

实验报告

课程名称：课程设计

实验名称——人脸识别表情动画

实验日期： 2017.5.10 班级： 数媒 1404

小组成员： 1030513312 那婉纯 1030514411 夏文利 1030514412 刘金玲

一、需求分析

人脸识别是所有的生物识别方法中应用最广泛的技术之一，人脸识别技术是一项近年来兴起的，但不大为人所知的新技术。针对目前年轻人越来越喜欢独特性自拍的现状，很多关于人脸识别添加特效的应用应运而生，我们小组也是针对这一现象，于是模仿市面上一些人脸识别的应用，像 Faceu、qq 视频聊天等，做一款人脸识别的表情软件。在这个应用中，我们添加了十几种不同的特效，有各种挂件以及各种好看的光效前景或背景，达到能根据人脸位置调整动画位置的目的，从而制作出不同的表情动画。

- (1) 目标用户：核心用户应为 15-35 岁的活跃在各社交平台上的年轻用户，其中喜欢自拍的女性用户为最主要群体
- (2) 核心价值：提供了动态贴纸、美颜功能，帮助用户简单创作出更有趣味、吸引力的自拍图片与视频，让用户在分享这些内容后获得更多关注与互动，满足了用户的虚荣心。

二、开发环境介绍

(一) openCVSharp

针对 c# 的计算机视觉库主要有两种，EmguCV 和 openCVSharp，其中 OpenCVSharp 包含以下几个.dll 文件：

OpenCvSharp.dll

OpenCvSharp.MachineLearning.dll

OpenCvSharp.Blob.dll

OpenCvSharp.UserInterface.dll

OpenCvSharp.Extensions.dll

OpenCvSharp.CPlusPlus.dll

OpenCvSharpExtern.dll

注：OpenCvSharpExtern.dll 并非 .Net 链接库，而是 C++ 链接库。在导出 exe 过程中最容易出现问题。

(二) 主要函数

针对人脸检测函数，opencv1 中的函数是 cvHaarDetectObjects，opencv2 中人脸检测使用的是 detectMultiScale 函数。函数 DetectMultiScale 可以检测出图片中所有的人脸，并将人脸用 vector 保存各个人脸的坐标、大小（用矩形表示），函数由分类器对象调用。

(三) 分类器 (.xml)

1. 加载分类器

```
haarCascade = new
```

```
CascadeClassifier(Application.streamingAssetsPath+"/haarcascades/haarcascade_frontalface_alt2.xml");
```

2. xml 的结构

haarcascade_frontalface_alt.xml 与 haarcascade_frontalface_alt2.xml 都是人脸识别的 Haar 特征分类器了。训练得到的分类器以 xml 形式保存，整体上它包括 stageType、featureType、height、width、stageParams、featureParams、stages、features 几个节点。

```
<?xml version="1.0"?>
- <opencv_storage>
- <cascade>
    <stageType>BOOST</stageType>
    <featureType>HOG</featureType>
    <height>30</height>
    <width>90</width>
    + <stageParams>
    + <featureParams>
    <stageNum>15</stageNum>
    + <stages>
    + <features>
    </cascade>
- </opencv_storage>
```

分类器的 Xml 文件整体结构

除 **stages** 和 **features** 外，其他主要是一些分类器的参数。**Stages** 中包含 15 个 **stage**（训练程序设定），每个 **stage** 中包含多个 **weakClassifiers**，而每个 **weakClassifier** 中又包含一个 **internalNodes** 和一个 **leafValues**。**internalNodes** 中四个变量代表一个 **node**，分别为 **node** 中的 **left/right** 标记、特征池中的 ID 和 **threshold**。**leafValues** 中两个变量代表一个 **node**，分别为 **left leaf** 的值和 **right leaf** 的值。



分类器的 Xml 文件具体结构

3. xml 的 read 结果

3.1.初始化了分类器的特征类型、最小检测窗口 size 等参数；

3.2.建立级联的分类器树；

3.3.提取了 xml 中的特征池。

3.4.data 成员变量的读取中同样可以分为两部分：分类器参数读取和 stage 分类树的建立。

3.5.完成 data 部分的载入后，接下来就是特征计算器（featureEvaluator）的载入。每一个 **node** 中都会计算特征池中的某一个特征，这个特征以 **featureIdx** 出现在 **node** 中。

4.其他分类器

haarcascade_eye.xml

haarcascade_eye_tree_eyeglasses.xml

haarcascade_frontalface_alt.xml

haarcascade_frontalface_alt_tree.xml

haarcascade_frontalface_default.xml

haarcascade_fullbody.xml

haarcascade_lefteye_2splits.xml

haarcascade_lowerbody.xml

haarcascade_mcs_eyepair_big.xml

haarcascade_mcs_eyepair_small.xml

haarcascade_mcs_leftear.xml

haarcascade_mcs_lefteye.xml

haarcascade_mcs_mouth.xml

haarcascade_mcs_nose.xml

haarcascade_mcs_rightear.xml

haarcascade_mcs_righteye.xml

haarcascade_mcs_upperbody.xml

haarcascade_profileface.xml

haarcascade_righteye_2splits.xml

haarcascade_smile.xml

haarcascade_upperbody.xml

注：如果要识别人体的其它部位，只需将上面的 **haarcascade_frontalface_alt2.xml** 分类器替换即可。

（四）调用摄像头，转换格式，进行读取

代码段 1:

```
cameraName = devices[0].name;
cameraTexture = new WebCamTexture(cameraName, mPreviewWidth, mPreviewHeight);
cameraTexture.Play();
```

代码段2:

```
Texture2D GetTexture2D(WebCamTexture wct)
{
    temp.SetPixels(wct.GetPixels());
    temp.Apply();
    return temp;
}
```

代码段3:

```
rt.LoadImage(bs);
rt.Apply();
m_material.mainTexture = rt;
```

(五) detectMultiScale 函数

在进入detectMultiScale函数之前，首先需要CascadeClassifier做初始化。

VS2015中对detectMultiScale函数定义如下:

```
OpenCvSharp.Rect[] CascadeClassifier.DetectMultiScale(Mat image, [double scaleFactor = 1.1], [int minNeighbors = 3], [HaarDetectionType flags = (HaarDetectionType)0], [Size? minSize = null], [Size? maxSize = null])
```

具体介绍如下:

void detectMultiScale(

const Mat& image, 待检测图片，一般为灰度图像加快检测速

CV_OUT vector<Rect>& objects, 被检测物体的矩形框向量组;

double scaleFactor = 1.1,

每一个图像尺度中的尺度参数，默认值为 1.1。

表示在前后两次相继的扫描中，搜索窗口的比例系数。默认为 1.1 即每次搜索窗口依次扩大 10%;

int minNeighbors = 3,

表示构成检测目标的相邻矩形的最小个数(默认为 3 个)。

如果组成检测目标的小矩形的个数和小于 min_neighbors - 1 都会被排除。

如果 min_neighbors 为 0, 则函数不做任何操作就返回所有的被检候选矩形框，这种设定值一般用在用户自定义对检测结果的组合程

int flags = 0,

要么使用默认值，要么使用 CV_HAAR_DO_CANNY_PRUNING

如果设置为 CV_HAAR_DO_CANNY_PRUNING，那么函数将会使用 Canny 边缘检测来排除边缘过多或过少的区域，因此这些区域通常不会是人脸所在区域;

Size minSize = Size(), 用来限制得到的目标区域的范围。

Size maxSize = Size()

);

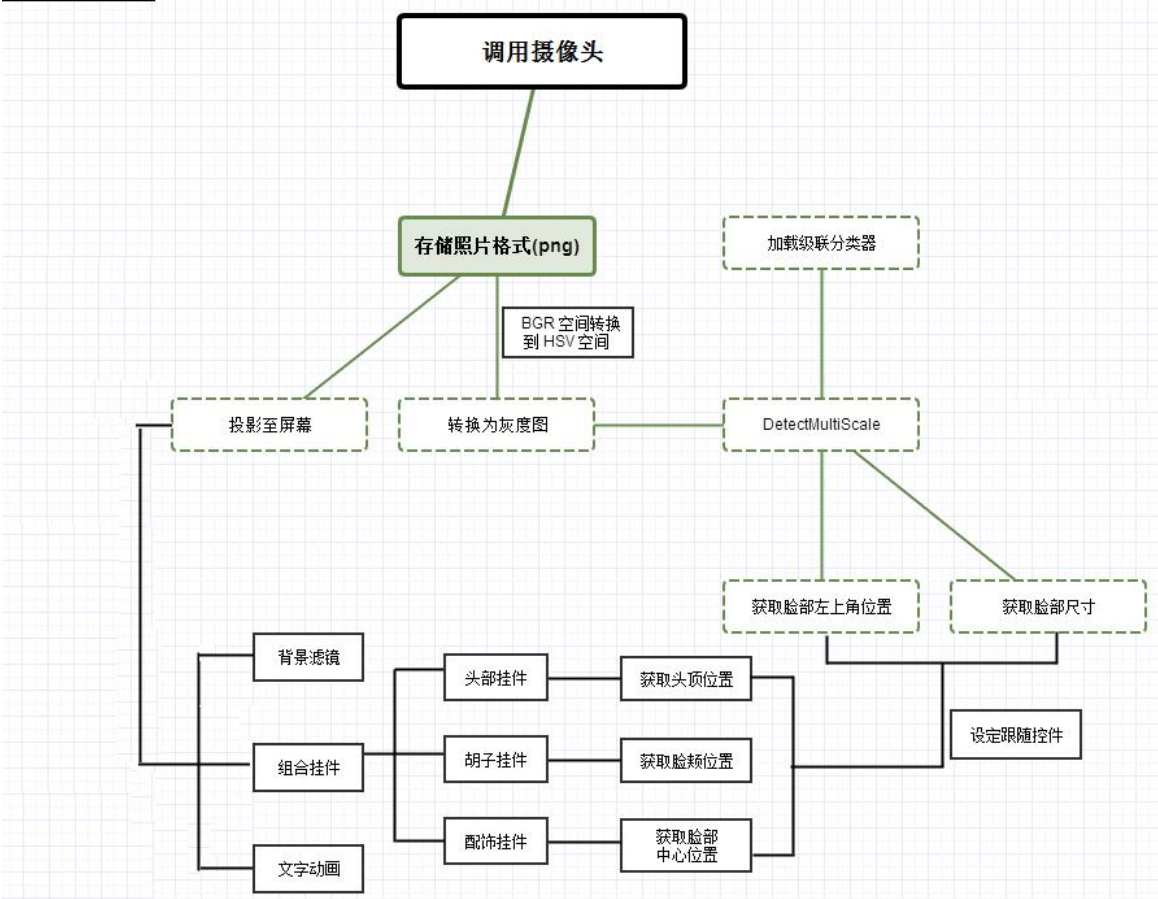
(六) 各种动画效果

文字动画：用户可以自己输入内容以后，点击空白区域发送输入的内容，点击不同的按钮就会给输入的内容添加不同的文字效果。

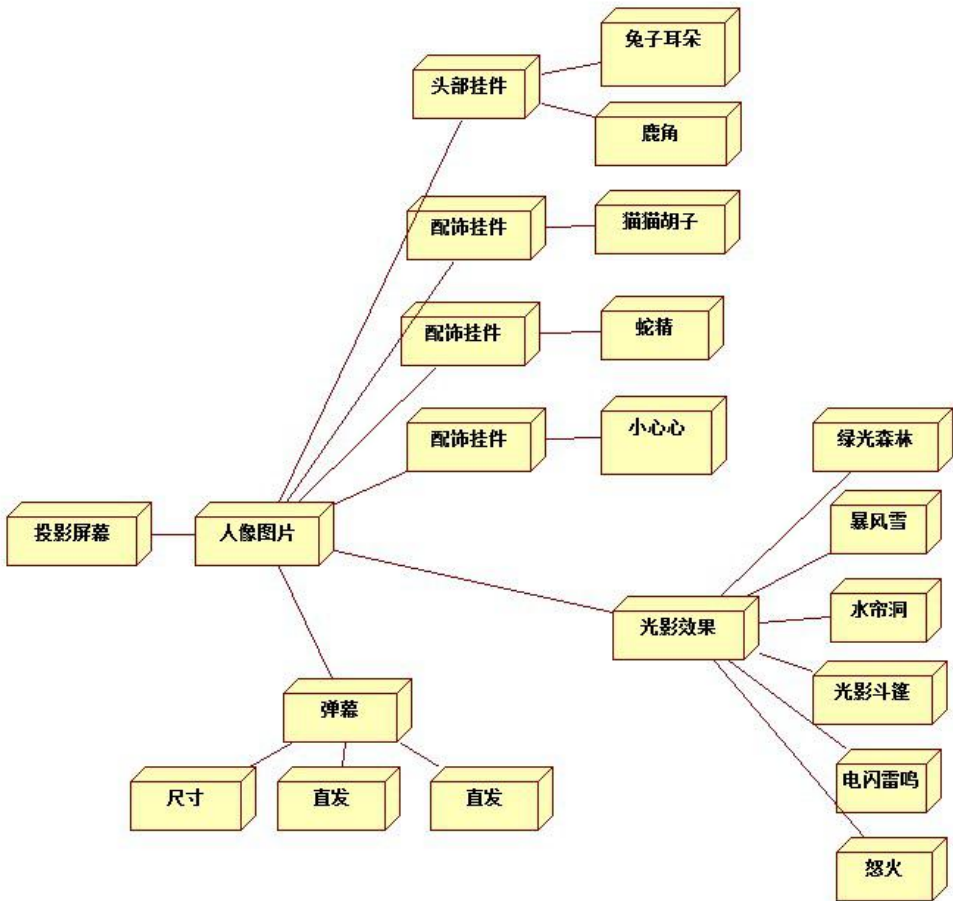
序列帧动画：按钮点击切换不同动画效果。

骨骼动画：不同组成部分的不同动画效果。

三、架构设计



四、模块设计



五、分工方案

介于这一次的实验内容和游戏有很大不同，没有分关卡设置内容，所以大部分实验内容都是小组成员合作完成，而呈现的每一动画效果均有多种方式复合，故具体分工如下：

openCv 人脸识别学习应用：那婉纯 刘金玲

序列帧动画：那婉纯 刘金玲 夏文利

关键帧动画：那婉纯 刘金玲 夏文利

骨骼动画：刘金玲 夏文利

粒子动画：那婉纯 刘金玲 夏文利

随机行为动画：那婉纯 夏文利（文字动画）

界面设计：刘金玲

六、遇到的问题以及解决方案

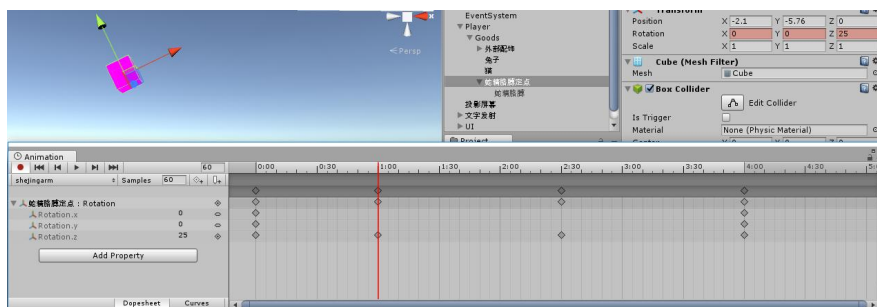
1. 定点旋转

通过动画编辑器设定的旋转是整体的一致旋转，但是在“蛇精”的组合动画中，需要让手臂呈现定点旋转效果。但是又不想通过代码设定。最后解决方案如下：

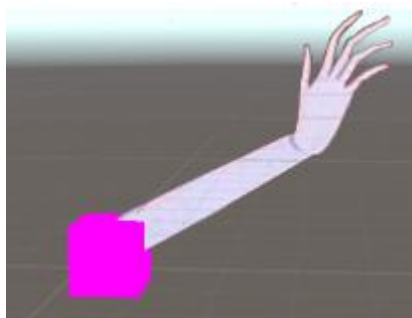


初始图片素材

在界面设定一个旋转的 Cube，设定好固定轴线的旋转效果。



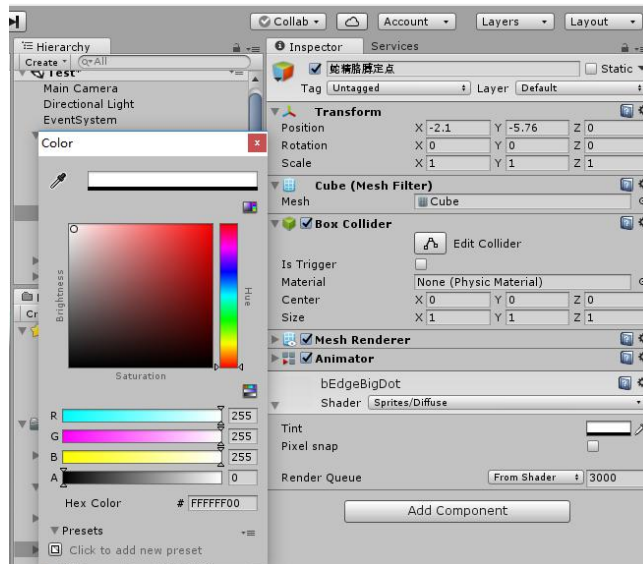
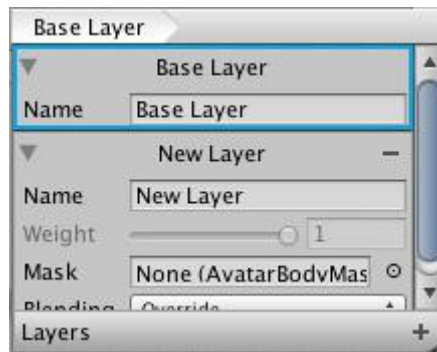
将素材固定在 Cube 上，连接处为定点。



将 Cube 设为透明。

2.动画层

Unity 使用“动画层”来管理身体不同部分的复杂状态机。可以使用 Animator 控制器左上角的层权重来管理动画层。



3.如何点击按钮换发射的文字预设

解决方法：在人物游戏物体上添加一个 public static 的属性值来判断点击哪个按钮，然后通过这个值来实例化不同的预知体

七、结果演示

