



中山大學  
SUN YAT-SEN UNIVERSITY

# 本科生毕业论文（设计）

Undergraduate Graduation Thesis  
(Design)

题目 Title: 基于 Hu 矩和普氏距离的  
人脸轮廓匹配算法

院 系  
School (Department): 数据科学与计算机学院

专 业  
Major: 软件工程（数字媒体方向）

学生姓名  
Student Name: 杨柳

学 号  
Student No.: 12330366

指导教师(职称)  
Supervisor (Title): 曾坤（副教授）

时间： 二〇一六 年 五 月 四 日

Date: Month 5 Day 4 Year 2016

**表一：毕业论文（设计）开题报告**  
**Form 1: Research Proposal of Graduation Thesis**  
**(Design)**

论文（设计）题目

Thesis (Design) Title: 基于 Hu 矩和普氏距离的人脸轮廓匹配算法

**课题背景:**

在如今高速人脸识别是人类视觉中最突出的能力之一。近年来，自动人脸识别逐渐成熟，取得了广泛的应用。计算机人脸识别技术主要包括：人脸检测和定位，对人脸进行特征提取和识别(匹配)。人的脸型包含了很多特征信息，人脸轮廓匹配很大程度上决定了识别的成功与否。

本文中的算法是“微漫”人脸漫画仿真系统的一部分，该系统致力于使用漫画仿真出跟用户相似的肖像画。在“微漫”系统中，用户输入肖像照片，系统切割出五官与脸型轮廓等人脸部件，分别与素材库中已手工画好漫画的照片原图做匹配，使用相似度最高的照片的漫画做部件拼凑并对其进行位置大小调整

**课题目标:**

为了“微漫”系统能够得到一个更加相似的结果，需要我们找到一个适用于所有情况的效果较好的人脸轮廓匹配算法，在匹配算法中，我们使用用户输入头像的脸型轮廓与素材库中的轮廓做匹配，找到效果最好的那个用来做最后的漫画人脸拼凑。

本课题的目标就是找到一个合适的人脸轮廓匹配算法。在该算法中，将会单独实现两个轮廓匹配算法，并综合两种算法得到一个效果更好的匹配算法。

**研究计划:**

2015. 11. 21    -    2015. 12. 31

OpenCV (C++与 Python) 环境搭建

2016. 1. 1        -    2016. 2. 1

实现 Hu 矩匹配算法

2016. 2. 2        -    2016. 3. 1

实现普氏距离匹配算法

2016. 3. 2        -    2016. 3. 30

实现加权评分算法

2016.4.1 - 2016.4.20

编写并完善论文

使用技术:

OpenCV、C++、Python

参考文献:

- 1、王俊艳, 苏光大. 基于下颌轮廓线的人脸分类方法. 红外与激光工程. 2004, (2): 159-163
- 2、陈慧静, 夏小玲. 基于形状上下文的人脸匹配算法. 计算机工程. 2009, 35(3): 208-210
- 3、李启娟, 李金屏. 基于轮廓信息的人脸检测. 计算机技术与发展. 2008, 18(9): 108-118
- 4、冯军华. 人脸面部轮廓特征提取及在身份验证中的应用[M]. 太原科技大学, 2010

Student Signature:

Date:

指导教师意见

Comments from Supervisor:

1. 同意开题                      2. 修改后开题                      3. 重新开题  
1. Approved(      )              2. Approved after Revision (      )              3.  
Disapproved(      )

Supervisor Signature:

Date:

**表二：毕业论文（设计）过程检查情况记录表**  
**Form 2: Process Check-up Form**

**第 1 次检查 (First Check-up) :**

学生总结

Student Self-summary:

- 1、阅读已有的论文
- 2、分别实现了 Hu 矩匹配算法和普氏距离匹配算法

指导教师意见

Comments of Supervisor:

- 1、对两种匹配算法做不变性实验，观察并分析实验结果
- 2、思考如何将两种算法做综合，产生更好的匹配算法
- 3、加快毕设速度

**第 2 次检查 (Second Check-up):**

学生总结

Student Self-summary:

- 1、设计并实现了加权评分匹配算法
- 2、编写了论文的正文部分

指导教师意见

Comments of Supervisor:

- 1、流程图不规范，需要重新画
- 2、ASM 识别人脸图标标注点太大，须使用更小的点标注
- 3、实验结果输出不能使用截图，应用表格表示数据
- 4、普氏距离公式不容易理解，要采用更好的表示方式
- 5、加权评分算法效果不太好，应继续改善，注意两种算法之间的互补

**第 3 次检查 (Third Check-up):**

学生总结

Student Self-summary:

- 1、改善加权评分匹配算法，得到更好的测试结果
- 2、完成论文全部内容

指导教师意见

<p>Comments of Supervisor:</p> <p>1、按照学院网最新公告修改论文格式</p> <p>2、摘要部分应概括的更为详细</p>	
<p>学生签名 (Student Signature):</p> <p>指导教师签名 (Supervisor Signature):</p> <p>日期 (Date) :</p> <p>日期 (Date) :</p>	
<p>总体完成情况 (Overall Assessment)</p>	<p>指导教师意见 Comments of Supervisor:</p> <p>1、按计划完成，完成情况优 (Excellent): ( )</p> <p>2、按计划完成，完成情况良 (Good): ( )</p> <p>3、基本按计划完成，完成情况合格 (Fair): ( )</p> <p>4、完成情况不合格 (Poor): ( )</p> <p>指导教师签名 (Supervisor Signature):</p> <p>日期 (Date) :</p>

表三：毕业论文（设计）答辩情况登记表  
Form 3: Thesis Defense Performance Form

答辩人 Student Name		专 业 Major	
论文（设计）题目 Thesis (Design) Title			
答辩小组成员 Committee Members			
<div>答辩记录 Records of Defense Performance:</div> <div></div> <div>记录人签名 (Clerk Signature):</div> <div>日期 (Date) :</div>			

## 学术诚信声明

本人所呈交的毕业论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。本毕业论文的知识产权归属于培养单位。本人完全意识到本声明的法律结果由本人承担。

本人签名：

日期：

## Statement of Academic Integrity

I hereby acknowledge that the thesis submitted is a product of my own independent research under the supervision of my supervisor, and that all the data, statistics, pictures and materials are reliable and trustworthy, and that all the previous research and sources are appropriately marked in the thesis, and that the intellectual property of the thesis belongs to the school. I am fully aware of the legal effect of this statement.

Student Signature:

Date:

## 【摘 要】

人脸识别是人类视觉中最突出的能力之一。近年来,自动人脸识别逐渐成熟,取得了广泛的应用<sup>[1]</sup>。计算机人脸识别技术主要包括:人脸检测和定位,对人脸进行特征提取和识别(匹配)<sup>[2]</sup>。人的脸型包含了很多特征信息,人脸轮廓匹配很大程度上决定了识别的成功与否。

本文正是针对这轮廓匹配这一问题展开研究。本文中的算法是“微漫”人脸漫画仿真系统的一部分,该系统致力于使用漫画仿真出跟用户相似的肖像画。在“微漫”系统中,用户输入肖像照片,系统切割出五官与脸型轮廓等人脸部件,分别与素材库中已手工画好漫画的照片原图做匹配,使用相似度最高的照片的漫画做部件拼凑并对其进行位置大小调整。

本文首先介绍了两种常用的轮廓匹配算法——Hu 矩匹配算法和普氏距离匹配算法,并通过实验验证了算法的鲁棒性,通过观察实验结果客观评价了算法的优缺点。接着,综合两种算法,创新出一个新的匹配算法——加权评分匹配算法,该算法同时使用 Hu 矩匹配算法和普氏距离匹配算法进行匹配,并对匹配结果排序打分并进行加权整合,最终得到一个最好的匹配结果。该算法综合两种现有有效的匹配算法,达到了更好的匹配效果。

本文把人脸轮廓概括为一些点的集合,从而把形状匹配问题转化为点集之间的匹配问题,每个匹配算法先提取点集内的特征信息,再将点集的特征信息相互比较(匹配)。

最后,本文对本文的工作进行了总结,指出了本文的贡献点和不足。针对本文的不足之处,分析并讨论了下一步的研究计划,给出了可行的改进方案。

**【关键词】** 人脸;轮廓;匹配;漫画;仿真



## [ABSTRACT]

Face recognition is one of the most prominent human visual ability. In recent years, automatic face recognition, which is more and more mature, has been applied widely<sup>[1]</sup>. Technologies of face recognition include: face detection, face location, human face feature extraction and recognition(matching)<sup>[2]</sup>. The outline of human' s face contains lots of feature information, so to a large extent, the success of recognition is determined by the matching of facial contour.

This paper focuses on the matching of facial contour. The algorithms in this paper are part of the “Wei Man” Comic Human Face Simulation System, which aims at using cartoon drawing to simulate a portrait of user. In the “Wei Man” system, user inputs portrait photo, then the system cuts out facial components such as facial features and facial contour. We compare these components with the photos that have been drawn. Finally, we choose the most similar components and put the cartoon drawing of them together into a complete face.

In this paper, at first two common contour matching algorithms, Hu Moment Matching Algorithm and Procrustes Distance Matching Algorithm, are introduced. We use experiments to verify that these two algorithms are effective and robust and observe experiment results to give an objective evaluation about the strengths and weaknesses of algorithms. Then, combining the two algorithms , we create a new matching algorithm called Weighted Score Matching Algorithm. This algorithm runs Hu Moment Matching Algorithm and Procrustes Distance Matching Algorithm at the same time, sorts the matching results, scores every result according to the order, carries on the weighting conformity and gets the most best matching result finally. Weighted Score Matching Algorithm combines the two existing efficient matching algorithms together and achieves a much better matching effect.

In this paper, the face contour is summarized as a set of points. Thus, the contour matching problem is transformed into the matching problem between two point sets. Each

matching algorithm first extracts the feature information from the point sets, and then compares the feature information of each set(matching).

Finally, this paper makes a summary, pointing out the contribution points and deficiencies. Based on the results and deficiencies, we analyze and discuss about the next research plan and give a feasible improvement scheme.

**[Keywords]** Human Face; Facial Contour; Matching; Comic; Simulation

# 目录

<b>第一章 概述</b>	<b>3</b>
1.1 人脸轮廓匹配问题的背景和意义	3
1.2 人脸轮廓匹配问题的描述	3
1.3 本文的工作	6
1.4 论文结构简介	6
<b>第二章 HU 矩匹配与普氏距离匹配算法综述</b>	<b>7</b>
2.1 HU 矩匹配算法	7
2.1.1 HU 矩概述	7
2.1.2 计算 HU 相似度	8
2.1.3 HU 矩不变性分析	9
2.1.4 HU 矩评价	11
2.2 普氏距离匹配算法	12
2.2.1 普氏距离概述	12
2.2.2 普氏距离计算	14
2.2.3 普氏距离不变性分析	15
2.2.4 普氏距离评价	16
<b>第三章 提出加权评分匹配的方法</b>	<b>18</b>
3.1 单个算法匹配	18
3.2 加权评分匹配	27
<b>第四章 加权评分匹配算法</b>	<b>28</b>
4.1 加权评分匹配算法概述	28
4.2 加权评分算法实现	28
4.3 加权评分匹配算法实验结果	29
4.4 加权评分匹配算法分析	31
<b>第五章 总结与展望</b>	<b>32</b>
5.1 本文工作的总结	32
5.1.1 本文的工作	32
5.1.2 本文工作的不足	32

5.2 未来工作的展望 .....	32
5.2.1 改进原有的不足 .....	32
5.2.2 进一步的研究方向.....	33
<b>参考文献</b> .....	<b>35</b>
<b>致 谢</b> .....	<b>34</b>

## 第一章 概述

### 1.1 人脸轮廓匹配问题的背景和意义

随着计算机技术的发展，人的身份认证显得越来越重要<sup>[3]</sup>。由于人脸图像在非接触环境下容易获取，人脸识别对用户无侵害且具有隐蔽性<sup>[4]</sup>，因此广泛应用于身份验证、视频监视、人机交互和图像数据库的检索等领域，受到了人们极大的重视与关注<sup>[5]</sup>。所谓人脸识别，是指对输入的人脸图像或者视频，判断其中是否存在人脸，如果存在人脸，则进一步给出每张人脸的位置、大小和各个主要面部器官的位置信息，并且依据这些信息，进一步提取每张人脸蕴含的身份特征，并将其与已知人脸库中的人脸进行对比，从而识别每张人脸的身份，进而达到身份验证的目的<sup>[4]</sup>。人脸轮廓匹配是人脸识别中的一个十分重要的步骤与前提，是当前一个非常活跃的研究课题。

“微漫”人脸漫画仿真系统是一个使用漫画仿真出跟用户相似肖像的系统，类似于“脸萌”，但是与脸萌不同的是用户需要输入个人照片，在进行少量用户交互的情况下输出漫画人脸。这就要求我们使用更加智能的算法来对用户图片和漫画素材进行分析、匹配与调整。轮廓匹配算法就是其中一个较为关键的算法。

### 1.2 人脸轮廓匹配问题的描述

本文所讲的人脸轮廓匹配算法是“微漫”人脸漫画仿真系统中的一部分，用于匹配用户输入照片中的人脸轮廓与标准素材。在“微漫”人脸漫画仿真系统的素材库中，我们手工画了很多标准人脸的漫画用于匹配用户上传的照片，如图 1-1 和 1-2 所示：

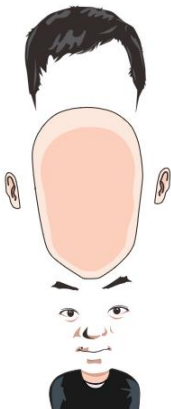


图 1-1 手工漫画图

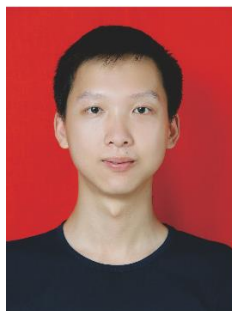


图 1-2 手工漫画图

系统的流程图如图 1-3 所示：

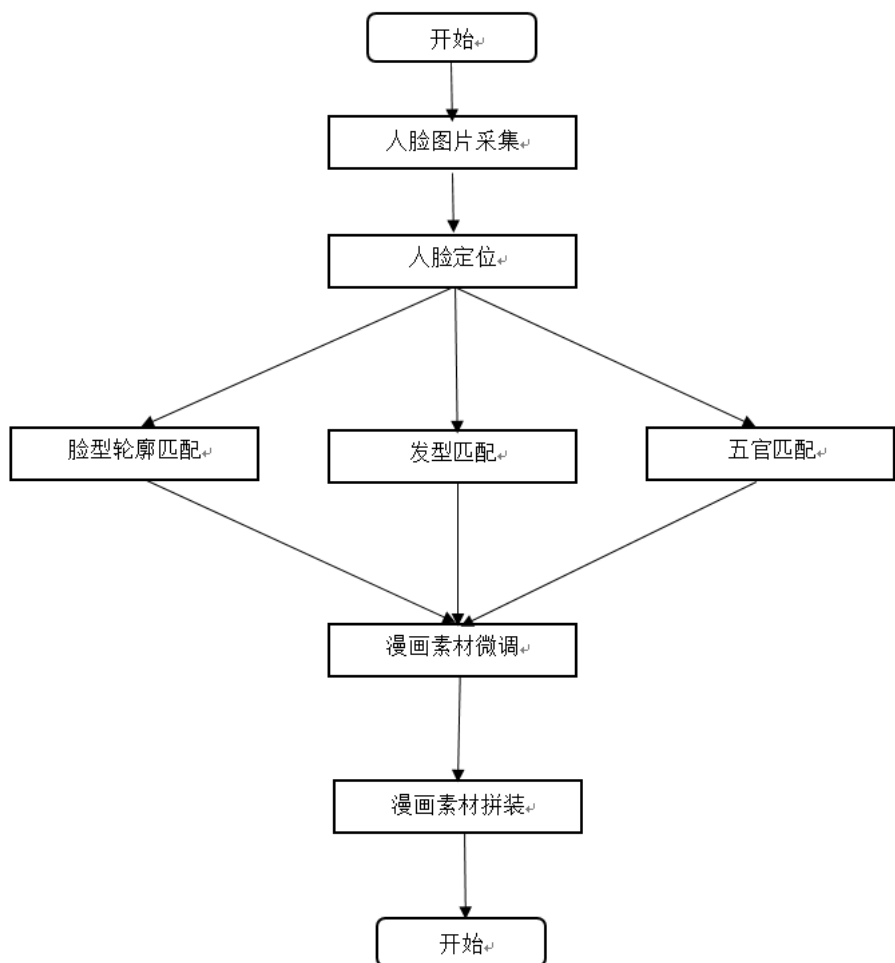


图 1-3 “微漫”系统流程图

我们首先使用 ASM 算法进行人脸定位，获得描述人脸的 77 个点，如图 1-4 和 1-5 所示：



图 1-4 人脸输入原图

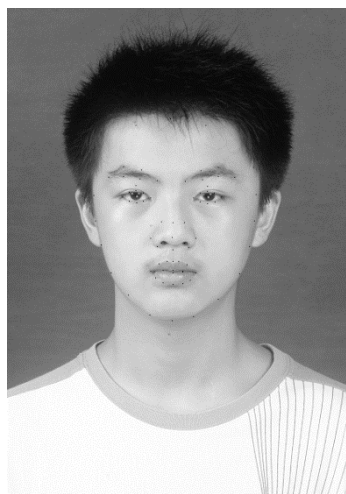


图 1-5 ASM 算法定位图

使用 ASM 算法得出的 77 个点，我们可以将五官和脸型轮廓描述出来，即：

- 1) 0-15 描述脸型轮廓
- 2) 16-27 描述眉毛
- 3) 28-47 描述眼睛
- 4) 48-58 描述鼻子
- 5) 59-76 描述嘴巴

接着我们将用户的五官脸型与素材库中的人脸原图进行匹配，选出匹配度最高的，拿对应的漫画素材（五官与轮廓）进行用户人脸拼装还原。五官匹配算法（LBP）与本文介绍算法原理不同，故在本文中不作详细描述。

人脸轮廓匹配包含两个部分，一个是人脸轮廓特征提取，另一个是人脸轮廓特征的匹配。输入的参数就是上文提过的 0-15 描述点，如图 1-6 所示：



图 1-6 轮廓匹配输入描述点

既然使用 16 个描述点代替了人脸轮廓，那么问题就转换成了用户输入的 16 个描述点与素材库中每个图片的 16 个描述点的匹配问题，目的是在素材库中匹配出与用户最为相似的一个人脸轮廓。

对于两组点之间的匹配而言，有以下三个难点与挑战：

- 1) 16 个点描述的轮廓过于粗略，但是由于 ASM 给出的描述点的个数是固定的，我们只能寻找其他方式来对轮廓细节进行详细描述，对现有点做插值就是方法之一。
- 2) 提取描述点中的特征时，既要考虑提取的特征能否包含轮廓中的关键信息，更要考虑所提取因子的不变性（旋转、缩放、平移不变性）
- 3) 使用所提取的特征进行匹配时，需要考虑匹配的方式（相似度还是差别值），

这个多半和特征点的选取有关，而且两者之间大多可以相互转换。

综上所述，对于轮廓匹配，其关键点在于提取特征的种类与方式。

### 1.3 本文的工作

本文分开描述了轮廓匹配的两种算法——Hu 矩匹配与普氏距离匹配，这两种算法均是形状匹配的算法，但用于人脸轮廓匹配的情况却不多，因为尽管两种算法优点突出，但其本身都或多或少存在缺陷，在此我综合了两种轮廓匹配的算法，使得两种算法的优缺点可以相互补充，最终获得了不错的结果。

### 1.4 论文结构简介

- 1) 概述，本章主要阐述了论文的研究背景和意义，对算法使用的系统做了介绍，综述了算法解决的问题，并简要介绍了本文的主要工作和内容安排。
- 2) Hu 矩匹配与普氏距离匹配算法综述，本章主要介绍了前人已经用于解决轮廓匹配问题的两种算法，并分别对两种算法做了实验，分析其旋转、缩放与平移不变性。
- 3) 提出加权评分匹配的方法，本章首先分别对 Hu 矩匹配算法和普氏距离匹配算法进行实验，使用微调的原始数据与原始数据做匹配，接着观察并分析实验结果，结合实际情况提出加权评分匹配算法的概念。
- 4) 加权评分匹配算法，本章解释了加权评分匹配算法的概念，详细描述了算法实现的过程，并对加权评分算法惊醒了第三章中的实验，分析实验结果，结合第三章评价算法效果。
- 5) 总结与展望，本章对全文的工作进行了总结，指出了本文的贡献点和不足。针对存在的不足，给出了可行的改进方案，提出了进一步的研究方向。



## 第二章 Hu 矩匹配与普氏距离匹配算法综述

### 2.1 Hu 矩匹配算法

#### 2.1.1 Hu 矩概述

几何矩和中心矩是由 Ming-Kuei Hu 在 1962 年提出的。在连续情况下，图像定义为  $f(x, y)$ ，

$p+q$  阶的几何矩（标准矩）定义为：

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad p, q = 0, 1, 2, \dots \quad (2.1)$$

$p+q$  阶中心矩定义为：

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - x_0)^p (y - y_0)^q f(x, y) dx dy \quad p, q = 0, 1, 2, \dots \quad (2.2)$$

其中矩心  $x_0, y_0$  为：

$$x_0 = \frac{m_{10}}{m_{00}}, \quad y_0 = \frac{m_{01}}{m_{00}} \quad (2.3)$$

对于数字图像处理领域，图像一般都是离散的，在这种情况下，

$p+q$  阶的几何矩定义为：

$$m_{pq} = \sum_{x=1}^M \sum_{y=1}^N x^p y^q f(x, y) \quad p, q = 0, 1, 2, \dots \quad (2.4)$$

$p+q$  阶中心矩定义为：

$$\mu_{pq} = \sum_{x=1}^M \sum_{y=1}^N (x - x_0)^p (y - y_0)^q f(x, y) \quad p, q = 0, 1, 2, \dots \quad (2.5)$$

归一化的中心矩定义为：

$$\gamma_{pq} = \frac{\mu_{pq}}{\mu_{00}^r} \quad (2.6)$$

式中：

$$r = \frac{p + q + 2}{2} \quad p + q = 2, 3, 4, \dots \quad (2.7)$$

利用二阶和三阶的归一化中心矩构造了七个不变矩：

$$I_1 = \gamma_{20} + \gamma_{02} \quad (2.8)$$

$$I_2 = (\gamma_{20} - \gamma_{02})^2 + 4\gamma_{11}^2 \quad (2.9)$$

$$I_3 = (y_{30} - 3y_{12})^2 + (3y_{21} - y_{03})^2 \quad (2.10)$$

$$I_4 = (y_{30} + y_{12})^2 + (y_{21} + y_{03})^2 \quad (2.11)$$

$$I_5 = (y_{30} - 3y_{12})(y_{30} + y_{12})[(y_{30} + y_{12})^2 - 3(y_{21} + y_{03})^2] \\ + (3y_{21} - y_{03})(y_{21} + y_{03})[3(y_{30} + y_{12})^2 - (y_{21} + y_{03})^2] \quad (2.12)$$

$$I_6 = (y_{20} - y_{02})[(y_{30} + y_{12})^2 - (y_{21} + y_{03})^2] + 4y_{11}(y_{30} + y_{12})(y_{21} + y_{03}) \quad (2.13)$$

$$I_7 = (3y_{21} - y_{03})(y_{30} + y_{12})[(y_{30} + y_{12})^2 - 3(y_{21} + y_{03})^2] - (y_{30} - 3y_{12})(y_{12} \\ + y_{03})[3(y_{30} + y_{12})^2 - (y_{21} + y_{03})^2] \quad (2.14)$$

这 7 个不变矩构成一组特征量，并且 Hu 在同一篇论文<sup>[6]</sup>中证明了它们具有旋转、放缩和平移不变性。

### 2.1.2 计算 Hu 相似度

得到 7 个 Hu 不变矩之后，要使用这七个数值计算两个轮廓之间的相似度。计算两组不变矩相似度的算法有很多种，在这里我简单列举两种。

算法一：

```
Initialize: I_a = orthogonal invariants of a
           I_b = orthogonal invariants of b
           sigma_ab = 0
           sigma_a = 0
           sigma_b = 0
for i = 0 : 6
    tmp = fabs(I_a[i] * I_b[i])
    sigma_ab += tmp
    sigma_a += pow(I_a[i], 2)
    sigma_b += pow(I_b[i], 2);
end
similarity = sigma_ab / (sqrt(sigma_a) * sqrt(sigma_b))
```

算法二：

```
Initialize: I_a = orthogonal invariants of a
           I_b = orthogonal invariants of b
           tmp_1 = 0
           tmp_2 = 0
for i = 0 : 6
    tmp_1 += fabs(I_a[i] - I_b[i])
    tmp_2 += fabs(I_a[i] + I_b[i])
end
similarity = 1 - (tmp_1 * 1.0) / (tmp_2)
```

### 2.1.3 Hu 矩不变性分析

为了验证 Hu 矩的旋转、放缩与平移不变性，我做了以下实验：

画出一个椭圆（A），在椭圆上均匀取点（16 个），如图 2-1 所示：

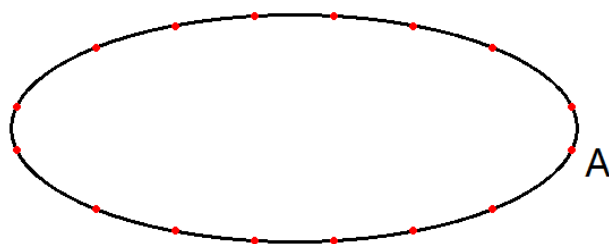


图 2-1 原椭圆 A

将椭圆 A 绕椭圆中心顺时针一次旋转  $15^\circ$ 、 $30^\circ$ 、 $45^\circ$ 、 $60^\circ$ 、 $75^\circ$  和  $90^\circ$ ，分别均匀取点并计算 Hu 矩与椭圆 A 比较，得出相似程度，如图 2-2 和表 2-1 所示：

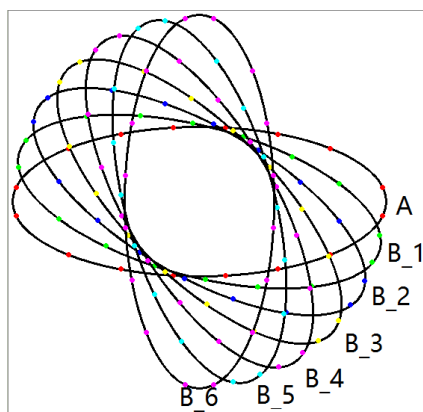


图 2-2 旋转后的椭圆

表 2-1 旋转后椭圆和原椭圆的 Hu 矩相似度

旋转后的椭圆	与原椭圆的 Hu 相似度
B_1	0.999961
B_2	0.999793
B_3	0.996088
B_4	0.999793
B_5	0.999961
B_6	1

将椭圆 A 的坐标分别乘以 0.4、0.8、1.2、1.6 和 2.0 做放缩操作，分别均匀取点并计算 Hu 矩与椭圆 A 比较，得出相似程度，如图 2-3 和表 2-2 所示：

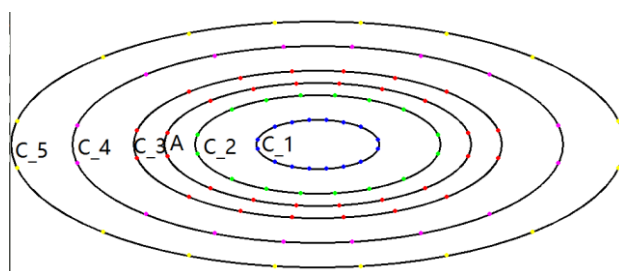


图 2-3 放缩后的椭圆

表 2-2 放缩后椭圆和原椭圆的 Hu 矩相似度

放缩后的椭圆	与原椭圆的 Hu 相似度
C_1	0.0502207
C_2	0.54532
C_3	0.000344273
C_4	0.0959548
C_5	0.117753

将椭圆 A 分别向上、下、左、右四个方向做平移，分别均匀取点并计算 Hu 矩与椭圆 A 比较，得出相似度，如图 2-4 和表 2-3 所示：

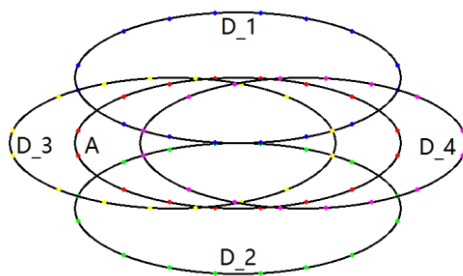


图 2-4 平移后的椭圆

表 2-3 平移后椭圆和原椭圆的 Hu 矩相似度

平移后的椭圆	与原椭圆的 Hu 相似度
D_1	1
D_2	1
D_3	1
D_4	1

从实验结果可以看出，Hu 矩对于旋转和平移的不变性非常好，但是缩放不变性的效果比较差。关于这一点，陈瑞森先生在论文《应用于数字图像识别的 Hu 矩缩放不变性分析》<sup>[7]</sup>中使用数学推导进行了证明。

#### 2.1.4 Hu 矩评价

有关 Hu 矩匹配的优缺点，请见表 2-4：

表 2-4 Hu 矩匹配的优缺点

Hu 匹配优点	Hu 匹配缺点
算法复杂度低，识别速度快。	识别率比较低。
在数字图像中的旋转、平移不变性良好。	在数字图像中不具有放缩不变性。
$I_1$ 到 $I_7$ 七个不变矩，从粗略到细节描述了轮廓形状，含盖信息较为全面。	实际上，在对图片中物体的识别过程中，只有 $I_1$ 和 $I_2$ 不变性保持的比较好，其他的几个不变矩带来的误差比较大，有学者认为只有基于二阶矩的不变矩对二维物体的描述才是真正的具有旋转、缩放和平移不变性。

## 2.2 普氏距离匹配算法

### 2.2.1 普氏距离概述

欧氏距离 (Euclidean Distance) 是一个通常采用的距离定义，如图 2-5 所示：

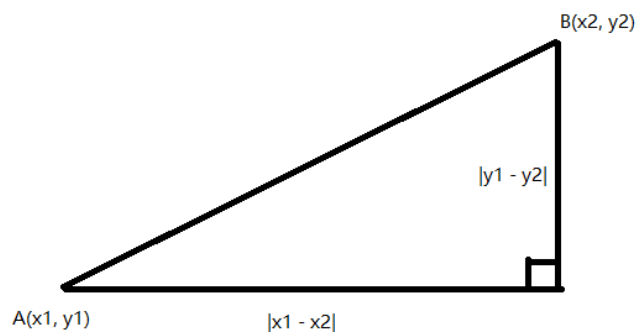


图 2-5 欧氏距离示意图

指在  $m$  维空间中两个点之间的真实距离，或者向量的自然长度（即该点到原点的距离）。在二维空间中，A、B 两点之间的欧氏距离的公式为：

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.15)$$

普氏距离 (Procrustes Distance) 就是两个轮廓上对应描述点之间欧氏距离的和<sup>[9]</sup>，如图 2-6 所示：

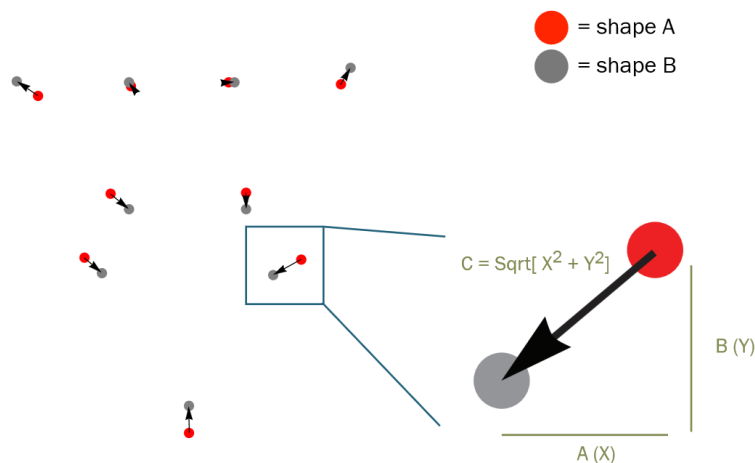


图 2-6 普氏距离示意图

但在实际应用中，想要比较两个脸型轮廓的普氏距离，却没有只是求解欧氏距离这么简单。首先，计算两个轮廓普氏距离之前，我们做了三个假设：

- 1) 两个轮廓分别对应两组点
- 2) 两组点之间存在一对一的关联

### 3) 点之间的关联是给定的

接着，我们需要引入形状（shape）的概念，形状是大量与平移、旋转和放缩无关的点的集合。因为普氏距离的输入为形状，所以我们需要对轮廓描述点做一些预处理，去掉其中的相似性变换（Similarity Transformation），这个预处理包含了三个步骤：

#### 1) 移除平移变换

将描述点中心平移到坐标原点。

#### 2) 移除放缩变换

假设描述点的数量为  $n$ ，坐标为  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ ，现将所有点归一化：

$$\sum_{i=1}^n x_i^2 + y_i^2 = 1 \quad (2.16)$$

我们可以想象一下，这组点的坐标是位于一个  $2(n-1)$  维度空间的单位超球面上的。经过前两步的操作，我们得到了两组归一化点（Normalized Points） $U$  和  $X$ 。到这里我们只完成了预定型（pre-shape），因为还未移除旋转变换。

#### 3) 移除旋转变换

如果我们考虑一组归一化点的所有旋转可能，可以形成一个一维闭合曲线。

我们可以选择固定其中一组点，旋转另一组。

在以上前提下，定义了三种普氏距离<sup>[9]</sup>：

#### 1) 全普氏距离（Full Procrustes Distance）

$$D_F = \min(s, \Theta) \|U - sXR(\Theta)\| \quad (2.17)$$

找到一个对轮廓  $X$  的旋转  $R(\Theta)$  和放缩  $s$  操作，使得轮廓  $X$  与轮廓  $U$  之间的欧氏距离最小，这时轮廓  $X$  与轮廓  $U$  之间的欧式距离就是全普氏距离。

#### 2) 偏普氏距离（Partial Procrustes Distance）

$$D_P = \min(\Theta) \|U - XR(\Theta)\| \quad (2.18)$$

找到一个对轮廓  $X$  的旋转  $R(\Theta)$  操作，使得轮廓  $X$  与轮廓  $U$  之间的欧氏距离最小，这时轮廓  $X$  与轮廓  $U$  之间的欧式距离就是偏普氏距离。

#### 3) 普氏距离（Procrustes Distance）

$$\rho = \min(d) \|U - X\| \quad (2.19)$$

对轮廓 X 进行旋转操作，使得在球面上轮廓 X 与轮廓 U 之间的测地线距离（Geodesic Distance）最小，这时轮廓 X 与轮廓 U 之间的欧式距离就是偏普氏距离。

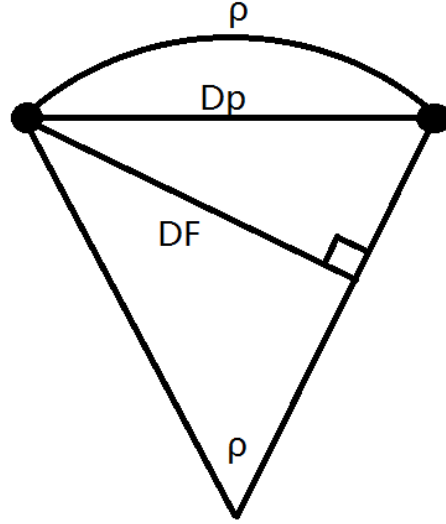


图 2-7 三种普氏距离示意

图 2-7 对三种普氏距离做了一个简单的解释，图中是超球面的一个切面，A、B 两点之间的三中普氏距离已标出。

### 2.2.2 普氏距离计算

本文所介绍匹配算法使用的是全普氏距离。

对于二维的平移、放缩与旋转变换计算，我们可以使用矩阵乘法来表示，如图 2-8 所示：

$$\begin{pmatrix} u_1 & u_2 & \cdot & \cdot & \cdot & u_n \\ v_1 & v_2 & & & & v_n \end{pmatrix} = s \begin{pmatrix} \cos \theta & \sin \theta & t_x \\ -\sin \theta & \cos \theta & t_y \end{pmatrix} \begin{pmatrix} x_1 & x_2 & \cdot & \cdot & \cdot & x_n \\ y_1 & y_2 & & & & y_n \\ 1 & 1 & & & & 1 \end{pmatrix}$$

$$\begin{pmatrix} a & b & t_x \\ -b & a & t_y \end{pmatrix} \begin{pmatrix} x_1 & x_2 & \cdot & \cdot & \cdot & x_n \\ y_1 & y_2 & & & & y_n \\ 1 & 1 & & & & 1 \end{pmatrix}$$

with  $a = s \cos \theta$ ,  $b = s \sin \theta$

图 2-8 矩阵乘法表示平移、缩放与旋转变换<sup>[8]</sup>

X 经过三种变换得到 X'，计算 U 与 X' 的欧氏距离，当欧氏距离最小时，该欧氏距离就为轮廓 X 与 U 之间的普氏距离。



在 Graham McNeill 与 Sethu Vijayakumar 的论文《2D Shape Classification and Retrieval》中，给出了计算两个轮廓之间全普氏距离的公式<sup>[10]</sup>：

$$D_F(z, w) = 1 - \frac{|w^*z|^2}{w^*wz^*z} \quad (2.20)$$

公式中  $z$  和  $w$  是复数集合：

$$z_j = x_j + iy_j \quad j = 0, 1, 2, \dots, n \quad (2.21)$$

$$w_j = u_j + iv_j \quad j = 0, 1, 2, \dots, n \quad (2.22)$$

$w^*$ 、 $z^*$ 分别是  $w$  与  $z$  的共轭转置。

### 2.2.3 普氏距离不变性分析

与 Hu 矩相同，我做了同样的实验来验证普氏距离的旋转、放缩与平移不变性。

画出一个椭圆（A），在椭圆上均匀取点（16 个），如图 2-1 所示。

将椭圆 A 绕椭圆中心顺时针一次旋转  $15^\circ$ 、 $30^\circ$ 、 $45^\circ$ 、 $60^\circ$ 、 $75^\circ$  和  $90^\circ$ ，分别均匀取点并计算与椭圆 A 之间的普氏距离，得出两者差距，如图 2-2 和表 2-5 所示：

表 2-5 旋转后椭圆和原椭圆的普氏距离差距

旋转后的椭圆	与原椭圆的普氏距离差距
B_1	5.84126e-006
B_2	1.90735e-005
B_3	2.5034e-006
B_4	2.13385e-005
B_5	5.42402e-006
B_6	1.19209e-005

将椭圆 A 的坐标分别乘以 0.4、0.8、1.2、1.6 和 2.0 做放缩操作，分别均匀取点并计算与椭圆 A 之间的普氏距离，得出两者差距，如图 2-4 和表 2-6 所示：

表 2-6 放缩后椭圆和原椭圆的普氏距离差距

放缩后的椭圆	与原椭圆的普氏距离差距
C_1	8.96454e-005
C_2	3.52263e-005
C_3	1.36495e-005
C_4	6.79493e-006
C_5	3.63588e-006

将椭圆 A 分别向上、下、左、右四个方向做平移，分别均匀取点并计算与椭圆 A 之间的普氏距离，得出两者差距，如图 2-6 和表 2-7 所示：

表 2-7 平移后椭圆和原椭圆的普氏距离差距

平移后的椭圆	与原椭圆的普氏距离差距
D_1	1.10865e-005
D_2	8.88109e-006
D_3	7.21216e-006
D_4	1.29938e-005

从实验结果可以看出，与 Hu 矩不同的是，普氏距离对于旋转、缩放和平移的不变性都非常好，这个很大程度上得益于我们对描述点的预处理。

#### 2.2.4 普氏距离评价

有关普氏距离匹配的优缺点，请见表 2-8：

表 2-8 普氏距离匹配的优缺点

普氏距离匹配优点	普氏距离匹配缺点
算法复杂度低，计算速度快。	没有提取物体特征，不包含物体信息，无法进一步对其进行分析。
在数字图像中的旋转、放缩与平移不变性良好。	在进行计算之前，必须先找到描述点之间的对应关系，导致无法适应部分特殊匹配。
检验两个物体是否匹配，自然会想到计算两者差别，这时使用普氏距离就最为自然合适。	

### 第三章 提出加权评分匹配的方法

#### 3.1 单个算法匹配

在轮廓匹配的算法中，没有任何一个是可以完美适配所有情况的，不同算法的侧重方面也都是不同的，使用单个匹配算法对轮廓进行匹配或多或少都会存在问题。

对于素材库中的所有图像(共 45 张)，使用 ASM 算法求出每个轮廓的描述点坐标，记为数据集 $C_1$ ，如表 3-1 所示：

表 3-1 所有轮廓描述点的坐标

$C_1$ 中轮廓标号	轮廓坐标
1	163 226 170 255 177 282 189 311 212 336 235 353 258 355 284 346 307 322 321 292 327 265 329 241 331 213 274 144 235 138 199 147
2	149 319 149 352 152 384 158 420 181 461 216 489 247 495 276 487 303 458 318 420 324 387 327 356 327 323 286 232 241 222 197 230
3	117 222 120 254 126 283 139 314 157 341 180 365 205 372 230 367 258 346 283 316 295 282 300 251 302 219 246 139 205 132 164 140
...	...
45	99 218 101 257 107 293 119 330 142 361 174 386 206 393 235 387 267 365 293 332 305 297 312 260 315 218 250 130 205 121 160 130

对其中的部分坐标做调整（微调，不会对大致轮廓造成影响）之后，我们得到了另一组数据，记为数据集 $C_2$ ，如表 3-2 所示：

表 3-2 调整后的描述点坐标

$C_2$ 中轮廓标号	轮廓坐标
1	164 226 169 252 176 279 190 310 212 334 235 353 258 356 283 348 306 323 324 293 331 266 334 241 336 212 273 141 233 136 195 146
2	150 318 151 352 156 385 165 421 188 453 219 479 247 484 273 477 296 450 312 420 321 387 325 355 327 321 285 233 241 222 197 230
3	109 221 113 253 119 282 132 313 154 342 179 365 204 372 228 367 255 345 281 316 293 281 299 250 301 218 247 138 206 132 166 140
...	...
45	99 217 101 255 107 291 119 328 143 358 174 384 206 392 234 386 266 363 294 331 307 296 313 259 316 217 251 128 206 119 160 128

如果我们单独使用 Hu 矩或普氏距离对数据集 $C_1$ 中的 45 个轮廓的描述点进行两两匹配，总能成功匹配到原轮廓，匹配正确率为 100%，匹配结果如表 3-3 和表 3-4 所示：

表 3-3  $C_1$ 对 $C_1$ 单独 Hu 矩匹配算法匹配结果

$C_1$ 中轮廓	匹配到的 $C_1$ 中 轮廓	相似度		$C_1$ 中轮廓	匹配到的 $C_1$ 中 轮廓	相似度
0	0	1		23	23	1
1	1	1		24	24	1
2	2	1		25	25	1
3	3	1		26	26	1
4	4	1		27	27	1
5	5	1		28	28	1
6	6	1		29	29	1
7	7	1		30	30	1
8	8	1		31	31	1
9	9	1		32	32	1
10	10	1		33	33	1
11	11	1		34	34	1
12	12	1		35	35	1
13	13	1		36	36	1
14	14	1		37	37	1
15	15	1		38	38	1
16	16	1		39	39	1
17	17	1		40	40	1
18	18	1		41	41	1
19	19	1		42	42	1
20	20	1		43	43	1
21	21	1		44	44	1
22	22	1				
匹配正确率				100%		

表 3-4  $C_1$ 对 $C_1$ 单独普氏距离算法匹配结果

$C_1$ 中轮廓	匹配到的 $C_1$ 中轮廓	差距	$C_1$ 中轮廓	匹配到的 $C_1$ 中轮廓	差距
0	0	1.32918e-005	23	23	2.18749e-005
1	1	1.23382e-005	24	24	3.25441e-005
2	2	1.23978e-005	25	25	9.95398e-006
3	3	4.35114e-006	26	26	8.46386e-006
4	4	8.70228e-006	27	27	1.12653e-005
5	5	3.78489e-005	28	28	3.20673e-005
6	6	5.90086e-006	29	29	1.44243e-005
7	7	1.14441e-005	30	30	9.05991e-005
8	8	9.95398e-006	31	31	1.97887e-005
9	9	4.8995e-005	32	32	1.2219e-005
10	10	9.31025e-005	33	33	6.49691e-006
11	11	9.23872e-006	34	34	1.28746e-005
12	12	1.055e-005	35	35	1.35899e-005
13	13	1.0848e-005	36	36	1.32322e-005
14	14	1.29342e-005	37	37	1.38283e-005
15	15	1.44839e-005	38	38	1.508e-005
16	16	5.90086e-006	39	39	1.19209e-005
17	17	9.29832e-006	40	40	9.71556e-006
18	18	8.22544e-006	41	41	1.26362e-005
19	19	2.08616e-005	42	42	1.30534e-005
20	20	3.77297e-005	43	43	1.42455e-005
21	21	8.52346e-006	44	44	9.29832e-006
22	22	7.15256e-006			
匹配正确率			100%		

为了查看相似脸型之间的匹配结果，检验算法的鲁棒性，我们将数据集 $C_1$ 中的轮廓描述点作为素材库，逐行输入数据集 $C_2$ 中的轮廓描述点，从素材库中寻找最匹配项。

对于数据集 $C_2$ 中的每组轮廓描述点，单独使用 Hu 矩匹配素材库(数据集 $C_1$ )中的最匹配项，匹配结果如表 3-5 与图 3-1 所示。

同样对于数据集 $C_2$ 中的每组轮廓描述点，单独使用普氏距离匹配素材库(数据集 $C_1$ )中的最匹配项，匹配结果如表 3-6 与图 3-2 所示。

可以看到，单独 Hu 矩匹配或普氏距离匹配的正确率均约为 53%，刚刚到达半数，表示有一半 $C_2$ 中的轮廓都能匹配到 $C_1$ 中原本的轮廓。当我们仔细观察轮廓比较图时，我们有了以下两个发现：

- 1) 在大部分匹配错误情况下，尽管有的调整后轮廓无法匹配到原轮廓，但其匹配到的结果相似度是很高的，表明匹配准确率受到了素材的影响，素材中脸型区分度不大。
- 2) 虽然 Hu 矩匹配和普氏距离匹配的正确率都为 53%，但是两者匹配的情况是不同的，即同样的轮廓，使用 Hu 矩匹配与使用普氏距离匹配到的结果可能不同。



表 3-5  $C_2$ 对 $C_1$ 单独 Hu 矩匹配算法匹配结果

$C_2$ 中轮廓	匹配到的 $C_1$ 中轮廓	相似度	$C_2$ 中轮廓	匹配到的 $C_1$ 中轮廓	相似度
0	36	0.89855	23	23	0.881113
1	1	0.925046	24	24	0.939447
2	38	0.921372	25	25	0.873563
3	3	0.790106	26	25	0.893572
4	26	0.870195	27	1	0.89062
5	20	0.83161	28	24	0.75736
6	6	0.949656	29	29	0.911077
7	7	0.931773	30	17	0.891805
8	29	0.915466	31	19	0.591313
9	9	0.938447	32	14	0.931847
10	10	0.951934	33	11	0.916842
11	26	0.86617	34	39	0.898876
12	12	0.853018	35	2	0.879321
13	13	0.830445	36	36	0.955528
14	14	0.962833	37	41	0.95307
15	15	0.971568	38	38	0.890927
16	16	0.940785	39	39	0.938076
17	40	0.892256	40	17	0.859266
18	18	0.889887	41	38	0.939096
19	19	0.879824	42	42	0.917221
20	9	0.898767	43	38	0.706694
21	21	0.960578	44	44	0.9693
22	22	0.861949			
匹配正确率			53.3333%		



图 3-1  $C_2$ 对 $C_1$ 单独 Hu 矩匹配配对轮廓比较图

表 3-6  $C_2$ 对 $C_1$ 单独普氏距离算法匹配结果

$C_2$ 中轮廓	匹配到的 $C_1$ 中轮廓	差距	$C_2$ 中轮廓	匹配到的 $C_1$ 中轮廓	差距
0	0	0.000258565	23	23	0.000108063
1	32	0.000326991	24	24	5.25117e-005
2	44	0.000293911	25	26	5.78165e-005
3	42	0.000172913	26	25	0.000305474
4	4	5.79953e-005	27	24	0.000665247
5	41	0.000444353	28	14	0.000391364
6	6	4.82798e-005	29	29	0.000199139
7	7	0.000106394	30	40	0.00022012
8	43	0.000634253	31	21	0.000284195
9	9	0.000354528	32	32	0.000375688
10	10	0.000235498	33	34	0.00053525
11	18	0.000740469	34	34	9.94802e-005
12	12	0.000151932	35	4	0.00025171
13	13	0.000197291	36	36	7.33733e-005
14	38	0.000244021	37	37	0.000108182
15	15	6.34193e-005	38	38	0.000143111
16	16	0.000160396	39	27	0.000100672
17	44	0.00017482	40	36	0.000442684
18	18	9.2864e-005	41	3	0.000334382
19	34	0.000389814	42	42	0.000375688
20	20	0.000237048	43	24	0.000507116
21	21	0.000165224	44	44	2.61068e-005
22	22	0.00016135			
匹配正确率			53.3333%		



图 3-2  $C_2$ 对 $C_1$ 单独普氏距离匹配配对轮廓比较图

### 3.2 加权评分匹配

从上一小节的实验中，我们可以发现，对于轮廓匹配，Hu 矩匹配和普氏距离匹配都是比较有效的匹配算法，但是两种算法侧重不同。有的轮廓在 Hu 矩匹配中匹配不到正确结果，但是在普氏距离匹配中却可以，反之同样成立。

在 Hu 矩匹配算法和普氏距离匹配算法的基础上，本文提出了加权评分匹配算法。在该算法中，我们分别使用 Hu 矩匹配算法和普氏距离匹配算法对匹配结果进行计算、排序并打分，综合考虑两者评分结果，得到一个新的排名，取排名最高的轮廓作为最佳匹

## 第四章 加权评分匹配算法

### 4.1 加权评分匹配算法概述

加权平均匹配算法综合了两种匹配算法，看重单个匹配算法中匹配度高的部分，并尽可能减少匹配度小的部分对最后结果的影响。

加权平均匹配算法的重点在于加权与评分。

对于 Hu 矩匹配算法和普氏距离匹配算法，我们认为两者侧重点不同，且匹配效果相差不多，所以我们赋予两者的权重一样，各占 $\frac{1}{2}$ ，希望两种匹配算法能够相互弥补。

对于相似度与差距的排名，越靠后的结果是越离谱的，所以我们认为排名靠前的轮廓更为重要应该给予更好的分数，排名靠后的轮廓给予更差的分数。

### 4.2 加权评分算法实现

对于任意一个 $C_2$ 中的轮廓  $c$ ，分别使用两种算法，计算  $c$  与 $C_1$ 中所有轮廓的相似度或差距，得到序列 $D_h$ 与 $D_p$ 。

接着分别对序列 $D_h$ 与 $D_p$ 进行排名，相似度越高匹配度越高，所以 $D_h$ 按照降序排名，而差距越大匹配度越小，所以 $D_p$ 按照升序排名，得到有序序列 $R_h$ 与 $R_p$ 。

然后分别对有序序列 $R_h$ 与 $R_p$ 打分，得到 $C_1$ 中的每个轮廓与  $c$  的匹配得分 $S_h$ 与 $S_p$ ，两个序列的评分方式相同：

$$S_{hi} = i^2 \quad i = 1, 2, 3, \dots \quad (4.1)$$

$$S_{pj} = j^2 \quad j = 1, 2, 3, \dots \quad (4.2)$$

$i$  和  $j$  分别对应 $R_h$ 与 $R_p$ 中的排名。

最后将 $S_h$ 与 $S_p$ 中的分数相加得到 $C_1$ 中的每个轮廓与  $c$  的匹配总得分得分  $S$ ，对其进行排名之后取分数最好的轮廓即为最佳匹配。

### 4.3 加权评分匹配算法实验结果

同样对于数据集 $C_2$ 中的每组轮廓描述点，加权平均匹配算法寻找素材库(数据集 $C_1$ )中的最匹配项，匹配结果如表 4-1 与图 4-1 所示。

表 4-1  $C_2$ 对 $C_1$ 加权平均匹配算法匹配结果

$C_2$ 中轮廓	匹配到的 $C_1$ 中轮廓	$C_2$ 中轮廓	匹配到的 $C_1$ 中轮廓
0	36	23	23
1	13	24	24
2	35	25	25
3	3	26	25
4	27	27	21
5	41	28	24
6	6	29	29
7	7	30	43
8	5	31	2
9	9	32	21
10	10	33	32
11	37	34	34
12	12	35	5
13	13	36	36
14	28	37	37
15	15	38	38
16	16	39	39
17	38	40	23
18	18	41	38
19	19	42	42
20	5	43	24
21	21	44	44
22	39		
匹配正确率		53.3333%	

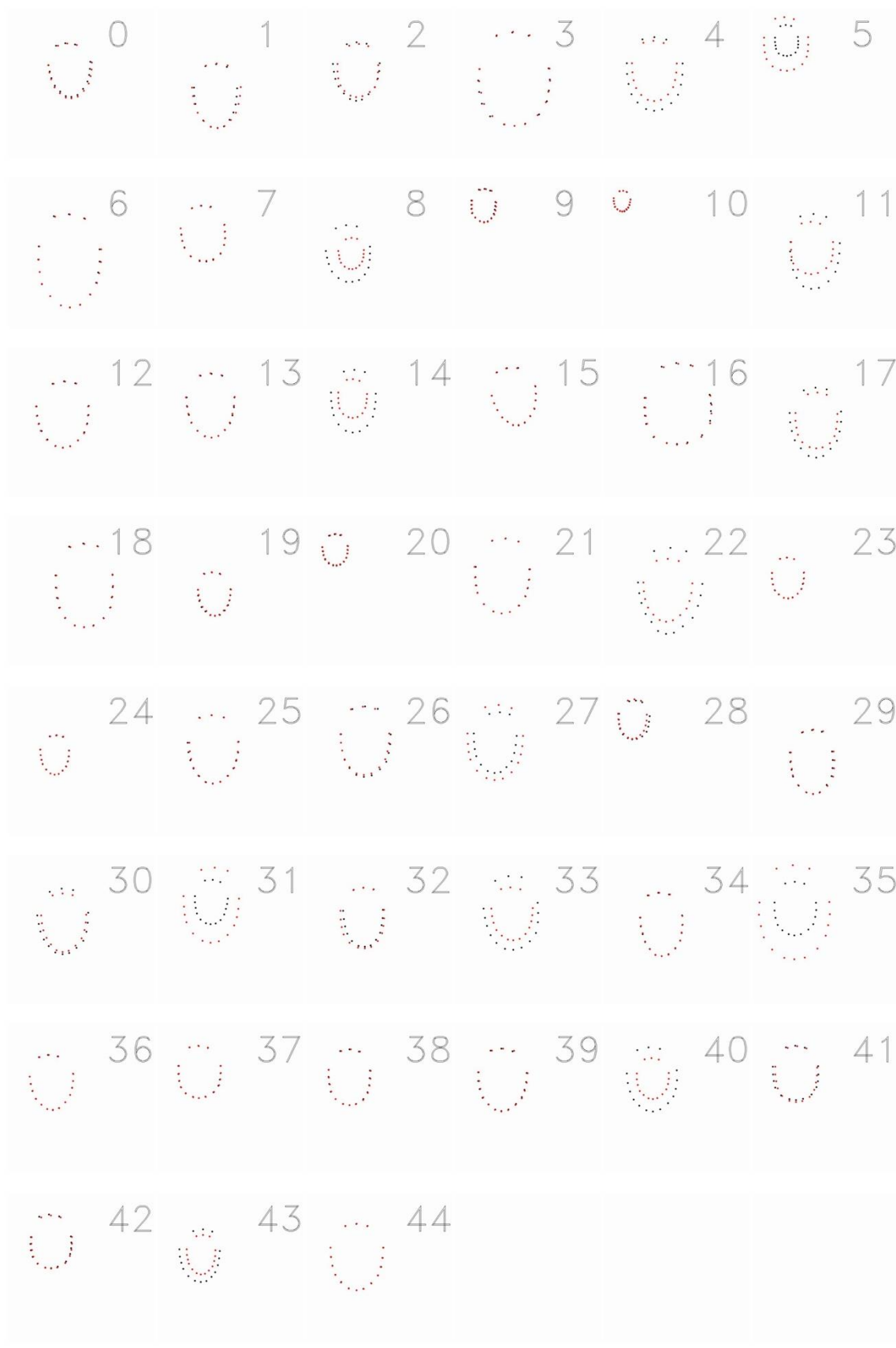


图 4-1  $C_2$ 对 $C_1$ 加权平均匹配配对轮廓比较图



#### 4.4 加权评分匹配算法分析

从实验结果中，我们可以看到加权评分匹配算法的匹配准确率为 53.3333%，与前面单个匹配算法的准确率一样，表明单单从本文样本数据上来讲，加权平均匹配算法没有提升匹配准确率，同样也没有降低匹配准确率。这样的结果是由以下两个原因造成的：

- 1) 素材库中现有轮廓之间的相似度较高，脸型类型分布不均。
- 2) 素材库中的素材没有进行素材标准化。

这个原因会导致在对轮廓坐标进行微调之后，与非原轮廓之间的匹配度可能会高于与原轮廓之间的匹配度，从而导致匹配失误。在仔细对轮廓比较图进行分析之后，我们发现微调之后的轮廓与匹配到的轮廓之间非常相似，从项目需求上来分析，匹配结果都是较为准确的。

理论上，加权评分匹配算法综合了两种不同的匹配算法，吸取了两种算法的优点，并很大程度上弥补了两种算法的不足。这样看来，加权评分匹配算法确实是一个优于 Hu 矩匹配算法和普氏距离匹配算法的算法。

## 第五章 总结与展望

### 5.1 本文工作的总结

#### 5.1.1 本文的工作

本文介绍了两种常见的人脸轮廓匹配算法——Hu 矩匹配算法和普氏距离匹配算法，分别对两种算法进行了实验，分析两个匹配算法的旋转、平移与放缩不变性。接着，本文综合 Hu 矩匹配算法和普氏距离匹配算法，创新出了一个新的轮廓匹配算法——加权评分算法。使用此算法，我们可以实现计算两个轮廓的匹配度以及在素材库中寻找与给定轮廓匹配度最高的轮廓。

除此之外，本文还建立了一个素材库，素材中的每一个素材都有漫画图像与之对应，我们还根据人脸轮廓计算出点集，通过点集来表示和操作每个轮廓。

#### 5.1.2 本文工作的不足

- 1) 本文所用素材之间的相似度较高，不同类型的脸型分布不均匀，导致算法匹配不到正确的脸型轮廓，拉低了匹配正确率。
- 2) 本文所用素材质量参差不齐，存在侧脸、照相时采光不好、佩戴饰品等不同路况，导致在识别时产生影响与干扰。
- 3) 本文所用素材大小与人脸位置没有固定，导致在算法中要实现多尺度、多旋转角度匹配这一附加步骤，拖慢程序速度。
- 4) 本文加权评分算法所综合的单个匹配算法（弱分类器）只有两个，过少，不足以应对所有情况，不够鲁棒。

### 5.2 未来工作的展望

#### 5.2.1 改进原有的不足

针对以上提出的不足，在未来的工作中，将从以下几个方面进行改进：

- 1) 根据脸型对素材进行分类，如图 5-1 所示，每种脸型内放置 5-8 个素材，每次计算每种脸型的平均脸，使用素材与平均脸进行匹配。
- 2) 规定使用标准证件照作为素材。
- 3) 使用标准化脚本对素材进行筛选和调整，使素材大小、人脸大小、人脸

位置等参数保持一致。

- 4) 多引进几种轮廓匹配算法(弱分类器),使用 Adaboost 对算法进行筛选和计算权重,最终生成一个效果更好、更加鲁棒的匹配算法(强分类器)。

### 5.2.2 进一步的研究方向

在“微漫”人脸漫画仿真系统中,人脸轮廓匹配是较为重要的一个步骤,但是我们发现,单单是匹配到最好的轮廓是不够的,有的时候看上去两者的相似度不高,这就需要我们z对匹配的脸型进行调整。

在进一步的研究中,我们会根据用户自拍中的真实数据来对匹配到的轮廓进行微调甚至夸张,使轮廓与用户更加相似,从而达到一种更好的漫画仿真效果。

#### 参考文献:

- [1]. 王俊艳, 苏光大. 基于下颌轮廓线的人脸分类方法. 红外与激光工程. 2004, (2): 159-163
- [2]. 陈慧静, 夏小玲. 基于形状上下文的人脸匹配算法. 计算机工程. 2009, 35(3): 208-210
- [3]. 李启娟, 李金屏. 基于轮廓信息的人脸检测. 计算机技术与发展. 2008, 18(9): 108-118
- [4]. 冯军华. 人脸部轮廓特征提取及在身份验证中的应用[M]. 太原科技大学, 2010
- [5]. 龚文胜, 陈宇, 辛国江. 一种基于轮廓图像的人脸检测方法. 计算机工程与设计. 2006, 27(21): 4068-4070
- [6]. MING-KUEI HU. Visual Pattern Recognition by Moment Invariants. IRE TRANSACTIONS ON INFORMATION THEORY. 1962: 179-187
- [7]. 陈瑞森. 应用于数字图像识别的Hu矩缩放不变性分析. 微型机与应用. 2011, 30(1): 29-31
- [8]. P. David Polly. Procrustes, PCA, and 3D coordinates. Indiana University, 2012
- [9]. YARON LIPMAN, REEMA AL-AIFARI, INGRID DAUBECHIES. THE CONTINUOUS PROCRUSTES DISTANCE BETWEEN TWO SURFACES[D]. PRINCETON UNIVERSITY, 2011
- [10]. Graham McNeill, Sethu Vijayakumar. 2D Shape Classification and Retrieval. In Proceedings of International Joint Conference on Artificial Intelligence(IJCAI' 05). Edinburgh. UK. 2005
- [11]. Gary Bradski, Adrian Kaehler. Learning OpenCV Computer Vision with the OpenCV Library. O'Reilly Media. 2008
- [12]. Stanley B. Lippman, Josée Lajoie, Barbara E. Moo. C++ Primer(5th Edition). Addison-Wesley. 2015

## 致 谢

经过大学四年的学习，籍论文完成之际，我特向指导和帮助我的老师、同学、同事、朋友及关心支持我的家人表示诚挚的谢意。

首先要感谢我的导师曾坤副教授。本文是在曾坤导师的精心指导下完成的，从论文的选题、设计方案直至完成论文的整个过程中，都得到了曾坤老师耐心细致的指导。

感谢中山大学数据科学与计算机学院所有的领导和老师，前三年的学习基础对我非常重要，是你们让我能够静静地坐下来，在知识的海洋里吸取更多的营养，从而能够为自己进一步地加油充电。通过论文的撰写，使我能够更系统、全面地学习有关计算机科学方面的理论知识，并得以借鉴众多专家学者的宝贵经验，这对于我今后的工作和我为之服务的企业，无疑是不可多得的宝贵财富。

同时感谢我亲爱的同学们，在学习中我们相互帮助，互相激励和关心。是你们让我在学习很生活中收获到了更多的东西。

最后感谢我的家人，这么多年来，正是你们的支持和鼓励，才使我顺利地完学业；正是你们的关心和默默的奉献，给我创造了优越的条件，使我在学习的道路上乐观向上、勇往直前。

**毕业论文（设计）成绩评定记录**  
Grading Sheet of the Graduation Thesis (Design)

<p>指导教师评语 Comments of Supervisor:</p>          <p>成绩评定 Grade:</p>          <p>指导教师签名 Supervisor Signature : _____ Date: _____</p>
<p>答辩小组或专业负责人意见 Comments of the Defense Committee:</p>          <p>成绩评定 Grade:</p>          <p>签名（章）：_____ Date: _____ Signatures of Committee Members(Stamp)</p>
<p>院系负责人意见 Comments of the Academic Chief of School:</p>          <p>成绩评定 Grade:</p>          <p>签名（章） Signature(Stamp): _____ Date: _____</p>