

## 原 最简单的视音频播放示例1：总述

2014年10月19日 00:09:00 阅读数：17763

最简单的视音频播放示例系列文章列表：

[最简单的视音频播放示例1：总述](#)

[最简单的视音频播放示例2：GDI播放YUV, RGB](#)

[最简单的视音频播放示例3：Direct3D播放YUV, RGB（通过Surface）](#)

[最简单的视音频播放示例4：Direct3D播放RGB（通过Texture）](#)

[最简单的视音频播放示例5：OpenGL播放RGB/YUV](#)

[最简单的视音频播放示例6：OpenGL播放YUV420P（通过Texture，使用Shader）](#)

[最简单的视音频播放示例7：SDL2播放RGB/YUV](#)

[最简单的视音频播放示例8：DirectSound播放PCM](#)

[最简单的视音频播放示例9：SDL2播放PCM](#)

## 前言

最近研究了一下Windows平台下的视音频播放的技术。在Windows平台下的视频播放技术主要有以下三种：GDI，Direct3D和OpenGL；音频播放技术主要是DirectSound。这些技术属于比较底层的技术，因此使用起来相对来说还是比较复杂的。我在学习的过程中也发现这一领域一直缺少比较简单直观的示例程序，因此打算做些示例程序，同时写一些相关的文章，方便这一领域的人学习相关的知识。

打算重点记录一下视频播放的技术。之前的研究一直集中于视频的编解码方面的技术，而很少接触到视频播放显示方面的技术。此前的程序一直采用了调用SDL的方式播放显示视频画面。而事实上，SDL本身并不具有播放显示的功能，它只是封装了底层播放显示的代码。为了做到“知其然知其所以然”，需要分析一下视频播放的底层技术。

在Windows平台下视频的播放显示主要可以使用以下两种技术：Direct3D和OpenGL。以上两种显示技术是最好的。除了以上两种之外，还有其他的选择。比如说DirectDraw或者GDI。但是微软已经停止了对DirectDraw的支持。而GDI也并不是主要用于视频显示领域（主要用于软件开发）。有关于他们这几个显示技术之间的比较的文章比较多，在这里不再细说。计划写5篇文章记录三种视频显示技术：GDI，Direct3D，OpenGL。其中Direct3D包含简单和复杂的两种显示方式：使用Surface和使用Texture；OpenGL也包含简单和复杂的两种显示方式：直接画像素和使用Texture。最后再写一篇文章记录一下使用SDL2显示视频的技术，与以上技术的代码做一个对比。需要注意的是，Direct3D，OpenGL这些技术在3D的领域拥有大量的功能，但这些功能并不是这些文章的侧重点。这些文章面向于视频技术人员而非3D游戏开发人员，因而更加侧重于它们在视频显示领域的应用。对于3D方面的功能则基本上会一笔带过。

先说一点我自己的感受。我发现作为一个搞视频技术的人研究Direct3D，OpenGL这些显示技术还是有一些困难的。一方面，尽管网络上有大量的Direct3D，OpenGL方面的资料（感觉比视频编解码领域的资料要多很多），但是这些资料基本上都是面向于3D游戏编程的。因而翻阅了大量的资料，会学到很多和视频显示相关性不大的知识：比如说投影，光线等等。尽管经过这样“坚持不懈”的学习，最后也可以掌握与视频显示相关性比较强的东西（比如说像纹理这方面的知识），但是这样会消耗掉大量的时间与精力。正可谓“时间就是金钱”，这样学习不是很经济。另一方面，尽管偶尔还是可以找到一些OpenGL，Direct3D的显示视频的例子，又比较复杂。这些例子基本上都是一些“大神”的劳动成果。他们不但为了使用方便而对很多函数进行了封装，而且还考虑到很多可能出错的地方而写了很多错误判断的语句。这样做对于成熟的应用来说是一件好事，但是对新手来说却是一件坏事。因为过多的封装会让新手搞不清到底哪些函数是Direct3D，OpenGL的API，哪些函数是“大神”后加的。而增加过多的错误判断等方面的语句之后代码会变得很长，新手就很难分辨出究竟哪些函数才是关键的API。由于以上问题的存在，使得Direct3D，OpenGL这些显示技术的学习门槛比较高，很多人也不愿意投入太多的精力去研究它。

我记录这几篇文章正是为了解决之几方面的问题而写的。这几篇文章在知识叙述方面只记录OpenGL，Direct3D中和视频显示相关的技术，其他方面的技术尽量不写，做到“功夫用在刀刃上”；在示例代码方面，尽量剔除无关紧要的代码，只保留最关键的函数。相信这样会有助于更多的人学习视频显示方面的技术。

这几篇文章中的示例程序，输入都是YUV/RGB的像素数据（注意是没有文件头的那种），输出都是一个显示视频的窗口。由于像素数据体积比较大，因此输入视频的长度很短，只有几秒钟。

## Simplest Media Play

有关使用Direct3D，OpenGL，GDI，DirectSound这些技术播放视音频的代码已经整合到了一个名为“Simplest Media Play”的工程里。该工程已经配置好并且包含了测试素材（YUV，RGB，PCM数据），直接运行即可看到结果。

## 项目主页

SourceForge：<https://sourceforge.net/projects/simplestmediaplay/>

Github：[https://github.com/leixiaohua1020/simplest\\_media\\_play](https://github.com/leixiaohua1020/simplest_media_play)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_media\\_play](http://git.oschina.net/leixiaohua1020/simplest_media_play)

CSDN下载地址：<http://download.csdn.net/detail/leixiaohua1020/8054395>

1.1版更新（2014.12.21）=====

修复了一些小问题

CSDN下载地址：<http://download.csdn.net/detail/leixiaohua1020/8284113>

SourceForge已经更新。

上述工程包含了使用各种API（Direct3D，OpenGL，GDI，DirectSound，SDL2）播放多媒体例子。其中音频输入为PCM采样数据。输出至系统的声卡播放出来。视频输入为YUV/RGB像素数据。输出至显示器上的一个窗口播放出来。

通过本工程的代码初学者可以快速学习使用这几个API播放视频和音频的技术。

一共包括了如下几个子工程：

simplest\_audio\_play\_directsound:

使用DirectSound播放PCM音频采样数据。

simplest\_audio\_play\_sdl2:

使用SDL2播放PCM音频采样数据。

simplest\_video\_play\_direct3d:

使用Direct3D的Surface播放RGB/YUV视频像素数据。

simplest\_video\_play\_direct3d\_texture:使用Direct3D的Texture播放RGB视频像素数据。

simplest\_video\_play\_gdi:

使用GDI播放RGB/YUV视频像素数据。

simplest\_video\_play\_opengl:

使用OpenGL播放RGB/YUV视频像素数据。

simplest\_video\_play\_opengl\_texture:

使用OpenGL的Texture播放YUV视频像素数据。

simplest\_video\_play\_sdl2:

使用SDL2播放RGB/YUV视频像素数据。

工程中除了几个子工程的源代码之外，还包括了以下内容：

## 关键函数调用结构图

工程包含了每个工程的关键函数调用。例如下图是使用Direct3D的Surface播放视频的时候的函数调用结构图。不再一一列举。

□

## 测试素材

test\_bgr24\_320x180.rgb：在硬盘上的存储顺序是B1G1R1, B2G2R2, B3G3R3.....

test\_bgra\_320x180.rgb：在硬盘上的存储顺序是B1G1R1A1, B2G2R2A2, B3G3R3A3.....

test\_rgb24\_320x180.rgb：在硬盘上的存储顺序是R1G1B1, R2G2B2, R3G3B3.....

test\_yuv420p\_320x180.yuv：

在硬盘上的存储顺序是

YYYYYYYYYY...

YYYYYYYYYY...

UUUUU...

VVVVV...

内容截图如下所示：

□

分辨率为320x180，长度为50帧（2秒）。内容为阴天的故宫。

注：可以使用Vooya查看RGB/YUV数据，强大而且方便。地址：<http://www.offminor.de/>

NocturneNo2inEflat\_44.1k\_s16le.pcm：在硬盘上的存储顺序是L1L1,R1R1,L2L2,R2R2,...  
采样率为44.1kHz，采样位数为16bit，立体声。内容为肖邦的《夜曲》（Nocturne in E flat major, Op9 No2）中的起始片段。

注：可以使用Audition这类的音频编辑软件查看PCM数据。

## 结果

工程中的视频显示的结果都是一样的。弹出的窗口都是统一的500x500的大小，显示的内容也是一模一样，这样也有利于比对它们之间的技术。

同理，音频播放的素材也是一模一样。

下面截图显示一下它们的运行结果：

Simplest Audio Play DirectSound

Simplest Audio Play SDL2

Simplest Video Play Direct3D (Surface)

Simplest Video Play Direct3D (Texture)

Simplest Video Play GDI

Simplest Video Play OpenGL

Simplest Video Play OpenGL (Texture)

Simplest Video Play SDL2

版权声明：本文为博主原创文章，未经博主允许不得转载。https://blog.csdn.net/leixiaohua1020/article/details/40246783

文章标签：[Direct3D](#) [OpenGL](#) [GDI](#) [DirectSound](#) [SDL](#)

个人分类：[OpenGL](#) [我的开源项目](#) [GDI](#) [DirectSound](#) [Direct3D](#) [SDL](#)