

東南大學

# 毕业设计(论文)报告

题 目 人脸标记与建模技术及其整形医学

应用研究

仪器科学与工程学院 院（系） 测控技术与仪器专业

学 号 22014419

学生姓名 李英昊

指导教师 李 滢；朱利丰

起止日期 2017-11-01——2018-05-28

设计地点 中心楼

## 摘要

目前，使用计算机三维建模在教育、工业、医疗等领域已获得了普遍的应用，本文将针对整形医学应用研究三维人脸的采集重建技术。通过深度相机、摄像机阵列等传统方法获取目标的三维模型存在设备价格昂贵、对场地和环境有较高要求等不足。而利用单张图片进行三维可形变人脸模型拟合法生成的三维人脸模型由于无法获取图像中人脸的深度信息而难免存在拟合误差。另一方面，尽管整形医学对美丽人脸的研究已经得出了很多成果，但是这类研究受制于样本获取的难度，通常难以同时对大量数据进行研究。

因此，本文研究了一种通过对目标的正脸与侧脸分别进行人脸检测、特征点定位，并结合正脸与侧脸图像和特征点进行三维可形变模人脸型拟合的方法。相对于传统的可形变模型拟合，该方法通过侧脸图片中标定的特征点提供了额外的人脸 Z 轴方向的深度信息，使得三维可形变模型的拟合结果更接近于被拟合人脸的真实形状。之后，本文论述了一些拟合后的三维人脸模型如何简化整形医学研究的样本获取过程，以及三维模型可以对整形手术做出的贡献。

关键词：三维可形变模型，整形医学，人脸特征点

## Abstract

Nowadays, the usage of three-dimensional model on education, industry and medical field is rather general and receiving positive feedbacks. However, getting 3D model of an object through depth camera or camera array is expensive, demanding ideal environmental condition. On the other hand, the traditional way of fitting a 3D morphable model to a single image is impossible to achieve high accuracy for the lack of depth information.

Despite that recent research have done a lot on attractive faces of different kind of people, the study on plastic surgery is still limited in generalization due to the unavailability of large amount of samples.

Under this condition, an approach is raised in this paper to acquire a 3D model with high accuracy by detecting the faces of a frontal image and a profile of the same person, locating the face landmarks and fitting a three-dimensional morphable face model to these images simultaneously. Compared with the traditional way, this method introduces extra shape information along Z axis, which leads to a more reliable description of the truth. After that, it is showed that how the 3D model contributes to the study of plastic surgery.

**KEY WORDS:** 3D Morphable Model, Plastic Surgery, face landmarks

# 目 录

摘要.....	I
Abstract.....	II
第一章 概述.....	1
1.1 背景概述.....	1
1.1.1 机器视觉.....	1
1.1.2 机器学习.....	1
1.1.3 整形医学.....	2
1.2 主要工作.....	3
1.3 研究意义.....	4
第二章 人脸检测.....	5
2.1 原理与算法.....	5
2.2 应用与测试.....	7
2.2.1 正脸检测的代码与结果.....	7
2.2.2 侧脸检测器的训练代码及测试结果.....	8
第三章 人脸对齐.....	10
3.1 背景与原理.....	10
3.2 正脸对齐算法.....	11
3.3 侧脸对齐算法.....	11
3.3.1 模型组成.....	12
3.3.2 模型建立.....	13
3.3.3 模型拟合.....	14
3.4 应用与测试.....	14
3.4.1 正脸对齐程序及表现.....	14
3.4.2 侧脸对齐程序及表现.....	15
第四章 三维人脸重建.....	19
4.1 研究背景.....	19
4.2 模型介绍.....	19
4.3 模型拟合.....	21

4.3.1	拟合算法 .....	21
4.3.2	程序流程 .....	24
4.3.3	拟合表现 .....	25
4.4	误差分析 .....	27
4.4.1	人脸检测误差 .....	27
4.4.2	人脸对齐误差 .....	28
4.4.3	三维模型拟合误差 .....	28
第五章	三维人脸模型在整形医学上的应用 .....	29
5.1	辅助医学研究 .....	29
5.2	辅助确定整形需求 .....	31
5.3	利用三维模型预览手术结果 .....	32
第六章	分析与讨论 .....	33
6.1	开发平台及编程语言的选取 .....	33
6.2	基于神经网络的人脸标记与重建 .....	33
6.3	非线性回归 .....	34
6.4	本项目的贡献 .....	34
致 谢	.....	35
参考文献	.....	36

# 第一章 概述

## 1.1 背景概述

近年来，数据处理技术受益于电子硬件设备的更新、互联网的发展、信息的爆发式增长和算法逻辑的进步，水平得到了飞速提升。作为其中的一个重要分支，人脸识别与重构一直是各路学者关注的焦点问题之一。即使如此，目前构建一个还原度较高的三维人脸模型的普遍做法仍然需要使用复杂的扫描设备，耗费时间与财力。而当前通过单张正脸图片进行人脸三维建模的方法虽然可以很好地拟合单一视图，但由于客观已知条件的不足，拟合结果很难保证面部深度信息的准确性。使用多张普通的二维图片来进行高精度的人脸建模亟待研究。

在整形医学中，对于人脸“美观”问题的研究也十分全面，各国学者对不同民族、性别、年龄与地区的居民相貌做了详细分析，对于“有吸引力的面孔”（Attractive Faces）中五官与各部分的比例进行了深入的探讨。尽管如此，研究者们所用的实验样本十分有限，而且实验测量方法复杂，容易受误差影响。在这种情况下，我们可以将上述高精度的三维人脸模型应用在整形医学的研究中，使得样本采集过程轻而易举，在简化实验流程的同时提高精度；我们还可以设计程序与算法，使用这些整形医学的探索结果来指导整形手术的进行。

### 1.1.1 机器视觉

机器视觉（Computer Vision, CV）是一个高度交叉的学科，主要致力于研究如何从图片中获取更多更精确的信息。从工程角度来说，计算机视觉的目的就是模拟人眼的视觉系统，其主要应用有：

- 自动识别：包括物体识别（Object recognition）、物体检测（Identification and Detection）、图片信息检测（Content-based image retrieval）、人脸识别（Face recognition）、姿态判断（Pose estimation）、光学字符识别（Optical character recognition, OCR）等；
- 运动分析：包括相机位置跟踪（Egomotion）、物体跟踪（Tracking）、光流跟踪（Optical Flow）等；
- 场景还原：使用一张或多张图片对给定场景进行三维建模。该过程不需要对场景进行三维扫描（使用激光雷达或深度相机）
- 图片复原：修复图片中的畸变和噪声。

事实上，机器视觉的应用在生活中屡见不鲜，相机与手机拍照时电子取景器中在人脸部分显示的方框、火车站进站安检与解锁 iPhoneX 时使用的面部识别功能、PhotoShop 中的智能图片拼接工具等都属于机器视觉应用的范畴。

目前，机器视觉领域的相关研究已经做得十分深入，各类算法也得到了大力发展。计算机视觉与计算机视觉相关领域有很多顶尖的国际学术会议，。这些会议每年都会发布数百篇高质量的机器视觉的算法开发与应用的论文，为整个领域的发展做出的巨大的贡献。

### 1.1.2 机器学习

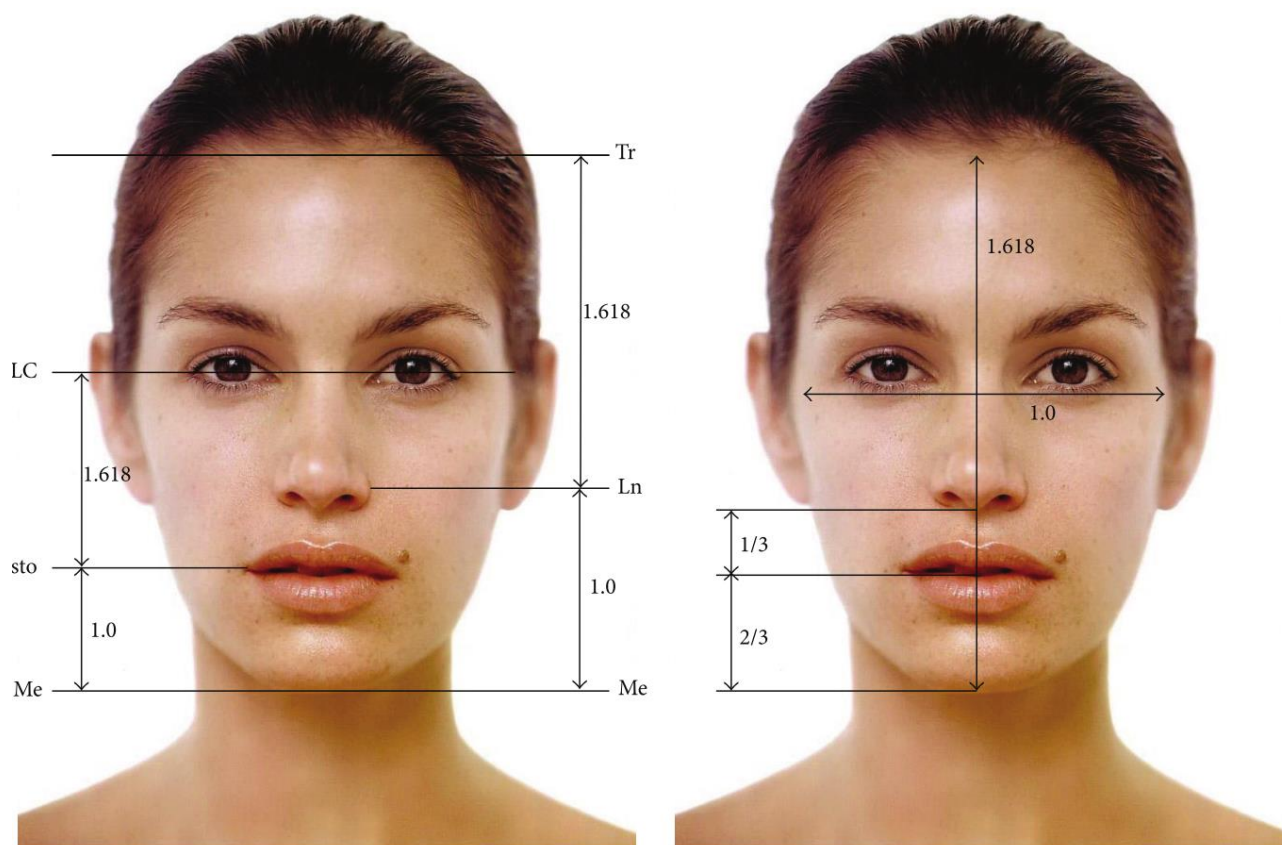
当下最火的学科必定属于计算机科学，而计算机科学中最热门的方向则毫无疑问地是机器学习（Machine Learning, ML）。在阿尔法围棋（AlphaGo，围棋英文名称为 the Game of Go）击败李世石后，全世界的目光都集中在了“人工智能”上。。

机器学习由亚瑟·塞缪尔（Arthur Samuel）在 1959 年提出<sup>[1]</sup>，主要目标是让计算机系统通过对指定数据进行“学习”，而非直接进行算法编程，来使计算机系统可以完成模式识别等功能。机器学习包括决策树学习（Decision Tree learning, DT）、规则学习（Association Rule learning, AR）、归纳逻辑编程（Inductive Logic Programming, ILP）、支持向量机（Support Vector Machines, SVMs）、聚类（Clustering）、贝叶斯网络（Bayesian Networks）、人工神经网络（Artificial Neural Networks, ANNs，又简称神经网络，Neural Networks, NNs）以及强化学习（Reinforcement Learning, RL）等等。机器学习的一个重点应用是数据分析，机器学习可以将复杂的模型与算法应用于结果预测；这些分析模型使研究人员与工程师能够给出可靠的、可重用的决策与结论，并且从对数据的历史关系和走势的分析中揭示隐含的信息；另一个重要的应用领域便是机器视觉中的模式识别。即使现在存在深度强化学习网络（Deep Q Network, DQN）算法等可以让机器通过学习，在玩一些简单的游戏时水平超过人类，我们仍然没有理由认为除了一些特定的工作外（例如通过经验来进行物种识别的专家等），机器学习算法可以大规模取代人类的脑力劳动。相反，机器学习高精度的拟合与识别结果能够让我们摆脱模糊信息的干扰，帮助我们在各个领域更方便地实现精细工作或做出可靠的预测。

### 1.1.3 整形医学

整形手术包含对人体构造的复原、重建或改变。它包含两个部分，分别为包括颅面手术、手部手术、显微手术和烧伤医疗的用以重构身体的一部分外观或机能的重建手术（Reconstructive Surgery）以及以改善外观为目的的美容手术（Cosmetic Surgery or Aesthetic Surgery）。在 1980 年代，只有约 25% 的患者去做美容手术，而这一数字在 2006 年增长为了 75%。本文将主要涉及美容手术的话题。

美容手术基于人类对于面部美学的普遍认可。面部美学的基础是对于面部轮廓类型的分类以及大众普遍接受的吸引力标准。对理想面部比例的定量考据是美容手术取得理想效果的关键<sup>[2-6]</sup>。然而，人脸平均轮廓形状与面部特征因人种、性别和地理区域的不同而产生差异。尽管存在一个定义理想人脸的黄金分割比例，如图 1-1，找到一个通用的标准来评判面部是否美观仍然是一件十分困难的事情。

图 1-1 人脸的理想比例<sup>[6]</sup>

## 1.2 主要工作

本文首先对上述计算机视觉与机器学习算法进行了比较，选择出合适的方法分别对包含正面人脸与接近正面的人脸的照片和侧脸照片进行物体检测，判断照片中是否包含正脸图像或者侧脸图像。如果有，以边界框（Bounding Box, bbox）的形式确定面孔图像的位置与大小；其次，对于已经确定了边界框的正脸或侧脸图像分别进行特征点拟合。

其次是本项目的主要工作，本文改进了已有的三维可形变模型（3D Morphable Model, 3DMM）拟合算法，设计并编写了自己的 3DMM 拟合程序，使得该模型能够同时对正脸与侧脸的特征点进行拟合，生成的结果更为精确。

最后，对于输入样本，我们分别采集他的正脸与侧脸图像，使用上述方法获得正脸与侧脸的特征点并进行三维重建，生成其面部三维网格模型后对该模型进行美学分析，提出整形医学方面的建议。具体流程如图 1-2 所示。

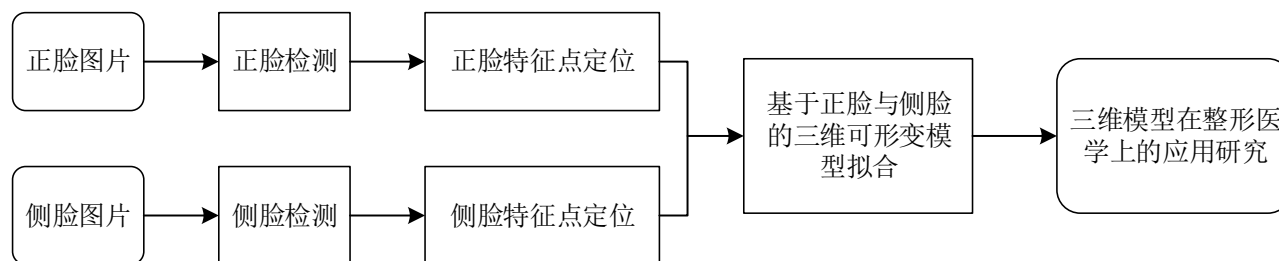


图 1-2 本项目的主要流程



论文内容一共分为六个部分，本部分为绪论，主要介绍本项目的技术背景、主要工作和研究意义。

第二部分将介绍本文使用的人脸检测算法，并通过代码介绍和示例展示来验证算法的可靠性；

第三部分将比较几种特征点检测（又称特征点定位、特征点回归，**Landmark Location**）算法，阐述选择本文使用的算法的原因，并通过代码介绍和示例对比来验证所选择算法的可靠性；

第四部分将详细讲述使用两种视图进行人脸三维重建的算法流程，并比较本文改进后的算法与原有算法在同一数据集上的表现；

第五部分将通过几个示例介绍重建出来的三维人脸在整形医学上的应用；

第六部分将分析本文方法存在的缺陷，拟定可行的改进方案，对本项目的工作进行总结，并对该研究方向进行进一步展望。

### 1.3 研究意义

尽管计算机视觉和机器学习已经得到了长足的发展，但是这些算法还远非完善，存在这很大的提升空间。同时，它们的应用还没有普及，有更多领域的研究需要这些先进技术的辅助。

在整形医学的实验中，研究者使用的方法大多是直接测量人脸各部分距离，采集人脸数据后再手工通过软件标注人脸特征点并计算距离，或者通过在被采集者面部粘贴可视点之后，再用计算机识别的方式获取特征点<sup>[2-6]</sup>。这些方法费时费力，还容易造成精度损失。不仅如此，在各个诊所，需要进行重建手术的病人的手术建议也完全由医生开具，难免会存在主观判断的情况。

本项目除了给出了一种精度更高，而且算法复杂度和硬件开销没有显著增加的三维人脸拟合方法以外，还能够通过拟合出来的人脸自动给出具有参考价值的医学建议，拓展了计算机视觉和机器学习领域的应用思路。

## 第二章 人脸检测

人脸检测是目标检测 (Object Detection) 的一个特例, 目标是获取图片中所有人脸目标的位置与大小。值得注意的是, 与人脸识别 (Face Recognition) 不同, 人脸检测 (Face Detection) 并不关注被检测对象的身份信息。传统的人脸检测致力于发现正面人脸 (Frontal Human Faces), 设计的程序大多数也只能检测出照片中的正脸。由于本项目的特殊性, 侧脸 (Profiles) 的检测应该受到同样的重视, 因此, 我们通过自己标注的侧脸数据集, 训练了自己的侧脸检测器。

### 2.1 原理与算法

经过近二十年的发展, 有许多算法被设计出来用于检测人脸, 从原始的基于颜色的算法到目前最为先进的神经网络算法, 不一而足。随着更多的如局部二值模式 (Local Binary Patterns, LBP)<sup>[7]</sup>、尺度不变特征转换 (Scale Invariant Feature Transform, SIFT)<sup>[8]</sup>和方向梯度直方图 (Histogram of Oriented Gradient, HOG)<sup>[9]</sup>等稳定性较好的特征提取方法的提出、如带标注的自然状态人脸数据库 (Labeled Faces in the Wild Home, LFW)<sup>[10]</sup>等大规模带有标注的开放图像数据库的建立和支持向量机、深度学习 (Deep Learning, DL) 等机器学习方法的引入, 人脸检测的精度目前已经十分优异, 在图片模糊的条件下甚至可以超过人类的检测水平。常用的人脸检测算法包括自适应增强人脸检测算法 (Adaptive Boosting, Adaboost)<sup>[11]</sup>、基于特征的方法<sup>[12]</sup>、基于模板的方法和基于统计理论的方法。

基于统计理论的方法是目目前高精度人脸检测最为常用的方法, 主要包括神经网络方法、支持向量机方法<sup>[15]</sup>和隐马尔可夫模型 (Hidden Markov Model, HMM)<sup>[14]</sup>方法。然而, 这种方法需要大量的统计学特性, 即需要大量带有标注的样本作为训练数据, 而且神经网络方法还需要计算机有较快的运算速度与较大的内存容量。在最近几年数据量呈爆发式增长之前该条件难以满足, 因此尽管理论提出较早, 该方法在近几年才变得流行。

本项目主要使用 Dlib 库<sup>[16]</sup>中的人脸检测与物体检测算法进行人脸检测。该算法结合了经典的方向梯度直方图特征 (HOG Feature)、线性分类器 (Linear Classifier)、图像金字塔 (Image Pyramid) 和滑动窗检测方案 (Sliding Window Detection Scheme), 不仅适用于人脸检测, 还适用于大多数的形状不会剧烈变化的半刚性物体 (Semi-rigid Objects)。由于 Dlib 库中预训练的正脸检测模型<sup>1</sup>只适用于人脸朝向角度 (Yaw, 通常译作航向角) 较小 (大概在  $-60^{\circ}$  ~  $60^{\circ}$ ) 时进行人脸检测, 而本项目需要识别出侧脸 ( $-100^{\circ} < \text{Yaw} < 100^{\circ}$ ), 故在侧脸照片中将使用自己训练的检测器进行侧脸检测。Dlib 库中的算法使得两种检测共用一种特征, 可以加快程序的运行速度。

HOG 描述器具体的实现方法是: 首先, 将图像分成由几个相邻像素组成的小区域——细胞单元 (cell), 然后计算细胞单元中各像素的梯度的直方图; 然后, 在几个相邻细胞单元组成的区域——区块 (block) 内计算这些直方图的密度, 据其对细胞单元进行对比度归一化; 最后, 把这些直方图重新组合成图像, 构成图像特征描述器 (Window Descriptor)。该描述器

---

<sup>1</sup>见 [https://github.com/davisking/Dlib/blob/master/Dlib/image\\_processing/frontal\\_face\\_detector.h](https://github.com/davisking/Dlib/blob/master/Dlib/image_processing/frontal_face_detector.h)

能很好地免疫亮度与对比度的变化和几何图形的形变。获得 HOG 特征的流程如图 2-1，具体文字介绍请参阅论文<sup>[9]</sup>。

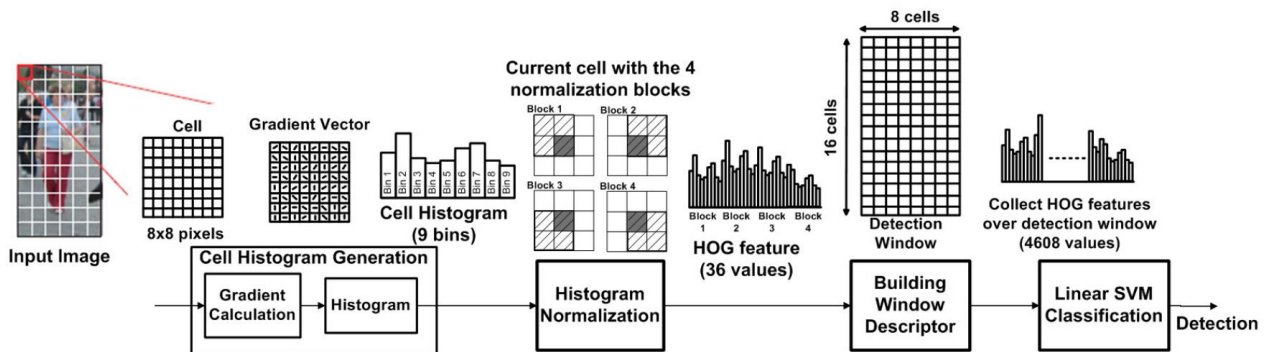


图 2-1 使用 HOG 特征的物体检测算法<sup>[9]</sup>

线性分类器通过对特征向量 (Eigenvector) 的线性组合与聚类来做出分类，是生活中将具有相似特征的物体进行分类的常用分类手段。目前，在计算机图形学中 HOG 特征检测加上支持向量机 (SVM)<sup>[17]</sup> 分类是人脸检测的一个主要研究思路。支持向量机可以通过一个核函数将原始的样本特征变换到另一特征空间，使原来不线性可分的样本变得线性可分，应用于图像分类时能够显著提高检测准确度。卷积神经网络 (Convolution Neural Network, CNN)<sup>[18]</sup> 也是一个很好的图像特征分类器。不幸的是，在训练样本较少的情况下，卷积神经网络在测试集上表现效果一般。

图像金字塔是图像预处理的一种方式。它通过将图像进行一层一层的下采样或上采样来对原始图像创建多种分辨率的副本，以此构建图像的多尺度，使得检测器能够检测出不同尺寸的人脸图像。Dlib 中图像金字塔的以  $5/6$  进行下采样，即不断将原图像缩小为原来的  $5/6$ ，并用过程中生成的图像构成多级图像金字塔。

滑动窗则是使用一个固定大小的窗口 (Dlib 中默认为  $80 \times 80$  像素) 对图像中的像素进行遍历，使用上述 HOG 特征与 SVM 分类器判断每一个窗口中的内容是否是人脸。当图像大小小于滑动窗的大小时，停止对图像进行下采样。人脸检测的算法流程如图 2-2 所示。

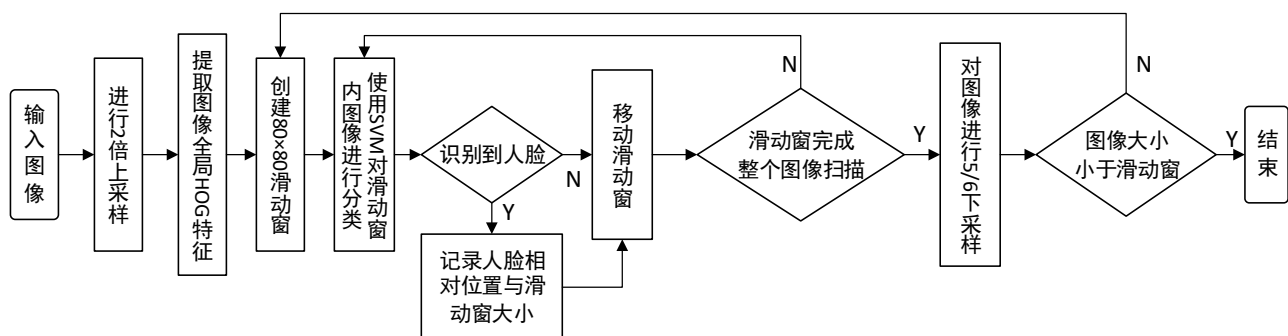


图 2-2 Dlib 人脸检测算法流程

侧脸检测模型的训练过程与检测过程与上述流程基本一致，训练时只要将给出的边界框内的 HOG 特征 (作为正特征) 与其他部分的 HOG 特征 (作为负特征) 输入 SVM 训练，使得生成的模型能够分辨正负人脸特征即可。

## 2.2 应用与测试

本项目的侧脸检测部分使用的是 Menpo 项目<sup>[20]</sup>封装的 Dlib 检测器构造函数与检测函数。Menpo 中的 Dlib 检测器训练函数和构造函数使用的算法与源代码与 Dlib C++库完全一致；不同的只是 Menpo 将算法流程语句封装为了一个函数，将滑动窗大小、SVM 正则化参数 C、SVM 训练停止条件  $\varepsilon$  等所有训练选项作为参数传递进去，而不是像 Dlib 一样需要在程序中通过改变类的成员变量进行设置。除此之外，Menpo 库还提供了基于 Jupyter Notebook<sup>[21]</sup>与 Matplotlib<sup>[22]</sup>的良好的图片显示与交互界面，可以方便的展示训练结果。

正脸检测部分实际上被直接基于 Dlib 库的 Python 代码实现。但是，为了方便展示，下面仍然使用 Menpo 提供的函数做展示。再次强调，这里使用的 Menpo 与 Dlib 在算法上完全一致，二者的检测结果理论上也不存在区别。

### 2.2.1 正脸检测的代码与结果

首先载入测试图片。本项目中使用的正脸测试图片来自 LFPW<sup>[23]</sup>测试集。LFPW 数据集一共包含 1432 个从网络上下载的自然状态下的人脸数据，同时，每个人脸都被以不同的方式标注了特征点，是进行人脸检测以及特征点检测模型测试的理想数据集。

首先载入数据集：

```
from pathlib import Path
import menpo.io as mio
from menpo.visualize import print_progress

path_to_lfpw = Path(r'F:\PythonFiles\FaceLandmarks\DataSets\lfpw\testset')
images = mio.import_images(path_to_lfpw)
images_list = []
for img in print_progress(images):
    images_list.append(img)
```

接着，载入 Dlib 正脸检测器，并对测试样本中的每一张图片进行人脸检测。检测出来的边界框参数被直接添加到图片 Image 类的成员中。

```
from menpodetect import load_dlib_frontal_face_detector
detect = load_dlib_frontal_face_detector()
for image in images_list:
    detect(image)
```

最后，我们通过下面的代码显示随机六张带有边界框的图片，观察 Dlib 检测器的效果。结果如图 2-3 所示。

```
%matplotlib inline
import matplotlib.pyplot as plt
plt.figure(1)
for i in range(2):
    for j in range(4):
        plt.subplot(2, 4, i * 4 + j + 1)
        images_list[i * 4 + j].view_landmarks(group='dlib_0', line_colour='red', render_markers=False, line_width=4)
plt.show()
```

值得注意的是，Menpo 可以通过下面的代码实现带有交互的图片显示功能，使用者能够通过滑动条、复选框和下拉菜单来控制图片的通道选择、图片缩放、边界框的渲染方式等。由于文章中无法体现上述功能，因此在这里不做展示。

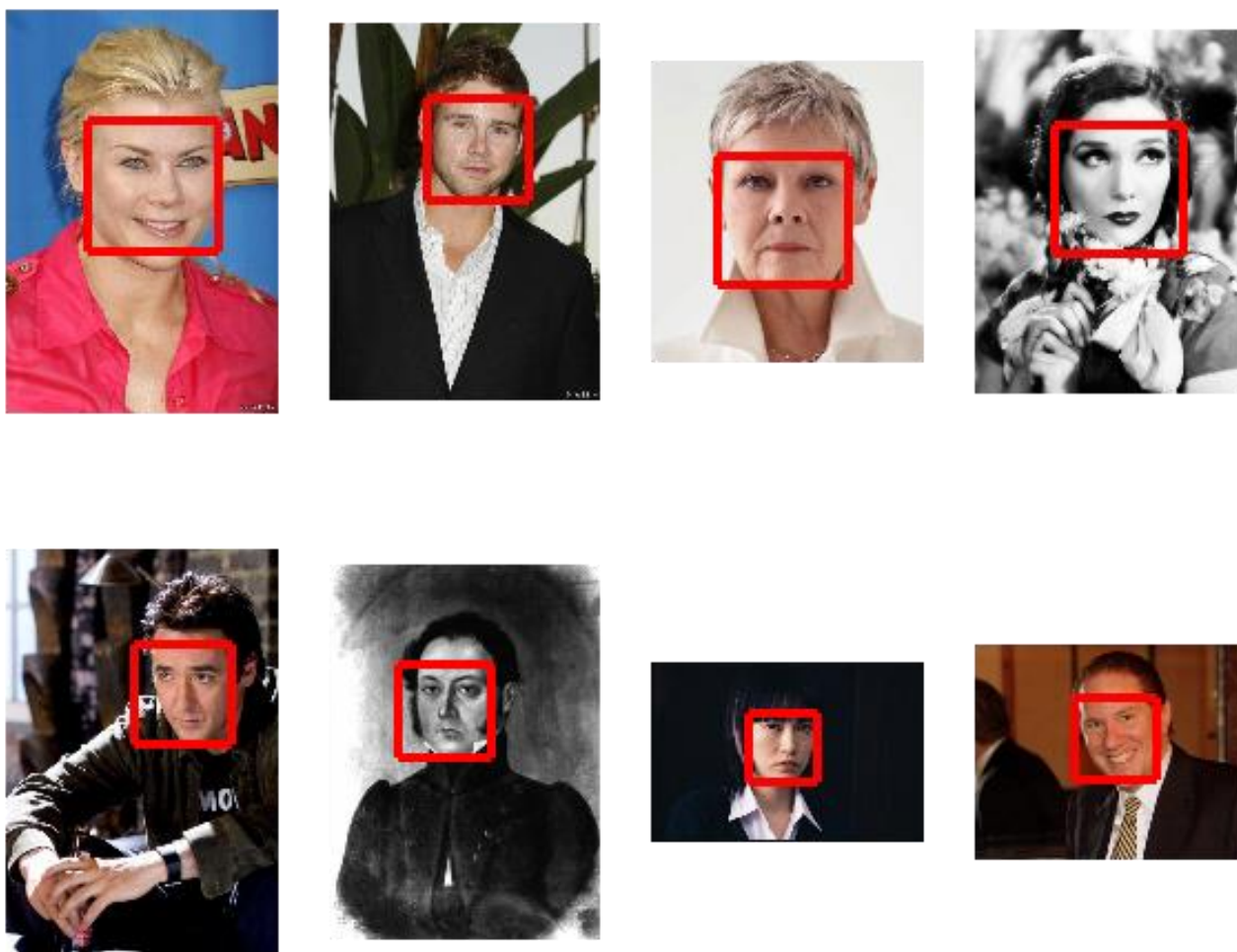


图 2-3 Dlib 正脸检测器的输出结果（红色方框）

### 2.2.2 侧脸检测器的训练代码及测试结果

侧脸检测器的训练原理上面已经详细说明了，请参考 2.1.2 节。由于目前网络上还没有大规模公开的侧脸数据集，作者自行在 FERET<sup>[24]</sup>数据集的侧脸照片（文件名带有“pr”字眼）集中随机标注了 99 张照片作为训练集训练侧脸检测模型，将剩余 1125 张照片作为测试集测试模型的表现。FERET 数据集中包含有大量同一个人不同角度的照片，是本项目重要的测试数据来源，在第三章、第四章中均会涉及。

首先载入训练集：

```
from pathlib import Path
import menpo.io as mio
from menpo.visualize import print_progress
path_to_images = r'F:\PythonFiles\FaceLandmarks\DataSets\FERET\FERET_pr_train'
train_images = mio.import_images(path_to_images)
training_images = []
for img in print_progress(train_images):
    training_images.append(img)
```

[=====] 100% (99/99) - done.

接下来训练侧脸检测模型。注意，在载入训练集时，训练集的边界框自动以点云的形式存储在类成员里，因此不用再次显示调用。

```
from menpodetect.dlib import train_dlib_detector
in_memory_detector = train_dlib_detector(training_images, verbose_stdout=True)
```



上面的代码中并没有指明训练参数，因此训练参数将取默认值。默认值如下：2 倍上采样；图像金字塔下采样比例为 5/6； $\varepsilon=0.01$ ； $C=5$ ；滑动窗大小为  $80\times 80$  像素。

最后，使用训练好的侧脸检测模型构建检测器，载入训练集并检测：

```
from menpodetect.dlib import DlibDetector
profile_detector = DlibDetector(in_memory_detector)

test_images = mio.import_images(r'F:\PythonFiles\FaceLandmarks\DataSets\FERET\FERET_pr_test')
testing_images = []
for img in test_images:
    testing_images.append(img)

for img in testing_images:
    profile_detector(img)
```

随机选取测试集中的八张图片进行侧脸检测，检测效果如图 2-4 所示：

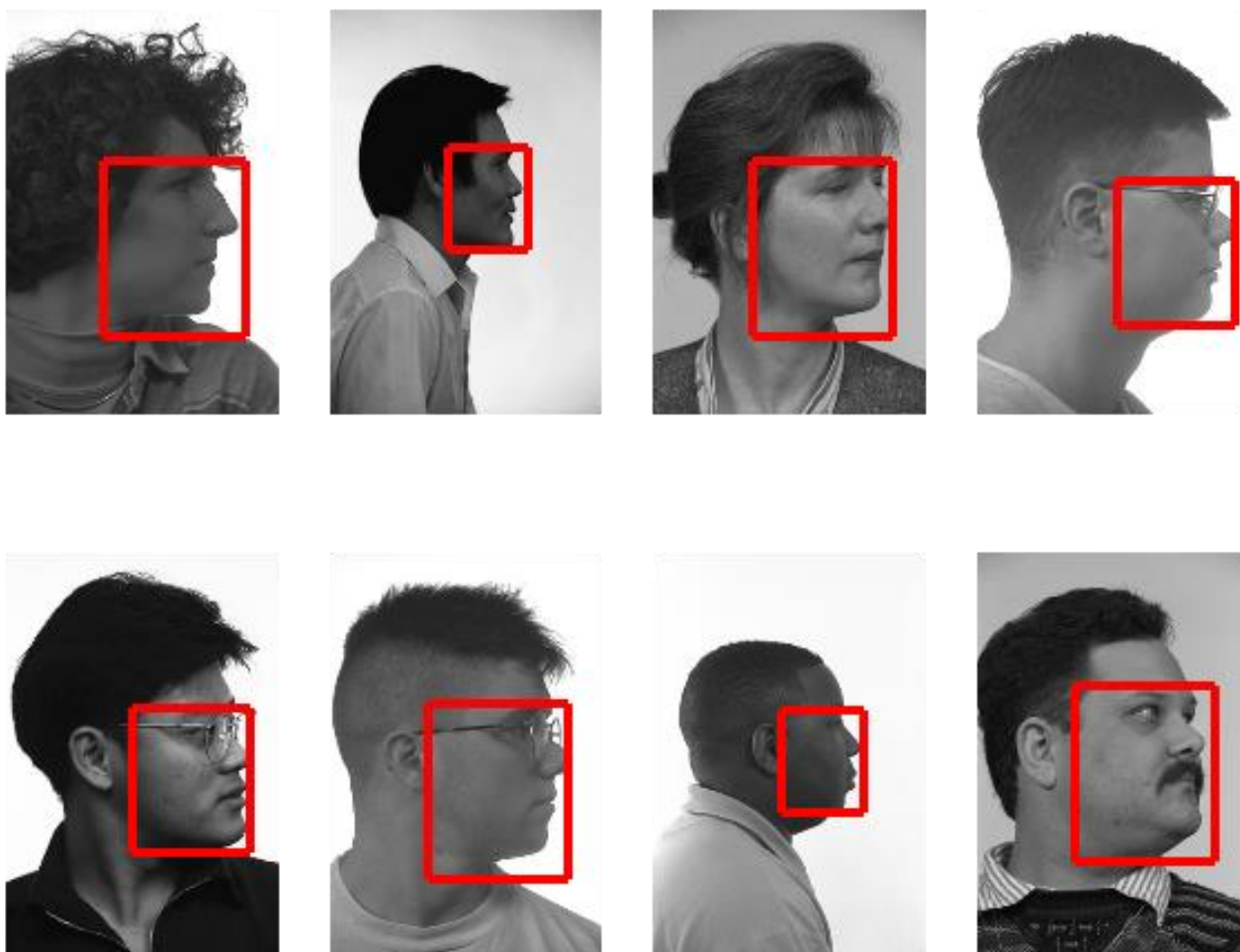


图 2-4 训练出的侧脸检测器的检测结果

可以看出，即使在训练集较小的情况下，该算法仍然能够较为准确地检测出图像中的侧脸信息，能够很好地完成本项目的要求。

## 第三章 人脸对齐

### 3.1 背景与原理

人脸对齐 (Face Alignment), 又称人脸特征点检测 (Facial Landmark Detection), 是机器视觉领域又一重要的研究方向, 目的是在已知人脸边界框的情况下 (人脸边界框可以由上一章中的方法获得), 通过一些算法与模型尽可能精确地判断出人脸的一些特征点的位置。目前, 网络上存在大量带标注的人脸数据集, 这些数据集可以被分为两种类型: 在受控条件 (Controlled Condition) 下创建的数据集, 如 Multi-PIE<sup>[25]</sup>, XM2VTS<sup>[26]</sup>, FRGC-V2<sup>[27]</sup>, 与 AR<sup>[28]</sup> 等, 与在自然状态下 (in-the-wild) 生成的数据集, 如 LFPW<sup>[29]</sup>, HELEN<sup>[30]</sup>, AFW<sup>[31]</sup>, AFLW<sup>[32]</sup>, IBUG<sup>[33]</sup>, 300-W<sup>[36]</sup> 等。这些数据集都包含了诸如光照、背景、姿态、表情与遮挡状态等条件的变化, 十分适于人脸对齐的研究。关于这些数据库内容的详细信息与使用许可, 请参阅相关论文。每个数据库中提供的人脸特征点标注主要为正脸特征点标注, 而且不完全一致, 具体如图 3-1 所示; 在本项目中使用的正脸标注为 Multi-PIE/IBUG 68 点标注方式。在目前公开的数据集中并没有对侧脸特征点的标注。因此, 本文采用的侧脸标注为笔者自行根据正脸特征点标注与经验拟定的侧脸 26 特征点标注方式, 如图 3-2 所示。其它侧脸标注方式可参阅<sup>[34]</sup>, 但是其标注文件数据并未公开。

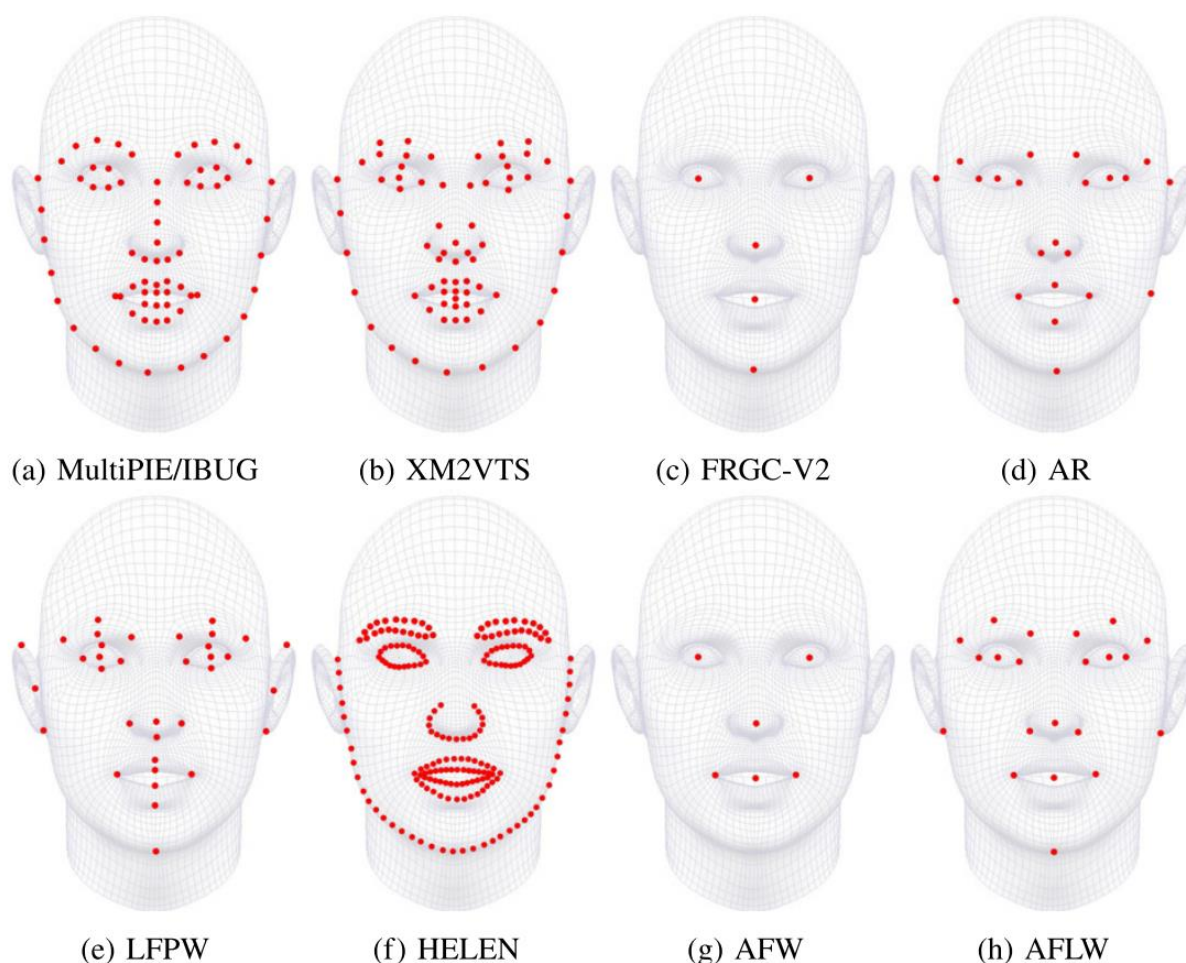


图 3-1 一些带标注的数据集的特征点标注方式<sup>[35]</sup>

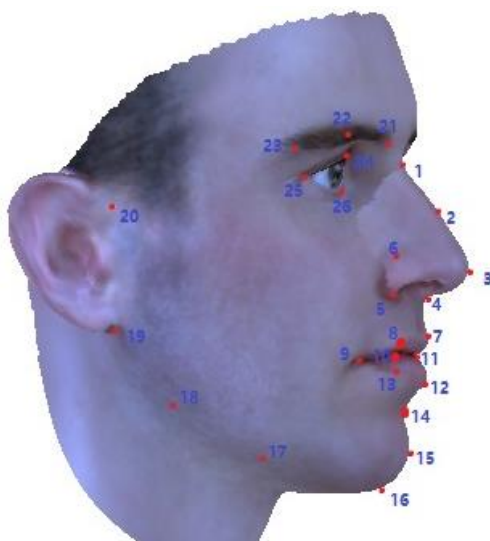


图 3-2 自行拟定的侧脸特征点标注方式（从序号“1”开始）

自 1995 年以来，已经有许多在精度与算法效率的表现上都十分出色的算法被开发出来。这些算法可以大致被分为四类：基于主动形状模型（Active Shape Model, ASM）<sup>[37]</sup>与主动外观模型（Active Appearance Model, AAM）<sup>[38]</sup>的传统主动模型方法、基于受限局部模型（Constrained Local Model, CLM）<sup>[39]</sup>的方法、基于级联形状回归（Cascade Pose Regression, CPR）<sup>[40]</sup>的方法与其他方法，例如基于神经网络与深度学习的特征点检测，包括深度卷积神经网络（Deep Convolutional Neural Network, DCNN）<sup>[41]</sup>、多任务级联卷积神经网络（Multi-task Cascaded Convolutional Neural Network, MTCNN）<sup>[42]</sup>等、深度对齐网络（Deep Alignment Networks, DAN）<sup>[43]</sup>等。目前的人脸对齐已经可以取得相当高的精度，甚至有分析指出，人脸对齐算法在现有的数据集上已经接近性能的饱和。由于人脸对齐算法繁多复杂，限于文章篇幅与作者水平不再对所有方法一一介绍，而重点讲述在本项目中应用的两种对齐算法。

### 3.2 正脸对齐算法

本文中正脸的人脸对齐直接使用 Dlib 库中的函数预训练的正脸特征点检测模型实现。该函数基于聚合回归树（Ensemble of Regression Trees, ERT）算法，应用聚合回归树直接从稀疏像素强度子集中估算人脸特征点坐标，速度极快（单独人脸特征点检测耗时约 1ms）的同时估算结果十分精确。具体算法流程请参考论文<sup>[44]</sup>。

### 3.3 侧脸对齐算法

不同与正脸对齐，本文中的侧脸对齐使用的是主动外观模型（AAM），由主动轮廓模型（Active Contour Model, Snake）与主动形状模型（ASM）发展而来，是 ASM 的扩展形式，使用覆盖目标区域的所有的图像区域信息，而不是如 ASM 一样仅仅使用边界附近的信息。AAM 是可形变对象类（Deformable Object Class）的形状和外观的统计学可形变模型（Statistical Deformable Model）。它是一个生成模型（Generative Model），在拟合过程中旨在通过一些优化方法来恢复某个对象的参数描述。AAM 是人脸对齐的经典算法，能够应对大多数如光照等环境条件的变化。



### 3.3.1 模型组成

若要建立一个 AAM 模型，需要定义一个可形变模型的形状实例（shape instance） $\mathbf{s} = [x_1, y_1, \dots, x_L, y_L]^T$  为一个  $2L \times 1$  的向量，包含  $L$  个特征点，它们的坐标为  $(x_i, y_i), \forall i = 1, \dots, L$ ，训练集一共包含了分别标注了  $L$  个特征点  $N$  张图片  $\{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_N\}$ 。完整的 AAM 模型包含以下部分形状模型（Shape Model）、外观模型（Appearance Model）和表情模型（Motion Model）三个部分<sup>[48]</sup>：

形状模型的建立首先需要把训练集的形状实例（shape instance） $\{s_1, s_2, \dots, s_N\}$  使用广义普鲁克分析（Generalized Procrustes Analysis, GPA）<sup>[45]</sup> 进行对齐（align），并应用主成分分析（Principal Component Analysis, PCA）<sup>[46,54]</sup> 方法提取出训练集的正交基底（orthonormal basis）。接着，使用四组表示相似变换（similarity transform）的特征向量（eigenvector），分别为比例参数（scaling）、平面内的旋转参数（in-plane rotation）与平移参数（translation），对训练形状进行增强，得到  $\{\bar{\mathbf{s}}, \mathbf{U}_s\}$ 。其中， $\mathbf{U}_s \in \mathbb{R}^{2L \times n}$  为  $n$  个特征向量（包括四组相似变换元素）的正交基底， $\bar{\mathbf{s}} \in \mathbb{R}^{2L \times 1}$  为形状向量的均值。至此，我们可以得到以均值和特征向量表示的形状实例：

$$\mathbf{s}_p = \bar{\mathbf{s}} + \mathbf{U}_s \mathbf{p} \quad (1)$$

其中  $\mathbf{p} = [p_1, p_2, \dots, p_n]^T$ ，为形状参数向量。

表情模型中包含一个必不可少的形变函数（warp function） $\mathbf{W}(\mathbf{p})$ ，用于将有关于由参数  $\mathbf{p}$  生成的形状实例中的纹理变形为通用的参考形状（reference shape）。参考形状默认问平均形状  $\bar{\mathbf{s}}$ ，也可以在创建 AAM 时自己指定。

要训练外观模型，首先需要将特征方程  $\mathbf{F}()$  作用在所有训练集图像中，如  $\mathbf{F}(\mathbf{I}_i), \forall i = 1, 2, \dots, N$ ，提取出 HOG、SIFT 等图像特征；接着，将基于特征的图片变形至参考形状并向量化，得到：

$$\mathbf{a}_i = \mathbf{F}(\mathbf{I}_i)(\mathbf{W}(\mathbf{p}_i)), \forall i = 1, 2, \dots, N \quad (2)$$

其中， $\mathbf{a}_i \in \mathbb{R}^{M \times 1}$ 。

最后，对  $\mathbf{a}_i$  进行主成分分析，得到  $\{\bar{\mathbf{a}}, \mathbf{U}_a\}$ 。其中， $\mathbf{U}_a \in \mathbb{R}^{M \times m}$  为  $m$  个特征向量组成的正交基底， $\bar{\mathbf{a}} \in \mathbb{R}^{M \times 1}$  为平均外观向量。至此，我们用下面的公式来生成符合高斯过程的外观实例：

$$\mathbf{a}_c = \bar{\mathbf{a}} + \mathbf{U}_a \mathbf{c} \quad (3)$$

其中， $\mathbf{c} = [c_1, c_2, \dots, c_m]^T$  为外观参数向量。

建立外观模型时，我们定义

$$\mathbf{t}(\mathbf{W}(\mathbf{p})) = \mathbf{F}(\mathbf{I})(\mathbf{W}(\mathbf{p})) \quad (4)$$

为通过参数  $\mathbf{p}$  给定形状实例的图片  $\mathbf{I}$  的基于特征（feature-based）的  $M \times 1$  形变向量。本项目中使用的模型为 AAM 模型中的基于局部的主动外观模型（Patch-based AAM, PatchAAM）<sup>[47]</sup> 版本。与全局主动外观模型（Holistic AAM）不同，该模型的形变函数  $\mathbf{t}(\mathbf{W}(\mathbf{p}))$  只提取由参数  $\mathbf{p}$  生成的形状实例中的每一个特征点周围的矩形区域的图像特征，而非特征点三角形包围范围内的全局特征。

### 3.3.2 模型建立

公式(1)与公式(3)描述了 AAM 的形状与外观的变化。

当给定一个图片数量为  $N$ ，每幅图片的特征点数量为  $L$  的训练集后， $\bar{\mathbf{s}}$  可以通过将图像中的特征点对齐（将特征点通过仿射变换移至参考模型的位置），并对每一个特征点的位置  $(\mathbf{x}_i, \mathbf{y}_i), \forall i=1, \dots, N$ ，其中  $\mathbf{x}_i = [x_i^1, x_i^2, \dots, x_i^L]^T$ ， $\mathbf{y}_i = [y_i^1, y_i^2, \dots, y_i^L]^T$  求取平均值得到。 $\mathbf{U}_s$  可以通过对训练形状  $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_L\}$  与平均形状  $\bar{\mathbf{s}}$  的差进行 PCA 操作后，取对应特征值最大的前  $n$  个特征向量得到。 $n$  的大小可以自己定义，不过，通常特征向量对应的特征值的能量总和占全部能量的 90% 以上。调整形状参数向量  $\mathbf{p}$ ，相当于调整不同的形状特征向量的表达权重，可得到不同的形状实例，如图 3-3 所示。

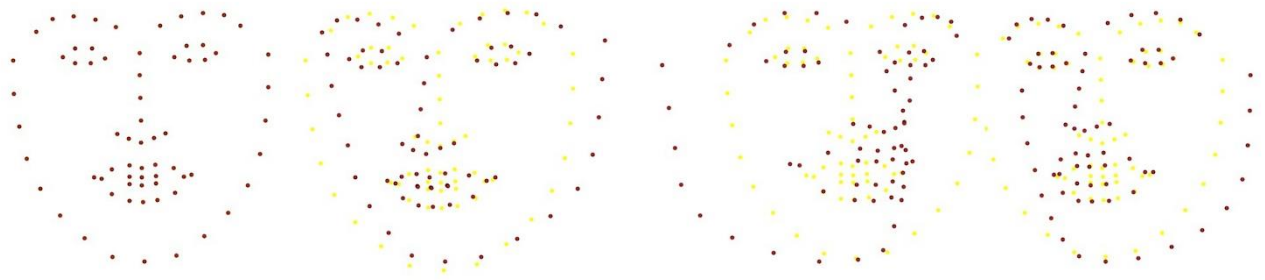


图 3-3 形状模型实例。最左边为平均形状，其他三个为在  $[-3, 3]$  范围内随机生成前五个特征向量权重得到的实例（黄色部分为平均形状）。

外观模型中  $\bar{\mathbf{a}}$  与  $\mathbf{U}_a$  的计算过程与  $\bar{\mathbf{s}}$  与  $\mathbf{U}_s$  的计算基本一致。不过，需要注意的是，不同于形状模型计算中对特征点的坐标进行操作，PatchAAM 外观模型的计算对象是以每特征点为中心的周围  $X \times Y$  大小的矩形区域中的图像特征。调整外观参数向量  $\mathbf{c}$ ，相当于调整不同的外观特征向量的表达权重，即可得到不同的外观实例，如图 3-4 所示。

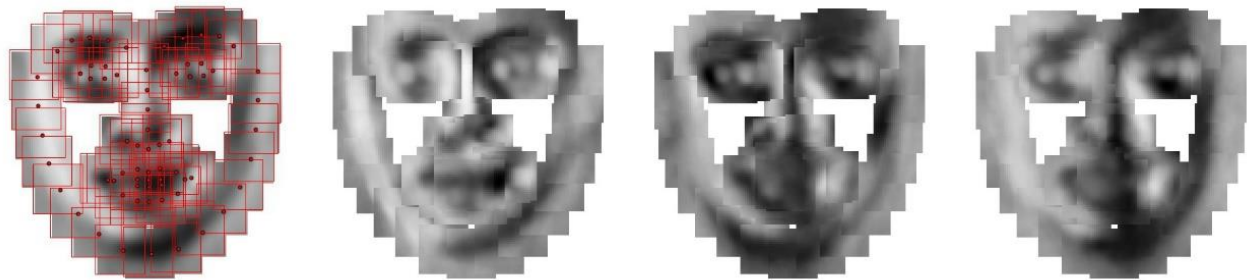


图 3-4 外观模型实例。从左到右依次为：平均外观（包括特征点与边界轮廓）； $\mathbf{c} = [1, 1, 1, 1, 1]$ ； $\mathbf{c} = [1.5, 1.5, 1.5, 1.5, 1.5]$ ； $\mathbf{c} = [1, 1, 0, -1, -1]$ 。

PatchAAM 的形变函数相对于全局主动外观模型也更简单。对于全局主动外观模型，形变函数  $\mathbf{W}(\mathbf{p})$  需要计算特征点构成的所有三角形从原形状  $\mathbf{s}_0$  到目标形状  $\mathbf{s}_p$  的仿射变换（Affine Transformation）；对于基于局部的主动外观模型来说，这样的操作就可以省略了：因为无论是变换前后，我们感兴趣的总是一个特征点周围的图像特征，而非全局的图像特征。这种特性也导致了 PatchAAM 在每个特征点的相对位置方面的稳定性略逊于全局主动外观模型。

### 3.3.3 模型拟合

主动外观模型的拟合，实际上就是找到  $\mathbf{p}$  和  $\mathbf{c}$  的值，使得由  $\mathbf{p}$  和  $\mathbf{c}$  为参数向量生成的模型实例中的图像特征与被拟合图片中的图像特征的残差最小。以方程形式表示即为：

$$\arg \min_{\mathbf{p}, \mathbf{c}} \|\mathbf{t}(\mathbf{W}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{U}_a \mathbf{c}\|^2 \quad (5)$$

本文中主要使用的优化方法为卢卡斯-卡纳德优化器（Lucas-Kanade Optimization）。它本质上也是梯度下降算法<sup>[51]</sup>的一种，能够解决光流方程的内在模糊性的问题，并且具有对噪声不敏感的优点。本文将不再对该优化器进行讨论，如有需要，请参阅维基百科或论文<sup>[49-50]</sup>。

## 3.4 应用与测试

### 3.4.1 正脸对齐程序及表现

本项目中正脸对齐部分直接采用 Dlib 库中的 Python 函数接口与预训练的模型实现。Dlib 库提供的函数由 C++代码实现，具有优秀的效率与良好的精度，开放源码且使用方便，在如 Massachusetts Institute of Technology Information Extraction toolkit、Lawrence Livermore National Laboratory's ROSE project、OpenFace 等项目中均有使用。

首先，使用官方模型构造人脸检测器 detector 与特征提取器（特征点定位函数）predictor。其中，调用 Dlib.get\_frontal\_face\_detector()函数即可加载默认的人脸检测模型，不用显示传入。具体请参考 2.1.2 节。

```
detector = dlib.get_frontal_face_detector()
predictor_path = r"\\path\\to\\face_landmarks\\shape_predictor_68_face_landmarks.dat"
predictor = dlib.shape_predictor(predictor_path)
```

测试集仍然采用 LFPW 测试集。读取测试集中所有 png 图像的文件名：

```
img_path = r'F:\PythonFiles\FaceLandmarks\DataSets\lfpw\testset'
img_names = []
for _,_,file_names in os.walk(img_path):
    for file_name in file_names:
        if file_name.endswith('.png'):
            img_names.append(file_name)
```

随机载入六张测试图片：

```
imgs = []
for i in range(6):
    img = cv2.imread(os.path.join(img_path, img_names[np.random.randint(224)]))
    imgs.append(img)
```

对于每一张图片，先使用人脸检测器检测人脸的边界框，以便在特征点回归过程中确定特征点的初始位置。接着，使用特征提取器进行正脸特征点回归，得到每一个特征点的预测坐标值。为了方便展示结果，我在图像中以红色圆点标出了输出的特征点的位置。

```
for img in imgs:
    dets = detector(img, 1)
    for bbox in dets:
        points = predictor(img, bbox)
        marks = np.array([[p.x, p.y] for p in points.parts()], copy=False, dtype=np.int16)
        for mark in marks:
            cv2.circle(img, tuple(mark.tolist()), 3, (0,0,255), -1)
```

图 3-5 展示了正脸对齐的结果。可以看出，Dlib 特征点检测函数可以在不同光照条件、姿态、表情与图片大小的条件下都有良好的表现。



图 3-5 随机对六幅图片进行正脸特征点检测的结果

### 3.4.2 侧脸对齐程序及表现

本项目中侧脸对齐部分应用了 Menpo 项目提供的 AAM 模型训练与特征点检测函数进行图片基于局部特征的主动外观模型的构建。该函数的算法请参考 3.3 节。

首先载入训练集。与 2.2.2 节相同，我们使用的训练集的是 FERET 数据集中经过自己手工标注的 99 张侧脸图片。特征点数据以点云的形式保存在与图像同名的 pts 文件中。载入训练集时，Menpo 提供的函数会自动载入点云并将它与图片关联。对于每张训练图像，我们只保留了图像中特征点最小边界框内及外面 20% 的图像，并且通过把图片转换成灰度图，以此将原来的三个或四个通道（channel）（RGB 通道或者 RGBa 通道）变成一个通道，显著降低了训练时对内存的占用。一些转换后的训练图片和对应的特征点如图 3-6 所示。可以看出，侧脸图像上特征点可以被算法标记，尽管有一些特征点标注存在少量误差。

```
path_to_images = r'F:\PythonFiles\FaceLandmarks\DataSets\FERET\FERET_pr_train'
training_images = []
for img in print_progress(mio.import_images(path_to_images, verbose=True)):
    if img.n_channels == 3:
        img = img.as_greyscale()
    img = img.crop_to_landmarks_proportion(0.2)
    d = img.diagonal()
    if d > 400:
        img = img.rescale(400.0 / d)
    training_images.append(img)
```

```
Found 99 assets, index the returned LazyList to import.
[=====] 100% (99/99) - done.
```



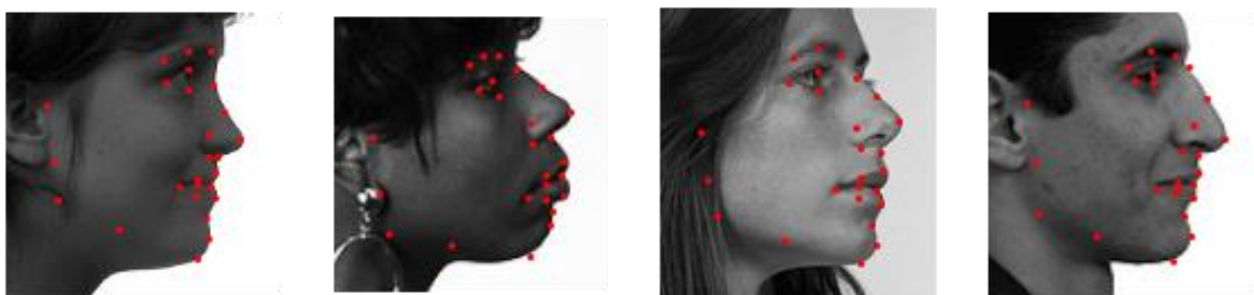


图 3-6 一些标注了侧脸 26 特征点的训练集图片

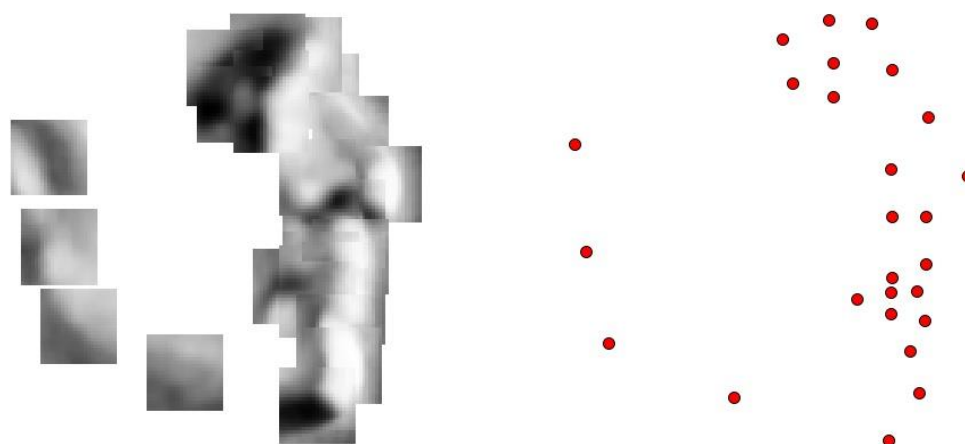
接着，通过 Menpo 库函数建立 PatchAAM 模型。该模型采用了 SIFT 图像特征，最大外观特征向量数量为 20 个，最大形状特征向量数量为 150 个。因为这里训练集大小为  $99 < 150$ ，所以形状特征向量的数量实际取值为 99。最终生成的 AAM 模型中的平均外观  $\bar{\mathbf{a}}$  与平均形状  $\bar{\mathbf{s}}$ ，如图 3-7 所示。

```
from menpofit.aam import PatchAAM
from menpo.feature import fast_dsift
```

```
patch_aam = PatchAAM(training_images, group='PTS', patch_shape=[(15, 15), (23, 23)],
                      diagonal=150, scales=(0.5, 1.0), holistic_features=fast_dsift,
                      max_shape_components=20, max_appearance_components=150,
                      verbose=True)
```

```
- Computing reference shape
- Building models
  - Scale 0: Done
  - Scale 1: Done
```

```
Computing batch 0
```

图 3-7 由 FERET 侧脸训练集生成的 PatchAAM 模型的  $\bar{\mathbf{a}}$  (左) 与  $\bar{\mathbf{s}}$  (右)

最后，将 AAM 模型的优化器设置为卢卡斯-卡纳德优化器，设置迭代模式为反向组合迭代，建立特征提取器。

```
from menpofit.aam import LucasKanadeAAMFitter, WibergInverseCompositional

fitter = LucasKanadeAAMFitter(patch_aam, lk_algorithm_cls=WibergInverseCompositional,
                              n_shape=[5, 20], n_appearance=[30, 150])
```

下面测试 AAM 模型在测试集上的表现，测试所用的数据集与 2.2.2 节相同。首先随机载入测试集中的六张图片，并且将图像转为灰度图，因为训练 AAM 模型时使用的是灰度图：

```
import os
import cv2
test_path = r'F:\PythonFiles\FaceLandmarks\DataSets\FERET\FERET_pr_test'
img_names = []
for _, file_names in os.walk(test_path):
    for file_name in file_names:
        if file_name.endswith('.jpg'):
            img_names.append(file_name)
imgs = []
for i in range(6):
    img = mio.import_image(os.path.join(test_path, img_names[np.random.randint(len(img_names))]))
    img = img.as_greyscale()
    imgs.append(img)
```

接着，载入 2.2.2 节训练的侧脸检测器：

```
from menpodetect.dlib import DlibDetector
detector = DlibDetector('profile_detector.svm')
```

对于测试集中的每一张图片，先使用侧脸检测器检测侧脸的边界框，以便在特征点定位过程中确定特征点的初始位置。接着，使用特征提取器进行侧脸特征点定位，得到每个特征点的预测坐标值。特征点的最终位置如图 3-8 所示；迭代过程如图 3-9 所示。

```
result_imgs = []
for img in imgs:
    bbox = detector(img)[0]
    result_imgs.append(fitter.fit_from_bb(img, bbox, max_iters=[15, 5]))
```



图 3-8 对测试集中随机侧脸图片的特征点拟合结果

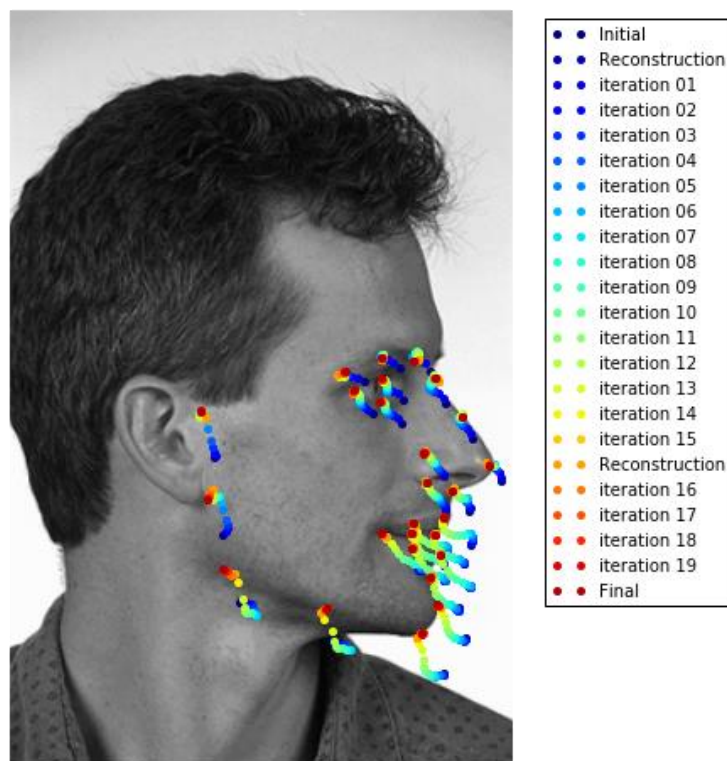


图 3-9 AAM 模型拟合的迭代过程

从图中可以看出，该模型可以较好地定位侧脸特征点，在一些图片中由模型算法得到的特征点精度甚至高于手工标注。因此，PatchAAM 模型能够满足本项目的要求。

## 第四章 三维人脸重建

### 4.1 研究背景

目前，有很多种技术被用于三维建模，使用最普遍的两种方式为基于多视图几何的三维重建以及基于深度相机的三维重建<sup>[52]</sup>。基于多视图几何的三维重建需要通过多台相机从不同角度同时拍摄物体照片，再通过多视图几何算法得到目标物体的三维模型。该方法虽然能够获得很高的建模精度，但是同时也对拍摄条件与设备有很高的要求。

基于深度相机的三维重建利用普通相机获取 RGB 通道，利用飞行时间（Time of Flight, TOF）方法获取包含深度信息的 D 通道，组合成具有 X、Y 与 Z 三维坐标的深度图像。Kinect、iPhone X 的面部解锁系统等均是这种三维建模方式的应用。然而，如此获得的深度信息易受待测物体与环境光线的影响存在大量噪声，建模过程会被严重干扰；而且，此类设备造价昂贵，使用成本较高。

基于以上原因，本项目决定设计一种以两张或两张以上的多视角普通人脸照片，通过三维可形变模型（3D Morphable Model, 3DMM）<sup>[53]</sup>完成人脸重建过程的方法。与上面提到的基于多视图几何的三维重建不同，此方法对人脸照片的拍摄设备与条件没有要求，而且最少只需要正面与侧面两张照片即可完成人脸三维重建过程，省时省力。两张不同视图的人脸照片可以提供人脸三个维度的完整信息，以此得到的建模准确性必然高于基于单张正脸的 3DMM 人脸合成。因此，本方法相比于基于多视图几何、基于深度相机与基于单张照片的 3DMM 三维重建过程都具有一定的优势。

### 4.2 模型介绍

三维可形变模型方法于 1999 年在论文<sup>[53]</sup>中被首次提出，其主要思想类似于 AAM 模型（见 3.3 节），不过模型的建立与计算不再在二维平面内，而是在三维空间内进行。

首先，通过基于多视图几何的三维重建技术等方法获取一个含有  $m$  个三维人脸模型的数据集。三维人脸的几何特征以形状向量（shape vector） $\mathbf{S} = [X_1, Y_1, Z_1, X_2, \dots, Y_n, Z_n]^T \in \mathbb{R}^{3n}$  表示，形状向量中包含  $n$  个顶点（vertex）的  $X, Y, Z$  三维坐标；三维人脸的纹理特征以纹理向量（texture vector） $\mathbf{T} = [R_1, G_1, B_1, R_2, \dots, G_n, B_n]^T \in \mathbb{R}^{3n}$  表示，纹理向量中包含形状向量中每一个顶点对应的  $R, G, B$  数值；数据集中每个人脸的形状向量和纹理向量分别以  $\mathbf{S}_i$  和  $\mathbf{T}_i$  来表示。通过这  $m$  个样例，我们能够建立一个可形变的人脸模型。将模型对齐后，新的形状模型  $\mathbf{S}_{model}$  与纹理模型  $\mathbf{T}_{model}$  可以用  $m$  个样例线性组合后的重心坐标表示：

$$\mathbf{S}_{mod} = \sum_{i=1}^m a_i \mathbf{S}_i, \mathbf{T}_{mod} = \sum_{i=1}^m b_i \mathbf{T}_i, \sum_{i=1}^m a_i = \sum_{i=1}^m b_i = 1 \quad (6)$$

令  $\mathbf{a} = [a_1, a_2, \dots, a_m]^T$ ， $\mathbf{b} = [b_1, b_2, \dots, b_m]^T$ ，通过调整参数向量  $\mathbf{a}$  和  $\mathbf{b}$  就能够得到不同形状与材质的三维人脸模型。

为了使得到的  $\mathbf{S}_{model}$  与  $\mathbf{T}_{model}$  接近真实人脸，我们必须对参数向量  $\mathbf{a}$  和  $\mathbf{b}$  分布的概率进行评估。为此，我们在数据集中引入多变量正态分布（Multivariate Normal Distribution，又名高斯分布，Gaussian Distribution），得到平均形状模型  $\bar{\mathbf{S}}$ 、平均纹理模型  $\bar{\mathbf{T}}$  和通过样例与平均值的



差值  $\Delta \mathbf{S}_i = \mathbf{S}_i - \bar{\mathbf{S}}$ 、 $\Delta \mathbf{T}_i = \mathbf{T}_i - \bar{\mathbf{T}}$  计算出的协方差矩阵  $\mathbf{C}_S$ 、 $\mathbf{C}_T$ 。接着，使用主成分分析<sup>[54]</sup>进行数据压缩，将  $\mathbf{S}_i, \mathbf{T}_i, \forall i=1, \dots, m$  用协方差矩阵的特征向量  $\mathbf{s}_i$  和  $\mathbf{t}_i$ （以对应的特征值由大到小排列）构成的正交基底表示：

$$\mathbf{S}_{model} = \bar{\mathbf{S}} + \sum_{i=1}^{m-1} \alpha_i \mathbf{s}_i, \mathbf{T}_{model} = \bar{\mathbf{T}} + \sum_{i=1}^{m-1} \beta_i \mathbf{t}_i \quad (7)$$

其中， $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_{m-1}]^T, \boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_{m-1}]^T \in \mathbb{R}^{m-1}$ 。参数  $\boldsymbol{\alpha}$  的概率  $p(\boldsymbol{\alpha})$  为：

$$p(\boldsymbol{\alpha}) \propto \exp \left[ -\frac{1}{2} \sum_{i=1}^{m-1} \left( \frac{\alpha_i}{\sigma_i} \right)^2 \right] \quad (8)$$

其中， $\sigma_i^2$  为形状协方差矩阵  $\mathbf{C}_S$  的特征值。参数  $\boldsymbol{\beta}$  的概率  $p(\boldsymbol{\beta})$  的计算方法同理。如此，我们就可以得到以平均形状模型  $\bar{\mathbf{S}} \in \mathbb{R}^{3n}$ 、平均纹理模型  $\bar{\mathbf{T}} \in \mathbb{R}^{3n}$ 、形状特征向量集合  $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{3n}\} \in \mathbb{R}^{m-1}$  和对应的特征值集合  $\{\sigma_{s,1}^2, \sigma_{s,2}^2, \dots, \sigma_{s,3n}^2\}$ 、纹理特征向量集合  $\{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_{3n}\} \in \mathbb{R}^{m-1}$  和对应的特征值集合  $\{\sigma_{t,1}^2, \sigma_{t,2}^2, \dots, \sigma_{t,3n}^2\}$  表示的三维可形变模型。根据模型依赖的编程语言不同，形状特征向量与纹理特征向量集合可能以长度为  $3n$  的列表或容器存储，或以  $3n \times (m-1)$  或  $(m-1) \times 3n$  的矩阵存储。通常，模型中还会提供表情特征向量集合  $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_t\} \in \mathbb{R}^n$ ，其中  $t$  为向量数量，根据模型变化而变化，但通常取愤怒、高兴、恐惧、伤心、惊讶和厌恶 6 个特征。此外，模型中还有顶点（vertex）之间的三角关系和相邻顶点、相邻三角面片信息、顶点之间的对称信息等。

网络上可以用于学术研究的公开三维人脸模型包括巴塞尔大学（University of Basel）建立的 BFM2009（Basel Face Model 2009）<sup>[55]</sup>、BFM2017（Basel Face Model 2017）<sup>[56]</sup>、伦敦大学学院（London's Global University）发布的包含一万多个人脸数据的 LSFM（Large Scale Facial Model）<sup>[57]</sup>、浙江大学发布的包含了多种表情信息的 FaceWarehouse<sup>[58]</sup>和萨里大学（University of Surrey）的 eos 项目<sup>[59]</sup>中提供的包含不同分辨率的三维可形变人脸模型。其中一些模型的平均形状模型如图 4-1。其中，BFM2009 是目前最常使用的可形变模型。如有需要，请参阅论文或项目网站上的方式进行申请与下载。

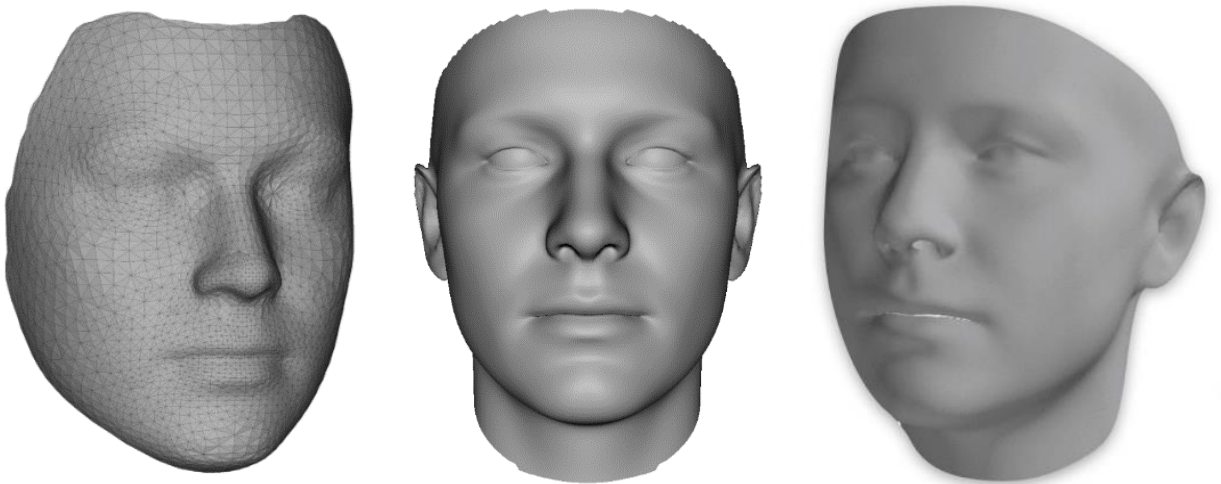


图 4-1 一些 3DMM 的平均形状模型。左：eos；中：BFM2009；右：LSFM<sup>[59,55,57]</sup>

### 4.3 模型拟合

令三维可形变人脸模型对包含两个或两个以上人脸视图的照片同时进行拟合是本项目重点研究的问题。本项目的算法与程序流程参考了论文<sup>[59-60]</sup>的算法与开源代码（访问 Github 以获取相应程序），并使用 eos 项目提供的 3447 顶点 3DMM、29587 顶点 3DMM 和 BFM2009 三种模型进行了实验验证。不同于<sup>[59]</sup>使用的 C++ 编程语言与<sup>[60]</sup>使用的 Matlab 语言，本项目的代码全部使用 Python 完成，尽管运行效率更低，但是代码含义更明确、易于调试，适用于实验与理论验证。若需要高性能的程序，请以 C++ 编译语言和 CUDA 重构本文代码。

#### 4.3.1 拟合算法

本项目使用的拟合算法基于特征点对应与最小二乘回归。使用此算法的前提条件是对于正脸图片中标注的特征点  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{68} \in \mathbb{R}^2$ ，在三维模型中存在且已知与之对应的顶点  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{68} \in \mathbb{R}^3$ ；该条件同样适用于侧脸图像。正脸与侧脸图像中的特征点在三维模型中可以对应同一个点。假设印堂（glabella）的位置在正脸图片中标注为 10 号点  $(x_{f,10}, y_{f,10})$ ，在侧脸图片中标注为 27 号点  $(x_{p,27}, y_{p,27})$ ，那么这两个点就同时对应于三维模型中的 526 号坐标点  $\mathbf{v}_{526} = (X_{526}, Y_{526}, Z_{526})$ 。当然，允许有些点只由正脸对应，另一些只由侧脸对应。特征点的选取可以自由拟定，只要二维图片中的特征点与三维模型中的顶点映射关系准确即可。

在二维与三维特征点映射关系已知的情况下，用三维人脸模型去拟合二维特征点的首要工作就是求取一个摄像机矩阵，使得三维模型在照片的二维空间内的投影位置与二维特征点大体相同。为简单起见，本项目中我们使用的是无穷远摄像机中的仿射摄像机（Affine Camera），即假设拟合所采用的二维图片是用分辨率无穷大的相机从无穷远处拍摄的。该假设会忽略照片拍摄的时候产生的透视效果，如图 4-2，从而引入一定的误差，但可以简化计算过程。



图 4-2 摄像机与被摄物体距离增加时，透视效应会逐渐减小直至消失<sup>[61]</sup>

令世界坐标系与物体坐标系重合，则从物体坐标系到世界坐标系的转换矩阵（Transform Matrix）为单位矩阵  $\mathbf{I}_{4 \times 4}$ 。如此一来，我们只需要计算目标人脸从世界坐标系到摄像机坐标系的转换矩阵  $[\mathbf{R}|\mathbf{t}]$  和摄像机的内部参数  $\mathbf{K}$  即可。其中，

$$[\mathbf{R}|\mathbf{t}] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} a_x & & x_0 \\ & a_y & y_0 \\ & & 1 \end{bmatrix} \quad (9)$$

$\mathbf{R}$  为一个正交矩阵。于是，一个  $3 \times 4$  齐次摄像机投影矩阵可以写为<sup>[61]</sup>：

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}] \quad (10)$$

取  $\mathbf{R}$  最后一行的前三个元素为 0 即可得到仿射摄像机投影矩阵  $\mathbf{P}_\infty$

$$\mathbf{P}_\infty = \mathbf{K} \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ 0 & 0 & 0 & t_z \end{bmatrix} \quad (11)$$

应用摄像机投影矩阵将三维模型中的点映射到二维平面内的变换可以写为：

$$\mathbf{x} = \mathbf{P}_\infty \mathbf{v} \quad (12)$$

因此，要计算  $\mathbf{P}_\infty$ ，首先需要获得旋转矩阵  $\mathbf{R}$ 。 $\mathbf{R}$  可以直接通过用用齐次坐标（Homogeneous Coordinates）表示的三维特征点与二维特征点计算得出。首先，用所有三维特征点与二维特征点坐标分别构成矩阵  $\mathbf{A}$  和  $\mathbf{b}$ ：

$$\mathbf{A} = \begin{bmatrix} \mathbf{v}_1^T | 1 \\ \mathbf{v}_2^T | 1 \\ \vdots \\ \mathbf{v}_n^T | 1 \end{bmatrix} \in \mathbb{R}^{n \times 4}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} \in \mathbb{R}^{n \times 2} \quad (13)$$

与上一节不同，这里的  $n$  代表特征点（landmark）的数量，而非三维模型中的顶点数量。下同。

利用伪逆矩阵求解矩阵方程  $\mathbf{A}\mathbf{r} = \mathbf{b}$  可以得出中间变量  $\mathbf{r} \in \mathbb{R}^{4 \times 2}$ 。令

$$\mathbf{r}^T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \end{bmatrix} = [\hat{\mathbf{R}} | \mathbf{t}] \quad (14)$$

缩放参数  $s$  可以由

$$s = (\|\mathbf{r}^{1T}\|^2 + \|\mathbf{r}^{2T}\|^2) / 2 \quad (15)$$

得到。其中， $\mathbf{r}^{1T}$  为矩阵  $\hat{\mathbf{R}}$  的第一行， $\mathbf{r}^{2T}$  为矩阵  $\hat{\mathbf{R}}$  的第二行。单位平移矩阵  $\mathbf{t} = [t_1/s, t_2/s, 0]^T$ ，单位旋转矩阵  $\mathbf{R}$  由

$$\mathbf{R} = \begin{bmatrix} 1/\|\mathbf{r}^{1T}\|^2 & & \\ & 1/\|\mathbf{r}^{2T}\|^2 & \\ & & 1/\|\mathbf{r}^{1T} \times \mathbf{r}^{2T}\|^2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{r}^{1T} \\ \mathbf{r}^{2T} \\ \mathbf{r}^{1T} \times \mathbf{r}^{2T} \end{bmatrix} \quad (16)$$

得到。此时， $\mathbf{R}$  不一定是正交矩阵。因此，需要对  $\mathbf{R}$  进行奇异值分解（Singular Value Decomposition, SVD） $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \text{svd}(\mathbf{R})$ ，并令  $\mathbf{R} = \mathbf{U}\mathbf{V}^T$  得到正交的旋转矩阵。最后，通过内参  $\mathbf{K}$  将转换矩阵  $[\mathbf{R}|\mathbf{t}]$  映射到图像空间中，并令矩阵最后一行等于  $[0 \ 0 \ 0 \ 1]$ ，即最终完成了仿射摄像机投影矩阵  $\mathbf{P}_\infty \in \mathbb{R}^{3 \times 4}$  的构建。

分别通过三维模型、正脸图片与侧脸图片中相对应的特征点得到仿射摄像机投影矩阵  $\mathbf{P}_f$  和  $\mathbf{P}_p$  后, 我们就可以开始进行三维特征点对图像的拟合了。本项目的拟合算法基于论文<sup>[62]</sup>进行修改而得到。

首先, 通过摄像机投影矩阵建立一个大小为  $3(n_f + n_p) \times 4(n_f + n_p)$  的分块对角矩阵  $\mathbf{P}_{diag}$ , 其中  $n_f$  表示正脸特征点的数量,  $n_p$  表示侧脸特征点的数量。矩阵的前  $n_f$  个元素为正脸摄像机投影矩阵  $\mathbf{P}_f$ , 其余的元素为侧脸摄像机投影矩阵  $\mathbf{P}_p$ :

$$\mathbf{P}_{diag} = \begin{bmatrix} \mathbf{P}_f & & & \\ & \ddots & & \\ & & \mathbf{P}_f & \\ & & & \mathbf{P}_p \\ & & & & \ddots \\ & & & & & \mathbf{P}_p \end{bmatrix} \in \mathbb{R}^{3(n_f + n_p) \times 4(n_f + n_p)} \quad (17)$$

接着, 将正脸特征点对应三维坐标点的形状特征向量  $\mathbf{S}_{vf,1}, \mathbf{S}_{vf,2}, \dots, \mathbf{S}_{vf,n_f}$  和侧脸特征点对应三维坐标点的形状特征向量  $\mathbf{S}_{vp,1}, \mathbf{S}_{vp,2}, \dots, \mathbf{S}_{vp,n_p}$  构成一个  $4(n_f + n_p) \times m$  大小的特征向量矩阵  $\hat{\mathbf{V}}$ 。其中,  $m$  为需要拟合的参数数量, 即特征向量的长度。该长度可以等于或小于三维可形变模型建立时的三维人脸数据数量。因为每个坐标点在 X, Y, Z 方向上各有一个特征向量, 而且矩阵运算时坐标以齐次形式表示, 因此特征向量矩阵  $\hat{\mathbf{V}}$  共包含  $3(n_f + n_p)$  个特征向量和  $n_f + n_p$  个 0。故

$$\mathbf{S}_{v_f,i} = \begin{bmatrix} \mathbf{s}_{Xfi}^T \\ \mathbf{s}_{Yfi}^T \\ \mathbf{s}_{Zfi}^T \\ 0 \end{bmatrix} \in \mathbb{R}^{4 \times m}, i \in [1, n_f], \quad \mathbf{S}_{v_p,i} = \begin{bmatrix} \mathbf{s}_{Xpi}^T \\ \mathbf{s}_{Ypi}^T \\ \mathbf{s}_{Zpi}^T \\ 0 \end{bmatrix} \in \mathbb{R}^{4 \times m}, i \in [1, n_p] \quad (18)$$

那么, 特征向量矩阵  $\hat{\mathbf{V}}$  可以表示为:

$$\hat{\mathbf{V}} = \begin{bmatrix} \mathbf{S}_{v_f,1} \\ \vdots \\ \mathbf{S}_{v_f,n_f} \\ \mathbf{S}_{v_p,1} \\ \vdots \\ \mathbf{S}_{v_p,n_p} \end{bmatrix} \in \mathbb{R}^{4(n_f + n_p) \times m} \quad (19)$$

用齐次坐标表示的平均形状向量  $\bar{\mathbf{V}}$  可以表示为:

$$\bar{\mathbf{V}} = \begin{bmatrix} X_{f,1} & Y_{f,1} & Z_{f,1} & 1 & \cdots & Z_{f,n_f} & X_{p,1} & \cdots & Z_{p,n_p} & 1 \end{bmatrix}^T \in \mathbb{R}^{4(n_f + n_p) \times 1} \quad (20)$$

同理, 正脸与侧脸图像内标注的特征点也可以以齐次坐标在一个向量内表示为:

$$\mathbf{x} = \begin{bmatrix} x_{f,1} & y_{f,1} & 1 & \cdots & y_{f,n_f} & x_{p,1} & \cdots & y_{p,n_p} & 1 \end{bmatrix}^T \in \mathbb{R}^{3(n_f + n_p) \times 1} \quad (21)$$

最后, 我们可以列出方程:

$$\mathbf{A}\mathbf{a}=\mathbf{b} \quad (22)$$

其中， $\mathbf{a}$ 为我们要求解的形状模型参数向量， $\mathbf{A}$ 表示特征向量在二维平面内的投影：

$$\mathbf{A}=\mathbf{P}_{diag}\hat{\mathbf{V}} \quad (23)$$

$\mathbf{b}$ 为二维特征点 $\mathbf{x}$ 与平均形状向量 $\bar{\mathbf{V}}$ 在二维平面内的投影的差值：

$$\mathbf{b}=\mathbf{x}-\mathbf{P}_{diag}\bar{\mathbf{V}} \quad (24)$$

由于模型需要符合高斯过程，因此在计算过程中引入对角矩阵 $\Omega$ 来控制每一个特征向量参数的置信程度：

$$\Omega=\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{3(n_f+n_p)} \end{bmatrix} \quad (25)$$

同时，正则化参数 $\lambda$ 被用于控制模型的平滑程度。综上所述，方程的解 $\mathbf{a} \in \mathbb{R}^{m \times 1}$ 可以由下式计算得到：

$$\mathbf{a}=(\mathbf{A}^T\Omega\mathbf{A}+\lambda\mathbf{I})^{-1}(\mathbf{A}^T\Omega\mathbf{b}) \quad (26)$$

其中， $\mathbf{I}$ 为单位对角矩阵。

表情拟合的过程与模型拟合基本一致，只有一点需要注意：表情特征向量的参数不能为负数，因此需要用非负最小二乘法（Non-negative Least-squares, NNLS）进行拟合。

### 4.3.2 程序流程

基于 4.3.1 节介绍的算法，我使用 Python 编写了 3DMM 拟合程序。算法流程如图 4-3 所示：

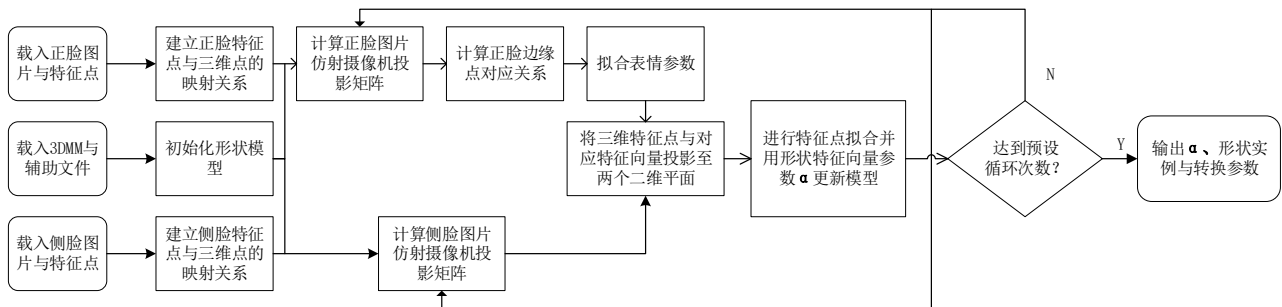


图 4-3 三维特征点拟合程序流程图

由于 Dlib 正脸特征点检测程序定位的轮廓特征点（Contour Landmark）不是真正意义上的如下颌骨沿线的人脸轮廓点，而是图像中人脸与环境接触的边缘位置上的点。对于不同的姿态，正脸检测程序输出的人脸轮廓点在三维人脸模型上的映射会发生变化。因此，对于每张正脸图片，我们都需要重新计算图片中正脸边缘特征点与三维人脸模型上顶点的映射关系。

计算正脸边缘点的对应关系的方法为：利用存在对应关系的特征点（如眼睛、鼻子、嘴的轮廓特征点）计算放射摄像机投影矩阵，并将三维模型投影到二维平面内后，利用反投影变

换和 k-d 树在三维模型的可见 (Visible) 投影上搜索距离目标二维特征点最近的点, 并建立两点之间的映射关系。

### 4.3.3 拟合表现

FERET 数据集中的六组图片被随机挑选出来作为测试数据。正脸和侧脸图片均为  $256 \times 384$  像素的灰度图片, 人脸占图片的比例不确定。图片中的正脸特征点用 Dlib 特征点定位程序进行标注, 侧脸特征点用 3.4.2 节训练的模型进行标注, 标注的点的位置以与图像同名的 pts 文件的形式存放在与图像文件相同的目录下。进行拟合前, 程序首先读取图片和特征点文件, 并对图片与特征点位置等比缩放到图片高度为 2000 像素。拟合所用的正脸特征点数量为 66 个 (有两个嘴唇位置的点没有用到), 侧脸特征点数量为 15 个 (因为个人标注的特征点存在较大误差, 而且难以确定在三维模型中的特征点编号, 所以项目只取了一些位置比较明确而且对应编号在模型文档中已经给出的点)。在代码中, 给  $\Omega$  的赋值为常数与单位对角矩阵的乘积, 因而每一个特征点的  $x$ 、 $y$  坐标在拟合时占相同权重。而正脸特征点较多, 而且无论是正脸图片还是侧脸图片都包括坐标点在三维空间内的  $Y$  坐标, 因此可以预见, 三维模型将在正脸图片上和  $Y$  空间内做出更好的拟合效果。拟合结果如图 4-4、4-5 与 4-6 所示。图 4-4 与 4-6 中的蓝色正方形是特征点拟合程序确定的人脸特征点, 红色原点代表特征点对应的三维模型上的顶点在二维平面的的投影; 绿色到蓝色渐变的透明掩模 (Mask) 为拟合所得的三维人脸形状网格实例经过仿射摄像机矩阵在图片平面内的投影。图 4-5 为图 4-4 中的形状网格实例保存为 .ply 文件后用 Windows 10 自带的 3D Builder 软件查看结果。

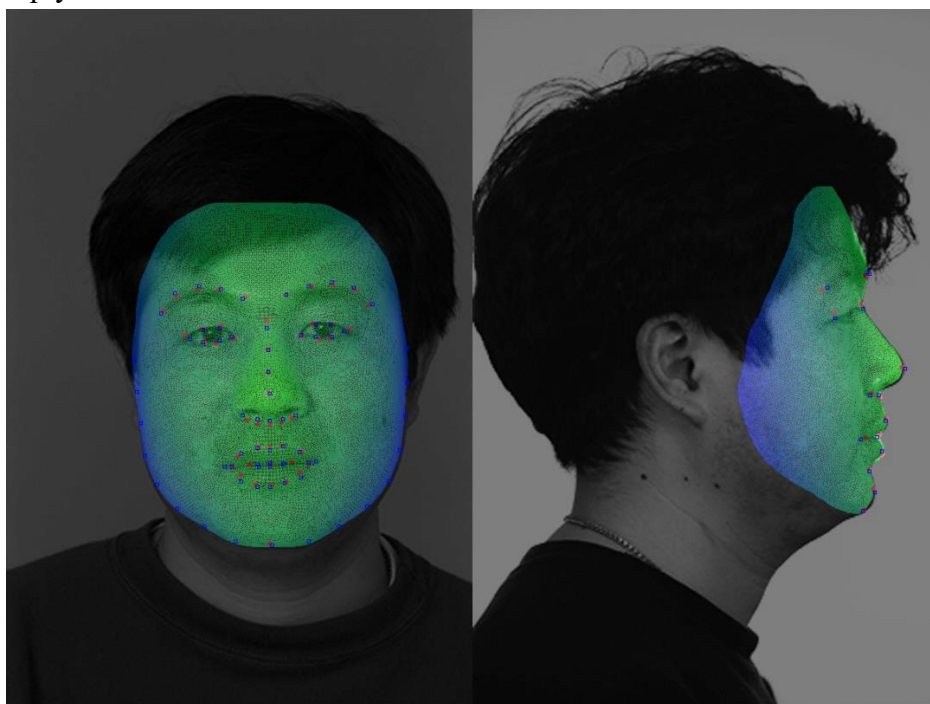


图 4-4 3DMM 对 FERET 数据集上 00029ba010\_960521.tif 和 00029pr010\_940128.tif 文件进行拟合的结果。



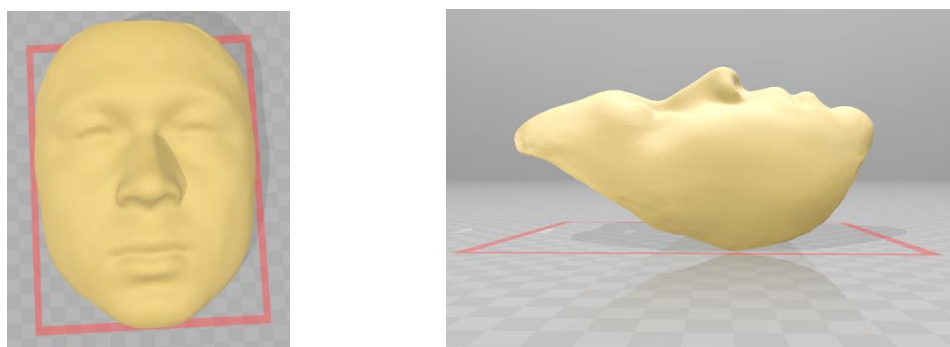


图 4-5 图 4-4 中人脸拟合生成的三维模型的正面与侧面，用 3D Builder 软件打开

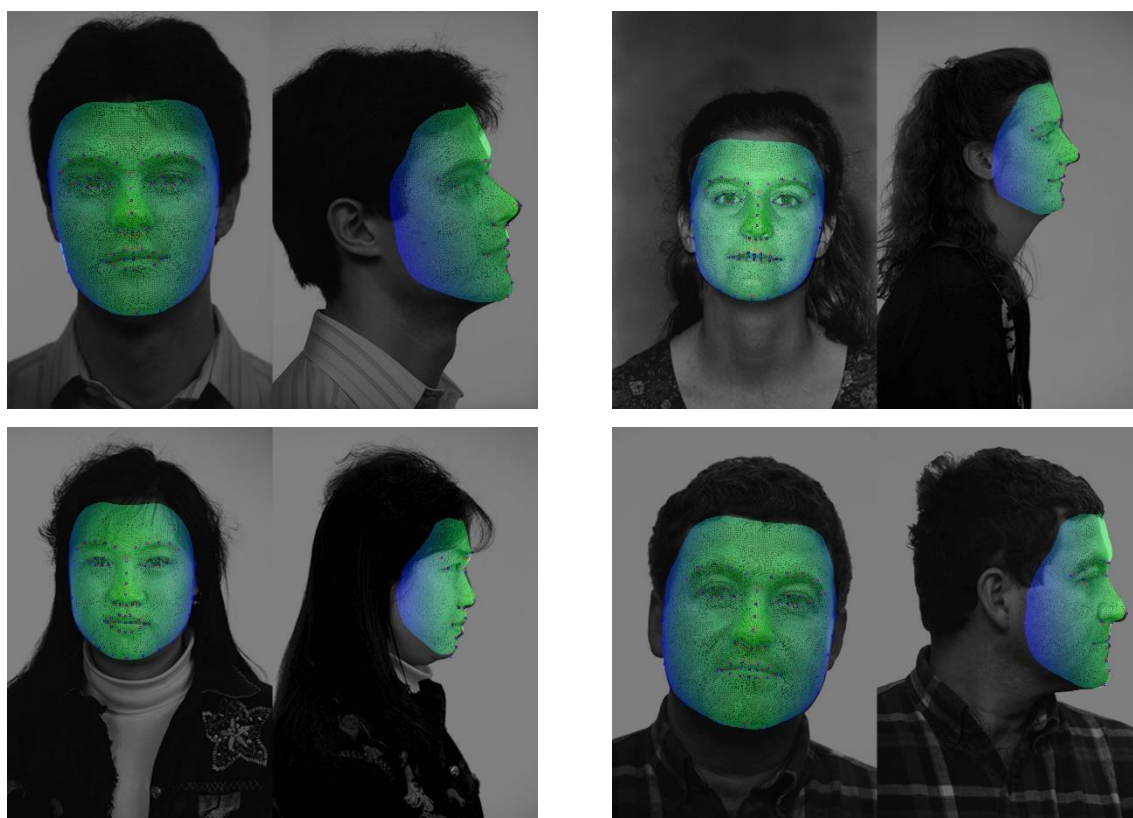


图 4-6 对 FERET 数据集中其他组人脸进行拟合的结果

由图 4-6 可以看出，3DMM 拟合对于有人脸特征点约束的位置都可以取得很好的拟合效果。因此，理论上，只要以足够多的特征点（事实上，只需要多规定一些侧脸轮廓和额头部分的特征点），就能够使 3DMM 较为真实地还原人脸的全部细节。

为了方便起见，本项目目前使用的三维可形变模型为 eos 项目中提供的 3448 点与 29587 点三维模型（图 4-4~4-6 均使用 eos29587 点三维模型拟合）。然而，BFM2009 模型具有更多的顶点（53490 个），可以表达更多的细节，而且具有更多的特征向量，可以拟合更多样的人脸，更适于实际工程中使用。理论上，LSFM 模型应具有更优良的特性，但是使用此模型的工程目前还很少，实际效果有待验证。

最后，图 4-7 展示了目前工程研究中最经常使用的只用三维可形变模型拟合正脸特征点，再将输出模型投影到侧脸图片上的输出结果。与图 4-4 对比，可以明显看出本文提出的算法拟合结果要比常用方法更能体现人脸的真实外观。这种优越性普遍存在。

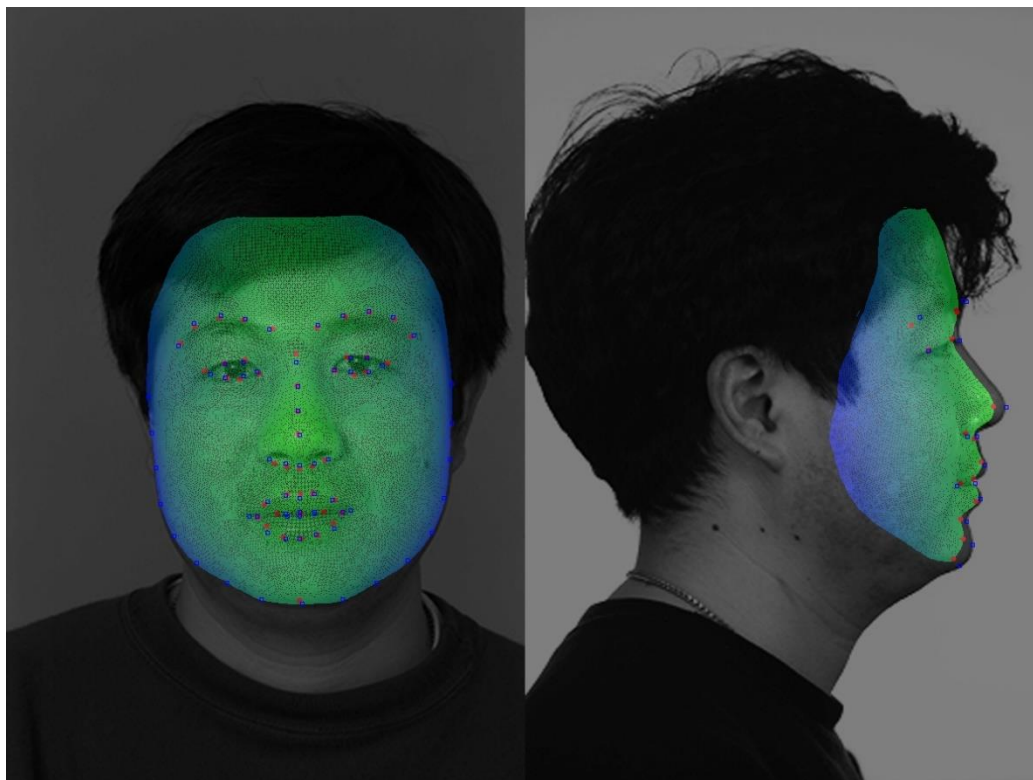


图 4-7 只用三维可形变模型做正脸特征点拟合的拟合效果

论文<sup>[63-64]</sup>也提出了通过多张图片进行三位人脸建模的方法。然而，他们使用的多张图片姿态差距不大，全都是正脸或接近正脸的照片，而且标注方式全部为 Multi-PIE/IBUG 68 标注。因此，从信息论的角度来说，本文使用的标注方法含有更多的信息。尽管<sup>[63-64]</sup>使用了更加先进的拟合方法——基于深度学习进行非线性回归，从拟合结果来看最终的精度与本文差别不大。非线性回归方式对于本项目使用的线性回归方式还是具有一定的优势，此话题将在第六章讨论。

## 4.4 误差分析

由于各种因素，在三维可形变模型拟合过程中有很多步骤可能会引入误差。这些误差中，有些是随机误差，有些是系统误差。下面就是关于模型拟合过程可能产生的误差的分析。

### 4.4.1 人脸检测误差

人脸检测程序输出的边界框的位置和大小可能会与真实值存在一定的误差。然而，此误差不会对最终结果产生影响：边界框在本程序中只用于在特征点定位过程中确定人脸平均形状实例的初始位置；即使人脸平均形状实例位置的初始化不精确，人脸对齐算法也会通过迭代和梯度下降来消除这个初始误差。尽管如此，如果由于各种原因人脸检测程序没有检测到人脸，或者误检测到人脸，那么整个流程就会彻底崩溃。减小人脸检测器的误差的方法就是用更多的训练数据去训练人脸检测模型。这样，该检测器就能够更好地区分正负样本，从而做出更加准确的判断。



#### 4.4.2 人脸对齐误差

人脸对齐过程也不可避免地会出现误差。误差的来源主要有三个方面：

第一，训练集数据太少。当训练集过小时，建立的 AAM 模型中的平均外观模型和平均形状模型不能真正反映出外观和形状的平均值，这将加大特征点的回归难度；更主要的是，过小的训练集不能够覆盖到大多数的人脸形状与外观信息，导致最终生成的 PCA 模型中的特征向量无论如何设置参数也无法描述一些实例。例如，在本项目中，我只随机标注了 99 张 FERET 数据集中的侧脸图片。尽管这些侧脸图片几乎覆盖了全部的人脸形状实例，但是由于 FERET 数据集是在实验条件下获得的，人物背景与光照等环境条件十分单一；由此生成的外观模型就没有完全覆盖到自然条件下人物侧脸的与周边环境的外观信息。因此，理论上来说，本文训练的 PatchAAM 模型对自然状态下的侧脸照片的特征点定位效果不会十分理想（没有测试）。

第二，训练集特征点标注误差。这个误差主要来源与手工标注特征点的时候没有正确判断和标注特征点的位置，属于人为误差，无法避免。标注误差会对生成的 AAM 模型中的形状模型与外观模型，尤其是外观模型，的平均模型和 PCA 特征向量产生干扰。在训练集较小的时候，这种干扰尤为严重。在训练集足够大，而且绝大多数数据标注准确的情况下，特征点标注误差带来的影响可以忽略。除了人为失误外，训练集特征点的数据精度也会带来标注误差。例如对一张分辨率为的照片，使用整型数据以分辨率精度标注特征点就会带来标注误差。这种误差可以由预处理放大照片、利用浮点数存储标注数据等手段减小。

第三，特征提取器在回归特征点时产生的误差。这类误差难以控制，主要通过使用更复杂的算法构建特征点定位模型来减小。目前，有无数的研究都致力于构建更精确的特征点定位模型，具体可参考<sup>[39-43,65]</sup>。

#### 4.4.3 三维模型拟合误差

除了人脸对齐误差会对三维可形变模型的拟合造成影响外，三维模型拟合过程本身也存在一定的误差。这类误差包括但不限于：二维特征点与三维顶点的对应关系误差，即它们没有精确地表现人脸的同一个位置；摄像机投影矩阵的系统误差：算法中使用了仿射摄像机投影矩阵进行平行投影，而现实世界中的人脸图片全部使用透视摄像机（Perspective Camera）拍摄。使用放射摄像机投影矩阵构建出来的三维模型为带有透视效果的三维人脸模型，而非人脸的真实形状；转换矩阵构造时产生的误差等。

值得一提的是，因为本项目使用同一三维模型同时对正脸与侧脸特征点拟合，因此正脸与侧脸照片最理想的状态是除了角度以外其他信息完全一致。如果不一致则会影响拟合精度。例如图 4-6 中上面一组图片的人脸表情有所变化，最后侧脸嘴角特征点的投影位置就发生了较大偏差。

上述所有误差都会或多或少地影响算法生成的三位人脸模型实例效果，需要加以注意。

## 第五章 三维人脸模型在整形医学上的应用

利用三维可形变人脸模型生成的三维人脸实例在整形医学的研究与应用领域可以发挥十分重要的作用。下面,将以三维模型辅助整形医学研究、以三维模型参数辅助确定整形需求和以三维模型预览手术结果三个方向讨论三维人脸模型在整形医学上的应用。

### 5.1 辅助医学研究

目前,对于有吸引力的人脸研究(Study on Attractive Faces)过程主要由人脸图像采集、研究点位置标注、点与点之间距离的测量以及距离分析四个步骤组成<sup>[2-6]</sup>。虽然此类对于有吸引力的人脸各器官等部分的大小、比例的定量研究硕果累累,但是研究手段仍然比较原始。即使标注点的距离测定是由计算机软件辅助完成,样本的采集与标定都是在实验条件下手工进行的,容易受到仪器设备和人员的限制。同时,在距离测定时,人脸绕三个坐标轴的微小旋转都会给实验引入误差。在计算机辅助设计手段如此发达的今天,我们完全可以利用二到四章所述的三维建模过程通过正脸与侧脸两张图片直接给出被采集人脸的三维模型,并基于三维人脸模型的顶点位置进行研究。此方法建立样本模型的方法如此简单,以至于研究人员在短时间内就能够采集大量样本数据,并且可以突破地域条件的限制,获取来自世界各地的研究样本,从而得出适用性更广泛的结论,或者结合机器学习等先进手段探索更深层次的规律。以三维模型辅助整形医学研究的具体过程如下所述:

首先是样本采集工作。样本采集工作可以由研究人员完成,也可以由受试者自行完成。因为此步骤十分简单,只需要提供一组表情一致的正侧脸照片即可。理想状态下,这些照片应该使用长焦镜头在远处拍摄,但实际上该条件很难也不需要得到满足。

其次是利用第二章到第四章所述方法进行三维人脸建模。如果要能够自动定位侧脸图片的特征点,那么就需要用更多的自然状态下(in-the-wild)带标注的侧脸图片对本项目的侧脸数据集进行扩充。AFLW 数据库是一个不错的选择,但是特征点仍然需要自行标定。

实际应用中,三维人脸模型可以选择 BFM2009。因为该模型不仅提供了平均形状模型和形状特征向量,还提供了平均纹理模型和纹理特征向量。在此条件下,该模型不但可以拟合人脸的空间形状,还能够拟合人脸的纹理(即皮肤、瞳孔颜色等信息),使得模型的美观程度更容易观察,并以此进行面容评判。

然后是观察点的序号和坐标获取。一般来说,在整形医学中关注的特征点主要是 LG Farkas 在论文<sup>[66]</sup>中提出的位置,如图 5-1 所示。顶点序号的获取可以通过本项目的辅助程序来完成。该程序以 Python 和 OpenGL 完成,受制于 Python 缓慢的循环速度,该程序效率低下,但仍能够完成要求的功能。使用者可以用键盘 WASD 控制三维模型的旋转角度,用 IJKL 控制模型的位置。程序初始时会在鼻凸点初始化一个白点,在其相邻的任一顶点初始化一个红点;使用者可以按小键盘上的 4 或 6 键选择上一个或下一个以红点表示的白点的相邻点,按 5 可以选择白点位置为当前点,并在控制台打印该点的序号。如此,尽管速度比较慢,我们最终可以获取任何需要获取的任何点的序号。获取序号后,即可通过下标轻松地获取该点的位置。如图 5-2 与图 5-3 所示。

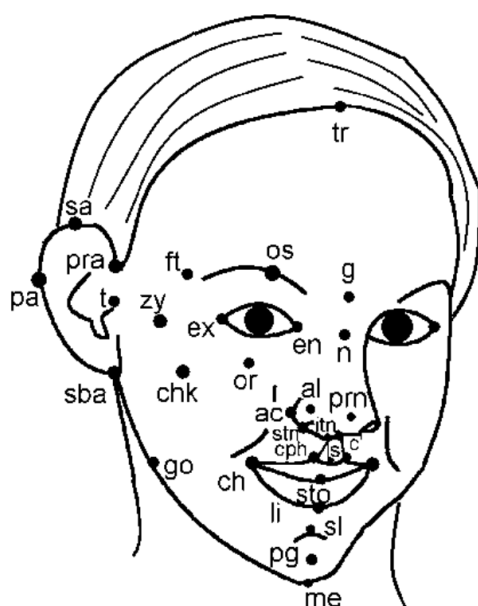


图 5-1 数字人脸坐标<sup>[67]</sup>。中线特征点：tr，发际中点；g，印堂；n，鼻根点；prn，鼻突点；c'，鼻小柱；sn，鼻底点；ls，上唇中点；sto，口点；li，下唇中点；sl，唇底点；pg，颏前点；me，下巴；对称特征点：ex，外眼角；en，内眼角；os，上眶点；or，眶点；ft，额颞点；chk，面颊；zy，颧点；t，耳屏点；al，鼻翼端；ac，鼻翼顶；itm，鼻孔轴内端；stn，鼻孔轴外端；cph，上唇突点；ch，口角；go，下颌角点；pra，耳廓前点；sa，耳上点；pa，耳廓后点；sba，耳下点。

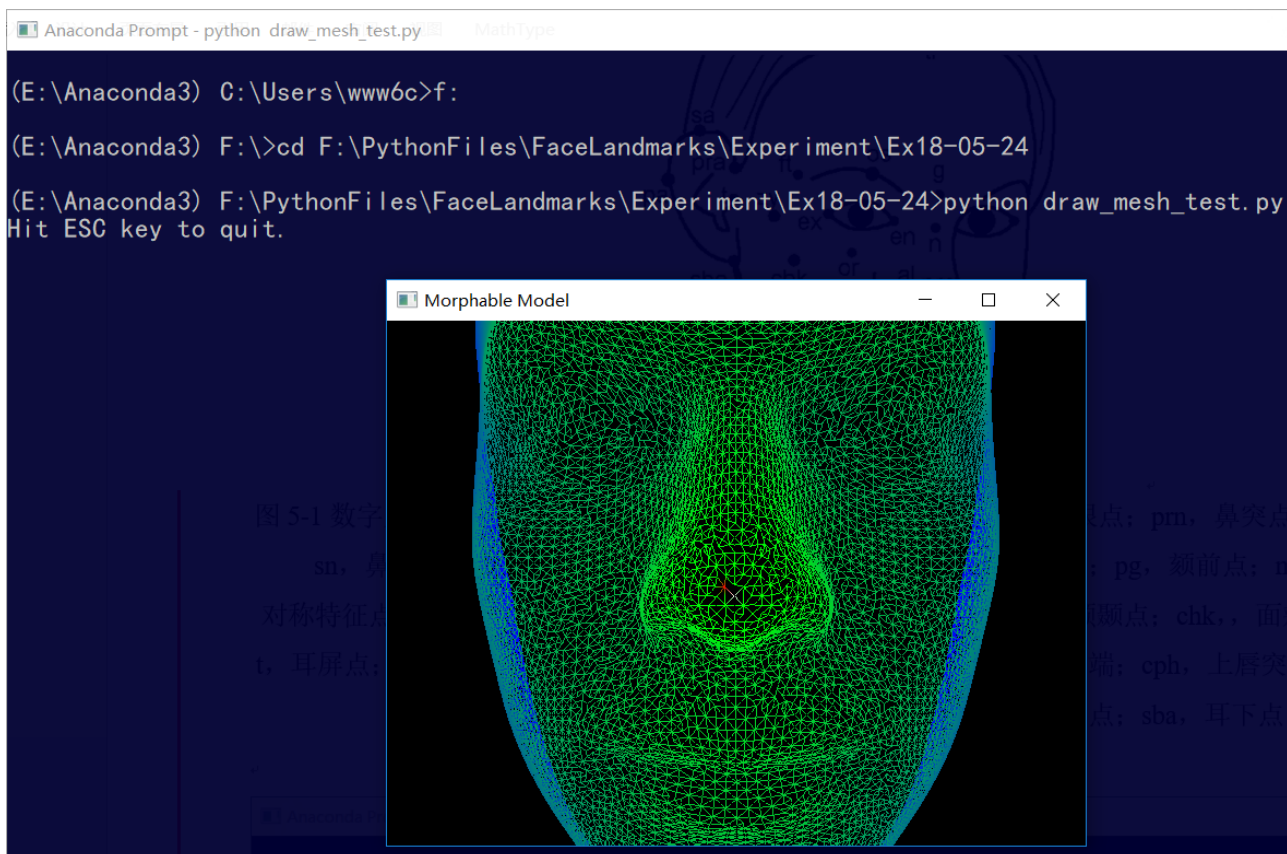


图 5-2 特征点序号获取程序程序开始运行时的界面。

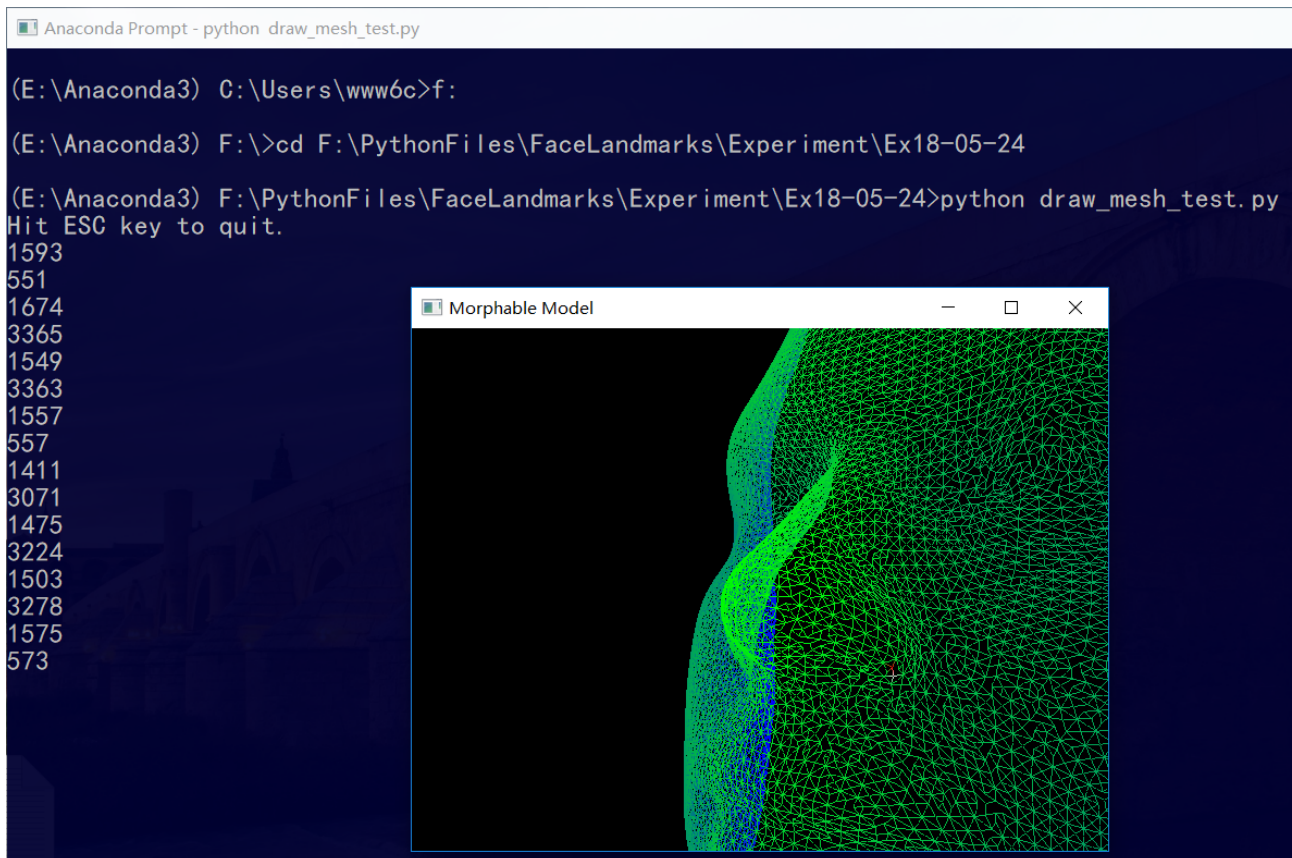


图 5-3 特征点序号获取程序经过几次移动后获取到鼻子一侧的一个顶点序号的界面。获取到的序号在控制台输出。

一个顶点在所有三维模型中的相对位置均相同。因此，只要获取一次所需的特征点，就可以输出所有三维模型中该点的三维坐标。这种方法要远远快于分别测量每一个样本各点的位置，可以提高效率。不仅如此，当研究人员想利用上述点外的其他点的数据进行研究时，只需要用程序获取该点的编号就能够更新各点之间的位置及比例，而不需要重新采集并测量样本。而且，本方法能够获得个点的三维坐标，与传统方法中获得的二维坐标相比，本方法能够提供更多的信息。

## 5.2 辅助确定整形需求

对有吸引力的面孔的研究通常会作为结论给出有吸引力的面孔中各部分的比例或数值<sup>[2-6]</sup>。尽管一些论文中得出了“人脸越小越有吸引力”的结论<sup>[6]</sup>，使用各部分的比例而非绝对数值来评价人脸是否美观也是普遍为人接受的一种研究方法。对于一个人脸样本，利用论文中发布的“有吸引力面孔”的比例数据，我们可以利用人脸三维建模的方式直接获取该人脸样本各部分的比例。将获取的比例与“有吸引力人脸”的比例进行比对，我们就能够获得定量的人脸缺陷数据，并用这些数据来指导整形手术。例如，论文<sup>[2]</sup>提出漂亮的汉族女性的颧骨宽度（左右 *chk* 点在人脸坐标系中 X 轴上投影的距离）与下颌角点宽度（左右 *go* 点在 X 轴上投影的距离）的比例为 1.151，而中脸长（*g* 点与 *c'* 点在 Y 轴上投影的距离）与下脸长（*c'* 点与 *me* 点在 Y 轴上投影的距离）的比例为 1.016。若现有一汉族女性样本颧骨宽度与下颌角

点宽度比例为 0.9, 而中脸长与下脸长比例为 1.2, 那么我们就可以安全地给她提供“削颧骨”与“垫下巴”的基础建议。

### 5.3 利用三维模型预览手术结果

拟定手术方案后, 我们就会得知某一特征点的位置变化。例如, 设一汉族女性样本  $g$  点  $Y$  轴坐标为 30,  $c'$  点  $Y$  轴坐标为 10, 现在想对她进行垫下巴手术, 手术后中脸长与下脸长的比例为 1。若已知该女目前  $me$  点的坐标为, 那么我们可以预测手术后该女  $me$  点坐标为那么, 以该女其他特征点当前的三维坐标 (当然, 应该排除  $sl$ 、 $pg$  等相关点) 和  $me$  点的新三维坐标进行三维模型拟合, 就能够获得手术结果的三维预览模型。该算法十分简单, 如有需要可自行完成。

## 第六章 分析与讨论

### 6.1 开发平台及编程语言的选取

本项目各部分均使用 Python 作为编程语言。然而，在第四章和第五章均有提到，Python 的运行效率很低。编写第四章所述的程序时，尽管本文作者花费了大量时间对程序进行优化，最后的运行效率是程序刚刚调试成功时的几千倍，但是仍然不及用 C++，甚至是 Matlab 完成的同类型程序，仅仅是 C++ 程序的几十分之一。而且，在第五章的程序中，尽管基于 Python 的 OpenGL: PyOpenGL 库支持 GPU 加速，但受制于 Python 缓慢的 for 循环系统，GPU 难以满负荷工作，发挥应有的作用。本项目使用 python 作为编程语言的考虑在于：

第一，作为交互式的语言，Python 程序调试效率较高，在此特性支持下，本文使用的 eos 项目代码用 Python 重构后再修改算法，而未在原来的 C++ 项目里直接修改；第二，Python 语法简单，完成同样的功能所需的代码量要远远少于 C++，而且程序逻辑清晰，有强大的内置序列化功能，方便保存模型；第三，因为项目庞大复杂，要调用很多库（外部依赖项），而 C++ 每一个外部依赖项都对同一个功能用不同的类实现，导致经常需要将数据在不同的类型之间来回转换。举例来说，对于向量与矩阵，eos 项目代码中就使用了 `std::vector`、`std::vector<std::vector>`、`Eigen::Vector`、`Eigen::Matrix`、`glm::Matrix` 等形式保存和实现不同功能，而 Python 只需要 `numpy.array` 一种。第四，帮助文档完善。如果要了解 `numpy.array` 的特性，只需要在命令行中输入 `help(numpy.array)` 即可得到详细准确的文档，而不用耗时间在网络中搜索。第五，相对于 Matlab，Python 对于非数学功能的支持要完善很多，而数学功能因为 `numpy` 的存在也不逊色于 Matlab 太多。

综上，可以看出，尽管 Python 运行效率低下，但它的开发效率极高，非常适用于实验研究等场合。事实上，目前有很多很有影响力的项目均由 Python 完成。另一方面，本项目针对的应用对计算性能要求尚可，只需交互式的完成人脸建模而无须进行实时的建模跟踪，因此，本项目选择 Python 作为基础编译语言可以完成目标。

### 6.2 基于神经网络的人脸标记与重建

项目起初也对基于 CNN 的人脸特征点检测技术进行了尝试性探索，构建了一个三层卷积两层全连接的以电脑 2G 的 GPU 能带得动的最复杂卷积神经网络模型，然后用它在一个 400 张带标注图片的侧脸训练集上训练（最初在 PIE 项目的生成模型中构建的，探索中发现这个数据集实用性太低。如图 3-2 所示为来自训练集的一张图，不过 3-2 的特征点是人为交互式标注的，而训练集是正经标在了字典里然后固化成 `pkl` 文件的）。初期结果表明模型在训练集上表现良好，但在测试集上的拟合效果较差。出现这种现象的原因应该不是模型构建的问题，因为同样的技术用全连接层构建的神经网络在机械假手肌电信号识别上就取得了几乎 100% 的准确率。所以导致此方法表现不良的原因主要是在于是训练集太小。鉴于自行构建与深度学习算法相容的训练集使用时间成本较高，本项目未采用基于深度学习的方法。后期可在此方面继续进行探索。

## 6.3 非线性回归

第四章提到的非线性回归的优势在于，它可以对整个拟合过程，包括摄像机矩阵计算、表情拟合和形状拟合的过程进行回归，使最后二维与三维投影特征点之间的残差尽可能小。而本文使用的算法只对形状特征向量进行了回归，表情特征向量、摄像机矩阵和边缘特征点的对应尽管在每个循环都被计算，但是它们也仅仅是被计算了而已，新计算出的结果的误差为进行详尽分析对比。事实上，当循环次数足够大时，三者确实发生了发散，而使用非线性回归对全局进行优化则不会出现这种问题。本文算法最后仅计算边缘特征点的对应关系一次、表情特征向量一次、摄像机矩阵参数三次。在这些约束条件下，该算法取得了最好的效果。

在今后类似项目的研究中，可以考虑用梯度下降算法等非线性回归方法取代本项目中使用的线性回归，查看该情况下迭代的收敛效果。

## 6.4 本项目的贡献

本项目最主要的贡献为开发了一种三维人脸模型拟合的新手段，取得了比传统方法更加准确的拟合结果，并提出了利用该模型辅助整形医学研究的方法。

除此之外，本项目的贡献主要有：

- 定义了一种侧脸 26 特征点标注标准。
- 在 FERET 数据集上建立了一个包含 99 张图片标注的侧脸数据集；
- 基于 Python 和 OpenCV 实现了任意数量人脸特征点标注和存储程序；
- 基于 Python 和 OpenCV 实现了修改带标注图片的标注点位置的程序；
- 基于 Python 和 OpenGL 实现了确定三维模型中顶点序号（Index）的程序。

本项目的程序整理后将全部上传至 Github，请搜索账号 Yinghao-Li 以获取程序源码。

## 致 谢

本项目在朱利丰老师的指导下完成，在整个过程中老师一直给予我及时与无私的帮助。即使远在美国，朱老师还会隔着十二三个小时的时差通过网络视频与屏幕分享来解决我的千奇百怪的问题。在此，我向朱利丰老师致以衷心的感谢和美好的祝愿。

感谢完定阳同学允许我使用他的肖像进行学术研究。

感谢所有提供开源代码的程序员，你们的贡献是本项目顺利进行必不可少的条件。



## 参考文献

- [1] Samuel A L. Some studies in machine learning using the game of checkers[M]// Computers & thought. MIT Press, 1995:206-226.
- [2] Zhao Q, Zhou R, Zhang X D, et al. Morphological Quantitative Criteria and Aesthetic Evaluation of Eight Female Han Face Types[J]. Aesthetic Plastic Surgery, 2013, 37(2):445-453.
- [3] Choe K S, Sclafani A P, Litner J A, et al. The Korean American Woman's Face[J]. Archives of Facial Plastic Surgery, 2004, 6(4):244-52.
- [4] Gao Y, Niddam J, Noel W, et al. Comparison of aesthetic facial criteria between Caucasian and East Asian female populations: An esthetic surgeon's perspective[J]. Asian Journal of Surgery, 2016, 41(1).
- [5] Edler R, Agarwal P, Wertheim D, et al. The use of anthropometric proportion indices in the measurement of facial attractiveness[J]. European Journal of Orthodontics, 2006, 28(3):274.
- [6] Milutinovic J, Zelic K, Nedeljkovic N. Evaluation of Facial Beauty Using Anthropometric Proportions[J]. The Scientific World Journal, 2014, (2014-2-20), 2014, 2014(2):428250.
- [7] Ojala T, Pietikainen M, Harwood D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions[C]// Iapr International Conference on Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision & Image Processing. IEEE, 2002:582-585 vol.1.
- [8] Lowe D G. Object Recognition from Local Scale-Invariant Features[C]// The Proceedings of the Seventh IEEE International Conference on Computer Vision. IEEE, 2002:1150.
- [9] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]// IEEE Computer Society Conference on Computer Vision & Pattern Recognition. IEEE Computer Society, 2005:886-893.
- [10] Huang G B, Mattar M, Berg T, et al. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments[J]. Month, 2007.
- [11] Freund, Yoav, Schapire, Robert E. A decision-theoretic generalization of on-line learning and an application to boosting[J]. Journal of Computer and System Sciences, 1997, 55(1):23-37.
- [12] Hsu R L, Abdel-Mottaleb M, Jain A K. Face detection in color images[C]// International Conference on Image Analysis and Recognition. Springer-Verlag, 2010:454-463.
- [13] Viola P, Jones M J. Robust Real-Time Face Detection[C]// Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. IEEE Xplore, 2004:747.
- [14] Baum L. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process[J]. Inequalities, 1972, 3:1-8.
- [15] Osuna E, Freund R, Girosi F. Training Support Vector Machines: an Application to Face Detection[C]// Computer Vision and Pattern Recognition, 1997. Proceedings. 1997 IEEE Computer Society Conference on. IEEE, 2002:130-136.
- [16] King D E. Dlib-ml: A Machine Learning Toolkit[M]. JMLR.org, 2009.
- [17] Cortes C, Vapnik V. Support-vector networks[J]. Machine Learning, 1995, 20(3):273-297.

- 
- [18] Lawrence S, Giles C L, Tsoi A C. Convolutional neural networks for face recognition[C]// Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on. IEEE, 1997:217-222.
- [19] Suleiman A, Sze V. Energy-efficient HOG-based object detection at 1080HD 60 fps with multi-scale support[C]// Signal Processing Systems. IEEE, 2014:1-6.
- [20] Alabort-I-Medina J, Antonakos E, Booth J, et al. Menpo: A Comprehensive Platform for Parametric Image Alignment and Visual Deformable Models[C]// ACM International Conference on Multimedia. ACM, 2014:679-682.
- [21] Kluyver T, Ragan-Kelley B, Pérez F, et al. Jupyter Notebooks – a publishing format for reproducible computational workflows[C]// International Conference on Electronic Publishing. 2016.
- [22] Hunter J D. Matplotlib: A 2D Graphics Environment[J]. Computing in Science & Engineering, 2007, 9(3):90-95.
- [23] Belhumeur P N, Jacobs D W, Kriegman D J, et al. Localizing Parts of Faces Using a Consensus of Exemplars[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2011:545-552.
- [24] Phillips P J, Wechsler H, Huang J, et al. The FERET database and evaluation procedure for face-recognition algorithms[J]. Image & Vision Computing J, 1998, 16(5):295-306.
- [25] Gross R, Matthews I, Cohn J, et al. Multi-PIE[J]. Image & Vision Computing, 2010, 28(5):807-813.
- [26] Messer K. XM2VTSDB : The Extended M2VTS Database[J]. Proc.intl.conf.on Audio & Video Based Biometric Person Authentication, 1999:72--77.
- [27] Phillips P J, Flynn P J, Scruggs T, et al. Overview of the Face Recognition Grand Challenge[C]// IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2005:947-954.
- [28] Martinez A M. The AR face database[J]. Cvc Technical Report, 1998, 24.
- [29] Belhumeur P N, Jacobs D W, Kriegman D J, et al. Localizing Parts of Faces Using a Consensus of Exemplars[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2011:545-552.
- [30] Le V, Brandt J, Lin Z, et al. Interactive Facial Feature Localization[C]// European Conference on Computer Vision. Springer-Verlag, 2012:679-692.
- [31] Köstinger M, Wohlhart P, Roth P M, et al. Annotated Facial Landmarks in the Wild: A large-scale, real-world database for facial landmark localization[C]// IEEE International Conference on Computer Vision Workshops. IEEE, 2012:2144-2151.
- [32] Sagonas C, Tzimiropoulos G, Zafeiriou S, et al. 300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge[C]// IEEE International Conference on Computer Vision Workshops. IEEE Computer Society, 2013:397-403.
- [33] Alabort-I-Medina J, Antonakos E, Booth J, et al. Menpo: A Comprehensive Platform for Parametric Image Alignment and Visual Deformable Models[C]// ACM International Conference on Multimedia. ACM, 2014:679-682.

- 
- [34] Zafeiriou S, Trigeorgis G, Chrysos G, et al. The Menpo Facial Landmark Localisation Challenge: A Step Towards the Solution[C]// IEEE Conference on Computer Vision and Pattern Recognition Workshops. IEEE Computer Society, 2017:2116-2125.
- [35] Sagonas C, Antonakos E, Tzimiropoulos G, et al. 300 Faces In-The-Wild Challenge: database and results \*, \*\*[J]. Image & Vision Computing, 2016, 47:3-18.
- [36] Sagonas C, Tzimiropoulos G, Zafeiriou S, et al. A Semi-automatic Methodology for Facial Landmark Annotation[C]// Computer Vision and Pattern Recognition Workshops. IEEE, 2013:896-903.
- [37] Cootes T F, Taylor C J, Cooper D H, et al. Active Shape Models-Their Training and Application[J]. Computer Vision & Image Understanding, 1995, 61(1):38-59.
- [38] Cootes T F, Edwards G J, Taylor C J. Active Appearance Models[C]// European Conference on Computer Vision. Springer-Verlag, 1998:484-498.
- [39] Cristinacce D, Cootes T F. Feature Detection and Tracking with Constrained Local Models[C]// British Machine Vision Conference 2006, Edinburgh, Uk, September. DBLP, 2006:929-938.
- [40] Dollar P, Welinder P, Perona P. Cascaded pose regression[J]. IEEE, 2010, 238(6):1078-1085.
- [41] Sun Y, Wang X, Tang X. Deep Convolutional Network Cascade for Facial Point Detection[C]// Computer Vision and Pattern Recognition. IEEE, 2013:3476-3483.
- [42] Zhang K, Zhang Z, Li Z, et al. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks[J]. IEEE Signal Processing Letters, 2016, 23(10):1499-1503.
- [43] Kowalski M, Naruniec J, Trzcinski T. Deep Alignment Network: A Convolutional Neural Network for Robust Face Alignment[J]. 2017:2034-2043.
- [44] Kazemi V, Sullivan J. One millisecond face alignment with an ensemble of regression trees[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2014:1867-1874.
- [45] Gower J C. Generalized procrustes analysis[J]. Psychometrika, 1975, 40(1):33-51.
- [46] Pearson, K. On lines and planes of closest fit to systems of points in space[C]// The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science. 1901:559-572.
- [47] Wilms M, Handels H, Ehrhardt J. Representative Patch-based Active Appearance Models Generated from Small Training Populations[C]// International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Cham, 2017:152-160.
- [48] Matthews I, Baker S. Active Appearance Models Revisited[J]. International Journal of Computer Vision, 2004, 60(2):135-164.
- [49] Lucas B D, Kanade T. An iterative image registration technique with an application to stereo vision[C]// International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers Inc. 1981:674-679.
- [50] Baker S, Matthews I. Lucas-Kanade 20 Years On: A Unifying Framework[J]. International Journal of Computer Vision, 2004, 56(3):221-255.
- [51] Avriel M. Nonlinear Programming: Analysis and Methods[J]. 1976:54.
- [52] 吴怀宇. 3D 打印:三维智能数字化创造[M]. 电子工业出版社, 2015.
- [53] Blanz V. A morphable model for the synthesis of 3D faces[J]. Acm Siggraph, 1999:187-194.
- [54] Wold S, Esbensen K, Geladi P. Principal component analysis[J]. Chemom.intell.lab, 1987, 2(1):37-52.

- 
- [55] Paysan P, Knothe R, Amberg B, et al. A 3D Face Model for Pose and Illumination Invariant Face Recognition[C]// IEEE International Conference on Advanced Video & Signal Based Surveillance. IEEE Computer Society, 2009:296-301.
- [56] Gerig T, Morel-Forster A, Blumer C, et al. Morphable Face Models - An Open Framework[J]. 2017.
- [57] Booth J, Roussos A, Zafeiriou S, et al. A 3D Morphable Model Learnt from 10,000 Faces[C]// Computer Vision and Pattern Recognition. IEEE, 2016:5543-5552.
- [58] Cao C, Weng Y, Zhou S, et al. FaceWarehouse: A 3D Facial Expression Database for Visual Computing[J]. IEEE Transactions on Visualization & Computer Graphics, 2013, 20(3):1.
- [59] Huber P, Hu G, Tena R, et al. A Multiresolution 3D Morphable Face Model and Fitting Framework[C]// Visapp. 2016.
- [60] Bas A, Smith W A P, Bolkart T, et al. Fitting a 3D Morphable Model to Edges: A Comparison Between Hard and Soft Correspondences[C]// Asian Conference on Computer Vision. Springer, Cham, 2016:377-391.
- [61] Hartley R, Zisserman A. Multiple view geometry in computer vision. With foreword by Olivier Faugeras. 2nd edition[J]. 2003.
- [62] Aldrian O, Smith W A. Inverse rendering of faces with a 3D morphable model[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 35(5):1080-1093.
- [63] Tran A T, Hassner T, Masi I, et al. Regressing Robust and Discriminative 3D Morphable Models with a Very Deep Neural Network[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2017:1493-1502.
- [64] Piotraschke M, Blanz V. Automated 3D Face Reconstruction from Multiple Images Using Quality Measures[C]// Computer Vision and Pattern Recognition. IEEE, 2016:3418-3427.
- [65] Bulat A, Tzimiropoulos G. How Far are We from Solving the 2D & 3D Face Alignment Problem? (and a Dataset of 230,000 3D Facial Landmarks)[J]. 2017:1021-1030.
- [66] Farkas L G. Anthropometry of the head and face[J]. Annals of Occupational Hygiene, 1994, 52(4):773-782.
- [67] Sforza C, Laino A, D'Alessio R, et al. Three-dimensional facial morphometry of attractive children and normal children in the deciduous and early mixed dentition.[J]. Angle Orthodontist, 2007, 77(6):1025-1033.