度。与MPEG-1/2/4不同，H.264在keyint值很大是并不受DCT漂移效应影响。

keyint_min=<1-keyint/2>

设置IDR帧间的最小间隔 (默认值：25)。如果这个间隔中出现了场景切换，那么这个切换仍然编码为I帧，但不产生新的GOP。在H.264中，I帧并不一定约束出一个封闭的GOP，因为这里允许P帧在其前面的一帧之前预测 出来 (另参见frameref)。所以，I帧并不一定可用于播放定位。IDR帧限制其后续P帧，不让它们参照该IDR帧之前的帧。

scenecut=<-1-100>

控制插入额外帧行为的激进程度 (默认值：40)。当scenecut值小时，编解码器在即将超过keyint所规定值时总是强制使用I帧。scenecut值设置得好可能为I帧找到一个更好的位置。较大的数值导致使用多于所需要的帧，从而浪费了比特。-1表示禁用场景切换检测，这样I帧只有每过keyint个帧时才会插入一个，即使场景切换之前就已发生。这种方式不推荐使用，并且其浪费比特率，因为场景切换编码为P帧与编码为I帧几乎差不多大，但其并不重置'keyint计数器'。

frameref=<1-16>

B帧和P帧中的预测器里所使用的之前出现的帧的数量 (默认值：1)。该选项在动画是有效果的，但在实况视频素材中，大约6个参照帧之后参照帧的优化效果急剧下降。该

选项对于解码速度没有影响，但确实增加了解码所需的内存量。某些解码器最多只能处理15个参照帧。

bframes=<0-16>

帧和P帧之间连续出现的B帧的最大数量（默认值：0）

(no)b_adapt

自动决定何时使用B帧以及使用多少，数量上限为以上所指定的最大值（默认值：开启）。如果禁用了这个选项，那么将使用的B帧数为最大值。

b_bias=<-100-100>

控制b_adapt所做的决策。b_bias值最高产生的B帧越多（默认值：0）。

(no)b_pyramid

允许B帧作用预测其它帧的参照帧。例如，考虑3个连续的B帧：I0 B1 B2 B3 P4。不用这个选项的话，B帧的样式与MPEG-[124]中的一样。这样这些帧将以I0 P4 B1 B2 B3的次序编码，而所有的B帧都是从I0和P4中预测出来的。使用了这个选项后，这些帧将编码为I0 04 B2 B1 B3。B2与前面所述的一样，但B1是从I0和B2预测出来的，而B3是从B2和P4预测出来的。这种方式通常产生稍许好一些的压缩效果，而几乎没有运行速度上的开销。然而，这是一个实验性的选项：没有完全调整好并且可能不能总是起作用。要求bframes >= 2。缺点：将解码延迟量增加至2帧。

(no)deblock

使用反块效应滤镜（默认值：开启）。由于相对于其提高的质量而言，该选项占用极少时间，所以不推荐禁用这个选项。

deblock=<-6-6>,<-6-6>

第一个参数是AlphaC0（默认值：0）。该参数调整的是H.264内循环反块效应滤镜所用的阈值。第一，该参数调整的是滤镜对于任何一个像素所允许产生的变动的最大数量。第二，该参数影响的是将被滤除的边界两侧差别的阈值。正数值使更多的块效应损伤得到削减，但同时也会损伤画面细节。

第二个参数是Beta（默认值：0）。该参数影响的是画面细节的阈值。细节很多的块将不被过滤，因为滤镜产生的平滑效果会比原来的块效应效果更加明显。

滤镜的默认行为几乎总是带来最优的质量，所以最好要么不调这个选项，要么只对其作少量调整。然而，如果你的源视频素材已经有一些块效应或噪声效果你想要去除，那么可能将这个选项调高一点是个不错的办法。

(no)cabac

使用CABAC模式（Context-Adaptive Binary Arithmetic Coding，上下文自适应的二进制算术编码）（默认值：开启）。稍稍减慢编码和解码的速度，但应该可以节省10-15%的比特率。除非你要解码速度，否则你不该禁用这个选项。

qp_min=<1-51>（用于ABR或二阶段编码模式）

最小量化参数，10-30似乎是一个有用的范围（默认值：10）。

qp_max=<1-51>（用于ABR或二阶段编码模式）

最大量化参数（默认值：51）

qp_step=<1-50>（用于ABR或二阶段编码模式）

量化参数在帧之间增加/降低的最大数值（默认值：4）

ratetol=<0.1-100.0>（用于ABR或二阶段编码模式）

相对于平均比特率的所允许的变化程度（不针对特定编码单元）（默认值：1.0）

vbv_maxrate=<value>（用于ABR或二阶段编码模式）

局部最大的比特率，单位为千比特/秒（默认值：禁用）

vbv_bufsize=<value>（用于ABR或二阶段编码模式）

计算vbv_maxrate时所使用的平均化周期，单位为千比特（默认值：无，如果启用了vbv_maxrate那么必须指定这个选项）

vbv_init=<0.0-1.0>（用于ABR或二阶段编码模式）

初始缓冲占用量，为相对于vbv_bufsize值的一个分数（默认值：0.9）

ip_factor=<value>
I帧和P帧间的量化参数因数（默认值：1.4）

pb_factor=<value>
P帧和B帧间的量化参数因数（默认值：1.3）

qcomp=<0-1>（用于ABR或二阶段编码模式）
量化器压缩率（默认值：0.6）。值越小使比特率越恒定，而值越大使量化参数越恒定。

cplx_blur=<0-999>（仅用于二阶段编码模式）
估计出的帧复杂度的时间模糊度，应用于曲线压缩之前（默认值：20）。值越低则让量化参数值浮动得越多，值越高则使其变化得越平滑。cplx_blur保证每个I帧的质量与之后的P帧相当，并保证复杂度高低交替变化的帧（例如，低帧率的动画）不会因为量化参数的波动而浪费比特。

qblur=<0-99>（仅用于二阶段编码模式）
量化参数的时间模糊度，应用于曲线压缩之后（默认值：0.5）。值越低则允许量化参数浮动得越多，值越高则使其变化得越平滑。

zones=<zone0>[/<zone1>[/...]]
用户指定的用于影片特定部分（片尾、演职员表、……）的质量值。每个zone的格式是<start-frame>,<end-frame>,<option>，其中option可能是

q=<0-51>
量化参数

b=<0.01-100.0>
比特率的倍数

注意：量化参数选项不是严重执行的。其影响的只是码率控制过程中的计划阶段，并且仍然受制于溢出补偿和qp_min/qp_max选项。

direct_pred=<name>
决定用于B帧中直接模式宏块的运动预测的类型。

none
不使用直接模式的宏块。

spatial
运动矢量由邻接块推断出来。（默认值）

temporal
运动矢量由其后的P帧推断出来。

auto
编解码器为每个帧分别选择是用spatial还是用temporal。
spatial和temporal速度上和PSNR上大致相同，选择两者中的哪一个取决于视频的内容。
auto稍稍好一些，但运行慢一些。当与多阶段编码模式一同使用时，auto非常有效。
direct_pred=none不仅运行速度较慢而且质量也较差。

(no)weight_b
使用B帧中带权重的预测模式。不用这个选项的话，双向预测出的宏块给每个所参照的帧相等的权重值。使用了这个选项后，权重值是根据B帧相对参照帧的时间位置而决定的。要求bframes > 1。

partitions=<list>
启用一些可选的宏块类型（默认值：p8x8,b8x8,i8x8,i4x4）。

p8x8
启用p16x8、p8x16、p8x8类型。

p4x4
启用p8x4、p4x8、p4x4类型。p4x4只在subq >= 5，并且分辨率低时才推荐使用。

b8x8
启用b16x8、b8x16、b8x8类型。

	i8x8
启用i8x8类型。 除非启用了8x8dct，否则i8x8没有任何效果。	
	i4x4
启用i4x4类型。	
	all
启用以上所有类型。	
	none
禁用以上所有类型。	
不管这个选项设为何值，p16x16、b16x16和i16x16三种宏块类型总是启用的。 其思想是找到最适合描绘画面某一区域的宏块类型和尺寸。 例如，全局摇摄镜头较好以16x16的块来表示，而小型移动物体较好以小一点的块来表示。	
(no)8x8dct	
自适应空间变换尺寸：允许宏块在4x4和8x8的DCT间选择一种。 同时允许使用i8x8的宏块类型。 不使用这个选项，则只使用4x4的DCT。	
me=<name>	
选择全像素运动估计算法。	
	dia
菱形搜索，半径为1（运行快）	
	hex
六边形搜索，半径为2（默认值）	
	umh
非均匀的多六边形搜索（运行慢）	
	esa
彻底性搜索（运行非常慢，而且不比umh好多少）	
me_range=<4-64>	
彻底性运动搜索或多六边形运动搜索的半径（默认值：16）	
subq=<1-7>	
调整亚像素优化质量。 该参数控制的是运动估计决策过程中质量与速度的权衡。 subq=5能比subq=1多压缩掉10%。	
	1
对于所有候选宏块类型运行全像素精度的运动估计操作。 然后选择最佳的类型。 然后优化这种类型的运动值，使其达到快速四分之一像素模式的精度（运行最快）。	
	2
对于所有候选宏块类型运行半像素精度的运动估计操作。 然后选择最佳的类型。 然后优化这种类型的运动值，使其达到快速四分之一像素模式的精度。	
	3
与2相似，但采用较慢的四分之一像素优化模式。	
	4
对于所有候选宏块类型运行快速四分之一像素精度的运动估计操作。 然后选择最佳的类型。 然后完成对于此种类型的四分之像素模式的优化操作。	
	5
在选择最佳类型之前，对于所有候选宏块类型运行最佳质量的四分之一像素精度的运动估计 操作（默认值）。	
	6
启用I帧和P帧中宏块类型的码率失真优化模式。	
	7
启用运动矢量和帧内模式的码率失真优化模式。（最佳） 以上内容中，“所有的候选宏块类型”并不恰恰意味着所有已启用的类型： 4x4、4x8、8x4 只有当8x8比16x16好时才尝试采用。	

(no)chroma_me
在亚像素运动搜索中考虑色度信息（默认值：启用）。 要求subq>=5。

(no)mixed_refs
允许每个8x8或16x8的运动部分独立地选取一个参照帧。 不用这个选项的话，整个宏块必须采用同一个参照帧。 要求frameref>1。

(no)brdo
启用B帧中宏块类型的码率失真优化。 要求subq>=6。

(no)bime
优化双向宏块中所用的两个运动矢量，而不是重复使用向前和向后搜索中得到的矢量。没有B帧时这个选项没有任何作用。

trellis=<0-2>
码率失真最优的量化模式

禁用（默认值）	0
仅对最终编码启用	1
启用所有模式下的决策（运行慢，要求subq>=6）	2

deadzone_inter=<0-32>
设置非格子因子量化模式中帧间亮度量化无效区的大小（默认值：21）。 较小的值有助于保留最好的细节和影片的粒度感（特别是对于高比特率/质量编码有用）， 而较大的值有助于滤除这些细节从而省下比特以用在其它宏块和帧上（特别是对于低比特率的编码有用）。 推荐你在更改这个参数先调试一下deadzone_intra。

deadzone_intra=<0-32>
置非格子因子量化模式中帧内亮度量化无效区的大小（默认值：11）。 该选项与deadzone_inter有相同的效果，不同在于其影响的是内部帧。 推荐你在更改deadzone_inter之前先调试一下这个参数。

(no)fast_pskip
执行P帧中的早期跳跃侦测功能（默认值：启用）。 该选项通常不花任何代价而提升编码速度，但其有时可能在缺乏细节的画面区域，如天空， 产生画面损伤。

(no)dct_decimate
去除只含有单个微小系数的P帧中的DCT块（默认值：启用）。 该选项会去除一切画面细节，所以其会省下一些比特以用在其它帧上，从而有可能提升整体的主观质量。 如果你正以较高的目标比特率压缩非动画视频内容，那么你可能希望禁用这个选项以尽可能保留画面细节。

nr=<0-100000>
噪声削减强度，0表示禁用。 100-1000对于典型内容来说是一个有用的范围，但你可能希望对噪声很强的视频内容调高一些这个选项。 考虑到这个选项对于运行速度影响小，你可能希望倾向于使用这个选项，而不是使用诸如 denoise3d或hqdn3d之类的视频滤镜来滤除噪声。

chroma_qp_offset=<-12-12>
相对于亮度来说为色度使用一个不同的量化参数。 有用值的范围是<-2-2>（默认值：0）。

cqm=<flat|jvt|<filename>>
要么使用一个预先定义好的自定义量化矩阵，要么加一个JM格式的矩阵文件。

flat	使用预先定义好的平坦16矩阵（默认值）。
jvt	使用预先定义好的JVT矩阵。

<filename>

使用所给出的JM格式矩阵文件。

注意：Windows CMD.EXE的用户如果尝试使用全部CQM列表时，可能在解析命令行时遇到问题。这是因为命令行长度有限制。在这种情况下推荐将列表存在一个JM格式的CQM文件中，然后像上面所说那样加载进来。

cqm4iy=<list> (另参见cqm)

自定义的4x4帧内亮度矩阵，以由16个逗号分隔值域为1-255的值组成的列表的形式给出。

cqm4ic=<list> (另参见cqm)

自定义的4x4帧内色度矩阵，以由16个逗号分隔值域为1-255的值组成的列表的形式给出。

cqm4py=<list> (另参见cqm)

自定义的4x4帧间亮度矩阵，以由16个逗号分隔值域为1-255的值组成的列表的形式给出。

cqm4pc=<list> (另参见cqm)

自定义的4x4帧间色度矩阵，以由16个逗号分隔值域为1-255的值组成的列表的形式给出。

cqm8iy=<list> (另参见cqm)

自定义的8x8帧内亮度矩阵，以由64个逗号分隔值域为1-255的值组成的列表的形式给出。

cqm8py=<list> (另参见cqm)

自定义的8x8帧间亮度矩阵，以由64个逗号分隔值域为1-255的值组成的列表的形式给出。

level_idc=<10-51>

将比特率的等级设置为H.264标准附件A中定义的值（默认值：51 - 5.1级）。该选项用于告诉解码器其需要支持怎么样的性能。只有当你知道这个参数是什么意思，并且需要设置它时，才使用这个参数。

threads=<0-16>

生成线程以在多个CPU上平行编码（默认值：1）。该选项对于压缩质量稍有一些影响。0或'auto'告诉x264让其侦测你有多少个CPU，并选取一个适当的线程数。

(no)global_header

使SPS和PPS只出现一次，即在比特流的开始部分（默认值：禁用）。某些播放器，诸如Sony PSP，需要使用这个选项。默认的行为是使SPS和PPS在每个IDR帧前重复出现。

(no)interlaced

将视频内容作为隔行扫描内容对待。

log=<-1-3>

调整打印在屏幕上的日志信息的量。

-1

无

0

只打印错误信息。
1

警告信息

2

当编码结束时打印PSNR以及其它分析统计信息（默认值）

3

每个帧的PSNR、QP、帧类型、大小，以及其它统计信息

(no)psnr

打印信噪比统计信息。

注意：PSNR的‘Y’、‘U’、‘V’和‘Avg’域在数学上并不完美（这些值只是逐帧PSNR的平均值）。这些只是为了与JM标准的编解码器相比较，而保留下来的。出于其它目的的话，请采用log=3打印的‘Global’PSNR，或是逐帧的PSNR。

(no)ssim

打印结构相似度计量结果。该值为PSNR的一种替代计量值，并且可能与压缩后视频上观察到的质量有更好的相关性。

(no)visualize

启用x264在编码时的可视化效果。如果你系统上的x264支持这个选项，那么在编码过程中将打开一个新窗口，在这个窗口中，x264会尝试总体性地显示出每一帧是如何编码。在可视化模式下影片的每种块类型将以如下方式着色：

红/粉红

帧内块

蓝

帧间块

绿

跳跃

黄

B块

该功能可以认为是实验性，并且容易变化。特别是，其需要x264编译时启用可视化功能。注意在写这部分内容时，x264在编码并可视化显示每一帧后会暂停，等待用户按一个键，在此之后才编码下一帧。

xvfw (-xvfwopts)

使用Windows视频编解码器编码基本上是过时的做法，除非你希望编码为某种难懂而极端的编解码器格式。

codec=<name>

用于编码的二进制编解码器文件的文件名。

compdata=<file>

vfw2menc创建的编解码器设置文件的文件名（如firstpass.mcf）。

MPEG muxer (-mpegopts)

MPEG muxer可以生成5种类型的视频流，每种都有较好的默认参数，而用户可以更改这些参数。通常，当生成MPEG文件时，建议禁用MEncoder的跳帧指令（参见-noskip、-mc以及视频滤镜harddup和softskip）。

示例：

format=mpeg2:tsaf:vbitrate=8000

format=<mpeg1 | mpeg2 | xvc1 | xsvcd | dvd | pes1 | pes2>

视频流格式（默认值：mpeg2）。pes1与pes2是非常有问题的格式（没有打包头，没有填充），但VDR使用这些格式；不要选用这些格式，除非你确切知道你在做什么。

size=<最大65535>

单位为字节的打包大小，不要更改这个值除非你确切知道你在做什么（默认值：2048）。

muxrate=<int>

打包头部信息中的单位kbit/s的标称复合流速率（默认值：1800 kb/s）。当‘format=mpeg1’或‘mpeg2’时将根据需要获得更新。

tsaf

如果可能，在所有帧上设置时间戳值；推荐当format=dvd时使用。如果dvdauthor报警输出信息，例如“..audio sector out of range...”时，那么你可能不该启用这个选项。

interleaving2

使用一种更好的算法来交织音频和视频数据包，其所基于的原则是流合并器将总是尝试在空闲空间所占百分比最大的条件下填充流。

vdelay=<1-32760>

初始视频延迟时间，单位为毫秒（默认值：0），如果你想让视频相对音频有一定延迟，那么就使用这个选项。这个选项在使用了:drop时无效。

adelay=<1-32760>

初始音频延迟时间，单位为毫秒（默认值：0），如果你想让音频相对视频有一定延迟，那么就使用这个选项。

drop

当与vdelay一起使用时，流合并器将丢弃先于视频出现的那部分音频。

vwidth, vheight=<1-4095>

当视频是MPEG-1/2时，设置视频的宽度和高度。

vpswidth, vpsheight=<1-4095>

当视频是MPEG-2时，设置全景显示模式（pan-scan）下视频的宽度和高度。

vaspect=<1 | 4/3 | 16/9 | 221/100>

设置MPEG-2视频的显示宽高比。不要对MPEG-1视频使用这个选项，否则产生的宽高比将完全错误。

vbitrate=<int>

设置MPEG-1/2视频单位为kbit/s的视频比特率。

vframerate=<24000/1001 | 24 | 25 | 30000/1001 | 30 | 50 | 60000/1001 | 60 >

设置MPEG-1/2视频的帧率。如果与telecine选项一同使用，那么将忽略这个选项。

telecine

启用3:2下拉软性电视图像化模式：流合并器将使视频流看上去像是以30000/1001fps编码的。只有当输出帧率为24000/1001fps时这个选项才对MPEG-2有效，如需要则使用-ofps改变输出帧率。其它的帧率与该选项不兼容。

film2pal

启用FILM和NTSC至PAL软性电视图像化模式：流合并器将使视频流看上去像是以25fps编码的。只有当输出帧率为24000/1001fps时这个选项才对MPEG-2有效，如需要则使用-ofps改变输出帧率。其它的帧率与该选项不兼容。

tele_src and tele_dest

使用Donand Graft的DGPulldown代码启用任意电视图像化模式。你需要指定原始帧率和所期望的帧率；流合并器将使视频流看上去像是以所期望的帧率编码的。只有当输入的帧率比输出的帧率小，并且帧率的增加量 ≤ 1.5 时，这个选项才对MPEG-2视频有效。

示例：

tele_src=25,tele_dest=30000/1001

PAL至NTSC的电视图像化处理

vbuf_size=<40-1194>

设置视频解码器的缓冲大小，以千字节为单位表示。只有当视频流的比特率对于所选的格式来说太高了，并且你完全了解你在做什么时，才指定这个选项。太高的值可能导致影片无法播放，这取决于播放器的能力。当合并HDTV视频流时，400的值应该足够了。

abuf_size=<4-64>

设置音频解码器的缓冲大小，以千字节为单位表示。针对vbuf_size的原则在此同样适用。

FFmpeg libavformat demuxers (-lavfdopts)

probesize=<value>

在侦测阶段所探测的最大数据量。在MPEG-TS的情况下，这个值指定了所扫描TS包的

最大个数。

FFmpeg libavformat muxers (—lavfopts) (另参见—of lavf)

delay=<value>

当前只对MPEG[12]有意义：单位为秒的，对于任意出现的流，输出流参照计时器（SCR）和 解码时间戳（DTS）之间的最大所允许的差距。默认值为0.7（与MPEG标准所定义的强制执行值相同）。较高的值要求较大的缓冲，因而不应使用。

format=<container_format>

重新定义所合并输出的是哪种容器格式（默认值：根据输出文件的扩展名自动侦测）。

mpg

MPEG-1系统以及MPEG-2 PS

asf

高级流媒体格式

avi

音视频交织文件

wav

波形音频

swf

Macromedia Flash

flv

Macromedia Flash视频文件

rm

RealAudio和RealVideo

au

SUN AU格式

nut

NUT开放容器格式（实验性）

mov

QuickTime

mp4

MPEG-4格式

dv

Sony数字视频容器格式

muxrate=<rate>

流合流时的标称比特率，单位为比特每秒；当前该选项只对MPEG[12]有意义。有时需要提高这个值以防止“缓冲下溢”。

packetsize=<size>

所选格式单个数据包的大小，以字节为单位表示。当将流合并至MPEG[12]的实现格式时，默认值为：对于[S]VCD为2324，对于所有其它格式为2048。

preload=<distance>

当前只对MPEG[12]有意义：单位为秒，对于任意出现的流，输出流参照计时器（SCR）和解 码时间戳（DTS）间的初始差距（流分离至解码间的延迟）。

环境变量

有环境变量用于控制 MPlayer 和 MEncoder 的行为。

MPLAYER_CHARSET (也请参见 —msgcharset)

转换控制台的消息显示为指定字符集(默认值是自动检测: autodetect)。值 "noconv" 表示不要转换。

MPLAYER_HOME

MPlayer 寻找设置的目录。

MPLAYER_VERBOSE (也请参见 `-v` 和 `-msglevel`)

对所有的消息模块设置初始冗赘级别 (默认是: 0)。 负的值结果是越少的消息而正的值会越多。

libaf:

LADSPA_PATH

如果设置了 LADSPA_PATH, 它会寻找指定的文件。 如果没设置, 你就必须提供完整的路径名。 FIXME: 这也在 ladspa 一节有提到。

libdvdcss:

DVDCSS_CACHE

指定目录以储存标题的钥匙值。 这会加速 DVD 在缓存中的译码过程。 如果 DVDCSS_CACHE 目录不存在就创建它, 并创建以 DVD 标题或生产日期为名称的子目录。 如果 DVDCSS_CACHE 没设置或是空的, libdvdcss 会使用默认值为 Unix 的 `"${HOME}/.dvdcss/"` 或 Win32 的 `"C:\Documents and Settings\%USER%\Application Data\dvdcss\"`。 特殊值 "off" 停用缓存。

DVDCSS_METHOD

设置 libdvdcss 用于解读加密盘片的认证和解密方法。 可以是 title(标题), key(钥匙) 或 disc(盘片) 的其中之一。

key

是默认方法。 libdvdcss 会使用一套计算过的播放器密钥, 尝试取得盘片的钥匙。 如果驱动器不认识播放器的任何钥匙就会失败。

disc

是钥匙方法失败的备用方法。 代之使用播放器钥匙, libdvdcss 会使用蛮横暴力的算法破解盘片钥匙。 此过程需要 CPU 高强度运行和 64 MB 内存以储存临时的数据。

title

是所有其它方法的备用方法。 它不依靠与 DVD 设备交换钥匙, 而是使用一种密码机(crypto)攻击法来猜测标题钥匙。 此失败的情况极少, 因为盘片上没有足够的加密数据来低档统计攻击法, 但是另一方面这是仅有的方法来破译储存在硬盘上的 DVD, 和 RPC2 设备上不同地区码的 DVD。

DVDCSS_RAW_DEVICE

指定使用的原始设备。 确切的用法依赖于你的操作系统, 例如 Linux 工具建立的原始设备是 raw(8)。 请注意在大多数操作系统上, 使用原始设备需要高度对齐的缓冲(aligned buffers): Linux 需要 2048 字节(等于 DVD 扇区大小)的对齐。

DVDCSS_VERBOSE

设置 libdvdcss 冗赘级别。

0

一点都不输出消息。

1

输出错误消息到 stderr。

2

输出错误和调试消息到 stderr。

DVDREAD_NOKEYS

启动时跳过访问所有的钥匙。 目前停用。

HOME

FIXME:

待解释。

libao2:

AO_SUN_DISABLE_SAMPLE_TIMING

FIXME: 待解释。

AUDIODEV						
FIXME: 待解释。						
AUDIOSERVER						
指定 nas 音频输出驱动应该去连接的和传输的 网络音频系统(Network Audio System)服务器。 如果没设置, 就使用 DISPLAY。 传输可以是 tcp 和 unix 其中之一。 语法是 tcp/<某主机>:<某端口>, <某主机>:<实例号> 或 [unix]:<实例号>。 NAS 基端口是 8000 而 <实例号> 在其之上再增加。						
示例:						
AUDIOSERVER=somehost:0						
连接到 NAS 服务器, 某主机上使用默认的端口和传输协议。						
AUDIOSERVER=tcp/somehost:8000						
连接到 NAS 服务器, 某主机上监听 TCP 端口8000。						
AUDIOSERVER=(unix)?:0						
连接到 NAS 服务器, 本地主机上的实例 0 使用 unix 域套接口。						
DISPLAY						
FIXME: 待解释。						
vidix:						
VIDIX_CRT						
FIXME: 待解释。						
osdep:						
		TERM	FIXME:			
待解释。						
libvo:						
DISPLAY						
FIXME: 待解释。						
FRAMEBUFFER						
FIXME: 待解释。						
		HOME	FIXME:			
待解释。						
libmpdemux:						
		HOME	FIXME:			
待解释。						
HOMEPATH						
FIXME: 待解释。						
http_proxy						
FIXME: 待解释。						
LOGNAME						
FIXME: 待解释。						
USERPROFILE						
FIXME: 待解释。						
libmpcodecs:						
XANIM_MOD_DIR						
FIXME: 待解释。						
GUI:						
CHARSET						
FIXME: 待解释。						
DISPLAY						
FIXME: 待解释。						
		HOME	FIXME:			

待解释。

libavformat:

AUDIO_FLIP_LEFT

FIXME: 待解释。

BKTR_DEV

FIXME: 待解释。

BKTR_FORMAT

FIXME: 待解释。

BKTR_FREQUENCY

FIXME: 待解释。

http_proxy

FIXME: 待解释。

no_proxy

FIXME: 待解释。

相关文件

/usr/local/etc/mplayer/mplayer.conf

MPlayer 范围的设置

/usr/local/etc/mplayer/mencoder.conf

MEncoder 范围的设置

~/.mplayer/config

MPlayer 用户设置

~/.mplayer/mencoder.conf

MEncoder 用户设置

~/.mplayer/input.conf

输入绑定 (按键的完整列表请参见 "—input keylist" 的输出)

~/.mplayer/gui.conf

GUI 配置文件

~/.mplayer/gui.pl

GUI 播放列表

~/.mplayer/font/

字体目录 (里面必须有一个 font.desc 文件和带 .RAW 后缀的文件。)

~/.mplayer/DVDkeys/

缓存的 CSS 钥匙

假如播放文件是 /path/to/movie.avi, MPlayer 按以下顺序搜索字幕文件:

/path/to/movie.sub

~/.mplayer/sub/movie.sub

MPLAYER的使用范例

快速上手播放 DVD:

```
mplayer dvd://1
```

播放日文对话和英文字幕:

```
mplayer dvd://1 -alang ja -slang en
```

只播放第 5, 6, 7 场景:

```
mplayer dvd://1 -chapter 5-7
```

只播放第 5, 6, 7 标题:

```
mplayer dvd://5-7
```

播放多视角 DVD:

```
mplayer dvd://1 -dvdangle 2
```

从不同的 DVD 设备播放:

```
mplayer dvd://1 -dvd-device /dev/dvd2
```

从目录中的 VOB 文件播放 DVD 视频:

```
mplayer dvd://1 -dvd-device /path/to/directory/
```

复制 DVD 标题到硬盘, 保存文件名为 title1.vob :

```
mplayer dvd://1 -dumpstream -dumpfile title1.vob
```

播放 HTTP 流:

```
mplayer http://mplayer.hq/example.avi
```

播放 RTSP 流:

```
mplayer rtsp://server.example.com/streamName
```

转换字幕为 MPsub 格式:

```
mplayer dummy.avi -sub source.sub -dumpmpsub
```

转换字幕为 MPsub 格式但不播放:

```
mplayer /dev/zero -rawvideo pal:fps=xx -demuxer rawvideo -vc null -vo null -noframedrop -benchmark
```

从标准的 V4L 输入:

```
mplayer tv:// -tv driver=v4l:width=640:height=480:outfmt=i420 -vc rawi420 -vo xv
```

回放 Zoran 卡 (旧方法, 废弃):

```
mplayer -vo zr -vf scale=352:288 file.avi
```

回放 Zoran 卡 (新方法):

```
mplayer -vo zr2 -vf scale=352:288,zrmjpeg file.avi
```

只用两个扬声器播放 6-声道 AAC 文件:

```
mplayer -rawaudio format=0xff -demuxer rawaudio -af pan=2:.32:.32:.39:.06:.06:.39:.17:-.17:-.17:.17
```

你可能想用 pan 值播放片断 (例如用放大值) 以增加音量或者避免剪切。

使用geq滤镜产生的颠倒的棋盘：

```
mplayer -vf geq='128+(p(XY)-128)*(0.5-gt(mod(X/SW128)64))*(0.5-gt(mod(Y/SH128)64))*4'
```

MENCODER的使用范例

编码 DVD 标题 #2 中选择的场景：

```
mencoder dvd://2 -chapter 10-15 -o title2.avi -oac copy -ovc lavc -lavcopts vcodec=mpeg4
```

编码 DVD 标题 #2, 缩放到 640x480:

```
mencoder dvd://2 -vf scale=640:480 -o title2.avi -oac copy -ovc lavc -lavcopts vcodec=mpeg4
```

编码 DVD 标题 #2, 缩放到 512xHHH (保持宽高比):

```
mencoder dvd://2 -vf scale -zoom -xy 512 -o title2.avi -oac copy -ovc lavc -lavcopts vcodec=mpeg4
```

同上, 但使用比特率 1800kbit 和优化的 macroblocks:

```
mencoder dvd://2 -o title2.avi -oac copy -ovc lavc -lavcopts vcodec=mpeg4:mbd=1:vbitrate=1800
```

同上, 但使用 MJPEG 压缩:

```
mencoder dvd://2 -o title2.avi -oac copy -ovc lavc -lavcopts vcodec=mjpeg:mbd=1:vbitrate=1800
```

编码当前目录下的所有 *.jpg 文件:

```
mencoder "mf://*.jpg" -mf fps=25 -o output.avi -ovc lavc -lavcopts vcodec=mpeg4
```

从电视调谐器编码 (指定格式用 -vf format):

```
mencoder -tv driver=v4l:width=640:height=480 tv:// -o tv.avi -ovc raw
```

从管道编码:

```
rar p test-SVCD.rar | mencoder -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=800 -ofps 24 -
```

程序错误

不要惊慌。如果你发现错误, 报告我们, 但请首先确认你已阅读了所有的文档。也要小心那些表情符号 :) 许多错误是不正确的设置和参数使用引起的。错误报告的文档一节 (<http://www.mplayerhq.hu/DOCS/HTML/en/bugreports.html>) 解释了如何创建有用的错误报告。

程序的作者

MPlayer 最初由 Arpad Gereoffy 制作。其它许多贡献者的部分清单, 请参见 AUTHORS(作者) 文件。

MPlayer 是 (C) 2000–2007 The MPlayer Team (MPlayer 团队) 所有

本(英文)手册页主要由 Gabucino, Jonas Jermann 和 Diego Biurrun 编写, 由 Diego Biurrun 维护。视

频滤镜、通用编解码器选项、特定编解码器专用选项三个章节由 JRaSH <jrash06 AT 163.com> 重新翻译 参与翻译本中文手册页的有: Lu Ran <hephooey AT fastmail.fm>, Sheldon Jin <jinsh2 AT yahoo.com>, JRaSH<jrash06 At 163.com> 等。 有关本文的联系请发送邮件到 MPlayer-DOCS 邮件列表。 与翻译相关的邮件属于 MPlayer-translations 邮件列表。
