

# 浙江大学



## 浙江大学实验报告

课 程 : 计算机视觉

实验名称 : 制作个人视频

姓 名 : 吕皓明

专 业 : 计算机科学与技术

学 号 : 3190103303

指导老师 : 宋明黎

日 期 : 2021/11/24

# 1 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，最后按视频原来速度播放输入的视频；
4. 新视频中要在底部打上含自己学号与姓名等信息的字幕；
5. 有能力的同学，可以编程实现镜头切换效果；
6. 视频文件无需上载，但在实验报告里必须贴图展示输入输出效果

# 2 实验内容和原理

1. CMake
2. Opencv

# 3 实验步骤和分析

## 3.1 Mac OS安装opencv

使用homebrew安装opencv

```
brew install opencv
```

## 3.2 编写CMakeLists

```
cmake_minimum_required(VERSION 3.15)
set(CMAKE_CXX_STANDARD 14)
project(hw1)

find_package( OpenCV REQUIRED )
include_directories( ${OpenCV_INCLUDE_DIRS} )

set(SOURCE_FILES hw1.cpp)
add_executable(hw1 ${SOURCE_FILES})
target_link_libraries (hw1 ${OpenCV_LIBS})
```

## 3.3 代码编写

1. 初始化 `cv::VideoWriter` 对象

先确定 `VideoWriter` 所需要的参数，然后根据这些参数创建一个新的对象。

```

int frameFPS = 30;
int frameWidth = 640, frameHeight = 360;
Size frameSize = Size(frameWidth, frameHeight);
string resultName = "result.avi";
bool isColor = true;
VideoWriter writer = VideoWriter(resultName, VideoWriter::fourcc('M', 'J', 'P', 'G'), frameFPS,
frameSize, isColor);

```

## 2. 将图片变换为幻灯片形式播放

首先将文件夹中所有的图片取出，然后将其进行大小变换，接着打印自己的个人信息，最后设置以2s一张的速度进行播放。

```

vector<string> imgPaths;
glob(filePath + "/*.jpg", imgPaths);
for (auto &each : imgPaths) {
    Mat img = imread(each);
    Mat imgResized;
    resize(img, imgResized, frameSize);
    putText(
        imgResized,
        info,
        Point(30, 320),
        FONT_HERSHEY_SCRIPT_COMPLEX,
        1,
        Scalar(200, 200, 250),
        2,
        LINE_AA
    );
    for(int i = 0; i < 60; i++) {
        writer.write(imgResized);
    }
}

```

## 3. 将视频连接到幻灯片播放后原速播放

将视频的每一帧读取出来后与图片处理阶段进行一样的操作即可。

```

vector<string> aviPaths;
glob(filePath + "/*.avi", aviPaths);
string videoPath = aviPaths[0];
VideoCapture capture = VideoCapture(videoPath);
Mat frame, frameResized;
while (1) {
    capture >> frame;
}

```

```
if (frame.empty()) break;
resize(frame, frameResized, frameSize);
putText(
    frameResized,
    info,
    Point(30, 320),
    FONT_HERSHEY_SCRIPT_COMPLEX,
    1,
    Scalar(200, 200, 250),
    2,
    LINE_AA
);
writer.write(frameResized);
}
```

## 4 实验环境及运行方法

- 实验环境: Mac OS
- 运行方法

./hw1 "路径名"

## 5 实验结果展示

1. 编译运行后可以看到文件夹下生成了result.avi

注：终端中的红色输出是warning，并不影响运行

The screenshot shows the VS Code interface. The Explorer sidebar on the left lists project files: .vscode, assets, CMakeFiles, report, cmake\_install.cmake, CMakeCache.txt, CMakeLists.txt, hw1, hw1.cpp, Makefile, and result.avi. The Editor tab displays the hw1.cpp file with code for video processing. The Terminal tab at the bottom shows the command line output:

```
apple: ~ /Desktop/code/cv/hw1 make  
Consolidate compiler generated dependencies of target hw1  
[ 50%] Building CXX object CMakeFiles/hw1.dir/hw1.cpp.o  
[100%] Linking CXX executable hw1  
[100%] Built target hw1  
apple: ~ /Desktop/code/cv/hw1 ./hw1 ~ /Desktop/code/cv/hw1  
[cinepak @ 0x12e820a0] cinepak_decode failed  
apple: ~ /Desktop/code/cv/hw1
```

## 2. 查看视频输出

图片幻灯片放映举例：





视频输出举例：





## 6 心得体会

一开始觉得第一个作业难度挺大的，但是上网查了一些资料以后发现opencv提供的接口是真的好用，一下就做完了。过程中也没有碰到什么大的问题，只有一开始看的博客时间太老了，在调用putText的时候有一些参数写错了，比如CV\_FOURCC现在已经弃用了，该写成 [VideoWriter::fourcc](#)。