

INTRODUCTION TO VIDEO

A horizontal, glowing orange-yellow light streak that spans across the width of the slide, positioned just below the title text.

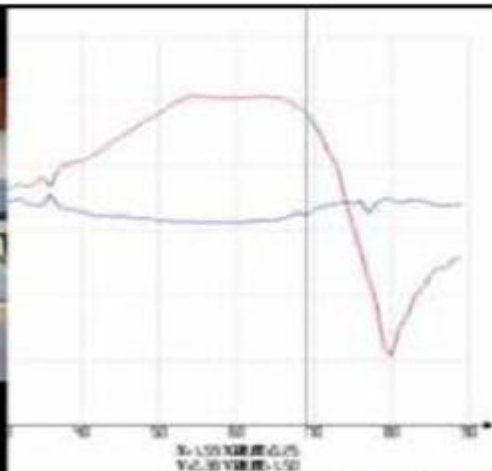
APPLICATIONS

- Track



APPLICATIONS

- Analysis



BASIC CONCEPTS

- Image sequence (序列)
 - FPS (Frames Per Second, > 24fps)
 - Resolution 分辨率
 - 视频显示格式 1080p, 720i, DVD, VCD, HD
- Encode/Decode (编码/解码)
 - 概念 MPEG, H.264
 - 文件格式 WMV, MP4, AVI

分辨率

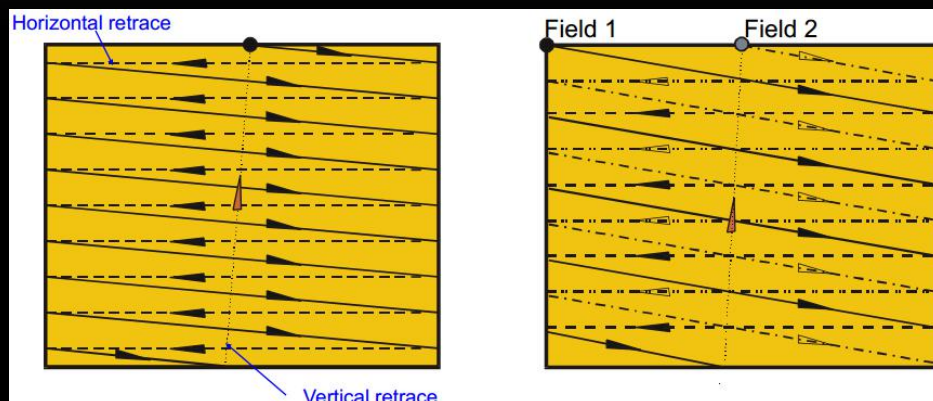
- 视频画面的大小（以像素度量）
 - PAL制式：768/720 x 576
 - NTSC制式：720/704/640 下480
 - 全高清电视：1920 x 1080
 - 4K电视：3840 x 2160

帧 率

- 帧 率frame rate: number of frames per second (fps)
- 最 基 本 要 求: >16 fps
- 目前的主流需求 (游戏, 高清视频) : >30 fps
- 传统电影: 24fps
- PAL电视制式: 25fps (i50)
- NTSC电视制式: 30fps (i60)

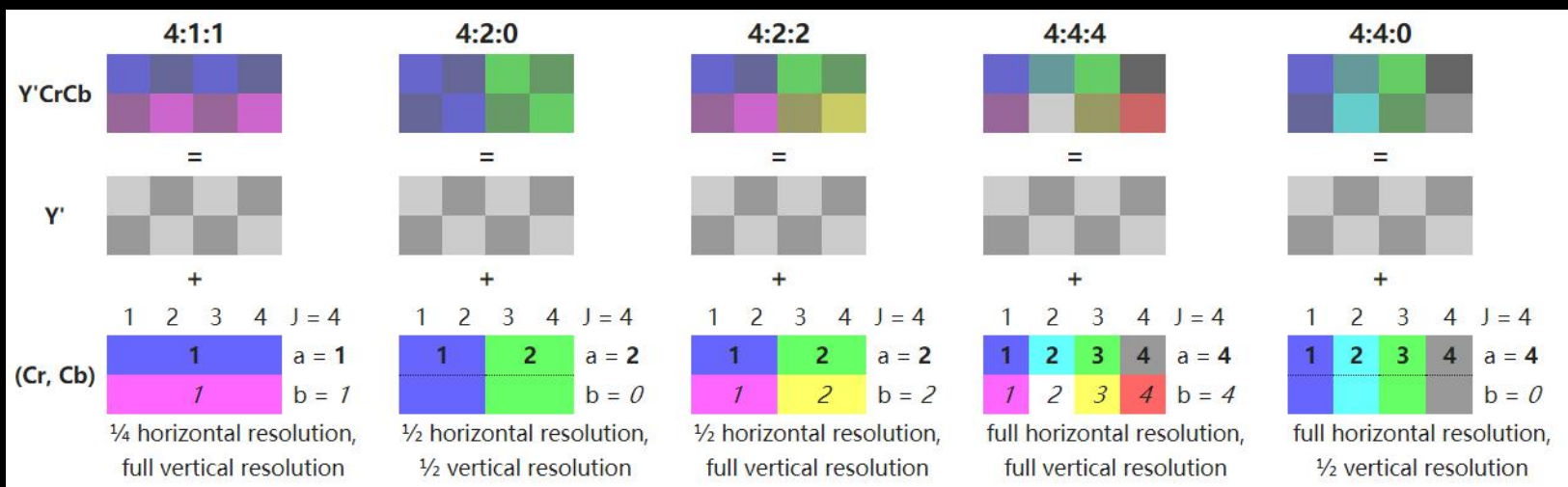
逐行和隔行

- Progressive 和 Interlaced
- why 隔行?
 - 提高图像分辨率（清晰度）
 - 会带来一定的闪烁
 - NTSC、PAL电视制式都是隔行
 - 新的高清电视具有转换的能力
- 视频描述
 - 1080p
 - 720p



色彩模式

- 视频中基本都使用基于灰度的Yxx模式
 - NTSC: YIQ
 - PAL: YUV
 - 数字视频: YCbCr
- 视频中并非每个像素的每个分量都占有一个字节(8位)
 - 一般每个像素的灰度占8位, 其他按比例计算



视频压缩和解压缩：编解码

- 必要性
 - 4:2:0 情况下2小时1080p电影（24fps）所需要的存储
 - $2 \text{ 小时} \times 60 \text{ 分} \times 60 \text{ 秒} \times 24 \text{ 帧} \times 1920 \times 1080 \text{ 像素} \times 1.5 \text{ (字节)} = 5375 \text{ GB}$
- 基本思路
 - 时间上的冗余：相邻帧的强相关性
 - 空间上的冗余：相邻像素的强相关性
 - 感知上的冗余：人眼对信息的敏感性
 - 统计上的冗余：统计的概率
- 有损压缩
 - **5 4 2 -> compression -> decompression ->**
 - **3 2 0, high error**
 - **4.9 4.1 1.9, low error**
 - **5 4 2, lossless**

编解码标准

- 国际电联（ITU）：H.26x
 - 视频传输通信。传统的电视、视频监控等领域
 - H.261、H.262、H.263、H.264、H.265
- 国际标准化组织（ISO）：MPEG
 - 视频存储和结构化表示。计算机、互联网等领域
 - MPEG-1、MPEG-2、MPEG-4、MPEG-H
 - MPEG-4 Part 2也是定位网络视频传输
 - divx和xvid（开源）
 - H.264 = MPEG-4 Part 10 = MPEG-4 AVC
 - H.265 = MPEG-H Part 2 = MPEG-H HEVC
- MicroSoft VC-1（WMV）
- RealVideo（RMVB）

视频文件格式

- MP4、RMVB、MKV、AVI?
 - 视频文件扩展名，视频文件的封装格式（容器：视频、音频、字幕、章节等信息的封装标准）
 - mp4是MPEG-4 Part 14所制定的容器
 - mkv是开放性容器
 - 同一种容器可以使用不同的编码，所以不能简单通过文件格式或容器来区分视频

码 率

- 视频数据（视频色彩量、亮度量、像素量）每秒输出的位数。一般用的单位是kbps
- 编码时候输入条件包含码率要求
 - 固定码率
 - 动态码率
- 评价编码技术：相同码率下，质量更好；相同质量下，码率更低
- 基本思想
 - I、B、P帧
 - GOP组：一个关键帧所在的组的长度

不同的应用需求

- 实时编码 -> 实时解码
 - 视频会议
 - 直播
- 非实时编码 -> 实时解码（对质量要求高）
 - 高清电影

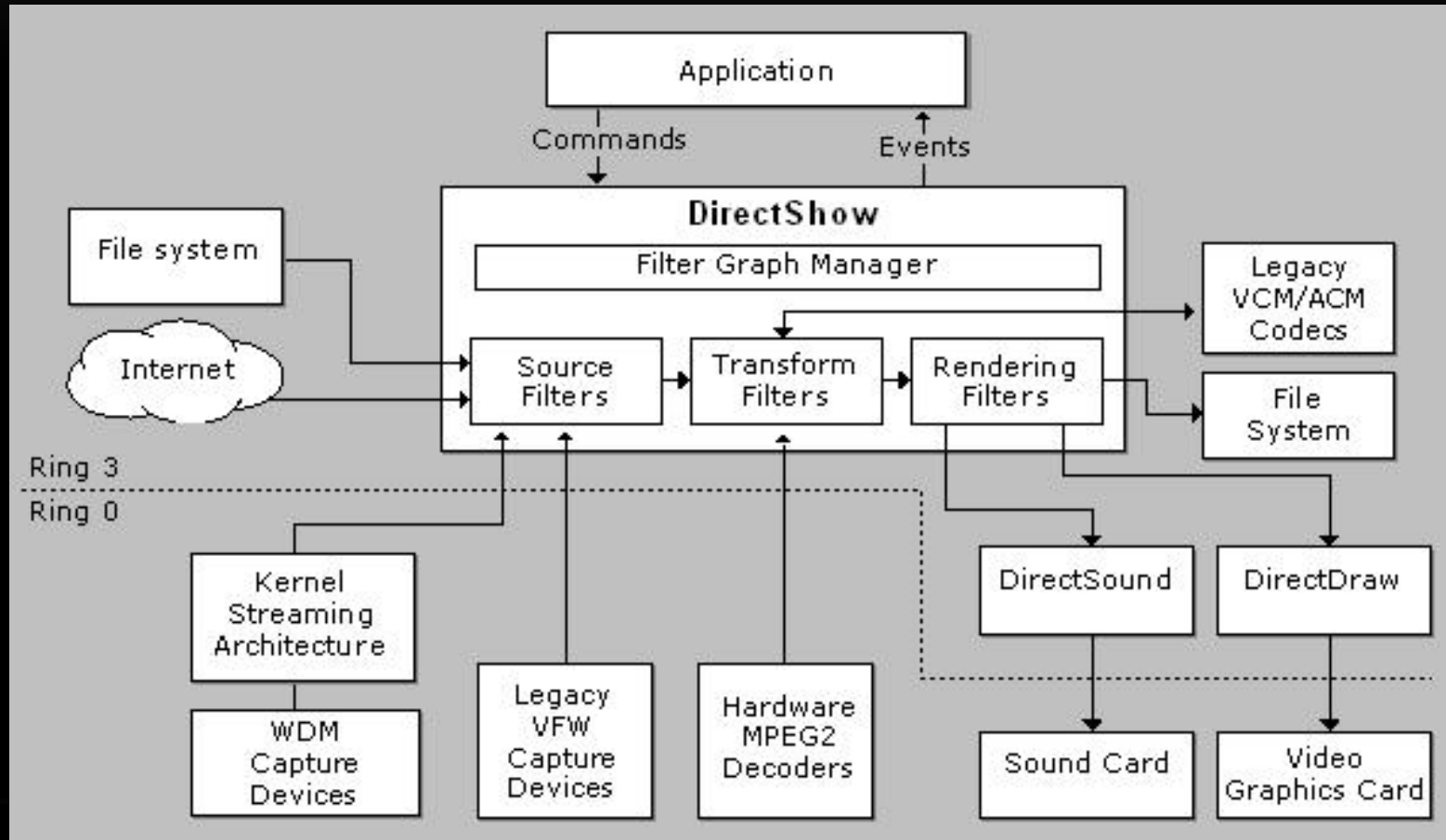
FIELDS

- Theory
- Decoder/Encoder Development
- **DirectShow** / Mplayer / FFMpeg, etc
- **Applications using OpenCV**
- **Computational Video**

DIRECTSHOW

- **Part of DirectX before**
 - **Direct3D, DirectShow, DirectInput, DirectSound, ...**
 - **Removed to Windows Platform SDK in 2005**
 - **COM based**
 - **Interface based**

FRAMEWORKS (框架)



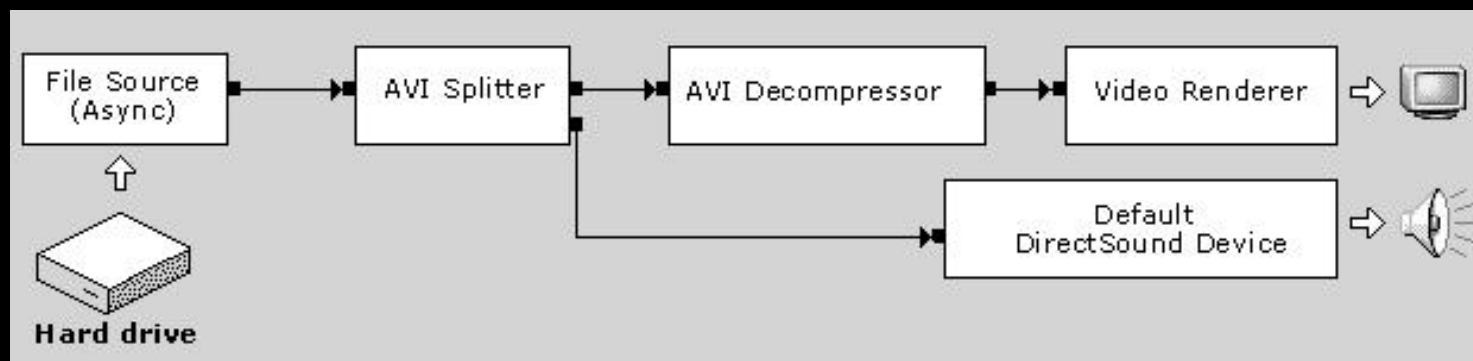
DIRECTSHOW FILTER

- 构建图的基本模块，功能（类型）包括
 - 读取文件
 - 从采集设备获取数据流
 - 编解码流媒体
 - Filter之间的数据转换和传递
 - 输出：显示、文件、网络流
 - Input/Output Pins



FILTER GRAPH

- 一系列Filter通过互相连接构成的具有完整功能的运行图
 - 包含输入、输出和中间Filter



DIRECTSHOW的使用

- 通过应用软件（媒体播放器）
- 自己构建（编程或者使用构建工具）
- 自己编写Filter完成特定功能

DIRECTSHOW

- **GraphEdit samples**
 - <https://www.videohelp.com/software/GraphStudio>
 - <https://github.com/Nevcairiel/LAVFilters>
 - <https://www.visioforge.com/encoding-filters-pack>
- **GraphEdt**
 - 可视化构建运行图的工具，无需编程
 - 可查看系统当前可用的各种Filter
- 显示播放
- 转码
- 存储

C++ CODES

- **// Create the filter graph**
- `hr = CoCreateInstance(CLSID_FilterGraph, NULL, CLSCTX_INPROC_SERVER, IID_IGraphBuilder, (void **) & g_pGraph);`
- **// Create the capture graph builder**
- `hr = CoCreateInstance(CLSID_CaptureGraphBuilder2, NULL, CLSCTX_INPROC_SERVER, IID_ICaptureGraphBuilder2, (void **) & g_pCapture);`
- **// Add the filter**
- `hr = g_pGraph->AddFilter(pSrcFilter, L"Video Capture");`
- **// Auto construct the graph**
- `hr = g_pCapture->RenderStream(&PIN_CATEGORY_PREVIEW, &MEDIATYPE_Video, pSrcFilter, pGrabber, pNullRender);`

OPENCV

- **Video Files**
 - **FFMpeg based**
- **Cameras**
 - **DirectShow based**

INPUT SOURCES

- `VideoCapture video(0);`
- `VideoCapture video(1);`
- `VideoCapture video("hello.avi");`

READ FRAMES

- `cv::Mat frame;`
- `video.read(frame);`

VIDEO应用示例

- OpenCV Tracker
- OpenCV Stabilization