

HW01-制作无声小短片

3180101941 杨锐

软件开发说明

- @Windows
- Python == 3.7.6
- Numpy == 1.8.1
- Opencv-python == 4.2.0.34
- Pyinstaller == 4.1(用于打包python文件 -> 可执行文件)

算法设计思路

视频分为三个部分：

- 片头：浙大照片+个人照片+个人信息
- 简笔画：吃豆人+豆子
- 片尾：随机射线

下面分别详细讲述实现

片头

浙大照片 -> 向下平移出 -> 个人照片 -> 个人信息 -> 向左平移出

首先读取一张图片，并初始化视频参数：

```
# 读取图片
inPath = './img/zju.jpg'
outPath = './output.mp4'
image = cv2.imread(inPath)
if image is None:
    print("The image is null")
    image = np.zeros((640, 960, 3), np.uint8)
# 初始化视频
rows = 640
cols = 960
fps = 30
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
video = cv2.VideoWriter(outPath, fourcc, fps,
                        (cols, rows)) # (cols,rows)|
```

对图片进行`resize`，静态写入两秒视频。

使用`np.float32([[1, 0, x], [0, 1, y]])`设定平移矩阵，使用`cv2.warpAffine()`每一帧更新图片并读入视频。

在每一帧读入视频的同时，使用 `cv2.imshow` 展示当前图片，并使用 `cv2.waitKey()` 等待用户键入空格来暂停/继续（这意味着我们对图片的处理也暂停了）

```
# 读入图片ZJU
image = cv2.resize(image, (960, 640))
# 写入视频
for _ in range(1, 2*fps):
    video.write(image)
    cv2.imshow('image', image)
    if cv2.waitKey(30) & 0xFF == ord(' '):
        cv2.waitKey(0)
# [1,0,x],[0,1,y]设定平移矩阵
M = np.float32([[1, 0, 0], [0, 1, +5]])
for _ in range(1, 4*fps):
    video.write(image)
    image = cv2.warpAffine(image, M, (cols, rows))
    cv2.imshow('image', image)
    if cv2.waitKey(30) & 0xFF == ord(' '):
        cv2.waitKey(0)
```

读入个人图片的思路完全一样，增加一个 `cv2.putText()` 添加个人信息

简笔画

吃豆人出现 > 第一个豆子> 第二个豆子> ... > 吃豆人发射射线

由于要绘制多个豆子，而他们的区别仅在于位置和颜色的不同，因此封装成一个类并配合 `@staticmethod` 实现代码复现：

```
class AnimateDraw:
    @staticmethod
    def drawPacman(img, pacmanBodyCenter):
        @staticmethod
        def drawBeans(img, beanHeadCenter, bodyColor):
            @staticmethod
            def drawAnimate(video, fps):
```

吃豆人：使用 `cv2.ellipse()` 绘制一个3/4的扇形，使用 `cv2.circle` 绘制眼睛

```
def drawPacman(img, pacmanBodyCenter):
    cv2.ellipse(img, pacmanBodyCenter, (80, 80),
               0, 30, 330, (0, 255, 255), -1)
    pacmanEyeCenter = (110, 200)
    cv2.circle(img, pacmanEyeCenter, 12, (255, 255, 255), -1)
```

豆子：由于豆子涉及的点数较多，如果将每一个坐标写死，显然不利于复用，因此通过传入参数 `beanHeadCenter` 和 `bodysize` 计算出其余坐标，再使用 `cv2.fillPoly()` 绘制并填充多边形

```

def drawBeans(img, beanHeadCenter, bodyColor):
    # Draw head
    bodySize = (25, 22)
    cv2.ellipse(img, beanHeadCenter, bodySize,
                180, 0, 180, bodyColor, -1)
    # Draw eyes
    eyeLeft = (beanHeadCenter[0]-8, beanHeadCenter[1]-5)
    eyeRight = (beanHeadCenter[0]+8, beanHeadCenter[1]-5)
    cv2.circle(img, eyeLeft, 3, (255, 255, 255), -1)
    cv2.circle(img, eyeRight, 3, (255, 255, 255), -1)
    # Draw body
    bodyLeft = [beanHeadCenter[0]-bodySize[0], beanHeadCenter[1]]
    bodyRight = [beanHeadCenter[0]+bodySize[0], beanHeadCenter[1]]
    bodyLength = 50
    pts = np.array([
        [bodyLeft, [bodyLeft[0]-2, bodyLeft[1]+bodyLength],
         [bodyLeft[0]+7, bodyRight[1]+bodyLength-10],
         [bodyLeft[0]+16, bodyRight[1]+bodyLength],
         [bodyLeft[0]+25, bodyRight[1]+bodyLength-10],
         [bodyRight[0]-16, bodyRight[1]+bodyLength],
         [bodyRight[0]-7, bodyRight[1]+bodyLength-10],
         [bodyRight[0]+2, bodyRight[1]+bodyLength], bodyRight], np.int32]
    ])
    # 顶点个数: 9, 矩阵变成9*1*2维
    pts = pts.reshape((-1, 1, 2))
    cv2.fillPoly(img, [pts], bodyColor)

```

绘制动画:

通过每隔一定帧调用 `AnimateDraw.drawBeans()` 函数，并传入更新后的坐标和颜色，实现间隔出现，其他绘制和之前思路一致

```

# Draw beans
colors = (
    (255, 0, 0),
    (0, 255, 0),
    (0, 0, 255),
    (255, 0, 255),
    (255, 255, 0),
    (0, 255, 255),
)
for i in range(0, 7*fps):
    if(i % 30 == 0 and i < 180):
        AnimateDraw.drawBeans(img, (250+i*3, 256), colors[int(i/30)])
    video.write(img)
    cv2.imshow('image', img)
    if cv2.waitKey(30) & 0xFF == ord(' '):
        cv2.waitKey(0)

```

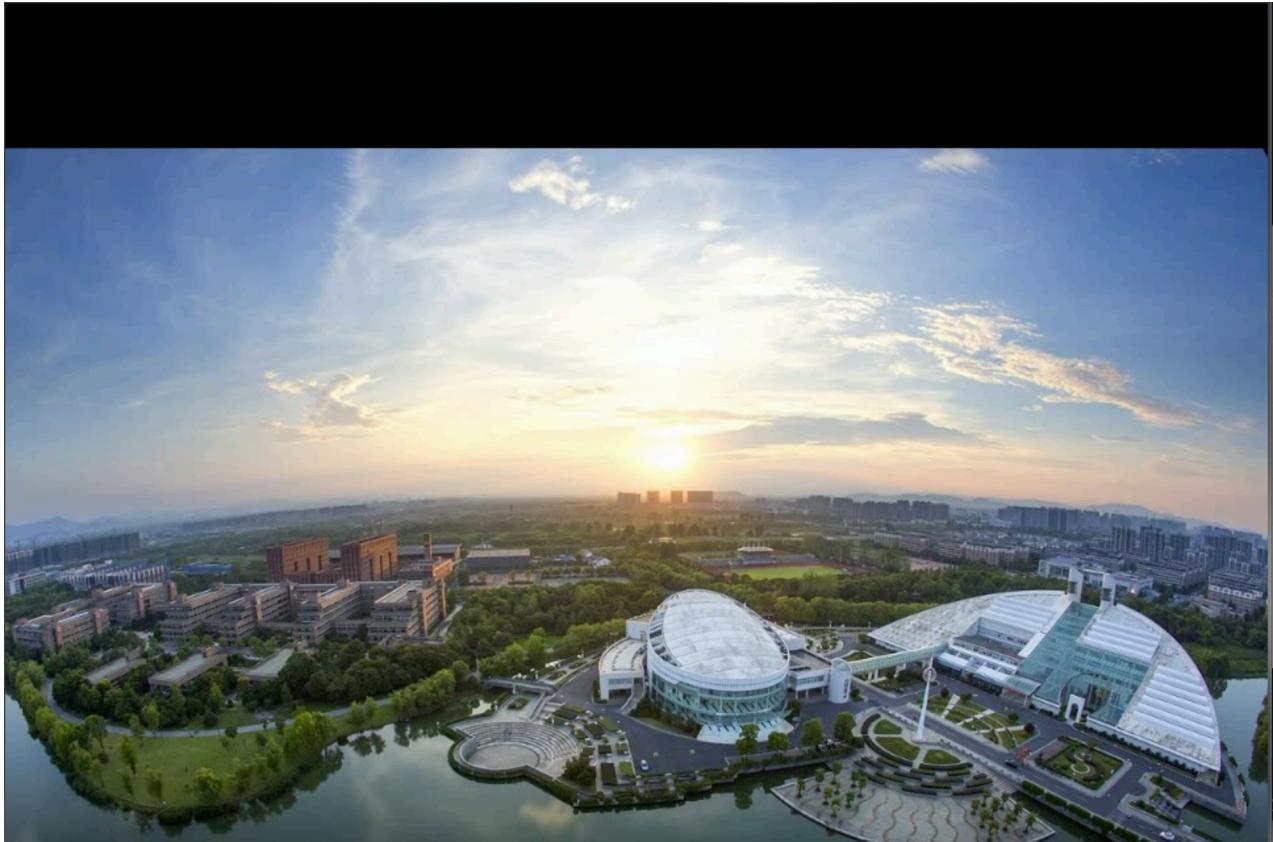
结尾

| 发射随机射线杀死豆子，结尾出现Thanks

```
# Random ray
for _ in range(0, 3*fps):
    pt1 = (int(random.random()*W3-W), int(random.random()*H3-H))
    pt2 = (int(random.random()*W3-W), int(random.random()*H3-H))
    pt3 = (int(random.random()*W3-W), int(random.random()*H3-H))
    pt4 = (int(random.random()*W3-W), int(random.random()*H3-H))
    pt5 = (int(random.random()*W3-W), int(random.random()*H3-H))
    pt6 = (int(random.random()*W3-W), int(random.random()*H3-H))
    pts = np.array([pt1, pt2, pt3, pt4, pt5, pt6], np.int32)
    pts = pts.reshape((-1, 1, 2))
    color = (random.randint(0, 255), random.randint(
        0, 255), random.randint(0, 255))
    cv2.polylines(image, [pts], True, color, 10, 0)
    video.write(image)
    cv2.imshow('image', image)
    if cv2.waitKey(delay) & 0xFF == ord(' '):
        cv2.waitKey(0)
    cv2.putText(image, 'Thanks', (350, 300), font,
                2, (255, 255, 255), 3, lineType=cv2.LINE_AA)
```

实验结果展示和分析

开头动画



个人信息

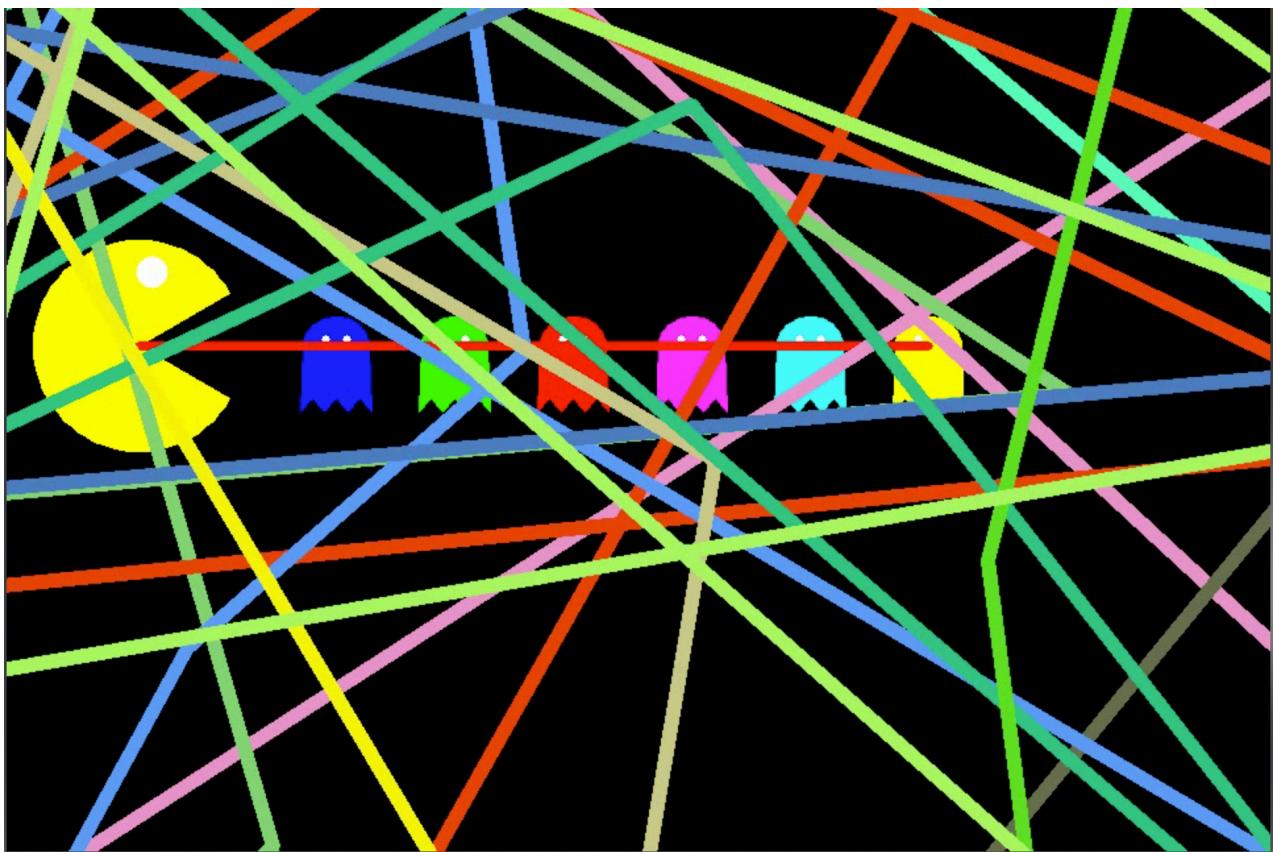


YangRui 3180101941

吃豆人



动画



结尾



完整版见exe/main.exe运行结果或output.mp4

编程体会

- 熟悉了opencv读/写图片的基本操作
- 熟悉了opencv读/写视频的基本操作
- 熟悉了opencv绘制图形的基本操作
- 熟悉了opencv图形变换（平移、旋转、缩放）的基本操作

个人照片

