## 目录

* [1 概要](#Synopsis)
* [2 说明](#Description)
* [3 详细说明](#Detailed-description)
  + [3.1 过滤](#Filtering)
    - [3.1.1 简单的过滤图](#Simple-filtergraphs)
    - [3.1.2 复杂的过滤图](#Complex-filtergraphs)
  + [3.2 流复制](#Stream-copy)
* [4 码流选择](#Stream-selection)
  + [4.1 说明](#Description-1)
    - [4.1.1 自动码流选择](#Automatic-stream-selection)
    - [4.1.2 手动流选择](#Manual-stream-selection)
    - [4.1.3 复杂的过滤图](#Complex-filtergraphs-1)
    - [4.1.4 流处理](#Stream-handling)
  + [4.2 例子](#Examples)
* [5个选项](#Options)
  + [5.1 流说明符](#Stream-specifiers-1)
  + [5.2 通用选项](#Generic-options)
  + [5.3 AV选项](#AVOptions)
  + [5.4 主要选项](#Main-options)
  + [5.5 视频选项](#Video-Options)
  + [5.6 高级视频选项](#Advanced-Video-options)
  + [5.7 音频选项](#Audio-Options)
  + [5.8 高级音频选项](#Advanced-Audio-options)
  + [5.9 字幕选项](#Subtitle-options)
  + [5.10 高级字幕选项](#Advanced-Subtitle-options)
  + [5.11 高级选项](#Advanced-options)
  + [5.12 预设文件](#Preset-files)
    - [5.12.1 ff预设文件](#ffpreset-files)
    - [5.12.2 av预设文件](#avpreset-files)
* [6 例子](#Examples-1)
  + [6.1 视频和音频抓取](#Video-and-Audio-grabbing)
  + [6.2 X11抓取](#X11-grabbing)
  + [6.3 视音频文件格式转换](#Video-and-Audio-file-format-conversion)
* [7 语法](#Syntax)
  + [7.1 引用和转义](#Quoting-and-escaping)
    - [7.1.1 例子](#Examples-2)
  + [7.2 日期](#Date)
  + [7.3 持续时间](#Time-duration)
    - [7.3.1 例子](#Examples-3)
  + [7.4 视频大小](#Video-size)
  + [7.5 视频码率](#Video-rate)
  + [7.6 比例](#Ratio)
  + [7.7 颜色](#Color)
  + [7.8 频道布局](#Channel-Layout)
* [8 表达式评估](#Expression-Evaluation)
* [9 编解码器选项](#Codec-Options)
* [10个解码器](#Decoders)
* [11个视频解码器](#Video-Decoders)
  + [11.1 AV1](#av1)
    - [11.1.1 选项](#Options-1)
  + [11.2 原始视频](#rawvideo)
    - [11.2.1 选项](#Options-2)
  + [11.3 libdav1d](#libdav1d)
    - [11.3.1 选项](#Options-3)
  + [11.4 libdavs2](#libdavs2)
  + [11.5 libuavs3d](#libuavs3d)
    - [11.5.1 选项](#Options-4)
  + [11.6 QSV译码器](#QSV-Decoders)
    - [11.6.1 常用选项](#Common-Options)
    - [11.6.2 HEVC 选项](#HEVC-Options)
  + [11.7 v210](#v210)
    - [11.7.1 选项](#Options-5)
* [12个音频解码器](#Audio-Decoders)
  + [12.1 ac3](#ac3)
    - [12.1.1 AC-3 解码器选项](#AC_002d3-Decoder-Options)
  + [12.2 后高音](#flac-1)
    - [12.2.1 FLAC 解码器选项](#FLAC-Decoder-options)
  + [12.3 ffwavesynth](#ffwavesynth)
  + [12.4 libcelt](#libcelt)
  + [12.5 libgsm](#libgsm)
  + [12.6 libilbc](#libilbc)
    - [12.6.1 选项](#Options-6)
  + [12.7 libopencore-amrnb](#libopencore_002damrnb)
  + [12.8 libopencore-amrwb](#libopencore_002damrwb)
  + [12.9 libopus](#libopus)
* [13个字幕解码器](#Subtitles-Decoders)
  + [13.1 libaribb24](#libaribb24)
    - [13.1.1 libaribb24 解码器选项](#libaribb24-Decoder-Options)
  + [13.2 dvbsub](#dvbsub)
    - [13.2.1 选项](#Options-7)
  + [13.3 DVD 子](#dvdsub)
    - [13.3.1 选项](#Options-8)
  + [13.4 libzvbi-图文电视](#libzvbi_002dteletext)
    - [13.4.1 选项](#Options-9)
* [14个编码器](#Encoders)
* [15 个音频编码器](#Audio-Encoders)
  + [15.1 音频](#aac)
    - [15.1.1 选项](#Options-10)
  + [15.2 ac3 和 ac3\_fixed](#ac3-and-ac3_005ffixed)
    - [15.2.1 AC-3 元数据](#AC_002d3-Metadata)
      * [15.2.1.1 元数据控制选项](#Metadata-Control-Options)
      * [15.2.1.2 缩混级别](#Downmix-Levels)
      * [15.2.1.3 音频制作信息](#Audio-Production-Information)
      * [15.2.1.4 其他元数据选项](#Other-Metadata-Options)
    - [15.2.2 扩展比特流信息](#Extended-Bitstream-Information)
      * [15.2.2.1 扩展比特流信息——第 1 部分](#Extended-Bitstream-Information-_002d-Pa)
      * [15.2.2.2 扩展比特流信息——第 2 部分](#Extended-Bitstream-Information-_002d-Pa)
    - [15.2.3 其他 AC-3 编码选项](#Other-AC_002d3-Encoding-Options)
    - [15.2.4 仅浮点数 AC-3 编码选项](#Floating_002dPoint_002dOnly-AC_002d3-En)
  + [15.3 后高音](#flac-2)
    - [15.3.1 选项](#Options-11)
  + [15.4 作品](#opus)
    - [15.4.1 选项](#Options-12)
  + [15.5 libfdk\_aac](#libfdk_005faac)
    - [15.5.1 选项](#Options-13)
    - [15.5.2 例子](#Examples-4)
  + [15.6 libmp3lame](#libmp3lame-1)
    - [15.6.1 选项](#Options-14)
  + [15.7 libopencore-amrnb](#libopencore_002damrnb-1)
    - [15.7.1 选项](#Options-15)
  + [15.8 力宝](#libopus-1)
    - [15.8.1 选项映射](#Option-Mapping)
  + [15.9 libshine](#libshine-1)
    - [15.9.1 选项](#Options-16)
  + [15.10 libtwolame](#libtwolame)
    - [15.10.1 选项](#Options-17)
  + [15.11 libvo-amrwbenc](#libvo_002damrwbenc)
    - [15.11.1 选项](#Options-18)
  + [15.12 libvorbis](#libvorbis)
    - [15.12.1 选项](#Options-19)
  + [15.13 mjpeg](#mjpeg)
    - [15.13.1 选项](#Options-20)
  + [15.14 wavpack](#wavpack)
    - [15.14.1 选项](#Options-21)
      * [15.14.1.1 共享选项](#Shared-options)
      * [15.14.1.2 私有选项](#Private-options)
* [16 个视频编码器](#Video-Encoders)
  + [16.1 a64\_multi, a64\_multi5](#a64_005fmulti_002c-a64_005fmulti5)
  + [16.2 Cinepak](#Cinepak)
    - [16.2.1 选项](#Options-22)
  + [16.3 动图](#GIF)
    - [16.3.1 选项](#Options-23)
  + [16.4 哈普](#Hap)
    - [16.4.1 选项](#Options-24)
  + [16.5 jpeg2000](#jpeg2000)
    - [16.5.1 选项](#Options-25)
  + [16.6 图书馆](#librav1e)
    - [16.6.1 选项](#Options-26)
  + [16.7 libaom-av1](#libaom_002dav1)
    - [16.7.1 选项](#Options-27)
  + [16.8 libsvtav1](#libsvtav1)
    - [16.8.1 选项](#Options-28)
  + [16.9 libjxl](#libjxl)
    - [16.9.1 选项](#Options-29)
  + [16.10 libkvazaar](#libkvazaar)
    - [16.10.1 选项](#Options-30)
  + [16.11 libopenh264](#libopenh264)
    - [16.11.1 选项](#Options-31)
  + [16.12 libtheora](#libtheora)
    - [16.12.1 选项](#Options-32)
    - [16.12.2 例子](#Examples-5)
  + [16.13 libvpx](#libvpx)
    - [16.13.1 选项](#Options-33)
  + [16.14 libwebp](#libwebp)
    - [16.14.1 像素格式](#Pixel-Format)
    - [16.14.2 选项](#Options-34)
  + [16.15 libx264、libx264rgb](#libx264_002c-libx264rgb)
    - [16.15.1 支持的像素格式](#Supported-Pixel-Formats)
    - [16.15.2 选项](#Options-35)
  + [16.16 libx265](#libx265)
    - [16.16.1 选项](#Options-36)
  + [16.17 libxavs2](#libxavs2)
    - [16.17.1 选项](#Options-37)
  + [16.18 libxvid](#libxvid)
    - [16.18.1 选项](#Options-38)
  + [16.19 媒体基金会](#MediaFoundation)
  + [16.20 mpeg2](#mpeg2)
    - [16.20.1 选项](#Options-39)
  + [16.21 PNG](#png)
    - [16.21.1 私人期权](#Private-options-1)
  + [16.22 专业分辨率](#ProRes)
    - [16.22.1 prores-ks 的私有选项](#Private-Options-for-prores_002dks)
    - [16.22.2 速度考虑](#Speed-considerations)
  + [16.23 QSV 编码器](#QSV-Encoders)
    - [16.23.1 速率控制方法](#Ratecontrol-Method)
    - [16.23.2 全局选项 -> MSDK 选项](#Global-Options-_002d_003e-MSDK-Options)
    - [16.23.3 常用选项](#Common-Options-1)
    - [16.23.4 运行时选项](#Runtime-Options)
    - [16.23.5 H264 选项](#H264-options)
    - [16.23.6 HEVC 选项](#HEVC-Options-1)
    - [16.23.7 MPEG2 选项](#MPEG2-Options)
    - [16.23.8 VP9 选项](#VP9-Options)
    - [16.23.9 AV1 选项](#AV1-Options)
  + [16.24雪](#snow)
    - [16.24.1 选项](#Options-40)
  + [16.25 VAAPI 编码器](#VAAPI-encoders)
  + [16.26 vbn](#vbn)
    - [16.26.1 选项](#Options-41)
  + [16.27 VC2](#vc2)
    - [16.27.1 选项](#Options-42)
* [17个字幕编码器](#Subtitles-Encoders)
  + [17.1 DVD 子](#dvdsub-1)
    - [17.1.1 选项](#Options-43)
* [18 个比特流过滤器](#Bitstream-Filters)
  + [18.1 aac\_adtstoasc](#aac_005fadtstoasc)
  + [18.2 av1\_元数据](#av1_005fmetadata)
  + [18.3 咬](#chomp)
  + [18.4 dca\_core](#dca_005fcore)
  + [18.5 dump\_extra](#dump_005fextra)
  + [18.6 dv\_error\_marker](#dv_005ferror_005fmarker)
  + [18.7 eac3\_core](#eac3_005fcore)
  + [18.8 extract\_extradata](#extract_005fextradata)
  + [18.9 filter\_units](#filter_005funits)
  + [18.10 hapqa\_extract](#hapqa_005fextract)
  + [18.11 h264\_元数据](#h264_005fmetadata)
  + [18.12 h264\_mp4toannexb](#h264_005fmp4toannexb)
  + [18.13 h264\_redundant\_pps](#h264_005fredundant_005fpps)
  + [18.14 hevc\_metadata](#hevc_005fmetadata)
  + [18.15 hevc\_mp4toannexb](#hevc_005fmp4toannexb)
  + [18.16 imxdump](#imxdump)
  + [18.17 mjpeg2jpeg](#mjpeg2jpeg)
  + [18.18 mjpegadump](#mjpegadump)
  + [18.19 mov2textsub](#mov2textsub-1)
  + [18.20 mp3 分解](#mp3decomp)
  + [18.21 mpeg2\_元数据](#mpeg2_005fmetadata)
  + [18.22 mpeg4\_unpack\_bframes](#mpeg4_005funpack_005fbframes)
  + [18.23噪音](#noise)
    - [18.23.1 例子](#Examples-6)
  + [18.24 空](#null)
  + [18.25 pcm\_rechunk](#pcm_005frechunk)
  + [18.26 pgs\_frame\_merge](#pgs_005fframe_005fmerge)
  + [18.27 prores\_metadata](#prores_005fmetadata)
  + [18.28 remove\_extra](#remove_005fextra)
  + [18.29 套](#setts)
  + [18.30 text2movsub](#text2movsub-1)
  + [18.31 trace\_headers](#trace_005fheaders)
  + [18.32 truehd\_core](#truehd_005fcore)
  + [18.33 vp9\_元数据](#vp9_005fmetadata)
  + [18.34 vp9\_superframe](#vp9_005fsuperframe)
  + [18.35 vp9\_superframe\_split](#vp9_005fsuperframe_005fsplit)
  + [18.36 vp9\_raw\_reorder](#vp9_005fraw_005freorder)
* [19 格式选项](#Format-Options)
  + [19.1 格式流说明符](#Format-stream-specifiers-1)
* [20 个解复用器](#Demuxers)
  + [20.1AA](#aa)
  + [20.2 音频](#aac-1)
  + [20.3 磅](#apng)
  + [20.4 自动对焦](#asf-1)
  + [20.5 连续](#concat-1)
    - [20.5.1 语法](#Syntax-1)
    - [20.5.2 选项](#Options-44)
    - [20.5.3 例子](#Examples-7)
  + [20.6 破折号](#dash-1)
    - [20.6.1 选项](#Options-45)
  + [20.7个](#ea)
    - [20.7.1 选项](#Options-46)
  + [20.8 国际货币基金组织](#imf)
  + [20.9 flv、live\_flv、kux](#flv_002c-live_005fflv_002c-kux)
  + [20.10 动图](#gif-1)
  + [20.11 小时](#hls-1)
  + [20.12 图片 2](#image2-1)
    - [20.12.1 例子](#Examples-8)
  + [20.13 libgme](#libgme)
  + [20.14 libmodplug](#libmodplug)
  + [20.15 libopenmpt](#libopenmpt)
  + [20.16 mov/mp4/3gp](#mov_002fmp4_002f3gp)
    - [20.16.1 选项](#Options-47)
    - [20.16.2 有声 AAX](#Audible-AAX)
  + [20.17 mpegts](#mpegts)
  + [20.18 mpjpeg](#mpjpeg)
  + [20.19 原始视频](#rawvideo-1)
  + [20.20 sbg](#sbg)
  + [20.21 字幕](#tedcaptions)
  + [20.22 蒸汽合成器](#vapoursynth)
* [21 个多路复用器](#Muxers)
  + [21.1 a64](#a64-1)
  + [21.2 广告](#adts-1)
    - [21.2.1 选项](#Options-48)
  + [21.3 艾夫](#aiff-1)
    - [21.3.1 选项](#Options-49)
  + [21.4 阿尔卑斯山](#alp-1)
    - [21.4.1 选项](#Options-50)
  + [21.5 英尺](#asf-2)
    - [21.5.1 选项](#Options-51)
  + [21.6.avi](#avi-1)
    - [21.6.1 选项](#Options-52)
  + [21.7 套印](#chromaprint-1)
    - [21.7.1 选项](#Options-53)
  + [21.8CRC](#crc-1)
    - [21.8.1 例子](#Examples-9)
  + [21.9 破折号](#dash-2)
  + [21.10 先进先出](#fifo-1)
    - [21.10.1 例子](#Examples-10)
  + [21.11 flv](#flv)
  + [21.12 帧crc](#framecrc-1)
    - [21.12.1 例子](#Examples-11)
  + [21.13 帧哈希](#framehash-1)
    - [21.13.1 例子](#Examples-12)
  + [21.14 framemd5](#framemd5-1)
    - [21.14.1 例子](#Examples-13)
  + [21.15 动图](#gif-2)
  + [21.16哈希](#hash-1)
    - [21.16.1 例子](#Examples-14)
  + [21.17 小时](#hls-2)
    - [21.17.1 选项](#Options-54)
  + [21.18 图标](#ico-1)
  + [21.19 图片 2](#image2-2)
    - [21.19.1 选项](#Options-55)
    - [21.19.2 例子](#Examples-15)
  + [21.20 马特罗斯卡](#matroska)
    - [21.20.1 元数据](#Metadata)
    - [21.20.2 选项](#Options-56)
  + [21.21 MD5](#md5-1)
    - [21.21.1 例子](#Examples-16)
  + [21.22 mov、mp4、ismv](#mov_002c-mp4_002c-ismv)
    - [21.22.1 选项](#Options-57)
    - [21.22.2 例子](#Example)
  + [21.23 mp3](#mp3)
  + [21.24 mpegts](#mpegts-1)
    - [21.24.1 选项](#Options-58)
    - [21.24.2 例子](#Example-1)
  + [21.25 mxf，mxf\_d10，mxf\_opatom](#mxf_002c-mxf_005fd10_002c-mxf_005fopato)
    - [21.25.1 选项](#Options-59)
  + [21.26 空](#null-1)
  + [21.27 螺母](#nut)
  + [21.28 奥格](#ogg)
  + [21.29 原始混合器](#raw-muxers-1)
    - [21.29.1 ac3](#ac3-1)
    - [21.29.2 adx](#adx)
    - [21.29.3 aptx](#aptx)
    - [21.29.4 aptx\_hd](#aptx_005fhd)
    - [21.29.5 avs2](#avs2)
    - [21.29.6 cavsvideo](#cavsvideo)
    - [21.29.7 编解码器 2raw](#codec2raw)
    - [21.29.8数据](#data)
    - [21.29.9 狄拉克](#dirac)
    - [21.29.10 dnxhd](#dnxhd)
    - [21.29.11 dts](#dts)
    - [21.29.12 eac3](#eac3)
    - [21.29.13 g722](#g722)
    - [21.29.14 g723\_1](#g723_005f1)
    - [21.29.15 g726](#g726)
    - [21.29.16 g726le](#g726le)
    - [21.29.17 gsm](#gsm)
    - [21.29.18 h261](#h261)
    - [21.29.19 h263](#h263)
    - [21.29.20 h264](#h264)
    - [21.29.21 hevc](#hevc)
    - [21.29.22 m4v](#m4v)
    - [21.29.23 mjpeg](#mjpeg-1)
    - [21.29.24 mlp](#mlp)
    - [21.29.25 mp2](#mp2)
    - [21.29.26 mpeg1 视频](#mpeg1video)
    - [21.29.27 mpeg2video](#mpeg2video)
    - [21.29.28 奥布](#obu)
    - [21.29.29 原始视频](#rawvideo-2)
    - [21.29.30 标准广播公司](#sbc)
    - [21.29.31 真高清](#truehd)
    - [21.29.32 vc1](#vc1)
  + [21.30 段，stream\_segment，ssegment](#segment_002c-stream_005fsegment_002c-ss)
    - [21.30.1 选项](#Options-60)
    - [21.30.2 例子](#Examples-17)
  + [21.31 平滑流](#smoothstreaming)
  + [21.32 流哈希](#streamhash-1)
    - [21.32.1 例子](#Examples-18)
  + [21.33 发球台](#tee-1)
    - [21.33.1 选项](#Options-61)
    - [21.33.2 例子](#Examples-19)
  + [21.34 webm\_chunk](#webm_005fchunk)
    - [21.34.1 选项](#Options-62)
    - [21.34.2 例子](#Example-2)
  + [21.35 webm\_dash\_manifest](#webm_005fdash_005fmanifest)
    - [21.35.1 选项](#Options-63)
    - [21.35.2 例子](#Example-3)
* [22元数据](#Metadata-1)
* [23 协议选项](#Protocol-Options)
* [24个协议](#Protocols)
  + [24.1 amqp](#amqp)
  + [24.2 异步](#async)
  + [24.3蓝光](#bluray)
  + [24.4缓存](#cache)
  + [24.5 连续](#concat-2)
  + [24.6 连接](#concatf)
  + [24.7 加密](#crypto)
  + [24.8数据](#data-1)
  + [24.9 英尺](#fd)
  + [24.10档](#file)
  + [24.11 FTP](#ftp)
  + [24.12地鼠](#gopher)
  + [24.13 地鼠](#gophers)
  + [24.14 小时](#hls-3)
  + [24.15 HTTP](#http)
    - [24.15.1 HTTP Cookie](#HTTP-Cookies)
  + [24.16 冰铸](#Icecast)
  + [24.17 ipfs](#ipfs)
  + [24.18 毫米](#mmst)
  + [24.19 毫米](#mmsh)
  + [24.20 MD5](#md5-2)
  + [24.21管](#pipe)
  + [24.22 prompeg](#prompeg)
  + [24.23 瑞斯特](#rist)
  + [24.24 rtmp](#rtmp)
  + [24.25 rmp](#rtmpe)
  + [24.26 rtmps](#rtmps)
  + [24.27 rtmpt](#rtmpt)
  + [24.28 rtmpte](#rtmpte)
  + [24.29 rtmpts](#rtmpts)
  + [24.30 libsmb 客户端](#libsmbclient)
  + [24.31 libssh](#libssh)
  + [24.32 librtmp rtmp、rtmpe、rtmps、rtmpt、rtmpte](#librtmp-rtmp_002c-rtmpe_002c-rtmps_002c)
  + [24.33 分钟](#rtp)
  + [24.34 rtsp](#rtsp)
    - [24.34.1 混合器](#Muxer)
    - [24.34.2 解复用器](#Demuxer)
    - [24.34.3 例子](#Examples-20)
  + [24.35闷棍](#sap)
    - [24.35.1 混合器](#Muxer-1)
    - [24.35.2 解复用器](#Demuxer-1)
  + [24.36 sctp](#sctp)
  + [24.37 秒](#srt)
  + [24.38 srtp](#srtp)
  + [24.39 子文件](#subfile)
  + [24.40 开球](#tee-2)
  + [24.41 TCP](#tcp)
  + [24.42 吨](#tls)
  + [24.43 开发协议](#udp)
    - [24.43.1 例子](#Examples-21)
  + [24.44 操作系统](#unix)
  + [24.45 zmq](#zmq)
* [25 设备选项](#Device-Options)
* [26个输入设备](#Input-Devices)
  + [26.1 阿尔萨](#alsa)
    - [26.1.1 选项](#Options-64)
  + [26.2 android\_camera](#android_005fcamera)
    - [26.2.1 选项](#Options-65)
  + [26.3 基础](#avfoundation)
    - [26.3.1 选项](#Options-66)
    - [26.3.2 例子](#Examples-22)
  + [26.4 桶](#bktr)
    - [26.4.1 选项](#Options-67)
  + [26.5 甲板链接](#decklink)
    - [26.5.1 选项](#Options-68)
    - [26.5.2 例子](#Examples-23)
  + [26.6 显示](#dshow)
    - [26.6.1 选项](#Options-69)
    - [26.6.2 例子](#Examples-24)
  + [26.7 fbdev](#fbdev)
    - [26.7.1 选项](#Options-70)
  + [26.8 gdigrab](#gdigrab)
    - [26.8.1 选项](#Options-71)
  + [26.9 iec61883](#iec61883)
    - [26.9.1 选项](#Options-72)
    - [26.9.2 例子](#Examples-25)
  + [26.10 千斤顶](#jack)
    - [26.10.1 选项](#Options-73)
  + [26.11 kmsgrab](#kmsgrab)
    - [26.11.1 选项](#Options-74)
    - [26.11.2 例子](#Examples-26)
  + [26.12 拉菲](#lavfi)
    - [26.12.1 选项](#Options-75)
    - [26.12.2 例子](#Examples-27)
  + [26.13 libcdio](#libcdio)
    - [26.13.1 选项](#Options-76)
  + [26.14 libdc1394](#libdc1394)
    - [26.14.1 选项](#Options-77)
  + [26.15开](#openal)
    - [26.15.1 选项](#Options-78)
    - [26.15.2 例子](#Examples-28)
  + [26.16 奥斯](#oss)
    - [26.16.1 选项](#Options-79)
  + [26.17脉搏](#pulse)
    - [26.17.1 选项](#Options-80)
    - [26.17.2 例子](#Examples-29)
  + [26.18 sndio](#sndio)
    - [26.18.1 选项](#Options-81)
  + [26.19 video4linux2，v4l2](#video4linux2_002c-v4l2)
    - [26.19.1 选项](#Options-82)
  + [26.20 vfwcap](#vfwcap)
    - [26.20.1 选项](#Options-83)
  + [26.21 x11grab](#x11grab)
    - [26.21.1 选项](#Options-84)
* [27 输出设备](#Output-Devices)
  + [27.1 阿尔萨](#alsa-1)
    - [27.1.1 例子](#Examples-30)
  + [27.2 音频工具箱](#AudioToolbox)
    - [27.2.1 选项](#Options-85)
    - [27.2.2 例子](#Examples-31)
  + [27.3卡卡](#caca)
    - [27.3.1 选项](#Options-86)
    - [27.3.2 例子](#Examples-32)
  + [27.4甲板链接](#decklink-1)
    - [27.4.1 选项](#Options-87)
    - [27.4.2 例子](#Examples-33)
  + [27.5 fbdev](#fbdev-1)
    - [27.5.1 选项](#Options-88)
    - [27.5.2 例子](#Examples-34)
  + [27.6 开放式](#opengl)
    - [27.6.1 选项](#Options-89)
    - [27.6.2 例子](#Examples-35)
  + [27.7 奥斯](#oss-1)
  + [27.8脉搏](#pulse-1)
    - [27.8.1 选项](#Options-90)
    - [27.8.2 例子](#Examples-36)
  + [27.9 升](#sdl)
    - [27.9.1 选项](#Options-91)
    - [27.9.2 交互式命令](#Interactive-commands)
    - [27.9.3 例子](#Examples-37)
  + [27.10 sndio](#sndio-1)
  + [27.11 v4l2](#v4l2)
  + [27.12 xv](#xv)
    - [27.12.1 选项](#Options-92)
    - [27.12.2 例子](#Examples-38)
* [28 重采样器选项](#Resampler-Options)
* [29 缩放器选项](#Scaler-Options)
* [30 过滤介绍](#Filtering-Introduction)
* [31 图2点](#graph2dot)
* [32 过滤图说明](#Filtergraph-description)
  + [32.1 过滤图语法](#Filtergraph-syntax-1)
  + [32.2 关于过滤图转义的注意事项](#Notes-on-filtergraph-escaping)
* [33 时间线编辑](#Timeline-editing)
* [34 在运行时使用命令更改选项](#Changing-options-at-runtime-with-a-comm)
* [具有多个输入的过滤器的 35 个选项（帧同步）](#Options-for-filters-with-several-inputs)
* [36个音频过滤器](#Audio-Filters)
  + [36.1 压缩机](#acompressor)
    - [36.1.1 命令](#Commands)
  + [36.2对比](#acontrast)
  + [36.3 一份](#acopy)
  + [36.4 淡入淡出](#acrossfade)
    - [36.4.1 例子](#Examples-39)
  + [36.5跨度](#acrossover)
    - [36.5.1 例子](#Examples-40)
  + [36.6破碎机](#acrusher)
    - [36.6.1 命令](#Commands-1)
  + [36.7 锐](#acue)
  + [36.8 declick](#adeclick)
  + [36.9 剪辑](#adeclip)
  + [36.10 去相关](#adecorrelate)
  + [36.11 延迟](#adelay)
    - [36.11.1 例子](#Examples-41)
  + [36.12 腺病毒](#adenorm)
    - [36.12.1 命令](#Commands-2)
  + [36.13 衍生物，积分](#aderivative_002c-aintegral)
  + [36.14 阿德克](#adrc)
    - [36.14.1 命令](#Commands-3)
    - [36.14.2 例子](#Examples-42)
  + [36.15 动态均衡器](#adynamicequalizer)
    - [36.15.1 命令](#Commands-4)
  + [36.16 adynamicsmooth](#adynamicsmooth)
    - [36.16.1 命令](#Commands-5)
  + [36.17回声](#aecho)
    - [36.17.1 例子](#Examples-43)
  + [36.18 强调](#aemphasis)
    - [36.18.1 命令](#Commands-6)
  + [36.19 评估](#aeval)
    - [36.19.1 例子](#Examples-44)
  + [36.20 激励器](#aexciter)
    - [36.20.1 命令](#Commands-7)
  + [36.21 淡出](#afade-1)
    - [36.21.1 命令](#Commands-8)
    - [36.21.2 例子](#Examples-45)
  + [36.22 阿夫顿](#afftdn)
    - [36.22.1 命令](#Commands-9)
    - [36.22.2 例子](#Examples-46)
  + [36.23 关系](#afftfilt)
    - [36.23.1 例子](#Examples-47)
  + [36.24 冷](#afir-1)
    - [36.24.1 例子](#Examples-48)
  + [36.25格式](#aformat-1)
  + [36.26 调频](#afreqshift)
    - [36.26.1 命令](#Commands-10)
  + [36.27 AFWDN](#afwtdn)
    - [36.27.1 命令](#Commands-11)
  + [36.28玛瑙](#agate)
    - [36.28.1 命令](#Commands-12)
  + [36.29 空气](#aiir)
    - [36.29.1 例子](#Examples-49)
  + [36.30 限制器](#alimiter)
  + [36.31 全通](#allpass)
    - [36.31.1 命令](#Commands-13)
  + [36.32 单循环](#aloop)
  + [36.33 合并](#amerge-1)
    - [36.33.1 例子](#Examples-50)
  + [36.34 混音](#amix)
    - [36.34.1 例子](#Examples-51)
    - [36.34.2 命令](#Commands-14)
  + [36.35乘以](#amultiply)
  + [36.36 均衡器](#anequalizer)
    - [36.36.1 例子](#Examples-52)
    - [36.36.2 命令](#Commands-15)
  + [36.37 anlmdn](#anlmdn)
    - [36.37.1 命令](#Commands-16)
  + [36.38 anlmf，anlms](#anlmf_002c-anlms)
    - [36.38.1 例子](#Examples-53)
    - [36.38.2 命令](#Commands-17)
  + [36.39 无效](#anull)
  + [36.40 帕德](#apad)
    - [36.40.1 例子](#Examples-54)
  + [36.41 移相器](#aphaser)
  + [36.42 相移](#aphaseshift)
    - [36.42.1 命令](#Commands-18)
  + [36.43 精神分裂症](#apsyclip)
    - [36.43.1 命令](#Commands-19)
  + [36.44 脉动器](#apulsator)
  + [36.45 样品](#aresample-1)
    - [36.45.1 例子](#Examples-55)
  + [36.46 反向](#areverse)
    - [36.46.1 例子](#Examples-56)
  + [36.47 恩恩](#arnndn)
    - [36.47.1 命令](#Commands-20)
  + [36.48 asdr](#asdr)
  + [36.49 个样本](#asetnsamples)
  + [36.50 asetrate](#asetrate)
  + [36.51 显示信息](#ashowinfo)
  + [36.52 软剪辑](#asoftclip)
    - [36.52.1 命令](#Commands-21)
  + [36.53 方面统计](#aspectralstats)
  + [36.54 阿斯里](#asr)
  + [36.55 统计数据](#astats-1)
  + [36.56 asubboost](#asubboost)
    - [36.56.1 命令](#Commands-22)
  + [36.57 次切](#asubcut)
    - [36.57.1 命令](#Commands-23)
  + [36.58 超切](#asupercut)
    - [36.58.1 命令](#Commands-24)
  + [36.59 超级通行证](#asuperpass)
    - [36.59.1 命令](#Commands-25)
  + [36.60 超级站](#asuperstop)
    - [36.60.1 命令](#Commands-26)
  + [36.61 节奏](#atempo)
    - [36.61.1 例子](#Examples-57)
    - [36.61.2 命令](#Commands-27)
  + [36.62 倾斜度](#atilt)
    - [36.62.1 命令](#Commands-28)
  + [36.63 阿特里姆](#atrim)
  + [36.64 轴相关](#axcorrelate)
    - [36.64.1 例子](#Examples-58)
  + [36.65 带通](#bandpass)
    - [36.65.1 命令](#Commands-29)
  + [36.66 带阻](#bandreject)
    - [36.66.1 命令](#Commands-30)
  + [36.67 低音，低架](#bass_002c-lowshelf)
    - [36.67.1 命令](#Commands-31)
  + [36.68 双二阶](#biquad)
    - [36.68.1 命令](#Commands-32)
  + [36.69 bs2b](#bs2b)
  + [36.70 频道地图](#channelmap)
    - [36.70.1 例子](#Examples-59)
  + [36.71 频道分割](#channelsplit)
    - [36.71.1 例子](#Examples-60)
  + [36.72合唱](#chorus)
    - [36.72.1 例子](#Examples-61)
  + [36.73压缩比](#compand)
    - [36.73.1 例子](#Examples-62)
  + [36.74补偿延迟](#compensationdelay)
    - [36.74.1 命令](#Commands-33)
  + [36.75 交叉馈电](#crossfeed)
    - [36.75.1 命令](#Commands-34)
  + [36.76结晶器](#crystalizer)
    - [36.76.1 命令](#Commands-35)
  + [36.77 直流偏移](#dcshift)
  + [36.78 去嘶声](#deesser)
  + [36.79 对话增强](#dialoguenhance)
    - [36.79.1 命令](#Commands-36)
  + [36.80 drmeter](#drmeter)
  + [36.81 dynaudnorm](#dynaudnorm)
    - [36.81.1 命令](#Commands-37)
  + [36.82 耳垢](#earwax)
  + [36.83均衡器](#equalizer)
    - [36.83.1 例子](#Examples-63)
    - [36.83.2 命令](#Commands-38)
  + [36.84 超立体声](#extrastereo)
    - [36.84.1 命令](#Commands-39)
  + [36.85 防火等级](#firequalizer)
    - [36.85.1 例子](#Examples-64)
  + [36.86镶边](#flanger)
  + [36.87哈斯](#haas)
  + [36.88 硬盘](#hdcd)
  + [36.89耳机](#headphone)
    - [36.89.1 例子](#Examples-65)
  + [36.90 高通](#highpass)
    - [36.90.1 命令](#Commands-40)
  + [36.91加入](#join)
  + [36.92 拉兹帕](#ladspa)
    - [36.92.1 例子](#Examples-66)
    - [36.92.2 命令](#Commands-41)
  + [36.93 响度](#loudnorm)
  + [36.94 低通](#lowpass)
    - [36.94.1 例子](#Examples-67)
    - [36.94.2 命令](#Commands-42)
  + [36.95 等级 2](#lv2)
    - [36.95.1 例子](#Examples-68)
    - [36.95.2 命令](#Commands-43)
  + [36.96 mcompand](#mcompand)
  + [36.97 平底锅](#pan-1)
    - [36.97.1 混合示例](#Mixing-examples)
    - [36.97.2 重映射示例](#Remapping-examples)
  + [36.98重播增益](#replaygain)
  + [36.99 重新取样](#resample)
  + [36.100 橡皮筋](#rubberband)
    - [36.100.1 命令](#Commands-44)
  + [36.101 侧链压缩](#sidechaincompress)
    - [36.101.1 命令](#Commands-45)
    - [36.101.2 示例](#Examples-69)
  + [36.102 侧链门](#sidechaingate)
    - [36.102.1 命令](#Commands-46)
  + [36.103 静默检测](#silencedetect)
    - [36.103.1 示例](#Examples-70)
  + [36.104 沉默移除](#silenceremove)
    - [36.104.1 示例](#Examples-71)
  + [36.105 舒缓剂](#sofalizer)
    - [36.105.1 示例](#Examples-72)
  + [36.106语音规范](#speechnorm)
    - [36.106.1 命令](#Commands-47)
    - [36.106.2 示例](#Examples-73)
  + [36.107 立体工具](#stereotools)
    - [36.107.1 命令](#Commands-48)
    - [36.107.2 示例](#Examples-74)
  + [36.108 立体加宽](#stereowiden)
    - [36.108.1 命令](#Commands-49)
  + [36.109 超级均衡器](#superequalizer)
  + [36.110环绕声](#surround)
  + [36.111 倾斜架](#tiltshelf)
    - [36.111.1 命令](#Commands-50)
  + [36.112 高音，高架](#treble_002c-highshelf)
    - [36.112.1 命令](#Commands-51)
  + [36.113颤音](#tremolo)
  + [36.114抖音](#vibrato)
  + [36.115 虚拟低音](#virtualbass)
  + [36.116卷](#volume)
    - [36.116.1 命令](#Commands-52)
    - [36.116.2 示例](#Examples-75)
  + [36.117 体积检测](#volumedetect)
    - [36.117.1 示例](#Examples-76)
* [37 个音源](#Audio-Sources)
  + [37.1 缓冲区](#abuffer)
    - [37.1.1 例子](#Examples-77)
  + [37.2 aevalsrc](#aevalsrc)
    - [37.2.1 例子](#Examples-78)
  + [37.3afdelaysrc](#afdelaysrc)
  + [37.4 afirsrc](#afirsrc)
  + [37.5 anullsrc](#anullsrc)
    - [37.5.1 例子](#Examples-79)
  + [37.6 片](#flite)
    - [37.6.1 例子](#Examples-80)
  + [37.7 噪声源](#anoisesrc)
    - [37.7.1 例子](#Examples-81)
  + [37.8 希尔伯特](#hilbert)
  + [37.9正弦](#sinc)
  + [37.10 正弦](#sine)
    - [37.10.1 例子](#Examples-82)
* [38 个音频接收器](#Audio-Sinks)
  + [38.1 缓冲槽](#abuffersink)
  + [38.2 空槽](#anullsink)
* [39个视频过滤器](#Video-Filters)
  + [39.1 加德罗伊](#addroi)
    - [39.1.1 例子](#Examples-83)
  + [39.2 阿尔法提取物](#alphaextract)
  + [39.3 字母合并](#alphamerge)
  + [39.4 放大](#amplify)
    - [39.4.1 命令](#Commands-53)
  + [39.5 屁股](#ass)
  + [39.6 噪音](#atadenoise)
    - [39.6.1 命令](#Commands-54)
  + [39.7平均模糊](#avgblur)
    - [39.7.1 命令](#Commands-55)
  + [39.8 背景键](#backgroundkey)
    - [39.8.1 命令](#Commands-56)
  + [39.9 bbox](#bbox)
    - [39.9.1 命令](#Commands-57)
  + [39.10双边](#bilateral)
    - [39.10.1 命令](#Commands-58)
  + [39.11双边\_cuda](#bilateral_005fcuda)
    - [39.11.1 例子](#Examples-84)
  + [39.12 位平面噪声](#bitplanenoise)
  + [39.13 黑检测](#blackdetect)
  + [39.14 黑框](#blackframe)
  + [39.15 混合](#blend-1)
    - [39.15.1 例子](#Examples-85)
    - [39.15.2 命令](#Commands-59)
  + [39.16 块检测](#blockdetect-1)
    - [39.16.1 例子](#Examples-86)
  + [39.17 模糊检测](#blurdetect-1)
    - [39.17.1 例子](#Examples-87)
  + [39.18 bm3d](#bm3d)
    - [39.18.1 例子](#Examples-88)
  + [39.19 盒模糊](#boxblur)
    - [39.19.1 例子](#Examples-89)
  + [39.20 bwdif](#bwdif)
  + [39.21 约](#cas)
    - [39.21.1 命令](#Commands-60)
  + [39.22 色度保持](#chromahold)
    - [39.22.1 命令](#Commands-61)
  + [39.23 色键](#chromakey-1)
    - [39.23.1 命令](#Commands-62)
    - [39.23.2 例子](#Examples-90)
  + [39.24 chromekey\_cuda](#chromakey_005fcuda)
    - [39.24.1 例子](#Examples-91)
  + [39.25 色满](#chromanr)
    - [39.25.1 命令](#Commands-63)
  + [39.26 色移](#chromashift)
    - [39.26.1 命令](#Commands-64)
  + [39.27 ciescope](#ciescope)
  + [39.28编解码器视图](#codecview)
    - [39.28.1 例子](#Examples-92)
  + [39.29 色彩平衡](#colorbalance)
    - [39.29.1 例子](#Examples-93)
    - [39.29.2 命令](#Commands-65)
  + [39.30 颜色对比](#colorcontrast)
    - [39.30.1 命令](#Commands-66)
  + [39.31 色彩校正](#colorcorrect)
    - [39.31.1 命令](#Commands-67)
  + [39.32 彩色通道混合器](#colorchannelmixer)
    - [39.32.1 例子](#Examples-94)
    - [39.32.2 命令](#Commands-68)
  + [39.33 着色](#colorize)
    - [39.33.1 命令](#Commands-69)
  + [39.34 色键](#colorkey)
    - [39.34.1 例子](#Examples-95)
    - [39.34.2 命令](#Commands-70)
  + [39.35 色保持](#colorhold)
    - [39.35.1 命令](#Commands-71)
  + [39.36 色阶](#colorlevels)
    - [39.36.1 例子](#Examples-96)
    - [39.36.2 命令](#Commands-72)
  + [39.37 色图](#colormap)
  + [39.38 颜色矩阵](#colormatrix)
  + [39.39 色彩空间](#colorspace)
  + [39.40 colorspace\_cuda](#colorspace_005fcuda)
  + [39.41色温](#colortemperature)
    - [39.41.1 命令](#Commands-73)
  + [39.42卷积](#convolution)
    - [39.42.1 命令](#Commands-74)
    - [39.42.2 例子](#Examples-97)
  + [39.43卷积](#convolve)
  + [39.44 副本](#copy)
  + [39.45 核心图像](#coreimage-1)
    - [39.45.1 例子](#Examples-98)
  + [39.46 更正](#corr)
  + [39.47 cover\_rect](#cover_005frect)
    - [39.47.1 例子](#Examples-99)
  + [39.48 作物](#crop)
    - [39.48.1 例子](#Examples-100)
    - [39.48.2 命令](#Commands-75)
  + [39.49 cropdetect](#cropdetect)
    - [39.49.1 例子](#Examples-101)
    - [39.49.2 命令](#Commands-76)
  + [39.50 球杆](#cue-1)
  + [39.51 曲线](#curves-1)
    - [39.51.1 命令](#Commands-77)
    - [39.51.2 例子](#Examples-102)
  + [39.52 数据范围](#datascope)
    - [39.52.1 命令](#Commands-78)
  + [39.53 模糊](#dblur)
    - [39.53.1 命令](#Commands-79)
  + [39.54 dctdnoiz](#dctdnoiz)
    - [39.54.1 例子](#Examples-103)
  + [39.55 解带](#deband)
    - [39.55.1 命令](#Commands-80)
  + [39.56 解块](#deblock)
    - [39.56.1 例子](#Examples-104)
    - [39.56.2 命令](#Commands-81)
  + [39.57 抽取](#decimate-1)
  + [39.58 去卷积](#deconvolve)
  + [39.59分](#dedot)
  + [39.60放气](#deflate)
    - [39.60.1 命令](#Commands-82)
  + [39.61 去闪烁](#deflicker)
  + [39.62 抖动](#dejudder)
  + [39.63 标志](#delogo)
    - [39.63.1 例子](#Examples-105)
  + [39.64 脱雨](#derain-1)
  + [39.65 去抖](#deshake)
  + [39.66 倒溢](#despill)
    - [39.66.1 命令](#Commands-83)
  + [39.67 删除](#detelecine)
  + [39.68 膨胀](#dilation-1)
    - [39.68.1 命令](#Commands-84)
  + [39.69 排量](#displace)
    - [39.69.1 例子](#Examples-106)
  + [39.70 dnn\_classify](#dnn_005fclassify)
  + [39.71 dnn\_detect](#dnn_005fdetect)
  + [39.72 dnn\_processing](#dnn_005fprocessing-1)
    - [39.72.1 例子](#Examples-107)
  + [39.73 抽屉箱](#drawbox)
    - [39.73.1 例子](#Examples-108)
    - [39.73.2 命令](#Commands-85)
  + [39.74 绘图](#drawgraph-1)
  + [39.75 拉网](#drawgrid)
    - [39.75.1 例子](#Examples-109)
    - [39.75.2 命令](#Commands-86)
  + [39.76 绘图文本](#drawtext-1)
    - [39.76.1 语法](#Syntax-2)
    - [39.76.2 文本扩展](#Text-expansion)
    - [39.76.3 命令](#Commands-87)
    - [39.76.4 例子](#Examples-110)
  + [39.77 边缘检测](#edgedetect)
    - [39.77.1 例子](#Examples-111)
  + [39.78 磅](#elbg)
  + [39.79 熵](#entropy)
  + [39.80 epx](#epx)
  + [39.81当量](#eq)
    - [39.81.1 命令](#Commands-88)
  + [39.82侵蚀](#erosion-1)
    - [39.82.1 命令](#Commands-89)
  + [39.83 estdif](#estdif)
    - [39.83.1 命令](#Commands-90)
  + [39.84曝光](#exposure)
    - [39.84.1 命令](#Commands-91)
  + [39.85 提取平面](#extractplanes)
    - [39.85.1 例子](#Examples-112)
  + [39.86 褪色](#fade)
    - [39.86.1 例子](#Examples-113)
  + [39.87 反馈](#feedback)
    - [39.87.1 例子](#Examples-114)
  + [39.88 fftdnoiz](#fftdnoiz)
  + [39.89 fftfilt](#fftfilt)
    - [39.89.1 例子](#Examples-115)
  + [39.90场](#field)
  + [39.91 现场提示](#fieldhint)
  + [39.92 现场比赛](#fieldmatch)
    - [39.92.1 p/c/n/u/b 含义](#p_002fc_002fn_002fu_002fb-meaning-1)
      * [39.92.1.1 p/c/n](#p_002fc_002fn)
      * [39.92.1.2 u/b](#u_002fb)
    - [39.92.2 例子](#Examples-116)
  + [39.93 场序](#fieldorder)
  + [39.94 fifo，afifo](#fifo_002c-afifo)
  + [39.95 填充边框](#fillborders)
    - [39.95.1 命令](#Commands-92)
  + [39.96 查找矩形](#find_005frect)
    - [39.96.1 例子](#Examples-117)
  + [39.97 填洪](#floodfill)
  + [39.98格式](#format-1)
    - [39.98.1 例子](#Examples-118)
  + [39.99 帧/秒](#fps-1)
    - [39.99.1 例子](#Examples-119)
  + [39.100 帧包](#framepack)
  + [39.101 帧率](#framerate)
  + [39.102 帧步](#framestep)
  + [39.103 冻结检测](#freezedetect)
  + [39.104 定格画面](#freezeframes)
  + [39.105 frei0r](#frei0r-1)
    - [39.105.1 示例](#Examples-120)
    - [39.105.2 命令](#Commands-93)
  + [39.106 fspp](#fspp)
  + [39.107 模糊](#gblur)
    - [39.107.1 命令](#Commands-94)
  + [39.108 geq](#geq)
    - [39.108.1 示例](#Examples-121)
  + [39.109 gradfun](#gradfun)
    - [39.109.1 示例](#Examples-122)
  + [39.110 图形监视器](#graphmonitor-1)
  + [39.111 灰色世界](#grayworld)
  + [39.112 灰边](#greyedge)
    - [39.112.1 例子](#Examples-123)
  + [39.113引导](#guided)
    - [39.113.1 命令](#Commands-95)
    - [39.113.2 例子](#Examples-124)
  + [39.114 杂乱无章](#haldclut-1)
    - [39.114.1 命令](#Commands-96)
    - [39.114.2 工作流程示例](#Workflow-examples)
      * [39.114.2.1 暂停 CLUT 视频流](#Hald-CLUT-video-stream)
      * [39.114.2.2 暂停 CLUT 预览](#Hald-CLUT-with-preview)
  + [39.115 hflip](#hflip)
  + [39.116 希斯特克](#histeq)
  + [39.117 直方图](#histogram-1)
    - [39.117.1 示例](#Examples-125)
  + [39.118 hqdn3d](#hqdn3d-1)
    - [39.118.1 命令](#Commands-97)
  + [39.119 硬件下载](#hwdownload-1)
  + [39.120 硬件地图](#hwmap)
  + [39.121 hwupload](#hwupload-1)
  + [39.122 hwupload\_cuda](#hwupload_005fcuda-1)
  + [39.123 hqx](#hqx)
  + [39.124 堆栈](#hstack-1)
  + [39.125 hsvhold](#hsvhold)
  + [39.126 hsvkey](#hsvkey)
  + [39.127色相](#hue)
    - [39.127.1 示例](#Examples-126)
    - [39.127.2 命令](#Commands-98)
  + [39.128 色调饱和度](#huesaturation)
  + [39.129 迟滞](#hysteresis)
  + [39.130 iccdetect](#iccdetect)
  + [39.131 iccgen](#iccgen)
  + [39.132 身份](#identity)
  + [39.133 艾特](#idet)
  + [39.134 伊尔](#il)
    - [39.134.1 命令](#Commands-99)
  + [39.135 充气](#inflate)
    - [39.135.1 命令](#Commands-100)
  + [39.136 交错](#interlace)
  + [39.137 核心](#kerndeint)
    - [39.137.1 示例](#Examples-127)
  + [39.138 樱桃酒](#kirsch)
    - [39.138.1 命令](#Commands-101)
  + [39.139 拉格芬](#lagfun)
    - [39.139.1 命令](#Commands-102)
  + [39.140 镜头校正](#lenscorrection)
    - [39.140.1 选项](#Options-93)
    - [39.140.2 命令](#Commands-103)
  + [39.141 镜头乐趣](#lensfun)
    - [39.141.1 示例](#Examples-128)
  + [39.142 lib安慰剂](#libplacebo)
    - [39.142.1 选项](#Options-94)
      * [39.142.1.1 输出模式](#Output-mode)
      * [39.142.1.2 缩放](#Scaling)
      * [39.142.1.3 解带](#Debanding)
      * [39.142.1.4 颜色调整](#Color-adjustment)
      * [39.142.1.5 峰值检测](#Peak-detection)
      * [39.142.1.6 色调映射](#Tone-mapping)
      * [39.142.1.7 抖动](#Dithering)
      * [39.142.1.8 自定义着色器](#Custom-shaders)
      * [39.142.1.9 调试/性能](#Debugging-_002f-performance)
    - [39.142.2 命令](#Commands-104)
    - [39.142.3 例子](#Examples-129)
  + [39.143 libvmaf](#libvmaf)
    - [39.143.1 示例](#Examples-130)
  + [39.144 极限差异](#limitdiff)
    - [39.144.1 命令](#Commands-105)
  + [39.145 限制器](#limiter)
    - [39.145.1 命令](#Commands-106)
  + [39.146 循环](#loop)
    - [39.146.1 示例](#Examples-131)
  + [39.147 lut1d](#lut1d)
    - [39.147.1 命令](#Commands-107)
  + [39.148 lut3d](#lut3d-1)
    - [39.148.1 命令](#Commands-108)
  + [39.149 卢马基](#lumakey)
    - [39.149.1 命令](#Commands-109)
  + [39.150 lut，lutrgb，lutyuv](#lut_002c-lutrgb_002c-lutyuv)
    - [39.150.1 命令](#Commands-110)
    - [39.150.2 示例](#Examples-132)
  + [39.151 lut2，tlut2](#lut2_002c-tlut2)
    - [39.151.1 命令](#Commands-111)
    - [39.151.2 例子](#Examples-133)
  + [39.152 蒙面钳](#maskedclamp)
    - [39.152.1 命令](#Commands-112)
  + [39.153 蒙面最大](#maskedmax)
    - [39.153.1 命令](#Commands-113)
  + [39.154 蒙面合并](#maskedmerge)
    - [39.154.1 命令](#Commands-114)
  + [39.155 掩码最小值](#maskedmin)
    - [39.155.1 命令](#Commands-115)
  + [39.156 屏蔽阈值](#maskedthreshold)
    - [39.156.1 命令](#Commands-116)
  + [39.157 面具乐趣](#maskfun)
    - [39.157.1 命令](#Commands-117)
  + [39.158 麦克迪因特](#mcdeint)
  + [39.159 中位数](#median)
    - [39.159.1 命令](#Commands-118)
  + [39.160 合并平面](#mergeplanes)
    - [39.160.1 示例](#Examples-134)
  + [39.161 估计](#mestimate)
  + [39.162 中等均衡器](#midequalizer)
  + [39.163 最小插值](#minterpolate)
  + [39.164混合](#mix)
    - [39.164.1 命令](#Commands-119)
  + [39.165单色](#monochrome)
    - [39.165.1 命令](#Commands-120)
  + [39.166 吗啡](#morpho)
    - [39.166.1 命令](#Commands-121)
  + [39.167 mp 十进制](#mpdecimate)
  + [39.168 毫秒](#msad)
  + [39.169 相乘](#multiply)
    - [39.169.1 命令](#Commands-122)
  + [39.170 否定](#negate)
    - [39.170.1 命令](#Commands-123)
  + [39.171 均值](#nlmeans-1)
  + [39.172 尼迪](#nnedi)
    - [39.172.1 命令](#Commands-124)
  + [39.173 无格式](#noformat)
    - [39.173.1 示例](#Examples-135)
  + [39.174噪音](#noise-1)
    - [39.174.1 示例](#Examples-136)
  + [39.175 归一化](#normalize)
    - [39.175.1 命令](#Commands-125)
    - [39.175.2 示例](#Examples-137)
  + [39.176 无效](#null-2)
  + [39.177 光学字符识别](#ocr)
  + [39.178 超频](#ocv)
    - [39.178.1 膨胀](#dilate-1)
    - [39.178.2 侵蚀](#erode)
    - [39.178.3 光滑](#smooth)
  + [39.179 示波器](#oscilloscope)
    - [39.179.1 命令](#Commands-126)
    - [39.179.2 示例](#Examples-138)
  + [39.180 叠加](#overlay-1)
    - [39.180.1 命令](#Commands-127)
    - [39.180.2 示例](#Examples-139)
  + [39.181 overlay\_cuda](#overlay_005fcuda-1)
  + [39.182 低噪声](#owdenoise)
  + [39.183垫](#pad-1)
    - [39.183.1 示例](#Examples-140)
  + [39.184 调色板生成](#palettegen-1)
    - [39.184.1 示例](#Examples-141)
  + [39.185 调色板使用](#paletteuse)
    - [39.185.1 示例](#Examples-142)
  + [39.186透视](#perspective)
  + [39.187相](#phase)
    - [39.187.1 命令](#Commands-128)
  + [39.188 光敏性](#photosensitivity)
  + [39.189 pixdesctest](#pixdesctest)
  + [39.190 像素化](#pixelize)
    - [39.190.1 命令](#Commands-129)
  + [39.191像素范围](#pixscope)
    - [39.191.1 命令](#Commands-130)
  + [39.192 页](#pp)
    - [39.192.1 示例](#Examples-143)
  + [39.193 页 7](#pp7)
  + [39.194 预乘](#premultiply)
  + [39.195 普雷维特](#prewitt)
    - [39.195.1 命令](#Commands-131)
  + [39.196伪彩](#pseudocolor)
    - [39.196.1 命令](#Commands-132)
    - [39.196.2 示例](#Examples-144)
  + [39.197 伪信噪比](#psnr)
    - [39.197.1 示例](#Examples-145)
  + [39.198 上拉](#pullup-1)
  + [39.199 磅](#qp)
    - [39.199.1 示例](#Examples-146)
  + [39.200 随机](#random)
  + [39.201 readeia608](#readeia608)
    - [39.201.1 命令](#Commands-133)
    - [39.201.2 示例](#Examples-147)
  + [39.202 读维茨](#readvitc)
    - [39.202.1 示例](#Examples-148)
  + [39.203 重新映射](#remap)
  + [39.204 去除颗粒](#removegrain)
  + [39.205 删除徽标](#removelogo)
  + [39.206 重复字段](#repeatfields)
  + [39.207反向](#reverse)
    - [39.207.1 示例](#Examples-149)
  + [39.208 rgbashift](#rgbashift)
    - [39.208.1 命令](#Commands-134)
  + [39.209 罗伯茨](#roberts)
    - [39.209.1 命令](#Commands-135)
  + [39.210 旋转](#rotate)
    - [39.210.1 例子](#Examples-150)
    - [39.210.2 命令](#Commands-136)
  + [39.211 萨布](#sab)
  + [39.212规模](#scale-1)
    - [39.212.1 选项](#Options-95)
    - [39.212.2 例子](#Examples-151)
    - [39.212.3 命令](#Commands-137)
  + [39.213 scale\_cuda](#scale_005fcuda)
    - [39.213.1 例子](#Examples-152)
  + [39.214 scale\_npp](#scale_005fnpp-1)
  + [39.215 scale2ref](#scale2ref)
    - [39.215.1 例子](#Examples-153)
    - [39.215.2 命令](#Commands-138)
  + [39.216 scale2ref\_npp](#scale2ref_005fnpp)
    - [39.216.1 示例](#Examples-154)
  + [39.217 夏尔](#scharr)
    - [39.217.1 命令](#Commands-139)
  + [39.218 卷轴](#scroll)
    - [39.218.1 命令](#Commands-140)
  + [39.219 scdet](#scdet-1)
  + [39.220 选择性颜色](#selectivecolor-1)
    - [39.220.1 示例](#Examples-155)
  + [39.221 个单独的字段](#separatefields-1)
  + [39.222 setdar, 塞萨尔](#setdar_002c-setsar)
    - [39.222.1 例子](#Examples-156)
  + [39.223 设定场](#setfield-1)
  + [39.224 设置参数](#setparams-1)
  + [39.225 锐化\_npp](#sharpen_005fnpp)
  + [39.226 剪切](#shear)
    - [39.226.1 命令](#Commands-141)
  + [39.227 显示信息](#showinfo)
  + [39.228 显示调色板](#showpalette)
  + [39.229 随机帧](#shuffleframes)
    - [39.229.1 示例](#Examples-157)
  + [39.230 随机像素](#shufflepixels)
  + [39.231 shuffleplanes](#shuffleplanes)
    - [39.231.1 例子](#Examples-158)
  + [39.232 信号统计](#signalstats-1)
    - [39.232.1 例子](#Examples-159)
  + [39.233 签名](#signature-1)
    - [39.233.1 示例](#Examples-160)
  + [39.234 西蒂](#siti-1)
    - [39.234.1 示例](#Examples-161)
  + [39.235 智能模糊](#smartblur-1)
  + [39.236索贝尔](#sobel)
    - [39.236.1 命令](#Commands-142)
  + [39.237 人](#spp-1)
    - [39.237.1 命令](#Commands-143)
  + [39.238 高级](#sr-1)
  + [39.239 西姆卡](#ssim)
    - [39.239.1 例子](#Examples-162)
  + [39.240 stereo3d](#stereo3d)
    - [39.240.1 示例](#Examples-163)
  + [39.241 流选择，流选择](#streamselect_002c-astreamselect)
    - [39.241.1 命令](#Commands-144)
    - [39.241.2 例子](#Examples-164)
  + [39.242 字幕](#subtitles-1)
  + [39.243 super2xsai](#super2xsai)
  + [39.244 交换协议](#swaprect)
    - [39.244.1 命令](#Commands-145)
  + [39.245 swapuv](#swapuv)
  + [39.246 汤混合](#tblend)
  + [39.247 电视电影](#telecine)
  + [39.248 这个图](#thistogram)
  + [39.249 门槛](#threshold)
    - [39.249.1 命令](#Commands-146)
    - [39.249.2 示例](#Examples-165)
  + [39.250 缩略图](#thumbnail)
    - [39.250.1 示例](#Examples-166)
  + [39.251瓦](#tile-1)
    - [39.251.1 例子](#Examples-167)
  + [39.252 交织](#tinterlace)
  + [39.253 tmedian](#tmedian)
    - [39.253.1 命令](#Commands-147)
  + [39.254 tm均衡器](#tmidequalizer)
  + [39.255 混音](#tmix)
    - [39.255.1 示例](#Examples-168)
    - [39.255.2 命令](#Commands-148)
  + [39.256 色调映射](#tonemap-1)
    - [39.256.1 选项](#Options-96)
  + [39.257 吨](#tpad)
  + [39.258转置](#transpose-1)
  + [39.259 transpose\_npp](#transpose_005fnpp)
  + [39.260 修剪](#trim)
  + [39.261 未预乘](#unpremultiply)
  + [39.262 锐化](#unsharp-1)
    - [39.262.1 示例](#Examples-169)
  + [39.263 直到](#untile-1)
    - [39.263.1 示例](#Examples-170)
  + [39.264 美元](#uspp)
  + [39.265 v360](#v360)
    - [39.265.1 示例](#Examples-171)
    - [39.265.2 命令](#Commands-149)
  + [39.266 模糊降噪器](#vaguedenoiser)
  + [39.267 可变模糊](#varblur)
    - [39.267.1 命令](#Commands-150)
  + [39.268 矢量示波器](#vectorscope)
  + [39.269 vidstabdetect](#vidstabdetect-1)
    - [39.269.1 示例](#Examples-172)
  + [39.270 vidstab变换](#vidstabtransform-1)
    - [39.270.1 选项](#Options-97)
    - [39.270.2 示例](#Examples-173)
  + [39.271 翻转](#vflip)
  + [39.272 vfrdet](#vfrdet)
  + [39.273 活力](#vibrance)
    - [39.273.1 命令](#Commands-151)
  + [39.274 维夫](#vif)
  + [39.275 小插图](#vignette-1)
    - [39.275.1 表达式](#Expressions)
    - [39.275.2 示例](#Examples-174)
  + [39.276 vmaf运动](#vmafmotion)
  + [39.277 vstack](#vstack-1)
  + [39.278 w3fdif](#w3fdif)
    - [39.278.1 命令](#Commands-152)
  + [39.279 波形](#waveform)
  + [39.280 编织，双面编织](#weave_002c-doubleweave)
    - [39.280.1 例子](#Examples-175)
  + [39.281 xbr](#xbr)
  + [39.282 x相关](#xcorrelate)
  + [39.283 x淡入淡出](#xfade)
    - [39.283.1 例子](#Examples-176)
  + [39.284 x 中值](#xmedian)
    - [39.284.1 命令](#Commands-153)
  + [39.285 x堆栈](#xstack-1)
    - [39.285.1 示例](#Examples-177)
  + [39.286 亚迪夫](#yadif-1)
  + [39.287 yadif\_cuda](#yadif_005fcuda)
  + [39.288 亚普布勒](#yaepblur)
    - [39.288.1 命令](#Commands-154)
  + [39.289 缩放平移](#zoompan)
    - [39.289.1 例子](#Examples-178)
  + [39.290 zscale](#zscale-1)
    - [39.290.1 选项](#Options-98)
    - [39.290.2 命令](#Commands-155)
* [40 个 OpenCL 视频滤波器](#OpenCL-Video-Filters)
  + [40.1 avgblur\_opencl](#avgblur_005fopencl)
    - [40.1.1 示例](#Example-4)
  + [40.2 boxblur\_opencl](#boxblur_005fopencl)
    - [40.2.1 例子](#Examples-179)
  + [40.3 colorkey\_opencl](#colorkey_005fopencl)
    - [40.3.1 例子](#Examples-180)
  + [40.4 卷积\_opencl](#convolution_005fopencl)
    - [40.4.1 例子](#Examples-181)
  + [40.5 侵蚀\_opencl](#erosion_005fopencl)
    - [40.5.1 例子](#Example-5)
  + [40.6 deshake\_opencl](#deshake_005fopencl)
    - [40.6.1 例子](#Examples-182)
  + [40.7 dilation\_opencl](#dilation_005fopencl)
    - [40.7.1 例子](#Example-6)
  + [40.8 nlmeans\_opencl](#nlmeans_005fopencl)
  + [40.9 overlay\_opencl](#overlay_005fopencl)
    - [40.9.1 例子](#Examples-183)
  + [40.10 pad\_opencl](#pad_005fopencl)
  + [40.11 prewitt\_opencl](#prewitt_005fopencl)
    - [40.11.1 示例](#Example-7)
  + [40.12 程序\_opencl](#program_005fopencl-1)
  + [40.13 重映射\_opencl](#remap_005fopencl)
  + [40.14 罗伯茨\_opencl](#roberts_005fopencl)
    - [40.14.1 例子](#Example-8)
  + [40.15 sobel\_opencl](#sobel_005fopencl)
    - [40.15.1 例子](#Example-9)
  + [40.16 色调映射\_opencl](#tonemap_005fopencl)
    - [40.16.1 示例](#Example-10)
  + [40.17 unsharp\_opencl](#unsharp_005fopencl)
    - [40.17.1 例子](#Examples-184)
  + [40.18 xfade\_opencl](#xfade_005fopencl)
* [41 个 VAAPI 视频过滤器](#VAAPI-Video-Filters)
  + [41.1 overlay\_vaapi](#overlay_005fvaapi)
    - [41.1.1 例子](#Examples-185)
  + [41.2 色调映射\_vaapi](#tonemap_005fvaapi)
    - [41.2.1 示例](#Example-11)
  + [41.3 hstack\_vaapi](#hstack_005fvaapi)
  + [41.4 vstack\_vaapi](#vstack_005fvaapi)
  + [41.5 xstack\_vaapi](#xstack_005fvaapi)
* [42 个 QSV 视频滤波器](#QSV-Video-Filters)
  + [42.1 hstack\_qsv](#hstack_005fqsv)
  + [42.2 vstack\_qsv](#vstack_005fqsv)
  + [42.3 xstack\_qsv](#xstack_005fqsv)
* [43个视频源](#Video-Sources)
  + [43.1缓冲区](#buffer)
  + [43.2 大提琴](#cellauto)
    - [43.2.1 例子](#Examples-186)
  + [43.3 coreimagesrc](#coreimagesrc-1)
    - [43.3.1 例子](#Examples-187)
  + [43.4 ddagrab](#ddagrab)
    - [43.4.1 例子](#Examples-188)
  + [43.5 渐变](#gradients)
  + [43.6 曼德尔布罗](#mandelbrot)
  + [43.7 mptestsrc](#mptestsrc)
  + [43.8 frei0r\_src](#frei0r_005fsrc)
  + [43.9生命](#life)
    - [43.9.1 例子](#Examples-189)
  + [43.10 allrgb、allyuv、颜色、色图、色谱、haldclutsrc、nullsrc、pal75bars、pal100bars、rgbtestsrc、smptebars、smptehdbars、testsrc、testsrc2、yuvtestsrc](#allrgb_002c-allyuv_002c-color_002c-colo)
    - [43.10.1 例子](#Examples-190)
    - [43.10.2 命令](#Commands-156)
  + [43.11 openclsrc](#openclsrc)
  + [43.12谢尔宾斯基](#sierpinski)
* [44个视频接收器](#Video-Sinks)
  + [44.1 缓冲区接收器](#buffersink)
  + [44.2 空槽](#nullsink)
* [45个多媒体过滤器](#Multimedia-Filters)
  + [45.1 a3dscope](#a3dscope)
    - [45.1.1 命令](#Commands-157)
  + [45.2 位域](#abitscope)
  + [45.3 绘图](#adrawgraph)
  + [45.4 图形监视器](#agraphmonitor)
  + [45.5 直方图](#ahistogram)
  + [45.6相位计](#aphasemeter)
    - [45.6.1 相位检测](#phasing-detection)
    - [45.6.2 例子](#Examples-191)
  + [45.7 矢量图](#avectorscope)
    - [45.7.1 例子](#Examples-192)
    - [45.7.2 命令](#Commands-158)
  + [45.8 板凳，板凳](#bench_002c-abench)
    - [45.8.1 例子](#Examples-193)
  + [45.9 连续](#concat-3)
    - [45.9.1 例子](#Examples-194)
    - [45.9.2 命令](#Commands-159)
  + [45.10 ebur128](#ebur128-1)
    - [45.10.1 例子](#Examples-195)
  + [45.11交错，交错](#interleave_002c-ainterleave)
    - [45.11.1 例子](#Examples-196)
  + [45.12 潜伏期，潜伏期](#latency_002c-alatency)
  + [45.13 元数据，元数据](#metadata_002c-ametadata)
    - [45.13.1 例子](#Examples-197)
  + [45.14 烫发，烫发](#perms_002c-aperms)
  + [45.15 实时，实时](#realtime_002c-arealtime)
    - [45.15.1 命令](#Commands-160)
  + [45.16段，段](#segment_002c-asegment)
    - [45.16.1 例子](#Examples-198)
  + [45.17 选择，选择](#select_002c-aselect)
    - [45.17.1 例子](#Examples-199)
  + [45.18 发送命令，发送命令](#sendcmd_002c-asendcmd)
    - [45.18.1 命令语法](#Commands-syntax)
    - [45.18.2 例子](#Examples-200)
  + [45.19 setpts，asetpts](#setpts_002c-asetpts)
    - [45.19.1 例子](#Examples-201)
  + [45.20 设定范围](#setrange)
  + [45.21 settb, asettb](#settb_002c-asettb)
    - [45.21.1 例子](#Examples-202)
  + [45.22 showcqt](#showcqt)
    - [45.22.1 例子](#Examples-203)
  + [45.23 showcwt](#showcwt)
  + [45.24 显示频率](#showfreqs)
  + [45.25 显示空间](#showspatial)
  + [45.26 显示频谱](#showspectrum-1)
    - [45.26.1 例子](#Examples-204)
  + [45.27 显示光谱](#showspectrumpic)
    - [45.27.1 例子](#Examples-205)
  + [45.28 showvolume](#showvolume)
  + [45.29 秀波](#showwaves)
    - [45.29.1 例子](#Examples-206)
  + [45.30 showwavespic](#showwavespic)
    - [45.30.1 例子](#Examples-207)
  + [45.31 边数据，asidedata](#sidedata_002c-asidedata)
  + [45.32 频谱合成器](#spectrumsynth)
    - [45.32.1 例子](#Examples-208)
  + [45.33 分裂，分裂](#split_002c-asplit)
    - [45.33.1 例子](#Examples-209)
  + [45.34 zmq，azmq](#zmq_002c-azmq)
    - [45.34.1 例子](#Examples-210)
* [46个多媒体资源](#Multimedia-Sources)
  + [46.1 电影](#amovie)
  + [46.2 异步测试](#avsynctest)
  + [46.3电影](#movie-1)
    - [46.3.1 例子](#Examples-211)
    - [46.3.2 命令](#Commands-161)
* [47 个外部图书馆](#External-libraries)
  + [47.1 开放媒体联盟 (AOM)](#Alliance-for-Open-Media-_0028AOM_0029)
  + [47.2 AMD AMF/VCE](#AMD-AMF_002fVCE)
  + [47.3 Avi合成器](#AviSynth)
  + [47.4 色谱图](#Chromaprint)
  + [47.5编解码器2](#codec2)
  + [47.6 dav1d](#dav1d)
  + [47.7 戴维斯 2](#davs2)
  + [47.8 无人机 3d](#uavs3d)
  + [47.9游戏音乐鸸鹋](#Game-Music-Emu)
  + [47.10 Intel QuickSync 视频](#Intel-QuickSync-Video)
  + [47.11 克瓦扎尔](#Kvazaar)
  + [47.12跛脚](#LAME)
  + [47.13 libilbc](#libilbc-1)
  + [47.14 libjxl](#libjxl-1)
  + [47.15 libvpx](#libvpx-1)
  + [47.16 调制插件](#ModPlug)
  + [47.17 OpenCORE、VisualOn 和 Fraunhofer 库](#OpenCORE_002c-VisualOn_002c-and-Fraunho)
    - [47.17.1 开放核心 AMR](#OpenCORE-AMR)
    - [47.17.2 VisualOn AMR-WB 编码器库](#VisualOn-AMR_002dWB-encoder-library)
    - [47.17.3 弗劳恩霍夫 AAC 库](#Fraunhofer-AAC-library)
  + [47.18 OpenH264](#OpenH264)
  + [47.19 OpenJPEG](#OpenJPEG)
  + [47.20 rav1e](#rav1e)
  + [47.21 SVT-AV1](#SVT_002dAV1)
  + [47.22 TwoLAME](#TwoLAME)
  + [47.23 VapourSynth](#VapourSynth)
  + [47.24 x264](#x264)
  + [47.25 x265](#x265)
  + [47.26 xavs](#xavs)
  + [47.27 xavs2](#xavs2)
  + [47.28 ZVBI](#ZVBI)
* [48 种支持的文件格式、编解码器或功能](#Supported-File-Formats_002c-Codecs-or-F)
  + [48.1 文件格式](#File-Formats)
  + [48.2 图像格式](#Image-Formats)
  + [48.3 视频编解码器](#Video-Codecs)
  + [48.4 音频编解码器](#Audio-Codecs)
  + [48.5 字幕格式](#Subtitle-Formats)
  + [48.6 网络协议](#Network-Protocols)
  + [48.7 输入/输出设备](#Input_002fOutput-Devices)
  + [48.8 时间码](#Timecode)
* [49 另见](#See-Also)
* [50位作者](#Authors)

## 1 概要

ffmpeg [ global\_options ] {[ input\_file\_options ] -i input\_url } ... {[ output\_file\_options ] output\_url } ...

## 2 说明

ffmpeg是一种非常快速的视频和音频转换器，也可以从实时音频/视频源中获取。它还可以在任意采样率之间进行转换，并使用高质量的多相滤波器动态调整视频大小。

ffmpeg从-i选项指定的任意数量的输入“文件”（可以是常规文件、管道、网络流、抓取设备等）中读取，并写入任意数量的输出“文件”，这由普通输出 url 指定。在命令行上找到的任何不能解释为选项的内容都被视为输出 url。

原则上，每个输入或输出 url 可以包含任意数量的不同类型的流（视频/音频/字幕/附件/数据）。允许的流数量和/或类型可能受容器格式限制。选择哪些流从哪些输入进入哪些输出是自动完成的或使用-map选项（请参阅流选择章节）。

要在选项中引用输入文件，您必须使用它们的索引（从 0 开始）。例如，第一个输入文件是0 ，第二个是1 ，等等。类似地，文件中的流由它们的索引引用。例如2:3指的是第三个输入文件中的第四个流。另请参阅流说明符一章。

作为一般规则，选项应用于下一个指定的文件。因此，顺序很重要，您可以多次在命令行中使用相同的选项。然后将每次出现应用于下一个输入或输出文件。这条规则的例外是全局选项（例如冗长级别），它应该首先指定。

不要混合输入和输出文件——首先指定所有输入文件，然后是所有输出文件。也不要混合属于不同文件的选项。所有选项仅适用于下一个输入或输出文件，并在文件之间重置。

* 要将输出文件的视频比特率设置为 64 kbit/s：
* ffmpeg -i 输入.avi -b:v 64k -bufsize 64k 输出.avi
* 要将输出文件的帧速率强制为 24 fps：
* ffmpeg -i 输入.avi -r 24 输出.avi
* 要将输入文件的帧速率（仅对原始格式有效）强制为 1 fps，将输出文件的帧速率强制为 24 fps：
* ffmpeg -r 1 -i 输入.m2v -r 24 输出.avi

原始输入文件可能需要格式选项。

## 3 详细说明

ffmpeg中对每一个输出的转码过程可以用下图来描述：

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| | | |

|输入 |解复用器 |编码数据 |解码器

|文件 | ----------> |数据包 | -----+

|\_\_\_\_\_\_\_| |\_\_\_\_\_\_\_\_\_\_\_\_| |

v

\_\_\_\_\_\_\_\_\_\_

| |

|解码 |

|相框 |

|\_\_\_\_\_\_\_\_\_|

\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ | |

| | | | |

|输出 | <-------- |编码数据 | <----+

|文件 |混合器 |数据包 |编码器

|\_\_\_\_\_\_\_\_| |\_\_\_\_\_\_\_\_\_\_\_\_|

ffmpeg调用 libavformat 库（包含解复用器）来读取输入文件并从中获取包含编码数据的数据包。当有多个输入文件时， ffmpeg会尝试通过跟踪任何活动输入流上的最低时间戳来使它们保持同步。

然后将编码的数据包传递给解码器（除非为流选择了 streamcopy，请参阅进一步的说明）。解码器产生未压缩的帧（原始视频/PCM 音频/...），可以通过过滤进一步处理（见下一节）。过滤后，帧被传递给编码器，编码器对它们进行编码并输出编码数据包。最后，这些被传递给 muxer，它将编码的数据包写入输出文件。

* [过滤](#Filtering)
* [流复制](#Stream-copy)

### 3.1 过滤

在编码之前， ffmpeg可以使用 libavfilter 库中的过滤器处理原始音频和视频帧。几个链接的过滤器形成一个过滤器图。 ffmpeg区分两种类型的过滤图：简单的和复杂的。

* [简单的过滤图](#Simple-filtergraphs)
* [复杂的过滤图](#Complex-filtergraphs)

#### 3.1.1 简单的过滤图

简单的过滤图是那些只有一个输入和输出，并且类型相同的过滤图。在上图中，它们可以通过简单地在解码和编码之间插入一个额外的步骤来表示：

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| | | |

|解码 | |编码数据 |

|框架 |\ \_ |数据包 |

|\_\_\_\_\_\_\_\_\_| \ /||\_\_\_\_\_\_\_\_\_\_\_\_|

\ \_\_\_\_\_\_\_\_\_\_ /

简单\_\|| | /编码器

过滤图 |过滤 |/

|相框 |

|\_\_\_\_\_\_\_\_\_\_|

简单的过滤器图配置了每个流的-filter选项（分别为视频和音频使用-vf和-af别名）。一个简单的视频过滤器图可以看起来像这样：

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

| | | | | | | |

|输入 | ---> |去隔行 | ---> |规模 | ---> |输出 |

|\_\_\_\_\_\_\_| |\_\_\_\_\_\_\_\_\_\_\_\_\_\_| |\_\_\_\_\_\_\_| |\_\_\_\_\_\_\_\_|

请注意，某些过滤器会更改框架属性但不会更改框架内容。例如，上面示例中的fps过滤器会更改帧数，但不会触及帧内容。另一个例子是setpts过滤器，它只设置时间戳，否则不改变地传递帧。

#### 3.1.2 复杂的过滤图

复杂的过滤器图是那些不能简单地描述为应用于一个流的线性处理链的过滤器图。例如，当图形具有多个输入和/或输出时，或者输出流类型与输入不同时，就是这种情况。它们可以用下图表示：

\_\_\_\_\_\_\_\_\_\_

| |

|输入 0 |\ \_\_\_\_\_\_\_\_\_\_

|\_\_\_\_\_\_\_\_\_| \ | |

\ \_\_\_\_\_\_\_\_\_\_ /|输出 0 |

\ | | / |\_\_\_\_\_\_\_\_\_\_|

\_\_\_\_\_\_\_\_\_ \|复杂 | /

| | | |/

|输入 1 |---->|过滤器 |\

|\_\_\_\_\_\_\_\_\_| | | \ \_\_\_\_\_\_\_\_\_\_

/|图 | \ | |

/ | | \|输出 1 |

\_\_\_\_\_\_\_\_\_ / |\_\_\_\_\_\_\_\_\_| |\_\_\_\_\_\_\_\_\_\_|

| | /

|输入 2 |/

|\_\_\_\_\_\_\_\_\_|

复杂的过滤图配置有-filter\_complex选项。请注意，此选项是全局的，因为就其性质而言，复杂的过滤图不能明确地与单个流或文件相关联。

-lavfi选项等同于-filter\_complex 。

复杂滤镜图的一个简单示例是叠加滤镜，它有两个视频输入和一个视频输出，其中一个视频叠加在另一个视频之上。它的音频对应物是amix过滤器。

### 3.2 流复制

向-codec选项提供复制参数来选择的模式。它使ffmpeg省略指定流的解码和编码步骤，因此它只进行解复用和复用。它对于更改容器格式或修改容器级元数据很有用。在这种情况下，上图将简化为：

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

| | | | | |

|输入 |解复用器 |编码数据 |混合器 |输出 |

|文件 | ----------> |数据包 | ------> |文件 |

|\_\_\_\_\_\_\_| |\_\_\_\_\_\_\_\_\_\_\_\_| |\_\_\_\_\_\_\_\_|

由于没有解码或编码，所以速度非常快，没有质量损失。但是，由于许多因素，它在某些情况下可能不起作用。应用过滤器显然也是不可能的，因为过滤器对未压缩的数据起作用。

## 4 码流选择

ffmpeg提供-map选项用于手动控制每个输出文件中的流选择。用户可以跳过-map并让 ffmpeg 执行自动流选择，如下所述。 -vn / -an / -sn / -dn选项可用于分别跳过视频、音频、字幕和数据流的包含，无论是手动映射还是自动选择，除了那些作为复杂过滤图输出的流。

* [描述](#Description-1)
* [例子](#Examples)

### 4.1 说明

下面的小节描述了流选择中涉及的各种规则。接下来的例子展示了这些规则是如何在实践中应用的。

虽然已尽一切努力准确反映程序的行为，但 FFmpeg 仍在不断开发中，自撰写本文以来代码可能已更改。

* [自动流选择](#Automatic-stream-selection)
* [手动流选择](#Manual-stream-selection)
* [复杂的过滤图](#Complex-filtergraphs-1)
* [流处理](#Stream-handling)

#### 4.1.1 自动码流选择

在没有特定输出文件的任何映射选项的情况下，ffmpeg 检查输出格式以检查其中可以包含哪种类型的流，即。视频、音频和/或字幕。对于每种可接受的流类型，ffmpeg 将从所有输入中选择一个流（如果可用）。

它将根据以下标准选择该流：

* 对于视频，它是具有最高分辨率的流，
* 对于音频，它是通道最多的流，
* 对于字幕，这是找到的第一个字幕流，但有一个警告。输出格式的默认字幕编码器可以是基于文本的或基于图像的，并且只会选择相同类型的字幕流。

在相同类型的多个流速率相同的情况下，选择具有最低索引的流。

不会自动选择数据或附件流，只能使用-map将其包含在内。

#### 4.1.2 手动流选择

当使用-map时，只有用户映射的流包含在该输出文件中，过滤器图输出的一个可能例外如下所述。

#### 4.1.3 复杂的过滤图

如果有任何复杂的 filtergraph 输出流带有未标记的 pad，它们将被添加到第一个输出文件中。如果输出格式不支持流类型，这将导致致命错误。在没有 map 选项的情况下，包含这些流会导致跳过其类型的自动流选择。如果存在映射选项，则除了映射流之外，还包括这些过滤图流。

带有标记垫的复杂过滤器图输出流必须映射一次且恰好一次。

#### 4.1.4 流处理

流处理独立于流选择，但下文所述的字幕除外。流处理是通过-codec选项设置的，该选项针对特定输出文件中的流。特别是，编解码器选项在流选择过程之后由 ffmpeg 应用，因此不会影响后者。如果没有为流类型指定-codec选项，ffmpeg 将选择输出文件 muxer 注册的默认编码器。

字幕除外。如果为输出文件指定了字幕编码器，则将包含找到的任何类型（文本或图像）的第一个字幕流。 ffmpeg 不验证指定的编码器是否可以转换选定的流，或者转换后的流在输出格式中是否可接受。这通常也适用：当用户手动设置编码器时，流选择过程无法检查编码流是否可以混合到输出文件中。如果不能，ffmpeg 将中止并且所有输出文件将无法处理。

### 4.2 例子

以下示例说明了 ffmpeg 的流选择方法的行为、怪癖和限制。

他们假定以下三个输入文件。

输入文件'A.avi'

流 0：视频 640x360

流 1：音频 2 通道

输入文件'B.mp4'

流 0：视频 1920x1080

流 1：音频 2 通道

流 2：字幕（文本）

流 3：音频 5.1 声道

流 4：字幕（文本）

输入文件'C.mkv'

流 0：视频 1280x720

流 1：音频 2 通道

流 2：字幕（图像）

#### 示例：自动流选择

ffmpeg -i A.avi -i B.mp4 out1.mkv out2.wav -map 1:a -c:a copy out3.mov

指定了三个输出文件，前两个没有设置-map选项，因此 ffmpeg 会自动为这两个文件选择流。

out1.mkv是 Matroska 容器文件并接受视频、音频和字幕流，因此 ffmpeg 将尝试从每种类型中选择一种。对于视频，它将  
从B.mp4中选择流 0 ，它在所有输入视频流中具有最高分辨率。  
对于音频，它将从B.mp4中选择流 3 ，因为它具有最多的通道数。  
对于字幕，它将从B.mp4中选择流 2 ，这是A.avi和B.mp4中的第一个字幕流。

out2.wav仅接受音频流，因此仅选择来自B.mp4 的流 3 。

对于out3.mov ，由于设置了-map选项，因此不会发生自动流选择。 -map 1:a选项将从第二个输入B.mp4中选择所有音频流。此输出文件中不会包含其他流。

对于前两个输出，所有包含的流都将被转码。选择的编码器将是每个输出格式注册的默认编码器，它可能与所选输入流的编解码器不匹配。

对于第三个输出，音频流的编解码器选项已设置为copy ，因此不会发生或可能发生解码-过滤-编码操作。所选流的数据包应从输入文件中传送并在输出文件中进行多路复用。

#### 示例：自动字幕选择

ffmpeg -i C.mkv out1.mkv -c:s dvdsub -an out2.mkv

虽然out1.mkv是接受字幕流的 Matroska 容器文件，但只能选择视频和音频流。 C.mkv的字幕流是基于图像的，而 Matroska 混合器的默认字幕编码器是基于文本的，因此字幕的转码操作预计会失败，因此不会选择流。但是，在out2.mkv中，在命令中指定了字幕编码器，因此除了视频流之外，还选择了字幕流。 -an的存在禁用out2.mkv的音频流选择。

#### 示例：未标记的过滤图输出

ffmpeg -i A.avi -i C.mkv -i B.mp4 -filter\_complex "overlay" out1.mp4 out2.srt

-filter\_complex选项设置滤镜图，由单个视频滤镜组成。叠加过滤器正好需要两个视频输入，但没有指定，因此使用前两个可用的视频流，即A.avi和C.mkv的视频流。过滤器的输出垫没有标签，因此被发送到第一个输出文件out1.mp4 。因此，将跳过视频流的自动选择，这将选择B.mp4中的流。大多数频道的音频流即。 B.mp4中的流 3是自动选择的。但是没有选择字幕流，因为 MP4 格式没有注册默认的字幕编码器，并且用户没有指定字幕编码器。

第二个输出文件out2.srt只接受基于文本的字幕流。因此，即使第一个可用的字幕流属于C.mkv ，它也是基于图像的，因此会被跳过。所选流，即B.mp4中的流 2 ，是第一个基于文本的字幕流。

#### 示例：标记的过滤图输出

ffmpeg -i A.avi -i B.mp4 -i C.mkv -filter\_complex “[1:v]hue=s=0[outv];overlay;aresample”\

-map '[outv]' -an out1.mp4 \

out2.mkv \

-map '[outv]' -map 1:a:0 out3.mkv

[outv]的输出焊盘已被映射两次。不应处理任何输出文件。

ffmpeg -i A.avi -i B.mp4 -i C.mkv -filter\_complex “[1:v]hue=s=0[outv];overlay;aresample”\

-一个 out1.mp4 \

out2.mkv \

-地图 1:a:0 out3.mkv

上面的这个命令也会失败，因为色调过滤器输出有一个标签， [outv] ，并且没有被映射到任何地方。

该命令应修改如下，

ffmpeg -i A.avi -i B.mp4 -i C.mkv -filter\_complex “[1:v]hue=s=0,split=2[outv1][outv2];overlay;aresample”\

-map '[outv1]' -an out1.mp4 \

out2.mkv \

-map '[outv2]' -map 1:a:0 out3.mkv

B.mp4的视频流被发送到 hue 过滤器，其输出使用 split 过滤器被克隆一次，并且两个输出都被标记。然后将每个副本映射到第一个和第三个输出文件。

覆盖过滤器需要两个视频输入，使用前两个未使用的视频流。这些是来自A.avi和C.mkv的流。覆盖输出没有标记，因此它被发送到第一个输出文件out1.mp4 ，而不管-map选项是否存在。

aresample 过滤器发送第一个未使用的音频流，即A.avi的音频流。由于此过滤器输出也未标记，因此它也被映射到第一个输出文件。 -an的存在只会抑制音频流的自动或手动流选择，而不是从过滤器图发送的输出。这两个映射流都应在out1.mp4中的映射流之前排序。

out2.mkv的视频、音频和字幕流完全由自动选择流决定。

out3.mkv包含来自 hue 过滤器的克隆视频输出和来自B.mp4的第一个音频流。

## 5个选项

如果没有另外指定，所有数字选项都接受表示数字的字符串作为输入，该字符串后面可以跟一个 SI 单位前缀，例如：'K'、'M' 或 'G'。

如果“i”附加到 SI 单位前缀，则完整前缀将被解释为二进制倍数的单位前缀，二进制倍数基于 1024 的幂而不是 1000 的幂。将“B”附加到 SI 单位前缀会乘以值减 8。这允许使用，例如：“KB”、“MiB”、“G”和“B”作为数字后缀。

不带参数的选项是布尔选项，并将相应的值设置为 true。可以通过在选项名称前加上“no”来将它们设置为 false。例如，使用“-nofoo”会将名称为“foo”的布尔选项设置为 false。

* [流说明符](#Stream-specifiers-1)
* [通用选项](#Generic-options)
* [AV选项](#AVOptions)
* [主要选项](#Main-options)
* [视频选项](#Video-Options)
* [高级视频选项](#Advanced-Video-options)
* [音频选项](#Audio-Options)
* [高级音频选项](#Advanced-Audio-options)
* [字幕选项](#Subtitle-options)
* [高级字幕选项](#Advanced-Subtitle-options)
* [高级选项](#Advanced-options)
* [预设文件](#Preset-files)

### 5.1 流说明符

某些选项适用于每个流，例如比特率或编解码器。流说明符用于精确指定给定选项属于哪个流。

流说明符是一个字符串，通常附加到选项名称并由冒号分隔。例如-codec:a:1 ac3包含a:1流说明符，它匹配第二个音频流。因此，它会为第二个音频流选择 ac3 编解码器。

流说明符可以匹配多个流，因此该选项适用于所有流。例如， -b:a 128k中的流说明符匹配所有音频流。

空流说明符匹配所有流。例如， -codec copy或-codec:copy将复制所有流而不重新编码。

流说明符的可能形式是：

流索引

将流与该索引匹配。例如-threads:1 4会将第二个流的线程数设置为 4。如果stream\_index用作附加流说明符（见下文），则它会从匹配的流中选择流编号stream\_index 。流编号基于 libavformat 检测到的流顺序，除非还指定了程序 ID。在这种情况下，它基于程序中流的顺序。

stream\_type [: additional\_stream\_specifier ]

stream\_type是以下之一：“v”或“V”表示视频，“a”表示音频，“s”表示字幕，“d”表示数据，“t”表示附件。 'v' 匹配所有视频流，'V' 只匹配没有附加图片、视频缩略图或封面艺术的视频流。如果使用了additional\_stream\_specifier ，那么它将匹配具有此类型并匹配 additional\_stream\_specifier 的流。否则，它匹配指定类型的所有流。

p: program\_id [: additional\_stream\_specifier ]

匹配程序中具有 id program\_id的流。如果使用了additional\_stream\_specifier ，那么它匹配既是程序的一部分又匹配 additional\_stream\_specifier 的流。

# stream\_id或 i: stream\_id

通过流 ID（例如 MPEG-TS 容器中的 PID）匹配流。

米：键[：值]

具有指定值的元数据标签键的流。如果未给出值，则匹配包含具有任何值的给定标签的流。

你

匹配具有可用配置的流，必须定义编解码器，并且必须存在视频维度或音频采样率等基本信息。

请注意，在ffmpeg中，元数据匹配仅适用于输入文件。

### 5.2 通用选项

这些选项在 ff\* 工具之间共享。

-L

显示许可证。

-h, -?, -help, --help [参数]

显示帮助。可以指定一个可选参数来打印有关特定项目的帮助。如果未指定参数，则仅显示基本（非高级）工具选项。

arg的可能值为：

长的

除了基本工具选项外，还打印高级工具选项。

满的

打印完整的选项列表，包括编码器、解码器、多路分解器、多路复用器、过滤器等的共享和私有选项。

解码器=解码器名称

打印有关名为decoder\_name的解码器的详细信息。使用-decoders选项获取所有解码器的列表。

编码器=编码器名称

encoder\_name的编码器的详细信息。使用-encoders选项获取所有编码器的列表。

解复用器= demuxer\_name

打印有关名为demuxer\_name的解复用器的详细信息。使用-formats选项获取所有解复用器和复用器的列表。

混合器= muxer\_name

muxer\_name的 muxer 的详细信息。使用-formats选项获取所有多路复用器和多路分解器的列表。

过滤器=过滤器名称

打印有关名为filter\_name的过滤器的详细信息。使用-filters选项获取所有过滤器的列表。

bsf=比特流过滤器名称

打印有关名为bitstream\_filter\_name的比特流过滤器的详细信息。使用-bsfs选项获取所有比特流过滤器的列表。

协议=协议名称

打印有关名为protocol\_name的协议的详细信息。使用-protocols选项获取所有协议的列表。

-版本

显示版本。

-buildconf

显示构建配置，每行一个选项。

-格式

显示可用格式（包括设备）。

-解复用器

显示可用的分路器。

-混合器

显示可用的多路复用器。

-设备

显示可用设备。

-编解码器

显示 libavcodec 已知的所有编解码器。

请注意，本文档中使用的术语“编解码器”作为更准确地称为媒体比特流格式的快捷方式。

-解码器

显示可用的解码器。

-编码器

显示所有可用的编码器。

-bsfs

显示可用的比特流过滤器。

-协议

显示可用协议。

-过滤器

显示可用的 libavfilter 过滤器。

-pix\_fmts

显示可用的像素格式。

-sample\_fmts

显示可用的样本格式。

-布局

显示频道名称和标准频道布局。

-性格

显示流配置。

-颜色

显示可识别的颜色名称。

-源设备[, opt1 = val1 [, opt2 = val2 ]...]

显示输入设备的自动检测来源。某些设备可能会提供无法自动检测的依赖于系统的源名称。不能假设返回的列表总是完整的。

ffmpeg-源脉冲，服务器=192.168.0.4

-接收器设备[, opt1 = val1 [, opt2 = val2 ]...]

显示输出设备的自动检测接收器。某些设备可能会提供无法自动检测的依赖于系统的接收器名称。不能假设返回的列表总是完整的。

ffmpeg -sinks 脉冲，服务器=192.168.0.4

-loglevel [标志+] loglevel | -v [标志+]日志级别

设置库使用的日志记录级别和标志。

可选的标志前缀可以包含以下值：

'重复'

指示不应将重复的日志输出压缩到第一行，并且将省略“Last message repeated n times”行。

'水平'

表示日志输出应该为每个消息行添加一个[level]前缀。这可以用作日志着色的替代方法，例如在将日志转储到文件时。

标志也可以单独使用，通过添加 '+'/'-' 前缀来设置/重置单个标志，而不影响其他标志或更改loglevel 。当同时设置flags和loglevel时，在最后一个flags值和loglevel之前需要一个“+”分隔符。

loglevel是包含以下值之一的字符串或数字：

'安静，-8 '

什么都不显示；安静。

'恐慌，0 '

只显示可能导致进程崩溃的致命错误，例如断言失败。这目前没有用于任何事情。

'致命，8 '

只显示致命错误。这些是错误之后该过程绝对无法继续。

'错误，16 '

显示所有错误，包括可以从中恢复的错误。

'警告，24 '

显示所有警告和错误。将显示与可能不正确或意外事件相关的任何消息。

'信息，32 '

在处理过程中显示信息性消息。这是警告和错误的补充。这是默认值。

'详细，40 '

info相同，只是更冗长。

'调试，48 '

显示一切，包括调试信息。

'跟踪，56 '

例如要启用重复日志输出，添加level前缀，并将loglevel设置为verbose ：

ffmpeg -loglevel repeat+level+verbose -i 输入输出

级别前缀标志或loglevel的当前状态的情况下启用重复日志输出的示例：

ffmpeg [...] -loglevel +repeat

默认情况下，程序会记录到 stderr。如果终端支持着色，则颜色用于标记错误和警告。可以禁用日志着色设置环境变量AV\_LOG\_FORCE\_NOCOLOR ，或者可以强制设置环境变量AV\_LOG\_FORCE\_COLOR 。

-报告

当前目录中名为program - YYYYMMDD - HHMMSS .log的文件中。此文件可用于错误报告。它还暗示-loglevel debug 。

将环境变量FFREPORT设置为任何值都具有相同的效果。如果值是“:”分隔的键=值序列，这些选项将影响报表；如果选项值包含特殊字符或选项分隔符“:”，则必须对其进行转义（请参阅 ffmpeg-utils 手册中的“引用和转义”部分）。

识别以下选项：

文件

设置用于报告的文件名； %p扩展为程序名称， %t扩展为时间戳， %%扩展为普通%

等级

使用数值设置日志详细级别（请参阅-loglevel ）。

使用日志级别32 （日志级别信息的别名）将报告输出到名为ffreport.log的文件：

FFREPORT=file=ffreport.log:level=32 ffmpeg -i 输入输出

解析环境变量的错误不是致命的，不会出现在报告中。

-隐藏横幅

禁止打印横幅。

所有 FFmpeg 工具通常都会显示版权声明、构建选项和库版本。此选项可用于禁止打印此信息。

-cpuflags 标志（全局）

允许设置和清除 cpu 标志。此选项用于测试。除非您知道自己在做什么，否则不要使用它。

ffmpeg -cpuflags -sse+mmx ...

ffmpeg -cpuflags mmx ...

ffmpeg -cpuflags 0 ...

此选项的可能标志是：

' x86 '

' mmx '

' mmxext '

' sse '

' sse2 '

' sse2slow '

' sse3 '

' sse3slow '

' ssse3 '

'原子'

' sse4.1 '

' sse4.2 '

' avx '

' avx2 '

' xop '

' fma3 '

' fma4 '

' 3dnow '

' 3dnowext '

' bmi1 '

' bmi2 '

' cmov '

' ARM '

' armv5te '

' armv6 '

' armv6t2 '

' vfp '

' vfpv3 '

'霓虹灯'

'设置'

' AArch64 '

' armv8 '

' vfp '

'霓虹灯'

' PowerPC '

' altivec '

'特定处理器'

'奔腾2 '

'奔腾3 '

'奔腾4 '

' k6 '

' k62 '

“全能”

' athlonxp '

' k8 '

-cpucount计数（全局）

覆盖 CPU 计数检测。此选项用于测试。除非您知道自己在做什么，否则不要使用它。

ffmpeg -cpucount 2

-max\_alloc字节

通过 ffmpeg 的 malloc 函数系列设置在堆上分配块的最大大小限制。使用此选项时要格外**小心。**如果您不了解这样做的全部后果，请不要使用。默认值为 INT\_MAX。

### 5.3 AV选项

这些选项由 libavformat、libavdevice 和 libavcodec 库直接提供。要查看可用的 AVOptions 列表，请使用-help选项。它们分为两类：

通用的

可以为任何容器、编解码器或设备设置这些选项。通用选项列在容器/设备的 AVFormatContext 选项和编解码器的 AVCodecContext 选项下。

私人的

这些选项特定于给定的容器、设备或编解码器。私有选项列在其相应的容器/设备/编解码器下。

例如，要将 ID3v2.3 标头而不是默认的 ID3v2.4 写入 MP3 文件，请使用MP3 混合器的id3v2\_version私有选项：

ffmpeg -i 输入.flac -id3v2\_version 3 输出.mp3

所有编解码器 AVOptions 都是按流的，因此应将流说明符附加到它们：

ffmpeg -i multichannel.mxf -map 0:v:0 -map 0:a:0 -map 0:a:0 -c:a:0 ac3 -b:a:0 640k -ac:a:1 2 -c :a:1 aac -b:2 128k 输出.mp4

在上面的示例中，多声道音频流被映射两次以进行输出。第一个实例使用编解码器 ac3 和比特率 640k 进行编码。第二个实例被缩混为 2 个通道并使用编解码器 aac 进行编码。使用输出流的绝对索引为其指定 128k 的比特率。

注意： -nooption语法不能用于布尔 AVOptions，使用-option 0 / -option 1 。

注意：通过在选项名称前加上 v/a/s 来指定每个流 AVOptions 的旧的未记录的方法现在已经过时，并将很快被删除。

### 5.4 主要选项

-f fmt （输入/输出）

强制输入或输出文件格式。通常会自动检测输入文件的格式，并根据输出文件的文件扩展名猜测格式，因此在大多数情况下不需要此选项。

-i网址（输入）

输入文件地址

-y（全局）

不询问就覆盖输出文件。

-n（全局）

不覆盖输出文件，如果指定的输出文件已经存在，则立即退出。

-stream\_loop编号（输入）

设置输入流循环的次数。 Loop 0表示不循环，loop -1表示无限循环。

-recast\_media（全球）

允许强制解码器使用与多路分解器检测到或指定的媒体类型不同的媒体类型。用于解码混合为数据流的媒体数据。

-c[: stream\_specifier ]编解码器（输入/输出，每个流）

-codec[: stream\_specifier ]编解码器（输入/输出，每个流）

为一个或多个流选择编码器（在输出文件之前使用时）或解码器（在输入文件之前使用时）。 codec是解码器/编码器的名称或特殊值副本（仅输出）以指示流不被重新编码。

例如

ffmpeg -i INPUT -map 0 -c:v libx264 -c:a 复制输出

使用 libx264 编码所有视频流并复制所有音频流。

应用最后一个匹配的c选项，因此

ffmpeg -i 输入 -map 0 -c 复制 -c:v:1 libx264 -c:a:137 libvorbis 输出

将复制除第二个视频（将使用 libx264 编码）和第 138 个音频（将使用 libvorbis 编码）之外的所有流。

-t持续时间（输入/输出）

当用作输入选项时（在-i之前），限制从输入文件读取数据的持续时间。

duration后停止写入输出。

duration必须是持续时间规范，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)。

-to 和 -t 互斥，-t 优先。

-定位（输入/输出）

position停止写入输出或读取输入。 position必须是持续时间规范，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)。

-to 和 -t 互斥，-t 优先。

-fs limit\_size （输出）

设置文件大小限制，以字节表示。超过限制后不再写入字节块。输出文件的大小略大于请求的文件大小。

-ss位置（输入/输出）

当用作输入选项时（在-i之前），在此输入文件中查找到position 。请注意，在大多数格式中，不可能精确搜索，因此ffmpeg将搜索到position之前最近的搜索点。当启用转码和-accurate\_seek时（默认），搜索点和位置之间的这个额外片段将被解码并丢弃。在进行流复制或使用-noaccurate\_seek时，它将被保留。

当用作输出选项（在输出 url 之前）时，解码但丢弃输入，直到时间戳到达position 。

position必须是持续时间规范，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)。

-sseof位置（输入）

类似于-ss选项，但相对于“文件结尾”。也就是说负值在文件的前面，0 在 EOF。

-isync input\_index （输入）

将输入分配为同步源。

这将采用目标和参考输入的开始时间之间的差异，并通过该差异偏移目标文件的时间戳。两个输入的源时间戳应来自相同的时钟源以获得预期结果。如果设置了copyts，则还必须设置start\_at\_zero 。如果任一输入没有起始时间戳，则不进行同步调整。

可接受的值是那些引用有效 ffmpeg 输入索引的值。如果同步引用是目标索引本身或-1 ，则不会对目标时间戳进行任何调整。同步参考本身可能不会同步到任何其他输入。

默认值为-1 。

-itsoffset偏移量（输入）

设置输入时间偏移量。

offset必须是持续时间规范，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)。

偏移量被添加到输入文件的时间戳中。指定正偏移量意味着相应的流会延迟偏移量中指定的持续时间。

-itsscale规模（输入，每流）

重新缩放输入时间戳。 scale应该是一个浮点数。

-时间戳日期（输出）

在容器中设置录制时间戳。

date必须是日期规范，请参阅[(ffmpeg-utils)the Date section in the ffmpeg-utils(1) manual](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#date-syntax) 。

-metadata[:metadata\_specifier] key = value （输出，每个元数据）

设置元数据键/值对。

可以给出一个可选的metadata\_specifier来设置流、章节或节目的元数据。有关详细信息，请参阅-map\_metadata文档。

此选项会覆盖使用-map\_metadata设置的元数据。也可以使用空值删除元数据。

例如，要在输出文件中设置标题：

ffmpeg -i in.avi -metadata title="我的标题" out.flv

要设置第一个音频流的语言：

ffmpeg -i INPUT -metadata:s:a:0 language=eng 输出

-disposition[:stream\_specifier]值（输出，每个流）

设置流的配置。

默认情况下，处置是从输入流复制的，除非此选项适用的输出流由复杂的过滤图提供 - 在这种情况下，处置默认未设置。

值是由“+”或“-”分隔的一系列项目。第一项也可以以“+”或“-”为前缀，在这种情况下，此选项会修改默认值。否则（第一项没有前缀）此选项将覆盖默认值。 “+”前缀添加给定的配置，“-”删除它。也可以通过将其设置为 0 来清除处置。

如果没有为输出文件指定 -disposition 选项，当输出文件中有多个这种类型的流并且没有标记这种类型的流时，ffmpeg 将自动在每种类型的第一个流上设置“默认”处置作为默认值。

-dispositions选项列出已知的配置。

例如，要使第二个音频流成为默认流：

ffmpeg -i in.mkv -c copy -disposition:a:1 默认out.mkv

要使第二个字幕流成为默认流并从第一个字幕流中删除默认配置：

ffmpeg -i in.mkv -c copy -disposition:s:0 0 -disposition:s:1 默认out.mkv

添加嵌入式封面/缩略图：

ffmpeg -i in.mp4 -i IMAGE -map 0 -map 1 -c copy -c:v:1 png -disposition:v:1 attached\_pic out.mp4

并非所有混合器都支持嵌入式缩略图，支持的混合器仅支持几种格式，如 JPEG 或 PNG。

-program [title= title :][program\_num= program\_num :]st= stream [:st= stream ...] （输出）

title和program\_num的程序，并将指定的流添加到其中。

-目标类型（输出）

指定目标文件类型（ vcd 、 svcd 、 dvd 、 dv 、 dv50 ）。类型可以以pal- 、 ntsc-或film-为前缀，以使用相应的标准。然后自动设置所有格式选项（比特率、编解码器、缓冲区大小）。你可以输入：

ffmpeg -i myfile.avi -target vcd /tmp/vcd.mpg

不过，您可以指定其他选项，只要您知道它们与标准不冲突，例如：

ffmpeg -i myfile.avi -target vcd -bf 2 /tmp/vcd.mpg

为每个目标设置的参数如下。

**光碟**

朋友：

-f vcd -muxrate 1411200 -muxpreload 0.44 -packetsize 2324

-s 352x288 -r 25

-编解码器：v mpeg1video -g 15 -b:v 1150k -maxrate:v 1150k -minrate:v 1150k -bufsize:v 327680

-ar 44100 -ac 2

-编解码器:a mp2 -b:a 224k

制式：

-f vcd -muxrate 1411200 -muxpreload 0.44 -packetsize 2324

-s 352x240 -r 30000/1001

-codec:v mpeg1video -g 18 -b:v 1150k -maxrate:v 1150k -minrate:v 1150k -bufsize:v 327680

-ar 44100 -ac 2

-编解码器:a mp2 -b:a 224k

电影：

-f vcd -muxrate 1411200 -muxpreload 0.44 -packetsize 2324

-s 352x240 -r 24000/1001

-codec:v mpeg1video -g 18 -b:v 1150k -maxrate:v 1150k -minrate:v 1150k -bufsize:v 327680

-ar 44100 -ac 2

-编解码器:a mp2 -b:a 224k

**SVCD**

朋友：

-f svcd -packetsize 2324

-s 480x576 -pix\_fmt yuv420p -r 25

-codec:v mpeg2video -g 15 -b:v 2040k -maxrate:v 2516k -minrate:v 0 -bufsize:v 1835008 -scan\_offset 1

-ar 44100

-编解码器:a mp2 -b:a 224k

制式：

-f svcd -packetsize 2324

-s 480x480 -pix\_fmt yuv420p -r 30000/1001

-codec:v mpeg2video -g 18 -b:v 2040k -maxrate:v 2516k -minrate:v 0 -bufsize:v 1835008 -scan\_offset 1

-ar 44100

-编解码器:a mp2 -b:a 224k

电影：

-f svcd -packetsize 2324

-s 480x480 -pix\_fmt yuv420p -r 24000/1001

-codec:v mpeg2video -g 18 -b:v 2040k -maxrate:v 2516k -minrate:v 0 -bufsize:v 1835008 -scan\_offset 1

-ar 44100

-编解码器:a mp2 -b:a 224k

**DVD**

朋友：

-f DVD -muxrate 10080k -packetsize 2048

-s 720x576 -pix\_fmt yuv420p -r 25

-编解码器：v mpeg2video -g 15 -b:v 6000k -maxrate:v 9000k -minrate:v 0 -bufsize:v 1835008

-ar 48000

-编解码器：a ac3 -b：a 448k

制式：

-f DVD -muxrate 10080k -packetsize 2048

-s 720x480 -pix\_fmt yuv420p -r 30000/1001

-编解码器：v mpeg2video -g 18 -b:v 6000k -maxrate:v 9000k -minrate:v 0 -bufsize:v 1835008

-ar 48000

-编解码器：a ac3 -b：a 448k

电影：

-f DVD -muxrate 10080k -packetsize 2048

-s 720x480 -pix\_fmt yuv420p -r 24000/1001

-编解码器：v mpeg2video -g 18 -b:v 6000k -maxrate:v 9000k -minrate:v 0 -bufsize:v 1835008

-ar 48000

-编解码器：a ac3 -b：a 448k

**数码相机**

朋友：

-f dv

-s 720x576 -pix\_fmt yuv420p -r 25

-ar 48000 -ac 2

制式：

-f dv

-s 720x480 -pix\_fmt yuv411p -r 30000/1001

-ar 48000 -ac 2

电影：

-f dv

-s 720x480 -pix\_fmt yuv411p -r 24000/1001

-ar 48000 -ac 2

dv50目标与dv目标相同，只是所有三个标准的像素格式集都是yuv422p 。

上述参数的任何用户设置值都将覆盖目标预设值。在那种情况下，输出可能不符合目标标准。

-dn（输入/输出）

作为输入选项，阻止文件的所有数据流被过滤或自动选择或映射到任何输出。请参阅-discard选项以单独禁用流。

作为输出选项，禁用数据记录，即自动选择或映射任何数据流。对于完全手动控制，请参阅-map选项。

-dframes数（输出）

设置要输出的数据帧数。这是-frames:d的过时别名，您应该改用它。

-frames[: stream\_specifier ]帧数（输出，每个流）

framecount帧后停止写入流。

-q[: stream\_specifier ] q （输出，每个流）

-qscale[: stream\_specifier ] q （输出，每个流）

使用固定质量标度 (VBR)。 q / qscale的含义取决于编解码器。如果在没有stream\_specifier的情况下使用qscale ，则它仅适用于视频流，这是为了保持与先前行为的兼容性，并且在没有 stream\_specifier 时，将相同的编解码器特定值指定给 2 个不同的编解码器（音频和视频）通常不是预期的用来。

-filter[: stream\_specifier ] filtergraph ( output,per-stream )

filtergraph指定的 filtergraph并使用它来过滤流。

filtergraph是对应用于流的 filtergraph 的描述，并且必须具有与流相同类型的单个输入和单个输出。在 filtergraph 中，输入关联到标签in ，输出关联到标签out 。有关 filtergraph 语法的更多信息，请参阅 ffmpeg-filters 手册。

如果要创建具有多个输入和/或输出的过滤图，请参阅[-filter\_complex 选项。](#filter_005fcomplex_005foption)

-filter\_script[: stream\_specifier ]文件名（输出，每个流）

这个选项类似于-filter ，唯一的区别是它的参数是要从中读取过滤器图描述的文件的名称。

-reinit\_filter[: stream\_specifier ]整数（输入，每个流）

此布尔选项确定当输入帧参数在流中更改时，是否重新初始化此流所馈送的过滤器图。默认情况下启用此选项，因为大多数视频和所有音频过滤器都无法处理输入帧属性中的偏差。重新初始化后，现有的过滤器状态将丢失，例如某些过滤器中可用的帧计数n参考。在重新初始化时缓冲的任何帧都将丢失。更改触发重新初始化的属性是视频、帧分辨率或像素格式；用于音频、采样格式、采样率、通道数或通道布局。

-filter\_threads nb\_threads （全局）

定义用于处理过滤器管道的线程数。每个管道将生成一个线程池，其中包含可用于并行处理的多个线程。默认值是可用 CPU 的数量。

-pre[: stream\_specifier ] preset\_name （输出，每个流）

指定匹配流的预设。

-统计（全球）

打印编码进度/统计信息。默认情况下它是打开的，要明确禁用它，您需要指定-nostats 。

-stats\_period时间（全局）

设置更新编码进度/统计信息的时间段。默认值为 0.5 秒。

-进度网址（全球）

将程序友好的进度信息发送到url 。

进度信息定期写入，并在编码过程结束时写入。它由“ key = value ”行组成。密钥仅包含字母数字字符。一系列进度信息的最后一个关键字始终是“进度”。

-stats\_period设置。

-标准输入

在标准输入上启用交互。默认情况下打开，除非使用标准输入作为输入。要明确禁用交互，您需要指定-nostdin 。

禁用标准输入交互很有用，例如，如果 ffmpeg 在后台进程组中。使用ffmpeg ... < /dev/null可以获得大致相同的结果，但它需要一个 shell。

-debug\_ts（全局）

打印时间戳信息。它默认关闭。此选项主要用于测试和调试目的，输出格式可能会从一个版本更改为另一个版本，因此不应由可移植脚本使用。

另请参阅选项-fdebug ts 。

-附加文件名（输出）

将附件添加到输出文件。这被一些格式支持，例如 Matroska 用于渲染字幕的字体。附件作为特定类型的流实现，因此此选项将向文件添加新流。然后可以以通常的方式在此流上使用每个流的选项。使用此选项创建的附件流将在所有其他流（即使用-map或自动映射创建的流）之后创建。

请注意，对于 Matroska，您还必须设置 mimetype 元数据标签：

ffmpeg -i 输入 -attach DejaVuSans.ttf -metadata:s:2 mimetype=application/x-truetype-font out.mkv

（假设附件流将在输出文件中排在第三位）。

-dump\_attachment[: stream\_specifier ]文件名（输入，每个流）

filename的文件中。如果文件名是空的，那么将使用文件名元数据标签的值。

例如，将第一个附件提取到名为“out.ttf”的文件中：

ffmpeg -dump\_attachment:t:0 out.ttf -i 输入

要提取由文件名标签确定的文件的所有附件：

ffmpeg -dump\_attachment:t "" -i 输入

技术说明 – 附件作为编解码器额外数据实现，因此该选项实际上可用于从任何流中提取额外数据，而不仅仅是附件。

### 5.5 视频选项

-vframes数（输出）

设置要输出的视频帧数。这是-frames:v的过时别名，您应该改用它。

-r[: stream\_specifier ] fps （输入/输出，每个流）

设置帧速率（Hz 值、分数或缩写）。

作为一个输入选项，忽略文件中存储的任何时间戳，而是假设恒定帧速率fps生成时间戳。这与用于某些输入格式（如 image2 或 v4l2）的-framerate选项不同（它在旧版本的 FFmpeg 中是相同的）。如果有疑问，请使用-framerate而不是输入选项-r 。

作为输出选项：

视频编码

在编码之前复制或丢弃帧以实现恒定的输出帧速率fps 。

视频流复制

向复用器指示fps是流帧速率。在这种情况下，不会删除或复制任何数据。如果fps与数据包时间戳确定的实际流帧速率不匹配，这可能会产生无效文件。另请参阅setts比特流过滤器。

-fpsmax[: stream\_specifier ] fps （输出，每流）

设置最大帧速率（Hz 值、分数或缩写）。

当输出帧率自动设置且高于此值时，钳制输出帧率。在批处理或输入帧率被错误地检测为非常高时很有用。它不能与-r一起设置。它在流复制期间被忽略。

-s[: stream\_specifier ]大小（输入/输出，每个流）

设置帧大小。

video\_size私有选项的快捷方式，被一些帧大小未存储在文件中或可配置的解复用器识别——例如原始视频或视频采集器。

作为一个输出选项，这会将比例视频过滤器插入到相应过滤器图的末尾。请直接使用刻度过滤器将其插入开头或其他位置。

格式为“ wxh ”（默认 - 与来源相同）。

-aspect[: stream\_specifier ]方面（输出，每个流）

aspect指定的视频显示纵横比。

aspect可以是浮点数字符串，也可以是num : den形式的字符串，其中num和den是纵横比的分子和分母。例如“4:3”、“16:9”、“1.3333”和“1.7777”都是有效的参数值。

如果与-vcodec copy一起使用，它将影响存储在容器级别的纵横比，但不会影响存储在编码帧中的纵横比（如果存在）。

-display\_rotation[: stream\_specifier ]旋转（输入，每流）

设置视频旋转元数据。

rotation是一个十进制数，指定视频在显示之前应逆时针旋转的度数。

此选项会覆盖存储在文件中的旋转/显示转换元数据（如果有）。当视频正在转码（而不是复制）并启用-autorotate时，视频将在过滤阶段旋转。否则，如果 muxer 支持，元数据将被写入输出文件。

如果给出了-display\_hflip和/或-display\_vflip选项，它们将在此选项指定的旋转之后应用。

-display\_hflip[: stream\_specifier ]（输入，每个流）

设置显示图像时是否应水平翻转。

有关详细信息，请参阅-display\_rotation选项。

-display\_vflip[: stream\_specifier ]（输入，每个流）

设置显示图像时是否应垂直翻转。

有关详细信息，请参阅-display\_rotation选项。

-vn（输入/输出）

作为输入选项，阻止文件的所有视频流被过滤或自动选择或映射到任何输出。请参阅-discard选项以单独禁用流。

作为输出选项，禁用视频录制，即自动选择或映射任何视频流。对于完全手动控制，请参阅-map选项。

-vcodec编解码器（输出）

设置视频编解码器。这是-codec:v的别名。

-pass[: stream\_specifier ] n （输出，每个流）

选择通行证号码（1 或 2）。它用于进行二次视频编码。视频的统计信息在第一次通过时记录到日志文件中（另请参阅选项 -passlogfile），在第二次通过时，该日志文件用于以准确请求的比特率生成视频。在第 1 遍中，您可以只停用音频并将输出设置为空，Windows 和 Unix 的示例：

ffmpeg -i foo.mov -c:v libxvid -pass 1 -an -f rawvideo -y NUL

ffmpeg -i foo.mov -c:v libxvid -pass 1 -an -f rawvideo -y /dev/null

-passlogfile[: stream\_specifier ]前缀（输出，每个流）

将两次通过的日志文件名前缀设置为prefix ，默认文件名前缀为“ffmpeg2pass”。完整的文件名将是PREFIX-N.log ，其中 N 是特定于输出流的数字

-vf过滤图（输出）

filtergraph指定的 filtergraph并使用它来过滤流。

-filter:v的别名，请参阅[-filter 选项](#filter_005foption)。

-自动旋转

根据文件元数据自动旋转视频。默认启用，使用-noautorotate禁用它。

-自动缩放

根据第一帧的分辨率自动缩放视频。默认启用，使用-noautoscale禁用它。禁用自动缩放时，过滤器图形的所有输出帧可能不具有相同的分辨率，并且可能不适用于某些编码器/复用器。因此，除非您真的知道自己在做什么，否则不建议禁用它。禁用自动缩放需要您自担风险。

### 5.6 高级视频选项

-pix\_fmt[: stream\_specifier ]格式（输入/输出，每个流）

设置像素格式。使用-pix\_fmts显示所有支持的像素格式。如果无法选择所选的像素格式，ffmpeg 将打印警告并选择编码器支持的最佳像素格式。如果pix\_fmt以+为前缀，如果无法选择请求的像素格式，ffmpeg 将退出并显示错误，并且 filtergraphs 内的自动转换被禁用。如果pix\_fmt是单个+ ，则 ffmpeg 选择与输入（或图形输出）相同的像素格式并禁用自动转换。

-sws\_flags标志（输入/输出）

设置 SwScaler 标志。

-rc\_override[: stream\_specifier ]覆盖（输出，每个流）

特定间隔的速率控制覆盖，格式为“int，int，int”列表，用斜杠分隔。两个第一个值是开始和结束帧数，最后一个是量化器，如果是正数，或者质量因子，如果是负数。

-psnr

计算压缩帧的 PSNR。此选项已弃用，改为使用-flags +psnr将 PSNR 标志传递给编码器。

-vstats

将视频编码统计信息转储到vstats\_HHMMSS.log 。

-vstats\_file文件

将视频编码统计信息转储到文件中。

-vstats\_version文件

指定要使用的 vstats 格式版本。默认值为 2。

版本 = 1：

frame= %5d q= %2.1f PSNR= %6.2f f\_size= %6d s\_size= %8.0fkB time= %0.3f br= %7.1fkbits/s avg\_br= %7.1fkbits/s

版本 > 1:

out= %2d st= %2d frame= %5d q= %2.1f PSNR= %6.2f f\_size= %6d s\_size= %8.0fkB time= %0.3f br= %7.1fkbits/s avg\_br= %7.1fkbits/s

-top[: stream\_specifier ] n （输出，每个流）

top=1/bottom=0/auto=-1 字段优先

-vtag fourcc/标签（输出）

强制视频标签/fourcc。这是-tag:v的别名。

-qphist（全球）

显示 QP 直方图

-vbsf比特流过滤器

已弃用，请参阅 -bsf

-force\_key\_frames[: stream\_specifier ] time [, time ...] （输出，每个流）

-force\_key\_frames[: stream\_specifier ] expr: expr （输出，每个流）

-force\_key\_frames[: stream\_specifier ] 源（输出，每个流）

-force\_key\_frames[: stream\_specifier ] source\_no\_drop（输出，每流）

force\_key\_frames可以采用以下形式的参数：

时间[,时间...]

如果参数包含时间戳，ffmpeg 将根据编码器时基将指定时间四舍五入到最接近的输出时间戳，并在时间戳等于或大于计算时间戳的第一帧强制关键帧。请注意，如果编码器时基太粗糙，则关键帧可能会被强制在时间戳低于指定时间的帧上。默认编码器时基是输出帧率的倒数，但可以通过-enc\_time\_base进行其他设置。

如果其中一个时间是“ chapters [ delta ]”，则扩展为文件中所有章节的开始时间，移位delta ，表示为以秒为单位的时间。此选项可用于确保搜索点出现在章节标记或输出文件中的任何其他指定位置。

例如，要在5分钟插入一个关键帧，加上每章开始前0.1秒的关键帧：

-force\_key\_frames 0:05:00，chapters-0.1

表达式：表达式

如果参数以expr:为前缀，则字符串expr被解释为表达式并针对每一帧进行计算。如果评估不为零，则强制使用关键帧。

expr中的表达式可以包含以下常量：

n

当前处理的帧数，从0开始

n\_forced

强制帧数

prev\_forced\_n

前一个强制帧的编号，没有强制关键帧时为NAN

prev\_forced\_t

上一个强制帧的时间，没有强制关键帧时为NAN

吨

当前处理帧的时间

例如，每 5 秒强制执行一个关键帧，您可以指定：

-force\_key\_frames expr:gte(t,n\_forced\*5)

要在最后一个强制关键帧的时间后 5 秒强制关键帧，从第 13 秒开始：

-force\_key\_frames expr:if(isnan(prev\_forced\_t),gte(t,13),gte(t,prev\_forced\_t+5))

来源

如果参数是source ，如果正在编码的当前帧在其源中被标记为关键帧，ffmpeg 将强制使用关键帧。

source\_no\_drop

如果参数是source\_no\_drop ，如果正在编码的当前帧在其源中被标记为关键帧，ffmpeg 将强制使用关键帧。在必须删除此特定源帧的情况下，强制将下一个可用帧改为关键帧。

请注意，强制使用太多关键帧对某些编码器的前瞻算法非常有害：使用固定 GOP 选项或类似选项会更有效。

-copyinkf[: stream\_specifier ]（输出，每个流）

做流拷贝的时候，把开头找到的非关键帧也拷贝过来。

-init\_hw\_device类型[=名称][:设备[, key=value ...]]

使用给定的设备参数初始化一个类型为name的新硬件设备。如果没有指定名称，它将收到一个格式为“ type %d”的默认名称。

device和以下参数的含义取决于设备类型：

库达

device是 CUDA 设备的编号。

识别以下选项：

primary\_ctx

如果设置为 1，则使用主设备上下文而不是创建新设备上下文。

例子：

-init\_hw\_device 库达：1

选择系统上的第二个设备。

-init\_hw\_device cuda:0,primary\_ctx=1

选择第一个设备并使用主设备上下文。

dxva2

device是 Direct3D 9 显示适配器的编号。

d3d11va

device是 Direct3D 11 显示适配器的编号。

瓦皮

device是 X11 显示名称或 DRM 渲染节点。如果未指定，它将尝试打开默认的 X11 显示 ( $DISPLAY )，然后是第一个 DRM 渲染节点 ( /dev/dri/renderD128 )。

vdpau

device是一个 X11 显示名称。如果未指定，它将尝试打开默认的 X11 显示 ( $DISPLAY )。

qsv

设备在“ MFX\_IMPL\_\* ”中选择一个值。允许的值为：

汽车

开关

硬件

auto\_any

hw\_any

hw2

hw3

hw4

如果未指定，则使用“ auto\_any ”。 （请注意，通过创建适合平台的子设备（“ dxva2 ”或“ d3d11va ”或“ vaapi ”）然后从中派生 QSV 设备，可能更容易实现 QSV 的预期结果。）

或者，“ child\_device\_type ”有助于选择适合平台的子设备类型。在 Windows 上，“ d3d11va ”用作默认子设备类型。

例子：

-init\_hw\_device qsv:hw,child\_device\_type=d3d11va

d3d11va ”的 GPU 子设备，并使用“ MFX\_IMPL\_HARDWARE ”创建 QSV 设备。

-init\_hw\_device qsv:hw,child\_device\_type=dxva2

dxva2 ”的 GPU 子设备，并使用“ MFX\_IMPL\_HARDWARE ”创建 QSV 设备。

打开CL

device选择平台和设备作为platform\_index.device\_index 。

还可以使用键值对过滤设备集，以仅查找与特定平台或设备字符串匹配的设备。

可用作过滤器的字符串是：

平台配置文件

平台版本

平台名称

平台供应商

平台扩展

设备名称

设备供应商

驱动版本

设备版本

设备配置文件

设备扩展

设备类型

索引和过滤器必须一起唯一地选择一个设备。

例子：

-init\_hw\_device opencl:0.1

在第一个平台上选择第二个设备。

-init\_hw\_device opencl:,device\_name=Foo9000

选择名称包含字符串Foo9000 的设备。

-init\_hw\_device opencl:1,device\_type=gpu,device\_extensions=cl\_khr\_fp16

cl\_khr\_fp16扩展的第二个平台上选择 GPU 设备。

凡尔康

如果device是一个整数，它会根据系统相关设备列表中的索引选择设备。如果device是任何其他字符串，它会选择名称包含该字符串作为子字符串的第一个设备。

识别以下选项：

调试

如果设置为 1，则启用验证层（如果已安装）。

线性图像

如果设置为 1，则 hwcontext 分配的图像将是线性的并且可局部映射。

实例扩展

要启用的附加实例扩展的加号分隔列表。

设备扩展

要启用的附加设备扩展的加号分隔列表。

例子：

-init\_hw\_device vulkan:1

选择系统上的第二个设备。

-init\_hw\_device vulkan:RADV

选择名称包含字符串RADV的第一个设备。

-init\_hw\_device vulkan:0,instance\_extensions=VK\_KHR\_wayland\_surface+VK\_KHR\_xcb\_surface

选择第一个设备并启用 Wayland 和 XCB 实例扩展。

-init\_hw\_device type [= name ]@ source

类型为name的新硬件设备，它从名称为source的现有设备派生而来。

-init\_hw\_device 列表

列出此 ffmpeg 构建中支持的所有硬件设备类型。

-filter\_hw\_device名称

name的硬件设备传递给任何过滤器图中的所有过滤器。这可用于设置使用hwupload过滤器上传到的设备，或使用hwmap过滤器映射到的设备。其他过滤器在需要硬件设备时也可能使用此参数。请注意，这通常仅在输入不在硬件帧中时才需要——当它在硬件帧中时，过滤器将从它们作为输入接收的帧的上下文中派生出它们所需的设备。

这是一个全局设置，因此所有过滤器都将接收相同的设备。

-hwaccel[: stream\_specifier ] hwaccel （输入，每个流）

使用硬件加速来解码匹配的流。 hwaccel的允许值为：

没有任何

不要使用任何硬件加速（默认）。

汽车

自动选择硬件加速方式。

vdpau

使用 VDPAU（Unix 视频解码和演示 API）硬件加速。

dxva2

使用 DXVA2（DirectX 视频加速）硬件加速。

d3d11va

使用 D3D11VA（DirectX 视频加速）硬件加速。

瓦皮

使用 VAAPI（Video Acceleration API）硬件加速。

qsv

使用 Intel QuickSync 视频加速进行视频转码。

与大多数其他值不同，此选项不会启用加速解码（无论何时选择 qsv 解码器都会自动使用），但会加速转码，而无需将帧复制到系统内存中。

要使其正常工作，解码器和编码器都必须支持 QSV 加速，并且不得使用过滤器。

如果所选 hwaccel 不可用或所选解码器不支持，则此选项无效。

请注意，大多数加速方法都用于播放，不会比现代 CPU 上的软件解码更快。此外， ffmpeg通常需要将解码后的帧从 GPU 内存复制到系统内存中，从而导致进一步的性能损失。因此，此选项主要用于测试。

-hwaccel\_device[: stream\_specifier ] hwaccel\_device （输入，每流）

选择用于硬件加速的设备。

同时指定了-hwaccel选项时才有意义。它可以通过名称引用使用-init\_hw\_device创建的现有设备，或者它可以创建一个新设备，就像之前调用“ -init\_hw\_device ”类型： hwaccel\_device一样。

-hwaccels

列出在此 ffmpeg 构建中启用的所有硬件加速组件。实际运行时可用性取决于硬件及其安装的合适驱动程序。

-fix\_sub\_duration\_heartbeat[: stream\_specifier ]

将特定的输出视频流设置为心跳流，根据该心跳流在收到随机访问数据包时拆分并推送当前正在进行的字幕。

这降低了尚未接收到结束包或后续字幕的字幕的等待时间。作为一个缺点，这很可能会导致字幕事件重复以覆盖整个持续时间，因此在处理将字幕事件传递到输出的延迟不相关的用例时，不应使用此选项。

需要为相关的输入字幕流设置-fix\_sub\_duration才能生效，以及必须将输入字幕流直接映射到心跳流所在的相同输出。

### 5.7 音频选项

-帧数（输出）

设置要输出的音频帧数。这是-frames:a的过时别名，您应该改用它。

-ar[: stream\_specifier ] freq （输入/输出，每个流）

设置音频采样频率。对于输出流，它默认设置为相应输入流的频率。对于输入流，此选项仅对音频抓取设备和原始多路分解器有意义，并映射到相应的多路分解器选项。

-aq q （输出）

设置音频质量（特定于编解码器，VBR）。这是 -q:a 的别名。

-ac[: stream\_specifier ]通道（输入/输出，每个流）

设置音频通道数。对于输出流，它默认设置为输入音频通道数。对于输入流，此选项仅对音频抓取设备和原始多路分解器有意义，并映射到相应的多路分解器选项。

-an（输入/输出）

作为输入选项，阻止文件的所有音频流被过滤或自动选择或映射到任何输出。请参阅-discard选项以单独禁用流。

作为输出选项，禁用音频录制，即自动选择或映射任何音频流。对于完全手动控制，请参阅-map选项。

-acodec编解码器（输入/输出）

设置音频编解码器。这是-codec:a的别名。

-sample\_fmt[: stream\_specifier ] sample\_fmt （输出，每个流）

设置音频样本格式。使用-sample\_fmts获取支持的样本格式列表。

-af过滤图（输出）

filtergraph指定的 filtergraph并使用它来过滤流。

-filter:a的别名，请参阅[-filter 选项](#filter_005foption)。

### 5.8 高级音频选项

-atag fourcc/标签（输出）

强制音频标签/fourcc。这是-tag:a的别名。

-absf比特流过滤器

已弃用，请参阅 -bsf

-guess\_layout\_max通道（输入，每个流）

如果不知道某些输入通道布局，请尝试猜测它是否最多对应于指定数量的通道。例如，2 告诉ffmpeg将 1 个通道识别为单声道，将 2 个通道识别为立体声，但不将 6 个通道识别为 5.1。默认设置是始终尝试猜测。使用 0 禁用所有猜测。

### 5.9 字幕选项

-scodec编解码器（输入/输出）

设置字幕编解码器。这是-codec:s的别名。

-sn（输入/输出）

作为输入选项，阻止文件的所有字幕流被过滤或自动选择或映射到任何输出。请参阅-discard选项以单独禁用流。

作为输出选项，禁用字幕录制，即自动选择或映射任何字幕流。对于完全手动控制，请参阅-map选项。

-sbsf比特流过滤器

已弃用，请参阅 -bsf

### 5.10 高级字幕选项

-fix\_sub\_duration

修复字幕持续时间。对于每个字幕，等待同一流中的下一个数据包并调整第一个的持续时间以避免重叠。这对于某些字幕编解码器是必需的，尤其是 DVB 字幕，因为原始数据包中的持续时间只是粗略估计，结尾实际上由空字幕帧标记。由于非单调时间戳，必要时未能使用此选项可能会导致持续时间延长或多路复用失败。

请注意，此选项将延迟所有数据的输出，直到下一个字幕包被解码：它可能会增加内存消耗和延迟很多。

-canvas\_size大小

设置用于渲染字幕的画布大小。

### 5.11 高级选项

-map [-] input\_file\_id [: stream\_specifier ][?] | [链接标签] （输出）

在输出文件中创建一个或多个流。此选项有两种指定数据源的形式：第一种从某个输入文件中选择一个或多个流（用-i指定） ，第二种从某个复杂的过滤器图（用-filter\_complex或-filter\_complex\_script指定）中获取输出.

input\_file\_id的输入文件的每个流创建一个输出流。如果给出了stream\_specifier ，则仅使用与说明符匹配的流（有关stream\_specifier语法，请参阅[Stream 说明符部分](#Stream-specifiers)）。

A -流标识符之前的字符创建“负”映射。它禁用来自已创建映射的匹配流。

尾随？在流索引将允许地图成为可选之后：如果地图不匹配任何流，则地图将被忽略而不是失败。请注意，如果使用了无效的输入文件索引，映射仍然会失败；例如地图是否引用了不存在的输入。

另一种[linklabel]形式会将复杂过滤器图（请参阅-filter\_complex选项）的输出映射到输出文件。 linklabel必须对应于图中定义的输出链接标签。

可以多次指定此选项，每次都向输出文件添加更多流。任何给定的输入流也可以被映射任意次数作为不同输出流的源，例如为了使用不同的编码选项和/或过滤器。流在输出中的创建顺序与在命令行中给出-map选项的顺序相同。

使用此选项禁用此输出文件的默认映射。

例子：

映射一切

将第一个输入文件中的所有流映射到输出

ffmpeg -i INPUT -map 0 输出

选择特定流

0:0和0:1标识。您可以使用-map来选择要放置在输出文件中的流。例如：

ffmpeg -i INPUT -map 0:1 输出.wav

INPUT中的第二个输入流映射到out.wav中的（单个）输出流。

创建多个流

a.mov （由标识符0:2指定）中选择索引为 2 的流，并从输入文件 b.mov （由标识符1:6指定）中选择索引为 6 的流，并将它们复制到输出文件输出.mov :

ffmpeg -i a.mov -i b.mov -c copy -map 0:2 -map 1:6 out.mov

创建多个流 2

从输入文件中选择所有视频和第三个音频流：

ffmpeg -i 输入 -map 0:v -map 0:a:2 输出

负面地图

要映射除第二个音频之外的所有流，请使用负映射

ffmpeg -i 输入 -map 0 -map -0:a:1 输出

可选地图

从第一个输入映射视频和音频流，并使用尾随? ，如果第一个输入中不存在音频流，则忽略音频映射：

ffmpeg -i 输入 -map 0:v -map 0:a?输出

按语言映射

选择英语音频流：

ffmpeg -i INPUT -map 0:m:language:eng 输出

-ignore\_unknown

如果尝试复制此类流，则忽略具有未知类型的输入流而不是失败。

-copy\_unknown

如果尝试复制此类流，则允许复制未知类型的输入流而不是失败。

-map\_channel [ input\_file\_id . stream\_specifier 。 channel\_id |-1][?][: output\_file\_id 。 stream\_specifier ]

此选项已弃用并将被删除。它可以用盘式过滤器代替。在某些情况下，使用channelsplit 、 channelmap或amerge过滤器的某种组合可能会更容易。

将给定输入的音频通道映射到输出。如果output\_file\_id 。 stream\_specifier未设置，音频通道将映射到所有音频流。

使用 "-1" 而不是input\_file\_id 。 stream\_specifier 。 channel\_id将映射一个静音频道。

尾随？将允许 map\_channel 是可选的：如果 map\_channel 不匹配任何通道，则 map\_channel 将被忽略而不是失败。

例如，假设INPUT是一个立体声音频文件，您可以使用以下命令切换两个音频通道：

ffmpeg -i 输入 -map\_channel 0.0.1 -map\_channel 0.0.0 输出

如果要使第一个通道静音并保留第二个通道：

ffmpeg -i 输入 -map\_channel -1 -map\_channel 0.0.1 输出

“-map\_channel”选项的顺序指定输出流中通道的顺序。输出通道布局是根据映射的通道数量猜测的（如果一个“-map\_channel”是单声道，如果是两个则为立体声，等等）。如果输入和输出通道布局不匹配（例如两个“-map\_channel”选项和“-ac 6”），则结合使用“-ac”和“-map\_channel”可以更新通道增益水平。

您还可以将输入的每个通道提取到特定的输出；以下命令将INPUT音频流（文件 0，流 0）的两个通道提取到相应的OUTPUT\_CH0和OUTPUT\_CH1输出：

ffmpeg -i INPUT -map\_channel 0.0.0 OUTPUT\_CH0 -map\_channel 0.0.1 OUTPUT\_CH1

以下示例将立体声输入的通道拆分为两个单独的流，并将其放入同一输出文件中：

ffmpeg -i stereo.wav -map 0:0 -map 0:0 -map\_channel 0.0.0:0.0 -map\_channel 0.0.1:0.1 -y out.ogg

请注意，目前每个输出流只能包含来自单个输入流的通道；例如，您不能使用“-map\_channel”来选择包含在不同流中的多个输入音频通道（来自相同或不同的文件）并将它们合并到一个输出流中。因此，例如，目前不可能将两个独立的单声道流转换为单个立体声流。然而，将立体声流分成两个单声道单声道流是可能的。

如果您需要此功能，可能的解决方法是使用amerge过滤器。例如，如果您需要将具有 2 个单声道音频流的媒体（此处为input.mkv ）合并为一个立体声通道音频流（并保留视频流），您可以使用以下命令：

ffmpeg -i input.mkv -filter\_complex "[0:1] [0:2] amerge" -c:a pcm\_s16le -c:v 复制输出.mkv

要从第一个输入映射前两个音频通道，并使用尾随? ，如果第一个输入是单声道而不是立体声，则忽略音频通道映射：

ffmpeg -i 输入 -map\_channel 0.0.0 -map\_channel 0.0.1？输出

-map\_metadata[: metadata\_spec\_out ] infile [: metadata\_spec\_in ] （输出，每个元数据）

infile设置下一个输出文件的元数据信息。请注意，这些是文件索引（从零开始），而不是文件名。可选的metadata\_spec\_in/out参数指定要复制的元数据。元数据说明符可以具有以下形式：

G

全局元数据，即适用于整个文件的元数据

s [: stream\_spec ]

每个流元数据。 stream\_spec是流说明符一章中描述的[流说明符](#Stream-specifiers)。在输入元数据说明符中，第一个匹配的流是从中复制的。在输出元数据说明符中，所有匹配的流都被复制到。

c ：章节索引

每章元数据。 chapter\_index是从零开始的章节索引。

p ：程序索引

每个程序的元数据。 program\_index是从零开始的程序索引。

如果省略元数据说明符，则默认为全局。

默认情况下，全局元数据从第一个输入文件复制，每个流和每个章节的元数据与流/章节一起复制。通过创建相关类型的任何映射来禁用这些默认映射。负文件索引可用于创建仅禁用自动复制的虚拟映射。

例如，将元数据从输入文件的第一个流复制到输出文件的全局元数据：

ffmpeg -i in.ogg -map\_metadata 0:s:0 输出.mp3

相反，将全局元数据复制到所有音频流：

ffmpeg -i in.mkv -map\_metadata:s:a 0:g out.mkv

请注意，简单的0在此示例中也适用，因为默认情况下采用全局元数据。

-map\_chapters input\_file\_index （输出）

input\_file\_index的输入文件中的章节复制到下一个输出文件。如果未指定章节映射，则从第一个输入文件复制章节，其中至少有一个章节。使用负文件索引来禁用任何章节复制。

-基准（全球）

在编码结束时显示基准信息。显示实际使用的系统和用户时间以及最大内存消耗。并非所有系统都支持最大内存消耗，如果不支持，它通常会显示为 0。

-benchmark\_all（全球）

在编码期间显示基准信息。显示在各个步骤（音频/视频编码/解码）中使用的实际时间、系统时间和用户时间。

-timelimit持续时间（全球）

ffmpeg 在 CPU 用户时间内运行duration秒后退出。

-转储（全球）

将每个输入数据包转储到 stderr。

-十六进制（全局）

转储数据包时，也会转储有效负载。

-readrate速度（输入）

限制输入读取速度。

它的值是一个浮点正数，表示媒体的最大持续时间（以秒为单位），应该在挂钟时间的一秒内摄取。默认值为零，表示没有对摄取速度施加限制。值1表示实时速度，相当于-re 。

主要用于模拟捕获设备或实时输入流（例如，从文件中读取时）。当输入是实际的捕获设备或实时流时，不应使用较低的值，因为它可能会导致数据包丢失。

当输出数据包的流速很重要时，它很有用，例如直播。

-re（输入）

以原始帧速率读取输入。这相当于设置-readrate 1 。

-vsync参数（全局）

-fps\_mode[: stream\_specifier ]参数（输出，每流）

设置视频同步方式/帧率模式。 vsync 应用于所有输出视频流，但可以通过设置 fps\_mode 为流覆盖。 vsync 已弃用，将来会被删除。

出于兼容性原因，可以将 vsync 的某些值指定为数字（显示在下表的括号中）。

直通 (0)

每个帧连同其时间戳从多路分解器传递到多路复用器。

CFR (1)

帧将被复制和丢弃以准确地达到请求的恒定帧速率。

可变帧频 (2)

帧通过它们的时间戳传递或丢弃，以防止 2 个帧具有相同的时间戳。

降低

作为直通但会破坏所有时间戳，使复用器根据帧速率生成新的时间戳。

自动 (-1)

根据复用器功能在 cfr 和 vfr 之间进行选择。这是默认方法。

请注意，在此之后，多路复用器可能会进一步修改时间戳。例如，在启用格式选项avoid\_negative\_ts的情况下。

使用 -map 您可以选择从哪个流中获取时间戳。您可以保留视频或音频不变，并将剩余的流同步到未更改的流。

-frame\_drop\_threshold参数

丢帧阈值，它指定视频帧在被丢弃之前可以落后多少。在帧速率单位中， 1.0 是一帧。默认值为 -1.1。一种可能的用例是在时间戳嘈杂的情况下避免丢帧，或者在时间戳准确的情况下提高丢帧精度。

-adrift\_threshold时间

设置时间戳和音频数据之间的最小差异（以秒为单位）以触发添加/删除样本以使其与时间戳匹配。此选项实际上是在硬（添加/删除）和软（挤压/拉伸）补偿之间进行选择的阈值。 -async必须设置为正值。

-apad参数（输出，每个流）

填充输出音频流。这与应用-af apad相同。 Argument 是一串过滤器参数，组成与apad过滤器相同。 -shortest必须为此输出设置才能使选项生效。

-copyts

不要处理输入时间戳，但保留它们的值而不尝试对其进行清理。特别是，不要删除初始开始时间偏移值。

请注意，根据vsync选项或特定的多路复用器处理（例如，在启用格式选项avoid\_negative\_ts的情况下），即使选择了此选项，输出时间戳也可能与输入时间戳不匹配。

-start\_at\_zero

copyts一起使用时，移动输入时间戳，使它们从零开始。

这意味着使用例如-ss 50将使输出时间戳从 50 秒开始，而不管输入文件从什么时间戳开始。

-copytb模式

指定流复制时如何设置编码器时基。 mode是一个整数数值，可以采用以下值之一：

1个

使用多路分解器时基。

时基从相应的输入多路分解器复制到输出编码器。在复制具有可变帧速率的视频流时，有时需要这样做以避免非单调增加的时间戳。

0

使用解码器时基。

时基从相应的输入解码器复制到输出编码器。

-1

尝试自动做出选择，以生成合理的输出。

默认值为 -1。

-enc\_time\_base[: stream\_specifier ]时基（输出，每个流）

设置编码器时基。 timebase是一个浮点数，可以采用以下值之一：

0

根据媒体类型分配默认值。

对于视频 - 使用 1/framerate，对于音频 - 使用 1/samplerate。

-1

尽可能使用输入流时基。

如果输入流不可用，将使用默认时基。

>0

使用提供的数字作为时基。

该字段可以提供为两个整数的比率（例如 1:24、1:48000）或浮点数（例如 0.04166、2.0833e-5）

默认值为 0。

-bitexact（输入/输出）

为 (de)muxer 和 (de/en)coder 启用 bitexact 模式

-最短（输出）

当最短的输出流结束时完成编码。

请注意，此选项可能需要缓冲帧，这会引入额外的延迟。可以使用-shortest\_buf\_duration选项控制此延迟的最大量。

-shortest\_buf\_duration持续时间（输出）

-shortest选项可能需要在至少一个流“稀疏”时缓冲潜在的大量数据（即帧之间有很大的间隙——这通常是字幕的情况） 。

此选项控制缓冲帧的最大持续时间（以秒为单位）。较大的值可能允许-shortest选项产生更准确的结果，但会增加内存使用和延迟。

默认值为 10 秒。

-dts\_delta\_threshold

时间戳不连续增量阈值。

-dts\_error\_threshold秒

时间戳错误增量阈值。此阈值用于丢弃疯狂/损坏的时间戳，默认值为 30 小时，这是任意选择的并且非常保守。

-muxdelay秒（输出）

设置最大解复用解码延迟。

-muxpreload秒（输出）

设置初始多路分解解码延迟。

-streamid输出流索引：新值（输出）

为输出流分配一个新的 stream-id 值。此选项应在它适用的输出文件名之前指定。对于存在多个输出文件的情况，可以将streamid重新赋值给不同的值。

例如，要将输出 mpegts 文件的流 0 PID 设置为 33，将流 1 PID 设置为 36：

ffmpeg -i inurl -streamid 0:33 -streamid 1:36 out.ts

-bsf[: stream\_specifier ] bitstream\_filters （输出，每个流）

为匹配流设置比特流过滤器。 bitstream\_filters是逗号分隔的比特流过滤器列表。使用-bsfs选项获取比特流过滤器列表。

ffmpeg -i h264.mp4 -c:v 复制 -bsf:v h264\_mp4toannexb -an out.h264

ffmpeg -i file.mov -an -vn -bsf:s mov2textsub -c:s copy -f rawvideo sub.txt

-tag[: stream\_specifier ] codec\_tag （输入/输出，每个流）

强制使用标签/fourcc 来匹配流。

-时间码hh : mm : ss SEP ff

指定写入时间码。 SEP是 ':' 对于非丢弃时间码和 ';' （或'.'）用于丢弃。

ffmpeg -i 输入.mpg -时间码 01:02:03.04 -r 30000/1001 -s ntsc 输出.mpg

-filter\_complex过滤图（全局）

定义一个复杂的过滤图，即具有任意数量的输入和/或输出的过滤图。对于简单的图形——那些具有相同类型的一个输入和一个输出的图形——请参阅-filter选项。 filtergraph是对 filtergraph 的描述，在 ffmpeg-filters 手册的“Filtergraph syntax”部分有描述。

[file\_index:stream\_specifier]语法（即与-map使用的相同）引用输入流。如果stream\_specifier匹配多个流，将使用第一个。未标记的输入将连接到匹配类型的第一个未使用的输入流。

输出链接标签用-map引用。未标记的输出被添加到第一个输出文件。

请注意，使用此选项可以仅使用没有普通输入文件的 lavfi 源。

例如，将图像叠加在视频上

ffmpeg -i video.mkv -i image.png -filter\_complex '[0:v][1:v]overlay[out]' -map

'[out]' out.mkv

这里[0:v]指的是第一个输入文件中的第一个视频流，它链接到叠加过滤器的第一个（主要）输入。类似地，第二个输入中的第一个视频流链接到覆盖的第二个（覆盖）输入。

假设每个输入文件中只有一个视频流，我们可以省略输入标签，所以上面等价于

ffmpeg -i video.mkv -i image.png -filter\_complex 'overlay[out]' -map

'[out]' out.mkv

此外，我们可以省略输出标签，过滤图的单个输出将自动添加到输出文件中，因此我们可以简单地写

ffmpeg -i video.mkv -i image.png -filter\_complex '覆盖' out.mkv

作为一个特殊的例外，您可以使用位图字幕流作为输入：它将被转换为与文件中最大视频大小相同的视频，如果没有视频则为 720x576。请注意，这是一个实验性的临时解决方案。一旦 libavfilter 对字幕有适当的支持，它将被删除。

例如，要在以 MPEG-TS 格式存储的 DVB-T 记录之上硬编码字幕，将字幕延迟 1 秒：

ffmpeg -i input.ts -filter\_complex \

'[#0x2ef] setpts=PTS+1/TB [sub] ; [#0x2d0] [子] 覆盖'\

-sn -map '#0x2dc' 输出.mkv

（0x2d0、0x2dc 和 0x2ef 分别是视频、音频和字幕流的 MPEG-TS PID；0:0、0:3 和 0:7 也可以）

色源生成5秒纯红色视频：

ffmpeg -filter\_complex 'color=c=red' -t 5 输出.mkv

-filter\_complex\_threads nb\_threads （全局）

定义用于处理 filter\_complex 图的线程数。类似于 filter\_threads 但仅用于-filter\_complex图。默认值是可用 CPU 的数量。

-lavfi滤镜图（全局）

定义一个复杂的过滤图，即具有任意数量的输入和/或输出的过滤图。相当于-filter\_complex 。

-filter\_complex\_script文件名（全局）

这个选项类似于-filter\_complex ，唯一的区别是它的参数是要从中读取复杂过滤器图描述的文件的名称。

-accurate\_seek（输入）

此选项使用-ss选项启用或禁用在输入文件中的准确搜索。默认情况下启用，因此在转码时查找是准确的。使用-noaccurate\_seek禁用它，这可能很有用，例如在复制一些流和转码其他流时。

-seek\_timestamp（输入）

此选项使用-ss选项启用或禁用在输入文件中按时间戳查找。默认情况下禁用。如果启用， -ss选项的参数将被视为实际时间戳，并且不会被文件的开始时间偏移。这仅适用于不从时间戳 0 开始的文件，例如传输流。

-thread\_queue\_size大小（输入/输出）

对于输入，此选项设置从文件或设备读取时排队数据包的最大数量。对于低延迟/高速率的直播流，如果不及时读取数据包可能会被丢弃；设置此值可以强制 ffmpeg 使用单独的输入线程并在数据包到达时立即读取数据包。默认情况下，ffmpeg 仅在指定多个输入时才执行此操作。

对于输出，此选项指定了可以排队到每个 muxing 线程的最大数据包数。

-sdp\_file文件（全局）

将输出流的 sdp 信息打印到file 。这允许在至少一个输出不是 rtp 流时转储 sdp 信息。 （要求至少一种输出格式为 rtp）。

-丢弃（输入）

允许从流中丢弃特定的流或帧。任何输入流都可以使用 value all完全丢弃，而从流中选择性丢弃帧发生在多路分解器中，并非所有多路分解器都支持。

没有任何

不丢弃任何框架。

默认

默认，不丢弃任何帧。

noref

丢弃所有非参考帧。

比迪尔

丢弃所有双向帧。

无钥匙

丢弃除关键帧之外的所有帧。

全部

丢弃所有帧。

-abort\_on标志（全局）

在各种条件下停止和中止。以下标志可用：

空输出

没有数据包传递到多路复用器，输出为空。

空输出流

在一些输出流中没有数据包被传递到多路复用器。

-max\_error\_rate（全局）

设置所有输入的解码帧失败分数，当超过 ffmpeg 时将返回退出代码 69。超过此阈值不会终止处理。范围是 0 到 1 之间的浮点数。默认值为 2/3。

-xerror（全局）

出错时停止并退出

-max\_muxing\_queue\_size数据包（输出，每个流）

当转码音频和/或视频流时，ffmpeg 将不会开始写入输出，直到每个此类流都有一个数据包。在等待这种情况发生时，缓冲其他流的数据包。此选项为匹配的输出流设置此缓冲区的大小（以数据包为单位）。

此选项的默认值对于大多数用途来说应该足够高，因此只有在确定需要时才触摸此选项。

-muxing\_queue\_data\_threshold字节（输出，每流）

这是一个最小阈值，直到该阈值不考虑多路复用队列大小。默认为每个流 50 兆字节，并且基于传递给多路复用器的数据包的总体大小。

-auto\_conversion\_filters（全球）

在所有过滤器图中启用自动插入格式转换过滤器，包括那些由-vf 、 -af 、 -filter\_complex和-lavfi定义的过滤器。如果过滤器格式协商需要转换，则过滤器的初始化将失败。仍然可以通过在图表中插入相关的转换过滤器（scale、aresample）来执行转换。默认情况下，要明确禁用它，您需要指定-noauto\_conversion\_filters 。

-bits\_per\_raw\_sample[: stream\_specifier ]值（输出，每个流）

将给定输出流中每个原始样本的位数声明为value 。请注意，此选项设置提供给编码器/复用器的信息，它不会更改流以符合此值。设置与流属性不匹配的值可能会导致编码失败或输出文件无效。

-stats\_enc\_pre[: stream\_specifier ]路径（输出，每个流）

-stats\_enc\_post[: stream\_specifier ]路径（输出，每个流）

-stats\_mux\_pre[: stream\_specifier ]路径（输出，每个流）

path给定的文件中。

-stats\_enc\_pre在原始视频或音频帧发送进行编码之前写入有关它们的信息，而-stats\_enc\_post在从编码器接收到编码数据包时写入有关它们的信息。 -stats\_mux\_pre在数据包即将发送到 muxer 时写入有关数据包的信息。每个帧或数据包在指定文件中生成一行。该行的格式由-stats\_enc\_pre\_fmt / -stats\_enc\_post\_fmt / -stats\_mux\_pre\_fmt控制。

当多个流的统计信息写入单个文件时，对应于不同流的行将交错。这种交错的精确顺序没有指定，并且不能保证在程序的不同调用之间保持稳定，即使使用相同的选项也是如此。

-stats\_enc\_pre\_fmt[: stream\_specifier ] format\_spec （输出，每个流）

-stats\_enc\_post\_fmt[: stream\_specifier ] format\_spec （输出，每流）

-stats\_mux\_pre\_fmt[: stream\_specifier ] format\_spec （输出，每个流）

-stats\_enc\_pre / -stats\_enc\_post / -stats\_mux\_pre编写的行的格式。

format\_spec是一个字符串，可能包含{fmt}形式的指令。 format\_spec是反斜杠转义的——使用 \{、\} 和 \\ 分别将文字 {、} 或 \ 写入输出。

fmt给出的指令可能是以下之一：

fidx

输出文件的索引。

sidx

文件中输出流的索引。

n

帧数。预编码：到目前为止发送到编码器的帧数。编码后：到目前为止从编码器接收到的数据包数。 Muxing：到目前为止，为此流提交给 muxer 的数据包数。

你

输入帧号。对应于该输出帧或数据包的输入帧（即解码器输出）的索引。 -1 如果不可用。

结核病

编码器时基，作为有理数 num /den 。请注意，这可能与复用器使用的时基不同。

tbi

ptsi的时基，作为有理数num/den 。当ptsi可用时可用，否则为0/1 。

积分

帧或数据包的呈现时间戳，作为整数。应该乘以时基来计算演示时间。

ptsi

输入帧的呈现时间戳（参见ni ），作为整数。应乘以tbi以计算呈现时间。不可用时打印为 (2^63 - 1 = 9223372036854775807)。

吨

帧或数据包的呈现时间，以十进制数表示。等于pts乘以tb 。

钛

输入帧的呈现时间（参见ni ），作为十进制数。等于ptsi乘以tbi 。不可用时打印为 inf。

数据传输系统

数据包的解码时间戳，作为整数。应该乘以时基来计算演示时间。仅后编码。

dt

帧或数据包的解码时间，作为十进制数。等于dts乘以tb 。

锡

到目前为止发送到编码器的音频样本数。仅音频和预编码。

样本

帧中的音频样本数。仅音频和预编码。

尺寸

编码数据包的大小（以字节为单位）。仅后编码。

br

当前比特率（以比特每秒为单位）。仅后编码。

abr

到目前为止整个流的平均比特率，以比特每秒为单位，如果此时无法确定，则为 -1。仅后编码。

默认格式字符串是：

预编码

{fidx} {sidx} {n} {t}

编码后

{fidx} {sidx} {n} {t}

将来，可能会将新项目添加到默认格式字符串的末尾。依赖格式保持完全相同的用户应手动指定。

请注意，写入同一文件的不同流的统计信息可能具有不同的格式。

### 5.12 预设文件

预设文件包含一系列option = value对，每行一个，指定一系列选项，这些选项很难在命令行上指定。以井号 ('#') 字符开头的行将被忽略并用于提供注释。查看FFmpeg 源代码树中的预设目录以获取示例。

有两种类型的预设文件：ffpreset 和 avpreset 文件。

* [ff预设文件](#ffpreset-files)
* [av预设文件](#avpreset-files)

#### 5.12.1 ff预设文件

vpre 、 apre 、 spre和fpre选项指定。 fpre选项将预设的文件名而不是预设名称作为输入，可用于任何类型的编解码器。对于vpre 、 apre和spre选项，预设文件中指定的选项将应用于当前选择的与预设选项类型相同的编解码器。

传递给vpre 、 apre和spre预设选项的参数根据以下规则标识要使用的预设文件：

在目录$FFMPEG\_DATADIR （如果设置）和$HOME/.ffmpeg中搜索名为arg .ffpreset 的文件，并在配置时定义的数据目录（通常是PREFIX/share/ffmpeg ）或沿着按此顺序在 win32 上可执行。例如，如果参数是libvpx-1080p ，它将搜索文件libvpx-1080p.ffpreset 。

在上述目录中搜索名为codec\_name -arg .ffpreset 的文件，其中codec\_name是将应用预设文件选项的编解码器的名称。例如，如果您使用-vcodec libvpx选择视频编解码器并使用-vpre 1080p ，那么它将搜索文件libvpx-1080p.ffpreset 。

#### 5.12.2 av预设文件

pre选项指定。它们的工作方式类似于 ffpreset 文件，但它们只允许特定于编码器的选项。因此，不能使用指定编码器的选项=值对。

当指定pre选项时，ffmpeg 将在目录$AVCONV\_DATADIR （如果设置）和$HOME/.avconv以及配置时定义的数据目录（通常是PREFIX/share/ffmpeg ）中查找后缀为 .avpreset 的文件， 以该顺序。

在上述目录中搜索名为codec\_name - arg.avpreset的文件，其中codec\_name是预设文件选项将应用到的编解码器的名称。例如，如果您使用-vcodec libvpx选择视频编解码器并使用-pre 1080p ，那么它将搜索文件libvpx-1080p.avpreset 。

如果找不到这样的文件，则 ffmpeg 将在相同目录中搜索名为arg .avpreset 的文件。

## 6 例子

* [视频和音频抓取](#Video-and-Audio-grabbing)
* [X11抓取](#X11-grabbing)
* [视频和音频文件格式转换](#Video-and-Audio-file-format-conversion)

### 6.1 视频和音频抓取

如果指定输入格式和设备，ffmpeg 可以直接抓取视频和音频。

ffmpeg -f oss -i /dev/dsp -f video4linux2 -i /dev/video0 /tmp/out.mpg

或者使用 ALSA 音频源（单声道输入，卡 id 1）而不是 OSS：

ffmpeg -f alsa -ac 1 -i hw:1 -f video4linux2 -i /dev/video0 /tmp/out.mpg

请注意，在使用任何电视查看器（例如Gerd Knorr 的[xawtv）启动 ffmpeg 之前，您必须激活正确的视频源和频道。](http://linux.bytesex.org/xawtv/)您还必须使用标准调音台正确设置录音电平。

### 6.2 X11抓取

通过 ffmpeg 获取 X11 显示

ffmpeg -f x11grab -video\_size cif -framerate 25 -i :0.0 /tmp/out.mpg

0.0 是 X11 服务器的 display.screen 编号，与 DISPLAY 环境变量相同。

ffmpeg -f x11grab -video\_size cif -framerate 25 -i :0.0+10,20 /tmp/out.mpg

0.0 是 X11 服务器的 display.screen 编号，与 DISPLAY 环境变量相同。 10 是 x 偏移量，20 是抓取的 y 偏移量。

### 6.3 视音频文件格式转换

任何支持的文件格式和协议都可以作为 ffmpeg 的输入：

例子：

* 您可以使用 YUV 文件作为输入：
* ffmpeg -i /tmp/test%dY /tmp/out.mpg

它将使用文件：

/tmp/test0.Y, /tmp/test0.U, /tmp/test0.V,

/tmp/test1.Y、/tmp/test1.U、/tmp/test1.V 等...

Y 文件使用的分辨率是 U 和 V 文件的两倍。它们是原始文件，没有标题。它们可以由所有像样的视频解码器生成。如果 ffmpeg 无法猜测，则必须使用-s选项指定图像的大小。

* 您可以从原始 YUV420P 文件输入：
* ffmpeg -i /tmp/test.yuv /tmp/out.avi

test.yuv 是一个包含原始 YUV 平面数据的文件。每一帧由 Y 平面组成，然后是 U 和 V 平面，分辨率为垂直和水平的一半。

* 您可以输出到原始 YUV420P 文件：
* ffmpeg -i mydivx.avi hugefile.yuv
* 您可以设置多个输入文件和输出文件：
* ffmpeg -i /tmp/a.wav -s 640x480 -i /tmp/a.yuv /tmp/a.mpg

将音频文件 a.wav 和原始 YUV 视频文件 a.yuv 转换为 MPEG 文件 a.mpg。

* 您还可以同时进行音频和视频转换：
* ffmpeg -i /tmp/a.wav -ar 22050 /tmp/a.mp2

以 22050 Hz 采样率将 a.wav 转换为 MPEG 音频。

* 您可以同时编码为多种格式并定义从输入流到输出流的映射：
* ffmpeg -i /tmp/a.wav -map 0:a -b:a 64k /tmp/a.mp2 -map 0:a -b:a 128k /tmp/b.mp2

将 a.wav 转换为 64 kbits 的 a.mp2 和 128 kbits 的 b.mp2。 '-map file:index' 按照输出流定义的顺序指定每个输出流使用哪个输入流。

* 您可以对解密的 VOB 进行转码：
* ffmpeg -i snatch\_1.vob -f avi -c:v mpeg4 -b:v 800k -g 300 -bf 2 -c:a libmp3lame -b:a 128k snatch.avi

这是一个典型的 DVD 翻录示例；输入是 VOB 文件，输出是带有 MPEG-4 视频和 MP3 音频的 AVI 文件。请注意，在此命令中我们使用 B 帧，因此 MPEG-4 流与DivX5 兼容，GOP 大小为 300，这意味着对于 29.97fps 输入视频每 10 秒一个帧内帧。此外，音频流是 MP3 编码的，因此您需要通过将--enable-libmp3lame传递给配置来启用 LAME 支持。该映射对于 DVD 转码以获得所需的音频语言特别有用。

注意：要查看支持的输入格式，请使用ffmpeg -demuxers 。

* 您可以从视频中提取图像，或从许多图像中创建视频：

从视频中提取图像：

ffmpeg -i foo.avi -r 1 -s WxH -f image2 foo-%03d.jpeg

这将从视频中每秒提取一个视频帧，并将它们输出到名为foo-001.jpeg 、 foo-002.jpeg等的文件中。图像将重新缩放以适应新的 WxH 值。

如果只想提取有限数量的帧，可以将上述命令与-frames:v或-t选项结合使用，或者与 -ss 结合使用，从某个时间点开始提取。

从许多图像创建视频：

ffmpeg -f image2 -framerate 12 -i foo-%03d.jpeg -s WxH foo.avi

语法foo-%03d.jpeg指定使用由三位数字组成的十进制数字，并用零填充来表示序列号。它与 C printf 函数支持的语法相同，但只适用于接受普通整数的格式。

导入图像序列时，-i 还支持在内部扩展类似 shell 的通配符模式（通配符），方法是选择 image2-specific -pattern\_type glob选项。

例如，要从匹配 glob 模式foo-\*.jpeg 的文件名创建视频：

ffmpeg -f image2 -pattern\_type glob -framerate 12 -i 'foo-\*.jpeg' -s WxH foo.avi

* 您可以在输出中放置许多相同类型的流：
* ffmpeg -i test1.avi -i test2.avi -map 1:1 -map 1:0 -map 0:1 -map 0:0 -c copy -y test12.nut

生成的输出文件test12.nut将以相反的顺序包含输入文件中的前四个流。

* 强制 CBR 视频输出：
* ffmpeg -i myfile.avi -b 4000k -minrate 4000k -maxrate 4000k -bufsize 1835k out.m2v
* 四个选项 lmin、lmax、mblmin 和 mblmax 使用“lambda”单位，但您可以使用 QP2LAMBDA 常量轻松从“q”单位转换：
* ffmpeg -i src.ext -lmax 21\*QP2LAMBDA dst.ext

## 7 语法

本节记录了 FFmpeg 库和工具使用的语法和格式。

* [引用和转义](#Quoting-and-escaping)
* [日期](#Date)
* [持续时间](#Time-duration)
* [视频大小](#Video-size)
* [视频码率](#Video-rate)
* [比率](#Ratio)
* [颜色](#Color)
* [渠道布局](#Channel-Layout)

### 7.1 引用和转义

除非明确说明，否则 FFmpeg 采用以下引用和转义机制。应用以下规则：

* ' ' '和' \ '是特殊字符（分别用于引号和转义）。除了它们之外，可能还有其他特殊字符，具体取决于使用转义和引用的特定语法。
* 特殊字符通过在其前面加上“ \ ”进行转义。
* 在“ ” ”之间的所有字符都按字面意思包含在已解析的字符串中。引号字符 ' ' ' 本身不能被引用，因此您可能需要关闭引号并将其转义。
* 除非转义或引用，否则前导和尾随空格将从解析的字符串中删除。

请注意，您可能需要在使用命令行或脚本时添加二级转义，这取决于所采用的 shell 语言的语法。

libavutil/avstring.h中定义的函数av\_get\_token可用于根据上面定义的规则解析引用或转义的标记。

工具tools/ffescape可用于自动引用或转义脚本中的字符串。

* [例子](#Examples-2)

#### 7.1.1 例子

* 包含'特殊字符的字符串Crime d'Amour ：
* 爱情犯罪
* 上面的字符串包含一个引号，因此在引用它时需要对'进行转义：
* '犯罪'\''爱情'
* 使用引号包括前导或尾随空格：
* ' 这个字符串以空格开始和结束 '
* 转义和引用可以混合在一起：
* ' 字符串 '\'string\'' 是一个字符串 '
* 要包含文字“ \ ”，您可以使用转义或引用：
* 'c:\foo' 可以写成 c:\\foo

### 7.2 日期

接受的语法是：

[(YYYY-MM-DD|YYYYMMDD)[T|t| ]]((HH:MM:SS[.m...]]])|(HHMMSS[.m...]]]))[Z]

现在

如果值为“现在”，则采用当前时间。

时间是本地时间，除非附加 Z，在这种情况下，它被解释为 UTC。如果未指定年月日部分，则采用当前年月日。

### 7.3 持续时间

有两种可接受的语法来表示持续时间。

[-][ HH :] MM : SS [。米...]

HH表示小时数， MM表示分钟数，最多2位， SS表示秒数，最多2位。最后的 m 表示SS的十进制值。

或者

[-] S +[。 m ...][s|ms|us]

S表示秒数，小数部分m可选。可选的文字后缀“ s ”、“ ms ”或“ us ”分别表示将值解释为秒、毫秒或微秒。

在这两个表达式中，可选的“ - ”表示负持续时间。

* [例子](#Examples-3)

#### 7.3.1 例子

下面的例子都是有效的持续时间：

' 55 '

55秒

' 0.2 '

0.2秒

' 200 毫秒'

200毫秒，也就是0.2s

' 200000us '

200000微秒，也就是0.2s

' 12:03:45 '

12小时03分45秒

' 23.189 '

23.189 秒

### 7.4 视频大小

width x height形式的字符串，或者大小缩写的名称。

可识别以下缩写：

' ntsc '

720x480

'朋友'

720x576

' qntsc '

352x240

' qpal '

352x288

' sntsc '

640x480

' spal '

768x576

'电影'

352x240

' ntsc 电影'

352x240

' sqcif '

128x96

' qcif '

176x144

'到岸价'

352x288

' 4cif '

704x576

' 16cif '

1408x1152

' qqvga '

160x120

' qvga '

320x240

' VGA '

640x480

' svga '

800x600

' xga '

1024x768

' uxga '

1600x1200

' qxga '

2048x1536

' sxga '

1280x1024

' qsxga '

2560x2048

' hsxga '

5120x4096

' wvga '

852x480

' wxga '

1366x768

' wsxga '

1600x1024

' wuxga '

1920x1200

' woxga '

2560x1600

' wqsxga '

3200x2048

' wquxga '

3840x2400

' whsxga '

6400x4096

' whuxga '

7680x4800

' cga '

320x200

'加'

640x350

' hd480 '

852x480

' hd720 '

1280x720

'高清1080 '

1920x1080

' 2k '

2048x1080

' 2kflat '

1998x1080

' 2kscope '

2048x858

' 4k '

4096x2160

' 4kflat '

3996x2160

' 4kscope '

4096x1716

' nhd '

640x360

' hqvga '

240x160

' wqvga '

400x240

' fwqvga '

432x240

' hvga '

480x320

' qhd '

960x540

' 2kdci '

2048x1080

' 4kdci '

4096x2160

' uhd2160 '

3840x2160

' uhd4320 '

7680x4320

### 7.5 视频码率

指定视频的帧速率，表示为每秒生成的帧数。它必须是frame\_rate\_num / frame\_rate\_den格式的字符串、整数、浮点数或有效的视频帧率缩写。

可识别以下缩写：

' ntsc '

30000/1001

'朋友'

25/1

' qntsc '

30000/1001

' qpal '

25/1

' sntsc '

30000/1001

' spal '

25/1

'电影'

24/1

' ntsc 电影'

24000/1001

### 7.6 比例

比率可以表示为表达式，或以numerator : denominator 的形式表示。

请注意，具有无穷大 (1/0) 或负值的比率被认为是有效的，因此如果您想排除这些值，您应该检查返回的值。

未定义的值可以使用“0:0”字符串表示。

### 7.7 颜色

它可以是下面定义的颜色名称（不区分大小写的匹配）或[0x|#]RRGGBB[AA]序列，可能后跟 @ 和表示 alpha 分量的字符串。

alpha分量可以是由“0x”后跟一个介于0.0和1.0之间的十六进制数或十进制数组成的字符串，表示不透明度值（'0x00'或'0.0'表示完全透明， ' 0xff '或' 1.0 '完全不透明）。如果未指定 alpha 分量，则假定为“ 0xff ”。

字符串“ random ”将产生随机颜色。

识别以下颜色名称：

'爱丽丝蓝'

0xF0F8FF

'古董白'

0xFAEBD7

'水'

0x00FFFF

'海蓝宝石'

0x7FFFD4

'蔚蓝'

0xF0FFFF

'米色'

0xF5F5DC

'浓汤'

0xFFE4C4

'黑色'

0x000000

'漂白杏仁'

0xFFEBCD

'蓝色'

0x0000FF

'蓝紫罗兰'

0x8A2BE2

'棕色'

0xA52A2A

' BurlyWood '

0xDEB887

'学员蓝'

0x5F9EA0

'黄绿色'

0x7FFF00

'巧克力'

0xD2691E

'珊瑚'

0xFF7F50

'矢车菊蓝'

0x6495ED

'玉米须'

0xFFF8DC

'深红'

0xDC143C

'青色'

0x00FFFF

'深蓝'

0x00008B

'深青色'

0x008B8B

'暗金棒'

0xB8860B

'深灰色'

0xA9A9A9

'墨绿'

0x006400

'深卡其色'

0xBDB76B

'深洋红色'

0x8B008B

'深橄榄绿'

0x556B2F

'深橙色'

0xFF8C00

'黑暗兰花'

0x9932CC

'深红'

0x8B0000

'黑鲑鱼'

0xE9967A

'深海绿'

0x8FBC8F

'暗石蓝'

0x483D8B

'暗石灰'

0x2F4F4F

'深绿松石'

0x00CED1

'暗紫罗兰'

0x9400D3

'深粉红色'

0xFF1493

'深天蓝'

0x00BFFF

'暗灰色'

0x696969

'道奇蓝'

0x1E90FF

'耐火砖'

0xB22222

'花白'

0xFFFAF0

'森林绿'

0x228B22

'紫红色'

0xFF00FF

'盖恩斯伯勒'

0xDCDCDC

'幽灵白'

0xF8F8FF

'黄金'

0xFFD700

'金棒'

0xDAA520

'灰色'

0x808080

'绿色'

0x008000

'绿黄'

0xADFF2F

'蜜露'

0xF0FFF0

'热粉色'

0xFF69B4

'印度红'

0xCD5C5C

'靛蓝'

0x4B0082

'象牙'

0xFFFFF0

'卡其色'

0xF0E68C

'薰衣草'

0xE6E6FA

'薰衣草腮红'

0xFFF0F5

'草坪绿'

0x7CFC00

'柠檬雪纺'

0xFFFACD

'浅蓝'

0xADD8E6

'光珊瑚'

0xF08080

'浅青色'

0xE0FFFF

'浅金棒黄'

0xFAFAD2

'浅绿色'

0x90EE90

'浅灰色'

0xD3D3D3

'浅粉色'

0xFFB6C1

'轻鲑鱼'

0xFFA07A

' LightSeaGreen '

0x20B2AA

'浅天蓝'

0x87CEFA

' LightSlateGray '

0x778899

'轻钢蓝'

0xB0C4DE

'淡黄色'

0xFFFFE0

'石灰'

0x00FF00

'石灰绿'

0x32CD32

'亚麻布'

0xFAF0E6

'洋红色'

0xFF00FF

'栗色'

0x800000

' MediumAquaMarine '

0x66CDAA

'中蓝'

0x0000CD

' MediumOrchid '

0xBA55D3

'中紫'

0x9370D8

' MediumSeaGreen '

0x3CB371

' MediumSlateBlue '

0x7B68EE

' MediumSpringGreen '

0x00FA9A

' MediumTurquoise '

0x48D1CC

' MediumVioletRed '

0xC71585

'午夜蓝'

0x191970

'薄荷霜'

0xF5FFFA

'迷雾玫瑰'

0xFFE4E1

'鹿皮鞋'

0xFFE4B5

'纳瓦霍白人'

0xFF死

'海军'

0x000080

'老花边'

0xFDF5E6

'橄榄'

0x808000

' OliveDrab '

0x6B8E23

'橙色'

0xFFA500

'橙红'

0xFF4500

'兰花'

0xDA70D6

'淡金棒'

0xEEE8AA

'淡绿色'

0x98FB98

'淡绿松石'

0xAFEEEE

'淡紫红色'

0xD87093

'木瓜鞭'

0xFFEFD5

'桃子泡芙'

0xFFDAB9

'秘鲁人'

0xCD853F

'粉红色'

0xFFC0CB

'李子'

0xDDA0DD

'粉蓝'

0xB0E0E6

'紫色'

0x800080

'红色'

0xFF0000

'玫瑰色'

0xBC8F8F

'宝蓝'

0x4169E1

'马鞍布朗'

0x8B4513

'鲑鱼'

0xFA8072

'桑迪布朗'

0xF4A460

'海绿'

0x2E8B57

'贝壳'

0xFFF5EE

'西耶娜'

0xA0522D

'银'

0xC0C0C0

'天蓝'

0x87CEEB

'石板蓝'

0x6A5ACD

'石板灰'

0x708090

'雪'

0xFFFAFA

'春绿'

0x00FF7F

'钢铁蓝'

0x4682B4

'谭'

0xD2B48C

'蓝绿色'

0x008080

'蓟'

0xD8BFD8

'番茄'

0xFF6347

'绿松石'

0x40E0D0

'紫罗兰'

0xEE82EE

'小麦'

0xF5DEB3

'白色'

0xFFFFFF

'白烟'

0xF5F5F5

'黄色'

0xFFFF00

'黄绿'

0x9ACD32

### 7.8 频道布局

通道布局指定多通道音频流中通道的空间布置。为了指定通道布局，FFmpeg 使用了一种特殊的语法。

各个频道由 id 标识，如下表所示：

'佛罗里达州'

左前

' FR '

右前方

' FC '

前中心

'低频'

低频

' BL '

左后

' BR '

右后

' FLC '

前中偏左

'财务报告委员会'

前中右

'公元前'

后中心

' SL '

左边

' SR '

右边

' TC '

顶部中心

' TFL '

左上前

' TFC '

顶前中心

'总生育率'

右上前

'待定'

左上角

'待定'

顶部后中心

' TBR '

右上角

' DL '

缩混左

'博士'

右缩混

' WL '

左宽

'写'

右宽

' SDL '

直接左环绕

'特别提款权'

直接右环绕

' LFE2 '

低频2

可以使用以下标识符指定标准通道布局组合：

'单声道'

FC

'立体声'

飞行+飞行

' 2.1 '

FL+FR+LFE

' 3.0 '

FL+FR+FC

' 3.0（返回） '

FL+FR+BC

' 4.0 '

FL+FR+FC+BC

'四'

FL+FR+BL+BR

'四边形（边） '

FL+FR+SL+SR

' 3.1 '

FL+FR+FC+LFE

' 5.0 '

FL+FR+FC+BL+BR

' 5.0（边） '

FL+FR+FC+SL+SR

' 4.1 '

FL+FR+FC+LFE+BC

' 5.1 '

FL+FR+FC+LFE+BL+BR

' 5.1（边） '

FL+FR+FC+LFE+SL+SR

' 6.0 '

FL+FR+FC+BC+SL+SR

' 6.0（前） '

FL+FR+FLC+FRC+SL+SR

'六角形'

FL+FR+FC+BL+BR+BC

' 6.1 '

FL+FR+FC+LFE+BC+SL+SR

' 6.1 '

FL+FR+FC+LFE+BL+BR+BC

' 6.1（前） '

FL+FR+LFE+FLC+FRC+SL+SR

' 7.0 '

FL+FR+FC+BL+BR+SL+SR

' 7.0（前） '

FL+FR+FC+FLC+FRC+SL+SR

' 7.1 '

FL+FR+FC+LFE+BL+BR+SL+SR

' 7.1（宽） '

FL+FR+FC+LFE+BL+BR+FLC+FRC

' 7.1（宽边） '

FL+FR+FC+LFE+FLC+FRC+SL+SR

' 7.1（顶部） '

FL+FR+FC+LFE+BL+BR+TFL+TFR

'八角形'

FL+FR+FC+BL+BR+BC+SL+SR

'立方体'

FL+FR+BL+BR+TFL+TFR+TBL+TBR

'十六边形'

FL+FR+FC+BL+BR+BC+SL+SR+WL+WR+TBL+TBR+TBC+TFC+TFL+TFR

'缩混'

DL+DR

'22 .2 '

FL+FR+FC+LFE+BL+BR+FLC+FRC+BC+SL+SR+TC+TFL+TFC+TFR+TBL+TBC+TBR+LFE2+TSL+TSR+BFC+BFL+BFR

自定义频道布局可以指定为一系列术语，以“+”分隔。每个术语可以是：

* 单个通道的名称（例如“ FL ”、“ FR ”、“ FC ”、“ LFE ”等），每个通道可选地在“@”之后包含一个自定义名称（例如“ FL@Left ”、“ FR” @Right '、' FC@Center '、' LFE@Low\_Frequency ' 等）

可以通过以下方式指定标准通道布局：

* 单个通道的名称（例如' FL '、' FR '、' FC '、' LFE '等）
* 标准通道布局的名称（例如“ mono ”、“ stereo ”、“ 4.0 ”、“ quad ”、“ 5.0 ”等）
* 多个通道，以十进制表示，后跟“c”，产生该数量通道的默认通道布局（请参阅函数av\_channel\_layout\_default ）。请注意，并非所有通道数都有默认布局。
* 多个通道，以十进制表示，后跟“C”，产生具有指定通道数的未知通道布局。请注意，并非所有频道布局规范字符串都支持未知频道布局。
* 通道布局掩码，以“0x”开头的十六进制（请参阅libavutil/channel\_layout.h中的AV\_CH\_\*宏。

在 libavutil 版本 53 之前，用于指定多个通道的尾随字符“c”是可选的，但现在它是必需的，同时通道布局掩码也可以指定为十进制数（当且仅当后面没有“c”或“C”）。

libavutil/channel\_layout.h中定义的函数av\_channel\_layout\_from\_string 。

## 8 表达式评估

计算算术表达式时，FFmpeg 使用内部公式计算器，通过libavutil/eval.h接口实现。

表达式可以包含一元、二元运算符、常量和函数。

两个表达式expr1和expr2可以组合形成另一个表达式“ expr1 ; expr2 ”。依次计算expr1和expr2 ，新表达式的计算结果为expr2的值。

以下二元运算符可用： + 、 - 、 \* 、 / 、 ^ 。

以下一元运算符可用： + , - 。

以下功能可用：

绝对值（x）

x的绝对值。

acos(x)

计算x的反余弦。

asin(x)

计算x的反正弦。

阿坦(x)

计算x的反正切。

atan2(x, y)

y / x的反正切值的主值。

之间（x，最小值，最大值）

如果x大于或等于min且小于或等于max ，则返回 1，否则返回 0。

位和（x，y）

比特（x，y）

x和y进行按位计算和/或运算。

在执行按位运算之前， x和y的计算结果被转换为整数。

请注意，转换为整数和转换回浮点数都可能会丢失精度。当心大量（通常为 2^53 和更大）的意外结果。

上限（表达式）

expr的值向上舍入到最接近的整数。例如，“ceil(1.5)”是“2.0”。

剪辑（x，最小值，最大值）

介于min和max之间的x值。

余弦（x）

计算x的余弦。

cosh(x)

x的双曲余弦。

当量（x，y）

x和y相等则返回 1 ，否则返回 0。

指数（x）

计算x的指数（以e为底，欧拉数）。

地板（expr）

expr的值向下舍入到最接近的整数。例如，“floor(-1.5)”是“-2.0”。

高斯(x)

x的高斯函数，对应于exp(-x\*x/2) / sqrt(2\*PI) 。

gcd(x, y)

x和y的最大公约数。如果x和y均为 0 或其中之一或两者都小于零，则行为未定义。

GT(x, y)

x大于y则返回 1 ，否则返回 0。

gte(x, y)

x大于或等于y则返回 1 ，否则返回 0。

hypot(x, y)

这个函数类似于同名的C函数；它返回“sqrt( x \* x + y \* y )”，边长为x和y的直角三角形斜边的长度，或点 ( x , y ) 与原点的距离。

如果（x，y）

评估x ，如果结果非零，则返回y的评估结果，否则返回 0 。

如果（x，y，z）

计算x ，如果结果非零则返回y的计算结果，否则返回z的计算结果。

如果不是（x，y）

评估x ，如果结果为零，则返回y的评估结果，否则返回 0。

如果不是（x，y，z）

评估x ，如果结果为零，则返回y的评估结果，否则返回z的评估结果。

isinf(x)

如果x为 +/-INFINITY，则返回 1.0，否则返回 0.0。

伊南(x)

x是 NAN，则返回 1.0 ，否则返回 0.0。

ld(变量)

var加载内部变量的值，该值之前使用 st( var , expr ) 存储。该函数返回加载的值。

勒普（x，y，z）

z的量返回x和y之间的线性插值。

日志（x）

计算x的自然对数。

lt(x, y)

x小于y则返回 1 ，否则返回 0。

lte(x, y)

如果x小于或等于y则返回 1 ，否则返回 0。

最大（x，y）

x和y之间的最大值。

最小值（x，y）

x和y之间的最小值。

模型（x，y）

x除以y的余数。

不是（表达式）

expr为零，则返回 1.0 ，否则返回 0.0。

战俘（X，Y）

x提升y的幂，它等同于“( x )^( y )”。

打印（t）

打印（t，l）

日志级别为l的表达式t的值。如果未指定l ，则使用默认日志级别。返回打印的表达式的值。

以日志级别 l 打印 t

随机（x）

返回一个介于 0.0 和 1.0 之间的伪随机值。 x是将用于保存种子/状态的内部变量的索引。

根（表达式，最大值）

在区间 0.. max中找到一个输入值，其参数为ld(0)的expr表示的函数为 0 。

expr中的表达式必须表示连续函数，否则结果未定义。

ld(0)用于表示函数输入值，这意味着将使用表达式可以通过ld(0)访问的各种输入值对给定表达式进行多次求值。当表达式的计算结果为 0 时，将返回相应的输入值。

圆形（表达式）

expr的值四舍五入为最接近的整数。例如，“round(1.5)”是“2.0”。

信号（x）

计算x的符号。

罪恶（x）

计算x的正弦值。

辛（x）

计算x的双曲正弦值。

平方根（表达式）

expr的平方根。这等同于“( expr )^.5”。

挤压（x）

计算表达式1/(1 + exp(4\*x)) 。

st（变量，表达式）

expr的值存储在内部变量中。 var指定存储值的变量编号，取值范围为 0 到 9。函数返回存储在内部变量中的值。请注意，变量目前不在表达式之间共享。

tan(x)

计算x的正切。

tanh(x)

计算x的双曲正切值。

泰勒（表达式，x）

泰勒 (expr, x, id)

ld(id)阶导数的表达式，计算x处的泰勒级数。

当级数不收敛时，结果是不确定的。

ld(id)用于表示expr中的导数顺序，这意味着将使用表达式可以通过ld(id)访问的各种输入值对给定表达式进行多次求值。如果未指定id ，则假定为 0。

请注意，当您在 y 而不是 0 处获得导数时，可以使用taylor(expr, xy) 。

时间(0)

以秒为单位返回当前（挂钟）时间。

截断（表达式）

expr的值向零舍入到最接近的整数。例如，“trunc(-1.5)”是“-1.0”。

同时（条件，表达式）

在表达式cond为非零时计算表达式expr ，并返回最后一次expr计算的值，如果cond始终为假，则返回 NAN。

以下常量可用：

PI

单位圆盘的面积，大约 3.14

乙

exp(1)（欧拉数），大约为 2.718

个人健康指数

黄金比例(1+sqrt(5))/2，约1.618

假设如果表达式具有非零值，则该表达式被视为“真”，请注意：

\*像 AND 一样工作

+像或一样工作

例如构造：

如果（A 和 B）那么 C

相当于：

如果（A \* B，C）

在您的 C 代码中，您可以扩展一元和二元函数列表，并定义可识别的常量，以便它们可用于您的表达式。

评估器还识别国际系统单位前缀。如果'i'附加在前缀之后，则使用二进制前缀，它基于1024的幂而不是1000的幂。'B'后缀将值乘以8，可以附加在单位前缀之后或单独使用.这允许使用例如“KB”、“MiB”、“G”和“B”作为数字后缀。

可用的国际系统前缀列表如下，并指示相应的 10 和 2 的幂。

是

10^-24 / 2^-80

z

10^-21 / 2^-70

A

10^-18 / 2^-60

F

10^-15 / 2^-50

p

10^-12 / 2^-40

n

10^-9 / 2^-30

你

10^-6 / 2^-20

米

10^-3 / 2^-10

C

10^-2

d

10^-1

H

10^2

k

10^3 / 2^10

钾

10^3 / 2^10

米

10^6 / 2^20

G

10^9 / 2^30

吨

10^12 / 2^40

P

10^15 / 2^50

乙

10^18 / 2^60

Z

10^21 / 2^70

是

10^24 / 2^80

## 9 编解码器选项

libavcodec 提供了一些通用的全局选项，可以在所有编码器和解码器上设置。此外，每个编解码器都可以支持所谓的私有选项，这些选项特定于给定的编解码器。

有时，全局选项可能只影响特定种类的编解码器，可能无意义或被其他编解码器忽略，因此您需要了解指定选项的含义。还有一些选项仅用于解码或编码。

-option来设置选项 FFmpeg 工具中的值，或通过在AVCodecContext选项中显式设置值或使用libavutil/opt.h API 进行编程使用。

支持的选项列表如下：

b整数（编码、音频、视频）

以比特/秒为单位设置比特率。默认值为 200K。

ab整数（编码、音频）

设置音频比特率（以比特/秒为单位）。默认值为 128K。

bt整数（编码、视频）

设置视频比特率容差（以比特/秒为单位）。在 1-pass 模式下，比特率容限指定速率控制愿意偏离目标平均比特率值的程度。这与最小/最大比特率无关。过分降低公差会对质量产生不利影响。

标志标志（解码/编码、音频、视频、字幕）

设置通用标志。

可能的值：

' mv4 '

按宏块 (mpeg4) 使用四个运动矢量。

' qpel '

使用 1/4 像素运动补偿。

'循环'

使用循环过滤器。

'质量等级'

使用固定的 qscale。

'通过1 '

在第一遍模式下使用内部 2pass 速率控制。

' pass2 '

在第二遍模式下使用内部 2pass 速率控制。

'灰色'

仅解码/编码灰度。

' psnr '

在编码期间设置 error[?] 变量。

'截断'

输入比特流可能会被随机截断。

' drop\_changed '

不要输出参数与流中第一个解码帧不同的帧。丢帧时返回错误 AVERROR\_INPUT\_CHANGED。

' ildct '

使用隔行 DCT。

'低延迟'

强制低延迟。

' global\_header '

将全局标头放在 extradata 而不是每个关键帧中。

'双精确'

仅写入平台、构建和时间无关的数据。 （除了 (I)DCT）。这可确保文件和数据校验和可重现并在平台之间匹配。它的主要用途是回归测试。

'艾克'

应用 H263 高级帧内编码/mpeg4 ac 预测。

'伊尔梅'

应用隔行扫描运动估计。

' cgop '

使用封闭的 gop。

' output\_corrupt '

输出甚至可能损坏的帧。

时基有理数

设置编解码器时基。

它是表示帧时间戳的基本时间单位（以秒为单位）。对于固定 fps 内容，时基应为1 / frame\_rate ，时间戳增量应相同为 1。

g整数（编码、视频）

设置图片组 (GOP) 大小。默认值为 12。

ar整数（解码/编码，音频）

设置音频采样率（以 Hz 为单位）。

ac整数（解码/编码，音频）

设置音频通道数。

截止整数（编码，音频）

设置截止带宽。 （仅由选定的编码器支持，请参阅其各自的文档部分。）

frame\_size整数（编码、音频）

设置音频帧大小。

除了最后一个之外，每个提交的帧都必须包含每个通道的 frame\_size 个样本。当编解码器设置了 CODEC\_CAP\_VARIABLE\_FRAME\_SIZE 时可能为 0，在这种情况下帧大小不受限制。它由一些解码器设置以指示恒定的帧大小。

frame\_number整数

设置帧数。

延迟整数

qcomp浮点数（编码、视频）

设置视频量化比例压缩 (VBR)。它在速率控制方程中用作常数。默认 rc\_eq 的推荐范围：0.0-1.0。

qblur浮点数（编码、视频）

设置视频量化器尺度模糊 (VBR)。

qmin整数（编码、视频）

设置最小视频量化比例 (VBR)。必须包含在 -1 到 69 之间，默认值为 2。

qmax整数（编码、视频）

设置最大视频量化比例 (VBR)。必须包含在 -1 到 1024 之间，默认值为 31。

qdiff整数（编码、视频）

设置量化标度 (VBR) 之间的最大差异。

bf整数（编码，视频）

设置非 B 帧之间的最大 B 帧数。

必须是 -1 到 16 之间的整数。0 表示禁用 B 帧。如果使用值 -1，它将根据编码器选择一个自动值。

默认值为 0。

b\_qfactor float （编码，视频）

设置 P 和 B 帧之间的 qp 因子。

codec\_tag整数

错误标志（解码、视频）

解决方法不是自动检测到编码器错误。

可能的值：

'自动检测'

' xvid\_ilace '

Xvid 隔行扫描错误（如果 fourcc==XVIX 则自动检测）

' ump4 '

（如果 fourcc==UMP4 则自动检测）

'无填充'

填充错误（自动检测）

' amv '

' qpel\_chroma '

' std\_qpel '

旧标准 qpel（根据 fourcc/版本自动检测）

' qpel\_chroma2 '

' direct\_blocksize '

direct-qpel-blocksize 错误（根据 fourcc/版本自动检测）

'边缘'

边缘填充错误（根据 fourcc/版本自动检测）

' hpel\_chroma '

' dc\_clip '

'女士'

解决 Microsoft 损坏的解码器中的各种错误。

'截断'

截断帧

严格整数（解码/编码、音频、视频）

指定如何严格遵守标准。

可能的值：

'非常'

严格遵守规范或参考软件的更早更严格的版本

'严格'

无论后果如何，都严格遵守规范中的所有内容

'正常'

'非官方'

允许非官方扩展

'实验'

允许非标准化的实验性事物，实验性（未完成/进行中/未经过充分测试）解码器和编码器。注意：实验解码器可能会带来安全风险，请勿将其用于解码不受信任的输入。

b\_qoffset浮点数（编码、视频）

设置 P 帧和 B 帧之间的 QP 偏移量。

err\_detect标志（解码、音频、视频）

设置错误检测标志。

可能的值：

' crccheck '

验证嵌入式 CRC

'比特流'

检测比特流规范偏差

'缓冲区'

检测不正确的比特流长度

'爆炸'

检测到小错误时中止解码

'忽略错误'

忽略解码错误，继续解码。如果您想分析视频的内容并因此希望无论如何都解码所有内容，这将很有用。如果出现错误，此选项将不会生成令人赏心悦目的视频。

'小心'

将违反规范且在野外未被视为错误的事物视为错误

'合规'

将所有规范不合规视为错误

'咄咄逼人'

将理智的编码器不应做的事情视为错误

has\_b\_frames整数

block\_align整数

rc\_override\_count整数

最大速率整数（编码、音频、视频）

设置最大比特率容差（以比特/秒为单位）。需要设置 bufsize。

minrate整数（编码、音频、视频）

设置最小比特率容差（以比特/秒为单位）。在设置 CBR 编码时最有用。否则用处不大。

bufsize整数（编码、音频、视频）

设置速率控制缓冲区大小（以位为单位）。

i\_qfactor浮点数（编码、视频）

设置 P 和 I 帧之间的 QP 因子。

i\_qoffset浮点数（编码、视频）

设置 P 和 I 帧之间的 QP 偏移量。

dct整数（编码、视频）

设置 DCT 算法。

可能的值：

'汽车'

自动选择一个好的（默认）

'禁食'

快速整数

'整数'

准确的整数

' mmx '

' altivec '

'法安'

浮点 AAN DCT

lumi\_mask浮动（编码，视频）

压缩明亮区域比中等区域更强。

tcplx\_mask浮点数（编码、视频）

设置时间复杂度掩码。

scplx\_mask浮点数（编码、视频）

设置空间复杂度掩码。

p\_mask浮动（编码，视频）

设置帧间屏蔽。

dark\_mask浮点数（编码、视频）

压缩黑暗区域比中等区域更强。

idct整数（解码/编码，视频）

选择 IDCT 实施。

可能的值：

'汽车'

'整数'

'简单'

' simplemmx '

'简单自动'

自动挑选与简单兼容的IDCT

'手臂'

' altivec '

' sh4 '

'简单的手臂'

' simplearmv5te '

' simplearmv6 '

'简单霓虹灯'

' xvid '

'法尼'

浮点 AAN IDCT

slice\_count整数

ec标志（解码、视频）

设置错误隐藏策略。

可能的值：

' guess\_mvs '

迭代运动矢量 (MV) 搜索（慢）

'解封'

对损坏的 MB 使用强大的去块过滤器

' favor\_inter '

倾向于从前一帧而不是当前帧进行预测

bits\_per\_coded\_sample整数

方面有理数（编码，视频）

设置样本纵横比。

sar有理数（编码、视频）

设置样本纵横比。 aspect的别名。

调试标志（解码/编码、音频、视频、字幕）

打印特定的调试信息。

可能的值：

'图片'

图片信息

' rc '

速率控制

'比特流'

' mb\_type '

宏块 (MB) 类型

' qp '

每块量化参数 (QP)

' dct\_coeff '

'绿色元数据'

显示即将到来的帧、GoP 或给定持续时间的复杂性元数据。

'跳过'

'开始代码'

‘儿’

错误识别

' mmco '

内存管理控制操作 (H.264)

'错误'

'缓冲区'

图片缓冲区分配

' thread\_ops '

线程操作

'命名'

跳跃运动补偿

cmp整数（编码、视频）

设置完整的 pel me 比较功能。

可能的值：

'悲伤'

绝对差之和，快速（默认）

' sse '

误差平方和

'饱和'

绝对 Hadamard 变换差异之和

' dct '

绝对 DCT 转换差值之和

' psnr '

量化误差平方和（避免，低质量）

'位'

块所需的位数

'路'

率失真最佳，慢

'零'

0

' vsad '

绝对垂直差之和

' vs '

垂直差平方和

' nsse '

噪声保留平方差和

' w53 '

5/3 小波，仅用于雪地

' w97 '

9/7 小波，仅用于雪地

' dctmax '

'色度'

subcmp整数（编码、视频）

设置子像素比较功能。

可能的值：

'悲伤'

绝对差之和，快速（默认）

' sse '

误差平方和

'饱和'

绝对 Hadamard 变换差异之和

' dct '

绝对 DCT 转换差值之和

' psnr '

量化误差平方和（避免，低质量）

'位'

块所需的位数

'路'

率失真最佳，慢

'零'

0

' vsad '

绝对垂直差之和

' vs '

垂直差平方和

' nsse '

噪声保留平方差和

' w53 '

5/3 小波，仅用于雪地

' w97 '

9/7 小波，仅用于雪地

' dctmax '

'色度'

mbcmp整数（编码、视频）

设置宏块比较功能。

可能的值：

'悲伤'

绝对差之和，快速（默认）

' sse '

误差平方和

'饱和'

绝对 Hadamard 变换差异之和

' dct '

绝对 DCT 转换差值之和

' psnr '

量化误差平方和（避免，低质量）

'位'

块所需的位数

'路'

率失真最佳，慢

'零'

0

' vsad '

绝对垂直差之和

' vs '

垂直差平方和

' nsse '

噪声保留平方差和

' w53 '

5/3 小波，仅用于雪地

' w97 '

9/7 小波，仅用于雪地

' dctmax '

'色度'

ildctcmp整数（编码、视频）

设置隔行dct比较功能。

可能的值：

'悲伤'

绝对差之和，快速（默认）

' sse '

误差平方和

'饱和'

绝对 Hadamard 变换差异之和

' dct '

绝对 DCT 转换差值之和

' psnr '

量化误差平方和（避免，低质量）

'位'

块所需的位数

'路'

率失真最佳，慢

'零'

0

' vsad '

绝对垂直差之和

' vs '

垂直差平方和

' nsse '

噪声保留平方差和

' w53 '

5/3 小波，仅用于雪地

' w97 '

9/7 小波，仅用于雪地

' dctmax '

'色度'

dia\_size整数（编码、视频）

设置运动估计的钻石类型和大小。

' (1024, INT\_MAX) '

全运动估计（最慢）

' (768, 1024] '

嗯运动估计

' (512, 768] '

六边形运动估计

' (256, 512] '

l2s菱形运动估计

' [2,256] '

var 菱形运动估计

' (-1, 2) '

小钻石运动估计

' -1 '

有趣的钻石运动估计

' (INT\_MIN, -1) '

sab 钻石运动估计

last\_pred整数（编码、视频）

设置前一帧的运动预测量。

precmp整数（编码、视频）

设置运动前估计比较功能。

可能的值：

'悲伤'

绝对差之和，快速（默认）

' sse '

误差平方和

'饱和'

绝对 Hadamard 变换差异之和

' dct '

绝对 DCT 转换差值之和

' psnr '

量化误差平方和（避免，低质量）

'位'

块所需的位数

'路'

率失真最佳，慢

'零'

0

' vsad '

绝对垂直差之和

' vs '

垂直差平方和

' nsse '

噪声保留平方差和

' w53 '

5/3 小波，仅用于雪地

' w97 '

9/7 小波，仅用于雪地

' dctmax '

'色度'

pre\_dia\_size整数（编码、视频）

设置运动估计预通过的钻石类型和大小。

subq整数（编码、视频）

设置子像素运动估计质量。

me\_range整数（编码、视频）

设置限制运动矢量范围（DivX 播放器为 1023）。

global\_quality整数（编码、音频、视频）

slice\_flags整数

mbd整数（编码、视频）

设置宏块决策算法（高质量模式）。

可能的值：

'简单'

使用 mbcmp（默认）

'位'

使用最少的位

'路'

使用最佳率失真

rc\_init\_occupancy整数（编码、视频）

设置解码开始前应加载到 rc 缓冲区的位数。

flags2标志（解码/编码、音频、视频、字幕）

可能的值：

'快'

允许不符合规范的加速技巧。

'没有'

跳过比特流编码。

'忽略作物'

忽略来自 sps 的裁剪信息。

' local\_header '

将全局标头放在每个关键帧而不是额外数据中。

'大块'

帧数据可能被分成多个块。

' showall '

显示第一个关键帧之前的所有帧。

' export\_mvs '

支持它的编解码器的帧边数据（请参阅AV\_FRAME\_DATA\_MOTION\_VECTORS ）。另请参阅doc/examples/export\_mvs.c 。

'跳过手册'

不要跳过样本并将跳过信息导出为帧端数据。

' ass\_ro\_flush\_noop '

不要在刷新时重置 ASS ReadOrder 字段。

' icc\_profiles '

从比色标签生成/解析嵌入的 ICC 配置文件。

export\_side\_data标志（解码/编码、音频、视频、字幕）

可能的值：

' MV '

支持它的编解码器的帧边数据（请参阅AV\_FRAME\_DATA\_MOTION\_VECTORS ）。另请参阅doc/examples/export\_mvs.c 。

'公关'

将编码器生产者参考时间导出到支持它的编解码器的数据包端数据（请参阅AV\_PKT\_DATA\_PRFT ）。

' venc\_params '

通过帧端数据导出视频编码参数（参见AV\_FRAME\_DATA\_VIDEO\_ENC\_PARAMS ）以获取支持它的编解码器。目前，它们是 H.264 和 VP9。

'胶片颗粒'

通过帧侧数据导出胶片颗粒参数（参见AV\_FRAME\_DATA\_FILM\_GRAIN\_PARAMS ）。目前支持AV1解码器。

线程整数（解码/编码，视频）

设置要使用的线程数，以防所选编解码器实现支持多线程。

可能的值：

'自动，0 '

自动选择要设置的线程数

默认值为“自动”。

直流整数（编码，视频）

设置 intra\_dc\_precision。

nssew整数（编码、视频）

设置 nsse 权重。

skip\_top整数（解码、视频）

设置顶部被跳过的宏块行数。

skip\_bottom整数（解码、视频）

设置底部被跳过的宏块行数。

配置文件整数（编码、音频、视频）

设置编码器编解码器配置文件。默认值为“未知”。编码器特定配置文件记录在相关编码器文档中。

级别整数（编码、音频、视频）

可能的值：

'未知'

低分辨率整数（解码、音频、视频）

以 1= 1/2、2=1/4、3=1/8 分辨率解码。

mblmin整数（编码、视频）

设置最小宏块拉格朗日因子 (VBR)。

mblmax整数（编码、视频）

设置最大宏块拉格朗日因子 (VBR)。

skip\_loop\_filter整数（解码、视频）

skip\_idct整数（解码、视频）

skip\_frame整数（解码、视频）

根据选项值选择的帧类型进行解码器丢弃处理。

skip\_loop\_filter跳过帧循环过滤， skip\_idct跳过帧 IDCT/反量化， skip\_frame跳过解码。

可能的值：

'无'

不丢弃任何框架。

'默认'

丢弃无用的帧，如 0 大小的帧。

' noref '

丢弃所有非参考帧。

'双向'

丢弃所有双向帧。

'诺基'

丢弃除关键帧之外的所有帧。

' nointra '

丢弃除 I 帧之外的所有帧。

'全部'

丢弃所有帧。

默认值为“默认”。

bidir\_refine整数（编码、视频）

细化双向宏块中使用的两个运动矢量。

keyint\_min整数（编码、视频）

设置 IDR 帧之间的最小间隔。

参考整数（编码，视频）

设置参考帧以考虑运动补偿。

网格整数（编码、音频、视频）

设置率失真最佳量化。

mv0\_threshold整数（编码、视频）

compression\_level整数（编码、音频、视频）

bits\_per\_raw\_sample整数

channel\_layout整数（解码/编码，音频）

可能的值：

request\_channel\_layout整数（解码，音频）

可能的值：

rc\_max\_vbv\_use浮点数（编码、视频）

rc\_min\_vbv\_use浮点数（编码、视频）

ticks\_per\_frame整数（解码/编码、音频、视频）

color\_primaries整数（解码/编码，视频）

可能的值：

' bt709 '

BT.709

' bt470m '

BT.470M

' bt470bg '

BT.470 BG

' smpte170m '

SMPTE 170 M

' smpte240m '

SMPTE 240 米

'电影'

电影

' bt2020 '

BT.2020

' smpte428 '

' smpte428\_1 '

SMPTE ST 428-1

' smpte431 '

SMPTE 431-2

' smpte432 '

SMPTE 432-1

' jedec-p22 '

JEDEC P22

color\_trc整数（解码/编码，视频）

可能的值：

' bt709 '

BT.709

'伽玛22 '

BT.470M

'伽玛28 '

BT.470 BG

' smpte170m '

SMPTE 170 M

' smpte240m '

SMPTE 240 米

'线性'

线性的

'日志'

' log100 '

日志

' log\_sqrt '

' log316 '

对数平方根

' iec61966\_2\_4 '

' iec61966-2-4 '

IEC 61966-2-4

' bt1361 '

' bt1361e '

BT.1361

' iec61966\_2\_1 '

' iec61966-2-1 '

IEC 61966-2-1

' bt2020\_10 '

' bt2020\_10bit '

BT.2020 - 10 位

' bt2020\_12 '

' bt2020\_12bit '

BT.2020 - 12 位

' smpte2084 '

SMPTE ST 2084 格式

' smpte428 '

' smpte428\_1 '

SMPTE ST 428-1

' arib-std-b67 '

ARIB标准-B67

颜色空间整数（解码/编码，视频）

可能的值：

' RGB '

红绿蓝

' bt709 '

BT.709

' fcc '

美国联邦通信委员会

' bt470bg '

BT.470 BG

' smpte170m '

SMPTE 170 M

' smpte240m '

SMPTE 240 米

' ycocg '

YCOCG

' bt2020nc '

' bt2020\_ncl '

BT.2020 NCL

' bt2020c '

' bt2020\_cl '

BT.2020 CL

' smpte2085 '

SMPTE 2085 格式

'色度衍生-nc '

色度衍生的 NCL

'色度衍生-c '

色度衍生的 CL

' ictcp '

ICtCp

color\_range整数（解码/编码，视频）

如果用作输入参数，它可以作为解码器的提示，输入具有哪个 color\_range。可能的值：

'电视'

' mpeg '

MPEG (219\*2^(n-8))

'电脑'

' jpeg '

JPEG (2^n-1)

chroma\_sample\_location整数（解码/编码，视频）

可能的值：

'离开'

'中心'

'左上角'

'顶'

'左下角'

'底部'

log\_level\_offset整数

设置日志级别偏移量。

切片整数（编码，视频）

切片数，用于并行编码。

thread\_type标志（解码/编码，视频）

选择要使用的多线程方法。

使用“ frame ”会增加每个线程一帧的解码延迟，因此不能提供未来帧的客户端不应使用它。

可能的值：

'切片'

一次解码单个帧的多个部分。

使用切片的多线程仅在视频使用切片编码时才有效。

'框架'

一次解码多个帧。

默认值为“ slice+frame ”。

audio\_service\_type整数（编码，音频）

设置音频服务类型。

可能的值：

'妈'

主要音频服务

' ef '

效果

'六'

视障人士

'嗨'

听力障碍

'滴'

对话

'合作'

评论

'他们'

紧急情况

' vo '

画外音

'卡'

卡拉OK

request\_sample\_fmt sample\_fmt （解码，音频）

设置样本格式音频解码器应该更喜欢。默认值为无。

pkt\_timebase有理数

sub\_charenc编码（解码，字幕）

设置输入字幕字符编码。

field\_order field\_order （视频）

设置/覆盖视频的场序。可能的值：

'进步'

渐进式视频

' tt '

隔行扫描视频，顶场首先编码和显示

' bb '

隔行扫描视频，底场首先编码和显示

'结核病'

隔行扫描视频，顶部先编码，底部先显示

' BT '

隔行扫描视频，底部先编码，顶部先显示

skip\_alpha bool (解码, 视频)

设置为 1 以禁用处理 alpha（透明度）。这类似于标志选项中的“灰色”标志，它跳过色度信息而不是 alpha。默认为 0。

codec\_whitelist列表（输入）

"," 分隔的允许解码器列表。默认情况下所有都被允许。

dump\_separator字符串（输入）

Separator 用于分隔打印在命令行上的关于 Stream 参数的字段。例如，要用换行符和缩进分隔字段：

ffprobe -dump\_separator“

“-i ~/videos/matrixbench\_mpeg2.mpg

max\_pixels整数（解码/编码，视频）

每个图像的最大像素数。该值可用于避免由于大图像而导致的内存不足故障。

apply\_cropping bool (解码,视频)

如果裁剪参数是左侧和顶部参数所需对齐方式的倍数，则启用裁剪。如果不符合对齐要求，将部分应用裁剪以保持对齐。默认值为 1（启用）。注意：所需的对齐取决于是否设置了AV\_CODEC\_FLAG\_UNALIGNED和 CPU。 AV\_CODEC\_FLAG\_UNALIGNED不能从命令行更改。硬件解码器也不会应用左/上裁剪。

## 10个解码器

解码器是 FFmpeg 中的配置元素，它允许解码多媒体流。

当您配置 FFmpeg 构建时，默认情况下会启用所有受支持的本机解码器。需要外部库的解码器必须通过相应的--enable-lib选项手动启用。您可以使用配置选项--list-decoders列出所有可用的解码器。

您可以使用配置选项--disable-decoders禁用所有解码器，并使用选项--enable-decoder= DECODER / --disable-decoder= DECODER选择性地启用/禁用单个解码器。

选项-decoders将显示启用的解码器列表。

## 11个视频解码器

下面是一些当前可用的视频解码器的描述。

* [AV1](#av1)
* [原始视频](#rawvideo)
* [libdav1d](#libdav1d)
* [libdavs2](#libdavs2)
* [libuavs3d](#libuavs3d)
* [QSV 解码器](#QSV-Decoders)
* [v210](#v210)

### 11.1 AV1

AOMedia 视频 1 (AV1) 解码器。

* [选项](#Options-1)

#### 11.1.1 选项

操作点

选择可缩放 AV1 比特流 (0 - 31) 的操作点。默认为 0。

### 11.2 原始视频

原始视频解码器。

该解码器解码原始视频流。

* [选项](#Options-2)

#### 11.2.1 选项

顶部top\_field\_first

指定输入视频的假定字段类型。

-1

假定视频是渐进式的（默认）

0

假定底场优先

1个

假定顶场优先

### 11.3 libdav1d

dav1d AV1 解码器。

libdav1d 允许 libavcodec 解码 AOMedia Video 1 (AV1) 编解码器。在配置期间需要存在 libdav1d 标头和库。您需要使用--enable-libdav1d显式配置构建。

* [选项](#Options-3)

#### 11.3.1 选项

libdav1d 包装器支持以下选项。

框架线程

设置解码期间要使用的帧线程数。默认值为 0（自动检测）。对于 libdav1d >= 1.0，此选项已弃用，将来会被删除。请改用选项max\_frame\_delay和全局选项线程。

瓷砖线

设置解码期间要使用的图块线程数。默认值为 0（自动检测）。对于 libdav1d >= 1.0，此选项已弃用，将来会被删除。请改用全局选项线程。

最大帧延迟

设置解码器可以在内部缓冲的最大帧数。默认值为 0（自动检测）。

胶片颗粒

如果比特流中存在胶片颗粒，则将胶片颗粒应用于解码视频。默认为库的内部默认值。此选项已弃用，将来会被删除。请参阅全局选项export\_side\_data以导出 Film Grain 参数而不是应用它。

反对

选择可缩放 AV1 比特流 (0 - 31) 的操作点。默认为库的内部默认值。

全层

输出可缩放 AV1 比特流的所有空间层。默认值为假。

### 11.4 libdavs2

AVS2-P2/IEEE1857.4 视频解码器包装器。

此解码器允许 libavcodec 使用 davs2 库解码 AVS2 流。

### 11.5 libuavs3d

AVS3-P2/IEEE1857.10视频解码器。

libuavs3d 允许 libavcodec 解码 AVS3 流。在配置期间需要存在 libuavs3d 标头和库。您需要使用--enable-libuavs3d显式配置构建。

* [选项](#Options-4)

#### 11.5.1 选项

libuavs3d 包装器支持以下选项。

框架线程

设置解码期间要使用的帧线程数。默认值为 0（自动检测）。

### 11.6 QSV译码器

Intel QuickSync 视频解码器系列（VC1、MPEG-2、H.264、HEVC、JPEG/MJPEG、VP8、VP9、AV1）。

* [常用选项](#Common-Options)
* [HEVC 选项](#HEVC-Options)

#### 11.6.1 常用选项

所有 qsv 解码器都支持以下选项。

异步深度

内部并行化深度，值越高延迟越高。

gpu\_copy

视频和系统内存之间的 GPU 加速复制

'默认'

'上'

'关闭'

#### 11.6.2 HEVC 选项

hevc\_qsv 的额外选项。

加载插件

在内部会话中加载的用户插件

'无'

' hevc\_sw '

' hevc\_hw '

加载插件

A :- 在内部会话中加载的十六进制插件 UID 的单独列表

### 11.7 v210

未压缩的 4:2:2 10 位解码器。

* [选项](#Options-5)

#### 11.7.1 选项

自定义步幅

以字节为单位设置 v210 数据的行大小。默认值为 0（自动检测）。如 BOXX 文件所示，您可以为无跨步 v210 使用特殊的 -1 值。

## 12个音频解码器

下面是一些当前可用的音频解码器的描述。

* [ac3](#ac3)
* [高音](#flac-1)
* [合成器](#ffwavesynth)
* [自由意志](#libcelt)
* [libgsm](#libgsm)
* [libilbc](#libilbc)
* [libopencore-amrnb](#libopencore_002damrnb)
* [libopencore-amrwb](#libopencore_002damrwb)
* [力宝](#libopus)

### 12.1 ac3

AC-3 音频解码器。

该解码器实现了 ATSC A/52:2010 和 ETSI TS 102 366 的一部分，以及未记录的 RealAudio 3（又名 dnet）。

* [AC-3 解码器选项](#AC_002d3-Decoder-Options)

#### 12.1.1 AC-3 解码器选项

-drc\_scale值

动态范围比例因子。应用于来自 AC-3 流的动态范围值的因子。该因子以指数方式应用。默认值为 1。有 3 个值得注意的比例因子范围：

drc\_scale == 0

刚果民主共和国禁用。产生全频音频。

0 < drc\_scale <= 1

启用刚果民主共和国。应用一小部分流 DRC 值。音频再现介于全范围和全压缩之间。

drc\_scale > 1

启用刚果民主共和国。不对称地应用 drc\_scale。响亮的声音被完全压缩。柔和的声音得到增强。

### 12.2 后高音

FLAC 音频解码器。

该解码器旨在实现 Xiph 的完整 FLAC 规范。

* [FLAC 解码器选项](#FLAC-Decoder-options)

#### 12.2.1 FLAC 解码器选项

-use\_buggy\_lpc

lavc FLAC 编码器用于生成具有高 lpc 值（如默认值）的错误流。该选项可以通过使用 lavc 用于解码的旧错误 lpc 逻辑来正确解码此类流。

### 12.3 ffwavesynth

内部波合成器。

该解码器根据预定义的序列生成波形图。它的使用纯粹是内部的，它接受的数据格式没有公开记录。

### 12.4 libcelt

libcelt 解码器包装器。

libcelt 允许 libavcodec 解码 Xiph CELT 超低延迟音频编解码器。在配置期间需要存在 libcelt 标头和库。您需要使用--enable-libcelt显式配置构建。

### 12.5 libgsm

libgsm 解码器包装器。

libgsm 允许 libavcodec 解码 GSM 全速率音频编解码器。需要在配置期间存在 libgsm 标头和库。您需要使用--enable-libgsm显式配置构建。

此解码器支持普通 GSM 和 Microsoft 变体。

### 12.6 libilbc

libilbc 解码器包装器。

libilbc 允许 libavcodec 解码 Internet Low Bitrate Codec (iLBC) 音频编解码器。需要在配置期间存在 libilbc 标头和库。您需要使用--enable-libilbc显式配置构建。

* [选项](#Options-6)

#### 12.6.1 选项

libilbc 包装器支持以下选项。

提高

设置为 1 时启用解码音频的增强功能。默认值为 0（禁用）。

### 12.7 libopencore-amrnb

libopencore-amrnb 解码器包装器。

libopencore-amrnb 允许 libavcodec 解码自适应多速率窄带音频编解码器。使用它需要在配置期间存在 libopencore-amrnb 标头和库。您需要使用--enable-libopencore-amrnb显式配置构建。

存在用于 AMR-NB 的 FFmpeg 本机解码器，因此用户无需此库即可解码 AMR-NB。

### 12.8 libopencore-amrwb

libopencore-amrwb 解码器包装器。

libopencore-amrwb 允许 libavcodec 解码自适应多速率宽带音频编解码器。使用它需要在配置期间存在 libopencore-amrwb 标头和库。您需要使用--enable-libopencore-amrwb显式配置构建。

存在用于 AMR-WB 的 FFmpeg 本机解码器，因此用户无需此库即可解码 AMR-WB。

### 12.9 libopus

libopus 解码器包装器。

libopus 允许 libavcodec 解码 Opus Interactive Audio Codec。在配置期间需要存在 libopus 标头和库。您需要使用--enable-libopus显式配置构建。

存在用于 Opus 的 FFmpeg 本机解码器，因此用户可以在没有此库的情况下解码 Opus。

## 13个字幕解码器

* [libaribb24](#libaribb24)
* [dvb订阅](#dvbsub)
* [DVD子](#dvdsub)
* [libzvbit-图文电视](#libzvbi_002dteletext)

### 13.1 libaribb24

ARIB STD-B24 字幕解码器。

实施 ARIB STD-B24 标准的配置文件 A 和 C。

* [libaribb24 解码器选项](#libaribb24-Decoder-Options)

#### 13.1.1 libaribb24 解码器选项

-aribb24-base-path路径

设置 libaribb24 库的基本路径。这用于读取配置文件（用于自定义 unicode 转换），以及将非文本符号转储为该位置下的图像。

默认情况下取消设置。

-aribb24-skip-ruby-text布尔值

告诉解码器包装器跳过包含半高 ruby 文本的文本块。

默认启用。

### 13.2 dvbsub

* [选项](#Options-7)

#### 13.2.1 选项

计算杂乱

-2

如果流中没有匹配的 CLUT，则计算一次 clut。

-1

如果流中没有匹配的 CLUT，则计算 clut。

0

从不计算 CLUT

1个

始终计算 CLUT 并覆盖流中提供的那个。

dvb\_子流

选择 dvb 子流，如果默认为 -1，则选择所有子流。

### 13.3 DVD 子

该编解码器解码 DVD 中使用的位图字幕；在 VobSub 文件对和某些 Matroska 文件中也可以找到相同的字幕。

* [选项](#Options-8)

#### 13.3.1 选项

调色板

指定位图使用的全局调色板。当存储在 VobSub 中时，调色板通常在索引文件中指定；在 Matroska 中，调色板以与 VobSub 中相同的格式存储在编解码器额外数据中。在 DVD 中，调色板存储在 IFO 文件中，因此在从转储的 VOB 文件中读取时不可用。

此选项的格式是一个包含16个24位十六进制的数字（没有0x前缀）的字符串，例如逗号分隔，例如0d00ee，ee450d，101010，eaeaea，eaeaea，0ce60b，ec14ed，ec14ed，ebff0b，ebff0b，0d617a，0d617a，7b7b7b，7b7b7b，d1d1d1d1d1d1 d1d1，7b21,7b2a，0dd1,0d9999999999999999999999999999999999999999. 0f007b、cf0dec、cfa80c、7c127b 。

ifo\_调色板

指定从中获取全局调色板的 IFO 文件。 （实验性的）

forced\_subs\_only

仅解码标记为强制的字幕条目。有些标题在同一轨道中有强制和非强制字幕。将此标志设置为1将只保留强制字幕。默认值为0 。

### 13.4 libzvbi-图文电视

Libzvbi 允许 libavcodec 解码 DVB 图文电视页面和 DVB 图文电视字幕。在配置期间需要存在 libzvbi 标头和库。您需要使用--enable-libzvbi显式配置构建。

* [选项](#Options-9)

#### 13.4.1 选项

txt\_page

要解码的图文电视页码列表。与指定列表不匹配的页面将被删除。您可以使用特殊的\*字符串来匹配所有页面，或使用subtitle来匹配所有字幕页面。默认值为 \*。

txt\_default\_region

设置用于解码的默认字符集，值介于 0 和 87 之间（参见 ETS 300 706，第 15 节，表 32）。默认值为 -1，它不会覆盖 libzvbi 默认值。某些无法发出正确字符集信号的旧版 1.0 传输需要此选项。

txt\_chop\_top

丢弃顶部图文电视行。默认值为 1。

txt\_格式

指定解码字幕的格式。

位图

默认格式，图文页面应该使用这个，因为某些图形和颜色不能用简单的文本甚至 ASS 来表达。

文本

没有格式的简单基于文本的输出。

屁股

格式化的 ASS 输出、字幕页面和图文电视页面以不同的样式返回，字幕页面被剥离为文本，但努力保持文本对齐和格式。

txt\_left

生成位图的 X 偏移量，默认为 0。

txt\_top

生成位图的 Y 偏移量，默认为 0。

txt\_chop\_spaces

截断前导和尾随空格并从生成的文本中删除空行。此选项对于基于图文电视的字幕很有用，其中行首或行尾可能存在空格，或者由于双倍大小的图文电视字符，字幕行之间可能存在空行。默认值为 1。

txt\_duration

以毫秒为单位设置解码图文电视页面或字幕的显示持续时间。默认值为 -1，表示无穷大或直到下一个字幕事件到来。

txt\_透明

强制生成的图文电视位图的透明背景。默认值为 0，表示背景不透明。

txt\_不透明度

设置图文电视背景的不透明度 (0-255)。如果未设置txt\_transparent ，它只会影响开始框和结束框之间的字符，通常是字幕。如果设置了txt\_transparent ，则默认值为 0 ，否则为 255。

## 14个编码器

编码器是 FFmpeg 中的配置元素，它允许对多媒体流进行编码。

当您配置 FFmpeg 构建时，默认情况下会启用所有受支持的本机编码器。需要外部库的编码器必须通过相应的--enable-lib选项手动启用。您可以使用配置选项--list-encoders列出所有可用的编码器。

您可以使用配置选项--disable-encoders禁用所有编码器，并使用选项--enable-encoder= ENCODER / --disable-encoder= ENCODER选择性地启用/禁用单个编码器。

选项-encoders将显示已启用编码器的列表。

## 15 个音频编码器

下面是一些当前可用的音频编码器的描述。

* [aac](#aac)
* [ac3 和 ac3\_fixed](#ac3-and-ac3_005ffixed)
* [高音](#flac-2)
* [作品](#opus)
* [libfdk\_aac](#libfdk_005faac)
* [libmp3lame](#libmp3lame-1)
* [libopencore-amrnb](#libopencore_002damrnb-1)
* [力宝](#libopus-1)
* [libshine](#libshine-1)
* [libtwolame](#libtwolame)
* [libvo-amrwbenc](#libvo_002damrwbenc)
* [libvorbis](#libvorbis)
* [mjpeg](#mjpeg)
* [wavpack](#wavpack)

### 15.1 音频

高级音频编码 (AAC) 编码器。

此编码器是默认的 AAC 编码器，在 FFmpeg 中原生实现。

* [选项](#Options-10)

#### 15.1.1 选项

b

以比特/秒为单位设置比特率。设置此项会自动激活恒定比特率 (CBR) 模式。如果未指定此选项，则设置为 128kbps。

q

设置可变比特率 (VBR) 模式的质量。此选项仅在使用ffmpeg命令行工具时有效。对于库接口用户，请使用global\_quality 。

隔断

设置截止频率。如果未指定，将允许编码器动态调整截止以提高低比特率的清晰度。

aac\_coder

设置AAC编码器的编码方式。可能的值：

'双循环'

两个循环搜索（TLS）方法。这是默认方法。

该方法首先根据频带阈值设置量化器，然后尝试通过从所有量化器中添加或减去特定值并稍微调整一些单独的量化器来找到最佳组合。将根据是否启用aac\_is 、 aac\_ms和aac\_pns进行自我调整。

' anm '

基于平均噪声掩模比 (ANMR) 网格的解决方案。

这是一个实验性的编码器，目前产生的质量较低，更不稳定并且比默认的双循环编码器慢但有潜力。目前不支持aac\_is或aac\_pns选项。目前不推荐。

'快'

恒定量化器方法。

使用更便宜的双环算法版本，不会尝试做那么多巧妙的调整。低比特率（小于 64kbps）时更糟，但在更高的比特率下更好更快。

aac\_ms

设置中/侧编码模式。 “auto”的默认值将自动使用 M/S 和将受益于此类编码的频段。可以使用值“enable”强制所有波段，这主要用于调试或使用“disable”禁用。

aac\_is

设置强度立体声编码工具的使用。默认情况下，它是启用的，如果有用的话，它会自动为相似的立体声频段对切换 IS。可以通过将值设置为“禁用”来禁用调试。

aac\_pns

在解码过程中使用感知噪声替代将低熵高频段替换为不易察觉的白噪声。默认情况下，它是启用的，但可以通过使用“禁用”来禁用调试目的。

aac\_tns

允许使用跨越高频带的多抽头 FIR 滤波器，以隐藏编码过程中的量化噪声，并由解码器还原。除了减少高频中令人不快的伪影外，这还降低了高频中的熵，并允许中低频段使用更多比特。默认情况下它是启用的，但可以通过将选项设置为“禁用”来禁用调试。

aac\_ltp

允许使用长期预测扩展，它通过在整个帧中扩展频带中的恒定谐波峰值来提高非常低带宽情况下的编码效率，例如语音或独奏钢琴音乐的编码。此选项由 profile:a aac\_low 隐含，并且与 aac\_pred 不兼容。与-ar结合使用可降低采样率。

aac\_pred

允许使用更传统的预测方式，其中传输的频谱系数被当前系数减去先前“预测”系数的差值代替。在理论上，有时在实践中，这可以提高中低比特率音频的质量。此选项暗示 aac\_main 配置文件并且与 aac\_ltp 不兼容。

轮廓

设置编码配置文件，可能的值：

' aac\_low '

默认的 AAC“低复杂性”配置文件。是最兼容的并产生体面的质量。

' mpeg2\_aac\_low '

相当于-profile:a aac\_low -aac\_pns 0 。 PNS 是随 MPEG4 规范一起引入的。

' aac\_ltp '

长期预测配置文件，由aac\_ltp选项启用并将启用。在 MPEG4 中引入。

' aac\_main '

aac\_pred选项启用并将启用。在 MPEG2 中引入。

如果未指定此选项，则将其设置为“ aac\_low ”。

### 15.2 ac3 和 ac3\_fixed

AC-3 音频编码器。

这些编码器实现了 ATSC A/52:2010 和 ETSI TS 102 366 的一部分，以及未记录的 RealAudio 3（又名 dnet）。

ac3编码器使用浮点数学运算，而ac3\_fixed编码器仅使用定点整数数学运算。这并不意味着一个总是更快，只是一个或另一个可能更适合特定系统。 ac3\_fixed编码器不是任何输出格式的默认编解码器，因此必须使用选项-acodec ac3\_fixed明确指定才能使用它。

* [AC-3 元数据](#AC_002d3-Metadata)
* [扩展比特流信息](#Extended-Bitstream-Information)
* [其他 AC-3 编码选项](#Other-AC_002d3-Encoding-Options)
* [仅浮点 AC-3 编码选项](#Floating_002dPoint_002dOnly-AC_002d3-En)

#### 15.2.1 AC-3 元数据

AC-3 元数据选项用于设置描述音频的参数，但在大多数情况下不会影响音频编码本身。一些选项确实直接影响或影响结果比特流的解码和播放，而其他选项仅供参考。一些选项会向输出流添加位，否则这些位可能会用于音频数据，因此会影响输出的质量。这些将在下面的选项列表中用注释相应地指出。

这些参数在几个公开可用的文档中有详细描述。

* [A/52:2010 - 数字音频压缩 (AC-3) (E-AC-3) 标准](http://www.atsc.org/cms/standards/a_52-2010.pdf)
* [A/54 - ATSC 数字电视标准使用指南](http://www.atsc.org/cms/standards/a_54a_with_corr_1.pdf)
* [杜比元数据指南](http://www.dolby.com/uploadedFiles/zz-_Shared_Assets/English_PDFs/Professional/18_Metadata.Guide.pdf)
* [杜比数字专业编码指南](http://www.dolby.com/uploadedFiles/zz-_Shared_Assets/English_PDFs/Professional/46_DDEncodingGuidelines.pdf)
* [元数据控制选项](#Metadata-Control-Options)
* [缩混级别](#Downmix-Levels)
* [音频制作信息](#Audio-Production-Information)
* [其他元数据选项](#Other-Metadata-Options)

#### 15.2.1.1 元数据控制选项

-per\_frame\_metadata布尔值

允许每帧元数据。指定编码器是否应检查每个帧的元数据更改。

0

初始化时设置的元数据值将用于流中的每一帧。 （默认）

1个

可以在对每个帧进行编码之前更改元数据值。

#### 15.2.1.2 缩混级别

-center\_mixlev级别

中心混合电平。缩混为立体声时解码器应应用于中央声道的增益量。如果存在中心通道，则此字段将仅写入比特流。该值指定为比例因子。有 3 个有效值：

0.707

应用 -3dB 增益

0.595

应用 -4.5dB 增益（默认）

0.500

应用 -6dB 增益

-surround\_mixlev级别

环绕声混合电平。缩混为立体声时解码器应应用于环绕声道的增益量。如果存在一个或多个环绕声道，则此字段将仅写入比特流。该值指定为比例因子。有 3 个有效值：

0.707

应用 -3dB 增益

0.500

应用 -6dB 增益（默认）

0.000

静音环绕声道

#### 15.2.1.3 音频制作信息

音频制作信息是描述混合环境的可选信息。没有或两个字段都写入比特流。

-mixing\_level数

混合水平。指定制作环境中混音被掌握时的峰值声压级 (SPL)。有效值为 80 到 111，或 -1 表示未知或未指示。默认值为 -1，但如果将音频制作信息写入比特流，则无法使用该值。因此，如果room\_type选项不是默认值，则mixing\_level选项不能为 -1。

-room\_type类型

房型。描述在录音室或配音阶段的最后混音过程中使用的均衡。大房间配音舞台，行业标准X曲线均衡；一个小房间有平坦的均衡。如果mixing\_level选项和room\_type选项都具有默认值，则不会将此字段写入比特流。

0

未注明

未指示（默认）

1个

大的

大房间

2个

小的

小房间

#### 15.2.1.4 其他元数据选项

-版权布尔值

版权指示符。指定此音频是否存在版权。

0

离开

不存在版权（默认）

1个

在

版权存在

-dialnorm值

对话规范化。指示节目的平均对话电平低于数字 100% 满量程 (0 dBFS) 的程度。此参数确定音频再现期间的电平偏移，将对话的平均音量设置为预设电平。目标是匹配节目源之间的音量级别。 -31dB 的值将导致在音频再现期间相对于源音量没有音量级别变化。有效值为 -31 到 -1 范围内的整数，默认值为 -31。

-dsur\_mode模式

杜比环绕模式。指定立体声信号是否使用 Dolby Surround (Pro Logic)。如果音频流是立体声，则此字段将仅写入比特流。使用此选项并不**意味着**编码器将实际应用杜比环绕处理。

0

未注明

未指示（默认）

1个

离开

不是杜比环绕声编码

2个

在

杜比环绕声编码

-原始布尔值

原始比特流指示器。指定此音频是否来自原始来源而不是副本。

0

离开

非原始来源

1个

在

原始来源（默认）

#### 15.2.2 扩展比特流信息

扩展比特流选项是 A/52:2010 标准附件 D 中指定的替代比特流语法的一部分。它分为两部分。如果指定组中的任何一个参数，则该组中的所有值都将写入比特流。默认值用于写入但未指定的那些。如果写入混合级别，解码器将使用这些值而不是在center\_mixlev和surround\_mixlev选项中指定的值，前提是它支持备用比特流语法。

* [扩展比特流信息 - 第 1 部分](#Extended-Bitstream-Information-_002d-Pa)
* [扩展比特流信息 - 第 2 部分](#Extended-Bitstream-Information-_002d-Pa)

#### 15.2.2.1 扩展比特流信息——第 1 部分

-dmix\_mode模式

首选立体声缩混模式。允许用户选择 Lt/Rt（杜比环绕声）或 Lo/Ro（普通立体声）作为首选立体声缩混模式。

0

未注明

未指示（默认）

1个

轻轨

Lt/Rt 缩混首选

2个

鹦鹉

Lo/Ro 缩混首选

-ltrt\_cmixlev级别

Lt/Rt 中心混合电平。在 Lt/Rt 模式下缩混为立体声时，解码器应应用于中央声道的增益量。

1.414

应用 +3dB 增益

1.189

应用 +1.5dB 增益

1.000

应用 0dB 增益

0.841

应用 -1.5dB 增益

0.707

应用 -3.0dB 增益

0.595

应用 -4.5dB 增益（默认）

0.500

应用 -6.0dB 增益

0.000

沉默中心频道

-ltrt\_surmixlev级别

Lt/Rt 环绕混合电平。在 Lt/Rt 模式下缩混为立体声时，解码器应应用于环绕声道的增益量。

0.841

应用 -1.5dB 增益

0.707

应用 -3.0dB 增益

0.595

应用 -4.5dB 增益

0.500

应用 -6.0dB 增益（默认）

0.000

静音环绕声道

-loro\_cmixlev级别

Lo/Ro 中心混合水平。在 Lo/Ro 模式下缩混为立体声时，解码器应应用于中央声道的增益量。

1.414

应用 +3dB 增益

1.189

应用 +1.5dB 增益

1.000

应用 0dB 增益

0.841

应用 -1.5dB 增益

0.707

应用 -3.0dB 增益

0.595

应用 -4.5dB 增益（默认）

0.500

应用 -6.0dB 增益

0.000

沉默中心频道

-loro\_surmixlev级别

Lo/Ro 环绕声混合电平。在 Lo/Ro 模式下缩混为立体声时，解码器应应用于环绕声道的增益量。

0.841

应用 -1.5dB 增益

0.707

应用 -3.0dB 增益

0.595

应用 -4.5dB 增益

0.500

应用 -6.0dB 增益（默认）

0.000

静音环绕声道

#### 15.2.2.2 扩展比特流信息——第 2 部分

-dsurex\_mode模式

杜比环绕 EX 模式。指示流是否使用 Dolby Surround EX（7.1 矩阵转换为 5.1）。使用此选项并不**意味着**编码器将实际应用 Dolby Surround EX 处理。

0

未注明

未指示（默认）

1个

在

杜比环绕声 EX 关闭

2个

离开

杜比环绕声 EX 开启

-dheadphone\_mode模式

杜比耳机模式。指示流是否使用 Dolby Headphone 编码（多声道矩阵化为 2.0 以便与耳机一起使用）。使用此选项并不**意味着**编码器将实际应用杜比耳机处理。

0

未注明

未指示（默认）

1个

在

杜比耳机关闭

2个

离开

杜比耳机开启

-ad\_conv\_type类型

A/D 转换器类型。指示音频是否经过HDCD A/D 转换。

0

标准

标准 A/D 转换器（默认）

1个

硬盘驱动器

HDCD 模数转换器

#### 15.2.3 其他 AC-3 编码选项

-stereo\_rematrixing布尔值

立体声再矩阵化。启用/禁用对立体声输入重新矩阵化的使用。这是一个可选的 AC-3 功能，通过有选择地将左/右声道编码为中/侧来提高质量。默认情况下启用此选项，强烈建议将其保留为启用状态，除非用于测试目的。

截止频率

设置低通截止频率。如果未指定，编码器会选择由各种其他编码参数确定的默认值。

#### 15.2.4 仅浮点数 AC-3 编码选项

这些选项仅对浮点编码器有效，对定点编码器不存在，因为相应的功能未在定点中实现。

-channel\_coupling布尔值

启用/禁用通道耦合，这是一项可选的 AC-3 功能，可通过将来自多个通道的高频信息组合到单个通道中来提高质量。每通道高频信息在频域和时域中都以较低的精度发送。这允许更多的比特用于较低的频率，同时保留足够的信息来重建高频。此选项默认为浮点编码器启用，通常应保持启用状态，除非用于测试目的或提高编码速度。

-1

汽车

由编码器选择（默认）

0

离开

禁用通道耦合

1个

在

启用通道耦合

-cpl\_start\_band编号

耦合启动带。设置通道耦合起始频带，从 1 到 15。如果使用高于带宽的值，它将减少到耦合结束频带小于 1。如果使用auto ，起始波段将由编码器根据比特率、采样率和通道布局确定。如果禁用通道耦合，则此选项无效。

-1

汽车

由编码器选择（默认）

### 15.3 后高音

FLAC（免费无损音频编解码器）编码器

* [选项](#Options-11)

#### 15.3.1 选项

FFmpeg 的 flac 编码器支持以下选项。

压缩级别

设置压缩级别，如果没有明确设置，它会为许多其他选项选择默认值。有效值为 0 到 12，默认值为 5。

帧大小

设置每个通道样本中帧的大小。

lpc\_coeff\_precision

设置 LPC 系数精度，有效值为 1 到 15，默认值为 15。

lpc\_type

设置第一阶段LPC算法

'无'

未使用 LPC

'固定'

固定的 LPC 系数

'莱文森'

'乔斯基'

lpc\_passes

LPC 分析期间用于 Cholesky 分解的遍数

最小分区顺序

最小分区顺序

最大分区顺序

最大分区顺序

prediction\_order\_method

'估计'

' 2级'

' 4级'

' 8级'

'搜索'

暴力搜索

'日志'

ch\_mode

通道模式

'汽车'

为每一帧自动选择模式

'独立'

通道独立编码

'左边'

'右侧'

'中侧'

exact\_rice\_parameters

选择大米参数是精确计算还是近似计算。如果设置为 1 则它们会被准确选择，这会稍微减慢代码速度并稍微提高压缩率。

multi\_dim\_quant

多维量化。如果设置为 1，则在第一阶段之后应用第二阶段 LPC 算法来微调系数。这非常慢并且略微提高了压缩率。

### 15.4 作品

作品编码器。

这是 Opus 格式的原生 FFmpeg 编码器。目前它正在开发中，只实现了编解码器的 CELT 部分。它的质量通常较差，充其量与 libopus 编码器相当。

* [选项](#Options-12)

#### 15.4.1 选项

b

以比特/秒为单位设置比特率。如果未指定，它会使用通道数和布局来做出很好的猜测。

opus\_delay

设置以毫秒为单位的最大延迟。低于 20 毫秒的延迟会很快降低质量。

### 15.5 libfdk\_aac

libfdk-aac AAC（高级音频编码）编码器包装器。

libfdk-aac 库基于 Android 项目中的 Fraunhofer FDK AAC 代码。

在配置期间需要存在 libfdk-aac 标头和库。您需要使用--enable-libfdk-aac显式配置构建。该库也不兼容 GPL，因此如果您允许使用 GPL，则应使用--enable-gpl --enable-nonfree --enable-libfdk-aac进行配置。

此编码器支持 AAC-HE 配置文件。

vbr或flags +qscale选项启用的 VBR 编码是实验性的，仅适用于某些参数组合。

对编码 7.1 音频的支持仅适用于 libfdk-aac 0.1.3 或更高版本。

<http://sourceforge.net/p/opencore-amr/fdk-aac/>上的 fdk-aac 项目。

* [选项](#Options-13)
* [例子](#Examples-4)

#### 15.5.1 选项

以下选项映射到共享的 FFmpeg 编解码器选项。

b

以比特/秒为单位设置比特率。如果未明确指定比特率，则会根据所选配置文件自动将其设置为合适的值。

如果启用 VBR 模式，则忽略该选项。

ar

设置音频采样率（以 Hz 为单位）。

频道

设置音频通道数。

标志+qscale

启用固定质量、VBR（可变比特率）模式。请注意，当vbr值为正时，VBR 被隐式启用。

隔断

设置截止频率。如果未指定（或明确设置为 0），它将使用库自动计算的值。默认值为 0。

轮廓

设置音频配置文件。

识别以下配置文件：

' aac\_low '

低复杂度 AAC (LC)

' aac\_he '

高效 AAC (HE-AAC)

' aac\_he\_v2 '

高效 AAC 第 2 版 (HE-AACv2)

' aac\_ld '

低延迟 AAC (LD)

' aac\_field '

增强型低延迟 AAC (ELD)

如果未指定，则设置为“ aac\_low ”。

以下是 libfdk\_aac 编码器的私有选项。

加力

如果设置为 1，则启用加力燃烧室功能，如果设置为 0，则禁用。这会提高质量，但也会提高所需的处理能力。

默认值为 1。

字段\_sbr

如果设置为 1，则为 ELD 启用 SBR（光谱带复制），如果设置为 0，则禁用。

默认值为 0。

字段\_v2

如果设置为 1，则为 ELDv2 启用 ELDv2（ELD 立体声信号的 LD-MPS 扩展），如果设置为 0，则禁用。

请注意，当 fdk-aac 版本 (AACENCODER\_LIB\_VL0.AACENCODER\_LIB\_VL1.AACENCODER\_LIB\_VL2) > (4.0.0) 时，该选项可用。

默认值为 0。

信号

设置 SBR/PS 信令样式。

它可以采用以下值之一：

'默认'

隐式选择信号（默认情况下显式分层，如果禁用全局标头则隐式）

'隐式'

隐式向后兼容信号

' explicit\_sbr '

显式 SBR，隐式 PS 信令

'显式层次'

显式分层信令

默认值为“默认”。

拉美

如果设置为 1，则输出 LATM/LOAS 封装数据，如果设置为 0，则禁用。

默认值为 0。

header\_period

设置 StreamMuxConfig 和 PCE 重复周期（以帧为单位），用于在 LATM/LOAS 传输层内发送带内配置缓冲区。

必须是 16 位非负整数。

默认值为 0。

vbr

设置 VBR 模式，从 1 到 5。1 是最低质量（虽然仍然很好），5 是最高质量。值为 0 将禁用 VBR，并启用 CBR（恒定比特率）。

目前只有“ aac\_low ”配置文件支持 VBR 编码。

VBR 模式 1-5 大致对应于以下平均比特率：

' 1 '

32 kbps/通道

' 2 '

40 kbps/通道

' 3 '

48-56 kbps/通道

' 4 '

64 kbps/通道

' 5 '

大约 80-96 kbps/频道

默认值为 0。

#### 15.5.2 例子

* 使用ffmpeg将音频文件转换为 M4A (MP4) 容器中的 VBR AAC：
* ffmpeg -i input.wav -codec:a libfdk\_aac -vbr 3 输出.m4a
* 使用ffmpeg将音频文件转换为 CBR 64k kbps AAC，使用高效 AAC 配置文件：
* ffmpeg -i input.wav -c:a libfdk\_aac -profile:a aac\_he -b:a 64k output.m4a

### 15.6 libmp3lame

LAME（Lame Ain't an MP3 Encoder）MP3 编码器包装器。

在配置期间需要存在 libmp3lame 标头和库。您需要使用--enable-libmp3lame显式配置构建。

请参阅[libshine](#libshine)以获取定点 MP3 编码器，但质量较低。

* [选项](#Options-14)

#### 15.6.1 选项

libmp3lame 包装器支持以下选项。选项的 lame 等价物列在括号中。

b ( -b )

为 CBR 或 ABR 设置以比特/秒表示的比特率。 LAME比特率以千比特/秒表示。

q ( -V )

为 VBR 设置恒定质量设置。此选项仅在使用ffmpeg命令行工具时有效。对于库接口用户，请使用global\_quality 。

压缩级别 ( -q )

设置算法质量。有效参数是 0-9 范围内的整数，0 表示质量最高但最慢，9 表示速度最快但质量最差。

截止（ --lowpass ）

设置低通截止频率。如果未指定，编码器会动态调整截止值。

水库

设置为 1 时启用位库。默认值为 1。LAME 默认启用此功能，但可以使用--nores选项覆盖。

关节立体声（ -mj ）

启用编码器以使用（逐帧）L/R 立体声或中/侧立体声。默认值为 1。

abr ( --abr )

编码器使用 ABR。 --abr设置目标比特率，而这个选项只告诉 FFmpeg 使用 ABR 仍然依赖b来设置比特率。

### 15.7 libopencore-amrnb

OpenCORE 自适应多速率窄带编码器。

在配置期间需要存在 libopencore-amrnb 标头和库。您需要使用--enable-libopencore-amrnb --enable-version3显式配置构建。

这是一个单声道编码器。官方它只支持 8000Hz 采样率，但您可以通过将strict设置为“非官方”或更低来覆盖它。

* [选项](#Options-15)

#### 15.7.1 选项

b

以比特每秒为单位设置比特率。仅支持以下比特率，否则 libavcodec 将舍入到最接近的有效比特率。

4750

5150

5900

6700

7400

7950

10200

12200

数字传输系统

设置为 1 时允许不连续传输（产生舒适噪音）。默认值为 0（禁用）。

### 15.8 力宝

libopus Opus 交互式音频编解码器编码器包装器。

在配置期间需要存在 libopus 标头和库。您需要使用--enable-libopus显式配置构建。

* [选项映射](#Option-Mapping)

#### 15.8.1 选项映射

大多数 libopus 选项都是在opus-tools 的opusec实用程序之后建模的。以下是一个选项映射表，描述了 libopus 包装器支持的选项，以及它们在括号中的opusec等价物。

b（比特率）

以比特/秒为单位设置比特率。 FFmpeg 的b选项以 bits/s 表示，而opusec的比特率以 kilobits/s 表示。

vbr （ vbr 、 hard-cbr和cvbr ）

设置可变比特率模式。 FFmpeg vbr选项具有以下有效参数，括号中为opusenc等效选项：

'关闭（硬 cbr ） '

使用恒定比特率编码。

'在（动词） '

使用可变比特率编码（默认）。

'约束（ cvbr ） '

使用受约束的可变比特率编码。

压缩级别 ( comp )

设置编码算法复杂度。有效选项是 0-10 范围内的整数。 0 提供最快的编码但质量较低，而 10 提供最高质量但最慢的编码。默认值为 10。

帧持续时间（帧大小）

设置最大帧大小或帧的持续时间（以毫秒为单位）。参数必须完全如下：2.5、5、10、20、40、60。较小的帧大小可实现较低的延迟，但在给定的比特率下质量较差。大于 20 毫秒的大小仅在相当低的比特率下才有意义。默认值为 20 毫秒。

packet\_loss（预期损失）

设置预期的丢包率。默认值为 0。

fec (不适用)

启用带内前向纠错。 packet\_loss必须为非零才能利用 - FEC“边数据”的频率与预期的数据包丢失成正比。默认是禁用的。

申请（北美）

设置预期的应用程序类型。下面列出了有效选项：

'网络电话'

有利于提高语音清晰度。

'音频'

支持忠实于输入（默认）。

'低延迟'

仅限于最低延迟模式。

截止值 (NA)

以 Hz 为单位设置截止带宽。参数必须恰好为以下之一：4000、6000、8000、12000或20000，分别对应窄带、中带、宽带、超宽带和全带。默认值为 0（禁用截止）。

映射家族 (映射家族)

设置编码器要使用的通道映射系列。默认值 -1 对单声道和立体声输入使用映射系列 0，否则使用映射系列 1。默认还禁用 libopus 中的环绕声掩蔽和 LFE 带宽优化，并要求输入包含 8 个或更少的通道。

其他值包括 0 表示单声道和立体声，1 表示带掩蔽和 LFE 带宽优化的环绕声，255 表示具有未指定通道布局的独立流。

apply\_phase\_inv (NA)（需要 libopus >= 1.2）

如果设置为 0，则禁用强度立体声的相位反转，提高单声道缩混的质量，但略微降低正常立体声质量。默认值为 1（启用相位反转）。

### 15.9 libshine

Shine 定点 MP3 编码器包装器。

Shine 是一个定点 MP3 编码器。它在没有 FPU 的平台上有更好的性能，例如 armel CPU，以及一些手机和平板电脑。然而，由于它更注重性能而非质量，因此在质量方面无法与 LAME 和其他生产级编码器相提并论。此外，根据该项目的主页，该编码器可能并非没有错误，因为代码是很久以前编写的，并且该项目至少已停产 5 年。

此编码器仅支持立体声和单声道输入。这也是仅 CBR。

原始项目（最后更新于 2007 年初）位于<http://sourceforge.net/projects/libshine-fxp/>。我们仅支持 Savonet/Liquidsoap 项目在<https://github.com/savonet/shine>上更新的分支。

在配置期间需要存在 libshine 标头和库。您需要使用--enable-libshine显式配置构建。

另请参阅[libmp3lame](#libmp3lame) 。

* [选项](#Options-16)

#### 15.9.1 选项

libshine 包装器支持以下选项。选项的shineenc等价物列在括号中。

b ( -b )

为 CBR 设置以比特/秒表示的比特率。闪耀 -b选项以千比特/秒表示。

### 15.10 libtwolame

TwoLAME MP2 编码器包装器。

在配置期间需要存在 libtwolame 标头和库。您需要使用--enable-libtwolame显式配置构建。

* [选项](#Options-17)

#### 15.10.1 选项

libtwolame 包装器支持以下选项。 twolame等效选项跟在FFmpeg 选项之后并在括号中。

b ( -b )

为 CBR 设置以比特/秒表示的比特率。双脚 b选项以千比特/秒表示。默认值为 128k。

q ( -V )

为实验性 VBR 支持设置质量。最大值范围是 -50 到 50，有用范围是 -10 到 10。值越高，质量越好。此选项仅在使用ffmpeg命令行工具时有效。对于库接口用户，请使用global\_quality 。

模式（ --模式）

设置生成的音频的模式。可能的值：

'汽车'

根据输入自动选择模式。这是默认设置。

'立体声'

立体声

'联合立体声'

联合立体声

'双通道'

双通道

'单声道'

单核细胞增多症

心理模型（ --psyc-mode ）

设置心理声学模型以用于编码。参数必须是介于 -1 和 4 之间的整数，包括边界值。值越高，质量越好。默认值为 3。

energy\_levels ( --energy )

设置为 1 时启用能量级别扩展。默认值为 0（禁用）。

错误保护（ --保护）

设置为 1 时启用 CRC 错误保护。默认值为 0（禁用）。

版权（ --版权）

设置为 1 时设置 MPEG 音频版权标志。默认值为 0（禁用）。

原创（ --原创）

设置为 1 时设置 MPEG 音频原始标志。默认值为 0（禁用）。

### 15.11 libvo-amrwbenc

VisualOn 自适应多速率宽带编码器。

在配置期间需要存在 libvo-amrwbenc 标头和库。您需要使用--enable-libvo-amrwbenc --enable-version3显式配置构建。

这是一个单声道编码器。官方只支持 16000Hz 采样率，但您可以通过将strict设置为“非官方”或更低来覆盖它。

* [选项](#Options-18)

#### 15.11.1 选项

b

以比特/秒为单位设置比特率。仅支持以下比特率，否则 libavcodec 将舍入到最接近的有效比特率。

' 6600 '

' 8850 '

' 12650 '

' 14250 '

' 15850 '

' 18250 '

' 19850 '

' 23050 '

' 23850 '

数字传输系统

设置为 1 时允许不连续传输（产生舒适噪音）。默认值为 0（禁用）。

### 15.12 libvorbis

libvorbis 编码器包装器。

在配置期间需要存在 libvorbisenc 标头和库。您需要使用--enable-libvorbis显式配置构建。

* [选项](#Options-19)

#### 15.12.1 选项

libvorbis 包装器支持以下选项。选项的oggenc等价物列在括号中。

要获得更准确和更广泛的 libvorbis 选项文档，请查阅 libvorbisenc 和oggenc的文档。请参阅[http://xiph.org/vorbis/、http://wiki.xiph.org/Vorbis-tools](http://xiph.org/vorbis/)和 oggenc(1) [。](http://wiki.xiph.org/Vorbis-tools)

b ( -b )

为 ABR 设置以比特/秒表示的比特率。奥根 -b以千比特/秒表示。

q ( -q )

为 VBR 设置恒定质量设置。该值应该是一个介于 -1.0 到 10.0 之间的浮点数。值越高，质量越好。默认值为“ 3.0 ”。

ffmpeg命令行工具时有效。对于库接口用户，请使用global\_quality 。

截止 ( --advanced-encode-option lowpass\_frequency=N )

以 Hz 为单位设置截止带宽，值为 0 将禁用截止。 oggenc的相关选项以 kHz 表示。默认值为“ 0 ”（禁用截止）。

最小速率 ( -m )

设置以比特/秒表示的最小比特率。奥根 -m以千比特/秒表示。

最大速率 ( -M )

设置以比特/秒表示的最大比特率。奥根 -M以千比特/秒表示。这仅对 ABR 模式有影响。

iblock ( --advanced-encode-option impulse\_noisetune=N )

为脉冲块设置本底噪声偏置。该值是一个从 -15.0 到 0.0 的浮点数。负偏差指示编码器特别注意编码音频中瞬变的清晰度。更好的瞬态响应的折衷是更高的比特率。

### 15.13 mjpeg

运动 JPEG 编码器。

* [选项](#Options-20)

#### 15.13.1 选项

霍夫曼

设置霍夫曼编码策略。可能的值：

'默认'

使用默认的霍夫曼表。这是默认策略。

'最佳'

计算和使用最佳霍夫曼表。

### 15.14 wavpack

WavPack 无损音频编码器。

* [选项](#Options-21)

#### 15.14.1 选项

wavpack命令行实用程序的等效选项列在括号中。

* [共享选项](#Shared-options)
* [私人选项](#Private-options)

#### 15.14.1.1 共享选项

以下共享选项对此编码器有效。此处仅记录有关此特定编码器的特殊说明。有关选项的一般含义，请参阅[编解码器选项一章](#codec_002doptions)。

frame\_size ( --blocksize )

对于此编码器，此选项的范围介于 128 和 131072 之间。默认值是根据采样率和通道数自动确定的。

有关计算默认值的完整公式，请参阅libavcodec/wavpackenc.c 。

compression\_level（ -f 、 -h 、 -hh和-x ）

#### 15.14.1.2 私有选项

关节立体声（ -j ）

设置是否开启联合立体声。有效值为：

'在（ 1 ） '

强制中/侧音频编码。

'关闭（ 0 ） '

强制左/右音频编码。

'汽车'

让编码器自动决定。

优化单声道

设置是否启用单声道优化。此选项仅对非单声道流有效。可用值：

'上'

启用

'关闭'

禁用

## 16 个视频编码器

下面是一些当前可用的视频编码器的描述。

* [a64\_multi, a64\_multi5](#a64_005fmulti_002c-a64_005fmulti5)
* [电影胶片](#Cinepak)
* [动图](#GIF)
* [哈普](#Hap)
* [jpeg2000格式](#jpeg2000)
* [图书馆1e](#librav1e)
* [libaom-av1](#libaom_002dav1)
* [libsvtav1](#libsvtav1)
* [libjxl](#libjxl)
* [libkvazaar](#libkvazaar)
* [libopenh264](#libopenh264)
* [libtheora](#libtheora)
* [libvpx](#libvpx)
* [libwebp](#libwebp)
* [libx264、libx264rgb](#libx264_002c-libx264rgb)
* [libx265](#libx265)
* [libxavs2](#libxavs2)
* [libxvid](#libxvid)
* [媒体基金会](#MediaFoundation)
* [mpeg2格式](#mpeg2)
* [PNG](#png)
* [ProRes格式](#ProRes)
* [QSV 编码器](#QSV-Encoders)
* [雪](#snow)
* [VAAPI编码器](#VAAPI-encoders)
* [vbn](#vbn)
* [VC2](#vc2)

### 16.1 a64\_multi, a64\_multi5

A64 / Commodore 64 多色字符集编码器。 a64\_multi5扩展为第 5 种颜色 (colram)。

### 16.2 Cinepak

Cinepak 又名 CVID 编码器。与 Windows 3.1 和老式 MacOS 兼容。

* [选项](#Options-22)

#### 16.2.1 选项

g整数

关键帧间隔。至少每-g帧插入一个关键帧，有时更快。

q:v整数

品质因数。越低越好。较高的比特率较低。下表列出了使用-g 100为-q:v的各种值编码 akiyo\_cif.y4m 时的比特率：

-q:v 1 1918 kb/s

-q:v 2 1735 kb/s

-q:v 4 1500 kb/s

-q:v 10 1041 kb/s

-q:v 20 826 kb/s

-q:v 40 553 kb/s

-q:v 100 394 kb/s

-q:v 200 312 kb/s

-q:v 400 266 kb/s

-q:v 1000 237 kb/s

max\_extra\_cb\_iterations整数

Max extra codebook 重新计算通过，越多越好，越慢。

skip\_empty\_cb布尔值

避免浪费字节，忽略老式 MacOS 解码器。

max\_strips整数

min\_strips整数

要使用的条带的最小和最大数量。更宽的范围有时会提高质量。更多条带通常质量更好，但成本更高。更少的片段往往会产生更多的关键帧。复古兼容是 1..3。

strip\_number\_adaptivity整数

允许在帧之间更改多少条带。越高越好但越慢。

### 16.3 动图

GIF 图像/动画编码器。

* [选项](#Options-23)

#### 16.3.1 选项

giffflags整数

设置用于 GIF 编码的标志。

抵消

启用图片偏移。

默认启用。

转差

启用帧之间的透明度检测。

默认启用。

gifimage整数

启用每帧编码一个完整的 GIF 图像，而不是动画 GIF。

默认值为0 。

global\_palette整数

在可行的情况下将调色板写入全局 GIF 标头。

如果禁用，即使提供了全局调色板，每一帧都将始终写入一个调色板。

默认值为1 。

### 16.4 哈普

Vidvox Hap 视频编码器。

* [选项](#Options-24)

#### 16.4.1 选项

格式整数

指定要编码的 Hap 格式。

哈普

hap\_alpha

hap\_q

默认值为hap 。

块整数

指定要将帧拆分成的块数，介于 1 到 64 之间。这允许对大帧进行多线程解码，可能会以数据速率为代价。编码器可以修改这个值来均匀地划分帧。

默认值为1 。

压缩机整数

指定要使用的第二级压缩器。如果设置为none ，块将被限制为 1，因为分块的未压缩帧没有任何好处。

没有任何

活泼的

默认值为snappy 。

### 16.5 jpeg2000

默认情况下，原生 jpeg 2000 编码器是有损的， -q:v选项可用于设置编码质量。可以使用-pred 1选择无损编码。

* [选项](#Options-25)

#### 16.5.1 选项

格式整数

可以设置为j2k或jp2 （默认值），这样可以存储非 rgb pix\_fmts。

tile\_width整数

设置图块宽度。范围是 1 到 1073741824。默认值为 256。

tile\_height整数

设置图块高度。范围是 1 到 1073741824。默认值为 256。

预测整数

允许设置离散小波变换 (DWT) 类型

dwt97int（有损）

dwt53（无损）

默认为dwt97int

布尔值

启用此项以在每个数据包的开头添加 SOP 标记。默认情况下禁用。

eph布尔值

启用此项以在每个数据包标头的末尾添加 EPH 标记。默认情况下禁用。

编程整数

设置编码器使用的顺序。可能的值是：

lrcp

循环控制程序

rpcl

聚合酶链反应

控制程序

默认设置为lrcp 。

layer\_rates字符串

默认情况下，如果不使用此选项，则使用质量指标进行压缩。此选项允许使用压缩比进行压缩。可以指定每个级别的压缩率。层l的压缩率指定前l层中包含的总文件大小的比率。

用法示例：

ffmpeg -i input.bmp -c:v jpeg2000 -layer\_rates "100,10,1" output.j2k

这会将图像压缩为包含 3 层，其中第一层中包含的数据将被压缩 1000 倍，前两层中将压缩 100 倍，并且在使用所有 3 层时应包含所有数据。

### 16.6 图书馆

rav1e AV1 编码器包装器。

需要在配置期间存在 rav1e 标头和库。您需要使用--enable-librav1e显式配置构建。

* [选项](#Options-26)

#### 16.6.1 选项

最大值

设置使用比特率模式时要使用的最大量化器。

分钟

设置使用比特率模式时要使用的最小量化器。

qp

使用量化器模式以给定的量化器 (0-255) 进行编码。

速度

选择用于编码的速度预设 (0-10)。

瓷砖

选择要编码的图块数。

瓷砖行

选择要编码的图块行数。

瓷砖列

选择要编码的图块列数。

rav1e参数

以“:”分隔的key = value对列表设置 rav1e 选项。有关选项列表，请参阅rav1e --help 。

例如，使用-rav1e-params指定 librav1e 编码选项：

ffmpeg -i input -c:v librav1e -b:v 500K -rav1e-params speed=5:low\_latency=true 输出.mp4

### 16.7 libaom-av1

libaom AV1 编码器包装器。

在配置期间需要存在 libaom 标头和库。您需要使用--enable-libaom显式配置构建。

* [选项](#Options-27)

#### 16.7.1 选项

包装器支持以下标准 libavcodec 选项：

b

以比特/秒为单位设置比特率目标。默认情况下，这将使用可变比特率模式。如果maxrate和minrate也设置为相同的值，那么它将使用恒定比特率模式，否则如果crf也设置为好，那么它将使用受限质量模式。

g keyint\_min

设置关键帧位置。 GOP 大小设置关键帧之间的最大距离；如果为零，则输出流将是仅限内部的。除非最小距离与 GOP 大小相同，否则将忽略最小距离，在这种情况下，关键帧将始终以固定间隔出现。默认情况下未设置，因此如果没有此选项，库可以完全自由地选择放置关键帧的位置。

qmin qmax

设置最小/最大量化值。有效范围是从 0 到 63（警告：这与 AV1 实际使用的量化器值不匹配 - 除以 4 以将实际量化器值映射到该范围）。默认为最小值/最大值（无限制）。

minrate maxrate bufsize rc\_init\_occupancy

设置速率控制缓冲参数。如果未设置则不使用 - 默认为不受约束的可变比特率。

线程

设置编码时要使用的线程数。这可能需要将tiles或row-mt选项也设置为实际完全使用指定数量的线程。默认为主机支持的硬件线程数。

轮廓

设置编码配置文件。默认使用与输入的位深度和色度子采样相匹配的配置文件。

包装器还有一些特定的选项：

CPU使用

设置质量/编码速度权衡。有效范围为 0 到 8，数字越大表示速度越快但质量越差。默认值为 1，这将是缓慢且高质量的。

自动替代参考

启用备用参考帧的使用。默认为库的内部默认值。

arnr-max-frames (帧)

设置 alref 降噪最大帧数。默认为 -1。

arnr-强度（强度）

设置 alref 降噪滤波器强度。范围是 -1 到 6。默认值为 -1。

aq-模式 ( aq-模式)

设置自适应量化模式。可能的值：

'无（ 0 ） '

禁用。

'方差（ 1 ） '

基于方差。

'复杂性（ 2 ） '

基于复杂性。

'循环（ 3 ） '

循环刷新。

调（调）

设置编码器调谐的失真度量。默认为psnr 。

' psnr ( 0 ) '

' ssim ( 1 ) '

滞后帧

设置编码器可以在任何时间保持飞行的最大帧数以进行前瞻。默认为库的内部默认值。

容错性

启用错误恢复功能：

默认

提高对整个帧丢失的恢复能力。

默认情况下不启用。

crf

为恒定质量（无比特率目标）和受限质量（具有最大比特率目标）模式设置质量/大小权衡。有效范围为 0 到 63，数字越大表示质量越低且输出大小越小。仅在设置时使用；默认情况下只使用比特率目标。

静态阈值

在块上设置一个更改阈值，低于该阈值它们将被编码器跳过。以任意单位定义为非负整数，默认为零（不跳过任何块）。

下降阈值

设置接近速率控制边界时丢帧的阈值。定义为目标缓冲区的百分比 - 当速率控制缓冲区低于此百分比时，帧将被丢弃，直到它重新填充到阈值以上。默认为零（不丢帧）。

降噪噪音水平（水平）

谷物合成要去除的噪音量。如果此选项未设置或设置为 0，则禁用颗粒合成。

降噪块大小（像素）

用于谷物合成去噪的块大小。如果未设置，AV1 编解码器将使用默认值 32。

下冲-pct ( pct )

设置目标比特率的数据率下冲（最小）百分比。范围是 -1 到 100。默认值为 -1。

过冲-pct ( pct )

设置目标比特率的数据率过冲（最大）百分比。范围是 -1 到 1000。默认值为 -1。

minsection-pct ( pct )

GOP 比特率与目标比特率的最小百分比变化。如果未设置 minsection-pct，libaomenc 包装器将按如下方式计算它： (minrate \* 100 / bitrate) 。范围是 -1 到 100。默认值为 -1（未设置）。

最大截面-pct ( pct )

GOP 比特率与目标比特率的最大百分比变化。如果未设置 maxsection-pct，libaomenc 包装器将按如下方式计算它： (maxrate \* 100 / bitrate) 。范围是 -1 到 5000。默认值为 -1（未设置）。

帧平行（布尔值）

启用帧并行解码功能。默认为真。

瓷砖

设置用于对输入视频进行编码的图块数，作为列 x 行。较大的数字允许编码和解码中的更大并行性，但可能会降低编码效率。默认为输入视频大小所需的最小图块数（对于最大并包括 4K 的尺寸，这是 1x1（即单个图块））。

平铺列 平铺行

将图块数设置为图块行数和列数的 log2。提供与 libvpx/VP9 的兼容性。

row-mt（需要 libaom >= 1.0.0-759-g90a15f4f2）

启用基于行的多线程。默认情况下禁用。

启用 cdef（布尔值）

启用约束定向增强滤波器。 libaom-av1 编码器默认启用 CDEF。

启用恢复（布尔值）

启用循环恢复过滤器。 libaom-av1 的默认值为真。

启用全局运动（布尔值）

允许使用全局运动进行块预测。默认为真。

启用-intrabc（布尔值）

为帧内块预测启用块复制模式。此模式对屏幕内容很有用。默认为真。

enable-rect-partitions ( boolean ) (需要 libaom >= v2.0.0)

启用矩形分区。默认为真。

enable-1to4-partitions ( boolean ) (需要 libaom >= v2.0.0)

启用 1:4/4:1 分区。默认为真。

enable-ab-partitions ( boolean ) (需要 libaom >= v2.0.0)

启用 AB 形分区。默认为真。

启用角度增量（布尔值）（需要 libaom >= v2.0.0）

启用角度增量帧内预测。默认为真。

enable-cfl-intra ( boolean ) (需要 libaom >= v2.0.0)

启用从亮度帧内预测预测的色度。默认为真。

enable-filter-intra ( boolean ) （需要 libaom >= v2.0.0）

启用过滤器内部预测器。默认为真。

enable-intra-edge-filter ( boolean ) (需要 libaom >= v2.0.0)

启用内部边缘过滤器。默认为真。

enable-smooth-intra ( boolean ) (需要 libaom >= v2.0.0)

启用平滑帧内预测模式。默认为真。

enable-paeth-intra ( boolean ) (需要 libaom >= v2.0.0)

在帧内预测中启用路径预测器。默认为真。

启用调色板（布尔值）（需要 libaom >= v2.0.0）

启用调色板预测模式。默认为真。

enable-flip-idtx ( boolean ) (需要 libaom >= v2.0.0)

启用扩展变换类型，包括FLIPADST\_DCT、DCT\_FLIPADST、FLIPADST\_FLIPADST、ADST\_FLIPADST、FLIPADST\_ADST、IDTX、V\_DCT、H\_DCT、V\_ADST、H\_ADST、V\_FLIPADST、H\_FLIPADST。默认为真。

启用 tx64（布尔值）（需要 libaom >= v2.0.0）

启用 64 磅变换。默认为真。

reduced-tx-type-set ( boolean ) (需要 libaom >= v2.0.0)

使用减少的转换类型集。默认为假。

use-intra-dct-only ( boolean ) (需要 libaom >= v2.0.0)

仅将 DCT 用于 INTRA 模式。默认为假。

use-inter-dct-only ( boolean ) (需要 libaom >= v2.0.0)

仅将 DCT 用于 INTER 模式。默认为假。

use-intra-default-tx-only ( boolean ) (需要 libaom >= v2.0.0)

仅对 INTRA 模式使用默认转换。默认为假。

enable-ref-frame-mvs ( boolean ) (需要 libaom >= v2.0.0)

启用时间 mv 预测。默认为真。

enable-reduced-reference-set ( boolean ) (需要 libaom >= v2.0.0)

使用简化的单一和复合参考文献集。默认为假。

enable-obmc ( boolean ) (需要 libaom >= v2.0.0)

启用 obmc。默认为真。

启用双过滤器（布尔值）（需要 libaom >= v2.0.0）

启用双过滤器。默认为真。

enable-diff-wtd-comp ( boolean ) (需要 libaom >= v2.0.0)

启用差异加权复合。默认为真。

enable-dist-wtd-comp（布尔值）（需要 libaom >= v2.0.0）

启用距离加权复合。默认为真。

enable-onesided-comp ( boolean ) (需要 libaom >= v2.0.0)

启用单面复合。默认为真。

enable-interinter-wedge ( boolean ) (需要 libaom >= v2.0.0)

启用interinter wedge compound。默认为真。

enable-interintra-wedge ( boolean ) (需要 libaom >= v2.0.0)

启用内部楔形复合。默认为真。

enable-masked-comp ( boolean ) (需要 libaom >= v2.0.0)

启用掩蔽化合物。默认为真。

enable-interintra-comp ( boolean ) (需要 libaom >= v2.0.0)

启用内部化合物。默认为真。

enable-smooth-interintra ( boolean ) (需要 libaom >= v2.0.0)

启用平滑帧内模式。默认为真。

aom参数

以“:”分隔的key = value对列表设置 libaom 选项。有关受支持选项的列表，请参阅“AV1 特定选项”部分下的aomenc --help 。

例如，使用-aom-params指定 libaom 编码选项：

ffmpeg -i 输入 -c:v libaom-av1 -b:v 500K -aom-params tune=psnr:enable-tpl-model=1 output.mp4

### 16.8 libsvtav1

SVT-AV1 编码器包装器。

需要在配置期间存在 SVT-AV1 标头和库。您需要使用--enable-libsvtav1显式配置构建。

* [选项](#Options-28)

#### 16.8.1 选项

轮廓

设置编码配置文件。

'主要'

'高'

'专业'

等级

设置操作点级别。例如：'4.0'

高级

设置分层预测级别。

' 3级'

' 4级'

这是默认值。

等级

设置操作点层。

'主要'

这是默认设置。

'高'

最大值

设置使用比特率模式时要使用的最大量化器。

分钟

设置使用比特率模式时要使用的最小量化器。

crf

crf 速率控制模式中使用的恒定速率因子值 (0-63)。

qp

设置 cqp 速率控制模式中使用的量化器 (0-63)。

sc\_检测

启用场景变化检测。

la\_depth

设置要向前看的帧数 (0-120)。

预设

在 0 到 13 的范围内设置质量-速度权衡。值越高速度越快但质量越低。

tile\_rows

设置要使用的图块行数的 log2 (0-6)。

tile\_columns

设置要使用的图块列数的 log2 (0-4)。

svtav1参数

以“:”分隔的键=值对列表设置 SVT-AV1 选项。有关可接受的参数列表，请参阅 SVT-AV1 编码器用户指南。

### 16.9 libjxl

libjxl JPEG XL 编码器包装器。

在配置期间需要存在 libjxl 标头和库。您需要使用--enable-libjxl显式配置构建。

* [选项](#Options-29)

#### 16.9.1 选项

libjxl 包装器支持以下选项：

距离

设置目标 Butteraugli 距离。这是一个质量设置：较低的距离产生较高的质量，distance=1.0 大致相当于摄影内容的 libjpeg Quality 90。设置 distance=0.0 会产生真正的无损编码。有效值介于 0.0 和 15.0 之间，正常值很少超过 5.0。设置 distance=0.1 通常可以获得大多数输入的透明度。默认值为 1.0。

努力

设置使用的编码效果。更高的努力值会产生更一致的质量，并且通常会产生更好的质量/bpp 曲线，但代价是需要更多的 CPU 时间。有效值范围为 1 到 9，默认值为 7。

模块化的

强制编码器使用模块化模式而不是自动选择。默认是使用 VarDCT 进行有损编码，使用 Modular 进行无损编码。对于有损编码，VarDCT 通常优于 Modular，但不支持无损编码。

### 16.10 libkvazaar

Kvazaar H.265/HEVC 编码器。

在配置期间需要存在 libkvazaar 标头和库。您需要使用--enable-libkvazaar显式配置构建。

* [选项](#Options-30)

#### 16.10.1 选项

b

以 bit/s 为单位设置目标视频比特率并启用速率控制。

kvazaar参数

逗号 (,) 分隔的名称=值对列表。有关选项列表，请参阅 kvazaar 文档。

### 16.11 libopenh264

Cisco libopenh264 H.264/MPEG-4 AVC 编码器包装器。

此编码器需要在配置期间存在 libopenh264 标头和库。您需要使用--enable-libopenh264显式配置构建。使用pkg-config检测库。

有关该库的更多信息，请参阅<http://www.openh264.org>。

* [选项](#Options-31)

#### 16.11.1 选项

以下 FFmpeg 全局选项影响 libopenh264 编码器的配置。

b

设置比特率（每秒比特数）。

G

设置 GOP 大小。

最大速率

设置最大比特率（每秒比特数）。

标志 +global\_header

在比特流中设置全局标头。

切片

设置切片数，用于并行编码。默认值为 0。这仅在slice\_mode设置为“ fixed ”时使用。

环路滤波器

启用环路过滤器，如果设置为 1（自动启用）。要禁用，请将值设置为 0。

轮廓

设置配置文件限制。如果设置为“ main ”的值，则启用 CABAC（将SEncParamExt.iEntropyCodingModeFlag标志设置为 1）。

最大尺寸

以字节为单位设置最大 NAL 大小。

允许跳过帧

如果设置为 1，则允许跳帧达到目标比特率。

### 16.12 libtheora

libtheora Theora 编码器包装器。

需要在配置期间存在 libtheora 标头和库。您需要使用--enable-libtheora显式配置构建。

有关 libtheora 项目的更多信息，请参阅<http://www.theora.org/>。

* [选项](#Options-32)
* [例子](#Examples-5)

#### 16.12.1 选项

以下全局选项映射到影响编码流的质量和比特率的内部 libtheora 选项。

b

为 CBR（恒定比特率）模式设置以比特/秒为单位的视频比特率。如果启用 VBR（可变比特率）模式，则忽略此选项。

旗帜

用于通过qscale标志启用恒定质量模式 (VBR) 编码，并启用pass1和pass2模式。

G

设置 GOP 大小。

全局质量

将全局质量设置为以 lambda 为单位的整数。

仅当使用标志 +qscale启用 VBR 模式时相关。该值通过除以FF\_QP2LAMBDA转换为 QP 单位，在 [0 - 10] 范围内裁剪，然后乘以 6.3 以获得原生 libtheora 范围 [0-63] 中的值。较高的值对应于较高的质量。

q

设置为非负值时启用 VBR 模式，并将恒定质量值设置为以 QP 为单位的双精度浮点值。

该值被裁剪在 [0-10] 范围内，然后乘以 6.3 以获得原生 libtheora 范围 [0-63] 中的值。

ffmpeg命令行工具时有效。对于库接口用户，请使用global\_quality 。

#### 16.12.2 例子

* ffmpeg设置最大恒定质量 (VBR) 编码：
* ffmpeg -i 输入 -codec:v libtheora -q:v 10 输出.ogg
* 使用ffmpeg转换 CBR 1000 kbps Theora 视频流：
* ffmpeg -i 输入 -codec:v libtheora -b:v 1000k 输出.ogg

### 16.13 libvpx

通过 libvpx 支持的 VP8/VP9 格式。

在配置期间需要存在 libvpx 标头和库。您需要使用--enable-libvpx显式配置构建。

* [选项](#Options-33)

#### 16.13.1 选项

libvpx 包装器支持以下选项。 vpxenc等效选项或值列在括号中以便于迁移。

为了减少文档的重复，这里只记录私有选项和其他一些需要特别注意的选项。有关未记录的通用选项的文档，请参阅[编解码器选项一章](#codec_002doptions)。

要获得有关 libvpx 选项的更多文档，请调用命令ffmpeg -h encoder=libvpx 、 ffmpeg -h encoder=libvpx-vp9或vpxenc --help 。 libvpx API 文档中提供了更多信息。

b（目标比特率）

以比特/秒为单位设置比特率。请注意，FFmpeg 的b选项以 bits/s 表示，而vpxenc的target-bitrate以 kilobits/s 为单位。

g ( kf-最大距离)

keyint\_min ( kf-min-dist )

qmin (最小-q )

最小（最佳质量）量化器。

qmax (最大-q )

最大（最差质量）量化器。可以每帧更改。

bufsize ( buf-sz , buf-optimal-sz )

设置速率控制缓冲区大小（以位为单位）。注意vpxenc的选项以毫秒为单位指定，libvpx 包装器按如下方式转换此值： buf-sz = bufsize \* 1000 / bitrate ， buf-optimal-sz = bufsize \* 1000 / bitrate \* 5 / 6 。

rc\_init\_occupancy ( buf-initial-sz )

设置解码开始前应加载到 rc 缓冲区的位数。注意vpxenc的选项以毫秒为单位指定，libvpx 包装器按如下方式转换此值： rc\_init\_occupancy \* 1000 / bitrate 。

下冲-pct

设置目标比特率的数据率下冲（最小）百分比。

过冲-pct

设置目标比特率的数据率过冲（最大）百分比。

skip\_threshold (丢帧)

qcomp (偏置-pct )

最大速率（最大部分-pct ）

以 bits/s 为单位设置 GOP 最大比特率。注意vpxenc的选项指定为目标比特率的百分比，libvpx 包装器按如下方式转换此值： (maxrate \* 100 / bitrate) 。

最小速率 ( minsection-pct )

以比特/秒为单位设置 GOP 最小比特率。注意vpxenc的选项指定为目标比特率的百分比，libvpx 包装器按如下方式转换此值： (minrate \* 100 / bitrate) 。

minrate, maxrate, b最终使用=cbr

（最小速率 == 最大速率 == 比特率） 。

crf（最终用途=cq ， cq-level ）

调（调）

' psnr ( psnr ) '

' ssim ( ssim ) '

质量、截止日期（ deadline ）

'最好的'

使用最佳质量期限。命名不当且速度很慢，应避免使用此选项，因为它可能会产生质量较差的输出。

'好'

使用高质量的截止日期。当与cpu-used选项一起使用时，这是速度和质量之间的一个很好的权衡。

'实时'

使用实时质量期限。

速度，cpu-used ( cpu-used )

设置质量/速度比修改器。较高的值会以质量为代价加快编码速度。

nr（噪音敏感度）

静态阈值

在块上设置一个更改阈值，低于该阈值它们将被编码器跳过。

切片（令牌部分）

请注意，FFmpeg 的slices选项给出了分区总数，而vpxenc的token-parts给出为log2(partitions) 。

最大帧内速率

将最大 I 帧比特率设置为目标比特率的百分比。值为 0 表示无限制。

force\_key\_frames

VPX\_EFLAG\_FORCE\_KF

备用参考系相关

自动替代参考

启用备用参考帧（仅限 2-pass）。大于 1 的值启用多层交替参考帧（仅限 VP9）。

arr-maxframes

设置 alref 降噪最大帧数。

arr类型

设置 alref 降噪滤波器类型：向后、向前、居中。

兵力

设置 alref 降噪滤波器强度。

rc-lookahead, 滞后帧 ( lag-in-frames )

设置帧数以预测帧类型和速率控制。

最小 gf 间隔

设置最小黄金/备用参考帧间隔（仅限 VP9）。

容错

启用错误弹性功能。

锐度整数

以降低 PSNR 为代价提高清晰度。有效范围是 [0, 7]。

ts参数

使用 : 分隔的键=值对列表设置时间可伸缩性配置。例如，要使用ffmpeg指定时间可伸缩性参数：

ffmpeg -i INPUT -c:v libvpx -ts-parameters ts\_number\_layers=3:\

ts\_target\_bitrate=250,500,1000:ts\_rate\_decimator=4,2,1:\

ts\_periodicity=4:ts\_layer\_id=0,2,1,2:ts\_layering\_mode=3 输出

更多细节请参考vpx/vpx\_encoder.h中的struct vpx\_codec\_enc\_cfg 。

ts\_number\_layers

时间编码层数。

ts\_target\_bitrate

每个时间层的目标比特率（以 kbps 为单位）。 （比特率应该包括较低的时间层）。

ts\_rate\_decimator

每个时间层的帧速率抽取因子。

ts\_periodicity

定义帧时间层成员资格的序列长度。

ts\_layer\_id

定义帧成员到时间层的模板。

ts\_layering\_mode

（可选）从一组预定义的时间分层模式中选择时间结构。目前支持以下选项。

0

内部不提供时间分层标志，依赖于使用带有以下键的AVFrame中的元数据字段传递的标志。

vp8-标志

设置传递给编码器的标志以指示当前帧的引用方案。有关详细信息，请参阅vpx/vpx\_encoder.h中的函数vpx\_codec\_encode 。

temporal\_id

显式设置要编码的当前帧的时间 ID。

2个

两个时间层。 0-1...

3个

三个时间层。 0-2-1-2...；具有单一参考系。

4个

与选项“3”相同，只是时间周期内的两个时间层 2 帧之间存在依赖关系。

VP9 特定选项

无损的

启用无损模式。

瓷砖列

设置要使用的图块列数。请注意，这是作为log2(tile\_columns)给出的。例如，通过将tile-columns选项设置为 3 将请求 8 个 tile 列。

瓷砖行

设置要使用的图块行数。请注意，这是作为log2(tile\_rows)给出的。例如，通过将tile-rows选项设置为 2 将请求 4 个 tile 行。

帧平行

启用帧并行解码功能。

水模式

设置自适应量化模式（0：关闭（默认），1：方差 2：复杂性，3：循环刷新，4：equator360）。

色彩空间色彩空间

设置输入色彩空间。 VP9 比特流支持用信号发送以下色彩空间：

'rgb' sRGB

' BT709'BT709

'未指定'未知

'bt470bg' bt601

'smpte170m' smpte170

'smpte240m' smpte240

'bt2020\_ncl' bt2020

行山布尔

启用基于行的多线程。

调整内容

设置内容类型：默认 (0)、屏幕 (1)、电影 (2)。

语料库复杂性

Corpus VBR 模式是标准 VBR 的一种变体，其中复杂度分布中点是传入的，而不是针对特定剪辑或块计算的。

有效范围是 [0, 10000]。 0（默认值）使用标准 VBR。

启用-tpl布尔值

启用时间依赖模型。

参考框架配置

在vpx/vp8cx.h中设置结构vpx\_svc\_ref\_frame\_config\_t的成员以精细控制引用方案和帧缓冲区管理。  
使用 : 分隔的键=值对列表。例如，

av\_dict\_set(&av\_frame->metadata, "ref-frame-config", \

“rfc\_update\_buffer\_slot=7:rfc\_lst\_fb\_idx=0:rfc\_gld\_fb\_idx=1:rfc\_alt\_fb\_idx=2:rfc\_reference\_last=0:rfc\_reference\_golden=0:rfc\_reference\_alt\_ref=0”);

rfc\_update\_buffer\_slot

指示要更新的缓冲区槽号

rfc\_update\_last

指示是否更新最后一帧

rfc\_update\_golden

指示是否更新 GOLDEN 框架

rfc\_update\_alt\_ref

指示是否更新 ALT\_REF 帧

rfc\_lst\_fb\_idx

最后帧缓冲区索引

rfc\_gld\_fb\_idx

GOLDEN 帧缓冲区索引

rfc\_alt\_fb\_idx

ALT\_REF 帧缓冲区索引

rfc\_reference\_last

指示是否引用最后一帧

rfc\_reference\_golden

指示是否引用 GOLDEN 框架

rfc\_reference\_alt\_ref

指示是否引用 ALT\_REF 帧

rfc\_reference\_duration

指示帧持续时间

有关 libvpx 的更多信息，请参阅： http: [//www.webmproject.org/](http://www.webmproject.org/)

### 16.14 libwebp

libwebp WebP 图像编码器包装器

libwebp 是 Google 的官方 WebP 图像编码器。它可以以有损或无损模式编码。有损图像本质上是 VP8 帧的包装器。无损图像是由谷歌开发的独立编解码器。

* [像素格式](#Pixel-Format)
* [选项](#Options-34)

#### 16.14.1 像素格式

目前，libwebp 由于格式和 libwebp 的限制，仅支持有损的 YUV420 和无损的 RGB。两种模式都支持 Alpha。由于API的限制，如果有损编码时传入RGB或无损编码时传入YUV，像素格式将自动使用libwebp中的函数进行转换。这并不理想，这样做只是为了方便。

#### 16.14.2 选项

-无损布尔值

启用/禁用无损模式的使用。默认为 0。

-compression\_level整数

对于有损，这是质量/速度的权衡。较高的值以增加编码时间为代价为给定大小提供更好的质量。对于无损，这是大小/速度的权衡。较高的值以增加编码时间为代价提供较小的大小。更具体地说，它控制使用的额外算法和压缩工具的数量，并改变这些工具的组合。这映射到libwebp 中的方法选项。有效范围为 0 到 6。默认值为 4。

-质量浮动

对于有损编码，这控制图像质量。对于无损编码，这控制了压缩所花费的精力和时间。范围是 0 到 100。默认值为 75。

预设类型

配置预设。这会根据图像的一般类型进行一些自动设置。

没有任何

不要使用预设。

默认

使用编码器默认值。

图片

数码照片，如人像、内景

照片

室外照片，自然采光

绘画

手绘或线条图，具有高对比度细节

图标

小尺寸彩色图像

文本

类文字

### 16.15 libx264、libx264rgb

x264 H.264/MPEG-4 AVC 编码器包装器。

此编码器需要在配置期间存在 libx264 标头和库。您需要使用--enable-libx264显式配置构建。

libx264 支持大量功能，包括 8x8 和 4x4 自适应空间变换、自适应 B 帧放置、CAVLC/CABAC 熵编码、隔行扫描 (MBAFF)、无损模式、用于细节保留的 psy 优化（自适应量化、psy-RD、psy -格子）。

许多 libx264 编码器选项映射到 FFmpeg 全局编解码器选项，而独特的编码器选项通过私有选项提供。此外， x264opts和x264-params私有选项允许传递 libx264 x264\_param\_parse函数接受的 key=value 元组列表。

x264 项目网站位于<http://www.videolan.org/developers/x264.html>。

libx264rgb 编码器与 libx264 相同，除了它接受打包的 RGB 像素格式而不是 YUV 作为输入。

* [支持的像素格式](#Supported-Pixel-Formats)
* [选项](#Options-35)

#### 16.15.1 支持的像素格式

x264 支持 8 到 10 位色彩空间。确切的位深度在 x264 的配置时间进行控制。

#### 16.15.2 选项

libx264 包装器支持以下选项。 x264等效选项或值列在括号中以便于迁移。

为了减少文档的重复，这里只记录私有选项和其他一些需要特别注意的选项。有关未记录的通用选项的文档，请参阅[编解码器选项一章](#codec_002doptions)。

要获得更准确和更广泛的 libx264 选项文档，请调用命令x264 --fullhelp或查阅 libx264 文档。

b（比特率）

以比特/秒为单位设置比特率。请注意，FFmpeg 的b选项以 bits/s 表示，而x264的比特率以 kilobits/s 为单位。

bf（ bframes ）

g (键盘输入)

最小值 (最小值)

最小量化标度。

qmax ( qpmax )

最大量化标度。

qdiff ( qpstep )

量化标度之间的最大差异。

qblur（ qblur ）

量化曲线模糊

qcomp ( qcomp )

量化曲线压缩因子

参考 (参考)

每个 P 帧可以使用的参考帧数。范围是从0-16 。

sc\_threshold（场景剪辑）

设置场景变化检测的阈值。

格子（格子）

执行网格量化以提高效率。默认启用。

天然橡胶 (天然橡胶)

me\_range (梅兰奇)

运动搜索的最大范围（以像素为单位）。

我\_方法（我）

设置运动估计方法。按速度降序排列的可能值：

'直径（直径） '

' epzs（直径） '

半径为 1 的菱形搜索（最快）。 “ epzs ”是“ dia ”的别名。

'十六进制（十六进制） '

半径为 2 的六边形搜索。

'嗯（嗯） '

不均匀的多六边形搜索。

'欧空局（欧空局） '

详尽的搜索。

'德莎（德莎） '

Hadamard 穷举搜索（最慢）。

强制IDR

通常，在强制I帧类型时，编码器可以选择任何类型的I帧。这个选项强制它选择一个 IDR 框架。

subq ( subme )

亚像素运动估计方法。

b\_strategy ( b-adapt )

自适应 B 帧放置决策算法。仅在第一次通过时使用。

keyint\_min (最小键值)

最小 GOP 大小。

编码员

设置熵编码器。可能的值：

'交流'

启用 CABAC。

' vlc '

启用 CAVLC 并禁用 CABAC。它产生与x264的--no-cabac选项相同的效果。

厘米

设置全像素运动估计比较算法。可能的值：

'色度'

在运动估计中启用色度。

'悲伤'

忽略运动估计中的色度。它产生与x264的--no-chroma-me选项相同的效果。

线程（线程）

编码线程数。

线程类型

设置多线程技术。可能的值：

'切片'

基于切片的多线程。它产生与x264的--sliced-threads选项相同的效果。

'框架'

基于帧的多线程。

旗帜

设置编码标志。它可用于禁用封闭 GOP 并通过将其设置为-cgop来启用开放 GOP 。结果类似于x264的--open-gop选项的行为。

rc\_init\_occupancy ( vbv-init )

预设（预设）

设置编码预设。

调（调）

设置编码参数的调整。

个人资料（个人资料）

设置配置文件限制。

快速通过

设置为 1 时启用编码第一遍时的快速设置。设置为 0 时，它与x264的--slow-firstpass选项具有相同的效果。

手抄报 (手抄报)

设置恒定质量模式的质量。

crf\_max ( crf-max )

在 CRF 模式下，防止 VBV 降低质量超过这一点。

qp ( qp )

设置恒定量化率控制方法参数。

aq-模式 ( aq-模式)

设置 AQ 方法。可能的值：

'无（ 0 ） '

禁用。

'方差（ 1 ） '

方差 AQ（复杂性掩码）。

'自方差（ 2 ） '

自动方差 AQ（实验）。

水质强度 ( aq-strength )

设置 AQ 强度，减少平坦和纹理区域的阻塞和模糊。

心理学

设置为 1 时使用心理视觉优化。设置为 0 时，它与x264的--no-psy选项具有相同的效果。

psy-rd ( psy-rd )

psy-rd中设置心理视觉优化的强度： psy-trellis格式。

rc-lookahead ( rc-lookahead )

设置帧数以预测帧类型和速率控制。

权重b

设置为 1 时启用 B 帧的加权预测。设置为 0 时，它与x264的--no-weightb选项具有相同的效果。

权重p（权重p ）

为 P 帧设置加权预测方法。可能的值：

'无（ 0 ） '

残疾人

'简单（ 1 ） '

仅启用加权参考

'聪明（ 2 ） '

启用加权参考和重复

西姆 (西姆)

编码后启用计算和打印 SSIM 统计信息。

帧内刷新（帧内刷新）

设置为 1 时启用定期帧内刷新而不是 IDR 帧。

avcintra 类（类）

配置编码器以生成 AVC-Intra。有效值为 50,100 和 200

蓝光兼容（蓝光兼容）

配置编码器以兼容蓝光标准。它是设置“bluray-compat=1 force-cfr=1”的简写。

b 偏差 ( b 偏差)

设置对 B 帧使用频率的影响。

b-金字塔（ b-金字塔）

设置保留一些 B 帧作为参考的方法。可能的值：

'无（无） '

禁用。

'严格（严格） '

严格的层次金字塔。

'正常（正常） '

非严格（不兼容蓝光）。

混合参考

启用每个分区使用一个引用，而不是设置为 1 时每个宏块使用一个引用。设置为 0 时，它与x264的--no-mixed-refs选项具有相同的效果。

8x8dct

x264的--no-8x8dct选项具有相同的效果。

快跳

设置为 1 时启用 P 帧的早期跳过检测。设置为 0 时，它与x264的--no-fast-pskip选项具有相同的效果。

澳元（澳元）

设置为 1 时启用访问单元定界符。

mb树

x264的--no-mbtree选项具有相同的效果。

去块（去块）

alpha : beta形式设置环路滤波器参数。

cplxblur（ cplxblur ）

设置 QP 的波动减少（在曲线压缩之前）。

分区（分区）

将分区设置为以逗号分隔的列表。列表中的可能值：

' p8x8 '

8x8 P 帧分区。

' p4x4 '

4x4 P 帧分区。

' b8x8 '

4x4 B 帧分区。

' i8x8 '

8x8 I 帧分区。

' i4x4 '

4x4 I 帧分区。 （启用“ p4x4 ”需要启用“ p8x8 ”。启用“ i8x8 ”需要启用自适应空间变换（ 8x8dct选项）。）

'无（无） '

不考虑任何分区。

'所有（全部） '

考虑每个分区。

直接预测（直接）

设置直接 MV 预测模式。可能的值：

'无（无） '

禁用 MV 预测。

'空间（空间） '

启用空间预测。

'时间（时间） '

启用时间预测。

'自动（自动） '

自动决定。

切片最大尺寸 (切片最大尺寸)

以字节为单位设置每个切片的大小限制。如果未指定但指定了 RTP 有效负载大小 ( ps )，则使用它。

统计数据（统计数据）

设置多通道统计的文件名。

nal-hrd（ nal-hrd ）

设置信号HRD信息（需要设置vbv-bufsize ）。可能的值：

'无（无） '

禁用 HRD 信息信号。

' vbr ( vbr ) '

可变比特率。

' cbr（ cbr ） '

恒定比特率（在 MP4 容器中不允许）。

x264opts（北美）

设置任何 x264 选项，请参阅x264 --fullhelp以获取列表。

由“:”分隔的键=值对列表。在过滤器和psy-rd选项中使用“：”作为分隔符本身，使用“，”代替。他们很久以前就接受了它，但由于某种原因，这没有记录在案。

例如，使用ffmpeg指定 libx264 编码选项：

ffmpeg -i foo.mpg -c:v libx264 -x264opts keyint=123:min-keyint=20 -an out.mkv

a53cc布尔值

将隐藏式字幕（必须是 ATSC 兼容格式）导入输出。只有 mpeg2 和 h264 解码器提供这些。默认值为 1（开）。

udu\_sei布尔值

如果可用，将用户数据未注册的 SEI 导入输出。默认值为 0（关闭）。

x264 参数 (NA)

使用 : 分隔的 key=value 参数列表覆盖 x264 配置。

x264opts相同，但为了与 Libav 分支兼容而重复。

例如，使用ffmpeg指定 libx264 编码选项：

ffmpeg -i INPUT -c:v libx264 -x264-params level=30:bframes=0:weightp=0:\

cabac=0:ref=1:vbv-maxrate=768:vbv-bufsize=2000:analyse=all:me=umh:\

no-fast-pskip=1:subq=6:8x8dct=0:trellis=0 输出

提供了用于常见用法的编码 ffpreset，因此它们可以与一般预设系统一起使用（例如，传递pre选项）。

### 16.16 libx265

x265 H.265/HEVC 编码器封装器。

此编码器需要在配置期间存在 libx265 标头和库。您需要使用--enable-libx265显式配置构建。

* [选项](#Options-36)

#### 16.16.1 选项

b

设置目标视频比特率。

高炉

G

设置 GOP 大小。

keyint\_min

最小 GOP 大小。

参考

每个 P 帧可以使用的参考帧数。范围是从1-16 。

预设

设置 x265 预设。

调

设置 x265 调整参数。

轮廓

设置配置文件限制。

crf

设置恒定质量模式的质量。

qp

设置恒定量化率控制方法参数。

分钟

最小量化标度。

最大值

最大量化标度。

差分

量化标度之间的最大差异。

模糊

量化曲线模糊

qcomp

量化曲线压缩因子

i\_q因子

b\_q因子

强制IDR

通常，在强制I帧类型时，编码器可以选择任何类型的I帧。这个选项强制它选择一个 IDR 框架。

udu\_sei布尔值

如果可用，将用户数据未注册的 SEI 导入输出。默认值为 0（关闭）。

x265-参数

由“:”分隔的key = value对列表设置 x265 选项。有关选项列表，请参阅x265 --help 。

例如，使用-x265-params指定 libx265 编码选项：

ffmpeg -i 输入 -c:v libx265 -x265-params crf=26:psy-rd=1 输出.mp4

### 16.17 libxavs2

xavs2 AVS2-P2/IEEE1857.4 编码器包装器。

此编码器需要在配置期间存在 libxavs2 标头和库。您需要使用--enable-libxavs2显式配置构建。

使用以下标准 libavcodec 选项：

* b /比特率
* g / gop\_size
* bf / max\_b\_frames

编码器也有自己的特定选项：

* [选项](#Options-37)

#### 16.17.1 选项

lcu\_row\_threads

将行的并行线程数设置为 1 到 8（默认为 5）。

initial\_qp

将 xavs2 量化参数设置为 1 到 63（默认值 34）。这用于设置第一帧的初始 qp。

qp

将 xavs2 量化参数设置为 1 到 63（默认值 34）。用于设置constant-QP模式下的qp值。

最大qp

将速率控制的最大 qp 设置为 1 到 63（默认 55）。

最小qp

将速率控制的最小 qp 设置为 1 到 63（默认为 20）。

速度等级

将速度级别设置为 0 到 9（默认为 0）。越高越好但越慢。

日志级别

将日志级别设置为 -1 到 3（默认为 0）。 -1：无，0：错误，1：警告，2：信息，3：调试。

xavs2-参数

由“:”分隔的key = value对列表设置 xavs2 选项。

例如，使用-xavs2-params指定 libxavs2 编码选项：

ffmpeg -i 输入 -c:v libxavs2 -xavs2-params RdoqLevel=0 output.avs2

### 16.18 libxvid

Xvid MPEG-4 第 2 部分编码器包装器。

此编码器需要在配置期间存在 libxvidcore 标头和库。您需要使用--enable-libxvid --enable-gpl显式配置构建。

本机mpeg4编码器支持 MPEG-4 Part 2 格式，因此用户无需此库即可编码为该格式。

* [选项](#Options-38)

#### 16.18.1 选项

libxvid 包装器支持以下选项。以下一些选项已列出但未记录，并且对应于共享编解码器选项。有关其文档，请参阅[编解码器选项一章](#codec_002doptions)。其他未列出的共享选项对 libxvid 编码器没有影响。

b

G

分钟

最大值

mpeg\_quant

线程

高炉

b\_q因子

b\_qoffset

旗帜

设置特定的编码标志。可能的值：

' mv4 '

按宏块使用四个运动矢量。

'艾克'

启用高质量的 AC 预测。

'灰色'

只编码灰度。

' gmc '

启用全局运动补偿 (GMC)。

' qpel '

启用四分之一像素运动补偿。

' cgop '

启用封闭 GOP。

' global\_header '

将全局标头放在 extradata 而不是每个关键帧中。

格子

我\_方法

设置运动估计方法。按速度降序和质量升序排列的可能值：

'零'

不使用运动估计（默认）。

'豆荚'

' x1 '

'日志'

为 16x16 块启用高级菱形区域搜索，为 16x16 块启用半像素细化。 “ x1 ”和“ log ”是“ phods ”的别名。

' epzs '

启用上述所有功能，以及针对 8x8 块的高级菱形区域搜索、针对 8x8 块的半像素细化以及色度平面上的运动估计。

'满'

启用上述所有内容，以及扩展的 16x16 和 8x8 块搜索。

mbd

设置宏块决策算法。质量递增顺序的可能值：

'简单'

使用宏块比较功能算法（默认）。

'位'

为 16x16 块启用基于速率失真的半像素和四分之一像素细化。

'路'

启用上述所有功能，加上针对 8x8 块的基于速率失真的半像素和四分之一像素细化，以及使用方形模式的基于速率失真的搜索。

lumi\_aq

设置为 1 时启用 lumi masking 自适应量化。默认值为 0（禁用）。

variance\_aq

设置为 1 时启用方差自适应量化。默认值为 0（禁用）。

与lumi\_aq结合使用时，生成的质量不会优于单独指定的两个中的任何一个。换句话说，最终的质量将是两种效果中较差的一种。

西姆

设置结构相似性（SSIM）显示方法。可能的值：

'关闭'

禁止显示 SSIM 信息。

'平均'

将编码结束时的平均 SSIM 输出到标准输出。显示平均 SSIM 的格式为：

平均 SSIM：%f

对于不熟悉 C 的用户，%f 表示浮点数或小数（例如 0.939232）。

'框架'

将编码期间的每帧 SSIM 数据和编码结束时的平均 SSIM 数据输出到标准输出。每帧信息的格式为：

SSIM：平均值：%1.3f 最小值：%1.3f 最大值：%1.3f

对于不熟悉 C 的用户，%1.3f 表示圆点后四舍五入为 3 位的浮点数（例如 0.932）。

ssim\_acc

设置 SSIM 精度。有效选项是 0-4 范围内的整数，0 给出最准确的结果，4 计算最快。

### 16.19 媒体基金会

这为 MediaFoundation 框架中的编码器（音频和视频）提供了包装器。它可以访问软件和硬件编码器。视频编码器可以采用 nv12 或 yuv420p 形式的输入（有些编码器同时支持这两种形式，有些只支持其中一种——实际上，nv12 是更安全的选择，尤其是在硬件编码器中）。

### 16.20 mpeg2

MPEG-2 视频编码器。

* [选项](#Options-39)

#### 16.20.1 选项

轮廓

选择要编码的 mpeg2 配置文件：

' 422 '

'高'

' SS '

空间可扩展

'信噪比'

信噪比可扩展

'主要'

'简单'

等级

选择要编码的 mpeg2 级别：

'高'

'高1440 '

'主要'

'低'

seq\_disp\_ext整数

指定编码器是否应将 sequence\_display\_extension 写入输出。

-1

汽车

通过检查要写入的数据是否与默认值或未指定的值不同，自动决定是否写入（这是默认值）。

0

绝不

永远不要写它。

1个

总是

总是写它。

视频格式整数

指定写入序列显示扩展的video\_format，表示视频图片的来源。默认为' unspecified '，可以是' component '、' pal '、' ntsc '、' secam '或' mac '。为了获得最大的兼容性，请使用“组件”。

a53cc布尔值

将隐藏式字幕（必须是 ATSC 兼容格式）导入输出。默认值为 1（开）。

### 16.21 PNG

PNG图像编码器。

* [私人选项](#Private-options-1)

#### 16.21.1 私人期权

dpi整数

设置像素的物理密度，以每英寸点数为单位，默认不设置

dpm整数

设置像素的物理密度，以每米点数为单位，默认不设置

### 16.22 专业分辨率

Apple ProRes 编码器。

FFmpeg 包含 2 个 ProRes 编码器，prores-aw 和 prores-ks 编码器。可以使用-vcodec选项选择使用的编码器。

* [个人期权](#Private-Options-for-prores_002dks)
* [速度考虑](#Speed-considerations)

#### 16.22.1 prores-ks 的私有选项

文件整数

选择要编码的 ProRes 配置文件

'代理'

'它'

'标准'

'总部'

' 4444 '

' 4444xq '

quant\_mat整数

选择量化矩阵。

'汽车'

'默认'

'代理'

'它'

'标准'

'总部'

如果设置为auto ，将选择与配置文件匹配的矩阵。如果未设置，将选择提供最高质量的矩阵default 。

bits\_per\_mb整数

为编码一个宏块分配多少位。不同的配置文件每个宏块使用 200 到 2400 位，最大值为 8000。

mbs\_per\_slice整数

每个切片中的宏块数（1-8）；默认值 (8) 几乎适用于所有情况。

供应商字符串

覆盖 4 字节供应商 ID。像apl0这样的自定义供应商 ID会声称流是由 Apple 编码器生成的。

alpha\_bits整数

指定 alpha 分量的位数。可能的值为0 、 8和16 。使用0禁用 alpha 平面编码。

#### 16.22.2 速度考虑

在默认操作模式下，编码器必须遵守帧约束（即不生成尺寸大于请求的帧），同时仍使输出图像尽可能好。包含很多小细节的帧更难压缩，编码器将花费更多时间为每个切片搜索合适的量化器。

设置更高的bits\_per\_mb限制将提高速度。

为了获得最快的编码速度，请设置qscale参数（推荐值为 4）并且不要设置大小限制。

### 16.23 QSV 编码器

Intel QuickSync 视频编码器系列（MPEG-2、H.264、HEVC、JPEG/MJPEG 和 VP9）

* [速率控制方法](#Ratecontrol-Method)
* [全局选项 -> MSDK 选项](#Global-Options-_002d_003e-MSDK-Options)
* [常用选项](#Common-Options-1)
* [运行时选项](#Runtime-Options)
* [H264 选项](#H264-options)
* [HEVC 选项](#HEVC-Options-1)
* [MPEG2 选项](#MPEG2-Options)
* [VP9 选项](#VP9-Options)
* [AV1 选项](#AV1-Options)

#### 16.23.1 速率控制方法

ratecontrol方法选择如下：

* global\_quality时，将使用基于质量的模式。具体来说，这意味着
  + CQP - 恒定量化标度，当还设置了qscale编解码器标志时（ -qscale ffmpeg 选项）。
  + LA\_ICQ - 当look\_ahead选项也被设置时，具有 lookahead 的智能恒定质量。
  + ICQ – 否则智能恒定质量。对于 ICQ 模式，全局质量范围是 1 到 51，其中 1 是最好的质量。
* 否则，使用基于比特率的模式。对于所有这些，您至少应该使用b选项指定所需的平均比特率。
  + LA - 当指定look\_ahead选项时，具有前瞻性的 VBR 。
  + VCM - 视频会议模式，当设置了vcm选项时。
  + CBR - 恒定比特率，当指定最大速率并等于平均比特率时。
  + VBR - 可变比特率，当指定最大速率时，但高于平均比特率。
  + AVBR - 平均 VBR 模式，当未指定maxrate时， avbr\_accuracy和avbr\_convergence均设置为非零。此模式适用于 Windows 上的 H264 和 HEVC。

请注意，根据您的系统，编码器可能会选择与您指定的模式不同的模式。将详细级别设置为详细或更高，以查看 QSV 运行时使用的实际设置。

#### 16.23.2 全局选项 -> MSDK 选项

额外的 libavcodec 全局选项映射到 MSDK 选项如下：

* g/gop\_size -> GopPicSize
* bf/max\_b\_frames +1 -> GopRefDist
* rc\_init\_occupancy/rc\_initial\_buffer\_occupancy -> InitialDelayInKB
* 切片-> NumSlice
* 参考-> NumRefFrame
* b\_strategy/b\_frame\_strategy -> BRefType
* cgop/CLOSED\_GOP编解码器标志 -> GopOptFlag
* 对于CQP模式， i\_qfactor/i\_qoffset和b\_qfactor/b\_qoffset分别设置QPP和QPI以及QPP和QPB之间的差异。
* 编码器选项设置为值vlc将使 H.264 编码器使用 CAVLC 而不是 CABAC。

#### 16.23.3 常用选项

所有 qsv 编码器都使用以下选项。

异步深度

指定在应用程序显式同步结果之前应用程序执行了多少异步操作。如果为零，则不指定值。

预设

此选项列出了从非常快（最佳速度）到非常慢（最佳质量）的一系列选择。

'非常快'

'更快'

'快'

'中号'

'慢'

'慢'

'很慢'

强制IDR

将 I 帧强制为 IDR 帧。

低电量

对于编码器，将此标志设置为 ON 以降低功耗和 GPU 使用率。

#### 16.23.4 运行时选项

在 qsv 编码过程中可以使用以下选项。

全局质量

i\_quant\_factor

i\_quant\_offset

b\_quant\_factor

b\_quant\_offset

在 h264\_qsv 和 hevc\_qsv 中受支持。更改这些值以重置 qsv 编解码器的 qp 配置。

最大帧大小

在 h264\_qsv 和 hevc\_qsv 中受支持。更改此值以重置 qsv 编解码器的 MaxFrameSize 配置。

gop\_size

更改此值以重置 qsv 编解码器的 gop 配置。

int\_ref\_type

int\_ref\_cycle\_size

int\_ref\_qp\_delta

int\_ref\_cycle\_dist

在 h264\_qsv 和 hevc\_qsv 中受支持。更改这些值以重置 qsv 编解码器的帧内刷新配置。

最大值

分钟

max\_qp\_i

min\_qp\_i

最大qp\_p

min\_qp\_p

max\_qp\_b

min\_qp\_b

在 h264\_qsv 中受支持。更改这些值以重置 qsv 编解码器的最大/最小 qp 配置。

low\_delay\_brc

在 h264\_qsv 和 hevc\_qsv 中受支持。更改此值以重置 qsv 编解码器的 low\_delay\_brc 配置。

帧率

更改此值以重置 qsv 编解码器的帧率配置。

比特率

rc\_buffer\_size

rc\_initial\_buffer\_occupancy

rc\_max\_rate

更改这些值以重置 qsv 编解码器的比特率控制配置。

pic\_timing\_sei

在 h264\_qsv 和 hevc\_qsv 中受支持。更改此值以重置 qsv 编解码器的 pic\_timing\_sei 配置。

#### 16.23.5 H264 选项

这些选项由 h264\_qsv 使用

分机

扩展比特率控制。

recovery\_point\_sei

设置此标志以在每个内部刷新周期开始时插入恢复点 SEI 消息。

rdo

启用率失真优化。

最大帧大小

以字节为单位的最大编码帧大小。

max\_frame\_size\_i

I 帧的最大编码帧大小（以字节为单位）。如果此值设置为大于零，则对于 I 帧，将忽略 max\_frame\_size 设置的值。

max\_frame\_size\_p

P 帧的最大编码帧大小（以字节为单位）。如果此值设置为大于零，则对于 P 帧，将忽略 max\_frame\_size 设置的值。

最大切片大小

以字节为单位的最大编码切片大小。

比特率限制

切换比特率限制。将比特率修改为在 QSV 编码器强加的范围内。关闭此标志可能会导致违反 HRD 一致性。请注意，指定低于 QSV 编码器范围的比特率可能会显着影响质量。如果这个选项在非 CQP 模式下生效：如果比特率不在 QSV 编码器强加的范围内，它将被更改为在该范围内。

mbbrc

设置此标志启用宏块级别的比特率控制，通常可以提高主观视觉质量。启用此标志可能会对性能和客观视觉质量指标产生负面影响。

low\_delay\_brc

设置此标志可打开或关闭 qsv 插件中的 LowDelayBRC 功能，它提供更准确的比特率控制，以最大限度地减少逐帧比特流大小的变化。值：-1-默认0-关1-开

自适应我

该标志控制 QSV 编码器插入 I 帧。打开此标志以允许将帧类型从 P 和 B 更改为 I。

adaptive\_b

该标志控制帧类型从 B 到 P 的变化。

p\_strategy

启用 P-金字塔：0-默认 1-简单 2-金字塔（bf 需要设置为 0）。

b\_strategy

此选项控制 B 帧作为参考的使用。

dblk\_idc

此选项禁用去块。它的取值范围为0~2。

骑士

如果设置，则使用 CAVLC；如果未设置，CABAC 用于编码。

变频调速器

视频会议模式，参见码率控制方法。

idr\_interval

IDR 帧之间的距离（在 I 帧中）。

pic\_timing\_sei

使用 pic\_struct\_syntax 元素插入图片时序 SEI。

single\_sei\_nal\_unit

将所有的 SEI 消息放到一个 NALU 中。

max\_dec\_frame\_buffering

DPB 中缓冲的最大帧数。

展望

使用具有前瞻性的 VBR 算法。

前瞻深度

数字框架中的前瞻深度。

look\_ahead\_downsampling

为前瞻分析保存的帧的缩小比例因子。

'未知'

'汽车'

'关闭'

' 2x '

' 4x '

int\_ref\_type

指定内部刷新类型。帧内刷新的主要目标是在不显着影响由 I 帧引起的编码比特流大小的情况下提高错误恢复能力。 SDK 编码器通过使用帧内 MB 在刷新周期中对每个帧的一部分进行编码来实现这一点。 none表示不刷新。 vertical表示垂直刷新，按 MB 列。 horizontal表示水平刷新，按 MB 行。 slice表示按切片水平刷新，不重叠。在slice的情况下， in\_ref\_cycle\_size 被忽略。要启用帧内刷新，B 帧应设置为 0。

int\_ref\_cycle\_size

指定刷新周期内的图片数量，从 2 开始。0 和 1 是无效值。

int\_ref\_qp\_delta

指定插入的内部 MB 的 QP 差异。如果亮度样本的目标编码位深度为 8，则这是 [-51, 51] 范围内的有符号值，对于 10 位深度，此范围为 [-63, 63]，对于 12 位深度，则为 [-75, 75]分别。

int\_ref\_cycle\_dist

帧内刷新周期开始之间的距离。

轮廓

'未知'

'基线'

'主要'

'高'

a53cc

使用 A53 隐藏式字幕（如果可用）。

音频

插入访问单元定界符 NAL。

调频模式

多帧模式。

'关闭'

'汽车'

重复pps

对每一帧重复 pps。

max\_qp\_i

I 帧的最大视频量化比例。

min\_qp\_i

I 帧的最小视频量化比例。

最大qp\_p

P 帧的最大视频量化器标度。

min\_qp\_p

P 帧的最小视频量化比例。

max\_qp\_b

B 帧的最大视频量化比例。

min\_qp\_b

B 帧的最小视频量化比例。

设想

向编码器提供有关编码会话场景的提示。

'未知'

'显示远程处理'

“视频会议”

'存档'

'直播'

'相机捕捉'

'视频监控'

'游戏串流'

'远程游戏'

avbr\_accuracy

AVBR 速率控制的精度（单位为百分之一）。

avbr\_convergence

AVBR码率控制的收敛（以100帧为单位）

参数avbr\_accuracy和avbr\_convergence用于平均可变比特率控制 (AVBR) 算法。该算法关注整体编码质量，同时在avbr\_Convergence周期后满足精度范围avbr\_accuracy内的指定比特率target\_bitrate 。此方法不遵循 HRD，并且即时比特率没有上限或填充。

跳过帧

使用每帧元数据“qsv\_skip\_frame”在编码时跳过帧。此选项定义此元数据的用法。

'不跳过'

跳帧被禁用。

' insert\_dummy '

编码器插入比特流帧，其中所有宏块都被编码为已跳过。

' insert\_nothing '

类似于 insert\_dummy，但编码器不向比特流中插入任何内容。跳过的帧仍在 brc 中使用。例如，gop仍然包含跳帧，跳帧之后的帧尺寸会更大。

' brc\_only '

skip\_frame 元数据指示当前帧之前丢失的帧数。

#### 16.23.6 HEVC 选项

这些选项由 hevc\_qsv 使用

分机

扩展比特率控制。

recovery\_point\_sei

设置此标志以在每个内部刷新周期开始时插入恢复点 SEI 消息。

rdo

启用率失真优化。

最大帧大小

以字节为单位的最大编码帧大小。

max\_frame\_size\_i

I 帧的最大编码帧大小（以字节为单位）。如果此值设置为大于零，则对于 I 帧，将忽略 max\_frame\_size 设置的值。

max\_frame\_size\_p

P 帧的最大编码帧大小（以字节为单位）。如果此值设置为大于零，则对于 P 帧，将忽略 max\_frame\_size 设置的值。

最大切片大小

以字节为单位的最大编码切片大小。

mbbrc

设置此标志启用宏块级别的比特率控制，通常可以提高主观视觉质量。启用此标志可能会对性能和客观视觉质量指标产生负面影响。

low\_delay\_brc

设置此标志可打开或关闭 qsv 插件中的 LowDelayBRC 功能，它提供更准确的比特率控制，以最大限度地减少逐帧比特流大小的变化。值：-1-默认0-关1-开

自适应我

该标志控制 QSV 编码器插入 I 帧。打开此标志以允许将帧类型从 P 和 B 更改为 I。

adaptive\_b

该标志控制帧类型从 B 到 P 的变化。

p\_strategy

启用 P-金字塔：0-默认 1-简单 2-金字塔（bf 需要设置为 0）。

b\_strategy

此选项控制 B 帧作为参考的使用。

dblk\_idc

此选项禁用去块。它的取值范围为0~2。

idr\_interval

IDR 帧之间的距离（在 I 帧中）。

' begin\_only '

仅在流的开头输出 IDR 帧。

加载插件

在内部会话中加载的用户插件。

'无'

' hevc\_sw '

' hevc\_hw '

加载插件

一个 :- 十六进制插件 UID 的单独列表，用于在内部会话中加载。

前瞻深度

以数字帧表示的前瞻深度，在启用 extbrc 选项时可用。

轮廓

设置编码配置文件（scc 要求 libmfx >= 1.32）。

'未知'

'主要'

' main10 '

'主要'

'下一个'

' scc '

等级

设置编码层级（只有level >= 4 才能支持high tier）。此选项仅在指定级别选项时生效。

'主要'

'高'

总计

1：GPB（广义P/B帧）

0：常规P帧。

tile\_cols

分块编码的列数。

tile\_rows

分块编码的行数。

音频

插入访问单元定界符 NAL。

pic\_timing\_sei

使用 pic\_struct\_syntax 元素插入图片时序 SEI。

转换跳过

打开此选项以启用 transformskip。它在等于或更新于 ICL 的平台上受支持。

int\_ref\_type

指定内部刷新类型。帧内刷新的主要目标是在不显着影响由 I 帧引起的编码比特流大小的情况下提高错误恢复能力。 SDK 编码器通过使用帧内 MB 在刷新周期中对每个帧的一部分进行编码来实现这一点。 none表示不刷新。 vertical表示垂直刷新，按 MB 列。 horizontal表示水平刷新，按 MB 行。 slice表示按切片水平刷新，不重叠。在slice的情况下， in\_ref\_cycle\_size 被忽略。要启用帧内刷新，B 帧应设置为 0。

int\_ref\_cycle\_size

指定刷新周期内的图片数量，从 2 开始。0 和 1 是无效值。

int\_ref\_qp\_delta

指定插入的内部 MB 的 QP 差异。如果亮度样本的目标编码位深度为 8，则这是 [-51, 51] 范围内的有符号值，对于 10 位深度，此范围为 [-63, 63]，对于 12 位深度，则为 [-75, 75]分别。

int\_ref\_cycle\_dist

帧内刷新周期开始之间的距离。

max\_qp\_i

I 帧的最大视频量化比例。

min\_qp\_i

I 帧的最小视频量化比例。

最大qp\_p

P 帧的最大视频量化器标度。

min\_qp\_p

P 帧的最小视频量化比例。

max\_qp\_b

B 帧的最大视频量化比例。

min\_qp\_b

B 帧的最小视频量化比例。

设想

向编码器提供有关编码会话场景的提示。

'未知'

'显示远程处理'

“视频会议”

'存档'

'直播'

'相机捕捉'

'视频监控'

'游戏串流'

'远程游戏'

avbr\_accuracy

AVBR 速率控制的精度（单位为百分之一）。

avbr\_convergence

AVBR码率控制的收敛（以100帧为单位）

参数avbr\_accuracy和avbr\_convergence用于平均可变比特率控制 (AVBR) 算法。该算法关注整体编码质量，同时在avbr\_Convergence周期后满足精度范围avbr\_accuracy内的指定比特率target\_bitrate 。此方法不遵循 HRD，并且即时比特率没有上限或填充。

跳过帧

使用每帧元数据“qsv\_skip\_frame”在编码时跳过帧。此选项定义此元数据的用法。

'不跳过'

跳帧被禁用。

' insert\_dummy '

编码器插入比特流帧，其中所有宏块都被编码为已跳过。

' insert\_nothing '

类似于 insert\_dummy，但编码器不向比特流中插入任何内容。跳过的帧仍在 brc 中使用。例如，gop仍然包含跳帧，跳帧之后的帧尺寸会更大。

' brc\_only '

skip\_frame 元数据指示当前帧之前丢失的帧数。

#### 16.23.7 MPEG2 选项

这些选项由 mpeg2\_qsv 使用

轮廓

'未知'

'简单'

'主要'

'高'

#### 16.23.8 VP9 选项

vp9\_qsv 使用这些选项

轮廓

'未知'

'配置文件0 '

'配置文件1 '

'配置文件2 '

'配置文件3 '

tile\_cols

分块编码的列数（需要 libmfx >= 1.29）。

tile\_rows

平铺编码的行数（需要 libmfx >= 1.29）。

#### 16.23.9 AV1 选项

这些选项由 av1\_qsv 使用（需要 libvpl）。

轮廓

'未知'

'主要'

tile\_cols

分块编码的列数。

tile\_rows

分块编码的行数。

自适应我

该标志控制 QSV 编码器插入 I 帧。打开此标志以允许将帧类型从 P 和 B 更改为 I。

adaptive\_b

该标志控制帧类型从 B 到 P 的变化。

b\_strategy

此选项控制 B 帧作为参考的使用。

分机

扩展比特率控制。

前瞻深度

以数字帧表示的前瞻深度，在启用 extbrc 选项时可用。

low\_delay\_brc

设置此标志可打开或关闭 qsv 插件中的 LowDelayBRC 功能，它提供更准确的比特率控制，以最大限度地减少逐帧比特流大小的变化。值：-1-默认0-关1-开

最大帧大小

为每个帧设置允许的最大大小（以字节为单位）。如果帧大小超过限制，编码器将调整 QP 值来控制帧大小。在CQP码率控制模式下无效。

### 16.24雪

* [选项](#Options-40)

#### 16.24.1 选项

迭代直径大小

迭代运动估计的直径大小

### 16.25 VAAPI 编码器

可通过 VAAPI 访问的硬件编码器包装器。

这些编码器只接受 VAAPI 硬件表面的输入。如果您在软件帧中有输入，请使用hwupload过滤器将它们上传到 GPU。

使用以下标准 libavcodec 选项：

* g / gop\_size
* bf / max\_b\_frames
* 轮廓

如果未设置，这将根据输入帧的格式和驱动程序支持的配置文件自动确定。

* 等级
* b /比特率
* 最大速率/ rc\_max\_rate
* 缓冲区大小/ rc\_buffer\_size
* rc\_init\_occupancy / rc\_initial\_buffer\_occupancy
* 压缩级别

速度/质量权衡：更高的值更快/更差的质量。

* q / global\_quality

大小/质量权衡：值越高越小/质量越差。

* 分钟
* 最大值
* i\_qfactor / i\_quant\_factor
* i\_qoffset / i\_quant\_offset
* b\_qfactor / b\_quant\_factor
* b\_qoffset / b\_quant\_offset
* 切片

所有编码器都支持以下选项：

低电量

某些驱动程序/平台为某些编解码器提供了第二个编码器，旨在使用比默认编码器更少的功率；设置此选项将尝试使用该编码器。请注意，它可能支持缩减的功能集，因此某些其他选项可能在此模式下不可用。

idr\_interval

设置开放 GOP 模式下完全刷新 (IDR) 帧之间的正常帧内帧数。帧内帧仍然是 IRAP，但不包括全局标头并且可能具有不可解码的前导图片。

b\_depth

设置 B 帧参考深度。当设置为 1（默认值）时，所有 B 帧将仅引用 P 或 I 帧。当设置为更大的值时，将出现多层 B 帧，每一层中的帧仅引用更高层中的帧。

异步深度

最大处理并行度。增加它以提高单通道性能。如果驱动程序未实现 vaSyncBuffer 函数，则此选项不起作用。如果使用大量的 async\_depth，请确保分配了足够的 hw\_frames。

最大帧大小

为每个帧设置允许的最大大小（以字节为单位）。如果帧大小超过限制，编码器将调整 QP 值来控制帧大小。在CQP码率控制模式下无效。

rc\_mode

设置要使用的速率控制模式。给定的驱动程序可能只支持模式的一个子集。

可能的模式：

汽车

根据驱动程序支持和其他选项自动选择模式。这是默认设置。

CQP

质量稳定。

CBR

恒定比特率。

可变比特率

可变比特率。

ICQ

智能恒质。

QVBR

质量定义的可变比特率。

AVBR

平均可变比特率。

每个编码器也有自己的特定选项：

h264\_vaapi

profile设置profile\_idc和constraint\_set\*\_flag的值。 level设置level\_idc的值。

编码员

设置熵编码器（默认为cabac ）。可能的值：

'交流'

' cabac '

使用 CABAC。

' vlc '

'骑士'

使用 CAVLC。

音频

在流中包含访问单元定界符（默认情况下不包含）。

清

设置要包含的 SEI 消息类型。以下值的一些组合：

'标识符'

包括一条包含有关编码器信息的user\_data\_unregistered消息。

'计时'

包括图片计时参数（ buffering\_period和pic\_timing消息）。

'恢复点'

在适当的地方包括恢复点（ recovery\_point消息）。

hevc\_vaapi

profile和level分别设置general\_profile\_idc和general\_level\_idc的值。

音频

在流中包含访问单元定界符（默认情况下不包含）。

等级

设置general\_tier\_flag 。如果未明确指定，这可能会影响为流选择的级别。

清

设置要包含的 SEI 消息类型。以下值的一些组合：

'人类发展报告'

如果输入帧有 HDR 元数据（ mastering\_display\_colour\_volume和content\_light\_level消息），则包括 HDR 元数据。

瓷砖

设置用于对输入视频进行编码的图块数，作为列 x 行。较大的数字允许编码和解码中的更大并行性，但可能会降低编码效率。

mjpeg\_vaapi

仅支持基线 DCT 编码。编码器始终使用标准量化表和霍夫曼表 - global\_quality缩放标准量化表（范围 1-100）。

对于 YUV，支持 4:2:0、4:2:2 和 4:4:4 子采样模式。还支持 RGB，并将创建 RGB JPEG。

jfif

在每个帧中包含 JFIF 标头（默认情况下不包含）。

霍夫曼

包括标准霍夫曼表（默认打开）。关闭此功能将在每个输出帧中节省几百个字节，但可能会失去与某些不能完全处理 MJPEG 的 JPEG 解码器的兼容性。

mpeg2\_vaapi

profile和level设置profile\_and\_level\_indication的值。

vp8\_vaapi

不支持 B 帧。

global\_quality设置用于非关键帧的q\_idx （范围 0-127）。

loop\_filter\_level

loop\_filter\_sharpness

手动设置环路滤波器参数。

vp9\_vaapi

global\_quality设置用于 P 帧的q\_idx （范围 0-255）。

loop\_filter\_level

loop\_filter\_sharpness

手动设置环路滤波器参数。

支持 B 帧，但输出流始终按编码顺序而不是显示顺序。如果启用 B 帧，则可能需要使用vp9\_raw\_reorder 比特流过滤器修改输出流以正确的顺序显示帧。

仅生成正常帧 -可能需要vp9\_superframe比特流过滤器来生成可用于所有解码器的流。

### 16.26 vbn

Vizrt 二进制图像编码器。

此格式由广播供应商 Vizrt 用于快速纹理流。不支持该格式的高级功能，例如纹理数据的 LZW 压缩或 mipmap 的生成。

* [选项](#Options-41)

#### 16.26.1 选项

格式字符串

设置 VBN 文件使用的纹理压缩。可以是dxt1 、 dxt5或raw 。默认为dxt5 。

### 16.27 VC2

SMPTE VC-2（以前称为 BBC Dirac Pro）。此编解码器主要针对专业广播，但由于它支持 8（有限范围或全范围）、10 或 12 位的 yuv420、yuv422 和 yuv444，这使其适用于需要低开销和低压缩的其他任务（如屏幕录制） ).

* [选项](#Options-42)

#### 16.27.1 选项

b

设置目标视频比特率。通常这大约是未压缩视频比特率的 1:6（例如，对于 1920x1080 50fps yuv422p10，大约是 400Mbps）。较高的值（接近未压缩的比特率）打开无损压缩模式。

字段顺序

为隔行扫描输入设置时启用场编码（例如设置为 tt - 顶场优先）。应该增加对隔行扫描内容的压缩，因为它会拆分字段并分别编码。

小波深度

设置要应用的小波变换的总量，介于 1 和 5 之间（默认值）。较低的值会降低压缩率和质量。能力较差的解码器可能无法处理超过 3的wavelet\_depth值。

小波类型

设置变换类型。当前仅实现了5\_3 (LeGall) 和9\_7 (Deslauriers-Dubuc)，其中 9\_7 具有更好的压缩率，因此是默认值。

切片宽度

切片高度

设置每个切片的切片大小。值越大，压缩效果越好。为了与其他更有限的解码器兼容，使用slice\_width为 32 和slice\_height为 8。

宽容

以百分比设置速率控制系统的下冲容差。这是为了防止运行昂贵的搜索。

平方米

将量化矩阵预设设置为默认使用或wavelet\_depth设置为 5时使用

* default使用规范中的默认量化矩阵，并使用第五级的值进行扩展。这在保留细节和省略伪影之间提供了良好的平衡。
* flat使用完全归零的量化矩阵。这会增加 PSNR，但可能会降低感知。在虚假基准中使用。
* color减少细节但尝试在极低的比特率下保留颜色。

## 17个字幕编码器

* [DVD子](#dvdsub-1)

### 17.1 DVD 子

此编解码器对 DVD 中使用的位图字幕格式进行编码。通常它们存储在 VOBSUB 文件对 (\*.idx + \*.sub) 中，它们也可以用于 Matroska 文件中。

* [选项](#Options-43)

#### 17.1.1 选项

调色板

指定位图使用的全局调色板。

此选项的格式是一个包含16个24位十六进制的数字（没有0x前缀）的字符串，例如逗号分隔，例如0d00ee，ee450d，101010，eaeaea，eaeaea，0ce60b，ec14ed，ec14ed，ebff0b，ebff0b，0d617a，0d617a，7b7b7b，7b7b7b，d1d1d1d1d1d1 d1d1，7b21,7b2a，0dd1,0d9999999999999999999999999999999999999999. 0f007b、cf0dec、cfa80c、7c127b 。

even\_rows\_fix

设置为 1 时，启用一种变通方法，使所有字幕中的像素行数均匀。这解决了一些玩家在数字为奇数时切断底行的问题。如果需要，解决方法只是添加一个完全透明的行。开销很低，通常平均每个字幕一个字节。

默认情况下，此解决方法被禁用。

## 18 个比特流过滤器

当您配置 FFmpeg 构建时，默认情况下会启用所有支持的比特流过滤器。您可以使用配置选项--list-bsfs列出所有可用的。

您可以使用配置选项--disable-bsfs禁用所有比特流过滤器，并使用选项--enable-bsf=BSF选择性地启用任何比特流过滤器，或者您可以使用选项--disable-bsf禁用特定的比特流过滤器=BSF 。

选项-bsfs将显示您的构建中包含的所有支持的比特流过滤器的列表。

ff\* 工具为每个流应用了一个 -bsf 选项，采用逗号分隔的过滤器列表，其参数在“=”之后跟随过滤器名称。

ffmpeg -i 输入 -c:v 复制 -bsf:v filter1[=opt1=str1:opt2=str2][,filter2] 输出

下面是对当前可用的比特流过滤器及其参数（如果有）的描述。

* [aac\_adtstoasc](#aac_005fadtstoasc)
* [av1\_元数据](#av1_005fmetadata)
* [咀嚼](#chomp)
* [dca\_core](#dca_005fcore)
* [dump\_extra](#dump_005fextra)
* [dv\_error\_marker](#dv_005ferror_005fmarker)
* [eac3\_核心](#eac3_005fcore)
* [提取\_额外数据](#extract_005fextradata)
* [过滤器单位](#filter_005funits)
* [hapqa\_extract](#hapqa_005fextract)
* [h264\_元数据](#h264_005fmetadata)
* [h264\_mp4toannexb](#h264_005fmp4toannexb)
* [h264\_redundant\_pps](#h264_005fredundant_005fpps)
* [hevc\_元数据](#hevc_005fmetadata)
* [hevc\_mp4toannexb](#hevc_005fmp4toannexb)
* [imx转储](#imxdump)
* [mjpeg2jpeg](#mjpeg2jpeg)
* [转储](#mjpegadump)
* [mov2文本子](#mov2textsub-1)
* [mp3分解](#mp3decomp)
* [mpeg2\_元数据](#mpeg2_005fmetadata)
* [mpeg4\_unpack\_bframes](#mpeg4_005funpack_005fbframes)
* [噪音](#noise)
* [无效的](#null)
* [pcm\_rechunk](#pcm_005frechunk)
* [pgs\_frame\_merge](#pgs_005fframe_005fmerge)
* [prores\_metadata](#prores_005fmetadata)
* [remove\_extra](#remove_005fextra)
* [定居点](#setts)
* [text2movsub](#text2movsub-1)
* [trace\_headers](#trace_005fheaders)
* [真核](#truehd_005fcore)
* [vp9\_元数据](#vp9_005fmetadata)
* [vp9\_superframe](#vp9_005fsuperframe)
* [vp9\_superframe\_split](#vp9_005fsuperframe_005fsplit)
* [vp9\_raw\_reorder](#vp9_005fraw_005freorder)

### 18.1 aac\_adtstoasc

将 MPEG-2/4 AAC ADTS 转换为 MPEG-4 音频特定配置比特流。

此过滤器从 MPEG-2/4 ADTS 标头创建 MPEG-4 AudioSpecificConfig 并删除 ADTS 标头。

例如，当将 AAC 流从原始 ADTS AAC 或 MPEG-TS 容器复制到 MP4A-LATM、FLV 文件或 MOV/MP4 文件和相关格式（如 3GP 或 M4A）时，需要此过滤器。请注意，对于 MP4A-LATM 和 MOV/MP4 及相关格式，它是自动插入的。

### 18.2 av1\_元数据

修改嵌入在 AV1 流中的元数据。

达标

在流的所有时间单元中插入或删除时间定界符 OBU。

'插入'

在每个还没有 TU 的开头插入一个 TD。

'删除'

从每个有一个 TU 的开头删除 TD。

原色

传输特性

矩阵系数

设置流中的颜色描述字段（参见 AV1 第 6.4.2 节）。

颜色范围

设置流中的颜色范围（参见 AV1 第 6.4.2 节；请注意，不能为使用 BT.709 原色、sRGB 传输特性和身份 (RGB) 矩阵系数的流设置此范围）。

'电视'

范围有限。

'电脑'

全系列。

色度样本位置

设置流中的色度采样位置（参见 AV1 第 6.4.2 节）。这只能为 4:2:0 流设置。

'垂直'

左侧位置（匹配 MPEG-2 和 H.264 中的默认值）。

'并置'

左上位置。

tick\_rate

在序列头的计时信息中设置节拍率 ( time\_scale / num\_units\_in\_display\_tick )。

num\_ticks\_per\_picture

设置每张图片中的刻度数，以指示流具有固定的帧率。如果tick\_rate也没有设置则忽略。

删除填充

删除填充 OBU。

### 18.3 咬

删除数据包末尾的零填充。

### 18.4 dca\_core

从 DCA/DTS 流中提取核心，删除 DTS-HD 等扩展。

### 18.5 dump\_extra

将额外数据添加到过滤数据包的开头，除非所述数据包已经完全以要添加的额外数据开始。

频率

附加参数指定应过滤哪些数据包。它接受以下值：

' k '

'关键帧'

向所有密钥数据包添加额外数据

' e '

'全部'

向所有数据包添加额外数据

如果未指定，则假定为“ k ”。

例如，以下ffmpeg命令在libx264编码器生成的 H.264 数据包中强制使用全局标头（从而禁用单个数据包标头） ，但通过将存储在 extradata 中的标头添加到密钥数据包中来纠正它们：

ffmpeg -i INPUT -map 0 -flags:v +global\_header -c:v libx264 -bsf:v dump\_extra out.ts

### 18.6 dv\_error\_marker

DV 中标记为损坏的块被指定颜色的块替换。

颜色

替换损坏方块的颜色

斯塔

一个 16 位掩码，它指定 16 个可能的错误状态值中的哪些值将被彩色块替换。 0xFFFE 是替换所有非 0 错误状态值的默认值。

'好的'

没有错误，没有隐瞒

'错误'

错误，无隐瞒

'水库'

预订的

'诺托克'

错误或隐瞒

'巴黎'

未保留

' Aa, Ba, Ca, Ab, Bb, Cb, A, B, C, a, b, erri, erru '

具体错误状态码

请参阅第 44-46 页或[http://web.archive.org/web/20060927044735/http://www.smpte.org/smpte\_store/standards/pdf/s314m.pdf的第 5.5 节](http://web.archive.org/web/20060927044735/http://www.smpte.org/smpte_store/standards/pdf/s314m.pdf)

### 18.7 eac3\_core

从 E-AC-3 流中提取核心，丢弃额外的通道。

### 18.8 extract\_extradata

提取带内额外数据。

某些编解码器允许长期标头（例如 MPEG-2 序列标头或 H.264/HEVC (VPS/)SPS/PPS）在“带内”（即作为包含编码的比特流的一部分）传输帧）或“带外”（例如在容器级别）。后一种形式在 FFmpeg 术语中称为“额外数据”。

该比特流过滤器检测带内标头并将它们作为额外数据提供。

消除

启用此选项后，长期标头会在提取后从比特流中删除。

### 18.9 filter\_units

从流中删除给定集合中或不具有类型的单元。

pass\_types

移除所有其他单元类型或单元类型范围的列表。这被指定为以“|”分隔的单位类型值列表或带有“-”的值范围。

移除类型

pass\_types相同，除了给定集合中的单位被删除，所有其他单位都通过。

此转换未更改额外数据，但请注意，如果流包含内联参数集，则如果删除它们，输出可能无法使用。

例如，要从 H.264 流中删除所有非 VCL NAL 单元：

ffmpeg -i 输入 -c:v 复制 -bsf:v 'filter\_units=pass\_types=1-5' 输出

要从 H.265 流中删除所有 AUD、SEI 和填充符：

ffmpeg -i 输入 -c:v 复制 -bsf:v 'filter\_units=remove\_types=35|38-40' 输出

### 18.10 hapqa\_extract

提取 HAPQA 文件的 Rgb 或 Alpha 部分，无需重新压缩，以创建 HAPQ 或 HAPAlphaOnly 文件。

质地

指定要保留的纹理。

颜色

α

将 HAPQA 转换为 HAPQ

ffmpeg -i hapqa\_inputfile.mov -c copy -bsf:v hapqa\_extract=texture=color -tag:v HapY -metadata:s:v:0 编码器="HAPQ" hapq\_file.mov

将 HAPQA 转换为 HAPAlphaOnly

ffmpeg -i hapqa\_inputfile.mov -c copy -bsf:v hapqa\_extract=texture=alpha -tag:v HapA -metadata:s:v:0 encoder="仅限 HAPAlpha" hapalphaonly\_file.mov

### 18.11 h264\_元数据

修改嵌入在 H.264 流中的元数据。

音频

在流的所有访问单元中插入或删除 AUD NAL 单元。

'通过'

'插入'

'删除'

默认为通过。

样本纵横比

在 VUI 参数中设置流的样本纵横比。请参见 H.264 表 E-1。

overscan\_appropriate\_flag

设置流是否适合使用过扫描显示（参见 H.264 E.2.1 节）。

视频格式

video\_full\_range\_flag

设置流中的视频格式（参见 H.264 E.2.1 部分和表 E-2）。

colour\_primaries

传输特性

矩阵系数

设置流中的颜色描述（参见 H.264 E.2.1 节和表 E-3、E-4 和 E-5）。

chroma\_sample\_loc\_type

设置流中的色度样本位置（参见 H.264 E.2.1 节和图 E-1）。

tick\_rate

在 VUI 参数中设置刻度率 (time\_scale / num\_units\_in\_tick)。这是流中可表示的最小时间单位，在许多情况下表示流的场速率（帧速率的两倍）。

fixed\_frame\_rate\_flag

设置流是否具有固定帧率 - 通常这表示帧率恰好是滴答率的一半，但确切含义取决于隔行扫描和图片结构（请参阅 H.264 部分 E.2.1 和表 E-6）。

zero\_new\_constraint\_set\_flags

SPS 中的 constraint\_set4\_flag 和 constraint\_set5\_flag 为零。这些位在 H.264 规范的早期版本中保留，因此一些硬件解码器要求这些位为零。将其归零的结果仍然是有效的比特流。

crop\_left

crop\_right

作物顶部

作物底部

在 SPS 中设置帧裁剪偏移量。如果流已被裁剪，这些值将替换当前值。

这些字段以像素为单位设置。请注意，如果对色度进行二次采样或对流进行隔行扫描，则某些尺寸可能无法表示（请参阅 H.264 第 7.4.2.1.1 节）。

sei\_user\_data

插入一个字符串作为 SEI 未注册用户数据。参数必须是UUID+string 的形式，其中 UUID 是可能由连字符分隔的十六进制数字，字符串可以是任何内容。

例如，“ 086f3693-b7b3-4f2c-9653-21492feee5b8+hello ”将插入与给定 UUID 关联的字符串“hello”。

删除\_填充

删除填充 NAL 单元和填充 SEI 消息。

显示方向

插入、提取或删除显示方向 SEI 消息。参见 H.264 的 D.1.27 和 D.2.27 节的句法和语义。

'通过'

'插入'

'删除'

'提取物'

默认为通过。

旋转和翻转选项结合使用。任何预先存在的显示方向消息都将在插入或删除模式下删除。提取模式将显示矩阵作为辅助数据附加到数据包。

旋转

设置显示方向 SEI 的旋转（以度为单位的逆时针角度）。范围是 -360 到 +360。默认值为 NaN。

翻动

在显示方向 SEI 中设置翻转。

'水平'

'垂直'

默认未设置。

等级

在 SPS 中设置级别。请参阅 H.264 A.3 节和表 A-1 至 A-5。

参数必须是级别名称（例如“ 4.2 ”）、level\_idc 值（例如“ 42 ”）或特殊名称“ auto ”，表示过滤器应尝试从输入中猜测级别流属性。

### 18.12 h264\_mp4toannexb

将 H.264 比特流从长度前缀模式转换为起始码前缀模式（如 ITU-T H.264 规范附件 B 中所定义）。

这是某些流格式所必需的，通常是 MPEG-2 传输流格式 (muxer mpegts )。

例如，要使用ffmpeg将包含 H.264 流的 MP4 文件重新混合为 mpegts 格式，您可以使用以下命令：

ffmpeg -i INPUT.mp4 -codec copy -bsf:v h264\_mp4toannexb OUTPUT.ts

mpegts ) 和原始 H.264 (muxer h264 ) 输出格式自动插入的。

### 18.13 h264\_redundant\_pps

这对某些包含冗余 PPS 的蓝光流应用了特定的修正，这些 PPS 修改了流的不相关参数，这混淆了需要正确额外数据的其他转换。

### 18.14 hevc\_metadata

修改嵌入在 HEVC 流中的元数据。

音频

在流的所有访问单元中插入或删除 AUD NAL 单元。

'插入'

'删除'

样本纵横比

在 VUI 参数中设置流中的样本纵横比。

视频格式

video\_full\_range\_flag

设置流中的视频格式（参见 H.265 E.3.1 节和表 E.2）。

colour\_primaries

传输特性

矩阵系数

设置流中的颜色描述（参见 H.265 E.3.1 节和表 E.3、E.4 和 E.5）。

chroma\_sample\_loc\_type

设置流中的色度采样位置（参见 H.265 E.3.1 节和图 E.1）。

tick\_rate

在 VPS 和 VUI 参数 (time\_scale / num\_units\_in\_tick) 中设置滴答率。结合num\_ticks\_poc\_diff\_one ，这可以在流中设置恒定的帧率。请注意，当流位于容器中时，它可能会被容器参数覆盖。

num\_ticks\_poc\_diff\_one

在 VPS 和 VUI 中设置 poc\_proportional\_to\_timing\_flag 并使用该值设置 num\_ticks\_poc\_diff\_one\_minus1（参见 H.265 sections 7.4.3.1 和 E.3.1）。如果tick\_rate也没有设置则忽略。

crop\_left

crop\_right

作物顶部

作物底部

在 SPS 中设置一致性窗口裁剪偏移量。如果流已被裁剪，这些值将替换当前值。

这些字段以像素为单位设置。请注意，如果对色度进行二次采样（H.265 第 7.4.3.2.1 节），则某些尺寸可能无法表示。

等级

在 VPS 和 SPS 中设置级别。参见 H.265 A.4 节和表 A.6 和 A.7。

参数必须是级别名称（例如“ 5.1 ”）、 general\_level\_idc值（例如“ 153 ”表示级别 5.1）或特殊名称“ auto ”，表示过滤器应尝试猜测级别来自输入流属性。

### 18.15 hevc\_mp4toannexb

将 HEVC/H.265 比特流从长度前缀模式转换为起始代码前缀模式（如 ITU-T H.265 规范附件 B 中所定义）。

这是某些流格式所必需的，通常是 MPEG-2 传输流格式 (muxer mpegts )。

例如，要使用ffmpeg将包含 HEVC 流的 MP4 文件重新混合为 mpegts 格式，您可以使用以下命令：

ffmpeg -i INPUT.mp4 -codec copy -bsf:v hevc\_mp4toannexb OUTPUT.ts

请注意，此过滤器是针对 MPEG-TS（多路复用器mpegts ）和原始 HEVC/H.265（多路复用器h265或hevc ）输出格式自动插入的。

### 18.16 imxdump

修改比特流以适合 MOV 并可供 Final Cut Pro 解码器使用。此过滤器仅适用于 mpeg2video 编解码器，Final Cut Pro 7 和具有适当-tag:v的更新版本可能不需要。

例如，要将 30 MB/秒的 NTSC IMX 重新混合到 MOV：

ffmpeg -i input.mxf -c copy -bsf:v imxdump -tag:v mx3n 输出.mov

### 18.17 mjpeg2jpeg

将 MJPEG/AVI1 数据包转换为完整的 JPEG/JFIF 数据包。

MJPEG 是一种视频编解码器，其中每个视频帧本质上都是一个 JPEG 图像。可以无损地提取各个帧，例如通过

ffmpeg -i ../some\_mjpeg.avi -c:v 复制帧\_%d.jpg

不幸的是，这些块是不完整的 JPEG 图像，因为它们缺少解码所需的 DHT 段。引自<http://www.digitalpreservation.gov/formats/fdd/fdd000063.shtml>：

Avery Lee 于 2001 年在 rec.video.desktop 新闻组中写道，评论说“MJPEG，或者至少是具有 MJPG fourcc 的 AVI 中的 MJPEG，是受限的 JPEG，具有固定的 – 并且\*省略\* – 哈夫曼表。JPEG 必须是 YCbCr 色彩空间，它必须是 4:2:2，而且它必须使用基本的 Huffman 编码，而不是算术或渐进。。。。你确实可以提取 MJPEG 帧并用常规的 JPEG 解码器解码它们，但是你必须在前面加上DHT 段给他们，否则解码器将不知道如何解压缩数据。OpenDML 规范中给出了所需的确切表格。

此位流过滤器修补从 MJPEG 流中提取的帧的标头（携带 AVI1 标头 ID 且缺少 DHT 段）以生成完全合格的 JPEG 图像。

ffmpeg -i mjpeg-movie.avi -c:v 复制 -bsf:v mjpeg2jpeg frame\_%d.jpg

exiftran -i -9 帧\*.jpg

ffmpeg -i frame\_%d.jpg -c:v 复制旋转.avi

### 18.18 mjpegadump

将 MJPEG A 标头添加到比特流，以启用 Quicktime 解码。

### 18.19 mov2textsub

从 MOV 字幕中提取可表示的文本文件，从每个字幕包中剥离元数据标头。

另请参阅[text2movsub](#text2movsub)过滤器。

### 18.20 mp3 分解

解压缩非标准压缩 MP3 音频标头。

### 18.21 mpeg2\_元数据

修改嵌入在 MPEG-2 流中的元数据。

显示纵横比

设置流中的显示纵横比。

支持以下固定值：

4/3

16/9

221/100

任何其他值将导致发出信号的是方形像素（参见 H.262 第 6.3.3 节和表 6-3）。

帧率

设置流中的帧速率。这是根据已知值表与一个小的乘数和除数组合而成的——如果提供的值不能精确表示，将使用最接近的可表示值（参见 H.262 第 6.3.3 节和表 6-4）。

视频格式

设置码流中的视频格式（见H.262 6.3.6节和表6-6）。

colour\_primaries

传输特性

矩阵系数

设置流中的颜色描述（参见H.262 section 6.3.6和表6-7、6-8和6-9）。

### 18.22 mpeg4\_unpack\_bframes

解压 DivX 风格打包的 B 帧。

DivX 样式打包的 B 帧不是有效的 MPEG-4，并且只是损坏的 Video for Windows 子系统的解决方法。它们占用更多空间，可能导致轻微的 AV 同步问题，需要更多 CPU 能力来解码（除非播放器有一些解码图片队列来补偿每个数据包样式的 2,0,2,0 帧）并且如果复制到标准中会造成麻烦像 mp4 或 mpeg-ps/ts 这样的容器，因为 MPEG-4 解码器可能无法解码它们，因为它们不是有效的 MPEG-4。

例如，要使用ffmpeg修复包含带有 DivX 样式打包 B 帧的 MPEG-4 流的 AVI 文件，您可以使用以下命令：

ffmpeg -i INPUT.avi -codec copy -bsf:v mpeg4\_unpack\_bframes 输出.avi

### 18.23噪音

损坏包装内的物品或直接将其掉落而不损坏容器。可用于模糊测试或测试错误弹性/隐藏。

参数：

数量

接受一个表达式，其对每个数据包的评估决定了该数据包中的字节将被修改的频率。低于 0 的值将导致可变频率。默认值为 0，即不进行任何修改。但是，如果既未指定 amount 也未指定 drop，则 amount 将设置为-1 。请参阅下文了解可接受的变量。

降低

接受按数据包评估的表达式，其值确定是否丢弃该数据包。评估为正值会导致数据包被丢弃。对负值的评估导致它被丢弃的可变机会，大致与值的大小成反比。默认值为 0，即不会掉落。请参阅下文了解可接受的变量。

掉落量

接受一个非负整数，它分配一个可变的机会，它被删除，大致与值成反比。默认值为 0，即不会掉落。保留此选项是为了向后兼容，相当于将drop 设置为具有相同幅度的负值，即dropamount=4与drop=-4相同。如果还指定了 drop，则忽略。

amount和drop都接受包含以下变量的表达式：

' n '

数据包的索引，从零开始。

'结核病'

数据包时间戳的时基。

'积分'

数据包表示时间戳。

' dts '

数据包解码时间戳。

' nopts '

表示 AV\_NOPTS\_VALUE 的常量。

'开始'

在流中看到的第一个非 AV\_NOPTS\_VALUE PTS。

'开始'

在流中看到的第一个非 AV\_NOPTS\_VALUE DTS。

'持续时间'

' d '

数据包持续时间，以时基为单位。

'位置'

输入中的数据包位置；未知或未设置时可能为 -1。

'尺寸'

数据包大小，以字节为单位。

'钥匙'

数据包是否被标记为关键帧。

'状态'

一个伪随机整数，主要来自数据包有效负载的内容。

* [例子](#Examples-6)

#### 18.23.1 例子

对每个字节应用修改，但不要丢弃任何数据包。

ffmpeg -i INPUT -c copy -bsf noise=1 输出.mkv

在时间戳 30 秒后丢弃每个未标记为关键帧的视频数据包，但不要修改任何剩余的数据包。

ffmpeg -i INPUT -c copy -bsf:v noise=drop='gt(t\,30)\*not(key)' 输出.mkv

每 10 秒减少一秒的音频，并在其余部分添加一些随机噪音。

ffmpeg -i INPUT -c copy -bsf:a noise=amount=-1:drop='between(mod(t\,10)\,9\,10)' 输出.mkv

### 18.24 空

这个比特流过滤器通过不改变的数据包。

### 18.25 pcm\_rechunk

将 PCM 音频重新打包为每个数据包的固定样本数或每秒固定的数据包速率。这类似于[(ffmpeg-filters)asetnsamples 音频过滤器](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-filters.html#asetnsamples)，但适用于音频数据包而不是音频帧。

nb\_out\_samples, n

设置每个输出音频数据包的样本数。该数字旨在作为每个通道的样本数。默认值为 1024。

垫，p

与之前的样本相同数量的样本（或大致相同数量的样本，请参阅frame\_rate ）。默认值为 1。

帧率，r

此选项使过滤器每秒输出固定数量的数据包，而不是每个数据包的固定数量的样本。如果音频采样率不能被帧率整除，那么样本数将不是常数，而是会略有变化，以便每个数据包都尽可能靠近帧边界开始。使用此选项优先于nb\_out\_samples 。

frame\_rate选项为 NTSC 帧速率生成众所周知的 1602-1601-1602-1601-1602 模式的 48kHz 音频。

ffmpeg -f lavfi -i sine=r=48000:d=1 -c pcm\_s16le -bsf pcm\_rechunk=r=30000/1001 -f framecrc -

### 18.26 pgs\_frame\_merge

将以“显示集结束”段结尾的一系列 PGS 字幕段合并到一个数据包中。

这是某些支持 PGS 字幕的容器所必需的（muxer matroska ）。

### 18.27 prores\_metadata

修改嵌入在 prores 流中的颜色属性元数据。

原色

设置颜色原色。可用值是：

'汽车'

保持相同的颜色原色属性（默认）。

'未知'

' bt709 '

' bt470bg '

BT601 625

' smpte170m '

BT601 525

' bt2020 '

' smpte431 '

DCI P3

' smpte432 '

P3 D65

传输特性

设置颜色转移。可用值是：

'汽车'

保持相同的传输特性属性（默认）。

'未知'

' bt709 '

英国电信 601、英国电信 709、英国电信 2020

' smpte2084 '

SMPTE ST 2084 格式

' arib-std-b67 '

ARIB标准-B67

矩阵系数

设置矩阵系数。可用值是：

'汽车'

保持相同的色彩空间属性（默认）。

'未知'

' bt709 '

' smpte170m '

蓝牙601

' bt2020nc '

为文件的每一帧设置 Rec709 色彩空间

ffmpeg -i INPUT -c copy -bsf:v prores\_metadata=color\_primaries=bt709:color\_trc=bt709:colorspace=bt709 输出.mov

为文件的每一帧设置 Hybrid Log-Gamma 参数

ffmpeg -i INPUT -c copy -bsf:v prores\_metadata=color\_primaries=bt2020:color\_trc=arib-std-b67:colorspace=bt2020nc 输出.mov

### 18.28 remove\_extra

从数据包中删除额外数据。

它接受以下参数：

频率

设置要从中删除额外数据的帧类型。

' k '

仅从非关键帧中删除额外数据。

'关键帧'

仅从关键帧中删除额外数据。

' e，所有'

从所有帧中删除额外数据。

### 18.29 套

在数据包中设置 PTS 和 DTS。

它接受以下参数：

TS

积分

数据传输系统

为 PTS、DTS 或两者设置表达式。

期间

设置持续时间的表达式。

时基

设置输出时基。

表达式通过 eval API 求值，可以包含以下常量：

否

输入数据包的计数。从0开始。

TS

ts或dts选项的情况下输入中的多路分解时间戳或在pts选项的情况下显示时间戳。

收银机

数据包文件中的原始位置，如果当前数据包未定义，则为未定义

数据传输系统

输入中的多路分解时间戳。

公关部

输入中的呈现时间戳。

期间

输入的持续时间。

启动DTS

第一个数据包的 DTS。

开始点

第一个数据包的 PTS。

PREV\_INDTS

上一个输入 DTS。

PREV\_INPTS

上一个输入PTS。

PREV\_INDURATION

上一个输入持续时间。

PREV\_OUTDTS

先前的输出 DTS。

PREV\_OUTPTS

之前的输出PTS。

PREV\_OUTDURATION

上一个输出持续时间。

下一个DTS

接下来输入DTS。

下一个PTS

接下来输入PTS。

NEXT\_DURATION

下一个输入持续时间。

结核病

流包所属的时基。

TB\_OUT

输出时基。

SR

流包所属的采样率。

NOPTS

AV\_NOPTS\_VALUE 常量。

### 18.30 text2movsub

将文本字幕转换为带有元数据标头的 MOV 字幕（由mov\_text编解码器使用）。

另请参阅[mov2textsub](#mov2textsub)过滤器。

### 18.31 trace\_headers

日志跟踪输出包含编码流标头中的所有语法元素（所有高于单个编码块的级别）。这对于调试低级流问题很有用。

支持 AV1、H.264、H.265、(M)JPEG、MPEG-2 和 VP9，但根据版本的不同，可能只有其中的一部分可用。

### 18.32 truehd\_core

从 TrueHD 流中提取核心，丢弃 ATMOS 数据。

### 18.33 vp9\_元数据

修改 VP9 流中嵌入的元数据。

色彩空间

在帧头中设置颜色空间值。请注意，任何设置为 RGB 的帧都将隐式设置为 PC 范围，并且 RGB 与配置文件 0 和 2 不兼容。

'未知'

' bt601 '

' bt709 '

' smpte170 '

' smpte240 '

' bt2020 '

' RGB '

颜色范围

在帧头中设置颜色范围值。请注意，颜色空间强加的任何值都将优先于此值。

'电视'

'电脑'

### 18.34 vp9\_superframe

将 VP9 不可见 (alt-ref) 帧合并回 VP9 超帧。这修复了拆分/分段 VP9 流的合并，其中 alt-ref 帧与其可见对应帧分离。

### 18.35 vp9\_superframe\_split

将 VP9 超帧拆分为单个帧。

### 18.36 vp9\_raw\_reorder

给定具有正确时间戳但可能乱序的 VP9 流，插入额外的 show-existing-frame 数据包以更正顺序。

## 19 格式选项

libavformat 库提供了一些通用的全局选项，可以在所有的 muxers 和 demuxers 上设置。此外，每个 muxer 或 demuxer 可能支持所谓的私有选项，这些选项特定于该组件。

-option来设置选项 FFmpeg 工具中的值，或通过在AVFormatContext选项中显式设置值或使用libavutil/opt.h API 进行编程使用。

支持的选项列表如下：

avioflags标志（输入/输出）

可能的值：

'直接'

减少缓冲。

探测大小整数（输入）

以字节为单位设置探测大小，即要分析以获取流信息的数据的大小。较高的值将能够检测到更多信息，以防它分散到流中，但会增加延迟。必须是不小于32的整数，默认为5000000。

max\_probe\_packets整数（输入）

设置探测编解码器时缓冲数据包的最大数量。默认为 2500 个数据包。

packetsize整数（输出）

设置数据包大小。

fflags标志

设置格式标志。有些是为有限数量的格式实现的。

输入文件的可能值：

'丢弃腐败'

丢弃损坏的数据包。

'快速搜索'

为某些格式启用快速但不准确的搜索。

'绅士'

如果存在 DTS，则生成缺失的 PTS。

'点燃'

如果设置了 PTS，则忽略 DTS。设置 nofillin 时呈惰性。

'点燃'

忽略索引。

'无缓冲'

减少在初始输入流分析期间缓冲引入的延迟。

' nofillin '

可以精确计算的报文字段不要填写缺失值。

'不解析'

禁用 AVParsers，这也需要+nofillin 。

'排序'

尝试通过 DTS 交错输出数据包。目前，仅适用于带索引的 AVI。

输出文件的可能值：

' autobsf '

根据输出格式的要求自动应用比特流过滤器。默认启用。

'双精确'

仅写入平台、构建和时间无关的数据。这可确保文件和数据校验和可重现并在平台之间匹配。它的主要用途是回归测试。

' flush\_packets '

立即写出数据包。

'最短'

在最短流的末尾停止复用。可能需要增加 max\_interleave\_delta 以避免在 EOF 之前刷新较长的流。

seek2any整数（输入）

如果设置为 1，则在支持时允许在解复用器级别上搜索非关键帧。默认值为 0。

analyzeduration整数（输入）

指定分析多少微秒以探测输入。较高的值将能够检测到更准确的信息，但会增加延迟。它默认为 5,000,000 微秒 = 5 秒。

加密密钥十六进制字符串（输入）

设置解密密钥。

indexmem整数（输入）

设置用于时间戳索引的最大内存（每个流）。

rtbufsize整数（输入）

设置用于缓冲实时帧的最大内存。

fdebug标志（输入/输出）

打印特定的调试信息。

可能的值：

' ' \_

max\_delay整数（输入/输出）

以微秒为单位设置最大多路复用或多路分解延迟。

fpsprobesize整数（输入）

设置用于探测 fps 的帧数。

audio\_preload整数（输出）

设置音频数据包应提前交错的微秒数。

chunk\_duration整数（输出）

为每个块设置微秒。

chunk\_size整数（输出）

为每个块设置大小（以字节为单位）。

err\_detect, f\_err\_detect标志（输入）

设置错误检测标志。 f\_err\_detect已弃用，只能通过ffmpeg工具使用。

可能的值：

' crccheck '

验证嵌入式 CRC。

'比特流'

检测比特流规范偏差。

'缓冲区'

检测不正确的比特流长度。

'爆炸'

检测到小错误时中止解码。

'小心'

将违反规范且在野外未被视为错误的事情考虑在内。

'合规'

将所有不符合规范的行为视为错误。

'咄咄逼人'

将理智的编码器不应做的事情视为错误。

max\_interleave\_delta整数（输出）

设置交织的最大缓冲持续时间。持续时间以微秒表示，默认为 10000000（10 秒）。

为确保所有流都正确交错，libavformat 将等到每个流至少有一个数据包，然后再实际将任何数据包写入输出文件。当某些流“稀疏”时（即连续数据包之间存在较大间隙），这可能会导致过度缓冲。

此字段指定 muxing 队列中第一个和最后一个数据包的时间戳之间的最大差异，libavformat 将输出一个数据包，无论它是否已为所有流排队数据包。

如果设置为 0，libavformat 将继续缓冲数据包，直到每个流都有一个数据包，而不管缓冲数据包之间的最大时间戳差异如何。

use\_wallclock\_as\_timestamps整数（输入）

如果设置为 1，则使用挂钟作为时间戳。默认值为 0。

avoid\_negative\_ts整数（输出）

可能的值：

' make\_non\_negative '

移动时间戳，使它们成为非负数。另请注意，这仅影响前导负时间戳，而不影响非单调负时间戳。

' make\_zero '

移动时间戳，使第一个时间戳为 0。

'自动（默认） '

在目标格式需要时启用移位。

'禁用'

禁用时间戳的移动。

启用移位后，所有输出时间戳都会移位相同的量。与没有移动的情况相比，音频、视频和字幕的不同步和相对时间戳差异得以保留。

skip\_initial\_bytes整数（输入）

如果设置为 1，则设置在读取标头和帧之前要跳过的字节数。默认值为 0。

correct\_ts\_overflow整数（输入）

如果设置为 1，则更正单个时间戳溢出。默认值为 1。

flush\_packets整数（输出）

在每个数据包之后刷新底层 I/O 流。默认为-1（auto），表示由底层协议决定，1启用，有降低延迟的效果，0禁用，在某些情况下可能增加IO吞吐量。

output\_ts\_offset偏移量（输出）

设置输出时间偏移量。

offset必须是持续时间规范，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)。

偏移量由复用器添加到输出时间戳。

指定正偏移量意味着相应的流被延迟 bt 中指定的持续时间offset 。默认值为0 （表示不应用偏移量）。

format\_whitelist列表（输入）

"," 分隔的允许分路器列表。默认情况下所有都被允许。

dump\_separator字符串（输入）

Separator 用于分隔打印在命令行上的关于 Stream 参数的字段。例如，要用换行符和缩进分隔字段：

ffprobe -dump\_separator“

“-i ~/videos/matrixbench\_mpeg2.mpg

max\_streams整数（输入）

指定最大流数。这可用于拒绝由于大量流而需要太多资源的文件。

skip\_estimate\_duration\_from\_pts布尔（输入）

使用 PTS 计算时跳过输入持续时间的估计。目前适用于MPEG-PS和MPEG-TS。

严格，f\_strict整数（输入/输出）

指定如何严格遵守标准。 f\_strict已弃用，只能通过ffmpeg工具使用。

可能的值：

'非常'

严格遵守规范或参考软件的更早更严格的版本

'严格'

无论后果如何，都严格遵守规范中的所有内容

'正常'

'非官方'

允许非官方扩展

'实验'

允许非标准化的实验性事物，实验性（未完成/进行中/未经过充分测试）解码器和编码器。注意：实验解码器可能会带来安全风险，请勿将其用于解码不受信任的输入。

* [格式流说明符](#Format-stream-specifiers-1)

### 19.1 格式流说明符

格式流说明符允许选择一个或多个匹配特定属性的流。

libavformat/avformat.h标头中声明的avformat\_match\_stream\_specifier()函数定义，并记录在[ffmpeg(1) 手册的 (ffmpeg)Stream specifiers 部分中](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg.html#Stream-specifiers)。

## 20 个解复用器

分路器是 FFmpeg 中的配置元素，可以从特定类型的文件中读取多媒体流。

当您配置 FFmpeg 构建时，默认情况下会启用所有受支持的分路器。您可以使用配置选项--list-demuxers列出所有可用的。

您可以使用配置选项--disable-demuxers禁用所有解复用器，并使用选项--enable-demuxer= DEMUXER选择性地启用单个解复用器，或使用选项--disable-demuxer= DEMUXER禁用它。

选项-demuxers将显示启用的 demuxers 列表。使用-formats查看已启用的多路分解器和多路复用器的组合列表。

下面是一些当前可用的多路分解器的描述。

* [一个](#aa)
* [aac](#aac-1)
* [apng](#apng)
* [asf](#asf-1)
* [连接](#concat-1)
* [短跑](#dash-1)
* [一个](#ea)
* [国际货币基金组织](#imf)
* [flv, live\_flv, kux](#flv_002c-live_005fflv_002c-kux)
* [动图](#gif-1)
* [hls](#hls-1)
* [图片2](#image2-1)
* [libgme](#libgme)
* [libmodplug](#libmodplug)
* [libopenmpt](#libopenmpt)
* [电影/mp4/3gp](#mov_002fmp4_002f3gp)
* [mpegts](#mpegts)
* [mpjpeg格式](#mpjpeg)
* [原始视频](#rawvideo-1)
* [sbg](#sbg)
* [字幕](#tedcaptions)
* [蒸汽合成器](#vapoursynth)

### 20.1AA

可听格式 2、3 和 4 解复用器。

此解复用器用于解复用可听格式 2、3 和 4 (.aa) 文件。

### 20.2 音频

原始音频数据传输流 AAC 解复用器。

此解复用器用于解复用包含单个 AAC 流以及其中的任何 ID3v1/2 或 APE 标记的 ADTS 输入。

### 20.3 磅

动画便携式网络图形分路器。

这个解复用器用于解复用 APNG 文件。所有标头，但 PNG 签名，直到（但不包括）第一个 fcTL 块都作为额外数据传输。然后将帧拆分为两个 fcTL 块之间或最后一个 fcTL 和 IEND 块之间的所有块。

-ignore\_loop布尔

如果已设置，则忽略文件中的循环变量。默认启用。

-max\_fps整数

以每秒帧数表示的最大帧率。默认值 0 不施加任何限制。

-default\_fps int

当文件中没有指定时，以每秒帧数为单位的默认帧率（0 表示尽可能快）。默认值为 15。

### 20.4 自动对焦

高级系统格式多路分解器。

该解复用器用于解复用 ASF 文件和 MMS 网络流。

-no\_resync\_search布尔

不要试图通过寻找某个可选的起始码来重新同步。

### 20.5 连续

虚拟级联脚本解复用器。

这个解复用器从一个文本文件中读取文件列表和其他指令，然后一个接一个地解复用它们，就好像它们的所有数据包都被复用在一起一样。

调整文件中的时间戳，以便第一个文件从 0 开始，每个下一个文件从前一个文件结束的地方开始。请注意，它是全局完成的，如果所有流的长度不完全相同，则可能会导致间隙。

所有文件必须具有相同的流（相同的编解码器、相同的时基等）。

每个文件的持续时间用于调整下一个文件的时间戳：如果持续时间不正确（例如，因为它是使用比特率计算的或因为文件被截断），它可能会导致伪影。 duration指令可用于覆盖存储在每个文件中的持续时间。

* [句法](#Syntax-1)
* [选项](#Options-44)
* [例子](#Examples-7)

#### 20.5.1 语法

该脚本是一个扩展 ASCII 格式的文本文件，每行一个指令。空行、前导空格和以“#”开头的行将被忽略。识别以下指令：

文件路径

要读取的文件的路径；特殊字符和空格必须用反斜杠或单引号转义。

所有后续的文件相关指令都适用于该文件。

ffconcat 版本 1.0

确定脚本类型和版本。

为了让 FFmpeg 自动识别格式，该指令必须在脚本的第一行完全按原样出现（没有额外的空格或字节顺序标记）。

持续时间

文件的持续时间。可以从文件中指定此信息；如果文件中的信息不可用或不准确，在此处指定它可能会更有效或有帮助。

如果为所有文件设置了持续时间，则可以在整个串联视频中搜索。

入点时间戳

在文件中。当分路器打开文件时，它会立即寻找指定的时间戳。完成搜索，以便所有流都可以在 In 点成功呈现。

该指令最适用于帧内编解码器，因为对于非帧内编解码器，您通常会在实际入点之前获得额外的数据包，并且解码内容很可能也包含入点之前的帧。

对于每个文件，文件 In 点之前的数据包的时间戳将小于文件的计算开始时间戳（第一个文件的情况下为负），并且文件的持续时间（如果未由持续时间指令指定）将减少基于他们指定的入点。

由于指定 In 点之前的潜在数据包，数据包时间戳可能会在两个连接的文件之间重叠。

出点时间戳

文件的出点。当解复用器在任何流中达到指定的解码时间戳时，它会将其作为文件结束条件进行处理，并跳过所有流中的当前数据包和所有剩余数据包。

Out point是exclusive的，就是说demuxer不会输出解码时间戳大于等于Out point的包。

该指令最适合所有流紧密交错的帧内编解码器和格式。对于非帧内编解码器，您通常会在出点之后获得带有显示时间戳的额外数据包，因此解码内容很可能也包含出点之后的帧。如果您的流没有紧密交错，您可能无法在出点之前从所有流中获取所有数据包，并且您可能只能解码最早的流直到出点。

文件的持续时间（如果持续时间指令未指定）将根据其指定的出点减少。

file\_packet\_metadata键=值

文件包的元数据。将为每个文件包设置指定的元数据。您可以多次指定此指令以添加多个元数据条目。该指令已弃用，请改用file\_packet\_meta 。

file\_packet\_meta密钥 价值

文件包的元数据。将为每个文件包设置指定的元数据。您可以多次指定此指令以添加多个元数据条目。

选项键 价值

访问、打开和探测文件的选项。可以出现多次。

溪流

在虚拟文件中引入一个流。所有后续的流相关指令都适用于最后引入的流。必须设置一些流属性以允许识别子文件中的匹配流。如果脚本中没有定义流，则复制第一个文件中的流。

exact\_stream\_id id

设置流的 ID。如果给出此指令，将使用子文件中具有相应 id 的字符串。这对于流顺序不可靠的 MPEG-PS (VOB) 文件特别有用。

stream\_meta键 价值

流的元数据。可以出现多次。

stream\_codec值

流的编解码器。

stream\_extradata十六进制字符串

字符串的额外数据，以十六进制编码。

章节编号 开始 结尾

加一章。 id是唯一标识符，可能很小且连续。

#### 20.5.2 选项

此分路器接受以下选项：

安全的

如果设置为 1，则拒绝不安全的文件路径和指令。如果文件路径不包含协议规范并且是相对的并且所有组件仅包含可移植字符集中的字符（字母、数字、句点、下划线和连字符）并且在组件的开头没有句点，则该文件路径被认为是安全的。

如果设置为 0，则接受任何文件名。

默认值为 1。

自动转换

如果设置为 1，尝试对数据包数据执行自动转换以使流可连接。默认值为 1。

目前，唯一的转换是将 h264\_mp4toannexb 比特流过滤器添加到 MP4 格式的 H.264 流中。如果分辨率发生变化，这尤其必要。

段时间元数据

如果设置为 1，每个数据包将包含lavf.concat.start\_time和lavf.concat.duration数据包元数据值，它们是连接输出中各个文件段的开始时间和持续时间，以微秒表示。仅当基于 concat 文件已知时才设置持续时间元数据。默认值为 0。

#### 20.5.3 例子

* 使用绝对文件名并包含一些注释：
* # 我的第一个文件名
* 文件/mnt/share/file-1.wav
* # 我的第二个文件名包括空格
* 文件 '/mnt/share/file 2.wav'
* # 我的第三个文件名包括空格和单引号
* 文件 '/mnt/share/file 3'\''.wav'
* 允许输入格式自动探测，使用安全文件名并设置第一个文件的持续时间：
* ffconcat 版本 1.0
* 文件 file-1.wav
* 持续时间 20.0
* 文件子目录/file-2.wav

### 20.6 破折号

基于 HTTP 多路分解器的动态自适应流。

此多路分解器呈现清单中找到的所有 AVStreams。通过在 AVStreams 上设置丢弃标志，调用者可以决定实际接收哪些流。每个流将<Representation>中的id和bandwidth属性镜像为分别名为“id”和“variant\_bitrate”的元数据键。

* [选项](#Options-45)

#### 20.6.1 选项

此分路器接受以下选项：

cenc\_decryption\_key

16 字节的十六进制密钥，用于解密使用 ISO 通用加密（CENC/AES-128 CTR；ISO/IEC 23001-7）加密的文件。

### 20.7个

Electronic Arts 多媒体格式解复用器。

这种格式被各种 Electronic Arts 游戏使用。

* [选项](#Options-46)

#### 20.7.1 选项

merge\_alpha布尔

通常，VP6 alpha 通道（如果存在）作为辅助视频流返回，通过设置此选项，您可以使分路器返回一个视频流，其中除了普通视频之外还包含 alpha 通道。

### 20.8 国际货币基金组织

可互操作的主格式多路分解器。

此多路分解器呈现在 IMF 组合中找到的音频和视频流。

### 20.9 flv、live\_flv、kux

Adobe Flash 视频格式分离器。

此解复用器用于解复用 FLV 文件和 RTMP 网络流。在实时网络流的情况下，如果您强制格式化，您可以使用 live\_flv 选项而不是 flv 来避免时间戳不连续。 KUX是优酷平台上使用的flv变体。

ffmpeg -f flv -i 我的文件.flv ...

ffmpeg -f live\_flv -i rtmp://<any.server>/anything/key ....

-flv\_metadata布尔

根据 onMetaData 数组内容分配流。

-flv\_ignore\_prevtag布尔

忽略前一个标签值的大小。

-flv\_full\_metadata布尔

输出 onMetadata 的所有上下文。

### 20.10 动图

动画 GIF 分离器。

它接受以下选项：

最小延迟

以百分之一秒为单位设置帧之间的最小有效延迟。范围是 0 到 6000。默认值为 2。

max\_gif\_delay

以百分之一秒为单位设置帧之间的最大有效延迟。范围是 0 到 65535。默认值为 65535（将近 11 分钟），这是规范允许的最大值。

默认延迟

以百分之一秒为单位设置帧之间的默认延迟。范围是 0 到 6000。默认值为 10。

忽略循环

GIF 文件可以包含循环一定次数（或无限次）的信息。如果ignore\_loop设置为 1，那么来自输入的循环设置将被忽略并且不会发生循环。如果设置为 0，则循环将发生，并根据 GIF 循环次数。默认值为 1。

例如，使用覆盖过滤器，将一个无限循环的 GIF 放在另一个视频上：

ffmpeg -i input.mp4 -ignore\_loop 0 -i input.gif -filter\_complex overlay=shortest=1 out.mkv

请注意，在上面的示例中，叠加过滤器的最短选项用于以最短输入文件的长度结束输出视频，在本例中为 input.mp4，因为此示例中的 GIF 无限循环。

### 20.11 小时

HLS 解复用器

Apple HTTP Live Streaming 多路分解器。

此多路分解器呈现来自所有变体流的所有 AVStream。 id 字段设置为比特率变体索引号。通过在 AVStreams 上设置丢弃标志（通过在 ffplay 中按“a”或“v”），调用者可以决定实际接收哪些变体流。流所属变体的总比特率在名为“variant\_bitrate”的元数据键中可用。

它接受以下选项：

live\_start\_index

开始直播流的分段索引（负值从末尾开始）。

prefer\_x\_start

如果它在播放列表中而不是 live\_start\_index，则更喜欢使用#EXT-X-START。

允许的扩展

',' 分隔的允许 hls 访问的文件扩展名列表。

最大重载

尝试重新加载不充分列表的最大次数。默认值为 1000。

m3u8\_hold\_counters

没有新段刷新时加载 m3u8 的最大次数。默认值为 1000。

http\_persistent

使用持久 HTTP 连接。仅适用于 HTTP 流。默认启用。

http\_multiple

使用多个 HTTP 连接来下载 HTTP 段。默认为 HTTP/1.1 服务器启用。

http\_seekable

使用 HTTP 部分请求下载 HTTP 段。 0 = 禁用，1 = 启用，-1 = 自动，默认为自动。

段格式选项

使用由:分隔的键=值对列表为媒体段的多路分解器设置选项。

段最大重试

错误时重新加载段的最大次数，当不需要在网络错误时跳过段时很有用。默认值为 0。

### 20.12 图片 2

图像文件分离器。

这个多路分解器从一个模式指定的图像文件列表中读取。模式的语法和含义由选项pattern\_type指定。

该模式可能包含一个后缀，用于自动确定文件中包含的图像的格式。

对于序列中的所有文件，每个图像的大小、像素格式和格式必须相同。

此分路器接受以下选项：

帧率

设置视频流的帧速率。它默认为 25。

环形

如果设置为 1，则循环遍历输入。默认值为 0。

模式类型

选择用于解释提供的文件名的模式类型。

pattern\_type接受以下值之一。

没有任何

禁用模式匹配，因此视频将只包含指定的图像。如果您不想从多个图像创建序列并且您的文件名可能包含特殊模式字符，则应使用此选项。

顺序

选择一个序列模式类型，用于指定由序号索引的文件序列。

序列模式可能包含字符串“%d”或“%0 N d”，它指定表示序列号的字符在与模式匹配的每个文件名中的位置。如果使用“%d0 N d”形式，表示每个文件名中的数字的字符串是用 0 填充的， N是表示数字的用 0 填充的数字的总数。文字字符“%”可以在带有字符串“%%”的模式中指定。

如果序列模式包含“%d”或“%0 N d”，则该模式指定的文件列表的第一个文件名必须包含一个包含在start\_number和start\_number + start\_number\_range -1之间的数字，并且后面的所有数字都必须是顺序的。

例如，模式“img-%03d.bmp”将匹配格式为img-001.bmp 、 img-002.bmp 、...、 img-010.bmp等的文件名序列；模式“i%%m%%g-%d.jpg”将匹配形式为i%m%g-1.jpg 、 i%m%g-2.jpg 、...、 i 的一系列文件名%m%g-10.jpg等

请注意，模式不一定包含“%d”或“%0 N d”，例如要转换单个图像文件img.jpeg ，您可以使用以下命令：

ffmpeg -i img.jpeg img.png

球体

选择 glob 通配符模式类型。

该模式被解释为glob()模式。只有在使用通配支持编译 libavformat 时才可选择此选项。

glob\_sequence （已弃用，将被删除）

选择混合的 glob 通配符/序列模式。

%\*?[]{}中至少包含一个 glob 元字符，前面是未转义的“%”，则该模式被解释为glob()模式，否则它被解释为序列模式。

所有 glob 特殊字符%\*?[]{}必须以“%”为前缀。要转义文字“%”，您应使用“%%”。

例如，模式foo-%\*.jpeg将匹配所有以“foo-”为前缀并以“.jpeg”结尾的文件名，而foo-%?%?%?.jpeg将匹配所有以“foo-”为前缀的文件名"，后跟三个字符的序列，并以 ".jpeg" 结尾。

这种模式类型已被弃用，取而代之的是glob和sequence 。

默认值为glob\_sequence 。

像素格式

设置要读取的图像的像素格式。如果未指定，则从序列中的第一个图像文件中猜测像素格式。

起始编号

设置与图像文件模式匹配的文件的索引以开始读取。默认值为 0。

start\_number\_range

设置索引间隔范围以在查找序列中的第一个图像文件时检查，从start\_number开始。默认值为 5。

ts\_from\_file

如果设置为 1，会将帧时间戳设置为图像文件的修改时间。请注意，未提供时间戳的单调性：图像的顺序与没有此选项的顺序相同。默认值为 0。如果设置为 2，会将帧时间戳设置为以纳秒为精度的图像文件的修改时间。

视频大小

设置要读取的图像的视频大小。如果未指定，则从序列中的第一个图像文件中猜测视频大小。

导出路径元数据

如果设置为 1，将向输入中的元数据添加两个额外字段，使它们也可用于其他过滤器（有关示例，请参见drawtext过滤器）。默认值为 0。额外字段说明如下：

lavf.image2dec.source\_path

对应于正在读取的输入文件的完整路径。

lavf.image2dec.source\_basename

对应于正在读取的文件的名称。

* [例子](#Examples-8)

#### 20.12.1 例子

* 使用ffmpeg从文件序列img-001.jpeg ， img-002.jpeg ，...中的图像创建视频，假设输入帧速率为每秒 10 帧：
* ffmpeg -framerate 10 -i 'img-%03d.jpeg' 输出.mkv
* 如上所述，但首先从序列中索引为 100 的文件中读取：
* ffmpeg -framerate 10 -start\_number 100 -i 'img-%03d.jpeg' 输出.mkv
* 读取匹配“\*.png”全局模式的图像，即所有以“.png”后缀结尾的文件：
* ffmpeg -framerate 10 -pattern\_type glob -i "\*.png" out.mkv

### 20.13 libgme

Game Music Emu 库是视频游戏音乐文件模拟器的集合。

有关详细信息，请参阅[https://bitbucket.org/mpyne/game-music-emu/overview 。](https://bitbucket.org/mpyne/game-music-emu/overview)

它接受以下选项：

跟踪索引

设置要解复用的轨道的索引。解复用器只能导出一个轨道。曲目索引从 0 开始。默认是选择第一首曲目。曲目数作为曲目元数据条目导出。

采样率

设置导出轨道的采样率。范围是 1000 到 999999。默认值为 44100。

最大大小（字节）

分路器将整个文件缓冲到内存中。调整此值以设置最大缓冲区大小，这反过来又充当可读取文件大小的上限。默认值为 50 MiB。

### 20.14 libmodplug

基于 ModPlug 的模块解复用器

请参阅<https://github.com/Konstanty/libmodplug>

它将导出一个 2 通道 16 位 44.1 kHz 音频流。可选地，可以导出带有或不带有打印元数据的pal8 16 色视频流。

它接受以下选项：

降噪

应用一个简单的低通滤波器。可以是 1（开）或 0（关）。默认为 0。

混响深度

设置混响量。范围 0-100。默认为 0。

混响延迟

以 ms 为单位设置延迟，固定为 40-250 ms。默认为 0。

低音量

应用低音扩展又名 XBass 或 megabass。范围是 0（安静）到 100（大声）。默认为 0。

低音范围

设置截止频率即低音频率的上限。范围是 10-100 赫兹。默认为 0。

环绕深度

应用 Dolby Pro-Logic 环绕声效果。范围是 0（安静）到 100（沉重）。默认为 0。

环绕延迟

以毫秒为单位设置环绕延迟，固定为 5-40 毫秒。默认为 0。

最大尺寸

分路器将整个文件缓冲到内存中。调整此值以设置最大缓冲区大小，这反过来又充当可读取文件大小的上限。范围是 0 到 100 MiB。 0 删除缓冲区大小限制（不推荐）。默认值为 5 MiB。

video\_stream\_expr

使用 eval API 对生成的视频流分配颜色进行评估的字符串。可以使用的变量有x , y , w , h , t , speed , tempo , order , pattern和row 。

视频流

生成视频流。可以是 1（开）或 0（关）。默认为 0。

video\_stream\_w

在“字符”中设置视频帧宽度，其中一个字符表示 8 个像素。范围是 20-512。默认值为 30。

video\_stream\_h

在“字符”中设置视频帧高度，其中一个字符表示 8 个像素。范围是 20-512。默认值为 30。

video\_stream\_ptxt

在视频流上打印元数据。包括speed 、 tempo 、 order 、 pattern 、 row和ts （以毫秒为单位的时间）。可以是 1（开）或 0（关）。默认值为 1。

### 20.15 libopenmpt

基于 libopenmpt 的模块解复用器

有关详细信息，请参阅[https://lib.openmpt.org/libopenmpt/ 。](https://lib.openmpt.org/libopenmpt/)

子歌曲选项设置。

它接受以下选项：

子歌

设置子歌曲索引。这可以是“全部”、“自动”或子歌曲的索引。 Subsong 索引从 0 开始。默认值为“自动”。

默认值是让 libopenmpt 选择。

布局

设置通道布局。有效值为 1、2 和 4 通道布局。默认值为立体声。

采样率

设置 libopenmpt 输出的采样率。范围从 1000 到 INT\_MAX。默认值为 48000。

### 20.16 mov/mp4/3gp

Quicktime 文件格式和 ISO/IEC 基本媒体文件格式（ISO/IEC 14496-12 或 MPEG-4 Part 12、ISO/IEC 15444-12 或 JPEG 2000 Part 12）的解复用器。

注册扩展：mov、mp4、m4a、3gp、3g2、mj2、psp、m4b、ism、ismv、isma、f4v

* [选项](#Options-47)
* [有声 AAX](#Audible-AAX)

#### 20.16.1 选项

此分路器接受以下选项：

enable\_drefs

启用加载外部曲目，默认禁用。启用此功能理论上可以在某些用例中泄漏信息。

使用\_绝对路径

允许通过绝对路径加载外部轨道，默认情况下禁用。启用此功能会带来安全风险。只有在已知源是非恶意的情况下才应启用它。

seek\_streams\_individually

寻找时，分别识别每个流中的最近点，并从识别点解复用该流中的数据包。与从一开始就线性解复用相比，这可能会导致不同的数据包序列。默认为真。

忽略编辑列表

忽略任何编辑列表原子。默认情况下，解复用器修改流索引以反映编辑列表描述的时间线。默认为假。

高级编辑列表

修改流索引以反映编辑列表描述的时间线。 ignore\_editlist必须设置为 false 才能使该选项生效。如果ignore\_editlist和此选项都设置为 false，则只有流索引的开始被修改以反映编辑列表描述的初始停留时间或开始时间戳。默认为真。

忽略章节

不要解析章节。这包括 GoPro“HiLight”标签/时刻。请注意，只有在输入可搜索时才会解析章节。默认为假。

使用\_mfra\_for

对于可搜索的分段输入，如果存在，则从媒体片段随机访问框设置片段的起始时间戳。

以下选项可用：

'汽车'

自动检测是否将 mfra 时间戳设置为 PTS 或 DTS （默认）

' dts '

将 mfra 时间戳设置为 DTS

'积分'

将 mfra 时间戳设置为 PTS

' 0 '

不要使用 mfra 框设置时间戳

使用\_tfdt

对于片段输入，将片段的起始时间戳设置为tfdt框中的baseMediaDecodeTime 。默认启用，这将更喜欢使用tfdt框来设置 DTS。禁止使用sidx框中的earliest\_presentation\_time 。在任何一种情况下，如果mfra框的时间戳可用且use\_mfra\_for设置为 pts 或 dts，则将使用它。

全部导出

udta框中无法识别的框导出为元数据条目。框类型的前四个字符设置为键。默认为假。

导出\_xmp

XMP\_ box 和uuid box的全部内容导出为带有键xmp 的字符串。请注意，如果设置了export\_all而未设置此选项，则XMP\_框的内容仍会导出，但密钥为XMP\_ 。默认为假。

激活字节

解密 Audible AAX 和 AAX+ 文件所需的 4 字节密钥。请参阅下面的 Audible AAX 小节。

audible\_fixed\_key

用于处理 Audible AAX/AAX+ 文件的固定键。它已预先设置，因此无需指定。

解密密钥

16 字节的十六进制密钥，用于解密使用 ISO 通用加密（CENC/AES-128 CTR；ISO/IEC 23001-7）加密的文件。

max\_stts\_delta

写入 trak 的 stts 框中的非常高的样本增量有时可能是有意的，但通常它们是错误写入的，或者在被视为带符号的 32 位整数时用于存储负值以进行 dts 校正。此选项允许用户设置上限，超过该上限增量将被限制为 1。当转换为 int32 时，如果为负，则大于该限制的值用于调整向前的 dts。

单位是轨道时间刻度。范围是 0 到 UINT\_MAX。默认值为UINT\_MAX - 48000\*10 ，它允许对 48 kHz 音频流进行最多 10 秒的 dts 校正，同时容纳 99.9% 的uint32范围。

#### 20.16.2 有声 AAX

Audible AAX 文件是加密的 M4B 文件，可以通过指定 4 字节的激活密码来解密。

ffmpeg -activation\_bytes 1CEB00DA -i test.aax -vn -c:复制输出.mp4

### 20.17 mpegts

MPEG-2 传输流解复用器。

此分路器接受以下选项：

重新同步大小

设置查找新同步的大小限制。默认值为 65536。

skip\_unknown\_pmt

跳过 PAT 中未定义的程序的 PMT。默认值为 0。

fix\_teletext\_pts

使用从第一个节目的 PCR 计算的时间戳覆盖图文数据包 PTS 和 DTS 值，图文电视流是其中的一部分并且未被丢弃。默认值为 1，如果您希望图文电视数据包 PTS 和 DTS 值保持不变，请将此选项设置为 0。

ts\_packetsize

以字节为单位携带原始数据包大小的输出选项。显示检测到的原始数据包大小，用户不能设置。

扫描所有pmts

扫描并组合所有 PMT。该值为-1到1之间的整数（-1表示自动设置，1表示启用，0表示禁用）。默认值为 -1。

merge\_pmt\_versions

当 PMT 的版本更新并且基本流移动到不同的 PID 时，重新使用现有流。默认值为 0。

最大数据包大小

设置分路器发出的数据包的最大大小（以字节为单位）。超过此大小的有效负载被拆分到多个数据包中。范围是 1 到 INT\_MAX/2。默认为 204800 字节。

### 20.18 mpjpeg

MJPEG 封装在多部分 MIME 分离器中。

此多路分解器允许读取 MJPEG，其中每个帧都表示为 multipart/x-mixed-replace 流的一部分。

strict\_mime\_boundary

默认实现将宽松的标准应用于多部分 MIME 边界检测，以防止大量现有端点无法生成正确的 MIME MJPEG 流的回归。通过将此选项设置为 1 打开此选项将导致对边界值进行更严格的检查。

### 20.19 原始视频

原始视频解复用器。

该解复用器允许读取原始视频数据。由于没有标头指定假定的视频参数，因此用户必须指定它们才能正确解码数据。

此分路器接受以下选项：

帧率

设置输入视频帧率。默认值为 25。

像素格式

设置输入视频像素格式。默认值为yuv420p 。

视频大小

设置输入视频大小。必须明确指定此值。

例如，要使用ffplay读取原始视频文件input.raw ，假设像素格式为rgb24 ，视频大小为320x240 ，帧速率为每秒 10 张图像，请使用以下命令：

ffplay -f rawvideo -pixel\_format rgb24 -video\_size 320x240 -framerate 10 input.raw

### 20.20 sbg

SBaGen 脚本多路分解器。

<http://uazu.net/sbagen/>使用的脚本语言来生成双耳节拍会话。 SBG 脚本如下所示：

-SE

一：300-2.5/3 440+4.5/0

b: 300-2.5/0 440+4.5/3

离开： -

现在==一个

+0:07:00 == 乙

+0:14:00 == 一个

+0:21:00 == 乙

+0:30:00 关闭

SBG 脚本可以混合使用绝对时间戳和相对时间戳。如果脚本仅使用绝对时间戳（包括脚本开始时间）或仅使用相对时间戳，则其布局是固定的，并且转换很简单。另一方面，如果脚本混合了这两种时间戳，那么相对时间戳的NOW参考将从脚本被读取时的当前时间获取，脚本布局将根据该参考冻结。这意味着如果直接播放脚本，实际时间将匹配绝对时间戳，直到声音控制器的时钟精度，但如果用户以某种方式暂停播放或搜索，所有时间都将相应地移动。

### 20.21 字幕

用于[TED 演讲的 JSON 字幕](http://www.ted.com/)。

TED 不提供标题链接，但可以从页面上猜到。 FFmpeg 源代码树中的文件tools/bookmarklets.html包含一个小书签来公开它们。

此分路器接受以下选项：

开始时间

设置 TED 演讲的开始时间，以毫秒为单位。默认值为 15000（15 秒）。它用于将字幕与可下载的视频同步，因为它们包含 15 秒的介绍。

示例：将字幕转换为大多数玩家都能理解的格式：

ffmpeg -i http://www.ted.com/talks/subtitles/id/1/lang/en talk1-en.srt

### 20.22 蒸汽合成器

Vapoursynth 包装纸。

出于安全考虑，不会自动检测 Vapoursynth 脚本，因此必须强制输入格式。对于 ff\* CLI 工具，在输入-i yourscript.vpy之前添加-f vapoursynth 。

此分路器接受以下选项：

最大脚本大小

分路器将整个脚本缓冲到内存中。调整此值以设置最大缓冲区大小，这反过来又充当可读取脚本大小的上限。默认值为 1 MiB。

## 21 个多路复用器

Muxers 是 FFmpeg 中的配置元素，它允许将多媒体流写入特定类型的文件。

当您配置 FFmpeg 构建时，默认情况下会启用所有受支持的混合器。您可以使用配置选项--list-muxers列出所有可用的多路复用器。

您可以使用配置选项--disable-muxers禁用所有多路复用器，并使用--enable-muxer= MUXER / --disable-muxer= MUXER选项选择性地启用/禁用单个多路复用器。

选项-muxers将显示启用的 muxers 列表。使用-formats查看已启用的多路分解器和多路复用器的组合列表。

下面是一些当前可用的多路复用器的描述。

* [a64](#a64-1)
* [广告](#adts-1)
* [爱夫](#aiff-1)
* [阿尔卑斯山](#alp-1)
* [asf](#asf-2)
* [avi](#avi-1)
* [套印](#chromaprint-1)
* [CRC](#crc-1)
* [短跑](#dash-2)
* [先进先出](#fifo-1)
* [flv](#flv)
* [帧crc](#framecrc-1)
* [帧哈希](#framehash-1)
* [framemd5](#framemd5-1)
* [动图](#gif-2)
* [散列](#hash-1)
* [hls](#hls-2)
* [图标](#ico-1)
* [图片2](#image2-2)
* [马特罗斯卡](#matroska)
* [MD5](#md5-1)
* [电影、mp4、ismv](#mov_002c-mp4_002c-ismv)
* [音乐播放器](#mp3)
* [mpegts](#mpegts-1)
* [mxf, mxf\_d10, mxf\_opatom](#mxf_002c-mxf_005fd10_002c-mxf_005fopato)
* [无效的](#null-1)
* [坚果](#nut)
* [奥格](#ogg)
* [原始混合器](#raw-muxers-1)
* [段，stream\_segment，ssegment](#segment_002c-stream_005fsegment_002c-ss)
* [流畅的流媒体](#smoothstreaming)
* [流哈希](#streamhash-1)
* [球座](#tee-1)
* [webm\_chunk](#webm_005fchunk)
* [webm\_dash\_manifest](#webm_005fdash_005fmanifest)

### 21.1 a64

用于 Commodore 64 视频的 A64 混合器。接受单个a64\_multi或a64\_multi5编解码器视频流。

### 21.2 广告

音频数据传输流复用器。它接受单个 AAC 流。

* [选项](#Options-48)

#### 21.2.1 选项

它接受以下选项：

write\_id3v2布尔

允许在流的开头写入 ID3v2.4 标签。默认是禁用的。

write\_apetag布尔

允许在流的末尾写入 APE 标签。默认是禁用的。

write\_mpeg2布尔

启用以将 ADTS 帧头中的 MPEG 版本位设置为 1，表示 MPEG-2。默认值为 0，表示 MPEG-4。

### 21.3 艾夫

音频交换文件格式混合器。

* [选项](#Options-49)

#### 21.3.1 选项

它接受以下选项：

write\_id3v2

设置为 1 时启用 ID3v2 标签写入。默认值为 0（禁用）。

id3v2\_版本

选择要写入的 ID3v2 版本。目前仅支持版本 3 和 4（也称为 ID3v2.3 和 ID3v2.4）。默认为版本 4。

### 21.4 阿尔卑斯山

用于 High Voltage Software 的 Lego Racers 游戏音频的 Muxer。它接受不超过 2 个通道且采样率不超过 44100 Hz 的单个 ADPCM\_IMA\_ALP 流。

扩展名：tun、pcm

* [选项](#Options-50)

#### 21.4.1 选项

它接受以下选项：

类型类型

设置文件类型。

'屯'

将文件类型设置为音乐。采样率必须为 22050 Hz。

' PCM '

将文件类型设置为 sfx。

'汽车'

根据输出文件扩展名设置文件类型。 .pcm导致类型pcm否则类型tun被设置。 （默认）

### 21.5 英尺

高级系统格式混合器。

请注意，Windows Media Audio (wma) 和 Windows Media Video (wmv) 也使用此多路复用器。

* [选项](#Options-51)

#### 21.5.1 选项

它接受以下选项：

数据包大小

设置多路复用器数据包大小。通过调整此设置，您可以减少数据碎片或多路复用器开销，具体取决于您的来源。默认值为 3200，最小值为 100，最大值为 64k。

### 21.6.avi

音频视频交错复用器。

* [选项](#Options-52)

#### 21.6.1 选项

它接受以下选项：

保留索引空间

为文件头中每个流的 OpenDML 主索引保留指定的字节数。默认情况下，如果第一个主索引中没有剩余空间，则附加主索引将嵌入数据包中，并作为索引链链接在一起。此索引结构可能会导致某些用例出现问题，例如第三方软件严格依赖 OpenDML 索引规范或文件查找速度缓慢。在文件头中保留足够的索引空间可以避免这些问题。

所需的索引空间取决于输出文件的大小，每 GB 大约需要 16 个字节。当此选项被省略或设置为零时，将猜测必要的索引空间。

写通道掩码

将通道布局掩码写入音频流标头。

默认情况下启用此选项。禁用通道掩码在特定情况下可能很有用，例如，当将多个音频流合并为一个以与仅支持 AVI 中的单个音频流的软件兼容时（请参阅（ffmpeg-filters）ffmpeg-filters 手册中的“amerge”[部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-filters.html#amerge)).

flipped\_raw\_rgb

如果设置为 true，则存储原始 RGB 位图的正高度，这表示位图是自下而上存储的。请注意，此选项不会翻转必须事先手动完成的位图，例如使用 vflip 过滤器。默认为false ，表示位图自上而下存储。

### 21.7 套印

色谱指纹仪。

此多路复用器将音频数据提供给 Chromaprint 库，后者为提供的音频数据生成指纹。请参阅<https://acoustid.org/chromaprint>

它采用最多 2 个通道的单个签名本机端 16 位原始音频流。

* [选项](#Options-53)

#### 21.7.1 选项

沉默阈值

检测静音的阈值。范围从 -1 到 32767，其中 -1 禁用静音检测。静音检测只能与算法的版本 3 一起使用。必须禁用静音检测才能与 AcoustID 服务一起使用。默认为 -1。

算法

用于指纹的算法版本。范围是 0 到 4。版本 3 启用静音检测。默认值为 1。

fp\_格式

将指纹输出为的格式。接受以下选项：

'生'

二进制原始指纹

'压缩'

二进制压缩指纹

' base64 '

Base64压缩指纹（默认）

### 21.8CRC

CRC（循环冗余校验）测试格式。

此多路复用器计算并打印所有输入音频和视频帧的 Adler-32 CRC。默认情况下，在计算 CRC 之前，音频帧被转换为带符号的 16 位原始音频，视频帧被转换为原始视频。

多路复用器的输出由一行形式组成： CRC=0x CRC ，其中CRC是一个十六进制数，用 0 填充到 8 位数字，包含所有解码输入帧的 CRC。

另请参阅[framecrc](#framecrc)多路复用器。

* [例子](#Examples-9)

#### 21.8.1 例子

例如计算输入的 CRC，并将其存储在文件out.crc中：

ffmpeg -i 输入 -f crc 输出.crc

您可以使用以下命令将 CRC 打印到标准输出：

ffmpeg -i 输入 -f crc -

通过指定音视频编解码器和格式来选择使用ffmpeg的每一帧的输出格式。例如，要计算转换为 PCM 无符号 8 位的输入音频和转换为 MPEG-2 视频的输入视频的 CRC，请使用以下命令：

ffmpeg -i 输入 -c:a pcm\_u8 -c:v mpeg2video -f crc -

### 21.9 破折号

HTTP 动态自适应流 (DASH) 多路复用器，根据 MPEG-DASH 标准 ISO/IEC 23009-1:2014 创建分段和清单文件。

有关详细信息，请参阅：

* ISO DASH 规范： <http://standards.iso.org/ittf/PubliclyAvailableStandards/c065274_ISO_IEC_23009-1_2014.zip>
* WebM DASH 规范： <https://sites.google.com/a/webmproject.org/wiki/adaptive-streaming/webm-dash-specification>

它为每个流创建一个 MPD 清单文件和分段文件。

段文件名可能包含与标准第 5.3.9.4.4 节中定义的 SegmentTemplate 一起使用的预定义标识符。可用的标识符是“$RepresentationID$”、“$Number$”、“$Bandwidth$”和“$Time$”。除了标准标识符之外，还支持 ffmpeg 特定的“$ext$”标识符。当指定 ffmpeg 时，会将文件名中的 $ext$ 替换为 muxing 格式的扩展名，如 mp4、webm 等，

ffmpeg -re -i <输入> -map 0 -map 0 -c:a libfdk\_aac -c:v libx264 \

-b:v:0 800k -b:v:1 300k -s:v:1 320x170 -profile:v:1 基线\

-profile:v:0 main -bf 1 -keyint\_min 120 -g 120 -sc\_threshold 0 \

-b\_strategy 0 -ar:a:1 22050 -use\_timeline 1 -use\_template 1 \

-window\_size 5 -adaptation\_sets "id=0,streams=v id=1,streams=a" \

-f破折号/path/to/out.mpd

seg\_duration持续时间

以秒为单位设置分段长度（可以设置分数值）。当启用use\_template并禁用use\_timeline时，该值被视为平均段持续时间，并被视为所有其他用例的最短段持续时间。

frag\_duration持续时间

设置段内片段的长度（以秒为单位）（可以设置分数值）。

frag\_type类型

设置分片的间隔类型。

window\_size大小

设置清单中保留的最大段数。

extra\_window\_size大小

在从磁盘中删除之前，设置保留在清单之外的最大段数。

remove\_at\_exit删除

完成后启用 (1) 或禁用 (0) 删除所有段。

use\_template模板

启用 (1) 或禁用 (0) 使用 SegmentTemplate 而不是 SegmentList。

use\_timeline时间线

在 SegmentTemplate 中启用 (1) 或禁用 (0) SegmentTimeline。

单文件单文件

启用 (1) 或禁用 (0) 将所有段存储在一个文件中，使用字节范围进行访问。

single\_file\_name文件名

用于 baseURL 的 DASH 模板名称。暗示single\_file设置为“1”。在模板中，“$ext$”替换为特定于段格式的文件扩展名。

init\_seg\_name初始名称

用于初始化段的 DASH 模板名称。默认为“init-stream$RepresentationID$.$ext$”。 “$ext$”替换为特定于段格式的文件扩展名。

media\_seg\_name段名

用于媒体片段的 DASH 模板名称。默认为“chunk-stream$RepresentationID$-$Number%05d$.$ext$”。 “$ext$”替换为特定于段格式的文件扩展名。

utc\_timing\_url utc\_url

将返回 ISO 格式的 UTC 时间戳的页面的 URL。示例：“https://time.akamai.com/?iso”

方法方法

使用给定的 HTTP 方法创建输出文件。一般设置为PUT或POST。

http\_user\_agent用户代理

覆盖 HTTP 标头中的 User-Agent 字段。仅适用于 HTTP 输出。

http\_persistent http\_persistent

使用持久 HTTP 连接。仅适用于 HTTP 输出。

hls\_playlist hls\_播放列表

同时生成 HLS 播放列表文件。使用文件名hls\_master\_name生成主播放列表。为每个流生成一个媒体播放列表文件，文件名为 media\_0.m3u8、media\_1.m3u8 等。

hls\_master\_name文件名

HLS 主播放列表名称。默认为“master.m3u8”。

流式传输

启用 (1) 或禁用 (0) 输出块流模式。在 chunk streaming 模式下，每一帧都是一个 moof 片段，形成一个 chunk。

adaptation\_sets适应集

将流分配给 AdaptationSet。语法为“id=x,streams=a,b,c id=y,streams=d,e”，其中 x 和 y 是适配集的 ID，a、b、c、d 和 e 是适配集的索引映射流。

要将所有视频（或音频）流映射到 AdaptationSet，可以使用“v”（或“a”）作为流标识符而不是 ID。

当没有定义分配时，这默认为每个流的 AdaptationSet。

可选语法是“id=x,seg\_duration=x,frag\_duration=x,frag\_type=type,descriptor=descriptor\_string,streams=a,b,c id=y,seg\_duration=y,frag\_type=type,streams=d,e”和依此类推，描述符对 ISO/IEC 23009-1:2014/Amd.2:2015 定义的方案有用。例如，-adaptation\_sets "id=0,descriptor=<SupplementalProperty schemeIdUri=\"urn:mpeg:dash:srd:2014\" value=\"0,0,0,1,1,2,2\"/> ,流=v”。请注意，描述符字符串应该是一个自闭合的 xml 标记。 seg\_duration、frag\_duration 和 frag\_type 覆盖每个适应集的全局选项值。例如，-adaptation\_sets "id=0,seg\_duration=2,frag\_duration=1,frag\_type=duration,streams=v id=1,seg\_duration=2,frag\_type=none,streams=a" type\_id 将适配集标记为包含流意味着用于引用适配集的技巧模式。例如，- adaptation\_sets "id=0,seg\_duration=2,frag\_type=none,streams=0 id=1,seg\_duration=10,frag\_type=none,trick\_id=0,streams=1"

超时超时

设置套接字 I/O 操作的超时时间。仅适用于 HTTP 输出。

索引校正索引校正

启用 (1) 或禁用 (0) 段索引校正逻辑。仅在启用use\_template且禁用use\_timeline时适用。

启用后，逻辑会监视段索引的流。如果流的段索引值不在预期的实时位置，则逻辑会更正该索引值。

通常在实时流媒体用例中需要此逻辑。网络带宽波动在长时间流式传输期间很常见。每次波动都会导致板块指数落后于预期的实时位置。

format\_options选项列表

:分隔的 key=value 参数列表设置容器格式 (mp4/webm) 选项。包含:特殊字符的值必须转义。

global\_sidx全局\_sidx

写入全局 SIDX 原子。仅适用于单个文件、mp4 输出、非流模式。

dash\_segment\_type dash\_segment\_type

可能的值：

汽车

如果设置此标志，将根据流编解码器选择破折号段文件格式。这是默认模式。

mp4

如果设置了此标志，破折号段文件将采用 ISOBMFF 格式。

网页管理器

如果设置了此标志，破折号段文件将采用 WebM 格式。

ignore\_io\_errors忽略\_io\_errors

在打开和写入过程中忽略 IO 错误。对于具有网络输出的长时间运行很有用。

lhls lhls

启用低延迟 HLS(LHLS)。添加带有当前段的 URI 的#EXT-X-PREFETCH 标签。 hls.js 玩家正在尝试标准化一个开放的 LHLS 规范。规范草案可在 https://github.com/video-dev/hlsjs-rfcs/blob/lhls-spec/proposals/0001-lhls.md 中找到。此选项试图遵守上述开放规范。它会自动启用流媒体和hls\_playlist选项。这是一项实验性功能。

注意：这不是 Apple 的 LHLS 版本。请参阅<https://datatracker.ietf.org/doc/html/draft-pantos-hls-rfc8216bis>

ldash ldash

通过限制某些元素的存在和值来启用低延迟 Dash。

master\_m3u8\_publish\_rate master\_m3u8\_publish\_rate

每隔指定数量的分段间隔重复发布主播放列表。

write\_prft写\_prft

在支持的流上写入生产者参考时间元素。这也允许在底层复用器中写入 prft 框。仅在启用utc\_url选项时适用。它默认设置为自动，在这种情况下，多路复用器将尝试仅在需要它的模式下启用它。

mpd\_profile mpd\_profile

设置一个或多个清单配置文件。

http\_opts http\_opts

传递给底层 HTTP 协议的 : 分隔的 key=value 选项列表。仅适用于 HTTP 输出。

目标延迟目标延迟

以秒为单位设置预期目标延迟（可以设置分数值）以进行服务。仅在启用流和write\_prft选项时适用。这是一个信息字段，客户端可以使用它来衡量服务的延迟。

min\_playback\_rate min\_playback\_rate

设置适当指示的最小播放速率，以便在客户端正常播放时自动调整播放延迟和缓冲区占用。

最大播放速率最大播放速率

设置适当指示的最大播放速率，以便在客户端正常播放时自动调整播放延迟和缓冲区占用。

update\_period update\_period

为动态内容设置 mpd 更新周期。单位是秒。

### 21.10 先进先出

fifo pseudo-muxer 通过使用先进先出队列并在单独的线程中运行实际的 muxer 来分离编码和 muxing。这在与[tee](#tee) muxer 结合时特别有用，可用于将数据发送到具有不同可靠性/写入速度/延迟的多个目的地。

API 用户应该知道在其 AVFormatContext 中使用的回调函数（interrupt\_callback、io\_open 和 io\_close）必须是线程安全的。

如果队列已满或输出失败，fifo 多路复用器的行为是可选的，

* 输出可以透明地重新启动，重试之间的可配置延迟基于实时或已处理流的时间。
* 编码可以在临时故障期间被阻止，或者在 fifo 队列已满的情况下继续透明地丢弃数据包。

fifo\_format

指定格式名称。如果无法从输出名称后缀猜到它，则很有用。

队列大小

指定队列的大小（数据包数）。默认值为 60。

格式选项

为底层多路复用器指定格式选项。 Muxer 选项可以指定为由“:”分隔的key = value对列表。

drop\_pkts\_on\_overflow布尔

如果设置为 1 (true)，万一 fifo 队列填满，数据包将被丢弃而不是阻塞编码器。这使得可以在不延迟输入的情况下继续流式传输，但代价是省略部分流。默认情况下，此选项设置为 0 (false)，因此在这种情况下，编码器将被阻塞，直到 muxer 处理了一些数据包并且没有任何数据包丢失。

attempt\_recovery布尔

如果发生故障，请尝试恢复输出。这在与网络输出一起使用时特别有用，因为它可以透明地重新启动流式传输。默认情况下，此选项设置为 0（假）。

最大恢复尝试次数

设置输出永久失败后连续不成功恢复尝试的最大次数。默认情况下，此选项设置为 0（无限制）。

recovery\_wait\_time持续时间

在上一次不成功的恢复尝试之后，下一次恢复尝试之前的等待时间。默认值为 5 秒。

recovery\_wait\_streamtime布尔

如果设置为 0 (false)，则在等待恢复尝试时使用实时（即至少在 recovery\_wait\_time 秒后尝试恢复）。如果设置为 1 (true)，则改为考虑处理流的时间（即，将在至少省略流的recovery\_wait\_time秒后尝试恢复）。默认情况下，此选项设置为 0（假）。

recover\_any\_error布尔

如果设置为 1 (true)，无论导致失败的错误类型如何，都将尝试恢复。默认情况下，此选项设置为 0（假），并且在某些（通常是永久性）错误的情况下，即使attempt\_recovery设置为 1，也不会尝试恢复。

restart\_with\_keyframe布尔

指定从队列溢出或失败中恢复后是否等待关键帧。默认情况下，此选项设置为 0（假）。

时移持续时间

缓冲指定数量的数据包并延迟写入输出。请注意， queue\_size必须足够大以存储用于时移的数据包。在输入结束时，fifo 缓冲区以实时速度刷新。

* [例子](#Examples-10)

#### 21.10.1 例子

* 将某些内容流式传输到 rtmp 服务器，即使在出现临时故障（网络中断）的情况下也继续以实时速率处理流式传输，并尝试无限期地每秒恢复流式传输。
* ffmpeg -re -i ... -c:v libx264 -c:a aac -f fifo -fifo\_format flv -map 0:v -map 0:a
* -drop\_pkts\_on\_overflow 1 -attempt\_recovery 1 -recovery\_wait\_time 1 rtmp://example.com/live/stream\_name

### 21.11 flv

Adobe Flash 视频格式混合器。

这个 muxer 接受以下选项：

flvflags标志

可能的值：

' aac\_seq\_header\_detect '

根据音频流数据放置AAC序列头。

' no\_sequence\_end '

禁用序列结束标记。

' no\_metadata '

禁用元数据标签。

' no\_duration\_filesize '

当流结束时元数据中的持续时间和文件大小为零时禁用它们。 （习惯于不可搜索的直播）。

' add\_keyframe\_index '

用于方便查找；特别是对于 HTTP 伪流。

### 21.12 帧crc

每数据包 CRC（循环冗余校验）测试格式。

此多路复用器为每个音频和视频数据包计算并打印 Adler-32 CRC。默认情况下，在计算 CRC 之前，音频帧被转换为带符号的 16 位原始音频，视频帧被转换为原始视频。

复用器的输出由以下形式的每个音频和视频数据包组成：

stream\_index , packet\_dts , packet\_pts , packet\_duration , packet\_size , 0x CRC

CRC是一个十六进制数，用 0 填充到包含数据包 CRC 的 8 位数字。

* [例子](#Examples-11)

#### 21.12.1 例子

例如，计算INPUT中音频和视频帧的 CRC ，转换为原始音频和视频数据包，并将其存储在文件out.crc中：

ffmpeg -i INPUT -f framecrc 输出.crc

要将信息打印到标准输出，请使用以下命令：

ffmpeg -i 输入 -f framecrc -

使用ffmpeg ，您可以在通过指定音频和视频编解码器为每个数据包计算 CRC 之前选择音频和视频帧编码的输出格式。例如，要计算转换为 PCM 无符号 8 位的每个解码输入音频帧和转换为 MPEG-2 视频的每个解码输入视频帧的 CRC，请使用以下命令：

ffmpeg -i 输入 -c:a pcm\_u8 -c:v mpeg2video -f framecrc -

另请参阅[crc](#crc)复用器。

### 21.13 帧哈希

每数据包哈希测试格式。

此多路复用器为每个音频和视频数据包计算并打印加密哈希。这可以用于逐个数据包的相等性检查，而不必对每个数据包单独进行二进制比较。

默认情况下，在计算哈希值之前，音频帧被转换为带符号的 16 位原始音频，视频帧被转换为原始视频，但也可以使用显式转换为其他编解码器的输出。它默认使用 SHA-256 加密哈希函数，但支持其他几种算法。

复用器的输出由以下形式的每个音频和视频数据包组成：

stream\_index , packet\_dts , packet\_pts , packet\_duration , packet\_size , hash

hash是一个十六进制数，表示为数据包计算的哈希值。

哈希算法

算法指定的加密哈希函数。支持的值包括MD5 、 murmur3 、 RIPEMD128 、 RIPEMD160 、 RIPEMD256 、 RIPEMD320 、 SHA160 、 SHA224 、 SHA256 （默认）、 SHA512/224 、 SHA512/256 、 SHA384 、 SHA512 、 CRC32和adler32 。

* [例子](#Examples-12)

#### 21.13.1 例子

INPUT中音频和视频帧的 SHA-256 哈希，转换为原始音频和视频数据包，并将其存储在文件out.sha256中：

ffmpeg -i 输入 -f framehash out.sha256

要使用 MD5 哈希函数将信息打印到标准输出，请使用以下命令：

ffmpeg -i 输入 -f framehash -hash md5 -

另请参阅[哈希](#hash)混合器。

### 21.14 framemd5

Per-packet MD5 测试格式。

[framehash muxer](#framehash)的变体。与那个 muxer 不同，它默认使用 MD5 哈希函数。

* [例子](#Examples-13)

#### 21.14.1 例子

计算INPUT中音频和视频帧的 MD5 哈希，转换为原始音频和视频数据包，并将其存储在文件out.md5中：

ffmpeg -i 输入 -f framemd5 输出.md5

要将信息打印到标准输出，请使用以下命令：

ffmpeg -i 输入 -f framemd5 -

另请参阅[framehash](#framehash)和[md5](#md5) muxers。

### 21.15 动图

动画 GIF 混合器。

它接受以下选项：

环形

设置循环输出的次数。使用-1表示不循环，使用0表示无限循环（默认）。

final\_delay

在最后一帧之后强制延迟（以厘秒表示）。每一帧都延迟到下一帧结束。默认值为-1 ，这是一个特殊值，用于告知复用器重新使用之前的延迟。在循环的情况下，您可能希望自定义此值以标记暂停。

例如，要对循环 10 次的 gif 进行编码，循环之间有 5 秒的延迟：

ffmpeg -i INPUT -loop 10 -final\_delay 500 输出.gif

注意 1：如果你想将帧提取到单独的 GIF 文件中，你需要强制使用[image2](#image2) muxer：

ffmpeg -i 输入 -c:v gif -f image2 "out%d.gif"

注 2：GIF 格式的时基非常大：因此两帧之间的延迟不能小于一厘秒。

### 21.16哈希

哈希测试格式。

此多路复用器计算并打印所有输入音频和视频帧的加密哈希。这可用于相等性检查，而无需进行完整的二进制比较。

默认情况下，在计算哈希值之前，音频帧被转换为带符号的 16 位原始音频，视频帧被转换为原始视频，但也可以使用显式转换为其他编解码器的输出。时间戳被忽略。它默认使用 SHA-256 加密哈希函数，但支持其他几种算法。

muxer 的输出由一行形式组成： algo = hash ，其中algo是一个短字符串，表示所使用的哈希函数，而hash是一个十六进制数，表示计算出的哈希值。

哈希算法

算法指定的加密哈希函数。支持的值包括MD5 、 murmur3 、 RIPEMD128 、 RIPEMD160 、 RIPEMD256 、 RIPEMD320 、 SHA160 、 SHA224 、 SHA256 （默认）、 SHA512/224 、 SHA512/256 、 SHA384 、 SHA512 、 CRC32和adler32 。

* [例子](#Examples-14)

#### 21.16.1 例子

要计算转换为原始音频和视频的输入的 SHA-256 哈希值，并将其存储在文件out.sha256中：

ffmpeg -i 输入 -f 哈希输出.sha256

要将 MD5 哈希打印到标准输出，请使用以下命令：

ffmpeg -i 输入 -f hash -hash md5 -

另请参阅[framehash](#framehash) muxer。

### 21.17 小时

Apple HTTP Live Streaming muxer，根据 HTTP Live Streaming (HLS) 规范对 MPEG-TS 进行分段。

它创建一个播放列表文件，以及一个或多个片段文件。输出文件名指定播放列表文件名。

默认情况下，muxer 为每个生成的段创建一个文件。这些文件与播放列表同名，后跟序列号和 .ts 扩展名。

确保在编码时需要一个封闭的 GOP，并设置 GOP 大小以适应您的分段时间限制。

例如，要使用ffmpeg转换输入文件：

ffmpeg -i in.mkv -c:v h264 -flags +cgop -g 30 -hls\_time 1 out.m3u8

此示例将生成播放列表、 out.m3u8和片段文件： out0.ts 、 out1.ts 、 out2.ts等。

另请参阅[段](#segment)复用器，它提供了更通用和更灵活的分段器实现，可用于执行 HLS 分段。

* [选项](#Options-54)

#### 21.17.1 选项

此多路复用器支持以下选项：

hls\_init\_time持续时间

设置初始目标段长度。默认值为0 。

duration必须是持续时间规范，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)。

在第一个 m3u8 列表上经过此时间后，将在下一个关键帧切割片段。初始播放列表填满后， ffmpeg将以等于hls\_time 的持续时间剪切片段

hls\_持续时间

设置目标段长度。默认值为 2。

duration必须是持续时间规范，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)。这段时间过去后，片段将在下一个关键帧被剪切。

hls\_list\_size大小

设置播放列表条目的最大数量。如果设置为 0，列表文件将包含所有段。默认值为 5。

hls\_delete\_threshold大小

hls\_flags delete\_segments删除它们之前保留在磁盘上的未引用段的数量。增加此值以允许继续的客户端下载播放列表中最近引用的片段。默认值为 1，表示将删除早于hls\_list\_size+1的段。

hls\_start\_number\_source

根据指定的来源启动播放列表序列号（ #EXT-X-MEDIA-SEQUENCE ）。除非hls\_flags single\_file被设置，它还指定片段和字幕文件名的起始序列号的来源。在任何情况下，如果设置了hls\_flags append\_list并且读取的播放列表序列号大于指定的起始序列号，则该值将用作起始值。

它接受以下值：

通用（默认）

根据start\_number选项值设置起始序号。

时代

起始编号将是自纪元 (1970-01-01 00:00:00) 以来的秒数

大纪元

起始编号将是自纪元 (1970-01-01 00:00:00) 以来的微秒数

约会时间

开始编号将基于当前日期/时间 YYYYmmddHHMMSS。例如 20161231235759。

start\_number号码

当hls\_start\_number\_source值为generic时，从指定编号开始播放列表序列号 ( #EXT-X-MEDIA-SEQUENCE ) 。 （这是默认情况。）除非设置了hls\_flags single\_file ，否则它还会指定片段和字幕文件名的起始序列号。默认值为 0。

hls\_allow\_cache允许缓存

显式设置客户端是可以（1）还是不能（0）缓存媒体段。

hls\_base\_url baseurl

将baseurl附加到播放列表中的每个条目。用于生成具有绝对路径的播放列表。

请注意，每个片段的播放列表序列号必须是唯一的，并且不要与可以循环的片段文件名序列号混淆，例如，如果指定了 wrap选项。

hls\_segment\_filename文件名

设置段文件名。除非设置了hls\_flags single\_file ，否则文件名将用作带段号的字符串格式：

ffmpeg -i in.nut -hls\_segment\_filename '文件%03d.ts' out.m3u8

此示例将生成播放列表、 out.m3u8和片段文件： file000.ts 、 file001.ts 、 file002.ts等。

文件名可能包含完整路径或相对路径规范，但只有没有任何路径信息的文件名部分将包含在 m3u8 段列表中。如果指定了相对路径，则创建的段文件的路径将相对于当前工作目录。当设置strftime\_mkdir 时， filename的整个扩展值将被写入m3u8 段列表。

当var\_stream\_map设置了两个或多个变体流时，文件名模式必须包含字符串“%v”，该字符串指定变体流索引在生成的段文件名中的位置。

ffmpeg -i in.ts -b:v:0 1000k -b:v:1 256k -b:a:0 64k -b:a:1 32k \

-map 0:v -map 0:a -map 0:v -map 0:a -f hls -var\_stream\_map "v:0,a:0 v:1,a:1" \

-hls\_segment\_filename 'file\_%v\_%03d.ts' out\_%v.m3u8

此示例将生成播放列表段文件集： file\_0\_000.ts 、 file\_0\_001.ts 、 file\_0\_002.ts等和file\_1\_000.ts 、 file\_1\_001.ts 、 file\_1\_002.ts等。

字符串“%v”可能出现在文件名中，也可能出现在包含该文件的最后一个目录名中，但只能出现在其中之一中。 （此外，%v 可能在最后一个子目录或文件名中出现多次。）如果字符串 %v 出现在目录名称中，则在扩展目录名称模式后创建子目录。这使得能够创建对应于子目录中不同变体流的段。

ffmpeg -i in.ts -b:v:0 1000k -b:v:1 256k -b:a:0 64k -b:a:1 32k \

-map 0:v -map 0:a -map 0:v -map 0:a -f hls -var\_stream\_map "v:0,a:0 v:1,a:1" \

-hls\_segment\_filename 'vs%v/file\_%03d.ts' vs%v/out.m3u8

此示例将生成播放列表段文件集： vs0/file\_000.ts 、 vs0/file\_001.ts 、 vs0/file\_002.ts等以及vs1/file\_000.ts 、 vs1/file\_001.ts 、 vs1/file\_002.ts等.

战斗时间

文件名上使用 strftime()以使用本地时间扩展段文件名。段号在此模式下也可用，但要使用它，您需要指定 second\_level\_segment\_index hls\_flag 并且 %%d 将作为说明符。

ffmpeg -i in.nut -strftime 1 -hls\_segment\_filename '文件-%Y%m%d-%s.ts' out.m3u8

此示例将生成播放列表、 out.m3u8和片段文件： file-20160215-1455569023.ts 、 file-20160215-1455569024.ts等。注意：在某些系统/环境中， %s说明符不可用。请参阅strftime()文档。

ffmpeg -i in.nut -strftime 1 -hls\_flags second\_level\_segment\_index -hls\_segment\_filename '文件-%Y%m%d-%%04d.ts' out.m3u8

此示例将生成播放列表、 out.m3u8和片段文件： file-20160215-0001.ts 、 file-20160215-0002.ts等。

strftime\_mkdir

与 -strftime\_mkdir 一起使用，它将创建在filename中扩展的所有子目录。

ffmpeg -i in.nut -strftime 1 -strftime\_mkdir 1 -hls\_segment\_filename '%Y%m%d/file-%Y%m%d-%s.ts' out.m3u8

此示例将创建一个目录 201560215（如果它不存在），然后生成播放列表、 out.m3u8和片段文件： 20160215/file-20160215-1455569023.ts 、 20160215/file-20160215-1455569024.ts等.

ffmpeg -i in.nut -strftime 1 -strftime\_mkdir 1 -hls\_segment\_filename '%Y/%m/%d/file-%Y%m%d-%s.ts' out.m3u8

此示例将创建目录层次结构 2016/02/15（如果其中任何一个不存在），然后生成播放列表、 out.m3u8和片段文件： 2016/02/15/file-20160215-1455569023.ts ， 2016/02/15/file-20160215-1455569024.ts等

hls\_segment\_options选项列表

使用 : 分隔的 key=value 参数列表设置输出格式选项。包含:特殊字符的值必须转义。

hls\_key\_info\_file key\_info\_file

key\_info\_file中的信息进行段加密。 key\_info\_file的第一行指定写入播放列表的密钥 URI。密钥 URL 用于在播放期间访问加密密钥。第二行指定在加密过程中用于获取密钥的密钥文件的路径。密钥文件被读取为二进制格式的 16 个八位字节的单个打包数组。可选的第三行将初始化向量 (IV) 指定为要使用的十六进制字符串，而不是用于加密的段序列号（默认值）。如果启用了hls\_flags periodic\_rekey，则对key\_info\_file的更改将导致使用新密钥/IV 进行段加密，并在播放列表中为新密钥 URI/IV生成一个条目。

关键信息文件格式：

密钥 URI

密钥文件路径

四（可选）

示例关键 URI：

http://服务器/文件.key

/路径/到/file.key

文件.key

示例密钥文件路径：

文件.key

/路径/到/file.key

例四：

0123456789ABCDEF0123456789ABCDEF

关键信息文件示例：

http://服务器/文件.key

/路径/到/file.key

0123456789ABCDEF0123456789ABCDEF

示例 shell 脚本：

#!/bin/sh

BASE\_URL=${1:-'.'}

openssl rand 16 > 文件密钥

echo $BASE\_URL/file.key > file.keyinfo

echo file.key >> file.keyinfo

echo $(openssl rand -hex 16) >> file.keyinfo

ffmpeg -f lavfi -re -i testsrc -c:v h264 -hls\_flags delete\_segments \

-hls\_key\_info\_file file.keyinfo out.m3u8

-hls\_enc编码

启用 (1) 或禁用 (0) AES128 加密。启用后，生成的每个片段都会被加密，加密密钥将保存为播放列表名称.key。

-hls\_enc\_key密钥

用于加密段的 16 字节密钥，默认情况下它是随机生成的。

-hls\_enc\_key\_url keyurl

如果设置， keyurl将代替baseurl到播放列表中的密钥文件名。

-hls\_enc\_iv iv

每个段的 16 字节初始化向量而不是自动生成的向量。

hls\_segment\_type标志

可能的值：

' mpegts '

以 MPEG-2 传输流格式输出片段文件。这与所有 HLS 版本兼容。

' fmp4 '

以碎片化的 MP4 格式输出片段文件，类似于 MPEG-DASH。 fmp4 文件可用于 HLS 版本 7 及更高版本。

hls\_fmp4\_init\_filename文件名

将文件名设置为片段文件头文件，默认文件名是init.mp4 。

在文件名上使用-strftime 1以使用本地时间扩展段文件名。

ffmpeg -i in.nut -hls\_segment\_type fmp4 -strftime 1 -hls\_fmp4\_init\_filename "%s\_init.mp4" out.m3u8

这将产生这样的初始化1602678741\_init.mp4

hls\_fmp4\_init\_resend

每次 m3u8 文件刷新后重新发送 init 文件，默认为0 。

当var\_stream\_map设置了两个或多个变体流时，文件名模式必须包含字符串“%v”，该字符串指定变体流索引在生成的 init 文件名中的位置。字符串“%v”可能出现在文件名中或包含该文件的最后一个目录名中。如果字符串出现在目录名称中，则在扩展目录名称模式后创建子目录。这使得能够在子目录中创建对应于不同变体流的初始化文件。

hls\_flags标志

可能的值：

' single\_file '

如果设置了此标志，则复用器会将所有片段存储在单个 MPEG-TS 文件中，并将使用播放列表中的字节范围。用这种方式生成的 HLS 播放列表的版本号为 4。例如：

ffmpeg -i in.nut -hls\_flags single\_file out.m3u8

将生成播放列表out.m3u8和单个片段文件out.ts 。

'删除段'

从播放列表中移除的片段文件在一段等于片段持续时间加上播放列表持续时间的时间后被删除。

'附加列表'

将新段附加到旧段列表的末尾，并从旧段列表中删除#EXT-X-ENDLIST 。

' round\_durations '

将播放列表文件段信息中的持续时间信息四舍五入为整数值，而不是使用浮点数。如果没有其他功能需要使用更高的 HLS 版本，那么这将允许 ffmpeg 输出 HLS 版本 2 m3u8。

' discont\_start '

#EXT-X-DISCONTINUITY标签添加到播放列表中，在第一个片段的信息之前。

' omit\_endlist '

不要在播放列表末尾附加EXT-X-ENDLIST标签。

' periodic\_rekey '

hls\_key\_info\_file指定的文件将定期检查并检测加密信息的更新。请务必自动替换此文件，包括包含 AES 加密密钥的文件。

'独立段'

#EXT-X-INDEPENDENT-SEGMENTS添加到具有视频片段的播放列表，并且保证该播放列表的所有片段都以关键帧开始。

' iframes\_only '

#EXT-XI-FRAMES-ONLY添加到具有视频片段且只能在#EXT-X-BYTERANGE模式下播放 I 帧的播放列表。

'按时间拆分'

允许分段从关键帧以外的帧开始。当关键帧之间的时间不一致时，这会改善某些玩家的行为，但可能会使其他人的情况变得更糟，并可能在搜索过程中导致一些奇怪的现象。此标志应与hls\_time选项一起使用。

' program\_date\_time '

生成EXT-X-PROGRAM-DATE-TIME标签。

' second\_level\_segment\_index '

当 strftime 打开时，除了日期/时间值之外，还可以在 hls\_segment\_filename 表达式中将段索引用作 %%d。要获得带有尾随零的固定宽度数字，可以使用 %%0xd 格式，其中 x 是所需的宽度。

' second\_level\_segment\_size '

当 strftime 打开时，除了日期/时间值之外，还可以在 hls\_segment\_filename 表达式中使用段大小（以字节计）作为 %%s。要获得带有尾随零的固定宽度数字，可以使用 %%0xs 格式，其中 x 是所需的宽度。

' second\_level\_segment\_duration '

当 strftime 打开时，除了日期/时间值之外，还可以在 hls\_segment\_filename 表达式中使用段持续时间（以微秒计算）作为 %%t。要获得带有尾随零的固定宽度数字，可以使用 %%0xt 格式，其中 x 是所需的宽度。

ffmpeg -i 样本.mpeg \

-f hls -hls\_time 3 -hls\_list\_size 5 \

-hls\_flags second\_level\_segment\_index+second\_level\_segment\_size+second\_level\_segment\_duration \

-strftime 1 -strftime\_mkdir 1 -hls\_segment\_filename "segment\_%Y%m%d%H%M%S\_%%04d\_%%08s\_%%013t.ts" stream.m3u8

这将产生这样的段： segment\_20170102194334\_0003\_00122200\_0000003000000.ts ， segment\_20170102194334\_0004\_00120072\_0000003000000.ts等。

'临时文件'

将段数据写入 filename.tmp 并仅在段完成后重命名为 filename。提供片段的网络服务器可以配置为拒绝对 \*.tmp 的请求，以防止在将片段添加到 m3u8 播放列表之前访问正在进行的片段。此标志还会影响 m3u8 播放列表文件的创建方式。如果设置此标志，所有播放列表文件将写入临时文件并在完成后重命名，类似于片段处理。但是不管这个标志如何，文件协议和类型（ hls\_playlist\_type ）不是vod的播放列表总是被写入临时文件。如果master\_pl\_publish\_rate值不为零，主播放列表文件 ( master\_pl\_name )，如果有的话，使用文件协议，总是被写入临时文件而不考虑这个标志。

hls\_playlist\_type 事件

在 m3u8 标头中发出#EXT-X-PLAYLIST-TYPE:EVENT 。强制hls\_list\_size为 0；播放列表只能附加到。

hls\_playlist\_type 点播

在 m3u8 标头中发出#EXT-X-PLAYLIST-TYPE:VOD 。强制hls\_list\_size为 0；播放列表不得更改。

方法

使用给定的 HTTP 方法创建 hls 文件。

ffmpeg -re -i in.ts -f hls -方法 PUT http://example.com/live/out.m3u8

本示例将使用 HTTP PUT 方法将所有 mpegts 段文件上传到 HTTP 服务器，并使用相同的方法在每次刷新时更新 m3u8 文件。请注意，HTTP 服务器必须支持给定的文件上传方法。

http\_user\_agent

覆盖 HTTP 标头中的 User-Agent 字段。仅适用于 HTTP 输出。

var\_stream\_map

映射字符串，指定如何将音频、视频和字幕流分组到不同的变体流中。变体流组由空格分隔。预期的字符串格式类似于“a:0,v:0 a:1,v:1 ....”。这里a:、v:、s:分别是指定音频、视频和字幕流的键。允许的值为 0 到 9（仅根据实际使用进行限制）。

当有两个或多个变体流时，输出文件名模式必须包含字符串“%v”，该字符串指定变体流索引在输出媒体播放列表文件名中的位置。字符串“%v”可能出现在文件名中或包含该文件的最后一个目录名中。如果字符串出现在目录名称中，则在扩展目录名称模式后创建子目录。这允许在子目录中创建变体流。

ffmpeg -re -i in.ts -b:v:0 1000k -b:v:1 256k -b:a:0 64k -b:a:1 32k \

-map 0:v -map 0:a -map 0:v -map 0:a -f hls -var\_stream\_map "v:0,a:0 v:1,a:1" \

http://example.com/live/out\_%v.m3u8

此示例创建两个 hls 变体流。第一个变体流将包含比特率 1000k 的视频流和比特率 64k 的音频流，第二个变体流将包含比特率 256k 的视频流和比特率 32k 的音频流。在这里，将创建文件名为 out\_0.m3u8 和 out\_1.m3u8 的两个媒体播放列表。如果您想要一些有意义的文本而不是结果名称中的索引，您可以为每个或某些变体指定名称，如以下示例所示。

ffmpeg -re -i in.ts -b:v:0 1000k -b:v:1 256k -b:a:0 64k -b:a:1 32k \

-map 0:v -map 0:a -map 0:v -map 0:a -f hls -var\_stream\_map "v:0,a:0,name:my\_hd v:1,a:1,name:my\_sd" \

http://example.com/live/out\_%v.m3u8

此示例与上一个示例一样创建两个 hls 变体流。但是在这里，将创建文件名为 out\_my\_hd.m3u8 和 out\_my\_sd.m3u8 的两个媒体播放列表。

ffmpeg -re -i in.ts -b:v:0 1000k -b:v:1 256k -b:a:0 64k \

-map 0:v -map 0:a -map 0:v -f hls -var\_stream\_map "v:0 a:0 v:1" \

http://example.com/live/out\_%v.m3u8

此示例创建三个 hls 变体流。第一个变体流将是视频比特率 1000k 的纯视频流，第二个变体流将是比特率 64k 的纯音频流，第三个变体流将是比特率 256k 的纯视频流。在这里，将创建文件名为 out\_0.m3u8、out\_1.m3u8 和 out\_2.m3u8 的三个媒体播放列表。

ffmpeg -re -i in.ts -b:v:0 1000k -b:v:1 256k -b:a:0 64k -b:a:1 32k \

-map 0:v -map 0:a -map 0:v -map 0:a -f hls -var\_stream\_map "v:0,a:0 v:1,a:1" \

http://example.com/live/vs\_%v/out.m3u8

此示例在子目录中创建变体流。在这里，第一个媒体播放列表是在http://example.com/live/vs\_0/out.m3u8创建的，第二个是在http://example.com/live/vs\_1/out.m3u8创建的。

ffmpeg -re -i in.ts -b:a:0 32k -b:a:1 64k -b:v:0 1000k -b:v:1 3000k \

-map 0:a -map 0:a -map 0:v -map 0:v -f hls \

-var\_stream\_map "a:0,agroup:aud\_low a:1,agroup:aud\_high v:0,agroup:aud\_low v:1,agroup:aud\_high" \

-master\_pl\_name master.m3u8 \

http://example.com/live/out\_%v.m3u8

此示例创建两个纯音频变体流和两个纯视频变体流。除了主播放列表中每个变体流的#EXT-X-STREAM-INF 标签之外，#EXT-X-MEDIA 标签也被添加到两个仅音频变体流，并将它们映射到两个仅视频变体流音频组名称为“aud\_low”和“aud\_high”。

默认情况下，会创建一个包含所有编码流的 hls 变体。

ffmpeg -re -i in.ts -b:a:0 32k -b:a:1 64k -b:v:0 1000k \

-map 0:a -map 0:a -map 0:v -f hls \

-var\_stream\_map "a:0,agroup:aud\_low,default:yes a:1,agroup:aud\_low v:0,agroup:aud\_low" \

-master\_pl\_name master.m3u8 \

http://example.com/live/out\_%v.m3u8

此示例创建两个纯音频变体流和一个纯视频变体流。除了主播放列表中每个变体流的#EXT-X-STREAM-INF 标签之外，#EXT-X-MEDIA 标签也被添加到两个仅音频变体流，并且它们被映射到一个仅视频变体流音频组名称为“aud\_low”，并且音频组的默认状态为“否”或“是”。

默认情况下，会创建一个包含所有编码流的 hls 变体。

ffmpeg -re -i in.ts -b:a:0 32k -b:a:1 64k -b:v:0 1000k \

-map 0:a -map 0:a -map 0:v -f hls \

-var\_stream\_map "a:0,agroup:aud\_low,default:yes,language:ENG a:1,agroup:aud\_low,language:CHN v:0,agroup:aud\_low" \

-master\_pl\_name master.m3u8 \

http://example.com/live/out\_%v.m3u8

此示例创建两个纯音频变体流和一个纯视频变体流。除了主播放列表中每个变体流的#EXT-X-STREAM-INF 标签之外，#EXT-X-MEDIA 标签也被添加到两个仅音频变体流，并且它们被映射到一个仅视频变体流音频组名称为“aud\_low”，音频组的默认状态为 NO 或 YES，一个音频的语言名为 ENG，另一种音频语言名为 CHN。

默认情况下，会创建一个包含所有编码流的 hls 变体。

ffmpeg -y -i input\_with\_subtitle.mkv \

-b:v:0 5250k -c:v h264 -pix\_fmt yuv420p -profile:v main -level 4.1 \

-b:a:0 256k \

-c:s webvtt -c:a mp2 -ar 48000 -ac 2 -map 0:v -map 0:a:0 -map 0:s:0 \

-f hls -var\_stream\_map "v:0,a:0,s:0,sgroup:subtitle" \

-master\_pl\_name master.m3u8 -t 300 -hls\_time 10 -hls\_init\_time 4 -hls\_list\_size \

10 -master\_pl\_publish\_rate 10 -hls\_flags \

删除片段+discont\_start+split\_by\_time ./tmp/video.m3u8

此示例在带有 webvtt 字幕组名称“subtitle”的主播放列表中添加带有TYPE=SUBTITLES的#EXT-X-MEDIA标签。请确保输入文件至少有一个文本字幕流。

cc\_stream\_map

指定不同隐藏式字幕组及其属性的映射字符串。隐藏式字幕流组由空格分隔。预期的字符串格式类似于“ccgroup:<group name>,instreamid:<INSTREAM-ID>,language:<language code> ....”。 'ccgroup' 和 'instreamid' 是强制属性。 “语言”是一个可选属性。通过在var\_stream\_map字符串中提供相同的“ccgroup”名称，使用此选项配置的隐藏式字幕组将映射到不同的变体流。如果未设置var\_stream\_map ，则将cc\_stream\_map中的第一个可用 ccgroup映射到输出变体流。下面给出了这两个用例的示例。

ffmpeg -re -i in.ts -b:v 1000k -b:a 64k -a53cc 1 -f hls \

-cc\_stream\_map "ccgroup:cc,instreamid:CC1,language:en" \

-master\_pl\_name master.m3u8 \

http://example.com/live/out.m3u8

此示例在组名为“cc”、语言为“en”（英语）和 INSTREAM-ID 为“CC1”的主播放列表中添加带有TYPE=CLOSED-CAPTIONS的#EXT-X-MEDIA标签。此外，它还为输出变体流添加组名为“cc”的CLOSED-CAPTIONS属性。

ffmpeg -re -i in.ts -b:v:0 1000k -b:v:1 256k -b:a:0 64k -b:a:1 32k \

-a53cc:0 1 -a53cc:1 1\

-map 0:v -map 0:a -map 0:v -map 0:a -f hls \

-cc\_stream\_map "ccgroup:cc,instreamid:CC1,language:en ccgroup:cc,instreamid:CC2,language:sp" \

-var\_stream\_map "v:0,a:0,ccgroup:cc v:1,a:1,ccgroup:cc" \

-master\_pl\_name master.m3u8 \

http://example.com/live/out\_%v.m3u8

此示例在 INSTREAM-ID“CC1”和“CC2”的主播放列表中添加两个带有TYPE=CLOSED-CAPTIONS的#EXT-X-MEDIA标签。此外，它还为两个输出变体流添加了组名为“cc”的CLOSED-CAPTIONS属性。

master\_pl\_name

使用给定名称创建 HLS 主播放列表。

ffmpeg -re -i in.ts -f hls -master\_pl\_name master.m3u8 http://example.com/live/out.m3u8

此示例创建名为 master.m3u8 的 HLS 主播放列表，并发布在 http://example.com/live/

master\_pl\_publish\_rate

每隔指定数量的段间隔重复发布主播放列表。

ffmpeg -re -i in.ts -f hls -master\_pl\_name master.m3u8 \

-hls\_time 2 -master\_pl\_publish\_rate 30 http://example.com/live/out.m3u8

此示例创建名为 master.m3u8 的 HLS 主播放列表，并每隔 30 段（即每 60 秒）重复发布一次。

http\_persistent

使用持久 HTTP 连接。仅适用于 HTTP 输出。

暂停

设置套接字 I/O 操作的超时时间。仅适用于 HTTP 输出。

-ignore\_io\_errors

在打开、写入和删除期间忽略 IO 错误。对于具有网络输出的长时间运行很有用。

标题

设置自定义 HTTP 标头，可以覆盖内置的默认标头。仅适用于 HTTP 输出。

### 21.18 图标

ICO 文件混合器。

Microsoft 的图标文件格式 (ICO) 有一些应注意的严格限制：

* 大小在任何维度上都不能超过 256 像素
* 只能存储BMP和PNG图片
* 如果使用 BMP 图片，它必须是以下像素格式之一：
* BMP 位深度 FFmpeg 像素格式
* 1bit pal8
* 4位pal8
* 8位pal8
* 16位RGB555LE
* 24 位 bgr24
* 32 位 bgra
* 如果使用 BMP 图像，则必须使用 BITMAPINFOHEADER DIB 标头
* 如果使用 PNG 图片，则必须使用 rgba 像素格式

### 21.19 图片 2

图像文件混合器。

图像文件混合器将视频帧写入图像文件。

输出文件名由模式指定，可用于生成按顺序编号的文件系列。该模式可能包含字符串“%d”或“%0 N d”，该字符串指定代表文件名中编号的字符的位置。如果使用“%0 N d”形式，表示每个文件名中数字的字符串将用 0 填充为N位数字。文字字符“%”可以在带有字符串“%%”的模式中指定。

如果模式包含“%d”或“%0 N d”，则指定的文件列表的第一个文件名将包含数字 1，后面的所有数字将是连续的。

该模式可能包含一个后缀，用于自动确定要写入的图像文件的格式。

例如，模式“img-%03d.bmp”将指定一系列文件名，格式为img-001.bmp 、 img-002.bmp 、...、 img-010.bmp等。模式“img% %-%d.jpg" 将指定一系列文件名，格式为img%-1.jpg 、 img%-2.jpg 、...、 img%-10.jpg等。

图像混合器支持 .YUV 图像文件格式。这种格式的特殊之处在于每个图像帧由三个文件组成，用于每个 YUV420P 组件。要读取或写入此图像文件格式，请指定“.Y”文件的名称。混合器将根据需要自动打开“.U”和“.V”文件。

* [选项](#Options-55)
* [例子](#Examples-15)

#### 21.19.1 选项

frame\_pts

如果设置为 1，则使用 pkt->pts 中的 pts 扩展文件名。默认值为 0。

起始编号

从指定编号开始序列。默认值为 1。

更新

如果设置为 1，文件名将始终被解释为文件名，而不是模式，相应的文件将不断被新图像覆盖。默认值为 0。

战斗时间

strftime()的日期和时间信息扩展文件名。默认值为 0。

原子写

将输出写入临时文件，写入完成后重命名为目标文件名。默认是禁用的。

protocol\_opts选项列表

将协议选项设置为以 : 分隔的键=值参数列表。包含:特殊字符的值必须进行转义。

#### 21.19.2 例子

以下示例展示了如何使用ffmpeg创建一系列文件img-001.jpeg 、 img-002.jpeg ……，每秒从输入视频中获取一张图像：

ffmpeg -i in.avi -vsync cfr -r 1 -f image2 'img-%03d.jpeg'

请注意，对于ffmpeg ，如果未使用-f选项指定格式并且输出文件名指定图像文件格式，则会自动选择 image2 muxer，因此前面的命令可以写为：

ffmpeg -i in.avi -vsync cfr -r 1 'img-%03d.jpeg'

另请注意，模式不一定包含“%d”或“%0 N d”，例如，要从输入视频的开头创建单个图像文件img.jpeg ，您可以使用以下命令：

ffmpeg -i in.avi -f image2 -frames:v 1 img.jpeg

strftime选项允许您使用日期和时间信息扩展文件名。检查strftime()函数的文档以了解语法。

例如，要从strftime() "%Y-%m-%d\_%H-%M-%S" 模式生成图像文件，可以使用以下ffmpeg命令：

ffmpeg -f v4l2 -r 1 -i /dev/video0 -f image2 -strftime 1 "%Y-%m-%d\_%H-%M-%S.jpg"

您可以使用当前帧的 PTS 设置文件名：

ffmpeg -f v4l2 -r 1 -i /dev/video0 -copyts -f image2 -frame\_pts true %d.jpg”

一个更复杂的示例是每秒将桌面内容直接发布到 WebDAV 服务器：

ffmpeg -f x11grab -framerate 1 -i :0.0 -q:v 6 -update 1 -protocol\_opts method=PUT http://example.com/desktop.jpg

### 21.20 马特罗斯卡

Matroska 容器混合器。

这个 muxer 实现了 matroska 和 webm 容器规范。

* [元数据](#Metadata)
* [选项](#Options-56)

#### 21.20.1 元数据

此 muxer 中可识别的元数据设置是：

标题

设置提供给单个曲目的标题名称。这将映射到作为附件写入的流的 FileDescription 元素。

语言

以 Matroska 语言形式指定轨道语言。

语言可以是 3 个字母的书目 ISO-639-2 (ISO 639-2/B) 形式（如法语的“fre”），或语言代码与语言专业的国家代码混合（如“fre- ca”代表加拿大法语）。

立体模式

在单个视频轨道中设置两个视图的立体 3D 视频布局。

识别以下值：

'单声道'

视频不是立体声

'左\_右'

两个视图并排排列，左眼视图在左侧

' bottom\_top '

两个视图都按上下方向排列，左眼视图在底部

' top\_bottom '

两个视图都按上下方向排列，左眼视图在顶部

' checkerboard\_rl '

每个视图都以棋盘交错的方式排列，左眼视图在第一位

' checkerboard\_lr '

每个视图都以棋盘交错的方式排列，右眼视图在第一位

' row\_interleaved\_rl '

每个视图由基于行的交错构成，右眼视图是第一行

' row\_interleaved\_lr '

每个视图由基于行的交错构成，左眼视图是第一行

' col\_interleaved\_rl '

两个视图都以基于列的交错方式排列，右眼视图是第一列

' col\_interleaved\_lr '

两个视图都以基于列的交错方式排列，左眼视图是第一列

' anaglyph\_cyan\_red '

所有帧均为浮雕格式，可通过红青色滤镜查看

'右\_左'

两个视图并排排列，右眼视图在左侧

' anaglyph\_green\_magenta '

所有帧均采用浮雕格式，可通过绿色-洋红色滤镜查看

' block\_lr '

双眼并排，左眼优先

' block\_rl '

双眼并排，右眼优先

例如，可以使用以下命令行创建 3D WebM 剪辑：

ffmpeg -i sample\_left\_right\_clip.mpg -an -c:v libvpx -metadata stereo\_mode=left\_right -y stereo\_clip.webm

#### 21.20.2 选项

此多路复用器支持以下选项：

保留索引空间

默认情况下，此 muxer 在文件末尾写入用于搜索的索引（在 Matroska 术语中称为 cues），因为它无法提前知道在文件开头为索引留出多少空间。然而，对于某些用例——例如，可以进行搜索但速度很慢的流式传输——将索引放在文件的开头是很有用的。

如果此选项设置为非零值，则 muxer 将在文件头中保留给定的空间量，然后在 muxing 完成时尝试将提示写入那里。如果预留空间不够，则不会写入任何 Cues，文件将完成，写入预告片将返回错误。大多数用例的安全大小应该是每小时视频 50kB 左右。

请注意，仅当输出可搜索时才会写入提示，否则此选项将无效。

提示到前面

如果设置，则 muxer 将在必要时通过移动主要数据将索引写入文件的开头。这可以与 reserve\_index\_space 结合使用，在这种情况下，仅当最初保留的空间不足时才移动数据。

如果无法找到输出，则忽略此选项。

默认模式

此选项控制如何设置输出轨道的 FlagDefault。它会影响播放器默认播放的曲目。默认模式是“ passthrough ”。

'推断'

每个具有处置默认值的轨道都将设置 FlagDefault。此外，对于每种类型的轨道（音频、视频或字幕），如果不存在具有该类型默认配置的轨道，则该类型的第一个轨道将被标记为默认（如果存在）。这确保以合理的方式设置默认标志，即使输入源自缺乏默认轨道概念的容器。

' infer\_no\_subs '

此模式与 infer 相同，只是如果不存在配置默认的字幕轨道，则不会将字幕轨道标记为默认。

'直通'

在此模式下，当且仅当在相应流的处置中设置了 AV\_DISPOSITION\_DEFAULT 标志时，才会设置 FlagDefault。

flipped\_raw\_rgb

如果设置为 true，则存储原始 RGB 位图的正高度，这表示位图是自下而上存储的。请注意，此选项不会翻转必须事先手动完成的位图，例如使用 vflip 过滤器。默认为false ，表示位图自上而下存储。

### 21.21 MD5

MD5 测试格式。

[哈希混合器](#hash)的变体。与那个 muxer 不同，它默认使用 MD5 哈希函数。

* [例子](#Examples-16)

#### 21.21.1 例子

要计算转换为原始音频和视频的输入的 MD5 哈希，并将其存储在文件out.md5中：

ffmpeg -i 输入 -f md5 输出.md5

您可以使用以下命令将 MD5 打印到标准输出：

ffmpeg -i 输入 -f md5 -

另请参阅[哈希](#hash)和[framemd5](#framemd5)混合器。

### 21.22 mov、mp4、ismv

MOV/MP4/ISMV（平滑流式传输）混合器。

mov/mp4/ismv muxer 支持分片。通常，一个 MOV/MP4 文件将所有数据包的所有元数据存储在一个位置（写在文件末尾，可以通过将faststart添加到movflags或使用qt-faststart将其移动到开头以便更好地播放工具）。分片文件由许多片段组成，其中数据包和有关这些数据包的元数据存储在一起。写入碎片文件的优点是即使写入中断文件也可解码（而正常的 MOV/MP4 如果未正确完成则无法解码），并且在写入非常长的文件时需要较少的内存（因为写入正常的 MOV /MP4 文件在文件关闭之前将有关每个数据包的信息存储在内存中）。缺点是它与其他应用程序的兼容性较差。

* [选项](#Options-57)
* [例子](#Example)

#### 21.22.1 选项

通过设置定义如何将文件切割成碎片的 AVOptions 之一来启用碎片：

-moov\_size字节

在文件开头为 moov 原子保留空间，而不是将 moov 原子放在末尾。如果预留空间不足，muxing 将失败。

-movflags frag\_keyframe

在每个视频关键帧处开始一个新片段。

-frag\_duration持续时间

创建持续时间为微秒的片段。

-frag\_size大小

创建包含最多size个有效负载数据字节的片段。

-movflags frag\_custom

允许调用者手动选择何时切割片段，方法是调用av\_write\_frame(ctx, NULL)以写入包含目前写入的数据包的片段。 （这仅对集成 libavformat 的其他应用程序有用，对ffmpeg无效。）

-min\_frag\_duration持续时间

不要创建短于持续时间微秒的片段。

如果指定了多个条件，则在满足指定条件之一时切割片段。例外情况是-min\_frag\_duration ，必须满足它才能应用任何其他条件。

此外，可以通过其他几个选项调整输出文件的写入方式：

-movflags empty\_moov

直接在文件的开头写一个初始的 moov 原子，而不描述其中的任何样本。通常，一个 mdat/moov 对写在文件的开头，作为一个普通的 MOV/MP4 文件，只包含文件的一小部分。设置此选项后，没有初始 mdat 原子，moov 原子仅描述轨道但持续时间为零。

在写入 ismv（平滑流）文件时隐式设置此选项。

-movflags separate\_moof

为每个轨道写一个单独的 moof（电影片段）原子。通常，所有轨道的数据包都写入一个 moof 原子（效率稍高），但设置此选项后，复用器会为每个轨道写入一对 moof/mdat，从而更容易分离轨道。

在写入 ismv（平滑流）文件时隐式设置此选项。

-movflags skip\_sidx

跳过 sidx 原子的写入。当 sidx 原子导致的比特率开销很高时，此选项可用于 sidx 原子不是强制性的情况。当启用 global\_sidx 标志时，此选项将被忽略。

-movflags 快速启动

运行第二遍，将索引（moov 原子）移动到文件的开头。该操作可能需要一段时间，并且在输出碎片等各种情况下将不起作用，因此默认不启用。

-movflags rtphint

将 RTP 提示轨道添加到输出文件。

-movflags disable\_chpl

禁用 Nero 章节标记（chpl 原子）。通常，Nero 章节和 QuickTime 章节轨道都会写入该文件。设置此选项后，将仅写入 QuickTime 章节轨道。当使用某些标记程序（如 mp3Tag 2.61a 和 iTunes 11.3）重新处理文件时，Nero 章节可能会导致失败，很可能其他版本也会受到影响。

-movflags omit\_tfhd\_offset

不要在 tfhd 原子中写入任何绝对 base\_data\_offset。这避免了将片段绑定到文件/流中的绝对字节位置。

-movflags default\_base\_moof

与 omit\_tfhd\_offset 类似，此标志避免在 tfhd 原子中写入绝对 base\_data\_offset 字段，而是通过使用新的 default-base-is-moof 标志来实现。此标志是 14496-12:2012 的新标志。这可能使片段在某些情况下更容易解析（避免将轨道片段位置计算基于前一个轨道片段的隐式结束）。

-write\_tmcd

指定on强制写入时间码轨道， off禁用它， auto只为 mov 和 mp4 输出写入时间码轨道（默认）。

-movflags negative\_cts\_offsets

允许使用 CTTS 盒的版本 1，其中 CTS 偏移量可以为负。这使得初始样本的 DTS/CTS 为零，并减少了某些情况下对编辑列表的需求，例如带有 B 帧的视频轨道。此外，简化了对 DASH-IF 互操作性指南的遵守。

在写入 ismv（平滑流）文件时隐式设置此选项。

-write\_btrt布尔值

强制或禁用在轨道的 stsd 框内写入比特率框。该框包含轨道的解码缓冲区大小（以字节为单位）、最大比特率和平均比特率。如果无法计算这些值，将跳过该框。默认为-1或auto ，仅在 MP4 模式下写入框。

-write\_prft

为 PRFT 框中的 NTP 字段写入具有指定时间源的生产者时间参考框 (PRFT)。将值设置为“ wallclock ”以将时间源指定为挂钟时间，将“ pts ”设置为将时间源指定为输入数据包的 PTS 值。

将值设置为“ pts ”仅适用于实时编码用例，其中 PTS 值设置为源的挂钟时间。例如，带有 decklink 捕获源的编码用例，其中video\_pts和audio\_pts设置为“ abs\_wallclock ”。

-empty\_hdlr\_name布尔

hdlr框中写入名称。默认为假。

-movie\_timescale刻度

设置写在电影标题框 ( mvhd ) 中的时间刻度。范围是 1 到 INT\_MAX。默认值为 1000。

-video\_track\_timescale刻度

设置用于视频轨道的时间刻度。范围是 0 到 INT\_MAX。如果设置为0 ，时间刻度将根据本机流时基自动设置。默认为 0。

#### 21.22.2 例子

可以使用此混合器将平滑流内容实时推送到 IIS 上的发布点。例子：

ffmpeg -re <正常输入/转码选项> -movflags isml+frag\_keyframe -f ismv http://server/publishingpoint.isml/Streams(Encoder1)

### 21.23 mp3

MP3 多路复用器写入具有以下可选功能的原始 MP3 流：

* 开头的 ID3v2 元数据标头（默认启用）。支持版本 2.3 和 2.4， id3v2\_version私有选项控制使用哪一个（3 或 4）。将id3v2\_version设置为 0 会完全禁用 ID3v2 标头。

muxer 支持将附加图片（APIC 帧）写入 ID3v2 标头。图片以带有单个数据包的视频流的形式提供给复用器。可以有任意数量的流，每个都对应一个 APIC 帧。流元数据标签title和comment分别映射到 APIC描述和图片类型。有关允许的图片类型，请参阅<http://id3.org/id3v2.4.0-frames>。

请注意，APIC 帧必须写在开头，因此 muxer 将缓冲音频帧，直到它获取所有图片。因此，建议尽快提供图片以避免过度缓冲。

* 在 ID3v2 标头（如果存在）之后的 Xing/LAME 帧。它默认启用，但只有在输出可搜索时才会写入。 write\_xing私有选项可用于禁用它。该帧包含可能对解码器有用的各种信息，例如音频持续时间或编码器延迟。
* 文件末尾的旧版 ID3v1 标记（默认情况下禁用）。它可以通过write\_id3v1私有选项启用，但由于其功能非常有限，因此不推荐使用。

例子：

编写一个带有 ID3v2.3 标题和 ID3v1 页脚的 mp3：

ffmpeg -i 输入 -id3v2\_version 3 -write\_id3v1 1 out.mp3

地图的音频和图片流：

ffmpeg -i input.mp3 -i cover.png -c copy -map 0 -map 1

-metadata:s:v title="专辑封面" -metadata:s:v comment="封面（正面）" out.mp3

编写一个没有任何额外功能的“干净”MP3：

ffmpeg -i 输入.wav -write\_xing 0 -id3v2\_version 0 输出.mp3

### 21.24 mpegts

MPEG 传输流复用器。

该多路复用器执行 ISO 13818-1 和 ETSI EN 300 468 的一部分。

mpegts muxer 中可识别的元数据设置是service\_provider和service\_name 。如果未设置，则service\_provider的默认值为“ FFmpeg ”， service\_name的默认值为“ Service01 ”。

* [选项](#Options-58)
* [例子](#Example-1)

#### 21.24.1 选项

复用器选项是：

mpegts\_transport\_stream\_id整数

设置“ transport\_stream\_id ”。这标识 DVB 中的转发器。默认为0x0001 。

mpegts\_original\_network\_id整数

设置“ original\_network\_id ”。这是 DVB 中网络的唯一标识符。它的主要用途是通过路径“ Original\_Network\_ID, Transport\_Stream\_ID ”来唯一标识服务。默认为0x0001 。

mpegts\_service\_id整数

设置“ service\_id ”，在 DVB 中也称为节目。默认为0x0001 。

mpegts\_service\_type整数

设置程序“ service\_type ”。默认为digital\_tv 。接受以下选项：

'十六进制值'

ETSI 300 468 中定义的0x01和0xff之间的任何十六进制值。

'数字电视'

数字电视服务。

'数字收音机'

数字广播服务。

'图文电视'

图文服务。

' advanced\_codec\_digital\_radio '

高级编解码器数字广播服务。

' mpeg2\_digital\_hdtv '

MPEG2 数字高清电视服务。

' advanced\_codec\_digital\_sdtv '

高级编解码器数字标清电视服务。

' advanced\_codec\_digital\_hdtv '

高级编解码器数字高清电视服务。

mpegts\_pmt\_start\_pid整数

为 PMT 设置第一个 PID。默认为0x1000 ，最小为0x0020 ，最大为0x1ffa 。此选项在 PMT PID 固定为0x0100的 m2ts 模式下无效。

mpegts\_start\_pid整数

设置基本流的第一个 PID。默认为0x0100 ，最小为0x0020 ，最大为0x1ffa 。此选项在基本流 PID 固定的 m2ts 模式中无效。

mpegts\_m2ts\_mode布尔值

1则启用 m2ts 模式。默认值为-1 ，即禁用 m2ts 模式。

muxrate整数

设置一个常数 muxrate。默认为 VBR。

pes\_payload\_size整数

以字节为单位设置最小 PES 数据包负载。默认值为2930 。

mpegts\_flags标志

设置 mpegts 标志。接受以下选项：

'重新发送标题'

在写入下一个数据包之前重新发送 PAT/PMT。

'拉美'

对 AAC 使用 LATM 打包。

' pat\_pmt\_at\_frames '

在每个视频帧重新发送 PAT 和 PMT。

'系统\_b '

符合系统 B (DVB) 而不是系统 A (ATSC)。

' initial\_discontinuity '

将每个流的初始数据包标记为不连续。

'尼特'

发出 NIT 表。

' omit\_rai '

禁用随机访问指示器的写入。

mpegts\_copyts布尔值

如果值设置为1 ，则保留原始时间戳。默认值为-1 ，这会导致移动时间戳，以便它们从 0 开始。

omit\_video\_pes\_length布尔值

省略视频数据包的 PES 数据包长度。默认值为1 （真）。

pcr\_period整数

以毫秒为单位覆盖默认的 PCR 重传时间。默认值为-1 ，这意味着 PCR 间隔将自动确定：20 ms 用于 CBR 流，帧持续时间的最高倍数小于 100 ms 用于 VBR 流。

pat\_period持续时间

PAT/PMT 表之间的最长时间（以秒为单位）。默认值为0.1 。

sdt\_period持续时间

SDT 表之间的最长时间（以秒为单位）。默认值为0.5 。

nit\_period持续时间

NIT 表之间的最长时间（以秒为单位）。默认值为0.5 。

tables\_version整数

设置 PAT、PMT、SDT 和 NIT 版本（默认为0 ，有效值从 0 到 31，包括在内）。此选项允许更新流结构，以便标准消费者可以检测到更改。为此，重新打开输出AVFormatContext （在使用 API 的情况下）或重新启动ffmpeg实例，循环更改tables\_version值：

ffmpeg -i source1.ts -codec copy -f mpegts -tables\_version 0 udp://1.1.1.1:1111

ffmpeg -i source2.ts -codec copy -f mpegts -tables\_version 1 udp://1.1.1.1:1111

...

ffmpeg -i source3.ts -codec copy -f mpegts -tables\_version 31 udp://1.1.1.1:1111

ffmpeg -i source1.ts -codec copy -f mpegts -tables\_version 0 udp://1.1.1.1:1111

ffmpeg -i source2.ts -codec copy -f mpegts -tables\_version 1 udp://1.1.1.1:1111

...

#### 21.24.2 例子

ffmpeg -i 文件.mpg -c 复制\

-mpegts\_original\_network\_id 0x1122 \

-mpegts\_transport\_stream\_id 0x3344 \

-mpegts\_service\_id 0x5566 \

-mpegts\_pmt\_start\_pid 0x1500 \

-mpegts\_start\_pid 0x150 \

-metadata service\_provider="一些提供者" \

-metadata service\_name="一些频道" \

输出.ts

### 21.25 mxf，mxf\_d10，mxf\_opatom

MXF 混合器。

* [选项](#Options-59)

#### 21.25.1 选项

复用器选项是：

store\_user\_comments布尔

设置是否应存储用户评论（如果可用）或从不存储。 IRT D-10 不允许用户评论。因此默认是为 mxf 和 mxf\_opatom 而不是为 mxf\_d10 编写它们

### 21.26 空

空混合器。

此 muxer 不生成任何输出文件，它主要用于测试或基准测试目的。

例如，要使用ffmpeg进行基准解码，您可以使用以下命令：

ffmpeg -benchmark -i INPUT -f null out.null

注意上面的命令并没有读写out.null文件，但是指定输出文件是ffmpeg语法要求的。

或者，您可以将命令编写为：

ffmpeg -benchmark -i 输入 -f null -

### 21.27 螺母

-同步点标志

更改 nut 中的同步点用法：

默认使用正常的低开销搜索辅助工具。

none根本不使用同步点，减少了开销但使流不可搜索；

不建议使用此选项，因为生成的文件对损坏非常敏感并且无法进行搜索。通常，同步点的开销也可以忽略不计。请注意，- write\_index 0 可用于禁用所有不断增长的数据表，允许在内存有限且没有这些缺点的情况下混合无穷无尽的流。

timestamped使用 wallclock 字段扩展同步点。

none和timestamped标志是实验性的。

-write\_index布尔

最后写索引，默认是写索引。

ffmpeg -i INPUT -f\_strict 实验性 -syncpoints 无 - |处理器

### 21.28 奥格

Ogg 容器混合器。

-page\_duration持续时间

首选页面持续时间，以微秒为单位。 muxer 将尝试创建大约duration微秒长的页面。这允许用户在查找粒度和容器开销之间进行折衷。默认值为 1 秒。值为 0 将填充所有段，使页面尽可能大。在大多数情况下，值为 1 将有效地使用每页 1 个数据包，以额外的容器开销为代价提供较小的查找粒度。

-serial\_offset值

从中设置流序列号的序列值。将其设置为不同且足够大的值可确保生成的 ogg 文件可以安全链接。

### 21.29 原始混合器

原始混合器接受与指定编解码器匹配的单个流。它们不存储时间戳或元数据。除非另有说明，否则识别的扩展名与 muxer 名称相同。

* [ac3](#ac3-1)
* [adx](#adx)
* [aptx](#aptx)
* [aptx\_hd](#aptx_005fhd)
* [AV2](#avs2)
* [骑士视频](#cavsvideo)
* [编解码器2raw](#codec2raw)
* [数据](#data)
* [狄拉克](#dirac)
* [dnxhd](#dnxhd)
* [数据传输系统](#dts)
* [eac3](#eac3)
* [g722](#g722)
* [g723\_1](#g723_005f1)
* [G726](#g726)
* [g726le](#g726le)
* [全球移动通信系统](#gsm)
* [h261](#h261)
* [h263](#h263)
* [h264](#h264)
* [高压](#hevc)
* [m4v](#m4v)
* [mjpeg](#mjpeg-1)
* [mlp](#mlp)
* [mp2](#mp2)
* [mpeg1视频](#mpeg1video)
* [mpeg2视频](#mpeg2video)
* [奥布](#obu)
* [原始视频](#rawvideo-2)
* [南北广播公司](#sbc)
* [真高清](#truehd)
* [VC1](#vc1)

#### 21.29.1 ac3

杜比数字，也称为 AC-3，音频。

#### 21.29.2 adx

CRI 中间件 ADX 音频。

当输出可搜索时，此多路复用器将在第一个数据包开始附近写出总样本计数，并且计数可以存储在 32 位中。

#### 21.29.3 aptx

aptX（蓝牙音频处理技术）音频。

#### 21.29.4 aptx\_hd

aptX HD（蓝牙音频处理技术）音频。

扩展：aptxhd

#### 21.29.5 avs2

AVS2-P2/IEEE1857.4 视频。

扩展名：avs、avs2

#### 21.29.6 cavsvideo

中文AVS（音频视频标准）视频。

扩展名：骑士

#### 21.29.7 编解码器 2raw

编解码器 2 音频。

没有注册扩展名，因此必须提供格式名称，例如使用 ffmpeg CLI 工具-f codec2raw 。

#### 21.29.8数据

数据复用器接受具有任何类型的任何编解码器的单个流。必须使用ffmpeg CLI 工具的-map选项来选择输入流。

没有注册扩展名，因此必须提供格式名称，例如使用 ffmpeg CLI 工具-f 数据。

#### 21.29.9 狄拉克

BBC狄拉克视频。 Dirac Pro 编解码器是一个子集，被标准化为 SMPTE VC-2。

扩展名：drc、vc2

#### 21.29.10 dnxhd

狂热的 DNxHD 视频。它被标准化为 SMPTE VC-3。接受 DNxHR 流。

扩展名：dnxhd、dnxhr

#### 21.29.11 dts

DTS 相干声学 (DCA) 音频。

#### 21.29.12 eac3

杜比数字+，也称为增强型 AC-3，音频。

#### 21.29.13 g722

ITU-T G.722 音频。

#### 21.29.14 g723\_1

ITU-T G.723.1 音频。

扩展名：tco、rco

#### 21.29.15 g726

ITU-T G.726 大端（“左对齐”）音频。

没有注册扩展名，因此必须提供格式名称，例如使用 ffmpeg CLI 工具-f g726 。

#### 21.29.16 g726le

ITU-T G.726 little-endian（“右对齐”）音频。

没有注册扩展名，因此必须提供格式名称，例如使用 ffmpeg CLI 工具-f g726le 。

#### 21.29.17 gsm

全球移动通信音频系统。

#### 21.29.18 h261

ITU-T H.261 视频。

#### 21.29.19 h263

ITU-T H.263 / H.263-1996、H.263+ / H.263-1998 / H.263 版本 2 视频。

#### 21.29.20 h264

ITU-T H.264 / MPEG-4 第 10 部分 AVC 视频。如果比特流处于长度前缀模式，则应将其转换为附件 B 语法。

扩展名：h264、264

#### 21.29.21 hevc

ITU-T H.265 / MPEG-H 第 2 部分 HEVC 视频。如果比特流处于长度前缀模式，则应将其转换为附件 B 语法。

扩展：hevc、h265、265

#### 21.29.22 m4v

MPEG-4 第 2 部分视频。

#### 21.29.23 mjpeg

动态 JPEG 视频。

扩展名：mjpg、mjpeg

#### 21.29.24 mlp

Meridian Lossless Packing，也称为Packed PCM，音频。

#### 21.29.25 mp2

MPEG-1 音频层 II 音频。

扩展名：mp2、m2a、mpa

#### 21.29.26 mpeg1 视频

MPEG-1 第 2 部分视频。

扩展名：mpg、mpeg、m1v

#### 21.29.27 mpeg2video

ITU-T H.262 / MPEG-2 第 2 部分视频。

扩展：m2v

#### 21.29.28 奥布

AV1 低开销开放比特流单元复用器。时间定界符 OBU 将被插入到流的所有时间单元中。

#### 21.29.29 原始视频

原始未压缩视频。

扩展：yuv、rgb

#### 21.29.30 标准广播公司

Bluetooth SIG 低复杂度子带编解码器音频。

扩展名：sbc、msbc

#### 21.29.31 真高清

杜比 TrueHD 音频。

扩展名：thd

#### 21.29.32 vc1

SMPTE 421M / VC-1 视频。

### 21.30 段，stream\_segment，ssegment

基本流分段器。

此多路复用器将流输出到许多持续时间几乎固定的独立文件。可以以类似于[image2 的方式设置输出文件名模式](#image2)，或者如果启用了strftime选项，则可以使用strftime模板。

stream\_segment是用于写入流输出格式的多路复用器的变体，即不需要全局标头，建议用于输出，例如 MPEG 传输流段。 ssegment是stream\_segment的较短别名。

每个片段都以所选参考流的关键帧开始，该参考流是通过reference\_stream选项设置的。

请注意，如果要对视频文件进行精确分割，则需要使输入关键帧与分段器预期的精确分割时间相对应，否则分段混合器将使用指定开始后的下一个关键帧开始新分段时间。

分段复用器最适合单个恒定帧速率视频。

segment\_list来生成已创建段的列表。列表类型由segment\_list\_type选项指定。段列表中的条目文件名默认设置为相应段文件的基本名称。

另请参阅[hls](#hls) muxer，它为 HLS 分段提供了更具体的实现。

* [选项](#Options-60)
* [例子](#Examples-17)

#### 21.30.1 选项

段复用器支持以下选项：

增量\_tc 1|0

如果设置为1 ，则在每个段之间增加时间码如果选择此项，则输入需要在第一个视频流中具有时间码。默认值为0 。

reference\_stream说明符

设置引用流，由字符串说明符指定。如果说明符设置为auto ，则自动选择引用。否则它必须是指定参考流的流说明符（请参阅 ffmpeg 手册中的“流说明符”一章）。默认值为自动。

segment\_format格式

覆盖内部容器格式，默认情况下由文件扩展名猜测。

segment\_format\_options选项列表

使用 : 分隔的 key=value 参数列表设置输出格式选项。包含:特殊字符的值必须进行转义。

段列表名称

还生成一个名为name的列表文件。如果未指定，则不会生成列表文件。

segment\_list\_flags标志

设置影响段列表生成的标志。

它目前支持以下标志：

'缓存'

允许缓存（仅影响 M3U8 列表文件）。

'现场'

允许实时友好的文件生成。

segment\_list\_size大小

更新列表文件，使其最多包含size段。如果为 0，列表文件将包含所有段。默认值为 0。

segment\_list\_entry\_prefix前缀

添加前缀。用于生成绝对路径。默认情况下不应用前缀。

segment\_list\_type类型

选择列表格式。

识别以下值：

'平'

为创建的段生成一个平面列表，每行一个段。

' csv，分机'

为创建的段生成一个列表，每行一个段，每行匹配格式（逗号分隔值）：

段文件名、段开始时间、段结束时间

segment\_filename是 muxer 根据提供的模式生成的输出文件的名称。如果需要，将应用 CSV 转义（根据 RFC4180）。

segment\_start\_time和segment\_end\_time指定以秒为单位的段开始和结束时间。

后缀为“.csv”或“.ext”的列表文件将自动选择此格式。

' ext ' 已弃用，取而代之的是 ' csv '。

' ffconcat '

为创建的段生成一个 ffconcat 文件。可以使用 FFmpeg [concat demuxer](#concat)读取生成的文件。

后缀为“.ffcat”或“.ffconcat”的列表文件将自动选择此格式。

' m3u8 '

生成一个扩展的 M3U8 文件，版本 3，符合<http://tools.ietf.org/id/draft-pantos-http-live-streaming>。

后缀为“.m3u8”的列表文件将自动选择此格式。

如果未指定，则从列表文件名后缀中猜测类型。

segment\_time时间

将段持续时间设置为time ，该值必须是持续时间规范。默认值为“2”。另请参阅segment\_times选项。

请注意，拆分可能不准确，除非您在给定时间强制参考流关键帧。请参阅下面的介绍性通知和示例。

min\_seg\_duration时间

将最短段持续时间设置为time ，该值必须是持续时间规范。这可以防止多路复用器以低于该值的持续时间结束段。仅对segment\_time有效。默认值为“0”。

segment\_atclocktime 1|0

如果设置为“1”，则从 00:00 点开始以规则的时钟时间间隔拆分。 segment\_time中指定的时间值用于设置分割间隔的长度。

例如，将segment\_time设置为“900”可以在 12:00、12:15、12:30 等创建文件。

默认值为“0”。

segment\_clocktime\_offset持续时间

segment\_atclocktime时，将段分割时间延迟指定的持续时间。

例如，将segment\_time设置为“900”，将 segment\_clocktime\_offset设置为“300”，这样就可以在 12:05、12:20、12:35 等创建文件。

默认值为“0”。

segment\_clocktime\_wrap\_duration持续时间

如果数据包在分段时钟时间后的指定持续时间内到达复用器，则强制分段器仅开始新分段。通过这种方式，您可以使分段器对向后的本地时间跳跃更具弹性，例如闰秒或从夏令时过渡到标准时间。

默认值是最大可能持续时间，这意味着开始一个新段，而不管自上一个时钟时间以来经过的时间。

segment\_time\_delta增量

在选择段的开始时间时指定准确时间，表示为持续时间规范。默认值为“0”。

当指定增量时，如果关键帧的 PTS 满足以下关系，它将开始一个新的片段：

PTS >= start\_time - time\_delta

此选项在分割视频内容时很有用，视频内容始终在 GOP 边界处分割，以防在指定的分割时间之前找到关键帧。

特别是可以与ffmpeg选项force\_key\_frames结合使用。由于舍入问题，由force\_key\_frames指定的关键帧时间可能设置不准确，结果可能会在指定时间之前设置关键帧时间。对于恒定帧率视频，值 1/(2\* frame\_rate ) 应该解决指定时间与force\_key\_frames设置的时间之间的最坏情况不匹配。

segment\_times次

指定分割点列表。 times包含以逗号分隔的持续时间规范列表，按递增顺序排列。另请参阅segment\_time选项。

segment\_frames帧

指定分割视频帧数列表。 frames包含以逗号分隔的整数列表，按升序排列。

此选项指定在找到参考流关键帧并且帧的序号（从 0 开始）大于或等于列表中的下一个值时开始新段。

segment\_wrap限制

limit就环绕段索引。

segment\_start\_number编号

设置第一段的序号。默认为0 。

strftime 1|0

使用strftime函数定义要写入的新段的名称。如果选择此项，输出段名称必须包含一个strftime函数模板。默认值为0 。

break\_non\_keyframes 1|0

如果启用，则允许段从关键帧以外的帧开始。当关键帧之间的时间不一致时，这会改善某些玩家的行为，但可能会使其他人的情况变得更糟，并可能在搜索过程中导致一些奇怪的现象。默认为0 。

重置时间戳1|0

在每个段的开头重置时间戳，以便每个段都以接近零的时间戳开始。它旨在简化生成的片段的播放。可能不适用于混合器/编解码器的某些组合。默认设置为0 。

initial\_offset偏移量

指定时间戳偏移量以应用于输出数据包时间戳。参数必须是持续时间规范，默认为 0。

write\_empty\_segments 1|0

如果启用，如果在段通常跨越的时间段内没有数据包，则写入一个空段。否则，该段将被写入的下一个数据包填充。默认为0 。

确保在编码时需要一个封闭的 GOP，并设置 GOP 大小以适应您的分段时间限制。

#### 21.30.2 例子

* in.mkv的内容重新混合到out-000.nut 、 out-001.nut等段列表中，并将生成的段列表写入out.list ：
* ffmpeg -i in.mkv -codec hevc -flags +cgop -g 60 -map 0 -f segment -segment\_list out.list out%03d.nut
* 段输入并为输出段设置输出格式选项：
* ffmpeg -i in.mkv -f segment -segment\_time 10 -segment\_format\_options movflags=+faststart out%03d.mp4
* 根据segment\_times选项指定的分割点对输入文件进行分割：
* ffmpeg -i in.mkv -codec copy -map 0 -f segment -segment\_list out.csv -segment\_times 1,2,3,5,8,13,21 out%03d.nut
* 使用ffmpeg force\_key\_frames选项强制指定位置输入中的关键帧，与段选项segment\_time\_delta一起考虑设置关键帧时间时可能操作的舍入。
* ffmpeg -i in.mkv -force\_key\_frames 1,2,3,5,8,13,21 -codec:v mpeg4 -codec:a pcm\_s16le -map 0 \
* -f segment -segment\_list out.csv -segment\_times 1,2,3,5,8,13,21 -segment\_time\_delta 0.05 out%03d.nut

为了在输入文件上强制关键帧，需要进行转码。

* segment\_frames选项指定的帧编号序列分割输入文件来分割输入文件：
* ffmpeg -i in.mkv -codec copy -map 0 -f segment -segment\_list out.csv -segment\_frames 100,200,300,500,800 out%03d.nut
* libx264和aac编码器将in.mkv转换为 TS 片段：
* ffmpeg -i in.mkv -map 0 -codec:v libx264 -codec:a aac -f ssegment -segment\_list out.list out%03d.ts
* 分割输入文件，创建M3U8直播播放列表（可作为直播HLS源）：
* ffmpeg -re -i in.mkv -codec copy -map 0 -f segment -segment\_list playlist.m3u8 \
* -segment\_list\_flags +live -segment\_time 10 out%03d.mkv

### 21.31 平滑流

Smooth Streaming muxer 生成一组适用于传统 Web 服务器的文件（清单、块）。

窗口大小

指定清单中保留的片段数。默认 0（保留所有）。

extra\_window\_size

指定在从磁盘中删除之前保留在清单之外的片段数。默认 5。

前瞻计数

指定先行片段的数量。默认 2。

min\_frag\_duration

指定最短片段持续时间（以微秒为单位）。默认 5000000。

remove\_at\_exit

指定是否在完成时删除所有片段。默认 0（不删除）。

### 21.32 流哈希

每个流哈希测试格式。

该多路复用器以每个流为基础计算并打印所有输入帧的加密哈希。这可用于相等性检查，而无需进行完整的二进制比较。

默认情况下，在计算哈希值之前，音频帧被转换为带符号的 16 位原始音频，视频帧被转换为原始视频，但也可以使用显式转换为其他编解码器的输出。时间戳被忽略。它默认使用 SHA-256 加密哈希函数，但支持其他几种算法。

muxer 的输出由以下形式的每个流一行组成： streamindex , streamtype , algo = hash ，其中streamindex是映射流的索引， streamtype是指示流类型的单个字符， algo是一个短字符串，表示使用的散列函数，散列是表示计算散列的十六进制数。

哈希算法

算法指定的加密哈希函数。支持的值包括MD5 、 murmur3 、 RIPEMD128 、 RIPEMD160 、 RIPEMD256 、 RIPEMD320 、 SHA160 、 SHA224 、 SHA256 （默认）、 SHA512/224 、 SHA512/256 、 SHA384 、 SHA512 、 CRC32和adler32 。

* [例子](#Examples-18)

#### 21.32.1 例子

要计算转换为原始音频和视频的输入的 SHA-256 哈希值，并将其存储在文件out.sha256中：

ffmpeg -i 输入 -f streamhash out.sha256

要将 MD5 哈希打印到标准输出，请使用以下命令：

ffmpeg -i 输入 -f streamhash -hash md5 -

另请参阅[哈希](#hash)和[帧哈希](#framehash)混合器。

### 21.33 发球台

tee muxer 可用于将相同的数据写入多个输出，例如文件或流。例如，它可用于通过网络传输视频并同时将其保存到磁盘。

它不同于为ffmpeg命令行工具指定多个输出。使用 tee muxer，音频和视频数据将只编码一次。对于传统的多输出，并行启动多个编码操作，这可能是一个非常昂贵的过程。当直接使用 libavformat API 时，tee muxer 没有用，因为它可以直接将相同的数据包提供给多个 muxer。

由于 tee muxer 不代表任何特定的输出格式，ffmpeg 无法自动选择输出流。因此，所有用于输出的流都必须使用-map指定。请参阅下面的示例。

一些编码器可能需要不同的选项，具体取决于输出格式；对此的自动检测不能与 tee muxer 一起使用，因此需要明确指定它们。主要示例是global\_header标志。

从属输出在提供给多路复用器的文件名中指定，以“|”分隔。如果任何从名称包含'|'分隔符、前导或尾随空格或任何特殊字符，它们必须被转义（请参阅[（ffmpeg-utils）ffmpeg-utils(1) 手册中的“引用和转义”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#quoting_005fand_005fescaping)）。

* [选项](#Options-61)
* [例子](#Examples-19)

#### 21.33.1 选项

use\_fifo布尔值

[fifo多路复用器](#fifo)在单独的线程中处理。这允许补偿输出的不同速度/延迟/可靠性并设置透明恢复。默认情况下，此功能处于关闭状态。

fifo\_选项

传递给 fifo 伪多路复用器实例的选项。见[先进先出](#fifo)。

可以通过在方括号之间将它们作为由“：”分隔的key = value对列表预先添加来为每个从属指定 Muxer 选项。如果选项值包含特殊字符或“:”分隔符，则必须对其进行转义；请注意，这是二级转义。

还可以识别以下特殊选项：

F

指定格式名称。如果无法从输出 URL 中猜出，则为必填项。

bsfs[/规格]

指定要应用于指定输出的比特流过滤器列表。

/分隔的选项，可以指定给定比特流过滤器应用于哪些流。 spec必须是流说明符（请参阅[格式流说明符](#Format-stream-specifiers)）。

如果未指定流说明符，则比特流过滤器将应用于输出中的所有流。如果输出包含无法应用比特流过滤器的流，这将导致输出操作失败，例如h264\_mp4toannexb被应用于包含音频流的输出。

opt=value的形式指定。

可以指定多个比特流过滤器，以“,”分隔。

use\_fifo布尔值

这允许覆盖单个从复用器的 tee muxer use\_fifo 选项。

fifo\_选项

这允许覆盖单个从复用器的 tee muxer fifo\_options。见[先进先出](#fifo)。

选择

选择应映射到从属输出的流，由流说明符指定。如果未指定，则默认为所有映射流。如果输出格式不接受所有映射流，这将导致输出操作失败。

您可以使用以逗号 ( , ) 分隔的多个流说明符，例如： a:0,v

失败时

指定输出失败时的行为。这可以设置为abort （默认）或ignore 。如果此从属输出失败，则中止将导致整个过程失败。 ignore将忽略此输出上的失败，因此其他输出将继续而不会受到影响。

#### 21.33.2 例子

* 编码一些东西并将其存档在 WebM 文件中并通过 UDP 将其作为 MPEG-TS 流式传输：
* ffmpeg -i ... -c:v libx264 -c:a mp2 -f tee -map 0:v -map 0:a
* “archive-20121107.mkv|[f=mpegts]udp://10.0.1.255:1234/”
* 如上所述，但即使输出到本地文件失败（例如本地驱动器已满），也要继续流式传输：
* ffmpeg -i ... -c:v libx264 -c:a mp2 -f tee -map 0:v -map 0:a
* “[onfail=ignore]archive-20121107.mkv|[f=mpegts]udp://10.0.1.255:1234/”
* 使用ffmpeg对输入进行编码，并将输出发送到三个不同的目的地。根据 MPEG-TS 格式的要求，dump\_extra比特流过滤器用于将额外数据信息添加到所有输出视频关键帧数据包。选择选项应用于out.aac以使其仅包含音频数据包。
* ffmpeg -i ... -map 0 -flags +global\_header -c:v libx264 -c:a aac
* -f tee “[bsfs/v=dump\_extra=freq=keyframe]out.ts|[movflags=+faststart]out.mp4|[select=a]out.aac”
* 同上，但只选择流a:1作为音频输出。请注意，必须执行二级转义，因为“:”是用于分隔选项的特殊字符。
* ffmpeg -i ... -map 0 -flags +global\_header -c:v libx264 -c:a aac
* -f tee “[bsfs/v=dump\_extra=freq=keyframe]out.ts|[movflags=+faststart]out.mp4|[select=\'a:1\']out.aac”

### 21.34 webm\_chunk

WebM 实时块混合器。

这个 muxer 将 WebM 标头和块作为单独的文件写出，这些文件可以被支持通过 DASH 的 WebM 直播流的客户端使用。

* [选项](#Options-62)
* [例子](#Example-2)

#### 21.34.1 选项

此多路复用器支持以下选项：

chunk\_start\_index

第一个块的索引（默认为 0）。

标头

将写入初始化数据的标头的文件名。

audio\_chunk\_duration

每个音频块的持续时间（以毫秒为单位）（默认为 5000）。

#### 21.34.2 例子

ffmpeg -f v4l2 -i /dev/video0 \

-f alsa -i hw:0 \

-地图 0:0 \

-c:v libvpx-vp9 \

-s 640x360 -keyint\_min 30 -g 30 \

-f webm\_chunk \

-header webm\_live\_video\_360.hdr \

-chunk\_start\_index 1 \

webm\_live\_video\_360\_%d.chk \

-地图 1:0 \

-c:一个libvorbis \

-b:a 128k \

-f webm\_chunk \

-header webm\_live\_audio\_128.hdr \

-chunk\_start\_index 1 \

-audio\_chunk\_duration 1000 \

webm\_live\_audio\_128\_%d.chk

### 21.35 webm\_dash\_manifest

WebM DASH 清单混合器。

此 muxer 实现 WebM DASH 清单规范以生成 DASH 清单 XML。它还支持 DASH 直播流的清单生成。

有关详细信息，请参阅：

* WebM DASH 规范： <https://sites.google.com/a/webmproject.org/wiki/adaptive-streaming/webm-dash-specification>
* ISO DASH 规范： <http://standards.iso.org/ittf/PubliclyAvailableStandards/c065274_ISO_IEC_23009-1_2014.zip>
* [选项](#Options-63)
* [例子](#Example-3)

#### 21.35.1 选项

此多路复用器支持以下选项：

适应集

此选项具有以下语法：“id=x,streams=a,b,c id=y,streams=d,e” 其中 x 和 y 是自适应集的唯一标识符，a、b、c、d 和e 是相应的音频和视频流的索引。使用此选项可以添加任意数量的适配集。

居住

将此设置为 1 以创建实时流 DASH 清单。默认值：0。

chunk\_start\_index

第一个块的起始索引。这将进入清单中“ SegmentTemplate ”元素的“ startNumber ”属性。默认值：0。

chunk\_duration\_ms

每个块的持续时间（以毫秒为单位）。这将进入清单中“ SegmentTemplate ”元素的“ duration ”属性。默认值：1000。

utc\_timing\_url

将返回 ISO 格式的 UTC 时间戳的页面的 URL。这将进入清单中“ UTCTiming ”元素的“ value ”属性。默认值：无。

time\_shift\_buffer\_depth

保证任何表示可用的最短时间（以秒为单位）移动缓冲区。这将进入“ MPD ”元素的“ timeShiftBufferDepth ”属性。默认值：60。

最小更新周期

清单的最短更新周期（以秒为单位）。这将进入“ MPD ”元素的“ minimumUpdatePeriod ”属性。默认值：0。

#### 21.35.2 例子

ffmpeg -f webm\_dash\_manifest -i video1.webm \

-f webm\_dash\_manifest -i video2.webm \

-f webm\_dash\_manifest -i audio1.webm \

-f webm\_dash\_manifest -i audio2.webm \

-地图0 -地图1 -地图2 -地图3 \

-c 复制\

-f webm\_dash\_manifest \

-adaptation\_sets“id=0,streams=0,1 id=1,streams=2,3”\

清单.xml

## 22元数据

FFmpeg 能够将媒体文件中的元数据转储到一个简单的 UTF-8 编码的类似 INI 的文本文件中，然后使用元数据复用器/解复用器将其加载回来。

文件格式如下：

1. 一个文件由一个标题和许多元数据标签组成，这些标签分为多个部分，每个部分都在自己的行上。
2. 标头是一个“ ;FFMETADATA ”字符串，后跟版本号（现在为 1）。
3. 元数据标签的形式为“ key=value ”
4. 紧跟在标头之后的是全局元数据
5. 在全局元数据之后，可能会有包含每个流/每个章节元数据的部分。
6. 一个部分以大写的部分名称（即 STREAM 或 CHAPTER）在方括号（' [ '，' ] '）中开始，并以下一个部分或文件结尾结束。
7. 在章节部分的开头可能有一个可选的时基用于开始/结束值。它必须采用“ TIMEBASE= num / den ”的形式，其中num和den是整数。如果缺少时基，则假定开始/结束时间以纳秒为单位。

接下来的章节部分必须包含章节开始和结束时间，格式为“ START= num ”，“ END= num ”，其中num是正整数。

1. 空行和以 ' ;开头的行' 或 ' # ' 被忽略。
2. 包含特殊字符（“ = ”、“ ; ”、“ # ”、“ \ ”和换行符）的元数据键或值必须使用反斜杠“ \ ”进行转义。
3. 请注意，元数据中的空格（例如“ foo = bar ”）被认为是标记的一部分（在上面的示例中，键是“ foo ”，值是“ bar ”）。

ffmetadata 文件可能如下所示：

;FF元数据1

标题=自行车\\车棚

;这是评论

艺术家=FFmpeg 巨魔团队

[章节]

时基=1/1000

开始=0

#chapter 结束于 0:01:00

结束=60000

标题=第 1 章

[溪流]

标题=多\

线

通过使用 ffmetadata muxer 和 demuxer，可以将元数据从输入文件提取到 ffmetadata 文件，然后将文件转码为带有编辑后的 ffmetadata 文件的输出文件。

ffmpeg提取 ffmetadata 文件如下：

ffmpeg -i 输入 -f ffmetadata FFMETADATAFILE

从 FFMETADATAFILE 文件中重新插入已编辑的元数据信息可以通过以下方式完成：

ffmpeg -i INPUT -i FFMETADATAFILE -map\_metadata 1 -codec 复制输出

## 23 协议选项

libavformat 库提供了一些通用的全局选项，可以在所有协议上设置。此外，每个协议都可能支持所谓的私有选项，这些选项特定于该组件。

-option来设置选项 FFmpeg 工具中的值，或通过在AVFormatContext选项中显式设置值或使用libavutil/opt.h API 进行编程使用。

支持的选项列表如下：

protocol\_whitelist列表（输入）

设置以“,”分隔的允许协议列表。 “ALL”匹配所有协议。以“-”为前缀的协议被禁用。默认情况下允许所有协议，但另一个协议（嵌套协议）使用的协议仅限于每个协议子集。

## 24个协议

协议是 FFmpeg 中的配置元素，可以访问需要特定协议的资源。

当您配置 FFmpeg 构建时，默认情况下会启用所有支持的协议。您可以使用配置选项“–list-protocols”列出所有可用的协议。

您可以使用配置选项“–disable-protocols”禁用所有协议，并使用选项“–enable-protocol= PROTOCOL ”有选择地启用协议，或者您可以使用选项“–disable-protocol=”禁用特定协议协议”。

ff\* 工具的选项“-protocols”将显示支持的协议列表。

所有协议都接受以下选项：

读写超时

等待（网络）读/写操作完成的最长时间，以微秒为单位。

以下是对当前可用协议的描述。

* [amqp](#amqp)
* [异步](#async)
* [蓝光](#bluray)
* [缓存](#cache)
* [连接](#concat-2)
* [连接](#concatf)
* [密码](#crypto)
* [数据](#data-1)
* [fd](#fd)
* [文件](#file)
* [FTP](#ftp)
* [地鼠](#gopher)
* [地鼠](#gophers)
* [hls](#hls-3)
* [网址](#http)
* [冰铸](#Icecast)
* [ipfs](#ipfs)
* [MMST](#mmst)
* [嗯嗯](#mmsh)
* [MD5](#md5-2)
* [管道](#pipe)
* [提示音](#prompeg)
* [手](#rist)
* [rtmp](#rtmp)
* [模板](#rtmpe)
* [rtmps](#rtmps)
* [提示符](#rtmpt)
* [模板](#rtmpte)
* [提示音](#rtmpts)
* [libsmb客户端](#libsmbclient)
* [libssh](#libssh)
* [librtmp rtmp, rtmpe, rtmps, rtmpt, rtmpte](#librtmp-rtmp_002c-rtmpe_002c-rtmps_002c)
* [实时传输协议](#rtp)
* [实时传输协议](#rtsp)
* [树液](#sap)
* [sctp](#sctp)
* [srt](#srt)
* [srtp协议](#srtp)
* [子文件](#subfile)
* [球座](#tee-2)
* [TCP](#tcp)
* [tls](#tls)
* [UDP](#udp)
* [unix](#unix)
* [zmq](#zmq)

### 24.1 amqp

高级消息队列协议 (AMQP) 版本 0-9-1 是一种基于代理的发布-订阅通信协议。

FFmpeg 必须使用 –enable-librabbitmq 进行编译以支持 AMQP。还必须运行一个单独的 AMQP 代理。示例开源 AMQP 代理是 RabbitMQ。

启动代理后，FFmpeg 客户端可以使用以下命令将数据流式传输到代理：

ffmpeg -re -i input -f mpegts amqp://[[用户]:[密码]@]主机名[:端口][/vhost]

其中主机名和端口（默认为 5672）是代理的地址。客户端还可以设置用于身份验证的用户/密码。这两个字段的默认值都是“guest”。可以使用 vhost 设置 broker 上的虚拟主机名称。默认值为“/”。

多个订阅者可以使用以下命令从代理流式传输：

ffplay amqp://[[用户]:[密码]@]主机名[:端口][/vhost]

在 RabbitMQ 中，发布到代理的所有数据都流经特定的交换器，并且每个订阅客户端都有一个分配的队列/缓冲区。当数据包到达交换器时，它可能会根据交换器和 routing\_key 字段被复制到客户端的队列中。

支持以下选项：

交换

设置要在经纪人上使用的交易所。 RabbitMQ 有几个预定义的交换器：“amq.direct”是默认的交换器，其中发布者和订阅者必须有一个匹配的 routing\_key； “amq.fanout”与广播操作相同（即数据被转发到独立于routing\_key的fanout exchange上的所有队列）； “amq.topic”类似于“amq.direct”，但允许更复杂的模式匹配（请参阅 RabbitMQ 文档）。

路由键

设置路由键。默认值为“amqp”。路由键用于“amq.direct”和“amq.topic”交换来决定是否将数据包写入订阅者的队列。

pkt\_size

发送/接收到代理的每个数据包的最大大小。默认值为 131072。最小值为 4096，最大值为任何大值（由 int 表示）。接收数据包时，这会在 FFmpeg 中设置一个内部缓冲区大小。它应该等于或大于发布到代理的数据包的大小。否则，接收到的消息可能会被截断，从而导致解码错误。

连接超时

与代理的初始连接期间的超时秒数。默认值为 rw\_timeout，如果未设置 rw\_timeout，则为 5 秒。

delivery\_mode模式

设置发送给代理的每条消息的传递模式。接受以下值：

'执着'

传递模式设置为“持久”(2)。这是默认值。根据其设置，消息可能会写入代理的磁盘。

'非持久'

传递模式设置为“非持久”(1)。除非代理处于内存压力之下，否则消息将保留在代理的内存中。

### 24.2 异步

输入流的异步数据填充包装器。

在后台线程中填充数据，以将 I/O 操作与多路分解线程分离。

异步：网址

异步：http://主机/资源

异步：缓存：http：//主机/资源

### 24.3蓝光

阅读蓝光播放列表。

接受的选项是：

角度

蓝光角度

章节

开始章节 (1...N)

播放列表

要阅读的播放列表 (BDMV/PLAYLIST/??????.mpls)

例子：

从蓝光读取最长的播放列表安装到 /mnt/bluray：

蓝光：/mnt/蓝光

从挂载到 /mnt/bluray 的 BluRay 读取播放列表 4 的角度 2，从第 2 章开始：

-playlist 4 -angle 2 -chapter 2 bluray:/mnt/bluray

### 24.4缓存

输入流的缓存包装器。

将输入流缓存到临时文件。它为直播带来了搜索能力。

接受的选项是：

预读限制

不支持搜索时可以提前读取的字节数。范围是 -1 到 INT\_MAX。 -1 表示无限制。默认值为 65536。

网址语法是

缓存：网址

### 24.5 连续

物理级联协议。

按顺序阅读和查找许多资源，就好像它们是一种独特的资源一样。

此协议接受的 URL 具有以下语法：

连接： URL1 |网址 2 |...|网址

其中URL1 、 URL2 、 ...、 URLN是要连接的资源的 url，每个都可能指定一个不同的协议。

使用ffplay读取一系列文件split1.mpeg 、 split2.mpeg 、 split3.mpeg使用命令：

ffplay 连接：split1.mpeg\|split2.mpeg\|split3.mpeg

请注意，您可能需要转义字符“|”这对许多 shell 来说是特殊的。

### 24.6 连接

使用换行符分隔的资源列表的物理级联协议。

按顺序阅读和查找许多资源，就好像它们是一种独特的资源一样。

此协议接受的 URL 具有以下语法：

concatf:网址

其中URL是包含以换行符分隔的要连接的资源列表的 url，每个资源都可能指定一个不同的协议。特殊字符必须用反斜杠或单引号转义。请参阅[(ffmpeg-utils)ffmpeg-utils(1) 手册中的“引用和转义”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#quoting_005fand_005fescaping)。

使用ffplay读取文件split.txt中单独行中列出的一系列文件split1.mpeg 、 split2.mpeg 、 split3.mpeg使用命令：

ffplay concatf：split.txt

其中split.txt包含以下行：

split1.mpeg文件

split2.mpeg文件

split3.mpeg文件

### 24.7 加密

AES 加密流读取协议。

接受的选项是：

钥匙

根据给定的十六进制表示设置 AES 解密密钥二进制块。

iv

根据给定的十六进制表示设置 AES 解密初始化向量二进制块。

接受的网址格式：

密码：网址

加密+网址

### 24.8数据

URI 中的内联数据。请参阅<http://en.wikipedia.org/wiki/Data_URI_scheme>。

例如，要转换内联给定的 GIF 文件ffmpeg ：

ffmpeg -i "" 笑脸.png

### 24.9 英尺

文件描述符访问协议。

接受的语法是：

fd: -fd文件描述符

如果未指定fd ，默认情况下将使用 stdout 文件描述符进行写入，使用 stdin 进行读取。与pipe协议不同，fd协议如果对应的是普通文件，则有seek支持。为了安全起见，fd 协议不支持通过 URL 传递文件描述符。

该协议接受以下选项：

块大小

设置 I/O 操作的最大块大小，以字节为单位。默认值为INT\_MAX ，这导致不限制请求的块大小。将此值设置得合理低可缩短用户终止请求的反应时间，这在数据传输速度较慢时非常有用。

fd

设置文件描述符。

### 24.10档

文件访问协议。

读取或写入文件。

文件 URL 可以采用以下形式：

文件：文件名

其中文件名是要读取的文件的路径。

没有协议前缀的 URL 将被假定为文件 URL。根据构建的不同，一个看起来像 Windows 路径且以驱动器号开头的 URL 也将被假定为文件 URL（在类 unix 系统的构建中通常不是这种情况）。

例如，要使用ffmpeg从文件input.mpeg中读取，请使用以下命令：

ffmpeg -i 文件：输入.mpeg 输出.mpeg

该协议接受以下选项：

截短

如果设置为 1，则写入时截断现有文件。值 0 可防止截断。默认值为 1。

块大小

设置 I/O 操作的最大块大小，以字节为单位。默认值为INT\_MAX ，这导致不限制请求的块大小。将此值设置得相当低可以改善用户终止请求的反应时间，这对于慢速介质上的文件很有价值。

跟随

如果设置为 1，协议将在文件末尾重试读取，允许读取仍在写入的文件。为了终止它，您需要使用 rw\_timeout 选项，或者使用中断回调（对于 API 用户）。

可寻找的

控制是否在文件上公布可搜索性。 0 表示不可搜索，-1 表示自动（普通文件可搜索，命名管道不可搜索）。

许多分路器以不同的方式处理可搜索和不可搜索的资源，覆盖这可能会以丢失某些功能（例如准确搜索）为代价加快打开某些文件的速度。

### 24.11 FTP

FTP（文件传输协议）。

使用 FTP 协议读取或写入远程资源。

需要以下语法。

ftp://[用户[:密码]@]服务器[:端口]/path/to/remote/resource.mpeg

该协议接受以下选项。

暂停

设置底层低级操作使用的套接字 I/O 操作的超时时间（以微秒为单位）。默认情况下，它设置为 -1，这意味着未指定超时。

ftp用户

设置用于对 FTP 服务器进行身份验证的用户。这被 FTP URL 中的用户覆盖。

ftp-密码

设置用于验证 FTP 服务器的密码。这将被 FTP URL 中的密码覆盖，如果未设置用户，则被ftp-anonymous-password覆盖。

ftp-匿名密码

以匿名用户身份登录时使用的密码。通常应使用电子邮件地址。

ftp 写入可寻址

在编码期间控制连接的可搜索性。如果设置为 1，资源应该是可搜索的，如果设置为 0，则假定资源不可搜索。默认值为 0。

注意：协议可以用作输出，但建议不要这样做，除非特别小心（测试、自定义服务器配置等）。不同的 FTP 服务器在搜索操作期间以不同的方式运行。由于服务器限制，ff\* 工具可能会生成不完整的内容。

### 24.12地鼠

地鼠协议。

### 24.13 地鼠

地鼠协议。

带有 TLS 封装的 Gopher 协议。

### 24.14 小时

将 Apple HTTP Live Streaming compliant segmented stream 看成一个统一的流。描述片段的 M3U8 播放列表可以是远程 HTTP 资源或本地文件，使用标准文件协议访问。嵌套协议通过在 hls URI 方案名称后指定“+ proto ”来声明，其中proto是“file”或“http”。

hls+http://host/path/to/remote/resource.m3u8

hls+file://path/to/local/resource.m3u8

不鼓励使用此协议 - hls demuxer 应该也能正常工作（如果不能，请报告问题）并且更完整。要改为使用 hls demuxer，只需使用指向 m3u8 文件的直接 URL。

### 24.15 HTTP

HTTP（超文本传输协议）。

该协议接受以下选项：

可寻找的

控制连接的可搜索性。如果设置为 1 资源应该是可搜索的，如果设置为 0 则假定不可搜索，如果设置为 -1 它将尝试自动检测它是否可搜索。默认值为 -1。

chunked\_post

如果设置为 1，则对帖子使用分块传输编码，默认值为 1。

内容类型

为 POST 消息或侦听模式设置特定的内容类型。

http\_proxy

将 HTTP 代理设置为通过例如 http://example.com:1234 建立隧道

标题

设置自定义 HTTP 标头，可以覆盖内置的默认标头。该值必须是对标头进行编码的字符串。

多个请求

如果设置为 1，则使用持久连接，默认值为 0。

发布数据

设置自定义 HTTP 发布数据。

推荐人

设置 Referer 标头。在 HTTP 请求中包含“Referer: URL”标头。

用户代理

覆盖 User-Agent 标头。如果未指定，协议将使用描述 libavformat 构建的字符串。 ("Lavf/<版本>")

重新连接

如果设置则 eof 被视为错误并导致重新连接，这对于直播/无限流很有用。

重新连接流

如果设置，则即使是流式/非可搜索流也会在出错时重新连接。

重新连接网络错误

如果连接期间出现 TCP/TLS 错误，则自动重新连接。

重新连接\_on\_http\_error

要重新连接的 HTTP 状态代码的逗号分隔列表。该列表可以包括特定的状态代码（例如“503”）或字符串“4xx”/“5xx”。

重新连接延迟最大值

设置以秒为单位的最大延迟，之后放弃重新连接

mime\_type

导出 MIME 类型。

http\_version

导出 HTTP 响应版本号。通常是“1.0”或“1.1”。

冰冷的

如果设置为 1，则从服务器请求 ICY (SHOUTcast) 元数据。如果服务器支持，则应用程序必须通过读取icy\_metadata\_headers和icy\_metadata\_packet选项来检索元数据。默认值为 1。

icy\_metadata\_headers

如果服务器支持 ICY 元数据，则它包含 ICY 特定的 HTTP 回复标头，由换行符分隔。

icy\_metadata\_packet

如果服务器支持 ICY 元数据，并且icy设置为 1，则这包含服务器发送的最后一个非空元数据包。它应该由对中流元数据更新感兴趣的应用程序定期轮询。

饼干

设置要在以后的请求中发送的 cookie。每个 cookie 的格式与 Set-Cookie HTTP 响应字段的值相同。多个 cookie 可以用换行符分隔。

抵消

设置初始字节偏移量。

end\_offset

尝试将请求限制为此偏移量之前的字节。

方法

当用作客户端选项时，它会为请求设置 HTTP 方法。

当用作服务器选项时，它设置预期来自客户端的 HTTP 方法。如果预期的和接收到的 HTTP 方法不匹配，客户端将收到一个 Bad Request 响应。取消设置时，暂时不检查 HTTP 方法。这将在未来被自动检测所取代。

听

如果设置为 1，则启用实验性 HTTP 服务器。这可用于在用作输出选项时发送数据，或在用作输入选项时使用 HTTP POST 从客户端读取数据。如果设置为 2，则启用实验性多客户端 HTTP 服务器。这尚未在 ffmpeg.c 中实现，因此不能用作命令行选项。

# 服务器端（发送）：

ffmpeg -i somefile.ogg -c copy -listen 1 -f ogg http://服务器:端口

# 客户端（接收）：

ffmpeg -i http://服务器:端口-c 复制 somefile.ogg

# 客户端也可以用wget来完成：

wget http://服务器：端口-O somefile.ogg

# 服务器端（接收）：

ffmpeg -listen 1 -i http://服务器:端口-c 复制 somefile.ogg

# 客户端（发送）：

ffmpeg -i somefile.ogg -chunked\_post 0 -c copy -f ogg http://服务器:端口

# 客户端也可以用wget来完成：

wget --post-file=somefile.ogg http://服务器:端口

发送\_期望\_100

为 POST 发送 Expect: 100-continue 标头。如果设置为 1 它将发送，如果设置为 0 则不会，如果设置为 -1 它将尝试发送（如果适用）。默认值为 -1。

授权类型

设置 HTTP 认证类型。没有 Digest 选项，因为此方法需要先从服务器获取 nonce 参数，并且不能像 Basic 那样直接使用。

没有任何

自动选择 HTTP 身份验证类型。这是默认设置。

基本的

选择 HTTP 基本身份验证。

基本身份验证发送一个 Base64 编码的字符串，其中包含客户端的用户名和密码。 Base64 不是一种加密形式，应被视为与以明文形式发送用户名和密码相同（Base64 是一种可逆编码）。如果需要保护资源，请强烈考虑使用基本身份验证以外的身份验证方案。 HTTPS/TLS 应该与基本身份验证一起使用。如果没有这些额外的安全增强功能，则不应使用基本身份验证来保护敏感或有价值的信息。

* [HTTP cookies](#HTTP-Cookies)

#### 24.15.1 HTTP Cookie

某些 HTTP 请求将被拒绝，除非 cookie 值随请求一起传递。 cookies选项允许指定这些 cookies。至少，每个 cookie 必须指定一个值以及路径和域。同时匹配域和路径的 HTTP 请求将自动在 HTTP Cookie 标头字段中包含 cookie 值。多个 cookie 可以用换行符分隔。

播放指定 cookie 的流所需的语法是：

ffplay -cookies“nlqptid=nltid=tsn；路径=/；域=somedomain.com；” http://somedomain.com/somestream.m3u8

### 24.16 冰铸

Icecast 协议（流向 Icecast 服务器）

该协议接受以下选项：

ice\_genre

设置流类型。

冰名

设置流名称。

冰的描述

设置流描述。

ice\_url

设置流网站 URL。

冰公共

设置流是否应该公开。默认值为 0（不公开）。

用户代理

覆盖 User-Agent 标头。如果未指定，将使用“Lavf/<version>”形式的字符串。

密码

设置 Icecast 挂载点密码。

内容类型

设置流内容类型。如果它不同于 audio/mpeg，则必须设置它。

legacy\_icecast

这启用了对 Icecast 版本 < 2.4.0 的支持，这些版本不支持 HTTP PUT 方法但支持 SOURCE 方法。

tls

与 Icecast 建立 TLS (HTTPS) 连接。

icecast://[用户名[:密码]@]服务器：端口/挂载点

### 24.17 ipfs

行星际文件系统 (IPFS) 协议支持。人们可以通过所谓的网关访问存储在 IPFS 网络上的文件。这些是 http(s) 端点。该协议包装了要发送到此类网关的 IPFS 本机协议（ipfs:// 和 ipns://）。用户可以（并且应该）托管自己的节点，这意味着该协议将使用一个人的本地网关来访问 IPFS 网络上的文件。

该协议接受以下选项：

网关

定义要使用的网关。如果未设置，协议将首先尝试通过按顺序查看$IPFS\_GATEWAY 、 $IPFS\_PATH和$HOME/.ipfs/来定位本地网关。

可以通过两种方式使用此协议。使用 IPFS：

ffplay ipfs://<散列>

或者 IPNS 协议（IPNS 是可变的 IPFS）：

ffplay ipns://<哈希>

### 24.18 毫米

基于 TCP 的 MMS（Microsoft 媒体服务器）协议。

### 24.19 毫米

基于 HTTP 的 MMS（Microsoft 媒体服务器）协议。

所需的语法是：

mmsh://服务器[:端口][/应用][/播放路径]

### 24.20 MD5

MD5 输出协议。

计算要写入的数据的 MD5 哈希，并在关闭时将其写入指定的输出或标准输出（如果未指定）。它可用于测试多路复用器而无需编写实际文件。

下面是一些示例。

# 将编码后的 AVI 文件的 MD5 哈希写入文件 output.avi.md5。

ffmpeg -i 输入.flv -f avi -y md5:输出.avi.md5

# 将编码后的 AVI 文件的 MD5 哈希写入标准输出。

ffmpeg -i input.flv -f avi -y md5:

请注意，某些格式（通常是 MOV）要求输出协议是可搜索的，因此它们将无法使用 MD5 输出协议。

### 24.21管

UNIX 管道访问协议。

从 UNIX 管道读取和写入。

接受的语法是：

管道：[数字]

如果未指定fd ，则 number是对应于管道文件描述符的数字（例如，0 代表 stdin，1 代表 stdout，2 代表 stderr）。如果未指定数字，默认情况下将使用 stdout 文件描述符进行写入，使用 stdin 进行读取。

例如，使用ffmpeg从 stdin 读取：

猫测试.wav | ffmpeg -i 管道：0

# ...这与...相同

猫测试.wav | ffmpeg -i 管道：

ffmpeg写入标准输出：

ffmpeg -i test.wav -f avi 管道：1 |猫 > test.avi

# ...这与...相同

ffmpeg -i test.wav -f avi 管道： |猫 > test.avi

该协议接受以下选项：

块大小

设置 I/O 操作的最大块大小，以字节为单位。默认值为INT\_MAX ，这导致不限制请求的块大小。将此值设置得合理低可缩短用户终止请求的反应时间，这在数据传输速度较慢时非常有用。

fd

设置文件描述符。

请注意，某些格式（通常是 MOV）要求输出协议是可搜索的，因此它们将无法使用管道输出协议。

### 24.22 prompeg

Pro-MPEG 实践代码#3 第 2 版 FEC 协议。

Pro-MPEG CoP#3 FEC 是一种二维奇偶校验前向纠错机制，用于通过 RTP 发送的 MPEG-2 传输流。

rtp\_mpegts muxer 和rtp协议结合使用。

所需的语法是：

-f rtp\_mpegts -fec prompeg= option = val ... rtp://主机名：端口

目标 UDP 端口是列 FEC 流的端口 + 2和行 FEC 流的端口 + 4 。

该协议接受以下选项：

l= n

列数（4-20，LxD <= 100）

d= n

行数（4-20，LxD <= 100）

用法示例：

-f rtp\_mpegts -fec prompeg=l=8:d=4 rtp://主机名:端口

### 24.23 瑞斯特

可靠的互联网流媒体传输协议

接受的选项是：

rist\_profile

支持的值：

'简单'

'主要'

这个是默认的。

'进阶'

缓冲区大小

以毫秒为单位设置内部 RIST 缓冲区大小以重新传输数据。默认值为 0，这意味着 librist 默认值（1 秒）。最大值为 30 秒。

fifo\_size

librist 接收器输出 fifo 的大小（以数据包数表示）。这必须是 2 的幂。默认为 8192（相对于 1024 的 librist 默认值）。

overrun\_nonfatal= 1|0

在 librist fifo 缓冲区溢出的情况下生存。默认值为 0。

pkt\_size

设置发送数据的最大数据包大小。默认为 1316。

日志级别

为 RIST 日志消息设置日志级别。如果您明确想要启用调试级别消息或数据包丢失模拟，则只需设置此项，否则将遵守常规日志级别。

秘密

设置覆盖加密秘密，默认情况下未设置。

加密

设置加密类型，默认禁用。可接受的值为 128 和 256。

### 24.24 rtmp

实时消息协议。

实时消息协议 (RTMP) 用于通过 TCP/IP 网络传输多媒体内容。

所需的语法是：

rtmp://[用户名:密码@]服务器[:端口][/ app ][/ instance ][/ playpath ]

接受的参数是：

用户名

一个可选的用户名（主要用于发布）。

密码

可选密码（主要用于发布）。

服务器

RTMP 服务器的地址。

港口

要使用的 TCP 端口号（默认为 1935）。

应用程序

它是要访问的应用程序的名称。它通常对应于应用程序在 RTMP 服务器上的安装路径（例如/ondemand/ 、 /flash/live/等）。您也可以通过rtmp\_app选项覆盖从 URI 解析的值。

游戏路径

app中指定的应用程序播放的资源路径或名称，可以以“mp4:”为前缀。您也可以通过rtmp\_playpath选项覆盖从 URI 解析的值。

听

充当服务器，侦听传入连接。

暂停

等待传入连接的最长时间。表示听。

AVOption s 在代码中）设置以下参数：

rtmp\_app

要连接到 RTMP 服务器的应用程序的名称。此选项会覆盖 URI 中指定的参数。

rtmp\_缓冲区

以毫秒为单位设置客户端缓冲时间。默认值为 3000。

rtmp连接

额外的任意 AMF 连接参数，从字符串解析，例如B:1 S:authMe O:1 NN:code:1.23 NS:flag:ok O:0 。每个值都带有一个表示类型的字符前缀，B 表示布尔值，N 表示数字，S 表示字符串，O 表示对象，或 Z 表示空值，后跟一个冒号。对于布尔值，对于 FALSE 或 TRUE，数据必须分别为 0 或 1。同样，对于对象，数据必须分别为 0 或 1 以结束或开始一个对象。子对象中的数据项可以通过在类型前加上“N”前缀并在值之前指定名称（即NB:myFlag:1 ）来命名。可以多次使用此选项来构造任意 AMF 序列。

rtmp\_flashver 文件

用于运行 SWF 播放器的 Flash 插件版本。默认值为 LNX 9,0,124,2。 （发布时，默认为 FMLE/3.0（兼容；<libavformat 版本>）。）

rtmp\_flush\_interval

同一请求中刷新的数据包数（仅限 RTMPT）。默认值为 10。

rtmp\_live

指定媒体是直播流。无法在实时流中恢复或搜索。默认值为any ，这意味着订阅者首先尝试播放播放路径中指定的直播流。如果找不到该名称的直播流，它会播放录制的流。其他可能的值是live和recorded 。

rtmp\_pageurl

嵌入媒体的网页的 URL。默认情况下不会发送任何值。

rtmp\_播放路径

要播放或发布的流标识符。此选项会覆盖 URI 中指定的参数。

rtmp\_订阅

要订阅的直播流的名称。默认情况下不会发送任何值。仅当指定选项或 rtmp\_live 设置为 live 时才会发送。

rtmp\_swf哈希

解压缩的 SWF 文件的 SHA256 哈希值（32 字节）。

rtmp\_swf大小

SWFVerification 所需的解压缩 SWF 文件的大小。

rtmp\_swfurl 文件

媒体的 SWF 播放器的 URL。默认情况下不会发送任何值。

rtmp\_swf验证

播放器 swf 文件的 URL，自动计算哈希/大小。

rtmp\_tcurl 文件

目标流的 URL。默认为 proto://host[:port]/app。

tcp\_nodelay= 1|0

设置 TCP\_NODELAY 以禁用 Nagle 算法。默认值为 0。

备注：写入套接字目前未优化以最小化系统调用并降低 TCP\_NODELAY 的效率/效果。

例如，使用ffplay从 RTMP 服务器“myserver”的应用程序“vod”中读取名为“sample”的多媒体资源：

ffplay rtmp://myserver/vod/sample

要发布到受密码保护的服务器，分别传递播放路径和应用程序名称：

ffmpeg -re -i <input> -f flv -rtmp\_playpath some/long/path -rtmp\_app long/app/name rtmp://用户名:密码@myserver/

### 24.25 rmp

加密实时消息协议。

加密实时消息传递协议 (RTMPE) 用于在标准加密原语中流式传输多媒体内容，由 Diffie-Hellman 密钥交换和 HMACSHA256 组成，生成一对 RC4 密钥。

### 24.26 rtmps

通过安全 SSL 连接的实时消息传递协议。

实时消息协议 (RTMPS) 用于通过加密连接流式传输多媒体内容。

### 24.27 rtmpt

通过 HTTP 隧道传输的实时消息传递协议。

通过 HTTP 隧道传输的实时消息传递协议 (RTMPT) 用于流式传输 HTTP 请求中的多媒体内容以穿越防火墙。

### 24.28 rtmpte

通过 HTTP 隧道传输的加密实时消息传递协议。

通过 HTTP 隧道传输的加密实时消息传递协议 (RTMPTE) 用于流式传输 HTTP 请求中的多媒体内容以穿越防火墙。

### 24.29 rtmpts

通过 HTTPS 传输的实时消息传递协议。

通过 HTTPS (RTMPTS) 隧道传输的实时消息协议用于流式传输 HTTPS 请求中的多媒体内容以穿越防火墙。

### 24.30 libsmb 客户端

libsmbclient 允许操作 CIFS/SMB 网络资源。

需要以下语法。

smb://[[域:]用户[:密码@]]服务器[/共享[/路径[/文件]]]

该协议接受以下选项。

暂停

设置底层低级操作使用的套接字 I/O 操作的超时时间（以毫秒为单位）。默认情况下，它设置为 -1，这意味着未指定超时。

截短

如果设置为 1，则写入时截断现有文件。值 0 可防止截断。默认值为 1。

工作组

设置用于建立连接的工作组。默认情况下未指定工作组。

有关详细信息，请参阅： <http://www.samba.org/>。

### 24.31 libssh

通过 libssh 的安全文件传输协议

使用 SFTP 协议读取或写入远程资源。

需要以下语法。

sftp://[用户[:密码]@]服务器[:端口]/path/to/remote/resource.mpeg

该协议接受以下选项。

暂停

设置底层低级操作使用的套接字 I/O 操作超时。默认情况下，它设置为 -1，这意味着未指定超时。

截短

如果设置为 1，则写入时截断现有文件。值 0 可防止截断。默认值为 1。

私钥

指定包含在授权期间使用的私钥的文件的路径。默认情况下，libssh 在~/.ssh/目录中搜索密钥。

示例：播放存储在远程服务器上的文件。

ffplay sftp://user:password@server\_address:22/home/user/resource.mpeg

### 24.32 librtmp rtmp、rtmpe、rtmps、rtmpt、rtmpte

实时消息传递协议及其通过 librtmp 支持的变体。

在配置期间需要存在 librtmp 标头和库。您需要使用“–enable-librtmp”显式配置构建。如果启用，这将取代本机 RTMP 协议。

该协议提供了支持 RTMP、HTTP 隧道 RTMP (RTMPT)、加密 RTMP (RTMPE)、基于 SSL/TLS 的 RTMP (RTMPS) 和这些加密类型的隧道变体（RTMPTE、RTMPTS）所需的大部分客户端功能和一些服务器功能.

所需的语法是：

rtmp\_proto :// server [: port ][/ app ][/ playpath ]选项

其中rtmp\_proto是字符串 "rtmp", "rtmpt", "rtmpe", "rtmps", "rtmpte", "rtmpts" 对应的每个 RTMP 变体之一， server , port , app和playpath与指定的含义相同对于 RTMP 本机协议。 options包含key = val形式的空格分隔选项列表。

有关详细信息，请参阅 librtmp 手册页 (man 3 librtmp)。

ffmpeg将文件实时流式传输到 RTMP 服务器：

ffmpeg -re -i myfile -f flv rtmp://myserver/live/mystream

ffplay播放相同的流：

ffplay "rtmp://myserver/live/mystream live=1"

### 24.33 分钟

实时传输协议。

RTP URL 所需的语法是：rtp:// hostname [: port ][?选项=值...]

端口指定要使用的 RTP 端口。

支持以下 URL 选项：

ttl= n

设置 TTL（生存时间）值（仅适用于多播）。

rtcpport= n

将远程 RTCP 端口设置为n 。

localrtpport= n

将本地 RTP 端口设置为n 。

localrtcpport= n '

将本地 RTCP 端口设置为n 。

pkt\_size = n

将最大数据包大小（以字节为单位）设置为n 。

buffer\_size=大小

以字节为单位设置最大 UDP 套接字缓冲区大小。

连接=0|1

在 UDP 套接字上执行connect() （如果设置为 1）或不执行（如果设置为 0）。

来源= ip [, ip ]

列出允许的源 IP 地址。

块= ip [, ip ]

列出不允许（阻止）的源 IP 地址。

write\_to\_source=0|1

将数据包发送到最新接收到的数据包的源地址（如果设置为 1）或默认远程地址（如果设置为 0）。

本地端口= n

将本地 RTP 端口设置为n 。

localaddr=地址

用于发送数据包或加入多播组的网络接口的本地 IP 地址。

超时= n

将套接字 I/O 操作的超时（以微秒为单位）设置为n 。

这是一个弃用的选项。相反，应该使用localrtpport 。

重要笔记：

1. 如果未设置rtcpport ，则 RTCP 端口将设置为 RTP 端口值加 1。
2. 如果未设置localrtpport （本地 RTP 端口），任何可用端口将用于本地 RTP 和 RTCP 端口。
3. 如果未设置localrtcpport （本地 RTCP 端口），它将被设置为本地 RTP 端口值加 1。

### 24.34 rtsp

实时流协议。

RTSP 在技术上不是 libavformat 中的协议处理程序，它是一个解复用器和复用器。解复用器支持普通 RTSP（数据通过 RTP 传输；这被 Apple 和 Microsoft 使用）和 Real-RTSP（数据通过 RDT 传输）。

muxer 可用于使用 RTSP ANNOUNCE 将流发送到支持它的服务器（目前是 Darwin Streaming Server 和 Mischa Spiegelmock 的[RTSP 服务器](https://github.com/revmischa/rtsp-server)）。

RTSP url 所需的语法是：

rtsp://主机名[:端口]/路径

ffmpeg / ffplay命令行上设置选项，或者通过AVOption或avformat\_open\_input在代码中设置。

* [复用器](#Muxer)
* [解复用器](#Demuxer)
* [例子](#Examples-20)

#### 24.34.1 混合器

支持以下选项。

rtsp\_传输

设置 RTSP 传输协议。

它接受以下值：

' UDP '

使用 UDP 作为下层传输协议。

' TCP '

使用 TCP（在 RTSP 控制通道内交织）作为较低的传输协议。

默认值为“ 0 ”。

rtsp\_标志

设置 RTSP 标志。

接受以下值：

'拉美'

对 AAC 使用 MP4A-LATM 打包而不是 MPEG4-GENERIC。

' rfc2190 '

对于 H.263，使用 RFC 2190 打包而不是 RFC 4629。

' skip\_rtcp '

不要发送 RTCP 发件人报告。

' h264\_mode0 '

在 RTP 中为 H.264 使用模式 0。

'再见'

完成时发送 RTCP BYE 数据包。

默认值为“ 0 ”。

最小端口

设置最小本地 UDP 端口。默认值为 5000。

最大端口

设置最大本地 UDP 端口。默认值为 65000。

缓冲区大小

以字节为单位设置最大套接字缓冲区大小。

pkt\_size

设置最大发送数据包大小（以字节为单位）。默认值为 1472。

#### 24.34.2 解复用器

支持以下选项。

初始暂停

如果设置为 1，则不立即开始播放流。默认值为 0。

rtsp\_传输

设置 RTSP 传输协议。

它接受以下值：

' UDP '

使用 UDP 作为下层传输协议。

' TCP '

使用 TCP（在 RTSP 控制通道内交织）作为较低的传输协议。

' udp\_multicast '

使用 UDP 多播作为较低的传输协议。

' http '

使用 HTTP 隧道作为较低的传输协议，这对于传递代理很有用。

' https '

使用 HTTPs 隧道作为较低的传输协议，这对于传递代理很有用，并广泛用于安全考虑。

可以指定多个较低的传输协议，在这种情况下，一次尝试一个（如果一个设置失败，则尝试下一个）。对于 muxer，仅支持“ tcp ”和“ udp ”选项。

rtsp\_标志

设置 RTSP 标志。

接受以下值：

'过滤器源'

仅接受来自协商的对等地址和端口的数据包。

'听'

充当服务器，侦听传入连接。

' prefer\_tcp '

如果 TCP 可用作 RTSP RTP 传输，请首先尝试使用 TCP 进行 RTP 传输。

' satip\_raw '

导出原始 MPEG-TS 流而不是解复用。该标志将简单地写出原始流，原始 PAT/PMT/PID 完好无损。

默认值为“无”。

允许的媒体类型

设置要从服务器接受的媒体类型。

接受以下标志：

'视频'

'音频'

'数据'

'字幕'

默认情况下它接受所有媒体类型。

最小端口

设置最小本地 UDP 端口。默认值为 5000。

最大端口

设置最大本地 UDP 端口。默认值为 65000。

监听超时

设置最大超时（以秒为单位）以建立初始连接。设置listen\_timeout > 0 会将rtsp\_flags设置为“ listen ”。默认值为 -1，这意味着设置“侦听”模式时无限超时。

重新排序队列大小

设置要缓冲的数据包数，以处理重新排序的数据包。

暂停

以微秒为单位设置套接字 TCP I/O 超时。

用户代理

覆盖 User-Agent 标头。如果未指定，则默认为 libavformat 标识符字符串。

缓冲区大小

以字节为单位设置最大套接字缓冲区大小。

当通过 UDP 接收数据时，分路器会尝试重新排序接收到的数据包（因为它们可能会乱序到达，或者数据包可能会完全丢失）。这可以通过将最大多路分解延迟设置为零（通过AVFormatContext 的max\_delay字段）来禁用。

使用ffplay观看多比特率 Real-RTSP 流时，可以使用-vst选择要显示的流 n和-ast n分别用于视频和音频，可以通过按v和a即时切换。

#### 24.34.3 例子

以下示例都使用了ffplay和ffmpeg工具。

* 通过 UDP 观看流，最大重新排序延迟为 0.5 秒：
* ffplay -max\_delay 500000 -rtsp\_transport udp rtsp://server/video.mp4
* 观看通过 HTTP 隧道传输的流：
* ffplay -rtsp\_transport http rtsp://server/video.mp4
* 实时发送流到 RTSP 服务器，供其他人观看：
* ffmpeg -re -i输入-f rtsp -muxdelay 0.1 rtsp://server/live.sdp
* 实时接收流：
* ffmpeg -rtsp\_flags listen -i rtsp://ownaddress/live.sdp输出

### 24.35闷棍

会话公告协议 (RFC 2974)。这在技术上不是 libavformat 中的协议处理程序，它是一个 muxer 和 demuxer。它用于通过在单独的端口上定期宣布流的 SDP 来发送 RTP 流的信号。

* [复用器](#Muxer-1)
* [解复用器](#Demuxer-1)

#### 24.35.1 混合器

提供给 muxer 的 SAP url 的语法是：

sap://目的地[:端口][?选项]

RTP 数据包发送到端口port上的目标，如果未指定端口，则发送到端口 5004。 options是一个&分隔的列表。支持以下选项：

announce\_addr=地址

指定用于发送公告的目标 IP 地址。如果省略，则通知将发送到常用的 SAP 通知多播地址 224.2.127.254 (sap.mcast.net)，如果目的地是 IPv6 地址，则发送到 ff0e::2:7ffe。

announce\_port=端口

指定发送公告的端口，如果不指定则默认为 9875。

ttl= ttl

指定公告和 RTP 数据包的生存时间值，默认为 255。

相同端口= 0|1

如果设置为 1，则在同一端口对上发送所有 RTP 流。如果为零（默认值），则所有流都在唯一的端口上发送，每个流在比前一个高 2 的端口号上发送。 VLC/Live555 要求将其设置为 1，以便能够接收流。用于接收的 libavformat 中的 RTP 堆栈要求所有流都在唯一端口上发送。

示例命令行如下。

要在本地子网上广播流，以便在 VLC 中观看：

ffmpeg -re -i input -f sap sap://224.0.0.255?same\_port=1

ffplay中观看：

ffmpeg -re -i输入-f sap sap://224.0.0.255

通过 IPv6在ffplay中观看：

ffmpeg -re -i输入-f sap sap://[ff0e::1:2:3:4]

#### 24.35.2 解复用器

提供给多路分解器的 SAP url 的语法是：

sap://[地址][:端口]

address是监听通知的多播地址，如果省略，则使用默认的 224.2.127.254 (sap.mcast.net)。 port是侦听的端口，如果省略则为 9875。

多路分解器监听给定地址和端口上的公告。收到通知后，它会尝试接收该特定流。

示例命令行如下。

要播放在普通 SAP 多播地址上宣布的第一个流：

ffplay sap://

要播放在默认 IPv6 SAP 多播地址上宣布的第一个流：

ffplay sap://[ff0e::2:7ffe]

### 24.36 sctp

流控制传输协议。

接受的 URL 语法是：

sctp://主机：端口[?选项]

该协议接受以下选项：

听

如果设置为任何值，则侦听传入连接。默认情况下完成传出连接。

最大流

设置最大流数。默认情况下没有设置限制。

### 24.37 秒

Haivision 通过 libsrt 的安全可靠传输协议。

SRT URL 支持的语法是：

srt://主机名:端口[?选项]

options包含以 & 分隔的选项列表，格式为key = val 。

或者

选项srt://主机名:端口

选项包含 '-键列表 val ' 选项。

该协议接受以下选项。

connect\_timeout=毫秒

连接超时; SRT 无法连接 RTT > 1500 毫秒（2 次握手交换），默认连接超时为 3 秒。此选项适用于呼叫者和集合点连接模式。连接超时是为集合点模式设置的值的 10 倍（可以用作早期版本中此连接问题的解决方法）。

ffs=字节

飞行标志大小（窗口大小），以字节为单位。 FFS 实际上是一个内部参数，您应该将其设置为不小于recv\_buffer\_size和mss 。默认值比较大，因此除非你设置一个非常大的接收缓冲区，否则你不需要改变这个选项。默认值为 25600。

inputbw=字节/秒

发件人标称输入速率，以字节/秒为单位。与oheadbw一起使用，当maxbw设置为相对 (0) 时，计算恢复数据包与主媒体流一起发送时的最大发送速率： inputbw \* (100 + oheadbw ) / 100 如果在设置maxbw时未设置inputbw相对于 (0)，实际输入速率在库内部进行评估。默认值为 0。

iptos=托斯

IP 服务类型。仅适用于发件人。默认值为 0xB8。

ipttl= ttl

IP 生存时间。仅适用于发件人。默认值为 64。

延迟=微秒

基于时间戳的数据包传递延迟。用于吸收突发的丢失数据包重传。此标志将rcvlatency和peerlatency设置为相同的值。请注意，在 1.3.0 版本之前，这是唯一设置延迟的标志，但是这实际上等同于设置peerlatency ，当 side 是 sender 和rcvlatency ，当 side 是 receiver ，并且不支持双向流发送。

listen\_timeout=微秒

设置套接字侦听超时。

maxbw=字节/秒

最大发送带宽，以字节/秒为单位。 -1 infinite (CSRTCC limit is 30mbps) 0 relative to input rate (see inputbw ) >0 absolute limit value 默认值为 0 (relative)

模式=呼叫者|侦听器|会合点

连接方式。调用者打开客户端连接。侦听器启动服务器以侦听传入连接。 rendezvous使用 Rendez-Vous 连接模式。默认值为来电者。

mss=字节

最大段大小，以字节为单位。用于缓冲区分配和速率计算，使用数据包计数器假设数据包已满。使用对等点之间的最小 MSS。在整个互联网中默认为 1500。这是 UDP 数据包的最大大小，只能减小，除非您有一些不寻常的专用网络设置。默认值为 1500。

nakreport= 1|0

如果设置为 1，Receiver 将定期发送“UMSG\_LOSSREPORT”消息，直到丢失的数据包被重新传输或有意丢弃。默认值为 1。

oheadbw=百分比

高于输入速率的恢复带宽开销，以百分比表示。见输入bw 。默认值为 25%。

密码=字符串

HaiCrypt 加密/解密密码字符串，长度从 10 到 79 个字符。密码是发送者和接收者之间的共享秘密。它用于使用 PBKDF2（基于密码的密钥派生函数）生成密钥加密密钥。它仅在pbkeylen非零时使用。只有当接收到的数据被加密时，它才会在接收器上使用。配置的密码无法恢复（只写）。

强制加密= 1|0

如果为true，则连接双方必须设置相同的密码（包括空的，即不加密）。如果密码不匹配或只有一侧未加密，连接将被拒绝。默认为真。

kmrefreshrate=数据包

加密密钥切换为新密钥后要传输的数据包数。默认为 -1。 -1 表示自动（srt 库中为 0x1000000）。此选项的范围是 0 - INT\_MAX之间的整数。

kmpreannounce=数据包

发送新加密密钥与发生切换之间的时间间隔。该值也适用于发生切换和旧加密密钥停用之间的后续间隔。默认为 -1。 -1 表示自动（srt 库中为 0x1000）。此选项的范围是 0 - INT\_MAX之间的整数。

snddropdelay=微秒

发送方在丢弃数据包之前的额外延迟。此延迟添加到默认丢弃延迟时间间隔值。

特殊值 -1：根本不丢弃发送方的数据包。

payload\_size=字节

设置在实时模式下单次调用发送函数期间传输的数据包的最大声明大小。如果不使用此值，则使用 0（文件模式中的默认值）。默认为 -1（自动），通常表示 MPEG-TS；如果您要使用 SRT 发送任何不同类型的有效负载，例如将实时流包装在非常小的帧中，那么您可以使用更大的最大帧大小，但不超过 1456 字节。

pkt\_size=字节

payload\_size ”的别名。

对等延迟=微秒

由发送方设置为接收方最小值的延迟值（如rcvlatency中所述）。

pbkeylen=字节

发件人加密密钥长度，以字节为单位。只能设置为 0、16、24 和 32。如果不是 0，则启用发送方加密。接收方不需要（设置为 0），在 HaiCrypt 握手中从发送方获得的密钥大小。默认值为 0。

rcvlatency=微秒

从数据包发送到接收函数中将其交付给接收方应用程序的那一刻起应该经过的时间。这个时间应该是一个足够大的缓冲时间，足以覆盖发送所花费的时间、意外延长的 RTT 时间以及重新传输丢失的 UDP 数据包所需的时间。有效延迟值将是此选项的值和对等方设置的对等延迟值的最大值。在 1.3.0 版本之前，此选项仅作为latency可用。

recv\_buffer\_size=字节

设置 UDP 接收缓冲区大小，以字节为单位。

send\_buffer\_size=字节

设置 UDP 发送缓冲区大小，以字节为单位。

超时=微秒

为读取、写入和连接操作设置引发错误超时。请注意，SRT 库具有可以单独控制的内部超时，此处设置的值只是一个上限。

tlpktdrop= 1|0

太晚的数据包丢弃。在接收器上启用时，它会跳过未及时传送的丢失数据包，并在播放时间到来时将以下数据包传送给应用程序。它还向发件人发送伪造的 ACK。当在发送方启用并在接收方启用时，发送方将丢弃没有机会及时交付的旧数据包。如果接收方支持，它会在发送方中自动启用。

sndbuf=字节

设置发送缓冲区大小，以字节表示。

rcvbuf=字节

设置接收缓冲区大小，以字节表示。

接收缓冲区不得大于ffs 。

lossmaxttl=数据包

Reorder Tolerance 可能增长到的值。当 Reorder Tolerance > 0 时，数据包丢失报告会延迟，直到收到该数量的数据包。每次“迟来的”数据包到来时，Reorder Tolerance 都会增加，但这不是由于重传（即，当 UDP 数据包趋于乱序），最新序列和这个数据包的序列之间的差异，并且不超过这个选项的值。默认情况下它是 0，这意味着这个机制被关闭，丢失报告总是在序列中遇到“间隙”时立即发送。

最小反转

对等方要求的最低 SRT 版本。与不满足最低版本要求的对等点的连接将被拒绝。

十六进制的版本格式是 0xXXYYZZ for xyz 以人类可读的形式。

streamid=字符串

限制为 512 个字符的字符串，可以在连接之前在套接字上设置。该流 ID 将能够由侦听器端从 srt\_accept 返回的套接字中检索，并由具有该设置流 ID 的套接字连接。 SRT 不强制对此字符串的内容进行任何特殊解释。此选项在 Rendezvous 连接中没有意义；结果可能只是一方将覆盖另一方的价值，这取决于运气

srt\_streamid=字符串

streamid ”的别名以避免与 ffmpeg 命令行选项发生冲突。

平滑=活|文件

用于该套接字传输的平滑器类型，负责传输和拥塞控制。连接双方的Smoother类型必须完全一致，否则拒绝连接。

消息接口= 1|0

设置时，此套接字使用 Message API，否则使用 Buffer API。请注意，在实时模式下（请参阅transtype ）只有消息 API 可用。在文件模式下，您可以选择使用以下两种模式之一：

Stream API（默认，当此选项为 false 时）。在这种模式下，您可以使用一条发送指令发送任意数量的数据，甚至可以使用直接从文件中读取的专用函数。内部设施将负责任何速度和拥堵控制。接收时，也可以接收任意多的数据，未提取的数据将等待下一次调用。在 Stream 模式中，数据部分之间没有边界。

消息 API。在这种模式下，您的单个发送指令正好传递一个有边界的数据（一条消息）。与实时模式相反，此消息可能跨越多个 UDP 数据包，唯一的大小限制是它应作为一个整体放入发送缓冲区。接收方应使用尽可能大的缓冲区来接收消息，否则消息不会被放弃。当消息不完整时（没有收到所有数据包或有数据包丢失），它不会被放弃。

transtype=活|文件

设置套接字的传输类型，特别是，设置此选项会将多个其他参数设置为特定传输类型所需的默认值。

live：设置选项为实时传输。在这种模式下，您应该通过一个发送指令只发送适合一个 UDP 数据包的数据，并且限制为payload\_size中首先定义的值（此模式下默认为 1316）。此模式下没有速度控制，只有带宽控制（如果已配置），以便不超过开销传输（重传和控制数据包）的带宽。

文件：设置非实时传输的选项。请参阅messageapi以获取更多说明

逗留=秒

套接字在关闭时等待未发送数据的秒数。默认为 -1。 -1 表示自动（实时模式下 0 秒关闭，文件模式下 180 秒打开）。此选项的范围是 0 - INT\_MAX之间的整数。

tsbpd= 1|0

为真时，使用基于时间戳的数据包传送模式。默认行为取决于传输类型：在实时模式下启用，在文件模式下禁用。

有关详细信息，请参阅： <https://github.com/Haivision/srt>。

### 24.38 srtp

安全实时传输协议。

接受的选项是：

srtp\_in\_suite

srtp\_out\_suite

选择输入和输出编码套件。

支持的值：

' AES\_CM\_128\_HMAC\_SHA1\_80 '

' SRTP\_AES128\_CM\_HMAC\_SHA1\_80 '

' AES\_CM\_128\_HMAC\_SHA1\_32 '

' SRTP\_AES128\_CM\_HMAC\_SHA1\_32 '

srtp\_in\_params

srtp\_out\_params

设置输入和输出编码参数，这些参数由二进制块的 base64 编码表示形式表示。这个二进制块的前 16 个字节用作主密钥，后面的 14 个字节用作主盐。

### 24.39 子文件

虚拟地提取文件或另一个流的一部分。底层流必须是可搜索的。

接受的选项：

开始

提取段的起始偏移量，以字节为单位。

结尾

提取段的结束偏移量，以字节为单位。如果设置为 0，提取到文件末尾。

例子：

从 DVD VOB 文件中提取一个章节（开始和结束扇区从外部获得并乘以 2048）：

子文件,,开始,153391104,结束,268142592,,:/media/dvd/VIDEO\_TS/VTS\_08\_1.VOB

直接从 TAR 存档播放 AVI 文件：

子文件,,start,183241728,end,366490624,,:archive.tar

从开始偏移到结束播放 MPEG-TS 文件：

子文件,,开始,32815239,结束,0,,:video.ts

### 24.40 开球

将输出写入多个协议。各个输出由 | 分隔

三通：file://path/to/local/this.avi|file://path/to/local/that.avi

### 24.41 TCP

传输控制协议。

TCP url 所需的语法是：

tcp://主机名:端口[?选项]

options包含以 & 分隔的选项列表，格式为key = val 。

支持的选项列表如下。

听= 2|1|0

侦听传入连接。 0 禁用侦听，1 启用单客户端模式侦听，2 启用多客户端模式侦听。默认值为 0。

超时=微秒

设置引发错误超时，以微秒表示。

此选项仅与读取模式相关：如果超过此时间间隔没有数据到达，则引发错误。

listen\_timeout=毫秒

设置监听超时，以毫秒表示。

recv\_buffer\_size=字节

设置接收缓冲区大小，以字节表示。

send\_buffer\_size=字节

设置发送缓冲区大小，以字节表示。

tcp\_nodelay= 1|0

设置 TCP\_NODELAY 以禁用 Nagle 算法。默认值为 0。

备注：写入套接字目前未优化以最小化系统调用并降低 TCP\_NODELAY 的效率/效果。

tcp\_mss=字节

设置传出 TCP 数据包的最大段大小，以字节表示。

以下示例显示了如何使用ffmpeg设置监听 TCP 连接，然后使用ffplay访问该连接：

ffmpeg -i输入-f格式tcp://主机名:端口?listen

ffplay tcp://主机名:端口

### 24.42 吨

传输层安全性 (TLS) / 安全套接字层 (SSL)

TLS/SSL url 所需的语法是：

tls://主机名:端口[?选项]

AVOption s 在代码中）设置以下参数：

ca\_file, cafile=文件名

一个包含证书颁发机构 (CA) 根证书的文件，被视为受信任的。如果链接的 TLS 库包含默认值，则可能不需要为验证工作指定该默认值，但并非所有库和设置都内置默认值。该文件必须采用 OpenSSL PEM 格式。

tls\_verify= 1|0

如果启用，请尝试验证我们正在与之通信的对等方。请注意，如果使用 OpenSSL，这目前只能确保对等证书由 CA 数据库中的根证书之一签名，但它不会验证该证书是否与我们尝试连接的主机名实际匹配。 （对于其他后端，主机名也会被验证。）

默认情况下这是禁用的，因为在许多情况下它需要调用者提供 CA 数据库。

证书文件，证书=文件名

包含用于与对等方握手的证书的文件。 （当作为服务器运行时，在侦听模式下，对等方更经常需要这样做，而客户端证书仅在某些设置中是必需的。）

key\_file, key=文件名

包含证书私钥的文件。

听= 1 | 0

如果启用，侦听提供的端口上的连接，并在握手中承担服务器角色而不是客户端角色。

http\_proxy

要通过隧道的 HTTP 代理，例如http://example.com:1234 。代理必须支持 CONNECT 方法。

示例命令行：

创建一个为输入流提供服务的 TLS/SSL 服务器。

ffmpeg -i input -f format tls:// hostname : port ?listen&cert= server.crt &key= server.key

ffplay从 TLS/SSL 服务器播放流：

ffplay tls://主机名:端口

### 24.43 开发协议

用户数据报协议。

UDP URL 所需的语法是：

udp://主机名:端口[?选项]

options包含以 & 分隔的选项列表，格式为key = val 。

如果在系统上启用线程，循环缓冲区用于存储传入数据，这允许减少由于 UDP 套接字缓冲区溢出而导致的数据丢失。 fifo\_size和overrun\_nonfatal选项与此缓冲区相关。

支持的选项列表如下。

buffer\_size=大小

以字节为单位设置 UDP 最大套接字缓冲区大小。这用于设置接收或发送缓冲区大小，具体取决于套接字的用途。输出默认为 32 KB，输入默认为 384 KB。另见fifo\_size 。

比特率=比特率

如果设置为非零，如果输入有足够的数据包来维持输出将具有指定的恒定比特率。

burst\_bits=位

当使用比特率时，这指定了数据包突发中的最大位数。

本地端口=端口

覆盖要绑定的本地 UDP 端口。

localaddr=地址

用于发送数据包或加入多播组的网络接口的本地 IP 地址。

pkt\_size=尺寸

设置 UDP 数据包的大小（以字节为单位）。

重用= 1|0

明确允许或禁止重复使用 UDP 套接字。

ttl= ttl

设置生存时间值（仅适用于多播）。

连接= 1|0

connect()初始化 UDP 套接字。在这种情况下，目标地址不能在以后用 ff\_udp\_set\_remote\_url 改变。如果开始时不知道目标地址，也可以在 ff\_udp\_set\_remote\_url 中指定此选项。这允许找到带有 getsockname 的数据包的源地址，并且如果收到“目的地无法到达”则使写入返回 AVERROR(ECONNREFUSED)。对于接收，这提供了仅从指定的对等地址/端口接收数据包的好处。

来源=地址[，地址]

只接收从指定地址发送的数据包。在多播的情况下，也仅订阅来自这些地址的多播流量。

块=地址[，地址]

忽略从指定地址发送的数据包。在多播的情况下，还排除多播订阅中的源地址。

fifo\_size=单位

设置UDP 接收循环缓冲区大小，以188 字节大小的数据包数表示。如果未指定，默认为 7\*4096。

overrun\_nonfatal= 1|0

在 UDP 接收循环缓冲区溢出的情况下生存。默认值为 0。

超时=微秒

设置引发错误超时，以微秒表示。

此选项仅与读取模式相关：如果超过此时间间隔没有数据到达，则引发错误。

广播= 1|0

明确允许或禁止 UDP 广播。

请注意，广播可能无法在具有广播风暴保护的网络上正常工作。

* [例子](#Examples-21)

#### 24.43.1 例子

* 使用ffmpeg通过 UDP 流式传输到远程端点：
* ffmpeg -i输入-f格式udp://主机名:端口
* 使用ffmpeg使用 188 大小的 UDP 数据包通过 UDP 以 mpegts 格式流式传输，使用大输入缓冲区：
* ffmpeg -i input -f mpegts udp://主机名:端口?pkt\_size=188&buffer\_size=65535
* 使用ffmpeg从远程端点通过 UDP 接收：
* ffmpeg -i udp://[多播地址]:端口...

### 24.44 操作系统

Unix 本地套接字

Unix 套接字 URL 所需的语法是：

unix://文件路径

AVOption s 在代码中）设置以下参数：

暂停

以毫秒为单位的超时

听

在侦听模式下创建 Unix 套接字。

### 24.45 zmq

使用 libzmq 库的 ZeroMQ 异步消息传递。

该库支持单播流式传输到多个客户端，而不依赖于外部服务器。

流式传输或连接到流所需的语法是：

zmq:tcp://ip地址:端口

示例：在端口 5555 上创建本地主机流：

ffmpeg -re -i 输入 -f mpegts zmq:tcp://127.0.0.1:5555

多个客户端可以使用以下方式连接到流：

ffplay zmq:tcp://127.0.0.1:5555

流式传输到多个客户端是使用 ZeroMQ Pub-Sub 模式实现的。服务器端绑定到一个端口并发布数据。客户端连接到服务器（通过 IP 地址/端口）并订阅流。服务器和客户端启动的顺序通常无关紧要。

ffmpeg 必须使用 –enable-libzmq 选项编译以支持此协议。

ffmpeg / ffplay命令行上设置选项。支持以下选项：

pkt\_size

强制发送/接收数据的最大数据包大小。默认值为 131,072 字节。在服务器端，这设置了通过 ZeroMQ 发送的数据包的最大大小。在客户端上，它设置用于接收数据包的内部缓冲区大小。请注意，客户端上的 pkt\_size 应等于或大于服务器上的 pkt\_size。否则，接收到的消息可能会被截断，从而导致解码错误。

## 25 设备选项

libavdevice 库提供与 libavformat 相同的接口。即，输入设备被视为多路分解器，输出设备被视为多路复用器，接口和通用设备选项与 libavformat 提供的相同（请参阅 ffmpeg-formats 手册）。

此外，每个输入或输出设备都可以支持所谓的私有选项，这些选项特定于该组件。

-option来设置选项 FFmpeg 工具中的值，或通过在设备AVFormatContext选项中显式设置值或使用libavutil/opt.h API 进行编程使用。

## 26个输入设备

输入设备是 FFmpeg 中的配置元素，可以访问来自连接到系统的多媒体设备的数据。

当您配置 FFmpeg 构建时，默认情况下会启用所有受支持的输入设备。您可以使用配置选项“–list-indevs”列出所有可用的。

INDEV ”有选择地启用输入设备，或者您可以使用选项“–disable”禁用特定输入设备-indev=独立”。

ff\* 工具的选项“-devices”将显示支持的输入设备列表。

以下是对当前可用输入设备的描述。

* [阿尔萨](#alsa)
* [安卓相机](#android_005fcamera)
* [理由](#avfoundation)
* [回车](#bktr)
* [甲板链接](#decklink)
* [显示](#dshow)
* [开发者](#fbdev)
* [gdigrab](#gdigrab)
* [IEC61883](#iec61883)
* [杰克](#jack)
* [公里数抓取](#kmsgrab)
* [拉夫菲](#lavfi)
* [libcdio](#libcdio)
* [libdc1394](#libdc1394)
* [公开的](#openal)
* [oss](#oss)
* [脉冲](#pulse)
* [sndio](#sndio)
* [video4linux2，v4l2](#video4linux2_002c-v4l2)
* [vfwcap](#vfwcap)
* [x11抢](#x11grab)

### 26.1 阿尔萨

ALSA（高级 Linux 声音架构）输入设备。

要在配置期间启用此输入设备，您需要在系统上安装 libasound。

该设备允许从 ALSA 设备捕获。要捕获的设备名称必须是 ALSA 卡标识符。

ALSA 标识符的语法如下：

hw: CARD [, DEV [, SUBDEV ]]

其中DEV和SUBDEV组件是可选的。

三个参数（按顺序： CARD 、 DEV 、 SUBDEV ）指定卡号或标识符、设备号和子设备号（-1 表示任意）。

要查看系统当前识别的卡列表，请检查文件/proc/asound/cards和/proc/asound/devices 。

例如，要使用ffmpeg从卡 ID 为 0 的 ALSA 设备捕获，您可以运行以下命令：

ffmpeg -f alsa -i hw:0 alsaout.wav

有关详细信息，请参阅： <http://www.alsa-project.org/alsa-doc/alsa-lib/pcm.html>

* [选项](#Options-64)

#### 26.1.1 选项

采样率

以 Hz 为单位设置采样率。默认值为 48000。

频道

设置通道数。默认值为 2。

### 26.2 android\_camera

安卓摄像头输入设备。

此输入设备使用 Android Camera2 NDK API，可在 API 级别 24+ 的设备上使用。 android\_camera 的可用性在配置期间自动检测。

此设备允许从 Android 设备上的所有摄像头进行捕获，这些摄像头已集成到 Camera2 NDK API 中。

可用的摄像头在内部枚举，可以使用camera\_index参数进行选择。输入文件字符串被丢弃。

通常后置摄像头的索引为 0，而前置摄像头的索引为 1。

* [选项](#Options-65)

#### 26.2.1 选项

视频大小

设置以字符串形式给出的视频大小，例如 640x480 或 hd720。如果请求的视频大小不可用或默认情况下，回退到 Android 报告的第一个可用配置。

帧率

设置视频帧率。如果请求的帧率不可用或默认情况下 (-1)，则回退到 Android 报告的第一个可用配置。

相机索引

设置要使用的相机的索引。默认为 0。

输入队列大小

设置要缓冲的最大帧数。默认值为 5。

### 26.3 基础

AVFoundation 输入设备。

AVFoundation 是 Apple 目前推荐的框架，用于在 OSX >= 10.7 以及 iOS 上进行流媒体抓取。

输入文件名必须按以下语法给出：

-i "[[视频]:[音频]]"

第一个条目选择视频输入，而后者选择音频输入。流必须由设备名称或设备索引指定，如设备列表所示。或者，可以使用-video\_device\_index <INDEX>和/或-audio\_device\_index <INDEX>按索引选择视频和/或音频输入设备，覆盖输入文件名中给出的任何设备名称或索引。

可以使用-list\_devices true枚举所有可用设备，列出所有设备名称和相应的索引。

有两种设备名称别名：

默认

选择对应类型的AVFoundation默认设备。

没有任何

不记录相应的媒体类型。这等同于指定一个空的设备名称或索引。

* [选项](#Options-66)
* [例子](#Examples-22)

#### 26.3.1 选项

AVFoundation 支持以下选项：

-list\_devices <TRUE|FALSE>

如果设置为 true，则会给出所有可用输入设备的列表，显示所有设备名称和索引。

-video\_device\_index <索引>

通过索引指定视频设备。覆盖输入文件名中给定的任何内容。

-audio\_device\_index <索引>

通过索引指定音频设备。覆盖输入文件名中给定的任何内容。

-pixel\_format <格式>

请求视频设备使用特定的像素格式。如果不支持指定的格式，则会给出可用格式的列表，并使用此列表中的第一个格式。可用的像素格式有： monob、rgb555be、rgb555le、rgb565be、rgb565le、rgb24、bgr24、0rgb、bgr0、0bgr、rgb0、bgr48be、uyvy422、yuva444p、yuva444p16le、yuv444p、yuv422p 16、yuv422p10、yuv444p10、yuv420p、nv12、yuyv422、灰色

-帧率

设置抓取帧速率。默认为ntsc ，对应于30000/1001的帧速率。

-video\_size

设置视频帧大小。

-capture\_cursor

捕获鼠标指针。默认为 0。

-capture\_mouse\_clicks

捕捉屏幕鼠标点击。默认为 0。

-capture\_raw\_data

捕获原始设备数据。默认值为 0。使用此选项可能会导致接收传送到 AVFoundation 框架的基础数据。例如，对于将原始 DV 数据发送到框架的多路复用设备（如基于磁带的摄像机），将此选项设置为 false 会导致提取的视频帧仅以指定的像素格式捕获。将此选项设置为 true 会导致接收未受影响的原始 DV 流。

#### 26.3.2 例子

* 打印 AVFoundation 支持的设备列表并退出：
* $ ffmpeg -f avfoundation -list\_devices true -i ""
* 将视频设备 0 的视频和音频设备 0 的音频录制到 out.avi 中：
* $ ffmpeg -f avfoundation -i "0:0" out.avi
* 将视频设备 2 的视频和音频设备 1 的音频录制到 out.avi 中：
* $ ffmpeg -f avfoundation -video\_device\_index 2 -i ":1" out.avi
* 使用像素格式 bgr0 从系统默认视频设备录制视频，并且不将任何音频录制到 out.avi 中：
* $ ffmpeg -f avfoundation -pixel\_format bgr0 -i "default:none" out.avi
* 从合适的输入设备记录原始 DV 数据并将输出写入 out.dv：
* $ ffmpeg -f avfoundation -capture\_raw\_data true -i "zr100:none" out.dv

### 26.4 桶

BSD 视频输入设备。

* [选项](#Options-67)

#### 26.4.1 选项

帧率

设置帧速率。

视频大小

设置视频帧大小。默认是vga 。

标准

可用值是：

'朋友'

' ntsc '

'骗局'

' paln '

'手掌'

' ntscj '

### 26.5 甲板链接

decklink 输入设备为 Blackmagic DeckLink 设备提供采集功能。

要启用此输入设备，您需要 Blackmagic DeckLink SDK 并且需要使用适当的--extra-cflags和--extra-ldflags进行配置。在 Windows 上，您需要通过widl运行 IDL 文件。

DeckLink 对其支持的格式非常挑剔。输入的像素格式可以用raw\_format设置。必须使用-list\_formats 1为您的设备确定帧速率和视频大小。音频采样率始终为 48 kHz，通道数可以是 2、8 或 16。请注意，所有音频通道都捆绑在一个单独的音轨中。

* [选项](#Options-68)
* [例子](#Examples-23)

#### 26.5.1 选项

列表设备

如果设置为true ，打印设备列表并退出。默认为假。此选项已弃用，请使用ffmpeg 的-sources选项列出可用的输入设备。

列表格式

如果设置为true ，则打印支持的格式列表并退出。默认为假。

format\_code <FourCC>

这会将输入视频格式设置为 FourCC 给出的格式。要查看您的设备支持的值，请使用list\_formats 。请注意，有一个 FourCC 'pal'也可以用作pal （3 个字母）。如果硬件支持，默认行为是自动检测输入视频格式。

原始格式

设置捕获视频的像素格式。可用值是：

'汽车'

这是默认值，如果使用格式自动检测，则表示 8 位 YUV 422 或 8 位 ARGB，否则为 8 位 YUV 422。

' uyvy422 '

8 位 YUV 422。

' yuv422p10 '

10 位 YUV 422。

' argb '

8 位 RGB。

' bgra '

8 位 RGB。

' rgb10 '

10 位 RGB。

图文电视线路

如果设置为非零，将从垂直辅助数据中捕获额外的图文电视流。支持标清 PAL (576i) 和高清（1080i 或 1080p）信号源。如果是高清源，则解码 OP47 数据包。

此选项是捕获的 SD PAL VBI 行的位掩码，特别是第 6 到 22 行以及第 318 到 335 行。第 6 行是掩码中的 LSB。不包含图文电视信息的所选行将被忽略。您可以使用特殊的all常量来选择所有可能的行，或者使用 standard来跳过第 6、318 和 319 行，这些行与所有接收器都不兼容。

--enable-libzvbi进行编译。对于 HD 信号源，在较旧的（4K 之前）DeckLink 卡型号上，您必须以 10 位模式采集。

频道

定义要捕获的音频通道数。必须是“ 2 ”、“ 8 ”或“ 16 ”。默认为“ 2 ”。

双工模式

设置 decklink 设备双工/配置文件模式。必须是“ unset ”、“ half ”、“ full ”、“ one\_sub\_device\_full ”、“ one\_sub\_device\_half ”、“ two\_sub\_device\_full ”、“ four\_sub\_device\_half ”默认为“ unset ”。

注意：DeckLink SDK 11.0 已将双工属性替换为配置文件属性。对于 DeckLink Duo 2 和 DeckLink Quad 2，配置文件在使用相同连接器的任何 2 个子设备之间共享。对于 DeckLink 8K Pro，一个配置文件在所有 4 个子设备之间共享。所以 DeckLink 8K Pro 支持四种配置文件。

DeckLink 8K Pro（使用 DeckLink SDK >= 11.0）的有效配置文件模式：' one\_sub\_device\_full '、' one\_sub\_device\_half '、' two\_sub\_device\_full '、' four\_sub\_device\_half '

DeckLink Quad 2 和 DeckLink Duo 2 的有效配置文件模式：“ half ”、“ full ”

时间码格式

要包含在帧和视频流元数据中的时间码类型。必须是“ none ”、“ rp188vitc ”、“ rp188vitc2 ”、“ rp188ltc ”、“ rp188hfr ”、 “ rp188any ”、“ vitc ”、“ vitc2 ”或“ serial ”。默认为“无”（不包括在内）。

为了正确支持 50/60 fps 时间码，“ rp188any ”的查询时间码类型的顺序是 HFR、VITC1、VITC2 和 >30 fps 内容的 LTC。请注意，这与 DeckLink API 使用的顺序略有不同，即 HFR、VITC1、LTC、VITC2。

视频输入

设置视频输入源。必须是“ unset ”、“ sdi ”、“ hdmi ”、“ optical\_sdi ”、“ component ”、“ composite ”或“ s\_video ”。默认为“未设置”。

音频输入

设置音频输入源。必须是“ unset ”、“ embedded ”、“ aes\_ebu ”、“ analog ”、“ analog\_xlr ”、“ analog\_rca ”或“ microphone ”。默认为“未设置”。

video\_pts

设置视频包时间戳来源。必须是“ video ”、“ audio ”、“ reference ”、“ wallclock ”或“ abs\_wallclock ”。默认为“视频”。

audio\_pts

设置音频数据包时间戳源。必须是“ video ”、“ audio ”、“ reference ”、“ wallclock ”或“ abs\_wallclock ”。默认为“音频”。

拉杆

如果设置为“ true ”，则在信号丢失的情况下绘制彩条。默认为“真”。

队列大小

以字节为单位设置最大输入缓冲区大小。如果缓冲达到此值，传入的帧将被丢弃。默认为“ 1073741824 ”。

音频深度

设置音频样本位深度。必须是“ 16 ”或“ 32 ”。默认为“ 16 ”。

decklink\_copyts

如果设置为true ，则按原样转发时间戳而不删除初始偏移量。默认为假。

时间戳对齐

以秒为单位捕获开始时间对齐。如果设置为非零，输入帧将被丢弃，直到系统时间戳与配置值对齐。最多一帧持续时间的对齐差异是可以容忍的。这对于在为“N 路”冗余部署的 N 个不同硬件设备之间保持输入同步很有用。使用该选项前，需要将不同硬件设备的系统时间与NTP或PTP等协议同步。请注意，此方法并非万无一失。在某些边界情况下，由于操作系统中的线程调度抖动，输入同步可能不会发生。同步可能会出错 1 帧，或者在极少数情况下会出错timestamp\_align秒。默认为“ 0 ”。

wait\_for\_tc (布尔)

丢帧直到接收到带有时间码的帧。有时第一个输入帧未收到串行时间码。如果发生这种情况，存储的流时间码将不准确。如果此选项设置为true ，输入帧将被丢弃，直到收到带有时间码的帧。必须指定选项timecode\_format 。默认为假。

enable\_klv（布尔）

如果设置为true ，则从 VANC 中提取 KLV 数据并输出 KLV 数据包。 KLV VANC 数据包根据 MID 和 PSC 字段连接并聚合为一个 KLV 数据包。默认为假。

#### 26.5.2 例子

* 列出输入设备：
* ffmpeg -sources 甲板链接
* 列出支持的格式：
* ffmpeg -f decklink -list\_formats 1 -i 'Intensity Pro'
* 以 1080i50 捕获视频剪辑：
* ffmpeg -format\_code Hi50 -f decklink -i 'Intensity Pro' -c:a 复制 -c:v 复制输出.avi
* 以 1080i50 10 位捕获视频剪辑：
* ffmpeg -raw\_format yuv422p10 -format\_code Hi50 -f decklink -i 'UltraStudio Mini Recorder' -c:a copy -c:v 复制输出.avi
* 使用 16 个音频通道捕获 1080i50 的视频剪辑：
* ffmpeg -channels 16 -format\_code Hi50 -f decklink -i 'UltraStudio Mini Recorder' -c:a copy -c:v 复制输出.avi

### 26.6 显示

Windows DirectShow 输入设备。

使用 mingw-w64 项目构建 FFmpeg 时启用 DirectShow 支持。目前仅支持音频和视频设备。

多个设备可以作为单独的输入打开，但它们也可以在同一输入上打开，这应该会提高它们之间的同步性。

输入名称的格式应为：

类型=名称[:类型=名称]

其中TYPE可以是audio或video ， NAME是设备的名称或替代名称..

* [选项](#Options-69)
* [例子](#Examples-24)

#### 26.6.1 选项

如果未指定任何选项，则使用设备的默认值。如果设备不支持请求的选项，它将无法打开。

视频大小

在捕获的视频中设置视频大小。

帧率

在捕获的视频中设置帧速率。

采样率

设置捕获音频的采样率（以 Hz 为单位）。

样本大小

设置捕获音频的样本大小（以位为单位）。

频道

设置捕获音频中的通道数。

列表设备

如果设置为true ，打印设备列表并退出。

列表选项

如果设置为true ，则打印所选设备选项的列表并退出。

video\_device\_number

为具有相同名称的设备设置视频设备编号（从 0 开始，默认为 0）。

音频设备编号

为具有相同名称的设备设置音频设备编号（从 0 开始，默认为 0）。

像素格式

选择 DirectShow 使用的像素格式。这可能仅在视频编解码器未设置或设置为 rawvideo 时设置。

音频缓冲区大小

以毫秒为单位设置音频设备缓冲区大小（这会直接影响延迟，具体取决于设备）。默认使用音频设备的默认缓冲区大小（通常是 500 毫秒的倍数）。将此值设置得太低会降低性能。另请参阅<http://msdn.microsoft.com/en-us/library/windows/desktop/dd377582(v=vs.85).aspx>

video\_pin\_name

选择要按名称或备用名称使用的视频捕获引脚。

audio\_pin\_name

选择要按名称或备用名称使用的音频捕获引脚。

crossbar\_video\_input\_pin\_number

选择交叉开关设备的视频输入引脚号。这将被路由到交叉开关设备的视频解码器输出引脚。请注意，更改此值可能会影响将来的调用（设置新的默认值），直到系统重新启动。

crossbar\_audio\_input\_pin\_number

选择交叉开关设备的音频输入引脚号。这将被路由到交叉开关设备的音频解码器输出引脚。请注意，更改此值可能会影响将来的调用（设置新的默认值），直到系统重新启动。

显示视频设备对话框

如果设置为true ，则在捕获开始之前，向最终用户弹出一个显示对话框，允许他们手动更改视频过滤器属性和配置。请注意，对于交叉开关设备，有时可能需要调整此对话框中的值以在 PAL (25 fps) 和 NTSC (29.97) 输入帧速率、大小、交错等之间切换。更改这些值可以启用不同的扫描速率/帧速率并避免底部的绿色条、闪烁的扫描线等。请注意，对于某些设备，更改这些属性也会影响未来的调用（设置新的默认值），直到系统重新启动。

显示音频设备对话框

如果设置为true ，则在捕获开始之前，向最终用户弹出一个显示对话框，允许他们手动更改音频过滤器属性和配置。

show\_video\_crossbar\_connection\_dialog

如果设置为true ，则在捕获开始之前，向最终用户弹出一个显示对话框，允许他们在打开视频设备时手动修改交叉开关引脚路由。

show\_audio\_crossbar\_connection\_dialog

如果设置为true ，则在捕获开始之前，向最终用户弹出一个显示对话框，允许他们在打开音频设备时手动修改交叉开关引脚路由。

show\_analog\_tv\_tuner\_dialog

如果设置为true ，则在捕获开始之前，向最终用户弹出一个显示对话框，允许他们手动修改电视频道和频率。

show\_analog\_tv\_tuner\_audio\_dialog

如果设置为true ，则在捕获开始之前，向最终用户弹出一个显示对话框，允许他们手动修改电视音频（如单声道与立体声、语言 A、B 或 C）。

音频设备负载

从文件加载音频捕获过滤器设备而不是按名称搜索它。如果过滤器支持将其属性序列化为，它也可以加载其他参数。要使用它，必须指定音频捕获源，但它可以是任何东西，甚至是假的。

音频设备保存

将当前使用的音频捕获过滤器设备及其参数（如果过滤器支持）保存到文件中。如果存在同名文件，它将被覆盖。

video\_device\_load

从文件加载视频捕获过滤器设备而不是按名称搜索。如果过滤器支持将其属性序列化为，它也可以加载其他参数。要使用它，必须指定视频捕获源，但它可以是任何东西，甚至是假的。

video\_device\_save

将当前使用的视频捕获过滤器设备及其参数（如果过滤器支持）保存到文件中。如果存在同名文件，它将被覆盖。

使用\_video\_device\_timestamps

如果设置为false ，视频帧的时间戳将来自挂钟而不是捕获设备提供的时间戳。这允许解决提供不可靠时间戳的设备。

#### 26.6.2 例子

* 打印 DirectShow 支持的设备列表并退出：
* $ ffmpeg -list\_devices true -f dshow -i dummy
* 打开视频设备相机：
* $ ffmpeg -f dshow -i video="相机"
* 打开第二个名为Camera的视频设备：
* $ ffmpeg -f dshow -video\_device\_number 1 -i video="相机"
* 打开视频设备摄像头和音频设备麦克风：
* $ ffmpeg -f dshow -i video="相机":audio="麦克风"
* 打印所选设备中支持的选项列表并退出：
* $ ffmpeg -list\_options true -f dshow -i video="相机"
* 按名称或备用名称指定要捕获的引脚名称，指定备用设备名称：
* $ ffmpeg -f dshow -audio\_pin\_name "音频输出" -video\_pin\_name 2 -i video=video="@device\_pnp\_\\?\pci#ven\_1a0a&dev\_6200&subsys\_62021461&rev\_01#4&e2c7dd6&0&00e1#{65e8773d-8f56-11d0- a3b9-00a0c9223196}\{ca465100-deb0- 4d59-818f-8c477184adf6}":audio="麦克风"
* 配置交叉开关设备，指定交叉开关引脚，允许用户在启动时调整视频捕获属性：
* $ ffmpeg -f dshow -show\_video\_device\_dialog true -crossbar\_video\_input\_pin\_number 0
* -crossbar\_audio\_input\_pin\_number 3 -i video="圆刚BDA模拟采集":audio="圆刚BDA模拟采集"

### 26.7 fbdev

Linux 帧缓冲输入设备。

Linux 帧缓冲区是一个独立于图形硬件的抽象层，用于在计算机显示器（通常是控制台）上显示图形。它通过文件设备节点访问，通常是/dev/fb0 。

有关更多详细信息，请阅读 Linux 源代码树中包含的文件 Documentation/fb/framebuffer.txt。

另请参见<http://linux-fbdev.sourceforge.net/>和 fbset(1)。

使用ffmpeg从帧缓冲设备/dev/fb0进行记录：

ffmpeg -f fbdev -framerate 10 -i /dev/fb0 out.avi

您可以使用以下命令获取单个屏幕截图图像：

ffmpeg -f fbdev -framerate 1 -i /dev/fb0 -frames:v 1 截图.jpeg

* [选项](#Options-70)

#### 26.7.1 选项

帧率

设置帧速率。默认值为 25。

### 26.8 gdigrab

基于 Win32 GDI 的屏幕捕获设备。

此设备允许您捕获 Windows 上的显示区域。

输入文件名有两个选项：

桌面

或者

标题= window\_title

第一个选项将捕获整个桌面或桌面的固定区域。第二个选项将捕获单个窗口的内容，而不管它在屏幕上的位置。

例如，要使用ffmpeg抓取整个桌面：

ffmpeg -f gdigrab -framerate 6 -i 桌面输出.mpg

10,20抓取一个 640x480 区域：

ffmpeg -f gdigrab -framerate 6 -offset\_x 10 -offset\_y 20 -video\_size vga -i desktop out.mpg

抓取名为“计算器”的窗口的内容

ffmpeg -f gdigrab -framerate 6 -i title=计算器输出.mpg

* [选项](#Options-71)

#### 26.8.1 选项

画鼠标

指定是否绘制鼠标指针。使用值0不绘制指针。默认值为1 。

帧率

设置抓取帧速率。默认值为ntsc ，对应于帧速率30000/1001 。

显示区域

在屏幕上显示抓取的区域。

如果show\_region指定为1 ，则抓取区域将显示在屏幕上。使用此选项，如果仅抓取了屏幕的一部分，则很容易知道抓取的是什么。

请注意， show\_region与抓取单个窗口的内容不兼容。

例如：

ffmpeg -f gdigrab -show\_region 1 -framerate 6 -video\_size cif -offset\_x 10 -offset\_y 20 -i desktop out.mpg

视频大小

设置视频帧大小。如果选择桌面，则默认为捕获全屏；如果选择title= window\_title ，则默认为捕获全屏。

偏移量\_x

video\_size捕获区域时，设置与屏幕或桌面左边缘的距离。

请注意，偏移计算是从 Windows 上主监视器的左上角开始的。如果您有一个监视器位于主监视器的左侧，您将需要使用负的offset\_x值将区域移动到该监视器。

偏移量\_y

video\_size捕获区域时，设置与屏幕或桌面上边缘的距离。

请注意，偏移计算是从 Windows 上主监视器的左上角开始的。如果您的主显示器上方有一个显示器，您将需要使用负的offset\_y值将区域移动到该显示器。

### 26.9 iec61883

使用 libiec61883 的火线 DV/HDV 输入设备。

要启用此输入设备，您需要在系统上安装 libiec61883、libraw1394 和 libavc1394。使用配置选项--enable-libiec61883在启用设备的情况下进行编译。

iec61883 捕获设备支持从通过 IEEE1394 (FireWire) 连接的视频设备捕获，使用 libiec61883 和新的 Linux FireWire 堆栈 (juju)。这是 Linux 内核 2.6.37 及更高版本中的默认 DV/HDV 输入方法，因为旧的 FireWire 堆栈已被删除。

指定要用作输入文件的 FireWire 端口，或“自动”以选择连接的第一个端口。

* [选项](#Options-72)
* [例子](#Examples-25)

#### 26.9.1 选项

数据类型

覆盖 DV/HDV 的自动检测。仅当自动检测不起作用或应禁止使用其他设备类型时才应使用此方法。将 DV 设备视为 HDV（反之亦然）将不起作用并导致未定义的行为。支持值auto 、 dv和hdv 。

缓冲区

为传入数据设置缓冲区的最大大小，以帧为单位。对于 DV，这是一个精确值。对于 HDV，它不是精确的帧，因为 HDV 没有固定的帧大小。

dvguid

通过指定其 GUID 选择捕获设备。捕获将仅从指定设备执行，如果未找到具有给定 GUID 的设备，则捕获失败。如果同时连接多个设备，这对于选择输入很有用。查看 /sys/bus/firewire/devices 以找出 GUID。

#### 26.9.2 例子

* 抓取并显示 FireWire DV/HDV 设备的输入。
* ffplay -f iec61883 -i 自动
* 抓取并记录 FireWire DV/HDV 设备的输入，如果源是 HDV，则使用 100000 个数据包的数据包缓冲区。
* ffmpeg -f iec61883 -i auto -dvbuffer 100000 输出.mpg

### 26.10 千斤顶

杰克输入设备。

要在配置期间启用此输入设备，您需要在系统上安装 libjack。

JACK 输入设备创建一个或多个 JACK 可写客户端，每个音频通道一个，名称为client\_name :input\_ N ，其中client\_name是应用程序提供的名称， N是标识通道的数字。每个可写客户端将获取的数据发送到 FFmpeg 输入设备。

一旦创建了一个或多个 JACK 可读客户端，就需要将它们连接到一个或多个 JACK 可写客户端。

要连接或断开 JACK 客户端，您可以使用jack\_connect和jack\_disconnect程序，或通过图形界面进行，例如使用qjackctl 。

要列出 JACK 客户端及其属性，您可以调用命令jack\_lsp 。

遵循一个示例，该示例显示如何使用ffmpeg捕获 JACK 可读客户端。

# 创建一个名为“ffmpeg”的 JACK 可写客户端。

$ ffmpeg -f jack -i ffmpeg -y 输出.wav

# 启动示例 jack\_metro 可读客户端。

$ jack\_metro -b 120 -d 0.2 -f 4000

# 列出当前的 JACK 客户端。

$ jack\_lsp -c

系统：捕获\_1

系统：捕获\_2

系统：回放\_1

系统：playback\_2

ffmpeg：输入\_1

地铁：120\_bpm

# 将 metro 连接到 ffmpeg 可写客户端。

$ jack\_connect metro:120\_bpm ffmpeg:input\_1

有关详细信息，请阅读： http: [//jackaudio.org/](http://jackaudio.org/)

* [选项](#Options-73)

#### 26.10.1 选项

频道

设置通道数。默认值为 2。

### 26.11 kmsgrab

KMS 视频输入设备。

捕获与指定 CRTC 或平面相关联的 KMS 扫描帧缓冲区作为可以传递给其他硬件功能的 DRM 对象。

需要 DRM 主机或 CAP\_SYS\_ADMIN 才能运行。

如果你不明白这一切意味着什么，你可能不想要这个。请查看x11grab 。

* [选项](#Options-74)
* [例子](#Examples-26)

#### 26.11.1 选项

设备

要捕获的 DRM 设备。默认为/dev/dri/card0 。

格式

帧缓冲区的像素格式。如果您运行的是 Linux 5.7 或更高版本，这可以自动检测到，但需要为早期版本提供。默认为bgr0 ，这是 Linux 控制台和 Xorg X 服务器最常用的格式。

格式修饰符

用于在输出帧上发出信号的格式修饰符。这是正确导入某些 API 所必需的。如果您运行的是 Linux 5.7 或更高版本，它可以被自动检测到，但在早期版本中需要时需要明确提供。有关可能的值，请参阅 libdrm 文档。

crtc\_id

用于定义捕获源的 KMS CRTC ID。将使用给定 CRTC 上的第一个活动平面。

平面编号

用于定义捕获源的 KMS 平面 ID。如果crtc\_id和plane\_id均未指定，则默认为找到的第一个活动平面。

帧率

要捕获的帧率。这不会与任何页面翻转或帧缓冲区更改同步 - 它只是定义帧缓冲区采样的时间间隔。比帧缓冲区更新速率更快的采样将生成具有相同内容的独立帧。默认为30 。

#### 26.11.2 例子

* 从第一个活动平面捕获，将结果下载到正常帧并进行编码。这仅在帧缓冲区既是线性的又是可映射的情况下才有效——否则，结果可能会被扰乱或无法下载。
* ffmpeg -f kmsgrab -i - -vf 'hwdownload,format=bgr0' 输出.mp4
* 以 60fps 从 CRTC ID 42 捕获，将结果映射到 VAAPI，转换为 NV12 并编码为 H.264。
* ffmpeg -crtc\_id 42 -framerate 60 -f kmsgrab -i - -vf 'hwmap=derive\_device=vaapi,scale\_vaapi=w=1920:h=1080:format=nv12' -c:v h264\_vaapi 输出.mp4
* 要仅捕获平面的一部分，可以裁剪输出 - 这可用于捕获单个窗口，只要它具有已知的绝对位置和大小即可。例如，要捕获和编码 1920x1080 平面的中间四分之一：
* ffmpeg -f kmsgrab -i - -vf 'hwmap=derive\_device=vaapi,crop=960:540:480:270,scale\_vaapi=960:540:nv12' -c:v h264\_vaapi 输出.mp4

### 26.12 拉菲

Libavfilter 输入虚拟设备。

此输入设备从 libavfilter 过滤器图的打开输出垫读取数据。

对于每个 filtergraph 打开输出，输入设备将创建一个对应的流，该流映射到生成的输出。目前仅支持视频数据。 filtergraph 通过选项graph指定。

* [选项](#Options-75)
* [例子](#Examples-27)

#### 26.12.1 选项

图形

指定用作输入的过滤图。每个视频打开输出必须用“out N ”形式的唯一字符串标记，其中N是从 0 开始的数字，对应于设备生成的映射输入流。第一个未标记的输出自动分配给“out0”标签，但所有其他输出都需要明确指定。

可以将后缀“+subcc”附加到输出标签以创建额外的流，其中隐藏式字幕数据包附加到该输出（实验性的；目前仅适用于 EIA-608 / CEA-708）。 subcc 流是在所有普通流之后创建的，按照相应流的顺序。例如，如果存在“out19+subcc”、“out7+subcc”直至“out42”，则流#43 是流#7 的subcc，流#44 是流#19 的subcc。

如果未指定，则默认为为输入设备指定的文件名。

图文件

设置要读取并发送到其他过滤器的过滤器图的文件名。 filtergraph 的语法与选项graph指定的语法相同。

垃圾图

将图表转储到标准错误。

#### 26.12.2 例子

* ffplay播放它：
* ffplay -f lavfi -graph "color=c=pink [out0]" 虚拟
* 与前面的示例一样，但使用文件名来指定图形描述，并省略“out0”标签：
* ffplay -f lavfi 颜色=c=粉色
* 创建三个不同的视频测试过滤源并播放它们：
* ffplay -f lavfi -graph "testsrc [out0]; testsrc,hflip [out1]; testsrc,negate [out2]" test3
* ffplay播放它：
* ffplay -f lavfi "amovie=test.wav"
* ffplay播放：
* ffplay -f lavfi "movie=test.avi[out0];amovie=test.wav[out1]"
* 将解码帧转储到图像并将隐藏式字幕转储到文件（实验性）：
* ffmpeg -f lavfi -i "movie=test.ts[out0+subcc]" -map v frame%08d.png -map s -c copy -f rawvideo subcc.bin

### 26.13 libcdio

基于 libcdio 的音频 CD 输入设备。

要在配置期间启用此输入设备，您需要在系统上安装 libcdio。它需要配置选项--enable-libcdio 。

该设备允许播放和抓取音频 CD。

例如，要使用ffmpeg复制/dev/sr0中的整个音频 CD ，您可以运行以下命令：

ffmpeg -f libcdio -i /dev/sr0 cd.wav

* [选项](#Options-76)

#### 26.13.1 选项

速度

设置驱动器读取速度。默认值为 0。

速度是指定光盘速度的单位。速度通过 libcdio cdio\_cddap\_speed\_set函数设置。在许多 CD-ROM 驱动器上，指定一个太大的值将导致使用最快的速度。

偏执模式

设置偏执恢复模式标志。它接受以下值之一：

'禁用'

'验证'

'重叠'

'永不跳过'

'满'

默认值为“禁用”。

有关可用恢复模式的更多信息，请参阅 paranoia 项目文档。

### 26.14 libdc1394

IIDC1394输入设备，基于libdc1394和libraw1394。

需要配置选项--enable-libdc1394 。

* [选项](#Options-77)

#### 26.14.1 选项

帧率

设置帧速率。默认为ntsc ，对应于30000/1001的帧速率。

像素格式

选择像素格式。默认为uyvy422 。

视频大小

设置以字符串形式给出的视频大小，例如640x480或hd720 。默认为qvga 。

### 26.15开

OpenAL 输入设备在所有具有有效 OpenAL 1.1 实现的系统上提供音频捕获。

--enable-openal配置 FFmpeg 。

OpenAL 标头和库应作为 OpenAL 实现的一部分或作为额外下载（SDK）提供。根据您的安装，您可能需要通过--extra-cflags和--extra-ldflags指定额外的标志，以允许构建系统定位 OpenAL 标头和库。

OpenAL 实现的不完整列表如下：

**有创造力的**

官方 Windows 实现，通过支持的设备和软件回退提供硬件加速。请参阅<http://openal.org/>。

**OpenAL软件**

便携式开源 (LGPL) 软件实施。包括 Windows、Linux、Solaris 和 BSD 操作系统上最常见的声音 API 的后端。请参阅<http://kcat.strangesoft.net/openal.html>。

**苹果**

OpenAL 是 Core Audio 的一部分，Core Audio 是官方的 Mac OS X 音频接口。请参阅<http://developer.apple.com/technologies/mac/audio-and-video.html>

该设备允许从通过 OpenAL 处理的音频输入设备进行捕获。

您需要在提供的文件名中指定要捕获的设备的名称。如果提供空字符串，设备将自动选择默认设备。您可以使用选项list\_devices获取支持的设备列表。

* [选项](#Options-78)
* [例子](#Examples-28)

#### 26.15.1 选项

频道

设置捕获音频中的通道数。当前仅支持值1 （单声道）和2 （立体声）。默认为2 。

样本大小

设置捕获音频的样本大小（以位为单位）。当前仅支持值8和16 。默认为16 。

采样率

设置捕获音频的采样率（以 Hz 为单位）。默认为44.1k 。

列表设备

如果设置为true ，打印设备列表并退出。默认为假。

#### 26.15.2 例子

打印 OpenAL 支持的设备列表并退出：

$ ffmpeg -list\_devices true -f openal -i dummy out.ogg

从 OpenAL 设备DR-BT101捕获：

$ ffmpeg -f openal -i 'DR-BT101 via PulseAudio' out.ogg

从默认设备捕获（注意空字符串”作为文件名）：

$ ffmpeg -f openal -i '' out.ogg

同时从两个设备捕获，写入两个不同的文件，在同一个ffmpeg命令中：

$ ffmpeg -f openal -i 'DR-BT101 via PulseAudio' out1.ogg -f openal -i 'ALSA Default' out2.ogg

注意：并非所有 OpenAL 实现都支持同时捕获多个 - 如果上述方法不起作用，请尝试最新的 OpenAL Soft。

### 26.16 奥斯

打开声音系统输入设备。

提供给输入设备的文件名是代表OSS输入设备的设备节点，通常设置为/dev/dsp 。

例如，要使用ffmpeg从/dev/dsp抓取，请使用以下命令：

ffmpeg -f oss -i /dev/dsp /tmp/oss.wav

有关 OSS 的更多信息，请参阅： <http://manuals.opensound.com/usersguide/dsp.html>

* [选项](#Options-79)

#### 26.16.1 选项

采样率

以 Hz 为单位设置采样率。默认值为 48000。

频道

设置通道数。默认值为 2。

### 26.17脉搏

PulseAudio 输入设备。

--enable-libpulse配置 FFmpeg 。

提供给输入设备的文件名是源设备或字符串“default”

要列出 PulseAudio 源设备及其属性，您可以调用命令pactl list sources 。

有关 PulseAudio 的更多信息，请访问<http://www.pulseaudio.org>。

* [选项](#Options-80)
* [例子](#Examples-29)

#### 26.17.1 选项

服务器

连接到由 IP 地址指定的特定 PulseAudio 服务器。未提供时使用默认服务器。

姓名

指定 PulseAudio 在显示活动客户端时将使用的应用程序名称，默认情况下它是LIBAVFORMAT\_IDENT字符串。

stream\_name

指定 PulseAudio 在显示活动流时将使用的流名称，默认情况下为“record”。

采样率

以 Hz 为单位指定采样率，默认使用 48kHz。

频道

指定使用的通道，默认设置为 2（立体声）。

帧大小

此选项不执行任何操作且已弃用。

片段大小

指定 PulseAudio 中最小缓冲片段的字节大小，它会影响音频延迟。默认情况下，它设置为 50 毫秒的数据量。

挂钟

使用当前时间设置初始 PTS。默认值为 1。

#### 26.17.2 例子

从默认设备录制流：

ffmpeg -f pulse -i 默认 /tmp/pulse.wav

### 26.18 sndio

sndio 输入设备。

要在配置期间启用此输入设备，您需要在系统上安装 libsndio。

提供给输入设备的文件名是代表 sndio 输入设备的设备节点，通常设置为/dev/audio0 。

例如，要使用ffmpeg从/dev/audio0抓取，请使用以下命令：

ffmpeg -f sndio -i /dev/audio0 /tmp/oss.wav

* [选项](#Options-81)

#### 26.18.1 选项

采样率

以 Hz 为单位设置采样率。默认值为 48000。

频道

设置通道数。默认值为 2。

### 26.19 video4linux2，v4l2

Video4Linux2 输入视频设备。

“v4l2”可以用作“video4linux2”的别名。

如果 FFmpeg 是使用 v4l-utils 支持构建的（通过使用--enable-libv4l2配置选项），则可以将它与-use\_libv4l2输入设备选项一起使用。

/dev/video N类型的名称，其中N是与设备关联的数字。

Video4Linux2 设备通常支持一组有限的宽度x高度大小和帧速率。您可以使用-list\_formats all来检查 Video4Linux2 设备支持哪些格式。某些设备（如电视卡）支持一种或多种标准。可以使用-list\_standards all列出所有支持的标准。

时间戳的时基是 1 微秒。根据内核版本和配置，时间戳可能来自实时时钟（起源于 Unix 纪元）或单调时钟（起源通常在启动时，不受 NTP 或手动更改时钟的影响）。 -timestamps abs或-ts abs选项可用于强制转换为实时时钟。

video4linux2 设备与ffmpeg和ffplay的一些使用示例：

* 列出 video4linux2 设备支持的格式：
* ffplay -f video4linux2 -list\_formats all /dev/video0
* 抓取并显示 video4linux2 设备的输入：
* ffplay -f video4linux2 -framerate 30 -video\_size hd720 /dev/video0
* 抓取并记录 video4linux2 设备的输入，保留之前设置的帧速率和大小：
* ffmpeg -f video4linux2 -input\_format mjpeg -i /dev/video0 out.mpeg

有关 Video4Linux 的更多信息，请访问<http://linuxtv.org/>。

* [选项](#Options-82)

#### 26.19.1 选项

标准

设定标准。必须是受支持标准的名称。要获取受支持标准的列表，请使用list\_standards选项。

渠道

设置输入通道号。默认为 -1，表示使用之前选择的通道。

视频大小

设置视频帧大小。参数必须是WIDTH x HEIGHT形式的字符串或有效的尺寸缩写。

像素格式

选择像素格式（仅对原始视频输入有效）。

输入格式

设置首选像素格式（用于原始视频）或编解码器名称。当有多种输入格式可用时，此选项允许您选择输入格式。

帧率

设置首选视频帧速率。

列表格式

列出可用格式（支持的像素格式、编解码器和帧大小）并退出。

可用值是：

'全部'

显示所有可用的（压缩和非压缩）格式。

'生'

仅显示原始视频（未压缩）格式。

'压缩'

仅显示压缩格式。

列表标准

列出支持的标准并退出。

可用值是：

'全部'

显示所有支持的标准。

时间戳，ts

设置抓取帧的时间戳类型。

可用值是：

'默认'

使用内核中的时间戳。

'腹肌'

使用绝对时间戳（挂钟）。

' mono2abs '

强制从单调时间戳转换为绝对时间戳。

默认值为default 。

使用\_libv4l2

使用 libv4l2 (v4l-utils) 转换函数。默认为 0。

### 26.20 vfwcap

VfW（Windows 视频）捕获输入设备。

作为输入传递的文件名是捕获驱动程序编号，范围从 0 到 9。您可以使用“列表”作为文件名来打印驱动程序列表。任何其他文件名都将被解释为设备号 0。

* [选项](#Options-83)

#### 26.20.1 选项

视频大小

设置视频帧大小。

帧率

设置抓取帧速率。默认值为ntsc ，对应于帧速率30000/1001 。

### 26.21 x11grab

X11 视频输入设备。

要在配置期间启用此输入设备，您需要在系统上安装 libxcb。它将在配置期间自动检测到。

该设备允许捕获 X11 显示的区域。

作为输入传递的文件名具有以下语法：

[主机名]：显示号码。 screen\_number [+ x\_offset , y\_offset ]

主机名：显示号码。 screen\_number指定要从中抓取的屏幕的 X11 显示名称。主机名可以省略，默认为“localhost”。环境变量DISPLAY包含默认显示名称。

x\_offset和y\_offset指定抓取区域相对于 X11 屏幕左上边框的偏移量。它们默认为 0。

查看 X11 文档（例如man X ）以获取更多详细信息。

使用xdpyinfo程序获取有关 X11 显示属性的基本信息（例如，grep 用于“名称”或“尺寸”）。

例如使用ffmpeg从:0.0抓取：

ffmpeg -f x11grab -framerate 25 -video\_size cif -i :0.0 输出.mpg

在位置10,20抓取：

ffmpeg -f x11grab -framerate 25 -video\_size cif -i :0.0+10,20 输出.mpg

* [选项](#Options-84)

#### 26.21.1 选项

选择地区

指定是否使用指针以图形方式选择抓取区域。值1提示用户通过单击和拖动以图形方式选择抓取区域。单击而不拖动将选择整个屏幕。宽度或高度为零的区域也将选择整个屏幕。此选项会覆盖video\_size 、 grab\_x和grab\_y选项。默认值为0 。

画鼠标

指定是否绘制鼠标指针。值0指定不绘制指针。默认值为1 。

跟随鼠标

使抓取区域跟随鼠标。该参数可以居中或像素数PIXELS 。

当指定为“centered”时，抓取区域跟随鼠标指针并保持鼠标指针在区域中心；否则，仅当鼠标指针到达区域边缘PIXELS （大于零）以内时，区域才会跟随。

例如：

ffmpeg -f x11grab -follow\_mouse 居中 -framerate 25 -video\_size cif -i :0.0 out.mpg

仅当鼠标指针到达边缘 100 像素以内时才跟随：

ffmpeg -f x11grab -follow\_mouse 100 -framerate 25 -video\_size cif -i :0.0 out.mpg

帧率

设置抓取帧速率。默认值为ntsc ，对应于帧速率30000/1001 。

显示区域

在屏幕上显示抓取的区域。

如果show\_region指定为1 ，则抓取区域将显示在屏幕上。使用此选项，如果仅抓取了屏幕的一部分，则很容易知道抓取的是什么。

区域边界

使用-show\_region 1 ，则设置区域边界厚度。范围是 1 到 128，默认值为 3（仅限基于 XCB 的 x11grab）。

例如：

ffmpeg -f x11grab -show\_region 1 -framerate 25 -video\_size cif -i :0.0+10,20 输出.mpg

随着follow\_mouse ：

ffmpeg -f x11grab -follow\_mouse centered -show\_region 1 -framerate 25 -video\_size cif -i :0.0 out.mpg

窗口编号

抓住这个窗口，而不是整个屏幕。默认值为 0，映射到整个屏幕（根窗口）。

xwininfo程序找到窗口的 ID ，可能使用选项 -tree 和 -root。

如果稍后放大窗口，则不会记录新区域。当窗口关闭、取消映射（即图标化）或缩小到超过视频大小（默认为初始窗口大小）时，视频结束。

此选项禁用选项follow\_mouse和select\_region 。

视频大小

设置视频帧大小。默认是整个桌面或窗口。

抓取\_x

抓取

设置抓取区域坐标。它们表示为距 X11 窗口左上角的偏移量，对应于设备名称中的x\_offset和y\_offset参数。两个选项的默认值为 0。

## 27 输出设备

输出设备是 FFmpeg 中的配置元素，可以将多媒体数据写入连接到系统的输出设备。

当您配置 FFmpeg 构建时，默认情况下会启用所有支持的输出设备。您可以使用配置选项“–list-outdevs”列出所有可用的。

您可以使用配置选项“–disable-outdevs”禁用所有输出设备，并使用选项“–enable-outdev= OUTDEV ”有选择地启用输出设备，或者您可以使用选项“–disable”禁用特定输入设备-outdev = OUTDEV “。

ff\* 工具的选项“-devices”将显示启用的输出设备列表。

当前可用的输出设备的描述如下。

* [阿尔萨](#alsa-1)
* [音频工具箱](#AudioToolbox)
* [卡卡](#caca)
* [甲板链接](#decklink-1)
* [开发者](#fbdev-1)
* [opengl](#opengl)
* [oss](#oss-1)
* [脉冲](#pulse-1)
* [sdl](#sdl)
* [sndio](#sndio-1)
* [v4l2](#v4l2)
* [十五](#xv)

### 27.1 阿尔萨

ALSA（高级 Linux 声音架构）输出设备。

* [例子](#Examples-30)

#### 27.1.1 例子

* 在默认的 ALSA 设备上播放文件：
* ffmpeg -i INPUT -f alsa 默认值
* 在声卡 1、音频设备 7 上播放文件：
* ffmpeg -i 输入 -f alsa hw:1,7

### 27.2 音频工具箱

AudioToolbox 输出设备。

允许本机输出到 OSX 上的 CoreAudio 设备。

输出文件名可以为空（或- ）以引用默认系统输出设备或引用设备索引的数字，如使用所示： -list\_devices true 。

-audio\_device\_index <INDEX>按索引选择音频输入设备，覆盖输入文件名中给出的任何设备名称或索引。

可以使用-list\_devices true枚举所有可用设备，列出所有设备名称、UID 和相应的索引。

* [选项](#Options-85)
* [例子](#Examples-31)

#### 27.2.1 选项

AudioToolbox 支持以下选项：

-audio\_device\_index <索引>

通过索引指定音频设备。覆盖输出文件名中给定的任何内容。

#### 27.2.2 例子

* 打印支持的设备列表并向默认设备输出正弦波：
* $ ffmpeg -f lavfi -i sine=r=44100 -f audiotoolbox -list\_devices true -
* 将正弦波输出到索引为 2 的设备，覆盖任何输出文件名：
* $ ffmpeg -f lavfi -i sine=r=44100 -f audiotoolbox -audio\_device\_index 2 -

### 27.3卡卡

CACA 输出设备。

此输出设备允许在 CACA 窗口中显示视频流。每个应用程序只允许一个 CACA 窗口，因此您在一个应用程序中只能有一个此输出设备的实例。

--enable-libcaca配置 FFmpeg 。 libcaca 是一个图形库，它输出文本而不是像素。

有关 libcaca 的更多信息，请查看： http: [//caca.zoy.org/wiki/libcaca](http://caca.zoy.org/wiki/libcaca)

* [选项](#Options-86)
* [例子](#Examples-32)

#### 27.3.1 选项

窗口标题

设置 CACA 窗口标题，如果未指定默认为为输出设备指定的文件名。

窗口大小

设置CACA窗口大小，可以是窗体宽×高的字符串，也可以是视频大小的缩写。如果未指定，则默认为输入视频的大小。

司机

设置显示驱动程序。

算法

设置抖动算法。抖动是必要的，因为渲染的图片通常比可用的调色板具有更多的颜色。接受的值与-list\_dither algorithms一起列出。

抗锯齿

设置抗锯齿方法。抗锯齿使渲染图像平滑并避免常见的阶梯效应。接受的值与-list\_dither antialiases一起列出。

字符集

设置在呈现文本时将使用哪些字符。接受的值与-list\_dither charsets一起列出。

颜色

设置渲染文本时要使用的颜色。接受的值与-list\_dither colors一起列出。

列表驱动程序

如果设置为true ，则打印可用驱动程序列表并退出。

列表抖动

列出与参数相关的可用抖动选项。参数必须是algorithms 、 antialiases 、 charsets 、 colors之一。

#### 27.3.2 例子

* 以下命令显示ffmpeg输出是一个 CACA 窗口，强制其大小为 80x25：
* ffmpeg -i INPUT -c:v rawvideo -pix\_fmt rgb24 -window\_size 80x25 -f caca -
* 显示可用驱动程序列表并退出：
* ffmpeg -i INPUT -pix\_fmt rgb24 -f caca -list\_drivers true -
* 显示可用抖动颜色列表并退出：
* ffmpeg -i INPUT -pix\_fmt rgb24 -f caca -list\_dither 颜色 -

### 27.4甲板链接

decklink 输出设备为 Blackmagic DeckLink 设备提供播放功能。

要启用此输出设备，您需要 Blackmagic DeckLink SDK 并且需要使用适当的--extra-cflags和--extra-ldflags进行配置。在 Windows 上，您需要通过widl运行 IDL 文件。

DeckLink 对其支持的格式非常挑剔。像素格式始终为 uyvy422，必须使用-list\_formats 1为您的设备确定帧速率、场序和视频大小。音频采样率始终为 48 kHz。

* [选项](#Options-87)
* [例子](#Examples-33)

#### 27.4.1 选项

列表设备

如果设置为true ，打印设备列表并退出。默认为假。此选项已弃用，请使用ffmpeg 的-sinks选项列出可用的输出设备。

列表格式

如果设置为true ，则打印支持的格式列表并退出。默认为假。

预卷

预卷视频的时间量（以秒为单位）。默认为0.5 。

双工模式

设置 decklink 设备双工/配置文件模式。必须是“ unset ”、“ half ”、“ full ”、“ one\_sub\_device\_full ”、“ one\_sub\_device\_half ”、“ two\_sub\_device\_full ”、“ four\_sub\_device\_half ”默认为“ unset ”。

注意：DeckLink SDK 11.0 已将双工属性替换为配置文件属性。对于 DeckLink Duo 2 和 DeckLink Quad 2，配置文件在使用相同连接器的任何 2 个子设备之间共享。对于 DeckLink 8K Pro，一个配置文件在所有 4 个子设备之间共享。所以 DeckLink 8K Pro 支持四种配置文件。

DeckLink 8K Pro（使用 DeckLink SDK >= 11.0）的有效配置文件模式：' one\_sub\_device\_full '、' one\_sub\_device\_half '、' two\_sub\_device\_full '、' four\_sub\_device\_half '

DeckLink Quad 2 和 DeckLink Duo 2 的有效配置文件模式：“ half ”、“ full ”

时序偏移量

在使用的输出上设置同步锁相时序像素偏移。默认为“未设置”。

关联

在使用的输出上设置 SDI 视频链接配置。必须是“ unset ”、“ single ”链接 SDI、“ dual ”链接 SDI 或“ quad ”链接 SDI。默认为“未设置”。

平方

为 Quad-link SDI 输出启用 Square Division Quad Split 模式。必须是“ unset ”、“ true ”或“ false ”。默认为取消设置。

level\_a

在使用的输出上启用 SMPTE Level A 模式。必须是“ unset ”、“ true ”或“ false ”。默认为取消设置。

#### 27.4.2 例子

* 列出输出设备：
* ffmpeg -sinks 甲板链接
* 列出支持的格式：
* ffmpeg -i test.avi -f decklink -list\_formats 1 'DeckLink Mini Monitor'
* 播放视频片段：
* ffmpeg -i test.avi -f decklink -pix\_fmt uyvy422 'DeckLink Mini Monitor'
* 播放非标准帧率或视频大小的视频剪辑：
* ffmpeg -i test.avi -f decklink -pix\_fmt uyvy422 -s 720x486 -r 24000/1001 'DeckLink Mini Monitor'

### 27.5 fbdev

Linux 帧缓冲输出设备。

Linux 帧缓冲区是一个独立于图形硬件的抽象层，用于在计算机显示器（通常是控制台）上显示图形。它通过文件设备节点访问，通常是/dev/fb0 。

Linux 源代码树中包含的文件Documentation/fb/framebuffer.txt 。

* [选项](#Options-88)
* [例子](#Examples-34)

#### 27.5.1 选项

x偏移量

偏移量

设置左上角的 x/y 坐标。默认为 0。

#### 27.5.2 例子

/dev/fb0上播放文件。所需的像素格式取决于当前的帧缓冲区设置。

ffmpeg -re -i INPUT -c:v rawvideo -pix\_fmt bgra -f fbdev /dev/fb0

另请参见<http://linux-fbdev.sourceforge.net/>和 fbset(1)。

### 27.6 开放式

OpenGL 输出设备。

--enable-opengl配置 FFmpeg 。

此输出设备允许渲染到 OpenGL 上下文。上下文可能由应用程序提供或创建默认 SDL 窗口。

当设备呈现到外部上下文时，应用程序必须为以下消息实现处理程序： AV\_DEV\_TO\_APP\_CREATE\_WINDOW\_BUFFER - 在当前线程上创建 OpenGL 上下文。 AV\_DEV\_TO\_APP\_PREPARE\_WINDOW\_BUFFER - 使 OpenGL 上下文成为当前上下文。 AV\_DEV\_TO\_APP\_DISPLAY\_WINDOW\_BUFFER - 交换缓冲区。 AV\_DEV\_TO\_APP\_DESTROY\_WINDOW\_BUFFER - 销毁 OpenGL 上下文。应用程序还需要通过发送AV\_APP\_TO\_DEV\_WINDOW\_SIZE消息通知设备当前分辨率。

* [选项](#Options-89)
* [例子](#Examples-35)

#### 27.6.1 选项

背景

设置背景颜色。黑色是默认值。

无窗口

设置为非零值时禁用默认 SDL 窗口。设置时，应用程序必须提供 OpenGL 上下文以及window\_size\_cb和window\_swap\_buffers\_cb回调。

窗口标题

设置 SDL 窗口标题，如果未指定默认为为输出设备指定的文件名。设置no\_window时忽略。

窗口大小

设置首选窗口大小，可以是 widthxheight 形式的字符串或视频大小缩写。如果未指定，则默认为输入视频的大小，根据宽高比缩小。在未设置no\_window时主要可用。

#### 27.6.2 例子

使用 OpenGL 渲染在 SDL 窗口上播放文件：

ffmpeg -i INPUT -f opengl "窗口标题"

### 27.7 奥斯

OSS（Open Sound System）输出设备。

### 27.8脉搏

PulseAudio 输出设备。

--enable-libpulse配置 FFmpeg 。

有关 PulseAudio 的更多信息，请访问<http://www.pulseaudio.org>

* [选项](#Options-90)
* [例子](#Examples-36)

#### 27.8.1 选项

服务器

连接到由 IP 地址指定的特定 PulseAudio 服务器。未提供时使用默认服务器。

姓名

指定 PulseAudio 在显示活动客户端时将使用的应用程序名称，默认情况下它是LIBAVFORMAT\_IDENT字符串。

stream\_name

指定显示活动流时 PulseAudio 将使用的流名称，默认情况下它设置为指定的输出名称。

设备

指定要使用的设备。未提供时使用默认设备。可以使用命令pactl list sinks获取输出设备列表。

缓冲区大小

缓冲持续时间

控制 PulseAudio 缓冲区的大小和持续时间。小缓冲区提供更多控制，但需要更频繁的更新。

buffer\_size 以字节为单位指定大小，而buffer\_duration以毫秒为单位指定持续时间。

当提供两个选项时，将使用最大值（使用流参数将持续时间重新计算为字节）。如果将它们设置为 0（默认值），设备将使用默认的 PulseAudio 持续时间值。默认情况下，PulseAudio 将缓冲持续时间设置为大约 2 秒。

预缓冲

以字节为单位指定预缓冲大小。在缓冲区中至少有prebuf字节可用之前，服务器不会开始播放。默认情况下，此选项初始化为与buffer\_size或buffer\_duration相同的值（以较大者为准）。

最低要求

以字节为单位指定最小请求大小。服务器不会从客户端请求少于minreq字节，而是等待缓冲区足够空闲以一次请求更多字节。建议不要设置此选项，这会将其初始化为服务器认为合理的值。

#### 27.8.2 例子

在默认服务器上的默认设备上播放文件：

ffmpeg -i INPUT -f pulse “流名称”

### 27.9 升

SDL（简单直接媒体层）输出设备。

“sdl2”可以用作“sdl”的别名。

此输出设备允许在 SDL 窗口中显示视频流。每个应用程序只允许一个 SDL 窗口，因此您在一个应用程序中只能有一个此输出设备的实例。

要启用此输出设备，您需要在配置构建时在系统上安装 libsdl。

有关 SDL 的更多信息，请查看： http: [//www.libsdl.org/](http://www.libsdl.org/)

* [选项](#Options-91)
* [交互式命令](#Interactive-commands)
* [例子](#Examples-37)

#### 27.9.1 选项

窗口标题

设置 SDL 窗口标题，如果未指定默认为为输出设备指定的文件名。

图标\_标题

设置图标化 SDL 窗口的名称，如果未指定，则设置为与window\_title相同的值。

窗口大小

设置SDL窗口大小，可以是窗体宽×高的字符串，也可以是视频大小的缩写。如果未指定，则默认为输入视频的大小，根据宽高比缩小。

窗口\_x

窗口\_y

设置窗口在屏幕上的位置。

window\_fullscreen

提供非零值时设置全屏模式。默认值为零。

window\_enable\_quit

提供非零值时启用退出操作（使用窗口按钮或键盘键）。默认值为 1（启用退出操作）

#### 27.9.2 交互式命令

设备创建的窗口可以通过以下交互命令进行控制。

q, 电调

立即退出设备。

#### 27.9.3 例子

以下命令显示ffmpeg输出是一个 SDL 窗口，强制其大小为 qcif 格式：

ffmpeg -i INPUT -c:v rawvideo -pix\_fmt yuv420p -window\_size qcif -f sdl “SDL 输出”

### 27.10 sndio

sndio 音频输出设备。

### 27.11 v4l2

Video4Linux2 输出设备。

### 27.12 xv

XV（XVideo）输出设备。

该输出设备允许在 X Window 系统窗口中显示视频流。

* [选项](#Options-92)
* [例子](#Examples-38)

#### 27.12.1 选项

显示名称

指定硬件显示名称，它确定要使用的显示和通信域。

显示名称或 DISPLAY 环境变量可以是格式为hostname [: number [.屏幕编号]]。

hostname指定显示物理连接到的主机的名称。 number指定该主机上显示服务器的编号。 screen\_number指定要在该服务器上使用的屏幕。

如果未指定，则默认为 DISPLAY 环境变量的值。

例如， dual-headed:0.1将在名为“dual-headed”的机器上指定显示器 0 的屏幕 1。

查看 X11 规范以获取有关显示名称格式的更多详细信息。

窗口编号

当设置为非零值时，设备不会创建新窗口，而是使用现有窗口和提供的window\_id 。默认情况下，此选项设置为零，并且设备会创建自己的窗口。

窗口大小

宽×高的字符串，也可以是视频大小的缩写。如果未指定，则默认为输入视频的大小。设置window\_id时忽略。

窗口\_x

窗口\_y

为创建的窗口设置 X 和 Y 窗口偏移。默认情况下，它们都设置为 0。窗口管理器可能会忽略这些值。设置window\_id时忽略。

窗口标题

设置窗口标题，如果未指定默认为为输出设备指定的文件名。设置window\_id时忽略。

有关 XVideo 的更多信息，请参阅<http://www.x.org/>。

#### 27.12.2 例子

* 同时使用ffmpeg解码、显示和编码视频输入：
* ffmpeg -i 输入输出 -f xv 显示
* 解码输入视频并将其显示到多个 X11 窗口：
* ffmpeg -i INPUT -f xv normal -vf negate -f xv 否定

## 28 重采样器选项

音频重采样器支持以下命名选项。

-option来设置选项 FFmpeg 工具中的值，选项= aresample 过滤器的值，通过在SwrContext选项中明确设置值或使用libavutil/opt.h API 进行编程使用。

uchl，used\_chlayout

设置使用的输入通道布局。默认未设置。此选项仅用于特殊重映射。

isr，in\_sample\_rate

设置输入采样率。默认值为 0。

osr，out\_sample\_rate

设置输出采样率。默认值为 0。

isf，in\_sample\_fmt

指定输入样本格式。它默认设置为none 。

osf，out\_sample\_fmt

指定输出样本格式。它默认设置为none 。

tsf, internal\_sample\_fmt

设置内部样本格式。默认值为无。如果未明确设置，将自动选择它。

ichl，in\_chlayout

ochl，out\_chlayout

设置输入/输出通道布局。

有关所需语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 通道布局部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#channel-layout-syntax)

clev，center\_mix\_level

设置中央混音电平。它是一个以分贝表示的值，并且必须在 [-32,32] 区间内。

slev, surround\_mix\_level

设置环绕声混音电平。它是一个以分贝表示的值，并且必须在 [-32,32] 区间内。

lfe\_mix\_level

将 LFE 混音设置为非 LFE 电平。当有 LFE 输入但没有 LFE 输出时使用。它是一个以分贝表示的值，并且必须在 [-32,32] 区间内。

rmvol, rematrix\_volume

设置再矩阵音量。默认值为 1.0。

rematrix\_maxval

设置再矩阵化的最大输出值。这可用于防止削波与防止音量减小。值为 1.0 可防止剪裁。

标志，swr\_flags

设置转换器使用的标志。默认值为 0。

它支持以下单独的标志：

资源

强制重采样，即使输入和输出采样率匹配，此标志也会强制使用重采样。

抖动比例

设置抖动比例。默认值为 1。

抖动方法

设置抖动方法。默认值为 0。

支持的值：

'长方形'

选择矩形抖动

'三角形'

选择三角抖动

' triangular\_hp '

选择高通三角抖动

'唇膏'

选择 Lipshitz 噪声整形抖动。

'柴田'

选择 Shibata 噪声整形抖动。

' low\_shibata '

选择低 Shibata 噪声整形抖动。

' high\_shibata '

选择高 Shibata 噪声整形抖动。

' f\_weighted '

选择 f 加权噪声整形抖动

' modified\_e\_weighted '

选择修改后的电子加权噪声整形抖动

' improved\_e\_weighted '

选择改进的电子加权噪声整形抖动

重采样器

设置重采样引擎。默认值为 swr。

支持的值：

'开关'

选择原生 SW 重采样器；过滤器选项 precision 和 cheby 在这种情况下不适用。

' soxr '

选择 SoX 重采样器（如果可用）；补偿和过滤器选项 filter\_size、phase\_shift、exact\_rational、filter\_type 和 kaiser\_beta 在这种情况下不适用。

过滤尺寸

仅适用于 swr，设置重采样过滤器大小，默认值为 32。

相移

仅针对swr，设置重采样相移，默认值为10，且必须在区间[0,30]内。

线性插值

启用时使用线性插值（默认）。如果您想在 exact\_rational 失败时保持速度而不是质量，请禁用它。

exact\_rational

仅对于 swr，启用时，尝试使用基于输入和输出采样率的精确 phase\_count。但是，如果它大于1 << phase\_shift ，phase\_count 将是1 << phase\_shift作为回退。默认启用。

隔断

设置截止频率（swr：6dB点；soxr：0dB点）比例；必须是介于 0 和 1 之间的浮点值。对于 swr，默认值为 0.97，对于 soxr，默认值为 0.91（采样率为 44100，将整个音频频带保持为 20kHz）。

精确

仅对于 soxr，将计算重采样信号的精度（以位为单位）。默认值 20（通过适当的抖动，适用于 16 的目标位深度）给出了 SoX 的“高质量”；值为 28 表示 SoX 的“非常高质量”。

切比

仅对于 soxr，选择无通带滚降（Chebyshev）和“非理性”比率的更高精度近似。默认值为 0。

异步

仅适用于 swr，简单的 1 参数音频使用拉伸、挤压、填充和修剪与时间戳同步。将此设置为 1 将启用填充和修剪，较大的值表示每秒可以拉伸或压缩数据的最大样本量。默认值为 0，因此不应用任何补偿来使样本与音频时间戳匹配。

first\_pts

仅对于 swr，假设第一个 pts 应该是这个值。时间单位为 1 / 采样率。这允许在流的开头进行填充/修剪。默认情况下，不会对第一帧的预期点做出任何假设，因此不会进行填充或修剪。例如，如果音频流在视频流之后开始，这可以设置为 0 以用静音填充开头，或者由于编码器延迟而修剪任何具有负点的样本。

最小补偿

仅对于 swr，设置时间戳和音频数据之间的最小差异（以秒为单位）以触发数据的拉伸/挤压/填充或修剪以使其与时间戳匹配。默认情况下禁用拉伸/挤压/填充和修剪 ( min\_comp = FLT\_MAX )。

min\_hard\_comp

仅对于 swr，设置时间戳和音频数据之间的最小差异（以秒为单位）以触发添加/删除样本以使其与时间戳匹配。此选项实际上是在硬（修剪/填充）和软（挤压/拉伸）补偿之间进行选择的阈值。请注意，默认情况下所有补偿都通过min\_comp禁用。默认值为 0.1。

comp\_duration

仅对于 swr，设置数据被拉伸/压缩以使其与时间戳匹配的持续时间（以秒为单位）。必须是非负双精度浮点值，默认值为 1.0。

最大软补偿

仅对于 swr，设置数据被拉伸/压缩以使其与时间戳匹配的最大因子。必须是非负双精度浮点值，默认值为 0。

矩阵编码

选择矩阵立体声编码。

它接受以下值：

'无'

选择无

'杜比'

选择杜比

' dplii '

选择杜比定向逻辑 II

默认值为无。

过滤类型

仅对于 swr，选择重采样过滤器类型。这只会影响重采样操作。

它接受以下值：

'立方体'

选择立方体

' blackman\_nuttall '

选择 Blackman Nuttall 加窗 sinc

'凯撒'

选择 Kaiser 加窗 sinc

凯撒测试版

仅对于 swr，设置 Kaiser 窗口 beta 值。必须是[2,16]区间内的double float值，默认值为9。

output\_sample\_bits

仅对于 swr，设置用于抖动的输出采样位数。必须是[0,64]区间内的整数，默认为0，表示不使用。

## 29 缩放器选项

视频缩放器支持以下命名选项。

-option来设置选项 FFmpeg 工具中的价值，下面提到了一些仅限 API 的例外情况。对于编程用途，可以在SwsContext选项中或通过libavutil/opt.h API显式设置它们。

sws\_flags

设置定标器标志。这也用于设置缩放算法。只能选择一个算法。默认值为“双三次”。

它接受以下值：

'快速双线性'

选择快速双线性缩放算法。

'双线性'

选择双线性缩放算法。

'双三次'

选择双三次缩放算法。

'实验'

选择实验缩放算法。

'邻居'

选择最近邻重新缩放算法。

'区域'

选择平均面积重新缩放算法。

'双胞胎'

为亮度分量选择双三次缩放算法，为色度分量选择双线性。

'高斯'

选择高斯重新缩放算法。

'正弦'

选择 sinc 重新缩放算法。

'兰佐斯'

选择 Lanczos 重新缩放算法。默认宽度 (alpha) 为 3，可以通过设置param0进行更改。

'样条'

选择自然双三次样条重新缩放算法。

'打印信息'

启用打印/调试日志记录。

'准确\_rnd '

启用精确舍入。

' full\_chroma\_int '

启用全色度插值。

' full\_chroma\_inp '

选择全色度输入。

'双精确'

启用 bitexact 输出。

srcw （仅限 API）

设置源宽度。

srch （仅限 API）

设置源高度。

dstw （仅限 API）

设置目标宽度。

dsth （仅限 API）

设置目标高度。

src\_format （仅限 API）

设置源像素格式（必须表示为整数）。

dst\_format （仅限 API）

设置目标像素格式（必须表示为整数）。

src\_range （布尔值）

如果值设置为1 ，表示源是全范围的。默认值为0 ，表示源是有限范围的。

dst\_range （布尔值）

如果值设置为1 ，则为目的地启用全范围。默认值为0 ，这会启用有限范围。

参数 0，参数 1

设置缩放算法参数。指定的值特定于某些缩放算法，并被其他算法忽略。指定的值是浮点数值。

sws\_dither

设置抖动算法。接受以下值之一。默认值为“自动”。

'汽车'

自动选择

'无'

没有抖动

'拜耳'

拜耳抖动

'编辑'

误差扩散抖动

'抖动'

算术抖动，基于使用加法

' x\_dither '

算术抖动，基于使用 xor（比 a\_dither 更随机/更不明显的模式）。

字母表

设置当输入有 alpha 但输出没有时使用的 alpha 混合。默认值为“无”。

'统一颜色'

混合成统一的背景色

'棋盘'

混合到棋盘上

'无'

没有混合

## 30 过滤介绍

FFmpeg 中的过滤是通过 libavfilter 库启用的。

在 libavfilter 中，一个过滤器可以有多个输入和多个输出。为了说明可能发生的事情的种类，我们考虑以下过滤图。

[主要的]

输入 --> 分割 ----------------------> 叠加 --> 输出

| ^

|[tmp] [翻转]|

+-----> 裁剪 --> vflip ------+

此过滤器图将输入流分成两个流，然后将一个流发送通过裁剪过滤器和 vflip 过滤器，然后通过将其叠加在顶部将其与另一个流合并。您可以使用以下命令来实现此目的：

ffmpeg -i INPUT -vf "split [main][tmp]; [tmp] crop=iw:ih/2:0:0, vflip [flip]; [main][flip] overlay=0:H/2" 输出

结果将是视频的上半部分镜像到输出视频的下半部分。

同一线性链中的过滤器以逗号分隔，不同线性过滤器链以分号分隔。在我们的示例中， crop、vflip在一个线性链中， split和overlay分别在另一个线性链中。线性链连接的点用方括号中的名称标记。在该示例中，拆分过滤器生成两个与标签[main]和[tmp]关联的输出。

split的第二个输出（标记为[tmp] ）的流通过裁剪过滤器进行处理，裁剪掉视频的下半部分，然后垂直翻转。 overlay filter 将 split filter（标记为[main] ）的第一个未更改输出作为输入，并在其下半部分覆盖crop,vflip filterchain生成的输出。

一些过滤器接受参数列表的输入：它们在过滤器名称和等号之后指定，并且彼此之间用冒号分隔。

没有音频/视频输入的所谓源过滤器和没有音频/视频输出的接收器过滤器。

## 31 图2点

FFmpeg工具目录中包含的graph2dot程序可用于解析 filtergraph 描述并以点语言发出相应的文本表示。

调用命令：

图2点-h

查看如何使用graph2dot 。

然后，您可以将点描述传递给点程序（来自 graphviz 程序套件）并获得过滤图的图形表示。

例如命令序列：

回声GRAPH\_DESCRIPTION | \

工具/graph2dot -o graph.tmp && \

点-Tpng graph.tmp -o graph.png && \

显示图.png

GRAPH\_DESCRIPTION字符串描述的图形的图像。请注意，此字符串必须是一个完整的自包含图，其输入和输出已明确定义。例如，如果您的命令行是以下形式：

ffmpeg -i infile -vf scale=640:360 输出文件

您的GRAPH\_DESCRIPTION字符串需要采用以下形式：

nullsrc,scale=640:360,nullsink

您可能还需要设置nullsrc参数并添加格式过滤器以模拟特定的输入文件。

## 32 过滤图说明

过滤器图是连接过滤器的有向图。它可以包含循环，一对过滤器之间可以有多个链接。每个链接的一侧有一个输入垫，将其连接到一个滤波器，从中获取输入，另一侧有一个输出垫，将其连接到一个接受其输出的滤波器。

filtergraph 中的每个过滤器都是在应用程序中注册的过滤器类的实例，它定义了过滤器的特征和输入输出焊盘的数量。

没有输入焊盘的滤波器称为“源”，没有输出焊盘的滤波器称为“汇”。

* [过滤图语法](#Filtergraph-syntax-1)
* [filtergraph转义注意事项](#Notes-on-filtergraph-escaping)

### 32.1 过滤图语法

filtergraph 具有文本表示，可由ffmpeg中的-filter / -vf / -af和-filter\_complex选项和ffplay中的-vf / -af选项以及libavfilter/avfilter.h中定义的avfilter\_graph\_parse\_ptr()函数识别。

过滤器链由一系列连接的过滤器组成，每个过滤器都连接到序列中的前一个过滤器。过滤器链由“,”分隔的过滤器描述列表表示。

过滤器图由一系列过滤器链组成。过滤器链序列由“;”分隔的过滤器链描述列表表示。

过滤器由以下形式的字符串表示： [ in\_link\_1 ]...[ in\_link\_N ] filter\_name @ id = arguments [ out\_link\_1 ] ... [ out\_link\_M ]

filter\_name是过滤器类的名称，所描述的过滤器是其实例，并且必须是在程序中注册的过滤器类之一的名称，可选地后跟“@id ” 。过滤器类的名称后面可以选择跟一个字符串“= arguments ”。

arguments是一个字符串，其中包含用于初始化过滤器实例的参数。它可能具有以下两种形式之一：

* 以“:”分隔的键=值对列表。
* 以“:”分隔的值列表。在这种情况下，假定键是按声明顺序排列的选项名称。例如，淡入淡出过滤器按此顺序声明三个选项 - type 、 start\_frame和nb\_frames 。然后参数列表in:0:30表示in的值赋值给option type ， 0赋给start\_frame ， 30赋给nb\_frames 。
* 以“:”分隔的混合直接值和长键=值对列表。直接值必须在键=值对之前，并遵循与前一点相同的约束顺序。可以按任何首选顺序设置以下键=值对。

如果选项值本身是一个项目列表（例如，格式过滤器采用像素格式列表），列表中的项目通常由' |分隔'.

参数列表可以使用字符 ' ' ' 作为起始和结束标记，字符 ' \ ' 用于转义引用文本中的字符；否则，当遇到下一个特殊字符（属于集合 ' []=;, '）时，认为参数字符串终止。

ffmpeg CLI 工具中实现的特殊语法允许从文件加载选项值。这是通过在选项名称前添加一个斜杠“/”来完成的，然后将提供的值解释为加载实际值的路径。例如

ffmpeg -i <输入> -vf drawtext=/text=/tmp/some\_text <输出>

/tmp/some\_text加载要绘制的文本。希望实现类似功能的 API 用户应将avfilter\_graph\_segment\_\*()函数与自定义 IO 代码一起使用。

过滤器的名称和参数可选地前后跟有链接标签列表。链接标签允许人们命名链接并将其关联到过滤器输出或输入板。前面的标签in\_link\_1 ... in\_link\_N与滤波器输入垫相关联，以下标签out\_link\_1 ... out\_link\_M与输出垫相关联。

当在过滤器图中找到两个具有相同名称的链接标签时，将创建相应输入和输出板之间的链接。

如果未标记输出垫，则默认情况下将其链接到过滤器链中下一个过滤器的第一个未标记输入垫。例如在过滤器链中

nullsrc, split[L1], [L2]overlay, nullsink

拆分过滤器实例有两个输出垫，叠加过滤器实例有两个输入垫。 split 的第一个输出焊盘标记为“L1”，overlay 的第一个输入焊盘标记为“L2”，split 的第二个输出焊盘链接到 overlay 的第二个输入焊盘，两者均未标记。

在过滤器描述中，如果没有指定第一个过滤器的输入标签，则假定为“in”；如果未指定最后一个过滤器的输出标签，则假定为“out”。

在完整的滤波器链中，所有未标记的滤波器输入和输出焊盘都必须连接。如果所有过滤器链的所有过滤器输入和输出焊盘都已连接，则过滤器图被认为是有效的。

Libavfilter 会自动在需要格式转换的地方插入[缩放过滤器。](#scale)可以通过在前面加上sws\_flags= flags来为那些自动插入的缩放器指定 swscale 标志；到过滤图描述。

下面是过滤图语法的 BNF 描述：

NAME ::= 字母数字字符序列和“\_”

FILTER\_NAME ::=名称["@"名称]

LINKLABEL ::=“[”名称“]”

链接标签::=链接标签[链接标签]

FILTER\_ARGUMENTS ::= 字符序列（可能被引用）

FILTER ::= [ LINKLABELS ] FILTER\_NAME ["=" FILTER\_ARGUMENTS ] [ LINKLABELS ]

过滤器链::=过滤器[,过滤器链]

FILTERGRAPH ::= [sws\_flags= flags ;]过滤器链[;过滤图]

### 32.2 关于过滤图转义的注意事项

Filtergraph 描述组合需要几个级别的转义。有关所采用的转义过程的更多信息，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils)“引用和转义”部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#quoting_005fand_005fescaping)

第一级转义会影响每个过滤器选项值的内容，其中可能包含特殊字符:用于分隔值，或其中一个转义字符\' 。

二级转义会影响整个过滤器描述，其中可能包含转义字符\'或特殊字符[]；由过滤器图描述使用。

最后，当您在 shell 命令行上指定 filtergraph 时，您需要对其包含的 shell 特殊字符执行第三级转义。

例如，考虑将以下字符串嵌入到[drawtext](#drawtext)过滤器描述文本值中：

这是一个“字符串”：可能包含一个或多个特殊字符

这个字符串包含'特殊转义符，和:特殊字符，所以需要这样转义：

text=this is a \'string\'\: 可能包含一个或多个特殊字符

在过滤图描述中嵌入过滤器描述时，需要进行第二级转义，以转义所有过滤图特殊字符。因此上面的例子变成：

drawtext=text=this is a \\\'string\\\'\\: 可能包含一个\或多个\特殊字符

（请注意，除了\'转义特殊字符外， , 也需要转义）。

最后，在 shell 命令中编写 filtergraph 描述时需要额外的转义级别，这取决于所采用的 shell 的转义规则。例如，假设\是特殊的并且需要用另一个\进行转义，那么前面的字符串最终将导致：

-vf "drawtext=text=this is a \\\\\\'string\\\\\\'\\\\: may contain one\\, or more\\, special characters"

## 33 时间线编辑

一些过滤器支持通用启用选项。对于支持时间线编辑的过滤器，此选项可以设置为在将帧发送到过滤器之前评估的表达式。如果评估不为零，将启用过滤器，否则帧将原封不动地发送到过滤器图中的下一个过滤器。

该表达式接受以下值：

' t '

以秒表示的时间戳，如果输入时间戳未知则为 NAN

' n '

输入帧的序号，从0开始

'位置'

输入帧在文件中的位置，如果未知则为 NAN

' w '

' h '

输入帧的宽度和高度（如果是视频）

此外，这些过滤器支持可用于重新定义表达式的启用命令。

与任何其他过滤选项一样，启用选项遵循相同的规则。

例如，要启用从 10 秒到 3 分钟的模糊滤镜 ( [smartblur](#smartblur) )，以及从 3 秒开始的[曲线滤镜：](#curves)

smartblur = enable='between(t,10,3\*60)',

curves = enable='gte(t,3)' : 预设=cross\_process

请参阅ffmpeg -filters以查看哪些过滤器具有时间线支持。

## 34 在运行时使用命令更改选项

可以使用命令在过滤器运行期间更改某些选项。这些选项在ffmpeg的输出上标记为“T” -h filter=<过滤器名称> .命令的名称是选项的名称，参数是新值。

## 具有多个输入的过滤器的 35 个选项（帧同步）

一些具有多个输入的过滤器支持一组通用选项。这些选项只能按名称设置，不能使用缩写符号。

eof\_action

在次级输入上遇到 EOF 时采取的操作；它接受以下值之一：

重复

重复最后一帧（默认）。

结束

结束两个流。

经过

通过主输入。

最短的

如果设置为 1，则强制输出在最短输入终止时终止。默认值为 0。

重复上次

如果设置为 1，则强制过滤器将辅助流的最后一帧扩展到主流结束。值为 0 将禁用此行为。默认值为 1。

ts\_sync\_mode

基于辅助输入时间戳同步流的严格程度；它接受以下值之一：

默认

来自具有最接近或等于主输入帧的时间戳的辅助输入帧。

最近的

具有与主输入帧绝对最近的时间戳的辅助输入帧。

## 36个音频过滤器

配置 FFmpeg 构建时，可以使用--disable-filters禁用任何现有过滤器。配置输出将显示构建中包含的音频过滤器。

下面是对当前可用的音频过滤器的描述。

* [压缩机](#acompressor)
* [对比](#acontrast)
* [复印件](#acopy)
* [淡入淡出](#acrossfade)
* [跨越](#acrossover)
* [破碎机](#acrusher)
* [敏锐的](#acue)
* [点按](#adeclick)
* [剪辑](#adeclip)
* [关联](#adecorrelate)
* [延迟](#adelay)
* [腺病毒](#adenorm)
* [导数，非积分](#aderivative_002c-aintegral)
* [adrc](#adrc)
* [动态均衡器](#adynamicequalizer)
* [不流畅的](#adynamicsmooth)
* [回声](#aecho)
* [强调](#aemphasis)
* [评估](#aeval)
* [兴奋剂](#aexciter)
* [离开](#afade-1)
* [后缀](#afftdn)
* [关系](#afftfilt)
* [阿菲尔](#afir-1)
* [格式](#aformat-1)
* [频移](#afreqshift)
* [AFWDN](#afwtdn)
* [玛瑙](#agate)
* [空气](#aiir)
* [限制器](#alimiter)
* [全部通过](#allpass)
* [一个循环](#aloop)
* [合并](#amerge-1)
* [混合物](#amix)
* [乘以](#amultiply)
* [均衡器](#anequalizer)
* [安明](#anlmdn)
* [anlmf, anlms](#anlmf_002c-anlms)
* [在那边](#anull)
* [应用程序](#apad)
* [移相器](#aphaser)
* [相移](#aphaseshift)
* [心电图](#apsyclip)
* [脉动器](#apulsator)
* [样品](#aresample-1)
* [反转](#areverse)
* [阿恩恩](#arnndn)
* [asdr](#asdr)
* [样本](#asetnsamples)
* [星光](#asetrate)
* [显示信息](#ashowinfo)
* [软剪辑](#asoftclip)
* [方面统计](#aspectralstats)
* [asr](#asr)
* [统计数据](#astats-1)
* [升压](#asubboost)
* [下切](#asubcut)
* [超级剪辑](#asupercut)
* [超级通行证](#asuperpass)
* [超级站](#asuperstop)
* [节奏](#atempo)
* [倾斜度](#atilt)
* [修剪](#atrim)
* [轴相关](#axcorrelate)
* [带通](#bandpass)
* [带阻](#bandreject)
* [低音，低架](#bass_002c-lowshelf)
* [双二阶](#biquad)
* [bs2b](#bs2b)
* [通道图](#channelmap)
* [频道分割](#channelsplit)
* [合唱](#chorus)
* [压缩](#compand)
* [补偿延迟](#compensationdelay)
* [交叉馈送](#crossfeed)
* [结晶器](#crystalizer)
* [直流偏移](#dcshift)
* [去嘶声](#deesser)
* [对话增强](#dialoguenhance)
* [测速表](#drmeter)
* [dynaudnorm](#dynaudnorm)
* [耳垢](#earwax)
* [均衡器](#equalizer)
* [超立体声](#extrastereo)
* [防火器](#firequalizer)
* [镶边器](#flanger)
* [哈斯](#haas)
* [硬盘驱动器](#hdcd)
* [耳机](#headphone)
* [高通](#highpass)
* [加入](#join)
* [水疗中心](#ladspa)
* [响度](#loudnorm)
* [低通](#lowpass)
* [lv2](#lv2)
* [mcompand](#mcompand)
* [平底锅](#pan-1)
* [回放增益](#replaygain)
* [重采样](#resample)
* [橡皮筋](#rubberband)
* [侧链压缩](#sidechaincompress)
* [侧链门](#sidechaingate)
* [静音检测](#silencedetect)
* [沉默移除](#silenceremove)
* [安抚剂](#sofalizer)
* [言语规范](#speechnorm)
* [立体工具](#stereotools)
* [立体加宽](#stereowiden)
* [超级均衡器](#superequalizer)
* [环绕](#surround)
* [倾斜架](#tiltshelf)
* [高音，高架](#treble_002c-highshelf)
* [颤音](#tremolo)
* [颤音](#vibrato)
* [低音提琴](#virtualbass)
* [体积](#volume)
* [体积检测](#volumedetect)

### 36.1 压缩机

压缩器主要用于降低信号的动态范围。尤其是现代音乐大多以高比例压缩以提高整体响度。这样做是为了获得听众的最高关注度，使声音“变肥”并为音轨带来更多“力量”。如果一个信号被压缩太多，之后它可能听起来沉闷或“死气沉沉”，或者它可能开始“泵”（这可能是一个强大的效果，但也可能完全破坏一个音轨）。正确的压缩是获得专业声音的关键，也是混音和母带处理的高等艺术。由于其复杂的设置，可能需要很长时间才能获得这种效果的正确感觉。

阈值的音量并将其除以用ratio设置的因子来完成的。因此，如果您将阈值设置为 -12dB，而您的信号达到 -6dB，则 2:1 的比率将导致 -9dB 的信号。因为对信号的精确操作会导致波形失真，所以可以随着时间的推移对减少进行调整。这是通过设置“攻击”和“释放”来完成的。 attack确定信号在任何减少发生之前必须上升到阈值以上的时间，而release设置信号必须下降到阈值以下以再次减少减少的时间。比所选启动时间短的信号将保持不变。信号的整体减少可以在之后用化妆设置来弥补。因此，将信号的峰值压缩大约 6dB 并将补偿提高到这个水平会导致信号的音量是源信号的两倍。为了在压缩中获得更柔和的入口，膝盖在所选分贝范围内的阈值处压平硬边。

过滤器接受以下选项：

level\_in

设置输入增益。默认值为 1。范围介于 0.015625 和 64 之间。

模式

设置压缩机运行模式。可以向上也可以向下。默认是向下的。

临界点

如果流信号上升到这个水平以上，它将影响增益降低。默认为 0.125。范围介于 0.00097563 和 1 之间。

比率

设置信号降低的比率。 1:2表示如果电平比阈值高出4dB，那么降低后只会比阈值高出2dB。默认值为 2。范围在 1 到 20 之间。

攻击

在增益降低开始之前信号必须上升到阈值以上的毫秒数。默认值为 20。范围介于 0.01 和 2000 之间。

发布

在再次减少减少之前信号必须低于阈值的毫秒数。默认值为 250。范围介于 0.01 和 9000 之间。

化妆品

通过处理后将放大多少信号来设置数量。默认值为 1。范围从 1 到 64。

膝盖

围绕阈值弯曲尖锐的膝盖以更柔和地进入增益减少。默认值为 2.82843。范围在 1 到 8 之间。

关联

选择是否输入流的所有通道之间的平均电平或输入流的较大（最大）通道影响降低。默认为平均。

检测

峰值的情况下应采用准确的信号，在rms的情况下应采用 RMS 信号。默认值为rms ，这通常更平滑。

混合

在输出中使用多少压缩信号。默认值为 1。范围介于 0 和 1 之间。

* [命令](#Commands)

#### 36.1.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.2对比

简单的音频动态范围压缩/扩展滤波器。

过滤器接受以下选项：

对比

设置对比度。默认值为 33。允许的范围在 0 到 100 之间。

### 36.3 一份

将输入音频源原样复制到输出。这主要用于测试目的。

### 36.4 淡入淡出

将一个输入音频流的淡入淡出应用到另一个输入音频流。淡入淡出应用于第一个流结束附近的指定持续时间。

过滤器接受以下选项：

nb\_样本，ns

指定交叉淡入淡出效果必须持续的样本数。在交叉淡入淡出效果结束时，第一个输入音频将完全静音。默认值为 44100。

持续时间，d

指定交叉淡入淡出效果的持续时间。有关可接受的语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分。默认情况下，持续时间由](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)nb\_samples确定。如果设置此选项而不是nb\_samples 。

重叠，o

如果第一个流结束与第二个流开始重叠。默认启用。

曲线1

为第一个流设置交叉淡入淡出过渡曲线。

曲线2

为第二个流设置淡入淡出过渡曲线。

有关可用曲线类型的描述，请参阅[淡入淡出](#afade)过滤器描述。

* [例子](#Examples-39)

#### 36.4.1 例子

* 从一个输入到另一个输入的淡入淡出：
* ffmpeg -i first.flac -i second.flac -filter\_complex acrossfade=d=10:c1=exp:c2=exp output.flac
* 从一个输入到另一个输入交叉淡入淡出但不重叠：
* ffmpeg -i first.flac -i second.flac -filter\_complex acrossfade=d=10:o=0:c1=exp:c2=exp output.flac

### 36.5跨度

将音频流分成几个波段。

此过滤器将音频流分成两个或多个频率范围。将所有流加起来将给出平坦的输出。

过滤器接受以下选项：

分裂

设置分割频率。这些必须是积极的并且在增加。

命令

为每个波段分割设置滤波器顺序。这控制滤波器滚降或滤波器传递函数的陡度。可用值是：

'第二'

每倍频程 12 分贝。

'第四'

每倍频程 24 分贝。

'第六'

每倍频程 36 分贝。

' 8日'

每倍频程 48 分贝。

'10号'

每倍频程 60 分贝。

'12号'

每倍频程 72 分贝。

'14日'

每倍频程 84 分贝。

'16日'

每倍频程 96 分贝。

'18号'

每倍频程 108 分贝。

'20日'

每倍频程 120 dB。

默认为第 4 位。

等级

设置输入增益电平。允许的范围是从 0 到 1。默认值为 1。

收益

为每个波段设置输出增益。所有波段的默认值为 1。

精确

设置处理样本时要使用的精度。

汽车

根据其他过滤器自动选择内部样本格式。

漂浮

始终使用单浮点精度样本格式。

双倍的

始终使用双浮点精度样本格式。

默认值为自动。

* [例子](#Examples-40)

#### 36.5.1 例子

* 将输入音频流分成两个频段（低频和高频），分割频率为 1500 Hz，每个频段将在单独的流中：
* ffmpeg -i in.flac -filter\_complex 'acrossover=split=1500[LOW][HIGH]' -map '[LOW]' low.wav -map '[HIGH]' high.wav
* 与上面相同，但具有更高的过滤器阶数：
* ffmpeg -i in.flac -filter\_complex 'acrossover=split=1500:order=8th[LOW][HIGH]' -map '[LOW]' low.wav -map '[HIGH]' high.wav
* 同上，但也有额外的中频带（频率在 1500 和 8000 之间）：
* ffmpeg -i in.flac -filter\_complex 'acrossover=split=1500 8000:order=8th[LOW][MID][HIGH]' -map '[LOW]' low.wav -map '[MID]' mid.wav -地图'[HIGH]' high.wav

### 36.6破碎机

降低音频位分辨率。

该过滤器是具有增强功能的比特破碎机。 bit crusher 用于在听觉上减少音频信号采样的位数。这根本不会改变位深度，它只会产生效果。位深度减少的材料听起来更刺耳和“数字化”。该过滤器甚至能够舍入为连续值而不是离散位深度。此外，它还有一个 D/C 偏移，这会导致信号的下半部分和上半部分产生不同的压缩。抗锯齿设置能够产生“更柔和”的破碎声音。

该滤波器的另一个特点是对数模式。此设置从位之间的线性距离切换为对数位。结果是一个更“自然”的声音破碎机，例如不会对低信号进行门控。人耳具有对数感知，所以这种碾压的感觉要悦耳得多。对数粉碎也能够消除锯齿。

过滤器接受以下选项：

level\_in

设置电平。

拉平

设置水平。

位

设置位减少。

混合

设定混合量。

模式

可以是线性的： lin或对数的： log 。

直流电

设置直流。

一个

设置抗锯齿。

样品

设置样本缩减。

低频

启用低频振荡器。默认禁用。

橙子

设置 LFO 范围。

无孔的

设置低频振荡器速率。

* [命令](#Commands-1)

#### 36.6.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.7 锐

延迟音频过滤直到给定的挂钟时间戳。请参阅[提示](#cue)过滤器。

### 36.8 declick

从输入音频中去除脉冲噪声。

使用自回归建模将检测为脉冲噪声的样本替换为插值样本。

窗口，w

设置窗口大小，以毫秒为单位。允许的范围是从10到100 。默认值为55毫秒。这设置将立即处理的窗口大小。

重叠，o

设置窗口重叠，以窗口大小的百分比表示。允许的范围是从50到95 。默认值为75 %。将此设置为非常高的值会增加脉冲噪声去除，但会使整个过程变慢。

订单，一个

设置自回归顺序，以窗口大小的百分比表示。允许的范围是从0到25 。默认值为2 %。此选项还控制使用相邻良好样本的插值样本的质量。

阈值，吨

设置阈值。允许的范围是从1到100 。默认值为2 。这控制了将被移除的脉冲噪声的强度。值越低，越多的样本将被检测为脉冲噪声。

爆裂，b

设置突发融合，以窗口大小的百分比表示。允许的范围是0到10 。默认值为2 。如果检测为噪声的任何两个样本的间距小于此值，则这两个样本之间的任何样本也将被检测为噪声。

方法，米

设置重叠方式。

它接受以下值：

添加，一个

选择重叠添加方法。使用这种方法，即使没有插值的样本也会有轻微的变化。

保存，s

选择重叠保存方法。未插值的样本保持不变。

默认值为一个.

### 36.9 剪辑

从输入音频中删除剪辑样本。

使用自回归建模将检测为裁剪的样本替换为插值样本。

窗口，w

设置窗口大小，以毫秒为单位。允许的范围是从10到100 。默认值为55毫秒。这设置将立即处理的窗口大小。

重叠，o

设置窗口重叠，以窗口大小的百分比表示。允许的范围是从50到95 。默认值为75 %。

订单，一个

设置自回归顺序，以窗口大小的百分比表示。允许的范围是从0到25 。默认值为8 %。此选项还控制使用相邻良好样本的插值样本的质量。

阈值，吨

设置阈值。允许的范围是从1到100 。默认值为10 。较高的值会使剪辑检测不那么激进。

尺寸，n

设置用于检测剪辑的直方图的大小。允许的范围是从100到9999 。默认值为1000 。较高的值会使剪辑检测不那么激进。

方法，米

设置重叠方式。

它接受以下值：

添加，一个

选择重叠添加方法。使用这种方法，即使没有插值的样本也会有轻微的变化。

保存，s

选择重叠保存方法。未插值的样本保持不变。

默认值为一个.

### 36.10 去相关

对输入音频流应用去相关。

过滤器接受以下选项：

阶段

设置过滤的去相关阶段。允许的范围是 1 到 16。默认值为 6。

种子

设置用于设置跨通道样本延迟的随机种子。

### 36.11 延迟

延迟一个或多个音频通道。

延迟通道中的样本充满了寂静。

过滤器接受以下选项：

延误

为以“|”分隔的每个通道设置延迟列表（以毫秒为单位）。未使用的延迟将被忽略。如果给定延迟的数量小于通道数，则所有剩余通道都不会被延迟。如果您想延迟确切数量的样本，请将“S”附加到数字。如果您想要以秒为单位延迟，请将“s”附加到数字。

全部

对所有剩余通道使用最后设置的延迟。默认情况下是禁用的。如果启用此选项，则会更改选项延迟的解释方式。

* [例子](#Examples-41)

#### 36.11.1 例子

* 将第一个通道延迟 1.5 秒，将第三个通道延迟 0.5 秒，并保持第二个通道（以及可能存在的任何其他通道）不变。
* 延迟=1500|0|500
* 将第二个通道延迟 500 个样本，将第三个通道延迟 700 个样本，并保持第一个通道（以及可能存在的任何其他通道）不变。
* 延迟=0|500S|700S
* 将所有通道延迟相同数量的样本：
* 延迟=延迟=64S：全部=1

### 36.12 腺病毒

通过添加极低级别的噪音来修复音频中的异常现象。

该过滤器应放置在任何可能产生异常的过滤器之前。

接受的参数的描述如下。

等级

以 dB 为单位设置添加噪声的级别。默认为-351 。允许的范围是从 -451 到 -90。

类型

设置添加噪声的类型。

直流电

添加直流信号。

交流电

添加交流信号。

正方形

添加方形信号。

脉冲

添加脉冲信号。

默认为直流。

* [命令](#Commands-2)

#### 36.12.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.13 衍生物，积分

计算音频流的导数/积分。

一个接一个地应用两个过滤器会产生原始音频。

### 36.14 阿德克

将频谱动态范围控制器过滤器应用于输入音频流。

接受的选项的描述如下。

转移

设置传递表达式。

表达式可以包含以下常量：

通道

当前频道号

锡

当前样品编号

nb\_频道

通道数

吨

以秒表示的时间戳

高级

采样率

p

当前频率功率值，单位为dB

F

当前频率（以 Hz 为单位）

默认值为p 。

攻击

以毫秒为单位设置攻击。默认值为50毫秒。允许的范围是 1 到 1000 毫秒。

发布

以毫秒为单位设置释放。默认值为100毫秒。允许的范围是 5 到 2000 毫秒。

频道

设置要过滤的通道，默认情况下过滤音频流中的所有通道。

* [命令](#Commands-3)
* [例子](#Examples-42)

#### 36.14.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

#### 36.14.2 例子

* 以 -50 dB 的阈值和 1:6 的比率对所有频率应用频谱压缩：
* adrc=transfer='if(gt(p,-50),-50+(p-(-50))/6,p)':attack=50:release=100
* 与上面类似，但比例为 1:2 且仅过滤前中央通道：
* adrc=transfer='if(gt(p,-50),-50+(p-(-50))/2,p)':attack=50:release=100:channels=FC
* 将频谱噪声门应用到所有频率，阈值为 -85 dB，启动时间和释放时间都很短：
* adrc=transfer='if(lte(p,-85),p-800,p)':attack=1:release=5
* 将频谱扩展应用于所有频率，阈值为 -10 dB，比率为 1:2：
* adrc=transfer='if(lt(p,-10),-10+(p-(-10))\*2,p)':attack=50:release=100
* 将限制器应用到最大 -60 dB 到所有频率，启动时间为 2 毫秒，释放时间为 10 毫秒：
* adrc=transfer='min(p,-60)':attack=2:release=10

### 36.15 动态均衡器

对输入音频流应用动态均衡。

接受的选项的描述如下。

临界点

设置用于触发均衡的检测阈值。阈值检测使用带通滤波器。默认值为 0。允许的范围为 0 到 100。

d频率

以 Hz 为单位设置用于触发均衡的带通滤波器的检测频率。默认值为 1000 赫兹。允许的范围在 2 到 1000000 赫兹之间。

dq因子

设置用于触发均衡的带通滤波器的检测共振因子。默认值为 1。允许的范围是从 0.001 到 1000。

频率

设置均衡滤波器的目标频率。默认值为 1000 赫兹。允许的范围在 2 到 1000000 赫兹之间。

因子

设置目标均衡滤波器的目标共振因子。默认值为 1。允许的范围是从 0.001 到 1000。

攻击

设置在均衡开始之前检测信号必须上升到检测阈值以上的毫秒数。默认值为 20。允许的范围在 1 到 2000 之间。

发布

设置在均衡结束之前检测信号必须低于检测阈值的毫秒数。默认值为 200。允许的范围在 1 到 2000 之间。

比率

设置提高均衡增益的比率。默认值为 1。允许的范围在 0 到 30 之间。

化妆品

设置提高均衡增益的补偿补偿。默认值为 0。允许的范围在 0 到 100 之间。

范围

设置允许的最大削减/提升量。默认值为 50。允许的范围为 1 到 200。

模式

设置过滤器运行的模式，可以是以下之一：

'听'

仅输出隔离的带通信号。

'切'

削减检测阈值以上的频率。

'提升'

提升频率低于检测阈值。

默认模式是“剪切”。

类型

设置目标过滤器的类型，可以是以下之一：

'铃'

'低架'

'高架'

默认类型是“ bell ”。

方向

设置相对于阈值的处理方向。

'向下'

如果阈值高于/低于检测到的音量，则提升/削减。

'向上'

如果阈值低于/高于检测到的音量，则提升/削减。

默认方向是“向下”。

汽车

自动从检测过滤器中收集阈值。默认情况下是“禁用”。此选项可用于检测输入音频流的特定时间范围内的阈值，在这种情况下，选项值会在运行时更改。

可用值是：

'禁用'

禁用使用自动收集的阈值。

'关闭'

停止选择阈值。

'上'

开始选择阈值。

* [命令](#Commands-4)

#### 36.15.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.16 adynamicsmooth

对输入音频流应用动态平滑。

接受的选项的描述如下。

灵敏度

设置对频率波动的敏感度。默认值为 2。允许的范围是从 0 到 1e+06。

基频

设置平滑的基频。默认值为 22050。允许的范围为 2 到 1e+06。

* [命令](#Commands-5)

#### 36.16.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.17回声

对输入音频应用回声。

回声是反射声，在说话或喊叫时自然会在山区（有时是大型建筑物）中出现；数字回声效果模仿这种行为，通常用于帮助填充单个乐器或人声的声音。原始信号和反射信号之间的时间差就是延迟，反射信号的响度就是衰减。多个回声可以有不同的延迟和衰减。

接受的参数的描述如下。

增益

设置反射信号的输入增益。默认值为0.6 。

输出增益

设置反射信号的输出增益。默认值为0.3 。

延误

设置以“|”分隔的原始信号和反射之间的时间间隔列表（以毫秒为单位）。每个延迟的允许范围是(0 - 90000.0] 。默认值为1000 。

衰变

设置以“|”分隔的反射信号的响度列表。每个衰减的允许范围是(0 - 1.0] 。默认值为0.5 。

* [例子](#Examples-43)

#### 36.17.1 例子

* 让它听起来好像有两倍于实际演奏的乐器：
* 回声=0.8:0.88:60:0.4
* 如果延迟很短，那么它听起来像是一个（金属）机器人在播放音乐：
* 回声=0.8:0.88:6:0.4
* 更长的延迟听起来就像山区的露天音乐会：
* 回声=0.8:0.9:1000:0.3
* 与上面相同，但多了一座山：
* aecho=0.8:0.9:1000|1800:0.3|0.25

### 36.18 强调

音频强调过滤器创建或恢复直接取自 LP 或具有不同过滤器曲线的强调 CD 的材料。例如，要在黑胶唱片上存储音乐，必须首先通过滤波器改变信号，以平衡这种记录介质的缺点。播放素材后，必须应用反向滤波器来恢复频率响应的失真。

过滤器接受以下选项：

level\_in

设置输入增益。

拉平

设置输出增益。

模式

设置过滤模式。恢复素材使用复制模式，否则使用生产模式。默认为再现模式。

类型

设置过滤器类型。选择介质。可以是以下之一：

山口

选择哥伦比亚。

emi

选择电磁干扰。

bsi

选择 BSI (78RPM)。

里亚阿

选择美国唱片业协会。

光盘

选择光盘 (CD)。

50调频

选择 50µs (FM)。

75调频

选择 75µs (FM)。

50kf

选择 50µs (FM-KF)。

75kf

选择 75µs (FM-KF)。

* [命令](#Commands-6)

#### 36.18.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.19 评估

根据指定的表达式修改音频信号。

该过滤器接受一个或多个表达式（每个通道一个），这些表达式被评估并用于修改相应的音频信号。

它接受以下参数：

表达式

为每个单独的通道设置以“|”分隔的表达式列表。如果输入通道的数量大于表达式的数量，则最后指定的表达式将用于剩余的输出通道。

频道布局，c

设置输出通道布局。如果未指定，通道布局由表达式数指定。如果设置为 ' same '，它将默认使用相同的输入通道布局。

exprs中的每个表达式都可以包含以下常量和函数：

通道

当前表情的通道号

n

评估样本的编号，从0开始

秒

采样率

吨

评估样品的时间以秒表示

nb\_in\_channels

nb\_out\_channels

输入输出通道数

值（CH）

CH的输入通道的值

注意：这个过滤器很慢。为了加快处理速度，您应该使用专用过滤器。

* [例子](#Examples-44)

#### 36.19.1 例子

* 半卷：
* aeval=val(ch)/2:c=相同
* 第二个通道的反相：
* aeval=val(0)|-val(1)

### 36.20 激励器

激励器用于产生原始信号中不存在的高音。这是通过在范围内限制并添加到原始信号中的信号产生谐波失真来实现的。 Exciter 提升音频信号的高端，而不是像均衡器那样简单地提升较高频率来创建更“清脆”或“明亮”的声音。

过滤器接受以下选项：

level\_in

设置信号处理前的输入电平。允许的范围是 0 到 64。默认值为 1。

拉平

设置信号处理后的输出电平。允许的范围是 0 到 64。默认值为 1。

数量

设置添加到原始信号的谐波量。允许的范围是 0 到 64。默认值为 1。

驾驶

设置新创建的谐波的数量。允许的范围是 0.1 到 10。默认值为 8.5。

混合

设置新创建的谐波的八度。允许的范围是从 -10 到 10。默认值为 0。

频率

以 Hz 为单位设置产生谐波的频率下限。允许的范围是从 2000 到 12000 赫兹。默认值为 7500 赫兹。

天花板

设置产生谐波的频率上限。允许的范围是从 9999 到 20000 赫兹。如果值低于 10000 Hz，则不应用任何限制。

听

将原始信号静音并仅输出添加的谐波。默认情况下是禁用的。

* [命令](#Commands-7)

#### 36.20.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.21 淡出

对输入音频应用淡入/淡出效果。

接受的参数的描述如下。

类型，吨

指定效果类型，可以是淡入淡入或淡出淡出效果。默认为.

开始样本，ss

指定开始应用淡入淡出效果的起始样本数。默认为 0。

nb\_样本，ns

指定淡入淡出效果必须持续的样本数。在淡入效果结束时，输出音频将具有与输入音频相同的音量，在淡出过渡结束时，输出音频将为静音。默认值为 44100。

开始时间，st

指定淡入淡出效果的开始时间。默认值为 0。该值必须指定为持续时间；有关可接受的语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)如果设置此选项而不是start\_sample 。

持续时间，d

指定淡入淡出效果的持续时间。有关可接受的语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)在淡入效果结束时，输出音频将具有与输入音频相同的音量，在淡出过渡结束时，输出音频将为静音。默认情况下，持续时间由nb\_samples确定。如果设置此选项而不是nb\_samples 。

曲线

设置淡入淡出过渡曲线。

它接受以下值：

三

选择三角形，线性斜率（默认）

qsin

选择正弦波的四分之一

新

选择正弦波的一半

树脂

选择指数正弦波

日志

选择对数

ipar

选择倒抛物线

质量

选择二次

幼兽

选择立方体

平方

选择平方根

CBR

选择立方根

标准杆

选择抛物线

exp

选择指数

iqsin

选择反四分之一的正弦波

爱信

选择反半正弦波

德塞

选择双指数位

德西

选择双指数 sigmoid

洛西

选择逻辑 sigmoid

正弦波

选择正弦基数函数

isinc

选择反正弦基数函数

不褪色

没有应用淡入淡出

安静

设置淡入的初始增益或淡出的最终增益。默认值为0.0 。

统一

设置淡出的初始增益或淡入的最终增益。默认值为1.0 。

* [命令](#Commands-8)
* [例子](#Examples-45)

#### 36.21.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

#### 36.21.2 例子

* 淡出前 15 秒的音频：
* afade=t=in:ss=0:d=15
* 淡出 900 秒音频的最后 25 秒：
* afade=t=out:st=875:d=25

### 36.22 阿夫顿

使用 FFT 对音频样本进行降噪。

接受的参数的描述如下。

降噪，nr

以 dB 为单位设置降噪，允许的范围是 0.01 到 97。默认值为 12 dB。

本底噪声，nf

以 dB 为单位设置本底噪声，允许的范围是 -80 到 -20。默认值为 -50 分贝。

噪声类型，nt

设置噪声类型。

它接受以下值：

白色

选择白噪声。

黑胶唱片，v

选择乙烯基噪音。

虫胶, s

选择虫胶噪音。

习俗，c

选择自定义噪声，在bn选项中定义。

默认值为白噪声。

带噪声，bn

为 15 个频段中的每一个频段设置自定义频段噪声配置文件。带由“”或“|”分隔。

残差地板，射频

以 dB 为单位设置残差底限，允许的范围是 -80 到 -20。默认值为 -38 分贝。

track\_noise, tn

启用本底噪声跟踪。默认情况下是禁用的。启用此功能后，本底噪声会自动调整。

track\_residual, tr

启用残留跟踪。默认情况下是禁用的。

输出模式，OM

设置输出模式。

它接受以下值：

输入，我

传递输入不变。

输出，o

通过噪声过滤掉。

噪声，n

只传递噪音。

默认值为输出。

适应性，广告

设置适应性因子，用于适应每个频率仓的增益调整的速度。值0启用即时适应，而较高的值反应慢得多。允许的范围是从0到1 。默认值为0.5 。

floor\_offset, fo

设置本底噪声偏移因子。此选项用于调整应用于测量本底噪声的偏移。它仅在启用本底噪声跟踪时有效。允许的范围是从-2.0到2.0 。默认值为1.0 。

noise\_link, nl

设置用于多声道音频的噪声链接。

它接受以下值：

没有任何

使用未更改的通道的本底噪声。

分钟

使用所有通道的测量最小本底噪声。

最大限度

使用所有通道的测量最大本底噪声。

平均的

使用所有通道的测量平均本底噪声。

默认值为min 。

band\_multiplier，bm

设置频带倍增因子，用于跨频率仓扩展频带的量。允许的范围是从0.2到5 。默认值为1.25 。

样本噪声，sn

切换从输入音频捕获和测量噪声曲线。

它接受以下值：

开始，开始

开始样本噪声捕获。

停止，结束

停止样本噪声捕获并测量新的噪声带配置文件。

默认值为无。

增益平滑，GS

设置增益平滑空间半径，用于平滑应用于每个频率仓的增益。有助于减少随机音乐噪声伪影。较高的值会增加增益的平滑度。允许的范围是从0到50 。默认值为0 。

* [命令](#Commands-9)
* [例子](#Examples-46)

#### 36.22.1 命令

这个过滤器支持上面提到的一些选项作为[命令](#commands)。

#### 36.22.2 例子

* 将白噪声降低 10dB，并使用之前测得的 -40dB 本底噪声：
* afftdn=nr=10:nf=-40
* 将白噪声降低 10dB，并将初始本底噪声设置为 -80dB，并启用本底噪声自动跟踪，使本底噪声在处理过程中逐渐变化：
* afftdn=nr=10:nf=-80:tn=1
* 将噪声降低 20dB，使用 -40dB 的本底噪声并使用命令获取输入音频前 0.4 秒的噪声曲线：
* asendcmd=0.0 afftdn sn 开始，asendcmd=0.4 afftdn sn 停止，afftdn=nr=20:nf=-40

### 36.23 关系

将任意表达式应用于频域中的样本。

真实的

为以“|”分隔的每个单独通道设置频域实数表达式。默认为“重新”。如果输入通道的数量大于表达式的数量，则最后指定的表达式将用于剩余的输出通道。

图片

为以“|”分隔的每个单独通道设置频域虚数表达式。默认为“im”。

real和imag中的每个表达式都可以包含以下常量和函数：

高级

采样率

b

当前频点数

注意事项

可用箱数

通道

当前表情的通道号

通道

通道数

积分

当前帧点

关于

当前通道频点的当前实部

我是

当前通道频点的当前虚部

真实的（b，ch）

bin ， channel ）处频率仓的实部值

图片(b, ch)

bin , channel )处频率仓的虚部值

win\_size

设置窗口大小。允许的范围是 16 到 131072。默认值为4096

win\_func

设置窗函数。

它接受以下值：

'正确'

'巴特利特'

'汉，汉宁'

'海明'

'黑人'

'韦尔奇'

'平顶'

'巴里斯'

' bnuttall '

'巴恩'

'正弦'

'坚果'

'兰佐斯'

'高斯'

'图基'

'海豚'

'柯西'

' parzen '

'泊松'

'波曼'

'凯撒'

默认为hann 。

重叠

设置窗口重叠。如果设置为 1，将选择所选窗口函数的建议重叠。默认值为0.75 。

* [例子](#Examples-47)

#### 36.23.1 例子

* 在音频中几乎只保留低频：
* afftfilt="'real=re \* (1-clip((b/nb)\*b,0,1))':imag='im \* (1-clip((b/nb)\*b,0,1) )'"
* 应用机器人效果：
* afftfilt="real='hypot(re,im)\*sin(0)':imag='hypot(re,im)\*cos(0)':win\_size=512:overlap=0.75"
* 应用耳语效果：
* afftfilt="real='hypot(re,im)\*cos((random(0)\*2-1)\*2\*3.14)':imag='hypot(re,im)\*sin((random(1)\* 2-1)\*2\*3.14)':win\_size=128:overlap=0.8"
* 应用相移：
* afftfilt="real=re\*cos(1)-im\*sin(1):imag=re\*sin(1)+im\*cos(1)"

### 36.24 冷

应用任意有限脉冲响应滤波器。

此滤波器设计用于应用长达 60 秒的长 FIR 滤波器。

它可以用作数字分频滤波器、房间均衡、串扰消除、波场合成、可听化、立体声、立体声和空间化的组件。

该过滤器使用高于第一个的流作为 FIR 系数。如果非第一流包含单个通道，它将用于第一流中的所有输入通道，否则非第一流中的通道数必须与第一流中的通道数相同。

它接受以下参数：

干燥

设置干增益。这设置输入增益。

湿的

设置湿增益。这设置了最终输出增益。

长度

设置脉冲响应滤波器长度。默认值为 1，表示处理整个 IR。

类型

启用应用从 IR 功率测量的增益。

设置用于自动增益测量的方法。

没有任何

不要应用任何增益。

顶峰

选择峰值增益，非常保守的做法。这是默认值。

直流电

选择直流增益，应用有限。

gn

选择增益噪声方法，这是最流行的一种。

交流电

选择交流增益。

有效值

选择 RMS 增益。

伊尔根

设置在过滤前应用于 IR 系数的增益。允许的范围是 0 到 1。此增益在使用gtype选项应用任何增益之后应用。

红外线

设置 IR 流的格式。可以是单声道或输入。默认为输入。

马克西尔

设置允许的最大脉冲响应滤波器持续时间（以秒为单位）。默认值为 30 秒。允许的范围是 0.1 到 60 秒。

回复

在附加视频流中显示 IR 频率响应、幅度（洋红色）、相位（绿色）和群延迟（黄色）。默认情况下它是禁用的。

渠道

设置显示频率响应的红外通道。默认情况下显示第一个频道。此选项仅在启用响应时使用。

尺寸

设置视频流大小。此选项仅在启用响应时使用。

速度

设置视频流帧率。此选项仅在启用响应时使用。

minp

设置用于卷积的最小分区大小。默认为8192 。允许的范围是从1到65536 。较低的值会以更高的 CPU 使用率为代价减少延迟。

最大值

设置用于卷积的最大分区大小。默认为8192 。允许的范围是从8到65536 。较低的值可能会增加 CPU 使用率。

nbirs

设置在运行时可切换的输入脉冲响应流的数量。允许的范围是从1到32 。默认为1 。

红外线

设置将用于卷积的 IR 流，从0开始，应始终低于nbirs选项提供的值。默认为0 。这个选项可以在运行时通过[命令](#commands)改变。

精确

设置处理样本时要使用的精度。

汽车

根据其他过滤器自动选择内部样本格式。

漂浮

始终使用单浮点精度样本格式。

双倍的

始终使用双浮点精度样本格式。

默认值为自动。

* [例子](#Examples-48)

#### 36.24.1 例子

* 使用单声道 IR 文件作为第二个输入将混响应用于流，使用 ffmpeg 完成命令：
* ffmpeg -i input.wav -i middle\_tunnel\_1way\_mono.wav -lavfi afir output.wav
* 给定输入立体声流应用真正的立体声处理，以及左右声道的两个立体声脉冲响应，脉冲响应文件是名为 l\_ir.wav 和 r\_ir.wav 的文件：
* "pan=4C|c0=FL|c1=FL|c2=FR|c3=FR[a];amovie=l\_ir.wav[LIR];amovie=r\_ir.wav[RIR];[LIR][RIR]合并[ ir];[a][ir]afir=irfmt=input:gtype=gn:irgain=-5dB,pan=stereo|FL<c0+c2|FR<c1+c3"

### 36.25格式

设置输入音频的输出格式约束。该框架将协商最合适的格式以最大程度地减少转换。

它接受以下参数：

sample\_fmts, f

以“|”分隔的请求示例格式列表。

采样率，r

以“|”分隔的请求采样率列表。

频道布局，cl

以“|”分隔的请求频道布局列表。

有关所需语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 通道布局部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#channel-layout-syntax)

如果省略参数，则允许所有值。

强制输出为无符号 8 位或有符号 16 位立体声

aformat=sample\_fmts=u8|s16:channel\_layouts=立体声

### 36.26 调频

将频移应用于输入音频样本。

过滤器接受以下选项：

转移

指定频移。允许的范围是 -INT\_MAX 到 INT\_MAX。默认值为 0.0。

等级

设置应用于最终输出的输出增益。允许的范围是从 0.0 到 1.0。默认值为 1.0。

命令

设置用于过滤的过滤顺序。允许的范围是 1 到 16。默认值为 8。

* [命令](#Commands-10)

#### 36.26.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.27 AFWDN

使用小波减少输入样本的宽带噪声。

接受的选项的描述如下。

西格玛

设置噪声西格玛，允许范围为 0 到 1。默认值为 0。此选项控制应用于输入样本的去噪强度。设置此选项最有用的方法是通过分贝，例如。 -45分贝。

等级

设置分解的小波级数。允许的范围是从 1 到 12。默认值为 10。设置太低会使去噪性能很差。

小波

设置输入帧分解的小波类型。它们按系数的数量从低到高排序。更多的系数意味着更差的过滤速度，但整体质量更好。可用的小波是：

' sym2 '

' sym4 '

' rbior68 '

' deb10 '

' sym10 '

' coif5 '

' bl3 '

百分

设置完全去噪的百分比。允许的范围是从 0% 到 100%。默认值为 85% 或部分去噪。

轮廓

如果启用，第一个输入帧将用作噪声配置文件。如果第一帧样本包含非噪声，性能会很差。

自适应的

如果启用，将分析输入帧是否存在噪声。如果很有可能检测到噪声，则输入帧配置文件将用于处理后续帧，直到检测到新的噪声帧。

样品

在样本数中设置单帧的大小。允许的范围是从 512 到 65536。默认帧大小是 8192 个样本。

柔软度

设置在阈值函数内应用的柔软度。允许的范围是从 0 到 10。默认柔和度是 1。

* [命令](#Commands-11)

#### 36.27.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.28玛瑙

门主要用于减少信号的较低部分。这种信号处理减少了有用信号之间的干扰噪声。

阈值的音量并将其除以用ratio设置的因子来完成的。本底噪声的底部通过范围设置。因为对信号的精确操作会导致波形失真，所以可以随着时间的推移对减少进行调整。这是通过设置attack和release来完成的。

attack决定了在发生任何减少之前信号必须下降到阈值以下的时间， release设置信号必须上升到阈值以上以再次减少减少的时间。比所选启动时间短的信号将保持不变。

level\_in

过滤前设置输入电平。默认值为 1。允许的范围是从 0.015625 到 64。

模式

设置操作模式。可以向上也可以向下。默认是向下的。如果设置为向上模式，信号的较高部分将被放大，向上扩展动态范围。否则，如遇向下信号会减弱。

范围

设置信号低于阈值时增益降低的级别。默认值为 0.06125。允许的范围是从 0 到 1。将此设置为 0 将禁用缩减，然后过滤器的行为类似于扩展器。

临界点

如果信号上升到该水平以上，则释放增益降低。默认值为 0.125。允许的范围是从 0 到 1。

比率

设置信号降低的比率。默认值为 2。允许的范围为 1 到 9000。

攻击

在增益减少停止之前信号必须上升到阈值以上的毫秒数。默认值为 20 毫秒。允许的范围是从 0.01 到 9000。

发布

在再次增加减少量之前，信号必须低于阈值的毫秒数。默认值为 250 毫秒。允许的范围是从 0.01 到 9000。

化妆品

设置处理后信号的放大量。默认值为 1。允许的范围为 1 到 64。

膝盖

围绕阈值弯曲尖锐的膝盖以更柔和地进入增益减少。默认值为 2.828427125。允许的范围是 1 到 8。

检测

选择是否应采用精确信号进行检测或采用 RMS 之类的信号。默认值为rms 。可以是峰值或有效值。

关联

选择是否所有通道之间的平均电平或较大通道影响降低。默认为平均。可以是平均值或最大值。

* [命令](#Commands-12)

#### 36.28.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.29 空气

应用任意无限脉冲响应滤波器。

它接受以下参数：

零点，z

设置 B/分子/零点/反射系数。

极点，p

设置 A/分母/极点/阶梯系数。

增益，k

设置通道增益。

干增益

设置输入增益。

增湿

设置输出增益。

格式，f

设置系数格式。

'会'

格梯函数

' sf '

模拟传递函数

' tf '

数字传递函数

' zp '

Z 平面零点/极点，笛卡尔（默认）

'公关'

Z 平面零点/极点、极坐标弧度

' PD '

Z 平面零点/极点，极度

' sp '

S 平面零点/极点

过程，r

设置处理类型。

' d '

直接加工

的\_ \_

串行处理

' p '

并行处理

精度，e

设置过滤精度。

' dbl '

双精度浮点数（默认）

'飞'

单精度浮点数

' i32 '

32 位整数

' i16 '

16 位整数

归一化，n

归一化滤波器系数，默认情况下启用。启用它会将 DC 的幅度响应归一化为 0dB。

混合

在输出中使用多少过滤信号。默认值为 1。范围介于 0 和 1 之间。

回复

在附加视频流中显示 IR 频率响应、幅度（洋红色）、相位（绿色）和群延迟（黄色）。默认情况下它是禁用的。

渠道

设置显示频率响应的红外通道。默认情况下显示第一个频道。此选项仅在启用响应时使用。

尺寸

设置视频流大小。此选项仅在启用响应时使用。

tf和sf格式的系数以空格分隔，并按升序排列。

zp格式的系数由空格分隔，系数的顺序无关紧要。 zp格式的系数是具有i虚数单位的复数。

可以为每个通道提供不同的系数和增益，在这种情况下使用'|'分离系数或增益。最后提供的系数将用于所有剩余的通道。

* [例子](#Examples-49)

#### 36.29.1 例子

* 对于 48000 Hz 采样率，在 5000Hz 左右应用 2 极椭圆陷波：
* aiir=k=1:z=7.957584807809675810E-1 -2.575128568908332300 3.674839853930788710 -2.57512875289799137 7.957586296317130880E-1:p= 1 -2.86950072432325953 3.63022088054647218 -2.28075678147272232 6.361362326477423500E-1: f=tf:r=d
* 与上面相同，但采用zp格式：
* aiir=k=0.79575848078096756:z=0.80918701+0.58773007i 0.80918701-0.58773007i 0.80884700+0.58784055i 0.80884700-0.58784055i:p= 0.63892345+0.59951235i 0.63892345-0.59951235i 0.79582691+0.44198673i 0.79582691-0.44198673i:f=zp:r=秒
* 应用三阶模拟归一化巴特沃斯低通滤波器，使用模拟传递函数格式：
* aiir=z=1.3057 0 0 0:p=1.3057 2.3892 2.1860 1:f=sf:r=d

### 36.30 限制器

限幅器防止输入信号上升超过所需的阈值。该限制器使用前瞻技术来防止您的信号失真。这意味着信号处理后有一个小的延迟。请记住，它产生的延迟是您设置的启动时间。

过滤器接受以下选项：

level\_in

设置输入增益。默认值为 1。

拉平

设置输出增益。默认值为 1。

限制

不要让高于此电平的信号通过限幅器。默认值为 1。

攻击

限制器将在这段时间内达到其衰减级别（以毫秒为单位）。默认值为 5 毫秒。

发布

在此毫秒数内从限制恢复到衰减 1.0。默认值为 50 毫秒。

升序

当始终需要降低增益时，ASC 会负责释放到平均降低水平，而不是在释放时间内达到 0 的降低。

asc\_level

选择释放时间受 ASC 影响的程度，0 表示释放时间几乎没有变化，而 1 表示释放时间更长。

等级

自动电平输出信号。默认启用。如果启用，这会将音频归一化为 0dB。

潜伏

通过使用带有攻击参数的先行缓冲区集来补偿引入的延迟。当流达到 EOF 时，还会刷新先行缓冲区中的有效音频数据。

根据选择的设置，建议在应用此过滤器之前使用[aresample将输入上采样 2 倍或 4 倍。](#aresample)

### 36.31 全通

frequency和 filter-width width 的双极全通滤波器。全通滤波器改变音频的频率与相位的关系，而不改变其频率与振幅的关系。

过滤器接受以下选项：

频率，f

以 Hz 为单位设置频率。

width\_type, t

设置指定滤波器带宽的方法。

H

赫兹

q

Q因子

o

八度

秒

坡

k

千赫

宽度，w

以 width\_type 单位指定过滤器的带宽。

混合，米

在输出中使用多少过滤信号。默认值为 1。范围介于 0 和 1 之间。

频道，c

指定要过滤的通道，默认情况下会过滤所有可用的通道。

归一化，n

Normalize biquad coefficients，默认情况下是禁用的。启用它会将 DC 的幅度响应归一化为 0dB。

订单，o

设置过滤器顺序，可以是 1 或 2。默认是 2。

变换，一个

设置 IIR 滤波器的变换类型。

滴

迪

时差

tdii

拉特

svf

zdf

精度，r

设置过滤精度。

汽车

根据环绕滤波器选择自动样本格式。

s16

始终使用带符号的 16 位。

s32

始终使用带符号的 32 位。

f32

始终使用 32 位浮点数。

f64

始终使用 64 位浮点数。

* [命令](#Commands-13)

#### 36.31.1 命令

该过滤器支持以下命令：

频率，f

改变全通频率。命令的语法是：“频率”

width\_type, t

更改全通 width\_type。命令的语法是：“ width\_type ”

宽度，w

改变全通宽度。命令的语法是：“ width ”

混合，米

更改全通混音。命令的语法是：“ mix ”

### 36.32 单循环

循环音频样本。

过滤器接受以下选项：

环形

设置循环次数。将此值设置为 -1 将导致无限循环。默认为 0。

尺寸

设置最大样本数。默认为 0。

开始

设置循环的第一个样本。默认为 0。

### 36.33 合并

将两个或多个音频流合并为一个多通道流。

过滤器接受以下选项：

输入

设置输入的数量。默认值为 2。

如果输入的通道布局不相交，因此兼容，输出的通道布局将相应设置，通道将根据需要重新排序。如果输入的通道布局不相交，则输出将具有第一个输入的所有通道，然后是第二个输入的所有通道，按此顺序，输出的通道布局将是对应于频道总数。

例如，如果第一个输入是 2.1 (FL+FR+LF)，第二个输入是 FC+BL+BR，则输出将是 5.1，通道顺序如下：a1、a2、b1、 a3, b2, b3（a1是第一个输入的第一个通道，b1是第二个输入的第一个通道）。

另一方面，如果两个输入均为立体声，则输出通道将采用默认顺序：a1、a2、b1、b2，通道布局将任意设置为 4.0，这可能是也可能不是预期值.

所有输入必须具有相同的采样率和格式。

如果输入的持续时间不同，则输出将以最短的时间停止。

* [例子](#Examples-50)

#### 36.33.1 例子

* 将两个单声道文件合并为一个立体声流：
* amovie=left.wav [l] ; amovie=right.mp3 [r] ; [l] [r] 合并
* input.mkv中有 1 个视频流和 6 个音频流：
* ffmpeg -i input.mkv -filter\_complex "[0:1][0:2][0:3][0:4][0:5][0:6] amerge=inputs=6" -c:a pcm\_s16le输出.mkv

### 36.34 混音

将多个音频输入混合到一个输出中。

请注意，此过滤器仅支持浮动样本（ amerge和pan音频过滤器支持多种格式）。如果amix输入有整数样本，则会自动插入[aresample](#aresample)以执行到浮点样本的转换。

它接受以下参数：

输入

输入的数量。如果未指定，则默认为 2。

期间

如何确定流结束。

最长

最长输入的持续时间。 （默认）

最短的

最短输入的持续时间。

第一的

第一个输入的持续时间。

辍学过渡

输入流结束时体积重新归一化的过渡时间（以秒为单位）。默认值为 2 秒。

权重

将每个输入音频流的权重指定为由空格分隔的数字序列。如果与输入数量相比指定的权重较少，则将最后的权重分配给剩余的输入。每个输入的默认权重为 1。

归一化

始终缩放输入而不是仅对样本求和。如果在禁用此选项的情况下在此过滤器过滤之前或之后未对输入进行归一化，请注意严重削波。默认情况下启用。

* [例子](#Examples-51)
* [命令](#Commands-14)

#### 36.34.1 例子

* 这会将 3 个输入音频流混合到一个输出，其持续时间与第一个输入相同，并且 dropout 转换时间为 3 秒：
* ffmpeg -i INPUT1 -i INPUT2 -i INPUT3 -filter\_complex amix=inputs=3:duration=first:dropout\_transition=3 输出
* 这会将一个人声和一个音乐输入音频流混合到一个持续时间与最长输入相同的输出。音乐的重量将是人声的四分之一，并且输入未标准化：
* ffmpeg -i VOCALS -i MUSIC -filter\_complex amix=inputs=2:duration=longest:dropout\_transition=0:weights="1 0.25":normalize=0 输出

#### 36.34.2 命令

该过滤器支持以下命令：

权重

归一化

语法与同名选项相同。

### 36.35乘以

将第一音频流与第二音频流相乘并将结果存储在输出音频流中。乘法是通过将来自第一个流的每个样本与来自第二个流的相同位置的样本相乘来完成的。

通过这种逐元素乘法，可以创建振幅衰减和振幅调制。

### 36.36 均衡器

每个通道的高阶参数多频带均衡器。

它接受以下参数：

参数

此选项字符串的格式为：“c chn f= cf w= w g= g t= f | ...” 每个均衡器频段由“|”分隔。

中文

设置将应用均衡的通道号。如果输入没有该通道，则忽略该条目。

F

设置频段的中心频率。如果输入不具有该频率，则忽略该条目。

w

以赫兹为单位设置带宽。

G

以 dB 为单位设置频带增益。

吨

设置波段的滤波器类型，可选，可以是：

' 0 '

Butterworth，这是默认设置。

' 1 '

切比雪夫 1 型。

' 2 '

切比雪夫 2 型。

曲线

使用此选项激活的均衡器频率响应显示在视频流中。

尺寸

设置视频流大小。仅在激活曲线选项时有用。

增益

设置将显示的最大增益。仅在激活曲线选项时有用。将此设置为一个合理的值可以显示增益，该增益来自彼此太近的相邻频段，因此当两个频段都被激活时会产生更高的增益。

规模

设置用于在视频输出中绘制频率响应的频率比例。可以是线性的或对数的。默认为对数。

颜色

为将要在视频流中显示的每个通道曲线设置颜色。这是由空格或“|”分隔的颜色名称列表。无法识别或丢失的颜色将被白色替换。

* [例子](#Examples-52)
* [命令](#Commands-15)

#### 36.36.1 例子

* 使用 Chebyshev 1 型滤波器将前 2 个通道的增益降低 10 的中心频率 200Hz 和宽度 100Hz：
* 均衡器=c0 f=200 w=100 g=-10 t=1|c1 f=200 w=100 g=-10 t=1

#### 36.36.2 命令

该过滤器支持以下命令：

改变

改变现有的过滤器参数。命令的语法是：“ fN |f= freq |w= width |g= gain ”

fN是现有的过滤器编号，从 0 开始，如果没有这样的过滤器，则返回错误。 freq设置新的频率参数。 width以赫兹为单位设置新的宽度参数。 gain以 dB 为单位设置新的增益参数。

使用 asendcmd 的完整过滤器调用可能如下所示：asendcmd=c='4.0 anequalizer change 0|f=200|w=50|g=1',anequalizer=...

### 36.37 anlmdn

使用非局部均值算法减少音频样本中的宽带噪声。

通过寻找具有相似上下文的其他样本来调整每个样本。这种上下文相似性是通过比较它们周围大小为p的块来定义的。在样本周围r的区域中搜索补丁。

过滤器接受以下选项：

力量，小号

设置去噪强度。允许的范围是从 0.00001 到 10000。默认值为 0.00001。

补丁，p

设置补丁半径持续时间。允许的范围是 1 到 100 毫秒。默认值为 2 毫秒。

研究, r

设置研究半径持续时间。允许的范围是 2 到 300 毫秒。默认值为 6 毫秒。

输出，o

设置输出模式。

它接受以下值：

我

传递输入不变。

o

通过噪声过滤掉。

n

只传递噪音。

默认值为o 。

光滑，米

设置平滑系数。默认值为11 。允许的范围是从1到1000 。

* [命令](#Commands-16)

#### 36.37.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.38 anlmf，anlms

使用第二个音频流将归一化最小均值（平方|第四）算法应用于第一个音频流。

该自适应滤波器用于通过找到与产生误差信号的最小均方相关的滤波器系数（所需的第二输入音频流与实际信号第一输入音频流之间的差异）来模拟所需的滤波器。

接受的选项的描述如下。

命令

设置过滤器顺序。

亩

设置过滤器亩。

每股收益

设置过滤器 eps。

泄漏

设置过滤器泄漏。

过时

它接受以下值：

我

传递第一个输入。

d

传递第二个输入。

o

通过过滤样品。

n

通过所需样本和过滤样本之间的差异。

默认值为o 。

* [例子](#Examples-53)
* [命令](#Commands-17)

#### 36.38.1 例子

* 这个过滤器的许多用途之一是降噪，输入音频使用延迟固定量的相同样本进行过滤，立体声音频的一个这样的例子是：
* asplit[a][b],[a]adelay=32S|32S[a],[b][a]anlms=order=128:leakage=0.0005:mu=.5:out\_mode=o

#### 36.38.2 命令

此过滤器支持与选项相同的命令，但不包括 option order 。

### 36.39 无效

将音频源原样传递到输出。

### 36.40 帕德

用静音填充音频流的末尾。

ffmpeg一起使用 -shortest将音频流扩展到与视频流相同的长度。

接受的选项的描述如下。

数据包大小

设置静音数据包大小。默认值为 4096。

pad\_len

设置要添加到末尾的静音样本数。达到该值后，流将终止。此选项与whole\_len互斥。

全长

设置输出音频流中的最小样本总数。如果该值比输入音频长度长，则在末尾添加静音，直到达到该值。此选项与pad\_len互斥。

pad\_dur

指定要添加的静音样本的持续时间。请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax) 对于接受的语法。仅在设置为非负值时使用。

whole\_dur

指定输出音频流中的最小总持续时间。有关可接受的语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)仅在设置为非负值时使用。如果该值比输入音频长度长，则在末尾添加静音，直到达到该值。该选项与pad\_dur互斥

如果pad\_len 、 whole\_len 、 pad\_dur或whole\_dur选项均未设置，则过滤器将无限期地在输入流的末尾添加静默。

请注意，对于 ffmpeg 4.4 及更早版本，零pad\_dur或whole\_dur也会导致过滤器无限期地添加静音。

* [例子](#Examples-54)

#### 36.40.1 例子

* 在输入末尾添加 1024 个静音样本：
* apad=pad\_len=1024
* 确保音频输出至少包含 10000 个样本，如果需要，用静音填充输入：
* apad=whole\_len=10000
* 使用ffmpeg用静音填充音频输入，这样视频流将始终是最短的，并且在使用最短选项时将转换到输出文件的结尾：
* ffmpeg -i VIDEO -i AUDIO -filter\_complex "[1:0]apad" -最短输出

### 36.41 移相器

为输入音频添加相位效果。

移相器滤波器在频谱中创建一系列波峰和波谷。波峰和波谷的位置经过调制，因此它们会随时间变化，从而产生扫荡效果。

接受的参数的描述如下。

增益

设置输入增益。默认值为 0.4。

输出增益

设置输出增益。默认值为 0.74

延迟

以毫秒为单位设置延迟。默认值为 3.0。

衰变

设置衰减。默认值为 0.4。

速度

以 Hz 为单位设置调制速度。默认值为 0.5。

类型

设置调制类型。默认为三角形。

它接受以下值：

'三角形， '

'正弦曲线，s '

### 36.42 相移

将相移应用于输入音频样本。

过滤器接受以下选项：

转移

指定相移。允许的范围是从 -1.0 到 1.0。默认值为 0.0。

等级

设置应用于最终输出的输出增益。允许的范围是从 0.0 到 1.0。默认值为 1.0。

命令

设置用于过滤的过滤顺序。允许的范围是 1 到 16。默认值为 8。

* [命令](#Commands-18)

#### 36.42.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.43 精神分裂症

将 Psychoacoustic clipper 应用于输入音频流。

过滤器接受以下选项：

level\_in

设置输入增益。默认为 1。范围为 [0.015625 - 64]。

拉平

设置输出增益。默认为 1。范围为 [0.015625 - 64]。

夹子

设置裁剪起始值。默认值为 0dBFS 或 1。

差异

仅输出差异样本，有助于听到引入的失真。默认情况下是禁用的。

自适应的

设置应用的自适应失真强度。默认值为 0.5。允许的范围是从 0 到 1。

迭代

设置心理声学剪刀的迭代次数。允许的范围是 1 到 20。默认值为 10。

等级

自动电平输出信号。默认是禁用的。如果启用，这会将音频标准化回 0dBFS。

* [命令](#Commands-19)

#### 36.43.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.44 脉动器

音频脉动器介于自动声相器和颤音器之间。但它也可以产生有趣的立体声效果。 Pulsator 基于具有不同波形和移相的 LFO（低频振荡器）改变左右声道的音量。此过滤器能够定义左右通道之间的偏移量。偏移量 0 表示两个 LFO 形状相互匹配。左右声道同样改变 - 传统的颤音。 50% 的偏移意味着右声道的形状在相位上完全偏移（或向后移动大约一半的频率）- 脉动器充当自动声相器。在 1 时，两条曲线再次匹配。两者之间的每个设置都会在所有阶段之间无缝移动相移，并产生一些带有正弦波和三角波的“旁路”声音。您将偏移量设置得越接近 1（从 0.5 开始），信号从左扬声器传递到右扬声器的速度就越快。

过滤器接受以下选项：

level\_in

设置输入增益。默认为 1。范围为 [0.015625 - 64]。

拉平

设置输出增益。默认为 1。范围为 [0.015625 - 64]。

模式

设置 LFO 将使用的波形形状。可以是以下之一：正弦波、三角形、正方形、锯齿波或锯齿波。默认为正弦。

数量

设置调制。定义受低频振荡器影响的原始信号量。

偏移量\_l

设置左声道偏移。默认值为 0。允许的范围是 [0 - 1]。

偏移量\_r

设置右声道偏移。默认值为 0.5。允许的范围是 [0 - 1]。

宽度

设置脉冲宽度。默认值为 1。允许的范围为 [0 - 2]。

定时

设置可能的计时模式。可以是以下之一：bpm、ms 或 hz。默认为赫兹。

节律

设置节律。默认值为 120。允许的范围为 [30 - 300]。仅在计时设置为 bpm 时使用。

多发性硬化症

设置女士默认值为 500。允许的范围为 [10 - 2000]。仅在计时设置为毫秒时使用。

赫兹

以 Hz 为单位设置频率。默认值为 2。允许的范围为 [0.01 - 100]。仅在计时设置为 hz 时使用。

### 36.45 样品

使用 libswresample 库将输入音频重新采样为指定参数。如果未指定，则过滤器将自动在其输入和输出之间进行转换。

此过滤器还能够拉伸/压缩音频数据以使其与时间戳匹配或注入静音/剪切音频以使其与时间戳匹配，执行两者的组合或两者都不执行。

过滤器接受语法 [ sample\_rate :] resampler\_options ，其中sample\_rate表示采样率， resampler\_options是key = value对的列表，由“:”分隔。有关受支持选项的完整列表，请参阅[ffmpeg-resampler(1) 手册中的(ffmpeg -resampler)“Resampler Options”部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-resampler.html#Resampler-Options)

* [例子](#Examples-55)

#### 36.45.1 例子

* 将输入音频重新采样为 44100Hz：
* 样品=44100
* 将样本拉伸/压缩到给定的时间戳，每秒补偿最多 1000 个样本：
* 采样=异步=1000

### 36.46 反向

反转音频剪辑。

警告：此滤镜需要内存来缓冲整个剪辑，因此建议进行修剪。

* [例子](#Examples-56)

#### 36.46.1 例子

* 取一个片段的前 5 秒，然后反转它。
* atrim=end=5,反转

### 36.47 恩恩

使用递归神经网络减少语音中的噪音。

此过滤器接受以下选项：

模型，米

设置要加载的火车模型文件。此选项始终是必需的。

混合

设置将过滤后的样本混合到最终输出中的程度。允许的范围是从 -1 到 1。默认值为 1。负值是特殊的，它们设置在最终滤波器输出中保留多少过滤噪声。将此选项设置为 -1 可听到从输入信号中移除的实际噪声。

* [命令](#Commands-20)

#### 36.47.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.48 asdr

测量音频信号失真比。

该过滤器采用两个音频流作为输入，并输出第一个音频流。结果以任一输入端每个通道的 dB 为单位。

### 36.49 个样本

设置每个输出音频帧的样本数。

最后一个输出数据包可能包含不同数量的样本，因为当输入音频发出结束信号时，过滤器将清除所有剩余的样本。

过滤器接受以下选项：

nb\_out\_samples, n

设置每个输出音频帧的帧数。该数字旨在作为每个通道的样本数。默认值为 1024。

垫，p

如果设置为 1，过滤器将用零填充最后一个音频帧，这样最后一帧将包含与前一帧相同数量的样本。默认值为 1。

例如，要将每帧样本数设置为 1234 并禁用最后一帧的填充，请使用：

asetnsamples=n=1234:p=0

### 36.50 asetrate

在不改变 PCM 数据的情况下设置采样率。这将导致速度和音高的变化。

过滤器接受以下选项：

采样率，r

设置输出采样率。默认值为 44100 赫兹。

### 36.51 显示信息

显示包含每个输入音频帧的各种信息的一行。输入音频未被修改。

显示的行包含一系列形式为 key : value的键/值对。

输出中显示以下值：

n

输入帧的（连续）编号，从 0 开始。

积分

输入帧的呈现时间戳，以时基为单位；时基取决于滤波器输入垫，通常为 1/ sample\_rate 。

pts\_time

输入帧的呈现时间戳，以秒为单位。

位置

帧在输入流中的位置，如果此信息不可用和/或无意义（例如在合成音频的情况下），则为 -1

调速器

样本格式。

布局

渠道布局。

速度

音频帧的采样率。

nb\_样本

帧中的样本数（每个通道）。

校验和

音频数据的 Adler-32 校验和（以十六进制打印）。对于平面音频，数据被视为所有平面都被串联起来。

平面校验和

每个数据平面的 Adler-32 校验和列表。

### 36.52 软剪辑

应用音频软剪辑。

软削波是一种失真效果，其中信号的幅度沿着平滑曲线饱和，而不是硬削波的突然形状。

此过滤器接受以下选项：

类型

设置软剪辑的类型。

它接受以下值：

难的

tanh

晒黑

立方体

exp

藻类

五次

罪

erf

临界点

设置开始裁剪的阈值。默认值为 0dB 或 1。

输出

设置应用于输出的增益。默认值为 0dB 或 1。

参数

设置控制 sigmoid 函数的附加参数。

过采样

设置过采样因子。

* [命令](#Commands-21)

#### 36.52.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.53 方面统计

显示有关音频通道的频域统计信息。计算统计数据并将其存储为每个音频通道和每个音频帧的元数据。

它接受以下选项：

win\_size

以样本为单位设置窗口长度。默认值为 2048。允许的范围是从 32 到 65536。

win\_func

设置窗函数。

它接受以下值：

'正确'

'巴特利特'

'汉，汉宁'

'海明'

'黑人'

'韦尔奇'

'平顶'

'巴里斯'

' bnuttall '

'巴恩'

'正弦'

'坚果'

'兰佐斯'

'高斯'

'图基'

'海豚'

'柯西'

' parzen '

'泊松'

'波曼'

'凯撒'

默认为hann 。

重叠

设置窗口重叠。允许的范围是从0到1 。默认值为0.5 。

措施

选择测量的参数。元数据键可以用作标志，默认值是衡量一切的一切。 none禁用所有测量。

每个元数据键的列表如下：

意思是

方差

质心

传播

偏度

峰度

熵

平坦度

波峰

通量

坡

减少

滚下

### 36.54 阿斯里

自动语音识别

此过滤器使用 PocketSphinx 进行语音识别。要启用此过滤器的编译，您需要使用--enable-pocketsphinx配置 FFmpeg 。

它接受以下选项：

速度

设置输入音频的采样率。默认值为16000 。这需要匹配语音模型，否则会得到很差的结果。

唔

设置包含声学模型文件的字典。

字典

设置发音词典。

我

设置语言模型文件。

控制面板

设置语言模型集。

名称

设置要使用的语言模型。

登录名

设置日志消息的输出。

过滤器将已识别的语音导出为帧元数据lavfi.asr.text 。

### 36.55 统计数据

显示音频通道的时域统计信息。计算并显示每个音频通道的统计数据，并在适用的情况下给出总体数字。

它接受以下选项：

长度

以秒为单位的短窗口长度，用于峰值和谷值 RMS 测量。默认值为0.05 （50 毫秒）。允许的范围是[0 - 10] 。

元数据

设置元数据注入。所有元数据键都以lavfi.astats.X为前缀，其中X是从 1 开始的频道编号或字符串Overall 。默认是禁用的。

每个通道的可用键是： Bit\_depth 波峰因数 DC\_offset 动态范围 熵 平面因子 最大差异 最大等级 平均差 最小差异 最低等级 本底噪声 噪音楼层数 Number\_of\_Infs Number\_of\_NaNs Number\_of\_denormals 峰值计数 Peak\_level RMS\_difference RMS\_peak RMS\_trough 过零 Zero\_crossings\_rate

总体而言： Bit\_depth DC\_offset 熵 平面因子 最大差异 最大等级 平均差 最小差异 最低等级 本底噪声 噪音楼层数 Number\_of\_Infs Number\_of\_NaNs Number\_of\_denormals Number\_of\_samples 样本数 峰值计数 Peak\_level RMS\_difference RMS\_level RMS\_peak RMS\_trough

例如，完整的密钥看起来像lavfi.astats.1.DC\_offset或lavfi.astats.Overall.Peak\_count 。

请阅读下面的按键说明。

重置

设置在重置之前计算累积统计数据的帧数。默认是禁用的。

measure\_perchannel

选择每个通道测量的参数。元数据键可以用作标志，默认值是衡量一切的一切。 none禁用所有每通道测量。

measure\_overall

选择整体测量的参数。元数据键可以用作标志，默认值是衡量一切的一切。 none禁用所有整体测量。

测量键的描述如下：

没有任何

没有措施

全部

所有措施

位深

音频的整体位深度，即用于每个样本的位数

波峰因数

峰值与 RMS 电平的标准比率（注意：不是以 dB 为单位）

DC\_offset

从零开始的平均振幅位移

动态范围

测量的音频动态范围（以 dB 为单位）

熵

在整个音频中测量的熵，值接近 1.0 的熵通常是针对白噪声测量的

平面因子

Min\_level或Max\_level ）的平坦度（即具有相同值的连续样本）

最大差异

两个连续样本之间的最大差异

最大等级

最大样本水平

平均差

mean difference between two consecutive samples，即两个连续样本之间的每个差异的平均值

最小差异

两个连续样本之间的最小差异

最低等级

最小样本水平

本底噪声

在短窗口内以 dBFS 测量的最小局部峰值

噪音楼层数

本底噪声的次数（不是样本数）

Number\_of\_Infs

具有无限值的样本数

Number\_of\_NaNs

具有 NaN（不是数字）值的样本数

Number\_of\_denormals

具有低于正常值的样本数

Number\_of\_samples 样本数

样品数量

峰值计数

Min\_level或Max\_level的次数（不是样本数）

Peak\_level

以 dBFS 为单位测量的标准峰值电平

RMS\_difference

两个连续样本之间的均方根差

RMS\_level

以 dBFS 为单位测量的标准 RMS 电平

RMS\_peak

RMS\_trough

在短窗口内测得的 RMS 电平的峰值和谷值，以 dBFS 为单位。

过零

波形与零电平轴相交的点数

过零率

过零率和音频样本数

### 36.56 asubboost

提升低音炮频率。

过滤器接受以下选项：

干燥

设置干增益，保留多少原始信号。允许的范围是从 0 到 1。默认值为 1.0。

湿的

设置湿增益，保留多少过滤后的信号。允许的范围是从 0 到 1。默认值为 1.0。

促进

设置最大升压系数。允许的范围是 1 到 12。默认值为 2。

衰变

设置延迟线衰减增益值。允许的范围是从 0 到 1。默认值为 0.0。

反馈

设置延迟线反馈增益值。允许的范围是从 0 到 1。默认值为 0.9。

隔断

以赫兹为单位设置截止频率。允许的范围是 50 到 900。默认值为 100。

坡

设置截止频率的斜率。允许的范围是 0.0001 到 1。默认值为 0.5。

延迟

设置延迟。允许的范围是 1 到 100。默认值为 20。

频道

设置要处理的通道。默认值都是可用的。

* [命令](#Commands-22)

#### 36.56.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.57 次切

削减低音炮频率。

该滤波器允许设置自定义的、比高通滤波器更陡峭的滚降，因此能够更多地衰减阻带中的频率内容。

过滤器接受以下选项：

隔断

以赫兹为单位设置截止频率。允许的范围是 2 到 200。默认值为 20。

命令

设置过滤器顺序。可用值从 3 到 20。默认值为 10。

等级

设置输入增益电平。允许的范围是从 0 到 1。默认值为 1。

* [命令](#Commands-23)

#### 36.57.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.58 超切

切断超频率。

过滤器接受以下选项：

隔断

以赫兹为单位设置截止频率。允许的范围是 20000 到 192000。默认值为 20000。

命令

设置过滤器顺序。可用值从 3 到 20。默认值为 10。

等级

设置输入增益电平。允许的范围是从 0 到 1。默认值为 1。

* [命令](#Commands-24)

#### 36.58.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.59 超级通行证

应用高阶巴特沃斯带通滤波器。

过滤器接受以下选项：

centerf

以赫兹为单位设置中心频率。允许的范围是 2 到 999999。默认值为 1000。

命令

设置过滤器顺序。可用值从 4 到 20。默认值为 4。

因子

设置 Q 因子。允许的范围是 0.01 到 100。默认值为 1。

等级

设置输入增益电平。允许的范围是从 0 到 2。默认值为 1。

* [命令](#Commands-25)

#### 36.59.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.60 超级站

应用高阶巴特沃斯带阻滤波器。

过滤器接受以下选项：

centerf

以赫兹为单位设置中心频率。允许的范围是 2 到 999999。默认值为 1000。

命令

设置过滤器顺序。可用值从 4 到 20。默认值为 4。

因子

设置 Q 因子。允许的范围是 0.01 到 100。默认值为 1。

等级

设置输入增益电平。允许的范围是从 0 到 2。默认值为 1。

* [命令](#Commands-26)

#### 36.60.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.61 节奏

调整音频速度。

过滤器只接受一个参数，即音频速度。如果未指定，则过滤器将采用标称 1.0 速度。 Tempo 必须在 [0.5, 100.0] 范围内。

请注意，速度大于 2 将跳过一些样本而不是将它们混合。如果出于任何原因这是一个问题，则始终可以菊花链连接多个 atempo 实例以实现所需的产品速度。

* [例子](#Examples-57)
* [命令](#Commands-27)

#### 36.61.1 例子

* 将音频减慢到 80% 的速度：
* 速度=0.8
* 将音频加速到 300% 的速度：
* 节奏=3
* 通过菊花链式连接两个 atempo 实例将音频加速到 300%：
* atempo=sqrt(3)，atempo=sqrt(3)

#### 36.61.2 命令

该过滤器支持以下命令：

速度

更改过滤器速度比例因子。命令的语法是：“ tempo ”

### 36.62 倾斜度

将频谱倾斜滤波器应用于音频流。

该滤波器在任何指定频带上应用任何频谱滚降斜率。

过滤器接受以下选项：

频率

以 Hz 为单位设置倾斜的中心频率。默认值为 10000 赫兹。

坡

设置倾斜的倾斜方向。默认值为 0。允许的范围是从 -1 到 1。

宽度

设置倾斜宽度。默认值为 1000。允许的范围为 100 到 10000。

命令

设置倾斜滤波器的顺序。

等级

设置输入音量级别。允许的范围是从 0 到 4。默认值为 1。

* [命令](#Commands-28)

#### 36.62.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.63 阿特里姆

修剪输入，使输出包含输入的一个连续子部分。

它接受以下参数：

开始

要保留的部分开始的时间戳（以秒为单位）。即以时间戳开始的音频样本将是输出中的第一个样本。

结尾

结束的音频样本之前的音频样本将是输出中的最后一个样本。

开始点

start相同，除了此选项以样本而不是秒为单位设置开始时间戳。

结束点

end相同，除了此选项以样本而不是秒为单位设置结束时间戳。

期间

输出的最大持续时间（以秒为单位）。

开始样本

应输出的第一个样本的编号。

结束样本

应该丢弃的第一个样本的数量。

start 、 end和duration表示为持续时间规范；请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)。

请注意，前两组开始/结束选项和持续时间选项查看帧时间戳，而 \_sample 选项仅计算通过过滤器的样本。因此，当时间戳错误、不准确或不从零开始时，start/end\_pts 和 start/end\_sample 将给出不同的结果。另请注意，此过滤器不会修改时间戳。如果您希望输出时间戳从零开始，请在 atrim 过滤器之后插入 asetpts 过滤器。

如果设置了多个开始或结束选项，则此过滤器会尝试贪婪并保留至少与指定约束之一匹配的所有样本。要一次仅保留与所有约束匹配的部分，请链接多个 atrim 过滤器。

默认值是保留所有输入。因此可以设置例如仅结束值以在指定时间之前保留所有内容。

例子：

* 删除除第二分钟输入之外的所有内容：
* ffmpeg -i 输入 -af atrim=60:120
* 只保留前 1000 个样本：
* ffmpeg -i 输入 -af atrim=end\_sample=1000

### 36.64 轴相关

计算两个输入音频流之间的归一化窗口互相关。

结果样本始终介于 -1 和 1 之间（含）。如果结果为 1，则表示两个输入样本在该选定段中高度相关。结果 0 表示它们根本不相关。如果结果为 -1，则意味着两个输入样本异相，这意味着它们相互抵消。

过滤器接受以下选项：

尺寸

设置计算互相关的段的大小。默认值为 256。允许的范围为 2 到 131072。

算法

设置互相关算法。可慢可快。默认为慢。快速算法假定任何给定段的平均值始终为零，因此需要进行的计算要少得多。这通常是不正确的，但对典型的音频流有效。

* [例子](#Examples-58)

#### 36.64.1 例子

* 计算立体声音频流中通道之间的相关性：
* ffmpeg -i stereo.wav -af 通道分割，axcorrelate=size=1024:algo=fast correlation.wav

### 36.65 带通

frequency和（3dB 点）带宽 width 的双极巴特沃斯带通滤波器。 csg选项选择恒定裙边增益（峰值增益 = Q）而不是默认值：恒定 0dB 峰值增益。滤波器以每倍频程 6dB（每十倍频程 20dB）的速度衰减。

过滤器接受以下选项：

频率，f

设置滤波器的中心频率。默认值为3000 。

南玻

如果设置为 1，则恒定裙边增益。默认为 0。

width\_type, t

设置指定滤波器带宽的方法。

H

赫兹

q

Q因子

o

八度

秒

坡

k

千赫

宽度，w

以 width\_type 单位指定过滤器的带宽。

混合，米

在输出中使用多少过滤信号。默认值为 1。范围介于 0 和 1 之间。

频道，c

指定要过滤的通道，默认情况下会过滤所有可用的通道。

归一化，n

Normalize biquad coefficients，默认情况下是禁用的。启用它会将 DC 的幅度响应归一化为 0dB。

变换，一个

设置 IIR 滤波器的变换类型。

滴

迪

时差

tdii

拉特

svf

zdf

精度，r

设置过滤精度。

汽车

根据环绕滤波器选择自动样本格式。

s16

始终使用带符号的 16 位。

s32

始终使用带符号的 32 位。

f32

始终使用 32 位浮点数。

f64

始终使用 64 位浮点数。

块大小，b

设置用于反向 IIR 处理的块大小。如果此值设置为足够高的值（高于接近零值时截断的脉冲响应长度），则滤波将变为线性相位，否则如果不够大，只会产生令人讨厌的伪像。

请注意，当设置为非零值时，滤波器延迟将恰好是这么多样本。

* [命令](#Commands-29)

#### 36.65.1 命令

该过滤器支持以下命令：

频率，f

改变带通频率。命令的语法是：“频率”

width\_type, t

更改带通 width\_type。命令的语法是：“ width\_type ”

宽度，w

改变带通宽度。命令的语法是：“ width ”

混合，米

改变带通混合。命令的语法是：“ mix ”

### 36.66 带阻

frequency和（3dB 点）带宽 width 的双极巴特沃斯带阻滤波器。滤波器以每倍频程 6dB（每十倍频程 20dB）的速度衰减。

过滤器接受以下选项：

频率，f

设置滤波器的中心频率。默认值为3000 。

width\_type, t

设置指定滤波器带宽的方法。

H

赫兹

q

Q因子

o

八度

秒

坡

k

千赫

宽度，w

以 width\_type 单位指定过滤器的带宽。

混合，米

在输出中使用多少过滤信号。默认值为 1。范围介于 0 和 1 之间。

频道，c

指定要过滤的通道，默认情况下会过滤所有可用的通道。

归一化，n

Normalize biquad coefficients，默认情况下是禁用的。启用它会将 DC 的幅度响应归一化为 0dB。

变换，一个

设置 IIR 滤波器的变换类型。

滴

迪

时差

tdii

拉特

svf

zdf

精度，r

设置过滤精度。

汽车

根据环绕滤波器选择自动样本格式。

s16

始终使用带符号的 16 位。

s32

始终使用带符号的 32 位。

f32

始终使用 32 位浮点数。

f64

始终使用 64 位浮点数。

块大小，b

设置用于反向 IIR 处理的块大小。如果此值设置为足够高的值（高于接近零值时截断的脉冲响应长度），则滤波将变为线性相位，否则如果不够大，只会产生令人讨厌的伪像。

请注意，当设置为非零值时，滤波器延迟将恰好是这么多样本。

* [命令](#Commands-30)

#### 36.66.1 命令

该过滤器支持以下命令：

频率，f

改变带阻频率。命令的语法是：“频率”

width\_type, t

更改 bandreject width\_type。命令的语法是：“ width\_type ”

宽度，w

改变带阻宽度。命令的语法是：“ width ”

混合，米

改变带阻混音。命令的语法是：“ mix ”

### 36.67 低音，低架

使用响应类似于标准高保真音响音调控制的双极搁架滤波器来提升或削减音频的低音（较低）频率。这也称为搁置均衡 (EQ)。

过滤器接受以下选项：

增益，克

给出 0 Hz 时的增益。它的有用范围约为 -20（大幅削减）到 +20（大幅提升）。使用正增益时要注意削波。

频率，f

设置滤波器的中心频率等可用于扩展或降低要提升或削减的频率范围。默认值为100赫兹。

width\_type, t

设置指定滤波器带宽的方法。

H

赫兹

q

Q因子

o

八度

秒

坡

k

千赫

宽度，w

确定过滤器的架子过渡有多陡。

极点，p

设置极数。默认值为 2。

混合，米

在输出中使用多少过滤信号。默认值为 1。范围介于 0 和 1 之间。

频道，c

指定要过滤的通道，默认情况下会过滤所有可用的通道。

归一化，n

Normalize biquad coefficients，默认情况下是禁用的。启用它会将 DC 的幅度响应归一化为 0dB。

变换，一个

设置 IIR 滤波器的变换类型。

滴

迪

时差

tdii

拉特

svf

zdf

精度，r

设置过滤精度。

汽车

根据环绕滤波器选择自动样本格式。

s16

始终使用带符号的 16 位。

s32

始终使用带符号的 32 位。

f32

始终使用 32 位浮点数。

f64

始终使用 64 位浮点数。

块大小，b

设置用于反向 IIR 处理的块大小。如果此值设置为足够高的值（高于接近零值时截断的脉冲响应长度），则滤波将变为线性相位，否则如果不够大，只会产生令人讨厌的伪像。

请注意，当设置为非零值时，滤波器延迟将恰好是这么多样本。

* [命令](#Commands-31)

#### 36.67.1 命令

该过滤器支持以下命令：

频率，f

改变低音频率。命令的语法是：“频率”

width\_type, t

更改低音 width\_type。命令的语法是：“ width\_type ”

宽度，w

更改低音宽度。命令的语法是：“ width ”

增益，克

改变低音增益。命令的语法是：“ gain ”

混合，米

更改低音混音。命令的语法是：“ mix ”

### 36.68 双二阶

应用具有给定系数的双二阶 IIR 滤波器。其中b0 、 b1 、 b2和a0 、 a1 、 a2分别是分子和分母系数。和channels ， c指定要过滤的通道，默认情况下过滤所有可用的通道。

* [命令](#Commands-32)

#### 36.68.1 命令

该过滤器支持以下命令：

a0

a1

a2

b0

b1

b2

更改双二阶参数。命令的语法是：“值”

混合，米

在输出中使用多少过滤信号。默认值为 1。范围介于 0 和 1 之间。

频道，c

指定要过滤的通道，默认情况下会过滤所有可用的通道。

归一化，n

Normalize biquad coefficients，默认情况下是禁用的。启用它会将 DC 的幅度响应归一化为 0dB。

变换，一个

设置 IIR 滤波器的变换类型。

滴

迪

时差

tdii

拉特

svf

zdf

精度，r

设置过滤精度。

汽车

根据环绕滤波器选择自动样本格式。

s16

始终使用带符号的 16 位。

s32

始终使用带符号的 32 位。

f32

始终使用 32 位浮点数。

f64

始终使用 64 位浮点数。

块大小，b

设置用于反向 IIR 处理的块大小。如果此值设置为足够高的值（高于接近零值时截断的脉冲响应长度），则滤波将变为线性相位，否则如果不够大，只会产生令人讨厌的伪像。

请注意，当设置为非零值时，滤波器延迟将恰好是这么多样本。

### 36.69 bs2b

Bauer 立体声到双耳转换，改善立体声音频记录的耳机聆听。

--enable-libbs2b配置 FFmpeg 。

它接受以下参数：

轮廓

预定义的交叉馈送级别。

默认

默认级别（fcut=700，feed=50）。

月

Chu Moy 电路（fcut=700，feed=60）。

杰梅尔

Jan Meier 电路（fcut=650，feed=95）。

切割

截止频率（以 Hz 为单位）。

喂养

进给水平（以 Hz 为单位）。

### 36.70 频道地图

将输入通道重新映射到新位置。

它接受以下参数：

地图

将通道从输入映射到输出。参数是一个以“|”分隔的映射列表，每个映射都是in\_channel - out\_channel或in\_channel形式。 in\_channel可以是输入通道的名称（例如 FL 表示左前）或其在输入通道布局中的索引。 out\_channel是输出通道的名称或其在输出通道布局中的索引。如果没有给出out\_channel则它是一个隐含的索引，从零开始并为每个映射增加 1。

频道布局

输出流的通道布局。

如果不存在映射，过滤器将隐式地将输入通道映射到输出通道，并保留索引。

* [例子](#Examples-59)

#### 36.70.1 例子

* 例如，假设一个 5.1+ 缩混输入 MOV 文件，
* ffmpeg -i in.mov -filter 'channelmap=map=DL-FL|DR-FR' 输出.wav

将从输入的缩混通道创建一个标记为立体声的输出 WAV 文件。

* 修复在 AAC 的本机通道顺序中编码不当的 5.1 WAV
* ffmpeg -i in.wav -filter 'channelmap=1|2|0|5|3|4:5.1' 输出.wav

### 36.71 频道分割

将输入音频流中的每个通道拆分为单独的输出流。

它接受以下参数：

频道布局

输入流的通道布局。默认为“立体声”。

频道

一种通道布局，描述要提取为单独的输出流的通道或“全部”以将每个输入通道提取为单独的流。默认值为“全部”。

选择输入通道布局中不存在的通道将导致错误。

* [例子](#Examples-60)

#### 36.71.1 例子

* 例如，假设一个立体声输入 MP3 文件，
* ffmpeg -i in.mp3 -filter\_complex channelsplit out.mkv

将创建一个包含两个音频流的输出 Matroska 文件，一个仅包含左声道，另一个包含右声道。

* 将 5.1 WAV 文件拆分为每个通道文件：
* ffmpeg -i in.wav -filter\_complex
* 'channelsplit=channel\_layout=5.1[FL][FR][FC][LFE][SL][SR]'
* -map '[FL]' front\_left.wav -map '[FR]' front\_right.wav -map '[FC]'
* front\_center.wav -map '[LFE]' lfe.wav -map '[SL]' side\_left.wav -map '[SR]'
* side\_right.wav
* 从 5.1 WAV 文件中仅提取 LFE：
* ffmpeg -i in.wav -filter\_complex 'channelsplit=channel\_layout=5.1:channels=LFE[LFE]'
* -map '[LFE]' lfe.wav

### 36.72合唱

为音频添加合唱效果。

可以发出像合唱一样的单声，但也可以应用于乐器。

合唱类似于具有短延迟的回声效果，但回声的延迟是恒定的，而合唱则使用正弦或三角调制进行变化。调制深度定义调制延迟在延迟之前或之后播放的范围。因此，延迟的声音听起来会更慢或更快，这是围绕原始声音调整的延迟声音，就像在合唱中一些人声稍微走调一样。

它接受以下参数：

增益

设置输入增益。默认值为 0.4。

输出增益

设置输出增益。默认值为 0.4。

延误

设置延迟。典型的延迟约为 40 毫秒至 60 毫秒。

衰变

设置衰减。

速度

设置速度。

深处

设置深度。

* [例子](#Examples-61)

#### 36.72.1 例子

* 单个延迟：
* 合唱=0.7:0.9:55:0.4:0.25:2
* 两个延迟：
* 合唱=0.6:0.9:50|60:0.4|0.32:0.25|0.4:2|1.3
* 具有三个延迟的更饱满的合唱：
* 合唱=0.5:0.9:50|60|40:0.4|0.32|0.3:0.25|0.4|0.3:2|2.3|1.3

### 36.73压缩比

压缩或扩展音频的动态范围。

它接受以下参数：

攻击

衰变

每个通道的时间列表（以秒为单位），对输入信号的瞬时电平进行平均以确定其音量。 attacks是指音量的增加， decays是指音量的减少。对于大多数情况，起音时间（对音频变大的响应）应短于衰减时间，因为人耳对突然响亮的音频比对突然柔和的音频更敏感。起音的典型值为 0.3 秒，衰减的典型值为 0.8 秒。如果指定的起音和衰减次数低于通道数，则最后设置的起音/衰减将用于所有剩余通道。

积分

传递函数的点列表，以相对于最大可能信号幅度的 dB 为单位指定。每个关键点列表必须使用以下语法定义： x0/y0|x1/y1|x2/y2|....或x0/y0 x1/y1 x2/y2 ....

输入值必须严格递增，但传递函数不必单调上升。假定点0/0但可能会被覆盖（通过0/out-dBn ）。传递函数的典型值为-70/-70|-60/-20|1/0 。

软膝

为所有关节设置以 dB 为单位的曲线半径。它默认为 0.01。

获得

以 dB 为单位设置要应用于传递函数上所有点的附加增益。这允许轻松调整总增益。它默认为 0。

体积

设置初始音量，以 dB 为单位，在过滤开始时为每个通道假定。这允许用户最初提供标称电平，因此，例如，在压缩扩展开始运行之前，不会将非常大的增益应用于初始信号电平。最初安静的音频的典型值为 -90 dB。它默认为 0。

延迟

设置延迟，以秒为单位。输入音频会立即被分析，但音频在被送到音量调节器之前会被延迟。指定大约等于启动/衰减时间的延迟允许滤波器有效地在预测模式而不是反应模式下运行。它默认为 0。

* [例子](#Examples-62)

#### 36.73.1 例子

* 制作适合在嘈杂环境中聆听的小声和大声段落的音乐：
* compand=.3|.3:1|1:-90/-60|-60/-40|-40/-30|-20/-20:6:0:-90:0.2

另一个带有耳语和爆炸部分的音频示例：

compand=0|0:1|1:-90/-900|-70/-70|-30/-9|0/-3:6:0:0:0

* 当噪声低于信号时的噪声门：
* compand=.1|.1:.2|.2:-900/-900|-50.1/-900|-50/-50:.01:0:-90:.1
* 这是另一个噪声门，这次是针对噪声高于信号的电平（在某些方面类似于静噪）：
* compand=.1|.1:.1|.1:-45.1/-45.1|-45/-900|0/-900:.01:45:-90:.1
* 2:1 压缩从 -6dB 开始：
* compand=分=-80/-80|-6/-6|0/-3.8|20/3.5
* 2:1 压缩从 -9dB 开始：
* compand=分=-80/-80|-9/-9|0/-5.3|20/2.9
* 2:1 压缩从 -12dB 开始：
* compand=点数=-80/-80|-12/-12|0/-6.8|20/1.9
* 2:1 压缩从 -18dB 开始：
* compand=分=-80/-80|-18/-18|0/-9.8|20/0.7
* 3:1 压缩从 -15dB 开始：
* compand=点数=-80/-80|-15/-15|0/-10.8|20/-5.2
* 压缩器/门：
* compand=点数=-80/-105|-62/-80|-15.4/-15.4|0/-12|20/-7.6
* 扩展器：
* compand=attacks=0:points=-80/-169|-54/-80|-49.5/-64.6|-41.1/-41.1|-25.8/-15|-10.8/-4.5|0/0|20/ 8.3
* -6dB 的硬限幅器：
* compand=attacks=0:points=-80/-80|-6/-6|20/-6
* -12dB 的硬限幅器：
* compand=攻击=0:points=-80/-80|-12/-12|20/-12
* -35 dB 的硬噪声门：
* compand=attacks=0:points=-80/-115|-35.1/-80|-35/-35|20/20
* 软限制器：
* compand=attacks=0:points=-80/-80|-12.4/-12.4|-6/-8|0/-6.8|20/-2.8

### 36.74补偿延迟

补偿延迟线是一种基于度量的延迟，用于补偿麦克风或扬声器的不同位置。

例如，您使用放置在不同位置的两个麦克风录制了吉他。由于声波前端在正常情况下具有固定速度，因此麦克风的相位可能会有所不同，具体取决于它们的位置和插入方式。当这些麦克风同相（同步）时，可以实现最佳混音。请注意，麦克风之间约 30 厘米的距离会使一个麦克风捕获与另一个麦克风反相的信号。这使得最终混音听起来喜怒无常。该滤波器通过向每个麦克风轨道添加不同的延迟并使它们同步来帮助解决相位问题。

当您以一首曲目为基础并将其他曲目与它一一同步时，可以达到最佳效果。请记住，同步/延迟容限也取决于采样率。更高的采样率将提供更大的容差。

过滤器接受以下参数：

毫米

设置毫米距离。这是微调的补偿距离。默认为 0。

厘米

设置厘米距离。这是用于拧紧距离设置的补偿距离。默认为 0。

米

设置米距离。这是硬距离设置的补偿距离。默认为 0。

干燥

设置干燥量。未处理（干）信号量。默认为 0。

湿的

设定湿量。处理的（湿）信号量。默认值为 1。

温度

以摄氏度为单位设置温度。这是环境的温度。默认值为 20。

* [命令](#Commands-33)

#### 36.74.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.75 交叉馈电

应用耳机交叉馈送过滤器。

交叉馈送是混合立体声录音的左右声道的过程。它主要用于减少低频的极端立体声分离。

目的是为听众产生更多像扬声器一样的声音。

过滤器接受以下选项：

力量

设置交叉进给的强度。默认值为 0.2。允许的范围是从 0 到 1。这为立体图像的侧面部分设置了低架滤波器的增益。默认为 -6dB。当强度设置为 1 时，允许的最大值为 -30db。

范围

设置声场宽度。默认值为 0.5。允许的范围是从 0 到 1。这设置了低架滤波器的截止频率。默认在 1550 Hz 附近被切断。范围设置为 1 时，截止频率设置为 2100 Hz。

坡

设置低架过滤器的曲线斜率。默认值为 0.5。允许的范围是从 0.01 到 1。

level\_in

设置输入增益。默认值为 0.9。

拉平

设置输出增益。默认值为 1。

块大小

设置用于反向 IIR 处理的块大小。如果此值设置为足够高的值（高于接近零值时截断的脉冲响应长度），则滤波将变为线性相位，否则如果不够大，只会产生令人讨厌的伪像。

请注意，当设置为非零值时，滤波器延迟将恰好是这么多样本。

* [命令](#Commands-34)

#### 36.75.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.76结晶器

音频噪声锐化的简单算法。

此过滤器线性增加每个音频样本之间的差异。

过滤器接受以下选项：

我

设置效果强度（默认值：2.0）。必须在 -10.0 到 0（未改变的声音）到 10.0（最大效果）之间的范围内。反向过滤使用负值。

C

启用剪裁。默认情况下启用。

* [命令](#Commands-35)

#### 36.76.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.77 直流偏移

对音频应用直流偏移。

这对于从音频中删除 DC 偏移（可能由录音链中的硬件问题引起）很有用。 DC 偏移的影响是减小了动态余量，从而减小了音量。 [astats](#astats)过滤器可用于确定信号是否具有直流偏移。

转移

设置DC shift，允许范围为[-1, 1]。它表示移动音频的量。

限制器增益

选修的。它的值应该远小于 1（例如 0.05 或 0.02），用于防止削波。

### 36.78 去嘶声

对音频样本应用 de-essing。

我

设置触发 de-essing 的强度。允许的范围是从 0 到 1。默认值为 0。

米

设置声音高音部分的闪避量。允许的范围是从 0 到 1。默认值为 0.5。

F

de-essing 时要保留多少原始频率内容。允许的范围是从 0 到 1。默认值为 0.5。

秒

设置输出模式。

它接受以下值：

我

传递输入不变。

o

pass ess过滤掉了。

电子

仅通过 ess。

默认值为o 。

### 36.79 对话增强

增强立体声音频中的对话。

此滤波器接受立体声输入并产生环绕声 (3.0) 声道输出。新制作的前置中置声道增强了原本在两个立体声声道中可用的语音对话。此滤波器输出与立体声输入中可用的相同的左前和右前声道。

过滤器接受以下选项：

原来的

设置原始中心因子以保持在前置中心通道输出。允许的范围是从 0 到 1。默认值为 1。

提高

将对话增强因子设置为置于前置中置声道输出。允许的范围是从 0 到 3。默认值为 1。

嗓音

设置语音检测系数。允许的范围是 2 到 32。默认值为 2。

* [命令](#Commands-36)

#### 36.79.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.80 drmeter

测量音频动态范围。

在非常动态的材料中发现 14 或更高的 DR 值。在过渡材料中发现 DR 为 8 至 13。任何少于 8 的都具有非常差的动态并且非常压缩。

过滤器接受以下选项：

长度

以秒为单位设置窗口长度，用于将音频分割成等长的片段。默认为 3 秒。

### 36.81 dynaudnorm

动态音频规范器。

该滤波器对输入音频应用一定量的增益，以使其峰值幅度达到目标水平（例如 0 dBFS）。然而，与更“简单”的归一化算法相比，动态音频归一化器\*动态\*重新调整输入音频的增益因子。这允许对音频的“安静”部分应用额外增益，同时避免失真或削波“响亮”部分。换句话说：动态音频规范器将“均衡”安静和响亮部分的音量，也就是说每个部分的音量都达到相同的目标水平。但是请注意，动态音频规范器\*无需\*应用“动态范围压缩”即可实现此目标。它将在音频文件的每个部分\*内保留 100% 的动态范围。

框架，f

以毫秒为单位设置帧长度。范围从 10 到 8000 毫秒。默认值为 500 毫秒。动态音频规范器以小块（称为帧）的形式处理输入音频。这是必需的，因为峰值幅度对于单个样本值没有意义。相反，我们需要确定连续样本值序列的峰值幅度。 “标准”规范器会简单地使用整个文件的峰值幅度，而动态音频规范器会单独确定每个帧的峰值幅度。帧的长度以毫秒为单位指定。默认情况下，Dynamic Audio Normalizer 使用 500 毫秒的帧长，已发现这对大多数文件都能产生良好的效果。请注意，将根据单个输入音频文件的采样率自动确定确切的帧长度（以采样数表示）。

高斯尺寸，g

设置高斯滤波器窗口大小。在 3 到 301 之间，必须是奇数。默认值为 31。动态音频规范器最重要的参数可能是高斯平滑滤波器的窗口大小。过滤器的窗口大小以帧为单位指定，以当前帧为中心。为了简单起见，这必须是奇数。因此，默认值 31 考虑了当前帧以及前面的 15 个帧和后面的 15 个帧。使用更大的窗口会导致更强的平滑效果，从而导致更小的增益变化，即更慢的增益自适应。相反，使用较小的窗口会导致较弱的平滑效果，从而导致更多的增益变化，即更快的增益自适应。换句话说，你增加这个值越多，Dynamic Audio Normalizer 就越像一个“传统的”规范化过滤器。相反，您将此值减小得越多，动态音频规范器的行为就越像动态范围压缩器。

峰，磷

设置目标峰值。这指定了标准化音频输入的最高允许幅度级别。该滤波器将尝试尽可能接近目标峰值幅度，但同时它也确保归一化信号永远不会超过峰值幅度。帧的最大局部增益因子直接由目标峰值幅度强加。默认值为 0.95，因此留有 5%\* 的余量。不建议超过此值。

最大增益，米

设置最大增益系数。范围从 1.0 到 100.0。默认值为 10.0。动态音频规范器确定每个输入帧的最大可能（本地）增益因子，即不会导致削波或失真的最大增益因子。最大增益因子由帧的最高幅度样本决定。然而，动态音频规范器额外地通过预定的（全局）最大增益因子来限制帧的最大增益因子。这样做是为了避免在“静音”或几乎静音的帧中出现过多的增益因子。默认情况下，最大增益因子为 10.0，对于大多数输入，默认值应该足够了，通常不建议增加该值。但是，对于整体音量极低的输入，可能需要允许更高的增益系数。但是请注意，动态音频规范器并不简单地应用“硬”阈值（即截止值高于阈值）。相反，将应用“sigmoid”阈值函数。这样，增益因子将平滑地接近阈值，但绝不会超过该值。

目标有效值，r

设置目标 RMS。范围从 0.0 到 1.0。默认值为 0.0 - 禁用。默认情况下，动态音频规范器执行“峰值”规范化。这意味着每个帧的最大局部增益因子（仅）由帧的最高幅度样本定义。这样，可以在不超过最大信号电平的情况下（即不削波）尽可能多地放大样本。然而，可选地，动态音频规范器也可以考虑帧的均方根，缩写为 RMS。在电气工程中，RMS 通常用于确定时变信号的功率。因此，RMS 被认为是“感知响度”的更好近似值，而不是仅查看信号的峰值幅度。因此，通过将所有帧调整为恒定的 RMS 值，可以建立统一的“感知响度”。如果指定了目标 RMS 值，则帧的本地增益因子被定义为可准确产生该 RMS 值的因子。但是请注意，最大局部增益因子仍受帧最高幅度样本的限制，以防止削波。

联轴器，n

启用通道耦合。默认情况下启用。默认情况下，Dynamic Audio Normalizer 会将所有通道放大相同的量。这意味着相同的增益系数将应用于所有通道，即最大可能的增益系数由“最响亮”的通道决定。然而，在某些录音中，可能会出现不同通道的音量不均匀的情况，例如，一个通道可能比另一个通道“更安静”。在这种情况下，此选项可用于禁用通道耦合。这样，增益因子将为每个通道独立确定，仅取决于单个通道的最高幅度样本。这允许协调不同通道的音量。

正确的dc, c

启用直流偏置校正。默认情况下是禁用的。音频信号（在时域中）是一系列样本值。在动态音频规范器中，无论原始输入格式如何，这些样本值都在 -1.0 到 1.0 范围内表示。通常，音频信号或“波形”应以零点为中心。这意味着如果我们计算一个文件或单个帧中所有样本的平均值，那么结果应该是 0.0 或至少非常接近该值。但是，如果平均值在正方向或负方向上与 0.0 存在显着偏差，则这称为 DC 偏置或 DC 偏移。由于 DC 偏置显然是不可取的，因此 Dynamic Audio Normalizer 提供了可选的 DC 偏置校正。启用 DC 偏差校正后，动态音频规范器将确定每个输入帧的平均值或“DC 校正”偏移量，并从所有帧的样本值中减去该值，以确保这些样本再次以 0.0 为中心。此外，为了避免帧边界处的“间隙”，DC校正偏移值将在相邻帧之间平滑地内插。

alt边界，b

启用替代边界模式。默认情况下是禁用的。 Dynamic Audio Normalizer 考虑了每个帧周围的特定邻域。这包括前面的帧和后面的帧。然而，对于位于音频文件最开始和最后的“边界”帧，并非所有相邻帧都可用。特别是，对于音频文件中的前几帧，前面的帧是未知的。并且，类似地，对于音频文件中的最后几帧，后续帧是未知的。因此，出现了应该为“边界”区域中的丢失帧假设哪些增益因子的问题。 Dynamic Audio Normalizer 实现了两种模式来处理这种情况。默认边界模式假定缺失帧的增益因子恰好为 1.0，从而分别在输入的开始和结束处产生平滑的“淡入”和“淡出”。

压缩，小号

设置压缩因子。范围从 0.0 到 30.0。默认值为 0.0。默认情况下，动态音频规范器不应用“传统”压缩。这意味着信号峰值不会被修剪，因此完整的动态范围将保留在每个局部邻域内。但是，在某些情况下，可能需要将 Dynamic Audio Normalizer 的规范化算法与更“传统”的压缩相结合。为此，动态音频规范器提供了一个可选的压缩（阈值）功能。如果（且仅当）启用压缩功能，所有输入帧将在实际归一化过程之前由软拐点阈值函数处理。简而言之，阈值函数将修剪所有幅度超过某个阈值的样本。然而，Dynamic Audio Normalizer 并不简单地应用一个固定的阈值。相反，将为每个单独的帧调整阈值。通常，较小的参数会导致更强的压缩，反之亦然。不建议低于 3.0 的值，因为可能会出现听觉失真。

阈值，吨

设置目标阈值。这指定了将被归一化的音频输入的最低允许幅度级别。如果输入帧音量高于此值帧将被归一化。否则帧可能根本无法归一化。默认值设置为 0，这意味着所有输入帧都将被归一化。如果不想放大数字噪声，此选项最有用。

渠道，小时

指定要过滤的通道，默认情况下过滤所有可用通道。

重叠，o

指定帧的重叠。如果设置为 0（默认值），则不进行帧重叠。使用 >0 和 <1 值将使增益调整不那么保守，例如当 framelen 选项设置为较小的值时，如果 framelen 选项值针对非零重叠进行补偿，则与零重叠情况相比，增益调整将在时间上更加平滑。

曲线，v

指定在计算应用于帧的增益时将使用的峰值映射曲线表达式。最大输出帧增益仍将受到前面提到的此过滤器的其他选项的限制。

表达式可以包含以下常量：

通道

当前频道号

锡

当前样品编号

nb\_频道

通道数

吨

以秒表示的时间戳

高级

采样率

p

当前帧峰值

* [命令](#Commands-37)

#### 36.81.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.82 耳垢

使音频更容易在耳机上收听。

此滤波器将“提示”添加到 44.1kHz 立体声（即音频 CD 格式）音频，以便在使用耳机收听时，立体声图像从您的头部（耳机标准）移动到外部和听者前方（扬声器标准） ).

从 SoX 移植。

### 36.83均衡器

应用双极峰值均衡 (EQ) 滤波器。使用此滤波器，可以增加或减少选定频率及其周围的信号电平，同时（与带通和带阻滤波器不同）所有其他频率的信号电平保持不变。

为了产生复杂的均衡曲线，可以多次给出该滤波器，每次具有不同的中心频率。

过滤器接受以下选项：

频率，f

以 Hz 为单位设置滤波器的中心频率。

width\_type, t

设置指定滤波器带宽的方法。

H

赫兹

q

Q因子

o

八度

秒

坡

k

千赫

宽度，w

以 width\_type 单位指定过滤器的带宽。

增益，克

以 dB 为单位设置所需的增益或衰减。使用正增益时要注意削波。

混合，米

在输出中使用多少过滤信号。默认值为 1。范围介于 0 和 1 之间。

频道，c

指定要过滤的通道，默认情况下会过滤所有可用的通道。

归一化，n

Normalize biquad coefficients，默认情况下是禁用的。启用它会将 DC 的幅度响应归一化为 0dB。

变换，一个

设置 IIR 滤波器的变换类型。

滴

迪

时差

tdii

拉特

svf

zdf

精度，r

设置过滤精度。

汽车

根据环绕滤波器选择自动样本格式。

s16

始终使用带符号的 16 位。

s32

始终使用带符号的 32 位。

f32

始终使用 32 位浮点数。

f64

始终使用 64 位浮点数。

块大小，b

设置用于反向 IIR 处理的块大小。如果此值设置为足够高的值（高于接近零值时截断的脉冲响应长度），则滤波将变为线性相位，否则如果不够大，只会产生令人讨厌的伪像。

请注意，当设置为非零值时，滤波器延迟将恰好是这么多样本。

* [例子](#Examples-63)
* [命令](#Commands-38)

#### 36.83.1 例子

* 在 1000 Hz 时衰减 10 dB，带宽为 200 Hz：
* 均衡器=f=1000:t=h:width=200:g=-10
* 使用 Q 1 在 1000 Hz 时应用 2 dB 增益并使用 Q 2 在 100 Hz 时衰减 5 dB：
* 均衡器=f=1000:t=q:w=1:g=2,均衡器=f=100:t=q:w=2:g=-5

#### 36.83.2 命令

该过滤器支持以下命令：

频率，f

改变均衡器频率。命令的语法是：“频率”

width\_type, t

更改均衡器 width\_type。命令的语法是：“ width\_type ”

宽度，w

更改均衡器宽度。命令的语法是：“ width ”

增益，克

改变均衡器增益。命令的语法是：“ gain ”

混合，米

更改均衡器组合。命令的语法是：“ mix ”

### 36.84 超立体声

线性增加左右声道之间的差异，为播放添加某种“现场”效果。

过滤器接受以下选项：

米

设置差异系数（默认值：2.5）。 0.0 表示单声道声音（两个声道的平均值），1.0 声音将不变，-1.0 左右声道将交换。

C

启用剪裁。默认情况下启用。

* [命令](#Commands-39)

#### 36.84.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.85 防火等级

使用任意频率响应应用 FIR 均衡。

过滤器接受以下选项：

获得

设置增益曲线方程（以 dB 为单位）。表达式可以包含变量：

F

评估的频率

高级

采样率

通道

通道号，禁用多通道评估时设置为 0

孩子

通道 ID，请参阅 libavutil/channel\_layout.h，在禁用多通道评估时设置为第一个通道 ID

通道

通道数

布局

channel\_layout，参见 libavutil/channel\_layout.h

和功能：

增益插值（f）

基于 gain\_entry 在频率 f 上插入增益

cubic\_interpolate(f)

与 gain\_interpolate 相同，但更平滑

此选项也可用作命令。默认为gain\_interpolate(f) 。

获得进入

为 gain\_interpolate 函数设置增益条目。表达式可以包含函数：

条目（f，g）

以 g 值存储频率 f 处的增益条目

此选项也可用作命令。

延迟

以秒为单位设置过滤器延迟。更高的值意味着更准确。默认值为0.01 。

准确性

以 Hz 为单位设置滤波器精度。较低的值意味着更准确。默认值为5 。

函数

设置窗函数。可接受的值是：

矩形的

矩形窗口，当增益曲线已经平滑时有用

汉

汉恩窗（默认）

海明

海明窗

黑人

布莱克曼窗

坚果3

3 项连续一阶导数 nuttall 窗口

mnuttall3

最小 3 项不连续 nuttall 窗口

坚果高

4 项连续一阶导数 nuttall 窗口

高大

最小 4 项不连续 nuttall (blackman-nuttall) 窗口

巴里斯

布莱克曼-哈里斯窗

图基

窗口

固定的

如果启用，则使用固定数量的音频样本。这提高了大延迟过滤时的速度。默认是禁用的。

多

启用多通道增益评估。默认是禁用的。

零相

通过减去时间戳来补偿延迟来启用零相位模式。默认是禁用的。

规模

设置增益使用的比例。可接受的值是：

琳琳

线性频率，线性增益

日志

线性频率，对数（以 dB 为单位）增益（默认）

登录

对数（以倍频程为单位，其中 20 Hz 为 0）频率、线性增益

日志日志

对数频率，对数增益

转储文件

设置转储文件，适用于 gnuplot。

垃圾秤

设置转储文件的比例。可接受的值与比例选项相同。默认是 linlog。

fft2

使用复数 FFT 启用 2 通道卷积。这显着提高了速度。默认是禁用的。

最小相位

启用最小相位脉冲响应。默认是禁用的。

* [例子](#Examples-64)

#### 36.85.1 例子

* 1000 Hz 时的低通：
* firequalizer=gain='if(lt(f,1000), 0, -INF)'
* 具有 gain\_entry 的 1000 Hz 低通：
* firequalizer=gain\_entry='entry(1000,0);条目（1001，-INF）'
* 自定义均衡：
* firequalizer=gain\_entry='条目（100,0）;条目（400，-4）；条目（1000，-6）；条目（2000，0）'
* 具有零相位的更高延迟以补偿延迟：
* firequalizer=delay=0.1:fixed=on:zero\_phase=on
* 左声道低通，右声道高通：
* firequalizer=gain='if(eq(chid,1),gain\_interpolate(f),if(eq(chid,2),gain\_interpolate(1e6+f),0))'
* :gain\_entry='条目(1000, 0);入口（1001，-INF）；入口(1e6+1000,0)':multi=on

### 36.86镶边

对音频应用镶边效果。

过滤器接受以下选项：

延迟

以毫秒为单位设置基本延迟。范围从 0 到 30。默认值为 0。

深度

以毫秒为单位设置增加的扫描延迟。范围从 0 到 10。默认值为 2。

再生

设置百分比再生（延迟信号反馈）。范围从 -95 到 95。默认值为 0。

宽度

设置延迟信号与原始信号混合的百分比。范围从 0 到 100。默认值为 71。

速度

设置每秒扫描次数 (Hz)。范围从 0.1 到 10。默认值为 0.5。

形状

设置扫频波形，可以是三角波或正弦波。默认值为正弦曲线。

阶段

为多通道设置扫描波百分比偏移。范围从 0 到 100。默认值为 25。

插话

设置延迟线插值，线性或二次。默认是线性的。

### 36.87哈斯

将 Haas 效应应用于音频。

请注意，这最适合应用于单声道信号。将此滤波器应用于单声道信号后，它会提供一定的方向性并拉伸其立体声图像。

过滤器接受以下选项：

level\_in

设置输入电平。默认为1或 0dB

拉平

设置输出电平。默认为1或 0dB。

边增益

设置应用于信号侧面部分的增益。默认为1 。

中间源

设置一种中间源。可以是以下之一：

'离开'

选择左声道。

'对'

选择正确的渠道。

'中'

拾取立体图像的中间部分信号。

'边'

拾取立体图像的边部分信号。

中间阶段

改变中间阶段。默认情况下是禁用的。

左延迟

设置左声道延迟。默认为2.05毫秒。

左平衡

设置左声道平衡。默认为-1 。

左增益

设置左声道增益。默认为1 。

左相

改变左相位。默认情况下是禁用的。

right\_delay

设置右声道延迟。默认为2.12毫秒。

正确的平衡

设置右声道平衡。默认为1 。

权利增益

设置右声道增益。默认为1 。

右相

改变正确的相位。默认情况下启用。

### 36.88 硬盘

解码高清兼容数字 (HDCD) 数据。带有嵌入式 HDCD 代码的 16 位 PCM 流被扩展为 20 位 PCM 流。

该滤波器支持 HDCD 的峰值扩展和低电平增益调整功能，并检测瞬态滤波器标志。

ffmpeg -i HDCD16.flac -af hdcd OUT24.flac

将过滤器与 wav 一起使用时，请注意 wav 的默认编码是 16 位，因此生成的 20 位流将被截断回 16 位。在过滤器之后使用类似-acodec pcm\_s24le的东西来获得 24 位 PCM 输出。

ffmpeg -i HDCD16.wav -af hdcd OUT16.wav

ffmpeg -i HDCD16.wav -af hdcd -c:a pcm\_s24le OUT24.wav

过滤器接受以下选项：

禁用自动转换

禁用过滤器图中的任何自动格式转换或重采样。

过程立体声

一起处理立体声通道。如果 target\_gain 在通道之间不匹配，则认为它无效并使用最后一个有效的 target\_gain。

cdt\_ms

以毫秒为单位设置代码检测定时器周期。

force\_pe

即使未发出 PE 信号，也始终将峰值扩展到 -3dBFS 以上。

分析模式

用纯音替换音频并调整幅度以表示解码过程的某些特定方面。输出文件可以与原始文件一起加载到音频编辑器中以帮助分析。

analyze\_mode=pe:force\_pe=true可以用来查看PE级别以上的所有样本。

模式有：

' 0，关闭'

残疾人

' 1, lle '

每个样本的增益调整级别

' 2, 体育'

峰延伸发生的样品

' 3, cdt '

代码检测计时器处于活动状态的示例

' 4, tgm '

目标增益在通道之间不匹配的样本

### 36.89耳机

应用头部相关传输函数 (HRTF) 在用户周围创建虚拟扬声器，通过耳机进行双耳聆听。 HRIR 通过附加流提供，每个通道都需要一个立体声输入流。

过滤器接受以下选项：

地图

设置卷积输入流的映射。该参数是一个以“|”分隔的通道名称列表，因为它们是作为过滤器的附加流输入给出的。这也指定输入流的数量。输入流的数量必须不少于第一个流中的通道数加一个。

获得

设置应用于音频的增益。值以 dB 为单位。默认为 0。

类型

设置处理类型。可以是时间或频率。 time正在处理时域中的音频，速度很慢。 freq正在快速处理频域中的音频。默认是频率。

生活

为 LFE 通道设置自定义增益。值以 dB 为单位。默认为 0。

尺寸

将帧的大小设置为一次处理的样本数。默认值为1024 。允许的范围是 1024 到 96000。

小时

设置 hrir 流的格式。默认值为立体声。替代值是multich 。如果值设置为stereo ，则附加流的数量应大于或等于第一个输入流中的输入通道数。此外，每个附加流都应具有立体声通道数。如果值设置为multich ，附加流的数量应该正好是一个。此外，附加流的输入通道数应等于或大于第一个输入流的通道数的两倍。

* [例子](#Examples-65)

#### 36.89.1 例子

* 完整示例使用 wav 文件作为 7.1 缩混的电影过滤器的系数，每个电影过滤器使用带有 IR 系数的立体声文件作为输入。这些文件给出了虚拟扬声器每个位置的系数：
* ffmpeg -i 输入.wav
* -filter\_complex "amovie=azi\_270\_ele\_0\_DFC.wav[sr];amovie=azi\_90\_ele\_0\_DFC.wav[sl];amovie=azi\_225\_ele\_0\_DFC.wav[br];amovie=azi\_135\_ele\_0\_DFC.wav[bl];amovie=azi\_0\_ele\_0\_DFC.wav,asplit[fc ][ lfe];amovie=azi\_35\_ele\_0\_DFC.wav[fl];amovie=azi\_325\_ele\_0\_DFC.wav[fr];[0:a][fl][fr][fc][lfe][bl][br][sl][sr]耳机=FL|FR|FC|LFE|BL|BR|SL|SR"
* 输出.wav
* 完整示例使用 wav 文件作为系数，带有用于 7.1 缩混的电影滤波器，但现在是多声道 hr格式。
* ffmpeg -i input.wav -filter\_complex "amovie=minp.wav[hrirs];[0:a][hrirs]headphone=map=FL|FR|FC|LFE|BL|BR|SL|SR:hrir=multich"
* 输出.wav

### 36.90 高通

应用具有 3dB 点频率的高通滤波器。滤波器可以是单极或双极（默认）。滤波器每倍频程每极点 6dB（每极点每十倍频程 20dB）滚降。

过滤器接受以下选项：

频率，f

以 Hz 为单位设置频率。默认值为 3000。

极点，p

设置极数。默认值为 2。

width\_type, t

设置指定滤波器带宽的方法。

H

赫兹

q

Q因子

o

八度

秒

坡

k

千赫

宽度，w

以 width\_type 单位指定过滤器的带宽。仅适用于双极滤波器。默认值为 0.707q 并给出 Butterworth 响应。

混合，米

在输出中使用多少过滤信号。默认值为 1。范围介于 0 和 1 之间。

频道，c

指定要过滤的通道，默认情况下会过滤所有可用的通道。

归一化，n

Normalize biquad coefficients，默认情况下是禁用的。启用它会将 DC 的幅度响应归一化为 0dB。

变换，一个

设置 IIR 滤波器的变换类型。

滴

迪

时差

tdii

拉特

svf

zdf

精度，r

设置过滤精度。

汽车

根据环绕滤波器选择自动样本格式。

s16

始终使用带符号的 16 位。

s32

始终使用带符号的 32 位。

f32

始终使用 32 位浮点数。

f64

始终使用 64 位浮点数。

块大小，b

设置用于反向 IIR 处理的块大小。如果此值设置为足够高的值（高于接近零值时截断的脉冲响应长度），则滤波将变为线性相位，否则如果不够大，只会产生令人讨厌的伪像。

请注意，当设置为非零值时，滤波器延迟将恰好是这么多样本。

* [命令](#Commands-40)

#### 36.90.1 命令

该过滤器支持以下命令：

频率，f

改变高通频率。命令的语法是：“频率”

width\_type, t

更改高通 width\_type。命令的语法是：“ width\_type ”

宽度，w

更改高通宽度。命令的语法是：“ width ”

混合，米

更改高通混音。命令的语法是：“ mix ”

### 36.91加入

将多个输入流合并为一个多通道流。

它接受以下参数：

输入

输入流的数量。它默认为 2。

频道布局

所需的输出通道布局。它默认为立体声。

地图

将通道从输入映射到输出。参数是一个以“|”分隔的映射列表，每个都在 input\_idx中。 in\_channel - out\_channel形式。 input\_idx是输入流的从 0 开始的索引。 in\_channel可以是输入通道的名称（例如左前的 FL）或其在指定输入流中的索引。 out\_channel是输出通道的名称。

当未明确指定时，过滤器将尝试猜测映射。它首先尝试找到一个未使用的匹配输入通道，如果失败，它会选择第一个未使用的输入通道。

加入 3 个输入（正确设置通道布局）：

ffmpeg -i INPUT1 -i INPUT2 -i INPUT3 -filter\_complex join=inputs=3 输出

从 6 个单声道流构建 5.1 输出：

ffmpeg -i fl -i fr -i fc -i sl -i sr -i lfe -filter\_complex

'join=inputs=6:channel\_layout=5.1:map=0.0-FL|1.0-FR|2.0-FC|3.0-SL|4.0-SR|5.0-LFE'

出去

### 36.92 拉兹帕

加载 LADSPA（Linux 音频开发人员的简单插件 API）插件。

--enable-ladspa配置 FFmpeg 。

文件，f

指定要加载的 LADSPA 插件库的名称。如果定义了环境变量LADSPA\_PATH ，则在LADSPA\_PATH中以冒号分隔列表指定的每个目录中搜索 LADSPA 插件，否则在标准 LADSPA 路径中搜索，顺序为： HOME/.ladspa/lib/ , / usr/local/lib/ladspa/ , /usr/lib/ladspa/ 。

插件，p

指定库中的插件。一些库只包含一个插件，但其他库包含许多插件。如果未设置，过滤器将列出指定库中的所有可用插件。

控制，c

设置'|'分隔的控件列表，它们是零个或多个浮点值，用于确定加载插件的行为（例如延迟、阈值或增益）。需要使用以下语法定义控件：c0= value0 |c1= value1 |c2= value2 |...，其中valuei是在第 i个控件上设置的值。或者，也可以使用以下语法定义它们： value0 |值 1 | value2 |...，其中valuei是在第 i个控件上设置的值。如果controls设置为help ，则打印所有可用控件及其有效范围。

采样率，s

指定采样率，默认为 44100。仅在插件具有零输入时使用。

nb\_样本，n

设置每个输出帧每个通道的样本数，默认值为 1024。仅在插件具有零输入时使用。

持续时间，d

设置源音频的最短持续时间。有关可接受的语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)请注意，生成的持续时间可能大于指定的持续时间，因为生成的音频总是在完整帧的末尾被剪切。如果未指定，或者表示的持续时间为负数，则音频应该永远生成。仅在插件具有零输入时使用。

潜伏期，升

启用延迟补偿，默认情况下是禁用的。仅在插件有输入时使用。

* [例子](#Examples-66)
* [命令](#Commands-41)

#### 36.92.1 例子

* 列出 amp（LADSPA 示例插件）库中的所有可用插件：
* ladspa=文件=放大器
* VCF库中vcf\_notch插件的所有可用控件及其有效范围：
* ladspa=f=vcf:p=vcf\_notch:c=帮助
* Computer Music Toolkit (CMT) 插件库模拟低质量音频设备：
* ladspa=文件=cmt:插件=lofi:控制=c0=22|c1=12|c2=12
* 使用 TAP 插件（Tom 的音频处理插件）为音频添加混响：
* ladspa=文件=tap\_reverb:tap\_reverb
* 生成具有 0.2 振幅的白噪声：
* ladspa=file=cmt:noise\_source\_white:c=c0=.2
* 使用插件C\* Click -来自C\* 音频插件套件(CAPS) 库的节拍器生成 20 bpm 节拍：
* ladspa=file=caps:点击:c=c1=20'
* 应用C\* Eq10X2 - 立体声 10 段均衡器效果：
* ladspa=帽:Eq10X2:c=c0=-48|c9=-24|c3=12|c4=2
* SWH 插件集合的快速前瞻限制器将音量增加 20dB ：
* ladspa=fast\_lookahead\_limiter\_1913:fastLookaheadLimiter:20|0|2
* SWH 插件集合中的多频带均衡器衰减低频：
* ladspa=mbeq\_1197:mbeq:-24|-24|-24|0|0|0|0|0|0|0|0|0|0|0|0
* C\* 音频插件套件(CAPS) 库中的Narrower减少立体声图像：
* ladspa=caps:更窄
* 另一个白噪声，现在使用C\* Audio Plugin Suite (CAPS) 库：
* ladspa=帽子：白色：.2
* 一些分形噪声，使用C\* 音频插件套件(CAPS) 库：
* ladspa=上限：分形：c=c1=1
* VLevel插件的动态体积标准化：
* ladspa=vlevel-ladspa:vlevel\_mono

#### 36.92.2 命令

该过滤器支持以下命令：

CN

修改第 N个控制值。

如果指定的值无效，则忽略它并保留先前的值。

### 36.93 响度

EBU R128 响度归一化。包括动态和线性归一化模式。支持单通道（直播、文件）和双通道（文件）模式。该算法可以针对 IL、LRA 和最大真实峰值。在动态模式下，为了准确检测真正的峰值，音频流将被上采样到 192 kHz。使用-ar选项或aresample过滤器明确设置输出采样率。

过滤器接受以下选项：

我，我

设置综合响度目标。范围是 -70.0 - -5.0。默认值为 -24.0。

上帝抵抗军，上帝抵抗军

设置响度范围目标。范围是 1.0 - 50.0。默认值为 7.0。

TP, TP

设置最大真实峰值。范围是 -9.0 - +0.0。默认值为 -2.0。

测量\_I，测量\_i

输入文件的测量 IL。范围是 -99.0 - +0.0。

测量\_LRA，测量\_lra

输入文件的测量 LRA。范围是 0.0 - 99.0。

实测TP, 实测TP

输入文件的测量真实峰值。范围是 -99.0 - +99.0。

测量阈值

输入文件的测量阈值。范围是 -99.0 - +0.0。

抵消

设置偏移增益。增益在真实峰值限制器之前应用。范围是 -99.0 - +99.0。默认为 +0.0。

线性的

通过线性缩放源音频进行归一化。 measured\_I 、 measured\_LRA 、 measured\_TP和measured\_thresh都必须指定。目标 LRA 不应低于源 LRA，并且综合响度的变化不应导致超过目标 TP 的真实峰值。如果不满足这些条件中的任何一个，规范化模式将恢复为动态。选项为true或false 。默认为true 。

双单声道

将单声道输入文件视为“双单声道”。如果单声道文件旨在在立体声系统上播放，则其 EBU R128 测量值将在感知上不正确。如果设置为true ，此选项将补偿此效果。多通道输入文件不受此选项的影响。选项为真或假。默认为假。

打印格式

设置统计信息的打印格式。选项有摘要、json 或无。默认值为无。

### 36.94 低通

应用具有 3dB 点频率的低通滤波器。滤波器可以是单极或双极（默认）。滤波器每倍频程每极点 6dB（每极点每十倍频程 20dB）滚降。

过滤器接受以下选项：

频率，f

以 Hz 为单位设置频率。默认值为 500。

极点，p

设置极数。默认值为 2。

width\_type, t

设置指定滤波器带宽的方法。

H

赫兹

q

Q因子

o

八度

秒

坡

k

千赫

宽度，w

以 width\_type 单位指定过滤器的带宽。仅适用于双极滤波器。默认值为 0.707q 并给出 Butterworth 响应。

混合，米

在输出中使用多少过滤信号。默认值为 1。范围介于 0 和 1 之间。

频道，c

指定要过滤的通道，默认情况下会过滤所有可用的通道。

归一化，n

Normalize biquad coefficients，默认情况下是禁用的。启用它会将 DC 的幅度响应归一化为 0dB。

变换，一个

设置 IIR 滤波器的变换类型。

滴

迪

时差

tdii

拉特

svf

zdf

精度，r

设置过滤精度。

汽车

根据环绕滤波器选择自动样本格式。

s16

始终使用带符号的 16 位。

s32

始终使用带符号的 32 位。

f32

始终使用 32 位浮点数。

f64

始终使用 64 位浮点数。

块大小，b

设置用于反向 IIR 处理的块大小。如果此值设置为足够高的值（高于接近零值时截断的脉冲响应长度），则滤波将变为线性相位，否则如果不够大，只会产生令人讨厌的伪像。

请注意，当设置为非零值时，滤波器延迟将恰好是这么多样本。

* [例子](#Examples-67)
* [命令](#Commands-42)

#### 36.94.1 例子

* Lowpass 只有 LFE 通道，它不存在 LFE 它什么都不做：
* 低通=c=LFE

#### 36.94.2 命令

该过滤器支持以下命令：

频率，f

改变低通频率。命令的语法是：“频率”

width\_type, t

更改低通 width\_type。命令的语法是：“ width\_type ”

宽度，w

更改低通宽度。命令的语法是：“ width ”

混合，米

更改低通混音。命令的语法是：“ mix ”

### 36.95 等级 2

加载 LV2（LADSPA 版本 2）插件。

--enable-lv2配置 FFmpeg 。

插件，p

指定插件 URI。您可能需要转义“：”。

控制，c

设置'|'分隔的控件列表，它们是零个或多个浮点值，用于确定加载插件的行为（例如延迟、阈值或增益）。如果controls设置为help ，则打印所有可用控件及其有效范围。

采样率，s

指定采样率，默认为 44100。仅在插件具有零输入时使用。

nb\_样本，n

设置每个输出帧每个通道的样本数，默认值为 1024。仅在插件具有零输入时使用。

持续时间，d

设置源音频的最短持续时间。有关可接受的语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)请注意，生成的持续时间可能大于指定的持续时间，因为生成的音频总是在完整帧的末尾被剪切。如果未指定，或者表示的持续时间为负数，则音频应该永远生成。仅在插件具有零输入时使用。

* [例子](#Examples-68)
* [命令](#Commands-43)

#### 36.95.1 例子

* 使用 Calf 的低音增强器插件：
* lv2=p=http\\\\://calf.sourceforge.net/plugins/BassEnhancer:c=amount=2
* 使用 Calf 的 vinyl 插件：
* lv2=p=http\\\\://calf.sourceforge.net/plugins/Vinyl:c=drone=0.2|aging=0.5
* 使用 ArtyFX 的 bit crusher 插件：
* lv2=p=http\\\\://www.openavproductions.com/artyfx#bitta:c=crush=0.3

#### 36.95.2 命令

此过滤器支持插件作为命令导出的所有选项。

### 36.96 mcompand

Multiband 压缩或扩展音频的动态范围。

使用 4 阶 Linkwitz-Riley IIR 将输入音频分成多个频段。这类似于扬声器的交叉，并且在没有压扩器作用时导致平坦的频率响应。

它接受以下参数：

参数

此选项语法为：attack,decay,[attack,decay..] 软拐点 crossover\_frequency [delay [initial\_volume [gain]]] | attack,decay ... 有关每个项目的解释，请参阅 compand 滤波器文档。

### 36.97 平底锅

混合具有特定增益水平的通道。过滤器接受输出通道布局，后跟一组通道定义。

此过滤器还旨在有效地重新映射音频流的通道。

过滤器接受以下形式的参数：“ l | outdef | outdef |...”

升

输出通道布局或通道数

outdef

输出通道规范，形式为：“ out\_name =[ gain \*] in\_name [(+-)[ gain \*] in\_name ...]”

out\_name

要定义的输出通道，通道名称（FL、FR 等）或通道编号（c0、c1 等）

获得

通道的乘法系数，1 保持音量不变

in\_name

要使用的输入通道，详见 out\_name；不能混合命名和编号的输入通道

如果通道规范中的“=”替换为“<”，则该规范的增益将被重新归一化，以便总和为 1，从而避免削波噪声。

* [混合示例](#Mixing-examples)
* [重映射示例](#Remapping-examples)

#### 36.97.1 混合示例

例如，如果您想从立体声缩混到单声道，但左声道的系数更大：

pan=1c|c0=0.9\*c0+0.1\*c1

自定义缩混至立体声，自动适用于 3、4、5 和 7 声道环绕声：

声像=立体声| FL < FL + 0.5\*FC + 0.6\*BL + 0.6\*SL | FR < FR + 0.5\*FC + 0.6\*BR + 0.6\*SR

请注意， ffmpeg集成了一个默认的下混（和上混）系统，除非您有非常具体的需求，否则应该是首选（请参阅“-ac”选项）。

#### 36.97.2 重映射示例

当且仅当：

* 增益系数为零或一，
* 每个通道输出只有一个输入，

如果满足所有这些条件，过滤器将通知用户（“检测到纯通道映射”），并使用优化的无损方法进行重新映射。

例如，如果您有一个 5.1 源并且想要通过删除额外通道来获得立体声音频流：

pan="立体声| c0=FL | c1=FR"

给定相同的来源，您还可以切换左前和右前声道并保持输入声道布局：

pan="5.1| c0=c1 | c1=c0 | c2=c2 | c3=c3 | c4=c4 | c5=c5"

如果输入是立体声音频流，您可以将左前声道静音（并仍然保持立体声声道布局）：

pan="立体声|c1=c1"

仍然使用立体声音频流输入，您可以在左前和右前复制右声道：

pan="立体声| c0=FR | c1=FR"

### 36.98重播增益

ReplayGain 扫描仪过滤器。该过滤器将音频流作为输入并原封不动地输出。在过滤结束时，它显示track\_gain和track\_peak 。

### 36.99 重新取样

转换音频采样格式、采样率和通道布局。它并不意味着可以直接使用。

### 36.100 橡皮筋

使用 librubberband 应用时间拉伸和音高变换。

--enable-librubberband配置 FFmpeg 。

过滤器接受以下选项：

速度

设置速度比例因子。

沥青

设置音高比例因子。

瞬态

设置瞬变检测器。可能的值是：

脆

混合的

光滑的

探测器

设置检测器。可能的值是：

化合物

打击乐的

柔软的

阶段

设置相位。可能的值是：

层

独立的

窗户

设置处理窗口大小。可能的值是：

标准

短的

长的

平滑

设置平滑。可能的值是：

离开

在

共振峰

移动音调时启用共振峰保留。可能的值是：

转移

保留

音高

设置音调质量。可能的值是：

质量

速度

一致性

频道

设置频道。可能的值是：

分开

一起

* [命令](#Commands-44)

#### 36.100.1 命令

该过滤器支持以下命令：

速度

更改过滤器速度比例因子。命令的语法是：“ tempo ”

沥青

更改滤波器间距比例因子。命令的语法是：“ pitch ”

### 36.101 侧链压缩

该滤波器的作用类似于普通压缩器，但能够使用第二个输入信号压缩检测到的信号。它需要两个输入流并返回一个输出流。第一个输入流将根据第二个流信号进行处理。然后可以在后期处理阶段用其他滤波器对滤波后的信号进行滤波。请参阅[平移](#pan)和[合并](#amerge)过滤器。

过滤器接受以下选项：

level\_in

设置输入增益。默认值为 1。范围介于 0.015625 和 64 之间。

模式

设置压缩机运行模式。可以向上也可以向下。默认是向下的。

临界点

如果第二个流的信号升高到这个水平以上，它将影响第一个流的增益降低。默认为 0.125。范围介于 0.00097563 和 1 之间。

比率

设置信号降低的比率。 1:2 表示如果电平比阈值提高 4dB，则降低后只会比阈值高 2dB。默认值为 2。范围在 1 到 20 之间。

攻击

在增益降低开始之前信号必须上升到阈值以上的毫秒数。默认值为 20。范围介于 0.01 和 2000 之间。

发布

在再次减少减少之前信号必须低于阈值的毫秒数。默认值为 250。范围介于 0.01 和 9000 之间。

化妆品

通过处理后将放大多少信号来设置数量。默认值为 1。范围从 1 到 64。

膝盖

围绕阈值弯曲尖锐的膝盖以更柔和地进入增益减少。默认值为 2.82843。范围在 1 到 8 之间。

关联

选择侧链流所有通道之间的平均电平或侧链流的较大（最大）通道是否影响降低。默认为平均。

检测

峰值的情况下应采用准确的信号，在rms的情况下应采用 RMS 信号。默认为rms ，主要是更平滑。

level\_sc

设置侧链增益。默认值为 1。范围介于 0.015625 和 64 之间。

混合

在输出中使用多少压缩信号。默认值为 1。范围介于 0 和 1 之间。

* [命令](#Commands-45)
* [例子](#Examples-69)

#### 36.101.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

#### 36.101.2 示例

* 完整的 ffmpeg 示例采用 2 个音频输入，根据第 2 个输入的信号压缩第 1 个输入，然后将压缩信号与第 2 个输入合并：
* ffmpeg -i main.flac -i sidechain.flac -filter\_complex “[1:a]asplit=2[sc][mix];[0:a][sc]sidechaincompress[compr];[compr][mix]amerge”

### 36.102 侧链门

侧链门的作用类似于普通（宽带）门，但能够在将检测到的信号发送到增益降低级之前对其进行过滤。通常，门使用全范围信号来检测高于阈值的电平。例如：如果您从侧链信号中切断所有较低的频率，则仅当出现的高音不足时，门才会降低音轨的音量。使用此技术，您可以减少自然鼓的共鸣或消除严重失真吉他的静音敲击的“隆隆声”。它需要两个输入流并返回一个输出流。第一个输入流将根据第二个流信号进行处理。

过滤器接受以下选项：

level\_in

过滤前设置输入电平。默认值为 1。允许的范围是从 0.015625 到 64。

模式

设置操作模式。可以向上也可以向下。默认是向下的。如果设置为向上模式，信号的较高部分将被放大，向上扩展动态范围。否则，如遇向下信号会减弱。

范围

设置信号低于阈值时增益降低的级别。默认值为 0.06125。允许的范围是从 0 到 1。将此设置为 0 将禁用缩减，然后过滤器的行为类似于扩展器。

临界点

如果信号上升到该水平以上，则释放增益降低。默认值为 0.125。允许的范围是从 0 到 1。

比率

设置信号降低的比率。默认值为 2。允许的范围为 1 到 9000。

攻击

在增益减少停止之前信号必须上升到阈值以上的毫秒数。默认值为 20 毫秒。允许的范围是从 0.01 到 9000。

发布

在再次增加减少量之前，信号必须低于阈值的毫秒数。默认值为 250 毫秒。允许的范围是从 0.01 到 9000。

化妆品

设置处理后信号的放大量。默认值为 1。允许的范围为 1 到 64。

膝盖

围绕阈值弯曲尖锐的膝盖以更柔和地进入增益减少。默认值为 2.828427125。允许的范围是 1 到 8。

检测

选择是否应采用精确信号进行检测或采用 RMS 之类的信号。默认值为有效值。可以是峰值或有效值。

关联

选择是否所有通道之间的平均电平或较大通道影响降低。默认为平均值。可以是平均值或最大值。

level\_sc

设置侧链增益。默认值为 1。范围从 0.015625 到 64。

* [命令](#Commands-46)

#### 36.102.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 36.103 静默检测

检测音频流中的静音。

当此过滤器检测到输入音频音量小于或等于噪声容限值且持续时间大于或等于检测到的最小噪声持续时间时，此过滤器会记录一条消息。

打印的时间和持续时间以秒表示。 lavfi.silence\_start或lavfi.silence\_start.X元数据键设置在时间戳等于或超过检测持续时间的第一帧上，它包含静音第一帧的时间戳。

lavfi.silence\_duration或lavfi.silence\_duration.X和lavfi.silence\_end或lavfi.silence\_end.X元数据键在静音后的第一帧上设置。如果启用单声道，并且单独评估每个通道，则使用.X后缀键， X对应于通道号。

过滤器接受以下选项：

噪声，n

设置噪声容限。可以以 dB 为单位指定（如果“dB”附加到指定值）或振幅比。默认值为 -60dB，或 0.001。

持续时间，d

设置静音持续时间直到通知（默认为 2 秒）。有关可接受的语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)

单声道，米

单独处理每个通道，而不是合并处理。默认情况下是禁用的。

* [例子](#Examples-70)

#### 36.103.1 示例

* 以 -50dB 噪声容限检测 5 秒静默：
* silencedetect=n=-50dB:d=5
* ffmpeg检测静默的完整示例，在silence.mp3中噪声容差为 0.0001 ：
* ffmpeg -i silence.mp3 -af silencedetect=noise=0.0001 -f null -

### 36.104 沉默移除

从音频的开头、中间或结尾删除静音。

过滤器接受以下选项：

开始\_期间

此值用于指示是否应在音频开头修剪音频。零值表示不应从一开始就修剪静音。当指定一个非零值时，它会修剪音频直到找到非静音。通常，当从音频开头修剪静音时，start\_periods将为1 ，但可以将其增加到更高的值以将所有音频修剪到特定数量的非静音周期。默认值为0 。

开始持续时间

指定在停止修剪音频之前必须检测到非静音的时间量。通过增加持续时间，噪音的爆发可以被视为静音并被修剪掉。默认值为0 。

开始阈值

这指示应将哪个样本值视为无声。对于数字音频，值为0可能没问题，但对于从模拟录制的音频，您可能希望增加该值以解决背景噪音。可以以 dB 为单位指定（如果“dB”附加到指定值）或振幅比。默认值为0 。

开始沉默

指定开始时的最大静音持续时间，修剪后将保留该静音时间。默认为 0，等于修剪所有检测为静音的样本。

启动模式

指定在多声道音频开始时检测静音结束的模式。可以是任何一个或全部。默认是任何。对于any ，任何被检测为非静音的样本都会导致静音修剪停止。对于all ，只有当所有通道都被检测为非静音时才会导致停止修剪静音。

停止\_周期

设置从音频末尾修剪静音的计数。要从文件中间移除静默，请指定一个负数的stop\_periods 。该值随后被视为正值，用于指示效果应按照start\_periods指定的方式重新开始处理，使其适用于移除音频中间的静音期。默认值为0 。

停止持续时间

指定在不再复制音频之前必须存在的静音持续时间。通过指定更长的持续时间，可以在音频中保留需要的静音。默认值为0 。

停止阈值

start\_threshold相同，但用于从音频结尾修剪静音。可以以 dB 为单位指定（如果“dB”附加到指定值）或振幅比。默认值为0 。

停止沉默

指定最后的最大静音持续时间，将在修剪后保留。默认为 0，等于修剪所有检测为静音的样本。

停止模式

指定多声道音频结束时静音开始的检测模式。可以是任何一个或全部。默认是任何。对于any ，任何被检测为非静音的样本都会导致静音修剪停止。对于all ，只有当所有通道都被检测为非静音时才会导致停止修剪静音。

检测

设置如何检测静音。可以是有效值或峰值。第二个速度更快，并且与正好为 0 的数字静音效果更好。默认值为rms 。

窗户

设置以秒为单位的持续时间，用于计算用于检测静音的样本数量的窗口大小。默认值为0.02 。允许的范围是从0到10 。

* [例子](#Examples-71)

#### 36.104.1 示例

* 以下示例显示了如何使用此过滤器开始录制，该录制不包含通常发生在按下录制按钮和表演开始之间的开始时的延迟：
* silenceremove=start\_periods=1:start\_duration=5:start\_threshold=0.02
* 在音频中超过 1 秒的静音时，修剪从头到尾遇到的所有静音：
* silenceremove=stop\_periods=-1:stop\_duration=1:stop\_threshold=-90dB
* 使用峰值检测从头到尾修剪所有数字静音样本，其中音频中的数字静音样本超过 0 个，并且在流中相同位置的所有通道中检测到数字静音：
* silenceremove=window=0:detection=peak:stop\_mode=all:start\_mode=all:stop\_periods=-1:stop\_threshold=0

### 36.105 舒缓剂

SOFAlizer 使用头部相关传输函数 (HRTF) 在用户周围创建虚拟扬声器，通过耳机进行双耳聆听（支持多达 9 个声道的音频格式）。 HRTF 存储在 SOFA 文件中（有关数据库，请参阅[http://www.sofacoustics.org/ ）。](http://www.sofacoustics.org/) SOFAlizer 由奥地利科学院声学研究所 (ARI) 开发。

--enable-libmysofa配置 FFmpeg 。

过滤器接受以下选项：

沙发

设置用于渲染的 SOFA 文件。

获得

设置应用于音频的增益。值以 dB 为单位。默认为 0。

回转

以度为单位设置虚拟扬声器的旋转。默认为 0。

海拔

以度为单位设置虚拟扬声器的仰角。默认为 0。

半径

使用近场 HRTF 设置扬声器和听众之间的距离（以米为单位）。默认值为 1。

类型

设置处理类型。可以是时间或频率。 time正在处理时域中的音频，速度很慢。 freq正在快速处理频域中的音频。默认是频率。

喇叭

设置虚拟扬声器的自定义位置。此选项的语法为：<CH> <AZIM> <ELEV>[|<CH> <AZIM> <ELEV>|...]。每个虚拟扬声器都用简短的通道名称进行描述，后跟方位角和仰角（以度为单位）。每个虚拟扬声器描述由“|”分隔。例如，要覆盖左前和右前声道位置，请使用：'speakers=FL 45 15|FR 345 15'。具有无法识别的频道名称的描述将被忽略。

获得生命

为 LFE 通道设置自定义增益。值以 dB 为单位。默认为 0。

帧大小

在样本数中设置自定义帧大小。默认值为 1024。允许的范围是从 1024 到 96000。仅当选项 ' type ' 设置为freq时才使用。

归一化

是否应在导入 SOFA 文件时对所有 IR 进行规范化。默认情况下启用。

插

如果确切位置不匹配，最近的 IR 是否应该与相邻的 IR 进行插值。默认情况下是禁用的。

最小相位

载入 SOFA 文件时对所有 IR 进行最小相位处理。默认情况下是禁用的。

角步

设置邻居搜索角度步长。仅在启用选项插值时使用。

辐射步

设置邻居搜索半径步长。仅在启用选项插值时使用。

* [例子](#Examples-72)

#### 36.105.1 示例

* 使用 ClubFritz6 沙发文件：
* sofalizer=sofa=/path/to/ClubFritz6.sofa:type=freq:radius=1
* 使用 ClubFritz12 沙发锉和较大的半径，旋转较小：
* sofalizer=sofa=/path/to/ClubFritz12.sofa:type=freq:radius=2:rotation=5
* 与上面类似，但具有左前、右前、左后和右后的自定义扬声器位置以及自定义增益：
* “sofalizer=sofa=/path/to/ClubFritz6.sofa:type=freq:radius=2:speakers=FL 45|FR 315|BL 135|BR 225:gain=28”

### 36.106语音规范

语音规范器。

此过滤器根据阈值扩展或压缩音频样本的每个半周期（本地样本集全部高于或全部低于零以及两个最近的零交叉点之间），因此音频在以下选项控制的条件下达到目标峰值。

过滤器接受以下选项：

峰，磷

设置扩展目标峰值。这指定了标准化音频输入的最高允许绝对振幅级别。默认值为 0.95。允许的范围是从 0.0 到 1.0。

扩张，电子

设置最大扩展系数。允许的范围是 1.0 到 50.0。默认值为 2.0。此选项控制样本扩展的最大局部半周期。最大扩展将使得局部峰值达到目标峰值但永远不会超过它并且新峰值与先前峰值之间的比率不超过该选项值。

压缩，c

设置最大压缩因子。允许的范围是 1.0 到 50.0。默认值为 2.0。此选项控制样本压缩的最大本地半周期。仅当阈值选项设置为大于 0.0 的值时才使用此选项，然后在这种情况下，当局部峰值低于或等于阈值设置的值时，属于该峰值半周期的所有样本都将按当前压缩因子进行压缩。

阈值，吨

设置阈值。默认值为 0.0。允许的范围是从 0.0 到 1.0。此选项指定样本的哪些半周期将被压缩，哪些将被扩展。任何局部峰值低于或等于此选项值的半周期样本将按当前压缩因子进行压缩，否则，如果大于阈值，将使用扩展因子进行扩展，使其达到峰值目标值但永远不会超过它。

提高, r

设置样本每半个周期的扩展提升量。默认值为 0.001。允许的范围是从 0.0 到 1.0。这控制了每个新的半周期增加膨胀因子的速度，直到它达到膨胀值。将此选项设置得太高可能会导致失真。

跌倒，f

设置每半个采样周期的压缩提升量。默认值为 0.001。允许的范围是从 0.0 到 1.0。这控制了每个新的半周期压缩因子增加的速度，直到它达到压缩值。

渠道，小时

指定要过滤的通道，默认情况下过滤所有可用通道。

反转，我

开启反向过滤，默认关闭。这颠倒了阈值选项的解释。启用时，局部峰值低于或等于阈值选项的任何半周期样本将被扩展，否则将被压缩。

链接，我

在计算应用于每个过滤通道样本的增益时链接通道，默认情况下处于禁用状态。禁用时，每个过滤通道的增益计算都是独立的，否则当启用此选项时，将使用每个过滤通道的所有可能增益的最小值。

均方根，米

设置扩展目标 RMS 值。这指定了标准化音频输入的最高允许 RMS 电平。默认值为 0.0，因此禁用。允许的范围是从 0.0 到 1.0。

* [命令](#Commands-47)
* [例子](#Examples-73)

#### 36.106.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

#### 36.106.2 示例

* 弱而缓慢的放大：
* speechnorm=e=3:r=0.00001:l=1
* 中慢速放大：
* speechnorm=e=6.25:r=0.00001:l=1
* 强而快的放大：
* speechnorm=e=12.5:r=0.0001:l=1
* 非常强大和快速的放大：
* speechnorm=e=25:r=0.0001:l=1
* 极速放大：
* speechnorm=e=50:r=0.0001:l=1

### 36.107 立体工具

这个过滤器有一些方便的实用程序来管理立体声信号，用于将 M/S 立体声录音转换为 L/R 信号，同时控制参数或传播主音轨的立体声图像。

过滤器接受以下选项：

level\_in

在过滤两个通道之前设置输入电平。默认值为 1。允许的范围是从 0.015625 到 64。

拉平

为两个通道过滤后设置输出电平。默认值为 1。允许的范围是从 0.015625 到 64。

balance\_in

设置两个通道之间的输入平衡。默认值为 0。允许的范围是从 -1 到 1。

平衡

设置两个通道之间的输出平衡。默认值为 0。允许的范围是从 -1 到 1。

软夹

启用软剪辑。产生模拟失真而不是刺耳的数字 0dB 削波。默认情况下禁用。

互助

使左声道静音。默认情况下禁用。

静音

将右声道静音。默认情况下禁用。

相位

改变左声道的相位。默认情况下禁用。

移相器

改变右声道的相位。默认情况下禁用。

模式

设置立体声模式。可用值是：

' lr>lr '

左/右到左/右，这是默认值。

' lr>ms '

左/右到中/侧。

' ms>lr '

中/侧到左/右。

' lr>ll '

左/右到左/左。

' lr>rr '

左/右到右/右。

' lr>l+r '

左/右到左 + 右。

' lr>rl '

左/右到右/左。

' ms>ll '

中/侧到左/左。

' ms>rr '

中/侧到右/右。

' ms>rl '

中/侧到右/左。

' lr>lr '

左/右到左 - 右。

slev

设置侧信号电平。默认值为 1。允许的范围是从 0.015625 到 64。

sbal

设置侧信号的平衡。默认值为 0。允许的范围是从 -1 到 1。

中低层

设置中间信号的电平。默认值为 1。允许的范围是从 0.015625 到 64。

潘

设置中间信号盘。默认值为 0。允许的范围是从 -1 到 1。

根据

在单声道和反向通道之间设置立体声基础。默认值为 0。允许的范围是从 -1 到 1。

延迟

以毫秒为单位设置延迟从右声道向左延迟多少，反之亦然。默认值为 0。允许的范围为 -20 到 20。

等级

设置 S/C 级别。默认值为 1。允许的范围为 1 到 100。

阶段

以度为单位设置立体相位。默认值为 0。允许的范围为 0 到 360。

bmode\_in, bmode\_out

为 balance\_in/balance\_out 选项设置平衡模式。

可以是以下之一：

'平衡'

经典平衡模式。一次衰减一个通道。增益提高到 1。

'振幅'

与上面的经典模式类似，但增益提高到 2。

'权力'

等功率分配，从-6dB 到+6dB 范围。

* [命令](#Commands-48)
* [例子](#Examples-74)

#### 36.107.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

#### 36.107.2 示例

* 应用类似卡拉 OK 的效果：
* 立体工具=mlev=0.015625
* 将 M/S 信号转换为 L/R：
* “stereotools=mode=ms>lr”

### 36.108 立体加宽

该滤波器通过抑制两个通道共有的信号并通过将左信号延迟到右信号来增强立体声效果，反之亦然，从而扩大立体声效果。

过滤器接受以下选项：

延迟

左信号进入右信号的延迟时间（以毫秒为单位），反之亦然。默认值为 20 毫秒。

反馈

右延迟信号的增益量，反之亦然。在右输出中提供左信号的延迟效果，反之亦然，从而产生加宽效果。默认值为 0.3。

交叉馈送

反相从左向右交叉馈电。这有助于抑制单声道。如果该值为 1，它将取消两个通道共有的所有信号。默认值为 0.3。

干混

设置原始通道的输入信号电平。默认值为 0.8。

* [命令](#Commands-49)

#### 36.108.1 命令

delay as [commands](#commands)之外的所有上述选项。

### 36.109 超级均衡器

应用 18 波段均衡器。

过滤器接受以下选项：

1b

设置 65Hz 频带增益。

2b

设置 92Hz 频带增益。

3b

设置 131Hz 频带增益。

4b

设置 185Hz 频带增益。

5b

设置 262Hz 频带增益。

6b

设置 370Hz 频带增益。

7b

设置 523Hz 频带增益。

8b

设置 740Hz 频带增益。

9b

设置 1047Hz 频带增益。

10b

设置 1480Hz 频带增益。

11b

设置 2093Hz 频带增益。

12b

设置 2960Hz 频带增益。

13b

设置 4186Hz 频带增益。

14b

设置 5920Hz 频段增益。

15b

设置 8372Hz 频带增益。

16b

设置 11840Hz 频段增益。

17b

设置 16744Hz 频段增益。

18b

设置 20000Hz 频带增益。

### 36.110环绕声

应用音频环绕上混滤波器。

这个过滤器允许从音频流中产生多声道输出。

过滤器接受以下选项：

chl\_out

设置输出通道布局。默认情况下，这是5.1 。

有关所需语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 通道布局部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#channel-layout-syntax)

chl\_in

设置输入通道布局。默认情况下，这是立体声。

有关所需语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 通道布局部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#channel-layout-syntax)

level\_in

设置输入音量级别。默认情况下，这是1 。

拉平

设置输出音量级别。默认情况下，这是1 。

生活

如果输出通道布局有 LFE 通道输出，则启用它。默认情况下，这是启用的。

生命低

设置 LFE 低截止频率。默认情况下，这是128赫兹。

生命值高

设置 LFE 高截止频率。默认情况下，这是256赫兹。

生活模式

设置 LFE 模式，可以是add或sub 。默认是添加。在添加模式下，LFE 通道从输入音频创建并添加到输出。在子模式下，LFE 通道从输入音频创建并添加到输出，但所有非 LFE 输出通道都减去输出 LFE 通道。

光滑的

设置时间平滑强度，用于在时间上变换立体声时逐渐改变因素。允许的范围是从0.0到1.0 。有助于提高焦点选项值大于0.0的输出质量。默认值为0.0 。只有在此范围内且没有边缘的值才有效。

角度

设置立体声环绕变换的角度，允许的范围是从0到360 。默认值为90 。

重点

设置立体声环绕变换的焦点，允许的范围是从-1到1 。默认为0 。

fc\_in

设置前中心输入音量。默认情况下，这是1 。

fc\_out

设置前中心输出音量。默认情况下，这是1 。

fl\_in

设置左前输入音量。默认情况下，这是1 。

飞出

设置左前输出音量。默认情况下，这是1 。

fr\_in

设置右前输入音量。默认情况下，这是1 。

fr\_out

设置右前输出音量。默认情况下，这是1 。

sl\_in

设置左侧输入音量。默认情况下，这是1 。

sl\_out

设置左侧输出音量。默认情况下，这是1 。

sr\_in

设置右侧输入音量。默认情况下，这是1 。

sr\_out

设置右侧输出音量。默认情况下，这是1 。

bl\_in

设置左输入音量。默认情况下，这是1 。

bl\_out

设置左输出音量。默认情况下，这是1 。

br\_in

设置右输入音量。默认情况下，这是1 。

br\_out

设置右输出音量。默认情况下，这是1 。

bc\_in

设置后中心输入音量。默认情况下，这是1 。

bc\_out

设置后中心输出音量。默认情况下，这是1 。

生活在

设置 LFE 输入音量。默认情况下，这是1 。

生命输出

设置 LFE 输出音量。默认情况下，这是1 。

allx

为所有通道设置立体图像在 X 轴上的传播使用。允许的范围是从-1到15 。默认情况下，此值为负-1 ，因此未使用。

盟友

为所有通道设置立体图像在 Y 轴上的传播使用。允许的范围是从-1到15 。默认情况下，此值为负-1 ，因此未使用。

fcx、flx、frx、blx、brx、slx、srx、bcx

为每个通道设置立体图像在 X 轴上的传播使用。允许的范围是从0.06到15 。默认情况下，此值为0.5 。

fcy、fly、fry、bly、bry、sly、sry、bcy

为每个通道设置立体图像在 Y 轴上的传播使用。允许的范围是从0.06到15 。默认情况下，此值为0.5 。

win\_size

设置窗口大小。允许的范围是从1024到65536 。默认大小为4096 。

win\_func

设置窗函数。

它接受以下值：

'正确'

'巴特利特'

'汉，汉宁'

'海明'

'黑人'

'韦尔奇'

'平顶'

'巴里斯'

' bnuttall '

'巴恩'

'正弦'

'坚果'

'兰佐斯'

'高斯'

'图基'

'海豚'

'柯西'

' parzen '

'泊松'

'波曼'

'凯撒'

默认为hann 。

重叠

设置窗口重叠。如果设置为 1，将选择所选窗口函数的建议重叠。默认值为0.5 。

### 36.111 倾斜架

使用响应类似于标准高保真音调控制的双极搁架滤波器，提升或削减音频的较低频率并削减或提升音频的较高频率。这也称为搁置均衡 (EQ)。

过滤器接受以下选项：

增益，克

给出 0 Hz 时的增益。它的有用范围约为 -20（大幅削减）到 +20（大幅提升）。使用正增益时要注意削波。

频率，f

设置滤波器的中心频率等可用于扩展或降低要提升或削减的频率范围。默认值为3000赫兹。

width\_type, t

设置指定滤波器带宽的方法。

H

赫兹

q

Q因子

o

八度

秒

坡

k

千赫

宽度，w

确定过滤器的架子过渡有多陡。

极点，p

设置极数。默认值为 2。

混合，米

在输出中使用多少过滤信号。默认值为 1。范围介于 0 和 1 之间。

频道，c

指定要过滤的通道，默认情况下会过滤所有可用的通道。

归一化，n

Normalize biquad coefficients，默认情况下是禁用的。启用它会将 DC 的幅度响应归一化为 0dB。

变换，一个

设置 IIR 滤波器的变换类型。

滴

迪

时差

tdii

拉特

svf

zdf

精度，r

设置过滤精度。

汽车

根据环绕滤波器选择自动样本格式。

s16

始终使用带符号的 16 位。

s32

始终使用带符号的 32 位。

f32

始终使用 32 位浮点数。

f64

始终使用 64 位浮点数。

块大小，b

设置用于反向 IIR 处理的块大小。如果此值设置为足够高的值（高于接近零值时截断的脉冲响应长度），则滤波将变为线性相位，否则如果不够大，只会产生令人讨厌的伪像。

请注意，当设置为非零值时，滤波器延迟将恰好是这么多样本。

* [命令](#Commands-50)

#### 36.111.1 命令

此过滤器支持一些选项作为[命令](#commands)。

### 36.112 高音，高架

使用响应类似于标准高保真音响音调控制的双极搁架滤波器来提升或削减音频的高音（上部）频率。这也称为搁置均衡 (EQ)。

过滤器接受以下选项：

增益，克

给出 ~22 kHz 和奈奎斯特频率中较低者的增益。它的有用范围约为 -20（大幅削减）到 +20（大幅提升）。使用正增益时要注意削波。

频率，f

设置滤波器的中心频率等可用于扩展或降低要提升或削减的频率范围。默认值为3000赫兹。

width\_type, t

设置指定滤波器带宽的方法。

H

赫兹

q

Q因子

o

八度

秒

坡

k

千赫

宽度，w

确定过滤器的架子过渡有多陡。

极点，p

设置极数。默认值为 2。

混合，米

在输出中使用多少过滤信号。默认值为 1。范围介于 0 和 1 之间。

频道，c

指定要过滤的通道，默认情况下会过滤所有可用的通道。

归一化，n

Normalize biquad coefficients，默认情况下是禁用的。启用它会将 DC 的幅度响应归一化为 0dB。

变换，一个

设置 IIR 滤波器的变换类型。

滴

迪

时差

tdii

拉特

svf

zdf

精度，r

设置过滤精度。

汽车

根据环绕滤波器选择自动样本格式。

s16

始终使用带符号的 16 位。

s32

始终使用带符号的 32 位。

f32

始终使用 32 位浮点数。

f64

始终使用 64 位浮点数。

块大小，b

设置用于反向 IIR 处理的块大小。如果此值设置为足够高的值（高于接近零值时截断的脉冲响应长度），则滤波将变为线性相位，否则如果不够大，只会产生令人讨厌的伪像。

请注意，当设置为非零值时，滤波器延迟将恰好是这么多样本。

* [命令](#Commands-51)

#### 36.112.1 命令

该过滤器支持以下命令：

频率，f

改变高音频率。命令的语法是：“频率”

width\_type, t

更改高音 width\_type。命令的语法是：“ width\_type ”

宽度，w

改变高音宽度。命令的语法是：“ width ”

增益，克

改变高音增益。命令的语法是：“ gain ”

混合，米

更改高音混音。命令的语法是：“ mix ”

### 36.113颤音

正弦幅度调制。

过滤器接受以下选项：

F

以赫兹为单位的调制频率。次谐波范围（20 Hz 或更低）中的调制频率会产生颤音效果。通过指定高于 20 Hz 的调制频率，该滤波器还可用作环形调制器。范围是 0.1 - 20000.0。默认值为 5.0 赫兹。

d

调制深度百分比。范围是 0.0 - 1.0。默认值为 0.5。

### 36.114抖音

正弦相位调制。

过滤器接受以下选项：

F

以赫兹为单位的调制频率。范围是 0.1 - 20000.0。默认值为 5.0 赫兹。

d

调制深度百分比。范围是 0.0 - 1.0。默认值为 0.5。

### 36.115 虚拟低音

应用音频虚拟低音过滤器。

该滤波器接受立体声输入并产生带有 LFE (2.1) 声道输出的立体声。新制作的 LFE 通道增强了原本从两个立体声通道获得的虚拟低音。此滤波器输出左前和右前声道不变，如立体声输入中可用。

过滤器接受以下选项：

隔断

设置虚拟低音截止频率。默认值为 250 赫兹。允许的范围是 100 到 500 赫兹。

力量

设置虚拟低音强度。允许的范围是从 0.5 到 3。默认值为 3。

### 36.116卷

调整输入音频音量。

它接受以下参数：

体积

设置音频音量表达式。

输出值被限制为最大值。

输出音频音量由以下关系给出：

output\_volume = volume \* input\_volume

音量的默认值为“1.0”。

精确

该参数表示数学精度。

它确定将允许哪些输入样本格式，这会影响音量缩放的精度。

固定的

8位定点；这将输入样本格式限制为 U8、S16 和 S32。

漂浮

32 位浮点数；这将输入样本格式限制为 FLT。 （默认）

双倍的

64 位浮点数；这将输入样本格式限制为 DBL。

回放增益

选择在输入帧中遇到 ReplayGain 端数据时的行为。

降低

删除 ReplayGain 边数据，忽略其内容（默认）。

忽略

忽略 ReplayGain 边数据，但将其保留在帧中。

追踪

如果存在，最好使用轨道增益。

专辑

喜欢专辑增益，如果存在的话。

回放增益前置放大器

以 dB 为单位的预放大增益应用于选定的重播增益。

replaygain\_preamp的默认值为0.0。

回放增益\_noclip

通过限制应用的增益来防止削波。

replaygain\_noclip的默认值为1。

评估

在计算体积表达式时设置。

它接受以下值：

'曾经'

仅在过滤器初始化期间或发送“ volume ”命令时对表达式求值一次

'框架'

评估每个传入帧的表达式

默认值为“一次”。

音量表达式可以包含以下参数。

n

帧数（从零开始）

nb\_频道

通道数

nb\_consumed\_samples

过滤器消耗的样本数

nb\_样本

当前帧中的样本数

位置

文件中的原始帧位置

积分

帧PTS

采样率

采样率

起始点

流开始时的 PTS

开始

流开始时间

吨

帧时间

结核病

时间戳时基

体积

最后设定音量值

请注意，当eval设置为“ once ”时，只有sample\_rate和tb变量可用，所有其他变量将计算为 NAN。

* [命令](#Commands-52)
* [例子](#Examples-75)

#### 36.116.1 命令

该过滤器支持以下命令：

体积

修改音量表达式。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

#### 36.116.2 示例

* 将输入音量减半：
* 体积=体积=0.5
* 体积=体积=1/2
* 音量=音量=-6.0206dB

卷的命名键都可以省略，例如：

音量=0.5

* 使用定点精度将输入音频功率提高 6 分贝：
* 音量=音量=6dB：精度=固定
* 时间 10 后衰减音量，湮灭期为 5 秒：
* volume='if(lt(t,10),1,max(1-(t-10)/5,0))':eval=frame

### 36.117 体积检测

检测输入视频的音量。

过滤器没有参数。它仅支持 16 位有符号整数样本，因此输入将在需要时进行转换。当到达输入流结束时，有关卷的统计信息将打印在日志中。

特别是，它将显示平均体积（均方根）、最大体积（基于每个样本）和注册体积值直方图的开始（从最大值到样本的累积 1/1000 ).

所有音量均以相对于最大 PCM 值的分贝为单位。

* [例子](#Examples-76)

#### 36.117.1 示例

以下是输出的摘录：

[Parsed\_volumedetect\_0 0xa23120] mean\_volume：-27 dB

[Parsed\_volumedetect\_0 0xa23120] 最大音量：-4 dB

[Parsed\_volumedetect\_0 0xa23120] histogram\_4db: 6

[Parsed\_volumedetect\_0 0xa23120] histogram\_5db: 62

[Parsed\_volumedetect\_0 0xa23120] histogram\_6db: 286

[Parsed\_volumedetect\_0 0xa23120] histogram\_7db: 1042

[Parsed\_volumedetect\_0 0xa23120] histogram\_8db: 2551

[Parsed\_volumedetect\_0 0xa23120] histogram\_9db: 4609

[Parsed\_volumedetect\_0 0xa23120] histogram\_10db: 8409

代表着：

* 均方能量约为 -27 dB，或 10^-2.7。
* 最大样本位于 -4 dB，或更精确地介于 -4 dB 和 -5 dB 之间。
* 在 -4 dB 时有 6 个样本，在 -5 dB 时有 62 个，在 -6 dB 时有 286 个，等等。

换句话说，将音量提高 +4 dB 不会导致任何削波，提高 +5 dB 会导致 6 个样本削波，等等。

## 37 个音源

以下是对当前可用音频源的描述。

* [缓冲区](#abuffer)
* [aevalsrc](#aevalsrc)
* [afdelaysrc](#afdelaysrc)
* [AFRSRC](#afirsrc)
* [一个空源](#anullsrc)
* [飞行](#flite)
* [噪声源](#anoisesrc)
* [希尔伯特](#hilbert)
* [正弦波](#sinc)
* [正弦波](#sine)

### 37.1 缓冲区

缓冲音频帧，并使它们可用于过滤器链。

libavfilter/buffersrc.h中定义的接口。

它接受以下参数：

时基

将用于提交帧的时间戳的时基。它必须是浮点数或分子/分母形式。

采样率

传入音频缓冲区的采样率。

sample\_fmt

传入音频缓冲区的示例格式。来自libavutil/samplefmt.h中的枚举 AVSampleFormat 的示例格式名称或其对应的整数表示

频道布局

传入音频缓冲区的通道布局。来自libavutil/channel\_layout.c中的 channel\_layout\_map 的通道布局名称或来自libavutil/channel\_layout.h中的 AV\_CH\_LAYOUT\_\* 宏的相应整数表示

频道

传入音频缓冲区的通道数。如果同时指定了channels和channel\_layout ，则它们必须一致。

* [例子](#Examples-77)

#### 37.1.1 例子

abuffer=sample\_rate=44100:sample\_fmt=s16p:channel\_layout=stereo

将指示源接受 44100Hz 的平面 16 位签名立体声。由于名称为“s16p”的示例格式对应于数字 6，而“立体声”通道布局对应于值 0x3，因此这相当于：

abuffer=sample\_rate=44100:sample\_fmt=6:channel\_layout=0x3

### 37.2 aevalsrc

生成由表达式指定的音频信号。

此源在输入中接受一个或多个表达式（每个通道一个），这些表达式被评估并用于生成相应的音频信号。

此来源接受以下选项：

表达式

为每个单独的通道设置以“|”分隔的表达式列表。如果未指定channel\_layout选项，则所选通道布局取决于提供的表达式的数量。否则，最后指定的表达式将应用于剩余的输出通道。

频道布局，c

设置通道布局。指定布局中的通道数量必须等于指定表达式的数量。

持续时间，d

设置源音频的最短持续时间。有关可接受的语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)请注意，生成的持续时间可能大于指定的持续时间，因为生成的音频总是在完整帧的末尾被剪切。

如果未指定，或者表示的持续时间为负数，则音频应该永远生成。

nb\_样本，n

设置每个输出帧每个通道的样本数，默认为 1024。

采样率，s

指定采样率，默认为 44100。

exprs中的每个表达式都可以包含以下常量：

n

评估样本的编号，从0开始

吨

评估样本的时间，以秒为单位，从 0 开始

秒

采样率

* [例子](#Examples-78)

#### 37.2.1 例子

* 产生沉默：
* aevalsrc=0
* 生成频率为 440 Hz 的正弦信号，将采样率设置为 8000 Hz：
* aevalsrc="sin(440\*2\*PI\*t):s=8000"
* 生成双通道信号，明确指定通道布局（前中心 + 后中心）：
* aevalsrc="sin(420\*2\*PI\*t)|cos(430\*2\*PI\*t):c=FC|BC"
* 产生白噪声：
* aevalsrc="-2+随机(0)"
* 生成调幅信号：
* aevalsrc="sin(10\*2\*PI\*t)\*sin(880\*2\*PI\*t)"
* 在 360 Hz 载波上生成 2.5 Hz 双耳节拍：
* aevalsrc="0.1\*sin(2\*PI\*(360-2.5/2)\*t) | 0.1\*sin(2\*PI\*(360+2.5/2)\*t)"

### 37.3afdelaysrc

生成分数延迟 FIR 系数。

[afir过滤器](#afir)一起使用以过滤音频信号。

过滤器接受以下选项：

延迟，d

设置小数延迟。默认为 0。

采样率，r

设置采样率，默认为 44100。

nb\_样本，n

设置每帧的样本数。默认值为 1024。

水龙头，吨

设置输出音频流中滤波器系数的数量。默认值为 0。

频道布局，c

指定频道布局，可以是表示频道布局的字符串。 channel\_layout的默认值为“stereo”。

### 37.4 afirsrc

使用频率采样方法生成 FIR 系数。

[afir过滤器](#afir)一起使用以过滤音频信号。

过滤器接受以下选项：

水龙头，吨

设置输出音频流中的滤波器系数数。默认值为 1025。

频率，f

从设置幅度和相位的位置设置频率点。这必须按非降序排列，第一个元素必须为 0，而最后一个元素必须为 1。元素由空格分隔。

震级，米

为由frequency设置的每个频率点设置幅度值。值的数量必须与频率点的数量相同。值由空格分隔。

相位，p

为frequency设置的每个频率点设置相位值。值的数量必须与频率点的数量相同。值由空格分隔。

采样率，r

设置采样率，默认为 44100。

nb\_样本，n

设置每帧的样本数。默认值为 1024。

win\_func, w

设置窗函数。默认是黑人。

### 37.5 anullsrc

空音频源，返回未处理的音频帧。它主要用作模板并用于分析/调试工具，或用作忽略输入数据的过滤器的源（例如 sox 合成器过滤器）。

此来源接受以下选项：

channel\_layout, cl

指定频道布局，可以是整数或表示频道布局的字符串。 channel\_layout的默认值为“stereo”。

libavutil/channel\_layout.c中的 channel\_layout\_map 定义，了解字符串和通道布局值之间的映射。

采样率，r

指定采样率，默认为 44100。

nb\_样本，n

设置每个请求帧的样本数。

持续时间，d

设置源音频的持续时间。有关可接受的语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)

如果未指定，或者表示的持续时间为负数，则音频应该永远生成。

* [例子](#Examples-79)

#### 37.5.1 例子

* 将采样率设置为 48000 Hz，将通道布局设置为 AV\_CH\_LAYOUT\_MONO。
* anullsrc=r=48000:cl=4
* 使用更明显的语法执行相同的操作：
* anullsrc=r=48000:cl=单声道

所有参数都需要明确定义。

### 37.6 片

使用 libflite 库合成语音话语。

--enable-libflite配置 FFmpeg 。

请注意，2.0 之前的 flite 库版本不是线程安全的。

过滤器接受以下选项：

列表声音

如果设置为 1，列出可用语音的名称并立即退出。默认值为 0。

nb\_样本，n

设置每帧的最大样本数。默认值为 512。

文本文件

设置包含要朗读的文本的文件名。

文本

设置要朗读的文字。

声音，v

设置用于语音合成的声音。默认值为kal 。另请参阅list\_voices选项。

* [例子](#Examples-80)

#### 37.6.1 例子

* 从文件speech.txt中读取，并使用标准的 flite 语音合成文本：
* flite=textfile=speech.txt
* slt语音朗读指定文本：
* flite=text='再会了，Sub-Sub 的可怜恶魔，我是它的评论员':voice=slt
* 输入文本到 ffmpeg：
* ffmpeg -f lavfi -i flite=text='再会了，Sub-Sub 的可怜恶魔，我是它的评论员':voice=slt
* flite和lavfi设备让ffplay说出指定的文本：
* ffplay -f lavfi flite=text='不再为你所做的而悲伤。

有关 libflite 的更多信息，请查看： <http://www.festvox.org/flite/>

### 37.7 噪声源

生成噪声音频信号。

过滤器接受以下选项：

采样率，r

指定采样率。默认值为 48000 赫兹。

幅度，一个

指定生成的音频流的振幅 (0.0 - 1.0)。默认值为 1.0。

持续时间，d

指定生成的音频流的持续时间。不指定此选项会导致无限长度的噪声。

颜色，颜色，c

指定噪音的颜色。可用的噪声颜色有白色、粉红色、棕色、蓝色、紫色和天鹅绒。默认颜色为白色。

种子

指定用于播种 PRNG 的值。

nb\_样本，n

设置每个输出帧的样本数，默认为 1024。

* [例子](#Examples-81)

#### 37.7.1 例子

* 生成 60 秒的粉红噪声，采样率为 44.1 kHz，振幅为 0.5：
* anoisesrc=d=60:c=粉红色:r=44100:a=0.5

### 37.8 希尔伯特

生成奇抽头希尔伯特变换 FIR 系数。

[afir过滤器](#afir)一起使用，以将信号相移 90 度。

这用于许多矩阵编码方案和分析信号生成。该过程通常写为乘以虚数单位 i（或 j）。

过滤器接受以下选项：

采样率，s

设置采样率，默认为 44100。

水龙头，吨

设置 FIR 滤波器的长度，默认为 22051。

nb\_样本，n

设置每帧的样本数。

win\_func, w

设置生成 FIR 系数时要使用的窗函数。

### 37.9正弦

生成 sinc 开窗低通、高通、带通或带阻 FIR 系数。

[afir过滤器](#afir)一起使用以过滤音频信号。

过滤器接受以下选项：

采样率，r

设置采样率，默认为 44100。

nb\_样本，n

设置每帧的样本数。默认值为 1024。

生命值

设置高通频率。默认为 0。

LP

设置低通频率。默认值为 0。如果高通频率低于低通频率且低通频率高于 0，则滤波器将创建带通滤波器系数，否则将创建带阻滤波器系数。

阶段

设置滤波器相位响应。默认值为 50。允许的范围为 0 到 100。

贝塔

设置 Kaiser 窗口测试版。

属性

设置阻带衰减。默认值为 120dB，允许范围为 40 至 180 dB。

圆形的

启用舍入，默认情况下是禁用的。

hptaps

设置高通滤波器的抽头数。

lptaps

设置低通滤波器的抽头数。

### 37.10 正弦

生成由振幅为 1/8 的正弦波组成的音频信号。

音频信号是位精确的。

过滤器接受以下选项：

频率，f

设置载波频率。默认值为 440 赫兹。

beep\_factor, b

每秒启用一次周期性蜂鸣，频率为beep\_factor乘以载波频率。默认值为 0，表示禁用蜂鸣声。

采样率，r

指定采样率，默认为 44100。

持续时间，d

指定生成的音频流的持续时间。

samples\_per\_frame

设置每个输出帧的样本数。

表达式可以包含以下常量：

n

输出音频帧的（连续）编号，从 0 开始。

积分

输出音频帧的PTS（Presentation TimeStamp），单位为TB 。

吨

输出音频帧的 PTS，以秒为单位。

结核病

输出音频帧的时基。

默认值为1024 。

* [例子](#Examples-82)

#### 37.10.1 例子

* 生成一个简单的 440 Hz 正弦波：
* 正弦波
* 生成每秒 880 赫兹蜂鸣声的 220 赫兹正弦波，持续 5 秒：
* 正弦=220:4:d=5
* 正弦=f=220:b=4:d=5
* 正弦=频率=220:beep\_factor=4:duration=5
* 1602、1601、1602、1601、1602 NTSC 模式生成 1 kHz 正弦波：
* 正弦=1000:samples\_per\_frame='st(0,mod(n,5)); 1602-不（不（eq（ld（0），1）+ eq（ld（0），3）））'

## 38 个音频接收器

以下是对当前可用音频接收器的描述。

* [缓冲池](#abuffersink)
* [空槽](#anullsink)

### 38.1 缓冲槽

缓冲音频帧，并使它们可用于过滤器链的末尾。

libavfilter/buffersink.h或选项系统中定义的接口。

它接受指向 AVABufferSinkContext 结构的指针，该结构定义传入缓冲区的格式，作为不透明参数传递给avfilter\_init\_filter进行初始化。

### 38.2 空槽

空音频接收器；对输入音频绝对不做任何事情。它主要用作模板和用于分析/调试工具。

## 39个视频过滤器

配置 FFmpeg 构建时，可以使用--disable-filters禁用任何现有过滤器。配置输出将显示构建中包含的视频过滤器。

以下是对当前可用的视频过滤器的描述。

* [地址](#addroi)
* [阿尔法提取物](#alphaextract)
* [字母合并](#alphamerge)
* [放大](#amplify)
* [屁股](#ass)
* [噪音](#atadenoise)
* [平均模糊](#avgblur)
* [背景键](#backgroundkey)
* [bbox](#bbox)
* [双边](#bilateral)
* [双边\_cuda](#bilateral_005fcuda)
* [位平面噪声](#bitplanenoise)
* [黑检](#blackdetect)
* [黑框](#blackframe)
* [混合](#blend-1)
* [块检测](#blockdetect-1)
* [模糊检测](#blurdetect-1)
* [bm3d](#bm3d)
* [框模糊](#boxblur)
* [bwdif](#bwdif)
* [中科院](#cas)
* [色度保持](#chromahold)
* [色度键](#chromakey-1)
* [chromakey\_cuda](#chromakey_005fcuda)
* [色元](#chromanr)
* [色移](#chromashift)
* [思科示波器](#ciescope)
* [编解码器视图](#codecview)
* [色彩均衡](#colorbalance)
* [颜色对比](#colorcontrast)
* [色彩校正](#colorcorrect)
* [颜色通道混合器](#colorchannelmixer)
* [着色](#colorize)
* [色键](#colorkey)
* [颜色保持](#colorhold)
* [色阶](#colorlevels)
* [色图](#colormap)
* [颜色矩阵](#colormatrix)
* [色彩空间](#colorspace)
* [色彩空间\_cuda](#colorspace_005fcuda)
* [色温](#colortemperature)
* [卷积](#convolution)
* [卷积](#convolve)
* [复制](#copy)
* [核心形象](#coreimage-1)
* [更正](#corr)
* [覆盖矩形](#cover_005frect)
* [庄稼](#crop)
* [作物检测](#cropdetect)
* [线索](#cue-1)
* [曲线](#curves-1)
* [数据范围](#datascope)
* [模糊](#dblur)
* [dctdnoiz](#dctdnoiz)
* [解带](#deband)
* [去块](#deblock)
* [抽取](#decimate-1)
* [去卷积](#deconvolve)
* [点](#dedot)
* [放气](#deflate)
* [去闪烁](#deflicker)
* [消颤器](#dejudder)
* [标志](#delogo)
* [下雨](#derain-1)
* [去抖动](#deshake)
* [排泄](#despill)
* [删除](#detelecine)
* [扩张](#dilation-1)
* [取代](#displace)
* [dnn\_分类](#dnn_005fclassify)
* [dnn\_detect](#dnn_005fdetect)
* [dnn\_processing](#dnn_005fprocessing-1)
* [抽屉](#drawbox)
* [画图](#drawgraph-1)
* [拉网](#drawgrid)
* [绘制文本](#drawtext-1)
* [边缘检测](#edgedetect)
* [电子书](#elbg)
* [熵](#entropy)
* [环氧树脂](#epx)
* [当量](#eq)
* [侵蚀](#erosion-1)
* [估计差](#estdif)
* [接触](#exposure)
* [提取平面](#extractplanes)
* [褪色](#fade)
* [反馈](#feedback)
* [fftdnoiz](#fftdnoiz)
* [fftfilt](#fftfilt)
* [场地](#field)
* [现场提示](#fieldhint)
* [现场比赛](#fieldmatch)
* [战地秩序](#fieldorder)
* [先进先出, afifo](#fifo_002c-afifo)
* [填充边框](#fillborders)
* [查找矩形](#find_005frect)
* [填洪](#floodfill)
* [格式](#format-1)
* [每秒帧数](#fps-1)
* [框架包](#framepack)
* [帧率](#framerate)
* [步幅](#framestep)
* [冻结检测](#freezedetect)
* [定格](#freezeframes)
* [弗雷尔](#frei0r-1)
* [fspp](#fspp)
* [模糊](#gblur)
* [格格](#geq)
* [毕业乐趣](#gradfun)
* [图形监视器](#graphmonitor-1)
* [灰色世界](#grayworld)
* [灰边](#greyedge)
* [指导的](#guided)
* [杂乱无章](#haldclut-1)
* [翻转](#hflip)
* [海斯特](#histeq)
* [直方图](#histogram-1)
* [hqdn3d](#hqdn3d-1)
* [硬件下载](#hwdownload-1)
* [地图](#hwmap)
* [上传](#hwupload-1)
* [hwupload\_cuda](#hwupload_005fcuda-1)
* [hqx](#hqx)
* [堆栈](#hstack-1)
* [保持](#hsvhold)
* [hsvkey](#hsvkey)
* [色调](#hue)
* [色相饱和度](#huesaturation)
* [迟滞](#hysteresis)
* [icc检测](#iccdetect)
* [iccgen](#iccgen)
* [身份](#identity)
* [意念](#idet)
* [伊尔](#il)
* [膨胀](#inflate)
* [交错](#interlace)
* [内核设计](#kerndeint)
* [樱桃酒](#kirsch)
* [拉格乐趣](#lagfun)
* [镜头矫正](#lenscorrection)
* [镜头乐趣](#lensfun)
* [安慰剂](#libplacebo)
* [libvmaf](#libvmaf)
* [限制差异](#limitdiff)
* [限制器](#limiter)
* [环形](#loop)
* [lut1d](#lut1d)
* [lut3d](#lut3d-1)
* [亮度键](#lumakey)
* [lut, lutrgb, lutyuv](#lut_002c-lutrgb_002c-lutyuv)
* [卢特2, 卢特2](#lut2_002c-tlut2)
* [蒙面钳](#maskedclamp)
* [掩码最大值](#maskedmax)
* [蒙面合并](#maskedmerge)
* [掩码最小值](#maskedmin)
* [屏蔽阈值](#maskedthreshold)
* [面具乐趣](#maskfun)
* [麦迪逊](#mcdeint)
* [中位数](#median)
* [合并平面](#mergeplanes)
* [估计](#mestimate)
* [中均衡器](#midequalizer)
* [插值法](#minterpolate)
* [混合](#mix)
* [单色](#monochrome)
* [吗啡](#morpho)
* [抽取](#mpdecimate)
* [悲伤](#msad)
* [乘](#multiply)
* [否定](#negate)
* [n均值](#nlmeans-1)
* [尼迪](#nnedi)
* [无格式](#noformat)
* [噪音](#noise-1)
* [归一化](#normalize)
* [无效的](#null-2)
* [文字识别](#ocr)
* [ocv](#ocv)
* [示波器](#oscilloscope)
* [覆盖](#overlay-1)
* [overlay\_cuda](#overlay_005fcuda-1)
* [低噪声](#owdenoise)
* [软垫](#pad-1)
* [调色板生成器](#palettegen-1)
* [调色板使用](#paletteuse)
* [看法](#perspective)
* [阶段](#phase)
* [光敏性](#photosensitivity)
* [像素测试](#pixdesctest)
* [像素化](#pixelize)
* [像素范围](#pixscope)
* [pp](#pp)
* [7页](#pp7)
* [预乘](#premultiply)
* [普雷维特](#prewitt)
* [伪彩](#pseudocolor)
* [压力信噪比](#psnr)
* [拉起](#pullup-1)
* [qp](#qp)
* [随机的](#random)
* [阅读雅608](#readeia608)
* [读维茨](#readvitc)
* [重新映射](#remap)
* [去除颗粒](#removegrain)
* [删除徽标](#removelogo)
* [重复字段](#repeatfields)
* [撤销](#reverse)
* [rgbashift](#rgbashift)
* [罗伯茨](#roberts)
* [旋转](#rotate)
* [萨布](#sab)
* [规模](#scale-1)
* [scale\_cuda](#scale_005fcuda)
* [scale\_npp](#scale_005fnpp-1)
* [scale2ref](#scale2ref)
* [scale2ref\_npp](#scale2ref_005fnpp)
* [夏尔](#scharr)
* [滚动](#scroll)
* [scdet](#scdet-1)
* [精选颜色](#selectivecolor-1)
* [单独的字段](#separatefields-1)
* [塞达尔，塞萨尔](#setdar_002c-setsar)
* [设定场](#setfield-1)
* [设置参数](#setparams-1)
* [锐化npp](#sharpen_005fnpp)
* [剪切](#shear)
* [显示信息](#showinfo)
* [显示调色板](#showpalette)
* [洗牌](#shuffleframes)
* [洗牌像素](#shufflepixels)
* [洗牌飞机](#shuffleplanes)
* [信号统计](#signalstats-1)
* [签名](#signature-1)
* [西提](#siti-1)
* [智能模糊](#smartblur-1)
* [索贝尔](#sobel)
* [菌种](#spp-1)
* [高级](#sr-1)
* [西姆](#ssim)
* [立体立体](#stereo3d)
* [流选择，流选择](#streamselect_002c-astreamselect)
* [字幕](#subtitles-1)
* [超级2xsai](#super2xsai)
* [交换协议](#swaprect)
* [交换紫外线](#swapuv)
* [混合](#tblend)
* [电视电影](#telecine)
* [直方图](#thistogram)
* [临界点](#threshold)
* [缩略图](#thumbnail)
* [瓦](#tile-1)
* [交织](#tinterlace)
* [中位数](#tmedian)
* [均衡器](#tmidequalizer)
* [混音](#tmix)
* [色调图](#tonemap-1)
* [键盘](#tpad)
* [转置](#transpose-1)
* [transpose\_npp](#transpose_005fnpp)
* [修剪](#trim)
* [不预乘](#unpremultiply)
* [锐化](#unsharp-1)
* [直到](#untile-1)
* [USPP](#uspp)
* [v360](#v360)
* [模糊降噪器](#vaguedenoiser)
* [可变模糊](#varblur)
* [矢量示波器](#vectorscope)
* [视频检测](#vidstabdetect-1)
* [视标变换](#vidstabtransform-1)
* [翻转](#vflip)
* [vfrdet](#vfrdet)
* [充满活力](#vibrance)
* [维夫](#vif)
* [小插图](#vignette-1)
* [vmaf运动](#vmafmotion)
* [vstack](#vstack-1)
* [w3fdif](#w3fdif)
* [波形](#waveform)
* [编织，双面编织](#weave_002c-doubleweave)
* [xbr](#xbr)
* [相关联](#xcorrelate)
* [淡入淡出](#xfade)
* [x中位数](#xmedian)
* [堆栈](#xstack-1)
* [亚迪夫](#yadif-1)
* [yadif\_cuda](#yadif_005fcuda)
* [雅普模糊](#yaepblur)
* [缩放平移](#zoompan)
* [zscale](#zscale-1)

### 39.1 加德罗伊

在视频帧中标记感兴趣区域。

帧数据原封不动地通过，但元数据附加到帧指示感兴趣的区域，这可能会影响以后编码的行为。可以通过多次应用过滤器来标记多个区域。

X

区域距帧左边缘的距离（以像素为单位）。

是

区域距框架上边缘的距离（以像素为单位）。

w

以像素为单位的区域宽度。

H

以像素为单位的区域高度。

参数x 、 y 、 w和h是表达式，可能包含以下变量：

我

输入框的宽度。

我

输入框的高度。

偏移量

在该区域内应用的量化偏移量。

这必须是 -1 到 +1 范围内的实数值。零值表示没有质量变化。负值要求更好的质量（更少的量化），而正值要求更差的质量（更大的量化）。

该范围经过校准，因此极值表示可能的最大偏移量 - 如果帧的其余部分以可能的最差质量编码，则偏移量 -1 表示该区域应该以可能的最佳质量编码。然后以某种依赖于编解码器的方式插入中间值。

例如，在 10 位 H.264 中，量化参数在 -12 和 51 之间变化。因此，典型的 qoffset 值 -1/10 表示该区域应该使用大约整个范围的十分之一的 QP 编码，优于框架的其余部分。因此，如果帧的大部分要用大约 30 的 QP 进行编码，则该区域将获得大约 24 的 QP（偏移量大约为 -1/10 \* (51 - -12) = -6.3）。极值 -1 表示该区域应该以可能的最佳质量编码，而不管帧的其余部分如何处理——也就是说，应该以 -12 的 QP 编码。

清除

如果设置为 true，则在添加新区域之前删除框架上标记的任何现有感兴趣区域。

* [例子](#Examples-83)

#### 39.1.1 例子

* 将框架的中心四分之一标记为有趣。
* addroi=iw/4:ih/4:iw/2:ih/2:-1/10
* 将帧左边缘的 100 像素宽区域标记为非常无趣（以比帧其余部分低得多的质量进行编码）。
* 地址=0:0:100:ih:+1/5

### 39.2 阿尔法提取物

从输入中提取 alpha 分量作为灰度视频。这对于alphamerge过滤器特别有用。

### 39.3 字母合并

使用第二个输入的灰度值添加或替换主要输入的 alpha 分量。这旨在与alphaextract一起使用，以允许传输或存储具有不支持 alpha 通道的格式的 alpha 的帧序列。

例如，要从正常的 YUV 编码视频和使用alphaextract创建的单独视频重建完整帧，您可以使用：

电影=in\_alpha.mkv [alpha]; [输入][alpha] alphamerge [输出]

### 39.4 放大

放大当前像素与同一像素位置相邻帧像素之间的差异。

此过滤器接受以下选项：

半径

设置框架半径。默认值为 2。允许的范围为 1 到 63。例如，半径为 3 将指示过滤器计算 7 帧的平均值。

因素

设置因子以放大差异。默认值为 2。允许的范围为 0 到 65535。

临界点

设置差异放大的阈值。任何大于或等于该值的差异都不会改变源像素。默认值为 10。允许的范围为 0 到 65535。

宽容

设置差异放大的公差。任何低于此值的差异都不会改变源像素。默认值为 0。允许的范围为 0 到 65535。

低的

设置更改源像素的下限。默认值为 65535。允许的范围是从 0 到 65535。此选项控制将减少源像素值的最大可能值。

高的

设置更改源像素的上限。默认值为 65535。允许的范围是从 0 到 65535。此选项控制将增加源像素值的最大可能值。

飞机

设置要过滤的平面。默认为全部。允许的范围是从 0 到 15。

* [命令](#Commands-53)

#### 39.4.1 命令

此过滤器支持以下与同名选项相对应的[命令：](#commands)

因素

临界点

宽容

低的

高的

飞机

### 39.5 屁股

[字幕过滤器](#subtitles)相同，只是它不需要 libavcodec 和 libavformat 才能工作。另一方面，它仅限于 ASS（Advanced Substation Alpha）字幕文件。

除了[字幕](#subtitles)过滤器的常用选项外，此过滤器还接受以下选项：

成型

设置整形引擎

可用值是：

'汽车'

默认的 libass 整形引擎，这是最好的。

'简单'

快速、与字体无关的整形器，只能进行替换

'复杂'

使用 OpenType 进行替换和定位的较慢的整形器

默认值为自动。

### 39.6 噪音

对视频输入应用自适应时间平均降噪器。

过滤器接受以下选项：

0a

为第一个平面设置阈值 A。默认值为 0.02。有效范围是 0 到 0.3。

0b

为第一个平面设置阈值 B。默认值为 0.04。有效范围是 0 到 5。

1a

为第二平面设置阈值 A。默认值为 0.02。有效范围是 0 到 0.3。

1b

为第二平面设置阈值 B。默认值为 0.04。有效范围是 0 到 5。

2a

为第 3 平面设置阈值 A。默认值为 0.02。有效范围是 0 到 0.3。

2b

为第 3 平面设置阈值 B。默认值为 0.04。有效范围是 0 到 5。

阈值 A 旨在对输入信号的突然变化作出反应，而阈值 B 旨在对输入信号的连续变化作出反应。

秒

设置帧数过滤器将用于平均。默认为 9。必须是 [5, 129] 范围内的奇数。

p

设置帧过滤器的平面将用于平均。默认为全部。

A

设置算法过滤器的变体将用于平均。默认为p平行。或者可以设置为s serial。

并行可以比串行更快，而其他方式永远不会成立。并行将在第一次变化大于阈值时提前中止，而串行将继续处理帧的另一侧，如果它们等于或低于阈值。

0秒

1秒

2秒

为第一平面、第二平面或第三平面设置西格玛。默认值为 32767。有效范围为 0 到 32767。此选项控制由大小定义的半径中每个像素的权重。默认值意味着每个像素具有相同的权重。将此选项设置为 0 会有效地禁用过滤。

* [命令](#Commands-54)

#### 39.6.1 命令

此过滤器支持与选项相同的[命令](#commands)，但选项s除外。该命令接受与相应选项相同的语法。

### 39.7平均模糊

应用平均模糊滤镜。

过滤器接受以下选项：

尺码X

设置水平半径大小。

飞机

设置要过滤的平面。默认情况下，所有平面都被过滤。

尺码Y

sizeX相同。默认为0 。

* [命令](#Commands-55)

#### 39.7.1 命令

此过滤器支持与选项相同的命令。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.8 背景键

将静态背景变为透明。

过滤器接受以下选项：

临界点

场景变化检测的阈值。

相似

与背景的相似度百分比。

混合

为不相似的像素设置混合量。

* [命令](#Commands-56)

#### 39.8.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.9 bbox

计算输入帧亮度平面中非黑色像素的边界框。

此过滤器计算包含亮度值大于最小允许值的所有像素的边界框。描述边界框的参数打印在过滤器日志中。

过滤器接受以下选项：

最小值

设置最小亮度值。默认值为16 。

* [命令](#Commands-57)

#### 39.9.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.10双边

应用双边滤波器，空间平滑，同时保留边缘。

过滤器接受以下选项：

σS

设置高斯函数的西格玛来计算空间权重。允许的范围是 0 到 512。默认值为 0.1。

西格玛R

设置高斯函数的西格玛来计算范围权重。允许的范围是 0 到 1。默认值为 0.1。

飞机

设置要过滤的平面。默认是第一个。

* [命令](#Commands-58)

#### 39.10.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.11双边\_cuda

CUDA 加速双边滤波器，一种边缘保留滤波器。由于使用了 GPU 加速，此过滤器在数学上是准确的。为获得最佳输出质量，请使用一对一色度子采样，即 yuv444p 格式。

过滤器接受以下选项：

σS

设置高斯函数的sigma来计算空间权重，也称为sigma空间。允许的范围是 0.1 到 512。默认值为 0.1。

西格玛R

设置高斯函数的sigma来计算颜色范围权重，也称为sigma color。允许的范围是 0.1 到 512。默认值为 0.1。

窗口大小

设置双边函数的窗口大小以确定要循环的邻居数。如果输入的数字是偶数，则自动添加一个。允许的范围是 1 到 255。默认值为 1。

* [例子](#Examples-84)

#### 39.11.1 例子

* 在视频上应用双边过滤器。
* ./ffmpeg -v 详细 \
* -hwaccel cuda -hwaccel\_output\_format cuda -i input.mp4 \
* -init\_hw\_device 库达\
* -filter\_complex \
* “\
* [0:v]scale\_cuda=格式=yuv444p[缩放视频]；
* [缩放视频]bilateral\_cuda=window\_size=9:sigmaS=3.0:sigmaR=50.0" \
* -an -sn -c:v h264\_nvenc -cq 20 out.mp4

### 39.12 位平面噪声

显示和测量位平面噪声。

过滤器接受以下选项：

位平面

设置要分析的平面。默认为1 。

筛选

位平面中过滤掉噪声像素。默认是禁用的。

### 39.13 黑检测

检测（几乎）全黑的视频间隔。可用于检测章节转换、商业广告或无效录音。

过滤器将其检测分析输出到日志和帧元数据。如果找到至少指定的最短持续时间的黑色段，则会将带有开始和结束时间戳以及持续时间的行打印到级别为info的日志中。此外，每帧打印一条具有调试级别的日志行，显示为该帧检测到的黑色量。

lavfi.black\_start的黑色片段的第一帧和黑色片段结束后的第一帧，并具有密钥lavfi.black\_end 。该值是帧的时间戳。无论指定的最短持续时间如何，都会添加此元数据。

过滤器接受以下选项：

black\_min\_duration, d

设置以秒为单位的检测到的最小黑色持续时间。它必须是一个非负浮点数。

默认值为 2.0。

picture\_black\_ratio\_th, pic\_th

设置考虑图片“黑色”的阈值。表示比率的最小值：

nb\_black\_pixels / nb\_pixels

图片被认为是黑色的。默认值为 0.98。

pixel\_black\_th, pix\_th

设置考虑像素“黑色”的阈值。

阈值表示像素被认为是“黑色”的最大像素亮度值。提供的值根据以下等式缩放：

absolute\_threshold = luminance\_minimum\_value + pixel\_black\_th \* luminance\_range\_size

luminance\_range\_size和luminance\_minimum\_value取决于输入的视频格式，范围是 [0-255] YUV 全范围格式和 [16-235] YUV 非全范围格式。

默认值为 0.10。

以下示例将最大像素阈值设置为最小值，并仅检测 2 秒或更长时间的黑色间隔：

blackdetect=d=2:pix\_th=0.00

### 39.14 黑框

检测（几乎）全黑的帧。可用于检测章节转换或商业广告。输出行包括检测到的帧的帧号、黑度百分比、文件中的位置（如果已知或 -1）以及以秒为单位的时间戳。

为了显示输出行，您需要至少将日志级别设置为 AV\_LOG\_INFO 值。

此过滤器导出帧元数据lavfi.blackframe.pblack 。该值表示图片中低于阈值的像素的百分比。

它接受以下参数：

数量

必须低于阈值的像素百分比；它默认为98 。

门槛，阈值

低于该阈值的像素值被视为黑色；它默认为32 。

### 39.15 混合

将两个视频帧相互混合。

混合过滤器采用两个输入流并输出一个流，第一个输入是“顶层”，第二个输入是“底部”层。默认情况下，输出在最长输入终止时终止。

tblend （时间混合）过滤器从一个流中获取两个连续的帧，并输出通过将新帧混合在旧帧之上而获得的结果。

接受的选项的描述如下。

c0\_模式

c1\_模式

c2\_模式

c3\_模式

全模式

all\_mode的情况下，为特定像素组件或所有像素组件设置混合模式。默认值为normal 。

组件模式的可用值是：

'加法'

'和'

'平均'

'漂白剂'

'燃烧'

'变暗'

'差异'

'划分'

‘闪避’

'排除'

'肢体'

'冻结'

'几何'

'发光'

'谷物提取物'

'谷物合并'

'强光'

'混合'

'硬覆盖'

'谐波'

'热'

'插值'

'减轻'

'线性光'

'乘法'

'乘128 '

'否定'

'正常'

'或'

'覆盖'

'凤凰'

'针灯'

'反映'

'屏幕'

'软差异'

'柔光'

'污点'

'减去'

'生动之光'

'异或'

c0\_opacity

c1\_opacity

c2\_opacity

c3\_opacity

全不透明

all\_opacity的情况下，为特定像素组件或所有像素组件设置混合不透明度。仅与像素分量混合模式结合使用。

c0\_expr

c1\_expr

c2\_expr

c3\_expr

all\_expr

all\_expr的情况下，为特定像素分量或所有像素分量设置混合表达式。请注意，如果设置了相关模式选项，将忽略这些选项。

表达式可以使用以下变量：

否

过滤帧的序号，从0开始。

X

是

当前样本的坐标

W

H

当前过滤平面的宽度和高度

软件

上海

被过滤平面的宽度和高度比例。它是当前平面与亮度平面尺寸之间的比率，例如对于yuv420p帧，亮度平面的值为1,1 ，色度平面的值为0.5,0.5 。

吨

当前帧的时间，以秒为单位。

顶部，一个

第一个视频帧（顶层）当前位置的像素分量值。

底部，乙

第二个视频帧（底层）当前位置的像素分量值。

混合滤镜还支持[帧同步](#framesync)选项。

* [例子](#Examples-85)
* [命令](#Commands-59)

#### 39.15.1 例子

* 在前 10 秒内应用从底层到顶层的过渡：
* blend=all\_expr='A\*(if(gte(T,10),1,T/10))+B\*(1-(if(gte(T,10),1,T/10)))'
* 应用从顶层到底层的线性水平过渡：
* blend=all\_expr='A\*(X/W)+B\*(1-X/W)'
* 应用 1x1 棋盘效果：
* blend=all\_expr='if(eq(mod(X,2),mod(Y,2)),A,B)'
* 应用 uncover left 效果：
* blend=all\_expr='if(gte(N\*SW+X,W),A,B)'
* 应用揭开效果：
* blend=all\_expr='if(gte(YN\*SH,0),A,B)'
* 应用揭开左上效果：
* blend=all\_expr='if(gte(T\*SH\*40+Y,H)\*gte((T\*40\*SW+X)\*W/H,W),A,B)'
* 沿对角线拆分视频并在每一侧显示顶层和底层：
* blend=all\_expr='if(gt(X,Y\*(W/H)),A,B)'
* 显示当前帧和上一帧之间的差异：
* tblend=all\_mode=grainextract

#### 39.15.2 命令

此过滤器支持与选项相同的[命令](#commands)。

### 39.16 块检测

在不改变输入帧的情况下确定帧的块度。

基于 Remco Muijs 和 Ihor Kirenko：“A no-reference blocking artifact measure for adaptive video processing.” 2005 第 13 届欧洲信号处理会议。

过滤器接受以下选项：

period\_min

period\_max

设置用于确定像素网格（周期）的最小值和最大值。默认值为 [3,24]。

飞机

设置要过滤的平面。默认是第一个。

* [例子](#Examples-86)

#### 39.16.1 例子

* 确定第一个平面的块度并在 [8,32] 内搜索周期：
* blockdetect=period\_min=8:period\_max=32:planes=1

### 39.17 模糊检测

在不改变输入帧的情况下确定帧的模糊度。

基于 Marziliano、Pina 等人。 “无参考感知模糊指标。”允许基于块的缩写。

过滤器接受以下选项：

低的

高的

设置 Canny 阈值算法使用的低阈值和高阈值。

高阈值选择“强”边缘像素，然后通过 8-连通性与低阈值选择的“弱”边缘像素连接。

low和high阈值必须在 [0,1] 范围内选择，并且low应小于或等于high 。

低的默认值为20/255 ，高的默认值为50/255 。

半径

定义半径以在边缘像素周围搜索局部最大值。

block\_pct

仅确定最重要块的模糊度，以百分比给出。

区块宽度

确定宽度为block\_width的块的模糊度。如果设置为任何小于 1 的值，则不使用任何块，并且无论block\_height是多少，整个图像都将作为一个图像处理。

区块高度

确定高度块的模糊度block\_height 。如果设置为任何小于 1 的值，则不使用任何块，并且无论block\_width是多少，整个图像都将作为一个图像处理。

飞机

设置要过滤的平面。默认是第一个。

* [例子](#Examples-87)

#### 39.17.1 例子

* 确定 80% 最重要的 32x32 块的模糊：
* blurdetect=block\_width=32:block\_height=32:block\_pct=80

### 39.18 bm3d

使用块匹配 3D 算法对帧进行降噪。

过滤器接受以下选项。

西格玛

设置去噪强度。默认值为 1。允许的范围为 0 到 999.9。去噪算法对sigma很敏感，所以根据来源调整。

堵塞

设置局部补丁大小。这将在 2D 中设置尺寸。

一步

设置处理块的滑动步长。默认值为 4。允许的范围是从 1 到 64。较小的值允许处理更多的参考块并且速度较慢。

团体

为第 3 维设置相似块的最大数量。默认值为 1。设置为 1 时，不进行块匹配。较大的值允许单个组中有更多块。允许的范围是 1 到 256。

范围

设置搜索块匹配的半径。默认值为 9。允许的范围是从 1 到 INT32\_MAX。

迈步

在块匹配的两个搜索位置之间设置步长。默认值为 1。允许的范围是 1 到 64。越小越慢。

泰晤士报

设置块匹配的均方误差阈值。有效范围是 0 到 INT32\_MAX。

高密度脂蛋白

为 3D 变换域中的硬阈值设置阈值参数。较大的值会在频域中产生更强的硬阈值滤波。

估计

设置滤波估计模式。可以是basic或final 。默认是基本的。

参考

如果启用，过滤器将使用第二个流进行块匹配。对于estim选项的基本值默认禁用，如果estim的值为final则始终启用。

飞机

设置要过滤的平面。除 alpha 外，默认值均可用。

* [例子](#Examples-88)

#### 39.18.1 例子

* bm3d 的基本过滤：
* bm3d=sigma=3:block=4:bstep=2:group=1:estim=basic
* 同上，但只过滤亮度：
* bm3d=sigma=3:block=4:bstep=2:group=1:estim=basic:planes=1
* 同上，但有两种估计模式：
* split[a][b],[a]bm3d=sigma=3:block=4:bstep=2:group=1:estim=basic[a],[b][a]bm3d=sigma=3:block= 4:bstep=2:group=16:estim=final:ref=1
* 与上面相同，但使用[nlmeans](#nlmeans)过滤器进行预过滤：
* split[a][b],[a]nlmeans=s=3:r=7:p=3[a],[b][a]bm3d=sigma=3:block=4:bstep=2:group= 16：估计=最终：参考=1

### 39.19 盒模糊

将 boxblur 算法应用于输入视频。

它接受以下参数：

亮度半径，lr

luma\_power, lp

色度半径，cr

色度功率，cp

alpha\_radius, ar

alpha\_power, ap

接受的选项的描述如下。

亮度半径，lr

色度半径，cr

alpha\_radius, ar

设置用于模糊相应输入平面的框半径的表达式（以像素为单位）。

半径值必须是非负数，并且不得大于亮度和 alpha 平面的表达式min(w,h)/2的值，以及色度的min(cw,ch)/2的值飞机。

luma\_radius的默认值为“2”。如果未指定， chroma\_radius和alpha\_radius默认为为luma\_radius设置的相应值。

表达式可以包含以下常量：

w

H

输入的宽度和高度（以像素为单位）。

连续波

通道

输入色度图像的宽度和高度（以像素为单位）。

hsub

vsub

水平和垂直色度子样本值。例如，对于像素格式“yuv422p”， hsub为 2， vsub为 1。

luma\_power, lp

色度功率，cp

alpha\_power, ap

指定将 boxblur 滤镜应用于相应平面的次数。

luma\_power的默认值为2。如果未指定， chroma\_power和alpha\_power默认为为luma\_power设置的相应值。

值为 0 将禁用该效果。

* [例子](#Examples-89)

#### 39.19.1 例子

* 应用亮度、色度和 alpha 半径设置为 2 的 boxblur 滤镜：
* boxblur=luma\_radius=2:luma\_power=1
* 方框模糊=2:1
* 将亮度半径设置为 2，并将 alpha 和色度半径设置为 0：
* boxblur=2:1:cr=0:ar=0
* 将亮度和色度半径设置为视频维度的一部分：
* boxblur=luma\_radius=min(h\,w)/10:luma\_power=1:chroma\_radius=min(cw\,ch)/10:chroma\_power=1

### 39.20 bwdif

去隔行输入视频（“bwdif”代表“Bob Weaver Deinterlacing Filter”）。

基于 yadif 的运动自适应去隔行，使用 w3fdif 和三次插值算法。它接受以下参数：

模式

采用的隔行扫描模式。它接受以下值之一：

0，发送帧

每一帧输出一帧。

1、发送字段

为每个字段输出一帧。

默认值为send\_field 。

平价

输入隔行扫描视频采用的图片场奇偶校验。它接受以下值之一：

0, tff

假设顶部字段是第一个。

1、好朋友

假设底部字段是第一个。

-1，汽车

启用字段奇偶校验的自动检测。

默认值为自动。如果隔行扫描未知或解码器不导出此信息，则假定顶场优先。

取消

指定要去隔行扫描的帧。接受以下值之一：

0, 全部

去隔行扫描所有帧。

1、交错

仅去隔行标记为隔行扫描的帧。

默认值为全部。

### 39.21 约

将对比度自适应锐化过滤器应用于视频流。

过滤器接受以下选项：

力量

设置锐化强度。默认值为 0。

飞机

设置要过滤的平面。默认值是过滤除 alpha 平面之外的所有平面。

* [命令](#Commands-60)

#### 39.21.1 命令

此过滤器支持与选项相同的[命令](#commands)。

### 39.22 色度保持

删除除某些颜色之外的所有颜色的所有颜色信息。

过滤器接受以下选项：

颜色

不会被中性彩度取代的颜色。

相似

与上述颜色的相似度百分比。 0.01 只匹配准确的键颜色，而 1.0 匹配所有颜色。

混合

混合百分比。 0.0 使像素完全变灰或完全不变灰。值越高，颜色保留得越多。

yuv

表示传递的颜色已经是 YUV 而不是 RGB 的信号。

启用此功能后，“绿色”或“红色”等文字颜色不再有意义。这可用于将准确的 YUV 值作为十六进制数传递。

* [命令](#Commands-61)

#### 39.22.1 命令

此过滤器支持与选项相同的[命令](#commands)。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.23 色键

YUV 色彩空间颜色/色度键控。

过滤器接受以下选项：

颜色

将被透明度替换的颜色。

相似

与关键颜色的相似度百分比。

0.01 只匹配准确的键颜色，而 1.0 匹配所有颜色。

混合

混合百分比。

0.0 使像素完全透明或完全不透明。

较高的值导致半透明像素，透明度越高，像素颜色与键颜色越相似。

yuv

表示传递的颜色已经是 YUV 而不是 RGB 的信号。

启用此功能后，“绿色”或“红色”等文字颜色不再有意义。这可用于将准确的 YUV 值作为十六进制数传递。

* [命令](#Commands-62)
* [例子](#Examples-90)

#### 39.23.1 命令

此过滤器支持与选项相同的[命令](#commands)。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

#### 39.23.2 例子

* 使输入图像中的每个绿色像素透明：
* ffmpeg -i input.png -vf chromakey=green out.png
* 在静态黑色背景上叠加绿屏视频。
* ffmpeg -f lavfi -i color=c=black:s=1280x720 -i video.mp4 -shortest -filter\_complex "[1:v]chromakey=0x70de77:0.1:0.2[ckout];[0:v][ckout]覆盖[out]" -map "[out]" output.mkv

### 39.24 chromekey\_cuda

CUDA 加速 YUV 色彩空间颜色/色度键控。

这个过滤器像普通的色键过滤器一样工作，但在 CUDA 帧上运行。有关详细信息和参数，请参阅[chromakey](#chromakey) 。

* [例子](#Examples-91)

#### 39.24.1 例子

* 使输入视频中的所有绿色像素透明，并将其用作另一个视频的叠加层：
* ./ffmpeg\
* -hwaccel cuda -hwaccel\_output\_format cuda -i input\_green.mp4 \
* -hwaccel cuda -hwaccel\_output\_format cuda -i base\_video.mp4 \
* -init\_hw\_device 库达\
* -filter\_complex \
* “\
* [0:v]chromakey\_cuda=0x25302D:0.1:0.12:1[overlay\_video]； \
* [1:v]scale\_cuda=格式=yuv420p[基础]； \
* [基础][overlay\_video]overlay\_cuda" \
* -an -sn -c:v h264\_nvenc -cq 20 输出.mp4
* 处理两个软件源，显式上传帧：
* ./ffmpeg -init\_hw\_device cuda=cuda -filter\_hw\_device cuda \
* -f lavfi -i color=size=800x600:color=white,format=yuv420p \
* -f lavfi -i yuvtestsrc=size=200x200,format=yuv420p \
* -filter\_complex \
* “\
* [0]hwupload[下]; \
* [1]hwupload,chromakey\_cuda=green:0.1:0.12[over]; \
* [下][上]overlay\_cuda" \
* -c:v hevc\_nvenc -cq 18 -preset 慢输出.mp4

### 39.25 色满

减少色度噪声。

过滤器接受以下选项：

thres

设置平均色度值的阈值。当前像素和相邻像素的Y、U、V像素分量的绝对差值之和低于该阈值将被用于平均。亮度分量保持不变并复制到输出。默认值为 30。允许的范围为 1 到 200。

大小w

设置用于平均的矩形的水平半径。允许的范围是 1 到 100。默认值为 5。

大小

设置用于平均的矩形的垂直半径。允许的范围是 1 到 100。默认值为 5。

步骤w

平均时设置水平步长。默认值为 1。允许的范围为 1 到 50。主要用于加速过滤。

斯蒂芬

平均时设置垂直步长。默认值为 1。允许的范围为 1 到 50。主要用于加速过滤。

三

设置平均色度值的 Y 阈值。为当前像素和相邻像素的 Y 分量之间的最大允许差异设置更精细的控制。默认值为 200。允许的范围为 1 到 200。

通过

设置平均色度值的 U 阈值。为当前像素和相邻像素的 U 分量之间的最大允许差异设置更精细的控制。默认值为 200。允许的范围为 1 到 200。

threv

设置平均色度值的 V 阈值。为当前像素和相邻像素的 V 分量之间的最大允许差异设置更精细的控制。默认值为 200。允许的范围为 1 到 200。

距离

设置计算中使用的距离类型。

'曼哈顿'

绝对的区别。

'欧几里得'

差异平方。

默认距离类型是曼哈顿。

* [命令](#Commands-63)

#### 39.25.1 命令

此过滤器支持与选项相同的[命令](#commands)。该命令接受与相应选项相同的语法。

### 39.26 色移

水平和/或垂直移动色度像素。

过滤器接受以下选项：

cbh

设置水平移动色度蓝色的量。

简历

设置垂直移动色度蓝色的量。

crh

设置水平移动色度红色的量。

车速

设置垂直移动色度红色的量。

边缘

设置边缘模式，可以是smear 、 default 或warp 。

* [命令](#Commands-64)

#### 39.26.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.27 ciescope

显示 CIE 色图，并在其上叠加像素。

过滤器接受以下选项：

系统

设置颜色系统。

' ntsc, 470m '

' ebu, 470bg '

'样本'

' 240m '

'苹果'

'更宽的'

' cie1931 '

' rec709, hdtv '

'超高清电视，rec2020 '

' dcip3 '

cie

设置 CIE 系统。

' xy '

' ucs '

'爱'

色域

设置要绘制的色域。

有关可用值，请参阅系统选项。

大小，小号

设置 ciescope 大小，默认设置为 512。

强度，我

设置用于将输入像素值映射到 CIE 图的强度。

对比

设置用于绘制超出活动颜色系统色域的舌头颜色的对比度。

校正伽玛

示波器上显示的正确伽马，默认情况下启用。

秀白

在 CIE 图上显示白点，默认禁用。

伽马

设置输入伽马。仅用于 XYZ 输入色彩空间。

充满

填充 CIE 颜色。默认情况下启用。

### 39.28编解码器视图

可视化某些编解码器导出的信息。

一些编解码器可以使用边数据或其他方式通过帧导出信息。例如，一些基于 MPEG 的编解码器通过编解码器flags2选项中的export\_mvs标志导出运动矢量。

过滤器接受以下选项：

堵塞

使用亮度平面显示块分区结构。

MV

设置运动矢量以可视化。

mv的可用标志是：

' pf '

P 帧的前向预测 MV

'男朋友'

B 帧的前向预测 MV

' bb '

B 帧的后向预测 MV

qp

使用色度平面显示量化参数。

mv\_type，mvt

将运动矢量类型设置为可视化。包括来自所有帧的 MV，除非由frame\_type选项指定。

mv\_type的可用标志是：

' fp '

前向预测 MV

'英国石油公司'

向后预测的 MV

frame\_type, 英尺

设置帧类型以可视化的运动矢量。

frame\_type的可用标志是：

'如果'

帧内编码帧（I 帧）

' pf '

预测帧（P 帧）

'男朋友'

双向预测帧（B 帧）

* [例子](#Examples-92)

#### 39.28.1 例子

* ffplay可视化所有帧的前向预测 MV ：
* ffplay -flags2 +export\_mvs input.mp4 -vf codecview=mv\_type=fp
* ffplay可视化 P 和 B 帧的多向 MV ：
* ffplay -flags2 +export\_mvs input.mp4 -vf codecview=mv=pf+bf+bb

### 39.29 色彩平衡

修改输入帧的原色（红色、绿色和蓝色）的强度。

该滤镜允许在阴影、中间调或高光区域调整输入帧以实现红-青、绿-品红或蓝-黄平衡。

正调整值将平衡向原色移动，负值向互补色移动。

过滤器接受以下选项：

rs

gs

废话

调整红色、绿色和蓝色阴影（最暗的像素）。

R M

通用汽车

bm

调整红色、绿色和蓝色中间调（中等像素）。

rh

高

bh

调整红色、绿色和蓝色高光（最亮的像素）。

允许的选项范围是[-1.0, 1.0] 。默认值为0 。

PL

改变色彩平衡时保持亮度。默认是禁用的。

* [例子](#Examples-93)
* [命令](#Commands-65)

#### 39.29.1 例子

* 为阴影添加红色色偏：
* 色彩平衡=rs=.3

#### 39.29.2 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.30 颜色对比

调整 RGB 分量之间的颜色对比度。

过滤器接受以下选项：

遥控

设置红青对比度。默认值为 0.0。允许的范围是从 -1.0 到 1.0。

通用汽车

设置绿色-洋红色对比度。默认值为 0.0。允许的范围是从 -1.0 到 1.0。

经过

设置蓝黄色对比度。默认值为 0.0。允许的范围是从 -1.0 到 1.0。

收件人

通用汽车

拜拜

通过选项值设置每个rc和gm的权重。默认值为 0.0。允许的范围是从 0.0 到 1.0。如果所有权重均为 0.0，则禁用过滤。

PL

设置保持亮度的量。默认值为 0.0。允许的范围是从 0.0 到 1.0。

* [命令](#Commands-66)

#### 39.30.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.31 色彩校正

有选择地调整黑色和白色的颜色白平衡。此滤镜在 YUV 色彩空间中运行。

过滤器接受以下选项：

rl

设置红色阴影点。允许的范围是从 -1.0 到 1.0。默认值为 0。

bl

设置蓝色阴影点。允许的范围是从 -1.0 到 1.0。默认值为 0。

rh

设置红色高光点。允许的范围是从 -1.0 到 1.0。默认值为 0。

bh

设置红色高光点。允许的范围是从 -1.0 到 1.0。默认值为 0。

饱和

设置饱和度。允许的范围是从 -3.0 到 3.0。默认值为 1。

分析

手动以外的任何设置，它将分析每一帧并使用派生参数来过滤输出帧。

可能的值是：

'手册'

'平均'

'最小最大'

'中位数'

默认值为manual 。

* [命令](#Commands-67)

#### 39.31.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.32 彩色通道混合器

通过重新混合颜色通道来调整视频输入帧。

此滤镜通过添加与相同像素的其他通道关联的值来修改颜色通道。例如，如果要修改的值为红色，则输出值为：

红色=红色\* rr +蓝色\* rb +绿色\* rg + alpha \* ra

过滤器接受以下选项：

rr

rg

r

拉

调整输入红色、绿色、蓝色和 alpha 通道对输出红色通道的贡献。 rr的默认值为1 ， rg 、 rb和ra的默认值为0 。

克

格格

国标

嘎

调整输入红色、绿色、蓝色和 alpha 通道对输出绿色通道的贡献。 gg的默认值为1 ， gr 、 gb和ga的默认值为0 。

br

背景

bb

吧

调整输入红色、绿色、蓝色和 alpha 通道对输出蓝色通道的贡献。 bb的默认值为1 ， br 、 bg和ba的默认值为0 。

ar

银

ab

一个

调整输入红色、绿色、蓝色和 alpha 通道对输出 alpha 通道的贡献。 aa的默认值为1 ， ar 、 ag和ab的默认值为0 。

允许的选项范围是[-2.0, 2.0] 。

个人电脑

设置保留颜色模式。接受的值是：

'无'

禁用颜色保留，这是默认设置。

' lum '

保持亮度。

'最大'

保留 RGB 三元组的最大值。

'平均'

保留 RGB 三元组的平均值。

'总和'

保留 RGB 三元组的总和值。

' nrm '

保留 RGB 三元组的归一化值。

' pwr '

保留 RGB 三元组的功率值。

帕

设置更改颜色时保留的颜色量。允许的范围是[0.0, 1.0] 。默认值为0.0 ，因此被禁用。

* [例子](#Examples-94)
* [命令](#Commands-68)

#### 39.32.1 例子

* 将源转换为灰度：
* 颜色通道混合器=.3:.4:.3:0:.3:.4:.3:0:.3:.4:.3
* 模拟棕褐色调：
* 颜色通道混合器=.393:.769:.189:0:.349:.686:.168:0:.272:.534:.131

#### 39.32.2 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.33 着色

在视频流上覆盖纯色。

过滤器接受以下选项：

色调

设置色调。允许的范围是 0 到 360。默认值为 0。

饱和

设置颜色饱和度。允许的范围是从 0 到 1。默认值为 0.5。

亮度

设置颜色亮度。允许的范围是从 0 到 1。默认值为 0.5。

混合

设置光源亮度的混合。默认设置为 1.0。允许的范围是从 0.0 到 1.0。

* [命令](#Commands-69)

#### 39.33.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.34 色键

RGB 色彩空间颜色键控。此滤镜通过将落在关键颜色的相似半径内的每个像素的 alpha 分量设置为 0 来对 8 位 RGB 格式帧进行操作。相似半径之外的像素的 alpha 值取决于混合选项的值。

过滤器接受以下选项：

颜色

设置 alpha 将被设置为 0（完全透明）的颜色。请参阅[ffmpeg-utils 手册中的 (ffmpeg-utils)“颜色”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。默认为黑色。

相似

设置键颜色的半径，在该范围内其他颜色也具有完全透明。计算出的距离与 3D 空间中关键颜色的 RGB 值和像素颜色之间的单位分数距离有关。范围是 0.01 到 1.0。 0.01 匹配精确键颜色周围非常小的半径，而 1.0 匹配所有内容。默认值为0.01 。

混合

设置如何计算落在相似半径之外的像素的 alpha 值。 0.0 使像素完全透明或完全不透明。较高的值会产生半透明像素，透明度越高，像素颜色与键颜色越相似。范围是 0.0 到 1.0。默认值为0.0 。

* [例子](#Examples-95)
* [命令](#Commands-70)

#### 39.34.1 例子

* 使输入图像中的每个绿色像素透明：
* ffmpeg -i input.png -vf colorkey=green out.png
* 在静态背景图像上叠加绿屏视频。
* ffmpeg -i background.png -i video.mp4 -filter\_complex "[1:v]colorkey=0x3BBD1E:0.3:0.2[ckout];[0:v][ckout]overlay[out]" -map "[out]"输出.flv

#### 39.34.2 命令

此过滤器支持与选项相同的[命令](#commands)。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.35 色保持

删除除某些颜色之外的所有 RGB 颜色的所有颜色信息。

过滤器接受以下选项：

颜色

不会被中性灰色取代的颜色。

相似

与上述颜色的相似度百分比。 0.01 只匹配准确的键颜色，而 1.0 匹配所有颜色。

混合

混合百分比。 0.0 使像素完全变灰。值越高，颜色保留得越多。

* [命令](#Commands-71)

#### 39.35.1 命令

此过滤器支持与选项相同的[命令](#commands)。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.36 色阶

使用级别调整视频输入帧。

过滤器接受以下选项：

利民

吉敏

比敏

爱民

调整红色、绿色、蓝色和 alpha 输入黑点。允许的选项范围是[-1.0, 1.0] 。默认值为0 。

里马克斯

极光

最大

艾玛克斯

调整红色、绿色、蓝色和 alpha 输入白点。允许的选项范围是[-1.0, 1.0] 。默认值为1 。

输入级别用于使高光变亮（亮色调），使阴影变暗（暗色调），改变亮色调和暗色调的平衡。

罗敏

五民

博敏

奥民

调整红色、绿色、蓝色和 alpha 输出黑点。允许的选项范围是[0, 1.0] 。默认值为0 。

罗马克斯

gomax

博美思

奥麦克斯

调整红色、绿色、蓝色和 alpha 输出白点。允许的选项范围是[0, 1.0] 。默认值为1 。

输出电平允许手动选择受限的输出电平范围。

保存

设置保留颜色模式。接受的值是：

'无'

禁用颜色保留，这是默认设置。

' lum '

保持亮度。

'最大'

保留 RGB 三元组的最大值。

'平均'

保留 RGB 三元组的平均值。

'总和'

保留 RGB 三元组的总和值。

' nrm '

保留 RGB 三元组的归一化值。

' pwr '

保留 RGB 三元组的功率值。

* [例子](#Examples-96)
* [命令](#Commands-72)

#### 39.36.1 例子

* 使视频输出更暗：
* 色阶=rimin=0.058:gimin=0.058:bimin=0.058
* 增加对比度：
* colorlevels=rimin=0.039:gimin=0.039:bimin=0.039:rimax=0.96:gimax=0.96:bimax=0.96
* 使视频输出更轻：
* 色阶=rimax=0.902:gimax=0.902:bimax=0.902
* 增加亮度：
* 色阶=romin=0.5:gomin=0.5:bomin=0.5

#### 39.36.2 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.37 色图

将自定义颜色映射应用于视频流。

此过滤器需要三个输入视频流。第一个流是将被过滤掉的视频流。第二个和第三个视频流指定源颜色到目标颜色映射的色块。

过滤器接受以下选项：

补丁大小

以像素为单位设置源和目标视频流补丁大小。

nb\_补丁

设置来自源和目标视频流的最大使用补丁数。默认值是附加视频流中可用的补丁数。允许的最大补丁数是64 。

类型

设置用于目标颜色的调整。可以是相对的也可以是绝对的。默认是绝对的。

核心

设置用于测量映射颜色之间颜色差异的内核。

接受的值是：

'欧几里得'

'我们'

默认为euclidean 。

### 39.38 颜色矩阵

转换颜色矩阵。

过滤器接受以下选项：

来源

时间

指定源和目标颜色矩阵。必须指定这两个值。

接受的值是：

' bt709 '

BT.709

' fcc '

美国联邦通信委员会

' bt601 '

BT.601

' bt470 '

BT.470

' bt470bg '

BT.470BG

' smpte170m '

SMPTE-170M

' smpte240m '

SMPTE-240M

' bt2020 '

BT.2020

例如，要从 BT.601 转换为 SMPTE-240M，请使用以下命令：

颜色矩阵=bt601:smpte240m

### 39.39 色彩空间

转换色彩空间、传输特性或原色。输入视频需要有一个均匀的大小。

过滤器接受以下选项：

全部

一次指定所有颜色属性。

接受的值是：

' bt470m '

BT.470M

' bt470bg '

BT.470BG

' bt601-6-525 '

BT.601-6 525

' bt601-6-625 '

BT.601-6 625

' bt709 '

BT.709

' smpte170m '

SMPTE-170M

' smpte240m '

SMPTE-240M

' bt2020 '

BT.2020

空间

指定输出色彩空间。

接受的值是：

' bt709 '

BT.709

' fcc '

美国联邦通信委员会

' bt470bg '

BT.470BG 或 BT.601-6 625

' smpte170m '

SMPTE-170M 或 BT.601-6 525

' smpte240m '

SMPTE-240M

' ycgco '

YCgCo

' bt2020ncl '

BT.2020 非恒定亮度

trc

指定输出传输特性。

接受的值是：

' bt709 '

BT.709

' bt470m '

BT.470M

' bt470bg '

BT.470BG

'伽玛22 '

恒定伽玛 2.2

'伽玛28 '

恒定伽玛 2.8

' smpte170m '

SMPTE-170M、BT.601-6 625 或 BT.601-6 525

' smpte240m '

SMPTE-240M

' srgb '

SRGB

' iec61966-2-1 '

iec61966-2-1

' iec61966-2-4 '

IEC61966-2-4

' xvycc '

xvycc

' bt2020-10 '

10 位内容的 BT.2020

' bt2020-12 '

BT.2020 用于 12 位内容

初选

指定输出颜色原色。

接受的值是：

' bt709 '

BT.709

' bt470m '

BT.470M

' bt470bg '

BT.470BG 或 BT.601-6 625

' smpte170m '

SMPTE-170M 或 BT.601-6 525

' smpte240m '

SMPTE-240M

'电影'

电影

' smpte431 '

SMPTE-431

' smpte432 '

SMPTE-432

' bt2020 '

BT.2020

' jedec-p22 '

JEDEC P22 荧光粉

范围

指定输出颜色范围。

接受的值是：

'电视'

电视（受限）范围

' mpeg '

MPEG（限制）范围

'电脑'

PC（全）系列

' jpeg '

JPEG（完整）范围

格式

指定输出颜色格式。

接受的值是：

' yuv420p '

YUV 4:2:0 平面 8 位

' yuv420p10 '

YUV 4:2:0 平面 10 位

' yuv420p12 '

YUV 4:2:0 平面 12 位

' yuv422p '

YUV 4:2:2 平面 8 位

' yuv422p10 '

YUV 4:2:2 平面 10 位

' yuv422p12 '

YUV 4:2:2 平面 12 位

' yuv444p '

YUV 4:4:4 平面 8 位

' yuv444p10 '

YUV 4:4:4 平面 10 位

' yuv444p12 '

YUV 4:4:4 平面 12 位

快速地

进行快速转换，跳过伽玛/初级校正。这将显着减少 CPU，但在数学上是不正确的。要获得与 colormatrix 过滤器生成的输出兼容的输出，请使用 fast=1。

抖动

指定抖动模式。

接受的值是：

'无'

无抖动

'外频'

Floyd-Steinberg 抖动

wpadapt

白点适配模式。

接受的值是：

'布拉德福德'

布拉德福德白点适应

'冯克斯'

von Kries 白点自适应

'身份'

身份白点适配（即无白点适配）

艾尔

一次覆盖所有输入属性。与[all](#all)相同的接受值。

空间

覆盖输入色彩空间。与[space](#space)相同的接受值。

初选

覆盖输入颜色原色。与[primaries](#primaries)相同的接受值。

国际贸易委员会

覆盖输入传输特性。接受的值与[trc相同](#trc)。

橙色

覆盖输入颜色范围。与[range](#range)相同的接受值。

滤镜将传输特性、色彩空间和原色转换为指定的用户值。如果未指定，输出值将设置为基于“all”属性的默认值。如果也未指定该属性，则过滤器将记录错误。输出颜色范围和格式默认为与输入颜色范围和格式相同的值。应在输入数据上设置输入传输特性、颜色空间、颜色原色和颜色范围。如果缺少其中任何一个，过滤器将记录错误并且不会进行任何转换。

例如，要将输入转换为 SMPTE-240M，请使用以下命令：

色彩空间=smpte240m

### 39.40 colorspace\_cuda

CUDA 加速了色彩空间过滤器的实现。

与软件色彩空间滤镜相比，它的功能并不完整，目前仅支持 jpeg/full 和 mpeg/limited range 之间的色彩范围转换。

过滤器接受以下选项：

范围

指定输出颜色范围。

接受的值是：

'电视'

电视（受限）范围

' mpeg '

MPEG（限制）范围

'电脑'

PC（全）系列

' jpeg '

JPEG（完整）范围

### 39.41色温

调整视频中的色温以模拟环境色温的变化。

过滤器接受以下选项：

温度

以开尔文为单位设置温度。允许的范围是 1000 到 40000。默认值为 6500 K。

混合

设置混合过滤输出。允许的范围是从 0 到 1。默认值为 1。

PL

设置保持亮度的量。允许的范围是从 0 到 1。默认值为 0。

* [命令](#Commands-73)

#### 39.41.1 命令

此过滤器支持与选项相同的[命令](#commands)。

### 39.42卷积

应用 3x3、5x5、7x7 或水平/垂直最多 49 个元素的卷积。

过滤器接受以下选项：

0米

1米

2米

3米

为每个平面设置矩阵。矩阵在方模式下是 9、25 或 49 个有符号整数的序列，在行模式下是 1 到 49 个奇数个有符号整数的序列。

0rdiv

1rdiv

2rdiv

第三分区

为每个平面的计算值设置乘数。如果未设置或为 0，它将是所有矩阵元素的总和。

0偏差

1偏差

2偏差

3偏置

为每个平面设置偏差。该值被添加到乘法的结果。用于使整体图像更亮或更暗。默认值为 0.0。

0模式

1模式

2模式

3模式

为每个平面设置矩阵模式。可以是正方形、行或列。默认为正方形。

* [命令](#Commands-74)
* [例子](#Examples-97)

#### 39.42.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

#### 39.42.2 例子

* 应用锐化：
* 卷积=“0 -1 0 -1 5 -1 0 -1 0:0 -1 0 -1 5 -1 0 -1 0:0 -1 0 -1 5 -1 0 -1 0:0 -1 0 - 1 5 -1 0 -1 0"
* 应用模糊：
* 卷积="1 1 1 1 1 1 1 1 1:1 1 1 1 1 1 1 1 1:1 1 1 1 1 1 1 1 1:1 1 1 1 1 1 1 1 1:1/9:1/9 :1/9:1/9"
* 应用边缘增强：
* 卷积=“0 0 0 -1 1 0 0 0 0:0 0 0 -1 1 0 0 0 0:0 0 0 -1 1 0 0 0 0:0 0 0 -1 1 0 0 0 0:5:1 :1:1:0:128:128:128"
* 应用边缘检测：
* 卷积=“0 1 0 1 -4 1 0 1 0:0 1 0 1 -4 1 0 1 0:0 1 0 1 -4 1 0 1 0:0 1 0 1 -4 1 0 1 0:5:5 :5:1:0:128:128:128"
* 应用包括对角线的拉普拉斯边缘检测器：
* 卷积=“1 1 1 1 -8 1 1 1 1:1 1 1 1 -8 1 1 1 1:1 1 1 1 -8 1 1 1 1:1 1 1 1 -8 1 1 1 1:5:5 :5:1:0:128:128:0"
* 应用浮雕：
* 卷积="-2 -1 0 -1 1 1 0 1 2:-2 -1 0 -1 1 1 0 1 2:-2 -1 0 -1 1 1 0 1 2:-2 -1 0 -1 1 1 0 1 2"

### 39.43卷积

使用第二个流作为脉冲在频域中应用视频流的二维卷积。

过滤器接受以下选项：

飞机

设置要处理的平面。

冲动

设置将处理哪些脉冲视频帧，可以是first或all 。默认为全部。

卷积滤波器还支持[帧同步](#framesync)选项。

### 39.44 副本

将输入视频源原样复制到输出。这主要用于测试目的。

### 39.45 核心图像

在 OSX 上使用 Apple 的 CoreImage API 在 GPU 上进行视频过滤。

硬件加速基于 OpenGL 上下文。通常，这意味着它是由视频硬件处理的。但是，存在基于软件的 OpenGL 实现，这意味着无法保证硬件处理。这取决于各自的 OSX。

Apple 提供了许多滤镜和图像生成器，它们具有多种选择。过滤器必须通过其名称及其选项来引用。

coreimage 过滤器接受以下选项：

列表过滤器

列出所有可用的过滤器和生成器及其各自的选项以及可能的最小值和最大值以及默认值。

list\_filters=真

筛选

按各自的名称和选项指定所有过滤器。使用list\_filters确定所有有效的过滤器名称和选项。数字选项由浮点值指定，并自动限制在它们各自的值范围内。矢量和颜色选项必须由空格分隔的浮点值列表指定。必须进行字符转义。一个特殊的选项名称default可用于使用过滤器的默认选项。

需要指定默认值或至少一个过滤器选项。所有省略的选项都使用它们的默认值。过滤器字符串的语法如下：

filter=<NAME>@<OPTION>=<VALUE>[@<OPTION>=<VALUE>][@...][#<NAME>@<OPTION>=<VALUE>[@<OPTION>=<VALUE >][@...]][#...]

输出矩形

指定一个矩形，将滤镜链的输出复制到输入图像中。它由空格分隔的浮点值列表给出：

output\_rect=x\ y\ width\ height

如果未给出，则输出矩形等于输入图像的尺寸。输出矩形在输入图像的边界处自动裁剪。负值对每个组件都有效。

output\_rect=25\ 25\ 100\ 100

多个过滤器可以链接起来进行连续处理，而无需 GPU-HOST 传输，从而可以快速处理复杂的过滤器链。目前，仅支持具有零个（生成器）或恰好一个（过滤器）输入图像和一个输出图像的过滤器。此外，过渡过滤器还不能按预期使用。

一些过滤器根据各自的过滤器内核生成带有额外填充的输出图像。自动删除填充以确保过滤器输出与输入图像具有相同的大小。

对于图像生成器，输出图像的大小分别由过滤器链的前一个输出图像或整个过滤器链的输入图像决定。生成器不使用此图像的像素信息来生成其输出。但是，生成的输出会混合到该图像上，从而导致部分或完全覆盖输出图像。

coreimagesrc视频源可用于生成直接输入过滤器链的输入图像[。](#coreimagesrc)通过使用它，不需要由另一个视频源或输入视频提供输入图像。

* [例子](#Examples-98)

#### 39.45.1 例子

* 列出所有可用的过滤器：
* 核心图像=列表过滤器=真
* 使用具有默认选项的 CIBoxBlur 过滤器模糊图像：
* coreimage=filter=CIBoxBlur@default
* 使用 CISepiaTone 默认值和 CIVignetteEffect 的过滤器链，其中心为 100x100，半径为 50 像素：
* coreimage=filter=CIBoxBlur@default#CIVignetteEffect@inputCenter=100\ 100@inputRadius=50
* 使用 nullsrc 和 CIQRCodeGenerator 为 FFmpeg 主页创建 QR 代码，作为 Apple 标准 bash shell 的完整转义命令行给出：
* ffmpeg -f lavfi -i nullsrc=s=100x100,coreimage=filter=CIQRCodeGenerator@inputMessage=https\\\\\://FFmpeg.org/@inputCorrectionLevel=H -frames:v 1 QRCode.png

### 39.46 更正

获得两个输入视频之间的相关性。

这个过滤器需要两个输入视频。

两个输入视频必须具有相同的分辨率和像素格式，此过滤器才能正常工作。它还假设两个输入具有相同数量的帧，将它们一一比较。

获得的每个组件、平均值、最小和最大相关性通过日志系统打印。

过滤器将计算出的每个帧的相关性存储在帧元数据中。

此过滤器还支持[帧同步](#framesync)选项。

在下面的示例中，将正在处理的输入文件main.mpg与参考文件ref.mpg进行比较。

ffmpeg -i main.mpg -i ref.mpg -lavfi corr -f null -

### 39.47 cover\_rect

覆盖一个矩形物体

它接受以下选项：

覆盖

可选封面图像的文件路径，需要在 yuv420 中。

模式

设置覆盖模式。

它接受以下值：

'封面'

用提供的图像覆盖它

'模糊'

通过插入周围的像素来覆盖它

默认值为模糊。

* [例子](#Examples-99)

#### 39.47.1 例子

* ffmpeg通过给定视频的提供图像覆盖矩形对象：
* ffmpeg -i 文件.ts -vf find\_rect=newref.pgm,cover\_rect=cover.jpg:mode=cover new.mkv

### 39.48 作物

将输入视频裁剪到给定尺寸。

它接受以下参数：

w,out\_w

输出视频的宽度。它默认为iw 。此表达式仅在过滤器配置期间或发送“ w ”或“ out\_w ”命令时评估一次。

h,out\_h

输出视频的高度。它默认为ih 。此表达式仅在过滤器配置期间或发送“ h ”或“ out\_h ”命令时评估一次。

X

输入视频中输出视频左边缘的水平位置。它默认为(in\_w-out\_w)/2 。该表达式是按帧计算的。

是

输出视频上边缘在输入视频中的垂直位置。它默认为(in\_h-out\_h)/2 。该表达式是按帧计算的。

保持纵横比

如果设置为 1 将强制输出显示纵横比与输入相同，通过更改输出样本纵横比。它默认为 0。

精确的

启用精确裁剪。如果启用，子采样视频将按照指定的精确宽度/高度/x/y 进行裁剪，并且不会四舍五入到最接近的较小值。它默认为 0。

out\_w 、 out\_h 、 x 、 y参数是包含以下常量的表达式：

X

是

x和y的计算值。它们针对每个新帧进行评估。

in\_w

在\_h

输入的宽度和高度。

我

我

in\_w和in\_h相同。

out\_w

out\_h

输出（裁剪）的宽度和高度。

噢

哦

out\_w和out\_h相同。

A

iw / ih相同

特区

输入样本纵横比

达尔

输入显示宽高比，同( iw / ih ) \* sar

hsub

vsub

水平和垂直色度子样本值。例如，对于像素格式“yuv422p”， hsub为 2， vsub为 1。

n

输入帧的编号，从0开始。

位置

输入帧在文件中的位置，如果未知则为 NAN

吨

以秒表示的时间戳。如果输入时间戳未知，则为 NAN。

out\_w的表达式可能取决于out\_h的值，而out\_h的表达式可能取决于out\_w ，但它们不能取决于x和y ，因为x和y在out\_w和out\_h之后求值。

x和y参数指定输出（未裁剪）区域左上角位置的表达式。它们针对每一帧进行评估。如果评估值无效，则将其近似为最接近的有效值。

x的表达式可能取决于y ，而y的表达式可能取决于x 。

* [例子](#Examples-100)
* [命令](#Commands-75)

#### 39.48.1 例子

* 在位置 (12,34) 裁剪大小为 100x100 的区域。
* 裁剪=100:100:12:34

使用命名选项，上面的示例变为：

作物=w=100:h=100:x=12:y=34

* 裁剪尺寸为 100x100 的中央输入区域：
* 作物=100:100
* 将中心输入区域裁剪为输入视频的 2/3 大小：
* 裁剪=2/3\*in\_w:2/3\*in\_h
* 裁剪输入视频中心正方形：
* crop=out\_w=in\_h
* 裁剪=in\_h
* 分隔矩形，左上角位于 100:100 位置，右下角对应于输入图像的右下角。
* 裁剪=in\_w-100:in\_h-100:100:100
* 从左右边框裁剪 10 像素，从上下边框裁剪 20 像素
* 裁剪=in\_w-2\*10:in\_h-2\*20
* 只保留输入图像的右下四分之一：
* 裁剪=in\_w/2:in\_h/2:in\_w/2:in\_h/2
* 获得希腊和谐的作物高度：
* crop=in\_w:1/PHI\*in\_w
* 应用颤抖效果：
* crop=in\_w/2:in\_h/2:(in\_w-out\_w)/2+((in\_w-out\_w)/2)\*sin(n/10):(in\_h-out\_h)/2 +((in\_h-out\_h)/ 2)\*罪恶(n/7)
* 根据时间戳应用不稳定的相机效果：
* crop=in\_w/2:in\_h/2:(in\_w-out\_w)/2+((in\_w-out\_w)/2)\*sin(t\*10):(in\_h-out\_h)/2 +((in\_h-out\_h)/ 2)\*罪恶(t\*13)"
* 根据 y 的值设置 x：
* crop=in\_w/2:in\_h/2:y:10+10\*sin(n/10)

#### 39.48.2 命令

该过滤器支持以下命令：

w,out\_w

h,out\_h

X

是

设置输出视频的宽度/高度和输入视频的水平/垂直位置。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.49 cropdetect

自动检测裁剪尺寸。

它计算必要的裁剪参数并通过日志系统打印推荐参数。根据模式，检测到的尺寸对应于输入视频的非黑色或视频区域。

它接受以下参数：

模式

根据模式裁剪检测基于周围像素的纯黑色值或运动矢量和边缘像素的组合。

'黑色'

检测播放视频周围的黑色像素。对于精细控制，请使用选项limit 。

'消息'

通过视频内部的运动矢量检测正在播放的视频，并扫描通常形成正在播放视频边界的边缘像素。

限制

设置更高的黑色值阈值，可以选择从无 (0) 到一切（对于基于 8 位的格式为 255）指定。大于设定值的强度值被认为是非黑色的。它默认为 24。您还可以指定一个介于 0.0 和 1.0 之间的值，该值将根据像素格式的位深度进行缩放。

圆形的

宽度/高度应被其整除的值。默认为 16。偏移量会自动调整为视频居中。使用 2 仅获得偶数尺寸（4:2:2 视频需要）。编码为大多数视频编解码器时，16 是最好的。

跳过

设置跳过评估的初始帧数。默认值为 2。范围为 0 到 INT\_MAX。

重置计数，重置

设置计数器，确定多少帧后 cropdetect 将重置先前检测到的最大视频区域并重新开始检测当前最佳裁剪区域。默认值为 0。

当频道徽标扭曲视频区域时，这会很有用。 0 表示“永不重置”，并返回播放过程中遇到的最大区域。

mv\_threshold

以像素为单位将运动设置为运动检测的阈值。它默认为 8。

低的

高的

设置 Canny 阈值算法使用的低阈值和高阈值。

高阈值选择“强”边缘像素，然后通过 8-连通性与低阈值选择的“弱”边缘像素连接。

low和high阈值必须在 [0,1] 范围内选择，并且low应小于或等于high 。

low的默认值为5/255 ， high 的默认值为15/255 。

* [例子](#Examples-101)
* [命令](#Commands-76)

#### 39.49.1 例子

* 查找被黑色边框包围的视频区域：
* ffmpeg -i file.mp4 -vf cropdetect,metadata=mode=print -f null -
* 找到嵌入的视频区域，预先生成运动矢量：
* ffmpeg -i file.mp4 -vf mestimate,cropdetect=mode=mvedges,metadata=mode=print -f null -
* 找到一个嵌入的视频区域，使用来自解码器的运动矢量：
* ffmpeg -flags2 +export\_mvs -i file.mp4 -vf cropdetect=mode=mvedges,metadata=mode=print -f null -

#### 39.49.2 命令

该过滤器支持以下命令：

限制

该命令接受与相应选项相同的语法。如果指定的表达式无效，它会保持其当前值。

### 39.50 球杆

延迟视频过滤直到给定的挂钟时间戳。过滤器首先传递预卷帧数，然后缓冲最多缓冲帧数并等待提示。到达提示后，它转发缓冲帧以及输入的任何后续帧。

该过滤器可用于同步多个 ffmpeg 进程的输出，用于实时输出设备，如 decklink。通过在过滤链和预缓冲帧中放置延迟，该过程可以在达到目标挂钟时间戳后几乎立即将数据传递到输出。

无法保证完美的帧精度，但结果对于某些用例来说已经足够好了。

线索

以微秒为单位的 UNIX 时间戳表示的提示时间戳。默认为 0。

预卷

以秒为单位表示的内容作为预滚动传递的持续时间。默认为 0。

缓冲

在等待以秒表示的提示之前要缓冲的内容的最大持续时间。默认为 0。

### 39.51 曲线

使用曲线应用颜色调整。

此滤镜类似于 Adobe Photoshop 和 GIMP 曲线工具。每个组件（红色、绿色和蓝色）的值由使用平滑曲线相互连接的N 个关键点定义。 x 轴表示输入帧的像素值，y 轴表示要为输出帧设置的新像素值。

(0;0)和(1;1)两点定义。这将创建一条直线，其中每个原始像素值都被“调整”为其自身的值，这意味着图像没有变化。

过滤器允许您重新定义这两点并添加更多。将定义一条新曲线以平滑地通过所有这些新坐标。新定义的点需要在 x 轴上严格递增，并且它们的x和y值必须在[0;1]区间内。曲线是通过使用自然或单调三次样条插值形成的，具体取决于interp选项（默认值： natural ）。自然样条通常产生更平滑的曲线，而单调 ( pchip ) 样条保证指定点之间的过渡是单调的。如果计算出的曲线恰好超出向量空间，则值将被相应地裁剪。

过滤器接受以下选项：

预设

选择一种可用的颜色预设。除了r 、 g 、 b参数之外，还可以使用此选项；在这种情况下，后面的选项优先于预设值。可用的预设是：

'无'

' color\_negative '

'交叉进程'

'较暗'

'增加对比度'

'打火机'

' linear\_contrast '

' medium\_contrast '

'否定'

'强对比'

'复古'

默认为无。

主人，米

设置主关键点。这些点将定义第二遍映射。它有时被称为“亮度”或“值”映射。它可以与r 、 g 、 b或全部一起使用，因为它就像一个后处理 LUT。

红, r

设置红色分量的关键点。

绿色，克

设置绿色组件的关键点。

蓝色，乙

设置蓝色分量的关键点。

全部

为所有组件（不包括master）设置关键点。可以使用除其他关键点组件选项。在这种情况下，未设置的组件将退回到此所有设置。

文件

要从中导入设置的Photoshop 曲线文件 ( .acv )。

阴谋

在指定文件中保存曲线的 Gnuplot 脚本。

插话

指定插值的种类。可用的算法是：

'自然'

使用两次连续可微分的分段三次多项式的自然三次样条。

'芯片'

使用分段三次 Hermite 插值多项式 (PCHIP) 的单调三次样条。

为了避免一些过滤图语法冲突，每个关键点列表需要使用以下语法定义： x0/y0 x1/y1 x2/y2 ...。

* [命令](#Commands-77)
* [例子](#Examples-102)

#### 39.51.1 命令

此过滤器支持与选项相同的[命令](#commands)。

#### 39.51.2 例子

* 稍微增加蓝色的中间水平：
* 曲线=蓝色='0/0 0.5/0.58 1/1'
* 复古效果：
* 曲线=r='0/0.11 .42/.51 1/0.95':g='0/0 0.50/0.48 1/1':b='0/0.22 .49/.44 1/0.8'

在这里，我们获得了每个组件的以下坐标：

红色的

(0;0.11) (0.42;0.51) (1;0.95)

绿色的

(0;0) (0.50;0.48) (1;1)

蓝色的

(0;0.22) (0.49;0.44) (1;0.80)

* 前面的示例也可以通过关联的内置预设来实现：
* 曲线=预设=复古
* 或者简单地说：
* 曲线=复古
* 使用 Photoshop 预设并重新定义绿色组件的点：
* 曲线=psfile='MyCurvesPresets/purple.acv':green='0/0 0.45/0.53 1/1'
* 使用ffmpeg和gnuplot检查cross\_process配置文件的曲线：
* ffmpeg -f lavfi -i color -vf curves=cross\_process:plot=/tmp/curves.plt -frames:v 1 -f 空 -
* gnuplot -p /tmp/curves.plt

### 39.52 数据范围

视频数据分析过滤器。

此过滤器显示部分视频的十六进制像素值。

过滤器接受以下选项：

大小，小号

设置输出视频大小。

X

设置从何处拾取像素的 x 偏移量。

是

从选择像素的位置设置 y 偏移量。

模式

设置范围模式，可以是以下之一：

'单声道'

在黑色背景上用白色绘制十六进制像素值。

'颜色'

在黑色背景上绘制带有输入视频像素颜色的十六进制像素值。

'颜色2 '

在从输入视频中选取的颜色背景上绘制十六进制像素值，以这种方式选取文本颜色，使其始终可见。

轴

在视频的左侧和顶部绘制行号和列号。

不透明度

设置背景不透明度。

格式

设置显示数字格式。可以是hex或dec 。默认为十六进制。

成分

设置要显示的像素组件。默认情况下显示所有像素组件。

* [命令](#Commands-78)

#### 39.52.1 命令

此过滤器支持与选项相同的[命令，但不包括](#commands)大小选项。

### 39.53 模糊

应用方向模糊滤镜。

过滤器接受以下选项：

角度

设置方向模糊的角度。默认值为45 。

半径

设置方向模糊的半径。默认值为5 。

飞机

设置要过滤的平面。默认情况下，所有平面都被过滤。

* [命令](#Commands-79)

#### 39.53.1 命令

此过滤器支持与选项相同的[命令](#commands)。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.54 dctdnoiz

使用 2D DCT（频域滤波）对帧进行降噪。

此过滤器不是为实时设计的。

过滤器接受以下选项：

σ, s

设置噪声西格玛常数。

这个sigma定义了一个硬阈值3 \* sigma ；低于该阈值的每个 DCT 系数（绝对值）都会被丢弃。

如果您需要更高级的过滤，请参阅expr 。

默认为0 。

重叠

为每个块设置重叠像素数。由于过滤器可能很慢，您可能希望降低此值，但代价是过滤器效率降低和出现各种伪像的风险。

如果重叠值不允许处理整个输入宽度或高度，则会显示警告，并且不会对相应的边框进行降噪。

默认值为blocksize -1，这是最佳设置。

表达式，e

设置系数因子表达式。

对于 DCT 块的每个系数，该表达式将被评估为系数的乘数值。

如果设置了这个选项，则sigma选项将被忽略。

系数的绝对值可以通过c变量访问。

n

使用位数设置块大小。 1<< n定义blocksize ，即处理块的宽度和高度。

默认值为3 (8x8)，对于16x16 的块大小可以提高到4 。请注意，更改此设置会对速度处理产生巨大影响。此外，更大的块大小并不一定意味着更好的去噪。

* [例子](#Examples-103)

#### 39.54.1 例子

sigma为4.5的降噪：

dctdnoiz=4.5

使用表达式系统可以实现相同的操作：

dctdnoiz=e='gte(c, 4.5\*3)'

16x16块大小的暴力降噪：

dctdnoiz=15:n=4

### 39.55 解带

从输入视频中移除条带伪像。它的工作原理是用参考像素的平均值替换带状像素。

过滤器接受以下选项：

1th

第二名

3th

4th

为每个平面设置条带检测阈值。默认值为 0.02。有效范围是 0.00003 到 0.5。如果当前像素与参考像素之间的差异小于阈值，则将其视为带状。

范围，r

以像素为单位的条带检测范围。默认为 16。如果为正数，将使用 0 到设置值范围内的随机数。如果为负，将使用精确的绝对值。该范围定义了当前像素周围四个像素的正方形。

方向，d

以弧度为单位设置方向，从中比较四个像素。如果为正，将选取从 0 到设置方向的随机方向。如果为负，将选择绝对值的精确值。例如，方向 0、-PI 或 -2\*PI 弧度将仅选取同一行上的像素，而 -PI/2 将仅选取同一列上的像素。

模糊，b

如果启用，则将当前像素与所有四个周围像素的平均值进行比较。默认是启用的。如果禁用，则将当前像素与所有四个周围像素进行比较。如果仅与周围像素的所有四个差异都小于阈值，则该像素被认为是带状的。

联轴器，c

如果启用，当且仅当所有像素分量都带状时，当前像素才会更改，例如，为所有颜色分量触发带状检测阈值。默认是禁用的。

* [命令](#Commands-80)

#### 39.55.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.56 解块

从输入视频中移除块伪影。

过滤器接受以下选项：

筛选

设置过滤器类型，可以是weak或strong 。默认是strong 。这控制应用哪种去块效应。

堵塞

设置块的大小，允许的范围是从 4 到 512。默认是8 。

α

贝塔

伽马

三角洲

设置阻塞检测阈值。允许的范围是 0 到 1。默认值是： alpha为0.098 ，其余为0.05 。使用更高的阈值可提供更大的去块强度。设置alpha控制块精确边缘的阈值检测。其余选项控制边缘附近的阈值检测。每个用于下方/上方或左侧/右侧。将其中任何一个设置为0将禁用去块。

飞机

设置要过滤的平面。默认是过滤所有可用的平面。

* [例子](#Examples-104)
* [命令](#Commands-81)

#### 39.56.1 例子

* 使用弱过滤器和 4 像素的块大小去块。
* 去块=过滤器=弱：块=4
* 使用强过滤器去块，块大小为 4 像素和自定义阈值以去块更多边缘。
* deblock=filter=strong:block=4:alpha=0.12:beta=0.07:gamma=0.06:delta=0.05
* 与上面类似，但只过滤第一个平面。
* deblock=filter=strong:block=4:alpha=0.12:beta=0.07:gamma=0.06:delta=0.05:planes=1
* 与上面类似，但只过滤第二个和第三个平面。
* deblock=filter=strong:block=4:alpha=0.12:beta=0.07:gamma=0.06:delta=0.05:planes=6

#### 39.56.2 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.57 抽取

定期删除重复的帧。

过滤器接受以下选项：

循环

设置将从中删除一帧的帧数。将此设置为N意味着将丢弃每批N帧中的一帧。默认值为5 。

双阈值

设置重复检测的阈值。如果一个帧的差异度量小于或等于这个值，那么它被声明为重复的。默认为1.1

阈值

设置场景变化阈值。默认值为15 。

块x

块状的

设置度量计算期间使用的 x 轴和 y 轴块的大小。较大的块可以更好地抑制噪声，但对小运动的检测也更差。必须是二的幂。默认值为32 。

ppsrc

将主输入标记为预处理输入并激活干净的源输入流。这允许使用各种过滤器对输入进行预处理，以帮助度量计算，同时保持帧选择无损。当设置为1 时，第一个流用于预处理输入，第二个流是从中选择保留帧的干净源。默认为0 。

色度

设置是否在度量计算中考虑色度。默认为1 。

混合的

设置输入是否仅部分包含要抽取的内容。默认为假。如果启用，视频输出流将采用可变帧速率。

### 39.58 去卷积

使用第二个流作为脉冲在频域中应用视频流的 2D 反卷积。

过滤器接受以下选项：

飞机

设置要处理的平面。

冲动

设置将处理哪些脉冲视频帧，可以是first或all 。默认为全部。

噪音

做除法时设置噪音。默认值为0.0000001 。当宽度和高度不相同且不是 2 的幂或者如果卷积之前的流有噪声时很有用。

反卷积过滤器还支持[帧同步](#framesync)选项。

### 39.59分

减少视频中的交叉亮度（点爬行）和交叉颜色（彩虹）。

它接受以下选项：

米

设置操作模式。可以结合dotcrawl以减少交叉亮度和/或彩虹以减少交叉颜色。

lt

设置空间亮度阈值。较低的值会增加交叉亮度的减少。

时间

设置时间亮度的公差。较高的值会增加交叉亮度的减少。

时间

设置色度时间变化的公差。较高的值会增加串色的减少。

ct

设置时间色度阈值。较低的值会增加串色的减少。

### 39.60放气

对视频应用缩小效果。

该过滤器通过仅考虑低于像素的值，将像素替换为局部 (3x3) 平均值。

它接受以下选项：

阈值0

阈值1

阈值2

阈值3

限制每个平面的最大变化，默认为 65535。如果为 0，平面将保持不变。

* [命令](#Commands-82)

#### 39.60.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.61 去闪烁

删除时间帧亮度变化。

它接受以下选项：

大小，小号

以帧为单位设置移动平均滤波器大小。默认值为 5。允许的范围为 2 - 129。

模式，米

设置平均模式以平滑时间亮度变化。

可用值是：

'是'

算术平均值

'通用汽车'

几何平均数

‘嗯’

调和平均值

' qm '

二次均值

'厘米'

三次均值

'下午'

功率平均值

'中位数'

中位数

旁路

不要实际修改框架。当只需要元数据时很有用。

### 39.62 抖动

消除由部分隔行扫描的电视电影内容产生的抖动。

例如，可以通过[上拉](#pullup)滤波器引入抖动。如果原始源是部分电视电影内容，则pullup,dejudder的输出将具有可变帧速率。可能会改变容器的记录帧率。除了那个变化，这个过滤器不会影响恒定帧率视频。

此过滤器中可用的选项是：

循环

指定抖动重复的窗口长度。

接受任何大于 1 的整数。有用的值是：

' 4 '

如果原件以 24 到 30 fps（电影到 NTSC）进行电视电影化。

' 5 '

如果原件是从 25 到 30 fps（PAL 到 NTSC）进行电视电影化的。

' 20 '

如果两者混合。

默认值为“ 4 ”。

### 39.63 标志

通过周围像素的简单插值来抑制电视台徽标。只需设置一个覆盖徽标的矩形并观察它消失（有时甚至会出现更丑陋的东西 - 你的里程可能会有所不同）。

它接受以下参数：

X

是

指定徽标的左上角坐标。必须指定它们。

w

H

指定要清除的徽标的宽度和高度。必须指定它们。

展示

设置为 1 时，会在屏幕上绘制一个绿色矩形以简化查找正确的x 、 y 、 w和h参数的过程。默认值为 0。

矩形绘制在最外面的像素上，这些像素将（部分）替换为插值。在每个方向紧邻该矩形之外的下一个像素值将用于计算矩形内的内插像素值。

* [例子](#Examples-105)

#### 39.63.1 例子

* 设置一个覆盖区域的矩形，其左上角坐标为 0,0，大小为 100x77：
* delogo=x=0:y=0:w=100:h=77

### 39.64 脱雨

通过应用基于卷积神经网络的去雨方法，去除输入图像/视频中的雨水。支持机型：

* 循环挤压和激发上下文聚合网络 (RESCAN)。参见<http://openaccess.thecvf.com/content_ECCV_2018/papers/Xia_Li_Recurrent_Squeeze-and-Excitation_Context_ECCV_2018_paper.pdf>。

<https://github.com/XueweiMeng/derain_filter.git>的存储库中提供了训练和模型生成脚本。

可以使用 tools/python/convert.py 从 TensorFlow 模型文件 (.pb) 生成原生模型文件 (.model)

过滤器接受以下选项：

过滤类型

指定要使用的过滤器。此选项接受以下值：

'下雨'

排水过滤器。要进行除雨过滤，您需要使用除雨模型。

'除霾'

除雾过滤器。要进行去雾过滤，您需要使用去雾模型。

默认值为“ derain ”。

dnn\_backend

指定用于模型加载和执行的 DNN 后端。此选项接受以下值：

'本土'

DNN 加载和执行的本机实现。

'张量流'

TensorFlow 后端。要启用此后端，您需要为 C 库安装 TensorFlow（请参阅[https://www.tensorflow.org/install/lang\_c ）并使用](https://www.tensorflow.org/install/lang_c)--enable-libtensorflow配置 FFmpeg

默认值为 ' native '。

模型

设置模型文件的路径，指定网络架构及其参数。请注意，不同的后端使用不同的文件格式。 TensorFlow 和本机后端只能加载其格式的文件。

要获得完整功能（例如异步执行），请使用[dnn\_processing](#dnn_005fprocessing)过滤器。

### 39.65 去抖

尝试修复水平和/或垂直移动的微小变化。此滤镜有助于消除手持相机、碰撞三脚架、在车辆上移动等引起的相机抖动。

过滤器接受以下选项：

X

是

w

H

指定一个矩形区域来限制对运动矢量的搜索。如果需要，可以将对运动矢量的搜索限制在由其左上角、宽度和高度定义的帧的矩形区域。这些参数与 drawbox 过滤器具有相同的含义，可用于可视化边界框的位置。

当运动矢量搜索可能将帧内对象的同时运动与相机运动混淆时，这很有用。

x 、 y 、 w和h中的任何一个或全部设置为 -1，则使用全帧。这允许在不指定运动矢量搜索的边界框的情况下设置以后的选项。

默认 - 搜索整个框架。

收据

赖

在 0-64 像素范围内指定 x 和 y 方向的最大移动范围。默认 16。

边缘

指定如何生成像素以填充帧边缘的空白。可用值是：

'空白，0 '

在空白位置填零

'原始，1 '

空白处的原图

'夹子，2 '

空白位置的拉伸边缘值

'镜子，3 '

空白位置的镜面边缘

默认值为“镜像”。

块大小

指定用于运动搜索的块大小。范围 4-128 像素，默认 8。

对比

指定块的对比度阈值。只有具有超过指定对比度（最暗和最亮像素之间的差异）的块才会被考虑。范围 1-255，默认 125。

搜索

指定搜索策略。可用值是：

'详尽无遗，0 '

设置穷举搜索

'少，1 '

设置不太详尽的搜索。

默认值为“详尽”。

文件名

如果设置，则运动搜索的详细日志将写入指定文件。

### 39.66 倒溢

去除由绿屏或蓝屏的反射颜色引起的前景颜色的不需要的污染。

此过滤器接受以下选项：

类型

设置要使用的去溢出类型。

混合

设置如何生成溢出图。

扩张

设置去除剩余溢出物的量。

红色的

控制溢出区域中的红色量。

绿色的

控制溢出区域中的绿色量。绿屏应为 -1。

蓝色的

控制溢出区域中的蓝色量。蓝屏应该是-1。

亮度

控制溢出区域的亮度，保留颜色。

α

从生成的溢出图中修改 alpha。

* [命令](#Commands-83)

#### 39.66.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.67 删除

应用电视电影操作的精确逆运算。它需要使用模式选项指定的预定义模式，该模式必须与传递给电视电影过滤器的模式相同。

此过滤器接受以下选项：

first\_field

'顶， '

顶场优先

'底部，b '

bottom field first 默认值为top 。

图案

一串数字，代表您希望应用的下拉模式。默认值为23 。

起始帧

表示第一帧相对于电视电影模式的位置的数字。如果流被切断，将使用它。默认值为0 。

### 39.68 膨胀

对视频应用膨胀效果。

此过滤器用局部 (3x3) 最大值替换像素。

它接受以下选项：

阈值0

阈值1

阈值2

阈值3

限制每个平面的最大变化，默认为 65535。如果为 0，平面将保持不变。

坐标

指定要引用的像素的标志。默认值为 255，即使用所有八个像素。

标记到本地 3x3 坐标映射，如下所示：

1 2 3 4 5 6 7 8

* [命令](#Commands-84)

#### 39.68.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.69 排量

按照第二个和第三个输入流的指示置换像素。

它需要三个输入流并输出一个流，第一个输入是源，第二个和第三个输入是置换贴图。

第二个输入指定沿 x 轴移动多少像素，而第三个输入指定沿 y 轴移动多少像素。如果其中一个置换贴图流终止，则将使用该置换贴图的最后一帧。

请注意，一旦生成，置换贴图就可以反复使用。

接受的选项的描述如下。

边缘

为超出范围的像素设置置换行为。

可用值是：

'空白'

缺失的像素被黑色像素取代。

'涂抹'

相邻的像素将散开以替换缺失的像素。

'包裹'

超出范围的像素被包裹起来，因此它们指向另一侧的像素。

'镜子'

超出范围的像素将被镜像像素替换。

默认为“涂抹”。

* [例子](#Examples-106)

#### 39.69.1 例子

* 为视频尺寸 hd720 的 rgb 输入添加波纹效果：
* ffmpeg -i 输入 -f lavfi -i nullsrc=s=hd720,lutrgb=128:128:128 -f lavfi -i nullsrc=s=hd720,geq='r=128+30\*sin(2\*PI\*X/ 400+T):g=128+30\*sin(2\*PI\*X/400+T):b=128+30\*sin(2\*PI\*X/400+T)'-lavfi'[0][ 1][2]取代'输出
* 将波浪效果添加到视频大小 hd720 的 rgb 输入：
* ffmpeg -i 输入 -f lavfi -i nullsrc=hd720,geq='r=128+80\*(sin(sqrt((XW/2)\*(XW/2)+(YH/2)\*(YH/2) )/220\*2\*PI+T)):g=128+80\*(sin(sqrt((XW/2)\*(XW/2)+(YH/2)\*(YH/2))/220\* 2\*PI+T)):b=128+80\*(sin(sqrt((XW/2)\*(XW/2)+(YH/2)\*(YH/2))/220\*2\*PI+ T))' -lavfi '[1]split[x][y],[0][x][y]displace' 输出

### 39.70 dnn\_classify

使用基于边界框的深度神经网络进行分类。

过滤器接受以下选项：

dnn\_backend

指定用于模型加载和执行的 DNN 后端。此选项现在仅接受 openvino，将添加 tensorflow 后端。

模型

设置模型文件的路径，指定网络架构及其参数。请注意，不同的后端使用不同的文件格式。

输入

设置 dnn 网络的输入名称。

输出

设置 dnn 网络的输出名称。

信心

设置置信度阈值（默认值：0.5）。

标签

设置标签文件的路径，指定标签 ID 和名称之间的映射。每个标签名称写在一行中，跳过尾部空格和空行。第一行是label id 0的名称，第二行是label id 1的名称，以此类推。如果没有提供label文件，则以label id为name。

后端配置

设置要传递到后端的配置

sess\_config选项设置其配置，请使用 tools/python/tf\_sess\_config.py 获取您系统的配置。

### 39.71 dnn\_detect

使用深度神经网络进行对象检测。

过滤器接受以下选项：

dnn\_backend

指定用于模型加载和执行的 DNN 后端。此选项现在仅接受 openvino，将添加 tensorflow 后端。

模型

设置模型文件的路径，指定网络架构及其参数。请注意，不同的后端使用不同的文件格式。

输入

设置 dnn 网络的输入名称。

输出

设置 dnn 网络的输出名称。

信心

设置置信度阈值（默认值：0.5）。

标签

设置标签文件的路径，指定标签 ID 和名称之间的映射。每个标签名称写在一行中，跳过尾部空格和空行。第一行是label id 0的名称（一般是'background'），第二行是label id 1的名称，以此类推。如果没有提供label文件，则以label id为name。

后端配置

设置要传递到后端的配置。要使用异步执行，请设置异步（默认值：设置）。如果后端不支持异步，回滚到同步执行。

### 39.72 dnn\_processing

使用深度神经网络进行图像处理。它与另一个过滤器一起工作，将帧的像素格式转换为 dnn 网络所需的格式。

过滤器接受以下选项：

dnn\_backend

指定用于模型加载和执行的 DNN 后端。此选项接受以下值：

'本土'

DNN 加载和执行的本机实现。

'张量流'

TensorFlow 后端。要启用此后端，您需要为 C 库安装 TensorFlow（请参阅[https://www.tensorflow.org/install/lang\_c ）并使用](https://www.tensorflow.org/install/lang_c)--enable-libtensorflow配置 FFmpeg

'开维诺'

打开 VINO 后端。要启用此后端，您需要为 C 库构建和安装 OpenVINO（请参阅<https://github.com/openvinotoolkit/openvino/blob/master/build-instruction.md>）并使用--enable-libopenvino （– extra-cflags=-I... –extra-ldflags=-L... 如果头文件和库未安装到系统路径中，则可能需要）

默认值为 ' native '。

模型

设置模型文件的路径，指定网络架构及其参数。请注意，不同的后端使用不同的文件格式。 TensorFlow、OpenVINO 和本机后端只能加载其格式的文件。

可以使用 tools/python/convert.py 从 TensorFlow 模型文件 (.pb) 生成原生模型文件 (.model)

输入

设置 dnn 网络的输入名称。

输出

设置 dnn 网络的输出名称。

后端配置

设置要传递到后端的配置。要使用异步执行，请设置异步（默认值：设置）。如果后端不支持异步，回滚到同步执行。

sess\_config选项设置其配置，请使用 tools/python/tf\_sess\_config.py 为您的系统获取 TensorFlow 后端的配置。

* [例子](#Examples-107)

#### 39.72.1 例子

* 使用 can.pb 去除 rgb24 帧中的雨水（请参阅[derain](#derain)过滤器）：
* ./ffmpeg -i rain.jpg -vf format=rgb24,dnn\_processing=dnn\_backend=tensorflow:model=can.pb:input=x:output=y derain.jpg
* 将格式为 gray32f 的帧的像素值减半：
* ffmpeg -i input.jpg -vf format=grayf32,dnn\_processing=model=halve\_gray\_float.model:input=dnn\_in:output=dnn\_out:dnn\_backend=native -y out.native.png
* yuv420p（支持平面 YUV 格式）的帧使用 srcnn.pb（参见[sr过滤器）](#sr)处理 Y 通道：
* ./ffmpeg -i 480p.jpg -vf format=yuv420p,scale=w=iw\*2:h=ih\*2,dnn\_processing=dnn\_backend=tensorflow:model=srcnn.pb:input=x:output=y -y srcnn .jpg
* [sr过滤器）](#sr)处理 Y 通道，它会改变帧大小，对于格式 yuv420p（支持平面 YUV 格式），请使用 tools/python/tf\_sess\_config.py 为您的系统获取 TensorFlow 后端的配置。
* ./ffmpeg -i 480p.jpg -vf format=yuv420p,dnn\_processing=dnn\_backend=tensorflow:model=espcn.pb:input=x:output=y:backend\_configs=sess\_config=0x10022805320e09cdccccccccccec3f20012a01303801 -y tmp.espcn。 jpg

### 39.73 抽屉箱

在输入图像上绘制一个彩色框。

它接受以下参数：

X

是

指定框左上角坐标的表达式。它默认为 0。

宽度，w

身高，小时

指定框的宽度和高度的表达式；如果为 0，则它们被解释为输入宽度和高度。它默认为 0。

颜色，c

指定要写入的框的颜色。有关此选项的一般语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。如果使用特殊值invert ，则框边缘颜色与具有反转亮度的视频相同。

厚度，吨

设置框边缘厚度的表达式。 fill值将创建一个填充框。默认值为3 。

请参阅下面的可接受常量列表。

代替

如果输入具有 alpha，则适用。值为1时，绘制框的像素将覆盖视频的颜色和 alpha 像素。默认值为0 ，它将框合成到输入上，保持视频的 alpha 不变。

x 、 y 、 w和h和t的参数是包含以下常量的表达式：

达尔

输入显示的纵横比，等同于 ( w / h ) \* sar 。

hsub

vsub

水平和垂直色度子样本值。例如，对于像素格式“yuv422p”， hsub为 2， vsub为 1。

在\_h，我

in\_w, iw

输入的宽度和高度。

特区

输入样本纵横比。

X

是

绘制框的 x 和 y 偏移坐标。

w

H

绘制框的宽度和高度。

box\_source

如果你想在side data的detection bboxes中使用box data，box source可以设置为side\_data\_detection\_bboxes。

如果设置了box\_source ，则x 、 y 、 width和height将被忽略，并且仍然在侧数据的检测 bboxes 中使用 box 数据。因此，如果您不确定盒子来源，请不要使用此参数。

吨

绘制框的厚度。

这些常量允许x 、 y 、 w 、 h和t表达式相互引用，因此您可以指定y=x/dar或h=w/dar 。

* [例子](#Examples-108)
* [命令](#Commands-85)

#### 39.73.1 例子

* 在输入图像的边缘画一个黑框：
* 抽屉
* 绘制一个红色框，不透明度为 50%：
* 抽屉=10:20:200:60:red@0.5

前面的示例可以指定为：

画框=x=10:y=20:w=200:h=60:color=red@0.5

* 用粉红色填充框：
* 画框=x=10:y=10:w=100:h=100:color=pink@0.5:t=fill
* 绘制一个 2 像素的红色 2.40:1 遮罩：
* drawbox=x=-t:y=0.5\*(ih-iw/2.4)-t:w=iw+t\*2:h=iw/2.4+t\*2:t=2:c=red

#### 39.73.2 命令

此过滤器支持与选项相同的命令。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.74 绘图

使用输入视频元数据绘制图形。

它接受以下参数：

立方米

设置第一帧元数据键，元数据值将用于绘制图形。

fg1

设置第一个前景颜色表达式。

平方米

设置第二帧元数据键，元数据值将用于绘制图形。

fg2

设置第 2 个前景颜色表达式。

立方米

设置第 3 帧元数据键，元数据值将用于绘制图形。

fg3

设置第 3 个前景颜色表达式。

立方米

设置第 4 帧元数据键，元数据值将用于绘制图形。

fg4

设置第 4 个前景颜色表达式。

分钟

设置元数据值的最小值。

最大限度

设置元数据值的最大值。

背景

设置图形背景颜色。默认为白色。

模式

设置图形模式。

模式的可用值为：

'酒吧'

'点'

'线'

默认是行。

滑动

设置幻灯片模式。

slide 的可用值是：

'框架'

到达右边界时绘制新框架。

'替换'

用新列替换旧列。

'滚动'

从右向左滚动。

'滚动'

从左向右滚动。

'图片'

画单张图。

默认为框架。

尺寸

设置图形视频的大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为900x256 。

率，r

设置输出帧率。默认值为25 。

前景色表达式可以使用以下变量：

最小值

元数据值的最小值。

最大限度

元数据值的最大值。

值

当前元数据键值。

颜色定义为 0xAABBGGRR。

使用来自[signalstats](#signalstats)过滤器的元数据的示例：

signalstats,drawgraph=lavfi.signalstats.YAVG:min=0:max=255

[ebur128过滤器](#ebur128)的元数据的示例：

ebur128=metadata=1,adrawgraph=lavfi.r128.M:min=-120:max=5

### 39.75 拉网

在输入图像上绘制网格。

它接受以下参数：

X

是

指定网格相交点坐标的表达式（用于配置偏移量）。两者都默认为 0。

宽度，w

身高，小时

指定网格单元宽度和高度的表达式，如果为 0，则它们分别被解释为输入宽度和高度减去thickness ，因此图像被框起来。默认为 0。

颜色，c

指定网格的颜色。有关此选项的一般语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。如果使用特殊值invert ，则网格颜色与具有反转亮度的视频相同。

厚度，吨

设置网格线粗细的表达式。默认值为1 。

请参阅下面的可接受常量列表。

代替

如果输入具有 alpha，则适用。使用1时，绘制网格的像素将覆盖视频的颜色和 alpha 像素。默认值为0 ，它将网格合成到输入上，使视频的 alpha 保持不变。

x 、 y 、 w和h和t的参数是包含以下常量的表达式：

达尔

输入显示的纵横比，等同于 ( w / h ) \* sar 。

hsub

vsub

水平和垂直色度子样本值。例如，对于像素格式“yuv422p”， hsub为 2， vsub为 1。

在\_h，我

in\_w, iw

输入网格单元格的宽度和高度。

特区

输入样本纵横比。

X

是

网格交叉点的 x 和 y 坐标（用于配置偏移量）。

w

H

绘制单元格的宽度和高度。

吨

绘制单元格的厚度。

这些常量允许x 、 y 、 w 、 h和t表达式相互引用，因此您可以指定y=x/dar或h=w/dar 。

* [例子](#Examples-109)
* [命令](#Commands-86)

#### 39.75.1 例子

* 绘制一个单元格为 100x100 像素、厚度为 2 像素、颜色为红色且不透明度为 50% 的网格：
* drawgrid=width=100:height=100:thickness=2:color=red@0.5
* 绘制一个不透明度为 50% 的白色 3x3 网格：
* drawgrid=w=iw/3:h=ih/3:t=2:c=white@0.5

#### 39.75.2 命令

此过滤器支持与选项相同的命令。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.76 绘图文本

使用 libfreetype 库在视频顶部绘制文本字符串或来自指定文件的文本。

要启用此过滤器的编译，您需要使用--enable-libfreetype配置 FFmpeg 。要启用默认字体回退和字体选项，您需要使用--enable-libfontconfig配置 FFmpeg 。要启用text\_shaping选项，您需要使用--enable-libfribidi配置 FFmpeg 。

* [句法](#Syntax-2)
* [文本扩展](#Text-expansion)
* [命令](#Commands-87)
* [例子](#Examples-110)

#### 39.76.1 语法

它接受以下参数：

盒子

用于使用背景颜色在文本周围绘制一个框。该值必须是 1（启用）或 0（禁用）。框的默认值为0。

框框w

boxcolor设置要在框周围绘制的边框宽度。 boxborderw的默认值为0。

盒子颜色

用于在文本周围绘制框的颜色。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。

boxcolor的默认值为“白色”。

行间距

box设置要在框周围绘制的边框的行间距（以像素为单位） 。 line\_spacing的默认值为0。

边框w

bordercolor设置要在文本周围绘制的边框宽度。 borderw的默认值为0。

边框颜色

设置用于在文本周围绘制边框的颜色。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。

bordercolor的默认值为“黑色”。

扩张

选择文本的展开方式。可以是none 、 strftime （已弃用）或normal （默认）。有关详细信息，请参阅下面的[文本扩展](#drawtext_005fexpansion)部分。

基准时间

设置计数的开始时间。值以微秒为单位。仅适用于已弃用的 strftime 扩展模式。要在正常扩展模式下进行仿真，请使用pts函数，提供开始时间（以秒为单位）作为第二个参数。

固定边界

如果为真，请检查并修复文本坐标以避免剪裁。

字体颜色

用于绘制字体的颜色。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。

fontcolor的默认值为“黑色”。

字体颜色表达式

文本一样展开的字符串，获取动态字体颜色值。默认情况下，此选项具有空值且不被处理。设置此选项后，它会覆盖fontcolor选项。

字体

用于绘制文本的字体系列。默认Sans。

字体文件

用于绘制文本的字体文件。必须包括路径。如果禁用 fontconfig 支持，则此参数是必需的。

α

绘制应用 alpha 混合的文本。该值可以是 0.0 到 1.0 之间的数字。该表达式也接受相同的变量x、y 。默认值为 1。请参阅fontcolor\_expr 。

字体大小

用于绘制文本的字体大小。 fontsize的默认值为16。

文本整形

如果设置为 1，则在绘制文本之前尝试调整文本形状（例如，反转从右到左文本的顺序并连接阿拉伯字符）。否则，只需完全按照给定的方式绘制文本。默认为 1（如果支持）。

ft\_load\_flags

用于加载字体的标志。

标志映射 libfreetype 支持的相应标志，并且是以下值的组合：

默认

no\_scale

没有暗示

使成为

无位图

垂直布局

强制自动提示

crop\_bitmap

迂腐的

ignore\_global\_advance\_width

无递归

忽略转换

单色

线性设计

没有自动提示

默认值为“默认”。

有关详细信息，请参阅 FT\_LOAD\_\* libfreetype 标志的文档。

阴影颜色

用于在绘制的文本后面绘制阴影的颜色。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。

shadowcolor的默认值为“黑色”。

影子x

阴暗的

文本阴影位置相对于文本位置的 x 和 y 偏移量。它们可以是正值或负值。两者的默认值为“0”。

起始编号

n/frame\_num 变量的起始帧编号。默认值为“0”。

标签大小

用于呈现选项卡的空格数大小。默认值为 4。

时间码

以“hh:mm:ss[:;.]ff”格式设置初始时间码表示。它可以与或不与文本参数一起使用。必须指定timecode\_rate选项。

timecode\_rate, 速率, r

设置时间码帧速率（仅限时间码）。值将四舍五入为最接近的整数。最小值为“1”。帧速率 30 和 60 支持丢帧时间码。

tc24hmax

如果设置为 1，时间码选项的输出将以 24 小时为单位循环。默认值为 0（禁用）。

文本

要绘制的文本字符串。文本必须是 UTF-8 编码字符序列。如果没有使用参数textfile指定文件，则此参数是必需的。

文本文件

包含要绘制的文本的文本文件。文本必须是 UTF-8 编码字符序列。

如果没有使用参数 text 指定文本字符串，则此参数是必需的。

如果同时指定了text和textfile ，则会抛出错误。

文本源

如果要在侧数据的检测 bboxes 中使用文本数据，则应将文本源设置为 side\_data\_detection\_bboxes。

如果设置了文本源， text和textfile将被忽略，并且仍然在侧数据的检测 bboxes 中使用文本数据。因此，如果您不确定文本来源，请不要使用此参数。

重新加载

文本文件将以指定的帧间隔重新加载。确保以原子方式更新文本文件，否则它可能被部分读取，甚至失败。范围是 0 到 INT\_MAX。默认为 0。

X

是

指定在视频帧内绘制文本的偏移量的表达式。它们相对于输出图像的上/左边框。

x和y的默认值为“0”。

请参阅下面的可接受常量和函数列表。

x和y的参数是包含以下常量和函数的表达式：

达尔

输入显示宽高比，同( w / h ) \* sar

hsub

vsub

水平和垂直色度子样本值。例如，对于像素格式“yuv422p”， hsub为 2， vsub为 1。

line\_h, lh

每个文本行的高度

main\_h, h, H

输入高度

main\_w, w, W

输入宽度

max\_glyph\_a，上升

对于所有呈现的字形，从基线到用于放置字形轮廓点的最高/上网格坐标的最大距离。这是一个正值，因为网格的方向是 Y 轴向上。

max\_glyph\_d，下降

对于所有呈现的字形，从基线到用于放置字形轮廓点的最低网格坐标的最大距离。这是一个负值，由于网格的方向，Y 轴向上。

最大字形 h

最大字形高度，即渲染文本中包含的所有字形的最大高度，相当于ascent - descent 。

最大字形宽度

最大字形宽度，即渲染文本中包含的所有字形的最大宽度

n

输入帧数，从0开始

随机数（最小值，最大值）

min和max之间的随机数

特区

输入样本纵横比。

吨

以秒表示的时间戳，如果输入时间戳未知则为 NAN

text\_h, th

渲染文本的高度

text\_w, tw

渲染文本的宽度

X

是

绘制文本的 x 和 y 偏移坐标。

这些参数允许x和y表达式相互引用，因此您可以指定y=x/dar 。

图片类型

当前帧图片类型的单字符描述。

pkt\_pos

当前数据包在输入文件或流中的位置（以字节为单位，从输入开始）。值 -1 表示此信息不可用。

期间

当前数据包的持续时间，以秒为单位。

pkt\_size

当前数据包的大小（以字节为单位）。

#### 39.76.2 文本扩展

如果expansion设置为strftime ，过滤器会识别提供的文本中的 strftime() 序列并相应地扩展它们。检查 strftime() 的文档。此功能已弃用。

如果expansion设置为none ，则逐字打印文本。

如果扩展设置为正常（默认），则使用以下扩展机制。

反斜杠字符“ \ ”后跟任何字符，始终扩展到第二个字符。

%{...}形式的序列被扩展。大括号之间的文本是一个函数名，可能后跟由“:”分隔的参数。如果参数包含特殊字符或分隔符（':' 或 '}'），则应将它们转义。

过滤器参数字符串中的文本选项的值和过滤器图描述中的过滤器参数进行转义，并且可能还需要对 shell 进行转义，这最多构成四个级别的转义；使用文本文件可以避免这些问题。

以下功能可用：

表达式，e

表达式评估结果。

它必须采用一个参数来指定要计算的表达式，该表达式接受与x和y值相同的常量和函数。请注意，并非所有常量都应使用，例如，在计算表达式时不知道文本大小，因此常量text\_w和text\_h将具有未定义的值。

expr\_int\_format, eif

评估表达式的值并输出为格式化整数。

第一个参数是要计算的表达式，就像expr函数一样。第二个参数指定输出格式。允许的值为“ x ”、“ X ”、“ d ”和“ u ”。它们的处理方式与printf函数中的完全相同。第三个参数是可选的，它设置输出所占的位置数。它可用于从左侧添加零填充。

时间

过滤器运行的时间，以 UTC 表示。它可以接受一个参数：一个 strftime() 格式字符串。格式字符串被扩展以支持变量%[1-6]N ，它打印秒的小数部分，可选择指定的位数。

当地时间

过滤器运行的时间，以本地时区表示。它可以接受一个参数：一个 strftime() 格式字符串。格式字符串被扩展以支持变量%[1-6]N ，它打印秒的小数部分，可选择指定的位数。

元数据

帧元数据。接受一个或两个参数。

第一个参数是必需的，用于指定元数据键。

第二个参数是可选的，它指定一个默认值，在元数据键未找到或为空时使用。

可以通过检查以 TAG 开头的条目来识别可用的元数据，这些条目包含在通过运行ffprobe -show\_frames打印的每个帧部分中。

在通向 drawtext 过滤器的过滤器中生成的字符串元数据也可用。

n, 帧数

帧号，从0开始。

图片类型

当前图片类型的单字符描述。

积分

当前帧的时间戳。它最多可以接受三个参数。

第一个参数是时间戳的格式；它默认为flt秒，作为具有微秒精度的十进制数； hms代表具有毫秒精度的格式化[-]HH:MM:SS.mmm时间戳。 gmtime代表格式为 UTC 时间的帧的时间戳； localtime代表格式化为本地时区时间的帧的时间戳。

第二个参数是添加到时间戳的偏移量。

如果格式设置为hms ，则可以提供第三个参数24HH以 24 小时格式 (00-23) 显示格式化时间戳的小时部分。

如果格式设置为localtime或gmtime ，则可以提供第三个参数：strftime() 格式字符串。默认情况下，将使用YYYY-MM-DD HH:MM:SS格式。

#### 39.76.3 命令

此过滤器支持通过命令更改参数：

重新初始化

改变现有的过滤器参数。

参数的语法与过滤器调用的语法相同，例如

fontsize=56:fontcolor=green:text='你好世界'

使用 sendcmd 的完整过滤器调用如下所示：

sendcmd=c='56.0 drawtext reinit fontsize=56\:fontcolor=green\:text=Hello\\ World'

如果整个参数不能被解析或应用为有效值，则过滤器将继续使用其现有参数。

#### 39.76.4 例子

* 使用可选参数的默认值，使用 FreeSerif 字体绘制“测试文本”。
* drawtext="fontfile=/usr/share/fonts/truetype/freefont/FreeSerif.ttf: text='测试文本'"
* 在 x=100 和 y=50 的位置（从屏幕的左上角算起）用大小为 24 的 FreeSerif 字体绘制“测试文本”，文本为黄色，周围有一个红色框。文本和框的不透明度均为 20%。
* drawtext="fontfile=/usr/share/fonts/truetype/freefont/FreeSerif.ttf: text='测试文本':\
* x=100: y=50: fontsize=24: fontcolor=yellow@0.2: box=1: boxcolor=red@0.2"

请注意，如果参数列表中未使用空格，则不需要双引号。

* 在视频帧的中心显示文本：
* drawtext="fontsize=30:fontfile=FreeSerif.ttf:text='hello world':x=(w-text\_w)/2:y=(h-text\_h)/2"
* 在随机位置显示文本，每 30 秒切换到一个新位置：
* drawtext="fontsize=30:fontfile=FreeSerif.ttf:text='hello world':x=if(eq(mod(t\,30)\,0)\,rand(0\,(w-text\_w)) \,x):y=if(eq(mod(t\,30)\,0)\,rand(0\,(h-text\_h))\,y)"
* 在视频帧的最后一行显示从右向左滑动的文本行。假定文件LONG\_LINE包含没有换行符的单行。
* drawtext="fontsize=15:fontfile=FreeSerif.ttf:text=LONG\_LINE:y=h-line\_h:x=-50\*t"
* 在框架底部显示文件CREDITS的内容并向上滚动。
* drawtext="fontsize=20:fontfile=FreeSerif.ttf:textfile=CREDITS:y=h-20\*t"
* 在输入视频的中心绘制一个绿色字母“g”。字形基线位于屏幕高度的一半。
* drawtext="fontsize=60:fontfile=FreeSerif.ttf:fontcolor=green:text=g:x=(w-max\_glyph\_w)/2:y=h/2-ascent"
* 每 3 秒显示文本 1 秒：
* drawtext="fontfile=FreeSerif.ttf:fontcolor=white:x=100:y=x/dar:enable=lt(mod(t\,3)\,1):text='blink'"
* 使用 fontconfig 设置字体。请注意，冒号需要转义。
* drawtext='fontfile=Linux Libertine O-40\:style=Semibold:text=FFmpeg'
* 绘制“测试文本”，字体大小取决于视频的高度。
* drawtext="text='测试文本': fontsize=h/30: x=(w-text\_w)/2: y=(h-text\_h\*2)"
* 打印实时编码的日期（请参阅 strftime(3)）：
* drawtext='fontfile=FreeSans.ttf:text=%{localtime\:%a %b %d %Y}'
* 显示文本淡入淡出（出现/消失）：
* #!/bin/sh
* DS=1.0 #显示开始
* DE=10.0 #显示结束
* FID=1.5 # 淡入淡出持续时间
* FOD=5 # 淡出持续时间
* ffplay -f lavfi "color,drawtext=text=TEST:fontsize=50:fontfile=FreeSerif.ttf:fontcolor\_expr=ff0000%{eif\\\\: clip(255\*(1\*between(t\\, $DS + $FID\\, $DE - $FOD) + ((t - $DS)/$FID)\*between(t\\, $DS\\, $DS + $FID) + (-(t - $DE) /$FOD)\*between(t\\, $DE - $FOD\\, $DE) )\\, 0\\, 255) \\\\: x\\\\: 2 }"
* 水平对齐多个单独的文本。请注意， max\_glyph\_a和字体大小值包含在y偏移量中。
* drawtext=fontfile=FreeSans.ttf:text=DOG:fontsize=24:x=10:y=20+24-max\_glyph\_a,
* drawtext=fontfile=FreeSans.ttf:text=cow:fontsize=24:x=80:y=20+24-max\_glyph\_a
* lavf.image2dec.source\_basename元数据绘制到每个帧上。否则，绘制字符串“NA”。请注意，image2 分路器必须具有选项-export\_path\_metadata 1才能使特殊元数据字段可用于过滤器。
* drawtext="fontsize=20:fontcolor=white:fontfile=FreeSans.ttf:text='%{metadata\:lavf.image2dec.source\_basename\:NA}':x=10:y=10"

有关 libfreetype 的更多信息，请查看： <http://www.freetype.org/>。

有关 fontconfig 的更多信息，请查看： <http://freedesktop.org/software/fontconfig/fontconfig-user.html>。

有关 libfribidi 的更多信息，请查看： <http://fribidi.org/>。

### 39.77 边缘检测

检测并绘制边缘。该过滤器使用 Canny 边缘检测算法。

过滤器接受以下选项：

低的

高的

设置 Canny 阈值算法使用的低阈值和高阈值。

高阈值选择“强”边缘像素，然后通过 8-连通性与低阈值选择的“弱”边缘像素连接。

low和high阈值必须在 [0,1] 范围内选择，并且low应小于或等于high 。

低的默认值为20/255 ，高的默认值为50/255 。

模式

定义绘图模式。

'电线'

在黑色背景上绘制白色/灰色电线。

'混色'

混合颜色以创建油漆/卡通效果。

'精明'

在所有选定的平面上应用 Canny 边缘检测器。

默认值为wires 。

飞机

选择要过滤的平面。默认情况下过滤所有可用的平面。

* [例子](#Examples-111)

#### 39.77.1 例子

* 具有滞后阈值自定义值的标准边缘检测：
* 边缘检测=低=0.1:高=0.4
* 不加阈值的绘画效果：
* 边缘检测=模式=colormix:高=0

### 39.78 磅

使用 ELBG（增强型 LBG）算法应用色调分离效果。

对于每个输入图像，过滤器将计算从输入到输出的最佳映射，给定码本长度，即不同输出颜色的数量。

此过滤器接受以下选项。

codebook\_length, l

设置码本长度。该值必须是正整数，表示不同输出颜色的数量。默认值为 256。

nb\_steps, n

设置用于计算最优映射的最大迭代次数。值越高，结果越好，计算时间也越长。默认值为 1。

种子

设置一个随机种子，必须是一个介于 0 和 UINT32\_MAX 之间的整数。如果未指定，或者如果明确设置为 -1，过滤器将尝试在尽力而为的基础上使用良好的随机种子。

pal8

设置 pal8 输出像素格式。此选项不适用于长度大于 256 的码本。默认为禁用。

use\_alpha

在量化计算中包括 alpha 值。允许创建具有多个 alpha 平滑混合的调色板输出图像（例如 PNG8）。

### 39.79 熵

测量视频帧颜色通道直方图中的灰度熵。

它接受以下参数：

模式

可以是normal或diff 。默认是正常的。

diff模式测量直方图增量值的熵，相邻直方图值之间的绝对差异。

### 39.80 epx

应用专为像素艺术设计的 EPX 放大滤镜。

它接受以下选项：

n

设置缩放维度： 2表示2xEPX ， 3表示3xEPX 。默认值为3 。

### 39.81当量

设置亮度、对比度、饱和度和近似伽马调整。

过滤器接受以下选项：

对比

设置对比表达式。该值必须是-1000.0到1000.0范围内的浮点值。默认值为“1”。

亮度

设置亮度表达式。该值必须是-1.0到1.0范围内的浮点值。默认值为“0”。

饱和

设置饱和度表达式。该值必须是0.0到3.0范围内的浮点数。默认值为“1”。

伽马

设置伽马表达式。该值必须是0.1到10.0范围内的浮点数。默认值为“1”。

伽玛\_r

设置红色的伽马表达式。该值必须是0.1到10.0范围内的浮点数。默认值为“1”。

伽马\_g

设置绿色的伽马表达式。该值必须是0.1到10.0范围内的浮点数。默认值为“1”。

伽玛\_b

设置蓝色的伽马表达式。该值必须是0.1到10.0范围内的浮点数。默认值为“1”。

伽马权重

设置伽马权重表达式。它可用于减少高伽玛值对明亮图像区域的影响，例如，防止它们过度放大而只是纯白色。该值必须是0.0到1.0范围内的浮点数。值0.0将伽马校正一直向下，而1.0则保持其最大强度。默认值为“1”。

评估

在评估亮度、对比度、饱和度和伽马表达式的表达式时设置。

它接受以下值：

'初始化'

仅在过滤器初始化期间或处理命令时对表达式求值一次

'框架'

评估每个传入帧的表达式

默认值为“ init ”。

这些表达式接受以下参数：

n

输入帧的帧数从0开始

位置

输入文件中相应数据包的字节位置，如果未指定则为 NAN

r

输入视频的帧率，如果输入帧率未知则为 NAN

吨

以秒表示的时间戳，如果输入时间戳未知则为 NAN

* [命令](#Commands-88)

#### 39.81.1 命令

过滤器支持以下命令：

对比

设置对比表达式。

亮度

设置亮度表达式。

饱和

设置饱和度表达式。

伽马

设置伽马表达式。

伽玛\_r

设置 gamma\_r 表达式。

伽马\_g

设置 gamma\_g 表达式。

伽玛\_b

设置 gamma\_b 表达式。

伽马权重

设置 gamma\_weight 表达式。

该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.82侵蚀

对视频应用侵蚀效果。

此过滤器用局部 (3x3) 最小值替换像素。

它接受以下选项：

阈值0

阈值1

阈值2

阈值3

限制每个平面的最大变化，默认为 65535。如果为 0，平面将保持不变。

坐标

指定要引用的像素的标志。默认值为 255，即使用所有八个像素。

标记到本地 3x3 坐标映射，如下所示：

1 2 3 4 5 6 7 8

* [命令](#Commands-89)

#### 39.82.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.83 estdif

去隔行输入视频（“estdif”代表“边缘斜率跟踪去隔行过滤器”）。

使用边缘斜率跟踪算法插入缺失线的仅限空间的过滤器。它接受以下参数：

模式

采用的隔行扫描模式。它接受以下值之一：

框架

每一帧输出一帧。

场地

为每个字段输出一帧。

默认值为字段。

平价

输入隔行扫描视频采用的图片场奇偶校验。它接受以下值之一：

tf

假设顶部字段是第一个。

好朋友

假设底部字段是第一个。

汽车

启用字段奇偶校验的自动检测。

默认值为自动。如果隔行扫描未知或解码器不导出此信息，则假定顶场优先。

取消

指定要去隔行扫描的帧。接受以下值之一：

全部

去隔行扫描所有帧。

交错的

仅去隔行标记为隔行扫描的帧。

默认值为全部。

坡度

指定边缘坡度追踪的搜索半径。默认值为 1。允许的范围为 1 到 15。

红边

指定最佳边缘匹配的搜索半径。默认值为 2。允许的范围为 0 到 15。

经济成本

指定边匹配的边成本。默认值为 1.0。允许的范围是从 0 到 9。

成本

指定边缘匹配的中间成本。默认值为 0.5。允许的范围是从 0 到 1。

成本

指定边缘匹配的距离成本。默认值为 0.5。允许的范围是从 0 到 1。

插话

指定使用的插值。默认为 4 点插值。它接受以下值之一：

2p

两点插值。

4p

四点插值。

6p

六点插值。

* [命令](#Commands-90)

#### 39.83.1 命令

此过滤器支持与选项相同的[命令](#commands)。

### 39.84曝光

调整视频流的曝光。

过滤器接受以下选项：

接触

在 EV 中设置曝光校正。允许范围为 -3.0 至 3.0 EV 默认值为 0 EV。

黑色的

设置黑电平校正。允许的范围是从 -1.0 到 1.0。默认值为 0。

* [命令](#Commands-91)

#### 39.84.1 命令

此过滤器支持与选项相同的[命令](#commands)。

### 39.85 提取平面

从输入视频流中提取颜色通道分量到单独的灰度视频流中。

过滤器接受以下选项：

飞机

设置要提取的平面。

平面的可用值是：

'是'

'你'

' v '

'一个'

' r '

' g '

' b '

选择输入中不可用的平面将导致错误。这意味着您不能同时选择r 、 g 、 b平面和y 、 u 、 v平面。

* [例子](#Examples-112)

#### 39.85.1 例子

* 从输入视频帧中提取亮度、u 和 v 颜色通道分量到 3 个灰度输出：
* ffmpeg -i video.avi -filter\_complex 'extractplanes=y+u+v[y][u][v]' -map '[y]' y.avi -map '[u]' u.avi -map '[ v]' v.avi

### 39.86 褪色

对输入视频应用淡入/淡出效果。

它接受以下参数：

类型，吨

效果类型可以是“in”（淡入）或“out”（淡出）。默认为.

起始帧，s

指定开始应用淡入淡出效果的帧数。默认为 0。

nb\_frames, n

淡入淡出效果持续的帧数。在淡入效果结束时，输出视频将具有与输入视频相同的强度。在淡出过渡结束时，输出视频将填充所选颜色。默认值为 25。

α

如果设置为 1，则仅淡化 alpha 通道（如果输入中存在）。默认值为 0。

开始时间，st

指定帧的时间戳（以秒为单位）以开始应用淡入淡出效果。如果同时指定了 start\_frame 和 start\_time，则淡入淡出将从最后一个开始。默认为 0。

持续时间，d

淡入淡出效果必须持续的秒数。在淡入效果结束时，输出视频将具有与输入视频相同的强度，在淡出过渡结束时，输出视频将填充所选颜色。如果指定了持续时间和 nb\_frames，则使用持续时间。默认为 0（默认使用 nb\_frames）。

颜色，c

指定渐变的颜色。默认为“黑色”。

* [例子](#Examples-113)

#### 39.86.1 例子

* 淡入视频的前 30 帧：
* 淡入淡出：0:30

上面的命令等同于：

淡入淡出=t=in:s=0:n=30

* 淡出 200 帧视频的最后 45 帧：
* 淡出=淡出：155:45
* fade=type=out:start\_frame=155:nb\_frames=45
* 淡入 1000 帧视频的前 25 帧并淡出最后 25 帧：
* 淡入：0:25，淡入：975:25
* 将前 5 帧设为黄色，然后从第 5-24 帧淡入：
* 淡入淡出：5:20：颜色=黄色
* 在视频的前 25 帧淡入 alpha：
* 淡入淡出：0:25:alpha=1
* 使前 5.5 秒变黑，然后淡入 0.5 秒：
* 淡入淡出=t=in:st=5.5:d=0.5

### 39.87 反馈

应用反馈视频过滤器。

该过滤器将裁剪后的输入帧传递给第二个输出。从那里可以使用其他视频过滤器对其进行过滤。过滤器从第 2 个输入接收到帧后，该帧在第 1 个输入的原始帧之上组合并传递到第 1 个输出。

典型的用法是只过滤部分帧。

过滤器接受以下选项：

X

是

设置左上裁剪位置。

w

H

设置裁剪尺寸。

* [例子](#Examples-114)

#### 39.87.1 例子

* 使用 gblur 过滤器仅模糊视频帧大小 100x100 的左上角矩形部分。
* [in][blurin]feedback=x=0:y=0:w=100:h=100[out][blurout];[blurout]gblur=8[blurin]
* 使用 drawbox 过滤器在大小为 100x100 的视频帧的左上部分绘制黑框。
* [in][blurin]feedback=x=0:y=0:w=100:h=100[out][blurout];[blurout]drawbox=x=0:y=0:w=100:h=100 :t=100[模糊]

### 39.88 fftdnoiz

使用 3D FFT（频域滤波）对帧进行降噪。

过滤器接受以下选项：

西格玛

设置噪声西格玛常数。这设置了去噪强度。默认值为 1。允许的范围是从 0 到 30。使用非常高的 sigma 和低重叠可能会产生块效应。

数量

设置去噪量。默认情况下，所有检测到的噪音都会减少。默认值为 1。允许的范围为 0 到 1。

堵塞

以像素为单位设置块的大小，默认为 32，可以是 8 到 256。

重叠

设置块重叠。默认值为 0.5。允许的范围是从 0.2 到 0.8。

方法

设置去噪方法。默认是wiener ，也可以是hard 。

上一页

设置用于去噪的先前帧数。默认设置为 0。

下一个

将下一帧的数量设置为用于去噪。默认设置为 0。

飞机

设置将被过滤的平面，默认情况下，除 alpha 外，所有可用的过滤。

### 39.89 fftfilt

将任意表达式应用于频域中的样本

直流Y

调整图像亮度平面的直流值（增益）。过滤器接受0到1000范围内的整数值。默认值设置为0 。

直流电U

调整图像第一个色度平面的直流值（增益）。过滤器接受0到1000范围内的整数值。默认值设置为0 。

直流电压

调整图像第二个色度平面的直流值（增益）。过滤器接受0到1000范围内的整数值。默认值设置为0 。

重

设置亮度平面的频域权重表达式。

权重\_U

设置第一个色度平面的频域权重表达式。

重量\_V

设置第二个色度平面的频域权重表达式。

评估

在计算表达式时设置。

它接受以下值：

'初始化'

在过滤器初始化期间只对表达式求值一次。

'框架'

评估每个传入帧的表达式。

默认值为“ init ”。

过滤器接受以下变量：

X

是

当前样本的坐标。

W

H

图像的宽度和高度。

否

输入帧的编号，从0开始。

WS

HS

用于水平和垂直处理的 FFT 数组的大小。

* [例子](#Examples-115)

#### 39.89.1 例子

* 高通：
* fftfilt=dc\_Y=128:weight\_Y='squish(1-(Y+X)/100)'
* 低通：
* fftfilt=dc\_Y=0:weight\_Y='squish((Y+X)/100-1)'
* 锐化：
* fftfilt=dc\_Y=0:weight\_Y='1+squish(1-(Y+X)/100)'
* 模糊：
* fftfilt=dc\_Y=0:weight\_Y='exp(-4 \* ((Y+X)/(W+H)))'

### 39.90场

使用步幅算法从交错图像中提取单个场以避免浪费 CPU 时间。输出帧标记为非隔行扫描。

过滤器接受以下选项：

类型

指定是提取顶部（如果值为0或top ）还是底部字段（如果值为1或bottom ）。

### 39.91 现场提示

通过从提示文件作为数字提供的周围帧复制顶部和底部字段来创建新帧。

暗示

设置包含提示的文件：绝对/相对帧数。

剪辑中的每一帧必须有一行。每行必须包含两个用逗号分隔的数字，后面可以选择-或+ 。每行文件中提供的数字不能超出 [N-1,N+1]，其中 N 是绝对模式的当前帧编号，或者超出相对模式的 [-1, 1] 范围。第一个数字表示从哪一帧拾取顶场，第二个数字表示从哪一帧拾取底场。

如果可选地后跟+输出帧将被标记为隔行扫描，否则如果后跟-输出帧将被标记为逐行扫描，否则它将被标记为与输入帧相同。如果可选地后跟t输出帧将仅使用顶场，或者在b的情况下它将仅使用底场。如果行以#或;开头该行被跳过。

模式

可以是 item absolute或relative或pattern 。默认是绝对的。模式模式与相对模式相同，除了在文件的最后一个条目如果要处理的帧多于提示文件寻找开始。

相对模式的前几行提示文件示例：

0,0 - # 第一帧

1,0 - # 第二帧，使用第三帧的顶场和第二帧的底场

1,0 - #第三帧，使用第四帧的顶场和第三帧的底场

1,0 -

0,0 -

0,0 -

1,0 -

1,0 -

1,0 -

0,0 -

0,0 -

1,0 -

1,0 -

1,0 -

0,0 -

### 39.92 现场比赛

用于反电视电影的场匹配滤波器。它旨在从电视电影流中重建逐行帧。该过滤器不会丢弃重复的帧，因此要实现完整的反电视电影场匹配，需要在过滤器图中进行抽取过滤器，例如[抽取。](#decimate)

场匹配和抽取的分离尤其受到在两者之间插入去隔行滤波器回退的可能性的推动。如果源混合了电视电影和真正的隔行扫描内容， fieldmatch将无法匹配隔行扫描部分的场。但是这些剩余的梳理帧将被标记为隔行扫描，因此可以在抽取之前由稍后的过滤器（例如[yadif）](#yadif)去隔行扫描。

除了各种配置选项外， fieldmatch还可以采用可选的第二个流，通过ppsrc选项激活。如果启用，帧重建将基于来自第二个流的字段和帧。这允许对第一个输入进行预处理，以帮助过滤器的各种算法，同时保持输出无损（假设字段匹配正确）。通常，场感知降噪器或亮度/对比度调整可以提供帮助。

请注意，此过滤器使用与 TIVTC/TFM（AviSynth 项目）和 VIVTC/VFM（VapourSynth 项目）相同的算法。后者是基于fieldmatch的TFM 的轻型克隆。虽然语义和用法非常接近，但某些行为和选项名称可能不同。

抽取过滤器目前仅适用于恒定帧速率输入[。](#decimate)如果您的输入混合了电视电影 (30fps) 和具有较低帧率（如 24fps）的渐进式内容，请使用以下过滤器链来生成必要的 cfr 流：dejudder, fps=30000/1001,fieldmatch,decimate 。

过滤器接受以下选项：

命令

指定输入流的假定字段顺序。可用值是：

'汽车'

自动检测奇偶校验（使用 FFmpeg 的内部奇偶校验值）。

'好朋友'

首先假设底场。

' tff '

首先假设顶场。

请注意，有时建议不要相信流宣布的奇偶校验。

默认值为自动。

模式

设置要使用的匹配模式或策略。 pc模式是最安全的，因为它不会在可能的情况下因重复帧而产生抖动，但如果有错误的编辑或混合字段，它最终会输出组合帧，而实际上可能存在良好的匹配。另一方面， pcn\_ub模式在产生抖动方面风险最大，但如果有的话，几乎总能找到好的帧。其他值都介于pc和pcn\_ub之间，就冒抖动和创建重复帧的风险与在具有错误编辑、孤立字段、混合字段等的部分中找到良好匹配而言。

有关 p/c/n/u/b 的更多详细信息，请参阅[p/c/n/u/b 含义](#p_002fc_002fn_002fu_002fb-meaning)部分。

可用值是：

'电脑'

2 向匹配 (p/c)

' pc\_n '

2 路匹配，如果仍然梳理则尝试第 3 次匹配 (p/c + n)

' pc\_u '

2 向匹配，如果仍然组合 (p/c + u)，则尝试第 3 次匹配（相同顺序）

' pc\_n\_ub '

2 向匹配，如果仍然是梳理则尝试第 3 次匹配，如果仍然是梳理则尝试第 4/5 次匹配 (p/c + n + u/b)

' pcn '

3 路匹配 (p/c/n)

' pcn\_ub '

3 向匹配，如果原始匹配的所有 3 个都被检测为组合 (p/c/n + u/b)，则尝试第 4/5 个匹配

order = tff （和auto或top上的字段）将用于该模式的匹配项。

就速度而言， pc模式是迄今为止最快的，而pcn\_ub是最慢的。

默认值为pc\_n 。

ppsrc

将主输入流标记为预处理输入，并启用辅助输入流作为从中选择字段的干净源。有关更多详细信息，请参阅过滤器介绍。它类似于VFM/TFM 中的clip2功能。

默认值为0 （禁用）。

场地

设置要匹配的字段。建议将此设置为与order相同的值，除非您遇到该设置的匹配失败。在某些情况下，更改用于匹配的字段可能会对匹配性能产生很大影响。可用值是：

'汽车'

自动（与order相同的值）。

'底部'

从底场比赛。

'顶'

从顶场比赛。

默认值为自动。

色度

设置匹配比较期间是否包含色度。在大多数情况下，建议启用此功能。仅当您的剪辑有严重的彩虹或其他瑕疵等糟糕的色度问题时，才应将此设置为0 。将此设置为0也可用于以牺牲某些准确性为代价来加快速度。

默认值为1 。

y0

y1

y0和y1之间的线排除在字段匹配决策之外。排除带可用于忽略字幕、徽标或其他可能干扰匹配的内容。 y0设置起始扫描线， y1设置结束线； y0和y1之间的所有行（包括y0和y1 ）都将被忽略。将y0和y1设置为相同的值将禁用该功能。 y0和y1默认为0 。

阈值

将场景变化检测阈值设置为亮度平面上最大变化的百分比。好的值在[8.0, 14.0]范围内。场景变化检测仅在combmatch = sc的情况下相关。 scthresh的范围是[0.0, 100.0] 。

默认值为12.0 。

比拼

当combatch不是none时， fieldmatch在决定使用哪场比赛作为最终比赛时会考虑比赛的综合得分。可用值是：

'无'

没有基于综合分数的最终匹配。

' sc '

仅当检测到场景变化时才使用 Combed 分数。

'满'

始终使用梳理分数。

默认为sc 。

梳理数据库

强制fieldmatch计算某些匹配项的组合指标并打印它们。此设置在 TFM/VFM 词汇表中称为micout 。可用值是：

'无'

没有强制计算。

' pcn '

强制 p/c/n 计算。

' pcnub '

强制进行 p/c/n/u/b 计算。

默认值为无。

阈值

这是用于梳状帧检测的区域梳状阈值。这基本上控制了必须检测到的“强”或“可见”梳理的程度。较大的值意味着梳理必须更明显，较小的值意味着梳理可以不那么明显或强烈，但仍然可以检测到。有效设置从-1 （每个像素都将被检测为梳状）到255 （没有像素将被检测为梳状）。这基本上是一个像素差值。一个好的范围是[8, 12] 。

默认值为9 。

色度

设置在组合帧决策中是否考虑色度。仅当您的源存在色度问题（彩虹等）导致启用色度的梳状帧检测出现问题时才禁用此选项。实际上，使用chroma = 0通常更可靠，除非源中只有色度组合。

默认值为0 。

块x

块状的

分别设置梳状帧检测时使用的窗口的x轴和y轴大小。这与区域的大小有关，在该区域中，需要将combpel像素检测为已梳理，以便将帧声明为已梳理。有关更多信息，请参阅combpel参数说明。可能的值是从 4 开始到 512 的任何 2 的幂。

默认值为16 。

强迫

帧上任何blocky by blockx大小的块内的梳理像素数，用于将帧检测为梳理。虽然cthresh控制组合必须“可见”的程度，但此设置控制帧上任何局部区域（由blockx和blocky设置定义的窗口）必须有“多少”组合。最小值为0 ，最大值为blocky x blockx （此时不会将任何帧检测为梳状）。此设置在 TFM/VFM 词汇表中称为MI 。

默认值为80 。

* [p/c/n/u/b 含义](#p_002fc_002fn_002fu_002fb-meaning-1)
* [例子](#Examples-116)

#### 39.92.1 p/c/n/u/b 含义

* [p/c/n](#p_002fc_002fn)
* [u/b](#u_002fb)

#### 39.92.1.1 p/c/n

我们假设以下电视电影流：

热门字段：1 2 2 3 4

底部字段：1 2 3 4 4

这些数字对应于字段相关的渐进帧。在这里，前两帧是渐进的，第 3 和第 4 帧是梳状的，依此类推。

当fieldmatch配置为从底部 ( field = bottom ) 运行匹配时，这就是此输入流的转换方式：

输入流：

总计 1 2 2 3 4

B 1 2 3 4 4 <-- 匹配引用

匹配：ccnnc

输出流：

总计 1 2 3 4 4

乙 1 2 3 4 4

作为场匹配的结果，我们可以看到一些帧被复制了。要执行完整的反电视电影，您需要在此操作后依赖于抽取滤波器。参见[抽取滤波器](#decimate)的例子。

field = top )匹配的相同操作如下所示：

输入流：

T 1 2 2 3 4 <-- 匹配引用

乙 1 2 3 4 4

匹配：ccppc

输出流：

总计 1 2 2 3 4

乙 1 2 2 3 4

在这些示例中，我们可以看到p 、 c和n的含义；基本上，它们指的是相反奇偶校验的帧和字段：

* p匹配上一帧中相反奇偶校验的字段
* c匹配当前帧中相反奇偶校验的字段
* n在下一帧中匹配相反奇偶校验的字段

#### 39.92.1.2 u/b

u和b匹配有点特殊，因为它们从相反的奇偶校验标志进行匹配。在以下示例中，我们假设我们当前正在匹配第 2 帧（顶部：2，底部：2）。根据匹配，每个匹配字段的上方和下方都会放置一个“x”。

使用底部匹配（ field = bottom ）：

匹配：cpnbu

xxxxx

顶部 1 2 2 1 2 2 1 2 2 1 2 2 1 2 2

底部 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3

xxxxx

输出帧：

2 1 2 2 2

2 2 2 1 3

使用顶部匹配（ field = top ）：

匹配：cpnbu

xxxxx

顶部 1 2 2 1 2 2 1 2 2 1 2 2 1 2 2

底部 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3

xxxxx

输出帧：

2 2 2 1 2

2 1 3 2 2

#### 39.92.2 例子

顶场第一电视电影流的简单 IVTC：

fieldmatch=order=tff:combmatch=none, 抽取

Advanced IVTC，在[yadif上回退](#yadif)用于静止梳理帧：

fieldmatch=order=tff:combmatch=full, yadif=deint=interlaced, decimate

### 39.93 场序

转换输入视频的场序。

它接受以下参数：

命令

输出字段顺序。有效值为顶场优先的tff或底场优先的bff 。

默认值为“ tff ”。

转换是通过将图片内容向上或向下移动一行，并用适当的图片内容填充剩余的行来完成的。此方法与大多数广播场顺序转换器一致。

如果输入视频没有被标记为隔行扫描，或者它已经被标记为具有所需的输出场序，则此过滤器不会改变输入视频。

在转换为底场优先的 PAL DV 材料或从 PAL DV 材料转换时，它非常有用。

例如：

ffmpeg -i in.vob -vf "fieldorder=bff" 输出.dv

### 39.94 fifo，afifo

缓冲输入图像并在请求时发送它们。

它主要在由 libavfilter 框架自动插入时有用。

它不带参数。

### 39.95 填充边框

填充输入视频的边界，而不改变视频流尺寸。有时视频的四个边缘可能有垃圾，您可能不希望裁剪视频输入以保持某个数字的大小倍数。

此过滤器接受以下选项：

左边

从左边界填充的像素数。

正确的

从右边界填充的像素数。

顶部

从上边框填充的像素数。

底部

从底部边框开始填充的像素数。

模式

设置填充模式。

它接受以下值：

'涂抹'

使用最外面的像素填充像素

'镜子'

使用镜像填充像素（半样本对称）

'固定'

用常量值填充像素

'反映'

使用反射填充像素（整个样本对称）

'包裹'

使用环绕填充像素

'褪色'

将像素淡化为常量值

'边距'

用边界附近的加权平均像素填充顶部和底部的像素

默认为smear 。

颜色

在固定或淡入淡出模式下为像素设置颜色。默认为黑色。

* [命令](#Commands-92)

#### 39.95.1 命令

此过滤器支持与选项相同的[命令](#commands)。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.96 查找矩形

找一个矩形物体

它接受以下选项：

目的

对象图像的文件路径，需要在gray8。

临界点

检测阈值，默认为 0.5。

贴图

mipmap 的数量，默认为 3。

xmin, ymin, xmax, ymax

指定要在其中搜索的矩形。

丢弃

丢弃未检测到对象的帧。默认是禁用的。

* [例子](#Examples-117)

#### 39.96.1 例子

* ffmpeg通过给定视频的提供图像覆盖矩形对象：
* ffmpeg -i 文件.ts -vf find\_rect=newref.pgm,cover\_rect=cover.jpg:mode=cover new.mkv

### 39.97 填洪

具有相同像素分量值和另一个值的泛洪区域。

它接受以下选项：

X

设置像素 x 坐标。

是

设置像素 y 坐标。

s0

设置源 #0 组件值。

s1

设置源 #1 组件值。

s2

设置源 #2 组件值。

s3

设置源 #3 组件值。

d0

设置目标#0 组件值。

d1

设置目标 #1 组件值。

d2

设置目标 #2 组件值。

d3

设置目标#3 组件值。

### 39.98格式

将输入视频转换为指定的像素格式之一。 Libavfilter 将尝试选择一个适合作为下一个过滤器输入的过滤器。

它接受以下参数：

pix\_fmts

以“|”分隔的像素格式名称列表，例如“pix\_fmts=yuv420p|monow|rgb24”。

* [例子](#Examples-118)

#### 39.98.1 例子

* 将输入视频转换为yuv420p格式
* 格式=pix\_fmts=yuv420p

将输入视频转换为列表中的任何格式

格式=pix\_fmts=yuv420p|yuv444p|yuv410p

### 39.99 帧/秒

根据需要复制或丢弃帧，将视频转换为指定的恒定帧速率。

它接受以下参数：

每秒帧数

所需的输出帧速率。它接受包含以下常量的表达式：

' source\_fps '

输入的帧率

' ntsc '

NTSC 帧率30000/1001

'朋友'

PAL 帧率25.0

'电影'

电影帧率24.0

' ntsc\_film '

NTSC-电影帧率24000/1001

默认值为25 。

开始时间

假设第一个 PTS 应该是给定的值，以秒为单位。这允许在流的开头进行填充/修剪。默认情况下，不会对第一帧的预期 PTS 做出任何假设，因此不会进行填充或修剪。例如，如果视频流在音频流之后开始，这可以设置为 0 以用第一帧的副本填充开头，或者用负 PTS 修剪任何帧。

圆形的

时间戳 (PTS) 舍入方法。

可能的值是：

零

向 0 舍入

信息

从 0 舍入

向下

向无穷大舍入

向上

向+无穷大舍入

靠近

四舍五入到最近

默认值接近。

eof\_action

读取最后一帧时执行的操作。

可能的值是：

圆形的

使用与其他帧相同的时间戳舍入方法。

经过

如果尚未达到输入持续时间，则通过最后一帧。

默认为圆形。

或者，可以将选项指定为平面字符串： fps [: start\_time [: round ]]。

另请参阅[setpts](#setpts)过滤器。

* [例子](#Examples-119)

#### 39.99.1 例子

* 将 fps 设置为 25 的典型用法：
* 每秒帧数=每秒帧数=25
* 将 fps 设置为 24，使用缩写和舍入方法舍入到最接近的值：
* fps=fps=电影:圆=近

### 39.100 帧包

将两个不同的视频流打包成一个立体视频，在支持的编解码器上设置适当的元数据。这两个视图应该具有相同的大小和帧率，并且当较短的视频结束时处理将停止。请注意，您可以使用[比例](#scale)和[fps过滤器](#fps)方便地调整视图属性。

它接受以下参数：

格式

所需的打包格式。支持的值为：

某人

视图彼此相邻（默认）。

标签

视图彼此重叠。

线条

视图按行打包。

列

视图按列打包。

帧序列

这些视图在时间上是交错的。

一些例子：

# 将左视图和右视图转换为帧序列视频

ffmpeg -i LEFT -i RIGHT -filter\_complex framepack=frameseq 输出

# 将视图转换为与输入具有相同输出分辨率的并排视频

ffmpeg -i 左 -i 右 -filter\_complex [0:v]scale=w=iw/2[左],[1:v]scale=w=iw/2[右],[左][右]framepack=sbs输出

### 39.101 帧率

通过从源帧插入新的视频输出帧来更改帧速率。

此过滤器未设计为与隔行扫描媒体一起正常工作。如果您希望更改隔行扫描媒体的帧速率，则需要在此过滤器之前取消隔行扫描并在此过滤器之后重新隔行扫描。

接受的选项的描述如下。

每秒帧数

指定每秒输出帧数。此选项也可以单独指定为一个值。默认值为50 。

interp\_start

指定范围的起点，输出帧将在该范围内创建为两帧的线性插值。范围是 [ 0 - 255 ]，默认是15 。

interp\_end

指定将输出帧创建为两个帧的线性插值的范围的末端。范围是 [ 0 - 255 ]，默认是240 。

场景

将检测到场景变化的级别指定为 0 到 100 之间的值，以指示新场景；较低的值反映当前帧引入新场景的可能性较低，而较高的值表示当前帧更有可能是新场景。默认值为8.2 。

旗帜

指定影响过滤过程的标志。

标志的可用值为：

scene\_change\_detect, scd

scene的值启用场景变化检测。默认情况下启用此标志。

### 39.102 帧步

每第 N 帧选择一帧。

此过滤器接受以下选项：

步

一步帧之后选择帧。允许值为大于 0 的正整数。默认值为1 。

### 39.103 冻结检测

检测冻结的视频。

当检测到输入视频在指定持续时间内内容没有显着变化时，此过滤器会记录一条消息并设置帧元数据。视频冻结检测计算视频帧所有分量的平均绝对差值，并将其与本底噪声进行比较。

打印的时间和持续时间以秒表示。 lavfi.freezedetect.freeze\_start元数据键设置在时间戳等于或超过检测持续时间的第一帧上，它包含冻结第一帧的时间戳。 lavfi.freezedetect.freeze\_duration和lavfi.freezedetect.freeze\_end元数据键在冻结后的第一帧上设置。

过滤器接受以下选项：

噪声，n

设置噪声容限。可以以 dB 为单位指定（如果“dB”附加到指定值）或作为 0 和 1 之间的差值比率。默认值为 -60dB，或 0.001。

持续时间，d

设置冻结持续时间直到通知（默认为 2 秒）。

### 39.104 定格画面

冻结视频帧。

此过滤器使用来自第二个输入的帧冻结视频帧。

过滤器接受以下选项：

第一的

设置开始冻结的第一帧数。

最后的

设置结束冻结的最后一帧的编号。

代替

设置来自第二个输入的帧数，它将被用来代替替换的帧。

### 39.105 frei0r

对输入视频应用 frei0r 效果。

要启用此过滤器的编译，您需要安装 frei0r 标头并使用--enable-frei0r配置 FFmpeg 。

它接受以下参数：

过滤器名称

要加载的 frei0r 效果的名称。如果定义了环境变量FREI0R\_PATH ，则会在FREI0R\_PATH中以冒号分隔的列表指定的每个目录中搜索 frei0r 效果。否则，将搜索标准的 frei0r 路径，顺序为： HOME/.frei0r-1/lib/ 、 /usr/local/lib/frei0r-1/ 、 /usr/lib/frei0r-1/ 。

过滤参数

要传递给 frei0r 效果的以“|”分隔的参数列表。

frei0r 效果参数可以是布尔值（其值为“y”或“n”）、双精度值、颜色（指定为R / G / B ，其中R 、 G和B是介于 0.0 和1.0，包括）或[ffmpeg-utils 手册中（ffmpeg-utils）“颜色”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)中指定的颜色描述，位置（指定为X / Y ，其中X和Y是浮点数）和/或细绳。

参数的数量和类型取决于加载的效果。如果未指定效果参数，则设置默认值。

* [例子](#Examples-120)
* [命令](#Commands-93)

#### 39.105.1 示例

* 应用 distort0r 效果，设置前两个双精度参数：
* frei0r=filter\_name=distort0r:filter\_params=0.5|0.01
* 应用颜色距离效果，将颜色作为第一个参数：
* frei0r=色差：0.2/0.3/0.4
* frei0r=色差：紫色
* frei0r=颜色距离：0x112233
* 应用透视效果，指定左上角和右上角的图像位置：
* frei0r=透视：0.2/0.2|0.8/0.2

有关详细信息，请参阅<http://frei0r.dyne.org>

#### 39.105.2 命令

此过滤器支持filter\_params选项作为[命令](#commands)。

### 39.106 fspp

应用快速简单的后处理。它是[spp](#spp)的更快版本。

它将 (I)DCT 分成水平/垂直通道。与简单的后处理过滤器不同，其中一个是每个块执行一次，而不是每个像素。这允许更高的速度。

过滤器接受以下选项：

质量

设置质量。此选项定义平均级别数。它接受 4-5 范围内的整数。默认值为4 。

qp

强制一个恒定的量化参数。它接受 0-63 范围内的整数。如果未设置，过滤器将使用视频流中的 QP（如果可用）。

力量

设置过滤强度。它接受 -15 到 32 范围内的整数。较低的值意味着更多的细节但也有更多的伪影，而较高的值使图像更平滑但也更模糊。默认值为0 − PSNR 最优。

使用\_bframe\_qp

1 ，则启用来自 B 帧的 QP 。使用此选项可能会导致闪烁，因为 B 帧通常具有较大的 QP。默认值为0 （未启用）。

### 39.107 模糊

应用高斯模糊滤镜。

过滤器接受以下选项：

西格玛

设置水平西格玛，高斯模糊的标准差。默认值为0.5 。

脚步

设置高斯近似的步数。默认为1 。

飞机

设置要过滤的平面。默认情况下，所有平面都被过滤。

西格玛V

sigma相同。默认为-1 。

* [命令](#Commands-94)

#### 39.107.1 命令

此过滤器支持与选项相同的命令。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.108 geq

将通用方程应用于每个像素。

过滤器接受以下选项：

lum\_expr, lum

设置亮度表达式。

cb\_expr, cb

设置色度蓝色表达式。

cr\_expr, cr

设置色度红色表达式。

alpha\_expr，一个

设置 alpha 表达式。

红色表达式，r

设置红色表达式。

绿色表达式，g

设置绿色表达式。

蓝色表达式，b

设置蓝色表达式。

根据指定的选项选择颜色空间。如果指定了lum\_expr 、 cb\_expr或cr\_expr选项之一，过滤器将自动选择 YCbCr 色彩空间。如果指定了red\_expr 、 green\_expr或blue\_expr选项之一，它将选择 RGB 色彩空间。

如果其中一个色度表达式未定义，则它会退回到另一个。如果未指定 alpha 表达式，它将计算为不透明值。如果未指定任何色度表达式，它们将计算为亮度表达式。

表达式可以使用以下变量和函数：

否

过滤帧的序号，从0开始。

X

是

当前样本的坐标。

W

H

图像的宽度和高度。

软件

上海

宽度和高度比例取决于当前过滤的平面。它是对应的亮度平面像素数与当前平面像素数的比值。例如，对于 YUV4:2:0 ，亮度平面的值为1,1 ，色度平面的值为0.5,0.5 。

吨

当前帧的时间，以秒为单位。

p(x, y)

当前平面位置 ( x , y ) 处的像素值。

亮度（x，y）

x , y ) 处的像素值。

cb(x, y)

蓝差色度平面位置 ( x , y ) 处的像素值。如果没有这样的平面，则返回 0。

cr(x, y)

红差色度平面位置 ( x , y ) 处的像素值。如果没有这样的平面，则返回 0。

r(x, y)

g(x, y)

b(x, y)

红/绿/蓝分量位置 ( x , y )处的像素值。如果没有这样的组件，则返回 0。

阿尔法（x，y）

alpha 平面位置 ( x , y ) 处的像素值。如果没有这样的平面，则返回 0。

psum(x,y), lumsum(x, y), cbsum(x,y), crsum(x,y), rsum(x,y), gsum(x,y), bsum(x,y), alphasum (x,y)

从 (0,0) 到 (x,y) 的矩形中的样本值的总和，这允许获得矩形内的样本总和。查看没有和后缀的函数。

插值

设置其中一种插值方法：

最近的，n

双线性，b

默认是双线性的。

对于函数，如果x和y在区域之外，该值将自动裁剪到更近的边缘。

请注意，此过滤器可以使用多个线程，在这种情况下，每个切片都有自己的表达式状态。如果你只想使用一个表达式状态，因为你的表达式依赖于以前的状态，那么你应该将过滤器线程数限制为 1。

* [例子](#Examples-121)

#### 39.108.1 示例

* 水平翻转图像：
* geq=p(WX\,Y)
* 生成一个二维正弦波，角度为PI/3 ，波长为 100 像素：
* geq=128 + 100\*sin(2\*(PI/100)\*(cos(PI/3)\*(X-50\*T) + sin(PI/3)\*Y)):128:128
* 生成奇特的神秘移动光：
* nullsrc=s=256x256,geq=random(1)/hypot(X-cos(N\*0.07)\*W/2-W/2\,Y-sin(N\*0.09)\*H/2-H/2) ^2\*1000000\*sin(N\*0.02):128:128
* 生成快速浮雕效果：
* 格式=灰色，geq=lum\_expr='(p(X,Y)+(256-p(X-4,Y-4)))/2'
* 根据像素位置修改 RGB 分量：
* geq=r='X/W\*r(X,Y)':g='(1-X/W)\*g(X,Y)':b='(HY)/H\*b(X,Y )'
* 创建一个与输入大小相同的径向渐变（另请参阅[晕影](#vignette)过滤器）：
* geq=lum=255\*gauss((X/W-0.5)\*3)\*gauss((Y/H-0.5)\*3)/gauss(0)/gauss(0),format=gray

### 39.109 gradfun

修复有时通过截断至 8 位颜色深度而引入近乎平坦区域的条带伪像。内插应该到达条带所在位置的梯度，并抖动它们。

它专为播放而设计。不要在有损压缩之前使用它，因为压缩往往会丢失抖动并带回频带。

它接受以下参数：

力量

过滤器将改变任何一个像素的最大数量。这也是检测接近平坦区域的阈值。可接受的值范围从 .51 到 64；默认值为 1.2。超出范围的值将被裁剪到有效范围内。

半径

适合梯度的邻域。较大的半径可以使渐变更平滑，但也可以防止过滤器修改细节区域附近的像素。可接受的值为 8-32；默认值为 16。超出范围的值将被裁剪到有效范围内。

或者，可以将选项指定为扁平字符串： strength [: radius ]

* [例子](#Examples-122)

#### 39.109.1 示例

* 3.5且半径为8的过滤器：
* gradfun=3.5:8
* 指定半径，省略强度（这将回退到默认值）：
* gradfun=半径=8

### 39.110 图形监视器

显示各种过滤图统计信息。

使用此过滤器，可以调试完整的过滤器图。尤其是链接填充排队帧的问题。

过滤器接受以下选项：

大小，小号

设置视频输出大小。默认为hd720 。

不透明度，o

设置视频不透明度。默认值为0.9 。允许的范围是从0到1 。

模式，米

设置输出模式，可以是fulll或compact 。在紧凑模式下，只有带有一些排队帧的过滤器才会显示统计信息。

标志，f

设置启用视频中显示哪些统计数据的标志。

标志的可用值是：

'队列'

显示每个链接中排队的帧数。

' frame\_count\_in '

显示从过滤器中获取的帧数。

' frame\_count\_out '

显示过滤器给出的帧数。

' frame\_count\_delta '

显示上述两个值之间的帧增量数。

'积分'

显示当前过滤的帧点。

' pts\_delta '

显示当前帧和上一帧之间的点增量。

'时间'

显示当前过滤的帧时间。

' time\_delta '

显示当前帧和上一帧之间的时间差。

'时基'

显示过滤器链接的时基。

'格式'

显示过滤器链接使用的格式。

'尺寸'

如果过滤器链接使用音频，则显示视频大小或音频通道数。

'率'

如果过滤器链接使用音频，则显示视频帧率或采样率。

' eof '

显示链路输出状态。

' sample\_count\_in '

显示从过滤器中获取的样本数。

' sample\_count\_out '

显示过滤器给出的样本数。

' sample\_count\_delta '

显示上述两个值之间的样本增量数。

率，r

设置输出流的视频速率上限，默认值为25 。这保证输出视频帧率不会高于此值。

### 39.111 灰色世界

基于灰色世界假设应用颜色校正的颜色恒常性过滤器

参见： <https://www.researchgate.net/publication/275213614_A_New_Color_Correction_Method_for_Underwater_Imaging>

该算法使用线性光，因此输入数据应事先线性化（并可能正确标记）。

ffmpeg -i 输入 -vf zscale=transfer=linear,grayworld,zscale=transfer=bt709,format=yuv420p 输出

### 39.112 灰边

一种颜色恒常变化滤波器，通过灰度边缘算法估计场景光照并相应地校正场景颜色。

请参阅： <https://staff.science.uva.nl/th.gevers/pub/GeversTIP07.pdf>

过滤器接受以下选项：

迪福德

要在场景中应用的差异化顺序。必须在 [0,2] 范围内选择，默认值为 1。

水貂规范

用于计算 Minkowski 距离的 Minkowski 参数。必须在 [0,20] 范围内选择，默认值为 1。设置为 0 以获得最大值而不是计算 Minkowski 距离。

西格玛

要应用于场景的高斯模糊的标准偏差。必须在 [0,1024.0] 范围内选择，默认值 = 1。如果difford大于 0 ，则 floor( sigma \* break\_off\_sigma(3) ) 不能等于 0 。

* [例子](#Examples-123)

#### 39.112.1 例子

* 灰边：
* greyedge=difford=1:minknorm=5:sigma=2
* 最大边缘：
* greyedge=difford=1:minknorm=0:sigma=2

### 39.113引导

应用引导滤波器进行保边平滑、去雾等。

过滤器接受以下选项：

半径

以像素为单位设置框半径。允许的范围是 1 到 20。默认值为 3。

每股收益

设置正则化参数（带正方形）。允许的范围是 0 到 1。默认值为 0.01。

模式

设置过滤模式。可以是基本的或快速的。默认是基本的。

子

设置快速模式的子采样率。范围是 2 到 64。默认值为 4。在基本模式下不进行子采样。

指导

设置引导模式。可以关闭或打开。默认为关闭。如果关闭，则需要单输入。如果打开，则需要两个具有相同分辨率和像素格式的输入。第二个输入作为指导。

飞机

设置要过滤的平面。默认是第一个。

* [命令](#Commands-95)
* [例子](#Examples-124)

#### 39.113.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

#### 39.113.2 例子

* 带导向滤波器的边缘保持平滑：
* ffmpeg -i in.png -vf 引导出.png
* 去雾、结构转移过滤、带引导过滤器的细节增强。引导图像的生成参考论文《Guided Image Filtering》。参见： http: [//kaiminghe.com/publications/pami12guidedfilter.pdf](http://kaiminghe.com/publications/pami12guidedfilter.pdf) 。
* ffmpeg -i in.png -i guidance.png -filter\_complex guided=guidance=on out.png

### 39.114 杂乱无章

将 Hald CLUT 应用于视频流。

第一个输入是要处理的视频流，第二个是 Hald CLUT。 Hald CLUT 输入可以是简单的图片或完整的视频流。

过滤器接受以下选项：

杂乱无章

设置将从第二个输入流处理的 CLUT 视频帧，可以是first或all 。默认为全部。

最短的

当最短输入终止时强制终止。默认为0 。

重复上次

在流结束后继续应用最后一个 CLUT。值为0会在到达 CLUT 的最后一帧后禁用过滤器。默认为1 。

haldclut还具有与[lut3d](#lut3d)相同的插值选项（两个过滤器共享相同的内部结构）。

此过滤器还支持[帧同步](#framesync)选项。

有关 Hald CLUT 的更多信息，请访问 Eskil Steenberg 的网站（Hald CLUT 作者），网址为<http://www.quelsolaar.com/technology/clut.html>。

* [命令](#Commands-96)
* [工作流程示例](#Workflow-examples)

#### 39.114.1 命令

此过滤器支持将interp选项作为[命令](#commands)。

#### 39.114.2 工作流程示例

* [持有 CLUT 视频流](#Hald-CLUT-video-stream)
* [暂停 CLUT 预览](#Hald-CLUT-with-preview)

#### 39.114.2.1 暂停 CLUT 视频流

生成具有各种效果的身份 Hald CLUT 流：

ffmpeg -f lavfi -i [haldclutsrc](#haldclutsrc) =8 -vf "hue=H=2\*PI\*t:s=sin(2\*PI\*t)+1, curves=cross\_process" -t 10 -c:v ffv1 clut.nut

注意：确保使用无损编解码器。

然后将它与haldclut一起使用以将其应用于一些随机流：

ffmpeg -f lavfi -i mandelbrot -i clut.nut -filter\_complex '[0][1] haldclut' -t 20 mandelclut.mkv

Hald CLUT 将应用于前 10 秒（ clut.nut的持续时间），然后该 CLUT 流的最新图片将应用于mandelbrot流的剩余帧。

#### 39.114.2.2 暂停 CLUT 预览

Hald CLUT 应该是Level\*Level\*Level乘以Level\*Level\*Level像素的方形图像。对于给定的 Hald CLUT，FFmpeg 将选择从图片左上角开始的最大可能正方形。剩余的填充像素（底部或右侧）将被忽略。该区域可用于添加 Hald CLUT 的预览。

haldclut过滤器将支持以下生成的 Hald CLUT ：

ffmpeg -f lavfi -i [haldclutsrc](#haldclutsrc) =8 -vf "

pad=iw+320 [padded\_clut];

smptebars=s=320x256，分割[a][b]；

[padded\_clut][a] overlay=W-320:h, curves=color\_negative [main];

[main][b] overlay=W-320" -frames:v 1 clut.png

它包含 CLUT 的原始效果和预览效果：SMPTE 颜色条显示在右上方，下方是经过颜色变化处理的相同颜色条。

然后，可以通过以下方式可视化此 Hald CLUT 的效果：

ffplay input.mkv -vf "movie=clut.png, [in] haldclut"

### 39.115 hflip

水平翻转输入视频。

例如，要使用ffmpeg水平翻转输入视频：

ffmpeg -i in.avi -vf "hflip" out.avi

### 39.116 希斯特克

此过滤器在每帧的基础上应用全局颜色直方图均衡。

它可用于校正具有压缩像素强度范围的视频。过滤器重新分配像素强度以均衡它们在强度范围内的分布。它可以被视为“自动调整对比度过滤器”。此过滤器仅适用于校正质量下降或捕获不佳的源视频。

过滤器接受以下选项：

力量

确定要应用的均衡量。随着强度的降低，像素强度的分布越来越接近输入帧的分布。该值必须是 [0,1] 范围内的浮点数，默认为 0.200。

强度

设置可以生成的最大强度并适当地缩放输出值。应根据需要设置强度，然后如果需要可以限制强度以避免洗脱。该值必须是 [0,1] 范围内的浮点数，默认为 0.210。

反带

设置反带级别。如果启用，过滤器将随机改变输出像素的亮度，以避免直方图出现条带。可能的值为none 、 weak或strong 。它默认为无。

### 39.117 直方图

计算并绘制输入视频的颜色分布直方图。

计算出的直方图表示图像中的颜色分量分布。

标准直方图显示图像中颜色成分的分布。显示每个颜色分量的颜色图。根据输入格式，显示当前帧中 Y、U、V、A 或 R、G、B 分量的分布。在每张图下方显示了一个颜色分量比例表。

过滤器接受以下选项：

level\_height

设置水平高度。默认值为200 。允许的范围是 [50, 2048]。

比例尺高度

设置色标的高度。默认值为12 。允许的范围是 [0, 40]。

显示模式

设置显示模式。它接受以下值：

'堆栈'

每个颜色分量图都放在彼此下面。

'游行'

每个颜色分量图并排放置。

'覆盖'

parade中的信息相同，除了表示颜色分量的图表直接叠加在彼此之上。

默认是堆栈。

关卡模式

设置模式。可以是线性的，也可以是对数的。默认是线性的。

成分

设置要显示的颜色分量。默认值为7 。

透明度

设置前景不透明度。默认值为0.7 。

透明度

设置背景不透明度。默认值为0.5 。

颜色模式

设置颜色模式。它接受以下值：

'白底黑字'

'黑对白'

'白灰'

'黑灰'

' coloronblack '

' coloronwhite '

' colorongray '

'黑色'

'白色'

'灰色'

默认为whiteonblack 。

* [例子](#Examples-125)

#### 39.117.1 示例

* 计算并绘制直方图：
* ffplay -i 输入 -vf 直方图

### 39.118 hqdn3d

这是一个高精度/高质量的 3d 降噪滤波器。它旨在减少图像噪声，产生平滑的图像并使静止图像真正静止。它应该增强可压缩性。

它接受以下可选参数：

空间亮度

指定空间亮度强度的非负浮点数。它默认为 4.0。

色度空间

指定空间色度强度的非负浮点数。它默认为 3.0\* luma\_spatial /4.0。

luma\_tmp

指定亮度时间强度的浮点数。它默认为 6.0\* luma\_spatial /4.0。

色度\_tmp

指定色度时间强度的浮点数。它默认为luma\_tmp \* chroma\_spatial / luma\_spatial 。

* [命令](#Commands-97)

#### 39.118.1 命令

此过滤器支持与选项相同的[命令](#commands)。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.119 硬件下载

将硬件框架下载到系统内存。

输入必须是硬件帧，输出是非硬件格式。并非所有格式都在输出上受支持 - 可能需要在图表中紧随其后插入一个额外的格式过滤器，以获得受支持格式的输出。

### 39.120 硬件地图

将硬件帧映射到系统内存或另一个设备。

该过滤器有几种不同的操作模式；使用哪一个取决于输入和输出格式：

* 硬件帧输入，普通帧输出

将输入帧映射到系统内存并将它们传递到输出。如果以后需要原始硬件框架（例如，在其中的一部分上覆盖其他东西之后），可以在下一个模式中再次使用hwmap过滤器来检索它。

* 普通帧输入，硬件帧输出

如果输入实际上是软件映射的硬件框架，则取消映射它——即返回原始硬件框架。

否则，必须提供设备。在该设备上为输出创建新的硬件表面，然后将它们映射回输入端的软件格式，并将这些帧提供给前面的过滤器。这将像hwupload过滤器一样工作，但当输入已经是兼容格式时，可能能够避免额外的副本。

* 硬件帧输入输出

derive\_device选项为输出提供设备。输入和输出设备必须是不同类型和兼容的 - 其确切含义取决于系统，但通常意味着它们必须引用相同的底层硬件上下文（例如，引用相同的图形卡）。

如果输入帧最初是在输出设备上创建的，则取消映射以检索原始帧。

否则，将帧映射到输出设备——在与输入帧相对应的输出上创建新的硬件帧。

接受以下附加参数：

模式

设置帧映射模式。一些组合：

读

映射的框架应该是可读的。

写

映射的框架应该是可写的。

覆盖

映射将始终覆盖整个帧。

这在某些情况下可能会提高性能，因为不需要加载框架的原始内容。

直接的

映射不得涉及任何复制。

在某些情况下，在无法直接映射或具有意外属性的情况下，会创建到帧副本的间接映射。设置此标志可确保映射是直接的，如果不可能则将失败。

如果未指定，则默认为读+写。

派生\_设备类型

不是使用初始化时提供的设备，而是从输入帧所在的设备派生类型类型的新设备。

撤销

在硬件到硬件的映射中，反向映射——在接收器中创建帧并将它们映射回源。在某些情况下，这可能是必要的，其中需要一个方向的映射，但所使用的设备仅支持相反的方向。

这个选项很危险——如果过滤器的输出有任何额外的限制，它可能会以未定义的方式破坏前面的过滤器。请勿在未完全理解其使用含义的情况下使用它。

### 39.121 hwupload

将系统内存帧上传到硬件表面。

初始化过滤器时必须提供要上传到的设备。如果使用 ffmpeg，请使用-filter\_hw\_device选项或derive\_device选项选择适当的设备。输入和输出设备必须是不同类型和兼容的 - 其确切含义取决于系统，但通常意味着它们必须引用相同的底层硬件上下文（例如，引用相同的图形卡）。

接受以下附加参数：

派生\_设备类型

不是使用初始化时提供的设备，而是从输入帧所在的设备派生类型类型的新设备。

### 39.122 hwupload\_cuda

将系统内存帧上传到 CUDA 设备。

它接受以下可选参数：

设备

要使用的 CUDA 设备的编号

### 39.123 hqx

应用专为像素艺术设计的高质量放大滤镜。这个过滤器最初是由 Maxim Stepin 创建的。

它接受以下选项：

n

设置缩放维度： hq2x为2 ， hq3x为3 ， hq4x为4 。默认值为3 。

### 39.124 堆栈

水平堆叠输入视频。

所有流必须具有相同的像素格式和相同的高度。

请注意，此过滤器比使用[覆盖](#overlay)和[垫](#pad)过滤器创建相同的输出更快。

过滤器接受以下选项：

输入

设置输入流的数量。默认值为 2。

最短的

如果设置为 1，则强制输出在最短输入终止时终止。默认值为 0。

### 39.125 hsvhold

将一定的 HSV 范围转换为灰度值。

此过滤器测量选项中设置的 HSV 颜色与视频流中测量的颜色之间的色差。根据选项，输出颜色可以更改为灰色或非灰色。

过滤器接受以下选项：

色调

设置将用于色差计算的色调值。允许的范围是从 -360 到 360。默认值为 0。

坐着

设置将用于色差计算的饱和度值。允许的范围是从 -1 到 1。默认值为 0。

值

设置将用于色差计算的值。允许的范围是从 -1 到 1。默认值为 0。

相似

设置与关键颜色的相似度百分比。允许的范围是从 0 到 1。默认值为 0.01。

0.00001 只匹配准确的键颜色，而 1.0 匹配所有颜色。

混合

混合百分比。允许的范围是从 0 到 1。默认值为 0。

0.0 使像素完全变灰或完全不变灰。

值越高，灰色像素越多，灰色像素越高，像素颜色与关键颜色越相似。

### 39.126 hsvkey

将一定的 HSV 范围变成透明度。

此过滤器测量选项中设置的 HSV 颜色与视频流中测量的颜色之间的色差。根据选项，可以通过添加 alpha 通道将输出颜色更改为透明。

过滤器接受以下选项：

色调

设置将用于色差计算的色调值。允许的范围是从 -360 到 360。默认值为 0。

坐着

设置将用于色差计算的饱和度值。允许的范围是从 -1 到 1。默认值为 0。

值

设置将用于色差计算的值。允许的范围是从 -1 到 1。默认值为 0。

相似

设置与关键颜色的相似度百分比。允许的范围是从 0 到 1。默认值为 0.01。

0.00001 只匹配准确的键颜色，而 1.0 匹配所有颜色。

混合

混合百分比。允许的范围是从 0 到 1。默认值为 0。

0.0 使像素完全透明或完全不透明。

较高的值导致半透明像素，透明度越高，像素颜色与键颜色越相似。

### 39.127色相

修改输入的色调和/或饱和度。

它接受以下参数：

H

将色调角指定为度数。它接受一个表达式，默认为“0”。

秒

在 [-10,10] 范围内指定饱和度。它接受一个表达式并默认为“1”。

H

将色调角指定为弧度数。它接受一个表达式，默认为“0”。

b

在 [-10,10] 范围内指定亮度。它接受一个表达式并默认为“0”。

h和H互斥，不能同时指定。

b 、 h 、 H和s选项值是包含以下常量的表达式：

n

输入帧的帧数从0开始

积分

输入帧的表示时间戳，以时基单位表示

r

输入视频的帧率，如果输入帧率未知则为 NAN

吨

以秒表示的时间戳，如果输入时间戳未知则为 NAN

结核病

输入视频的时基

* [例子](#Examples-126)
* [命令](#Commands-98)

#### 39.127.1 示例

* 将色调设置为 90 度，将饱和度设置为 1.0：
* 色调=h=90:s=1
* 相同的命令，但以弧度表示色调：
* 色调=H=PI/2:S=1
* 旋转色调并使饱和度在 1 秒内在 0 和 2 之间摆动：
* 色调="H=2\*PI\*t: s=sin(2\*PI\*t)+1"
* 从 0 开始应用 3 秒饱和淡入效果：
* 色调=“s=min(t/3\,1)”

一般的淡入表达式可以写成：

hue="s=min(0\, max((t-START)/DURATION\, 1))"

* 从 5 秒开始应用 3 秒饱和淡出效果：
* 色调=“s=max(0\, min(1\, (8-t)/3))”

一般的淡出表达式可以写成：

hue="s=max(0\, min(1\, (START+DURATION-t)/DURATION))"

#### 39.127.2 命令

该过滤器支持以下命令：

b

秒

H

H

修改输入视频的色调和/或饱和度和/或亮度。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.128 色调饱和度

对输入视频流应用色调-饱和度-强度调整。

此滤镜在 RGB 色彩空间中运行。

此过滤器接受以下选项：

色调

设置要应用的色调偏移度数。默认值为 0。允许的范围为 -180 到 180。

饱和

设置饱和度偏移。默认值为 0。允许的范围是从 -1 到 1。

强度

设置强度偏移。默认值为 0。允许的范围是从 -1 到 1。

颜色

设置要调整的原色和补色。此选项通过提供一个或多个值来设置。这可以一次选择多种颜色。默认情况下选择所有颜色。

' r '

调整红色。

'是'

调整黄色。

' g '

调整果岭。

' c '

调整青色。

' b '

调整布鲁斯。

'米'

调整洋红色。

'一个'

调整所有颜色。

力量

设置过滤强度。允许的范围是从 0 到 100。默认值为 1。

rw, gw, bw

为每个 RGB 组件设置权重。允许的范围是从 0 到 1。默认设置为 0.333、0.334、0.333。这些选项用于饱和度和无亮度处理。

亮度

设置保留亮度，默认情况下是禁用的。调整色调可以改变原始 RGB 三元组的亮度，启用此选项后亮度保持相同值。

### 39.129 迟滞

通过连接组件将第一个流扩展为第二个流。这使得构建更强大的边缘掩码成为可能。

此过滤器接受以下选项：

飞机

设置哪些平面将被处理为位图，未处理的平面将从第一个流中复制。默认值 0xf，将处理所有平面。

临界点

设置用于过滤的阈值。如果像素组件值高于此值，则用于连接组件的过滤器算法将被激活。默认值为 0。

滞后滤波器还支持[帧同步](#framesync)选项。

### 39.130 iccdetect

从嵌入式 ICC 配置文件（如果存在）中检测色彩空间，并相应地更新帧的标签。

此过滤器接受以下选项：

力量

如果为真，框架的现有色彩空间标签将始终被从 ICC 配置文件中检测到的值覆盖。否则，只有当它们包含unknown时才会被分配。默认启用。

### 39.131 iccgen

生成 ICC 配置文件并将它们附加到框架。

此过滤器接受以下选项：

原色

颜色\_trc

配置将为其生成 ICC 配置文件的色彩空间。 auto的默认值从输入帧的元数据中推断出值，默认为 BT.709/sRGB。

请参阅[setparams](#setparams)过滤器以获取可能值的列表，但请注意unknown不是此过滤器的有效值。

力量

如果为 true，将生成一个 ICC 配置文件，即使它会覆盖已经存在的 ICC 配置文件。默认情况下禁用。

### 39.132 身份

获得两个输入视频之间的身份分数。

这个过滤器需要两个输入视频。

两个输入视频必须具有相同的分辨率和像素格式，此过滤器才能正常工作。它还假设两个输入具有相同数量的帧，将它们一一比较。

获得的每个组件、平均、最小和最大身份分数通过日志系统打印。

过滤器将计算出的每个帧的身份分数存储在帧元数据中。

此过滤器还支持[帧同步](#framesync)选项。

在下面的示例中，将正在处理的输入文件main.mpg与参考文件ref.mpg进行比较。

ffmpeg -i main.mpg -i ref.mpg -lavfi identity -f null -

### 39.133 艾特

检测视频隔行扫描类型。

此过滤器尝试检测输入帧是隔行、逐行、顶场还是底场优先。它还将尝试检测相邻帧之间重复的场（电视电影的标志）。

单帧检测在对每个帧进行分类时仅考虑紧邻的帧。多帧检测结合了先前帧的分类历史。

过滤器将记录这些元数据值：

single.current\_frame

使用单帧检测检测到的当前帧类型。其中之一：“tff”（顶场优先）、“bff”（底场优先）、“progressive”或“undetermined”

单.tff

使用单帧检测检测为顶场优先的累计帧数。

多个.tff

使用多帧检测检测为顶场优先的累计帧数。

单身.bff

使用单帧检测检测为底场优先的累计帧数。

多个.current\_frame

使用多帧检测检测到的当前帧类型。其中之一：“tff”（顶场优先）、“bff”（底场优先）、“progressive”或“undetermined”

多重.bff

使用多帧检测检测为底场优先的累计帧数。

单一渐进式

使用单帧检测检测为渐进式的累计帧数。

多重渐进

使用多帧检测检测为渐进式的累计帧数。

单身未定

使用单帧检测无法分类的累计帧数。

多重未定

无法使用多帧检测进行分类的累计帧数。

重复.current\_frame

当前帧中的哪个字段是从上一个重复的。 “两者都不”、“顶部”或“底部”之一。

重复。都没有

无重复场的累计帧数。

重复.top

顶场从前一帧的顶场重复的帧的累积数量。

重复.底部

从前一帧的底场重复底场的累积帧数。

过滤器接受以下选项：

国际阈值

设置交错阈值。

prog\_thres

设置渐进阈值。

rep\_thres

重复场检测的阈值。

半衰期

给定帧对统计数据的贡献减半后的帧数（即，它仅对其分类贡献 0.5）。默认值 0 意味着所有看到的帧都永远被赋予 1.0 的完整权重。

analyze\_interlaced\_flag

当这不是 0 时，idet 将使用指定的帧数来确定隔行标志是否准确，它不会计算未确定的帧。如果发现标志是准确的，它将被使用而不需要任何进一步的计算，如果发现它不准确，它将被清除而不需要任何进一步的计算。这允许插入 idet 过滤器作为一种低计算方法来清理隔行扫描标志

### 39.134 伊尔

去交错或交错字段。

该过滤器允许人们处理交错图像场而不去交错它们。去交织将输入帧分成 2 个场（所谓的半画面）。奇数行移动到输出图像的上半部分，偶数行移动到下半部分。您可以独立处理（过滤）它们，然后重新交织它们。

过滤器接受以下选项：

亮度模式，l

色度模式，c

alpha\_mode，一个

luma\_mode 、 chroma\_mode和alpha\_mode的可用值是：

'无'

没做什么。

'去交织，d '

去交错字段，将一个字段放在另一个上方。

'交织，我'

交错字段。反转去交织的效果。

默认值为无。

亮度交换，ls

色度交换，CS

alpha\_swap，作为

交换 luma/chroma/alpha 域。交换偶数和奇数行。默认值为0 。

* [命令](#Commands-99)

#### 39.134.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.135 充气

对视频应用膨胀效果。

此过滤器通过仅考虑高于像素的值，将像素替换为局部 (3x3) 平均值。

它接受以下选项：

阈值0

阈值1

阈值2

阈值3

限制每个平面的最大变化，默认为 65535。如果为 0，平面将保持不变。

* [命令](#Commands-100)

#### 39.135.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.136 交错

来自渐进式内容的简单交错过滤器。这将奇数帧的上（或下）行与偶数帧的下（或上）行交错，将帧速率减半并保持图像高度。

原装原装新车架

帧 'j' 帧 'j+1' (tff)

=======================================

第 0 行 ----------------------> 帧“j”第 0 行

第 1 行第 1 行 ----> 帧 'j+1' 第 1 行

第 2 行 ----------------------> 帧“j”第 2 行

第 3 行第 3 行 ----> 帧 'j+1' 第 3 行

…………

新Frame + 1 将由Frame 'j+2' 和Frame 'j+3' 等生成

它接受以下可选参数：

扫描

这决定了隔行帧是从逐行帧的偶数（tff - 默认）还是奇数（bff）行中获取。

低通

垂直低通滤波器可避免 twitter 交错并减少波纹图案。

' 0，关闭'

禁用垂直低通滤波器

' 1，线性'

启用线性滤波器（默认）

' 2、复杂'

启用复杂过滤器。这将略微减少 twitter 和莫尔条纹，但更好地保留细节和主观清晰度印象。

### 39.137 核心

通过应用 Donald Graft 的自适应内核反交错来反交错输入视频。处理视频的交错部分以生成渐进帧。

接受的参数的描述如下。

脱粒

在确定是否必须处理像素线时，设置影响过滤器容差的阈值。它必须是 [0,255] 范围内的整数，默认为 10。值为 0 将导致在每个像素上应用该过程。

地图

如果设置为 1，则将超过阈值的像素绘制为白色。默认值为 0。

命令

设置字段顺序。如果设置为 1，则交换字段；如果设置为 0，则保留字段。默认值为 0。

锋利的

如果设置为 1，则启用额外锐化。默认值为 0。

双向

如果设置为 1，则启用双向锐化。默认值为 0。

* [例子](#Examples-127)

#### 39.137.1 示例

* 应用默认值：
* kerndeint=thresh=10:map=0:order=0:sharp=0:twoway=0
* 启用额外的锐化：
* kerndeint=尖锐=1
* 将处理过的像素涂成白色：
* 内核设计=地图=1

### 39.138 樱桃酒

将 kirsch 运算符应用于输入视频流。

过滤器接受以下选项：

飞机

设置要处理的平面，未处理的平面将被复制。默认值 0xf，将处理所有平面。

规模

设置将与过滤结果相乘的值。

三角洲

设置将添加到过滤结果的值。

* [命令](#Commands-101)

#### 39.138.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.139 拉格芬

慢慢更新较暗的像素。

此滤镜使短时闪光显得更长。此过滤器接受以下选项：

衰变

设置衰减因子。默认值为 0.95。允许的范围是从 0 到 1。

飞机

设置要过滤的平面。默认为全部。允许的范围是从 0 到 15。

* [命令](#Commands-102)

#### 39.139.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.140 镜头校正

校正径向镜头畸变

该滤镜可用于校正使用广角镜头可能导致的径向失真，从而重新校正图像。要找到正确的参数，可以使用可用的工具，例如作为 opencv 的一部分或简单地反复试验。要使用 opencv，请使用来自 opencv 源的校准样本（在 samples/cpp 下），并从结果矩阵中提取 k1 和 k2 系数。

请注意，KDE 项目的开源工具 Krita 和 Digikam 中实际上提供了相同的过滤器。

也可用于补偿镜头误差的[晕影滤镜](#vignette)相比，此滤镜可校正图像的失真，而[晕影](#vignette)可校正亮度分布，因此在某些情况下您可能希望同时使用这两种滤镜，尽管您会有照顾顺序，即是否应该在镜头校正之前或之后应用渐晕。

* [选项](#Options-93)
* [命令](#Commands-103)

#### 39.140.1 选项

过滤器接受以下选项：

cx

图像焦点的相对 x 坐标，也就是失真的中心。该值的范围为 [0,1]，表示为图像宽度的分数。默认值为 0.5。

赛

图像焦点的相对 y 坐标，也就是失真的中心。该值的范围为 [0,1]，表示为图像高度的分数。默认值为 0.5。

k1

二次校正项的系数。该值的范围为 [-1,1]。 0 表示没有修正。默认为 0。

k2

双二次校正项的系数。该值的范围为 [-1,1]。 0 表示没有修正。默认为 0。

我

设置插值类型。可以是最近的或双线性的。默认是最近的。

fc

指定未映射像素的颜色。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。默认颜色为black@0 。

生成修正的公式是：

r\_src = r\_tgt \* (1 + k1 \* ( r\_tgt / r\_0 )^2 + k2 \* ( r\_tgt / r\_0 )^4)

其中r\_0是图像对角线的一半， r\_src和r\_tgt分别是源图像和目标图像中距焦点的距离。

#### 39.140.2 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.141 镜头乐趣

通过 lensfun 库 ( <http://lensfun.sourceforge.net/>) 应用镜头校正。

lensfun滤镜需要相机品牌、相机型号和镜头型号才能应用镜头校正。过滤器将加载 lensfun 数据库并查询它以在数据库中找到相应的相机和镜头条目。只要可以使用给定的选项找到这些条目，过滤器就可以对帧进行校正。请注意，不完整的字符串将导致过滤器选择与给定选项的最佳匹配，并且过滤器将输出所选的相机和镜头型号（以“信息”级别记录）。您必须按要求提供品牌、相机型号和镜头型号。

要获得可用品牌和型号的列表，请省略一个或两个品牌和型号选项。过滤器会将完整列表发送到级别为INFO 的日志。第一列是品牌，第二列是型号。要获得可用镜头的列表，请为制造商和型号设置任何值并省略lens\_model选项。过滤器将在日志中发送完整的镜头列表，级别为INFO 。打印列表后，ffmpeg 工具将退出。

过滤器接受以下选项：

制作

相机品牌（例如“佳能”）。此选项是必需的。

模型

相机型号（例如“Canon EOS 100D”）。此选项是必需的。

镜头型号

镜头型号（例如，“Canon EF-S 18-55mm f/3.5-5.6 IS STM”）。此选项是必需的。

数据库路径

镜头数据库文件夹的完整路径。如果未设置，过滤器将尝试从构建库时的安装路径加载数据库。默认未设置。

模式

要应用的更正类型。以下值是有效选项：

'渐晕'

启用固定镜头渐晕。

'几何'

启用固定镜头几何形状。这是默认设置。

'子像素'

启用修复色差。

' vig\_geo '

启用修复镜头渐晕和镜头几何形状。

' vig\_subpixel '

启用修复镜头渐晕和色差。

'失真'

可以修复镜头几何形状和色差。

'全部'

启用所有可能的更正。

焦距

图像/视频的焦距（缩放；视频的预期常数）。例如，18-55mm 镜头的焦距范围为 [18-55]，因此在使用该镜头时应选择该范围内的值。默认 18。

光圈

图像/视频的光圈（视频的预期常数）。请注意，光圈仅用于渐晕校正。默认 3.5。

焦点距离

图像/视频的焦距（视频的预期常数）。请注意，焦距仅用于渐晕，并且只会轻微影响渐晕校正过程。如果未知，请将其保留为默认值（即 1000）。

规模

转换后应用的比例因子。校正后视频不再一定是矩形的。此参数控制生成的图像中有多少是可见的。值 0 表示将自动选择一个值，以便在输出图像中几乎没有或没有未映射区域。 1.0 表示不进行额外缩放。较低的值可能会导致更多的校正图像可见，而较高的值可能会避免输出中未映射的区域。

目标几何

输出图像/视频的目标几何。以下值是有效选项：

'直线（默认） '

'鱼眼'

'全景'

'等矩形'

'鱼眼正字法'

' fisheye\_stereographic '

' fisheye\_equisolid '

' fisheye\_thoby '

撤销

应用图像校正的反向（而不是校正失真，应用它）。

插值

校正失真时使用的插值类型。以下值是有效选项：

'最近的'

'线性（默认） '

'兰佐斯'

* [例子](#Examples-128)

#### 39.141.1 示例

* 使用品牌“佳能”、相机型号“佳能 EOS 100D”和镜头型号“佳能 EF-S 18-55mm f/3.5-5.6 IS STM”进行镜头校正，焦距为“18”，光圈为“8.0”。
* ffmpeg -i input.mov -vf lensfun=make=Canon:model="佳能 EOS 100D":lens\_model="佳能 EF-S 18-55mm f/3.5-5.6 IS STM":focal\_length=18:aperture=8 -c :v h264 -b:v 8000k 输出.mov
* 与之前一样应用，但仅适用于视频的前 5 秒。
* ffmpeg -i input.mov -vf lensfun=make=Canon:model="佳能 EOS 100D":lens\_model="佳能 EF-S 18-55mm f/3.5-5.6 IS STM":focal\_length=18:aperture=8:enable ='lte(t\,5)' -c:v h264 -b:v 8000k 输出.mov

### 39.142 lib安慰剂

基于 libplacebo ( <https://code.videolan.org/videolan/libplacebo>) 的灵活 GPU 加速处理过滤器。请注意，此过滤器当前仅接受 Vulkan 输入帧。

* [选项](#Options-94)
* [命令](#Commands-104)
* [例子](#Examples-129)

#### 39.142.1 选项

此过滤器的选项分为以下部分：

* [输出方式](#Output-mode)
* [缩放](#Scaling)
* [解带](#Debanding)
* [颜色调整](#Color-adjustment)
* [峰值检测](#Peak-detection)
* [色调映射](#Tone-mapping)
* [抖动](#Dithering)
* [自定义着色器](#Custom-shaders)
* [调试/性能](#Debugging-_002f-performance)

#### 39.142.1.1 输出模式

这些选项控制整体输出模式。默认情况下，libplacebo 会尽量保留源色度和尺寸，但它会应用任何嵌入的胶片颗粒、杜比视界元数据或源帧中存在的变形 SAR。

w

H

设置输出视频维度表达式。默认值是输入维度。

[比例过滤器](#scale)相同的表达式。

格式

设置输出格式覆盖。如果未设置（默认），帧将以与相应输入帧相同的格式输出。否则，将执行格式转换。

force\_original\_aspect\_ratio

force\_divisible\_by

[比例过滤器选项的](#scale)工作方式相同。

规范化\_sar

如果启用，输出帧将始终具有 1:1 的像素纵横比。这将根据需要引入填充/裁剪。如果禁用（默认设置），任何宽高比不匹配，包括来自变形视频源的宽高比，都会转发到输出像素宽高比。

pad\_crop\_ratio

当输入纵横比与输出纵横比不匹配并且normalize\_sar生效时，指定填充和裁剪之间的比率（介于0.0和1.0之间）。默认值0.0总是用黑色边框填充内容，而值1.0总是裁掉部分内容。中间值是可能的，导致两种方法的混合。

色彩空间

原色

颜色\_trc

范围

配置输出帧将在其中传递的颜色空间。自动输出帧的默认值与输入帧的格式相同，不会发生变化。对于任何其他值，将执行转换。

有关可能值的列表，请参阅[setparams过滤器。](#setparams)

apply\_filmgrain

如果源帧中存在胶片颗粒（例如 AV1 或 H.274），并从输出中去除它。默认启用。

申请杜比视界

如果源帧中存在杜比视界 RPU 元数据，则应用杜比视界 RPU 元数据，并将其从输出中剥离。默认启用。请注意，杜比视界将始终输出 BT.2020+PQ，覆盖通常的输入帧元数据。这些也将被选为相应帧输出选项的auto值。

#### 39.142.1.2 缩放

本节中的选项控制 libplacebo 如何执行放大和（如果需要）缩小。请注意，libplacebo 将始终在内部对 4:4:4 内容进行操作，因此任何子采样色度格式（例如yuv420p）都必须作为渲染过程的一部分进行上采样和下采样。这意味着即使源分辨率和目标分辨率相同，缩放也可能有效。

升频器

降频器

配置用于放大和缩小的滤波器内核。各自的默认值是spline36和mitchell 。有关可能值的完整列表，请将帮助传递给这些选项。最重要的价值观是：

'无'

强制使用内置 GPU 纹理采样（通常是双线性）。速度极快但质量很差，尤其是在缩小比例时。

'双线性'

双线性插值。通常可以在 GPU 上免费完成，除非这样做会导致锯齿。速度快，质量低。

'最近的'

最近邻插值。锐利但高度锯齿。

'过采样'

视觉上看起来类似于最近邻插值但试图保持像素纵横比的算法。适合像素艺术，因为它会导致艺术外观的失真最小。

'兰佐斯'

标准 sinc-sinc 插值内核。

'样条36 '

lanczos 的三次样条逼近。性能没有差异，但振铃声略有减少。

' ewa\_lanczos '

基于 jinc-sinc 内核的 lanczos 的椭圆加权平均版本。这也普遍称为“Jinc 缩放”。速度慢但质量很高。

'高斯'

高斯核。具有某些理想的数学性质，但主观上非常模糊。

'米切尔'

具有 Mitchell 和 Netravali 推荐参数的三次 BC 样条。很少有铃声。

lut\_entries

配置定标器 LUT 的大小，范围从1到256 。默认值0将选择 libplacebo 的内部默认值，通常为64 。

抗振铃

启用抗振铃（对于非 EWA 过滤器）。该值（介于0.0和1.0之间）配置抗振铃算法的强度。如果设置得太高可能会增加锯齿。默认情况下禁用。

乙状结肠

在放大期间启用 S 形压缩。稍微减少振铃。默认启用。

#### 39.142.1.3 解带

Libplacebo 带有一个内置的去频带过滤器，可以很好地抵消许多常见的条带和阻塞源。强烈建议在需要质量时打开此功能。

解带

启用（快速）去频带算法。默认情况下禁用。

deband\_iterations

去频带算法的去频带迭代次数。每次迭代都以逐渐增加的半径（和减小的阈值）执行。推荐值在1到4的范围内。默认为1 。

deband\_threshold

去带滤波器强度。更高的数字会导致更积极的去带。默认为4.0 。

deband\_radius

去带滤波器半径。较高的半径适合慢速梯度，而较小的半径适合陡峭的梯度。默认为16.0 。

deband\_grain

要添加的额外输出谷物的数量。有助于隐藏瑕疵。默认为6.0 。

#### 39.142.1.4 颜色调整

主观色彩控制的集合。不是很严格，所以确切的效果会因输入原色和色彩空间而有所不同。

亮度

亮度提升，介于-1.0和1.0之间。默认为0.0 。

对比

对比度增益，介于0.0和16.0之间。默认为1.0 。

饱和

饱和度增益，介于0.0和16.0之间。默认为1.0 。

色调

以弧度为单位的色调偏移，介于-3.14和3.14之间。默认为0.0 。这将旋转 UV 子向量，默认为 RGB 输入的 BT.709 系数。

伽马

Gamma 调整，介于0.0和16.0之间。默认为1.0 。

视锥细胞

用于色盲模拟的锥体模型。接受l 、 m和s的任意组合。这里有些例子：

'米'

绿色畸形/绿色盲（影响 3%-4% 的人口）

'我'

红色盲/红色盲（影响 1%-2% 的人口）

' l+m '

单色性（非常罕见）

' l+m+s '

色盲（白天视力完全丧失，极为罕见）

锥体强度

cones指定的锥体的增益因子，介于0.0和10.0之间。值为1.0不会导致色觉发生变化。值0.0 （默认值）模拟这些视锥细胞的完全丢失。高于1.0 的值会导致视锥细胞之间的差异被放大，这可能有助于补偿色觉下降。

#### 39.142.1.5 峰值检测

为了帮助处理只有静态 HDR10 元数据（或没有任何标记）的源，libplacebo 使用其自己的内部帧分析计算着色器来分析源帧并实时调整色调映射功能。如果这太慢，或者如果需要完全可重现的帧完美结果，建议关闭此功能。

峰值检测

启用 HDR 峰值检测。忽略静态 MaxCLL/MaxFALL 值，支持从输入进行动态检测。请注意，检测到的值不会写回输出帧，它们仅指导内部色调映射过程。默认启用。

平滑周期

峰值检测平滑周期，介于0.0和1000.0之间。较高的值会导致峰值检测对输入变化的响应变差。默认为100.0 。

最小峰值

检测到的峰值的下限（相对于 SDR 白色），介于0.0和100.0之间。默认为1.0 。

场景阈值低

场景阈值高

场景变化检测的下限和上限。以0.0和100.0之间的对数刻度表示。分别默认为5.5和10.0 。将其中一个设置为负值会禁用此功能。

过冲

峰值平滑过冲裕度，介于0.0和1.0之间。提供安全余量以防止峰值平滑导致的削波。默认为0.05 ，对应于 5% 的边距。

#### 39.142.1.6 色调映射

本节中的选项控制 libplacebo 在处理广色域或 HDR 内容之间的不匹配时如何执行色调映射和色域映射。一般来说，libplacebo 依靠准确的源标记和掌握显示色域信息来产生最佳结果。

意图

在不同的原色色域之间进行调整时使用的渲染意图（在色调映射之后）。

'感性'

感知色域映射。目前相当于相对比色法。

'亲戚'

相对比色。这是默认设置。

'绝对'

绝对比色法。

'饱和'

饱和度映射。强制将源色域拉伸到目标色域。

色域模式

如何处理色度色域映射可能导致的色域外颜色。

'剪辑'

什么也不做，只需将超出范围的颜色剪辑到 RGB 体积。这是默认设置。

'警告'

突出显示色域外的像素（通过将它们着色为粉红色）。

'变暗'

线性降低内容亮度以保留饱和的细节，然后剪裁剩余的色域外颜色。顾名思义，这会使一切变暗，但在保留细节和颜色之间提供了良好的平衡。

'去饱和'

将色域外的颜色硬去饱和为白色，同时保持亮度。有变色倾向。

色调映射

要使用的色调映射算法。可用值是：

'汽车'

基于内部启发式的自动选择。这是默认设置。

'剪辑'

不执行色调映射，仅剪辑超出范围的颜色。为范围内的颜色保留完美的颜色精度，但完全破坏范围外的信息。不执行任何黑点自适应。不可配置。

' st2094-40 '

SMPTE ST 2094-40 Annex B 中的 EETF，它基于贝塞尔曲线应用来自 HDR10+ 动态元数据的贝塞尔曲线来执行色调映射。所使用的 OOTF 是根据目标和实际显示峰值亮度之间的比率进行调整的。

' st2094-10 '

来自 SMPTE ST 2094-10 附件 B.2 的 EETF，除了最大/最小值之外，它还考虑了输入信号的平均亮度。可配置的对比度参数影响线性输出段的斜率，默认为1.0表示不增加/减少对比度。请注意，这目前不包括附件 B.3 中定义的主观增益/偏移/伽马控制。

' bt.2390 '

ITU-R 报告 BT.2390 中的 EETF，一种带有线性段的厄米样条滚降。拐点偏移是可配置的。请注意，此参数默认为1.0 ，而不是ITU-R 规范中的值0.5 。

' bt.2446a '

ITU-R 报告 BT.2446 中的 EETF，方法 A。专为良好掌握的 HDR 源而设计。可用于正向和反向色调映射。不可配置。

'样条'

由两个多项式组成的简单样条，由单个枢轴点连接。该参数给出了枢轴点（在 PQ 空间中），默认为0.30 。可用于正向和反向色调映射。

'莱茵哈德'

简单的非线性、全局色调映射算法。该参数指定显示峰值处的局部对比度系数。本质上， 0.5的参数意味着参考白色的亮度大约是剪裁时的一半。默认为0.5 ，这导致此函数的最简单公式。

'莫比乌斯'

泛化 reinhard 色调映射算法以支持接近黑色的附加线性斜率。色调映射参数表示线性部分和非线性部分之间的权衡。本质上，对于给定的参数x ， x以下的每个颜色值都将被线性映射，而较高的值将被非线性色调映射。接近1.0的值使该曲线表现得像clip ，而接近0.0 的值使该曲线表现得像reinhard 。默认值为0.3 ，它在色度精度和保留色域外细节之间提供了良好的平衡。

'有能力'

由 John Hable 开发的用于 Uncharted 2 的分段电影色调映射算法，灵感来自柯达使用的类似色调映射算法。因其在具有 HDR 渲染功能的视频游戏中的使用而流行。很好地保留了暗部和亮部细节，但缺点是平均亮度会发生显着变化。这有点类似于参数为0.24的reinhard 。

'伽玛'

拟合伽玛（幂）函数以在源色彩空间和目标色彩空间之间传输，有效地导致感知硬拐点连接两个大致线性的部分。这可以相当准确地保留所有比例的细节，但会导致图像显得柔和或暗淡。该参数用作截止点，默认为0.5 。

'线性'

在 PQ 空间中将输入范围线性扩展到输出范围。这将准确地保留所有细节，但会导致明显不同的平均亮度。除常规色调映射外，还可用于反向色调映射。该参数可用作附加线性增益系数（默认为1.0 ）。

色调映射参数

对于可调色调映射函数，此参数可用于微调曲线行为。请参阅色调映射的文档。默认值0.0替换为曲线的首选默认设置。

色调映射模式

映射指定的色调映射函数如何应用于场景中的颜色。可能的值是：

'汽车'

基于内部启发式的自动选择。这是默认设置。

' RGB '

在 RGB 颜色空间中应用每个通道的函数。 RGB 中的每通道色调映射。保证不剪裁并严重降低输出的饱和度，但会显着扭曲颜色。非常类似于“好莱坞”的外观和感觉。

'最大'

对信号中发现的最亮分量执行色调映射。擅长保留高光中的细节，但有压垮黑色的倾向。

'混合'

2.4系数曲线，高光的每通道色调映射和中间调/阴影的线性（基于亮度） 。

'亮度'

在亮度分量 (CIE Y) 上线性色调映射，并调整（去饱和）色度以使用简单的常数因子进行补偿。这本质上是 ITU-R BT.2446 方法 A 中使用的模式。

反色调映射

如果启用，此过滤器还将尝试拉伸 SDR 信号以填充 HDR 输出颜色体积。默认情况下禁用。

色调映射\_串扰

额外的色调映射串扰因子，介于0.0和0.3之间。这有助于减少对某些明亮光谱颜色进行色调映射的问题。默认为0.04 。

tonemapping\_lut\_size

色调映射 LUT 的大小，介于2和1024之间。默认为256 。请注意，当与peak\_detect组合时，这个数字是平方的。

#### 39.142.1.7 抖动

默认情况下，libplacebo 会在必要时抖动，包括呈现为低于 16 位精度的任何整数格式。建议始终保持打开状态，因为不这样做可能会导致输出中出现可见的条带，即使启用了去频带过滤器。如果需要最大性能，请使用ordered\_fixed而不是禁用抖动。

抖动

要使用的抖动方法。接受以下值：

'无'

完全禁用抖动。可能会导致可见的条带。

'蓝色'

抖动伪蓝噪声。这是默认设置。

'订购'

可调有序抖动模式。

' ordered\_fixed '

固定大小为6的更快有序抖动。无纹理。

'白色'

用白噪声抖动。无纹理。

dither\_lut\_size

抖动 LUT 大小，作为1和8之间的对数 base2 。默认为6 ，对应于64x64的 LUT 大小。

抖动时间

启用时间抖动。默认情况下禁用。

#### 39.142.1.8 自定义着色器

libplacebo 支持许多基于 mpv .hook GLSL 语法的自定义着色器。可以在此处找到此类着色器的集合： [https ://github.com/mpv-player/mpv/wiki/User-Scripts#user-shaders](https://github.com/mpv-player/mpv/wiki/User-Scripts#user-shaders)

mpv 着色器格式的完整描述超出了本节的范围，但可以在此处找到摘要： [https ://mpv.io/manual/master/#options-glsl-shader](https://mpv.io/manual/master/#options-glsl-shader)

自定义着色器路径

指定要在运行时加载的自定义着色器文件的路径。

custom\_shader\_bin

将完整的自定义着色器指定为原始字符串。

#### 39.142.1.9 调试/性能

此部分中的所有选项默认关闭。当试图以牺牲质量为代价来获得最大性能时，它们可能会有所帮助。

跳过aa

缩小时禁用抗锯齿。

极地截止

截断极坐标 (EWA) 定标器内核低于此绝对幅度，介于0.0和1.0之间。

禁用线性

禁用线性光缩放。

disable\_builtin

禁用内置 GPU 采样（强制 LUT）。

force\_icc\_lut

强制使用完整的 ICC 3DLUT 进行色域映射。

禁用\_fbos

强制禁用 FBO，导致几乎所有功能丢失，但提供最大可能的速度。

#### 39.142.2 命令

该过滤器支持几乎所有上述选项作为[命令](#commands)。

#### 39.142.3 例子

* 有关如何初始化 Vulkan 设备、将帧上传到 GPU、执行过滤器转换为 yuv420p 以及将帧下载回 CPU 以进行输出的完整示例。请注意，在特定情况下，您可以通过指定与输入帧对应的正确格式过滤器选项来避免执行格式转换的需要。
* ffmpeg -i $INPUT -init\_hw\_device vulkan -vf hwupload,libplacebo=format=yuv420p,hwdownload,format=yuv420p $OUTPUT
* 标准色域 BT.709 输出的色调映射输入：
* libplacebo=颜色空间=bt709:color\_primaries=bt709:color\_trc=bt709:range=tv
* 重新缩放输入以适应标准 1080p，具有高质量缩放：
* libplacebo=w=1920:h=1080:force\_original\_aspect\_ratio=decrease:normalize\_sar=true:upscaler=ewa\_lanczos:downscaler=ewa\_lanczos
* 将输入转换为标准 sRGB JPEG：
* libplacebo=format=yuv420p:colorspace=bt470bg:color\_primaries=bt709:color\_trc=iec61966-2-1:range=pc
* 使用更高质量的去带设置：
* libplacebo=deband=true:deband\_iterations=3:deband\_radius=8:deband\_threshold=6
* 在安装了 Mesa 的系统（并禁用了最昂贵的选项）的 CPU 上运行此过滤器：
* ffmpeg ...-init\_hw\_device vulkan:llvmpipe ...-vf libplacebo=upscaler=none:downscaler=none:peak\_detect=false
* 抑制解码器中基于 CPU 的 AV1/H.274 电影颗粒应用，支持使用此过滤器进行处理。请注意，如果帧已经在 GPU 上，或者如果您将 libplacebo 用于其他目的，这只是一个增益，因为否则 VRAM 往返将抵消任何预期的加速。
* ffmpeg -export\_side\_data +film\_grain ... -vf libplacebo=apply\_filmgrain=true

### 39.143 libvmaf

计算参考/失真输入视频对的 VMAF（视频多方法评估融合）分数。

第一个输入是失真视频，第二个输入是参考视频。

获得的 VMAF 分数通过日志系统打印出来。

它需要 Netflix 的 vmaf 库 (libvmaf) 作为先决条件。安装库后，可以使用以下命令启用它： ./configure --enable-libvmaf 。

过滤器有以下选项：

模型

一个'|' vmaf 模型的分隔列表。每个模型都可以配置许多参数。默认值： “版本=vmaf\_v0.6.1”

模型路径

弃用，使用 model='path=...'。

启用\_转换

弃用，使用 model='enable\_transform=true'。

手机型号

弃用，使用 model='enable\_transform=true'。

enable\_conf\_interval

弃用，使用 model='enable\_conf\_interval=true'。

特征

一个'|'分隔的功能列表。每个功能都可以配置多个参数。

压力信噪比

弃用，使用 feature='name=psnr'。

西姆

弃用，使用 feature='name=ssim'。

ms\_ssim

已弃用，使用 feature='name=ms\_ssim'。

日志路径

设置用于存储日志文件的文件路径。

log\_fmt

设置日志文件的格式（xml、json、csv 或 sub）。

n\_线程

设置初始化 libvmaf 时要使用的线程数。默认值： 0 ，无线程。

n\_子样本

设置要使用的帧子采样间隔。

此过滤器还支持[帧同步](#framesync)选项。

* [例子](#Examples-130)

#### 39.143.1 示例

* 在下面的示例中，将失真视频distorted.mpg与参考文件reference.mpg进行比较。
* 基本用法：
* ffmpeg -i distorted.mpg -i reference.mpg -lavfi libvmaf=log\_path=output.xml -f null -
* 具有多个模型的示例：
* ffmpeg -i distorted.mpg -i reference.mpg -lavfi libvmaf='model=version=vmaf\_v0.6.1\\:name=vmaf|version=vmaf\_v0.6.1neg\\:name=vmaf\_neg' -f null -
* 具有多个附加功能的示例：
* ffmpeg -i distorted.mpg -i reference.mpg -lavfi libvmaf='feature=name=psnr|name=ciede' -f null -
* 具有选项和不同容器的示例：
* ffmpeg -i distorted.mpg -i reference.mkv -lavfi "[0:v]settb=AVTB,setpts=PTS-STARTPTS[main];[1:v]settb=AVTB,setpts=PTS-STARTPTS[ref]; [main][ref]libvmaf=log\_fmt=json:log\_path=output.json" -f null -

### 39.144 极限差异

使用第二个和可选的第三个视频流应用有限差异过滤器。

过滤器接受以下选项：

临界点

设置允许视频流之间存在某些差异时要使用的阈值。任何低于或恰好低于此阈值的绝对差值将从第一个视频流中选取像素分量。

弹性

在处理视频流时设置软阈值的弹性。该值乘以第一个设置第二个阈值。任何大于或精确于第二阈值的绝对差值将从第二视频流中选取像素分量。对于这两个阈值之间的值，将使用第一视频流和第二视频流之间的线性插值。

参考

启用参考（第三）视频流处理。默认情况下是禁用的。如果设置，此视频流将用于计算与第一个视频流的绝对差异。

飞机

指定将处理哪些平面。默认为所有可用。

* [命令](#Commands-105)

#### 39.144.1 命令

此过滤器支持上述所有选项作为[命令，](#commands)但选项“ reference ”除外。

### 39.145 限制器

将像素分量值限制在指定范围 [min, max]。

过滤器接受以下选项：

分钟

下限。默认为输入的最低允许值。

最大限度

上限。默认为输入的最高允许值。

飞机

指定将处理哪些平面。默认为所有可用。

* [命令](#Commands-106)

#### 39.145.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.146 循环

循环视频帧。

过滤器接受以下选项：

环形

设置循环次数。将此值设置为 -1 将导致无限循环。默认为 0。

尺寸

设置帧数的最大大小。默认为 0。

开始

设置循环的第一帧。默认为 0。

* [例子](#Examples-131)

#### 39.146.1 示例

* 无限循环单个第一帧：
* 循环=循环=-1:大小=1:开始=0
* 循环单个第一帧 10 次：
* 循环=循环=10:大小=1:开始=0
* 循环 10 个第一帧 5 次：
* 循环=循环=5:大小=10:开始=0

### 39.147 lut1d

将 1D LUT 应用于输入视频。

过滤器接受以下选项：

文件

设置 1D LUT 文件名。

目前支持的格式：

'立方体'

虹膜

' csp '

电影空间

插话

选择插值模式。

可用值是：

'最近的'

使用最近定义点的值。

'线性'

使用线性插值插值。

'余弦'

使用余弦插值法对值进行插值。

'立方体'

使用三次插值插值。

'样条'

使用样条插值插值。

* [命令](#Commands-107)

#### 39.147.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.148 lut3d

将 3D LUT 应用于输入视频。

过滤器接受以下选项：

文件

设置 3D LUT 文件名。

目前支持的格式：

' 3dl '

后遗症

'立方体'

虹膜

'达'

达芬奇

' m3d '

潘多拉

' csp '

电影空间

插话

选择插值模式。

可用值是：

'最近的'

使用最近定义点的值。

'三线'

使用定义立方体的 8 个点插入值。

'四面体'

使用四面体插值。

'金字塔'

使用金字塔插入值。

'棱镜'

使用棱镜插值。

* [命令](#Commands-108)

#### 39.148.1 命令

此过滤器支持将interp选项作为[命令](#commands)。

### 39.149 卢马基

将某些亮度值转换为透明度。

过滤器接受以下选项：

临界点

设置将用作透明度基础的亮度。默认值为0 。

宽容

设置要抠出的亮度值范围。默认值为0.01 。

柔软度

设置柔和度范围。默认值为0 。使用它来控制从零到完全透明的逐渐过渡。

* [命令](#Commands-109)

#### 39.149.1 命令

此过滤器支持与选项相同的[命令](#commands)。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.150 lut，lutrgb，lutyuv

计算用于将每个像素分量输入值绑定到输出值的查找表，并将其应用于输入视频。

lutyuv将查找表应用于 YUV 输入视频， lutrgb应用于 RGB 输入视频。

这些过滤器接受以下参数：

c0

设置第一个像素分量表达式

c1

设置第二个像素分量表达式

c2

设置第三个像素分量表达式

c3

设置第四个像素分量表达式，对应alpha分量

r

设置红色分量表达式

G

设置绿色组件表达式

b

设置蓝色分量表达式

A

alpha 组件表达式

是

设置Y/亮度分量表达式

你

设置 U/Cb 分量表达式

v

设置V/Cr分量表达式

它们中的每一个都指定了用于计算相应像素分量值的查找表的表达式。

与每个c\*选项关联的确切组件取决于输入格式。

lut过滤器需要输入 YUV 或 RGB 像素格式， lutrgb需要输入 RGB 像素格式，而lutyuv需要YUV。

表达式可以包含以下常量和函数：

w

H

输入的宽度和高度。

值

像素分量的输入值。

剪辑版

输入值，裁剪到minval - maxval范围内。

最大值

像素分量的最大值。

minval

像素分量的最小值。

负值

像素分量值的负值，裁剪到minval - maxval范围内；它对应于表达式“maxval-clipval+minval”。

剪辑（val）

val中的计算值，被限制在minval - maxval范围内。

伽玛值（伽玛）

像素分量值的计算伽玛校正值，裁剪到minval - maxval范围。它对应于表达式“pow((clipval-minval)/(maxval-minval)\, gamma )\*(maxval-minval)+minval”

所有表达式默认为“clipval”。

* [命令](#Commands-110)
* [例子](#Examples-132)

#### 39.150.1 命令

此过滤器支持与选项相同的[命令](#commands)。

#### 39.150.2 示例

* 否定输入视频：
* lutrgb="r=maxval+minval-val:g=maxval+minval-val:b=maxval+minval-val"
* lutyuv="y=maxval+minval-val:u=maxval+minval-val:v=maxval+minval-val"

以上与以下内容相同：

lutrgb="r=negval:g=negval:b=negval"

lutyuv="y=negval:u=negval:v=negval"

* 否定亮度：
* lutyuv=y=negval
* 移除色度分量，将视频变成灰度图像：
* lutyuv="u=128:v=128"
* 应用亮度燃烧效果：
* lutyuv="y=2\*val"
* 移除绿色和蓝色分量：
* lutrgb="g=0:b=0"
* 在输入上设置一个常量 alpha 通道值：
* 格式=rgba,lutrgb=a="最大最小值/2"
* 将亮度伽玛修正为 0.5 倍：
* lutyuv=y=gammaval(0.5)
* 丢弃亮度的最低有效位：
* lutyuv=y='bitand(val, 128+64+32)'
* 类似Technicolor的效果：
* lutyuv=u='(val-maxval/2)\*2+maxval/2':v='(val-maxval/2)\*2+maxval/2'

### 39.151 lut2，tlut2

lut2过滤器接受两个输入流并输出一个流。

tlut2 （time lut2）过滤器从一个流中获取两个连续的帧。

此过滤器接受以下参数：

c0

设置第一个像素分量表达式

c1

设置第二个像素分量表达式

c2

设置第三个像素分量表达式

c3

设置第四个像素分量表达式，对应alpha分量

d

设置输出位深度，仅适用于lut2过滤器。默认情况下为 0，这意味着位深度是从第一个输入格式中自动选取的。

lut2滤镜还支持[帧同步](#framesync)选项。

它们中的每一个都指定了用于计算相应像素分量值的查找表的表达式。

与每个c\*选项关联的确切组件取决于输入中的格式。

表达式可以包含以下常量：

w

H

输入的宽度和高度。

X

像素分量的第一个输入值。

是

像素分量的第二个输入值。

bdx

第一个输入视频位深度。

bdy

第二个输入视频位深度。

所有表达式默认为“x”。

* [命令](#Commands-111)
* [例子](#Examples-133)

#### 39.151.1 命令

此过滤器支持上述所有选项作为[命令，](#commands)但选项d除外。

#### 39.151.2 例子

* 突出显示两个 RGB 视频流之间的差异：
* lut2='ifnot(xy,0,pow(2,bdx)-1):ifnot(xy,0,pow(2,bdx)-1):ifnot(xy,0,pow(2,bdx)-1) '
* 突出显示两个 YUV 视频流之间的差异：
* lut2='ifnot(xy,0,pow(2,bdx)-1):ifnot(xy,pow(2,bdx-1),pow(2,bdx)-1):ifnot(xy,pow(2, bdx-1),pow(2,bdx)-1)'
* 显示两个视频流之间的最大差异：
* lut2='if(lt(x,y),0,if(gt(x,y),pow(2,bdx)-1,pow(2,bdx-1))):if(lt(x,y ),0,if(gt(x,y),pow(2,bdx)-1,pow(2,bdx-1)):if(lt(x,y),0,if(gt(x, y),pow(2,bdx)-1,pow(2,bdx-1)))'

### 39.152 蒙面钳

用第二个输入和第三个输入流夹住第一个输入流。

返回介于第二个输入流 -下冲和第三个输入流 +过冲之间的第一个流的值。

此过滤器接受以下选项：

下冲

默认值为0 。

过冲

默认值为0 。

飞机

设置哪些平面将被处理为位图，未处理的平面将从第一个流中复制。默认值 0xf，将处理所有平面。

* [命令](#Commands-112)

#### 39.152.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.153 蒙面最大

使用第二个输入流和第一个输入流之间的绝对差以及第三个输入流和第一个输入流之间的绝对差将第二个和第三个输入流合并到输出流中。如果第二个绝对差大于第一个，则选取的值将来自第二个输入流，否则来自第三个输入流。

此过滤器接受以下选项：

飞机

设置哪些平面将被处理为位图，未处理的平面将从第一个流中复制。默认值 0xf，将处理所有平面。

* [命令](#Commands-113)

#### 39.153.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.154 蒙面合并

使用第三个输入流中的每个像素权重将第一个输入流与第二个输入流合并。

第三个流像素分量中的值为 0 表示来自第一个流的像素分量未更改地返回，而最大值（例如 8 位视频为 255）表示来自第二个流的像素分量未更改返回。中间值定义两个输入流的像素分量之间的合并量。

此过滤器接受以下选项：

飞机

设置哪些平面将被处理为位图，未处理的平面将从第一个流中复制。默认值 0xf，将处理所有平面。

* [命令](#Commands-114)

#### 39.154.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.155 掩码最小值

使用第二个输入流和第一个输入流之间的绝对差以及第三个输入流和第一个输入流之间的绝对差将第二个和第三个输入流合并到输出流中。如果第二个绝对差小于第一个，则选取的值将来自第二个输入流，否则来自第三个输入流。

此过滤器接受以下选项：

飞机

设置哪些平面将被处理为位图，未处理的平面将从第一个流中复制。默认值 0xf，将处理所有平面。

* [命令](#Commands-115)

#### 39.155.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.156 屏蔽阈值

选择像素比较两个视频流与固定阈值的绝对差异。

如果第一和第二视频流的像素分量之间的绝对差等于或低于用户提供的阈值，则从第一视频流中拾取像素分量，否则从第二视频流中拾取像素分量。

此过滤器接受以下选项：

临界点

设置从两个输入视频流的绝对差异中选取像素时使用的阈值。

飞机

设置哪些平面将被处理为位图，未处理的平面将从第二个流中复制。默认值 0xf，将处理所有平面。

模式

设置过滤器操作模式。可以是abs或diff 。默认为abs 。

* [命令](#Commands-116)

#### 39.156.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.157 面具乐趣

从输入视频创建遮罩。

tblend过滤器之后创建运动蒙版很有用。

此过滤器接受以下选项：

低的

设置低阈值。任何低于或恰好低于此值的像素分量都将设置为 0。

高的

设置高门槛。任何高于此值的像素分量都将设置为当前像素格式允许的最大值。

飞机

设置要过滤的平面，默认情况下过滤所有可用的平面。

充满

用这个值填充所有帧像素。

和

设置帧的最大平均像素值。如果所有像素分量的总和高于此平均值，则输出帧将完全填充填充选项设置的值。当与tblend过滤器结合使用时，通常对场景变化很有用。

* [命令](#Commands-117)

#### 39.157.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.158 麦克迪因特

应用运动补偿去隔行扫描。

它每帧需要一个字段作为输入，因此必须与 yadif=1/3 或等效项一起使用。

此过滤器仅在 ffmpeg 版本 4.4 或更早版本中可用。

此过滤器接受以下选项：

模式

设置去隔行模式。

它接受以下值之一：

'快'

'中号'

'慢'

使用迭代运动估计

' extra\_slow '

像“慢”，但使用多个参考帧。

默认值为“快速”。

平价

设置输入视频假定的图片字段奇偶校验。它必须是以下值之一：

' 0, tff '

首先假设顶场

' 1, 好朋友'

首先假设底场

默认值为 ' bff '。

qp

设置内部编码器使用的每块量化参数 (QP)。

较高的值应导致更平滑的运动矢量场，但不是最佳的单个矢量。默认值为 1。

### 39.159 中位数

从由半径定义的某个矩形中选取中值像素。

此过滤器接受以下选项：

半径

设置水平半径大小。默认值为1 。允许的范围是从 1 到 127 的整数。

飞机

设置要处理的平面。默认为15 ，这是所有可用的平面。

半径V

设置垂直半径大小。默认值为0 。允许的范围是从 0 到 127 的整数。如果它是 0，将从水平半径选项中选取值。

百分位数

设置中位数百分位数。默认值为0.5 。默认值0.5将始终选择中值，而0将选择最小值， 1将选择最大值。

* [命令](#Commands-118)

#### 39.159.1 命令

此过滤器支持与选项相同的[命令](#commands)。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.160 合并平面

合并来自多个视频流的颜色通道组件。

该过滤器最多接受 4 个输入流，并将选定的输入平面合并到输出视频中。

此过滤器接受以下选项：

映射

设置输入到输出平面映射。默认为0 。

映射被指定为位图。它应指定为 0xAa[Bb[Cc[Dd]]] 形式的十六进制数。 'Aa' 描述了输出流的第一个平面的映射。 'A' 设置要使用的输入流的编号（从 0 到 3），'a' 设置要使用的相应输入的平面编号（从 0 到 3）。其余映射类似，“Bb”描述了输出流第二平面的映射，“Cc”描述了输出流第三平面的映射，“Dd”描述了输出流第四平面的映射。

格式

设置输出像素格式。默认是yuva444p 。

地图0s

地图1s

map2s

地图3

设置输出第 N 平面的输入到输出流映射。默认为0 。

地图0p

地图1p

地图2P

地图3p

为输出第 N 个平面设置输入到输出平面映射。默认为0 。

* [例子](#Examples-134)

#### 39.160.1 示例

* 将三个相同宽度和高度的灰色视频流合并为单个视频流：
* [a0][a1][a2]mergeplanes=0x001020:yuv444p
* 将第一个 yuv444p 流和第二个灰色视频流合并到 yuva444p 视频流中：
* [a0][a1]mergeplanes=0x00010210:yuva444p
* 在 yuva444p 流中交换 Y 和 A 平面：
* 格式=yuva444p，mergeplanes=0x03010200:yuva444p
* 在 yuv420p 流中交换 U 和 V 平面：
* 格式=yuv420p，mergeplanes=0x000201:yuv420p
* 将 rgb24 剪辑投射到 yuv444p：
* 格式=rgb24，mergeplanes=0x000102:yuv444p

### 39.161 估计

使用块匹配算法估计和导出运动矢量。运动矢量存储在帧端数据中以供其他过滤器使用。

此过滤器接受以下选项：

方法

指定运动估计方法。接受以下值之一：

' esa '

详尽的搜索算法。

' ts '

三步搜索算法。

' tdls '

二维对数搜索算法。

' nts '

新的三步搜索算法。

' fs '

四步搜索算法。

' ds '

钻石搜索算法。

'六角'

基于六边形的搜索算法。

' epzs '

增强的预测区域搜索算法。

'嗯'

不均匀多六边形搜索算法。

默认值为“ esa ”。

mb\_size

宏块大小。默认16 。

搜索参数

搜索参数。默认7 。

### 39.162 中等均衡器

使用两个视频流应用中途图像均衡效果。

Midway Image Equalization 将一对图像调整为具有相同的直方图，同时尽可能保持它们的动态。例如，它可用于匹配一对立体相机的曝光。

此过滤器有两个输入和一个输出，它们必须具有相同的像素格式，但可以具有不同的大小。滤波器的输出首先是用两个输入的中间直方图调整的输入。

此过滤器接受以下选项：

飞机

设置要处理的平面。默认为15 ，这是所有可用的平面。

### 39.163 最小插值

使用运动插值将视频转换为指定的帧速率。

此过滤器接受以下选项：

每秒帧数

指定输出帧率。这可以是合理的，例如60000/1001 。如果fps低于源 fps，则会丢弃帧。默认60 。

mi\_mode

运动插补模式。接受以下值：

'傻瓜'

复制前一帧或下一帧以插入新帧。

'混合'

混合源帧。插值帧是前一帧和下一帧的平均值。

' mci '

运动补偿插值。选择此模式时，以下选项有效：

' mc\_mode '

运动补偿模式。接受以下值：

' obmc '

重叠块运动补偿。

' aobmc '

自适应重叠块运动补偿。根据相邻运动矢量的可靠性自适应地控制窗口加权系数以减少过度平滑。

默认模式是“ obmc ”。

'我模式'

运动估计模式。接受以下值：

'双向'

双向运动估计。在前向和后向两个方向上为每个源帧估计运动矢量。

'比拉特'

双边运动估计。直接为内插帧估计运动矢量。

默认模式是' bilat '。

'我'

用于运动估计的算法。接受以下值：

' esa '

详尽的搜索算法。

' ts '

三步搜索算法。

' tdls '

二维对数搜索算法。

' nts '

新的三步搜索算法。

' fs '

四步搜索算法。

' ds '

钻石搜索算法。

'六角'

基于六边形的搜索算法。

' epzs '

增强的预测区域搜索算法。

'嗯'

不均匀多六边形搜索算法。

默认算法是' epzs '。

' mb\_size '

宏块大小。默认16 。

'搜索参数'

运动估计搜索参数。默认32 。

' vsbmc '

启用可变大小块运动补偿。运动估计在对象边界处应用较小的块大小，以便使它们不那么模糊。默认值为0 （禁用）。

scd

场景变化检测方法。场景变化导致运动矢量处于随机方向。场景变化检测用重复的帧替换插值帧。其他模式可能不需要。接受以下值：

'无'

禁用场景变化检测。

'差异'

帧差异。比较相应的像素值，如果满足scd\_threshold则检测到场景变化。

默认方法是“ fdiff ”。

scd\_threshold

场景变化检测阈值。默认为10 ..

### 39.164混合

将多个视频输入流混合成一个视频流。

接受的选项的描述如下。

输入

输入的数量。如果未指定，则默认为 2。

权重

将每个输入视频流的权重指定为序列。每个权重由空格分隔。如果权重数小于帧数，最后指定的权重将用于所有剩余的未设置权重。

规模

指定比例，如果设置它，它将乘以每个权重乘以像素值的总和，以给出最终目标像素值。默认情况下，比例会自动缩放为权重之和。

飞机

设置要过滤的平面。默认为全部。允许的范围是从 0 到 15。

期间

指定如何确定流的结尾。

'最长'

最长输入的持续时间。 （默认）

'最短'

最短输入的持续时间。

'第一'

第一个输入的持续时间。

* [命令](#Commands-119)

#### 39.164.1 命令

该过滤器支持以下命令：

权重

规模

飞机

语法与同名选项相同。

### 39.165单色

使用自定义颜色过滤器将视频转换为灰色。

接受的选项的描述如下。

cb

设置色度蓝点。允许的范围是从 -1 到 1。默认值为 0。

铬

设置色度红点。允许的范围是从 -1 到 1。默认值为 0。

尺寸

设置滤色器尺寸。允许的范围是从 .1 到 10。默认值为 1。

高的

设置高光强度。允许的范围是从 0 到 1。默认值为 0。

* [命令](#Commands-120)

#### 39.165.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.166 吗啡

此过滤器允许应用主要形态学灰度变换、侵蚀和膨胀第二个输入流中设置的任意结构。

[侵蚀](#erosion)和[膨胀过滤器](#dilation)的简单实现和慢得多的性能不同，当速度很关键时，应该使用morpho过滤器。

接受的选项的描述如下，

模式

设置要应用的形态变换，可以是：

'侵蚀'

'扩张'

'打开'

'关闭'

'渐变'

'礼帽'

'黑帽'

默认是侵蚀。

飞机

设置要过滤的平面，默认情况下过滤除 alpha 之外的所有平面。

结构

设置将从第二个输入流处理的结构视频帧，可以是first或all 。默认为全部。

morpho过滤器还支持[帧同步](#framesync)选项。

* [命令](#Commands-121)

#### 39.166.1 命令

此过滤器支持与选项相同的[命令](#commands)。

### 39.167 mp 十进制

丢弃与前一帧差别不大的帧以降低帧速率。

该过滤器的主要用途是用于非常低比特率的编码（例如通过拨号调制解调器进行流式传输），但理论上它可以用于修复不正确的反向电视电影。

接受的选项的描述如下。

最大限度

设置可以丢弃的连续帧的最大数量（如果为正），或丢弃帧之间的最小间隔（如果为负）。如果该值为 0，则丢弃该帧，而不管之前顺序丢弃的帧数。

默认值为 0。

你好

罗

压裂

设置下降阈值。

hi和lo的值适用于 8x8 像素块，代表实际像素值差异，因此 64 的阈值对应于每个像素的 1 个差异单位，或者相同的像素在块上的不同分布。

hi的阈值，并且不超过frac块（1 表示整个图像）的差异超过lo的阈值，则帧是丢弃的候选者。

hi默认值为64\*12， lo默认值为64\*5， frac默认值为0.33。

### 39.168 毫秒

获取两个输入视频之间的 MSAD（绝对差的平均和）。

这个过滤器需要两个输入视频。

两个输入视频必须具有相同的分辨率和像素格式，此过滤器才能正常工作。它还假设两个输入具有相同数量的帧，将它们一一比较。

获得的每个组件、平均值、最小和最大 MSAD 通过日志系统打印。

过滤器将计算出的每个帧的 MSAD 存储在帧元数据中。

此过滤器还支持[帧同步](#framesync)选项。

在下面的示例中，将正在处理的输入文件main.mpg与参考文件ref.mpg进行比较。

ffmpeg -i main.mpg -i ref.mpg -lavfi msad -f null -

### 39.169 相乘

将第一视频流像素值与第二视频流像素值相乘。

过滤器接受以下选项：

规模

设置应用于第二个视频流的比例。默认为1 。允许的范围是从0到9 。

抵消

设置应用于第二个视频流的偏移量。默认为0.5 。允许的范围是从-1到1 。

飞机

从将要处理的输入视频流中指定平面。默认情况下处理所有平面。

* [命令](#Commands-122)

#### 39.169.1 命令

此过滤器支持与选项相同的[命令](#commands)。

### 39.170 否定

否定（反转）输入视频。

它接受以下选项：

成分

将组件设置为取反。

组件的可用值是：

'是'

'你'

' v '

'一个'

' r '

' g '

' b '

negate\_alpha

如果值为 1，它会否定 alpha 分量（如果存在）。默认值为 0。

* [命令](#Commands-123)

#### 39.170.1 命令

此过滤器支持与选项相同的[命令](#commands)。

### 39.171 均值

使用非局部均值算法对帧进行降噪。

通过寻找具有相似上下文的其他像素来调整每个像素。这种上下文相似性是通过比较它们周围大小为p x p的块来定义的。在像素周围r x r的区域中搜索补丁。

请注意，研究区域定义了补丁的中心，这意味着一些补丁将由该研究区域之外的像素组成。

过滤器接受以下选项。

秒

设置去噪强度。默认值为 1.0。必须在 [1.0, 30.0] 范围内。

p

设置补丁大小。默认为 7。必须是 [0, 99] 范围内的奇数。

个人电脑

p相同，但用于色度平面。

默认值为0 ，表示自动。

r

设置研究规模。默认为 15。必须是 [0, 99] 范围内的奇数。

遥控

r相同，但用于色度平面。

默认值为0 ，表示自动。

### 39.172 尼迪

使用神经网络边缘定向插值的去隔行视频。

此过滤器接受以下选项：

权重

强制选项，没有二进制文件过滤器不能工作。当前文件可以在这里找到：https://github.com/dubhater/vapoursynth-nnedi3/blob/master/src/nnedi3\_weights.bin

取消

设置去隔行的帧数，默认为all 。可以全部或交错。

场地

设置操作模式。

可以是以下之一：

' af '

使用帧标志，两个字段。

'一个'

使用帧标志，单场。

' t '

仅使用顶场。

' b '

仅使用底场。

' tf '

使用两个字段，顶部优先。

'男朋友'

使用两个字段，从底部开始。

飞机

设置要处理的平面，默认情况下过滤器处理所有帧。

尺寸

设置每个像素周围的局部邻域大小，供预测神经网络使用。

可以是以下之一：

' s8x6 '

' s16x6 '

' s32x6 '

' s48x6 '

' s8x4 '

' s16x4 '

' s32x4 '

nns

设置预测神经网络中的神经元数量。可以是以下之一：

' n16 '

' n32 '

' n64 '

' n128 '

' n256 '

质量

控制混合在一起以计算最终输出值的不同神经网络预测的数量。可以快，默认或慢。

类型

设置要在预测器中使用的权重集。可以是以下之一：

'一个，腹肌'

训练权重以最小化绝对误差

'' s, mse '

训练权重以最小化平方误差

密码

控制是否使用预筛选器神经网络来决定哪些像素应由预测器神经网络处理，哪些可以通过简单的三次插值处理。预筛选器经过训练以了解三次插值是否足以用于像素，或者它是否应该由预测器 nn 进行预测。预筛选器 nn 的计算复杂度远低于预测器 nn 的计算复杂度。由于大多数像素都可以通过三次插值来处理，因此使用预筛选器通常会导致更快的处理速度。预筛选器非常准确，所以使用它和不使用它之间的区别几乎总是不明显的。

可以是以下之一：

'无'

'原创'

'新'

' new2 '

' new3 '

默认是新的。

* [命令](#Commands-124)

#### 39.172.1 命令

此过滤器支持与选项相同的[命令](#commands)，但不包括权重选项。

### 39.173 无格式

强制 libavfilter 不使用任何指定的像素格式作为下一个过滤器的输入。

它接受以下参数：

pix\_fmts

以“|”分隔的像素格式名称列表，例如 pix\_fmts=yuv420p|monow|rgb24"。

* [例子](#Examples-135)

#### 39.173.1 示例

* 强制 libavfilter 使用不同于yuv420p 的格式作为 vflip 过滤器的输入：
* noformat=pix\_fmts=yuv420p,vflip
* 将输入视频转换为列表中未包含的任何格式：
* noformat=yuv420p|yuv444p|yuv410p

### 39.174噪音

在视频输入帧上添加噪声。

过滤器接受以下选项：

全种子

c0\_种子

c1\_种子

c2\_seed

c3\_seed

all\_seed的情况下，为特定像素分量或所有像素分量设置噪声种子。默认值为123457 。

all\_strength, 全部

c0\_强度，c0s

c1\_强度，c1s

c2\_strength, c2s

c3\_strength, c3s

为特定像素分量或所有像素分量设置噪声强度，以防all\_strength 。默认值为0 。允许的范围是 [0, 100]。

all\_flags，allf

c0\_flags，c0f

c1\_flags，c1f

c2\_flags，c2f

c3\_flags，c3f

all\_flags ，则设置像素组件标志或为所有组件设置标志。组件标志的可用值是：

'一个'

平均时间噪声（更平滑）

' p '

将随机噪声与（半）规则模式混合

' t '

时间噪声（帧之间的噪声模式变化）

'你'

均匀噪声（否则为高斯噪声）

* [例子](#Examples-136)

#### 39.174.1 示例

向输入视频添加时间和均匀的噪声：

噪声=alls=20:allf=t+u

### 39.175 归一化

归一化 RGB 视频（也称为直方图拉伸、对比度拉伸）。请参阅：https://en.wikipedia.org/wiki/Normalization\_(image\_processing)

对于每一帧的每个通道，过滤器计算输入范围并将其线性映射到用户指定的输出范围。输出范围默认为从纯黑到纯白的全动态范围。

时间平滑可用于输入范围，以减少当小的暗或亮物体进入或离开场景时引起的闪烁（亮度的快速变化）。这类似于摄像机上的自动曝光（自动增益控制），并且像摄像机一样，它可能会导致视频曝光过度或曝光不足。

R、G、B 通道可以独立归一化，这可能会导致一些颜色偏移，或者作为单个通道链接在一起，从而防止颜色偏移。链接归一化保留色调。独立归一化没有，所以它可以用来去除一些偏色。独立和链接归一化可以以任何比例组合。

规范化过滤器接受以下选项：

黑点

白点

定义输出范围的颜色。最小输入值映射到blackpt 。最大输入值映射到whitept 。默认值分别为黑色和白色。为blackpt指定 white并为whitept指定 black将提供颜色反转的标准化视频。灰色阴影可用于减小动态范围（对比度）。在此处指定饱和色可以创建一些有趣的效果。

平滑

用于时间平滑的先前帧数。使用当前帧的滚动平均值和平滑的先前帧来平滑每个通道的输入范围。默认值为 0（无时间平滑）。

独立

控制独立（颜色偏移）通道归一化与链接（颜色保留）归一化的比率。 0.0是完全链接，1.0是完全独立。默认为 1.0（完全独立）。

力量

过滤器的整体强度。 1.0 是满强度。 0.0 是一个相当昂贵的空操作。默认为 1.0（最大强度）。

* [命令](#Commands-125)
* [例子](#Examples-137)

#### 39.175.1 命令

此过滤器支持与选项相同的[命令，但不包括](#commands)平滑选项。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

#### 39.175.2 示例

拉伸视频对比度以使用全动态范围，没有时间平滑；可能会闪烁，具体取决于源内容：

归一化=blackpt=black:whitept=white:smoothing=0

同上，但有 50 帧的时间平滑；应减少闪烁，具体取决于源内容：

归一化=blackpt=black:whitept=white:smoothing=50

如上所述，但使用色调保持链接通道归一化：

归一化=blackpt=black:whitept=white:smoothing=50:independence=0

同上，但强度减半：

normalize=blackpt=black:whitept=white:smoothing=50:independence=0:strength=0.5

将最暗的输入颜色映射为红色，将最亮的输入颜色映射为青色：

归一化=blackpt=red:whitept=cyan

### 39.176 无效

将视频源原样传递到输出。

### 39.177 光学字符识别

光学字符识别

此过滤器使用 Tesseract 进行光学字符识别。要启用此过滤器的编译，您需要使用--enable-libtesseract配置 FFmpeg 。

它接受以下选项：

数据路径

将数据路径设置为 tesseract 数据。默认是使用安装时设置的任何内容。

语言

设置语言，默认为“eng”。

白名单

设置字符白名单。

黑名单

设置字符黑名单。

过滤器将识别的文本导出为帧元数据lavfi.ocr.text 。过滤器将已识别单词的置信度导出为帧元数据lavfi.ocr.confidence 。

### 39.178 超频

使用 libopencv 应用视频转换。

要启用此过滤器，请安装 libopencv 库和标头并使用--enable-libopencv配置 FFmpeg 。

它接受以下参数：

过滤器名称

要应用的 libopencv 过滤器的名称。

过滤参数

传递给 libopencv 过滤器的参数。如果未指定，则采用默认值。

更准确的信息请参考 libopencv 官方文档： <http://docs.opencv.org/master/modules/imgproc/doc/filtering.html>

支持多个 libopencv 过滤器；请参阅以下小节。

* [扩张](#dilate-1)
* [侵蚀](#erode)
* [光滑的](#smooth)

#### 39.178.1 膨胀

使用特定的结构元素对图像进行膨胀。它对应于 libopencv 函数cvDilate 。

它接受参数： struct\_el | nb\_iterations 。

struct\_el表示结构元素，语法为： cols x rows + anchor\_x x anchor\_y / shape

cols和rows表示结构元素的列数和行数， anchor\_x和anchor\_y表示锚点， shape表示结构元素的形状。 shape必须是“rect”、“cross”、“ellipse”或“custom”。

shape的值为“custom”，则它后面必须跟一个“= filename ”形式的字符串。假定名称为filename 的文件表示二进制图像，每个可打印字符对应一个亮像素。当使用自定义形状时，列和行将被忽略，取而代之的是读取文件的数字或列和行。

struct\_el的默认值为“3x3+0x0/rect”。

nb\_iterations指定变换应用于图像的次数，默认为 1。

一些例子：

# 使用默认值

ocv=膨胀

# 使用带有 5x5 交叉的结构元素进行扩张，迭代两次

ocv=filter\_name=dilate:filter\_params=5x5+2x2/cross|2

# 从文件 diamond.shape 中读取形状，迭代两次。

# 文件 diamond.shape 可能包含这样的字符模式

# \*

#\*\*\*

# \*\*\*\*\*

#\*\*\*

# \*

# 忽略指定的列和行

# 但锚点坐标不是

ocv=膨胀:0x0+2x2/自定义=菱形.形状|2

#### 39.178.2 侵蚀

使用特定的结构元素侵蚀图像。它对应于 libopencv 函数cvErode 。

它接受参数： struct\_el : nb\_iterations ，具有与[扩张](#dilate)过滤器相同的语法和语义。

#### 39.178.3 光滑

平滑输入视频。

过滤器采用以下参数： type |参数 1 |参数2 |参数 3 |参数 4 。

type是要应用的平滑过滤器的类型，并且必须是以下值之一：“blur”、“blur\_no\_scale”、“median”、“gaussian”或“bilateral”。默认值为“高斯”。

param1 、 param2 、 param3和param4的含义取决于平滑类型。 param1和param2接受整数正值或 0。param3和param4接受浮点值。

param1的默认值为3。其他参数的默认值为 0。

cvSmooth的参数。

### 39.179 示波器

二维视频示波器。

可用于测量空间脉冲、阶跃响应、色度延迟等。

它接受以下参数：

X

设置示波器中心 x 位置。

是

设置示波器中心 y 位置。

秒

设置范围大小，相对于框架对角线。

吨

设置示波器倾斜/旋转。

o

设置跟踪不透明度。

发送

设置轨迹中心 x 位置。

泰

设置跟踪中心 y 位置。

tw

设置轨迹宽度，相对于框架的宽度。

日

设置轨迹高度，相对于框架的高度。

C

设置要跟踪的组件。默认情况下，它跟踪前三个组件。

G

绘制跟踪网格。默认情况下启用。

英石

绘制一些统计数据。默认情况下启用。

SC

绘制范围。默认情况下启用。

* [命令](#Commands-126)
* [例子](#Examples-138)

#### 39.179.1 命令

此过滤器支持与选项相同的[命令](#commands)。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

#### 39.179.2 示例

* 检查完整的第一行视频帧。
* 示波器=x=0.5:y=0:s=1
* 检查完整的最后一行视频帧。
* 示波器=x=0.5:y=1:s=1
* 检查高度为 1080 的视频帧的完整第 5 行。
* 示波器=x=0.5:y=5/1080:s=1
* 检查视频帧的最后一列。
* 示波器=x=1:y=0.5:s=1:t=1

### 39.180 叠加

将一个视频叠加在另一个视频之上。

它需要两个输入并有一个输出。第一个输入是覆盖第二个输入的“主”视频。

它接受以下参数：

接受的选项的描述如下。

X

是

设置主视频上叠加视频的 x 和 y 坐标的表达式。两个表达式的默认值为“0”。在表达式无效的情况下，它被设置为一个巨大的值（意味着叠加层将不会显示在输出可见区域内）。

eof\_action

请参见[帧同步](#framesync)。

评估

计算x和y的表达式时设置。

它接受以下值：

'初始化'

仅在过滤器初始化期间或处理命令时对表达式求值一次

'框架'

评估每个传入帧的表达式

默认值为“框架”。

最短的

请参见[帧同步](#framesync)。

格式

设置输出视频的格式。

它接受以下值：

' yuv420 '

强制YUV420输出

' yuv420p10 '

强制 YUV420p10 输出

' yuv422 '

强制YUV422输出

' yuv422p10 '

强制 YUV422p10 输出

' yuv444 '

强制YUV444输出

' RGB '

强制压缩 RGB 输出

'英镑'

强制平面 RGB 输出

'汽车'

自动选择格式

默认值为“ yuv420 ”。

重复上次

请参见[帧同步](#framesync)。

α

设置叠加视频的 alpha 格式，可以是straight或premultiplied 。默认是直的。

x和y表达式可以包含以下参数。

main\_w, W

main\_h, H

主要输入宽度和高度。

overlay\_w, w

overlay\_h, h

覆盖输入宽度和高度。

X

是

x和y的计算值。它们针对每个新帧进行评估。

hsub

vsub

输出格式的水平和垂直色度子样本值。例如，对于像素格式“yuv422p”， hsub为 2， vsub为 1。

n

输入帧数，从0开始

位置

输入帧在文件中的位置，如果未知则为 NAN

吨

时间戳，以秒为单位。如果输入时间戳未知，则为 NAN。

此过滤器还支持[帧同步](#framesync)选项。

请注意， n 、 pos 、 t变量仅在对每帧进行评估时可用，并且在eval设置为“ init ”时将评估为 NAN。

请注意，帧是按时间戳顺序从每个输入视频中获取的，因此，如果它们的初始时间戳不同，最好通过 setpts=PTS-STARTPTS 过滤器传递两个输入，让它们以相同的零时间戳开始，就像电影过滤器的例子一样。

您可以将更多叠加层链接在一起，但您应该测试这种方法的效率。

* [命令](#Commands-127)
* [例子](#Examples-139)

#### 39.180.1 命令

该过滤器支持以下命令：

X

是

修改覆盖输入的 x 和 y。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

#### 39.180.2 示例

* 在距主视频右下角 10 像素处绘制叠加层：
* overlay=main\_w-overlay\_w-10:main\_h-overlay\_h-10

使用命名选项，上面的示例变为：

覆盖=x=main\_w-overlay\_w-10:y=main\_h-overlay\_h-10

* -filter\_complex选项的ffmpeg工具，在输入的左下角插入一个透明的 PNG 标志：
* ffmpeg -i input -i logo -filter\_complex 'overlay=10:main\_h-overlay\_h-10' 输出
* ffmpeg工具插入 2 个不同的透明 PNG 标志（第二个标志在右下角） ：
* ffmpeg -i 输入 -i logo1 -i logo2 -filter\_complex 'overlay=x=10:y=Hh-10,overlay=x=Ww-10:y=Hh-10' 输出
* 在主视频之上添加透明色层； WxH必须指定覆盖过滤器的主要输入的大小：
* color=color=red@.3:size=WxH [over]; [in][over] 覆盖 [out]
* ffplay工具并排播放原始视频和过滤后的版本（此处使用 deshake 过滤器） ：
* ffplay input.avi -vf 'split[a][b]; [a]pad=iw\*2:ih[src]; [b]去抖动[过滤]； [src][filt]overlay=w'

上面的命令与以下命令相同：

ffplay input.avi -vf 'split[b], pad=iw\*2[src], [b]deshake, [src]overlay=w'

* 从时间 2 开始，从屏幕的左上角到右上角制作一个滑动叠加层：
* overlay=x='if(gte(t,2), -w+(t-2)\*20, NAN)':y=0
* 通过将两个输入视频并排放置来组合输出：
* ffmpeg -i left.avi -i right.avi -filter\_complex "
* nullsrc=size=200x100 [背景]；
* [0:v] setpts=PTS-STARTPTS, scale=100x100 [left];
* [1:v] setpts=PTS-STARTPTS, scale=100x100 [右];
* [背景][左] overlay=shortest=1 [背景+左];
* [背景+左][右] overlay=shortest=1:x=100 [左+右]
* “
* 通过将 delogo 过滤器应用于某个部分来屏蔽视频的 10-20 秒
* ffmpeg -i test.avi -codec:v:0 wmv2 -ar 11025 -b:v 9000k
* -vf '[in]split[split\_main][split\_delogo];[split\_delogo]trim=start=360:end=371,delogo=0:0:640:480[delogoed];[split\_main][delogoed]overlay=eof\_action=通过[出]'
* 蒙版.avi
* 串联多个叠加层：
* nullsrc=s=200x200 [bg];
* testsrc=s=100x100, split=4 [in0][in1][in2][in3];
* [in0] lutrgb=r=0, [bg] overlay=0:0 [mid0];
* [in1] lutrgb=g=0, [mid0] overlay=100:0 [mid1];
* [in2] lutrgb=b=0, [mid1] overlay=0:100 [mid2];
* [in3] 空，[mid2] 覆盖=100:100 [out0]

### 39.181 overlay\_cuda

将一个视频叠加在另一个视频之上。

[覆盖过滤器](#overlay)的 CUDA 变体。它只接受 CUDA 框架。底层输入像素格式必须匹配。

它需要两个输入并有一个输出。第一个输入是覆盖第二个输入的“主”视频。

它接受以下参数：

X

是

设置主视频上叠加视频的 x 和 y 坐标表达式。

它们可以包含以下参数：

main\_w, W

main\_h, H

主要输入宽度和高度。

overlay\_w, w

overlay\_h, h

覆盖输入宽度和高度。

X

是

x和y的计算值。它们针对每个新帧进行评估。

n

主输入框的序号索引，从0开始。

位置

主输入帧在文件中的字节偏移位置，如果未知则为 NAN。

吨

主输入帧的时间戳，以秒表示，如果未知则为 NAN。

两个表达式的默认值为“0”。

评估

计算x和y的表达式时设置。

它接受以下值：

在里面

在过滤器初始化期间或处理命令时对表达式求值一次。

框架

评估每个传入帧的表达式

默认值为帧。

eof\_action

请参见[帧同步](#framesync)。

最短的

请参见[帧同步](#framesync)。

重复上次

请参见[帧同步](#framesync)。

此过滤器还支持[帧同步](#framesync)选项。

### 39.182 低噪声

应用 Overcomplete Wavelet 降噪器。

过滤器接受以下选项：

深度

设置深度。

较大的深度值会更多地对较低频率的分量进行去噪，但会减慢过滤速度。

必须是 8-16 范围内的整数，默认为8 。

亮度强度，ls

设置亮度强度。

必须是 0-1000 范围内的双精度值，默认为1.0 。

色度强度，CS

设置色度强度。

必须是 0-1000 范围内的双精度值，默认为1.0 。

### 39.183垫

向输入图像添加填充，并将原始输入放置在提供的x和y坐标处。

它接受以下参数：

宽度，w

身高，小时

为添加了填充的输出图像的大小指定一个表达式。如果宽度或高度的值为0，则相应的输入大小用于输出。

宽度表达式可以引用高度表达式设置的值，反之亦然。

宽度和高度的默认值为0。

X

是

相对于输出图像的上/左边框，指定将输入图像放置在填充区域内的偏移量。

x表达式可以引用y表达式设置的值，反之亦然。

x和y的默认值为0。

如果x或y的计算结果为负数，它们将被更改，因此输入图像位于填充区域的中心。

颜色

指定填充区域的颜色。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。

颜色的默认值为“黑色”。

评估

指定何时评估width 、 height 、 x和y表达式。

它接受以下值：

'初始化'

在过滤器初始化期间或处理命令时仅对表达式求值一次。

'框架'

评估每个传入帧的表达式。

默认值为“ init ”。

方面

填充到纵横比而不是分辨率。

width 、 height 、 x和y选项的值是包含以下常量的表达式：

in\_w

在\_h

输入视频的宽度和高度。

我

我

in\_w和in\_h相同。

out\_w

out\_h

和height表达式指定的输出宽度和高度（填充区域的大小） 。

噢

哦

out\_w和out\_h相同。

X

是

x 和 y 表达式指定的x和y偏移量，如果尚未指定，则为 NAN。

A

iw / ih相同

特区

输入样本纵横比

达尔

输入显示宽高比，同( iw / ih ) \* sar

hsub

vsub

水平和垂直色度子样本值。例如，对于像素格式“yuv422p”， hsub为 2， vsub为 1。

* [例子](#Examples-140)

#### 39.183.1 示例

* 向输入视频添加颜色为“紫色”的填充。输出视频大小为 640x480，输入视频的左上角位于第 0 列第 40 行
* 垫=640:480:0:40:紫色

上面的示例等效于以下命令：

pad=width=640:height=480:x=0:y=40:color=violet

* 填充输入以获得尺寸增加 3/2 的输出，并将输入视频放在填充区域的中心：
* pad="3/2\*iw:3/2\*ih:(ow-iw)/2:(oh-ih)/2"
* 填充输入以获得大小等于输入宽度和高度之间最大值的平方输出，并将输入视频放在填充区域的中心：
* pad="max(iw\,ih):ow:(ow-iw)/2:(oh-ih)/2"
* 填充输入以获得 16:9 的最终 w/h 比率：
* pad="ih\*16/9:ih:(ow-iw)/2:(oh-ih)/2"
* 在变形视频的情况下，为了正确设置输出显示宽高比，需要在表达式中使用sar ，根据以下关系：
* (ih \* X / ih) \* sar = output\_dar
* X = output\_dar / sar

因此前面的例子需要修改为：

pad="ih\*16/9/sar:ih:(ow-iw)/2:(oh-ih)/2"

* 将输出大小加倍并将输入视频放在输出填充区域的右下角：
* pad="2\*iw:2\*ih:ow-iw:oh-ih"

### 39.184 调色板生成

为整个视频流生成一个调色板。

它接受以下选项：

最大颜色

设置调色板中要量化的最大颜色数。注意：调色板仍将包含 256 种颜色；未使用的调色板条目将为黑色。

保留\_透明

创建最多 255 种颜色的调色板，并保留最后一种颜色作为透明度。保留透明颜色对于 GIF 优化很有用。如果未设置，调色板中的最大颜色将为 256。您可能希望为独立图像禁用此选项。默认设置。

透明度颜色

设置将用作透明背景的颜色。

统计模式

设置统计模式。

它接受以下值：

'满'

计算全帧直方图。

'差异'

仅计算与前一帧不同的部分的直方图。如果背景是静态的，这可能与更加重视输入的移动部分有关。

'单身'

为每一帧计算新的直方图。

默认值为full 。

该过滤器还导出帧元数据lavfi.color\_quant\_ratio ( nb\_color\_in / nb\_color\_out )，您可以使用它来评估调色板的颜色量化程度。此信息在信息日志记录级别也可见。

* [例子](#Examples-141)

#### 39.184.1 示例

* ffmpeg生成给定视频的代表性调色板：
* ffmpeg -i input.mkv -vf palettegen palette.png

### 39.185 调色板使用

使用调色板对输入视频流进行下采样。

过滤器接受两个输入：一个视频流和一个调色板。调色板必须是 256 像素的图像。

它接受以下选项：

抖动

选择抖动模式。可用的算法是：

'拜耳'

有序 8x8 拜耳抖动（确定性）

'赫伯特'

Paul Heckbert 在 1982 年定义的抖动（简单误差扩散）。注意：这种抖动有时被认为是“错误的”，并作为参考包含在内。

'弗洛伊德斯坦伯格'

Floyd 和 Steingberg 抖动（误差扩散）

'山脉2 '

Frankie Sierra 抖动 v2（误差扩散）

' sierra2\_4a '

Frankie Sierra 抖动 v2“精简版”（错误扩散）

'山脉 3 '

Frankie Sierra 抖动 v3（误差扩散）

'伯克斯'

Burkes 抖动（误差扩散）

'阿特金森'

Apple Computer 的 Bill Atkinson 的阿特金森抖动（错误扩散）

默认为sierra2\_4a 。

拜耳规模

拜耳抖动时，此选项定义图案的比例（交叉影线图案的可见程度）。低值意味着更明显的图案，更少的条纹，更高的值意味着更少可见的图案，但代价是更多的条纹。

该选项必须是 [0,5] 范围内的整数值。默认值为2 。

差异模式

如果设置，定义要处理的区域

'矩形'

只有变化的矩形会被重新处理。这类似于 GIF 裁剪/偏移压缩机制。如果只有一部分图像发生变化，此选项对提高速度很有用，并且有一些用例，例如将误差扩散抖动的范围限制在移动场景的边界矩形内（如果场景没有变化，它会导致更确定的输出） '变化很大，因此移动噪音更少，GIF 压缩效果更好）。

默认为无。

新的

为每个输出帧采用新的调色板。

alpha\_threshold

设置透明度的 alpha 阈值。高于此阈值的 Alpha 值将被视为完全不透明，低于此阈值的值将被视为完全透明。

该选项必须是 [0,255] 范围内的整数值。默认值为128 。

* [例子](#Examples-142)

#### 39.185.1 示例

* 使用调色板（例如使用[palettegen生成）使用](#palettegen)ffmpeg对 GIF 进行编码：
* ffmpeg -i input.mkv -i palette.png -lavfi paletteuse output.gif

### 39.186透视

视频的正确视角不是垂直于屏幕录制的。

接受的参数的描述如下。

x0

y0

x1

y1

x2

y2

x3

y3

设置左上角、右上角、左下角和右下角的坐标表达式。默认值为0:0:W:0:0:H:W:H，透视将保持不变。如果sense选项设置为source ，则指定的点将被发送到目的地的角落。如果sense选项设置为destination ，那么源的角将被发送到指定的坐标。

表达式可以使用以下变量：

W

H

视频帧的宽度和高度。

在

输入帧数。

在

输出帧数。

插值

为透视校正设置插值。

它接受以下值：

'线性'

'立方体'

默认值为“线性”。

感觉

设置坐标选项的解释。

它接受以下值：

' 0，来源'

将给定坐标指定的源中的点发送到目的地的角。

' 1、目的地'

将源的角发送到给定坐标指定的目标中的点。

默认值为“来源”。

评估

x0,y0,...x3,y3 的表达式时设置。

它接受以下值：

'初始化'

仅在过滤器初始化期间或处理命令时对表达式求值一次

'框架'

评估每个传入帧的表达式

默认值为“ init ”。

### 39.187相

将隔行扫描视频延迟一个场时间，以便改变场序。

预期用途是修复以与电影到视频传输相反的场序捕获的 PAL 电影。

接受的参数的描述如下。

模式

设置相位模式。

它接受以下值：

' t '

捕获字段顺序从上到下，传输从下到上。过滤器将延迟底部字段。

' b '

捕获字段顺序从下到上，传输从上到下。过滤器将延迟顶场。

' p '

使用相同的现场顺序捕获和传输。此模式仅供其他选项的文档参考，但如果您实际选择它，过滤器将忠实地不做任何事情。

'一个'

捕获字段顺序由字段标志自动确定，传输相反。过滤器使用字段标志逐帧选择“ t ”和“ b ”模式。如果没有可用的字段信息，那么这就像“ u ”一样工作。

'你'

捕获未知或变化，转移相反。过滤器通过分析图像并选择在场之间产生最佳匹配的备选方案，逐帧地在“ t ”和“ b ”之间进行选择。

' T '

优先捕获，转移未知或变化。过滤器使用图像分析在“ t ”和“ p ”之间进行选择。

' B '

从底部开始捕获，转移未知或变化。过滤器使用图像分析在“ b ”和“ p ”之间进行选择。

'一个'

捕获由字段标志决定，传输未知或变化。过滤器使用字段标志和图像分析在“ t ”、“ b ”和“ p ”之间进行选择。如果没有字段信息可用，那么这就像“ U ”一样工作。这是默认模式。

'你'

捕获和传输未知或变化。过滤器仅使用图像分析在“ t ”、“ b ”和“ p ”之间进行选择。

* [命令](#Commands-128)

#### 39.187.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.188 光敏性

减少视频中的各种闪光，从而帮助患有癫痫症的用户。

它接受以下选项：

框架，f

设置过滤时使用的帧数。默认值为 30。

阈值，吨

设置检测阈值因子。默认值为 1。越低越严格。

跳过

设置采样帧时要跳过的像素数。默认值为 1。允许的范围为 1 到 1024。

旁路

保持框架不变。默认是禁用的。

### 39.189 pixdesctest

像素格式描述符测试过滤器，主要用于内部测试。输出视频应等于输入视频。

例如：

format=monow, pixdesctest

可用于测试单白像素格式描述符定义。

### 39.190 像素化

将像素化应用于视频流。

过滤器接受以下选项：

宽度，w

身高，小时

设置将用于像素化的块尺寸。默认值为16 。

模式，米

设置使用的像素化模式。

可能的值是：

'平均'

'分钟'

'最大'

默认值为avg 。

飞机，p

设置要过滤的平面。默认是过滤所有平面。

* [命令](#Commands-129)

#### 39.190.1 命令

此过滤器支持所有选项作为[命令](#commands)。

### 39.191像素范围

显示颜色通道的样本值。主要用于检查颜色和级别。支持的最低分辨率为 640x480。

过滤器接受以下选项：

X

设置范围 X 位置，X 轴上的相对偏移。

是

设置范围 Y 位置，Y 轴上的相对偏移量。

w

设置范围宽度。

H

设置示波器高度。

o

设置窗口不透明度。该窗口还包含有关像素区域的统计信息。

wx

设置窗口 X 位置，X 轴上的相对偏移量。

怀

设置窗口 Y 位置，Y 轴上的相对偏移量。

* [命令](#Commands-130)

#### 39.191.1 命令

此过滤器支持与选项相同的[命令](#commands)。

### 39.192 页

使用 libpostproc 启用指定的后处理子过滤器链。该库应该使用 GPL 构建 ( --enable-gpl ) 自动选择。子过滤器必须用“/”分隔，并且可以通过在前面加上“-”来禁用。每个子过滤器和一些选项都有一个短名称和一个长名称，可以互换使用，即 dr/dering 是相同的。

过滤器接受以下选项：

子过滤器

设置后处理子过滤器字符串。

所有子过滤器共享共同的选项来确定它们的范围：

一个/汽车

遵守此子过滤器的质量命令。

c/铬

也进行色度过滤（默认）。

y/nochrom

仅进行亮度过滤（无色度）。

n/诺鲁马

仅进行色度过滤（无亮度）。

这些选项可以附加在子过滤器名称之后，以“|”分隔。

可用的子过滤器是：

hb/hdeblock[|差异[|平坦度]]

水平去块滤波器

不同之处

差异因子，其中较高的值意味着更多的解块（默认值： 32 ）。

平坦度

平坦度阈值，其中较低的值意味着更多的去块效应（默认值： 39 ）。

vb/vdeblock[|差异[|平坦度]]

垂直去块滤波器

不同之处

差异因子，其中较高的值意味着更多的解块（默认值： 32 ）。

平坦度

平坦度阈值，其中较低的值意味着更多的去块效应（默认值： 39 ）。

ha/hadeblock[|差异[|平坦度]]

精确的水平去块滤波器

不同之处

差异因子，其中较高的值意味着更多的解块（默认值： 32 ）。

平坦度

平坦度阈值，其中较低的值意味着更多的去块效应（默认值： 39 ）。

va/vadeblock[|差异[|平坦度]]

精确的垂直去块滤波器

不同之处

差异因子，其中较高的值意味着更多的解块（默认值： 32 ）。

平坦度

平坦度阈值，其中较低的值意味着更多的去块效应（默认值： 39 ）。

水平和垂直去块滤波器共享差异值和平坦度值，因此您不能设置不同的水平和垂直阈值。

h1/x1h去块

实验水平去块滤波器

v1/x1v去块

实验垂直去块滤波器

博士/订单

去环滤波器

tn/tmpnoise[|threshold1[|threshold2[|threshold3]]], 时间降噪器

阈值1

更大 -> 更强的过滤

阈值2

更大 -> 更强的过滤

阈值3

更大 -> 更强的过滤

al/autolevels[:f/fullyrange]，自动亮度/对比度校正

f/全范围

将亮度拉伸到0-255 。

磅/林混合度

(1 2 1)过滤器过滤所有行来对给定的块去隔行。

锂/利尼波尔登

线性插值反交错滤波器，通过每两行线性插值来反交错给定的块。

ci/立方体

三次插值去隔行滤波器通过每隔一行进行三次插值来对给定的块去隔行。

md/中位数

中值去隔行滤波器，通过对每隔一行应用一个中值滤波器来对给定的块进行去隔行。

fd/ffmpegdeint

(-1 4 2 4 -1)过滤器过滤每隔一行来对给定的块去隔行。

l5/低通5

(-1 2 6 2 -1)滤波器过滤所有行来对给定的块去隔行。

fq/forceQuant[|量化器]

使用您指定的常量量化器覆盖输入中的量化器表。

量化器

要使用的量化器

默认/默认

默认 pp 过滤器组合 ( hb|a,vb|a,dr|a )

发/快

快速 pp 过滤器组合 ( h1|a,v1|a,dr|a )

交流电

高品质 pp 过滤器组合 ( ha|a|128|7,va|a,dr|a )

* [例子](#Examples-143)

#### 39.192.1 示例

* 应用水平和垂直去块、去色斑和自动亮度/对比度：
* pp=hb/vb/dr/al
* 应用没有亮度/对比度校正的默认过滤器：
* pp=de/-al
* 应用默认过滤器和时间降噪器：
* pp=默认值/tmpnoise|1|2|3
* 仅对亮度应用去块效应，并根据可用的 CPU 时间自动打开或关闭垂直去块效应：
* pp=hb|y/vb|a

### 39.193 页 7

应用后处理过滤器 7。它是[spp](#spp)过滤器的变体，类似于 spp = 6 和 7 点 DCT，其中在 IDCT 之后仅使用中心样本。

过滤器接受以下选项：

qp

强制一个恒定的量化参数。它接受 0 到 63 范围内的整数。如果未设置，过滤器将使用视频流中的 QP（如果可用）。

模式

设置阈值模式。可用模式有：

'硬'

设置硬阈值。

'软'

设置软阈值（更好的去振铃效果，但可能更模糊）。

'中号'

设置中等阈值（好的结果，默认值）。

### 39.194 预乘

使用第二个流的第一个平面作为 alpha，将 alpha 预乘效果应用于输入视频流。

两个流必须具有相同的尺寸和相同的像素格式。

过滤器接受以下选项：

飞机

设置要处理的平面，未处理的平面将被复制。默认值 0xf，将处理所有平面。

到位

不需要第二个输入进行处理，而是使用输入流中的 alpha 平面。

### 39.195 普雷维特

将 prewitt 运算符应用于输入视频流。

过滤器接受以下选项：

飞机

设置要处理的平面，未处理的平面将被复制。默认值 0xf，将处理所有平面。

规模

设置将与过滤结果相乘的值。

三角洲

设置将添加到过滤结果的值。

* [命令](#Commands-131)

#### 39.195.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.196伪彩

使用伪彩色更改视频中的帧颜色。

此过滤器接受以下选项：

c0

设置像素第一分量表达式

c1

设置像素第二分量表达式

c2

设置像素第三分量表达式

c3

设置像素第四分量表达式，对应alpha分量

指数，我

设置组件以用作更改颜色的基础

预设，p

选择一个内置 LUT。默认设置为无。

可用的 LUT：

'岩浆'

'地狱'

'等离子'

'绿色'

'涡轮'

'公民'

'范围1 '

'范围2 '

'阴影'

'亮点'

'太阳能'

'名义'

'首选'

'总计'

'光谱'

不透明度

设置输出颜色的不透明度。允许的范围是从 0 到 1。默认值设置为 1。

每个表达式选项指定用于计算相应像素分量值的查找表的表达式。

表达式可以包含以下常量和函数：

w

H

输入的宽度和高度。

值

像素分量的输入值。

ymin, umin, vmin, amin

允许的最小组件值。

ymax, umax, vmax, amax

最大允许组件值。

所有表达式默认为“val”。

* [命令](#Commands-132)
* [例子](#Examples-144)

#### 39.196.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

#### 39.196.2 示例

* 将过高的亮度值更改为渐变：
* pseudocolor="'if(between(val,ymax,amax),lerp(ymin,ymax,(val-ymax)/(amax-ymax)),-1):if(between(val,ymax,amax),lerp (umax,umin,(val-ymax)/(amax-ymax)),-1):if(between(val,ymax,amax),lerp(vmin,vmax,(val-ymax)/(amax-ymax) ),-1):-1'"

### 39.197 伪信噪比

获取两个输入视频之间的平均、最大和最小 PSNR（峰值信噪比）。

该过滤器接收两个输入视频，第一个输入被认为是“主要”来源，并不变地传递到输出。第二个输入用作计算 PSNR 的“参考”视频。

两个视频输入必须具有相同的分辨率和像素格式，此过滤器才能正常工作。它还假设两个输入具有相同数量的帧，将它们一一比较。

通过日志系统打印获得的平均 PSNR。

过滤器存储每帧的累积 MSE（均方误差），并在处理结束时对所有帧进行平均，并应用以下公式获得 PSNR：

PSNR = 10\*log10(MAX^2/MSE)

其中 MAX 是图像每个分量的最大值的平均值。

接受的参数的描述如下。

统计文件，f

如果指定，过滤器将使用命名文件来保存每个单独帧的 PSNR。当文件名等于“-”时，数据被发送到标准输出。

统计版本

指定要使用的统计文件格式版本。每种格式的详细信息如下。默认值为 1。

stats\_add\_max

确定最大值是否输出到统计日志。默认值为 0。需要 stats\_version >= 2。如果已设置且 stats\_version < 2，过滤器将返回错误。

此过滤器还支持[帧同步](#framesync)选项。

stats\_file ，则打印的文件包含一系列键/值对，形式为键：每个比较的帧的值。

如果指定的stats\_version大于 1，则标题行位于每帧对统计信息列表之前，键值对位于具有以下参数的帧格式之后：

psnr\_log\_version

日志文件格式的版本。将匹配stats\_version 。

字段

日志中包含的每帧对参数的逗号分隔列表。

每个显示的每帧对参数的描述如下：

n

输入帧的序号，从1开始

mse\_avg

Mean Square Error 比较帧的逐像素平均差，对所有图像分量取平均值。

mse\_y, mse\_u, mse\_v, mse\_r, mse\_g, mse\_b, mse\_a

后缀指定的组件的比较帧的逐像素平均差均方误差。

psnr\_y、psnr\_u、psnr\_v、psnr\_r、psnr\_g、psnr\_b、psnr\_a

由后缀指定的组件的比较帧的峰值信噪比。

max\_avg, max\_y, max\_u, max\_v

每个通道的最大允许值，以及所有通道的平均值。

* [例子](#Examples-145)

#### 39.197.1 示例

* 例如：
* movie=ref\_movie.mpg, setpts=PTS-STARTPTS [main];
* [main][ref] psnr="stats\_file=stats.log" [out]

在此示例中，将正在处理的输入文件与参考文件ref\_movie.mpg进行比较。每个单独帧的 PSNR 存储在stats.log中。

* 另一个使用不同容器的例子：
* ffmpeg -i main.mpg -i ref.mkv -lavfi "[0:v]settb=AVTB,setpts=PTS-STARTPTS[main];[1:v]settb=AVTB,setpts=PTS-STARTPTS[ref]; [main][ref]psnr" -f null -

### 39.198 上拉

下拉反转（反向电视电影）过滤器，能够处理混合硬电视电影、24000/1001 fps 逐行和 30000/1001 fps 逐行内容。

上拉过滤器旨在利用未来的上下文来做出决定。这个过滤器是无状态的，因为它不会锁定要遵循的模式，而是期待以下字段以识别匹配项并重建渐进式帧。

要生成具有均匀帧率的内容，请在 pullup 后插入 fps 过滤器，如果输入帧率为 29.97fps，则使用fps=24000/1001 ，对于 30fps 和（罕见的）电视电影 25fps 输入，使用fps=24 。

过滤器接受以下选项：

jl

jr

jt

jb

这些选项分别设置图像左侧、右侧、顶部和底部要忽略的“垃圾”数量。左右以8个像素为单位，上下以2行为单位。默认值为每边 8 个像素。

某人

设置严格的休息时间。将此选项设置为 1 将减少过滤器偶尔生成不匹配帧的机会，但也可能导致在高运动序列期间丢弃过多的帧。相反，将其设置为 -1 将使过滤匹配字段更加容易。这可能有助于处理场之间存在轻微模糊的视频，但也可能导致输出中出现隔行扫描帧。默认值为0 。

议员

设置要使用的公制平面。它接受以下值：

'我'

使用亮度平面。

'你'

使用色度蓝色平面。

' v '

使用色度红色平面。

这个选项可以设置为使用色度平面而不是默认的亮度平面来进行过滤器的计算。这可能会提高非常干净的源材料的准确性，但更有可能会降低准确性，尤其是在存在色度噪声（彩虹效应）或任何灰度视频的情况下。将mp设置为色度平面的主要目的是减少 CPU 负载并使 pullup 在慢速机器上实时可用。

为获得最佳效果（输出文件中没有重复的帧），有必要更改输出帧速率。例如，反转电视电影 NTSC 输入：

ffmpeg -i 输入 -vf pullup -r 24000/1001 ...

### 39.199 磅

更改视频量化参数 (QP)。

过滤器接受以下选项：

qp

设置量化参数的表达式。

该表达式通过 eval API 进行评估，并且可以包含以下常量：

已知的

如果索引不是 129，则为 1，否则为 0。

qp

从 -129 到 128 的顺序索引。

* [例子](#Examples-146)

#### 39.199.1 示例

* 一些等式，如：
* qp=2+2\*sin(PI\*qp)

### 39.200 随机

将帧内部缓存中的视频帧刷新为随机顺序。没有帧被丢弃。灵感来自[frei0r](#frei0r)神经过滤器。

框架

设置内部缓存帧数的大小，范围从2到512 。默认值为30 。

种子

0和UINT32\_MAX之间的整数。如果未指定，或者如果明确设置为小于0 ，则过滤器将尝试在尽力而为的基础上使用良好的随机种子。

### 39.201 readeia608

从视频帧的顶行读取隐藏式字幕 (EIA-608) 信息。

此过滤器为lavfi.readeia608.X.cc和lavfi.readia608.X.line添加帧元数据，其中X是带有 EIA-608 数据的已识别行的编号（从 0 开始）。每个元数据值的描述如下：

lavfi.readeia608.X.cc

这两个字节存储为 EIA-608 数据（以十六进制打印）。

lavfi.readeia608.X.line

识别和读取 EIA-608 数据的行号。

此过滤器接受以下选项：

扫描分钟

设置线以开始扫描 EIA-608 数据。默认为0 。

扫描最大值

将线设置为结束扫描 EIA-608 数据。默认值为29 。

spw

设置为同步码检测预留的宽度比例。默认值为0.27 。允许的范围是[0.1 - 0.7] 。

芯片组

启用检查奇偶校验位。如果出现奇偶校验错误，过滤器将为该字符输出0x00 。默认为假。

LP

进一步处理之前的低通线。默认启用。

* [命令](#Commands-133)
* [例子](#Examples-147)

#### 39.201.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

#### 39.201.2 示例

* 输出一个包含演示时间和前两行已识别 EIA-608 字幕数据的 csv。
* ffprobe -f lavfi -i movie=captioned\_video.mov,readeia608 -show\_entries frame=pts\_time:frame\_tags=lavfi.readeia608.0.cc,lavfi.readeia608.1.cc -of csv 文件

### 39.202 读维茨

从视频帧的顶行读取垂直间隔时间码 (VITC) 信息。

过滤器会添加帧元数据键lavfi.readvitc.tc\_str和时间码值。进一步的元数据键lavfi.readvitc.found设置为 0/1，具体取决于是否找到时间码数据。

此过滤器接受以下选项：

扫描最大值

设置扫描 VITC 数据的最大行数。如果该值设置为-1，则扫描整个视频帧。默认值为45 。

thr\_b

设置黑色的亮度阈值。接受 [0.0,1.0] 范围内的浮点数，默认值为0.2 。该值必须等于或小于thr\_w 。

thr\_w

设置白色的亮度阈值。接受 [0.0,1.0] 范围内的浮点数，默认值为0.6 。该值必须等于或大于thr\_b 。

* [例子](#Examples-148)

#### 39.202.1 示例

* 检测 VITC 数据并将其绘制到视频帧上；如果没有检测到有效的 VITC，绘制--:--:--:--作为占位符：
* ffmpeg -i input.avi -filter:v 'readvitc,drawtext=fontfile=FreeMono.ttf:text=%{metadata\\:lavfi.readvitc.tc\_str\\:--\\\\\\:--\\ \\\\:--\\\\\\:--}:x=(w-tw)/2:y=400-ascent'

### 39.203 重新映射

使用第二个重新映射像素：Xmap 和第三个：Ymap 输入视频流。

位置 (X, Y) 的目标像素将从源 (x, y) 位置选取，其中 x = Xmap(X, Y) 和 y = Ymap(X, Y)。如果映射值超出范围，则像素的零值将用于目标像素。

Xmap 和 Ymap 输入视频流的尺寸必须相同。输出视频流将具有 Xmap/Ymap 视频流维度。 Xmap 和 Ymap 输入视频流是 16 位深度，单通道。

格式

指定此过滤器输出的像素格式。可以是彩色或灰色。默认是颜色。

充满

指定未映射像素的颜色。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。默认颜色为黑色。

### 39.204 去除颗粒

removegrain 过滤器是用于渐进式视频的空间降噪器。

m0

为第一个平面设置模式。

立方米

为第二个平面设置模式。

平方米

设置第三平面的模式。

立方米

为第四平面设置模式。

模式的范围是从 0 到 24。每个模式的说明如下：

0

保持输入平面不变。默认。

1个

裁剪具有 8 个相邻像素中最小值和最大值的像素。

2个

裁剪 8 个相邻像素中第二个最小值和第二个最大值的像素。

3个

裁剪具有 8 个相邻像素的第三个最小值和最大值的像素。

4个

裁剪具有 8 个相邻像素的第四个最小值和最大值的像素。这相当于一个中值滤波器。

5个

行敏感剪裁提供最小的变化。

6个

线敏感剪裁，中间。

7

线敏感剪裁，中间。

8个

线敏感剪裁，中间。

9

在相邻像素最近的线上进行线敏感剪裁。

10

用最近的邻居替换目标像素。

11

[1 2 1] 水平和垂直内核模糊。

12

与模式 11 相同。

13

Bob 模式，从相邻像素最近的行插入顶场。

14

Bob 模式，从相邻像素最近的行插入底场。

15

Bob 模式，插入顶场。与 13 相同，但具有更复杂的插值公式。

16

Bob 模式，内插底场。与 14 相同，但具有更复杂的插值公式。

17

裁剪具有每对相对相邻像素的最大值和最小值的最小值和最大值的像素。

18

使用与当前像素的最大距离最小的相对邻居进行线敏感裁剪。

19

用它的 8 个邻居的平均值替换像素。

20

平均 9 个像素（[1 1 1] 水平和垂直模糊）。

21

使用相对邻居的平均值裁剪像素。

22

与模式 21 相同，但更简单、更快。

23

小边缘和光晕移除，但被认为是无用的。

24

类似于23。

### 39.205 删除徽标

抑制电视台徽标，使用图像文件来确定徽标由哪些像素组成。它的工作原理是用相邻像素填充构成徽标的像素。

过滤器接受以下选项：

文件名，f

设置滤镜位图文件，可以是libavformat支持的任何图片格式。图像文件的宽度和高度必须与正在处理的视频流的宽度和高度相匹配。

提供的位图中值为零的像素不被视为徽标的一部分，非零像素被视为徽标的一部分。如果徽标使用白色 (255) 而其余部分使用黑色 (0)，那么您将是安全的。为了制作滤镜位图，建议截取带有可见徽标的黑框的屏幕截图，然后使用阈值滤镜，然后使用腐蚀滤镜一次或两次。

如果需要，可以手动修复小斑点。请记住，如果徽标像素未被覆盖，过滤器质量将大大降低。将太多像素标记为标志的一部分并没有那么大的伤害，但它会增加覆盖图像所需的模糊量，并会破坏不必要的信息，额外的像素会减慢大标志的速度。

### 39.206 重复字段

此过滤器使用视频 ES 标头中的 repeat\_field 标志，并根据其值硬重复字段。

### 39.207反向

反转视频剪辑。

警告：此滤镜需要内存来缓冲整个剪辑，因此建议进行修剪。

* [例子](#Examples-149)

#### 39.207.1 示例

* 取一个片段的前 5 秒，然后反转它。
* 修剪=结束=5，反转

### 39.208 rgbashift

水平和/或垂直移动 R/G/B/A 像素。

过滤器接受以下选项：

rh

设置水平移动红色的量。

房车

设置垂直移动红色的量。

高

设置水平移动绿色的量。

gv

设置垂直移动绿色的量。

bh

设置水平移动蓝色的量。

bv

设置垂直移动蓝色的量。

啊

设置水平移动 alpha 的量。

影音

设置垂直移动 alpha 的量。

边缘

设置边缘模式，可以是smear 、 default 或warp 。

* [命令](#Commands-134)

#### 39.208.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.209 罗伯茨

应用罗伯茨交叉运算符输入视频流。

过滤器接受以下选项：

飞机

设置要处理的平面，未处理的平面将被复制。默认值 0xf，将处理所有平面。

规模

设置将与过滤结果相乘的值。

三角洲

设置将添加到过滤结果的值。

* [命令](#Commands-135)

#### 39.209.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.210 旋转

以弧度表示的任意角度旋转视频。

过滤器接受以下选项：

可选参数的描述如下。

角度，一个

为输入视频顺时针旋转的角度设置一个表达式，以弧度数表示。负值将导致逆时针旋转。默认设置为“0”。

为每一帧评估此表达式。

out\_w, ow

设置输出宽度表达式，默认值为“iw”。该表达式在配置期间仅被评估一次。

out\_h, 哦

设置输出高度表达式，默认值为“ih”。该表达式在配置期间仅被评估一次。

双线性

如果设置为 1，则启用双线性插值，值为 0 将禁用它。默认值为 1。

填充颜色，c

设置用于填充未被旋转图像覆盖的输出区域的颜色。有关此选项的一般语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。如果选择特殊值“none”，则不打印背景（例如，如果背景从不显示时很有用）。

默认值为“黑色”。

角度和输出大小的表达式可以包含以下常量和函数：

n

输入帧的序号，从0开始。在过滤第一帧之前总是NAN。

吨

输入帧的时间（以秒为单位），在配置过滤器时设置为 0。在过滤第一帧之前总是 NAN。

hsub

vsub

水平和垂直色度子样本值。例如，对于像素格式“yuv422p”， hsub为 2， vsub为 1。

in\_w, iw

在\_h，我

输入视频的宽度和高度

out\_w, ow

out\_h, 哦

宽度和高度表达式指定的填充区域的大小

旋转（一）

罗斯（一）

完全包含按弧度旋转的输入视频所需的最小宽度/高度。

out\_w和out\_h表达式时可用。

* [例子](#Examples-150)
* [命令](#Commands-136)

#### 39.210.1 例子

* 将输入顺时针旋转 PI/6 弧度：
* 旋转=PI/6
* 将输入逆时针旋转 PI/6 弧度：
* 旋转=-PI/6
* 将输入顺时针旋转 45 度：
* 旋转=45\*PI/180
* 应用周期为 T 的恒定旋转，从 PI/3 的角度开始：
* 旋转=PI/3+2\*PI\*t/T
* 使输入视频旋转振荡，周期为 T 秒，振幅为 A 弧度：
* 旋转=A\*sin(2\*PI/T\*t)
* 旋转视频，选择输出大小，以便整个旋转输入视频始终完全包含在输出中：
* rotate='2\*PI\*t:ow=hypot(iw,ih):oh=ow'
* 旋转视频，减小输出大小，以便不显示任何背景：
* rotate=2\*PI\*t:ow='min(iw,ih)/sqrt(2)':oh=ow:c=none

#### 39.210.2 命令

过滤器支持以下命令：

一、角度

设置角度表达式。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.211 萨布

应用形状自适应模糊。

过滤器接受以下选项：

亮度半径，lr

设置亮度模糊滤镜强度，必须是0.1-4.0范围内的值，默认值为1.0。值越大，图像越模糊，处理速度越慢。

luma\_pre\_filter\_radius, lpfr

设置亮度预过滤半径，必须是0.1-2.0范围内的值，默认值为1.0。

亮度强度，ls

设置仍要考虑的像素之间的亮度最大差异，必须是 0.1-100.0 范围内的值，默认值为 1.0。

色度半径，cr

设置色度模糊滤镜强度，必须是-0.9-4.0范围内的值。值越大，图像越模糊，处理速度越慢。

chroma\_pre\_filter\_radius, cpfr

设置色度预过滤半径，必须是-0.9-2.0范围内的值。

色度强度，CS

设置仍要考虑的像素之间的色度最大差异，必须是 -0.9-100.0 范围内的值。

如果未明确指定，每个色度选项值将设置为相应的亮度选项值。

### 39.212规模

使用 libswscale 库缩放（调整大小）输入视频。

缩放过滤器通过更改输出样本纵横比强制输出显示纵横比与输入相同。

如果输入图像格式与下一个过滤器请求的格式不同，缩放过滤器会将输入转换为请求的格式。

* [选项](#Options-95)
* [例子](#Examples-151)
* [命令](#Commands-137)

#### 39.212.1 选项

过滤器接受以下选项，或 libswscale 缩放器支持的任何选项。

有关缩放选项的完整列表，请参阅[(ffmpeg-scaler)ffmpeg-scaler 手册。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-scaler.html#scaler_005foptions)

宽度，w

身高，小时

设置输出视频维度表达式。默认值是输入维度。

如果宽度或w值为 0，则输入宽度用于输出。如果高度或h值为 0，则输入高度用于输出。

如果一个且只有一个值是 -n 且 n >= 1，则缩放过滤器将使用一个值来保持输入图像的纵横比，该纵横比是从另一个指定维度计算得出的。但是，之后它将确保计算出的维度可以被 n 整除，并在必要时调整该值。

如果两个值都是 -n 并且 n >= 1，则行为将与两个值都设置为 0 相同，如前所述。

请参阅下面的维度表达式中可接受的常量列表。

评估

指定何时计算宽度和高度表达式。它接受以下值：

'初始化'

在过滤器初始化期间或处理命令时仅对表达式求值一次。

'框架'

评估每个传入帧的表达式。

默认值为“ init ”。

国际间

设置隔行模式。它接受以下值：

' 1 '

强制交错感知缩放。

' 0 '

不要应用隔行缩放。

' -1 '

根据源帧是否标记为隔行扫描来选择隔行扫描感知缩放。

默认值为“ 0 ”。

旗帜

设置 libswscale 缩放标志。有关完整的值列表，请参阅[(ffmpeg-scaler)ffmpeg-scaler 手册。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-scaler.html#sws_005fflags)如果未明确指定，过滤器将应用默认标志。

参数 0，参数 1

为需要它们的缩放算法设置 libswscale 输入参数。有关完整文档，请参阅[(ffmpeg-scaler)ffmpeg-scaler 手册。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-scaler.html#sws_005fparams)如果未明确指定，过滤器将应用空参数。

大小，小号

设置视频大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。

in\_color\_matrix

out\_color\_matrix

设置输入/输出 YCbCr 颜色空间类型。

这允许覆盖自动检测的值，并允许强制将特定值用于输出和编码器。

如果未指定，则颜色空间类型取决于像素格式。

可能的值：

'汽车'

自动选择。

' bt709 '

格式符合国际电信联盟 (ITU) Recommendation BT.709。

' fcc '

设置符合美国联邦通信委员会 (FCC) 联邦法规 (CFR) 标题 47 (2003) 73.682 (a) 的颜色空间。

' bt601 '

' bt470 '

' smpte170m '

设置颜色空间符合：

* 国际电联无线电通信部门 (ITU-R) 建议书 BT.601
* ITU-R 建议书。 BT.470-6 (1998) 系统 B、B1 和 G
* 电影电视工程师协会 (SMPTE) ST 170:2004

' smpte240m '

设置符合 SMPTE ST 240:1999 的色彩空间。

' bt2020 '

设置符合ITU-R BT.2020非恒亮系统的色彩空间。

在范围内

超出范围

设置输入/输出 YCbCr 采样范围。

这允许覆盖自动检测的值，并允许强制将特定值用于输出和编码器。如果未指定，范围取决于像素格式。可能的值：

'自动/未知'

自动选择。

' jpeg/完整/pc '

设置完整范围（在 8 位亮度的情况下为 0-255）。

' mpeg/有限/电视'

设置“MPEG”范围（在 8 位亮度的情况下为 16-235）。

force\_original\_aspect\_ratio

如有必要，启用减少或增加输出视频宽度或高度以保持原始纵横比。可能的值：

'禁用'

按指定缩放视频并禁用此功能。

'减少'

如果需要，输出视频尺寸将自动减小。

'增加'

如果需要，输出视频尺寸将自动增加。

此选项的一个有用实例是，当您知道特定设备的最大允许分辨率时，您可以使用它来将输出视频限制为该分辨率，同时保持宽高比。例如，设备 A 允许播放 1280x720，而您的视频是 1920x800。使用此选项（将其设置为减少）并在命令行中指定 1280x720 会使输出为 1280x533。

w或h指定 -1 不同，您仍然需要指定输出分辨率才能使此选项生效。

force\_divisible\_by

force\_original\_aspect\_ratio一起使用时，确保输出尺寸（宽度和高度）都可以被给定整数整除。这类似于在w和h选项中使用-n 。

force\_original\_aspect\_ratio设置的值，相应地增加或减少分辨率。视频的宽高比可能会略有修改。

force\_original\_aspect\_ratio使视频适合或超过定义的分辨率，但编码器对宽度或高度的可分性有限制，则此选项会很方便。

w和h选项的值是包含以下常量的表达式：

in\_w

在\_h

输入的宽度和高度

我

我

in\_w和in\_h相同。

out\_w

out\_h

输出（缩放）宽度和高度

噢

哦

out\_w和out\_h相同

A

iw / ih相同

特区

输入样本纵横比

达尔

输入显示纵横比。从(iw / ih) \* sar计算。

hsub

vsub

水平和垂直输入色度子样本值。例如，对于像素格式“yuv422p”， hsub为 2， vsub为 1。

哦sub

卵子

水平和垂直输出色度子样本值。例如，对于像素格式“yuv422p”， hsub为 2， vsub为 1。

n

输入帧的（连续）编号，从 0 开始。仅适用于eval=frame 。

吨

输入帧的呈现时间戳，以秒数表示。仅适用于eval=frame 。

位置

帧在输入流中的位置（字节偏移量），如果此信息不可用和/或无意义（例如在合成视频的情况下），则为 NaN。仅适用于eval=frame 。

#### 39.212.2 例子

* 将输入视频缩放到 200x100 的大小
* 比例=w=200:h=100

这相当于：

比例=200:100

或者：

比例=200x100

* 指定输出大小的大小缩写：
* 规模=qcif

也可以写成：

规模=尺寸=qcif

* 将输入缩放为 2 倍：
* 规模=w=2\*iw:h=2\*ih
* 以上与以下内容相同：
* 比例=2\*in\_w:2\*in\_h
* 使用强制交错缩放将输入缩放为 2x：
* 比例=2\*iw:2\*ih:interl=1
* 将输入缩放为一半大小：
* 规模=w=iw/2:h=ih/2
* 增加宽度，并将高度设置为相同大小：
* 比例=3/2\*iw:ow
* 寻求希腊和谐：
* 比例=iw:1/PHI\*iw
* 标度=ih\*PHI:ih
* 增加高度，并将宽度设置为高度的 3/2：
* 刻度=w=3/2\*oh:h=3/5\*ih
* 增加大小，使大小成为色度子样本值的倍数：
* scale="trunc(3/2\*iw/hsub)\*hsub:trunc(3/2\*ih/vsub)\*vsub"
* 将宽度增加到最大 500 像素，保持与输入相同的纵横比：
* scale=w='min(500\, iw\*3/2):h=-1'
* 通过组合 scale 和 setsar 使像素成为正方形：
* scale='trunc(ih\*dar):ih',setsar=1/1
* 通过组合 scale 和 setsar 使像素成为正方形，确保生成的分辨率是均匀的（某些编解码器需要）：
* scale='trunc(ih\*dar/2)\*2:trunc(ih/2)\*2',setsar=1/1

#### 39.212.3 命令

该过滤器支持以下命令：

宽度，w

身高，小时

设置输出视频维度表达式。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.213 scale\_cuda

使用加速的 CUDA 内核缩放（调整大小）和转换（像素格式）输入视频。设置输出宽度和高度的方式与[缩放过滤器](#scale)的方式相同。

过滤器接受以下选项：

w

H

设置输出视频维度表达式。默认值是输入维度。

[比例过滤器](#scale)相同的表达式。

interp\_algo

设置用于缩放的算法：

最近的

最近邻

如果输入参数与所需的输出匹配，则默认使用。

双线性

双线性

双三次

双三次

这是默认设置。

兰佐斯

兰佐斯

格式

控制输出像素格式。默认情况下，或者如果未指定，则使用输入像素格式。

滤镜不支持 YUV 和 RGB 像素格式之间的转换。

直通

如果设置为 0，则处理每一帧，即使不需要转换也是如此。此模式可用于将过滤器用作耗尽有限解码器帧池的下游帧消费者的缓冲区。

如果设置为 1，则当帧与所需的输出参数匹配时，它们将按原样传递。这是默认行为。

参数

算法特定参数。

影响双三次算法的曲线。

force\_original\_aspect\_ratio

force\_divisible\_by

[比例过滤器选项的](#scale)工作方式相同。

* [例子](#Examples-152)

#### 39.213.1 例子

* 将输入缩放到 720p，保持宽高比并确保输出为 yuv420p。
* scale\_cuda=-2:720:格式=yuv420p
* 使用最近邻算法升级到 4K。
* scale\_cuda=4096:2160:interp\_algo=最近的
* 不要做任何转换或缩放，而是将所有输入帧复制到新分配的帧中。这对于处理过滤器和编码链很有用，否则会耗尽解码器帧池。
* scale\_cuda=直通=0

### 39.214 scale\_npp

使用 NVIDIA Performance Primitives (libnpp) 在 CUDA 视频帧上执行缩放和/或像素格式转换。设置输出宽度和高度的方式与缩放过滤器的方式相同。

接受以下附加选项：

格式

输出 CUDA 帧的像素格式。如果设置为字符串“same”（默认），将保留输入格式。请注意，硬件帧尚不支持自动格式协商和转换

interp\_algo

用于调整大小的插值算法。以下之一：

nn

最近的邻居。

线性的

立方体

cubic2p\_bspline

2 参数立方 (B=1, C=0)

cubic2p\_catmullrom

2 参数立方 (B=0, C=1/2)

cubic2p\_b05c03

2 参数立方（B=1/2，C=3/10）

极好的

超级采样

兰佐斯

force\_original\_aspect\_ratio

如有必要，启用减少或增加输出视频宽度或高度以保持原始纵横比。可能的值：

'禁用'

按指定缩放视频并禁用此功能。

'减少'

如果需要，输出视频尺寸将自动减小。

'增加'

如果需要，输出视频尺寸将自动增加。

此选项的一个有用实例是，当您知道特定设备的最大允许分辨率时，您可以使用它来将输出视频限制为该分辨率，同时保持宽高比。例如，设备 A 允许播放 1280x720，而您的视频是 1920x800。使用此选项（将其设置为减少）并在命令行中指定 1280x720 会使输出为 1280x533。

w或h指定 -1 不同，您仍然需要指定输出分辨率才能使此选项生效。

force\_divisible\_by

force\_original\_aspect\_ratio一起使用时，确保输出尺寸（宽度和高度）都可以被给定整数整除。这类似于在w和h选项中使用-n 。

force\_original\_aspect\_ratio设置的值，相应地增加或减少分辨率。视频的宽高比可能会略有修改。

force\_original\_aspect\_ratio使视频适合或超过定义的分辨率，但编码器对宽度或高度的可分性有限制，则此选项会很方便。

评估

指定何时计算宽度和高度表达式。它接受以下值：

'初始化'

在过滤器初始化期间或处理命令时仅对表达式求值一次。

'框架'

评估每个传入帧的表达式。

w和h选项的值是包含以下常量的表达式：

in\_w

在\_h

输入的宽度和高度

我

我

in\_w和in\_h相同。

out\_w

out\_h

输出（缩放）宽度和高度

噢

哦

out\_w和out\_h相同

A

iw / ih相同

特区

输入样本纵横比

达尔

输入显示纵横比。从(iw / ih) \* sar计算。

n

输入帧的（连续）编号，从 0 开始。仅适用于eval=frame 。

吨

输入帧的呈现时间戳，以秒数表示。仅适用于eval=frame 。

位置

帧在输入流中的位置（字节偏移量），如果此信息不可用和/或无意义（例如在合成视频的情况下），则为 NaN。仅适用于eval=frame 。

### 39.215 scale2ref

根据参考视频缩放（调整大小）输入视频。

查看可用选项的缩放过滤器，scale2ref 支持相同但使用参考视频而不是主要输入作为基础。 scale2ref 还支持w和h选项的以下附加常量：

主w

main\_h

主输入视频的宽高

main\_a

main\_w / main\_h相同

main\_sar

主输入视频的样本纵横比

main\_dar, mdar

主输入视频的显示宽高比。从(main\_w / main\_h) \* main\_sar计算。

main\_hsub

main\_vsub

主输入视频的水平和垂直色度子样本值。例如，对于像素格式“yuv422p”， hsub为 2， vsub为 1。

main\_n

主输入帧的（连续）编号，从 0 开始。仅适用于eval=frame 。

main\_t

主输入帧的呈现时间戳，以秒数表示。仅适用于eval=frame 。

main\_pos

帧在主输入流中的位置（字节偏移量），如果此信息不可用和/或无意义（例如在合成视频的情况下），则为 NaN。仅适用于eval=frame 。

* [例子](#Examples-153)
* [命令](#Commands-138)

#### 39.215.1 例子

* 在叠加之前缩放字幕流 (b) 以匹配主视频 (a) 的大小
* 'scale2ref[b][a];[a][b]覆盖'
* 将徽标缩放到视频高度的 1/10，同时保持其显示纵横比。
* [标志输入][视频输入]scale2ref=w=oh\*mdar:h=ih/10[标志输出][视频输出]

#### 39.215.2 命令

该过滤器支持以下命令：

宽度，w

身高，小时

设置输出视频维度表达式。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

### 39.216 scale2ref\_npp

使用 NVIDIA Performance Primitives (libnpp) 根据参考视频缩放（调整大小）输入视频。

查看可用选项的[scale\_npp](#scale_005fnpp)过滤器，scale2ref\_npp 支持相同但使用参考视频而不是主要输入作为基础。 scale2ref\_npp 还支持w和h选项的以下附加常量：

主w

main\_h

主输入视频的宽高

main\_a

main\_w / main\_h相同

main\_sar

主输入视频的样本纵横比

main\_dar, mdar

主输入视频的显示宽高比。从(main\_w / main\_h) \* main\_sar计算。

main\_n

主输入帧的（连续）编号，从 0 开始。仅适用于eval=frame 。

main\_t

主输入帧的呈现时间戳，以秒数表示。仅适用于eval=frame 。

main\_pos

帧在主输入流中的位置（字节偏移量），如果此信息不可用和/或无意义（例如在合成视频的情况下），则为 NaN。仅适用于eval=frame 。

* [例子](#Examples-154)

#### 39.216.1 示例

* 在叠加之前缩放字幕流 (b) 以匹配主视频 (a) 的大小
* 'scale2ref\_npp[b][a];[a][b]overlay\_cuda'
* 将徽标缩放到视频高度的 1/10，同时保持其显示纵横比。
* [登录][视频输入]scale2ref\_npp=w=oh\*mdar:h=ih/10[标志输出][视频输出]

### 39.217 夏尔

将 scharr 运算符应用于输入视频流。

过滤器接受以下选项：

飞机

设置要处理的平面，未处理的平面将被复制。默认值 0xf，将处理所有平面。

规模

设置将与过滤结果相乘的值。

三角洲

设置将添加到过滤结果的值。

* [命令](#Commands-139)

#### 39.217.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.218 卷轴

以恒定速度水平和/或垂直滚动输入视频。

过滤器接受以下选项：

水平，高

设置水平滚动速度。默认值为 0。允许的范围是从 -1 到 1。负值会更改滚动方向。

垂直，v

设置垂直滚动速度。默认值为 0。允许的范围是从 -1 到 1。负值会更改滚动方向。

hpos

设置初始水平滚动位置。默认值为 0。允许的范围是从 0 到 1。

vpos

设置初始垂直滚动位置。默认值为 0。允许的范围是从 0 到 1。

* [命令](#Commands-140)

#### 39.218.1 命令

此过滤器支持以下[命令](#commands)：

水平，高

设置水平滚动速度。

垂直，v

设置垂直滚动速度。

### 39.219 scdet

检测视频场景变化。

该过滤器使用帧间 mafd 和场景得分设置帧元数据，并将帧转发到下一个过滤器，因此它们可以使用这些元数据来检测场景变化或其他。

此外，此过滤器会在检测到场景发生阈值变化时记录一条消息并设置帧元数据。

lavfi.scd.mafd元数据键是用 mafd 为每一帧设置的。

lavfi.scd.score元数据键设置有每一帧的场景变化分数以检测场景变化。

lavfi.scd.time元数据键设置为当前过滤的帧时间，该帧时间检测具有阈值的场景变化。

过滤器接受以下选项：

阈值，吨

将场景变化检测阈值设置为最大变化的百分比。好的值在[8.0, 14.0]范围内。阈值的范围是[0., 100.] 。

默认值为10 。

sc\_pass, s

设置标志以将场景变化帧传递给下一个过滤器。默认值为0如果您只想获取场景变化帧的快照，则可以启用它。

### 39.220 选择性颜色

将青色、品红色、黄色和黑色 (CMYK) 调整为特定颜色范围（例如“红色”、“黄色”、“绿色”、“青色”...）。调整范围由颜色的“纯度”（即已经饱和的程度）定义。

此滤镜类似于 Adobe Photoshop 可选颜色工具。

过滤器接受以下选项：

校正方法

选择颜色校正方法。

可用值是：

'绝对'

指定的调整“按原样”应用（添加/减去原始像素分量值）。

'亲戚'

指定的调整是相对于原始组件值的。

默认是绝对的。

红色

红色像素调整（红色分量最大的像素）

黄色

黄色像素调整（蓝色分量最小的像素）

青菜

绿色像素调整（绿色分量最大的像素）

青色

青色像素调整（红色分量最小的像素）

蓝调

蓝色像素调整（蓝色分量最大的像素）

洋红色

品红色像素调整（绿色分量最小的像素）

白人

白色像素调整（所有分量都大于 128 的像素）

中立者

调整除纯黑和纯白以外的所有像素

黑人

黑色像素调整（所有分量小于 128 的像素）

文件

要从中导入设置的Photoshop 选择性颜色文件 ( .asv )。

所有调整设置（红色、黄色、...）最多接受 [-1,1] 范围内的 4 个空格分隔的浮点调整值，分别用于调整像素的青色、品红色、黄色和黑色的量它的范围。

* [例子](#Examples-155)

#### 39.220.1 示例

* 每个绿色区域的青色增加 50%，黄色减少 33%，蓝色区域的洋红色增加 27%：
* 选择性颜色=绿色=.5 0 -.33 0:蓝色=0 .27
* 使用 Photoshop 选择性颜色预设：
* selectivecolor=psfile=MySelectiveColorPresets/Misty.asv

### 39.221 个单独的字段

separatefields采用基于帧的视频输入并将每个帧拆分为其组件场，从而生成具有两倍帧速率和两倍帧数的新半高剪辑。

该过滤器使用帧中的场优势信息来决定将每对场中的哪一个放在输出的第一位。如果它弄错了，请在separatefields过滤器之前使用[setfield过滤器](#setfield)。

### 39.222 setdar, 塞萨尔

setdar过滤器设置过滤器输出视频的显示纵横比。

这是通过根据以下等式更改指定的样本（又名像素）纵横比来完成的：

DAR = HORIZONTAL\_RESOLUTION / VERTICAL\_RESOLUTION \* SAR

请记住， setdar过滤器不会修改视频帧的像素尺寸。此外，此过滤器设置的显示纵横比可能会被过滤器链中的后续过滤器更改，例如在缩放的情况下或如果应用另一个“setdar”或“setsar”过滤器。

setsar过滤器设置过滤器输出视频的样本（又名像素）纵横比。

请注意，由于应用了此过滤器，输出显示纵横比将根据上述等式发生变化。

请记住，由setsar过滤器设置的样本纵横比可能会被过滤器链中的后续过滤器更改，例如，如果应用了另一个“setsar”或“setdar”过滤器。

它接受以下参数：

r、ratio、dar（仅setdar ）、sar（仅setsar ）

设置滤镜使用的纵横比。

num : den形式的字符串，其中num和den是纵横比的分子和分母。如果未指定参数，则假定值为“0”。如果使用“ num : den ”形式，则应转义:字符。

最大限度

将表达的纵横比缩小为有理数时，设置用于表达分子和分母的最大整数值。默认值为100 。

参数sar是一个包含以下常量的表达式：

E、PI、PHI

这些是数学常数 e（欧拉数）、pi（希腊圆周率）和 phi（黄金比例）的近似值。

w, h

输入的宽度和高度。

A

w / h相同。

特区

输入样本纵横比。

达尔

输入显示纵横比。它与 ( w / h )\* sar相同。

hsub, vsub

水平和垂直色度子样本值。例如，对于像素格式“yuv422p”， hsub为 2， vsub为 1。

* [例子](#Examples-156)

#### 39.222.1 例子

* 要将显示纵横比更改为 16:9，请指定以下选项之一：
* setdar=dar=1.77777
* setdar=dar=16/9
* 要将样本宽高比更改为 10:11，请指定：
* setsar=sar=10/11
* 要将显示纵横比设置为 16:9，并在纵横比缩小中指定最大整数值 1000，请使用以下命令：
* setdar=比率=16/9:最大=1000

### 39.223 设定场

输出视频帧的力场。

setfield过滤器标记输出帧的交错类型字段。它不会更改输入帧，而只会设置相应的属性，这会影响后续过滤器（例如fieldorder或yadif ）如何处理帧。

过滤器接受以下选项：

模式

可用值是：

'汽车'

保持相同的字段属性。

'好朋友'

将帧标记为底场优先。

' tff '

将帧标记为顶场优先。

'前卫'

将帧标记为渐进式。

### 39.224 设置参数

输出视频帧的强制帧参数。

setparams过滤器标记输出帧的交错和颜色范围。它不会更改输入帧，而只会设置相应的属性，这会影响过滤器/编码器如何处理帧。

现场模式

可用值是：

'汽车'

保持相同的字段属性（默认）。

'好朋友'

将帧标记为底场优先。

' tff '

将帧标记为顶场优先。

'前卫'

将帧标记为渐进式。

范围

可用值是：

'汽车'

保持相同的颜色范围属性（默认）。

'未指定，未知'

将框架标记为未指定的颜色范围。

'有限，电视，mpeg '

将框架标记为有限范围。

'完整，电脑，jpeg '

将帧标记为完整范围。

原色

设置颜色原色。可用值是：

'汽车'

保持相同的颜色原色属性（默认）。

' bt709 '

'未知'

' bt470m '

' bt470bg '

' smpte170m '

' smpte240m '

'电影'

' bt2020 '

' smpte428 '

' smpte431 '

' smpte432 '

' jedec-p22 '

颜色\_trc

设置颜色转移。可用值是：

'汽车'

保持相同的颜色 trc 属性（默认）。

' bt709 '

'未知'

' bt470m '

' bt470bg '

' smpte170m '

' smpte240m '

'线性'

' log100 '

' log316 '

' iec61966-2-4 '

' bt1361e '

' iec61966-2-1 '

' bt2020-10 '

' bt2020-12 '

' smpte2084 '

' smpte428 '

' arib-std-b67 '

色彩空间

设置色彩空间。可用值是：

'汽车'

保持相同的色彩空间属性（默认）。

' gbr '

' bt709 '

'未知'

' fcc '

' bt470bg '

' smpte170m '

' smpte240m '

' ycgco '

' bt2020nc '

' bt2020c '

' smpte2085 '

'色度衍生-nc '

'色度衍生-c '

' ictcp '

### 39.225 锐化\_npp

使用 NVIDIA Performance Primitives (libnpp) 通过边界控制执行图像锐化。

接受以下附加选项：

border\_type

要使用的广告框架边框的采样类型。以下之一：

复制

复制像素值。

### 39.226 剪切

对输入视频应用剪切变换。

此过滤器支持以下选项：

嘘

X 方向的剪切因子。默认值为 0。允许的范围为 -2 到 2。

害羞的

Y 方向的剪切因子。默认值为 0。允许的范围为 -2 到 2。

填充颜色，c

设置用于填充未被转换视频覆盖的输出区域的颜色。有关此选项的一般语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。如果选择特殊值“none”，则不打印背景（例如，如果背景从不显示时很有用）。

默认值为“黑色”。

插话

设置插值类型。可以是双线性的或最近的。默认是双线性的。

* [命令](#Commands-141)

#### 39.226.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.227 显示信息

显示包含每个输入视频帧的各种信息的一行。输入视频没有被修改。

此过滤器支持以下选项：

校验和

计算每个平面的校验和。默认启用。

显示的行包含一系列形式为 key : value的键/值对。

输出中显示以下值：

n

输入帧的（连续）编号，从 0 开始。

积分

输入帧的 Presentation TimeStamp，表示为多个时基单位。时基单位取决于滤波器输入焊盘。

pts\_time

输入帧的演示时间戳，以秒数表示。

位置

帧在输入流中的位置，如果此信息不可用和/或无意义（例如在合成视频的情况下），则为 -1。

调速器

像素格式名称。

特区

输入帧的样本纵横比，以num / den的形式表示。

秒

输入框的大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。

我

隔行扫描模式的类型（“P”代表“逐行扫描”，“T”代表顶场优先，“B”代表底场优先）。

关键

如果帧是关键帧，则为 1，否则为 0。

类型

输入帧的图片类型（“I”代表 I 帧，“P”代表 P 帧，“B”代表 B 帧，或者“？”代表未知类型）。另请参阅AVPictureType枚举和libavutil/avutil.h中定义的av\_get\_picture\_type\_char函数的文档。

校验和

输入帧所有平面的 Adler-32 校验和（以十六进制打印）。

平面校验和

输入帧的每个平面的Adler-32校验和（十六进制打印），以“[ c0 c1 c2 c3 ]”。

意思是

输入帧每个平面像素的平均值，以“[ mean0 均值1 均值2 均值 3 ]”。

标准开发

输入帧各平面像素值的标准差，以“[ stdev0 标准设备 1 标准开发2 标准设备 3 ]”。

### 39.228 显示调色板

显示每帧的 256 色调色板。此过滤器仅与pal8像素格式帧相关。

它接受以下选项：

秒

设置用于表示一个调色板颜色条目的框的大小。默认值为30 （对于30x30像素框）。

### 39.229 随机帧

重新排序和/或复制和/或删除视频帧。

它接受以下参数：

映射

设置输入帧的目标索引。这是空格或“|”将输入帧映射到输出帧的单独索引列表。索引数还设置了每个索引可能具有的最大值。 '-1' 索引有特殊含义，即丢帧。

第一帧的索引为 0。默认情况下保持输入不变。

* [例子](#Examples-157)

#### 39.229.1 示例

* 交换每三帧输入的第二帧和第三帧：
* ffmpeg -i 输入 -vf "shuffleframes=0 2 1" 输出
* 交换每十帧输入的第 10 帧和第 1 帧：
* ffmpeg -i 输入 -vf "shuffleframes=9 1 2 3 4 5 6 7 8 0" 输出

### 39.230 随机像素

重新排列视频帧中的像素。

此过滤器接受以下选项：

方向，d

设置随机播放方向。可以正向或反向。默认方向是向前的。

模式，米

设置随机模式。可以是水平、垂直或块模式。

宽度，w

身高，小时

设置 shuffle block\_size。在水平随机模式的情况下，仅使用尺寸的宽度部分，而在垂直随机模式的情况下，仅使用尺寸的高度部分。

种子

设置用于混洗像素的随机种子。主要用于设置能够反向过滤过程以获得原始输入。例如，要反转正向随机播放，您需要使用相同的参数和完全相同的种子并将方向设置为反向。

### 39.231 shuffleplanes

重新排序和/或复制视频平面。

它接受以下参数：

地图0

要用作第一个输出平面的输入平面的索引。

地图1

要用作第二个输出平面的输入平面的索引。

地图2

用作第三个输出平面的输入平面的索引。

地图3

用作第四输出平面的输入平面的索引。

第一个平面的索引为 0。默认情况下保持输入不变。

* [例子](#Examples-158)

#### 39.231.1 例子

* 交换输入的第二和第三平面：
* ffmpeg -i 输入 -vf shuffleplanes=0:2:1:3 输出

### 39.232 信号统计

评估有助于确定与模拟视频媒体数字化相关的问题的各种视觉指标。

默认情况下，过滤器将记录这些元数据值：

YMIN

显示输入帧中包含的最小 Y 值。在[0-255]范围内表示。

黄

在输入框内的 10% 百分位数处显示 Y 值。在[0-255]范围内表示。

YAVG

显示输入帧内的平均 Y 值。在[0-255]范围内表示。

雅高

在输入帧内的 90% 百分位数处显示 Y 值。在[0-255]范围内表示。

YMAX

显示输入框内包含的最大 Y 值。在[0-255]范围内表示。

UMIN

显示输入帧中包含的最小 U 值。在[0-255]范围内表示。

超低

显示输入框内 10% 百分位数的 U 值。在[0-255]范围内表示。

无人机

显示输入帧内的平均 U 值。在[0-255]范围内表示。

超高

在输入帧内的 90% 百分位数处显示 U 值。在[0-255]范围内表示。

优美思

显示输入帧中包含的最大 U 值。在[0-255]范围内表示。

最小值

显示输入帧中包含的最小 V 值。在[0-255]范围内表示。

低电压

在输入帧内的 10% 百分位数处显示 V 值。在[0-255]范围内表示。

变频调速器

显示输入帧内的平均 V 值。在[0-255]范围内表示。

高电平

在输入帧内的 90% 百分位数处显示 V 值。在[0-255]范围内表示。

VMAX

显示输入帧中包含的最大 V 值。在[0-255]范围内表示。

赛特敏

显示输入帧中包含的最小饱和度值。表示范围为[0-~181.02]。

萨特罗

显示输入帧内 10% 百分位数的饱和度值。表示范围为[0-~181.02]。

卫星电视

显示输入帧内的平均饱和度值。表示范围为[0-~181.02]。

萨提格

显示输入帧内 90% 百分位数的饱和度值。表示范围为[0-~181.02]。

卫星最大

显示输入帧中包含的最大饱和度值。表示范围为[0-~181.02]。

休美

显示输入帧内色调的中值。以[0-360]的范围表示。

华视

显示输入帧内色调的平均值。以[0-360]的范围表示。

YDIF

显示当前帧中 Y 平面的所有值与上一输入帧的对应值之间的样本值差异的平均值。在[0-255]范围内表示。

UDIF

显示当前帧中 U 平面的所有值与上一输入帧的对应值之间的样本值差异的平均值。在[0-255]范围内表示。

VDIF

显示当前帧中 V 平面的所有值与前一输入帧的相应值之间的样本值差异的平均值。在[0-255]范围内表示。

深度

显示当前帧中 Y 平面的位深度。以[0-16]的范围表示。

泛比特深度

显示当前帧中 U 平面的位深度。以[0-16]的范围表示。

位深度

显示当前帧中 V 平面的位深度。以[0-16]的范围表示。

过滤器接受以下选项：

状态

出去

stat指定图像分析的附加形式。 out输出视频，突出显示指定类型的像素。

两个选项都接受以下值：

'兜售'

识别时间异常值像素。时间异常值是与同一场的相邻像素不同的像素。时间异常值的示例包括视频丢失、磁头堵塞或磁带跟踪问题的结果。

' vrep '

识别垂直线重复。垂直线重复包括帧内的相似像素行。在原生数字视频中，垂直线重复很常见，但这种模式在从模拟源数字化的视频中并不常见。当它出现在由模拟源数字化产生的视频中时，它可以指示漏失补偿器的隐藏。

'带来'

识别超出合法广播范围的像素。

颜色，c

输出选项的高亮颜色。默认颜色为黄色。

* [例子](#Examples-159)

#### 39.232.1 例子

* 各种视频指标的输出数据：
* ffprobe -f lavfi movie=example.mov,signalstats="stat=tout+vrep+brng" -show\_frames
* 输出有关每帧 Y 平面的最小值和最大值的特定数据：
* ffprobe -f lavfi movie=example.mov,signalstats -show\_entries frame\_tags=lavfi.signalstats.YMAX,lavfi.signalstats.YMIN
* 播放视频，同时以红色突出显示广播范围之外的像素。
* ffplay example.mov -vf signalstats="out=brng:color=red"
* 在帧上绘制带有 signalstats 元数据的播放视频。
* ffplay example.mov -vf signalstats=stat=brng+vrep+tout,drawtext=fontfile=FreeSerif.ttf:textfile=signalstat\_drawtext.txt 复制代码

命令中使用的signalstat\_drawtext.txt内容为：

时间 %{pts:hms}

Y (%{metadata:lavfi.signalstats.YMIN}-%{metadata:lavfi.signalstats.YMAX})

U (%{metadata:lavfi.signalstats.UMIN}-%{metadata:lavfi.signalstats.UMAX})

V (%{元数据:lavfi.signalstats.VMIN}-%{元数据:lavfi.signalstats.VMAX})

最大饱和度：%{metadata:lavfi.signalstats.SATMAX}

### 39.233 签名

计算 MPEG-7 视频签名。过滤器可以处理多个输入。在这种情况下，可以额外计算输入之间的匹配。过滤器总是通过第一个输入。每个流的签名可以写入一个文件。

它接受以下选项：

检测模式

启用或禁用匹配过程。

可用值是：

'关闭'

禁用匹配计算（默认）。

'满'

计算整个视频的匹配并输出是整个视频匹配还是仅部分匹配。

'快'

仅计算直到找到匹配项或视频结束。在某些情况下应该更快。

nb\_输入

设置输入的数量。选项值必须是非负整数。默认值为 1。

文件名

设置写入输出的路径。如果有多个输入，路径必须是一个原型，即必须包含 %d 或 %0nd（其中 n 是一个正整数），它将被替换为输入数字。如果未指定文件名，则不会写入任何输出。这是默认设置。

格式

选择输出格式。

可用值是：

'二进制'

使用指定的二进制表示（默认）。

' XML '

使用指定的 xml 表示。

th\_d

设置阈值以将一个词检测为相似。选项值必须是大于零的整数。默认值为 9000。

th\_dc

设置阈值以将所有单词检测为相似。选项值必须是大于零的整数。默认值为 60000。

th\_xh

设置阈值以将帧检测为相似。选项值必须是大于零的整数。默认值为 116。

th\_di

以帧为单位设置序列的最小长度，以将其识别为匹配序列。选项值必须是非负整数值。默认值为 0。

那个

设置最小关系，匹配帧到所有帧必须具有的关系。选项值必须是介于 0 和 1 之间的双精度值。默认值为 0.5。

* [例子](#Examples-160)

#### 39.233.1 示例

* 计算输入视频的签名并将其存储在 signature.bin 中：
* ffmpeg -i input.mkv -vf signature=文件名=signature.bin -map 0:v -f null -
* 检测两个视频是否匹配，将签名以XML格式存储在signature0.xml和signature1.xml中：
* ffmpeg -i input1.mkv -i input2.mkv -filter\_complex "[0:v][1:v] signature=nb\_inputs=2:detectmode=full:format=xml:filename=signature%d.xml" -map :v -f 空 -

### 39.234 西蒂

计算视频的空间信息 (SI) 和时间信息 (TI) 分数，如 ITU-T P.910：多媒体应用的主观视频质量评估方法中所定义。可在<https://www.itu.int/rec/T-REC-P.910-199909-S/en>获取 PDF 。

它接受以下选项：

打印摘要

如果设置为 1，摘要统计信息将打印到控制台。默认 0。

* [例子](#Examples-161)

#### 39.234.1 示例

* 要计算 SI/TI 指标并打印摘要：
* ffmpeg -i input.mp4 -vf siti=print\_summary=1 -f null -

### 39.235 智能模糊

在不影响轮廓的情况下模糊输入视频。

它接受以下选项：

亮度半径，lr

设置亮度半径。选项值必须是 [0.1,5.0] 范围内的浮点数，指定用于模糊图像的高斯滤波器的方差（越大越慢）。默认值为 1.0。

亮度强度，ls

设置亮度强度。选项值必须是配置模糊的 [-1.0,1.0] 范围内的浮点数。 [0.0,1.0] 中的值会使图像模糊，而 [-1.0,0.0] 中的值会使图像锐化。默认值为 1.0。

亮度阈值，lt

设置用作系数的亮度阈值，以确定像素是否应该模糊。选项值必须是 [-30,30] 范围内的整数。值为 0 将过滤所有图像，包含在 [0,30] 中的值将过滤平坦区域，包含在 [-30,0] 中的值将过滤边缘。默认值为 0。

色度半径，cr

设置色度半径。选项值必须是 [0.1,5.0] 范围内的浮点数，指定用于模糊图像的高斯滤波器的方差（越大越慢）。默认值为luma\_radius 。

色度强度，CS

设置色度强度。选项值必须是配置模糊的 [-1.0,1.0] 范围内的浮点数。 [0.0,1.0] 中的值会使图像模糊，而 [-1.0,0.0] 中的值会使图像锐化。默认值为luma\_strength 。

色度阈值，ct

设置用作系数的色度阈值，以确定像素是否应该模糊。选项值必须是 [-30,30] 范围内的整数。值为 0 将过滤所有图像，包含在 [0,30] 中的值将过滤平坦区域，包含在 [-30,0] 中的值将过滤边缘。默认值为luma\_threshold 。

如果未明确设置色度选项，则会设置相应的亮度值。

### 39.236索贝尔

将 sobel 运算符应用于输入视频流。

过滤器接受以下选项：

飞机

设置要处理的平面，未处理的平面将被复制。默认值 0xf，将处理所有平面。

规模

设置将与过滤结果相乘的值。

三角洲

设置将添加到过滤结果的值。

* [命令](#Commands-142)

#### 39.236.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.237 人

质量级别6的情况下 - 所有）移位压缩和解压缩图像并平均结果。

过滤器接受以下选项：

质量

设置质量。此选项定义平均级别数。它接受 0-6 范围内的整数。如果设置为0 ，过滤器将无效。值为6意味着质量更高。对于该值的每个增量，速度下降大约 2 倍。默认值为3 。

qp

强制一个恒定的量化参数。如果未设置，过滤器将使用视频流中的 QP（如果可用）。

模式

设置阈值模式。可用模式有：

'硬'

设置硬阈值（默认）。

'软'

设置软阈值（更好的去振铃效果，但可能更模糊）。

使用\_bframe\_qp

1 ，则启用来自 B 帧的 QP 。使用此选项可能会导致闪烁，因为 B 帧通常具有较大的 QP。默认值为0 （未启用）。

* [命令](#Commands-143)

#### 39.237.1 命令

该过滤器支持以下命令：

质量、水平

设置质量级别。值max可用于设置最大级别，当前为6 。

### 39.238 高级

通过应用一种基于卷积神经网络的超分辨率方法来缩放输入。支持机型：

* 超分辨率卷积神经网络模型 (SRCNN)。请参阅<https://arxiv.org/abs/1501.00092>。
* 高效的亚像素卷积神经网络模型 (ESPCN)。请参阅<https://arxiv.org/abs/1609.05158>。

训练脚本和模型文件（.pb）保存脚本可以在[https://github.com/XueweiMeng/sr/tree/sr\_dnn\_native找到](https://github.com/XueweiMeng/sr/tree/sr_dnn_native)。原始存储库位于<https://github.com/HighVoltageRocknRoll/sr.git>。

可以使用 tools/python/convert.py 从 TensorFlow 模型文件 (.pb) 生成原生模型文件 (.model)

过滤器接受以下选项：

dnn\_backend

指定用于模型加载和执行的 DNN 后端。此选项接受以下值：

'本土'

DNN 加载和执行的本机实现。

'张量流'

TensorFlow 后端。要启用此后端，您需要为 C 库安装 TensorFlow（请参阅[https://www.tensorflow.org/install/lang\_c ）并使用](https://www.tensorflow.org/install/lang_c)--enable-libtensorflow配置 FFmpeg

默认值为 ' native '。

模型

设置模型文件的路径，指定网络架构及其参数。请注意，不同的后端使用不同的文件格式。 TensorFlow 后端可以加载两种格式的文件，而原生后端只能加载其格式的文件。

比例因子

为 SRCNN 模型设置比例因子。允许的值为2 、 3和4 。默认值为2 。比例因子对于 SRCNN 模型是必需的，因为它接受使用具有适当比例因子的双三次放大进行放大的输入。

要获得完整功能（例如异步执行），请使用[dnn\_processing](#dnn_005fprocessing)过滤器。

### 39.239 西姆卡

获取两个输入视频之间的 SSIM（结构相似性度量）。

该过滤器接收两个输入视频，第一个输入被认为是“主要”来源，并不变地传递到输出。第二个输入用作计算 SSIM 的“参考”视频。

两个视频输入必须具有相同的分辨率和像素格式，此过滤器才能正常工作。它还假设两个输入具有相同数量的帧，将它们一一比较。

过滤器存储计算出的每一帧的 SSIM。

接受的参数的描述如下。

统计文件，f

如果指定，过滤器将使用命名文件来保存每个单独帧的 SSIM。当文件名等于“-”时，数据被发送到标准输出。

stats\_file ，则打印的文件包含一系列键/值对，形式为键：每个比较的帧的值。

每个显示参数的说明如下：

n

输入帧的序号，从1开始

Y, U, V, R, G, B

后缀指定的组件的比较帧的 SSIM。

全部

整个帧的比较帧的 SSIM。

D b

同上，但以 dB 表示。

此过滤器还支持[帧同步](#framesync)选项。

* [例子](#Examples-162)

#### 39.239.1 例子

* 例如：
* movie=ref\_movie.mpg, setpts=PTS-STARTPTS [main];
* [main][ref] ssim="stats\_file=stats.log" [out]

在此示例中，将正在处理的输入文件与参考文件ref\_movie.mpg进行比较。每个单独帧的 SSIM 存储在stats.log中。

* 另一个同时使用 psnr 和 ssim 的例子：
* ffmpeg -i main.mpg -i ref.mpg -lavfi "ssim;[0:v][1:v]psnr" -f null -
* 另一个使用不同容器的例子：
* ffmpeg -i main.mpg -i ref.mkv -lavfi "[0:v]settb=AVTB,setpts=PTS-STARTPTS[main];[1:v]settb=AVTB,setpts=PTS-STARTPTS[ref]; [main][ref]ssim" -f null -

### 39.240 stereo3d

在不同的立体图像格式之间转换。

过滤器接受以下选项：

在

设置输入的立体图像格式。

输入图像格式的可用值是：

' sbsl '

并排平行（左眼在左，右眼在右）

' sbsr '

并排交叉眼（右眼左，左眼右）

' sbs2l '

并排平行于半宽分辨率（左眼在左，右眼在右）

' sbs2r '

半宽分辨率并排十字眼（右眼左，左眼右）

' abl '

'表'

上下（左眼在上，右眼在下）

' abr '

' tbr '

上下（右眼在上，左眼在下）

' ab2l '

' tb2l '

上下半高分辨率（左眼在上，右眼在下）

' ab2r '

' tb2r '

上下半高分辨率（右眼在上，左眼在下）

'艾尔'

交替帧（左眼第一，右眼第二）

'阿'

交替帧（右眼第一，左眼第二）

' irl '

交错行（左眼有第一行，右眼从下一行开始）

'错误'

交错行（右眼有第一行，左眼从下一行开始）

' icl '

交错列，左眼优先

' icr '

交错列，右眼优先

默认值为“ sbsl ”。

出去

设置输出的立体图像格式。

' sbsl '

并排平行（左眼在左，右眼在右）

' sbsr '

并排交叉眼（右眼左，左眼右）

' sbs2l '

并排平行于半宽分辨率（左眼在左，右眼在右）

' sbs2r '

半宽分辨率并排十字眼（右眼左，左眼右）

' abl '

'表'

上下（左眼在上，右眼在下）

' abr '

' tbr '

上下（右眼在上，左眼在下）

' ab2l '

' tb2l '

上下半高分辨率（左眼在上，右眼在下）

' ab2r '

' tb2r '

上下半高分辨率（右眼在上，左眼在下）

'艾尔'

交替帧（左眼第一，右眼第二）

'阿'

交替帧（右眼第一，左眼第二）

' irl '

交错行（左眼有第一行，右眼从下一行开始）

'错误'

交错行（右眼有第一行，左眼从下一行开始）

'套利'

浮雕红/蓝灰（左眼红色滤镜，右眼蓝色滤镜）

' argg '

浮雕红色/绿色灰色（左眼红色滤镜，右眼绿色滤镜）

' arcg '

浮雕红色/青灰色（左眼红色滤镜，右眼青色滤镜）

'拱门'

浮雕红色/青色半色（左眼红色滤镜，右眼青色滤镜）

' arcc '

浮雕红色/青色（左眼红色滤镜，右眼青色滤镜）

' arcd '

使用杜波瓦最小二乘投影优化的浮雕红色/青色（左眼红色滤镜，右眼青色滤镜）

'聚合'

浮雕绿色/洋红色灰色（左眼绿色滤镜，右眼洋红色滤镜）

'啊'

浮雕绿色/洋红色半色（左眼绿色滤镜，右眼洋红色滤镜）

' agmc '

浮雕绿色/洋红色（左眼绿色滤镜，右眼洋红色滤镜）

' agmd '

使用杜波瓦的最小二乘投影优化的浮雕绿色/洋红色（左眼绿色滤镜，右眼洋红色滤镜）

'哎呀'

立体黄色/蓝灰色（左眼黄色滤镜，右眼蓝色滤镜）

'哎呀'

浮雕黄色/蓝色半色（左眼黄色滤镜，右眼蓝色滤镜）

' aybc '

浮雕黄色/蓝色（左眼黄色滤镜，右眼蓝色滤镜）

'哎呀'

浮雕黄色/蓝色用 dubois 的最小二乘投影优化（左眼黄色滤镜，右眼蓝色滤镜）

'毫升'

单声道输出（仅左眼）

'先生'

单声道输出（仅右眼）

'氯'

棋盘格，左眼优先

'铬'

棋盘格，右眼优先

' icl '

交错列，左眼优先

' icr '

交错列，右眼优先

' HDMI '

HDMI帧包

默认值为“ arcd ”。

* [例子](#Examples-163)

#### 39.240.1 示例

* 将输入视频从并排平行转换为立体黄色/蓝色 dubois：
* stereo3d=sbsl:aybd
* 将输入视频从上方（左眼在上方，右眼在下方）转换为并排交叉眼。
* stereo3d=abl:sbsr

### 39.241 流选择，流选择

选择视频或音频流。

过滤器接受以下选项：

输入

设置输入数量。默认值为 2。

地图

设置输入索引以重新映射到输出。

* [命令](#Commands-144)
* [例子](#Examples-164)

#### 39.241.1 命令

streamselect和astreamselect过滤器支持以下命令：

地图

设置输入索引以重新映射到输出。

#### 39.241.2 例子

* 选择前 5 秒的第一个流和其余时间的第二个流：
* sendcmd='5.0 streamselect map 1',streamselect=inputs=2:map=0
* 同上，但对于音频：
* asendcmd='5.0 astreamselect map 1',astreamselect=inputs=2:map=0

### 39.242 字幕

使用 libass 库在输入视频上绘制字幕。

--enable-libass配置 FFmpeg 。此过滤器还需要使用 libavcodec 和 libavformat 进行构建，以将传递的字幕文件转换为 ASS（Advanced Substation Alpha）字幕格式。

过滤器接受以下选项：

文件名，f

设置要读取的字幕文件的文件名。必须指定。

原始尺寸

指定原始视频的大小，即为其编写 ASS 文件的视频。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。由于 ASS 纵横比算法的设计错误，如果纵横比已更改，则必须正确缩放字体。

字体目录

设置包含过滤器可以使用的字体的目录路径。除了字体提供者使用的任何字体之外，还将使用这些字体。

α

处理 alpha 通道，默认情况下 alpha 通道保持不变。

夏朗

设置字幕输入字符编码。仅字幕过滤器。只有在不是 UTF-8 时才有用。

stream\_index, si

设置字幕流索引。仅字幕过滤器。

force\_style

覆盖字幕的默认样式或脚本信息参数。它接受包含 ASS 样式格式KEY=VALUE 对的字符串，以“,”分隔。

如果未指定第一个键，则假定第一个值指定文件名。

例如，要在输入视频之上渲染文件sub.srt ，请使用以下命令：

字幕=sub.srt

这相当于：

字幕=文件名=sub.srt

要从文件video.mkv渲染默认字幕流，请使用：

字幕=video.mkv

要从该文件渲染第二个字幕流，请使用：

字幕=video.mkv:si=1

要使来自sub.srt的字幕流以 80% 透明的蓝色DejaVu Serif显示，请使用：

subtitles=sub.srt:force\_style='Fontname=DejaVu Serif,PrimaryColour=&HCCFF0000'

### 39.243 super2xsai

使用 Super2xSaI（缩放和插值）像素艺术缩放算法将输入缩放 2 倍并平滑。

用于在不降低清晰度的情况下放大像素艺术图像。

### 39.244 交换协议

交换视频中的两个矩形对象。

此过滤器接受以下选项：

w

设置对象宽度。

H

设置对象高度。

x1

设置第一个矩形 x 坐标。

y1

设置 1st rect y 坐标。

x2

设置第二个矩形 x 坐标。

y2

设置第二个矩形 y 坐标。

对每一帧计算一次所有表达式。

所有选项都是包含以下常量的表达式：

w

H

输入的宽度和高度。

A

w / h相同

特区

输入样本纵横比

达尔

输入显示宽高比，同( w / h ) \* sar

n

输入帧的编号，从0开始。

吨

以秒表示的时间戳。如果输入时间戳未知，则为 NAN。

位置

输入帧在文件中的位置，如果未知则为 NAN

* [命令](#Commands-145)

#### 39.244.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.245 swapuv

交换 U 和 V 平面。

### 39.246 汤混合

混合连续的视频帧。

见[混合](#blend)

### 39.247 电视电影

对视频应用电视电影过程。

此过滤器接受以下选项：

first\_field

'顶， '

顶场优先

'底部，b '

bottom field first 默认值为top 。

图案

一串数字，代表您希望应用的下拉模式。默认值为23 。

一些典型的模式：

NTSC 输出 (30i)：

27.5p：32222

24p: 23（经典）

24p：2332（首选）

20 便士：33

18页：334

16页：3444

PAL 输出 (25i)：

27.5p：12222

24p：222222222223（“欧元下拉”）

16.67p：33

16页：33333334

### 39.248 这个图

计算并绘制输入视频随时间变化的颜色分布直方图。

与仅显示特定时间单个输入帧的[直方图的直方图视频过滤器](#histogram)不同，此过滤器还显示过去由width选项定义的帧数的直方图。

计算出的直方图表示图像中的颜色分量分布。

过滤器接受以下选项：

宽度，w

设置单色分量输出的宽度。默认值为0 。值为0表示宽度将从输入视频中选取。这也设置了要保留的传递直方图的数量。允许的范围是 [0, 8192]。

显示模式，d

设置显示模式。它接受以下值：

'堆栈'

每个颜色分量图都放在彼此下面。

'游行'

每个颜色分量图并排放置。

'覆盖'

parade中的信息相同，除了表示颜色分量的图表直接叠加在彼此之上。

默认是堆栈。

levels\_mode，米

设置模式。可以是线性的，也可以是对数的。默认是线性的。

组件，c

设置要显示的颜色分量。默认值为7 。

b 透明度, b

设置背景不透明度。默认值为0.9 。

信封，e

显示信封。默认是禁用的。

ecolor, ec

设置信封颜色。默认为黄金。

滑动

设置幻灯片模式。

slide 的可用值是：

'框架'

到达右边界时绘制新框架。

'替换'

用新列替换旧列。

'滚动'

从右向左滚动。

'滚动'

从左向右滚动。

'图片'

画单张图。

默认是替换。

### 39.249 门槛

对视频流应用阈值效应。

此过滤器需要四个视频流来执行阈值处理。第一个流是我们正在过滤的流。第二个流保存阈值，第三个流保存最小值，最后第四个流保存最大值。

过滤器接受以下选项：

飞机

设置要处理的平面，未处理的平面将被复制。默认值 0xf，将处理所有平面。

例如，如果第一流像素的分量值小于来自第二阈值流的像素分量的阈值，则将拾取第三流值，否则将拾取第四流像素分量值。

使用彩色源过滤器可以执行各种类型的阈值处理：

* [命令](#Commands-146)
* [例子](#Examples-165)

#### 39.249.1 命令

此过滤器支持所有选项作为[命令](#commands)。

#### 39.249.2 示例

* 二进制阈值，使用灰色作为阈值：
* ffmpeg -i 320x240.avi -f lavfi -i color=gray -f lavfi -i color=black -f lavfi -i color=white -lavfi 阈值输出.avi
* 倒置二进制阈值，使用灰色作为阈值：
* ffmpeg -i 320x240.avi -f lavfi -i color=gray -f lavfi -i color=white -f lavfi -i color=black -lavfi 阈值输出.avi
* 截断二进制阈值，使用灰色作为阈值：
* ffmpeg -i 320x240.avi -f lavfi -i color=gray -i 320x240.avi -f lavfi -i color=gray -lavfi 阈值输出.avi
* 阈值为零，使用灰色作为阈值：
* ffmpeg -i 320x240.avi -f lavfi -i color=gray -f lavfi -i color=white -i 320x240.avi -lavfi 阈值输出.avi
* 将阈值倒置为零，使用灰色作为阈值：
* ffmpeg -i 320x240.avi -f lavfi -i color=gray -i 320x240.avi -f lavfi -i color=white -lavfi 阈值输出.avi

### 39.250 缩略图

在给定的连续帧序列中选择最具代表性的帧。

过滤器接受以下选项：

n

设置要分析的帧批量大小；在一组n帧中，过滤器将选择其中一个，然后处理下一批n帧，直到结束。默认值为100 。

日志

设置日志级别以显示拾取的帧统计信息。默认是信息。

由于过滤器跟踪整个帧序列，较大的n值将导致较高的内存使用，因此不建议使用较高的值。

* [例子](#Examples-166)

#### 39.250.1 示例

* 每50帧提取一张图片：
* 缩略图=50
* ffmpeg创建缩略图的完整示例：
* ffmpeg -i in.avi -vf thumbnail,scale=300:200 -frames:v 1 out.png

### 39.251瓦

将几个连续的帧平铺在一起。

[until](#untile)过滤器可以做相反的事情。

过滤器接受以下选项：

布局

COLUMNSxROWS 的形式设置网格大小。范围最多为 UINT\_MAX 个单元格。默认为6x5 。

nb\_帧

设置要在给定区域中渲染的最大帧数。它必须小于或等于w x h 。默认值为0 ，表示将使用所有区域。

利润

以像素为单位设置外边框边距。范围是 0 到 1024。默认值为0 。

填充

设置内边框厚度（即帧之间的像素数）。有关更高级的填充选项（例如具有不同的边缘值），请参阅填充视频过滤器。范围是 0 到 1024。默认值为0 。

颜色

指定未使用区域的颜色。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。颜色的默认值为“黑色”。

重叠

将多个连续帧平铺在一起时，设置要重叠的帧数。该值必须介于0和nb\_frames - 1之间。默认为0 。

初始填充

在显示第一个输出帧之前将帧数设置为最初为空。这控制了多久可以获得第一个输出帧。该值必须介于0和nb\_frames - 1之间。默认为0 。

* [例子](#Examples-167)

#### 39.251.1 例子

* 在影片中生成所有关键帧 ( -skip\_frame nokey )的 8x8 PNG 图块：
* ffmpeg -skip\_frame nokey -i file.avi -vf 'scale=128:72,tile=8x8' -an -vsync 0 关键帧%03d.png

-vsync 0是防止ffmpeg复制每个输出帧以适应最初检测到的帧速率所必需的。

* 3x2帧的区域中显示5 张图片，它们之间有7 个像素，初始边距为2 个像素，使用混合平面和命名选项：
* tile=3x2:nb\_frames=5:padding=7:margin=2

### 39.252 交织

执行各种类型的时间场交错。

帧从 1 开始计数，因此第一个输入帧被视为奇数。

过滤器接受以下选项：

模式

指定隔行扫描的模式。此选项也可以单独指定为一个值。有关此选项的值列表，请参见下文。

可用值是：

'合并，0 '

将奇数帧移动到上场，甚至移动到下场，以半帧速率生成双倍高度帧。

------> 时间

输入：

框架 1 框架 2 框架 3 框架 4

11111 22222 33333 44444

11111 22222 33333 44444

11111 22222 33333 44444

11111 22222 33333 44444

输出：

11111 33333

22222 44444

11111 33333

22222 44444

11111 33333

22222 44444

11111 33333

22222 44444

' drop\_even, 1 '

只输出奇数帧，偶数帧被丢弃，以半帧率生成高度不变的帧。

------> 时间

输入：

框架 1 框架 2 框架 3 框架 4

11111 22222 33333 44444

11111 22222 33333 44444

11111 22222 33333 44444

11111 22222 33333 44444

输出：

11111 33333

11111 33333

11111 33333

11111 33333

' drop\_odd, 2 '

只输出偶数帧，奇数帧被丢弃，以半帧率生成高度不变的帧。

------> 时间

输入：

框架 1 框架 2 框架 3 框架 4

11111 22222 33333 44444

11111 22222 33333 44444

11111 22222 33333 44444

11111 22222 33333 44444

输出：

22222 44444

22222 44444

22222 44444

22222 44444

'垫，3 '

将每个帧扩展到全高，但用黑色填充交替线，以相同的输入帧速率生成具有双倍高度的帧。

------> 时间

输入：

框架 1 框架 2 框架 3 框架 4

11111 22222 33333 44444

11111 22222 33333 44444

11111 22222 33333 44444

11111 22222 33333 44444

输出：

11111 ..... 33333 .....

..... 22222 ..... 44444

11111 ..... 33333 .....

..... 22222 ..... 44444

11111 ..... 33333 .....

..... 22222 ..... 44444

11111 ..... 33333 .....

..... 22222 ..... 44444

' interleave\_top, 4 '

将奇数帧的上场与偶数帧的下场交错，以半帧速率生成高度不变的帧。

------> 时间

输入：

框架 1 框架 2 框架 3 框架 4

11111<- 22222 33333<- 44444

11111 22222<- 33333 44444<-

11111<- 22222 33333<- 44444

11111 22222<- 33333 44444<-

输出：

11111 33333

22222 44444

11111 33333

22222 44444

' interleave\_bottom, 5 '

将奇数帧的下场与偶数帧的上场交错，以半帧速率生成高度不变的帧。

------> 时间

输入：

框架 1 框架 2 框架 3 框架 4

11111 22222<- 33333 44444<-

11111<- 22222 33333<- 44444

11111 22222<- 33333 44444<-

11111<- 22222 33333<- 44444

输出：

22222 44444

11111 33333

22222 44444

11111 33333

'隔行扫描2，6 '

双倍帧率，高度不变。插入帧，每个帧包含来自前一个输入帧的第二个时间场和来自下一个输入帧的第一个时间场。此模式依赖于 top\_field\_first 标志。对于没有场同步的隔行扫描视频显示很有用。

------> 时间

输入：

框架 1 框架 2 框架 3 框架 4

11111 22222 33333 44444

11111 22222 33333 44444

11111 22222 33333 44444

11111 22222 33333 44444

输出：

11111 22222 22222 33333 33333 44444 44444

11111 11111 22222 22222 33333 33333 44444

11111 22222 22222 33333 33333 44444 44444

11111 11111 22222 22222 33333 33333 44444

' mergex2, 7 '

将奇数帧移到上场，偶数帧移到下场，以相同的帧速率生成双倍高度的帧。

------> 时间

输入：

框架 1 框架 2 框架 3 框架 4

11111 22222 33333 44444

11111 22222 33333 44444

11111 22222 33333 44444

11111 22222 33333 44444

输出：

11111 33333 33333 55555

22222 22222 44444 44444

11111 33333 33333 55555

22222 22222 44444 44444

11111 33333 33333 55555

22222 22222 44444 44444

11111 33333 33333 55555

22222 22222 44444 44444

数字值已弃用，但出于向后兼容性原因而被接受。

默认模式是合并。

旗帜

指定影响过滤过程的标志。

标志的可用值为：

低通滤波器，vlpf

在滤波器中启用线性垂直低通滤波。从包含高频垂直细节的逐行源创建隔行扫描目标时，需要垂直低通滤波。过滤将减少隔行扫描“twitter”和波纹图案。

复杂过滤器，cvlpf

启用复杂的垂直低通滤波。这将略微减少隔行扫描“twitter”和莫尔图案，但会更好地保留细节和主观清晰度印象。

绕过\_il

绕过已经交错的帧，只调整帧速率。

只能为模式启用垂直低通滤波和绕过已经交错的帧 interleave\_top和interleave\_bottom 。

### 39.253 tmedian

从几个连续的输入视频帧中选取中值像素。

过滤器接受以下选项：

半径

设置中值滤波器的半径。默认值为 1。允许的范围为 1 到 127。

飞机

设置要过滤的平面。默认值为15 ，通过它处理所有平面。

百分位数

设置中位数百分位数。默认值为0.5 。默认值0.5将始终选择中值，而0将选择最小值， 1将选择最大值。

* [命令](#Commands-147)

#### 39.253.1 命令

此过滤器支持以上所有选项作为[命令](#commands)，不包括选项radius 。

### 39.254 tm均衡器

应用时间中途视频均衡效果。

中途视频均衡调整一系列视频帧以具有相同的直方图，同时尽可能保持其动态。它对于匹配来自视频帧序列的曝光等非常有用。

此过滤器接受以下选项：

半径

设置过滤半径。默认值为5 。允许的范围是 1 到 127。

西格玛

设置过滤西格玛。默认值为0.5 。这控制过滤的强度。将此选项设置为 0 实际上没有任何作用。

飞机

设置要处理的平面。默认为15 ，这是所有可用的平面。

### 39.255 混音

混合连续的视频帧。

接受的选项的描述如下。

框架

要混合的连续帧数。如果未指定，则默认为 3。

权重

指定每个输入视频帧的权重。每个权重由空格分隔。如果权重数小于帧数，最后指定的权重将用于所有剩余的未设置权重。

规模

指定比例，如果设置它，它将乘以每个权重乘以像素值的总和，以给出最终目标像素值。默认情况下，比例会自动缩放为权重之和。

飞机

设置要过滤的平面。默认为全部。允许的范围是从 0 到 15。

* [例子](#Examples-168)
* [命令](#Commands-148)

#### 39.255.1 示例

* 平均 7 个连续帧：
* tmix=frames=7:weights="1 1 1 1 1 1 1"
* 应用简单的时间卷积：
* tmix=frames=3:weights="-1 3 -1"
* 与上面类似，但仅显示时间差异：
* tmix=frames=3:weights="-1 2 -1":scale=1

#### 39.255.2 命令

该过滤器支持以下命令：

权重

规模

飞机

语法与同名选项相同。

### 39.256 色调映射

来自不同动态范围的色调映射颜色。

此过滤器需要单精度浮点数据，因为它需要对超出范围的值进行操作（并可以输出）。需要另一个过滤器（例如[zscale ）将生成的帧转换为可用格式。](#zscale)

实施的色调映射算法仅适用于线性光，因此输入数据应事先线性化（并可能正确标记）。

ffmpeg -i 输入 -vf zscale=transfer=linear,tonemap=clip,zscale=transfer=bt709,format=yuv420p 输出

* [选项](#Options-96)

#### 39.256.1 选项

过滤器接受以下选项。

色调图

设置要使用的色调映射算法。

可能的值是：

没有任何

不要应用任何色调贴图，只对过亮的像素进行去饱和处理。

夹子

硬剪辑任何超出范围的值。使用它可以使范围内的值达到完美的颜色精度，同时扭曲超出范围的值。

线性的

将整个参考色域扩展到显示器的线性倍数。

伽马

在色调曲线之间拟合对数转换。

莱茵哈德

使用非线性对比度通过简单曲线保持整体图像亮度，这会导致细节变平并降低颜色准确性。

有能力的

reinhard更好地保留黑暗和明亮的细节，代价是让所有东西都稍微变暗。当保留细节比颜色和亮度准确性更重要时使用它。

莫比乌斯

平滑映射超出范围的值，同时尽可能保留范围内材料的对比度和颜色。当颜色准确性比保留细节更重要时使用它。

默认为无。

参数

调整色调映射算法。

这会影响以下算法：

没有任何

忽略。

线性的

指定拉伸时要使用的比例因子。默认为 1.0。

伽马

指定函数的指数。默认为 1.8。

夹子

指定一个额外的线性系数以在削波前乘以信号。默认为 1.0。

莱茵哈德

指定显示峰值处的局部对比度系数。默认为 0.5，这意味着色域内值的亮度大约是裁剪时的一半。

有能力的

忽略。

莫比乌斯

指定从线性到莫比乌斯变换的过渡点。此点以下的每个值都保证按 1:1 映射。值越高，结果越准确，代价是丢失明亮的细节。默认为 0.3，由于陡峭的初始斜率仍然相当准确地保留了范围内的颜色。

去饱和

对超过此亮度级别的高光应用去饱和度。参数越高，保留的颜色信息越多。此设置通过（平滑地）变成白色来帮助防止超高光的颜色不自然地过曝。这使图像感觉更自然，但代价是减少了有关超出范围的颜色的信息。

默认值 2.0 有点保守，主要适用于天空或阳光直射的表面。设置为 0.0 会禁用此选项。

此选项仅在输入帧具有受支持的颜色标签时才有效。

顶峰

用这个值覆盖信号/标称/参考峰值。当显示元数据中嵌入的峰值信息不可靠或从较低范围到较高范围的色调映射时很有用。

### 39.257 吨

临时填充视频帧。

过滤器接受以下选项：

开始

指定输入视频流之前的延迟帧数。默认为 0。

停止

指定输入视频流后的填充帧数。设置为 -1 无限期填充。默认为 0。

启动模式

设置添加到流开头的帧类型。可以是add或clone 。添加纯色框架。克隆帧是第一帧的克隆。默认是添加。

停止模式

设置添加到流末尾的帧类型。可以是add或clone 。添加纯色框架。克隆帧是最后一帧的克隆。默认是添加。

开始持续时间、停止持续时间

指定启动/停止延迟的持续时间。有关可接受的语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)这些选项覆盖start和stop 。默认为 0。

颜色

指定填充区域的颜色。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。

颜色的默认值为“黑色”。

### 39.258转置

转置输入视频中的列和行，并可选择翻转它。

它接受以下参数：

目录

指定移调方向。

可以假定以下值：

' 0, 4, cclock\_flip '

逆时针旋转90度，垂直翻转（默认），即：

LL

. . -> 。 .

lrrr

' 1, 5, 时钟'

顺时针旋转90度，即：

左下角

. . -> 。 .

lrrR

' 2, 6, 时钟'

逆时针旋转90度，即：

左下角

. . -> 。 .

lr ll

' 3, 7, clock\_flip '

顺时针旋转90度垂直翻转，即：

左下角右下角

. . -> 。 .

lr lL

对于 4-7 之间的值，仅当输入视频几何图形是纵向而非横向时才进行转置。这些值已弃用，应该改用passthrough选项。

数值已被弃用，应放弃以支持符号常量。

直通

如果输入几何与指定值指定的几何相匹配，则不要应用转置。它接受以下值：

'无'

始终应用换位。

'肖像'

保留纵向几何图形（当height >= width时）。

'景观'

保留景观几何（当width >= height时）。

默认值为无。

例如顺时针旋转 90 度并保留纵向布局：

转置=目录=1:直通=纵向

上面的命令也可以指定为：

转置=1：纵向

### 39.259 transpose\_npp

转置输入视频中的列和行，并可选择翻转它。有关更深入的示例，请参阅[转置](#transpose)视频过滤器，它共享大部分相同的选项。

它接受以下参数：

目录

指定移调方向。

可以假定以下值：

' cclock\_flip '

逆时针旋转90度，垂直翻转。 （默认）

'时钟'

顺时针旋转 90 度。

'时钟'

逆时针旋转 90 度。

'时钟翻转'

顺时针旋转90度，垂直翻转。

直通

如果输入几何与指定值指定的几何相匹配，则不要应用转置。它接受以下值：

'无'

始终应用换位。 （默认）

'肖像'

保留纵向几何图形（当height >= width时）。

'景观'

保留景观几何（当width >= height时）。

### 39.260 修剪

修剪输入，使输出包含输入的一个连续子部分。

它接受以下参数：

开始

指定保留部分的开始时间，即具有时间戳开始的帧将是输出中的第一帧。

结尾

结束的帧之前的帧将是输出中的最后一帧。

开始点

start相同，除了此选项以时基单位而不是秒为单位设置开始时间戳。

结束点

end相同，除了此选项以时基单位而不是秒为单位设置结束时间戳。

期间

输出的最大持续时间（以秒为单位）。

起始帧

应该传递给输出的第一帧的编号。

结束帧

应该丢弃的第一帧的编号。

start 、 end和duration表示为持续时间规范；有关可接受的语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)

请注意，前两组开始/结束选项和持续时间选项查看帧时间戳，而 \_frame 变体仅计算通过过滤器的帧。另请注意，此过滤器不会修改时间戳。如果您希望输出时间戳从零开始，请在 trim 过滤器之后插入一个 setpts 过滤器。

如果设置了多个开始或结束选项，则此过滤器会尝试贪婪并保留所有匹配至少一个指定约束的帧。要一次仅保留与所有约束匹配的部分，请链接多个修剪过滤器。

默认值是保留所有输入。因此可以设置例如仅结束值以在指定时间之前保留所有内容。

例子：

* 删除除第二分钟输入之外的所有内容：
* ffmpeg -i 输入 -vf 修剪=60:120
* 只保留第一秒：
* ffmpeg -i 输入 -vf 修剪=持续时间=1

### 39.261 未预乘

使用第二个流的第一个平面作为 alpha，将 alpha unpremultiply 效果应用于输入视频流。

两个流必须具有相同的尺寸和相同的像素格式。

过滤器接受以下选项：

飞机

设置要处理的平面，未处理的平面将被复制。默认值 0xf，将处理所有平面。

如果格式有 1 或 2 个分量，则亮度为位 0。如果格式有 3 或 4 个分量：对于 RGB 格式，位 0 为绿色，位 1 为蓝色，位 2 为红色；对于 YUV 格式，第 0 位是亮度，第 1 位是色度 U，第 2 位是色度 V。如果存在，alpha 通道始终是最后一位。

到位

不需要第二个输入进行处理，而是使用输入流中的 alpha 平面。

### 39.262 锐化

锐化或模糊输入视频。

它接受以下参数：

luma\_msize\_x, lx

设置亮度矩阵水平大小。它必须是 3 到 23 之间的奇数整数。默认值为 5。

luma\_msize\_y, ly

设置亮度矩阵垂直大小。它必须是 3 到 23 之间的奇数整数。默认值为 5。

luma\_amount, la

设置亮度效果强度。它必须是一个浮点数，合理的值在-1.5到1.5之间。

负值将模糊输入视频，而正值将锐化它，零值将禁用效果。

默认值为 1.0。

chroma\_msize\_x, cx

设置色度矩阵水平大小。它必须是 3 到 23 之间的奇数整数。默认值为 5。

chroma\_msize\_y, cy

设置色度矩阵垂直大小。它必须是 3 到 23 之间的奇数整数。默认值为 5。

色度量，ca

设置色度效果强度。它必须是一个浮点数，合理的值在-1.5到1.5之间。

负值将模糊输入视频，而正值将锐化它，零值将禁用效果。

默认值为 0.0。

alpha\_msize\_x, 斧头

设置 alpha 矩阵水平大小。它必须是 3 到 23 之间的奇数整数。默认值为 5。

alpha\_msize\_y，哎呀

设置 alpha 矩阵垂直大小。它必须是 3 到 23 之间的奇数整数。默认值为 5。

alpha\_amount, aa

设置 alpha 效果强度。它必须是一个浮点数，合理的值在-1.5到1.5之间。

负值将模糊输入视频，而正值将锐化它，零值将禁用效果。

默认值为 0.0。

所有参数都是可选的，默认为字符串 '5:5:1.0:5:5:0.0' 的等价物。

* [例子](#Examples-169)

#### 39.262.1 示例

* 应用强大的亮度锐化效果：
* 锐化=luma\_msize\_x=7:luma\_msize\_y=7:luma\_amount=2.5
* 应用亮度和色度参数的强烈模糊：
* 不锐化=7:7:-2:7:7:-2

### 39.263 直到

将由平铺图像组成的视频分解为单个图像。

输出视频的帧率是输入视频的帧率乘以图块数。

[tile](#tile)相反的操作。

过滤器接受以下选项：

布局

设置网格大小（即行数和列数）。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。

* [例子](#Examples-170)

#### 39.263.1 示例

* 从由 25 帧垂直堆叠的静止图像文件生成 1 秒视频，就像模拟胶片卷轴一样：
* ffmpeg -r 1 -i image.jpg -vf untile=1x25 电影.mkv

### 39.264 美元

质量级别8的情况下- 全部）移位压缩和解压缩图像并平均结果。

这与 spp 行为的不同之处在于 uspp 实际上使用 libavcodec Snow 对每个案例进行编码和解码，而 spp 使用类似于 MJPEG 的简化内部仅 8x8 DCT。

此过滤器仅在 ffmpeg 版本 4.4 或更早版本中可用。

过滤器接受以下选项：

质量

设置质量。此选项定义平均级别数。它接受 0-8 范围内的整数。如果设置为0 ，过滤器将无效。值为8意味着更高的质量。对于该值的每个增量，速度下降大约 2 倍。默认值为3 。

qp

强制一个恒定的量化参数。如果未设置，过滤器将使用视频流中的 QP（如果可用）。

### 39.265 v360

在各种格式之间转换 360 度视频。

过滤器接受以下选项：

输入

输出

设置输入/输出视频的格式。

可用格式：

' e '

'等价'

等距柱状投影。

' c3x2 '

' c6x1 '

' c1x6 '

具有 3x2/6x1/1x6 布局的立方体贴图。

格式特定选项：

in\_pad

外接垫

设置输入/输出立方体贴图的填充比例。以小数表示的值。

示例值：

' 0 '

没有填充。

' 0.01 '

1% 的面是填充。例如，分辨率为 1920x1280 的人脸尺寸为 640x640，填充距离为每边 3 像素。 （640 \* 0.01 = 6 像素）

默认值为**“ 0 ”** 。最大值为**“ 0.1 ”** 。

脚垫

fout\_pad

为输入/输出立方体贴图设置固定填充。以像素为单位的值。

默认值为**“ 0 ”** 。如果大于零，它将覆盖其他填充选项。

inforder

out\_forder

设置输入/输出立方体贴图的面顺序。为每个位置选择一个方向。

方向指定：

' r '

正确的

'我'

左边

'你'

向上

' d '

向下

' f '

向前

' b '

后退

默认值为**“ rludfb ”** 。

开头

out\_frot

为输入/输出立方体贴图设置面的旋转。为每个位置选择一个角度。

角度名称：

' 0 '

顺时针0度

' 1 '

顺时针90度

' 2 '

顺时针180度

' 3 '

顺时针270度

默认值为**“ 000000 ”** 。

'每个'

等角立方体贴图。

'平'

'日行'

'直线'

常规视频。

格式特定选项：

h\_fov

v\_fov

d\_fov

设置输出水平/垂直/对角线视野。以度为单位的值。

如果设置了对角线视野，它将覆盖水平和垂直视野。

ih\_fov

iv\_fov

id\_fov

设置输入水平/垂直/对角线视野。以度为单位的值。

如果设置了对角线视野，它将覆盖水平和垂直视野。

'鱼眼'

双鱼眼。

格式特定选项：

h\_fov

v\_fov

d\_fov

设置输出水平/垂直/对角线视野。以度为单位的值。

如果设置了对角线视野，它将覆盖水平和垂直视野。

ih\_fov

iv\_fov

id\_fov

设置输入水平/垂直/对角线视野。以度为单位的值。

如果设置了对角线视野，它将覆盖水平和垂直视野。

'桶'

'脸书'

'桶裂'

Facebook 的 360 格式。

' sg '

立体格式。

格式特定选项：

h\_fov

v\_fov

d\_fov

设置输出水平/垂直/对角线视野。以度为单位的值。

如果设置了对角线视野，它将覆盖水平和垂直视野。

ih\_fov

iv\_fov

id\_fov

设置输入水平/垂直/对角线视野。以度为单位的值。

如果设置了对角线视野，它将覆盖水平和垂直视野。

'墨卡托'

墨卡托格式。

'球'

球格式，向后方产生明显的变形。

'锤子'

Hammer-Aitoff 地图投影格式。

'正弦'

正弦地图投影格式。

'鱼眼'

鱼眼投影。

格式特定选项：

h\_fov

v\_fov

d\_fov

设置输出水平/垂直/对角线视野。以度为单位的值。

如果设置了对角线视野，它将覆盖水平和垂直视野。

ih\_fov

iv\_fov

id\_fov

设置输入水平/垂直/对角线视野。以度为单位的值。

如果设置了对角线视野，它将覆盖水平和垂直视野。

'帕尼尼'

帕尼尼投影。

格式特定选项：

h\_fov

设置输出pannini参数。

ih\_fov

设置输入pannini参数。

'圆柱'

圆柱投影。

格式特定选项：

h\_fov

v\_fov

d\_fov

设置输出水平/垂直/对角线视野。以度为单位的值。

如果设置了对角线视野，它将覆盖水平和垂直视野。

ih\_fov

iv\_fov

id\_fov

设置输入水平/垂直/对角线视野。以度为单位的值。

如果设置了对角线视野，它将覆盖水平和垂直视野。

'透视'

透视投影。 *（仅输出）*

格式特定选项：

v\_fov

设置透视参数。

'四面体'

四面体投影。

'茶匙'

截断方形金字塔投影。

‘他’

' hequirect '

半等角投影。

'等固'

等固体格式。

格式特定选项：

h\_fov

v\_fov

d\_fov

设置输出水平/垂直/对角线视野。以度为单位的值。

如果设置了对角线视野，它将覆盖水平和垂直视野。

ih\_fov

iv\_fov

id\_fov

设置输入水平/垂直/对角线视野。以度为单位的值。

如果设置了对角线视野，它将覆盖水平和垂直视野。

'奥格'

正字法格式。

格式特定选项：

h\_fov

v\_fov

d\_fov

设置输出水平/垂直/对角线视野。以度为单位的值。

如果设置了对角线视野，它将覆盖水平和垂直视野。

ih\_fov

iv\_fov

id\_fov

设置输入水平/垂直/对角线视野。以度为单位的值。

如果设置了对角线视野，它将覆盖水平和垂直视野。

'八面体'

八面体投影。

'圆柱'

圆柱等积投影。

插话

设置插值方法。  
*注意：更复杂的插值方法需要更多的内存才能运行。*

可用方法：

'附近'

'最近的'

最近的邻居。

'线'

'线性'

双线性插值。

'拉格朗日9 '

拉格朗日9插值。

'立方体'

'立方体'

双三次插值。

'兰斯'

'兰佐斯'

Lanczos 插值法。

' sp16 '

'样条16 '

Spline16 插值。

'高斯'

'高斯'

高斯插值。

'米切尔'

米切尔插值法。

默认值为**“行”** 。

w

H

设置输出视频分辨率。

默认分辨率取决于格式。

非立体声

输出立体声

设置输入/输出立体声格式。

' 2d '

二维单声道

'某人'

并排

'结核病'

顶底

输入和输出格式的默认值为**“ 2d ” 。**

偏航

沥青

卷

设置输出视频的旋转。以度为单位的值。

订单

设置输出视频的旋转顺序。为每个位置选择一个项目。

'是，是'

偏航

' p, P '

沥青

' r, R '

卷

默认值为**' ypr '** 。

h\_flip

v\_flip

d\_flip

水平（左右交换）/垂直（上下交换）/深度（前后交换）翻转输出视频。布尔值。

ih\_flip

iv\_flip

设置输入视频是否水平/垂直翻转。布尔值。

in\_trans

设置输入视频是否被转置。布尔值，默认禁用。

out\_trans

设置输出视频是否需要转置。布尔值，默认禁用。

h\_offset

v\_offset

设置输出水平/垂直离轴偏移量。默认设置为 0。允许的范围是从 -1 到 1。

alpha\_mask

通过将所有未映射的像素标记为完全透明，在 alpha 平面中构建蒙版。布尔值，默认禁用。

重置\_rot

重置输出视频的旋转。布尔值，默认禁用。

* [例子](#Examples-171)
* [命令](#Commands-149)

#### 39.265.1 示例

* 使用双三次插值将 equirectangular 视频转换为具有 3x2 布局和 1% 填充的立方体贴图：
* ffmpeg -i input.mkv -vf v360=e:c3x2:cubic:out\_pad=0.01 输出.mkv
* 提取等角立方体贴图的后视图：
* ffmpeg -i input.mkv -vf v360=eac:flat:yaw=180 输出.mkv
* 将并排立体格式的转置和水平翻转的等角立方体贴图转换为等矩形上下立体格式：
* v360=eac:equirect:in\_stereo=sbs:in\_trans=1:ih\_flip=1:out\_stereo=tb

#### 39.265.2 命令

此过滤器支持上述选项的子集作为[命令](#commands)。

### 39.266 模糊降噪器

应用基于小波的降噪器。

它使用 Cohen-Daubechies-Feauveau 9/7 将每个帧从视频输入转换到小波域。然后它对获得的系数应用一些过滤。之后进行逆小波变换。由于小波的特性，它应该给出一个很好的平滑结果，并减少噪音，而不会使图片特征模糊。

此过滤器接受以下选项：

临界点

过滤强度。越高，视频过滤得越多。在视频看起来过度过滤之前，硬阈值可以使用比软阈值更高的阈值。默认值为 2。

方法

过滤器将使用的过滤方法。

它接受以下值：

'硬'

所有低于阈值的值都将归零。

'软'

所有低于阈值的值都将归零。以上所有值都将减少阈值。

'绞喉'

缩放或取消系数 -（更多）软阈值和（更少）硬阈值之间的中介。

默认是绞喉。

步数

小波将分解图片的次数。图片无法分解到特定点之外（对于 640x480 帧通常为 8 - 2^9 = 512 > 480）。有效值为 1 到 32 之间的整数。默认值为 6。

百分

部分完全去噪（有限系数收缩），从 0 到 100。默认值为 85。

飞机

要处理的平面列表。默认情况下处理所有平面。

类型

过滤器将使用的阈值类型。

它接受以下值：

'通用'

所有分解使用的阈值相同。

'贝叶斯'

使用的阈值还取决于每个分解系数。

默认是通用的。

### 39.267 可变模糊

通过使用第二个视频流设置模糊半径来应用可变模糊过滤器。第二个流必须具有相同的尺寸。

此过滤器接受以下选项：

最小值

设置最小允许半径。允许的范围是 0 到 254。默认值为 0。

最大\_r

设置最大允许半径。允许的范围是 1 到 255。默认值为 8。

飞机

设置要处理的平面。默认情况下，全部使用。

varblur滤镜还支持[帧同步](#framesync)选项。

* [命令](#Commands-150)

#### 39.267.1 命令

此过滤器支持上述所有选项作为[命令](#commands)。

### 39.268 矢量示波器

在二维图形（称为矢量示波器）中显示 2 个颜色分量值。

此过滤器接受以下选项：

模式，米

设置矢量示波器模式。

它接受以下值：

'灰色'

'色调'

灰度值显示在图形上，更高的亮度意味着更多的像素在图形位置上具有相同的分量颜色值。这是默认模式。

'颜色'

灰色值显示在图表上。视频帧中不存在的周围像素值以由选项x和y设置的 2 个颜色分量的渐变绘制。第三个颜色分量是静态的。

'颜色2 '

视频帧中存在的实际颜色分量值显示在图形上。

'颜色3 '

与 color2 类似，但图表上相同值x和y的频率更高会增加另一个颜色分量的值，默认值为x和y的亮度。

'颜色4 '

视频帧中存在的实际颜色显示在图表上。如果两种不同的颜色映射到图形上的相同位置，则选择图形中不存在的分量值较高的颜色。

'颜色5 '

灰色值显示在图表上。类似于颜色，但具有从径向渐变中选取的第三种颜色分量。

X

设置将在 X 轴上表示的颜色分量。默认为1 。

是

设置将在 Y 轴上表示的颜色分量。默认值为2 。

强度，我

设置强度，由以下模式使用：灰色、颜色、颜色 3 和颜色 5，用于增加表示图中 (X, Y) 位置频率的颜色分量的亮度。

信封，e

'无'

没有信封，这是默认设置。

'即时'

即时信封，即使是最暗的单个像素也会被清楚地突出显示。

'峰'

随着时间的推移保持图表中显示的最大值和最小值。这样您仍然可以发现超出范围的值而无需不断查看矢量示波器。

'巅峰+瞬间'

峰值和即时包络结合在一起。

标线，g

设置绘制什么样的格线。

'无'

'绿色'

'颜色'

'反转'

不透明度，o

设置格线不透明度。

标志，f

设置标线标志。

'白色'

绘制白点的刻度线。

'黑色'

为黑点画格线。

'名字'

绘制颜色点的简称。

b 透明度, b

设置背景不透明度。

l阈值，l

为 X 或 Y 轴上未表示的颜色分量设置低阈值。低于此值的值将被忽略。默认值为 0。请注意，此值乘以一个像素组件可以具有的实际最大可能值。所以对于 8 位输入和 0.1 的低阈值，实际阈值是 0.1 \* 255 = 25。

h阈值，h

为 X 或 Y 轴上未表示的颜色分量设置高阈值。高于此值的值将被忽略。默认值为 1。请注意，此值乘以一个像素组件可以具有的实际最大可能值。所以对于 8 位输入和 0.9 的高阈值，实际阈值是 0.9 \* 255 = 230。

色彩空间，c

设置在绘制标线时使用哪种颜色空间。

'汽车'

' 601 '

' 709 '

默认为自动。

色调 0, t0

色调 1, t1

设置灰色/色调矢量示波器模式的色调。默认情况下，这两个选项均为零。这意味着没有色调，输出将保持灰色。

### 39.269 vidstabdetect

分析视频稳定/去抖动。执行 pass 1 of 2，请参阅pass 2 的[vidstabtransform 。](#vidstabtransform)

[vidstabtransform](#vidstabtransform)过滤器使用。

--enable-libvidstab配置 FFmpeg 。

此过滤器接受以下选项：

结果

设置用于写入转换信息的文件的路径。默认值为transforms.trf 。

颤抖

设置视频的抖动程度和相机的速度。它接受 1-10 范围内的整数，值为 1 表示轻微抖动，值为 10 表示强烈抖动。默认值为 5。

准确性

设置检测过程的精度。它必须是 1-15 范围内的值。值为 1 表示精度低，值为 15 表示精度高。默认值为 15。

一步的大小

设置搜索过程的步长。以 1 像素分辨率扫描最小值周围的区域。默认值为 6。

最小对比度

设置最小对比度。低于此值，本地测量字段将被丢弃。必须是 0-1 范围内的浮点值。默认值为 0.3。

三脚架

设置三脚架模式的参考帧编号。

如果启用，帧的运动将与过滤流中的参考帧进行比较，由指定的数字标识。这个想法是补偿或多或少静态场景中的所有运动并保持相机视图绝对静止。

如果设置为 0，则禁用。帧从 1 开始计数。

展示

在结果帧中显示字段和转换。它接受 0-2 范围内的整数。默认值为 0，即禁用任何可视化。

* [例子](#Examples-172)

#### 39.269.1 示例

* 使用默认值：
* 视频检测
* 分析强烈抖动的电影并将结果放入文件mytransforms.trf中：
* vidstabdetect=shakiness=10:accuracy=15:result="mytransforms.trf"
* 在生成的视频中可视化内部转换的结果：
* vidstabdetect=显示=1
* ffmpeg分析中等抖动的视频：
* ffmpeg -i 输入 -vf vidstabdetect=shakiness=5:show=1 dummy.avi

### 39.270 vidstab变换

视频稳定/去抖动：pass 2 of 2，请参阅[vidstabdetect](#vidstabdetect) for pass 1。

读取包含每一帧变换信息的文件并应用/补偿它们。与[vidstabdetect](#vidstabdetect)过滤器一起使用，可用于去除视频抖动。另见<http://public.hronopik.de/vid.stab>。使用[反锐化](#unsharp)滤镜也很重要，请参见下文。

--enable-libvidstab配置 FFmpeg 。

* [选项](#Options-97)
* [例子](#Examples-173)

#### 39.270.1 选项

输入

设置用于读取转换的文件的路径。默认值为transforms.trf 。

平滑

设置用于对相机运动进行低通滤波的帧数（值\*2 + 1）。默认值为 10。

例如，数字 10 表示使用 21 帧（过去 10 帧，未来 10 帧）来平滑视频中的运动。较大的值会产生更流畅的视频，但会限制摄像机的加速度（平移/倾斜移动）。 0 是模拟静态相机的特殊情况。

光电

设置相机路径优化算法。

可接受的值为：

'高斯'

相机运动的高斯核低通滤波器（默认）

'平均'

平均转换

最大位移

设置最大像素数以翻译帧。默认值为-1，表示没有限制。

最大角度

以弧度（度\*PI/180）为单位设置最大角度以旋转框架。默认值为-1，表示没有限制。

庄稼

指定如何处理由于运动补偿而可能可见的边界。

可用值是：

'保持'

保留前一帧的图像信息（默认）

'黑色'

用黑色填充边框

倒置

如果设置为 1，则反转转换。默认值为 0。

相对的

如果设置为 1，则考虑相对于前一帧进行变换，如果设置为 0，则考虑绝对变换。默认值为 0。

飞涨

设置缩放百分比。正值会产生放大效果，负值会产生缩小效果。默认值为 0（无缩放）。

优化缩放

设置最佳缩放以避免边界。

可接受的值为：

' 0 '

禁用

' 1 '

确定最佳静态缩放值（只有非常强烈的移动才会导致可见边界）（默认）

' 2 '

确定最佳自适应缩放值（无边框可见），请参阅zoomspeed

请注意，缩放时给出的值会添加到此处计算的值。

变焦速度

设置百分比以最大程度地缩放每个帧（当optzoom设置为 2 时启用）。范围从 0 到 5，默认值为 0.25。

国际刑警组织

指定插值类型。

可用值是：

'不'

无插值

'线性'

线性仅水平

'双线性'

双向线性（默认）

'双三次'

双向立方（慢）

三脚架

如果设置为 1，则启用虚拟三脚架模式，相当于relative=0:smoothing=0 。默认值为 0。

还使用[vidstabdetect的](#vidstabdetect)三脚架选项。

调试

如果设置为 1，则增加日志详细程度。检测到的全局运动也会写入临时文件global\_motions.trf 。默认值为 0。

#### 39.270.2 示例

* 使用ffmpeg进行具有默认值的典型稳定性：
* ffmpeg -i inp.mpeg -vf vidstabtransform,unsharp=5:5:0.8:3:3:0.4 inp\_stabilized.mpeg

始终推荐使用[反锐化滤镜。](#unsharp)

* 放大一点并从给定文件加载转换数据：
* vidstabtransform=zoom=5:input="mytransforms.trf"
* 使视频更加流畅：
* vidstabtransform=平滑=30

### 39.271 翻转

垂直翻转输入视频。

ffmpeg垂直翻转视频：

ffmpeg -i in.avi -vf "vflip" out.avi

### 39.272 vfrdet

检测可变帧率视频。

此过滤器尝试检测输入是可变帧率还是恒定帧率。

最后，它将输出检测到的具有可变增量点的帧数和具有恒定增量点的帧数。如果存在具有可变增量的帧，那么它还会显示遇到的最小、最大和平均增量。

### 39.273 活力

提高或改变饱和度。

过滤器接受以下选项：

强度

如果为正值，则设置增强强度；如果为负值，则设置更改强度。默认值为 0。允许的范围是从 -2 到 2。

rbal

设置红色平衡。默认值为 1。允许的范围为 -10 到 10。

全球

设置绿色平衡。默认值为 1。允许的范围为 -10 到 10。

巴尔

设置蓝色平衡。默认值为 1。允许的范围为 -10 到 10。

朗姆酒

设置红色亮度系数。

闷闷不乐

设置绿色亮度系数。

布鲁姆

设置蓝色亮度系数。

备用

如果强度为负且设置为 1，颜色将发生变化，否则颜色将不那么饱和，更接近灰色。

* [命令](#Commands-151)

#### 39.273.1 命令

此过滤器支持以上所有选项作为[命令](#commands)。

### 39.274 维夫

获取两个输入视频之间的平均 VIF（视觉信息保真度）。

这个过滤器需要两个输入视频。

两个输入视频必须具有相同的分辨率和像素格式，此过滤器才能正常工作。它还假设两个输入具有相同数量的帧，将它们一一比较。

通过日志系统打印获得的平均 VIF 分数。

过滤器存储计算出的每一帧的 VIF 分数。

此过滤器还支持[帧同步](#framesync)选项。

在下面的示例中，将正在处理的输入文件main.mpg与参考文件ref.mpg进行比较。

ffmpeg -i main.mpg -i ref.mpg -lavfi vif -f null -

### 39.275 小插图

制作或反转自然渐晕效果。

过滤器接受以下选项：

角度，一个

将镜头角度表达式设置为弧度数。

该值被限制在[0,PI/2]范围内。

默认值： “PI/5”

x0

y0

设置中心坐标表达式。默认分别为“w/2”和“h/2” 。

模式

设置前进/后退模式。

可用模式有：

'前进'

距中心点的距离越大，图像越暗。

'落后'

距中心点的距离越大，图像变得越亮。这可用于反转晕影效果，但还没有自动检测来提取镜头角度和其他设置（目前）。它还可以用于创建燃烧效果。

默认值为“ forward ”。

评估

为表达式（ angle 、 x0 、 y0 ）设置评估模式。

它接受以下值：

'初始化'

在过滤器初始化期间仅对表达式求值一次。

'框架'

评估每个传入帧的表达式。这比“ init ”模式慢得多，因为它需要重新计算所有缩放器，但它允许高级动态表达式。

默认值为“ init ”。

抖动

设置抖动以减少圆形条纹效果。默认值为1 （启用）。

方面

设置小插图方面。此设置允许调整小插图的形状。将此值设置为输入的 SAR 将使矩形渐晕遵循视频的尺寸。

默认为1/1 。

* [表达式](#Expressions)
* [例子](#Examples-174)

#### 39.275.1 表达式

alpha 、 x0和y0表达式可以包含以下参数。

w

H

输入宽度和高度

n

输入帧数，从0开始

积分

过滤后的视频帧的 PTS（Presentation TimeStamp）时间，以TB为单位表示，如果未定义则为 NAN

r

输入视频的帧率，如果输入帧率未知则为 NAN

吨

过滤后的视频帧的 PTS（Presentation TimeStamp），以秒为单位，如果未定义则为 NAN

结核病

输入视频的时基

#### 39.275.2 示例

* 应用简单的强渐晕效果：
* 插图=PI/4
* 做一个闪烁的渐晕：
* vignette='PI/4+random(1)\*PI/50':eval=frame

### 39.276 vmaf运动

获取视频的平均 VMAF 运动分数。它是 VMAF 的组件指标之一。

获得的平均运动分数通过日志系统打印。

过滤器接受以下选项：

统计文件

如果指定，过滤器将使用命名文件保存每一帧相对于前一帧的运动分数。当文件名等于“-”时，数据被发送到标准输出。

例子：

ffmpeg -i ref.mpg -vf vmafmotion -f null -

### 39.277 vstack

垂直堆叠输入视频。

所有流必须具有相同的像素格式和相同的宽度。

请注意，此过滤器比使用[覆盖](#overlay)和[垫](#pad)过滤器创建相同的输出更快。

过滤器接受以下选项：

输入

设置输入流的数量。默认值为 2。

最短的

如果设置为 1，则强制输出在最短输入终止时终止。默认值为 0。

### 39.278 w3fdif

去隔行输入视频（“w3fdif”代表“Weston 3 Field Deinterlacing Filter”）。

基于 Martin Weston 为 BBC R&D 描述的过程，并基于 Jim Easterbrook 为 BBC R&D 编写的去隔行算法实现，Weston 3 场去隔行滤波器使用 BBC R&D 计算的滤波器系数。

此过滤器使用帧中的场优势信息来决定将每对场中的哪一个放在输出的第一位。如果它弄错了，请在w3fdif过滤器之前使用[setfield过滤器](#setfield)。

有两组滤波器系数，所谓的“简单”和“复杂”。可以通过传递一个可选参数来设置使用哪一组滤波器系数：

筛选

设置隔行扫描滤波器系数。接受以下值之一：

'简单'

简单的滤波器系数集。

'复杂'

更复杂的滤波器系数集。

默认值为“复杂”。

模式

采用的隔行扫描模式。它接受以下值之一：

框架

每一帧输出一帧。

场地

为每个字段输出一帧。

默认值为字段。

平价

输入隔行扫描视频采用的图片场奇偶校验。它接受以下值之一：

tf

假设顶部字段是第一个。

好朋友

假设底部字段是第一个。

汽车

启用字段奇偶校验的自动检测。

默认值为自动。如果隔行扫描未知或解码器不导出此信息，则假定顶场优先。

取消

指定要去隔行扫描的帧。接受以下值之一：

'全部'

去隔行扫描所有帧，

'交错'

仅去隔行标记为隔行扫描的帧。

默认值为“全部”。

* [命令](#Commands-152)

#### 39.278.1 命令

此过滤器支持与选项相同的[命令](#commands)。

### 39.279 波形

视频波形监视器。

波形监视器绘制颜色分量强度。默认情况下仅亮度。波形的每一列对应于源视频中的一列像素。

它接受以下选项：

模式，米

可以是行，也可以是列。默认为列。在行模式下，左侧的图形表示颜色分量值 0，右侧表示值 = 255。在列模式下，顶部表示颜色分量值 = 0，底部表示值 = 255。

强度，我

设置强度。较小的值有助于找出有多少相同亮度的值分布在输入行/列中。默认值为0.04 。允许的范围是 [0, 1]。

镜子, r

设置镜像模式。 0表示未镜像， 1表示已镜像。在镜像模式下，较高的值将在行模式的左侧和列模式的顶部表示。默认值为1 （镜像）。

显示，d

设置显示模式。它接受以下值：

'覆盖'

parade中的信息相同，除了表示颜色分量的图表直接叠加在彼此之上。

这种显示模式可以更容易地发现本应相同的颜色分量重叠区域的相对差异或相似性，例如中性白色、灰色或黑色。

'堆栈'

在行模式下并排显示颜色分量的单独图表，或在列模式下一个在另一个下方。

'游行'

列模式下并排显示颜色分量的单独图形，或在行模式下显示一个在另一个下方。

使用此显示模式可以通过比较每个波形的顶部和底部图形的轮廓，轻松发现图像高光和阴影中的色偏。由于白色、灰色和黑色的特征在于红色、绿色和蓝色的数量完全相等，因此图片的中性区域应显示三个宽度/高度大致相等的波形。如果不是，则可以通过对三个波形进行电平调整来轻松执行校正。

默认是堆栈。

组件，c

设置要显示的颜色分量。默认值为 1，这意味着如果输入在 RGB 色彩空间中，则只有亮度或红色分量。例如，如果设置为 7，它将显示所有 3 个（如果）可用的颜色分量。

信封，e

'无'

没有信封，这是默认设置。

'即时'

步长值很小，图中显示的即时包络线、最小值和最大值也很容易看到。

'峰'

保持图表中随时间显示的最小值和最大值。这样，您仍然可以发现超出范围的值，而无需不断查看波形。

'巅峰+瞬间'

峰值和即时包络结合在一起。

过滤器，f

'低通'

没有过滤，这是默认的。

'平'

亮度和色度组合在一起。

'平坦'

与上面类似，但显示蓝色和红色色度之间的差异。

' xflat '

与上面类似，但使用不同的颜色。

' yflat '

与上面类似，但又是不同的颜色。

'色度'

仅显示色度。

'颜色'

在波形上显示实际颜色值。

'颜色'

与上面类似，但亮度显示色度值的频率。

标线，克

设置要显示的格线。

'无'

不显示格线。

'绿色'

显示显示合法广播范围的绿色标线。

'橙色'

显示表示合法广播范围的橙色标线。

'反转'

显示显示合法广播范围的反转标线。

不透明度，o

设置格线不透明度。

佛罗里达州旗帜

设置标线标志。

'数字'

在线条上方画数字。默认启用。

'点'

画点而不是线。

规模，小号

设置用于显示格线的刻度。

'数字'

'毫伏'

'愤怒'

默认为数字。

b 透明度, b

设置背景不透明度。

色调 0, t0

色调 1, t1

设置输出色调。仅与低通滤波器一起使用，并且当显示未覆盖且输入像素格式不是 RGB 时。

健身模式，调频

设置视频输出帧的样本纵横比。可用于配置波形，使其不会在一个方向上拉伸太多。

'无'

将样本纵横比设置为 1/1。

'尺寸'

设置样本宽高比以匹配视频的输入大小

默认为“无”。

### 39.280 编织，双面编织

编织采用基于场的视频输入并将每两个连续场连接成单个帧，从而产生帧速率和帧数减半的新双倍高度剪辑。

doubleweave的工作原理与weave相同，但不会将帧速率和帧数减半。

它接受以下选项：

first\_field

设置第一个字段。可用值是：

'顶， '

将帧设置为顶场优先。

'底部，b '

将帧设置为底场优先。

* [例子](#Examples-175)

#### 39.280.1 例子

* [select](#select)和[separatefields过滤器的](#separatefields)隔行扫描视频：
* separatefields,select=eq(mod(n,4),0)+eq(mod(n,4),3),weave

### 39.281 xbr

应用专为像素艺术设计的 xBR 高质量放大滤镜。它遵循一组边缘检测规则，请参阅<https://forums.libretro.com/t/xbr-algorithm-tutorial/123>。

它接受以下选项：

n

设置缩放维度： 2xBR为2，3xBR为3，4xBR为4 。 \_默认值为3 。

### 39.282 x相关

在第一和第二输入视频流之间应用归一化互相关。

第二个输入视频流的尺寸必须小于第一个输入视频流。

过滤器接受以下选项：

飞机

设置要处理的平面。

次要的

设置将从第二个输入视频流中处理哪些辅助视频帧，可以是first或all 。默认为全部。

xcorrelate过滤器还支持[帧同步](#framesync)选项。

### 39.283 x淡入淡出

将一个输入视频流的淡入淡出应用于另一个输入视频流。淡入淡出应用于指定的持续时间。

两个输入必须是恒定的帧率并且具有相同的分辨率、像素格式、帧率和时基。

过滤器接受以下选项：

过渡

设置一种可用的过渡效果：

'自定义'

'褪色'

' wipeleft '

'擦拭'

'擦除'

'擦拭'

'向左滑动'

'向右滑动'

'幻灯片'

'下滑'

'循环作物'

'直肠'

'距离'

'淡黑'

'淡白'

'径向'

' smoothleft '

' smoothright '

'平滑'

'平滑'

'圆开'

'关闭'

' vertopen '

' vertclose '

' horzopen '

'关闭'

'溶解'

'像素化'

'诊断'

'诊断'

'恶魔'

'诊断'

'切片'

'小时片'

' Vuslice '

' vdslice '

'模糊'

'渐变灰'

'擦拭'

'擦拭器'

'擦除'

'擦拭布'

'挤压'

'挤压'

'放大'

'褪色'

'淡化'

默认过渡效果是淡入淡出。

期间

以秒为单位设置淡入淡出持续时间。范围是 0 到 60 秒。默认持续时间为 1 秒。

抵消

以秒为单位设置相对于第一个输入流的淡入淡出开始。默认偏移量为 0。

表达式

设置自定义过渡效果的表达式。

表达式可以使用以下变量和函数：

X

是

当前样本的坐标。

W

H

图像的宽度和高度。

P

过渡效果的进展。

飞机

当前处理的平面。

A

当前位置和平面的第一个输入的返回值。

乙

当前位置和平面的第二个输入的返回值。

a0(x, y)

a1(x, y)

a2(x, y)

a3(x, y)

第一个输入的第一/第二/第三/第四分量的位置 ( x , y )处的像素值。

b0(x, y)

b1(x, y)

b2(x, y)

b3(x, y)

第二个输入的第一/第二/第三/第四分量的位置 ( x , y )处的像素值。

* [例子](#Examples-176)

#### 39.283.1 例子

* 从一个输入视频到另一个输入视频的淡入淡出，淡入淡出过渡，过渡持续时间为 2 秒，从偏移量 5 秒开始：
* ffmpeg -i first.mp4 -i second.mp4 -filter\_complex xfade=transition=fade:duration=2:offset=5 output.mp4

### 39.284 x 中值

从多个输入视频中选取中值像素。

过滤器接受以下选项：

输入

设置输入数量。默认值为 3。允许的范围是从 3 到 255。如果输入的数量是偶数，则结果将是两个中值之间的平均值。

飞机

设置要过滤的平面。默认值为15 ，通过它处理所有平面。

百分位数

设置中位数百分位数。默认值为0.5 。默认值0.5将始终选择中值，而0将选择最小值， 1将选择最大值。

* [命令](#Commands-153)

#### 39.284.1 命令

此过滤器支持以上所有选项作为[命令](#commands)，不包括选项输入。

### 39.285 x堆栈

将视频输入堆叠到自定义布局中。

所有流必须具有相同的像素格式。

过滤器接受以下选项：

输入

设置输入流的数量。默认值为 2。

布局

指定输入布局。此选项需要用户明确设置所需的布局配置。这设置了每个视频输入在输出中的位置。每个输入由“|”分隔。第一个数字代表列，第二个数字代表行。数字从 0 开始，以“\_”分隔。可以选择使用 wX 和 hX，其中 X 是从中获取宽度或高度的视频输入。用“+”分隔时可以使用多个值。在这种情况下，值被加在一起。

请注意，如果输入的尺寸不同，可能会出现间隙，因为并非所有输出视频帧都将被填充。同样，如果视频的位置没有为相邻视频的全帧留出足够的空间，则视频可能会相互重叠。

设置了默认布局0\_0|w0\_0 （相当于grid=2x1 ）。在所有其他情况下，布局或网格必须由用户设置。一次可以指定网格或布局。指定两者将导致错误。

网格

指定固定大小的输入网格。此选项用于创建输入流的固定大小网格。以COLUMNSxROWS 的形式设置网格大小。必须有ROWS \* COLUMNS输入流，它们将被排列为具有ROWS行和COLUMNS列的网格。使用此选项时，一行中的每个输入流必须具有相同的高度，并且所有行必须具有相同的宽度。

如果设置了网格，则输入选项将被忽略并隐式设置为ROWS \* COLUMNS 。

设置默认网格2x1 （相当于layout=0\_0|w0\_0 ）。在所有其他情况下，布局或网格必须由用户设置。一次可以指定网格或布局。指定两者将导致错误。

最短的

如果设置为 1，则强制输出在最短输入终止时终止。默认值为 0。

充满

如果设置为有效颜色，则所有未使用的像素都将填充该颜色。默认情况下，填充设置为无，因此它被禁用。

* [例子](#Examples-177)

#### 39.285.1 示例

* 在 2x2 网格中显示 4 个输入。

布局：

输入 1(0, 0) |输入 3(w0, 0)

输入 2(0, h0) |输入 4(w0, h0)

xstack=输入=4:布局=0\_0|0\_h0|w0\_0|w0\_h0

请注意，如果输入的大小不同，则可能会出现间隙或重叠。

* 在 1x4 网格中显示 4 个输入。

布局：

输入 1(0, 0)

输入 2(0, h0)

输入 3(0, h0+h1)

输入 4(0, h0+h1+h2)

xstack=输入=4:布局=0\_0|0\_h0|0\_h0+h1|0\_h0+h1+h2

请注意，如果输入的宽度不同，则会出现未使用的空间。

* 在 3x3 网格中显示 9 个输入。

布局：

输入 1(0, 0) |输入 4(w0, 0) |输入 7(w0+w3, 0)

输入 2(0, h0) |输入 5(w0, h0) |输入 8(w0+w3, h0)

输入 3(0, h0+h1) |输入 6(w0, h0+h1) |输入 9(w0+w3, h0+h1)

xstack=输入=9:布局=0\_0|0\_h0|0\_h0+h1|w0\_0|w0\_h0|w0\_h0+h1|w0+w3\_0|w0+w3\_h0|w0+w3\_h0+h1

请注意，如果输入的大小不同，则可能会出现间隙或重叠。

* 在 4x4 网格中显示 16 个输入。

布局：

输入 1(0, 0) |输入 5(w0, 0) |输入 9 (w0+w4, 0) |输入 13(w0+w4+w8, 0)

输入 2(0, h0) |输入 6(w0, h0) |输入 10(w0+w4, h0) |输入 14(w0+w4+w8, h0)

输入 3(0, h0+h1) |输入 7(w0, h0+h1) |输入 11(w0+w4, h0+h1) |输入 15(w0+w4+w8, h0+h1)

输入4（0，h0+h1+h2）|输入 8(w0, h0+h1+h2)|输入12（w0+w4，h0+h1+h2）|输入 16(w0+w4+w8, h0+h1+h2)

xstack=输入=16:布局=0\_0|0\_h0|0\_h0+h1|0\_h0+h1+h2|w0\_0|w0\_h0|w0\_h0+h1|w0\_h0+h1+h2|w0+w4\_0|

w0+w4\_h0|w0+w4\_h0+h1|w0+w4\_h0+h1+h2|w0+w4+w8\_0|w0+w4+w8\_h0|w0+w4+w8\_h0+h1|w0+w4+w8\_h0+h1+h2

请注意，如果输入的大小不同，则可能会出现间隙或重叠。

### 39.286 亚迪夫

去隔行输入视频（“yadif”的意思是“又一个去隔行过滤器”）。

它接受以下参数：

模式

采用的隔行扫描模式。它接受以下值之一：

0，发送帧

每一帧输出一帧。

1、发送字段

为每个字段输出一帧。

2、send\_frame\_nospatial

类似于send\_frame ，但它跳过了空间交错检查。

3、send\_field\_nospatial

像send\_field ，但它跳过了空间交错检查。

默认值为send\_frame 。

平价

输入隔行扫描视频采用的图片场奇偶校验。它接受以下值之一：

0, tff

假设顶部字段是第一个。

1、好朋友

假设底部字段是第一个。

-1，汽车

启用字段奇偶校验的自动检测。

默认值为自动。如果隔行扫描未知或解码器不导出此信息，则假定顶场优先。

取消

指定要去隔行扫描的帧。接受以下值之一：

0, 全部

去隔行扫描所有帧。

1、交错

仅去隔行标记为隔行扫描的帧。

默认值为全部。

### 39.287 yadif\_cuda

[yadif算法](#yadif)对输入视频进行去隔行扫描，但在 CUDA 中实现，以便它可以作为 GPU 加速管道的一部分与 nvdec 和/或 nvenc 一起工作。

它接受以下参数：

模式

采用的隔行扫描模式。它接受以下值之一：

0，发送帧

每一帧输出一帧。

1、发送字段

为每个字段输出一帧。

2、send\_frame\_nospatial

类似于send\_frame ，但它跳过了空间交错检查。

3、send\_field\_nospatial

像send\_field ，但它跳过了空间交错检查。

默认值为send\_frame 。

平价

输入隔行扫描视频采用的图片场奇偶校验。它接受以下值之一：

0, tff

假设顶部字段是第一个。

1、好朋友

假设底部字段是第一个。

-1，汽车

启用字段奇偶校验的自动检测。

默认值为自动。如果隔行扫描未知或解码器不导出此信息，则假定顶场优先。

取消

指定要去隔行扫描的帧。接受以下值之一：

0, 全部

去隔行扫描所有帧。

1、交错

仅去隔行标记为隔行扫描的帧。

默认值为全部。

### 39.288 亚普布勒

在保留边缘的同时应用模糊滤镜（“yaepblur”的意思是“又一个边缘保留模糊滤镜”）。该算法在“JS Lee, Digital image enhancement and noise filtering by use of local statistics, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-2, 1980”中有所描述。

它接受以下参数：

半径，r

设置窗口半径。默认值为 3。

飞机，p

设置要过滤的平面。默认只有第一个平面。

σ, s

设置模糊强度。默认值为 128。

* [命令](#Commands-154)

#### 39.288.1 命令

此过滤器支持与选项相同的[命令](#commands)。

### 39.289 缩放平移

应用缩放和平移效果。

此过滤器接受以下选项：

缩放，z

设置缩放表达式。范围是 1-10。默认值为 1。

X

是

设置 x 和 y 表达式。默认为 0。

d

以帧数设置持续时间表达式。这设置了单个输入图像将持续多少帧效果。默认值为 90。

秒

设置输出图像大小，默认为'hd720'。

每秒帧数

设置输出帧率，默认为'25'。

每个表达式可以包含以下常量：

in\_w, iw

输入宽度。

在\_h，我

输入高度。

out\_w, ow

输出宽度。

out\_h, 哦

输出高度。

在

输入帧数。

在

输出帧数。

及时，它

输入时间戳，以秒为单位。如果输入时间戳未知，则为 NAN。

out\_time, 时间, ot

以秒为单位的输出时间戳。

X

是

根据当前输入帧的“x”和“y”表达式最后计算的“x”和“y”位置。

像素

py

前一个输入帧的最后一个输出帧的“x”和“y”，或者当还没有这样的帧（第一个输入帧）时为 0。

飞涨

当前输入帧的“z”表达式的最后计算缩放。

缩放

上一个输入帧的最后一个输出帧的最后计算缩放。

期间

当前输入帧的输出帧数。从每个输入帧的“d”表达式计算。

持续时间

为前一个输入帧创建的输出帧数

A

有理数：输入宽度/输入高度

特区

样品纵横比

达尔

显示纵横比

* [例子](#Examples-178)

#### 39.289.1 例子

* 放大 1.5 倍并同时平移到图片中心附近的某个点：
* zoompan=z='min(zoom+0.0015,1.5)':d=700:x='if(gte(zoom,1.5),x,x+1/a)':y='if(gte(zoom, 1.5),y,y+1)':s=640x360
* 放大 1.5 倍并始终在图片中心平移：
* zoompan=z='min(zoom+0.0015,1.5)':d=700:x='iw/2-(iw/zoom/2)':y='ih/2-(ih/zoom/2)'
* 与上面相同但没有暂停：
* zoompan=z='min(max(zoom,pzoom)+0.0015,1.5)':d=1:x='iw/2-(iw/zoom/2)':y='ih/2-(ih/缩放/2)'
* 仅在输入视频的第一秒将图片中心放大 2 倍：
* zoompan=z='if(between(in\_time,0,1),2,1)':d=1:x='iw/2-(iw/zoom/2)':y='ih/2-( ih/缩放/2)'

### 39.290 zscale

使用 z.lib 库缩放（调整大小）输入视频： <https://github.com/sekrit-twc/zimg>。要启用此过滤器的编译，您需要使用--enable-libzimg配置 FFmpeg 。

zscale 过滤器通过更改输出样本纵横比强制输出显示纵横比与输入相同。

如果输入图像格式与下一个过滤器请求的格式不同，zscale 过滤器会将输入转换为请求的格式。

* [选项](#Options-98)
* [命令](#Commands-155)

#### 39.290.1 选项

过滤器接受以下选项。

宽度，w

身高，小时

设置输出视频维度表达式。默认值是输入维度。

如果宽度或w值为 0，则输入宽度用于输出。如果高度或h值为 0，则输入高度用于输出。

如果一个且只有一个值是 -n 且 n >= 1，则 zscale 过滤器将使用一个值来保持输入图像的纵横比，该纵横比是从另一个指定维度计算得出的。但是，之后它将确保计算出的维度可以被 n 整除，并在必要时调整该值。

如果两个值都是 -n 并且 n >= 1，则行为将与两个值都设置为 0 相同，如前所述。

请参阅下面的维度表达式中可接受的常量列表。

大小，小号

设置视频大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。

抖动，d

设置抖动类型。

可能的值是：

没有任何

有序的

随机的

误差扩散

默认为无。

过滤器，f

设置调整大小过滤器类型。

可能的值是：

观点

双线性

双三次

样条16

样条36

兰佐斯

默认是双线性的。

范围，r

设置颜色范围。

可能的值是：

输入

有限的

满的

默认与输入相同。

初选，p

设置颜色原色。

可能的值是：

输入

709

未指定

170米

240米

2020

默认与输入相同。

转移，吨

设置传输特性。

可能的值是：

输入

709

未指定

601

线性的

2020\_10

2020\_12

smpte2084

iec61966-2-1

arib-std-b67

默认与输入相同。

矩阵，米

设置色彩空间矩阵。

可能的值是：

输入

709

未指定

470bg

170米

2020\_ncl

2020\_cl

默认与输入相同。

范围，林

设置输入颜色范围。

可能的值是：

输入

有限的

满的

默认与输入相同。

primariesin, 引脚

设置输入颜色原色。

可能的值是：

输入

709

未指定

170米

240米

2020

默认与输入相同。

转铁蛋白，锡

设置输入传输特性。

可能的值是：

输入

709

未指定

601

线性的

2020\_10

2020\_12

默认与输入相同。

基质，最小值

设置输入色彩空间矩阵。

可能的值是：

输入

709

未指定

470bg

170米

2020\_ncl

2020\_cl

彩色，c

设置输出色度位置。

可能的值是：

输入

左边

中心

左上方

顶部

左下方

底部

染色质，cin

设置输入色度位置。

可能的值是：

输入

左边

中心

左上方

顶部

左下方

底部

不良贷款

设置标称峰值亮度。

参数a

用于缩放过滤器的参数 A。 bicubic 的参数“b”和 lanczos 的滤波器抽头数。

参数\_b

用于缩放过滤器的参数 B。双三次的参数“c”。

w和h选项的值是包含以下常量的表达式：

in\_w

在\_h

输入的宽度和高度

我

我

in\_w和in\_h相同。

out\_w

out\_h

输出（缩放）宽度和高度

噢

哦

out\_w和out\_h相同

A

iw / ih相同

特区

输入样本纵横比

达尔

输入显示纵横比。从(iw / ih) \* sar计算。

hsub

vsub

水平和垂直输入色度子样本值。例如，对于像素格式“yuv422p”， hsub为 2， vsub为 1。

哦sub

卵子

水平和垂直输出色度子样本值。例如，对于像素格式“yuv422p”， hsub为 2， vsub为 1。

#### 39.290.2 命令

该过滤器支持以下命令：

宽度，w

身高，小时

设置输出视频维度表达式。该命令接受与相应选项相同的语法。

如果指定的表达式无效，它会保持其当前值。

## 40 个 OpenCL 视频滤波器

下面是对当前可用的 OpenCL 视频过滤器的描述。

--enable-opencl配置 FFmpeg 。

运行 OpenCL 过滤器需要您初始化硬件设备并将该设备传递给任何过滤器图中的所有过滤器。

-init\_hw\_device opencl[= name ][: device [, key=value ...]]

使用给定的设备参数初始化一个名为name的opencl类型的新硬件设备。

-filter\_hw\_device名称

name的硬件设备传递给任何过滤器图中的所有过滤器。

有关更多详细信息，请参阅<https://www.ffmpeg.org/ffmpeg.html#Advanced-Video-options>

* 在第二个平台上选择第一个设备并使用默认参数运行 avgblur\_opencl 过滤器的示例。
* -init\_hw\_device opencl=gpu:1.0 -filter\_hw\_device gpu -i INPUT -vf "hwupload, avgblur\_opencl, hwdownload" 输出

由于 OpenCL 过滤器无法访问普通内存中的帧数据，所有帧数据在使用前都需要上传 ( [hwupload](#hwupload) ) 到连接到适当设备的硬件表面，然后下载 ( [hwdownload](#hwdownload) ) 回普通内存。请注意， [hwupload](#hwupload)将上传到与软件框架具有相同布局的表面，因此可能需要在将输入转换为正确格式之前立即添加[格式过滤器，并且](#format)[hwdownload](#hwdownload)不支持输出上的所有格式 - 它可能有必要在图表中紧随其后插入一个额外的[格式过滤器，以获得受支持格式的输出。](#format)

* [avgblur\_opencl](#avgblur_005fopencl)
* [boxblur\_opencl](#boxblur_005fopencl)
* [colorkey\_opencl](#colorkey_005fopencl)
* [卷积\_opencl](#convolution_005fopencl)
* [侵蚀\_opencl](#erosion_005fopencl)
* [deshake\_opencl](#deshake_005fopencl)
* [扩张\_opencl](#dilation_005fopencl)
* [nlmeans\_opencl](#nlmeans_005fopencl)
* [overlay\_opencl](#overlay_005fopencl)
* [pad\_opencl](#pad_005fopencl)
* [prewitt\_opencl](#prewitt_005fopencl)
* [程序\_opencl](#program_005fopencl-1)
* [重新映射\_opencl](#remap_005fopencl)
* [罗伯茨\_opencl](#roberts_005fopencl)
* [sobel\_opencl](#sobel_005fopencl)
* [色调映射\_opencl](#tonemap_005fopencl)
* [unsharp\_opencl](#unsharp_005fopencl)
* [xfade\_opencl](#xfade_005fopencl)

### 40.1 avgblur\_opencl

应用平均模糊滤镜。

过滤器接受以下选项：

尺码X

设置水平半径大小。范围是[1, 1024] ，默认值为1 。

飞机

设置要过滤的平面。默认值为0xf ，通过它处理所有平面。

尺码Y

设置垂直半径大小。范围是[1, 1024] ，默认值为0 。如果为零，将使用sizeX值。

* [例子](#Example-4)

#### 40.1.1 示例

* 应用水平和垂直大小为 3 的平均模糊过滤器，将输出的每个像素设置为输入中以它为中心的 7x7 区域的平均值。对于图像边缘的像素，该区域不会超出图像边界，因此计算中不使用超出范围的坐标。
* -i 输入 -vf "hwupload, avgblur\_opencl=3, hwdownload" 输出

### 40.2 boxblur\_opencl

将 boxblur 算法应用于输入视频。

它接受以下参数：

亮度半径，lr

luma\_power, lp

色度半径，cr

色度功率，cp

alpha\_radius, ar

alpha\_power, ap

接受的选项的描述如下。

亮度半径，lr

色度半径，cr

alpha\_radius, ar

设置用于模糊相应输入平面的框半径的表达式（以像素为单位）。

半径值必须是非负数，并且不得大于亮度和 alpha 平面的表达式min(w,h)/2的值，以及色度的min(cw,ch)/2的值飞机。

luma\_radius的默认值为“2”。如果未指定， chroma\_radius和alpha\_radius默认为为luma\_radius设置的相应值。

表达式可以包含以下常量：

w

H

输入的宽度和高度（以像素为单位）。

连续波

通道

输入色度图像的宽度和高度（以像素为单位）。

hsub

vsub

水平和垂直色度子样本值。例如，对于像素格式“yuv422p”， hsub为 2， vsub为 1。

luma\_power, lp

色度功率，cp

alpha\_power, ap

指定将 boxblur 滤镜应用于相应平面的次数。

luma\_power的默认值为2。如果未指定， chroma\_power和alpha\_power默认为为luma\_power设置的相应值。

值为 0 将禁用该效果。

* [例子](#Examples-179)

#### 40.2.1 例子

为每个平面的 box-radiuses luma\_radius 、 chroma\_radius 、 alpha\_radius的平均值。滤镜会将luma\_power 、 chroma\_power 、 alpha\_power时间应用到相应的平面上。对于图像边缘的像素，半径不会超出图像边界，因此计算中不使用超出范围的坐标。

* 应用 boxblur 滤镜，其中亮度、色度和 alpha 半径设置为 2，亮度、色度和 alpha 功率设置为 3。对于图像的每个平面，滤镜将运行 3 次，box-radius 设置为 2。
* -i 输入 -vf "hwupload, boxblur\_opencl=luma\_radius=2:luma\_power=3, hwdownload" 输出
* -i 输入 -vf "hwupload, boxblur\_opencl=2:3, hwdownload" 输出
* 应用亮度半径设置为 2、luma\_power 为 1、chroma\_radius 为 4、chroma\_power 为 5、alpha\_radius 为 3 和 alpha\_power 为 7 的 boxblur 滤镜。

对于亮度平面，将运行一次 2x2 框半径。

对于色度平面，4x4 框半径将运行 5 次。

对于 alpha 平面，将运行 7 次 3x3 框半径。

-i 输入 -vf "hwupload, boxblur\_opencl=2:1:4:5:3:7, hwdownload" 输出

### 40.3 colorkey\_opencl

RGB 色彩空间颜色键控。

过滤器接受以下选项：

颜色

将被透明度替换的颜色。

相似

与关键颜色的相似度百分比。

0.01 只匹配准确的键颜色，而 1.0 匹配所有颜色。

混合

混合百分比。

0.0 使像素完全透明或完全不透明。

较高的值导致半透明像素，透明度越高，像素颜色与键颜色越相似。

* [例子](#Examples-180)

#### 40.3.1 例子

* 通过一些轻微的混合使输入中的每个半绿色像素透明：
* -i 输入 -vf "hwupload, colorkey\_opencl=green:0.3:0.1, hwdownload" 输出

### 40.4 卷积\_opencl

应用 3x3、5x5、7x7 矩阵的卷积。

过滤器接受以下选项：

0米

1米

2米

3米

为每个平面设置矩阵。矩阵是 9、25 或 49 个有符号数的序列。每个平面的默认值为0 0 0 0 1 0 0 0 0 。

0rdiv

1rdiv

2rdiv

第三分区

为每个平面的计算值设置乘数。如果未设置或为 0，它将是所有矩阵元素的总和。选项值必须是大于或等于0.0 的浮点数。默认值为1.0 。

0偏差

1偏差

2偏差

3偏置

为每个平面设置偏差。该值被添加到乘法的结果。用于使整体图像更亮或更暗。选项值必须是大于或等于0.0 的浮点数。默认值为0.0 。

* [例子](#Examples-181)

#### 40.4.1 例子

* 应用锐化：
* -i 输入 -vf "hwupload, convolution\_opencl=0 -1 0 -1 5 -1 0 -1 0:0 -1 0 -1 5 -1 0 -1 0:0 -1 0 -1 5 -1 0 -1 0:0 -1 0 -1 5 -1 0 -1 0，硬件下载“输出
* 应用模糊：
* -i 输入 -vf "hwupload, convolution\_opencl=1 1 1 1 1 1 1 1 1:1 1 1 1 1 1 1 1 1:1 1 1 1 1 1 1 1 1:1 1 1 1 1 1 1 1 1: 1/9:1/9:1/9:1/9, hwdownload" 输出
* 应用边缘增强：
* -i 输入 -vf "hwupload, convolution\_opencl=0 0 0 -1 1 0 0 0 0:0 0 0 -1 1 0 0 0 0:0 0 0 -1 1 0 0 0 0:0 0 0 -1 1 0 0 0 0:5:1:1:1:0:128:128:128，硬件下载“输出
* 应用边缘检测：
* -i 输入 -vf "hwupload, convolution\_opencl=0 1 0 1 -4 1 0 1 0:0 1 0 1 -4 1 0 1 0:0 1 0 1 -4 1 0 1 0:0 1 0 1 -4 1 0 1 0:5:5:5:1:0:128:128:128，硬件下载“输出
* 应用包括对角线的拉普拉斯边缘检测器：
* -i 输入 -vf "hwupload, convolution\_opencl=1 1 1 1 -8 1 1 1 1:1 1 1 1 -8 1 1 1 1:1 1 1 1 -8 1 1 1 1:1 1 1 1 -8 1 1 1 1:5:5:5:1:0:128:128:0，硬件下载“输出
* 应用浮雕：
* -i 输入 -vf "hwupload, convolution\_opencl=-2 -1 0 -1 1 1 0 1 2:-2 -1 0 -1 1 1 0 1 2:-2 -1 0 -1 1 1 0 1 2:- 2 -1 0 -1 1 1 0 1 2，硬件下载”输出

### 40.5 侵蚀\_opencl

对视频应用侵蚀效果。

此过滤器用局部 (3x3) 最小值替换像素。

它接受以下选项：

阈值0

阈值1

阈值2

阈值3

限制每个平面的最大变化。范围是[0, 65535] ，默认值为65535 。如果为0 ，平面将保持不变。

坐标

指定要引用的像素的标志。范围是[0, 255]并且默认值为255 ，即使用所有八个像素。

x为中心的局部 3x3 坐标区域的标志：

1 2 3

4×5

6 7 8

* [例子](#Example-5)

#### 40.5.1 例子

* 应用腐蚀过滤器，阈值 0 设置为 30，阈值 1 设置为 40，阈值 2 设置为 50，坐标设置为 231，将输出的每个像素设置为像素之间的局部最小值：3x3 的 1、2、3、6、7、8输入中以它为中心的区域。如果输入像素和局部最小值之间的差异大于相应平面的阈值，则输出像素将设置为输入像素 - 相应平面的阈值。
* -i 输入 -vf "hwupload, erosion\_opencl=30:40:50:coordinates=231, hwdownload" 输出

### 40.6 deshake\_opencl

基于特征点的视频稳定滤波器。

过滤器接受以下选项：

三脚架

通过防止任何相机从原始框架移动来模拟三脚架。默认为0 。

调试

是否应在处理后的输出和控制台中显示额外的调试信息。

请注意，为了查看控制台调试输出，您还需要将-v verbose传递给 ffmpeg。

输出视频中的视点匹配仅支持 RGB 输入。

默认为0 。

自适应作物

是否在边界处进行一点裁剪以减少镜像像素的数量。

默认为1 。

细化特征

是否应该在子像素级别细化特征点。

可以关闭此功能以牺牲精度为代价获得轻微的性能提升。

默认为1 。

平滑强度

0.0到1.0应用于相机路径的平滑强度。

1.0是最大平滑强度，而小于该值的值会导致平滑度降低。

0.0会导致过滤器在每帧的基础上自适应地选择平滑强度。

默认为0.0 。

平滑窗口乘数

控制平滑窗口的大小（缓冲的帧数以确定运动信息）。

平滑窗口的大小通过将视频的帧率乘以该数字来确定。

可接受的值范围从0.1到10.0 。

较大的值会增加可用于确定如何平滑相机路径的运动数据量，从而可能提高平滑度，但也会增加延迟和内存使用。

默认为2.0 。

* [例子](#Examples-182)

#### 40.6.1 例子

* 使用固定的中等平滑强度稳定视频：
* -i 输入 -vf "hwupload, deshake\_opencl=smooth\_strength=0.5, hwdownload" 输出
* 通过调试稳定视频（在控制台和渲染视频中）：
* -i 输入 -filter\_complex “[0:v]format=rgba, hwupload, deshake\_opencl=debug=1, hwdownload, format=rgba, format=yuv420p” -v 详细输出

### 40.7 dilation\_opencl

对视频应用膨胀效果。

此过滤器用局部 (3x3) 最大值替换像素。

它接受以下选项：

阈值0

阈值1

阈值2

阈值3

限制每个平面的最大变化。范围是[0, 65535] ，默认值为65535 。如果为0 ，平面将保持不变。

坐标

指定要引用的像素的标志。范围是[0, 255]并且默认值为255 ，即使用所有八个像素。

x为中心的局部 3x3 坐标区域的标志：

1 2 3

4×5

6 7 8

* [例子](#Example-6)

#### 40.7.1 例子

* 应用扩张过滤器，阈值 0 设置为 30，阈值 1 设置为 40，阈值 2 设置为 50，坐标设置为 231，将输出的每个像素设置为像素之间的局部最大值：3x3 的 1、2、3、6、7、8输入中以它为中心的区域。如果输入像素和局部最大值之间的差异大于相应平面的阈值，则输出像素将设置为输入像素 + 相应平面的阈值。
* -i 输入 -vf "hwupload, dilation\_opencl=30:40:50:coordinates=231, hwdownload" 输出

### 40.8 nlmeans\_opencl

通过 OpenCL 进行非本地均值降噪过滤器，此过滤器接受与[nlmeans相同的选项](#nlmeans)。

### 40.9 overlay\_opencl

将一个视频叠加在另一个视频之上。

它需要两个输入并有一个输出。第一个输入是覆盖第二个输入的“主”视频。此过滤器要求所有输入具有相同的内存布局。因此，可能需要格式转换。

过滤器接受以下选项：

X

设置叠加视频在主视频上的 x 坐标。默认值为0 。

是

设置叠加视频在主视频上的 y 坐标。默认值为0 。

* [例子](#Examples-183)

#### 40.9.1 例子

* 在 INPUT 视频的左上角叠加图像 LOGO。两个输入都是 yuv420p 格式。
* -i 输入 -i 标志 -filter\_complex “[0:v]hwupload[a], [1:v]format=yuv420p, hwupload[b], [a][b]overlay\_opencl, hwdownload” 输出
* 输入具有相同的颜色通道内存布局，叠加层具有额外的 alpha 平面，如 INPUT 是 yuv420p，LOGO 是 yuva420p。
* -i 输入 -i 标志 -filter\_complex “[0:v]hwupload[a], [1:v]format=yuva420p, hwupload[b], [a][b]overlay\_opencl, hwdownload” 输出

### 40.10 pad\_opencl

向输入图像添加填充，并将原始输入放置在提供的x和y坐标处。

它接受以下选项：

宽度，w

身高，小时

为添加了填充的输出图像的大小指定一个表达式。如果宽度或高度的值为0，则相应的输入大小用于输出。

宽度表达式可以引用高度表达式设置的值，反之亦然。

宽度和高度的默认值为0。

X

是

相对于输出图像的上/左边框，指定将输入图像放置在填充区域内的偏移量。

x表达式可以引用y表达式设置的值，反之亦然。

x和y的默认值为0。

如果x或y的计算结果为负数，它们将被更改，因此输入图像位于填充区域的中心。

颜色

指定填充区域的颜色。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。

方面

填充到一个方面而不是一个分辨率。

width 、 height 、 x和y选项的值是包含以下常量的表达式：

in\_w

在\_h

输入视频的宽度和高度。

我

我

in\_w和in\_h相同。

out\_w

out\_h

和height表达式指定的输出宽度和高度（填充区域的大小） 。

噢

哦

out\_w和out\_h相同。

X

是

x 和 y 表达式指定的x和y偏移量，如果尚未指定，则为 NAN。

A

iw / ih相同

特区

输入样本纵横比

达尔

输入显示宽高比，同( iw / ih ) \* sar

### 40.11 prewitt\_opencl

将 Prewitt 运算符 ( <https://en.wikipedia.org/wiki/Prewitt_operator>) 应用于输入视频流。

过滤器接受以下选项：

飞机

设置要过滤的平面。默认值为0xf ，通过它处理所有平面。

规模

设置将与过滤结果相乘的值。范围是[0.0, 65535] ，默认值为1.0 。

三角洲

设置将添加到过滤结果的值。范围是[-65535, 65535] ，默认值为0.0 。

* [例子](#Example-7)

#### 40.11.1 示例

* 应用 Prewitt 运算符，比例设置为 2，增量设置为 10。
* -i 输入 -vf "hwupload, prewitt\_opencl=scale=2:delta=10, hwdownload" 输出

### 40.12 程序\_opencl

使用 OpenCL 程序过滤视频。

来源

OpenCL 程序源文件。

核心

程序中的内核名称。

输入

过滤器的输入数。默认为 1。

大小，小号

输出帧的大小。默认与第一个输入相同。

program\_opencl过滤器还支持[帧同步](#framesync)选项。

程序源文件必须包含一个具有给定名称的内核函数，它将为输出的每个平面运行一次。平面上的每次运行都会作为一个单独的 2D 全局 NDRange 排队，每个要生成的像素都有一个工作项。因此，每个工作项的全局 ID 偏移量是目标图像中像素的坐标。

内核函数需要采用以下参数：

* 目标图像， \_\_write\_only image2d\_t 。

该图像将成为输出；内核应该全部写入。

* 帧索引，无符号整数。

这是一个从零开始的计数器，每帧加一。

* 源图像， \_\_read\_only image2d\_t 。

这些是每个输入的最新图像。内核可以从它们中读取以生成输出，但不能写入它们。

示例程序：

* 将输入复制到输出（输出必须与输入大小相同）。
* \_\_kernel void copy(\_\_write\_only image2d\_t 目的地，
* 无符号整数索引，
* \_\_read\_only image2d\_t 来源）
* {
* const sampler\_t 采样器 = CLK\_NORMALIZED\_COORDS\_FALSE；
* int2 位置 = (int2)(get\_global\_id(0), get\_global\_id(1));
* float4 value = read\_imagef(source, sampler, location);
* write\_imagef（目的地，位置，价值）；
* }
* 应用一个简单的转换，将输入旋转一个随索引计数器增加的量。像素值由采样器线性插值，输出不需要与输入具有相同的维度。
* \_\_kernel void rotate\_image(\_\_write\_only image2d\_t dst,
* 无符号整数索引，
* \_\_read\_only image2d\_t 源）
* {
* const sampler\_t 采样器 = (CLK\_NORMALIZED\_COORDS\_FALSE |
* CLK\_FILTER\_LINEAR);
* 浮动角度 = (float)index / 100.0f;
* float2 dst\_dim = convert\_float2(get\_image\_dim(dst));
* float2 src\_dim = convert\_float2(get\_image\_dim(src));
* float2 dst\_cen = dst\_dim / 2.0f;
* float2 src\_cen = src\_dim / 2.0f;
* int2 dst\_loc = (int2)(get\_global\_id(0), get\_global\_id(1));
* float2 dst\_pos = convert\_float2(dst\_loc) - dst\_cen;
* float2 源位置 = {
* cos(角度) \* dst\_pos.x - sin(角度) \* dst\_pos.y,
* sin(角度) \* dst\_pos.x + cos(角度) \* dst\_pos.y
* };
* src\_pos = src\_pos \* src\_dim / dst\_dim;
* float2 src\_loc = src\_pos + src\_cen;
* 如果（src\_loc.x < 0.0f || src\_loc.y < 0.0f ||
* src\_loc.x > src\_dim.x || src\_loc.y > src\_dim.y)
* write\_imagef(dst, dst\_loc, 0.5f);
* 别的
* write\_imagef(dst, dst\_loc, read\_imagef(src, sampler, src\_loc));
* }
* 将两个输入混合在一起，每个输入的使用量随索引计数器而变化。
* \_\_kernel void blend\_images(\_\_write\_only image2d\_t dst,
* 无符号整数索引，
* \_\_read\_only image2d\_t src1,
* \_\_read\_only image2d\_t src2)
* {
* const sampler\_t 采样器 = (CLK\_NORMALIZED\_COORDS\_FALSE |
* CLK\_FILTER\_LINEAR);
* 浮动混合 = (cos((float)index / 50.0f) + 1.0f) / 2.0f;
* int2 dst\_loc = (int2)(get\_global\_id(0), get\_global\_id(1));
* int2 src1\_loc = dst\_loc \* get\_image\_dim(src1) / get\_image\_dim(dst);
* int2 src2\_loc = dst\_loc \* get\_image\_dim(src2) / get\_image\_dim(dst);
* float4 val1 = read\_imagef(src1, sampler, src1\_loc);
* float4 val2 = read\_imagef(src2, 采样器, src2\_loc);
* write\_imagef(dst, dst\_loc, val1 \* blend + val2 \* (1.0f - blend));
* }

### 40.13 重映射\_opencl

使用第二个重新映射像素：Xmap 和第三个：Ymap 输入视频流。

位置 (X, Y) 的目标像素将从源 (x, y) 位置选取，其中 x = Xmap(X, Y) 和 y = Ymap(X, Y)。如果映射值超出范围，则像素的零值将用于目标像素。

Xmap 和 Ymap 输入视频流的尺寸必须相同。输出视频流将具有 Xmap/Ymap 视频流维度。 Xmap 和 Ymap 输入视频流是 32 位浮点像素格式，单通道。

插话

指定用于重新映射像素的插值。允许的值是near和linear 。默认值为linear 。

充满

指定未映射像素的颜色。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。默认颜色为黑色。

### 40.14 罗伯茨\_opencl

将 Roberts 交叉运算符 ( <https://en.wikipedia.org/wiki/Roberts_cross>) 应用于输入视频流。

过滤器接受以下选项：

飞机

设置要过滤的平面。默认值为0xf ，通过它处理所有平面。

规模

设置将与过滤结果相乘的值。范围是[0.0, 65535] ，默认值为1.0 。

三角洲

设置将添加到过滤结果的值。范围是[-65535, 65535] ，默认值为0.0 。

* [例子](#Example-8)

#### 40.14.1 例子

* 应用 Roberts 交叉运算符，比例设置为 2，增量设置为 10
* -i 输入 -vf "hwupload, roberts\_opencl=scale=2:delta=10, hwdownload" 输出

### 40.15 sobel\_opencl

将 Sobel 运算符 ( <https://en.wikipedia.org/wiki/Sobel_operator>) 应用于输入视频流。

过滤器接受以下选项：

飞机

设置要过滤的平面。默认值为0xf ，通过它处理所有平面。

规模

设置将与过滤结果相乘的值。范围是[0.0, 65535] ，默认值为1.0 。

三角洲

设置将添加到过滤结果的值。范围是[-65535, 65535] ，默认值为0.0 。

* [例子](#Example-9)

#### 40.15.1 例子

* 应用比例设置为 2 且增量设置为 10 的 sobel 运算符
* -i 输入 -vf "hwupload, sobel\_opencl=scale=2:delta=10, hwdownload" 输出

### 40.16 色调映射\_opencl

使用色调映射执行 HDR(PQ/HLG) 到 SDR 的转换。

它接受以下参数：

色调图

指定要使用的色调映射运算符。与[tonemap](#tonemap)中的 tonemap 选项相同。

参数

调整色调映射算法。与[tonemap](#tonemap)中的参数选项相同。

去饱和

对超过此亮度级别的高光应用去饱和度。参数越高，保留的颜色信息越多。此设置通过（平滑地）变成白色来帮助防止超高光的颜色不自然地过曝。这使图像感觉更自然，但代价是减少了有关超出范围的颜色的信息。

默认值是0.5，这里的算法和目前cpu版本的tonemap有点不同。设置为 0.0 会禁用此选项。

临界点

色调映射算法参数根据每个场景进行微调。并使用阈值来检测场景是否发生了变化。如果当前帧平均亮度与当前运行平均值之间的距离超过阈值，我们将重新计算场景平均亮度和峰值亮度。默认值为 0.2。

格式

指定输出像素格式。

目前支持的格式有：

p010

nv12

范围，r

设置输出颜色范围。

可能的值是：

电视/mpeg

电脑/jpeg

默认与输入相同。

初选，p

设置输出颜色原色。

可能的值是：

BT709

BT2020

默认与输入相同。

转移，吨

设置输出传输特性。

可能的值是：

BT709

BT2020

默认为 bt709。

矩阵，米

设置输出色彩空间矩阵。

可能的值是：

BT709

BT2020

默认与输入相同。

* [例子](#Example-10)

#### 40.16.1 示例

* 使用线性运算符将 HDR(PQ/HLG) 视频转换为 bt2020-transfer-characteristic p010 格式。
* -i 输入 -vf "format=p010,hwupload,tonemap\_opencl=t=bt2020:tonemap=linear:format=p010,hwdownload,format=p010" 输出

### 40.17 unsharp\_opencl

锐化或模糊输入视频。

它接受以下参数：

luma\_msize\_x, lx

设置亮度矩阵水平大小。范围是[1, 23] ，默认值为5 。

luma\_msize\_y, ly

设置亮度矩阵垂直大小。范围是[1, 23] ，默认值为5 。

luma\_amount, la

设置亮度效果强度。范围是[-10, 10] ，默认值是1.0 。

负值将模糊输入视频，而正值将锐化它，零值将禁用效果。

chroma\_msize\_x, cx

设置色度矩阵水平大小。范围是[1, 23] ，默认值为5 。

chroma\_msize\_y, cy

设置色度矩阵垂直大小。范围是[1, 23] ，默认值为5 。

色度量，ca

设置色度效果强度。范围是[-10, 10] ，默认值是0.0 。

负值将模糊输入视频，而正值将锐化它，零值将禁用效果。

所有参数都是可选的，默认为字符串 '5:5:1.0:5:5:0.0' 的等价物。

* [例子](#Examples-184)

#### 40.17.1 例子

* 应用强大的亮度锐化效果：
* -i 输入 -vf "hwupload, unsharp\_opencl=luma\_msize\_x=7:luma\_msize\_y=7:luma\_amount=2.5, hwdownload" 输出
* 应用亮度和色度参数的强烈模糊：
* -i 输入 -vf "hwupload, unsharp\_opencl=7:7:-2:7:7:-2, hwdownload" 输出

### 40.18 xfade\_opencl

使用 OpenCL 交叉淡入淡出两个具有自定义过渡效果的视频。

它接受以下选项：

过渡

设置一种可能的过渡效果。

风俗

选择自定义过渡效果，实际的过渡描述将从源和内核选项中选取。

褪色

左擦

擦拭权

擦干净

擦拭

向左滑动

向右滑动

向上滑动

滑下

默认过渡是淡入淡出。

来源

用于自定义转换的 OpenCL 程序源文件。

核心

设置内核名称以用于从程序源文件自定义转换。

期间

设置视频过渡的持续时间。

抵消

设置相对于第一个视频的过渡开始时间。

程序源文件必须包含一个具有给定名称的内核函数，它将为输出的每个平面运行一次。平面上的每次运行都会作为一个单独的 2D 全局 NDRange 排队，每个要生成的像素都有一个工作项。因此，每个工作项的全局 ID 偏移量是目标图像中像素的坐标。

内核函数需要采用以下参数：

* 目标图像， \_\_write\_only image2d\_t 。

该图像将成为输出；内核应该全部写入。

* 第一个源图像， \_\_read\_only image2d\_t 。第二个源图像， \_\_read\_only image2d\_t 。

这些是每个输入的最新图像。内核可以从它们中读取以生成输出，但不能写入它们。

* 过渡进度，浮动。该值始终介于 0 和 1 之间（含 0 和 1）。

示例程序：

* 应用点幕过渡效果：
* \_\_kernel void blend\_images(\_\_write\_only image2d\_t dst,
* \_\_read\_only image2d\_t src1,
* \_\_read\_only image2d\_t src2,
* 浮动进度）
* {
* const sampler\_t 采样器 = (CLK\_NORMALIZED\_COORDS\_FALSE |
* CLK\_FILTER\_LINEAR);
* int2 p = (int2)(get\_global\_id(0), get\_global\_id(1));
* float2 rp = (float2)(get\_global\_id(0), get\_global\_id(1));
* float2 dim = (float2)(get\_image\_dim(src1).x, get\_image\_dim(src1).y);
* rp = rp /暗淡；
* float2 点 = (float2)(20.0, 20.0);
* float2 中心 = (float2)(0,0);
* float2 未使用；
* float4 val1 = read\_imagef(src1, sampler, p);
* float4 val2 = read\_imagef(src2, sampler, p);
* bool next = distance(fract(rp \* dots, &unused), (float2)(0.5, 0.5)) < (progress / distance(rp, center));
* write\_imagef(dst, p, next ? val1 : val2);
* }

## 41 个 VAAPI 视频过滤器

VAAPI 视频过滤器通常与 VAAPI 解码器和 VAAPI 编码器一起使用。下面是对 VAAPI 视频过滤器的描述。

--enable-vaapi配置 FFmpeg 。

要使用 vaapi 过滤器，您需要正确设置 vaapi 设备。更多信息，请阅读<https://trac.ffmpeg.org/wiki/Hardware/VAAPI>

* [overlay\_vaapi](#overlay_005fvaapi)
* [色调映射\_vaapi](#tonemap_005fvaapi)
* [hstack\_vaapi](#hstack_005fvaapi)
* [vstack\_vaapi](#vstack_005fvaapi)
* [xstack\_vaapi](#xstack_005fvaapi)

### 41.1 overlay\_vaapi

将一个视频叠加在另一个视频之上。

它需要两个输入并有一个输出。第一个输入是覆盖第二个输入的“主”视频。

过滤器接受以下选项：

X

是

设置主视频上叠加视频的 x 和 y 坐标表达式。

两个表达式的默认值为“0”。

w

H

设置主视频上叠加视频的宽度和高度的表达式。

默认值为“w”的“overlay\_iw”和“h”的“overlay\_ih\*w/overlay\_iw”。

表达式可以包含以下参数：

main\_w, W

main\_h, H

主要输入宽度和高度。

overlay\_iw

overlay\_ih

覆盖输入宽度和高度。

overlay\_w, w

overlay\_h, h

覆盖输出宽度和高度。

overlay\_x, x

覆盖\_y，y

覆盖层在 main 内部的位置

α

设置叠加视频的透明度。允许的范围是 0.0 到 1.0。更高的价值意味着更低的透明度。默认值为1.0 。

eof\_action

请参见[帧同步](#framesync)。

最短的

请参见[帧同步](#framesync)。

重复上次

请参见[帧同步](#framesync)。

此过滤器还支持[帧同步](#framesync)选项。

* [例子](#Examples-185)

#### 41.1.1 例子

* 在 INPUT 视频的左上角叠加图像 LOGO。这个过滤器的两个输入都是 yuv420p 格式。
* -i 输入 -i 标志 -filter\_complex “[0:v]hwupload[a], [1:v]format=yuv420p, hwupload[b], [a][b]overlay\_vaapi” 输出
* 在距 INPUT 视频左上角的偏移量 (200, 100) 处叠加图像徽标。输入具有相同的颜色通道内存布局，叠加层具有额外的 alpha 平面，例如 INPUT 是 yuv420p，而 LOGO 是 yuva420p。
* -i INPUT -i LOGO -filter\_complex "[0:v]hwupload[a], [1:v]format=yuva420p, hwupload[b], [a][b]overlay\_vaapi=x=200:y=100:w =400:h=300:alpha=1.0, hwdownload, format=nv12" 输出

### 41.2 色调映射\_vaapi

使用色调映射执行 HDR（高动态范围）到 SDR（标准动态范围）的转换。它将 HDR10 内容的动态范围映射到 SDR 内容。它目前只接受 HDR10 作为输入。

它接受以下参数：

格式

指定输出像素格式。

目前支持的格式有：

p010

nv12

默认为 nv12。

初选，p

设置输出颜色原色。

默认与输入相同。

转移，吨

设置输出传输特性。

默认为 bt709。

矩阵，米

设置输出色彩空间矩阵。

默认与输入相同。

* [例子](#Example-11)

#### 41.2.1 示例

* 将HDR(HDR10)视频转换为bt2020-transfer-characteristic p010格式
* tonemap\_vaapi=格式=p010:t=bt2020-10

### 41.3 hstack\_vaapi

水平堆叠输入视频。

[hstack过滤器](#hstack)的 VA-API 变体，每个输入流可能具有不同的高度，此过滤器将缩小/放大每个输入流，同时保持原始方面。

它接受以下选项：

输入

请参阅[堆栈](#hstack)。

最短的

请参阅[堆栈](#hstack)。

高度

设置输出高度。如果设置为 0，此过滤器会将输出高度设置为第一个输入流的高度。默认值为 0。

### 41.4 vstack\_vaapi

垂直堆叠输入视频。

[vstack过滤器](#vstack)的 VA-API 变体，每个输入流可能具有不同的宽度，此过滤器将缩小/放大每个输入流，同时保持原始方面。

它接受以下选项：

输入

请参见[vstack](#vstack) 。

最短的

请参见[vstack](#vstack) 。

宽度

设置输出宽度。如果设置为 0，此过滤器会将输出宽度设置为第一个输入流的宽度。默认值为 0。

### 41.5 xstack\_vaapi

将视频输入堆叠到自定义布局中。

[xstack过滤器](#xstack)的 VA-API 变体，每个输入流可能有不同的大小，此过滤器会将每个输入流缩小/放大到给定的输出大小，或第一个输入流的大小。

它接受以下选项：

输入

请参见[xstack](#xstack) 。

最短的

请参见[xstack](#xstack) 。

布局

请参见[xstack](#xstack) 。此外，这允许用户为每个输入流提供输出大小。

xstack\_vaapi=输入=4:布局=0\_0\_1920x1080|0\_h0\_1920x1080|w0\_0\_1920x1080|w0\_h0\_1920x1080

网格

请参见[xstack](#xstack) 。

grid\_tile\_size

设置网格时为每个输入流设置输出大小。如果未设置此选项，此过滤器将默认将输出大小设置为第一个输入流的大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。

充满

请参见[xstack](#xstack) 。

## 42 个 QSV 视频滤波器

下面是对当前可用的 QSV 视频过滤器的描述。

--enable-libmfx或--enable-libvpl配置 FFmpeg 。

要使用 QSV 过滤器，您需要正确设置 QSV 设备。更多信息，请阅读<https://trac.ffmpeg.org/wiki/Hardware/QuickSync>

* [hstack\_qsv](#hstack_005fqsv)
* [vstack\_qsv](#vstack_005fqsv)
* [xstack\_qsv](#xstack_005fqsv)

### 42.1 hstack\_qsv

水平堆叠输入视频。

[hstack过滤器](#hstack)的 QSV 变体，每个输入流可能具有不同的高度，此过滤器将缩小/放大每个输入流，同时保持原始方面。

它接受以下选项：

输入

请参阅[堆栈](#hstack)。

最短的

请参阅[堆栈](#hstack)。

高度

设置输出高度。如果设置为 0，此过滤器会将输出高度设置为第一个输入流的高度。默认值为 0。

### 42.2 vstack\_qsv

垂直堆叠输入视频。

[vstack过滤器](#vstack)的 QSV 变体，每个输入流可能具有不同的宽度，此过滤器将缩小/放大每个输入流，同时保持原始方面。

它接受以下选项：

输入

请参见[vstack](#vstack) 。

最短的

请参见[vstack](#vstack) 。

宽度

设置输出宽度。如果设置为 0，此过滤器会将输出宽度设置为第一个输入流的宽度。默认值为 0。

### 42.3 xstack\_qsv

将视频输入堆叠到自定义布局中。

[xstack过滤器](#xstack)的 QSV 变体。

它接受以下选项：

输入

请参见[xstack](#xstack) 。

最短的

请参见[xstack](#xstack) 。

布局

请参见[xstack](#xstack) 。此外，这允许用户为每个输入流提供输出大小。

xstack\_qsv=输入=4:布局=0\_0\_1920x1080|0\_h0\_1920x1080|w0\_0\_1920x1080|w0\_h0\_1920x1080

网格

请参见[xstack](#xstack) 。

grid\_tile\_size

设置网格时为每个输入流设置输出大小。如果未设置此选项，此过滤器将默认将输出大小设置为第一个输入流的大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。

充满

请参见[xstack](#xstack) 。

## 43个视频源

以下是对当前可用视频源的描述。

* [缓冲](#buffer)
* [大提琴](#cellauto)
* [核心图像资源](#coreimagesrc-1)
* [抢夺](#ddagrab)
* [梯度](#gradients)
* [曼德尔布罗](#mandelbrot)
* [mptestsrc](#mptestsrc)
* [frei0r\_src](#frei0r_005fsrc)
* [生活](#life)
* [allrgb, allyuv, 颜色, colorchart, colorspectrum, haldclutsrc, nullsrc, pal75bars, pal100bars, rgbtestsrc, smptebars, smptehdbars, testsrc, testsrc2, yuvtestsrc](#allrgb_002c-allyuv_002c-color_002c-colo)
* [openclsrc](#openclsrc)
* [谢尔宾斯基](#sierpinski)

### 43.1缓冲区

缓冲视频帧，并使它们可用于过滤器链。

libavfilter/buffersrc.h中定义的接口。

它接受以下参数：

视频大小

指定缓冲视频帧的大小（宽度和高度）。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。

宽度

输入视频宽度。

高度

输入视频高度。

pix\_fmt

表示缓冲视频帧的像素格式的字符串。可以是像素格式对应的数字，也可以是像素格式名称。

时基

指定缓冲帧的时间戳假定的时基。

帧率

指定视频流的预期帧速率。

像素纵横比，sar

输入视频的样本（像素）纵横比。

hw\_frames\_ctx

当使用硬件像素格式时，这应该是对描述输入帧的 AVHWFramesContext 的引用。

例如：

缓冲区=宽度=320:高度=240:pix\_fmt=yuv410p:time\_base=1/24:sar=1

将指示源接受大小为 320x240 且格式为“yuv410p”的视频帧，假设 1/24 作为时间戳时基和方形像素（1:1 样本纵横比）。由于名称为“yuv410p”的像素格式对应于数字 6（检查libavutil/pixfmt.h中的枚举 AVPixelFormat 定义），此示例对应于：

缓冲区=大小=320x240:pixfmt=6:time\_base=1/24:pixel\_aspect=1/1

或者，可以将选项指定为平面字符串，但不推荐使用此语法：

宽度：高度： pix\_fmt ： time\_base.num ： time\_base.den ： pixel\_aspect.num ： pixel\_aspect.den

### 43.2 大提琴

创建由基本元胞自动机生成的模式。

元胞自动机的初始状态可以通过文件名和模式选项来定义。如果未指定此类选项，则会随机创建初始状态。

在每个新帧中，视频中的新行都会填充下一代元胞自动机的结果。整个框架被填满时的行为由滚动选项定义。

此来源接受以下选项：

文件名，f

从指定文件中读取初始元胞自动机状态，即起始行。在文件中，每个非空白字符都被视为一个活动单元格，换行符将终止该行，文件中的其他字符将被忽略。

模式，p

从指定字符串中读取初始元胞自动机状态，即起始行。

字符串中的每个非空白字符都被视为一个活动单元格，换行符将终止该行，字符串中的其他字符将被忽略。

率，r

设置视频速率，即每秒生成的帧数。默认值为 25。

random\_fill\_ratio, 比率

设置初始元胞自动机行的随机填充率。是一个浮点数，取值范围为 0 到 1，默认为 1/PHI。

指定文件或模式时忽略此选项。

随机种子，种子

设置用于随机填充初始行的种子，必须是介于 0 和 UINT32\_MAX 之间的整数。如果未指定，或者如果明确设置为 -1，过滤器将尝试在尽力而为的基础上使用良好的随机种子。

规则

设置元胞自动机规则，取值范围为0~255，默认为110。

大小，小号

设置输出视频的大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。

如果指定文件名或模式，则大小默认设置为指定初始状态行的宽度，高度设置为宽度\* PHI。

如果设置了大小，它必须包含指定模式字符串的宽度，并且指定模式将在较大的行中居中。

如果未指定文件名或模式字符串，则大小值默认为“320x518”（用于随机生成的初始状态）。

滚动

如果设置为 1，则当输出中的所有行都已填满时向上滚动输出。如果设置为 0，新生成的行将在底部行填充后立即覆盖顶部行。默认为 1。

start\_full, 满

如果设置为 1，则在输出第一帧之前用生成的行完全填充输出。这是默认行为，用于禁用将值设置为 0。

缝

如果设置为 1，则将左右行边缘缝合在一起。这是默认行为，用于禁用将值设置为 0。

* [例子](#Examples-186)

#### 43.2.1 例子

* pattern中读取初始状态，并指定大小为 200x400 的输出。
* cellauto=f=图案：s=200x400
* 生成一个随机初始行，宽度为 200 个单元格，填充率为 2/3：
* cellauto=比率=2/3:s=200x200
* 创建一个由规则 18 生成的模式，该模式从以宽度为 100 的初始行为中心的单个活动单元格开始：
* cellauto=p=@:s=100x400:full=0:rule=18
* 指定一个更详细的初始模式：
* cellauto=p='@@@@@':s=100x400:full=0:rule=18

### 43.3 coreimagesrc

在 OSX 上使用 Apple 的 CoreImage API 在 GPU 上生成的视频源。

[coreimage视频过滤器](#coreimage)的特殊版本。在应用的过滤器链的开头使用核心图像生成器来生成内容。

coreimagesrc 视频源接受以下选项：

列表生成器

列出所有可用的生成器及其各自的选项以及可能的最小值和最大值以及默认值。

list\_generators=真

大小，小号

指定源视频的大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为320x240 。

率，r

指定源视频的帧速率，作为每秒生成的帧数。它必须是frame\_rate\_num / frame\_rate\_den格式的字符串、整数、浮点数或有效的视频帧率缩写。默认值为“25”。

特区

设置源视频的样本纵横比。

持续时间，d

设置源视频的持续时间。有关可接受的语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)

如果未指定，或者表示的持续时间为负数，则视频应该永远生成。

此外，接受[coreimage视频过滤器的所有选项。](#coreimage)完整的过滤器链可用于进一步处理生成的输入，而无需 CPU-HOST 传输。有关详细信息，请参阅[coreimage](#coreimage)文档和示例。

* [例子](#Examples-187)

#### 43.3.1 例子

* 使用 CIQRCodeGenerator 为 FFmpeg 主页创建 QR 代码，作为 Apple 标准 bash shell 的完整转义命令行给出：
* ffmpeg -f lavfi -i coreimagesrc=s=100x100:filter=CIQRCodeGenerator@inputMessage=https\\\\\://FFmpeg.org/@inputCorrectionLevel=H -frames:v 1 QRCode.png

这个例子相当于[coreimage的QRCode例子](#coreimage)，不需要nullsrc视频源。

### 43.4 ddagrab

通过桌面复制 API 捕获 Windows 桌面。

该过滤器专门返回 D3D11 硬件帧，用于 on-gpu 编码或处理。因此，任何类型的软件处理都需要显式的[hwdownload 。](#hwdownload)

它接受以下选项：

输出\_idx

要捕获的 DXGI 输出索引。

通常对应于 Windows 给屏幕的索引减一，所以它从 0 开始。

默认输出 0。

画鼠标

是否绘制鼠标光标。

默认为真。

仅影响硬件游标。如果游戏或应用程序呈现自己的光标，它总是会被捕获。

帧率

捕获桌面的帧率。

默认为 30 FPS。

视频大小

指定捕获视频的大小。

默认为屏幕的完整尺寸。

如果小于屏幕尺寸，则从底部/右侧裁剪。

偏移量\_x

捕获的视频的水平偏移。

偏移量\_y

捕获视频的垂直偏移。

output\_fmt

所需的过滤器输出格式。默认为 8 位 BGRA。

它接受以下值：

'汽车'

将所有支持的输出格式传递给 DDA 并返回 DDA 决定使用的格式。

' 8位'

' bgra '

8 位格式始终有效，如果需要，DDA 将转换为它们。

'10位'

' x2bgr10 '

如果请求 10 位格式但不可用，过滤器初始化将失败。

* [例子](#Examples-188)

#### 43.4.1 例子

捕获主屏幕并使用 nvenc 进行编码：

ffmpeg -f lavfi -i ddagrab -c:v h264\_nvenc -cq 18 输出.mp4

也可以跳过lavfi设备，直接使用filter。还演示了使用 libx264 下载帧和编码。在这种情况下需要明确的输出格式规范：

ffmpeg -filter\_complex ddagrab=output\_idx=1:framerate=60,hwdownload,format=bgra -c:v libx264 -crf 18 output.mp4

如果只想捕获桌面的一部分，可以通过指定较小的尺寸及其在屏幕中的偏移量来实现：

ddagrab=video\_size=800x600:offset\_x=100:offset\_y=100

### 43.5 渐变

生成几个梯度。

大小，小号

设置帧大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为“640x480”。

率，r

设置帧速率，以每秒帧数表示。默认值为“25”。

c0, c1, c2, c3, c4, c5, c6, c7

设置 8 种颜色。颜色的默认值是随机选择一种。

x0, y0, y0, y1

设置渐变线源点和目标点。如果为负数或超出范围，则随机选择。

nb\_colors, n

设置一次使用的颜色数量。允许的范围是 2 到 8。默认值为 2。

种子

设置用于拾取渐变线点的种子。

持续时间，d

设置源视频的持续时间。有关可接受的语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)

如果未指定，或者表示的持续时间为负数，则视频应该永远生成。

速度

设置渐变旋转的速度。

类型，吨

设置渐变类型，可以是linear或radial或circular或spiral 。

### 43.6 曼德尔布罗

start\_x和start\_y指定的点。

此来源接受以下选项：

结束点

设置终端点值。默认值为 400。

end\_scale

设置终端刻度值。必须是浮点值。默认值为 0.3。

内

设置内部着色模式，即用于绘制 Mandelbrot 分形内部区域的算法。

它应采用以下值之一：

黑色的

设置黑色模式。

收敛

显示收敛前的时间。

民科

根据最接近迭代原点的点设置颜色。

时期

设置周期模式。

默认值为mincol 。

纾困

设置救助值。默认值为 10.0。

最大化

设置渲染算法执行的最大迭代次数。默认值为 7189。

外

设置外部着色模式。它应采用以下值之一：

迭代次数

设置迭代计数模式。

normalized\_iteration\_count

设置归一化迭代计数模式。

默认值为normalized\_iteration\_count 。

率，r

设置帧速率，以每秒帧数表示。默认值为“25”。

大小，小号

设置帧大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为“640x480”。

起始比例

设置初始比例值。默认值为 3.0。

开始\_x

设置初始 x 位置。必须是介于 -100 和 100 之间的浮点值。默认值为 -0.743643887037158704752191506114774。

开始\_y

设置初始 y 位置。必须是介于 -100 和 100 之间的浮点值。默认值为 -0.131825904205311970493132056385139。

### 43.7 mptestsrc

生成各种测试模式，由 MPlayer 测试过滤器生成。

生成的视频大小是固定的，为256x256。此源特别适用于测试编码功能。

此来源接受以下选项：

率，r

指定源视频的帧速率，作为每秒生成的帧数。它必须是frame\_rate\_num / frame\_rate\_den格式的字符串、整数、浮点数或有效的视频帧率缩写。默认值为“25”。

持续时间，d

设置源视频的持续时间。有关可接受的语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)

如果未指定，或者表示的持续时间为负数，则视频应该永远生成。

测试，吨

设置要执行的测试的编号或名称。支持的测试有：

直流亮度

直流色度

频率亮度

频率色度

amp\_luma

amp\_chroma

海关总署

MV

环1

环2

全部

最大帧数，m

设置每次测试生成的最大帧数，默认值为 30。

默认值为“全部”，这将循环遍历所有测试的列表。

一些例子：

mptestsrc=t=dc\_luma

将生成一个“dc\_luma”测试模式。

### 43.8 frei0r\_src

提供一个免费资源。

要启用此过滤器的编译，您需要安装 frei0r 标头并使用--enable-frei0r配置 FFmpeg 。

此源接受以下参数：

尺寸

要生成的视频的大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。

帧率

生成视频的帧率。它可以是num / den形式的字符串或帧率缩写。

过滤器名称

要加载的 frei0r 源的名称。有关 frei0r 以及如何设置参数的更多信息，请阅读视频过滤器文档中的[frei0r部分。](#frei0r)

过滤参数

要传递给 frei0r 源的以“|”分隔的参数列表。

例如，要生成一个大小为 200x200 且帧速率为 10 的 frei0r partik0l 源，它覆盖在叠加过滤器的主输入上：

frei0r\_src=size=200x200:framerate=10:filter\_name=partik0l:filter\_params=1234 [覆盖]; [in][overlay] 覆盖

### 43.9生命

产生一种生活模式。

这个来源是基于约翰康威的人生游戏的概括。

源输入代表一个生命网格，每个像素代表一个细胞，该细胞可以处于两种可能状态之一，存活或死亡。每个细胞与其八个邻居相互作用，这些邻居是水平、垂直或对角线相邻的细胞。

在每次交互中，网格都会根据采用的规则进行演化，该规则指定了使一个细胞存活或诞生的相邻活细胞的数量。规则选项允许指定要采用的规则。

此来源接受以下选项：

文件名，f

设置从中读取初始网格状态的文件。在文件中，每个非空白字符都被视为一个活动单元格，换行符用于分隔每一行的末尾。

如果未指定此选项，则随机生成初始网格。

率，r

设置视频速率，即每秒生成的帧数。默认值为 25。

random\_fill\_ratio, 比率

设置初始随机网格的随机填充率。是一个浮点数，取值范围为 0 到 1，默认为 1/PHI。指定文件时将忽略它。

随机种子，种子

设置用于填充初始随机网格的种子，必须是介于 0 和 UINT32\_MAX 之间的整数。如果未指定，或者如果明确设置为 -1，过滤器将尝试在尽力而为的基础上使用良好的随机种子。

规则

制定生活规则。

NS /B NB ”类型的代码指定规则，其中NS和NB是 0-8 范围内的数字序列， NS指定使活细胞保持存活的存活邻居细胞的数量，和NB使死细胞变成活细胞（即“出生”）的活邻居细胞的数量。 “s”和“b”可以分别代替“S”和“B”。

或者，可以通过 18 位整数指定规则。 9 个高位用于编码下一个细胞状态，如果它对于每个相邻的活细胞数量都是存活的，低位指定“出生”新细胞的规则。更高阶位编码更多数量的相邻小区。例如，数字 6153 = (12<<9)+9指定生存规则为 12，出生规则为 9，对应于“S23/B03”。

默认值为“S23/B3”，这是最初的康威生命游戏规则，如果一个细胞有 2 个或 3 个相邻的活细胞，它将使一个细胞存活，如果一个死细胞周围有 3 个活细胞，则会产生一个新细胞细胞。

大小，小号

设置输出视频的大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。

如果指定了文件名，则默认情况下将大小设置为与输入文件相同的大小。如果设置了大小，它必须包含输入文件中指定的大小，并且该文件中定义的初始网格位于较大结果区域的中心。

如果未指定文件名，则大小值默认为“320x240”（用于随机生成的初始网格）。

缝

如果设置为 1，则将左右网格边缘缝合在一起，顶部和底部边缘也缝合在一起。默认为 1。

模具

设置单元模具速度。如果设置，死细胞将从death\_color变为mold\_color ，步长为mold 。 mold 的值可以从 0 到 255。

生活色彩

设置活（或新生）细胞的颜色。

死亡颜色

设置死细胞的颜色。如果设置了模具，这是用于表示死细胞的第一种颜色。

模具颜色

设置模具颜色，用于绝对死细胞和发霉细胞。

对于这 3 个颜色选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。

* [例子](#Examples-189)

#### 43.9.1 例子

* pattern中读取一个网格，并将其置于 300x300 像素大小的网格的中心：
* 生活=f=图案：s=300x300
* 生成一个大小为 200x200 的随机网格，填充率为 2/3：
* 寿命=比率=2/3:s=200x200
* 指定用于演化随机生成的网格的自定义规则：
* 生活=规则=S14/B34
* ffplay 的缓慢死亡效果（霉菌）的完整示例：
* ffplay -f lavfi life=s=300x200:mold=10:r=60:ratio=0.1:death\_color=#C83232:life\_color=#00ff00,scale=1200:800:flags=16

### 43.10 allrgb、allyuv、颜色、色图、色谱、haldclutsrc、nullsrc、pal75bars、pal100bars、rgbtestsrc、smptebars、smptehdbars、testsrc、testsrc2、yuvtestsrc

allrgb源返回所有 rgb 颜色的大小为 4096x4096 的帧。

allyuv源返回所有 yuv 颜色的大小为 4096x4096 的帧。

颜色源提供统一颜色的输入。

colorchart源提供了一个颜色检查图表。

colorspectrum源提供色谱输入。

haldclutsrc源提供了一个标识 Hald CLUT。另请参见[haldclut](#haldclut)过滤器。

nullsrc源返回未处理的视频帧。它主要用于分析/调试工具，或作为忽略输入数据的过滤器的来源。

pal75bars源根据具有 75% 色阶的 EBU PAL 建议生成彩条图案。

pal100bars源根据 100% 色阶的 EBU PAL 建议生成彩条图案。

rgbtestsrc源生成一个 RGB 测试模式，可用于检测 RGB 与 BGR 问题。您应该看到从上到下的红色、绿色和蓝色条纹。

smptebars源根据 SMPTE 工程指南 EG 1-1990 生成彩条图案。

smptehdbars源基于 SMPTE RP 219-2002 生成彩条图案。

testsrc源生成一个测试视频模式，显示颜色模式、滚动渐变和时间戳。这主要用于测试目的。

testsrc2源类似于 testsrc，但支持更多像素格式而不仅仅是rgb24 。这允许将其用作其他测试的输入而无需格式转换。

yuvtestsrc源代码生成一个 YUV 测试模式。您应该看到 ay、cb 和 cr 从上到下呈条纹状。

来源接受以下参数：

等级

指定 Hald CLUT 的级别，仅在haldclutsrc源中可用。 N级生成N\*N\*N × N\*N\*N像素的图片，用作 3D 查找表的单位矩阵。每个组件都以1/(N\*N) 的比例编码。

颜色，c

指定源的颜色，仅在颜色源中可用。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)"Color" 部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#color-syntax)。

大小，小号

指定源视频的大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为320x240 。

此选项不适用于allrgb 、 allyuv和haldclutsrc过滤器。

率，r

指定源视频的帧速率，作为每秒生成的帧数。它必须是frame\_rate\_num / frame\_rate\_den格式的字符串、整数、浮点数或有效的视频帧率缩写。默认值为“25”。

持续时间，d

设置源视频的持续时间。有关可接受的语法，请参阅[ffmpeg-utils(1) 手册中的 (ffmpeg-utils) 持续时间部分。](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#time-duration-syntax)

如果未指定，或者表示的持续时间为负数，则视频应该永远生成。

由于帧速率用作时基，因此包括最后一帧在内的所有帧都将具有完整的持续时间。如果指定的持续时间不是帧持续时间的倍数，则会向上舍入。

特区

设置源视频的样本纵横比。

α

指定背景的 alpha（不透明度），仅在testsrc2源中可用。该值必须介于 0（完全透明）和 255（完全不透明，默认值）之间。

小数，n

设置时间戳中显示的小数位数，仅在testsrc源中可用。

显示的时间戳值将对应于原始时间戳值乘以指定值的 10 次方。默认值为 0。

类型

设置色谱的类型，仅在色谱源中可用。可以是以下之一：

'黑色'

'白色'

'全部'

补丁大小

设置单个色块的色块大小，仅在色卡源中可用。默认为64x64 。

预设

设置 colorchecker 颜色预设，仅在颜色图表源中可用。

可用值是：

'参考'

'肤色'

默认值为参考。

* [例子](#Examples-190)
* [命令](#Commands-156)

#### 43.10.1 例子

* 生成时长为 5.3 秒、大小为 176x144 且帧速率为每秒 10 帧的视频：
* testsrc=duration=5.3:size=qcif:rate=10
* 下面的图形描述将生成一个红色源，不透明度为 0.2，大小为“qcif”，帧速率为每秒 10 帧：
* 颜色=c=红色@0.2:s=qcif:r=10
* 如果要忽略输入内容，可以使用nullsrc 。以下命令通过使用geq过滤器在亮度平面中生成噪声：
* nullsrc=s=256x256，geq=随机（1）\*255:128:128

#### 43.10.2 命令

颜色源支持以下命令：

三、颜色

设置创建图像的颜色。接受相应颜色选项的相同语法。

### 43.11 openclsrc

使用 OpenCL 程序生成视频。

来源

OpenCL 程序源文件。

核心

程序中的内核名称。

大小，小号

要生成的帧的大小。这必须设置。

格式

用于生成的帧的像素格式。这必须设置。

率，r

每秒生成的帧数。默认值为“25”。

有关程序加载工作原理的详细信息，请参阅[program\_opencl](#program_005fopencl)过滤器。

示例程序：

* 通过设置输出图像中像素位置的像素值来生成色带。 （请注意，这将适用于所有像素格式，但生成的输出将不相同。）
* \_\_kernel void ramp(\_\_write\_only image2d\_t dst,
* 无符号整数索引）
* {
* int2 loc = (int2)(get\_global\_id(0), get\_global\_id(1));
* float4 值；
* val.xy = val.zw = convert\_float2(loc) / convert\_float2(get\_image\_dim(dst));
* write\_imagef(dst, loc, val);
* }
* 生成 Sierpinski 地毯图案，每帧平移一个像素。
* \_\_kernel void sierpinski\_carpet(\_\_write\_only image2d\_t dst,
* 无符号整数索引）
* {
* int2 loc = (int2)(get\_global\_id(0), get\_global\_id(1));
* float4 值 = 0.0f；
* int x = loc.x + 索引；
* int y = loc.y + 索引；
* 而 (x > 0 || y > 0) {
* 如果（x % 3 == 1 && y % 3 == 1）{
* 值 = 1.0f；
* 休息;
* }
* x /= 3;
* y /= 3;
* }
* write\_imagef(dst, loc, value);
* }

### 43.12谢尔宾斯基

生成 Sierpinski 地毯/三角形分形，并随机平移。

此来源接受以下选项：

大小，小号

设置帧大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为“640x480”。

率，r

设置帧速率，以每秒帧数表示。默认值为“25”。

种子

设置用于随机平移的种子。

跳

为单个平移目的地设置最大跳跃。允许的范围是 1 到 10000。

类型

设置分形类型，可以默认为carpet或triangle 。

## 44个视频接收器

以下是对当前可用视频接收器的描述。

* [缓冲槽](#buffersink)
* [空槽](#nullsink)

### 44.1 缓冲区接收器

缓冲视频帧，并使它们可用于过滤器图的末尾。

libavfilter/buffersink.h或选项系统中定义的接口。

它接受指向 AVBufferSinkContext 结构的指针，该结构定义传入缓冲区的格式，作为不透明参数传递给avfilter\_init\_filter进行初始化。

### 44.2 空槽

空视频接收器：对输入视频绝对不做任何事情。它主要用作模板和用于分析/调试工具。

## 45个多媒体过滤器

下面是对当前可用的多媒体过滤器的描述。

* [a3dscope](#a3dscope)
* [升位范围](#abitscope)
* [绘图](#adrawgraph)
* [图形监视器](#agraphmonitor)
* [直方图](#ahistogram)
* [相位计](#aphasemeter)
* [矢量图](#avectorscope)
* [长凳，长凳](#bench_002c-abench)
* [连接](#concat-3)
* [ebur128](#ebur128-1)
* [交错，交错](#interleave_002c-ainterleave)
* [潜伏期，潜伏期](#latency_002c-alatency)
* [元数据，元数据](#metadata_002c-ametadata)
* [烫发，烫发](#perms_002c-aperms)
* [实时的，实时的](#realtime_002c-arealtime)
* [段，段](#segment_002c-asegment)
* [选择，选择](#select_002c-aselect)
* [发送命令，发送命令](#sendcmd_002c-asendcmd)
* [setpts, asetpts](#setpts_002c-asetpts)
* [设定范围](#setrange)
* [setb, asettb](#settb_002c-asettb)
* [显示cqt](#showcqt)
* [showcwt](#showcwt)
* [显示频率](#showfreqs)
* [显示空间](#showspatial)
* [显示频谱](#showspectrum-1)
* [显示频谱](#showspectrumpic)
* [显示量](#showvolume)
* [秀波](#showwaves)
* [显示波浪](#showwavespic)
* [边数据，边数据](#sidedata_002c-asidedata)
* [光谱合成器](#spectrumsynth)
* [分裂，分裂](#split_002c-asplit)
* [zmq，azmq](#zmq_002c-azmq)

### 45.1 a3dscope

将输入音频转换为 3d 范围视频输出。

过滤器接受以下选项：

率，r

设置帧速率，以每秒帧数表示。默认值为“25”。

大小，小号

指定输出的视频大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为hd720 。

视野

设置相机视野。默认为 90 度。允许的范围是 40 到 150。

卷

设置相机胶卷。

沥青

设置相机间距。

偏航

设置相机偏航。

放大

在 X 轴上设置相机变焦。

缩放

在 Y 轴上设置相机变焦。

zzoom

在 Z 轴上设置相机变焦。

xpos

在 X 轴上设置相机位置。

错别字

在 Y 轴上设置相机位置。

zpos

在 Z 轴上设置相机位置。

长度

以帧数设置显示音频波的长度。

* [命令](#Commands-157)

#### 45.1.1 命令

Filter 支持上述一些选项作为[命令](#commands)。

### 45.2 位域

将输入音频转换为视频输出，显示音频位范围。

过滤器接受以下选项：

率，r

设置帧速率，以每秒帧数表示。默认值为“25”。

大小，小号

指定输出的视频大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为1024x256 。

颜色

指定由空格或“|”分隔的颜色列表它将用于绘制通道。无法识别或丢失的颜色将被白色替换。

模式，米

设置输出模式。可以是bars或trace 。默认为bars 。

### 45.3 绘图

使用输入音频元数据绘制图形。

见[图纸](#drawgraph)

### 45.4 图形监视器

请参阅[图形监视器](#graphmonitor)。

### 45.5 直方图

将输入音频转换为视频输出，显示音量直方图。

过滤器接受以下选项：

模式

指定直方图的计算方式。

它接受以下值：

'单身'

对所有通道使用单一直方图。

'分开'

为每个通道使用单独的直方图。

默认为single 。

率，r

设置帧速率，以每秒帧数表示。默认值为“25”。

大小，小号

指定输出的视频大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为hd720 。

规模

设置显示比例。

它接受以下值：

'日志'

对数的

'开方'

平方根

' cbrt '

立方根

'临'

线性的

' rlog '

逆对数

默认为日志。

规模

设置振幅刻度。

它接受以下值：

'日志'

对数的

'临'

线性的

默认为日志。

帐户

设置在直方图中累积多少帧。默认值为 1。将此设置为 -1 会累积所有帧。

高度

设置窗口高度的直方图比率。

滑动

设置超声图滑动。

它接受以下值：

'替换'

用新行替换旧行。

'滚动'

从上到下滚动。

默认是替换。

模式

设置直方图模式。

它接受以下值：

'腹肌'

使用样本的绝对值。

'标志'

使用未触及的样本值。

默认为abs 。

### 45.6相位计

测量输入音频的相位，导出为元数据lavfi.aphasemeter.phase ，表示当前音频帧的平均相位。还可以生成视频输出并默认启用。音频作为第一个输出传递。

如果音频具有不同的通道布局，它将被重新矩阵化为立体声。相位值在[-1, 1]范围内，其中-1表示左右声道完全异相， 1表示声道同相。

过滤器接受以下选项，所有选项都与其视频输出相关：

率，r

设置输出帧率。默认值为25 。

大小，小号

设置输出的视频大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为800x400 。

遥控

GC

公元前

指定红色、绿色、蓝色对比度。默认值为2 、 7和1 。允许的范围是[0, 255] 。

mpc

设置将用于绘制中相的颜色。如果颜色是默认的none ，则不会绘制中间相位值。

视频

启用视频输出。默认启用。

* [相位检测](#phasing-detection)
* [例子](#Examples-191)

#### 45.6.1 相位检测

该过滤器还检测立体声流中的异相和单声道序列。它记录序列开始、结束和持续时间更长或与最小设置一样长的持续时间。

过滤器接受以下检测选项：

定相

启用单声道和异相检测。默认是禁用的。

公差，吨

设置单声道检测的相位公差，以幅度比表示。默认为0 。允许的范围是[0, 1] 。

角度，一个

设置异相检测的角度阈值，以度为单位。默认值为170 。允许的范围是[90, 180] 。

持续时间，d

设置单声道或异相持续时间直到收到通知，以秒为单位。默认值为2 。

#### 45.6.2 例子

* ffmpeg检测具有 0.001 相位容差的 1 秒单声道的完整示例：
* ffmpeg -i stereo.wav -af aphasemeter=video=0:phasing=1:duration=1:tolerance=0.001 -f null -

### 45.7 矢量图

将输入音频转换为视频输出，表示音频矢量范围。

该滤波器用于测量立体声音频流通道之间的差异。由相同的左右信号组成的单声道信号会产生直线垂直线。任何立体声分离都可以看作是偏离这条线，从而形成李萨如图形。如果出现直线（或偏离直线）但水平的线，则表示左右声道异相。

过滤器接受以下选项：

模式，米

设置矢量示波器模式。

可用值是：

'利沙育'

Lissajous 旋转了 45 度。

' lissajous\_xy '

同上但不旋转。

'极地'

形状类似半圆。

默认值为 ' lissajous '。

大小，小号

设置输出的视频大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为400x400 。

率，r

设置输出帧率。默认值为25 。

遥控

GC

公元前

交流电

指定红色、绿色、蓝色和 alpha 对比度。默认值为40 、 160 、 80和255 。允许的范围是[0, 255] 。

射频

女友

高炉

自动对焦

指定红色、绿色、蓝色和 alpha 淡化。默认值为15 、 10 、 5和5 。允许的范围是[0, 255] 。

飞涨

设置缩放系数。默认值为1 。允许的范围是[0, 10] 。小于1 的值将自动调整缩放因子到最大可能值。

画

设置矢量示波器绘图模式。

可用值是：

'点'

为每个样本画点。

'线'

在上一个样本和当前样本之间画线。

'碱'

在上一个样本和当前样本之间绘制抗锯齿线。

默认值为“点”。

规模

指定音频样本的振幅比例。

可用值是：

'临'

线性的。

'开方'

平方根。

' cbrt '

立方根。

'日志'

对数。

交换

将左通道轴与右通道轴交换。

镜子

镜像轴。

'无'

没有镜子。

' x '

仅镜像 x 轴。

'是'

仅镜像 y 轴。

' xy '

镜像两个轴。

* [例子](#Examples-192)
* [命令](#Commands-158)

#### 45.7.1 例子

* ffplay 的完整示例：
* ffplay -f lavfi 'amovie=input.mp3, asplit [a][out1];
* [a] avectorscope=zoom=1.3:rc=2:gc=200:bc=10:rf=1:gf=8:bf=7 [out0]'

#### 45.7.2 命令

此过滤器支持上述所有选项作为命令，但选项size和rate除外。

### 45.8 板凳，板凳

过滤图的基准部分。

过滤器接受以下选项：

行动

启动或停止计时器。

可用值是：

'开始'

获取当前时间，将其设置为帧元数据（使用键lavfi.bench.start\_time ），并将帧转发到下一个过滤器。

'停止'

获取当前时间并从输入帧元数据中获取lavfi.bench.start\_time元数据以获得时差。然后打印时差、平均、最大和最小时间（分别为t 、 avg 、 max和min ）。时间戳以秒为单位表示。

* [例子](#Examples-193)

#### 45.8.1 例子

* 基准[选择性](#selectivecolor)滤色器：
* 长凳=开始，选择性颜色=红色=-.2 .12 -.49，长凳=停止

### 45.9 连续

连接音频和视频流，将它们一个接一个地连接在一起。

该过滤器适用于同步视频和音频流的片段。所有段必须具有相同数量的每种类型的流，这也将是输出流的数量。

过滤器接受以下选项：

n

设置段数。默认值为 2。

v

设置输出视频流的数量，即每段视频流的数量。默认值为 1。

A

设置输出音频流的个数，即每段音频流的个数。默认为 0。

不安全

激活不安全模式：如果段具有不同的格式，则不会失败。

过滤器有v + a输出：首先是v视频输出，然后是a音频输出。

有n x( v + a ) 个输入：首先是第一段的输入，顺序与输出相同，然后是第二段的输入，依此类推。

由于各种原因，包括编解码器帧大小或草率的创作，相关流并不总是具有完全相同的持续时间。出于这个原因，相关的同步流（例如视频及其音频轨道）应该立即连接起来。 concat 过滤器将使用每个段中最长流的持续时间（最后一个除外），并在必要时用静音填充较短的音频流。

要使此过滤器正常工作，所有段都必须从时间戳 0 开始。

所有相应的流必须在所有段中具有相同的参数；过滤系统会自动为视频流选择通用的像素格式，为音频流自动选择通用的采样格式、采样率和通道布局，但其他设置（例如分辨率）必须由用户明确转换。

不同的帧率是可以接受的，但会导致输出帧率可变；一定要配置输出文件来处理它。

* [例子](#Examples-194)
* [命令](#Commands-159)

#### 45.9.1 例子

* 连接一个开场、一集和一个结局，全部为双语版本（流 0 中的视频，流 1 和 2 中的音频）：
* ffmpeg -i opening.mkv -i episode.mkv -i ending.mkv -filter\_complex \
* '[0:0] [0:1] [0:2] [1:0] [1:1] [1:2] [2:0] [2:1] [2:2]
* concat=n=3:v=1:a=2 [v] [a1] [a2]' \
* -map '[v]' -map '[a1]' -map '[a2]' 输出.mkv
* 连接两部分，分别处理音频和视频，使用 (a) 电影源，并调整分辨率：
* movie=part1.mp4, scale=512:288 [v1] ; amovie=part1.mp4 [a1] ;
* movie=part2.mp4, scale=512:288 [v2] ; amovie=part2.mp4 [a2] ;
* [v1] [v2] 连接 [outv] ; [a1] [a2] concat=v=0:a=1 [outa]

请注意，如果音频和视频流在第一个文件中的持续时间不完全相同，则会在拼接处发生不同步。

#### 45.9.2 命令

该过滤器支持以下命令：

下一个

关闭当前段并进入下一个段

### 45.10 ebur128

EBU R128 扫描仪过滤器。此过滤器获取音频流并分析其响度级别。默认情况下，它以 10Hz 的频率记录一条消息，其中包含瞬时响度（由M标识）、短期响度 ( S )、综合响度 ( I ) 和响度范围 ( LRA )。

过滤器只能分析样本格式为双精度浮点数的流。如果需要，输入流将转换为此规范。用户可能需要在此过滤器之后插入格式和/或采样过滤器以获得原始参数。

该滤波器还有一个视频输出（参见视频选项），带有实时图表以观察响度演变。该图形包含上述记录的消息，因此设置此选项时不再打印它，除非设置了详细日志记录。主绘图区域包含短期响度（3 秒分析），右侧的仪表用于瞬时响度（400 毫秒），但可以选择配置为显示短期响度（参见仪表） 。

绿色区域标记围绕目标响度的 +/- 1LU 目标范围（默认情况下为 -23LUFS，除非通过target修改）。

有关响度建议 EBU R128 的更多信息，请访问<http://tech.ebu.ch/loudness>。

过滤器接受以下选项：

视频

激活视频输出。无论是否设置此选项，音频流都将不变地传递。如果激活，视频流将是第一个输出流。默认为0 。

尺寸

设置视频大小。此选项仅适用于视频。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认和最小分辨率为640x480 。

仪表

设置 EBU 刻度表。默认值为9 。 EBU 刻度计 +9 和 EBU 刻度计 +18 的常用值为9和18 。允许此范围内的任何其他整数值。

元数据

设置元数据注入。如果设置为1 ，音频输入将被分割成 100ms 的输出帧，每个帧在元数据中包含各种响度信息。所有元数据键都以lavfi.r128 为前缀。 .

默认为0 。

框架日志

强制帧记录级别。

可用值是：

'安静'

禁用日志记录

'信息'

信息记录级别

'冗长'

详细日志级别

默认情况下，日志记录级别设置为info 。如果设置了视频或元数据选项，它将切换为verbose 。

顶峰

设置峰值模式。

可以累积可用模式（选项是标志类型）。可能的值是：

'无'

禁用任何峰值模式（默认）。

'样本'

启用样本峰值模式。

寻找更高样本值的简单峰值模式。它记录 sample-peak 的消息（由SPK标识）。

'真实'

启用真峰值模式。

如果启用，峰值查找是在输入流的过采样版本上完成的，以获得更好的峰值精度。它记录真实峰值的消息。 （由TPK识别）和每帧真实峰值（由FTPK识别）。此模式需要使用libswresample进行构建。

双单声道

将单声道输入文件视为“双单声道”。如果单声道文件旨在在立体声系统上播放，则其 EBU R128 测量值将在感知上不正确。如果设置为true ，此选项将补偿此效果。多通道输入文件不受此选项的影响。

潘劳

设置用于测量双单声道文件的特定声相法则。此参数是可选的，默认值为 -3.01dB。

目标

设置在可视化中用作相对零的特定目标级别（在 LUFS 中）。此参数是可选的，默认值为 -23LUFS，由 EBU R128 指定。但是，在线发布的材料可能更喜欢 -16LUFS 的级别（例如，用于播客或视频平台）。

测量

设置仪表显示的值。有效值为momentary和 s shortterm 。默认情况下将使用瞬时值，但在某些情况下，观察短期值可能更有用（例如现场混音）。

规模

设置响度的显示比例。有效参数是相对于目标的绝对（LUFS）或相对（LU）。这只会影响视频输出，不会影响摘要或连续日志输出。

* [例子](#Examples-195)

#### 45.10.1 例子

* ffplay 的实时图表，EBU 比例尺 +18：
* ffplay -f lavfi -i "amovie=input.mp3,ebur128=video=1:meter=18 [out0][out1]"
* ffmpeg运行分析：
* ffmpeg -nostats -i input.mp3 -filter\_complex ebur128 -f null -

### 45.11交错，交错

暂时交错来自多个输入的帧。

interleave适用于视频输入， ainterleave适用于音频。

这些过滤器从多个输入读取帧并将最旧的排队帧发送到输出。

输入流必须具有明确定义的、单调递增的帧时间戳值。

为了将一帧提交到输出，这些过滤器需要为每个输入至少排队一帧，因此它们无法在一个输入尚未终止并且不会接收传入帧的情况下工作。

总是丢弃输入帧的选择过滤器的情况。交错过滤器将继续从该输入读取，但它永远无法发送新帧到输出，直到输入发送流结束信号。

此外，根据输入同步，过滤器将丢弃帧，以防一个输入接收到的帧多于其他输入，并且队列已满。

这些过滤器接受以下选项：

nb\_输入，n

设置不同输入的个数，默认为2个。

期间

如何确定流结束。

最长

最长输入的持续时间。 （默认）

最短的

最短输入的持续时间。

第一的

第一个输入的持续时间。

* [例子](#Examples-196)

#### 45.11.1 例子

* ffmpeg交错属于不同流的帧：
* ffmpeg -i bambi.avi -i pr0n.mkv -filter\_complex "[0:v][1:v] 交错" out.avi
* 添加闪烁模糊效果：
* select='if(gt(random(0), 0.2), 1, 2)':n=2 [tmp], boxblur=2:2, [tmp] 交错

### 45.12 潜伏期，潜伏期

测量过滤延迟。

报告以前的过滤器过滤延迟、音频过滤器的音频样本数延迟或视频过滤器的视频帧数。

在输入流结束时，过滤器将报告过滤器图中先前运行的过滤器的最小和最大测量延迟。

### 45.13 元数据，元数据

操纵帧元数据。

此过滤器接受以下选项：

模式

设置过滤器的操作模式。

可以是以下之一：

'选择'

如果同时设置了值和键，则选择具有此类元数据的帧。如果仅设置了键，则选择元数据中具有此类键的每个帧。

'添加'

添加新的元数据键和值。如果密钥已经可用，则什么也不做。

'修改'

修改已存在键的值。

'删除'

如果设置了值，则仅删除具有该值的键。否则，删除密钥。如果未设置键，则删除帧中的所有元数据值。

'打印'

如果找到元数据，则打印键及其值。如果未设置键，则打印框架中可用的所有元数据值。

钥匙

设置用于所有模式的键。必须为除print和delete之外的所有模式设置。

价值

设置将要使用的元数据值。此选项对于修改和添加模式是必需的。

功能

值时使用哪个函数。

可以是以下之一：

' same\_str '

value相同则返回 true 。

'开始于'

值被解释为字符串，如果元数据值以值选项字符串开头，则返回 true。

'少'

值被解释为浮点数，如果元数据值小于value则返回 true 。

'等于'

值被解释为浮点数，如果值与元数据值相等则返回真。

'更大'

值被解释为浮点数，如果元数据值大于value则返回 true 。

'表达式'

值被解释为浮点数，如果来自选项expr 的表达式的计算结果为真，则返回真。

' ends\_with '

值被解释为字符串，如果元数据值以值选项字符串结尾，则返回 true。

表达式

设置函数设置为expr时使用的表达式。表达式通过 eval API 求值，可以包含以下常量：

VALUE1，框架

来自元数据键的值的浮点表示。

VALUE2，USERVAL

值选项中提供的值的浮点表示。

文件

如果在打印模式中指定，输出将写入指定文件。可以指定任何可写的 url，而不是普通文件名。文件名“-”是标准输出的简写。如果未设置文件选项，输出将以 AV\_LOG\_INFO 日志级别写入日志。

直接的

当输出写入使用file的 URL 集时，减少打印模式下的缓冲。

* [例子](#Examples-197)

#### 45.13.1 例子

* lavfi.signalstats.YDIF且值介于 0 和 1 之间的帧的所有元数据值。
* signalstats,metadata=print:key=lavfi.signalstats.YDIF:value=0:function=expr:expr='between(VALUE1,0,1)'
* 将 silencedetect 输出打印到文件metadata.txt 。
* silencedetect,ametadata=mode=print:file=metadata.txt
* 将所有元数据定向到文件描述符为 4 的管道。
* metadata=mode=print:file='pipe\:4'

### 45.14 烫发，烫发

为输出帧设置读/写权限。

这些过滤器主要是针对开发人员在过滤器图中的以下过滤器中测试直接路径。

过滤器接受以下选项：

模式

选择权限模式。

它接受以下值：

'无'

没做什么。这是默认设置。

'罗'

将所有输出帧设置为只读。

'写'

将所有输出帧设置为直接可写。

'切换'

如果可写，则将框架设置为只读，如果只读，则将其设置为可写。

'随机'

将每个输出帧随机设置为只读或可写。

种子

随机模式的种子，必须是介于0和UINT32\_MAX之间的整数。如果未指定，或者如果明确设置为-1 ，过滤器将尝试在尽力而为的基础上使用良好的随机种子。

注意：如果在权限过滤器和下一个过滤器之间自动插入过滤器，则可能无法在下一个过滤器中按预期接收到权限。在 perms/aperms 过滤器之前插入一个[format](#format)或[aformat过滤器可以避免这个问题。](#aformat)

### 45.15 实时，实时

减慢过滤以近似匹配实时。

这些过滤器将暂停过滤一段可变的时间，以匹配输出速率与输入时间戳。它们类似于ffmpeg的re选项。

他们接受以下选项：

限制

暂停的时间限制。任何长于该时间的暂停都将被视为时间戳不连续并重置计时器。默认为 2 秒。

速度

处理速度系数。该值必须是大于零的浮点数。大于 1.0 的值将导致比实时处理更快，更小的值将减慢处理速度。限制会相应地自动调整。默认值为 1.0。

无法实现比没有这些过滤器时更快的处理速度。

* [命令](#Commands-160)

#### 45.15.1 命令

两个过滤器都支持上述所有选项作为[命令](#commands)。

### 45.16段，段

将单个输入流拆分为多个流。

此过滤器与连接过滤器相反。

segment作用于视频帧，而asegment作用于音频样本。

此过滤器接受以下选项：

时间戳

由“|”分隔的输出段的时间戳。第一段将从输入流的开头运行。最后一段将运行到输入流结束

框架，样品

分割段的精确帧/样本数。

在所有情况下，在每个段前加上“+”前缀将使它相对于前一个段。

* [例子](#Examples-198)

#### 45.16.1 例子

* 将输入音频流分成三个输出音频流，从输入音频流的开头开始并将其存储在第一个输出音频流中，然后在第 60 秒之后存储在第二个输出音频流中，最后在输入音频流存储的第 150 秒之后在第三个输出音频流中：
* asegment=timestamps="60|150"

### 45.17 选择，选择

选择要传入输出的帧。

此过滤器接受以下选项：

表达式，e

设置表达式，它针对每个输入帧进行评估。

如果表达式的计算结果为零，则丢弃该帧。

如果评估结果为负或 NaN，则将帧发送到第一个输出；否则它被发送到索引为ceil(val)-1的输出，假设输入索引从 0 开始。

例如，值1.2对应于索引为ceil(1.2)-1 = 2-1 = 1 的输出，即第二个输出。

输出，n

设置输出数量。将所选帧发送到的输出基于评估结果。默认值为 1。

表达式可以包含以下常量：

n

过滤帧的（连续）编号，从 0 开始。

selected\_n

所选帧的（连续）编号，从 0 开始。

prev\_selected\_n

最后选定帧的序号。如果未定义，则为 NAN。

结核病

输入时间戳的时基。

积分

过滤帧的PTS（Presentation TimeStamp），单位为TB 。如果未定义，则为 NAN。

吨

过滤帧的 PTS，以秒为单位。如果未定义，则为 NAN。

prev\_pts

先前过滤帧的 PTS。如果未定义，则为 NAN。

prev\_selected\_pts

先前过滤的最后一帧的 PTS。如果未定义，则为 NAN。

prev\_selected\_t

先前选择的最后一帧的 PTS，以秒为单位。如果未定义，则为 NAN。

开始点

流中不是 NAN 的第一个 PTS。如果找不到，它仍然是 NAN。

开始\_t

不是 NAN 的流中的第一个 PTS，以秒为单位。如果找不到，它仍然是 NAN。

pict\_type （仅限视频）

过滤帧的类型。它可以采用以下值之一：

我

P

乙

小号

国际单位制

SP

双

interlace\_type （仅限视频）

帧交错类型。它可以采用以下值之一：

进步

帧是渐进的（不是交错的）。

顶峰

帧是顶场优先的。

底部优先

帧是底场优先的。

consumed\_sample\_n （仅音频）

当前帧之前的选定样本数

samples\_n （仅音频）

当前帧中的样本数

sample\_rate （仅音频）

输入采样率

钥匙

如果过滤帧是关键帧，则为 1，否则为 0。

位置

过滤帧在文件中的位置，如果信息不可用（例如合成视频）则为 -1

场景（仅视频）

介于 0 和 1 之间的值表示一个新场景；较低的值反映当前帧引入新场景的可能性较低，而较高的值表示当前帧更有可能是一个新场景（请参见下面的示例）

concatdec\_select

concat demuxer 可以通过设置 inpoint 和 outpoint 仅选择 concat 输入文件的一部分，但输出数据包可能不会完全包含在所选间隔中。通过使用此变量，可以跳过由 concat demuxer 生成的未完全包含在所选间隔中的帧。

这是通过将帧 pts 与解码帧中也存在的lavf.concat.start\_time和lavf.concat.duration数据包元数据值进行比较来实现的。

帧pts 至少为 start\_time 并且缺少持续时间元数据或帧 pts 小于 start\_time + 持续时间，则 concatdec\_select 变量为 -1，否则为 0，如果缺少 start\_time 元数据则为 NaN 。

这基本上意味着如果输入帧的 pts 在 concat 分离器设置的间隔内，则选择输入帧。

选择表达式的默认值为“1”。

* [例子](#Examples-199)

#### 45.17.1 例子

* 选择输入中的所有帧：
* 选择

上面的示例与以下内容相同：

选择=1

* 跳过所有帧：
* 选择=0
* 仅选择 I 帧：
* select='eq(pict\_type\,I)'
* 每 100 选择一帧：
* 选择='不（mod（n \，100））'
* 仅选择包含在 10-20 时间间隔内的帧：
* 选择=之间（t\,10\,20）
* 仅选择包含在 10-20 时间间隔内的 I 帧：
* select=between(t\,10\,20)\*eq(pict\_type\,I)
* 选择最小距离为 10 秒的帧：
* select='isnan(prev\_selected\_t)+gte(t-prev\_selected\_t\,10)'
* 使用 aselect 仅选择样本数 > 100 的音频帧：
* aselect='gt(samples\_n\,100)'
* 创建第一个场景的马赛克：
* ffmpeg -i video.avi -vf select='gt(scene\,0.4)',scale=160:120,tile -frames:v 1 preview.png

场景与 0.3 到 0.5 之间的值进行比较通常是一个明智的选择。

* 将偶数帧和奇数帧发送到单独的输出，并组合它们：
* 选择=n=2:e='mod(n, 2)+1' [奇数][偶数]; [奇数] pad=h=2\*ih [tmp]; [tmp][偶数] 覆盖=y=h
* 从 ffconcat 文件中选择有用的帧，该文件使用入点和出点，但源文件不仅仅是帧内文件。
* ffmpeg -copyts -vsync 0 -segment\_time\_metadata 1 -i input.ffconcat -vf select=concatdec\_select -af aselect=concatdec\_select 输出.avi

### 45.18 发送命令，发送命令

向过滤器图中的过滤器发送命令。

这些过滤器读取要发送到过滤器图中其他过滤器的命令。

sendcmd必须插入两个视频过滤器之间， asendcmd必须插入两个音频过滤器之间，但除此之外它们的行为方式相同。

commands选项的过滤器参数中提供，或者在由filename选项指定的文件中提供。

这些过滤器接受以下选项：

命令，c

设置要读取并发送到其他过滤器的命令。

文件名，f

设置要读取并发送到其他过滤器的命令的文件名。

* [命令语法](#Commands-syntax)
* [例子](#Examples-200)

#### 45.18.1 命令语法

命令描述由一系列间隔规范组成，包括当与该间隔相关的特定事件发生时要执行的命令列表。发生的事件通常是当前帧时间进入或离开给定时间间隔。

间隔由以下语法指定：

启动[-结束]命令；

时间间隔由START和END时间指定。 END是可选的，默认为最大时间。

START , END ]区间内，即时间大于或等于START且小于END ，则认为当前帧时间在指定区间内。

COMMANDS由一系列一个或多个命令规范组成，由“,”分隔，与该间隔相关。命令规范的语法由下式给出：

[旗帜]目标 命令 ARG

FLAGS是可选的，它指定与允许发送指定命令的时间间隔相关的事件类型，并且必须是由“+”或“|”分隔的标识符标志的非空序列并包含在“[”和“]”之间。

识别以下标志：

进入

当当前帧时间戳进入指定时间间隔时发送该命令。换句话说，当前一帧时间戳不在给定的时间间隔内时发送命令，而当前是。

离开

当当前帧时间戳离开指定的时间间隔时发送该命令。换句话说，当前一帧时间戳在给定的时间间隔内时发送命令，而当前不在。

表达式

命令ARG被解释为表达式，表达式的结果作为ARG传递。

表达式通过 eval API 求值，可以包含以下常量：

收银机

帧文件中的原始位置，如果当前帧未定义，则为未定义。

公关部

输入中的呈现时间戳。

否

视频或音频的输入帧数，从 0 开始。

吨

当前帧的时间（以秒为单位）。

TS

当前命令间隔的开始时间（以秒为单位）。

TE

当前命令间隔的结束时间（以秒为单位）。

TI

当前命令间隔的插补时间，TI = (T - TS) / (TE - TS)。

W

视频帧宽度。

H

视频帧高度。

如果未指定FLAGS ，则假定为默认值[enter] 。

TARGET指定命令的目标，通常是过滤器类的名称或特定的过滤器实例名称。

COMMAND指定目标过滤器的命令名称。

ARG是可选的，它指定给定COMMAND的可选参数列表。

#开头直到行尾的字符序列将被忽略，并可用于注释注释。

命令规范语法的简化 BNF 描述如下：

COMMAND\_FLAG ::= "输入" | “离开”

COMMAND\_FLAGS ::= COMMAND\_FLAG [(+|"|") COMMAND\_FLAG ]

命令::= ["[" COMMAND\_FLAGS "]"]目标 命令[参数]

命令::=命令[,命令]

间隔::=开始[-结束]命令

间隔::=间隔[;间隔]

#### 45.18.2 例子

* 在第 4 秒指定音频速度变化：
* asendcmd=c='4.0 atempo 节奏 1.5',atempo
* 针对特定的过滤器实例：
* asendcmd=c='4.0 atempo@my tempo 1.5',atempo@my
* 在文件中指定 drawtext 和 hue 命令的列表。
* # 显示区间 5-10 的文本
* 5.0-10.0 [enter] drawtext reinit 'fontfile=FreeSerif.ttf:text=hello world',
* [离开] drawtext reinit 'fontfile=FreeSerif.ttf:text=';
* # 在 15-20 区间对图像去饱和
* 15.0-20.0 [输入] hue s 0,
* [输入] drawtext reinit 'fontfile=FreeSerif.ttf:text=nocolor',
* [离开] hue s 1,
* [离开] drawtext reinit 'fontfile=FreeSerif.ttf:text=color';
* # 应用指数饱和淡出效果，从时间 25 开始
* 25 [输入] hue s exp(25-t)

允许读取和处理存储在文件test.cmd中的上述命令列表的过滤器图可以指定为：

sendcmd=f=test.cmd,drawtext=fontfile=FreeSerif.ttf:text='',hue

### 45.19 setpts，asetpts

更改输入帧的 PTS（呈现时间戳）。

setpts适用于视频帧， asetpts适用于音频帧。

此过滤器接受以下选项：

表达式

为每个帧评估以构建其时间戳的表达式。

表达式通过 eval API 求值，可以包含以下常量：

FRAME\_RATE，FR

帧率，仅为恒定帧率视频定义

公关部

输入中的呈现时间戳

否

视频的输入帧计数或消耗的样本数，不包括音频的当前帧，从 0 开始。

NB\_CONSUMED\_SAMPLES 个样本

消耗的样本数，不包括当前帧（仅音频）

NB\_SAMPLES，小号

当前帧中的样本数（仅音频）

SAMPLE\_RATE，SR

音频采样率。

开始点

第一帧的PTS。

开始

第一帧的时间（以秒为单位）

隔行扫描

说明当前帧是否交错。

吨

当前帧的时间（以秒为单位）

收银机

帧文件中的原始位置，如果当前帧未定义，则未定义

PREV\_INPTS

上一个输入PTS。

PREV\_INT

以前的输入时间（以秒为单位）

PREV\_OUTPTS

之前的输出PTS。

PREV\_OUTT

以前的输出时间（以秒为单位）

RTC时间

以微秒为单位的挂钟 (RTC) 时间。这已被弃用，请改用 time(0)。

时钟启动

电影开始时的挂钟 (RTC) 时间，以微秒为单位。

结核病

输入时间戳的时基。

* [例子](#Examples-201)

#### 45.19.1 例子

* 从零开始计算PTS
* setpts=PTS-开始PTS
* 应用快动作效果：
* setpts=0.5\*PTS
* 应用慢动作效果：
* setpts=2.0\*PTS
* 设置每秒 25 帧的固定速率：
* setpts=N/(25\*TB)
* 设置固定速率 25 fps 并带有一些抖动：
* setpts='1/(25\*TB) \* (N + 0.05 \* sin(N\*2\*PI/25))'
* 对输入 PTS 应用 10 秒的偏移量：
* setpts=PTS+10/TB
* 从“实时源”生成时间戳并重新定位到当前时基：
* setpts='(RTCTIME - RTCSTART) / (TB \* 1000000)'
* 通过计数样本生成时间戳：
* asetpts=N/SR/TB

### 45.20 设定范围

强制输出视频帧的颜色范围。

setrange过滤器标记输出帧的颜色范围属性。它不会更改输入帧，而只会设置相应的属性，这会影响后续过滤器如何处理该帧。

过滤器接受以下选项：

范围

可用值是：

'汽车'

保持相同的颜色范围属性。

'未指定，未知'

将颜色范围设置为未指定。

'有限，电视，mpeg '

将颜色范围设置为有限。

'完整，电脑，jpeg '

将颜色范围设置为完整。

### 45.21 settb, asettb

设置用于输出帧时间戳的时基。它主要用于测试时基配置。

它接受以下参数：

表达式, tb

在输出时基中计算的表达式。

tb的值是表示有理数的算术表达式。该表达式可以包含常量“AVTB”（默认时基）、“intb”（输入时基）和“sr”（采样率，仅音频）。默认值为“intb”。

* [例子](#Examples-202)

#### 45.21.1 例子

* 将时基设置为 1/25：
* settb=expr=1/25
* 将时基设置为 1/10：
* settb=expr=0.1
* 将时基设置为 1001/1000：
* 设置b=1+0.001
* 将时基设置为 2\*intb：
* settb=2\*intb
* 设置默认时基值：
* 设置b=AVTB

### 45.22 showcqt

使用具有直接频域系数计算的 Brown-Puckette 常数 Q 变换算法将输入音频转换为以对数表示频谱的视频输出（但变换本身并不是真正的常数 Q，Q 因子实际上是可变的/钳位的），带有音乐音调规模，从 E0 到 D#10。

过滤器接受以下选项：

大小，小号

指定输出的视频大小。它必须是均匀的。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为1920x1080 。

fps, 速率, r

设置输出帧率。默认值为25 。

bar\_h

设置条形图高度。它必须是均匀的。默认值为-1 ，它会自动计算条形图高度。

轴\_h

设置轴高度。它必须是均匀的。默认值为-1 ，自动计算轴高。

sono\_h

设置超声图高度。它必须是均匀的。默认值为-1 ，自动计算超声图高度。

全高清

设置全高清分辨率。此选项已弃用，请改用size 、 s 。默认值为1 。

sono\_v，音量

指定超声图体积表达式。它可以包含变量：

bar\_v

bar\_v评估的表达式

频率，频率，f

评估的频率

时间夹, tc

timeclamp选项的值

和功能：

a\_weighting(f)

等响度A计权

b\_weighting(f)

等响度B计权

c\_weighting(f)

等响度的 C 计权。

默认值为16 。

bar\_v, volume2

指定条形图音量表达式。它可以包含变量：

sono\_v

sono\_v评估表达式

频率，频率，f

评估的频率

时间夹, tc

timeclamp选项的值

和功能：

a\_weighting(f)

等响度A计权

b\_weighting(f)

等响度B计权

c\_weighting(f)

等响度的 C 计权。

默认值为sono\_v 。

sono\_g, 伽玛

指定超声图伽玛。较低的伽玛使光谱对比度更高，较高的伽玛使光谱具有更大的范围。默认值为3 。可接受的范围是[1, 7] 。

bar\_g, 伽玛2

指定条形图伽玛。默认值为1 。可接受的范围是[1, 7] 。

bar\_t

指定条形图透明度级别。较低的值使条形图更清晰。默认值为1 。可接受的范围是[0, 1] 。

时间夹, tc

指定变换时间夹。在低频下，时域和频域精度之间存在折衷。如果 timeclamp 较低，则表示时域中的事件更准确（例如快速低音鼓），否则表示频域中的事件更准确（例如低音吉他）。可接受的范围是[0.002, 1] 。默认值为0.17 。

攻击

以秒为单位设置攻击时间。默认值为0 （禁用）。否则，它通过在时域中应用非对称窗口来限制未来的样本，这在需要低延迟时很有用。接受的范围是[0, 1] 。

基频

指定变换基频。默认值为20.01523126408007475 ，即低于 E0 50 美分的频率。可接受的范围是[10, 100000] 。

结束频率

指定变换结束频率。默认值为20495.59681441799654 ，频率比 D#10 高 50 美分。可接受的范围是[10, 100000] 。

系数钳

此选项已弃用并被忽略。

长度

指定时域中的变换长度。使用此选项可以控制每个频率样本的时域和频域之间的精度权衡。它可以包含变量：

频率，频率，f

评估的频率

时间夹, tc

timeclamp选项的值。

默认值为384\*tc/(384+tc\*f) 。

数数

指定每个视频帧的变换计数。默认值为6 。可接受的范围是[1, 30] 。

计数

指定每个像素的变换计数。默认值为0 ，这使得它自动计算。可接受的范围是[0, 10] 。

字体文件

指定用于 freetype 绘制轴的字体文件。如果未指定，则使用嵌入字体。请注意，使用自定义basefreq和endfreq未实现使用字体文件或嵌入字体进行绘图，请改用axisfile选项。

字体

指定字体配置模式。这比fontfile 的优先级低。模式中的:可以替换为|以避免不必要的逃避。

字体颜色

指定字体颜色表达。这是应返回整数值 0xRRGGBB 的算术表达式。它可以包含变量：

频率，频率，f

评估的频率

时间夹, tc

timeclamp选项的值

和功能：

中音（女）

频率f的midi号，一些midi号：E0(16), C1(24), C2(36), A4(69)

r(x), g(x), b(x)

强度 x 的红色、绿色和蓝色值。

默认值为st(0, (midi(f)-59.5)/12)； st(1, 如果(介于(ld(0),0,1), 0.5-0.5\*cos(2\*PI\*ld(0)), 0)); r(1-ld(1)) + b(ld(1)) 。

轴文件

指定图像文件来绘制轴。此选项覆盖fontfile和fontcolor选项。

轴、文字

启用/禁用向轴绘制文本。如果设置为0 ，则禁用绘制到轴，忽略fontfile和axisfile选项。默认值为1 。

太阳能电池板

设置色彩空间。接受的值是：

'未指定'

未指定（默认）

' bt709 '

BT.709

' fcc '

美国联邦通信委员会

' bt470bg '

BT.470BG 或 BT.601-6 625

' smpte170m '

SMPTE-170M 或 BT.601-6 525

' smpte240m '

SMPTE-240M

' bt2020ncl '

BT.2020 非恒定亮度

方案

设置频谱图配色方案。这是格式为left\_r|left\_g|left\_b|right\_r|right\_g|right\_b 的浮点值列表。默认值为1|0.5|0|0|0.5|1 。

* [例子](#Examples-203)

#### 45.22.1 例子

* 在显示频谱的同时播放音频：
* ffplay -f lavfi 'amovie=a.mp3, asplit [a][out1]; [a] showcqt [out0]'
* 同上，但帧速率为 30 fps：
* ffplay -f lavfi 'amovie=a.mp3, asplit [a][out1]; [a] showcqt=fps=30:count=5 [out0]'
* 以 1280x720 分辨率播放：
* ffplay -f lavfi 'amovie=a.mp3, asplit [a][out1]; [a] showcqt=s=1280x720:count=4 [out0]'
* 禁用超声图显示：
* sono\_h=0
* A1及其谐波：A1、A2、（近）E3、A3：
* ffplay -f lavfi 'aevalsrc=0.1\*sin(2\*PI\*55\*t)+0.1\*sin(4\*PI\*55\*t)+0.1\*sin(6\*PI\*55\*t)+0.1\*sin( 8\*PI\*55\*t),
* 拆分[a][out1]； [a] showcqt [out0]'
* 与上面相同，但在频域中具有更高的精度：
* ffplay -f lavfi 'aevalsrc=0.1\*sin(2\*PI\*55\*t)+0.1\*sin(4\*PI\*55\*t)+0.1\*sin(6\*PI\*55\*t)+0.1\*sin( 8\*PI\*55\*t),
* 拆分[a][out1]； [a] showcqt=timeclamp=0.5 [out0]'
* 自定义音量：
* bar\_v=10:sono\_v=bar\_v\*a\_weighting(f)
* 自定义伽玛，现在频谱与振幅呈线性关系。
* bar\_g=2:sono\_g=2
* 自定义长度方程：
* tc=0.33:tlength='st(0,0.17); 384\*tc / (384 / ld(0) + tc\*f /(1-ld(0))) + 384\*tc / (tc\*f / ld(0) + 384 /(1-ld(0)) )'
* 自定义fontcolor和fontfile，C-note为绿色，其他为蓝色：
* fontcolor='if(mod(floor(midi(f)+0.5),12), 0x0000FF, g(1))':fontfile=myfont.ttf
* 使用 fontconfig 自定义字体：
* font='Courier New,Monospace,mono|bold'
* 使用图像文件的自定义轴的自定义频率范围：
* axisfile=myaxis.png:basefreq=40:endfreq=10000

### 45.23 showcwt

使用连续小波变换和 Morlet 小波将输入音频转换为表示频谱的视频输出。

过滤器接受以下选项：

大小，小号

指定输出的视频大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为640x512 。

率，r

设置输出帧率。默认值为25 。

规模

设置使用的频率刻度。允许的值为：

线性的

log2

吠

梅尔

erbs

默认值为linear 。

分钟

设置将在输出中使用的最小频率。默认值为20赫兹。

最大限度

设置将在输出中使用的最大频率。默认值为20000赫兹。实际频率上限取决于输入音频的采样率，当它设置为大于奈奎斯特频率的值时，将对此值强制执行。

日志

将计算出的幅度值映射到像素值时，设置亮度强度的对数基础。允许的范围是从0到1 。默认值为0.0001 。

偏差

设置频率偏差。低于1的值更注重频率，而高于1的值更注重时间。允许的范围是从0到10 。默认值为1 。

pps

设置一行中每秒输出的像素数。允许的范围是从1到1024 。默认值为64 。

模式

设置输出视觉模式。允许的值为：

震级

显示幅度。

阶段

只显示阶段。

大相位

显示幅度和相位的组合。幅度映射到亮度，相位映射到颜色。

渠道

显示每个通道大小的独特颜色。

立体声

显示每个立体差异的独特颜色。

默认值为magnitude 。

滑动

设置输出幻灯片方法。允许的值为：

代替

滚动

框架

方向

设置输出滑动方式的方向方式。允许的值为：

lr

方向从左到右。

rl

方向从右到左。

乌德

方向从上到下。

杜

方向从下到上。

### 45.24 显示频率

将输入音频转换为表示音频功率谱的视频输出。音频幅度在 Y 轴上，而频率在 X 轴上。

过滤器接受以下选项：

大小，小号

指定视频的大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认为1024x512 。

率，r

设置视频速率。默认值为25 。

模式

设置显示模式。这设置了每个频率仓的表示方式。

它接受以下值：

'线'

'酒吧'

'点'

默认为bar 。

规模

设置振幅刻度。

它接受以下值：

'临'

线性刻度。

'开方'

平方根尺度。

' cbrt '

立方根尺度。

'日志'

对数刻度。

默认为日志。

规模

设置频率标度。

它接受以下值：

'临'

线性刻度。

'日志'

对数刻度。

' rlog '

反向对数刻度。

默认为lin 。

win\_size

设置窗口大小。允许的范围是 16 到 65536。

默认为2048

win\_func

设置窗口函数。

它接受以下值：

'正确'

'巴特利特'

‘汉宁’

'海明'

'黑人'

'韦尔奇'

'平顶'

'巴里斯'

' bnuttall '

'巴恩'

'正弦'

'坚果'

'兰佐斯'

'高斯'

'图基'

'海豚'

'柯西'

' parzen '

'泊松'

'波曼'

'凯撒'

默认是汉宁。

重叠

设置窗口重叠。在[0, 1]范围内。默认值为1 ，这意味着将选择所选窗口函数的最佳重叠。

平均

设置时间平均。将此设置为 0 将显示当前最大峰值。默认值为1 ，这意味着禁用时间平均。

颜色

指定由空格或“|”分隔的颜色列表它将用于绘制频道频率。无法识别或丢失的颜色将被白色替换。

模式

设置通道显示模式。

它接受以下值：

'合并'

'分开'

默认为合并。

最小放大器

数振幅定标器中使用的最小振幅。

数据

设置数据显示方式。

它接受以下值：

'幅度'

'阶段'

'延迟'

默认值为magnitude 。

频道

设置处理音频时要使用的通道。默认情况下所有被处理。

### 45.25 显示空间

将立体声输入音频转换为视频输出，表示两个通道之间的空间关系。

过滤器接受以下选项：

大小，小号

指定输出的视频大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为512x512 。

win\_size

设置窗口大小。允许的范围是从1024到65536 。默认大小为4096 。

win\_func

设置窗函数。

它接受以下值：

'正确'

'巴特利特'

'汉'

‘汉宁’

'海明'

'黑人'

'韦尔奇'

'平顶'

'巴里斯'

'坚果'

'巴恩'

'正弦'

'坚果'

'兰佐斯'

'高斯'

'图基'

'海豚'

'柯西'

' parzen '

'泊松'

博曼\_ \_

'凯撒'

默认值为hann 。

率，r

设置输出帧率。

### 45.26 显示频谱

将输入音频转换为视频输出，表示音频频谱。

过滤器接受以下选项：

大小，小号

指定输出的视频大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为640x512 。

滑动

指定频谱应如何沿窗口滑动。

它接受以下值：

'替换'

当样本到达右侧时，样本再次从左侧开始

'滚动'

样本从右向左滚动

'全画幅'

仅当样本到达右侧时才会产生帧

'滚动'

样本从左向右滚动

'替换'

样本到达左侧时再次从右侧开始

默认值为替换。

模式

指定显示模式。

它接受以下值：

'合并'

所有通道显示在同一行

'分开'

所有通道都显示在单独的行中

默认值为“组合”。

颜色

指定显示颜色模式。

它接受以下值：

'频道'

每个通道以单独的颜色显示

'强度'

每个通道都使用相同的配色方案显示

'彩虹'

每个通道都使用彩虹配色方案显示

'莫兰'

每个频道都使用 moreland 配色方案显示

'星云'

每个通道都使用星云配色方案显示

'火'

每个通道都使用火焰配色方案显示

'火热'

每个通道都使用火热的配色方案显示

'水果'

每个通道都使用水果配色方案显示

'酷'

每个频道都使用酷炫的配色方案显示

'岩浆'

每个通道都使用岩浆配色方案显示

'绿色'

每个通道都使用绿色配色方案显示

'绿色'

每个通道都使用 viridis 配色方案显示

'等离子'

每个通道都使用等离子配色方案显示

'公民'

每个通道都使用 cividis 配色方案显示

'地形'

每个通道都使用地形配色方案显示

默认值为“通道”。

规模

指定用于计算强度颜色值的比例。

它接受以下值：

'临'

线性的

'开方'

平方根，默认

' cbrt '

立方根

'日志'

对数的

'第四节'

第四根

' 5thr '

第五根

默认值为“ sqrt ”。

规模

指定频率标度。

它接受以下值：

'临'

线性的

'日志'

对数的

默认值为“ lin ”。

饱和

为显示的颜色设置饱和度修改器。负值提供替代配色方案。 0完全没有饱和度。饱和度必须在 [-10.0, 10.0] 范围内。默认值为1 。

win\_func

设置窗函数。

它接受以下值：

'正确'

'巴特利特'

'汉'

‘汉宁’

'海明'

'黑人'

'韦尔奇'

'平顶'

'巴里斯'

' bnuttall '

'巴恩'

'正弦'

'坚果'

'兰佐斯'

'高斯'

'图基'

'海豚'

'柯西'

' parzen '

'泊松'

'波曼'

'凯撒'

默认值为hann 。

方向

设置时间与频率轴的方向。可以是垂直的也可以是水平的。默认是垂直的。

重叠

设置重叠窗口的比例。默认值为0 。当值为1时，重叠设置为当前使用的特定窗口函数的推荐大小。

获得

设置用于计算强度颜色值的比例增益。默认值为1 。

数据

设置要显示的数据。可以是magnitude 、 default 或phase ，或展开的 phase ： uphase 。

回转

设置颜色旋转，必须在[-1.0, 1.0]范围内。默认值为0 。

开始

设置显示频谱图的起始频率。默认为0 。

停止

设置显示频谱图的停止频率。默认为0 。

每秒帧数

设置帧速率上限。默认为auto ，无限制。

传奇

绘制时间和频率轴和图例。默认是禁用的。

疯狂

设置用于计算强度颜色值的动态范围。默认值为 120 dBFS。允许的范围是 10 到 200。

限制

以 dBFS 为单位设置输入音频样本音量的上限。默认值为 0 dBFS。允许的范围是从 -100 到 100。

不透明度

使用带 alpha 组件的像素格式输出时设置不透明度强度。

用法与 showwaves 过滤器非常相似；请参阅该部分中的示例。

* [例子](#Examples-204)

#### 45.26.1 例子

* 具有对数颜色缩放比例的大窗口：
* showspectrum=s=1280x480:scale=log
* ffplay 的每个通道的彩色和滑动频谱的完整示例：
* ffplay -f lavfi 'amovie=input.mp3, asplit [a][out1];
* [a] showspectrum=mode=separate:color=intensity:slide=1:scale=cbrt [out0]'

### 45.27 显示光谱

将输入音频转换为单个视频帧，表示音频频谱。

过滤器接受以下选项：

大小，小号

指定输出的视频大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为4096x2048 。

模式

指定显示模式。

它接受以下值：

'合并'

所有通道显示在同一行

'分开'

所有通道都显示在单独的行中

默认值为“组合”。

颜色

指定显示颜色模式。

它接受以下值：

'频道'

每个通道以单独的颜色显示

'强度'

每个通道都使用相同的配色方案显示

'彩虹'

每个通道都使用彩虹配色方案显示

'莫兰'

每个频道都使用 moreland 配色方案显示

'星云'

每个通道都使用星云配色方案显示

'火'

每个通道都使用火焰配色方案显示

'火热'

每个通道都使用火热的配色方案显示

'水果'

每个通道都使用水果配色方案显示

'酷'

每个频道都使用酷炫的配色方案显示

'岩浆'

每个通道都使用岩浆配色方案显示

'绿色'

每个通道都使用绿色配色方案显示

'绿色'

每个通道都使用 viridis 配色方案显示

'等离子'

每个通道都使用等离子配色方案显示

'公民'

每个通道都使用 cividis 配色方案显示

'地形'

每个通道都使用地形配色方案显示

默认值为“强度”。

规模

指定用于计算强度颜色值的比例。

它接受以下值：

'临'

线性的

'开方'

平方根，默认

' cbrt '

立方根

'日志'

对数的

'第四节'

第四根

' 5thr '

第五根

默认值为“日志”。

规模

指定频率标度。

它接受以下值：

'临'

线性的

'日志'

对数的

默认值为“ lin ”。

饱和

为显示的颜色设置饱和度修改器。负值提供替代配色方案。 0完全没有饱和度。饱和度必须在 [-10.0, 10.0] 范围内。默认值为1 。

win\_func

设置窗函数。

它接受以下值：

'正确'

'巴特利特'

'汉'

‘汉宁’

'海明'

'黑人'

韦尔奇\_ \_

'平顶'

'巴里斯'

'坚果'

'债券'

'高级'

'坚果'

'兰佐斯'

'高斯'

'图基'

'海豚'

'柯西'

' parzen '

'泊松'

'波曼'

'凯撒'

默认值为hann 。

方向

设置时间与频率轴的方向。可以是垂直的也可以是水平的。默认是垂直的。

获得

设置用于计算强度颜色值的比例增益。默认值为1 。

传奇

绘制时间和频率轴和图例。默认启用。

回转

设置颜色旋转，必须在[-1.0, 1.0]范围内。默认值为0 。

开始

设置显示频谱图的起始频率。默认为0 。

停止

设置显示频谱图的停止频率。默认为0 。

疯狂

设置用于计算强度颜色值的动态范围。默认值为 120 dBFS。允许的范围是 10 到 200。

限制

以 dBFS 为单位设置输入音频样本音量的上限。默认值为 0 dBFS。允许的范围是从 -100 到 100。

不透明度

使用带 alpha 组件的像素格式输出时设置不透明度强度。

* [例子](#Examples-205)

#### 45.27.1 例子

* ffmpeg在 1024x1024 图片中提取整个音轨的音频频谱图：
* ffmpeg -i audio.flac -lavfi showspectrumpic=s=1024x1024 频谱图.png

### 45.28 showvolume

将输入音频音量转换为视频输出。

过滤器接受以下选项：

率，r

设置视频速率。

b

设置边框宽度，允许范围为[0, 5]。默认值为 1。

w

设置通道宽度，允许范围为[80, 8192]。默认值为 400。

H

设置通道高度，允许范围为 [1, 900]。默认值为 20。

F

设置淡入淡出，允许范围为 [0, 1]。默认值为 0.95。

C

设置体积颜色表达式。

表达式可以使用以下变量：

体积

通道的当前最大音量（以 dB 为单位）。

顶峰

电流峰值。

渠道

当前频道号，从0开始。

吨

如果设置，则显示通道名称。默认启用。

v

如果设置，则显示音量值。默认启用。

o

设置方向，可以是 horizontal: h或 vertical: v ，默认是h 。

秒

设置步长，允许范围为 [0, 5]。默认为 0，表示禁用步进。

p

设置背景不透明度，允许范围为 [0, 1]。默认为 0。

米

设置计量模式，可以是 peak: p或 rms: r ，默认是p 。

DS

设置显示比例，可以是线性： lin或对数： log ，默认为lin 。

分米

在第二。如果设置为 > 0.，则显示前几秒的最大水平线。默认禁用： 0。

管理委员会

最大线的颜色。当dm选项设置为 > 0时使用。默认为：橙色

### 45.29 秀波

将输入音频转换为视频输出，表示样本波。

过滤器接受以下选项：

大小，小号

指定输出的视频大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为600x240 。

模式

设置显示模式。

可用值是：

'点'

为每个样本画一个点。

'线'

为每个样本画一条垂直线。

'点对点'

为每个样本画一个点，并在它们之间画一条线。

'克莱恩'

为每个样本画一条居中的垂直线。

默认值为点。

n

设置在同一列上打印的样本数。较大的值会降低帧率。必须是正整数。仅当未明确指定速率值时才能设置此选项。

率，r

设置（近似）输出帧速率。这是通过设置选项n来完成的。默认值为“25”。

分离通道

设置通道是单独绘制还是重叠。默认值为 0。

颜色

设置由“|”分隔的颜色将用于绘制每个通道。

规模

设置振幅刻度。

可用值是：

'临'

线性的。

'日志'

对数。

'开方'

平方根。

' cbrt '

立方根。

默认是线性的。

画

设置绘制模式。这对于设置高n非常有用。

可用值是：

'规模'

缩放每个绘制样本的像素值。

'满'

直接绘制每个样本。

默认值为scale 。

* [例子](#Examples-206)

#### 45.29.1 例子

* 同时输出输入文件音频和相应的视频表示：
* amovie=a.mp3,asplit[out0],showwaves[out1]
* 创建一个合成信号并用 showwaves 显示它，强制帧速率为每秒 30 帧：
* aevalsrc=sin(1\*2\*PI\*t)\*sin(880\*2\*PI\*t):cos(2\*PI\*200\*t),asplit[out0],showwaves=r=30[out1]

### 45.30 showwavespic

将输入音频转换为单个视频帧，代表样本波。

过滤器接受以下选项：

大小，小号

指定输出的视频大小。有关此选项的语法，请查看[ffmpeg-utils 手册中的 (ffmpeg-utils)“视频大小”部分](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html#video-size-syntax)。默认值为600x240 。

分离通道

设置通道是单独绘制还是重叠。默认值为 0。

颜色

设置由“|”分隔的颜色将用于绘制每个通道。

规模

设置振幅刻度。

可用值是：

'临'

线性的。

'日志'

对数。

'开方'

平方根。

' cbrt '

立方根。

默认是线性的。

画

设置绘制模式。

可用值是：

'规模'

缩放每个绘制样本的像素值。

'满'

直接绘制每个样本。

默认值为scale 。

筛选

设置过滤模式。

可用值是：

'平均'

对每个绘制的样本使用平均样本值。

'峰'

为每个绘制的样本使用峰值样本值。

默认值为平均。

* [例子](#Examples-207)

#### 45.30.1 例子

* ffmpeg在 1024x800 图片中提取整个音轨波形的通道分割表示：
* ffmpeg -i audio.flac -lavfi showwavespic=split\_channels=1:s=1024x800 波形.png

### 45.31 边数据，asidedata

删除框架边数据，或基于它选择框架。

此过滤器接受以下选项：

模式

设置过滤器的操作模式。

可以是以下之一：

'选择'

类型边数据的每一帧。

'删除'

类型的边数据。如果未设置类型，则删除框架中的所有侧面数据。

类型

设置用于所有模式的端数据类型。必须设置为选择模式。有关帧端数据类型的列表，请参阅libavutil/frame.h中的AVFrameSideDataType枚举。例如，要选择AV\_FRAME\_DATA\_PANSCAN端数据，您必须指定PANSCAN 。

### 45.32 频谱合成器

从 2 个输入视频频谱合成音频，第一个输入流表示随时间变化的幅度，第二个表示随时间变化的相位。滤波器将从视频中显示的频域转换回音频输出中显示的时域。

这个过滤器主要是为了反转处理过的[showspectrum](#showspectrum)过滤器输出而创建的，但也可以从其他频谱图中合成声音。但在这种情况下，如果相位数据不可用，结果会很差，因为在这种情况下需要重新创建相位数据，通常只是从随机噪声中重新创建。为获得最佳结果，仅使用灰色输出（ [showspectrum过滤器中的](#showspectrum)通道颜色模式）和幅度视频的对数刻度和相位视频的线性刻度。要生成阶段，对于第二个视频，请使用数据选项。输入视频通常应使用全帧幻灯片模式，因为这样可以节省解码视频所需的资源。

过滤器接受以下选项：

采样率

指定输出音频的采样率，生成频谱的音频采样率可能不同。

频道

设置输入视频频谱中表示的通道数。

规模

设置生成幅度输入频谱时使用的比例。可以是lin或log 。默认为日志。

滑动

设置生成输入频谱时使用的幻灯片。可以是replace 、 scroll 、 fullframe或rscroll 。默认为全画幅。

win\_func

设置用于再综合的窗函数。

重叠

设置窗口重叠。在[0, 1]范围内。默认值为1 ，这意味着将选择所选窗口函数的最佳重叠。

方向

设置输入视频的方向。可以是垂直的也可以是水平的。默认是垂直的。

* [例子](#Examples-208)

#### 45.32.1 例子

* 首先从音频创建幅度和相位视频，假设音频是具有 44100 采样率的立体声，然后使用 spectrumsynth 将视频重新合成回音频：
* ffmpeg -i input.flac -lavfi showspectrum=mode=separate:scale=log:overlap=0.875:color=channel:slide=fullframe:data=magnitude -an -c:v 原始视频幅度.nut
* ffmpeg -i input.flac -lavfi showspectrum=mode=separate:scale=lin:overlap=0.875:color=channel:slide=fullframe:data=phase -an -c:v rawvideo phase.nut
* ffmpeg -i magnitude.nut -i phase.nut -lavfi spectrumsynth=channels=2:sample\_rate=44100:win\_func=hann:overlap=0.875:slide=fullframe output.flac

### 45.33 分裂，分裂

将输入拆分为几个相同的输出。

asplit适用于音频输入，适用于视频。

过滤器接受一个指定输出数量的参数。如果未指定，则默认为 2。

* [例子](#Examples-209)

#### 45.33.1 例子

* 从同一个输入创建两个独立的输出：
* [in] 拆分 [out0][out1]
* 要创建 3 个或更多输出，您需要指定输出数量，例如：
* [输入] asplit=3 [输出 0][输出 1][输出 2]
* 从同一输入创建两个单独的输出，一个被裁剪，一个被填充：
* [在]拆分[splitout1][splitout2];
* [splitout1] crop=100:100:0:0 [cropout];
* [splitout2] pad=200:200:100:100 [padout];
* ffmpeg创建 5 个输入音频副本：
* ffmpeg -i 输入 -filter\_complex asplit=5 输出

### 45.34 zmq，azmq

接收通过 libzmq 客户端发送的命令，并将它们转发给过滤器图中的过滤器。

zmq和azmq用作传递过滤器。 zmq必须插入两个视频过滤器之间， azmq必须插入两个音频过滤器之间。两者都能够将消息发送到任何过滤器类型。

要启用这些过滤器，您需要安装 libzmq 库和标头，并使用--enable-libzmq配置 FFmpeg 。

有关 libzmq 的更多信息，请参阅： http: [//www.zeromq.org/](http://www.zeromq.org/)

zmq和azmq过滤器用作 libzmq 服务器，它接收通过 bind\_address （或缩写“ b ”）选项定义的网络接口发送的消息。此选项的默认值为tcp://localhost:5555 。您可能希望根据需要更改此值，但不要忘记转义任何“:”符号（请参阅[filtergraph escaping](#filtergraph-escaping) ）。

收到的消息必须采用以下形式：

目标 命令[参数]

TARGET指定命令的目标，通常是过滤器类的名称或特定的过滤器实例名称。默认过滤器实例名称使用模式“ Parsed\_<filter\_name>\_<index> ”，但您可以使用“ filter\_name@id ”语法覆盖它（请参阅[Filtergraph 语法](#Filtergraph-syntax)）。

COMMAND指定目标过滤器的命令名称。

ARG是可选的，它指定给定COMMAND 的可选参数列表。

接收到消息后，将处理消息并将相应的命令注入到过滤器图中。根据结果，过滤器将向客户端发送回复，采用以下格式：

错误代码 错误原因

信息

MESSAGE是可选的。

* [例子](#Examples-210)

#### 45.34.1 例子

查看tools/zmqsend中的 zmq 客户端示例，它可用于发送由这些过滤器处理的命令。

考虑以下由ffplay生成的过滤图。在这个例子中，最后一个覆盖过滤器有一个实例名称。所有其他过滤器将具有默认实例名称。

ffplay -dumpgraph 1 -f lavfi“

颜色=s=100x100:c=红色[l];

颜色=s=100x100:c=蓝色[r];

nullsrc=s=200x100, zmq [bg];

[bg][l] 叠加 [bg+l]；

[bg+l][r] overlay@my=x=100 "

要更改视频左侧的颜色，可以使用以下命令：

echo Parsed\_color\_0 c 黄色 |工具/zmqsend

要更改右侧：

echo Parsed\_color\_1 c 粉色 |工具/zmqsend

要更改右侧的位置：

回声覆盖@我 x 150 |工具/zmqsend

## 46个多媒体资源

以下是对当前可用的多媒体资源的描述。

* [电影](#amovie)
* [同步测试](#avsynctest)
* [电影](#movie-1)

### 46.1 电影

[电影源](#movie)相同，只是它默认选择音频流。

### 46.2 异步测试

生成音频/视频同步测试。

生成的流定期显示 Flash 视频帧并发出蜂鸣声。用于检查 A/V 同步问题。

它接受以下选项：

大小，小号

设置输出视频大小。默认值为hd720 。

帧率，fr

设置输出视频帧率。默认值为30 。

采样率，sr

设置输出音频采样率。默认值为44100 。

幅度，一个

设置输出音频蜂鸣幅度。默认值为0.7 。

期间，p

以秒为单位设置输出音频蜂鸣周期。默认值为3 。

延迟，dl

以帧数设置输出视频闪光延迟。默认值为0 。

循环，c

启用视频延迟循环，默认情况下禁用。

持续时间，d

设置流输出持续时间。默认持续时间是无限的。

fg, bg, 银

设置前景/背景/附加颜色。

### 46.3电影

从电影容器中读取音频和/或视频流。

它接受以下参数：

文件名

要读取的资源的名称（不一定是文件；也可以是设备或通过某种协议访问的流）。

格式名称，f

指定要读取的电影采用的格式，可以是容器的名称或输入设备的名称。如果未指定，则从电影名称或通过探测猜测格式。

seek\_point, sp

以秒为单位指定搜索点。帧将从该搜索点开始输出。该参数是用av\_strtod评估的，因此数值可能以 IS 后缀为后缀。默认值为“0”。

溪流，小号

指定要读取的流。可以指定多个流，用“+”分隔。然后源将以相同的顺序具有尽可能多的输出。语法[在 ffmpeg 手册的 (ffmpeg)“Stream specifiers”部分进行了解释](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg.html#Stream-specifiers)。两个特殊名称“dv”和“da”分别指定默认（最适合）视频和音频流。如果过滤器被称为“电影”，则默认为“dv”或“da”。

stream\_index, si

指定要读取的视频流的索引。如果值为-1，将自动选择最合适的视频流。默认值为“-1”。已弃用。如果过滤器名为“电影”，它将选择音频而不是视频。

环形

指定顺序读取流的次数。如果该值为 0，则流将无限循环。默认值为“1”。

请注意，当影片循环播放时，源时间戳不会更改，因此它会生成非单调递增的时间戳。

间断性

指定帧之间的时间差，高于该时间点的点被视为时间戳不连续性，通过调整后面的时间戳将其删除。

dec\_线程

指定解码线程数

格式选项

为打开的文件指定格式选项。格式选项可以指定为由“:”分隔的键=值对列表。以下示例显示如何添加 protocol\_whitelist 和 protocol\_blacklist 选项：

ffplay -f lavfi

“movie=filename='1.sdp':format\_opts='protocol\_whitelist=file,rtp,udp\:protocol\_blacklist=http'”

它允许在过滤图的主要输入之上叠加第二个视频，如下图所示：

输入 ----------> deltapts0 --> 覆盖 --> 输出

^

|

电影 --> scale--> deltapts1 ------+

* [例子](#Examples-211)
* [命令](#Commands-161)

#### 46.3.1 例子

* 从 AVI 文件 in.avi 的开头跳过 3.2 秒，并将其覆盖在标有“in”的输入之上：
* movie=in.avi:seek\_point=3.2, scale=180:-1, setpts=PTS-STARTPTS [over];
* [in] setpts=PTS-STARTPTS [main];
* [main][over] overlay=16:16 [out]
* 从 video4linux2 设备读取，并将其覆盖在标有“in”的输入之上：
* movie=/dev/video0:f=video4linux2, scale=180:-1, setpts=PTS-STARTPTS [over];
* [in] setpts=PTS-STARTPTS [main];
* [main][over] overlay=16:16 [out]
* 从dvd.vob中读取第一个视频流和id为0x81的音频流；视频连接到名为“video”的 pad，音频连接到名为“audio”的 pad：
* 电影=dvd.vob:s=v:0+#0x81 [视频] [音频]

#### 46.3.2 命令

movie 和 amovie 都支持以下命令：

寻找

使用“av\_seek\_frame”执行搜索。语法是： seek stream\_index |时间戳|旗帜

* stream\_index ：如果 stream\_index 为 -1，则选择默认流，并且时间戳自动从 AV\_TIME\_BASE 单位转换为流特定的 time\_base。
* timestamp ：以 AVStream.time\_base 为单位的时间戳，如果未指定流，则以 AV\_TIME\_BASE 为单位。
* flags : 选择方向和搜索模式的标志。

获取持续时间

以 AV\_TIME\_BASE 为单位获取电影持续时间。

## 47 个外部图书馆

FFmpeg 可以与许多外部库连接以添加对更多格式的支持。默认情况下都不使用它们，必须通过将适当的标志传递给./configure来明确请求它们的使用。

* [开放媒体联盟 (AOM)](#Alliance-for-Open-Media-_0028AOM_0029)
* [AMD AMF/VCE](#AMD-AMF_002fVCE)
* [合成器](#AviSynth)
* [色谱图](#Chromaprint)
* [编解码器2](#codec2)
* [dav1d](#dav1d)
* [戴维斯2](#davs2)
* [无人机3d](#uavs3d)
* [游戏音乐鸸鹋](#Game-Music-Emu)
* [Intel QuickSync 视频](#Intel-QuickSync-Video)
* [克瓦扎尔](#Kvazaar)
* [瘸](#LAME)
* [libilbc](#libilbc-1)
* [libjxl](#libjxl-1)
* [libvpx](#libvpx-1)
* [模组插件](#ModPlug)
* [OpenCORE、VisualOn 和 Fraunhofer 库](#OpenCORE_002c-VisualOn_002c-and-Fraunho)
* [OpenH264](#OpenH264)
* [打开JPEG](#OpenJPEG)
* [rav1e](#rav1e)
* [SVT-AV1](#SVT_002dAV1)
* [两个LAME](#TwoLAME)
* [蒸汽合成器](#VapourSynth)
* [x264](#x264)
* [x265](#x265)
* [xavs](#xavs)
* [xavs2](#xavs2)
* [ZVBI](#ZVBI)

### 47.1 开放媒体联盟 (AOM)

FFmpeg 可以利用 AOM 库进行 AV1 解码和编码。

转到<http://aomedia.org/>并按照说明安装库。然后通过--enable-libaom进行配置以启用它。

### 47.2 AMD AMF/VCE

FFmpeg 可以使用 AMD 高级媒体框架库在带有视频编码引擎 (VCE) 的硬件上加速 H.264 和 HEVC（仅限 Windows）编码。

<https://github.com/GPUOpen-LibrariesAndSDKs/AMF.git>获取 AMF 框架头文件（版本 1.4.9+） 。

在系统包含路径中创建一个AMF/目录。将AMF/amf/public/include/的内容复制到该目录中。然后使用--enable-amf配置 FFmpeg 。

amf 编码器的初始化按以下顺序进行：1) 尝试通过 dx11（仅限 windows）进行初始化 2）尝试通过 dx9（仅限 windows）进行初始化 3）尝试通过 vulkan 进行初始化

要在 linux 上使用 h.264(AMD VCE) 编码器，需要 amdgru-pro 版本 19.20+ 和 amf-amdgpu-pro 包（amdgru-pro 包含，但不会自动安装）。

可以使用官方 amd 驱动程序存档中的 amdgpu-pro-install 脚本安装此驱动程序。

### 47.3 Avi合成器

FFmpeg 可以读取 AviSynth 脚本作为输入。要启用支持，请在安装[AviSynth+](https://github.com/AviSynth/AviSynthPlus)提供的标头后通过--enable-avisynth进行配置。 AviSynth+ 可以配置为仅安装标头，方法是将-DHEADERS\_ONLY:bool=on传递给基于 CMake 的普通构建系统，或使用提供的GNUmakefile 。

对于 Windows，支持的 AviSynth 变体是用于 32 位构建的[AviSynth 2.6 RC1 或更高版本，以及](http://avisynth.nl)用于 32 位和 64 位构建的[AviSynth+ r1718 或更高版本。](http://avisynth.nl/index.php/AviSynth+)

对于 Linux、macOS 和 BSD，唯一受支持的 AviSynth 变体是[AviSynth+](https://github.com/AviSynth/AviSynthPlus) ，从版本 3.5 开始。

2016 年，AviSynth+ 增加了对使用 GCC 构建的支持。然而，由于 Windows 调用约定的古怪之处，AviSynth+ 的 32 位 GCC 构建与典型的 32 位 FFmpeg 构建不兼容。

默认情况下，FFmpeg 假定与 AviSynth+ 的 32 位 MSVC 构建兼容，因为这是使用最广泛和根深蒂固的构建配置。用户可以通过在配置 FFmpeg 时将-DAVSC\_WIN32\_GCC32传递给--extra-cflags来覆盖它并启用对 AviSynth+ 的 32 位 GCC 构建的支持。

FFmpeg 的 64 位构建不受影响，并且可以使用 AviSynth+ 的 MSVC 或 GCC 构建而无需任何特殊标志。

**笔记**

AviSynth(+) 是动态加载的。发行商可以使用--enable-avisynth构建 FFmpeg ，无论最终用户是否安装了 AviSynth，二进制文件都可以工作。如果/当最终用户想要使用 AviSynth 脚本时，他们可以安装 AviSynth(+)，FFmpeg 将能够找到并使用它来打开脚本。

**笔记**

### 47.4 色谱图

FFmpeg 可以使用 Chromaprint 库来生成音频指纹。通过--enable-chromaprint进行配置以启用它。请参阅<https://acoustid.org/chromaprint>。

### 47.5编解码器2

FFmpeg 可以利用 codec2 库进行 codec2 解码和编码。目前没有原生解码器，所以必须使用 libcodec2 进行解码。

转到<http://freedv.org/>，下载“Codec 2 source archive”。使用 CMake 构建和安装。 Debian 用户可以安装 libcodec2-dev 软件包。安装 libcodec2 后，您可以通过--enable-libcodec2进行配置以启用它。

使用 codec2 的最简单方法是使用 .c2 文件，因为它们包含解码所需的模式信息。要对此类文件进行编码，请使用 .c2 文件扩展名并为 libcodec2 编码器提供 -mode 选项： ffmpeg -i input.wav -mode 700C output.c2 。回放就像ffplay output.c2一样简单。有关支持模式的列表，请运行ffmpeg -h encoder=libcodec2 。还支持原始编解码器 2 文件。为了理解它们，需要将使用的模式指定为格式选项： ffmpeg -f codec2raw -mode 1300 -i input.raw output.wav 。

### 47.6 dav1d

FFmpeg 可以使用 dav1d 库进行 AV1 视频解码。

转到<https://code.videolan.org/videolan/dav1d>并按照安装库的说明进行操作。然后通过--enable-libdav1d进行配置以启用它。

### 47.7 戴维斯 2

FFmpeg 可以利用 davs2 库进行 AVS2-P2/IEEE1857.4 视频解码。

转到<https://github.com/pkuvcl/davs2>并按照安装库的说明进行操作。然后通过--enable-libdavs2进行配置以启用它。

libdavs2 在 GNU Public License Version 2 或更高版本下（详见<http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>），你必须将 FFmpeg 的许可证升级到 GPL 才能使用它.

**笔记**

### 47.8 无人机 3d

FFmpeg 可以利用 uavs3d 库进行 AVS3-P2/IEEE1857.10 视频解码。

转到<https://github.com/uavs3/uavs3d>并按照安装库的说明进行操作。然后通过--enable-libuavs3d进行配置以启用它。

### 47.9游戏音乐鸸鹋

FFmpeg 可以使用 Game Music Emu 库从支持的视频游戏音乐文件格式中读取音频。通过--enable-libgme进行配置以启用它。请参阅<https://bitbucket.org/mpyne/game-music-emu/overview>。

### 47.10 Intel QuickSync 视频

FFmpeg 可以使用 Intel QuickSync Video (QSV) 来加速多种编解码器的解码和编码。要使用 QSV，FFmpeg 必须与加载实际解码库的libmfx调度程序链接。

调度程序是开源的，可以从[https://github.com/lu-zero/mfx\_dispatch.git下载](https://github.com/lu-zero/mfx_dispatch.git)。 FFmpeg 需要使用--enable-libmfx选项进行配置，并且pkg-config需要能够找到调度程序的.pc文件。

### 47.11 克瓦扎尔

FFmpeg 可以使用 Kvazaar 库进行 HEVC 编码。

转到<https://github.com/ultravideo/kvazaar>并按照安装库的说明进行操作。然后通过--enable-libkvazaar进行配置以启用它。

### 47.12跛脚

FFmpeg 可以使用 LAME 库进行 MP3 编码。

转到<http://lame.sourceforge.net/>并按照安装库的说明进行操作。然后通过--enable-libmp3lame进行配置以启用它。

### 47.13 libilbc

iLBC 是一种窄带语音编解码器，作为 WebRTC 项目的一部分由 Google 免费提供。 libilbc 是 iLBC 编解码器的打包友好副本。 FFmpeg 可以利用 libilbc 库进行 iLBC 解码和编码。

转到<https://github.com/TimothyGu/libilbc>并按照安装库的说明进行操作。然后通过--enable-libilbc进行配置以启用它。

### 47.14 libjxl

JPEG XL 是一种图像格式，旨在在更长的生命周期内完全取代传统的 JPEG。有关详细信息，请参阅[https://jpegxl.info/ ，](https://jpegxl.info/)有关库源，请参阅[https://github.com/libjxl/libjxl 。](https://github.com/libjxl/libjxl)您可以通过--enable-libjxl进行配置以启用 libjxl 包装器。

### 47.15 libvpx

FFmpeg 可以利用 libvpx 库进行 VP8/VP9 解码和编码。

转到<http://www.webmproject.org/>并按照安装库的说明进行操作。然后通过--enable-libvpx进行配置以启用它。

### 47.16 调制插件

FFmpeg 可以利用这个源自 Modplug-XMMS 的库来读取类 MOD 音乐文件。请参阅<https://github.com/Konstanty/libmodplug>。通过--enable-libmodplug进行配置以启用它。

### 47.17 OpenCORE、VisualOn 和 Fraunhofer 库

从 Google Android 源代码中分离出来，OpenCore、VisualOn 和 Fraunhofer 库为许多音频编解码器提供编码器。

OpenCORE 和 VisualOn 库在 Apache License 2.0 下（详见<http://www.apache.org/licenses/LICENSE-2.0>），与 LGPL 2.1 版和 GPL 2 版不兼容。您必须升级 FFmpeg 的许可证通过传递--enable-version3到 LGPL 版本 3（或者如果你启用了 GPL 组件，GPL 版本 3）以使用它。

Fraunhofer AAC 库的许可证与 GPL 不兼容。因此，对于 GPL 构建，您必须通过--enable-nonfree进行配置才能使用它。据我们所知，它与 LGPL 兼容。

**笔记**

* [开放核心 AMR](#OpenCORE-AMR)
* [VisualOn AMR-WB 编码器库](#VisualOn-AMR_002dWB-encoder-library)
* [弗劳恩霍夫 AAC 库](#Fraunhofer-AAC-library)

#### 47.17.1 开放核心 AMR

FFmpeg 可以利用 OpenCORE 库进行 AMR-NB 解码/编码和 AMR-WB 解码。

转到<http://sourceforge.net/projects/opencore-amr/>并按照安装库的说明进行操作。然后通过--enable-libopencore-amrnb和/或--enable-libopencore-amrwb进行配置以启用它们。

#### 47.17.2 VisualOn AMR-WB 编码器库

FFmpeg 可以使用 VisualOn AMR-WBenc 库进行 AMR-WB 编码。

转到<http://sourceforge.net/projects/opencore-amr/>并按照安装库的说明进行操作。然后通过--enable-libvo-amrwbenc进行配置以启用它。

#### 47.17.3 弗劳恩霍夫 AAC 库

FFmpeg 可以利用 Fraunhofer AAC 库进行 AAC 解码和编码。

转到<http://sourceforge.net/projects/opencore-amr/>并按照安装库的说明进行操作。然后通过--enable-libfdk-aac进行配置以启用它。

### 47.18 OpenH264

FFmpeg 可以利用 OpenH264 库进行 H.264 解码和编码。

转到<http://www.openh264.org/>并按照安装库的说明进行操作。然后通过--enable-libopenh264进行配置以启用它。

对于解码，这个库比 libavcodec 中的内置解码器要受限得多；目前，这个库不支持解码 B 帧和其他一些主要/高级功能。 （它目前仅支持受约束的基线配置文件和 CABAC。）使用它主要用于测试和利用 Cisco 的专利组合许可 ( <http://www.openh264.org/BINARY_LICENSE.txt>)。

### 47.19 OpenJPEG

FFmpeg 可以使用 OpenJPEG 库对 J2K 视频进行解码/编码。转到<http://www.openjpeg.org/>获取库并按照安装说明进行操作。要在 FFmpeg 中启用 OpenJPEG，请将--enable-libopenjpeg传递给./configure 。

### 47.20 rav1e

FFmpeg 可以通过其 C 绑定使用 rav1e（Rust AV1 编码器）来编码视频。转到<https://github.com/xiph/rav1e/>并按照说明构建 C 库。要在 FFmpeg 中启用 rav1e，请将--enable-librav1e传递给./configure 。

### 47.21 SVT-AV1

FFmpeg 可以使用 Scalable Video Technology for AV1 library 进行 AV1 编码。

转到<https://gitlab.com/AOMediaCodec/SVT-AV1/>并按照安装库的说明进行操作。然后通过--enable-libsvtav1进行配置以启用它。

### 47.22 TwoLAME

FFmpeg 可以使用 TwoLAME 库进行 MP2 编码。

转到<http://www.twolame.org/>并按照说明安装库。然后通过--enable-libtwolame进行配置以启用它。

### 47.23 VapourSynth

FFmpeg 可以读取 VapourSynth 脚本作为输入。要启用支持，请将--enable-vapoursynth传递给配置。 Vapoursynth 是通过pkg-config检测到的。支持 42 或更高版本。请参阅<http://www.vapoursynth.com/>。

出于安全考虑，不会自动检测 Vapoursynth 脚本，因此必须强制输入格式。对于 ff\* CLI 工具，在输入-i yourscript.vpy之前添加-f vapoursynth 。

### 47.24 x264

FFmpeg 可以使用 x264 库进行 H.264 编码。

转到<http://www.videolan.org/developers/x264.html>并按照说明安装库。然后通过--enable-libx264进行配置以启用它。

x264 是在 GNU Public License Version 2 或更高版本下（详见<http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>），你必须将 FFmpeg 的许可证升级到 GPL 才能使用它.

**笔记**

### 47.25 x265

FFmpeg 可以使用 x265 库进行 HEVC 编码。

转到<http://x265.org/developers.html>并按照安装库的说明进行操作。然后通过--enable-libx265进行配置以启用它。

x265 是在 GNU Public License Version 2 或更高版本下（详见<http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>），你必须将 FFmpeg 的许可证升级到 GPL 才能使用它.

**笔记**

### 47.26 xavs

FFmpeg 可以使用 xavs 库进行 AVS 编码。

转到<http://xavs.sf.net/>并按照安装库的说明进行操作。然后通过--enable-libxavs进行配置以启用它。

### 47.27 xavs2

FFmpeg 可以使用 xavs2 库进行 AVS2-P2/IEEE1857.4 视频编码。

转到<https://github.com/pkuvcl/xavs2>并按照安装库的说明进行操作。然后通过--enable-libxavs2进行配置以启用它。

libxavs2 是在 GNU Public License Version 2 或更高版本下（详见<http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>），你必须将 FFmpeg 的许可证升级到 GPL 才能使用它.

**笔记**

### 47.28 ZVBI

ZVBI 是一个 VBI 解码库，FFmpeg 可以使用它来解码 DVB 图文电视页面和 DVB 图文电视字幕。

转到<http://sourceforge.net/projects/zapping/>并按照安装库的说明进行操作。然后通过--enable-libzvbi进行配置以启用它。

## 48 种支持的文件格式、编解码器或功能

您可以使用-formats和-codecs选项来获得详尽的列表。

* [文件格式](#File-Formats)
* [图像格式](#Image-Formats)
* [视频编解码器](#Video-Codecs)
* [音频编解码器](#Audio-Codecs)
* [字幕格式](#Subtitle-Formats)
* [网络协议](#Network-Protocols)
* [输入/输出设备](#Input_002fOutput-Devices)
* [时间码](#Timecode)

### 48.1 文件格式

libavformat库支持以下文件格式：

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 编码 | 解码 | 评论 |
| 3dostr |  | X |  |
| 4x米 |  | X | 4X 技术格式，用于某些游戏。 |
| 8088flex TMV |  | X |  |
| AAX |  | X | Audible Enhanced Audio 格式，用于有声读物。 |
| AA |  | X | 有声读物中使用的 Audible Format 2、3 和 4。 |
| ACT语音 |  | X | 包含 G.729 音频 |
| Adobe 幻灯片 | X | X |  |
| 音频敌我识别 (AIFF) | X | X |  |
| 美国激光游戏MM |  | X | Mad Dog McCree 等游戏中使用的多媒体格式。 |
| 3GPP AMR | X | X |  |
| 惊人的工作室打包动画文件 |  | X | 游戏《黑暗之心》中使用的多媒体格式。 |
| 苹果 HTTP 直播 |  | X |  |
| Artworx 数据格式 |  | X |  |
| 相互作用的ACM |  | X | 某些 Interplay 游戏中使用的纯音频格式。 |
| ADP |  | X | Nintendo Gamecube 上使用的音频格式。 |
| 亚足联 |  | X | Nintendo Gamecube 上使用的音频格式。 |
| ADS/SS2 |  | X | PS2 上使用的音频格式。 |
| APNG | X | X |  |
| 非洲猪瘟 | X | X | 高级/活动流格式。 |
| AST | X | X | Nintendo Wii 上使用的音频格式。 |
| 视频文件 | X | X |  |
| 合成器 |  | X |  |
| 自动录像机 |  | X | Mac 上使用的音频格式。 |
| 自动驾驶系统 |  | X | Creature Shock 游戏使用的多媒体格式。 |
| Beam 软件上海国际电影节 |  | X | Beam Software 在某些游戏中使用的音频和视频格式。 |
| 贝塞斯达软件 VID |  | X | 在 Bethesda Softworks 的一些游戏中使用。 |
| 二进制文本 |  | X |  |
| 宾克 |  | X | 许多游戏使用的多媒体格式。 |
| 宾克音频 |  | X | 某些游戏使用的纯音频多媒体格式。 |
| 位图兄弟合资公司 |  | X | 用于 Z 和 Z95 游戏。 |
| BRP |  | X | 阿尔戈游戏格式。 |
| 蛮力与无知 |  | X | 用于游戏 Flash Traffic：天使之城。 |
| BFSTM |  | X | Nintendo WiiU 上使用的音频格式（基于 BRSTM）。 |
| BRSTM |  | X | Nintendo Wii 上使用的音频格式。 |
| BW64 |  | X | 广播波 64 位。 |
| 世界羽联 | X | X |  |
| 编解码器 2（原始） | X | X | 必须给出 -mode 格式选项才能正确解码。 |
| 编解码器 2（.c2 文件） | X | X | 包含带有版本和模式信息的标头，简化播放。 |
| 显色指数ADX | X | X | 控制台视频游戏中使用的纯音频格式。 |
| 爱克斯 |  | X |  |
| CRI HCA |  | X | 控制台视频游戏中使用的纯音频格式。 |
| 唱片世界 II BMV |  | X |  |
| 互动C93 |  | X | 在 Interplay 的游戏 Cyberia 中使用。 |
| Delphine 软件国际 CIN |  | X | Delphine Software 游戏使用的多媒体格式。 |
| 数字语音标准 (DSS) |  | X |  |
| CD+G |  | X | CD+G卡拉OK盘使用的视频格式 |
| 幻影电影 |  | X |  |
| 准将CDXL |  | X | Amiga CD 视频格式 |
| 核心音频格式 | X | X | 苹果核心音频格式 |
| CRC测试格式 | X |  |  |
| 创意之声 | X | X | 为 Sound Blaster Pro 创建。 |
| 低温APC |  | X | CRYO Interactive Entertainment 在某些游戏中使用的音频格式。 |
| D-影院音频 | X | X |  |
| 豪华油漆动画 |  | X |  |
| DCSTR |  | X |  |
| DFA |  | X | 这种格式用于 Chronomaster 游戏 |
| DirectDraw表面 |  | X |  |
| DSD 流文件 (DSF) |  | X |  |
| DV视频 | X | X |  |
| 数字XA |  | X | 此格式用于非 Windows 版本的 Feeble Files 游戏和重新打包以用于 ScummVM 的不同游戏过场动画。 |
| 艺电数据 |  | X |  |
| 艺电多媒体 |  | X | 用于各种EA游戏；文件具有 WVE 和 UV2 等扩展名。 |
| Ensoniq 巴黎音频文件 |  | X |  |
| FFM（FFserver 直播） | X | X |  |
| 闪光灯 (SWF) | X | X |  |
| 闪存 9 (AVM2) | X | X | 只有嵌入的音频被解码。 |
| FLI/FLC/FLX动画 |  | X | .fli/.flc 文件 |
| Flash 视频 (FLV) | X | X | Macromedia Flash 视频文件 |
| framecrc 测试格式 | X |  |  |
| FunCom ISS |  | X | FunCom 的各种游戏中使用的音频格式，例如 The Longest Journey。 |
| G.723.1 | X | X |  |
| G.726 |  | X | 左对齐和右对齐。 |
| G.729比特 | X | X |  |
| G.729 原始 |  | X |  |
| GENH |  | X | 各种游戏的音频格式。 |
| GIF动画 | X | X |  |
| 广信丰 | X | X | General eXchange Format SMPTE 360M，由 Thomson Grass Valley 播出服务器使用。 |
| 汉诺威 |  | X | 仅支持版本 4，用于 Cryo Interactive 的某些游戏 |
| iCEDraw文件 |  | X |  |
| 首次代币发售 | X | X | 微软视窗ICO |
| id 雷神之锤 II CIN 视频 |  | X |  |
| 身份证号 | X | X | 用于雷神之锤 III、绝地武士 2 和其他电脑游戏。 |
| IEC61937封装 | X | X |  |
| 敌对部队 |  | X | 交换文件格式 |
| 步兵战车 |  | X | 一些旧的 CCTV DVR 使用的格式。 |
| iLBC | X | X |  |
| 互动MVE |  | X | 用于各种 Interplay 电脑游戏的格式。 |
| 迭代系统 ClearVideo |  | X | 仅 I 帧 |
| IV8 |  | X | IndigoVision 8000 视频服务器生成的一种格式。 |
| 体外受精（On2） | X | X | libvpx 使用的一种格式 |
| 网络视频录制 |  | X |  |
| 红外摄像头 | X | X |  |
| 拉夫 |  | X | 无限的音频格式 |
| 拉丁美洲 | X | X |  |
| LMLM4 |  | X | 由 Linux Media Labs MPEG-4 PCI 板使用 |
| LOAS |  | X | 包含 LATM 复用 AAC 音频 |
| 法改会 | X | X |  |
| LVF |  | X |  |
| LXF |  | X | Leitch/Harris 的视频服务器使用的 VR 原生流格式。 |
| 魔灯视频 (MLV) |  | X |  |
| 马特罗斯卡 | X | X |  |
| Matroska 音频 | X |  |  |
| FFmpeg 元数据 | X | X | 文本格式的元数据。 |
| 明讯XA |  | X | 用于模拟城市 3000；文件扩展名 .xa。 |
| 马华 |  | X | 在 Capcom 的一些游戏中使用；文件扩展名 .mca。 |
| MD工作室 |  | X |  |
| 合金装备：双蛇 |  | X |  |
| Megalux 框架 |  | X | 由 Megalux Ultimate Paint 使用 |
| MobiClip MODS |  | X |  |
| MobiClip MOFLEX |  | X |  |
| 莫博蒂克斯.mxg |  | X |  |
| 猴子的音频 |  | X |  |
| 运动像素 MVI |  | X |  |
| MOV/QuickTime/MP4 | X | X | 支持 3GP、3GP2、PSP、iPod 变体 |
| MP2 | X | X |  |
| MP3 | X | X |  |
| MPEG-1系统 | X | X | 多路音频和视频，支持 VCD 格式 |
| MPEG-PS（节目流） | X | X | 也称为VOB文件，支持 SVCD 和 DVD 格式 |
| MPEG-TS（传输流） | X | X | 也称为 DVB 传输流 |
| MPEG-4 | X | X | MPEG-4 是 QuickTime 的变体。 |
| 无国界医生组织 |  | X | PS3 上使用的音频格式。 |
| Mirillis FIC 视频 |  | X | 没有光标渲染。 |
| MIDI 样本转储标准 |  | X |  |
| MIME 多部分 JPEG | X |  |  |
| MSN TCP 网络摄像头 |  | X | 由 MSN Messenger 网络摄像头流使用。 |
| 音乐电视 |  | X |  |
| 音乐包 |  | X |  |
| Musepack SV8 |  | X |  |
| 材料交换格式 (MXF) | X | X | SMPTE 377M，用于数字影院、广播行业。 |
| 材料交换格式 (MXF)，D-10 映射 | X | X | SMPTE 386M、D-10/IMX 映射。 |
| NC 相机进给 |  | X | NC (AVIP NC4600) 摄像机流 |
| NIST 语音标题资源 |  | X |  |
| 计算机化语音实验室 NSP |  | X |  |
| NTT TwinVQ (VQF) |  | X | 日本电报电话公司 TwinVQ。 |
| Nullsoft 流媒体视频 |  | X |  |
| Nuppel视频 |  | X |  |
| 坚果 | X | X | NUT 开放容器格式 |
| 奥格 | X | X |  |
| Playstation 便携式 PMP |  | X |  |
| 便携式语音格式 |  | X |  |
| RK 音频 (RKA) |  | X |  |
| TechnoTrend PVA |  | X | 由 TechnoTrend DVB PCI 板使用。 |
| 质量控制计划 |  | X |  |
| 原始 ADTS (AAC) | X | X |  |
| 原料 AC-3 | X | X |  |
| 原始AMR-NB |  | X |  |
| 原始 AMR-WB |  | X |  |
| 原始亚太地区 |  | X |  |
| 原始aptX | X | X |  |
| 原始aptX高清 | X | X |  |
| 原始邦克 |  | X |  |
| 原始中文 AVS 视频 | X | X |  |
| 原始DFPWM | X | X |  |
| 原始狄拉克 | X | X |  |
| 原始 DNxHD | X | X |  |
| 原始DTS | X | X |  |
| 原始DTS-HD |  | X |  |
| 生E-AC-3 | X | X |  |
| 原始 FLAC | X | X |  |
| 原始GSM |  | X |  |
| 原始 H.261 | X | X |  |
| 原始 H.263 | X | X |  |
| 原始 H.264 | X | X |  |
| 原始HEVC | X | X |  |
| 原始的 Ingenient MJPEG |  | X |  |
| 原始MJPEG | X | X |  |
| 原始 MLP |  | X |  |
| 原始MPEG |  | X |  |
| 原始 MPEG-1 |  | X |  |
| 原始 MPEG-2 |  | X |  |
| 原始 MPEG-4 | X | X |  |
| 原始空值 | X |  |  |
| 原始视频 | X | X |  |
| 原始编号 RoQ | X |  |  |
| 原始OBU | X | X |  |
| 原始单板计算机 | X | X |  |
| 原始缩短 |  | X |  |
| 原始TAK |  | X |  |
| 原始高清 | X | X |  |
| 原始VC-1 | X | X |  |
| 原始 PCM A 律 | X | X |  |
| 原始 PCM mu-law | X | X |  |
| 原始 PCM 阿基米德 VIDC | X | X |  |
| 原始 PCM 签名 8 位 | X | X |  |
| 原始 PCM 签名 16 位大端 | X | X |  |
| 原始 PCM 签名 16 位小端 | X | X |  |
| 原始 PCM 签名 24 位大端 | X | X |  |
| 原始 PCM 签名 24 位小端 | X | X |  |
| 原始 PCM 签名 32 位大端 | X | X |  |
| 原始 PCM 签名 32 位小端 | X | X |  |
| 原始 PCM 签名 64 位大端 | X | X |  |
| 原始 PCM 签名 64 位小端 | X | X |  |
| 原始 PCM 无符号 8 位 | X | X |  |
| 原始 PCM 无符号 16 位大端 | X | X |  |
| 原始 PCM 无符号 16 位小端 | X | X |  |
| 原始 PCM 无符号 24 位大端 | X | X |  |
| 原始 PCM 无符号 24 位小端 | X | X |  |
| 原始 PCM 无符号 32 位大端 | X | X |  |
| 原始 PCM 无符号 32 位小端 | X | X |  |
| 原始 PCM 16.8 浮点小端 |  | X |  |
| 原始 PCM 24.0 浮点小端 |  | X |  |
| 原始 PCM 浮点 32 位大端 | X | X |  |
| 原始 PCM 浮点 32 位小端 | X | X |  |
| 原始 PCM 浮点 64 位大端 | X | X |  |
| 原始 PCM 浮点 64 位小端 | X | X |  |
| RDT |  | X |  |
| 红码 R3D |  | X | RED 数码相机使用的文件格式，包含 JPEG 2000 帧和 PCM 音频。 |
| 真实媒体 | X | X |  |
| 重定向器 |  | X |  |
| 红火 |  | X |  |
| 渲染软件 TeXture 词典 |  | X |  |
| 分辨率 DXV |  | X |  |
| RF64 |  | X |  |
| RL2 |  | X | 娱乐软件合作伙伴在某些游戏中使用的音频和视频格式。 |
| RPL/ARM电影 |  | X |  |
| 乐高头脑风暴 RSO | X | X |  |
| 相对标准差 |  | X |  |
| 实时播放协议 | X | X | 通过发布流到 RTMP 服务器进行输出 |
| 实时传输协议 | X | X |  |
| RTSP协议 | X | X |  |
| 样本转储交换 |  | X |  |
| 树液 | X | X |  |
| SBG |  | X |  |
| SDNS |  | X |  |
| 开发计划署 |  | X |  |
| 序列号 |  | X |  |
| 数字图片 SGA |  | X |  |
| 世嘉电影/CPK | X | X | 用于许多 Sega Saturn 控制台游戏。 |
| 硅片电影 |  | X |  |
| 塞拉索尔 |  | X | Sierra 在线游戏中使用的 .sol 文件。 |
| 塞拉利昂VMD |  | X | 用于 Sierra CD-ROM 游戏。 |
| 斯马克 |  | X | 许多游戏使用的多媒体格式。 |
| SMJPEG格式 | X | X | 用于某些 Loki 游戏端口。 |
| SMPTE 337M封装 |  | X |  |
| 粉碎 |  | X | 一些 LucasArts 游戏中使用的多媒体格式。 |
| 索尼 OpenMG（OMA） | X | X | Sony Sonic Stage 和 Sony Vegas 使用的音频格式。 |
| 索尼 PlayStation STR |  | X |  |
| 索尼 Wave64 (W64) | X | X |  |
| SoX 原生格式 | X | X |  |
| SUN AU 格式 | X | X |  |
| SUP 原始 PGS 字幕 | X | X |  |
| SVAG |  | X | Konami PS2 游戏中使用的音频格式。 |
| TDSC |  | X |  |
| 文本文件 |  | X |  |
| 热泵 |  | X | 用于任天堂 GameCube。 |
| Tiertex 有限公司 SEQ |  | X | 在游戏 Flashback 的 DOS CD-ROM 版本中使用的 Tiertex .seq 文件。 |
| 真实音频 | X | X |  |
| VAG |  | X | 许多索尼 PS2 游戏中使用的音频格式。 |
| VC-1 测试比特流 | X | X |  |
| 维德沃克斯哈普 | X | X |  |
| 活体 |  | X |  |
| VPK |  | X | Sony PS 游戏中使用的音频格式。 |
| 大理石瓦迪 |  | X |  |
| WAV格式 | X | X |  |
| 波形存档器 |  | X |  |
| 音频包 | X | X |  |
| WebM | X | X |  |
| Windows 电视 (WTV) | X | X |  |
| 联队指挥官 III 电影 |  | X | Origin 的 Wing Commander III 电脑游戏中使用的多媒体格式。 |
| 韦斯特伍德工作室音频 | X | X | Westwood Studios 游戏中使用的多媒体格式。 |
| 韦斯特伍德工作室 VQA |  | X | Westwood Studios 游戏中使用的多媒体格式。 |
| 宽带单位数据 (WSD) |  | X |  |
| WVE |  | X |  |
| 科乐美XMD |  | X |  |
| XMV |  | X | Xbox 游戏中使用的 Microsoft 视频容器。 |
| XVAG |  | X | PS3 上使用的音频格式。 |
| xWMA |  | X | XAudio 2 使用的 Microsoft 音频容器。 |
| 扩展二进制文本 (XBIN) |  | X |  |
| YUV4MPEG管道 | X | X |  |
| 症状 YOP |  | X |  |

X表示支持该列中的功能（编码/解码）。

### 48.2 图像格式

FFmpeg 可以为视频序列的每一帧读取和写入图像。支持以下图像格式：

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 编码 | 解码 | 评论 |
| .YUV | X | X | 每个组件一个原始文件 |
| 别名 PIX | X | X | Alias/Wavefront PIX 图像格式 |
| 动画 GIF | X | X |  |
| APNG | X | X | 动画便携式网络图形 |
| 骨形态发生蛋白 | X | X | 微软 BMP 图片 |
| 渲染器 PIX |  | X | Argonaut BRender 3D 引擎图像格式。 |
| 显色指数 |  | X | 辛特尔RAW |
| DPX | X | X | 数字图片交换 |
| EXR |  | X | OpenEXR |
| 适合 | X | X | 灵活的图像传输系统 |
| 高动态范围 | X | X | Radiance HDR RGBE 图像格式 |
| IMG |  | X | GEM 光栅图像 |
| JPEG格式 | X | X | 不支持渐进式 JPEG。 |
| JPG 2000 | X | X |  |
| JPEG格式 | X | X |  |
| LJPEG格式 | X |  | 无损 JPEG |
| 媒体100 |  | X |  |
| MSP |  | X | 微软画图图像 |
| PAM | X | X | PAM 是具有 alpha 支持的 PNM 扩展。 |
| PBM | X | X | 便携式位图图像 |
| PCD |  | X | 光碟 |
| PCX | X | X | PC画笔 |
| PFM | X | X | 便携式 FloatMap 图像 |
| 铂族金属 | X | X | 便携式 GrayMap 图像 |
| PGMYUV | X | X | 带有 YUV 4:2:0 中 U 和 V 分量的 PGM |
| PGX |  | X | PGX文件解码器 |
| PHM | X | X | 便携式 HalfFloatMap 图像 |
| 事先知情同意书 |  | X | Pictor/PC 油漆 |
| PNG | X | X | 便携式网络图形图像 |
| 百万分率 | X | X | 便携式像素图图像 |
| PSD |  | X | Photoshop |
| PTX |  | X | 五、Flash PTX格式 |
| 生活质量指数 | X | X | 还行 图片格式 |
| SGI | X | X | SGI RGB 图像格式 |
| 太阳光栅文件 | X | X | Sun RAS 图像格式 |
| 国际电影节 | X | X | 尚不支持 YUV、JPEG 和一些扩展名。 |
| 真视塔加 | X | X | Targa (.TGA) 图像格式 |
| VBN | X | X | Vizrt 二进制图像格式 |
| WBMP | X | X | 无线应用协议位图图像格式 |
| WebP | 乙 | X | WebP 图片格式，通过外部库 libwebp 支持编码 |
| XBM | X | X | X 位图图像格式 |
| X脸 | X | X | X-Face图像格式 |
| XPM |  | X | X PixMap 图像格式 |
| 新伟达 | X | X | X 窗口转储图像格式 |

X表示支持该列中的功能（编码/解码）。

E表示通过外部库提供支持。

### 48.3 视频编解码器

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 编码 | 解码 | 评论 |
| 4X电影 |  | X | 用于某些电脑游戏。 |
| 8088flex TMV |  | X |  |
| A64多色 | X |  | 创建适合在 commodore 64（多色模式）上播放的视频。 |
| 惊人的工作室 PAF 视频 |  | X |  |
| 美国激光游戏MM |  | X | 用于 Mad Dog McCree 等游戏。 |
| 娱乐图形电影 |  | X |  |
| 视频 | X | X | 用于中国的MP3播放器。 |
| ANSI/ASCII 艺术 |  | X |  |
| Apple 中级编解码器 |  | X |  |
| 苹果 MJPEG-B |  | X |  |
| 苹果像素 |  | X |  |
| 苹果 ProRes | X | X | fourcc: apch,apcn,apcs,apco,ap4h,ap4x |
| 苹果快绘 |  | X | fourcc: qdrw |
| 阿尔戈视频 |  | X | 在一些阿尔戈英雄游戏中使用。 |
| 华硕v1 | X | X | fourcc: ASV1 |
| 华硕v2 | X | X | fourcc: ASV2 |
| ATI VCR1 |  | X | fourcc: VCR1 |
| ATI VCR2 |  | X | fourcc: VCR2 |
| 金光光环 |  | X |  |
| Auravision Aura 2 |  | X |  |
| Autodesk Animator Flic 视频 |  | X |  |
| 欧特克 RLE |  | X | fourcc: AASC |
| AV1 | 乙 | 乙 | 通过外部库 libaom、libdav1d、librav1e 和 libsvtav1 提供支持 |
| Avid 1:1 10 位 RGB 打包机 | X | X | fourcc: AVrp |
| AVS（音频视频标准）视频 |  | X | Creature Shock 游戏使用的视频编码。 |
| AVS2-P2/IEEE1857.4 | 乙 | 乙 | 通过外部库 libxavs2 和 libdavs2 支持 |
| AVS3-P2/IEEE1857.10 |  | 乙 | 通过外部库 libuavs3d 支持 |
| AYUV | X | X | Microsoft 未压缩打包 4:4:4:4 |
| 光束软件 VB |  | X |  |
| 贝塞斯达 VID 视频 |  | X | 在 Bethesda Softworks 的一些游戏中使用。 |
| 宾克视频 |  | X |  |
| BitJazz 纯视频 |  | X |  |
| 位图兄弟 JV 视频 |  | X |  |
| y41p Brooktree 未压缩 4:1:1 12 位 | X | X |  |
| Brooktree ProSumer 视频 |  | X | fourcc: BT20 |
| 蛮力与无知 |  | X | 用于游戏 Flash Traffic：天使之城。 |
| C93视频 |  | X | Cyberia 游戏中使用的编解码器。 |
| 摄像头工作室 |  | X | fourcc: CSCD |
| CD+G |  | X | CD+G 卡拉 OK 盘的视频编解码器 |
| CDXL |  | X | Amiga CD 视频编解码器 |
| 中文AVS视频 | 乙 | X | AVS1-P2，JiZhun profile，通过外部库libxavs编码 |
| Delphine Software International CIN 视频 |  | X | Delphine Software International 游戏中使用的编解码器。 |
| Discworld II BMV 视频 |  | X |  |
| 电影格式高清 | X | X |  |
| 老人星总部 |  | X |  |
| 老人星 HQA |  | X |  |
| 老人星 HQX |  | X |  |
| Canopus 无损编解码器 |  | X |  |
| CD卡通 |  | X | 各种 Broderbund 游戏中使用的编解码器。 |
| 电影胶片 |  | X |  |
| Cirrus Logic AccuPak | X | X | fourcc：CLJR |
| CPiA 视频格式 |  | X |  |
| 创意 YUV (CYUV) |  | X |  |
| DFA |  | X | Chronomaster 游戏中使用的编解码器。 |
| 狄拉克 | 乙 | X | 通过原生 vc2 (Dirac Pro) 编码器支持 |
| 豪华油漆动画 |  | X |  |
| DNxHD | X | X | 又名 SMPTE VC3 |
| 鸭子 TrueMotion 1.0 |  | X | fourcc: 鸭子 |
| 鸭子 TrueMotion 2.0 |  | X | fourcc: TM20 |
| 鸭子 TrueMotion 2.0 RT |  | X | fourcc: TR20 |
| DV（数字视频） | X | X |  |
| Dxtory 捕获格式 |  | X |  |
| 微弱文件/ScummVM DXA |  | X | 最初用于 Feeble Files 游戏的编解码器。 |
| 艺电 CMV 视频 |  | X | 用于 NHL 95 比赛。 |
| 艺电疯牛视频 |  | X |  |
| 美国艺电 TGV 视频 |  | X |  |
| 艺电TGQ视频 |  | X |  |
| 美国艺电TQI视频 |  | X |  |
| 逃生124 |  | X |  |
| 逃生130 |  | X |  |
| FFmpeg 视频编解码器 #1 | X | X | 无损编解码器（fourcc：FFV1） |
| 闪屏视频 v1 | X | X | fourcc: FSV1 |
| 闪屏视频 v2 | X | X |  |
| Flash 视频 (FLV) | X | X | Flash 中使用的 Sorenson H.263 |
| FM 屏幕捕获编解码器 |  | X |  |
| 前向未压缩 |  | X |  |
| 护腕 |  | X |  |
| Go2会议 |  | X | fourcc: G2M2, G2M3 |
| Go2网络研讨会 |  | X | fourcc: G2M4 |
| Gremlin 数字视频 |  | X |  |
| H.261 | X | X |  |
| H.263 / H.263-1996 | X | X |  |
| H.263+ / H.263-1998 / H.263 版本 2 | X | X |  |
| H.264 / AVC / MPEG-4 AVC / MPEG-4 第 10 部分 | 乙 | X | 通过外部库 libx264 和 OpenH264 支持的编码 |
| HEVC | X | X | 通过外部库 libx265 和 libkvazaar 支持编码 |
| HNM 版本 4 |  | X |  |
| 哈夫YUV | X | X |  |
| HuffYUV FFmpeg 变体 | X | X |  |
| IBM Ultimotion |  | X | fourcc: 奥蒂 |
| id 电影视频 |  | X | 用于雷神之锤 II。 |
| 身份证 RoQ 视频 | X | X | 用于雷神之锤 III、绝地武士 2 和其他电脑游戏。 |
| 敌我识别导弹 |  | X | IFF 交错位图 |
| IFF ByteRun1 |  | X | IFF 运行长度编码位图 |
| Infinity IMM4 |  | X |  |
| 英特尔 H.263 |  | X |  |
| 英特尔 Indeo 2 |  | X |  |
| 英特尔 Indeo 3 |  | X |  |
| 英特尔 Indeo 4 |  | X |  |
| 英特尔 Indeo 5 |  | X |  |
| 互动C93 |  | X | 在 Interplay 的游戏 Cyberia 中使用。 |
| 互动 MVE 视频 |  | X | 在 Interplay .MVE 文件中使用。 |
| J2K | X | X |  |
| 卡尔莫顿的视频编解码器 |  | X | 蠕虫游戏中使用的编解码器。 |
| Kega 游戏视频 (KGV1) |  | X | Kega 模拟器屏幕捕获编解码器。 |
| 拉加里斯 |  | X |  |
| LCL（无损编解码器库）MSZH |  | X |  |
| LCL（无损编解码器库）ZLIB | 乙 | 乙 |  |
| 机车 |  | X |  |
| LucasArts SANM/Smush |  | X | 用于 LucasArts 游戏/SMUSH 动画。 |
| 无损 MJPEG | X | X |  |
| 魔术YUV视频 | X | X |  |
| Mandsoft 屏幕捕获编解码器 |  | X |  |
| 微软 ATC 屏幕 |  | X | 也称为 Microsoft Screen 3。 |
| Microsoft 表达式编码器屏幕 |  | X | 也称为 Microsoft Titanium Screen 2。 |
| 微软RLE |  | X |  |
| 微软屏幕 1 |  | X | 也称为 Windows Media Video V7 Screen。 |
| 微软屏幕 2 |  | X | 也称为 Windows Media Video V9 Screen。 |
| 微软视频 1 |  | X |  |
| 模仿 |  | X | 在 MSN Messenger 网络摄像头流中使用。 |
| 米罗VideoXL |  | X | fourcc: VIXL |
| MJPEG（动态 JPEG） | X | X |  |
| Mobotix MxPEG 视频 |  | X |  |
| 运动像素视频 |  | X |  |
| MPEG-1 视频 | X | X |  |
| MPEG-2 视频 | X | X |  |
| MPEG-4 第 2 部分 | X | X | libxvidcore 可以替代地用于编码。 |
| MPEG-4 第 2 部分 Microsoft 变体版本 1 |  | X |  |
| MPEG-4 第 2 部分 Microsoft 变体版本 2 | X | X |  |
| MPEG-4 第 2 部分 Microsoft 变体版本 3 | X | X |  |
| 纽泰克SpeedHQ | X | X |  |
| 任天堂 Gamecube THP 视频 |  | X |  |
| 缺口LC |  | X |  |
| Nuppel视频/RTjpeg |  | X | NuppelVideo 文件中使用的视频编码。 |
| On2 VP3 |  | X | 仍处于实验阶段 |
| On2 VP4 |  | X | fourcc: VP40 |
| On2 VP5 |  | X | fourcc: VP50 |
| On2 VP6 |  | X | fourcc: VP60,VP61,VP62 |
| On2 VP7 |  | X | fourcc: VP70,VP71 |
| VP8 | 乙 | X | fourcc：VP80，通过外部库 libvpx 支持的编码 |
| VP9 | 乙 | X | 通过外部库 libvpx 支持的编码 |
| 品尼高 TARGA CineWave YUV16 |  | X | fourcc: Y216 |
| Q-团队 QPEG |  | X | fourccs: QPEG, Q1.0, Q1.1 |
| QuickTime 8BPS 视频 |  | X |  |
| QuickTime 动画 (RLE) 视频 | X | X | fourcc: 'rle' |
| QuickTime 图形 (SMC) | X | X | fourcc: 'smc' |
| QuickTime 视频 (RPZA) | X | X | fourcc: rpza |
| R10K AJA Kona 10 位 RGB 编解码器 | X | X |  |
| R210 Quicktime 未压缩 RGB 10 位 | X | X |  |
| 原始视频 | X | X |  |
| RealVideo 1.0 | X | X |  |
| RealVideo 2.0 | X | X |  |
| RealVideo 3.0 |  | X | 离理想还很远 |
| RealVideo 4.0 |  | X |  |
| Renderware TXD（纹理字典） |  | X | Renderware 引擎使用的纹理字典。 |
| RL2视频 |  | X | 娱乐软件合作伙伴在某些游戏中使用 |
| 网版 |  | X |  |
| 丝网印刷 |  | X |  |
| 屏幕录像机黄金编解码器 |  | X |  |
| 塞拉利昂 VMD 视频 |  | X | 在 Sierra VMD 文件中使用。 |
| Silicon Graphics 运动视频压缩器 1 (MVC1) |  | X |  |
| Silicon Graphics 运动视频压缩器 2 (MVC2) |  | X |  |
| Silicon Graphics RLE 8 位视频 |  | X |  |
| Smacker 视频 |  | X | Smacker 中使用的视频编码。 |
| SMPTE VC-1 格式 |  | X |  |
| 雪 | X | X | 实验小波编解码器（fourcc：SNOW） |
| 索尼 PlayStation MDEC（动态解码器） |  | X |  |
| Sorenson 矢量量化器 1 | X | X | fourcc: SVQ1 |
| 索伦森矢量量化器 3 |  | X | fourcc: SVQ3 |
| 凌阳 JPEG (SP5X) |  | X | fourcc: SP5X |
| TechSmith 屏幕捕获编解码器 |  | X | fourcc：TSCC |
| TechSmith 屏幕捕获编解码器 2 |  | X | fourcc: TSC2 |
| 西奥拉 | 乙 | X | 通过外部库 libtheora 支持的编码 |
| Tiertex 有限公司 SEQ 视频 |  | X | DOS CD-ROM FlashBack 游戏中使用的编解码器。 |
| 视频 | X | X |  |
| v210 QuickTime 未压缩 4:2:2 10 位 | X | X |  |
| v308 QuickTime 未压缩 4:4:4 | X | X |  |
| v408 QuickTime 未压缩 4:4:4:4 | X | X |  |
| v410 QuickTime 未压缩 4:4:4 10 位 | X | X |  |
| VBLE 无损编解码器 |  | X |  |
| VMware 屏幕编解码器/VMware 视频 |  | X | VMware 捕获的视频中使用的编解码器。 |
| Westwood Studios VQA（矢量量化动画）视频 |  | X |  |
| Windows 媒体图像 |  | X |  |
| Windows 媒体视频 7 | X | X |  |
| Windows 媒体视频 8 | X | X |  |
| Windows 媒体视频 9 |  | X | 没有完全工作 |
| 联队指挥官 III / Xan |  | X | 用于 Wing Commander III .MVE 文件。 |
| 联队指挥官 IV / Xan |  | X | 用于 Wing Commander IV。 |
| 温诺夫WNV1 |  | X |  |
| WMV7 | X | X |  |
| 雅马哈SMAF | X | X |  |
| Psygnosis YOP视频 |  | X |  |
| yuv4 | X | X | libquicktime 未压缩打包 4:2:0 |
| ZeroCodec 无损视频 |  | X |  |
| 零立库 | X | X | LCL的一部分，编码器实验 |
| Zip 运动块视频 | X | X | 编码器仅适用于 PAL8。 |

X表示支持该列中的功能（编码/解码）。

E表示通过外部库提供支持。

### 48.4 音频编解码器

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 编码 | 解码 | 评论 |
| 8SVX指数 |  | X |  |
| 8SVX斐波那契 |  | X |  |
| 有声学协会 | 前任 | X | 通过内部编码器和外部库 libfdk-aac 支持编码 |
| 音频+ | 乙 | 九 | 通过外部库 libfdk-aac 支持的编码 |
| AC-3 | 九 | 九 |  |
| 艾尔普·开尔文 |  | X |  |
| ADPCM 4X 电影 |  | X |  |
| ADPCM 雅马哈爱克 |  | X |  |
| ADPCM AmuseGraphics 电影 |  | X |  |
| ADPCM 阿尔戈游戏 | X | X |  |
| ADPCM CDROM XA |  | X |  |
| ADPCM创意科技 |  | X | 16 -> 4、8 -> 4、8 -> 3、8 -> 2 |
| ADPCM电子艺界 |  | X | 用于各种 EA 游戏。 |
| ADPCM 艺电 Maxis CDROM XS |  | X | 用于模拟城市 3000。 |
| ADPCM艺电R1 |  | X |  |
| ADPCM艺电R2 |  | X |  |
| ADPCM艺电R3 |  | X |  |
| ADPCM 艺电 XAS |  | X |  |
| ADPCM G.722 | X | X |  |
| ADPCM G.726 | X | X |  |
| ADPCM IMA Acorn 回放 |  | X |  |
| ADPCM IMA AMV | X | X | 在 AMV 文件中使用 |
| ADPCM IMA 狡猾的发展 |  | X |  |
| ADPCM IMA 艺电 EACS |  | X |  |
| ADPCM IMA 艺电 SEAD |  | X |  |
| ADPCM IMA 丰康 |  | X |  |
| ADPCM IMA 高压软件 ALP | X | X |  |
| ADPCM IMA Mobiclip MOFLEX |  | X |  |
| ADPCM IMA QuickTime | X | X |  |
| ADPCM IMA 西蒙与舒斯特互动 | X | X |  |
| ADPCM IMA 育碧 APM | X | X |  |
| ADPCM IMA Loki SDL MJPEG |  | X |  |
| ADPCM IMA WAV | X | X |  |
| ADPCM IMA 韦斯特伍德 |  | X |  |
| ADPCM ISS IMA |  | X | 用于 FunCom 游戏。 |
| ADPCM IMA 对话 |  | X |  |
| ADPCM IMA 鸭子 DK3 |  | X | 用于某些 Sega Saturn 控制台游戏。 |
| ADPCM IMA 鸭子 DK4 |  | X | 用于某些 Sega Saturn 控制台游戏。 |
| ADPCM IMA 激进派 |  | X |  |
| 微软 | X | X |  |
| ADPCM MS IMA | X | X |  |
| ADPCM 任天堂 Gamecube AFC |  | X |  |
| ADPCM 任天堂 Gamecube DTK |  | X |  |
| ADPCM 任天堂 THP |  | X |  |
| ADPCM游戏机 |  | X |  |
| ADPCM QT IMA | X | X |  |
| ADPCM 世嘉 CRI ADX | X | X | 用于世嘉 Dreamcast 游戏。 |
| ADPCM 冲击波闪光 | X | X |  |
| ADPCM Sound Blaster Pro 2 位 |  | X |  |
| ADPCM Sound Blaster Pro 2.6 位 |  | X |  |
| ADPCM Sound Blaster Pro 4 位 |  | X |  |
| ADPCM维玛 |  | X | 用于 LucasArts SMUSH 动画。 |
| ADPCM科乐美XMD |  | X |  |
| ADPCM 韦斯特伍德工作室 IMA | X | X | 用于命令与征服等 Westwood Studios 游戏。 |
| ADPCM 雅马哈 | X | X |  |
| ADPCM 僵尸 |  | X |  |
| AMR-NB | 乙 | X | 通过外部库 libopencore-amrnb 支持的编码 |
| AMR-WB | 乙 | X | 通过外部库 libvo-amrwbenc 支持的编码 |
| 惊人的工作室 PAF 音频 |  | X |  |
| 苹果无损音频 | X | X | QuickTime fourcc 'alac' |
| aptX | X | X | 用于蓝牙A2DP |
| 高清 | X | X | 用于蓝牙A2DP |
| ATRAC1 |  | X |  |
| ATRAC3 |  | X |  |
| ATRAC3+ |  | X |  |
| ATRAC9 |  | X |  |
| 宾克音频 |  | X | 在许多游戏的 Bink 和 Smacker 文件中使用。 |
| 邦克音响 |  | X |  |
| 凯尔特 |  | 乙 | 通过外部库 libcelt 支持解码 |
| 编解码器2 | 乙 | 乙 | 通过外部库 libcodec2 支持编码/解码 |
| CRI HCA |  | X |  |
| Delphine Software International CIN 音频 |  | X | Delphine Software International 游戏中使用的编解码器。 |
| 双频PWM | X | X |  |
| 数字语音标准 - 标准播放模式 (DSS SP) |  | X |  |
| Discworld II BMV 音频 |  | X |  |
| 厨师 |  | X | 支持除 5.1 以外的所有版本。 |
| DCA（DTS 相干声学） | X | X | 支持的扩展：XCh、XXCH、X96、XBR、XLL、LBR（部分） |
| 杜比E |  | X |  |
| DPCM Cuberoot-Delta-精确 |  | X | 在少数游戏中使用。 |
| DPCM小精灵 |  | X |  |
| DPCM 编号 RoQ | X | X | 用于雷神之锤 III、绝地武士 2 和其他电脑游戏。 |
| DPCM 大理石瓦迪 |  | X |  |
| DPCM相互作用 |  | X | 用于各种 Interplay 电脑游戏。 |
| DPCM Squareroot-Delta-精确 |  | X | 用于各种游戏。 |
| DPCM Sierra 在线 |  | X | 用于 Sierra Online 游戏音频文件。 |
| DPCM溶胶 |  | X |  |
| DPCM Xan |  | X | 用于 Origin 的 Wing Commander IV AVI 文件。 |
| DPCM 希拉姆 DERF |  | X |  |
| DSD（Direct Stream Digital），最低有效位优先 |  | X |  |
| DSD（Direct Stream Digital），最高有效位优先 |  | X |  |
| DSD（直接流数字），最低有效位优先，平面 |  | X |  |
| DSD（Direct Stream Digital），最高有效位优先，平面 |  | X |  |
| DSP 集团 TrueSpeech |  | X |  |
| DST（直接流传输） |  | X |  |
| 数码影音 |  | X |  |
| 增强型 AC-3 | X | X |  |
| EVRC（增强型可变速率编解码器） |  | X |  |
| FLAC（免费无损音频编解码器） | X | 九 |  |
| FTR语音 |  | X |  |
| G.723.1 | X | X |  |
| G.729 |  | X |  |
| 全球移动通信系统 | 乙 | X | 通过外部库 libgsm 支持的编码 |
| GSM 微软变体 | 乙 | X | 通过外部库 libgsm 支持的编码 |
| IAC（Indeo 音频编码器） |  | X |  |
| iLBC（互联网低比特率编解码器） | 乙 | 前任 | 通过外部库 libilbc 支持编码和解码 |
| IMC（英特尔音乐编码器） |  | X |  |
| 相互作用的ACM |  | X |  |
| MACE（Macintosh 音频压缩/扩展）3:1 |  | X |  |
| MACE（Macintosh 音频压缩/扩展）6:1 |  | X |  |
| 玛丽安的 A-pac 音频 |  | X |  |
| MI-SC4（Micronas SC-4 音频） |  | X |  |
| MLP（子午线无损包装） | X | X | 用于 DVD 音频光盘。 |
| 猴子的音频 |  | X |  |
| MP1（MPEG 音频层 1） |  | 九 |  |
| MP2（MPEG 音频第 2 层） | 九 | 九 | 也通过外部库 TwoLAME 支持编码 |
| MP3（MPEG 音频第 3 层） | 乙 | 九 | 通过外部库支持的编码 LAME、ADU MP3 和 MP3onMP4 也支持 |
| MPEG-4 音频无损编码 (ALS) |  | X |  |
| MobiClip 快速音频 |  | X |  |
| Musepack SV7 |  | X |  |
| Musepack SV8 |  | X |  |
| Nellymoser 麻生 | X | X |  |
| On2 AVC（视频编解码器的音频） |  | X |  |
| 作品 | 乙 | X | 通过外部库 libopus 支持的编码 |
| PCM A律 | X | X |  |
| PCM μ律 | X | X |  |
| PCM 阿基米德 VIDC | X | X |  |
| PCM 签名 8 位平面 | X | X |  |
| PCM 签名 16 位大端平面 | X | X |  |
| PCM 签名的 16 位小端平面 | X | X |  |
| PCM 签名的 24 位小端平面 | X | X |  |
| PCM 签名的 32 位小端平面 | X | X |  |
| PCM 32 位浮点大端 | X | X |  |
| PCM 32 位浮点小端 | X | X |  |
| PCM 64 位浮点大端 | X | X |  |
| PCM 64 位浮点小端 | X | X |  |
| PCM D-Cinema 音频签名 24 位 | X | X |  |
| PCM 签名 8 位 | X | X |  |
| PCM 签名 16 位大端 | X | X |  |
| PCM 签名的 16 位小端 | X | X |  |
| PCM 签名 24 位大端 | X | X |  |
| PCM 签名的 24 位小端 | X | X |  |
| PCM 签名的 32 位大端 | X | X |  |
| PCM 签名的 32 位小端 | X | X |  |
| PCM 在 MPEG-TS 中签署了 16/20/24 位 big-endian |  | X |  |
| PCM 无符号 8 位 | X | X |  |
| PCM 无符号 16 位大端 | X | X |  |
| PCM 无符号 16 位小端 | X | X |  |
| PCM 无符号 24 位大端 | X | X |  |
| PCM 无符号 24 位小端 | X | X |  |
| PCM 无符号 32 位大端 | X | X |  |
| PCM 无符号 32 位小端 | X | X |  |
| PCM SGA |  | X |  |
| QCELP /纯音 |  | X |  |
| QDesign 音乐编解码器 1 |  | X |  |
| QDesign 音乐编解码器 2 |  | X | 仍然存在一些扭曲。 |
| RealAudio 1.0 (14.4K) | X | X | 真正的 14400 位/秒编解码器 |
| RealAudio 2.0 (28.8K) |  | X | 真正的 28800 位/秒编解码器 |
| RealAudio 3.0 (dnet) | 九 | X | 真正的低比特率 AC-3 编解码器 |
| 真实音频无损 |  | X |  |
| 真实音频 SIPR / ACELP.NET |  | X |  |
| RK 音频 (RKA) |  | X |  |
| SBC（低复杂度子带编解码器） | X | X | 用于蓝牙A2DP |
| 缩短 |  | X |  |
| Sierra VMD 音频 |  | X | 在 Sierra VMD 文件中使用。 |
| Smacker音频 |  | X |  |
| SMPTE 302M AES3 音频 | X | X |  |
| 索尼克 | X | X | 实验编解码器 |
| 声波无损 | X | X | 实验编解码器 |
| 斯佩克斯 | 乙 | 前任 | 通过外部库 libspeex 支持 |
| TAK（汤姆的无损音频压缩器） |  | X |  |
| 真实音频 (TTA) | X | X |  |
| 真高清 | X | X | 用于 HD-DVD 和蓝光光盘。 |
| TwinVQ（VQF风味） |  | X |  |
| 维玛 |  | X | 用于 LucasArts SMUSH 动画。 |
| ViewQuest VQC |  | X |  |
| Vorbis | 乙 | X | 存在一个原生但非常原始的编码器。 |
| Voxware 元声 |  | X |  |
| 波形存档器 |  | X |  |
| 音频包 | X | X |  |
| 韦斯特伍德音响 (SND1) |  | X |  |
| Windows 媒体音频 1 | X | X |  |
| Windows 媒体音频 2 | X | X |  |
| Windows Media 音频无损 |  | X |  |
| Windows 媒体音频专业版 |  | X |  |
| Windows Media 音频语音 |  | X |  |
| Xbox 媒体音频 1 |  | X |  |
| Xbox 媒体音频 2 |  | X |  |

X表示支持该列中的功能（编码/解码）。

E表示通过外部库提供支持。

我的意思是也可以使用仅整数版本（确保在没有硬件浮点支持的系统上的高性能）。

### 48.5 字幕格式

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 | 沐星 | 去木星 | 编码 | 解码 |
| 3GPP 定时文本 |  |  | X | X |
| AQ标题 |  | X |  | X |
| 数字电视广播 | X | X | X | X |
| DVB图文电视 |  | X |  | 乙 |
| DVD | X | X | X | X |
| JACOsub | X | X |  | X |
| 微型DVD | X | X |  | X |
| MPL2 |  | X |  | X |
| MPsub (MPlayer) |  | X |  | X |
| PGS |  |  |  | X |
| PJS（凤凰城） |  | X |  | X |
| 真实文本 |  | X |  | X |
| 萨米 |  | X |  | X |
| 云杉格式 (STL) |  | X |  | X |
| SSA/屁股 | X | X | X | X |
| 订阅 (SRT) | X | X | X | X |
| 子查看器 v1 |  | X |  | X |
| 子查看器 |  | X |  | X |
| TED 演讲字幕 |  | X |  | X |
| TTML | X |  | X |  |
| VobSub (IDX+SUB) |  | X |  | X |
| 虚拟播放器 |  | X |  | X |
| WebVTT | X | X | X | X |
| XSUB |  |  | X | X |

X表示支持该功能。

E表示通过外部库提供支持。

### 48.6 网络协议

|  |  |
| --- | --- |
| 姓名 | 支持 |
| AMQP | 乙 |
| 文件 | X |
| 文件传输协议 | X |
| 地鼠 | X |
| 地鼠 | X |
| 高水平LS | X |
| HTTP | X |
| HTTPS | X |
| 冰铸 | X |
| MMSH | X |
| MMST | X |
| 管道 | X |
| Pro-MPEG前向纠错 | X |
| 实时播放协议 | X |
| RTMPE | X |
| RTMPS | X |
| 实时传输协议 | X |
| RTMPTE | X |
| RTMPTS | X |
| 实时传输协议 | X |
| 桑巴 | 乙 |
| SCTP | X |
| SFTP协议 | 乙 |
| TCP | X |
| TLS | X |
| UDP协议 | X |
| ZMQ | 乙 |

X表示支持该协议。

E表示通过外部库提供支持。

### 48.7 输入/输出设备

|  |  |  |
| --- | --- | --- |
| 姓名 | 输入 | 输出 |
| 阿尔萨 | X | X |
| BKTR | X |  |
| 卡卡 |  | X |
| DV1394 | X |  |
| Lavfi虚拟设备 | X |  |
| Linux 帧缓冲区 | X | X |
| 杰克 | X |  |
| LIBCDIO | X |  |
| LIBDC1394 | X |  |
| OpenAL | X |  |
| OpenGL |  | X |
| 开放源码软件 | X | X |
| 脉冲音频 | X | X |
| SDL |  | X |
| 视频4Linux2 | X | X |
| VfW 捕获 | X |  |
| X11抓取 | X |  |
| Win32抓取 | X |  |

X表示支持输入/输出。

### 48.8 时间码

|  |  |  |
| --- | --- | --- |
| 编解码器/格式 | 读 | 写 |
| 视频文件 | X | X |
| 数码相机 | X | X |
| 广信丰 | X | X |
| MOV | X | X |
| MPEG1/2 | X | X |
| MXF | X | X |

## 49 另见

[ffmpeg](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg.html) [ffplay](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffplay.html) ， [ffprobe](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffprobe.html) ， [ffmpeg-utils](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-utils.html) ， [ffmpeg-scaler](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-scaler.html) ， [ffmpeg-resampler](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-resampler.html) ， [ffmpeg-codecs](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-codecs.html) ， [ffmpeg-bitstream-filters](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-bitstream-filters.html) ， [ffmpeg-formats](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-formats.html) ， [ffmpeg-devices](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-devices.html) ， [ffmpeg-protocols](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-protocols.html) ， [ffmpeg-filters](file:///D:\gitHub\ffmpeg-6.0_ffmpeg-all.html\ffmpeg-filters.html)

## 50位作者

FFmpeg 开发人员。

在 FFmpeg 源目录中键入命令git log ，或浏览在线存储库[http://来源.ffmpeg.org](http://source.ffmpeg.org) 。

特定组件的维护者列在源代码树的文件MAINTAINERS中。