

# 计算机组织与体系结构实习 Lab 4.1

2019/12/02

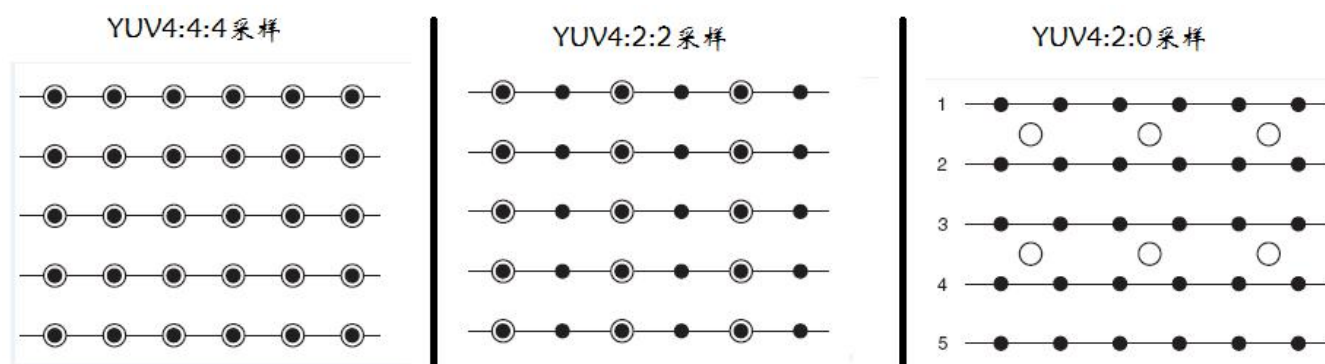
## 基于SIMD扩展指令的图像处理加速

该lab希望通过在x86主流处理器上，使用SIMD指令来加速实现图像的渐变和叠加，加深同学对SIMD指令的基本认识，并为lab 4.2做准备。

### part1、背景知识简介

#### 1. YUV420 图像格式简介

- 定义：每个像素分为三个分量，**Y** 表示明亮度，也就是灰度值；**U**和**V** 表示的则是色度，作用是描述影像色彩及饱和度，用于指定像素的颜色。
- 特点：即使没有U、V分量也可以显示完整的图片，只是没有颜色。在传输时可以占用较小的带宽。
- 采样方式：



- 存储格式：

假设一个分辨率为8X4的YUV图像，Y分量数目 =  $8 * 4$ ；U分量数目 =  $Y/4$ ；V分量数目 =  $Y/4$ 。

#### 1. ARGB8888 图像格式简介

- 定义：每个像素四个分量：ARGB，分别是Alpha（透明度）、Red、Green、Blue三色。
- 存储格式：分别用8位来记录4个值，所以每个像素会占用32位。

#### 1. RGB888 图形格式简介

- 定义：就是RGB24，具有三个分量RGB，每个分量8位。
- 存储格式：就是位图的存放格式，BMP就是在RGB数据基础上增加位图头数据而形成的文件格式。

1. YUV与RGB转换、Alpha混合等操作，同学们可自行上网搜索公式，注意有些公式精度处理不好，会产生锯齿状图形。

### part2、单幅图像的淡入淡出

1. 从网盘目录lab中下载相关内容。

2. 处理步骤如下：

- 从demo目录中读入一幅YUV420格式的图像。demo目录下共存放了两个单帧的yuv文件：dem1.yuv和dem2.yuv，大小都是1920 \* 1080 ( width \* height ) ,截取于parkscene.yuv文件。
- 进行YUV420到ARGB8888转换。其中，Alpha的取值范围：1~255，每隔3进行一次转换，共生成84幅图像；
- 根据A计算alpha混合后的RGB值 ( $A * RGB / 256$ ) ,得到不同亮度的alpha混合；
- 将alpha混合后的图像转换成YUV420格式，存入输出文件。
- 可以使用yuv播放器对生成的图像进行播放。

## part3、两幅图形叠加的渐变处理

---

1. 处理步骤如下：

- 读入demo目录下的二幅YUV420格式的图像；
- 分别进行YUV420到RGB888的转换；
- Alpha的取值范围：1~255，每隔3取值一次，并将两幅图像相加，共生成84幅图像；
- 将叠加后的图像转换成YUV420格式，存入输出文件。
- 可以使用yuv播放器对生成的图像进行播放。

## 实习要求

---

1. 分别采用X86的MMX, SSE2, AVX扩展指令对part2进行加速处理，统计加速后处理时间变化。
2. 分别采用x86的MMX, SSE2, AVX扩展指令对part3进行加速处理，统计加速后处理时间变化。

**注意：**

- 需要加速处理的计算包括：YUV2RGB转换、RGB2YUV转换、alpha混合计算。
- 提供程序模板

## 提交要求

**每人需单独提交：**

1. 实验报告1份。具体内容参照报告模板。
2. part2中优化前后的程序代码。提交内容中还应包含README文档，简要描述程序执行方法。
3. part3中优化前后的程序代码。提交内容中还应包含README文档，简要描述程序执行方法。