

# 浙江大学实验报告

专业：求是科学班（计算机）

姓名：蒋仕彪

学号：3170102587

日期：2019/11/23

课程名称：计算机视觉 指导老师：宋明黎 成绩：

实验名称：HW#1: 制作个人视频

## 一、实验目的和要求

对输入的一个彩色视频与五张以上照片，用 OpenCV 实现以下功能或要求：

1. 命令行格式：“xxx.exe 放视频与照片的文件夹路径”，（例如 MyMakeVideo.exe C:\input ）【假设该文件夹下面只有一个 avi 视频文件与若干 jpg 文件】
2. 将输入的视频与照片处理成同样长宽后，合在一起生成一个视频；
3. 这个新视频中，编程生成一个片头，然后按幻灯片形式播放这些输入照片，
4. 最后按视频原来速度播放输入的视频；
5. 新视频中要在底部打上含自己学号与姓名等信息的字幕；
6. 有能力的同学，可以编程实现镜头切换效果；
7. 视频文件无需上载，但在实验报告里必须贴图展示。

## 二、实验内容和原理

### 2.1 视频和图片的存储和表示

OpenCV 里用 VideoCapture 类来表示视频。

读入一个目标 avi 文件可以直接调用构造函数：VideoCapture(path)。调取 VideoCapture 成员的方法有点奇怪：统一用 get(k) 函数，不同的 k 决定了不同的参数。例如：用 CV\_CAP\_PROP\_FPS 表示帧速，用 CV\_CAP\_PROP\_FRAME\_WIDTH 是表示宽度，用 CV\_CAP\_PROP\_FRAME\_COUNT 表示总帧数。

OpenCV 里用 Mat 类来表示图片。

Mat 里的成员有：cols（总列数），rows（总行数），data（存储像素的指针首地址）。imread 可以直接从目标地址读取一个 Mat 类的图片。用流操作 Video << Mat 可以把一张图片附到给定视频后面。

OpenCV 里 VideoWriter 类来输入视频。构造 VideoWriter 时需要尺寸、帧速、输出地址等参数。

此外，还有一些常用的类，如 Size(width, height) 是表示图片尺寸的类。

### 2.2 拉取目标文件夹里的所有文件

我使用了 <io.h> 里的 \_finddata\_t 来辅助文件的拉取。\_findfirst 和 \_findnext 可以遍历文件夹里的所有文件。依次判断后缀名是否是 jpg 或者 avi，如果是的话就读入。有一个注意点是：在 Windows10 下，hFile 必须定义成 long long（Win 7 以前都是 long），否则会出现运行时错误。以下是代码：

```
pair<vector<Mat>, string>getPicturePath(char *buffer){
    vector<Mat> files;
    string video = "";
    string path = buffer; path = path + "\\";
    long long hFile = 0;
    struct _finddata_t fileInfo;
    string pathName;
```

```

hFile=_findfirst(pathName.assign(buffer).append("\\*").c_str(),&fileInfo);
if (hFile != -1) {
    do {
        if (!(fileInfo.attrib&_A_SUBDIR)) {
            string name = fileInfo.name;
            string ex = name.substr(name.length() - 3, 3);
            if (ex == "jpg")
                files.push_back(imread(path + name));
            else if (ex == "avi") {
                if (video != "Result.avi")
                    video = path + name;
            }
        }
    } while (_findnext(hFile, &fileInfo) == 0);
}
return make_pair(files, video);
}

```

## 2.3 加入字幕

加入字幕的核心函数是 `putText(Mat, String, point, font, size, color)`：它会把给定字符串 `string` 打印到图片 `Mat` 上，左上角的像素点是 `point`，字体是 `font`，大小是 `size`，颜色是 `color`。

所以我们需要知道被加入字幕的图片的长和宽，并通过计算，精确定位放入字幕的位置。`getTextSize` 会帮我们算出，给定形式的字幕的宽度和高度。最终我把黑色字幕打印至整张图靠下的位置，并居中。

```

Mat drawWords(Mat img) {
    for (int line = 0; line < 2; line++) {
        string text = line == 0 ? "Name: Jiang Shibiao" : "ID: 3170102587";
        Size textsize = getTextSize(text, FONT_HERSHEY_COMPLEX, 2, 1, 0);
        Point org((img.cols - textsize.width) / 2, (img.rows - textsize.height)
            / 32 * (30 + line * 2));
        putText(img, text, org, FONT_HERSHEY_COMPLEX, 2, Scalar(0, 0, 0), 2);
    }
    return img;
}

```

## 2.4 将图片并入视频

每张图片的大小不一定符合视频，所以我们首先要做一步调整大小的操作。`resize(A, B, size)` 表示把 `Mat A` 缩放成 `Mat B`，新图片的大小是 `size`。

如果直接把图片添加进视频流，它只会被播放一帧。所以我设了一个常量 `PIC_LEN`，每次加入一张图片时往视频流里连续放入那么多张相同的视频。这样就达到了幻灯片的效果。如下：

```

for (int i = 0; i < PIC_LEN; i++) writer << file.first[j];

```

## 2.5 生成视频片头

如何才能简单地生成一段动画呢？

我想到了分形。分形有一个好处是，微调参数后分形的变化是连续的。这样我们设一个常量 `START_PIC` 表示片头的帧数，每次用当前的循环变量 `i` 操控分形的参数，使其缓缓地变化。

这里我取了 Julia 集，参数为  $c = 0.285$ ,  $0 \sim 0.01$ 。以下是分形的代码：

```
Mat drawPic(int i, int max_pic, int width, int height){
    double XMax = 1, XMin = -1, YMax = 2, YMin = -2; //控制分形的边界
    double deltaX = (XMax - XMin) / width;
    double deltaY = (YMax - YMin) / height;
    int max_iterations = MAX_COLOR; //分形的最大轮数
    double max_size = 4.0;
    Mat img(Size(width, height), CV_8UC3);
    for (int row = 0; row < height; row++){
        for (int col = 0; col < width; col++){
            int color = 0; MyComplex c, z;
            z.real = XMin + col * deltaX;
            z.imag = YMin + row * deltaY;
            c.real = 0.285;
            c.imag = i * 1.0 / max_pic / 100;
            while ((color < max_iterations) && ((z.imag * z.imag + z.real * z.r
eal) < max_size)){
                double tmp = z.real * z.real - z.imag * z.imag + c.real;
                z.imag = z.imag * z.real + z.real * z.imag + c.imag;
                z.real = tmp; color++;
            }
            if (color >= max_iterations) color = 255;
            color %= MAX_COLOR;
            for (int i = 0; i < 3; i++) {
                int nowc = i == 0 ? B[color] : (i == 1 ? G[color] : R[color]);
                *(img.data + img.step[0] * row + img.step[1] * col + img.elemSi
ze1() * i) = nowc;
            }
        }
    }
    cvtColor(img, img, CV_HSV2BGR); //将 HSV 转化成 RGB
    return img;
}
```

## 2.7\* 镜头切换效果

镜头切换效果其实很简单。我们只需在两张图片之间生成一些“补间图片”加入视频流。例如，将前一张图 A 划成一个个单位正方形，交替染成黑色和白色。在过渡的时候先把其中一半变成目标图片，就有

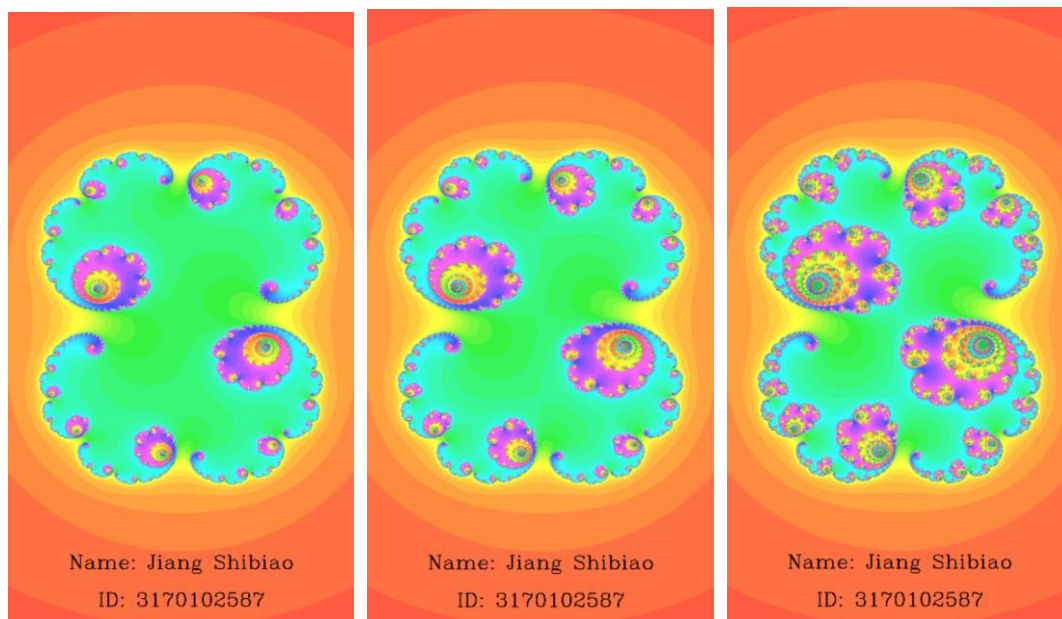
一种“百叶窗”式的效果了。

我实现的是一个均匀过渡：设一个常量 `PIC_CONVERT`，用循环变量 `I` 枚举它。假设原始图像某一个像素值是 `acol`，新图像对应像素 `bcol`，我就把第 `i` 张的过渡像素设成  $acol + (bcol - acol) * i / PIC\_CONVERT$ 。

需要注意的一个问题是，C++ 里类拷贝的时候，所有指针都是浅拷贝。在复制一个 `Mat` 图像的时候，记得把 `data` 数组重新 `malloc` 一下，或者直接用 `Mat(size, CV_8UC3)` 来新建。

### 三、成果展示

```
C:\Users\jiang\Desktop>lab1 E:\ZJU大学生活\课程学习\计算机视觉\lab1\lab1
Find picture: 1.jpg
Find picture: 2.jpg
[ INFO:0] VIDEOIO: Enabled backends(5, sorted by priority): FFMPEG(1000); MSMF(990); DSHOW(980); CV_IMAGES(970); CV_MJPEG(960)
Total frame: 711
请按任意键继续. . .
```



分形片头展示



镜头切换展示 1





镜头切换展示 2



结尾处原视频展示

## 四、感想

本次实验探究了 opencv 的基本操作。由于之前没有接触过，网上的博客又特别乱，一些基本的接口学起来挺累的。但是最后做成视频成就感满满。

Opencv 的调试很有难度。我遇到了两个调了很久才发现的错误：

①在某个博客上学习了 C++ 文件遍历的方法，使用起来却一直 RE。仔细排查原因和网上搜索后我发现，从 Win10 开始那个函数的返回类型有变化，从 long 变成了 long long……

②在镜头切换时，切换效果一直不明显。后来我发现我在 a 变换 b 时，所得结果 c 的初始化是 c=a。但是 Mat 类里的像素数组 data 是指针，等号的时候是浅拷贝，所以生成 c 后把 a 也改掉了……