

摘要

手势是人与人之间一种非常重要的交流方式，在人机交互的研究中也有非常大的实用性。随着计算机技术的发展，以及人与计算机之间的交互的多样化，手势识别技术涉及图像处理、模式识别、计算机视觉等领域的一个交叉学科，加之手势的多样性、多义性，以及时间和空间上的差异性等特点，因此手势识别自然成为一个极具吸引力的研究课题。手势识别的过程可分为手势建模、手部数据的获取、手部数据预处理、手部特征的提取、手部特征的识别五个阶段。本文研究的是静态手势识别，主要通过对目标手势进行建模分析得到可能有效的特征参数，然后分析基于轮廓图像的特征参数的提取方法，最后通过基于模版匹配的手势识别算法对提取到的特征参数进行实验，比较它们有效性。

首先，本文对基于三维模型的手势建模和基于表现的手势建模进行比较分析，针对本文研究的待识别目标样本设计了一种基于表现的手势模型——人手部 Tortoise 模型，它的优点是将手指和手掌区分开来，能够很好的描述手部的基本特征，便于分析特征参数。通过对人手部 Tortoise 模型的分析，得到若干个可能有效的特征参数，具体包括手势中伸出的手指的个数、手部轮廓的缺陷个数和手势轮廓的面积与周长的比值。另外，图像的不变矩特征作为图像识别中的重要参数，本研究中也分析计算了 7 个 Hu 矩不变量作为特征参数。

本文预处理部分重点比较了一般自适应阈值分割方法和基于大津法的阈值分割方法对手势分割的效果，实验证明利用基于大津法的阈值分割方法进行手势分割效果比一般自适应阈值分割方法好。

本文特征参数提取部分通过对人手部 Tortoise 模型的分析，研究出计算手掌轮廓、手掌半径、手掌重心、缺陷个数、手指个数和轮廓面积与周长的比值的方法，尤其是通过凸多边形验证等四个规则求手掌轮廓的方法，以及基于手掌轮廓的手指个数的计算方法。另外，本文研究 Hu 不变量，实现了基于轮廓图像的 Hu 矩不变量的计算，得到 7 个 Hu 矩不变量。

最后，本文对十个特征参数的计算结果进行详细分析，将所有特征参数分为三组，为了比较三组特征参数的有效性，设计了一套基于模版匹配的手势识别系统，实验结果表明提取的十个特征参数用于手势识别是有效的。

关键字： 手势识别，特征提取，识别算法，不变矩，手势建模

Abstract

The gesture is a very important means of communication between people in the study of human-computer interaction is also very large practical. With the diversification of the interaction between the development of computer technology, computer, gesture recognition technology involves an interdisciplinary field of image processing, pattern recognition, computer vision, coupled with the diversity of gestures, multi ambiguity, and time and the spatial variability of characteristics, gesture recognition will naturally become an attractive research topic. The gesture recognition process can be divided into gesture modeling, hand data acquisition, identify the five stages of the hand data preprocessing, feature extraction of hand, hand characteristics. In this paper, the static gesture recognition, mainly through modeling analysis may be effective in the characteristic parameters of the target gesture, and then analysis based on the outline of the image feature parameter extraction method, and finally by the gesture recognition algorithm based on template matching to extract the characteristics The parameters of the experiment, comparison of their validity.

Firstly, the comparative analysis of three-dimensional model of the gesture-based modeling and performance-based gesture modeling, pending identification of the target sample for this study design the gesture of a performance-based model - staffing the Department of Tortoise model, it has the advantage fingers and the palm to distinguish a good description of the basic features of the hand, to facilitate the analysis of characteristic parameters. Tortoise model staffing department to get a number of potentially effective characteristic parameters, including the number of fingers outstretched in the gesture, the ratio of the number of defects of hand contour and gesture contour area and perimeter. In addition, the image of the moment invariant features as image recognition of the important parameters in this study analysis of seven Hu moment invariants as characteristic parameters.

The focus of this article pretreatment part of general adaptive thresholding method and the gesture segmentation segmentation method based on Otsu's threshold effect, the experiments show that the use of gesture segmentation method based on

Otsu threshold segmentation than the general adaptive threshold segmentation method.

This article feature parameter extraction part by the Ministry Tortoise staffing model analysis to work out the outline of calculation of the palm of your hand, the radius of the palm of your hand, palm center of gravity, the ratio of the defects in the number of fingers the number and profile area and perimeter, especially through the convex polygon verification four rules seek the palm of your hand contour and contour based on the palm of your hand, fingers the number of the calculation method. In addition, this study Hu invariants, to achieve a calculation based on the outline image of Hu moments invariants, seven Hu moments invariants.

Finally, a detailed analysis of the calculation results of the 10 feature parameters, all the characteristic parameters are divided into three groups, three groups were compared the effectiveness of the characteristic parameters, the design of a gesture recognition system based on template matching, experimental results show that extraction ten characteristic parameters for gesture recognition is valid.

Keywords: Gesture recognition, Feature extraction, Recognition algorithm, Invariant moments, Gesture modeling

第1章 绪论

1.1 研究背景和意义

随着信息时代的到来，计算机技术在现代社会中的影响在不断的增大^[1]。随着人们使用计算机的水平的不断提高，各行各业对于计算机的使用也不断深入，对于计算机的性能以及适用环境的要求也越来越高^[2]。信息技术的快速发展激发了人们对人机交互更加拟人化、自然化的渴望。如何建立和谐自然的人机交互环境，使用户可方便地以人类所熟知的方式使用计算机是我们急需解决的难题。传统的人机交互方式由当初的只有键盘输入到目前的鼠标、遥杆、无线输入设备等等都大大的便利了人们与计算机之间的交互^[3]，使得人们能够更方便的操作计算机，从而能够快速的完成一定的任务，然而这些交互方式既要依赖额外的输入设备，又不符合人们的交互习惯，因此并未能完全满足人们与计算机交互的需要^[4]。随着计算机技术的进步，特别是图像处理和识别技术的进步，人们的目光不再局限于传统输入方式的改进上，如何利用人自身的生物特征，让人与计算机直接进行交互成为当前人机交互的研究热点。这种利用人的生物特征进行人机交互的领域统称为生物识别^[5]。

目前，人体可以利用的特征主要包括人脸、指纹、掌纹、手势、虹膜及步姿步态等，而这些特征主要应用于信息安全及身份识别等较窄的领域^[6]。其中，人手是人体最灵活的部位，可以表达丰富多变的含义，手语是人们日常生活中的第二语言^[7]。由此，想到将人的手势作为人机交互的手段，相比其他人体特征而言更为自然、直接、丰富^[8]。另外，相对于传统的人与人之间丰富多彩的交互方式，人们与机器的交流往往没有表情，没有动作，机械呆板不自然。因而，使机器能够感知人体语言，对于提高人机交互水平和加强人机接口的可实用性具有不可估量的意义^[9]。而手势具有很强的视觉冲击和效果，它生动、形象、直观，并且富含大量的交互信息，有着与口语、书面语言等自然语言相同的表达能力，是人体语言的一个重要组成部分，因此它完全可以充当人机自然交流的手段，在人机交互领域中起到至关重要的作用^[10]。然而由于人手本身是一个复杂的可变形体，而且手势具有多样性、多义性及不确定性^[11]，因此这是一个极富挑战性的多学科交叉的研究课题，是近年来人机交互的研究热点及难点。研

究手势识别对于改善人机交互的意义很大^[12]。

1.2 国内外发展现状

从上个世纪 90 年代起，人们就开始了手势识别的研究，手势识别主要分为两个方面：基于数据手套的手势识别和基于视觉的手势识别^[13]。基于数据手套的手势识别目前采用较多的神经网络等模式识别的方法，要求用户戴上特殊的传感器手套，利用传感器手套来跟踪和标记人手在空间坐标中运动的轨迹和时序信息，通过一些信号处理手段达到人机交互的目的^[14]。这种方式的优点在于输入的数据量少，识别率非常高，而且实时性很强；但它昂贵的设备，交互动作的不自然也成为了明显的缺点^[15]。

基于计算机视觉的手势识别主要是利用摄像头来获取手势图像数据，通过对图像数据的处理及分析，包括图像预处理、手势分割、手势建模、手势分析和特征提取，从而识别手势的含义达到与计算机交互的目的^[16]。该方法的最大优点是交互方式自然简单，对设备要求也非常低，然而，它的识别率相对比较低，实时性较差，在实时性要求比较高的场合这种方法还值得我们研究^[17]。目前，国内外主要研究基于传感器手套的手势识别，而且取得非常可喜的成绩，然而随着信息技术尤其的模式识别技术的提高，越来越多的研究者把目光投向基于视觉的手势识别上来了^[18]。以下简单介绍了在基于传感器手套和计算机视觉两个方面比较重大的研究成果。

在利用传感器手套等典型传感设备的方法中，台湾大学的 Liang 等人利用单个 VPL Data Gloves（美国加州 VPL 公司研发的数据手套）作为手语输入设备，可识别台湾手语课本中的 250 个基本词条，识别率为 90.5%^[20]，CMU 的 Christopher Lee 和 Xu 在 1995 年完成了一个操纵机器人的手势控制系统^[21]；M. W. Kadous 用 Power Gloves（超能手套）作为手语输入设备，识别由 95 个孤立词构成的词汇集，正确率为 80%^[22]。

在基于视觉的方法方面，具有代表性的研究成果包括：1991 年富士通实验室完成了对 46 个手语符号的识别工作^[23]；J. Davis 和 M. Shah 将戴上指尖具有高亮标记的视觉手套的手势作为系统的输入，可识别 7 种手势^[24]；Starner 等在对美国手语中带有词性的 40 个词汇随机组成的短句子识别率达到 99.2%^[25]；K. Grobel 和 M. Assam 从视频录像中提取特征，采用 HMM（隐马尔科夫模型）技术识别 262 个孤立词，正确率为 91.3%^[26]。

1.3 研究的难点

目前，然手势识别系统的研究已取得很大的进展，并且在不同的领域有着较高的识别率，但是手势识别还是存在着一些难点带我们去解决。

在不同的应用环境下，手势往往含有多种含义，具有多种表示方式，在时间和空间上也具有较大的差异性，这些特点决定了手势识别研究任务的艰巨性，另外，由于人手是复杂变形体具有柔软性以及计算机视觉的不适应，增加了手势识别技术研究难度，这就要求我们在不同应用环境下分析各种手势的几何特征，选取正确的特征向量才行^[27]。本文以普通表示数字 0 到 9 的手势作为研究的样本。在进行手势分割时，由于客观环境的影响，如光照强度和角度的变化、手势动作的旋转和平移、手势背景的复杂性（例如类肤色的事物存在于背景中）等因素的存在使得手势区域的位置检测不够准确或者发生偏离，这就增加了准确分割目标图像的难度^[28]。因此，对于基于只拥有单个普通摄像头的计算机来说，要在复杂背景下像人眼一样正确的分割出手势，目前还存在很大的困难，这就迫使我们在手势识别系统的实际开发和设计过程中，对环境强加了一些限制。最后，针对不同类型的手势计算不同的不变矩作为基本的特征向量进行分类识别则是手势识别的另一大难点^[29]。目前已有 Hu 不变矩、Legendre 矩、Tchebichef 矩和 Zernike 矩在图像识别方面的研究，各有各的特点，效果一般，而且在计算不变量的过程相对来说比较复杂^[30]，本文将分析传统的离散几何不变矩，求出七个不变量，选择其中四个作为识别参数。总的来说，本文会从以下三个关键点来研究手势识别技术。

（1）如何建立一个简单实用的手势模型，并且提高手势分割的精确度，获取更多的有效特征参数；

（2）通过大量的几何分析，研究提取各种有效的几何特征参数的方法，并且验证各个特征参数的有效性；

（3）根据实际需要，选取一种合适的分类模型或者分类算法也是手势识别技术的关键。

1.4 本文的主要工作和组织结构

本文的研究目标的具体目标就是识别十种手势图像，具体的是基于人手部 Tortoise 模型分析手部的几何特征，主要选择手势图像中伸出的手指的个数、缺

陷个数和图像的不变矩总共十个识别特征参数。在得到识别参数后，将它们分为三组，本文利用基于模版匹配的识别算法，分析比较三组特征参数的有效性和识别率。以下文章各个章节的内容安排：

第 1 章叙述了手势识别的研究背景和意义，查阅国内外论文，分析了当前国内外手势识别的研究现状，最后分析了手势识别技术的难点技术、待解决问题以及本文研究的目标。

第 2 章首先比较基于三维模型的手势建模和基于表现的手势建模，建立一种名字为人手部 Tortoise 模型的手势模型，通过对这个模型分析，总结出识别手势需要计算的几个几何特征参数。然后，简述了手势识别系统基本框架图，并且分析各部分的作用。最后重点叙述了手势预处理和手势阈值分割的过程，其中对阈值分割的方法做了简单的对比实验，并且选择一种比较好的阈值分割方法。

第 3 章主要根据前面总结的若干个特征参数分析手势图像的几何特征，研究特征参数的计算方法，重点阐述了手指个数和 Hu 矩不变量的计算方法。具体地是从感兴趣图像的轮廓入手，先计算出手的整体轮廓，提取轮廓图像的外接多边形和缺陷点，然后结合手掌的几何特征分析外接多边形的顶点和轮廓缺陷点的分布情况，研究得到计算近似手掌轮廓外的外接多边形，最后根据手掌轮廓的外接多边形得到两种计算手指个数的方法。最后总共计算了 10 个特征参数，包括 7 个 Hu 矩不变量、面积周长比、缺陷个数和手指个数。

第 4 章分析了整个识别系统的详细设计框架，并且对其中特征参数的提取方法和识别算法的实现进行了分析。最后通过该识别系统对三组不同的特征参数进行了对比实验，通过计算三组特征参数的识别率来分析比较 Hu 矩不变量与几何特征参数的有效性、几何特征中手指个数和缺陷个数的有效性，最终选择了 (Hu1, Hu2, Hu3, Hu4, 面积周长比, 手指个数) 的组合作为特征参数。

第 5 章对本文的研究进行总结和展望。

第2章 手势建模和预处理

2.1 手势建模

手势建模简单的说就是结合实际的应用，分析手部（包括手腕、手掌和手指）所有有效的特征，构造一个有效的手势识别的分类器。手势建模具体分为两个步骤，第一步是基于真实手形的特征和手势辨识的需要抽象出一个数学模型，我们把这个模型叫做人手部模型；第二步是根据人手部模型提取的人手部几何和纹理特征是建立手势模式库。这两步有着密切的关系，人手部模型是建立手势模式库和进行手形匹配的基础，而且手势模式库和人手部模型共同构成手势识别的分类器，用于确定那些特征值或者识别参数需要我们计算，便于更大程度的区分各种手势^[31]。在基于视觉的手势识别框架中，手势模型是最基本部分之一。根据不同的应用背景，手势识别采用的模型会有所不同，而对于不同的手势模型，采用的手势检测、特征提取、识别技术也会有差别。目前手势建模主要分为基于表观的手势模型与基于三维的手势模型^[32]。手势模型的详细分类如图 2-1 所示。

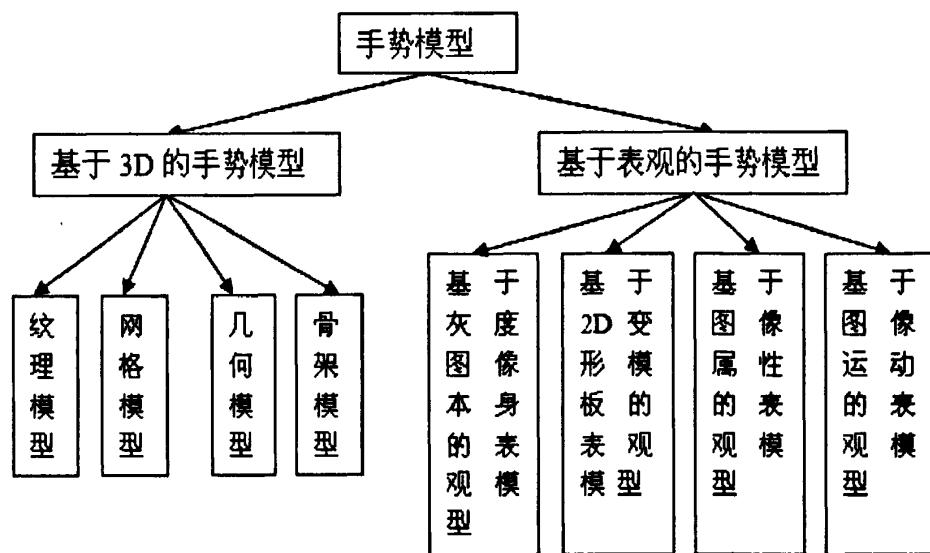


图 2-1 手势模型分类

2.1.1 基于三维模型的手势建模

基于三维模型的手势建模是对手部的关节结构进行三维建模，描述手的各个关节的位置与角度等属性。在基于模型的方法中，手势特征具有很高的自由度，模型的参数也非常多。

2.1.2 基于表观的手势模型

基于表观的手势建模是从二维平面观察得到的平面图像信息描述手的特征。基于表观的手势模型主要包括基于颜色、基于轮廓的模型两种。

基于颜色的手势模型是把手势图像看作像素颜色的集合，通过提取手部的颜色的特征来描述手势。基于颜色的手势模型的常用特征是手部区域的颜色直方图、高斯模型。

基于轮廓的手势模型是把手看作一个轮廓，通过提取手部图像中手的轮廓的几何特征来描述手势。

2.1.3 两种模型的比较

基于模型的手势模型的优点是能够很精确地描述手势，但它的不足之处是模型中手的模型参数多，要处理的数据量比较大，所以在手势识别过程中模型参数的估值也比较困难。基于视觉的手势模型常通过多目摄像头来采集图像，以此来建立手势的立体三维模型；合成的过程中也引入噪声，所以鲁棒性不强。基于表观的方法从二维平面观察手势，从手势图像是提取颜色特征、几何形状、轮廓等信息，计算量少，可靠性和提取效率都比较高。两种模型都有它的适用场景，实际应用开发中可以根据实际需要做两类手势模型中做一个选择。

2.1.4 人手部 Tortoise 模型

目前用于手势识别的手势模型有很多种，由于本文研究的手势识别对象在简单背景下获得的，识别目标比较明显，特征也比较明确，所以建立一种人手部 Tortoise 模型，他是一种基于表现的手势模型，具体的说是基于轮廓的手势模型。

人手部 Tortoise 模型简单的说就是将手部分解成饼状的手掌和 n 根被看作长条的手指，这种模型能够很好的描述手部的基本特征，如图 2-2 所示为模型图。

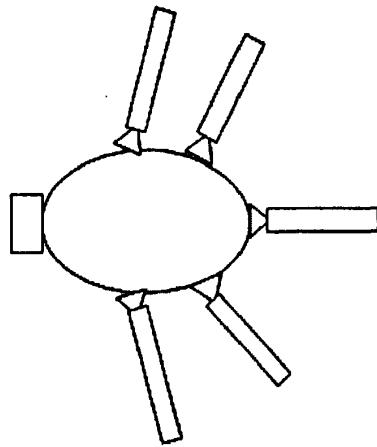


图 2-2 人手部 Tortoise 模型

Tortoise 模型包括的参数可以用公式的形式表示, 如下式:

$$y = f(r_1, r_2, n, L_1, \dots, L_n, W_1, \dots, W_n, \theta_1, \theta_n, R, G, B) \quad (2-1)$$

公式 (2-1) 满足的约束条件如下:

$$\begin{cases} r_1 \geq 1.5r_2 \\ n \in [0, 5] \\ 1.2r_1 \geq L_i \geq 0.3r_1, \quad i = 1, \dots, n \\ r_i \geq 3W_i, \quad i = 1, \dots, n \\ \theta_i \in [0, 90], \quad i = 1, \dots, n \\ R \in [0, 255], \quad G \in [0, 255], \quad B \in [0, 255] \end{cases} \quad (2-2)$$

其中, r_1 为手掌部半径, r_2 为手腕部半径, n 为手指个数, $L_1 \sim L_n$ 为每根手指的长度, $W_1 \sim W_n$ 为每根手指的宽度, $\theta \sim \theta_n$ 为每根手指与手腕间夹角, (R, G, B) 为皮肤颜色。

Tortoise 模型具有以下优势:

(1) 手掌被近似成为对称的饼状模型, 这使得模型更加的简洁, 进而提高了手形匹配速度, 而且具有对称性的模型也减少了由于手部旋转造成的错误识别概率。

(2) 将手指看作分布比较随意的条状模型, 使得在具体应用中, 可以通过分析每种手势中伸出来的手指的几何分布, 利用特征提取的方法从不同二维图像中计算总的手指数量来判断手形, 这样就降低了由于遮挡带来的匹配误差。

(3) 将手掌和手指作为一个整体的同时, 又将他们分开来分析, 使用手掌和手指的相对尺寸作为区分不同手势的依据, 这样就减少了在采集目标图像的过程中因为距离原因产生的手形缩放所带来的匹配误差。

本文识别的对象是用手势表示的数字 0 到 9 的集合, 为了识别这几个手势, 除了常用的不变矩特征作为重要的识别参数以外, 本文根据人手部 Tortoise 模型, 选择手指个数重要的特征参数, 因为每个手势伸出来的手指个数不尽相同, 有个别手势伸出的手指个数一样, 但是伸出的是不通的手指, 这个通过其他的特征参数来区分开来, 本文选择手势图像的面积和周长的比作为第二个参数, 分析每个手势轮廓的面积, 手掌面积占主要部分, 由于伸出的手指不同, 每种手势轮廓图像的面积和周长的比值区分度比较大。本文将在下面的章节中详细介绍获取这两个特征参数的方法。

2.2 手势识别系统的基本框架图

一般图像处理和模式识别系统的基本框架主要包括图像样本的采集、图像预处理、分析图像特征并提取识别参数、利用识别参数对图像进行分类识别^[33], 最后输出识别结果, 手势识别系统整体框架图如图 2-3 所示。

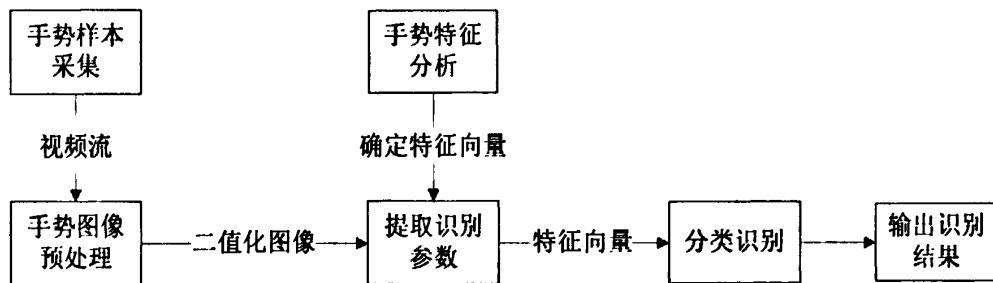


图 2-3 手势识别系统整体框架图

框图简单介绍如下:

(1) 手势样本图像的采集: 本文都是以 AVI 视频文件作为处理对象, 为了更好的提取手部特征, 拍摄之前, 先选好固定的背景, 利用普通的摄像头录制手势视频。

(2) 手势图像预处理: 主要是图像色彩空间的转换、提取灰度图像、中值滤波和阈值分割, 其中阈值分割是得到手势轮廓图像的基础, 好的阈值分割对

手势轮廓的提取有很大的帮助，在2.4.3节会详细介绍基于大津法的自适应的阈值分割。

(3) 分析图像特征并提取需要的识别参数：结合人手部 Tortoise 模型，分析手部几何特征，总共提取六个识别参数，其中有四个是基于手势轮廓图像提取的几何不变矩参数，另外两个分别是轮廓图像中伸出的手指的个数以及手势轮廓图像面积与周长的比值。

(4) 分类识别：针对已经提取的特征识别参数，组成一组特征向量，利用识别算法对手势图像进行训练和分类识别，本文中利用模版匹配的思想设计分类算法。

(5) 输出识别结果：识别结果以数据的形式储存到文本文件中，并且会计算每种手势的识别率。

2.3 手势图像样本采集和研究对象

手势识别一般分为静态手势识别和动态手势识别，静态手势识别是识别手型，读出手型表达的意义^[34]，而动态手势识别是识别手在空间中的运动轨迹，通过得到的运动轨迹参数来执行相应的操作，比如在投影上播放课件的时候，通过手势来进行的上下翻页、暂停、开始等操作，就是这方面的应用。本文中主要研究静态手势识别。为了使研究更带有目的性，本文给定了十个特定的手势作为识别对象，如图2-4所示的手势样本图中的十个手势图像分别表示数字0到9。简单的说，本文的直接的研究目标就是要让计算机识别这十个手势分别表示的意思，算出识别率。

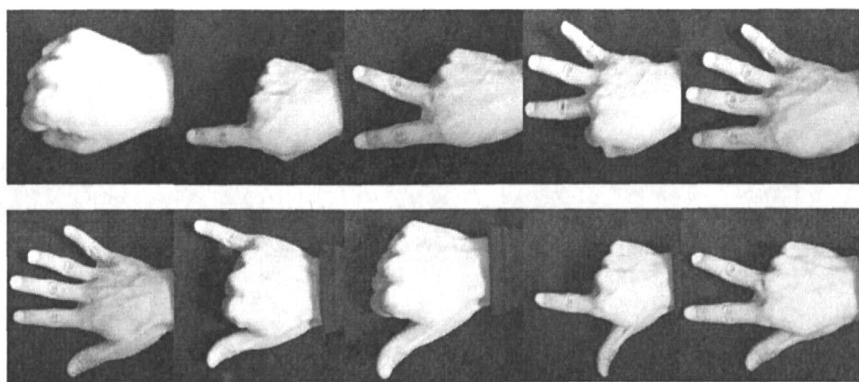


图 2-4 手势样本图

采集样本和建立样本集对于研究图像处理和识别是非常重要的，前面提到，本本文在采集样本的时候都是在简单背景下采集的，光照明暗程度、距离和角度都是差不多的，采集的到处理文件对象都是 AVI 视频流文件，具体的采集过程如下：

(1) 在数字视频摄像头前面按照上面规定的样本，从 0 到 9 的顺序连续做标准的手势，每个手势都要维持几秒钟，并且每个手势做好后作适当的平移，这个过程完了后会得到一个 AVI 视频文件。

(2) 观看这个 AVI 视频文件，将其中标准手势固定了的片段剪切下来，也就是出去其中变换手势的部分，留下完整的手势和手势平移的片段，这样会得到 10 个 AVI 视频文件，这十个文件就对应着上面这十个手势。

2.4 基于 OpenCV 的手势图像预处理

本文在研究过程中利用了 OpenCV 函数库，对手势图像进行预处理的过程主要包括色彩空间转换、中值滤波和阈值分割，手势图像的预处理过程框图如图 2-5 所示。

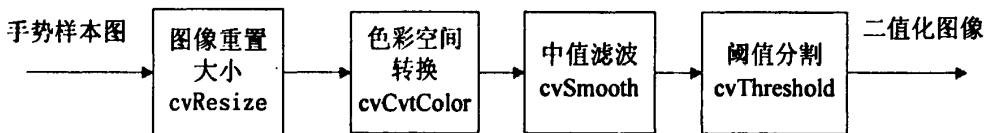


图 2-5 手势图像预处理过程

2.4.1 色彩空间转换

在平常的生活中，我们对颜色的讨论集中在通过红、绿、蓝三色混合而产生颜色的机制上，这种机制就是我们常说的 RGB 颜色模型，它有利于我们理解计算机中如何表示颜色，然而在图像处理的过程中，还需要一些更直观的颜色参数。为了寻求在各种特定环境中对于颜色的特征和行为的计算机解释方法，人们研究很多颜色模型或者叫做色彩空间，例如 RGB 彩色空间、CMY 彩色空间、YUV 彩色空间、YIQ 彩色空间和 HSV 彩色空间（也叫 HSI 彩色空间），但是，在这么多颜色空间描述方法中，没有哪一种颜色模型能解释完全所有的颜色问题，但是在实际的应用当中，我们选择相应的颜色空间模型来帮助描述我们能看到的各种颜色特征^[35]。在本文中，为了更好的描述手势图像的颜色特征，

选择了 HSV 颜色模型。

与传统的 RGB 颜色模型相比, HSV 模型使用了对用户更直观的颜色空间描述方法。HSV 色彩空间是从人的视觉系统出发, 用色调 (Hue)、色饱和度 (Saturation 或 Chroma) 和亮度 (Intensity 或 Brightness) 来描述色彩。HSV 色彩空间可以用一个圆锥空间模型来描述。尽管 HSV 色彩空间的圆锥模型相当复杂, 但它的确能把色调、色饱和度和亮度的变化情形表现得很清楚^[36]。如图 2-6 所示为 HSV 三维空间的六棱锥模型。

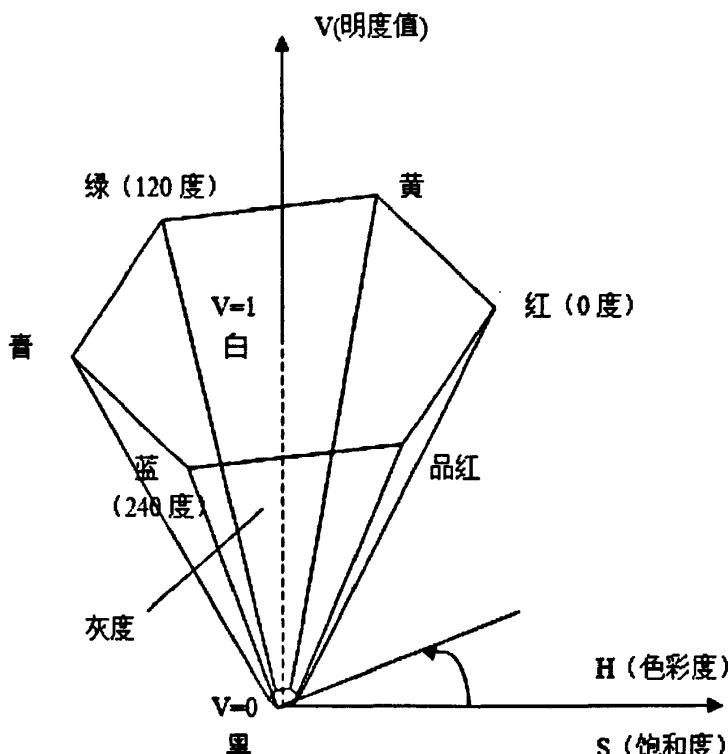


图 2-6 HSV 六棱锥

在六棱锥中, 色饱和度 S 沿水平轴测量, 从 0 到 1 变化, 而明度值 V 沿着六棱锥中心的垂直轴进行测量, 从 0 到 1 变化, 而色彩则使用与水平轴之间的角度来表示, 范围从 0 度到 360 度。总的来说 HSV 彩色空间有以下亮点和优点:

(1) 我们通常把色调和饱和度通称为色度, 用来表示颜色的类别与深浅程度。由于人的视觉对亮度的敏感程度远强于对颜色浓淡的敏感程度, 为了便于色彩处理和识别, 人的视觉系统经常采用 HSV 色彩空间, 它比 RGB 色彩空间更符合人的视觉特性^[37]。

(2) 采用 HSV 彩色空间减少彩色图象处理的复杂性, 增加快速性, 它更接近人对彩色的认识和解释。例如对色调、饱和度和亮度通过算法进行操作。在图象处理和计算机视觉中的大量算法, 都可以在 HSV 空间中方便的使用。它们可以分开处理而且是互相独立的, 因此 HSV 彩色空间可以大大简化图像分析和处理的工作量。

为了更好的描述手势图像的颜色特征，为了后面处理图像做准备，一开始对手势图像进行了色彩空间的转换，具体是从 RGB 色彩空间转换到 HSV 色彩空间，转换公式如下：

RGB → HSV (CV_BGR2HSV, CV_RGB2HSV) :

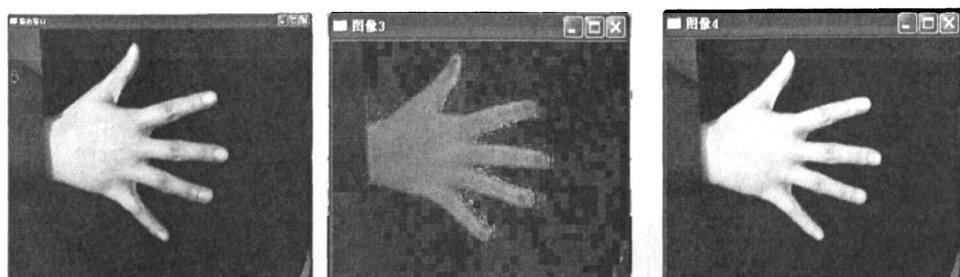
$$V = \max(R, G, B)$$

$$S = \begin{cases} [V - \min(R, G, B)] \times 255 \div V & , \text{ 若 } V \neq 0 \\ 0 & , \text{ 其他} \end{cases} \quad (2-1)$$

$$H = \begin{cases} (G-B) \times 60 \div S & , \text{ 若 } V = R \\ 180 + (B-R) \times 60 \div S & , \text{ 若 } V = G \\ 240 + (R-G) \times 60 \div S & , \text{ 若 } V = B \end{cases}$$

$$H = H + 360 \quad , \quad \text{若 } H < 0$$

使用上面从 0 度到 360 度变化的公式计算色调值，可以确保它们被 2 除后能适用于 8 位。



(a) 原图像 (b) 色彩转换后的图像 (c) 手势的灰度图像

图 2-7 色彩空间转换和灰度图像提取

在 OpenCV 中，通过 `cvCvtColor()` 函数对原手势图像进行色彩空间转换，对手势图像进行色彩空间转换的目的是为了得到手势图像的灰度图像，利用 OpenCV 的函数 `cvSplit()` 可以得到，效果如图 2-7 所示，其中图 (a) 为原图像，

图(b)为进行色彩空间转换后的图像, 图(c)为进行色彩空间转换后, 进行灰度提取的结果图像。有图可以看出, 经过色彩转换和灰度提取后, 手部区域的灰度值与背景有明显的区分, 这一步为后面手势图像阈值分割打好了基础。

2.4.2 中值滤波

本文在采集手势图像过程中, 由于光照、角度和采集设备的影响, 采集得到的视频文件中手势图像难免会有很多噪声, 为了去除这些噪声获得比较好的手势图像, 研究中采用了两个方法, 一是对于一段连续做完一整套手势的 AVI 视频文件进行剪切, 去除中间变换手势的过程, 只保留做好手势影像, 这样就得到一组视频文件 (文中是 10 个手势, 所以得到 10 个 AVI 文件), 在进行识别处理的时候就已经去除了因为变换手势等因素产生的噪声; 另外的一种方法就是进行图像平滑处理。

由于在采集手势图像的过程中, 手势会出现平移和旋转, 这就会给视频文件带来一些图片不清晰的状况, 尤其是手部边缘轮廓的模糊。在比较图像平滑方法中, 中值滤波有保存边缘的特性, 在不希望出现图像边缘模糊的场合很有用, 所以尝试用中值滤波的方法来对手势图像进行平滑处理。中值滤波的思想是检查输入信号中的采样并判断它是否代表了信号, 使用奇数个采样组成的观察窗实现这项功能。观察窗口中的数值进行排序, 位于观察窗中间的中值作为输出。然后, 丢弃最早的值, 取得新的采样, 重复上面的计算过程^[38]。简单的说就是将对应像素点的灰度值用窗口内的中值代替。

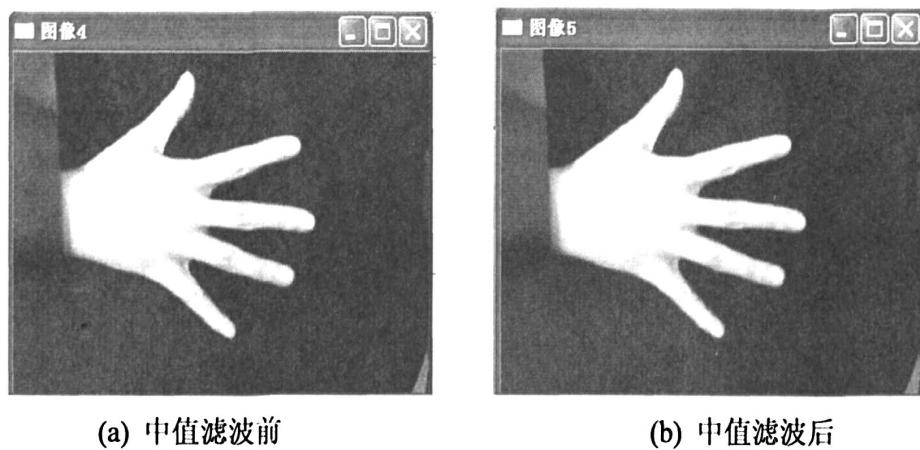


图 2-8 中值滤波前后对比

如图 2-8 所示为对手势灰度图像进行中值滤波后的效果。实现中值滤波时，为了简化编程工作，可以定义一个 $n \times n$ 的模版数组，需要注意的是，在窗口扫描图像的过程中，对于图像的四个边缘的像素点可以不处理或者使用灰度值为零的像素点来扩展图像的边缘，本文在得到色彩空间转换完成的手势图像后，对图像进行卷积核大小为 3×3 的中值滤波，具体使用了 OpenCV 中的 cvSmooth() 函数。由于背景图像本身比较干净，使用中值滤波主要为了滤除灰度提取后的一些细小杂质，从图 2-8 中看效果还不错。

2.4.3 图像的阈值分割

在对图像去除噪声以后，接着对手势图像进行了阈值分割。进行阈值分割的目的主要有两个，一是去掉噪声，主要是过滤掉像素值很小和很大的图像点，另一个就是使图像二值化，是的手势图像中，手的部分更加凸显，便于后面进行手部的轮廓提取，总的来说阈值分割的好坏直接影响后面手势轮廓的提取效果。本文采用一般自适应阈值分割方法和基于 OTSU 大津法的阈值分割方法对图像进行阈值分割。

(1) 一般自适应阈值分割方法

一般自适应阈值分割方法的阈值 $T(x, y)$ 在每个像素点都不同，基本方法是通过计算像素点周围的 $b \times b$ 区域的加权平均，然后减去一个常数来得到自适应阈值。本文利用 OpenCV 函数库中的函数 cvAdaptiveThreshold() 来实现。处理效果如图 2-9 所示。

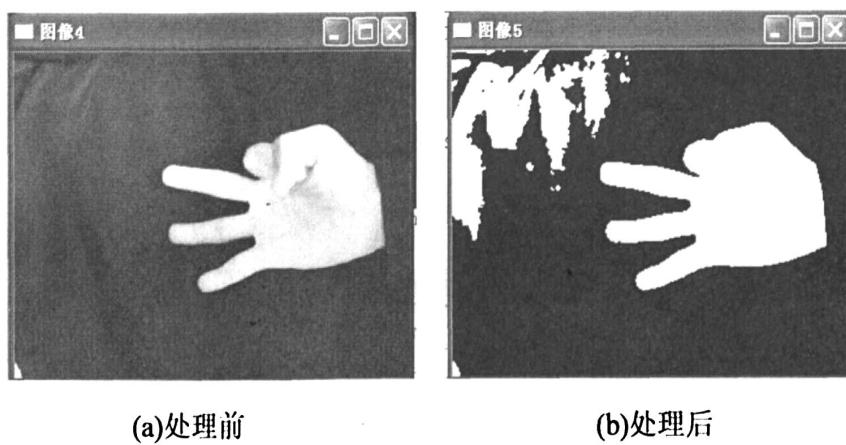


图 2-9 一般自适应阈值分割效果

分析图 2-9 的实验效果,发现一般自适应阈值分割方法并不能去除背景中与手部区域灰度值近似的区域,获得的二值化图像手部的轮廓也不够清晰。

(2) 基于 OTSU 大津法的阈值分割方法

OTSU 大津法又叫做最大类间方差法,它是日本学者大津于 1979 年提出的,是一种自适应的阈值确定的方法,在进行图像处理的时候经常用到,它是按图像的灰度特性,将图像分成背景和目标两个部分。背景和目标的类间方差越大,说明构成图像的两个部分的差别越大,当部分目标错分成了背景,或部分背景错分成了目标都会导致两个部分差别变小。因此,使类间方差最大的分割意味着错分概率最小^[39]。

对于图像 $I(x, y)$ 前景(即目标)和背景的分割阈值记作 T , 属于前景的像素点数总共占整幅图像的比例记为 ω_0 , 其平均灰度 μ_0 ; 背景像素点数总共占整幅图像的比例为 ω_1 , 其平均灰度为 μ_1 。图像的总平均灰度记为 μ , 类间方差记为 g 。假设图像的背景较暗,并且图像的大小为 $M \times N$, 图像中像素的灰度值小于阈值 T 的像素个数记作 N_0 , 像素灰度大于阈值 T 的像素个数记作 N_1 。

则有如下系列等式:

$$\omega_0 = N_0 / (M \times N) \quad (2-2)$$

$$\omega_1 = N_1 / (M \times N) \quad (2-3)$$

$$N_0 + N_1 = M \times N \quad (2-4)$$

$$\omega_0 = \omega_1 = 1 \quad (2-5)$$

$$\mu = \omega_0 \mu_0 + \omega_1 \mu_1 \quad (2-6)$$

$$g = \omega_0 (\mu_0 - \mu)^2 + \omega_1 (\mu_1 - \mu)^2 \quad (2-7)$$

将式(2-4)代入式(2-5), 得到等价公式:

$$g = \omega_0 \omega_1 (\mu_0 - \mu_1)^2 \quad (2-8)$$

采用遍历的方法得到使类间方差最大的阈值 T , 即为所求。

将 OpenCV 中阈值分割函数 `cvThreshold()` 中的 `threshold_type` 参数写成 `CV_THRESH_OTSU` 类型, 这样该函数就会使用大津法 OTSU 得到的全局自适应阈值来进行手势图像的二值化, 而参数中的 `threshold` 不再起作用。采用基于大津法的阈值分割方法对同一张手势图像进行处理后的效果如图 2-10 所示。

由图 2-10 可以看出, 基于大津法的阈值分割基本滤除了背景中与手势灰度值相近的区域的干扰, 获得的手部区域非常完整。

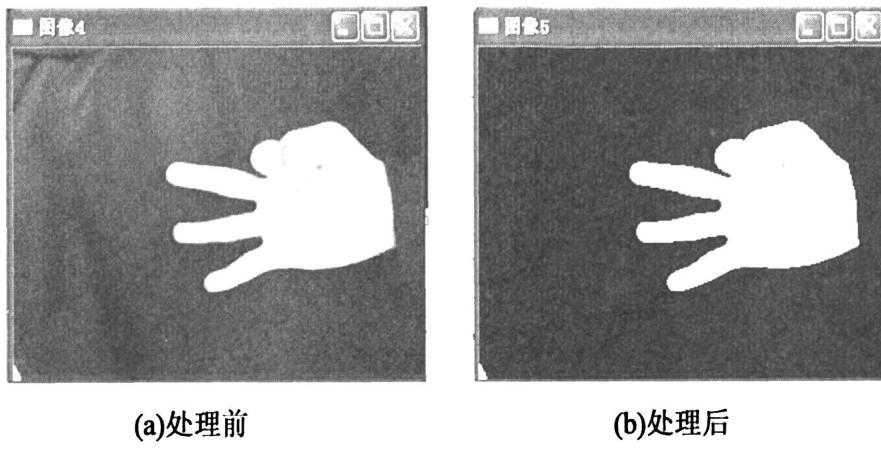


图 2-10 OTSU 大津法的阈值分割效果

比较上面两种阈值分割处理方法的结果, 基于大津法的阈值分割效果明显好于一般自适应阈值分割方法, 基于大津法的阈值分割不仅去除了背景中近似灰度值的影响, 而且获得的图像中手势的轮廓非常清晰。手势预处理过程主要目的是手势分割, 是手势轮廓和几何特征参数提取的基础。

2.5 本章小结

本章对基于三维模型的手势建模和基于表现的手势建模进行比较分析, 针对本文研究的待识别目标样本设计了人手部 Tortoise 模型的手势模型, 通过对这个模型分析, 总结出识别手势需要计算的几个几何特征参数。然后, 简述了手势识别系统基本框架图, 并且分析各部分的作用。最后重点阐述了手势预处理和手势阈值分割的过程, 通过对一般自适应阈值分割和基于大津法的阈值分割的对比实验, 最后选择基于大津法的阈值分割方法。这部分内容是整个研究的基础, 为后面提取手势图像的轮廓, 进而计算特征向量做好了准备。

第3章 基于手掌区域的手势图像特征分析与提取

前面提到，图像预处理完成之后，我们得到了手势图像的二值化的灰度图像，接下来就是在这个基础上分析和提取手部的特征向量。特征向量主要分为两个部分：一是从手部轮廓的几何特征方面着手获得的几何特征向量，主要包括手势中手势的轮廓、手掌的轮廓、每种手势伸出的手指的个数、缺陷个数以及手势轮廓图像的面积与周长的比值；另外一个是基于图像不变矩的特征提取，本文总共计算了基于轮廓图像的7个Hu矩不变量。本文详细分析了手势轮廓、手掌轮廓、手指个数、以及Hu矩不变量的计算方法。

3.1 手势轮廓的提取

3.1.1 开运算

一般由于噪声的影响，在对手势图像进行阈值分割以后，得到的图像的边界是不平滑的，手部区域具有一些噪声孔，背景区域上散布着一些小的噪声物体，连续的开运算和闭运算可以改善这种状况，所以在提取手部轮廓之前，我们对图像进行了开运算。

本文在对手势二值图像进行开运算时，分了两种情况，一种是以小模板进行开运算，在此基础上根据面积求整个手势图像的轮廓，计算手势轮廓图像的不变矩和面积周长比；另一种情况是以大模板进行开运算，类似的也是先根据面积求出整个手势图像的轮廓，再根据一定的规则（后面详细介绍）求出手掌的轮廓图像，然后由手掌轮廓信息确定每个手势图像中伸出的手指的个数作为一个重要的识别参数。

3.1.2 根据面积提取手的轮廓

在对手势二值图像进行开运算后，就开始提取手的轮廓。在进行以上处理后，我们看到的图像只有两种颜色，黑色和白色，其中黑色是背景的部分，白色区域为前景，前景中包括手和其他无关的部分，有的是噪声，能够通过寻找二值化图像中的连通域边界就可以获得前景物体的轮廓了，由于本文在简单背

景下提取手势样本图像，所以二值化后图像中代表手的部分的连通域具有明显特征，最大的特征是它的面积是所有连通域中最大的，基于此方法，就得到了手部的轮廓图像。

具体实现过程中，先利用 OpenCV 中 cvFindContours(tempImg[3], storage, &contour, sizeof(CvContour), CV_RETR_EXTERNAL, CV_CHAIN_APPROX_SIMPLE) 函数求出图像中所有的轮廓，并且将所有轮廓存储到容器 storage 中，指针 &contour 指向第一个外轮廓，然后从轮廓集中选出手的轮廓，具体做法是选择面积大于 10000 个像素的轮廓，因为样本背景本身比较简单，图像中没有比手更大的物体，具体函数实现如下：

```
void getHandContour(CvSeq* c, CvSeq** hc)
{
    while (c != NULL)
    {
        CvRect r = ((CvContour*)c)->rect;
        if (r.height*r.width>10000) //选出大于 10000 个像素的轮廓
        {
            *hc = c;
        }
        c = c->h_next;    }}}
```

值得注意的是，如果没有大于 10000 个像素的轮廓说明图像中没有手。在得到手的轮廓后，通过 OpenCv 提供的函数 cvDrawContours() 绘制手部轮廓，绘制的效果如图 3-1 所示。



图 3-1 手势图像的轮廓

图 3-1 中的红色线就是手势的轮廓，从图中可以看出，获得的手势轮廓线条非常清晰，并且完全是贴合手部区域的边缘线。

3.2 手掌轮廓的提取

在提取手掌轮廓之前还有两个准备工作要做，一是求出手掌轮廓的外接多边形，另一个是求出手掌轮廓的缺陷点，完成这两步是为了获得手部轮廓中的凸点集和凹点集，为后面删除不必要的点获得手掌轮廓做准备。

3.2.1 获取手部完整轮廓的外界多边形

由于我们已经获得手部的完整轮廓，并且已经获得轮廓线的点集，所以获得轮廓的外接多边形可以从完整的轮廓点集中选取比较特殊的点，顺序连接这些点获得的外接多边形必须包括手部完整轮廓中所有的点。具体做法是利用 OpenCV 中的序列访问函数 cvGetSeqElem()访问完整的轮廓序列，然后选取其中距离相隔 30 个像素的点，具体实现代码如下：

```
void simplyConvexHull(CvSeq* h, vector<CvPoint>& pts)
{
    int i;
    int count = h->total;
    pts.clear();
    //通过指针遍历轮廓线上的每一个点，将轮廓稀疏
    CvPoint** pt0 = (CvPoint**)(cvGetSeqElem(h, count-1));
    for (i = 0; i<count; i++)
    {
        CvPoint** pt = (CvPoint**)(cvGetSeqElem(h, i));
        //printf("x = %d \n", (*pt)->x);
        //排除相邻点
        if ((abs((*pt)->x - (*pt0)->x) + abs((*pt)->y - (*pt0)->y)) >30)
        {
            pts.push_back(cvPoint((*pt)->x, (*pt)->y));
            pt0 = pt;
        }
    }
}
```

```

    }
}

```

这样就获得了手部轮廓的外界多边形顶点的集合，接着通过 OpenCV 中的函数 drawConvexHullArray()画出外接多边形，处理后的结果如图 3-2 所示。

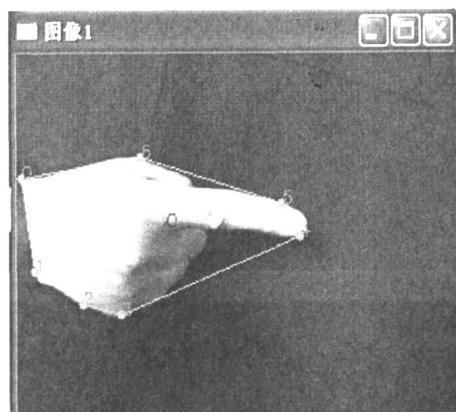


图 3-2 画好外接多边形的手势图像

由图 3-2 可以看出，通过以上方法获得的手势轮廓的外接多边形基本上可以包括整个手部区域。

3.2.2 获取外接多边形中的轮廓凸点和凹点

通过观察每个手势轮廓图像，我们发现每种手势轮廓都有不同的凸点和凹点，我们的目标是提取手掌区域的轮廓，因此除去手指部分就是手掌部分，而轮廓上的凸点和凹点一般都处于手指和手掌连接的地方，因此找出这些点，然后删除掉就可以获得手掌的轮廓了。

由于每个轮廓都有不同的轮廓凸包和凹点缺陷，这些缺陷可以用来描述物体的特征，而 OpenCV 函数库为我们提供了描述图像轮廓缺陷的方法。OpenCV 函数库定义了一个专门描述缺陷的结构体，具体定义如下：

```

typedef struct CvConvexityDefect
{
    CvPoint* start; /*缺陷开始的轮廓点*/
    CvPoint* end; /*缺陷结束的轮廓点*/
    CvPoint* depth_point; /*缺陷中离凸形最远的轮廓点（谷底） */
}

```

```

    float depth; /*谷底离凸形的深度*/
} CvConvexityDefect;

```

确定缺陷点的思想是由谷底离凸形的深度来确定谷底点，此时的谷底点就是我们所要求的凹陷点，在本文实际应用中，我们选取深度 depth_point 大于 10 个距离的点为谷底，即为我们所求的缺陷点，然后将所得的缺陷点序列存入数组，具体程序实现如下：

```

void getConvexityDefectArray(CvSeq* h, vector<CvPoint>& pts)
{
    int i, count = h->total;
    //printf("缺陷数量 = %d \n", count);
    pts.clear();
    for (i = 0; i < count; i++)
    {
        //定义一个缺陷点变量 cd
        CvConvexityDefect* cd = (CvConvexityDefect*)cvGetSeqElem(h, i);
        if ((cd) && (cd->depth>10)) //缺陷深度大于 10
        {
            CvPoint* pt = cd->depth_point;
            pts.push_back(cvPoint(pt->x, pt->y));
        }
    }
}

```

在得到缺陷点集合后，我们可以通过 drawConvexityDefectArray()函数在手势轮廓图像上画出缺陷点，如图 3-3 中的红色点即为缺陷点。

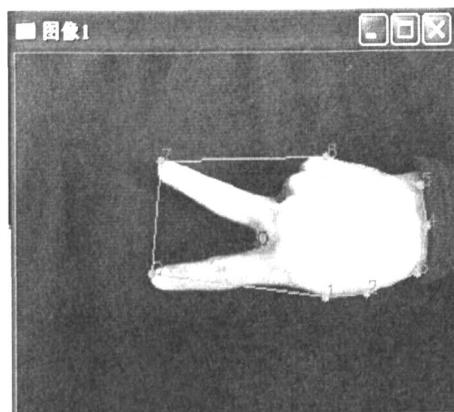


图 3-3 画好缺陷点的手势图像

从前面的研究不难发现，我们获得的缺陷点即为凹点集，而外界多边形的顶点即为凸点集，为了便于确定每个凸点和凹点具体位置，将凸点和凹点按原始轮廓的顺序，排列成向量。这样我们就得到了描述手势轮廓的外接多边形的每个点，并且确定了是凸点还是凹点。

3.2.3 求手掌区域轮廓的具体实现

前面已经得到手势图像的完整轮廓以及轮廓中凸点和凹点的信息，接下来就是如何从这些特征点集中找出规律选择合适的点来组成手掌区域的轮廓了。经过不断的分析，提取手掌区域轮廓主要分为以下几个步骤。

（1）规则一：删除两个凹点之间的凸点

观察上面获得的手部轮廓图像和凸凹点信息，我们发现两个凹点之间必有一个凸点，这个凸点往往是手指的指尖点，因此选择删除这样的凸点就可以删掉部分手指了。前面我们将凸点和凹点按原始轮廓的顺序，排列成向量，这其中我们将凸点和凹点赋上了对应的值，其中凸点的值为 1，凹点的值为 0，这样我们得到的向量会是类似 010111110101 这样的序列，因此我们将这个规则又叫做“010”规则，这也给我们在程序实现上带来了很大的便利，只需要删除两个值为 0 的点中间的那个点就行了。具体核心的实现代码如下。

```

for (i = 0; i < count; i++)
{
    //凸点的值为 1，凹点的值为 0，按照 010 规则循环删除
    if (i == 0)
    {
        //第一个点为凹点的情况
        if (!((pts[count-2].type == 0) && (pts[count-1].type == 1)
              && (pts[0].type == 0)))
        {
            v.push_back(pts[count-1]);
        }
    }
    else if (i == 1)
    {
        //第二个点为凸点的情况
        if (!((pts[count-1].type == 0) && (pts[0].type == 1)
              && (pts[1].type == 0)))
    }
}

```

```

{
    v.push_back(pts[0]);
}
}
else
{
    //中间一般循环过程
    if (!((pts[i-2].type == 0) && (pts[i-1].type == 1)
        && (pts[i].type == 0)))
    {
        v.push_back(pts[i-1]);
    }
}
}
}

```

删除两个凹点间的凸点后手势的轮廓图像如图 3-4 所示。

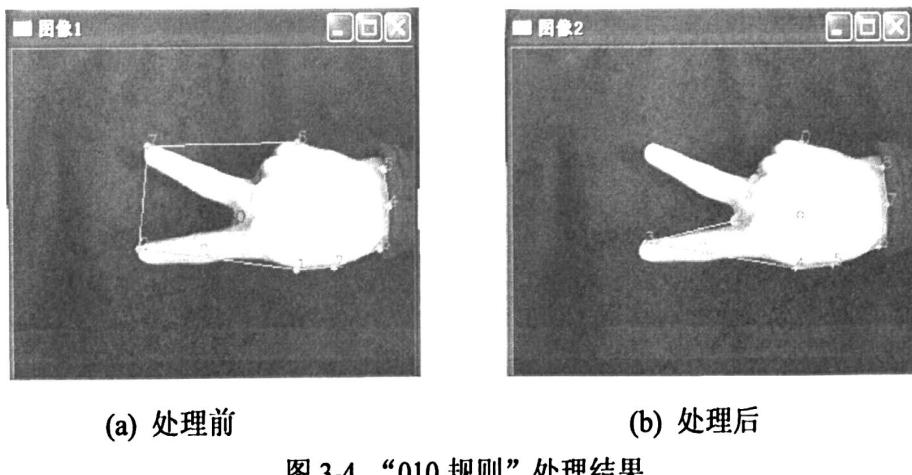


图 3-4 “010 规则”处理结果

由图 3-4 可以看出，通过“010 规则”处理后的外接多边形有效去除了两个缺陷点之间的手指部分，使得外界多边形结果更靠近手掌的外接多边形。

(2) 规则二：五个连续的凸点保留中央的凸点

观察轮廓序列点发现其中有一段连续的凸点，这部分对于手势图像中手掌靠近手腕的部分，我们的目标获得手掌部分，所以这部分的凸点是需要保留的，具体实现的时候，遍历左右的点，如果发现 5 个连续的点的值都为 1，那么就保

留中间的那个点，同样的原因，我们也叫这个规则为“11111”规则，这个规则和检测凸多边形的时候是联合起来用的，在删除不符合条件的点后会进行凸多边形检测，因为检测凸多边形的算法的原因（下面的章节会介绍），如果没有“11111”规则的限制，和可能删除手掌轮廓上的点。具体的程序实现如下。

```
count = pts.size();
for (i = 0; i < count; i++) //遍历，连续凸点中间点不能被删除
{
    if (v[i].type & v[i+1].type & v[i+2].type & v[i+3].type & v[i+4].type)
    {
        pts[i].type = 0;
    }
}
```

（3）规则三：缺陷点不能被删除

在手势轮廓图像上面看的很清楚，缺陷点一般都处于手指根部，由于我们要提取的是手掌的轮廓，很明显所有的缺陷点在手掌轮廓线上面，因此在删除不符合条件的点时，不能删除值为 0 的点，这也给后面检测轮廓的凸多边形提供了基础。

（4）规则四：检测是否是凸多边形

由于手掌轮廓的外接多边形是一个凸多边形，所以我们想到利用判断得到的外接多边形是否是凸多边形，如果不是凸多边形，那说明还有手指包括在轮廓中，这时需要删除一些点来保证得到的结果是凸多边形，结果越接近凸多边形就越接近手掌的轮廓，因此在具体实现的时候，利用一个循环来判断结果是否是凸多边形，如果不是就删除不满足条件的点，知道满足凸多边形的条件为止，这时得到的凸多边形也就近似为手掌的轮廓了。

在叙述如何检测是否是凸多边形之前我们先来看下凸多边形的定义。所谓凸多边形，就是把一个边形任意一边向两方无限延长成为一条直线，如果多边形的其他各边均在此直线的同旁，那么这个多边形就叫做凸多边形，也可以理解为通过凸多边形的任意一条边作平面，并与此多边形所在的平面相异，那么凸多边形的其他所有部分都在所作平面的同一侧^[40]。例如手势 0 的外界多边形就是一个凸多边形，如图 3-5 所示。从几何的角度思考，我们可以考虑利用凸多边形的一些性质来判断，但是那样会比较复杂，因为我们在实现了 010 规则和 11111 规则后得到的多边形有明显的特征，如图 3-6 所示。

由图 3-6 可以看出，只需要删除手指指尖的点就可以去除所有伸出的手指部分，并且得到外接多边形近似手掌轮廓的凸多边形。

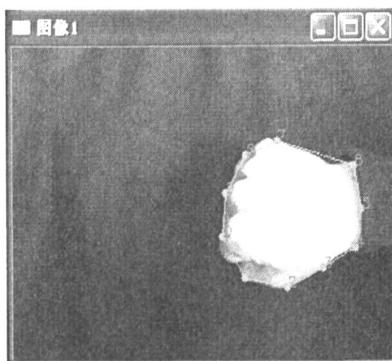


图 3-5 手势 0 的外接多边形

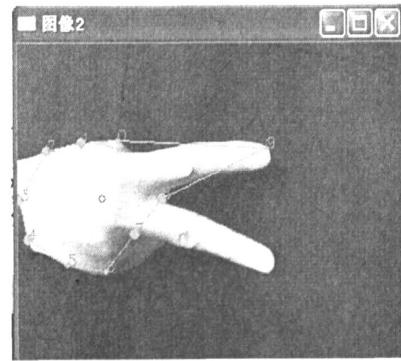


图 3-6 按规则删除一些点后的图像

下面我们来研究以下如何删除这个不满足凸多边形规则的点。单独把经过以上三个规则处理后的外接多边形拿出来分析，很明显的要保证所得到的外接多边形是凸多边形，必须设法删除点 p_3 ，为了方便叙述原理，将多边形放到二维的坐标系中，如图 3-7 所示，设定图中 p_1, p_2, p_3 三个点的左边分别是 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ ，从 p_1 的前一个点开始，我们按照逆时针的方向顺序观察图中的点，发现从 p_2 前面的点都是按照逆时针方向旋转的，但是从 p_2 到 p_3 ，方向变成了顺时针方向，具体的从数学的角度来，我们分析相邻两点之间的线段的斜率变化。

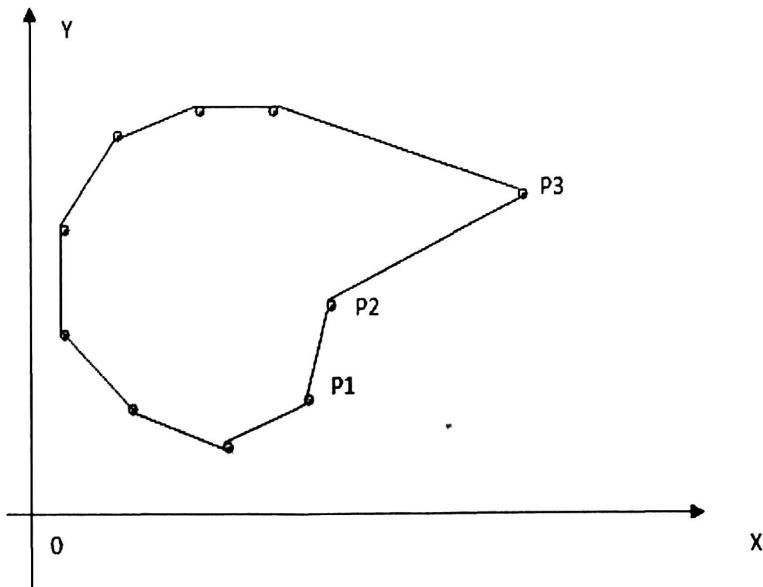


图 3-7 外接多边形的坐标系表

按照逆时针方向看, 线段 $\vec{p_1p_2}$ 与它之前相邻的线段的斜率是不断增大的, 但是从线段 $\vec{p_1p_2}$ 到线段 $\vec{p_2p_3}$ 的斜率发生突变, 相反的从顺时针方向看, 斜率变化情况改变也是从 p_3 这一点开始, 从这一点分析, 我们可以得出删除不满足凸多边形的那个点, 在图 3-7 中就是点 p_3 。利用计算斜率的公式, 我们得到线段 $\vec{p_1p_2}$ 的斜率和线段 $\vec{p_2p_3}$ 的斜率分别如下:

$$\frac{dy_0}{dx_0} = \frac{y_2 - y_1}{x_2 - x_1}, \quad \frac{dy}{dx} = \frac{y_3 - y_2}{x_3 - x_2} \quad (3-1)$$

由于斜率有正有负, 单纯的拿来比较会出现多种情况需要讨论, 我们将公式 (3-1) 同时都乘以 dx_0dx , 这样我们得到如下的结果:

$$dy_0dx = \frac{y_2 - y_1}{x_2 - x_1} \cdot dx_0, \quad dydx = \frac{y_3 - y_2}{x_3 - x_2} \cdot dx \quad (3-2)$$

这样也就只需要考虑讨论两种情况了, 分别是 dx_0dx 大于 0 和 dx_0dx 小于 0 的情况。我们以 $dydx_0 > dy_0dx$ 作为判断的标准, 遍历整个多边形的所有的点。

当 $dx_0dx > 0$ 时, 如果 $dydx_0 > dy_0dx$, 两边同时除以 dx_0dx , 不等式不要反号, 我们得到的不等式就变成了如下情况:

$$\frac{dy}{dx} > \frac{dy_0}{dx_0} \quad (3-3)$$

当 $dx_0dx < 0$ 时, 如果 $dydx_0 > dy_0dx$, 两边同时除以 dx_0dx , 这时不等式需要反号, 我们得到的不等式就变成了如下情况:

$$\frac{dy}{dx} < \frac{dy_0}{dx_0} \quad (3-4)$$

从上面的讨论可以看出, 当连续的按照顺序遍历多边形的点的时候, 不等式 $dydx_0 > dy_0dx$ 的结果情况完全可以表示斜率的变化情况。

在程序的实现上面, 我们运用了位运算来判断点斜率的变化, 关键性的代码只有一行, 如下:

```
orientation |= (dydx0 > dx0dy) ? 1 : ((dydx0 < dx0dy) ? 2 : 3);
```

从上面的程序看到, 变量 orientation 可以分别得到 1, 2 和 3 几个数值, 当不等式 $dydx_0 > dy_0dx$ 成立时, $orientation=1$, 表示成二进制数就是 $orientation=01$; 当不等式 $dydx_0 > dy_0dx$ 不成立而不等式 $dydx_0 < dy_0dx$ 成立时, $orientation=2$, 表示成二进制数就是 $orientation=10$ 。我们做一个遍历循环, 利用位运算 “|” 来计

算前后相邻的两个点的斜率变化,如果不等式 $dydx_0 > dy_0dx$ 一直成立,那么 orientation 结果就一直是二进制数 01 和 01 做或运算,得到的结果仍然是 01,或者当不等式 $dydx_0 < dy_0dx$ 一直成立时, orientation 结果就一直是二进制数 10 和 10 做或运算,得到的结果就是 10,只有当前一个点不等式 $dydx_0 < dy_0dx$ 成立,而紧接着的不等式 $dydx_0 > dy_0dx$ 一直成立,或者反过来前一个点不等式 $dydx_0 > dy_0dx$ 一直成立,而紧接着的点不等式 $dydx_0 < dy_0dx$ 成立时,才会出现 10 和 01 做或运算或者 01 和 10 做或运算时,这时得到的 orientation 结果的结果才会是 11,很明显,当 orientation=3 时,表示在当前点斜率发生变化,该点就是要删除的不满足凸多边形条件的点。经过这条规则删除点后得到的就是近视于手掌轮廓的外接多变形了,如图 3-8 所示。

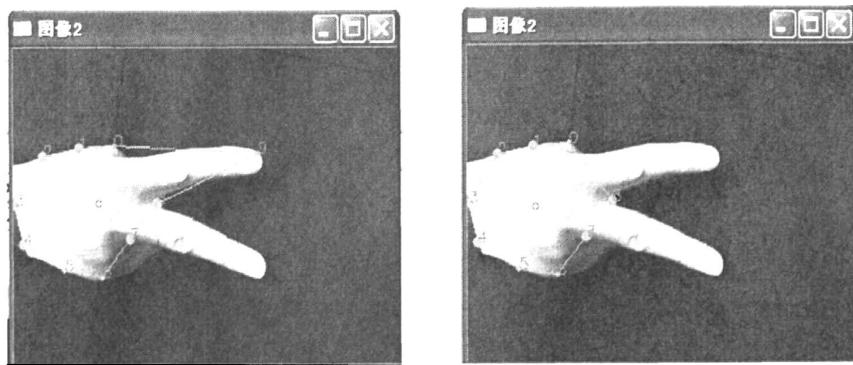


图 3-8 凸多边形检查结果

由图 3-8 可以看出,去除编号为 9 的顶点后得到的外接多边形刚好近似与手掌轮廓图像的外接多边形,通过手掌的外接多边形我们就可以近似求出手掌的重心和半径了。

3.3 提取手指个数作为识别的特征向量

提取手掌轮廓的最终目的是提取每个手势图像中伸出来的手指的个数,尽管有些手势伸出的手指个数是一样的,例如手势 2、手势 6 和手势 8 都是伸出了两个手指,但是他们伸出的手指的状态是不一样的,所以手指个数作为一个特征向量来识别手势是非常有利的。本文在提取手掌轮廓的基础上,提出了几种求手指个数的方法,下面详细介绍。

3.3.1 画圆法求手指个数

为了提取手指的个数，最开始的想法是利用整个手势轮廓的重心，以重心为圆心画圆，圆弧与手指相交的特征很明显，只要我们数出相交的小圆弧的段数就可以知道手指的个数了，这个方法在实际的研究过程中发现效果并不是很理想，有时候手指没有被相交，产生这个问题的原因在于整个手势轮廓的重心太靠近手指的方向，如果以它为圆心画圆，半径选择稍微不当就可能出现这种状况。因此，要利用画圆的方法来求手指的个数，圆的圆心和半径如何确定是关键。观察所有手势的几何特征发现，手掌区域轮廓的外界多边形近似一个圆形，因此想到如果我们能找到手掌的重心，以手掌重心画比手掌稍微大一点的圆刚好就可以切到所有伸出来的手指了。

由于前面的研究我们已经得到手掌轮廓的外接多边形和外界多边形的顶点信息，因此确定手中的重心近似与求这个外接多边形的形心，更确切的说就是计算手掌外界多边形的顶点集的形心，下面的程序段求出了这个中心点的位置，并且把它画在了完整的手势轮廓图像上，画在完整手势轮廓图像上的原因是后面画的圆是要与手指相交的，因此圆也是画在完整的手势轮廓图像上。

```
//计算手掌点集 pts 的形心，并画在 src 上
void calCenter(IplImage* src,vector<CvPointEx> pts,CvPoint* center){
    int count = pts.size();
    int x = 0, y = 0;
    int xcen = 0, ycen = 0;
    for(int i = 0; i < count; i++){
        x += pts[i].pt.x;
        y += pts[i].pt.y;
    }
    xcen = (x / count); //求出圆心的横坐标
    ycen = (y / count); //求出圆心的纵坐标
    center->x = xcen;
    center->y = ycen;
    cvCircle(src,*center,3,CV_RGB(233,233,0),CV_FILLED,8,0); //画出圆心
}
```

求出了圆心，下面的工作就是确定要画的圆的半径，求出与手指相交的个

数了。画圆的目的是为了与手势中伸出来的手指相交，我假设手掌轮廓的重心为 R_0 ，要画的圆的半径为 r ，一个是整个手势轮廓中距离手掌轮廓重心最远的凸点 R_1 ，另一个是手掌轮廓的外接多边形中距离 R_0 最近的点 R_2 ，将 R_0 和 R_1 之间的距离定为 D ， R_0 和 R_2 直接的距离定为 d ，显然，如果 $r > D$ 或者 $r < d$ 的时候，圆与伸出的手指是没有相交的，合理的半径范围应该是 $d < r < D$ 。考虑到每个手势的中手指伸出的情况不同，每个手指的长度也不同，为了能把所有伸出来的手指都找出来，选择了 $r = 0.7 \times D$ 作为圆的半径。

在确定了圆心和确定圆的半径之后，求出每个手势伸出的手指个数就很好得到了，如图 3-9 中的图(a)和图(b)所示，整个手势轮廓图像中，画出手掌重心作为圆心，以设定好的半径画圆会伸出来的手指相交，数出相交的小弧段数就可以确定手指个数了，由于圆弧可能与手掌相交留下圆弧段，我们设定大于手掌半径一半的圆弧不被计算，因为手指圆弧段长度一般都小于手掌半径的一半。

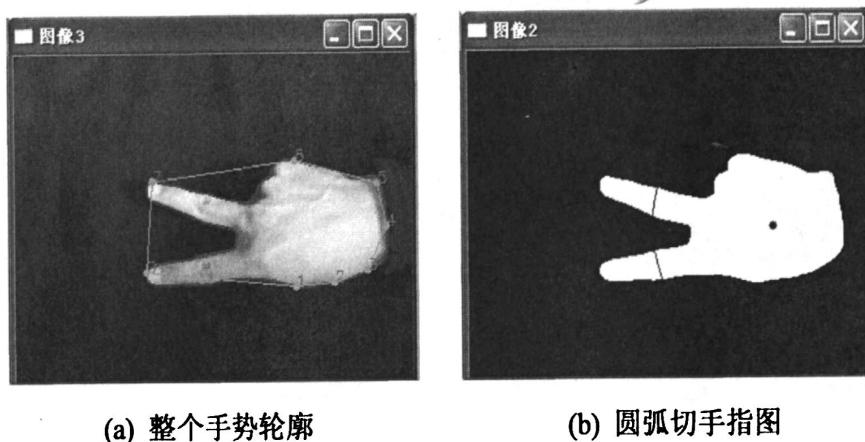


图 3-9 手指个数的计算结果

由图 3-9 的结果可以看出，通过手掌重心和半径确定的圆弧刚好可以切到手指中部，通过与手指相交的小圆弧段的个数得到手指的个数，求出的手指个数也非常准确。

3.3.2 顶点数法求手指个数

观察完整手势轮廓图像的外接多边形，大多数情况下手指指尖部分都有一个或者两个顶点，结合这一点，在完整手势轮廓上面画圆之后，我们发现，手指指尖上的点都在圆的外面，于是想到如果能够确定圆外手指指尖点的个数不

就得到了伸出的手指的个数了。但是这种方法有一个弊端，就是当一个手指的指尖处有两个顶点的画，得到的手指个数就不准确了，如图 3-10 所示，当前手势伸出的手指个数是 1，但是数出来的顶点个数是 2，这样就会出现较大的误差。

为了解决这个问题，我们先对所有的手势轮廓外接多边形上的点编号，如图 3-10 中图(a)所示。观察这些的分布情况，比较相邻两个顶点之间的距离，发现指尖上的两个顶点之间的距离相比较其他顶点的距离要小很多，基于这个原理，我们设定，在得到圆外面的顶点集合之后，需要对他们相互之间的距离进行检查，如果两个相邻点之间的距离大于手掌半径的，则保留相应的点，如果不满足条件则删除相应的点，这样统计下来的顶点个数就是手势中伸出来的手指的个数了。



(a) 指尖有两个顶点的情况

(b) 手指个数输出

图 3-10 指尖有两个顶点的情况

从图 3-10 中的图(a)可以看出，手指之间有两个顶点，编号为 4 和 5，此时伸出的手指个数为 1，但是圆外顶点有两个，通过对相邻点之间的距离排除 5 号点，得到手指个数为 1 个，输出结果如图 3-10 中的图(b)。由输出结果看出，计算得到的手指个数非常准确。

3.4 基于轮廓图像的不变矩的提取

在手势识别的研究中，由于识别对象比较单一，手势的几何特征并不能完全区分各种手势要表达的意义，因此出了提取手势的几何特征作为识别特征向量之外，还必须提取其他的特征向量作为识别参数。在数字图像处理中，图像的矩函数是一种非常有区分度的图像描述子，自 Hu 提出矩的不变量理论以来，

不变矩在模式识别及图像分析等领域都有广泛而成功的应用。图像不变矩分为灰度直方图不变矩和空间不变矩，本文中只研究空间不变矩，空间不变矩是指在对图像平移、旋转和尺度变换后，对比度变化具有恒定性。不变矩就是一种通过提取平移、旋转和比例不变性的图像特征，不变矩的主要思想是使用对变换不敏感的基于区域的几个矩作为特征向量，从而进行图像识别的方法^[41]。本文中分析研究手势图像的几种图像矩，计算他们的不变量，并且应用于手势识别中。

3.4.1 Hu 矩的不变量的研究

Hu 矩又称为几何不变矩，它由 Hu 在 1962 年提出来的，具有平移、旋转和尺度不变性^[42]。我们先看看图像基本的几何矩的定义。

对于灰度分布为 $f(x, y)$ 的图像，它的 $(p+q)$ 阶几何不变矩 m_{pq} 定义为：

$$m_{pq} = \iint x^p y^q f(x, y) dx dy \quad p, q = 0, 1, 2, \dots \quad (3-5)$$

$(p+q)$ 阶中心距 μ_{pq} 定义为：

$$\mu_{pq} = \iint (x - x_0)^p (y - y_0)^q f(x, y) dx dy \quad (3-6)$$

其中，矩心（或者叫重心） (x_0, y_0) 为：

$$x_0 = \frac{m_{10}}{m_{00}}, y_0 = \frac{m_{01}}{m_{00}} \quad (3-7)$$

相应的，在数字图像处理中，对于离散为 $M \times N$ 的数字图像 $f(x, y)$ ，它的 $(p+q)$ 普通几何矩 m_{pq} 定义为：

$$m_{pq} = \sum_{x=1}^M \sum_{y=1}^N x^p y^q f(x, y) \quad (3-8)$$

其中中心矩 μ_{pq} 定义为：

$$\mu_{pq} = \sum_{x=1}^M \sum_{y=1}^N (x - x_0)^p (y - y_0)^q f(x, y) \quad p, q = 0, 1, 2, \dots \quad (3-9)$$

其中，矩心（或者叫重心） (x_0, y_0) 为：

$$x_0 = \frac{m_{10}}{m_{00}}, y_0 = \frac{m_{01}}{m_{00}} \quad (3-10)$$

其中， m_{10} 和 m_{01} 是图像的一阶几何矩， m_{00} 是图像的 0 阶几何矩。

由式(3-8)、式(3-9)和式(3-10)可以得到图像的不大于三阶的中心矩分别为 μ_{00} , μ_{01} , μ_{10} , μ_{11} , μ_{02} , μ_{20} , μ_{12} , μ_{21} , μ_{03} , μ_{30} 。对于一般的灰度图像而言,中心矩有以下规律:

(1) μ_{02} 和 μ_{20} 是区域灰度分别围绕通过灰度中心的垂直轴线和水平轴线的惯性矩。若 $\mu_{20} > \mu_{02}$, 则说明图像在水平方向上拉长;反过来则说明图像在垂直方向上拉长。

(2) μ_{03} 和 μ_{30} 的幅值可以用来度量物体分别对于垂直轴线和水平轴线的对称性。若 $\mu_{03} = 0$, 则物体对于垂直轴线是对称的;若 $\mu_{30} = 0$, 则物体对于水平轴线是对称的。

当图像发生变化的时候, m_{pq} 也发生变化, 导致 μ_{pq} 具有平移不变性, 但对旋转依然敏感, 以此, 直接用普通矩和中心矩进行特征表示, 不能使特征向量同时具有平移、旋转和比例不变性, 如果利用归一化的中心距, 则特征向量不仅具有平移不变性, 而且还具有平移不变性^[43]。

归一化的中心矩 λ_{pq} 的定义为:

$$\lambda_{pq} = \frac{\mu_{pq}}{\mu_{00}}, \quad (3-11)$$

其中:

$$r = \frac{p+q+2}{2} \quad p+q = 2, 3, 4, \dots \quad (3-12)$$

Hu 利用 2 阶中心矩和 3 阶中心矩构造了七个不变矩的集合 I , 它们在连续图像条件下可保持平移、缩放和旋转不变^[33], 具体的定义如下:

$$I_1 = \lambda_{20} + \lambda_{02} \quad (3-13)$$

$$I_2 = (\lambda_{20} + \lambda_{02})^2 + 4\lambda_{11}^2 \quad (3-14)$$

$$I_3 = (\lambda_{30} + 3\lambda_{12})^2 + (3\lambda_{21} - \lambda_{03})^2 \quad (3-15)$$

$$I_4 = (\lambda_{30} + \lambda_{12})^2 + (\lambda_{21} + \lambda_{03})^2 \quad (3-16)$$

$$I_5 = (\lambda_{30} - \lambda_{12})(\lambda_{30} + \lambda_{12}) \left[(\lambda_{30} + \lambda_{12})^2 - 3(\lambda_{21} + \lambda_{03})^2 \right] + (3\lambda_{21} - \lambda_{03})(\lambda_{21} + \lambda_{03}) \left[3(\lambda_{30} + \lambda_{12})^2 - (\lambda_{21} + \lambda_{03})^2 \right] \quad (3-17)$$

$$I_6 = (\lambda_{20} - \lambda_{02}) \left[(\lambda_{30} + \lambda_{12})^2 - (\lambda_{21} + \lambda_{03})^2 \right] + 4\lambda_{11}(\lambda_{30} + \lambda_{12})(\lambda_{21} + \lambda_{03}) \quad (3-18)$$

$$I_7 = (3\lambda_{21} + \lambda_{03})(\lambda_{30} + \lambda_{12}) \left[(\lambda_{30} + \lambda_{12})^2 - 3(\lambda_{21} + \lambda_{03})^2 \right] + (\lambda_{30} - 3\lambda_{12})(\lambda_{21} + \lambda_{30}) \left[3(\lambda_{30} + \lambda_{12})^2 - (\lambda_{21} + \lambda_{03})^2 \right] \quad (3-19)$$

计算 Hu 矩的七个不变量的具体程序实现在下一节中会有详细的介绍。

3.4.2 基于轮廓图像的不变矩计算

与传统的图像识别问题一样，本文也选取每种手势图像矩不变量作为重要的识别参数。在得到了手势图像完整的轮廓之后，我们对轮廓图像求不变矩。基于上一节叙述的几何不变矩原理，我们得到计算七个不变量的方法。计算离散图像不变矩的主要步骤如下：

- (1) 根据公式 (3-8) 计算手势轮廓图像的 $(p+q)$ 普通几何矩 m_{pq} ，然后根据公式 (3-10) 得到手势图像的重心 (x_0, y_0) 。
- (2) 利用公式 (3-9) 计算图像的 $(p+q)$ 阶中心距 μ_{pq} ，然后根据公式 (3-11) 和 (3-12) 对距 μ_{pq} 进行归一化得到归一化中心矩 λ_{pq} 。
- (3) 在得到归一化中心矩 λ_{pq} 之后，根据公式 (3-13) 到公式 (3-19) 就可以计算得到七个不变量。

在程序实现上，主要运用了 OpenCV 函数库中的专门计算 Hu 矩的函数，函数 `cvGetHuMoments()`，计算不变量的主要代码如下：

```
void getHu(CvSeq* HandContour,IplImage* handcontour_img,CvHuMoments* hu)
{
    CvMoments moments;
    cvZero(handcontour_img); //所有元素置零*
    cvDrawContours(handcontour_img, HandContour, CV_RGB(255, 255, 255),
                   CV_RGB(255, 255, 255), 0, 2, 8); //绘制轮廓
    //Hu 矩计算
    cvMoments(handcontour_img,&moments, 0);
    cvGetHuMoments(&moments,hu);
}
```

计算得到的 7 个不变量存到结构体 `CvHuMoments{}` 中，定义如下：

```
typedef struct CvHuMoments{
```

```

    double hu1, hu2, hu3, hu4, hu5, hu6, hu7; /* Hu invariants (不变量) */
}CvHuMoments;

```

其中 hu1, hu2, hu3, hu4, hu5, hu6, hu7 分别表示七个不变量，在识别之前，会对这 7 个参数进行分析，选择其中相对有效的特征参数，这部分内容会在第 4 章中详细分析。利用上面叙述的方法分别对图 2-2 中的十个手势图像计算不变量，需要说明的是，本研究中计算 Hu 矩不变量都是基于轮廓图像的，计算得到的结果如表 3-1 所示。

表 3-1 Hu1~Hu7 的计算列表（保留五位小数）

手势	Hu1 (10^{-3})	Hu2 (10^{-7})	Hu3 (10^{-8})	Hu4 (10^{-9})	Hu5 (10^{-17})	Hu6 (10^{-12})	Hu7 (10^{-18})
0	8.97225	8.77079	1.62901	4.59926	4.48586	5.42537	-3.44809
1	1.03715	3.62643	1.09437	3.08521	1.34760	2.62817	4.06743
2	8.66334	2.63198	2.26593	4.56550	1.38996	2.26736	4.21721
3	6.73831	9.64397	1.88530	4.41835	1.20042	1.31682	-2.15506
4	5.56104	4.96254	3.05617	3.07682	9.36274	6.45037	-2.43342
5	5.01566	1.06899	1.59692	1.18169	3.37921	1.16277	-3.85446
6	7.45781	1.27794	1.34423	8.88622	5.52411	-4.58480	1.06819
7	7.90620	9.59369	5.41559	4.78611	-6.69167	1.37454	4.61635
8	7.53965	5.85173	5.29166	8.02492	1.22880	8.69439	5.69259
9	6.41362	5.31489	2.41673	1.67781	3.14284	3.25590	-9.31758

从表中可以看出，每种手势的不同不变量之间的差别还是比较大的，能够作为有效特征参数区分各种手势。

3.5 本章小结

本章先求出手势图像的轮廓图像和外接多边形，然后分析外接多边形的各个顶点缺陷点的特点，将所有顶点按照凸点和缺陷点分类并且排序编号，在此基础上利用四个规则求出手掌轮廓的近似外接多边形，接着分析特征参数手指个数的计算方法，最后计算基于轮廓图像的 Hu 矩不变量。本章求出了十个特征参数，为后面对各种手势做了准备。

第4章 方案实现和结果比较分析

4.1 识别系统的整体设计

在获取识别手势图像的所有特征参数之后，设计了一套手势识别系统，系统框图如图 4-1 所示。

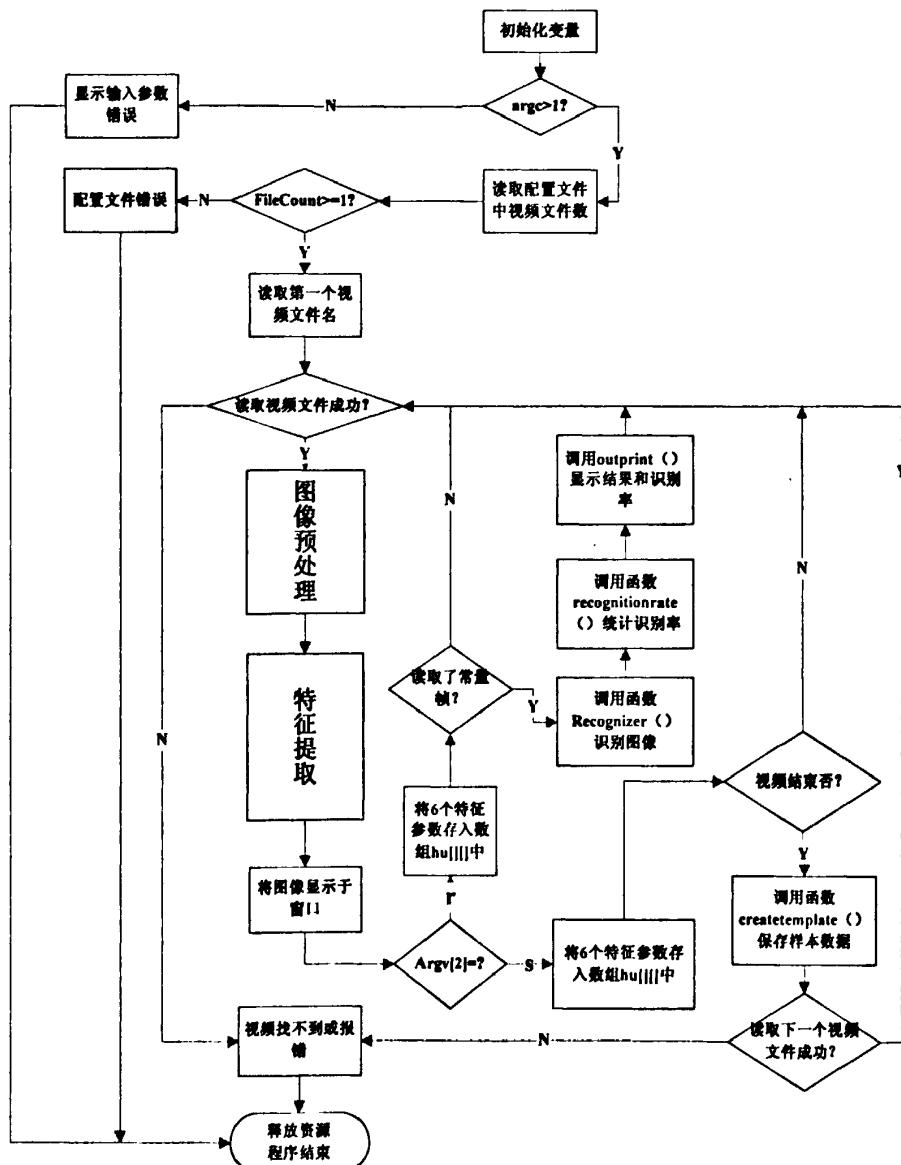


图 4-1 系统整体程序流程图

由图 4-1 可以看出，整个系统架构包括图像预处理、特征提取和图像识别三个大的部分，具体步骤如下：

(1) 首先是通过 xml 文件读取视频文件，将文件信息存到结构体 ImageProperty{} 中^[38]。结构体定义如下：

```
struct ImageProperty
{
    queue<string> fName;      //每一帧图像的文件名
    vector<int> fingercount;    //对应的理论手势值
    IplImage* pimage;
    bool isFirstFrame;         //标记是否是文件首帧
    bool isChange;              //标记是否更换手势值
    int position;               //记录图像属于第几个文件
    int frameCount;             //标记图像是同一手势文件的第几帧
} imagePro;
```

这里通过 isFirstFrame 标记是否是文件第一帧图像，通过 isChange 标记是否是下一个文件，由此来循环读取处理每一个手势文件。另外此处通过 OpenCV 函数库中的 cvGetCaptureProperty() 计算视频文件的总的图像帧数，方便后面计算特征参数的平均值。

(2) 读取视频文件成功后就顺序读取每个文件的每一帧图像，进行图像预处理过程，得到手势的二值化图像。

(3) 计算每一帧图像的特征参数将结果保存到结构体 GestureParams{} 中。结构体定义如下：

```
struct GestureParams
{
    float Hu1;      //Hu 矩第一个不变量
    float Hu2;      //Hu 矩第二个不变量
    float Hu3;      //Hu 矩第三个不变量
    float Hu4;      //Hu 矩第四个不变量
    float FingerCount; //手指个数
    float LenAreaRatio; //轮廓图像的面积与周长的比值
} gesResult;
```

其中第五个参数和第六个特征参数有三种情况，第一种情况是 Hu 矩不变量 Hu5 和 Hu6，第二种情况是缺陷个数和轮廓图像的面积与周长的比值，第三种情况是

手指个数和轮廓图像的面积与周长的比值。后面的对比试验中，我们通过测试这三组特征参数的识别率来分析所有特征参数的有效性。

(4) 识别具体的手势图像，这里通过主函数 int main(int argc, char* argv[])入口参数 argv[]的值确定是标准样本学习过程还是新样本识别过程，当 argv[]的值为 s 时就是学习模式，当 argv[]的值是 r 时就是识别模式。

(5) 输出识别结果。

由于图像预处理模块程序比较简单，这里就不详细介绍了，大致流程可以参考图 2-5。下面分别介绍特征参数计算模块和识别结果输出模块。

4.1.1 特征参数计算模块

根据人手部 Tortoise 模型的分析，得到研究需要计算的 10 个特征参数，分别是 7 个 Hu 矩不变量(Hu1, Hu2, Hu3, Hu4, Hu5, Hu6, Hu7)、手势轮廓图像的面积与周长的比值 LenAreaRatio、手势伸出的手指个数 FingerCount 和缺陷个数。具体计算程序流程图如图 4-2 所示。

在计算计算手指个数的 FingerCount 的值的过程中，定义了两个重要的结构体，用于存储中间结果。具体定义如下：

```
typedef struct CvPointEx
{
    CvPoint pt;
    int type; //凸点 1, 凹点 0,
    int position;//用于标注凸点和凹点的位置信息
}CvPointEx;

struct GestureImage
{
    IplImage* pFrImg;
    IplImage* handcontour_image;
    IplImage* templImg[7];
    CvSeq* contourX;           //大模板轮廓
    CvSeq* HandContourX;
    CvSeq* HandContour2X;
    vector<CvPoint> HandPt;
    vector<CvPoint> HandPt2;
```

```

vector<CvPointEx> HandPoints;
CvSeq* contour;           //小模版轮廓
CvSeq* HandContour;
CvSeq* HandContour2;
} gesimage;

```

其中结构体 GestureImage 前三个元素用来存在中间图像，contour、HandContour、HandContour2 分别用来存储整个手的轮廓、简化的轮廓和缺陷信息的轮廓，其中的两个 vector 用来存储轮廓中凸点和凹点的信息。

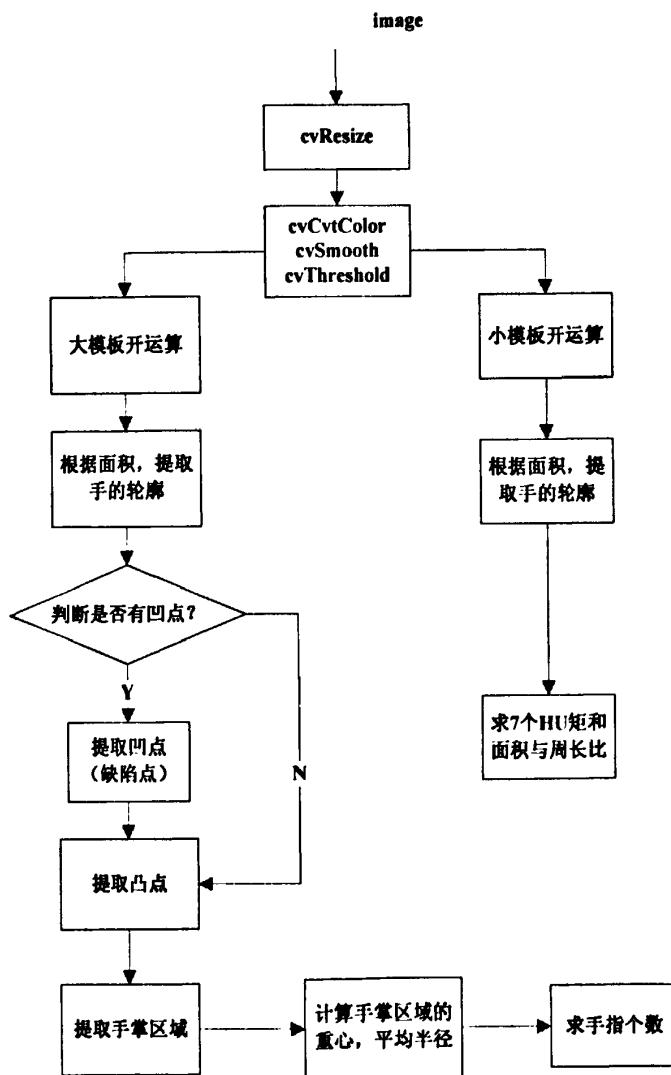


图 4-2 特征参数计算程序流程图

4.1.2 识别结果输出模块

在识别了 10 帧图像后统计识别结果，并且计算每一帧图像的识别率，结果存入结构体 ResultParameter{} 中。识别结果输出模块的程序流程如图 4-3 所示。

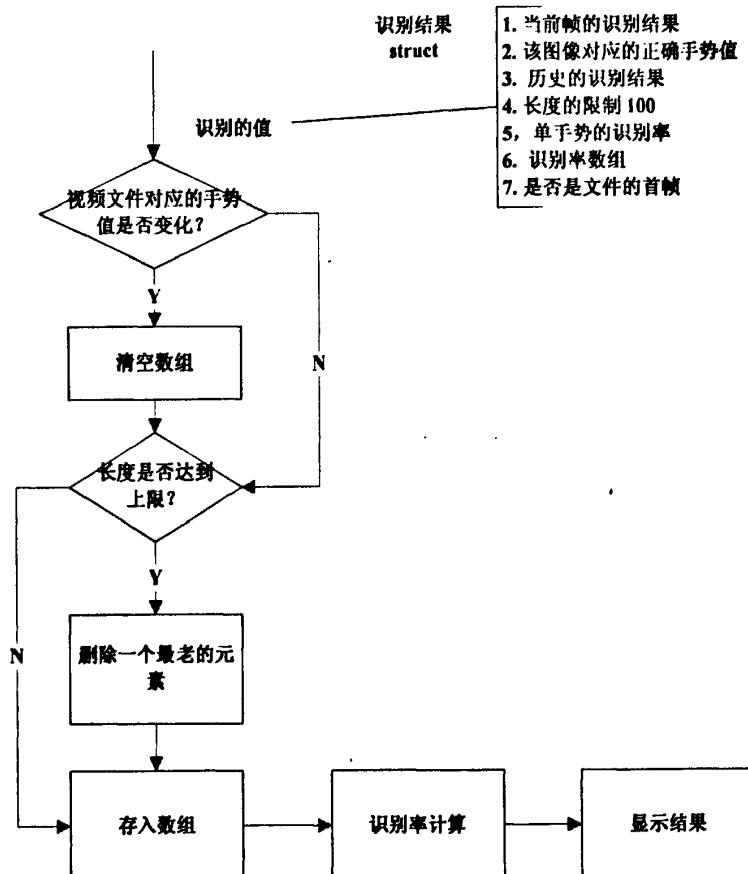


图 4-3 识别结果输出流程图

结构体的定义如下：

```

struct ResultParameter
{
    int curResult[FRAME_NUM]; //当前及历史的识别结果
    int RSLabel[FRAME_NUM]; //当前手势的正确性标志
    char Ratio[GESTURE_COUNT]; //所有手势的识别率
    CvSeq* HandContour; //手的轮廓
    vector<CvPointEx> HandPaw; //手掌轮廓
}

```

```

vector<CvPoint> HandTU;      //凸点
vector<CvPoint> HandAO;      //凹点、缺陷点
CvPoint Center;              //重心
int FingerCount;             //手指个数

} resParameter;

```

可以看出，结构体包括每帧图像的所有结果信息，其中手的轮廓、手掌的轮廓、凸点、缺陷点、重心的确定可以帮助我们在原图像中画出这些信息。

4.2 基于模版匹配的手势识别算法

为了验证前面求出的十个特征参数的有效性，本文并没有采用识别能力很强的识别方法，比如基于 SVM（支持向量机）或者神经网络的识别方法，而是采用了识别能力稍差的模版匹配的方法，因此识别能力太强的识别算法并不能明显区分各个特征参数的有效性，这样在后面的实验中，三组特征参数的有效性测试结果对比更加明显。

在机器识别事物的过程中，常常需要把不同传感器或同一传感器在不同时间、不同成像条件下对同一景象获取的两幅或多幅图像在空间上对准，或根据已知模式到另一幅图像中寻找相应的模式，这就叫匹配。模版匹配就是根据已知模版来计算待识别目标与模版的匹配程度，匹配程度最高的就是识别结果。这里有两个前期工作需要做，一个是建立标准的已知模版，另一个是建立评判匹配程度的机制。

本文先计算的手势图像文件（如图 2-2 所示）的每一帧图像的 6 个特征参数和每个视频文件的帧数 frame_count，将它们分别存到数组 hu[][][6]中，这样就得到一个行数为 frame_count，列数为 6 的数组，分别对每一列计算数学平均值，就可以得到一组代表该手势的特征参数。

分别对代表十个手势文件进行上述的计算，就可以得到代表 10 组分别代表每一种手势的标准的特征参数了，我们以这 10 组特征参数作为匹配每一种手势的模版，即总共有十个模版。至于匹配程度的评判机制，本文利用了欧式距离的公式来计算。欧式距离（Euclidean distance）也称欧几里得距离，它是一个通常采用的距离定义，它是在 n 维空间中两个点之间的真实距离^[44]。在 n 维空间中，点 (x_1, x_2, \dots, x_n) 和点 (y_1, y_2, \dots, y_n) 的欧式距离公式具体定义如下：

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (4-1)$$

由公式 (4-1) 可以看出在二维和三维空间中的欧式距离的就是两点之间的距离。基于这个原理, 我们将样本空间看成一个 6 维的空间, 将手势图像的 6 个特征参数组成一个向量, 很明显这个向量就是空间中的一个点, 前面提到的 10 个模版也可以表示成空间中的 10 个点, 这样我们就可以利用公式计算待识别的图像在 6 维空间中代表的点与 10 个模版在 6 维空间中代表的点之间的距离, 这个点与十个点中的哪个的距离最短, 识别的结果就是哪个点代表的手势。

利用以上原理, 设计了一种识别算法, 算法流程图如图 4-4 所示。其中函数 `createtemplate(float*, int,int)` 是用来计算标准手势图像的 6 个特征值的, 即我们说的模版, 并将结果写入 xml 文件中。而函数 `Recognizer(float *hu_r)` 用于计算代表目标图像的点与代表模版图像的点之间的最小欧式距离, 即匹配的过程。

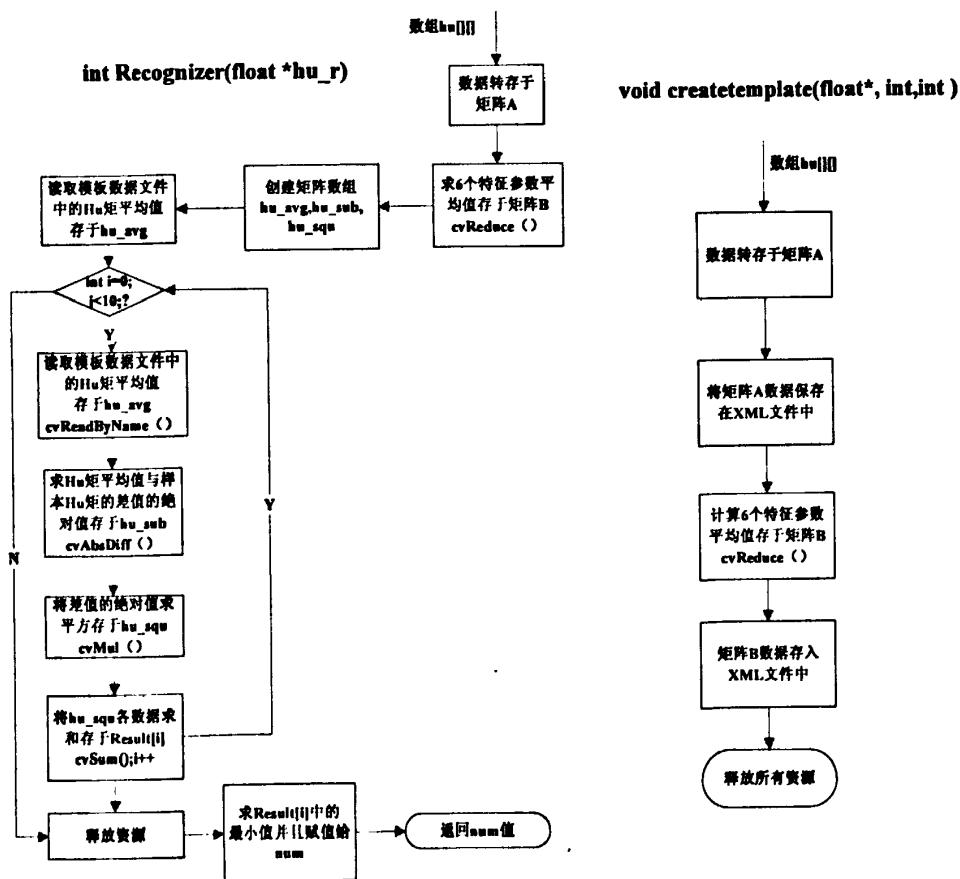


图 4-4 识别算法流程图

4.3 实验结果的比较与分析

4.3.1 特征参数结果分析

根据图 4-2 中描述的程序实现, 可以计算得到总共 10 个不同的特征参数, 分别是 7 个基于轮廓图像的 Hu 矩不变量 Hu1~Hu7、手势图像中伸出的手指个数、缺陷个数和手势轮廓图像的面积与周长的比值。针对标准的手势样本空间计算以上 10 个特征参数, 得到的结果如表 4-1 所示。

表 4-1 特征参数计算结果 (保留三位小数)

手势	Hu1 (10 ⁻³)	Hu2 (10 ⁻⁷)	Hu3 (10 ⁻⁸)	Hu4 (10 ⁻⁹)	Hu5 (10 ⁻¹⁷)	Hu6 (10 ⁻¹²)	Hu7 (10 ⁻¹⁸)	手指 个数	缺陷 个数	面积周 长比
0	8.972	8.771	1.629	4.599	4.486	5.425	-3.448	0.000	0.000	1.494
1	1.037	3.626	1.094	3.085	1.348	2.628	4.067	1.000	0.927	2.193
2	8.663	2.632	2.266	4.566	1.390	2.267	4.217	2.000	1.500	3.227
3	6.738	9.644	1.885	4.418	1.200	1.317	-2.155	3.000	2.234	4.003
4	5.561	4.963	3.056	3.079	9.363	6.450	-2.433	3.943	3.029	5.088
5	5.016	1.069	1.597	1.182	3.379	1.163	-3.854	5.000	3.946	5.925
6	7.459	1.278	1.344	8.886	5.524	-4.585	1.068	2.040	1.069	2.860
7	7.906	9.594	5.416	4.786	-6.692	1.375	4.616	1.000	2.000	2.083
8	7.540	5.852	5.292	8.025	1.229	8.694	5.693	2.000	2.357	3.024
9	6.414	5.315	2.417	1.678	3.143	3.256	-9.318	3.000	2.957	3.947

分析表 4-1 中的结果, 单看每个特征参数的值, 这十个特征参数对各个手势相互之间有一定的区分, 说明它们都有可能作为有效的特征参数。下面对它们逐一分析。

(1) 由前面对 Hu 矩不变量的研究知道, Hu1 和 Hu2 是基于二阶归一化中心矩构造得到的, Hu3、Hu4、Hu5、Hu6 和 Hu7 是基于三阶归一化中心矩构造得到的, 它们对二维物体的描述具有旋转、平移和缩放不变性, 但是由于较高阶中心矩对于成像过程中的误差、微小的变形等因素非常敏感, 所以由较高阶中心矩构造得到的不变量 Hu3、Hu4、Hu5、Hu6 和 Hu7 用于物体的识别有效性相对较差, 而低阶中心矩构造的不变量 Hu1 和 Hu2 用于物体识别有效性相当较

高；从另一个方面来看，我们观察表 4-1 中 Hu1、Hu2、Hu3、Hu4、Hu5、Hu6 和 Hu7 具体的值，Hu1、Hu2、Hu3、Hu4、Hu5、Hu6 和 Hu7 的数量级分别为 10^{-3} 、 10^{-7} 、 10^{-8} 、 10^{-9} 、 10^{-17} 、 10^{-12} 和 10^{-18} ，很明显，前 4 个不变量的数量级要小的多，相对于他们来说，Hu5、Hu6 和 Hu7 的值可以忽略不计；同样的，7 个 Hu 矩不变量相对于其它三个几何特征参数的值来说要小很多，因此在选定特征参数后要对计算结果进行相应的归一化处理。

(2) 从计算的结果来看，计算每个手势伸出的手指个数得到的结果是非常准确的，完全可以作为有效的特征参数，计算得到的缺陷个数的结果对于每个手势也有比较好的区分，不过没有手指个数那么明显。

单纯从手指个数和缺陷个数的计算结果看，经过平均以后的手指个数都几乎近视于整数，而经过平均以后的缺陷个数都是小数，这一点说明缺我们在求得的缺陷个数不稳定，这是因为每个手势伸出的手指个数是确定的，只要计算的正确，得到的结果几乎是不变的，例如手势 3 的平均手指个数是 3，是一个正确的结果，这说明计算每一帧图像得到的手指个数几乎都是 3，说明这个特征参数是非常稳定有效的；而缺陷个数因为对于缺陷的定义有一定的活动范围，定义多少的深度作为缺陷对于计算缺陷个数影响很大，另外，由于每个人做的手势会有千差万别，导致同一种手势不同的人做计算得到的缺陷个数很有可能不同，还是以手势 3 为例，我们计算得到的缺陷个数的平均值是 2.23998，出现这种结果的原因是计算每一帧图像的缺陷个数得到的结果有时候是 2 个有时候是 3 个，而且 2 个缺陷的情况居多，这说明缺陷个数的不稳定性。

另外，缺陷个数是基于手势图像的轮廓图像来计算的，因此获得的手势轮廓图像的好坏直接影响计算缺陷个数的正确性，这就会导致计算缺陷个数的前提条件更加精细，具体的说，就是在采集图像的时候必须保证适当的光照条件使得图像中手的部分亮度比较均匀，另外在做图像预处理的时候要寻求最适当的阈值分割方法，尽量在图像中把完整的手分割出来。相比较缺陷个数，求手指个数的条件就没有那么苛刻了，只要我们求出比较正确的轮廓，用上面提到的计算手指个数的方法得到的结果比较稳定的。当然在实际的应用中，预处理部分的效果越好，实验的结果也会越好。

(3) 第十个特征参数是手势轮廓图像的面积与周长的比值，结合人手部 Tortoise 模型分析每个手势，手掌面积和轮廓由于手势不同并不是一个固定值，它们的之间会有差别，而且每种手势伸出的手指长度不同，这些因素会直接影响手势轮廓图像的面积与周长的比值。从这个方面看，轮廓图像的面积与周长

的比作为特征参数可以有效的区分每个手势。

从理论上讲，以上这十个特征参数对于识别每个手势都有一定有效性，其中 7 个 Hu 矩不变量由于阶数过高，计算得到的结果在数量级上与其他几个特征参数的差距比较大，因此在选取特征参数是手势识别实验之前，必须对他们进行相应处理，保证选取的每个特征参数都有效。

4.3.2 归一化处理

由于摄像头采集手势图像的距离和角度变化会使得得到的样本出现缩放和旋转，这直接影响计算得到的特征参数的结果，为了减小这种影响，在计算得到所有的特征参数后，需要将结果进行归一化处理。另外，由于本文采用的是基于欧式距离的模版匹配识别算法，从前面章节介绍的识别算法的原理可以看出，训练得到的标准手势的特征参数和待识别样本的特征参数必须保持一致性，这样才能保证识别结果的正确性。基于以上两点原因，本文采用传统的缩放方法对每一列特征参数进行缩放，具体做法如下公式：

$$\text{value2} = \text{min} + \frac{(\text{max}-\text{min}) * (\text{value1}-\text{fmin})}{(\text{fmax}-\text{fmin})} \quad (4-1)$$

其中 value1 是原值， value2 是缩放后的值， $[\text{fmin}, \text{fmax}]$ 是缩放前的每一列特征参数的范围，而 $[\text{min}, \text{max}]$ 是缩放后的特征值范围。本文缩放后的特征值范围自定义为 $[-1, 1]$ ，因此公式 (4-2) 中的 min 和 max 的取值分别为 -1 和 1。本文在求得的表 4-1 中每一列特征参数后，遍历所有特征参数得到缩放前的最大值和最小值分别存到两个 XML 文件中，这里总结到表 4-2 中。

表 4-2 缩放前每一列的最大值和最小值（保留三位小数）

	Hu1 (10 ⁻³)	Hu2 (10 ⁻⁷)	Hu3 (10 ⁻⁸)	Hu4 (10 ⁻⁹)	Hu5 (10 ⁻¹⁷)	Hu6 (10 ⁻¹²)	Hu7 (10 ⁻¹⁸)	手指 个数	缺陷 个数	面积周 长比
fmax	14.165	78.843	46.418	97.757	53.863	43.225	-3.138	5.000	6.000	6.852
fmin	0.538	0.551	0.810	0.334	-8.048	-7.321	-9.887	0.000	0.000	1.230

经过上面的方法将特征参数缩放到 -1 到 1 之间，结果存入 txt 文件中，在后面识别的时候直接读取。

4.3.3 实验结果比较分析

实验中,为了测试各个特征参数的有效性,结合4.3.1节中对十个特征参数的分析,这里从十个特征参数中选择三组特征参数利用模版匹配算法进行分类识别,并且计算它们的识别率。本实验用于训练和测试的样本视频数都是10个,但是总的图像帧数不一样,训练视频总共300帧图像,而用于测试的总共有916帧图像。针对前面基于模版匹配的识别算法,本文从十个特征参数中选择三组六维空间特征向量,分别如下:

第一组特征向量: (Hu1, Hu2, Hu3, Hu4, Hu5, Hu6);

第二组特征向量: (Hu1, Hu2, Hu3, Hu4, 面积周长比, 缺陷个数);

第三组特征向量: (Hu1, Hu2, Hu3, Hu4, 面积周长比, 手指个数);

从以上的分组选择可以看出,第一组特征向量选择了6个Hu矩不变量,第二组是前4个Hu矩不变量的基础上加了面积周长比和缺陷个数这两个几何特征参数,而第三组特征向量是在第二组的基础上改变了几何特征参数,主要是缺陷个数和手指个数的区别。

针对以上三组特征参数,先对用于训练的样本利用函数 `createtemplate(float*, int,int)`计算得到模版,然后利用函数 `Recognizer(float *hu_r)`计算识别率。三组特征向量的识别结果如表4-3、表4-4和表4-5所示。表中测试次数就是总共识别了多少帧图像。计算识别率的过程是先读取每个视频的总的帧数,然后对每一帧图像进行识别,最后统计正确识别的次数和错误识别的次数,用正确识别的次数除以总的帧数就得到了识别率。表中最后一行是十种手势的总识别率。

表4-3 第一组特征参数的识别结果

手势	测试次数	正确识别	错误识别	识别率
0	76	58	18	76.32%
1	86	62	24	72.09%
2	78	56	22	71.79%
3	118	80	38	67.80%
4	89	67	22	75.28%
5	89	70	19	78.65%
6	100	74	26	74.00%
7	92	71	21	77.17%
8	87	59	29	67.82%
9	101	73	28	72.28%
总计	916	669	247	73.03%

表 4-4 第二组特征参数的识别结果

手势	测试次数	正确识别	错误识别	识别率
0	76	65	11	85.52%
1	86	74	12	86.05%
2	78	66	12	84.62%
3	118	93	15	78.81%
4	89	73	16	82.02%
5	89	73	16	82.02%
6	100	82	18	82.00%
7	92	78	14	84.78%
8	87	69	18	79.31%
9	101	84	17	83.17%
总计	916	757	159	82.64%

表 4-5 第三组特征参数的识别结果

手势	测试次数	正确识别	错误识别	识别率
0	76	71	5	93.42%
1	86	79	7	91.86%
2	78	72	6	92.31%
3	118	108	10	91.53%
4	89	83	6	93.26%
5	89	82	7	92.13%
6	100	90	10	90.00%
7	92	84	8	91.30%
8	87	80	7	91.95%
9	101	91	10	90.10%
总计	916	840	76	91.70%

分析以上三个表格计算的结果, 第一组只选择 6 个 Hu 矩不变量作为特征参数的总识别率是 73.03%, 而第二组特征参数在第一组的基础上引进了缺陷个数和面积周长比两个几何特征参数, 总的识别率是 82.64%, 最后一组参数在第二组特征参数的基础上将两个几何特征参数换成手指个数和面积周长比, 计算得到总的识别率为 91.70%。由此我们得到结论:

(1) 第一组特征参数和第二组特征参数识别结果比较, 发现完全利用 Hu 矩不变量作为特征参数的识别效果比在 Hu 矩不变量作为特征参数的基础上引进手势的几何特征的识别效果要差很多。

(2) 第二组特征参数和第三组特征参数识别结果比较, 尽管两者都是采用 4 个 Hu 矩不变量和两个几何特征参数, 但是手指个数作为特征参数的识别效果要比缺陷个数作为特征参数的识别效果好很多。

(3) 在图像识别技术中, Hu 矩不变量作为基本的特征参数是非常有必要的, 他可以弥补几何特征参数需要大量集合分析计算的不足, 但是具体的应用中, 有效的几何特征参数能将识别率提高很多。由此, 本文最终采用第三组特征向量 (Hu1, Hu2, Hu3, Hu4, 面积周长比, 手指个数) 作为最终的特征参数。

再次说明, 本文为了测试特征参数的有效性, 采用了基于模版匹配的识别算法, 如果利用基于 SVM 的识别算法, 整体的识别效果会增强, 但是对于各个特征参数的有效性区分就没那么明显。

4.4 本章小结

本章详细分析了整个识别系统的详细设计框架, 并且对其中特征参数的提取方法和识别算法的实现进行了详细分析。最后通过改识别系统对三组不同的特征参数进行了对比实验, 通过计算三组特征参数的识别率来分析比较 Hu 矩不变量与几何特征参数的有效性、几何特征中手指个数和缺陷个数的有效性, 最终选择了 (Hu1, Hu2, Hu3, Hu4, 面积周长比, 手指个数) 的组合作为特征参数。

第 5 章 总结与展望

5.1 本课题总结

手势具有多样性、多义性及不确定性，因此手势识别这是一个极富挑战性的多学科交叉性研究课题，是人机交互的研究热点及难点，研究手势识别对于改善人机交互的意义很大，提取有效的特征参数和设计一套有效的识别算法对于手势识别至关重要。在硕士论文的研究期间，主要完成了如下工作：

(1) 本文对基于三维模型的手势建模和基于表现的手势建模进行比较分析，针对本文研究的待识别目标样本设计了一种人手部 Tortoise 模型，将手指、手掌区分开来，然后针对具体要识别的样本图像的特征，分析得到几种可能有效的特征参数，具体包括手势中伸出的手指的个数、手部轮廓的缺陷个数和手势轮廓的面积与周长的比值。另外，图像的不变矩特征作为图像识别中的重要参数，我们也选取了基本的 7 个 Hu 矩不变量作为特征参数。

(2) 图像预处理的结果直接影响特征参数的计算结果。本文通过一系列的预处理过程，包括光照、色彩空间转换、中值滤波和阈值分割几个步骤，其中图像阈值分割会直接影响后面手势轮廓图像的提取，本文分析了几种阈值分割的效果，最终采用基于大津法的自适应阈值分割对图像进行处理，得到了比较好的二值化手势图像，手部区域很好的从图像中分割出来。

(3) 在计算各个特征参数的时，分析手势轮廓图像的特征，提出了先求出手势轮廓的外接多边形，分析外接多边形的顶点特征，求出其中的凸点和凹点并进行编号，在这个基础上分析求出手掌轮廓的近似外接多边形以及手掌的重心，在此基础上提出了几种计算手势中伸出的手指个数的方法。其中重点是画圆和求顶点数的方法。

(4) 对于特征参数的提取，本文总共计算了 10 个特征参数，并且对它们的计算结果进行详细分析，7 个 Hu 不变量对二维物体的描述具有旋转、平移和缩放不变性，由于较高阶中心矩对于成像过程中的误差、微小的变形等因素非常敏感，使得 $Hu1$ 和 $Hu2$ 用于物体识别有效性比后面 5 个要高，另外缺陷个数对于手势轮廓的提取精度要求比较高，而且缺陷的计算方法没有一个标准，需要大量的实验去验证，导致同一种手势求出的缺陷个数不稳定，而手指个数是

每种手势固定的特征信息，将它作为特征参数是非常有效的，但是其计算方法比较复杂。

(5) 为了验证各个特征参数的有效性，本文设计实现了一种简单的基于模版匹配的识别算法，从中 10 个特征参数中选取了三组特征参数进行对比实验，通过计算三组特征参数的识别率来分析比较 Hu 矩不变量与几何特征参数的有效性、几何特征中手指个数和缺陷个数的有效性。

5.2 展望

在实际用于人机交互的手势识别系统中，识别的实时性准确性要求都会高很多，针对基于计算视觉的手势识别的研究，本文对识别过程中需要提取的有效特征参数进行了研究，还有很多需要改进的地方。具体分为以下几点：

(1) 手势建模是特征参数提取的基础，好的数学模型可以帮助我们分析手势的几何特征，本文只结合待识别的 10 种手势的几何特征设计了人手部 Tortoise 模型，尽管通过分析得到了很多有效的特征参数，对于其他模型的研究还不够，因此，尝试研究新的手势模型，获取更多有效特征参数是进一步研究的重点。

(2) 图像的预处理过程有待完善，由于提取的特征参数都是基于手部轮廓图像的，所以提取手部轮廓很重要。本文只是在简单的背景下提取手部的轮廓图像，对于复杂背景图像，就需要更复杂有效的预处理了，特别是自适应的阈值分割。

(3) 图像的不变矩特征在图像识别的应用中有比较好的效果，本文只是采用了传统的几何不变矩作为特征参数，能否提取出更好的不变矩特征参数是一个很值得研究的问题。通过对普通不变矩的计算公式进行相应改变得到更有效的特征参数，比如将手的几何特征中手掌的半径作为一个系数乘到公式中会不会对手势识别更有实用性，或者在求不变矩的时候对于每个像素点的值改成原来的平方值会不会使计算得到的不变量直接的差异更加大。因此不变矩特征是进一步研究的重点。

(4) 本文采用的基于模版匹配的识别算法比较简单，用于比较特征参数的有效性效果不错，但是在实际的应用中，识别算法的要求是非常高的，更好的识别算法对于特征参数提取不足能够有效的弥补。因此在尽可能提取最有效的特征参数之后，研究更好的识别算法也是下一步研究的重点。

致谢

毕业论文即将完成，我的学习生涯也要告一段落了。借此机会，我要对一直以来帮助过我的人表示深深的感谢。

首先，感谢我的导师杨春金副教授，本课题是在杨老师的精心指导下立题完成的。在课题进展的过程中我遇到了许多困难，杨老师在各个方面都给予了我认真的指导和帮助，让我接触到许多新鲜的知识，学到了许多书本上学不到的知识。杨老师的谆谆教诲与悉心指导使我受益匪浅。谢谢他对我的帮助和支持！

同时，我要感谢所有教导过我、关心过我的老师，特别是信息学院的王虹教授、阙大顺教授、杨杰教授、王绪国副教授、孟哲副教授、沈维聪副教授、胡君萍副教授、封小钰副教授、胡德明老师以及信息学院的其他老师们，你们为我们的学业倾注了大量的心血，感谢你们对信息学院一代又一代的学生无私的付出。

还要感谢一直帮助我的老师兼师兄阮军副教授，他在我的研究生的学习和生活中，给了我许多指导性的建议，帮助我解决了很多困难，他学识的全面，严谨的作风，对工作认真负责的态度，钻研的精神都是我学习的榜样。我还要感谢与我一起生活了三年的同学，是他们陪伴我度过了最难忘的研究生生活，在学习和生活上都给了我巨大的帮助。

最后，我要感谢我的父母，感谢从小到大一直支持鼓励着我的所有的亲人，正是因为有他们的全力的支持，我才能走到今天。感谢所有支持关心我的朋友们！谢谢大家！

参考文献

- [1] 陈帮民. 手势追踪研究与手势识别应用平台实现[D]. 中山大学, 2011.
- [2] 邹洪. 实时动态手势识别关键技术研究[D]. 华南理工大学, 2011.
- [3] Gupta, L. Suwei Ma. Gesture-based interaction and communication: automated classification of hand gesture contours[C]. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2009.
- [4] Malgireddy, M.R. Corso, J.J. Setlur, S. Govindaraju, V. Mandalapu, D. A Framework for Hand Gesture Recognition and Spotting Using Sub-gesture Modeling[C]. Pattern Recognition (ICPR), 2010 20th International Conference on, 2010.
- [5] 刘信. 基于视频的中国手语识别算法的研究[D]. 中国科学技术大学, 2008.
- [6] Nguyen Dang Binh, Ejima, T., Intelligence Media Lab., Kyushu Inst. of Technol., Fukuoka. Real-Time hand Gesture Recognition Using Pseudo 3-D Hidden Markov Model[C]. Cognitive Informatics, 2006. ICCI 2006. 5th IEEE International Conference on, 2007.9.
- [7] Moni, M.A., Ali, A.B.M.S. HMM based hand gesture recognition: A review on techniques and approaches[C]. Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on, 2009.9.
- [8] JongShill Lee, YoungJoo Lee, EungHyuk Lee, SeungHong Hong. Hand region extraction and gesture recognition from video stream with complex background through entropy analysis[C]. Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE, 2005.3.
- [9] 吴莹. 基于单目视觉的实时手势识别系统[D]. 哈尔滨工业大学, 2009.
- [10] 邬大鹏. 基于视觉的手势识别及人机交互研究[D]. 南京航空航天大学, 2010.
- [11] 江立. 基于 CAS-GLOVE 数据手套的手势识别技术研究[D]. 北京交通大学, 2006.
- [12] 丁跃等. 基于手机手势识别的媒体控制界面[J]. 计算机工程, 2011, 36(23): 152-154.
- [13] Hyung-Ji Lee, Jae-Ho Chung. Hand gesture recognition using orientation histogram[C]. TENCON 99. Proceedings of the IEEE Region 10 Conference, 2009.8.
- [14] Pham The Bao, Nguyen Thanh Binh, Tu Duy Khoa. A New Approach to Hand Tracking and Gesture Recognition by a New Feature Type and HMM[C]. Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Sixth International Conference on, 2009.12.
- [15] 贾建军. 基于视觉的手势识别技术研究[D]. 哈尔滨工业大学, 2008.
- [16] 温正东. 基于视觉手势识别的人——机器人交互系统研究[D]. 华南理工大学, 2011.

- [17] 张毅. 静态手势识别的神经网络方法研究[D]. 电子科技大学, 2011.
- [18] 谈国新, 程文山. 基于肤色分割和 Camshift 的手势识别研究[D]. 华中师范大学, 2010.
- [19] 孙丽娟, 张立材, 郭彩龙. 基于视觉的手势识别技术[J]. 计算机技术与发展. 2008, 18(10): 214-221.
- [20] Pham The Bao, Nguyen Thanh Binh, Tu Duy Khoa. A New Approach to Hand Tracking and Gesture Recognition by a New Feature Type and HMM[C]. Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Sixth International Conference on, 2009.12.
- [21] Mu-Chun Su, Inst. of Comput. Sci. & Inf. Eng., Nat. Central Univ., Chung-Li. A fuzzy rule-based approach to spatio-temporal hand gesture recognition[C]. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2010.8.
- [22] 戴文华. 基于笔手势的笔式人机交互系统研究[D]. 南开大学, 2010.
- [23] 王修辉. 基于自适应遗传算法的手势识别[J]. 计算机辅助设计与图形学学报. 2007, 19(8): 1056-1061.
- [24] Ip, H.H.S., Chan, S.C.S., Lam, M.S.W. Hand gesture animation from static postures using an anatomy-based model[C]. Computer Graphics International, 2000. Proceedings, 2008.8.
- [25] Verma, R., Dev, A. Vision based hand gesture recognition using finite state machines and fuzzy logic[C]. Ultra Modern Telecommunications & Workshops, 2009. ICUMT '09. International Conference on, 2009.
- [26] Ge, S.S., Yang, Y., Lee, T.H. Hand Gesture Recognition and Tracking based on Distributed Locally Linear Embedding[C]. Robotics, Automation and Mechatronics, 2010 IEEE Conference on, 2010.
- [27] 杨青. 手势识别技术的研究[D]. 大连理工大学, 2010.
- [28] 李勇, 冷彪. 基于图像处理和神经网络的岩体识别[J]. 大众科技. 2009(7): 54-56.
- [29] 谭旭, 肖南峰. 基于改进 RCE 和 RBF 神经网络的静态手势识别[J]. 计算机工程与应用. 2011, 47(7): 172-176.
- [30] 宋瑞华. 基于傅立叶描述子的手势识别算法[D]. 西安电子科技大学, 2008.
- [31] 王向慧, 刘文光. 颜色模型浅析[J]. 科技信息. 2010(16): 592-593.
- [32] Moon-Jin Jeon, Seung-Eun Yang, Zeungnam Bien. User adaptive hand gesture recognition using multivariate fuzzy decision tree and fuzzy garbage model[C]. Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on, 2009.10.
- [33] Tusor, B., Varkonyi-Koczy, A.R. Circular fuzzy neural network based hand gesture and posture modeling[C]. Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE, 2010.
- [34] 胡卫谊. 基于动态手势的人机交互系统研究[D]. 武汉理工大学, 2010.

- [35] 吴浩. 基于视觉的应用于智能控制的手势识别技术的研究[D]. 电子科技大学, 2008.
- [36] 徐剑辉. 基于感兴趣区域的运动人体检测与跟踪[D]. 江苏大学, 2009.
- [37] 高娜等. 改进的最大类间方差法在 PCB 图像分割中的应用[J]. 数字技术与应用. 2009(12): 48-53.
- [38] 许杏. 基于隐马尔科夫模型的手势识别研究[D]. 华南理工大学, 2011.
- [39] Ishikawa, M, Matsumura, H. Recognition of a hand-gesture based on self-organization using a DataGlove[A]. Neural Information Processing, 2009. Proceedings. ICONIP '99. 6th International Conference on, 2009(2).
- [40] 魏伟波, 范筱亭. 不变矩方法研究[J]. 火力与指挥控制. 2007, 32(11): 114-117.
- [41] Ziaie, P., Muller, T., Knoll, A. A Novel Approach to Hand-Gesture Recognition in a Human-Robot Dialog System[C]. Image Processing Theory, Tools and Applications, 2008. IPTA 2008. First Workshops on, 2009.
- [42] 讲能兴. 基于计算机视觉技术的机器零件装配图数字标号的自动识别[J]. 计算机与现代化. 2011(5): 88-93.
- [43] 赵麟. 基于颜色不变性的图像检索算法研究[D]. 北京交通大学, 2009.
- [44] Byung-Woo Min, Ho-Sub Yoon, Jung Soh, Yun-Mo Yang, Toshiaki Ejima. Hand gesture recognition using hidden Markov models[C]. Systems, Man, and Cybernetics, 2007.