# 实验二 直方图均衡化

### 实验目的

1. 理解点操作图像增强方法。
2. 理解直方图均衡化算法的原理，掌握算法的实现。

### 实验任务

读出存储的黑白灰度图像并显示，编程实现将所给seed.yuv图像进行直方图均衡化处理，并给出均衡化前后的对比图。

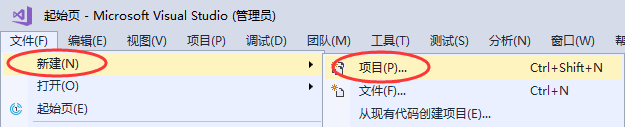
提示：可借助所给YUV播放器实现对视频图像的观看，注意界面参数设置。

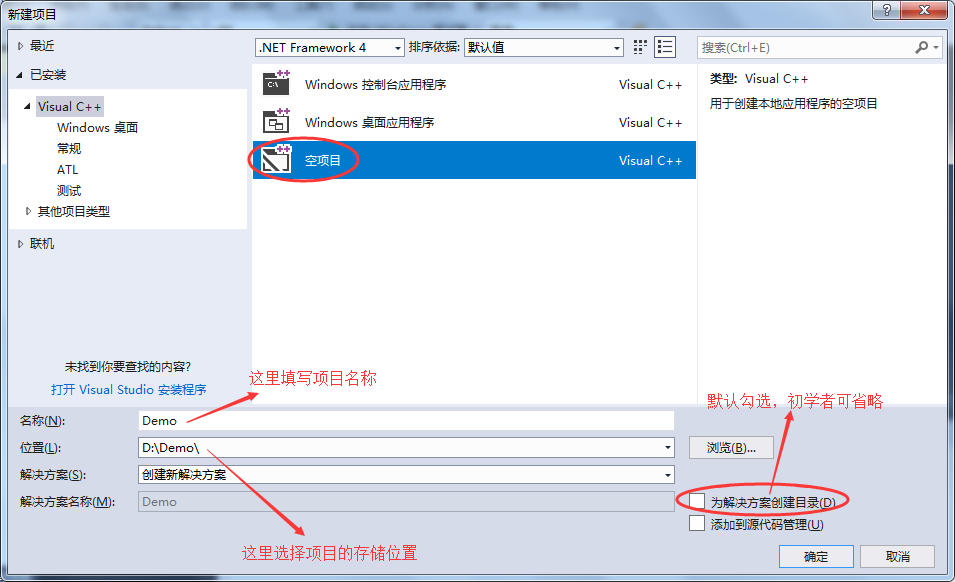
### 实验环境和搭建

本次实验建议在Visual Studio中进行程序的编写和调试。下面以Visual Studio 2017为例简单介绍项目的创建和源文件的创建。

1. **创建新项目**

打开 VS 2017，在菜单栏中依次选择 “文件 --> 新建 --> 项目”：



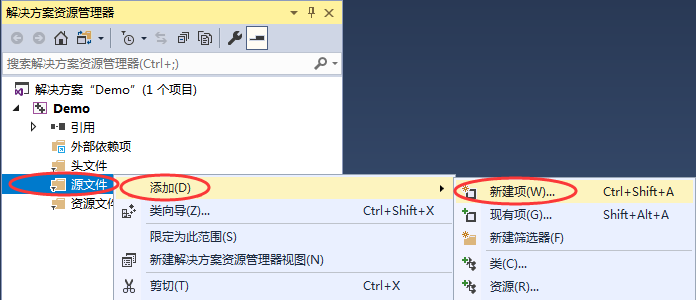


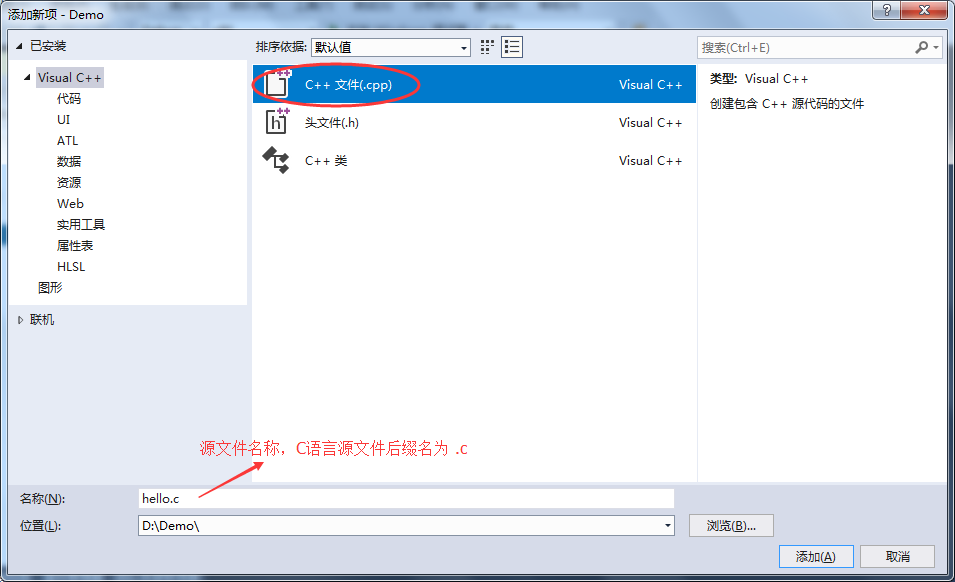
另外，项目名称和存储路径中最好不要包含中文。

点击 “确定” 按钮后，会直接进入项目可操作界面，我们将在这个界面完成所有的编程工作。

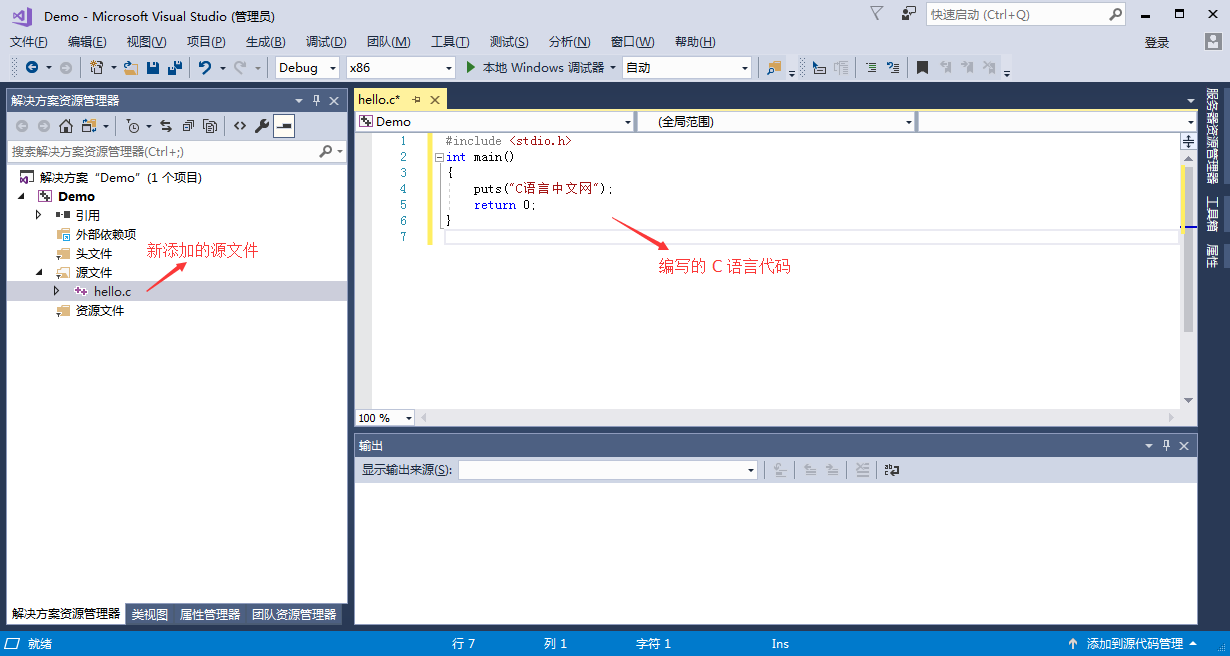
1. **添加源文件**

在 “源文件” 处右击鼠标，在弹出菜单中选择 “添加 --> 新建项” ，如下图所示：





点击 “添加” 按钮，就添加上了一个新的源文件。即可在编辑窗口中进行代码输入。



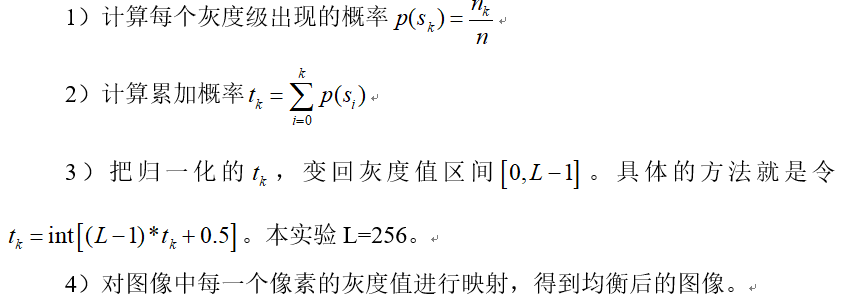
### 实验材料

seed.yuv （宽500 高500），采样格式为4:4:4.

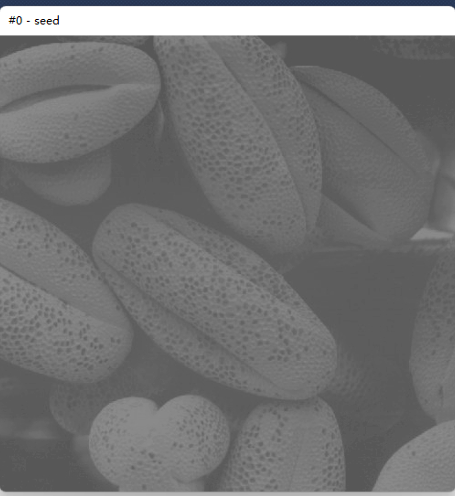
可以用YUVviewerPlus看上述图像以及处理后的图像，注意正确选择宽、高和采样格式。

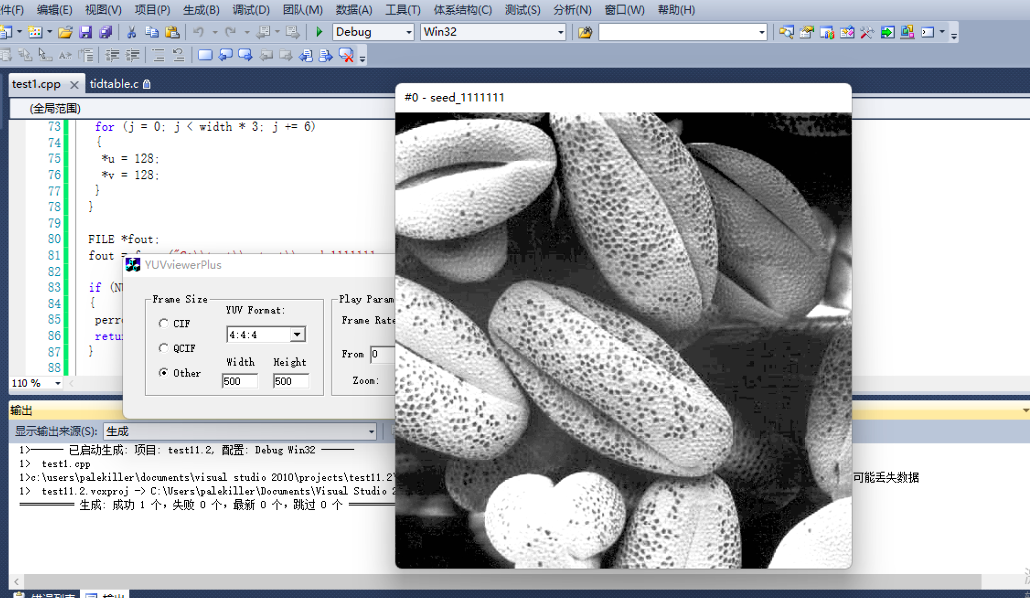
### 实验步骤

1. 只需要对Y分量进行直方图均衡。U、V分量直接置为128即可。
2. 直方图均衡的具体步骤：



注：本次实验已经给出读写操作的基本框架，因此只要实现算法即可。





#define \_CRT\_SECURE\_NO\_DEPRECATE  
#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
 int width=500;                          // 图片的宽度  
 int height=500;           // 图片的高度  
  
 int len = width \* height;   
  
 FILE \*fin;  
 fin = fopen("C:\\seed.yuv", "rb");  
 if (NULL == fin)  
 {  
  perror("open file is failed\n");  
  return -2;  
 }  
  
 unsigned char \*buff1 = (unsigned char \*)malloc(len\*3);           // 转换后数据存放的位置  
 if (NULL == buff1)  
 {  
  fprintf(stderr, "malloc data failed\n");  
  fclose(fin);  
  return -3;  
 }  
  
 fread(buff1,1,len\*3,fin);//读取fin，每次读取一个字节，读取width个项，最终放入buff1中  
  
 unsigned char \*buff2= buff1;//设置缓存区  
 if (NULL == buff2)  
 {  
  fprintf(stderr, "malloc buff failed\n");  
  fclose(fin);  
  return -4;  
 }  
  
 unsigned int sum[256] = {};  
 unsigned int sum\_all[256] = {};//用于存放累加灰度值  
  
  
//统计每个灰度级像素点个数  
 int i,j;  
 for(i=0;i<len\*3;i+=3)  
 {  
  sum[\*buff2]+=1;  
  buff2++;  
 }  
   
 //计算累加灰度值  
 for(i=0;i<256;i++)  
 {  
  for(j=0;j<=i;j++)  
  {  
  sum\_all[i]+=sum[j];  
  }  
 }  
  
 buff2 =buff1;//重置指针  
   
 for(i=0;i<500\*500\*3;i+=3)  
 {  
  \*buff2 = 255\*sum\_all[\*buff2]/500/500+0.5;//归一化处理  
  buff2++;  
 }  
  
 unsigned char \*u = buff1 + width \* height;   // u数据存放的位置  
 unsigned char \*v = u + width \* height / 2;  // v数据存放的位置   
  for (i = 0; i < height; i++)  
 {  
  fread(buff2, 1, width \* 3, fin);  
  for (j = 0; j < width \* 3; j += 6)  
  {  
   \*u = 128;  
   \*v = 128;  
  }  
 }  
  
 FILE \*fout;  
 fout = fopen("C:\\test\\output\\seed\_1111111.yuv", "wb");  
  
 if (NULL == fout)  
 {  
  perror("open newfile is failed\n");  
  return -3;  
 }  
  
 fwrite(buff1, 1, len\*3, fout);//将buff1中len\*3写入fout  
 fclose(fout);  
  
 return 0;  
}