山东大学 计算机科学与技术 学院

计算机视觉 课程实验报告

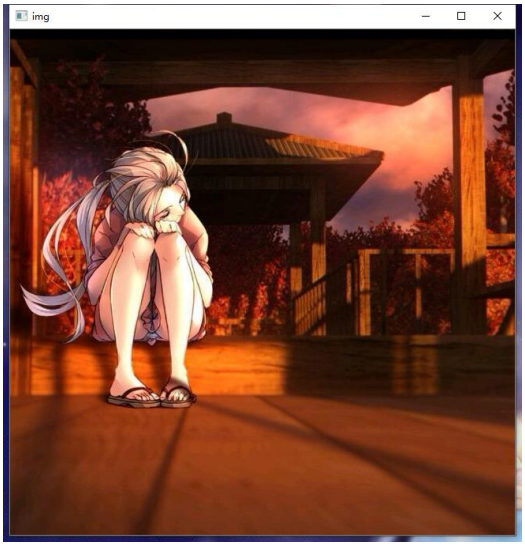
学号：201822130233	姓名： 李云龙	
实验题目：图像基本操作		

实验过程中遇到和解决的问题：

（记录实验过程中遇到的问题，以及解决过程和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）

教学 ppt 中图像通道分离主要用指针的移动，其实就是一个 2 层循环，由于图像坐标轴方向原因，用 y 遍历行，x 遍历列。以 jpg 格式图像为例，位于（y，x）的像素是 RGB 类型的，占据 3 字节，根据想读取的通道数进行一个位移到某字节并赋值到输出中的相应位置中。由于输出图像的格式为 CV_8UC1，因此 1 个字节表现出来就是一个灰度值，相应的输出图像是一个灰度图。

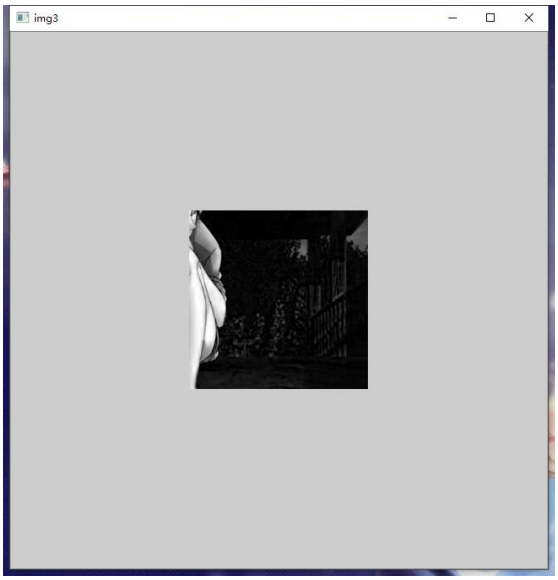
原图：



以获取B通道（channelToGet=0）为例：

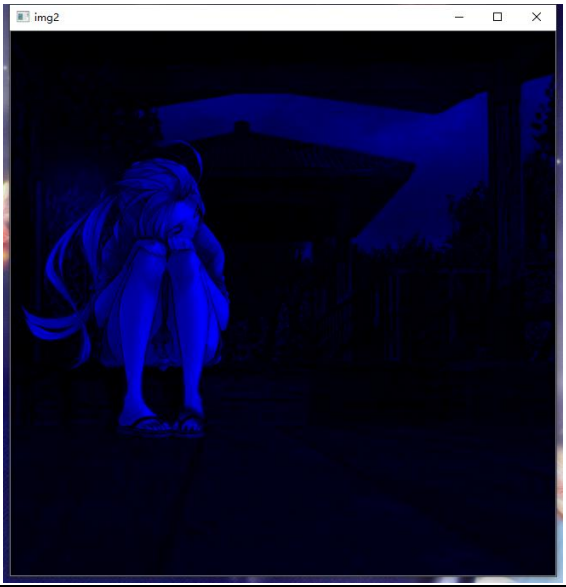
图像子区域操作即给定一个感兴趣区域（cv::Rect表示），调用getChannel函数实现子区域通道的获取和输出，`void getChannel(const uchar *input, int width, int height, int inStep, int inChannels, uchar *output, int outStep, int channelToGet)` 中传入的input指针指向ROI的第一个像素的地址（通过`char* get_pixel(const uchar *img, int x, int y, int step, int nc)` 获取），另外若也想直接输出到子区域，可传入指向对应位置的output指针（同样通过`char* get_pixel(const uchar *img, int x, int y, int step, int nc)` 获取），或者其他子区域。因此，最终实现为`getChannel(get_pixel(input, x1, y1, inStep, inChannels), rect.width, rect.height, inStep, inChannels, get_pixel(output, x1, y1, outStep, 1), outStep, channelToGet);`

子区域（以中间的矩形区域为例）：



若想转换成RGB图，可以把格式设为`CV_8UC3`，并把读取通道的值写到对应位置后（如`output[y * outStep + x * inChannels + channelToGet] = input[y * inStep + x * inChannels + channelToGet];`）其他通道上的值设为0，子区域操作中output指针需要修改为`get_pixel(output, x1, y1, outStep, inChannels)` 才能得到正确地址

以获取B通道（channelToGet=0）为例：



由于 channelToGet 在代码中算作一个偏移量，因此如果取值超过通道数，偏移量会多偏移，以 500 为例，余 3 得 2，输出 R 通道分离后的图像，同时可以看出输出图像整体上有一个偏移。最终输出为：

结果分析与体会：

由于图像在内存中的存储方式（交叉存储，顺序存储），程序中访问图像中的某像素或像素的某通道也需要相应的方式。图像的类型可能不同，这决定了图像的通道数及位深度等信息，如 CV_8UC3 代表 uchar，3 通道，每通道占 8 位。

图像的 step 与像素宽乘以通道数是不一样的，由于在内存中图像的像素可能只占总存储区域的一部分，而像素宽乘以通道数是这一部分的宽度，但加上其他部分的宽度才是 step，想要访问下一行的像素必须加 step。