

浙江大学

计算机视觉作业报告

作业名称：简易小视频制作

姓 名：陈润健

学 号：3160103989

电子邮箱：3160103989@zju.edu.cn

联系电话：18868104871

导 师：潘纲



2018 年 11 月 24 日

简易小视频制作

一、 作业已实现的功能简述及运行简要说明

1. 生成一段小视频，片头要含有自己的学号和姓名。
2. 视频中的每一帧都要调用 GUI 生成。
3. 实现镜头变换的效果。

二、 作业的开发与运行环境

IDE: Xcode 10.1

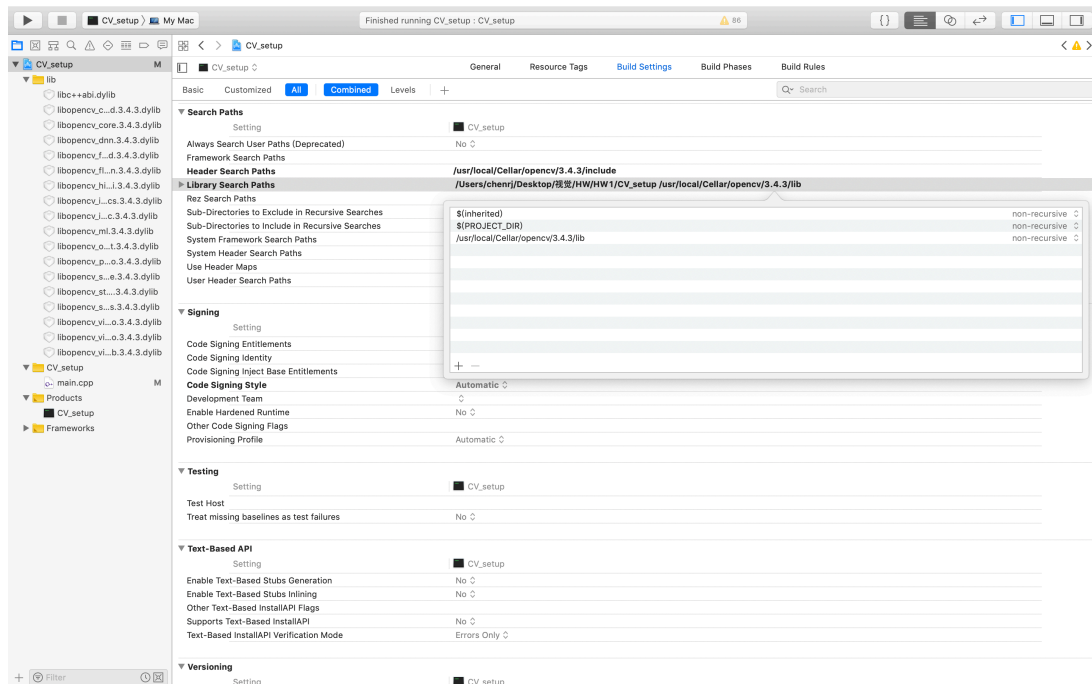
操作系统: MacOS 10.14.1

SDK: macosx10.14

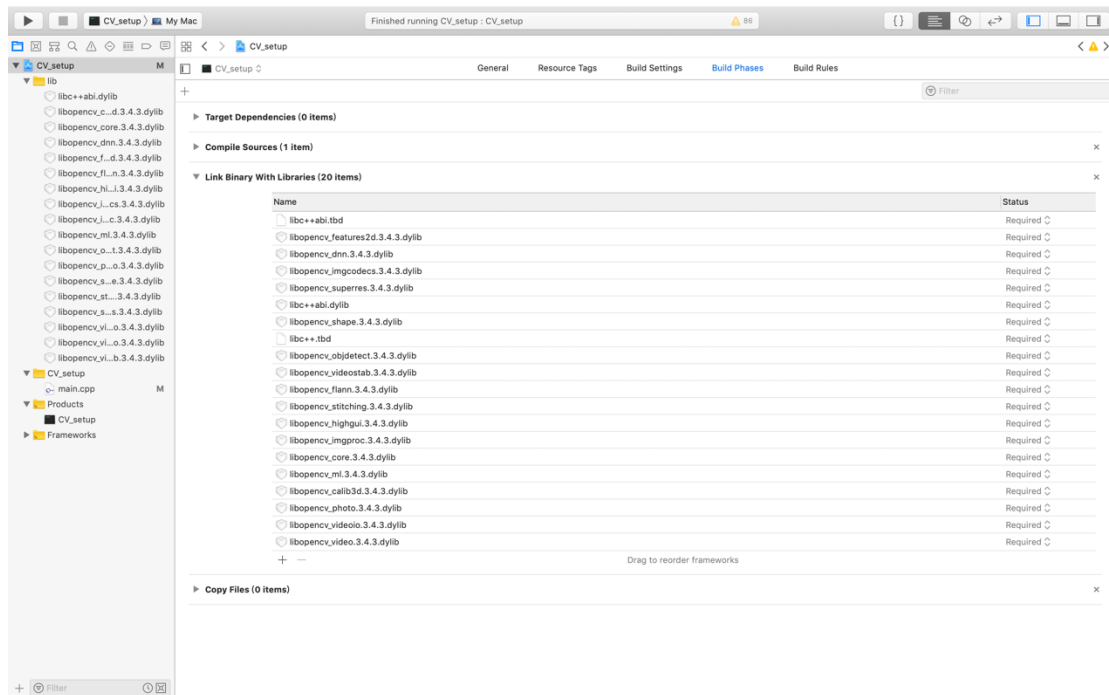
三、 基本思路，用到的函数及流程

1. 配置 OpenCV 以及工程中路径等的设置

- 1) 在 mac 系统下，直接使用 brew 安装 OpenCV 即可。
- 2) Xcode 中工程的配置：
 - a) 找到 OpenCV 的路径，配置头文件搜索路径和库搜索路径。



b) 添加需要的 dylib



2. 新建视频，生成每一帧图像，并加到视频中：

1) 一开始使用课件上的相关类和函数进行尝试，大致如下：

```
int fps = 25; //define frequency
int H = 574, W = 750, isColor=1; //define Height and Weight
int num = 1000; //define number of frame
CvVideoWriter *new_video; //a new video
new_video = cvCreateVideoWriter ("hw1.avi", CV_FOURCC('P', 'I', 'M', '1'), fps, cvSize(W, H), isColor);

IplImage* img = 0;
int nFrames = 50; for(i=0; i<nFrames; i++)
{
    img = cvQueryFrame(capture);
    cvWriteFrame(new_video, img);
}
```

- 2) 运行报错，无法创建视频文件。在网上查找之后发现，这些函数是 opencv1 版本的，而我在系统中安装的是 opencv3，因此会出现上述运行错误。
- 3) 重新找到 opencv3 的例程之后，发现可以使用以下函数进行图像的绘制和视频的创建：

视频的创建:

```
int fps = 25; //define frequency
int H = 2560, W = 1600; //define Height and Weight
int num = 100; //define number of frame
VideoWriter new_video; //a new video

Size S = Size(H,W); //define video size

new_video.open("/Users/chenrj/Desktop/视觉/HW/HW1/hw1.avi", 0, fps, S, true); //create video

//check if video is opened
if (!new_video.isOpened())
{
    std::cout << "Could not open the output video for write: " << std::endl;
    return -1;
}
```

图像的绘制:

```
Mat basic_image(int H, int W, int Is_man=1)
{
    Mat new_image = Mat(W, H, CV_8UC3, cvScalar(255, 255, 255)); //a new image

    //center of the image
    Point Center;
    Center.x = H/2;
    Center.y = W/2;

    //put texts onto the image
    Point ID_pos(Center.x, Center.y);
    Point Name_pos(Center.x, Center.y);
    Name_pos.y -= 100;

    putText(new_image, "3160103989", ID_pos, CV_FONT_HERSHEY_COMPLEX, 3, cvScalar(0, 0, 255));
    putText(new_image, "Chen Runjian", Name_pos, CV_FONT_HERSHEY_COMPLEX, 5.5, cvScalar(0, 0, 255));

    rectangle(new_image, *new Point(200, 200), *new Point(Center.x, Center.y), Scalar(255, 0, 0));
    //draw a man
    if (Is_man)
    {
        Point head_pos(Center.x, Center.y+200);

        circle(new_image, head_pos, 150, Scalar(0, 0, 0));

        line(new_image, *new Point(head_pos.x, head_pos.y+150), *new Point(head_pos.x, head_pos.y+450), Scalar(0, 0, 0));

        line(new_image, *new Point(head_pos.x-200, head_pos.y+250), *new Point(head_pos.x+200, head_pos.y+250), Scalar(0, 0, 0));

        line(new_image, *new Point(head_pos.x, head_pos.y+450), *new Point(head_pos.x-300, head_pos.y+700), Scalar(0, 0, 0));

        line(new_image, *new Point(head_pos.x, head_pos.y+450), *new Point(head_pos.x+300, head_pos.y+700), Scalar(0, 0, 0));
    }

    return new_image;
}
```

视频帧的插入:

```
//put images into the video
new_video << new_image;
```

3. 产生镜头变换效果:

- 1) 移动的物体: 对于每一帧产生的图像, 字和图形位置渐进改变即可。

```
for (int i = 1 ; i <= num ; i++)
{
    Mat new_image = Mat(W,H , CV_8UC3,cvScalar(255,255,255)); //a new image

    //center of the image
    Point Center;
    Center.x = H/2;
    Center.y = W/2;

    //put moving characters on the image
    int moving_step = H/(2*num);

    Point ID_pos(Center.x-H/2+i*moving_step,Center.y);
    Point Name_pos(Center.x-H/2+i*moving_step,Center.y);
    Name_pos.y -= 100;

    putText(new_image, "3160103989", ID_pos, CV_FONT_HERSHEY_COMPLEX, 3, CvScalar(0,0,255));
    putText(new_image, "Chen Runjian", Name_pos, CV_FONT_HERSHEY_COMPLEX, 5.5, CvScalar(0,0,255));

    rectangle(new_image, *new Point(200+i*moving_step-H/2,200) , *new Point(Center.x+i*moving_step-H/2,Center.y),Scalar(255,0,0));

    //draw a moving man
    Point head_pos(Center.x+H/2-i*moving_step,Center.y+200);

    circle(new_image,head_pos,150,Scalar(0,0,0));

    line(new_image,* new Point(head_pos.x,head_pos.y+150),* new Point(head_pos.x,head_pos.y+450),Scalar(0,0,0));

    line(new_image,* new Point(head_pos.x-200,head_pos.y+250),* new Point(head_pos.x+200,head_pos.y+250),Scalar(0,0,0));

    line(new_image,* new Point(head_pos.x,head_pos.y+450),* new Point(head_pos.x-300,head_pos.y+700),Scalar(0,0,0));

    line(new_image,* new Point(head_pos.x,head_pos.y+450),* new Point(head_pos.x+300,head_pos.y+700),Scalar(0,0,0));

    //put images into the video
    new_video << new_image;
}
```

- 2) 随机多边形的效果: 采用 OpenCV 提供的随机函数, 随机生成多边形的顶点和多边形的填充颜色, 即可随机生成多边形图案。(一开始产生的都是一样的多边形, 后来发现是随机化的种子没有设置好, 改成时间作为随机化种子之后, 就产生了不同的多边形)

```
Mat random_Pol(int W,int H)
{
    double x1 = -W/2, x2 = W*3/2, y1 = -H/2, y2 = H*3/2;
    Mat image = Mat(H, W, CV_8UC3,cvScalar(0,0,0)); //create a new image

    //random generator
    RNG rng(((unsigned)time(NULL)));
    Point pt1, pt2;

    int lineType = LINE_8;

    //randomly create vertex
    Point pts[2][3];
    pts[0][0].x = rng.uniform(x1, x2);
    pts[0][0].y = rng.uniform(y1, y2);
    pts[0][1].x = rng.uniform(x1, x2);
    pts[0][1].y = rng.uniform(y1, y2);
    pts[0][2].x = rng.uniform(x1, x2);
    pts[0][2].y = rng.uniform(y1, y2);
    pts[1][0].x = rng.uniform(x1, x2);
    pts[1][0].y = rng.uniform(y1, y2);
    pts[1][1].x = rng.uniform(x1, x2);
    pts[1][1].y = rng.uniform(y1, y2);
    pts[1][2].x = rng.uniform(x1, x2);
    pts[1][2].y = rng.uniform(y1, y2);
    const Point* ppt[2] = {pts[0], pts[1]};
    int npt[] = {3, 3};
    //fill the poly
    fillPoly(image, ppt, npt, 2, randomColor(rng), lineType);

    return image;
}
```

四、 实验结果与分析：

生成了一段符合要求的小视频，由于原视频比较大，进行压缩之后上交，如果老师需要检查原 avi 视频，麻烦老师运行可执行文件，即可在桌面->视觉->HW->HW1 目录下找到对应的视频，麻烦老师了！

五、 结论与心得体会

OpenCV 是一个强大的图像处理库，在这一次的实验中，我完成了 OpenCV 的安装，简单调用和调试的工作，初步体验到了它的魅力，希望在接下来的课程和作业中，能够接触到更多实用的函数，同时学习到更多关于计算机视觉的理论知识，尝试自己实现各种有趣的功能（特征检测，跟踪等等）。