.NET Conf China
2022

扫码下载代码示例和PPT

群名称:C#/.NET计算机视觉技术交流
群　号:579060605

长沙.NET技术社区
CHANGSHA DOTNET COMMUNITY

.NET玩转音视频操作 FFmpeg

周杰 (来自长沙)
Handshakes by DC Frontiers

https://github.com/sdcb/Sdcb.FFmpeg

.NET使用FFmpeg两类方法和相关库

命令行

- FFmpeg.NET
- MediaToolkit
- Xabe.Ffmpeg
- 自己写

C API平台调用

- FFmpeg.AutoGen
- EmguFFmpeg
- **Sdcb.FFmpeg**

命令行

优点：容易学习、入门方便、不与GPL开源协议冲突

基于进程互操作，依赖于标准流重定向管理状态

输入和输出依赖于文件，很难精细控制

C API平台调用

缺点：C API代码比较复杂

输入和输出可基于内存，可精细控制每一帧

业界普遍使用FFmpeg.AutoGen，在C#的基础上糅合C/指针，写起来比C API更复杂

Sdcb.FFmpeg

保留所有直接调用C API的能力、保留跨平台的能力

删掉并重写了依赖ClangMacroParser，比原版支持更多的宏解析

动态库加载方式从手动LoadLibrary改为了自动的[DllImport]，这在.NET Core中可以自动从NuGet包中加载dll，这更符合.NET社区共识

除了底层封装，还提供了中层（类）封装和高层（帮助类）封装

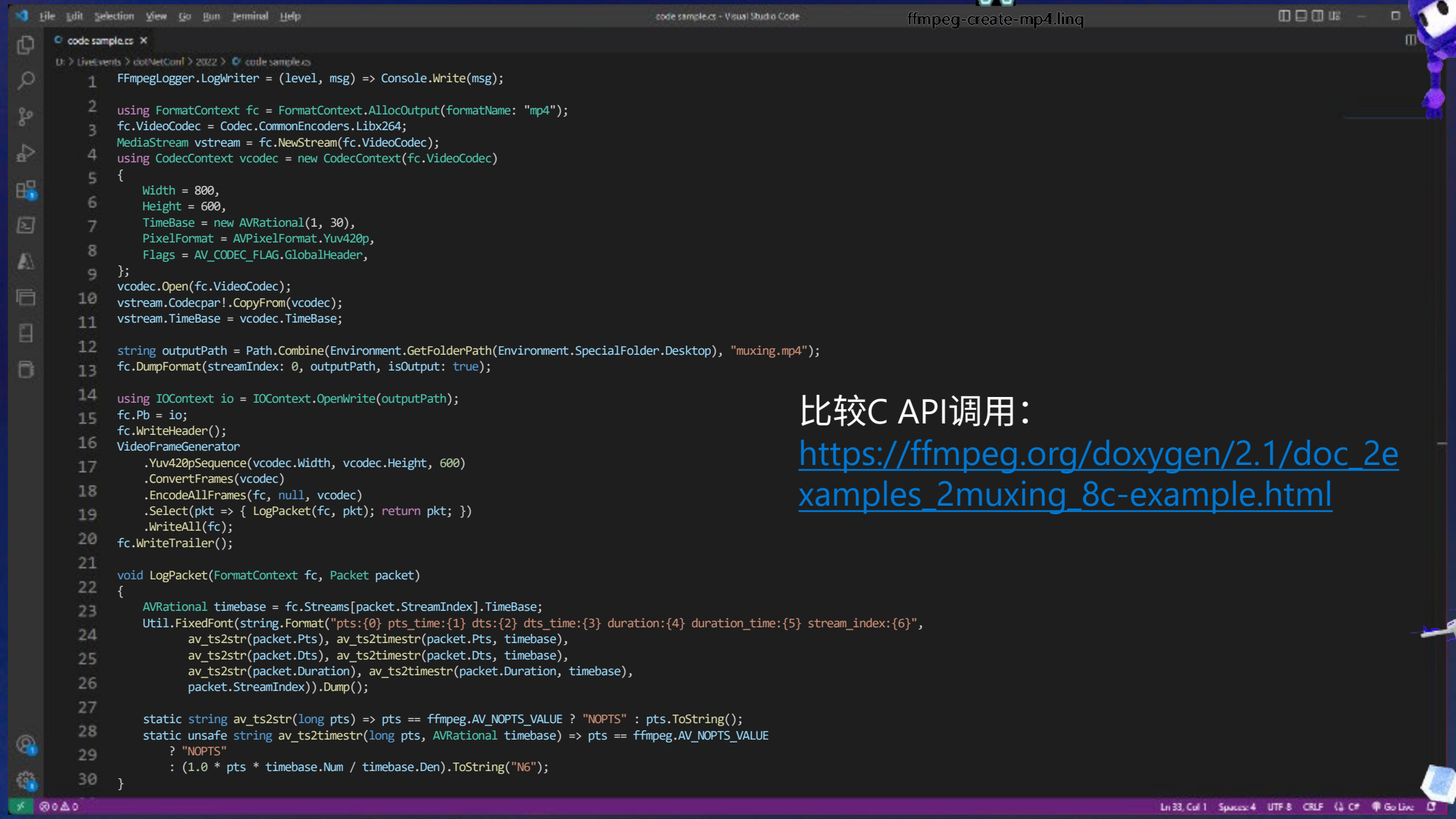
简化了名字，如`AVCodecID.AV_CODEC_ID_H264 -> AVCodecID.H264`

为许多C宏改造成了C#枚举，如`ffmpeg.AV_DICT_MATCH_CASE -> AV_DICT_READ.MatchCase`

5个基于Sdcb.FFmpeg的代码示例

扫码下载代码示例和PPT

示例1 纯代码生成视频




```
FFmpegLogger.LogWriter = (level, msg) => Console.Write(msg);

using FormatContext fc = FormatContext.AllocOutput(formatName: "mp4");
fc.VideoCodec = Codec.CommonEncoders.Libx264;
MediaStream vstream = fc.NewStream(fc.VideoCodec);
using CodecContext vcodec = new CodecContext(fc.VideoCodec)
{
    Width = 800,
    Height = 600,
    TimeBase = new AVRational(1, 30),
    PixelFormat = AVPixelFormat.Yuv420p,
    Flags = AV_CODEC_FLAG.GlobalHeader,
};
vcodec.Open(fc.VideoCodec);
vstream.Codecpar!.CopyFrom(vcodec);
vstream.TimeBase = vcodec.TimeBase;

string outputPath = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Desktop), "muxing.mp4");
fc.DumpFormat(streamIndex: 0, outputPath, isOutput: true);

using IOContext io = IOContext.OpenWrite(outputPath);
fc.Pb = io;
fc.WriteHeader();
VideoFrameGenerator
    .Yuv420pSequence(vcodec.Width, vcodec.Height, 600)
    .ConvertFrames(vcodec)
    .EncodeAllFrames(fc, null, vcodec)
    .Select(pkt => { LogPacket(fc, pkt); return pkt; })
    .WriteAll(fc);
fc.WriteTrailer();

void LogPacket(FormatContext fc, Packet packet)
{
    AVRational timebase = fc.Streams[packet.StreamIndex].TimeBase;
    Util.FixedFont(string.Format("pts:{0} pts_time:{1} dts:{2} dts_time:{3} duration:{4} duration_time:{5} stream_index:{6}",
           av_ts2str(packet.Pts), av_ts2timestr(packet.Pts, timebase),
           av_ts2str(packet.Dts), av_ts2timestr(packet.Dts, timebase),
           av_ts2str(packet.Duration), av_ts2timestr(packet.Duration, timebase),
           packet.StreamIndex)).Dump();

    static string av_ts2str(long pts) => pts == ffmpeg.AV_NOPTS_VALUE ? "NOPTS" : pts.ToString();
    static unsafe string av_ts2timestr(long pts, AVRational timebase) => pts == ffmpeg.AV_NOPTS_VALUE
        ? "NOPTS"
        : (1.0 * pts * timebase.Num / timebase.Den).ToString("N6");
}
```

比较C API调用：
https://ffmpeg.org/doxygen/2.1/doc_2examples_2muxing_8c-example.html

示例1 纯代码生成视频·重点代码解析

```
fc.WriteHeader();
VideoFrameGenerator.Yuv420pSequence(vcodec.Width, vcodec.Height, 600)
    .ConvertFrames(vcodec)
    .EncodeAllFrames(fc, null, vcodec)
    .WriteAll(fc);
fc.WriteTrailer();
```

扫码下载代码示例和PPT

示例1 纯代码生成视频·技术要点

1. Yuv420pSequence
2. IEnumerable<Frame>
3. IEnumerable<Packet>

```
3 个引用 | 0/1 通过
public static unsafe void FillYuv420p(Frame frame, int i)
{
    int_array8 linesize = frame.Linesize;
    int linesize0 = linesize._[0];
    int linesize1 = linesize._[1];
    int linesize2 = linesize._[2];

    byte* data0 = (byte*)frame.Data._0;
    byte* data1 = (byte*)frame.Data._1;
    byte* data2 = (byte*)frame.Data._2;

    /* prepare a dummy image */
    /* Y */
    for (int y = 0; y < frame.Height; y++)
    {
        for (int x = 0; x < frame.Width; x++)
        {
            data0[y * linesize0 + x] = (byte)(x + y + i * 3);
        }
    }
    /* Cb and Cr */
    for (int y = 0; y < frame.Height / 2; y++)
    {
        for (int x = 0; x < frame.Width / 2; x++)
        {
            data1[y * linesize1 + x] = (byte)(128 + y + i * 2);
            data2[y * linesize2 + x] = (byte)(64 + x + i * 5);
        }
    }

    frame.Pts = i;
}
```

```
0 个引用
public static class PacketsExtensions
{
    /// <summary> Calling every frame with following apis: av_packet_clone av_packet ...
    1 个引用 | 0/1 通过
    public static IEnumerable<Packet> CloneMakeWritable(this IEnumerable<Packet> packets, bool unref = true)...

    /// <summary> packets -> frames
    2 个引用 | 0/2 通过
    public static IEnumerable<Frame> DecodePackets(this IEnumerable<Packet> packets, CodecContext c)...

    /// <summary> packets -> frames
    0 个引用
    public static IEnumerable<Frame> DecodeAllPackets(this IEnumerable<Packet> packets, FormatContext fc,
        CodecContext? audioCodec = null,
        CodecContext? videoCodec = null)...

    3 个引用 | 0/3 通过
    public static void WriteAll(this IEnumerable<Packet> packets, FormatContext fc, bool unref = true)...

    0 个引用
    public unsafe static void SetData(this Packet packet, IntPtr data, int size)...
}
```

FramesExtensions.cs

Sdcb.FFmpeg (net48) — Sdcb.FFmpeg.Toolboxs.Extensions.FramesExtensions

```
267     /// <returns>Caller must call <see cref="Frame.Unref"/> to the result when not used.</returns>
        0 个引用
268     public static IEnumerable<Frame> ApplyAudioFilters(this IEnumerable<Frame> srcFrames, AudioFilterContext ctx, bool unref = true)...
291
292     /// <returns>Caller must call <see cref="Frame.Unref"/> to the result when not used.</returns>
        0 个引用
293     public static IEnumerable<Frame> ApplyFilters(this IEnumerable<Frame> srcFrames,
294         AudioFilterContext? audioCtx = null,
295         VideoFilterContext? videoCtx = null,
296         bool unref = true)...
336
337     /// <returns>Caller must call <see cref="Frame.Unref"/> to the result when not used.</returns>
        0 个引用
338     public static IEnumerable<Frame> AudioFifo(this IEnumerable<Frame> frames, CodecContext encoder, bool unref = true)...
380
381     /// <summary> frames -> packets
        3 个引用 | 0/1 通过
385     public static IEnumerable<Packet> EncodeFrames(this IEnumerable<Frame> frames, CodecContext c, bool makeSequential = false)...
406
407     /// <summary> frames -> packets
        3 个引用 | 0/3 通过
411     public static IEnumerable<Packet> EncodeAllFrames(this IEnumerable<Frame> frames, FormatContext fc,
412         CodecContext? audioEncoder = null,
413         CodecContext? videoEncoder = null,
414         bool allowSkipFrame = true)...
501
        0 个引用
502     public static IEnumerable<Frame> ConvertVideoFrames(this IEnumerable<Frame> sourceFrames, Func<(int width, int height)> sizeAccessor, AVP
545 }
```

示例2 压制视频为微信不二压的码率

要点：

- 视频编码：H264
- 视频码率：600kbps以下
- 视频分辨率：未限制，但推荐长边960
- 音频编码：AAC
- 音频码率：48kbps
- <代码见附件>

扫码下载代码示例和PPT

示例3 gif表情包生成器

要点：

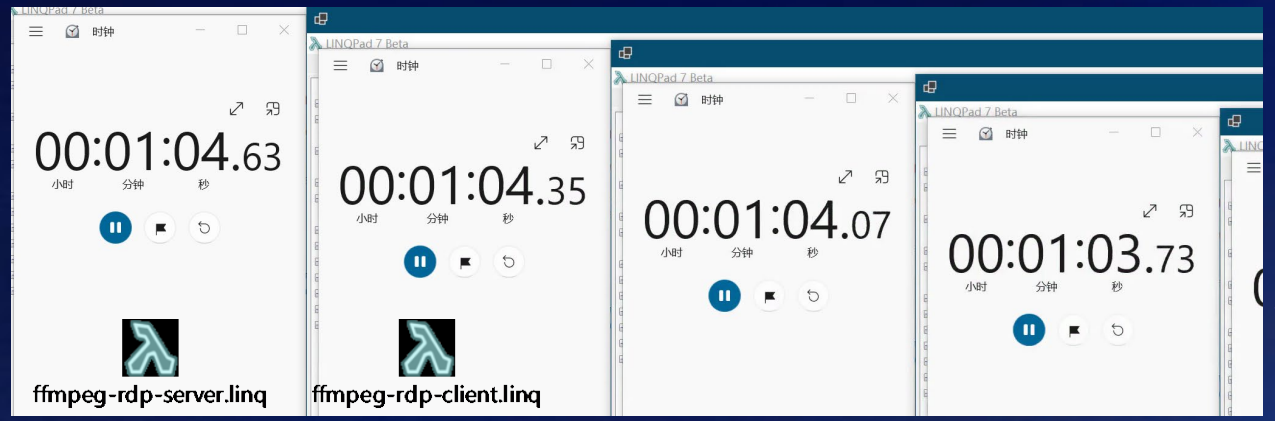
- 视频解码
- 将每一帧转换为BGRA像素格式
- 使用Direct2D读取并绘制字幕
- 将每一帧输入视频过滤器，转换为PAL8格式
- 将PAL8编码像素格式的帧编码为gif
- <源码和demo见下面链接>

扫码下载代码示例和PPT

https://ffmpeg-sorry-demo.starworks.cc:88/
https://github.com/sdcb/ffmpeg-wjz-sorry-generator
https://github.com/sdcb/sdcb.ffmpeg

示例4 投屏

- 创建网络流，写入视频大小、像素格式、编码等元数据信息
- 截屏，然后将屏幕图片转换为YUV420P等常用编码像素格式
- 使用libx264编码时，需要指定[“preset”]=“zerolatency”，否则延时会太高
- 将每个编码的视频帧以byte[]的形式读出，写入到网络流
- 读取端读取网络流，读出相关元数据信息，每次读取一个包
- 将包解码，并调用Direct2D完成显示
- <源代码见下面的2个附件>



扫码下载代码示例和PPT

示例5 播放/录制RTSP摄像头监控视频

要点：

- 默认会使用UDP环境，但需要配置电脑防火墙和路由器的uPnP设置
- 如果无法完成nPnP相关配置，可输入[“rtsp_transport”] = “tcp”
- 不解码录制成连续多文件时，需要注意每一个Packet的Pts/Dts
- <Demo见2个附件>

扫码下载代码示例和PPT

Thank you!

周杰 (来自长沙)
Handshakes by DC Frontiers

扫码下载代码示例和PPT

群名称:C#/.NET计算机视觉技术交流
群　号:579060605