国内图书分类号:G40-057

国际图书分类号:378 密级：公开

**西 南 交 通 大 学**

**研 究 生 学 位 论 文**

基于Kinect的手势识别技术

在智能电视交互场景中的应用研究

年 级二〇一一级

姓 名朱明茗

申请学位级别 硕 士

专 业 教育技术学

指 导 老 师 景红 教授

二〇一四年四月十九日

Classified Index: G40-057

U.D.C: 378

Southwest Jiaotong University

Master Degree Thesis

THE RESEARCH ON APPLYING THE SIGN LANGUAGE RECOGNITION TECHNOLOGY FOR DEAF EDUCATION BASED ON KINECT

Grade: 2011

Candidate:Zhu Mingming

Academic Degree Applied for : Master Degree

Speciality: Educational Technology

Supervisor: Prof. Jing Hong

Apr.19, 2014

**西南交通大学**

**学位论文版权使用授权书**

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权西南交通大学可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复印手段保存和汇编本学位论文。

本学位论文属于

1．保密□，在 年解密后适用本授权书；

2．不保密□，使用本授权书。

（请在以上方框内打“√” ）

学位论文作者签名： 指导老师签名：

日期： 日期：

西南交通大学硕士学位论文主要工作（贡献）声明

本人在学位论文中所做的主要工作或贡献如下：

【贡献】

本人郑重声明：所呈交的学位论文，是在导师指导下独立进行研究工作所得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中作了明确说明。本人完全了解违反上述声明所引起的一切法律责任将由本人承担。

学位论文作者签名：

日期：

摘 要

【摘要】

关键词：【关键词】

Abstract

【English Abstract】

**Keywords: 【English Keyword】**

目 录

[摘 要 I](#_Toc417155856)

[Abstract II](#_Toc417155857)

[第1章 绪论 1](#_Toc417155858)

[1.1. 课题研究的意义 1](#_Toc417155859)

[1.2. Kinect体感技术简介 2](#_Toc417155860)

[1.2.1. Kinect原理 3](#_Toc417155861)

[1.2.2. Kinect关键技术 3](#_Toc417155862)

[1.2.3. Kinect局限性 4](#_Toc417155863)

[1.3. 国内外研究现状 5](#_Toc417155864)

[1.3.1 人机交互 5](#_Toc417155865)

[1.3.2 手势识别 6](#_Toc417155866)

[1.3.3 动态手势识别 7](#_Toc417155867)

[1.4. 研究内容和组织结构 7](#_Toc417155868)

[第2章 手势识别技术概述 9](#_Toc417155869)

[2.1 基于数据手套的手势识别 9](#_Toc417155870)

[2.2 基于视觉的手势识别 10](#_Toc417155871)

[2.2.1 静态手势识别 11](#_Toc417155872)

[2.2.2 动态手势识别 12](#_Toc417155873)

[2.3 OpenCV简介 13](#_Toc417155874)

[2.4 智能电视交互场景下的手势 14](#_Toc417155875)

[2.5 本章小结 15](#_Toc417155876)

[第3章 基于深度图像的静态手势识别 16](#_Toc417155877)

[3.1 基于阈值的手势分割 16](#_Toc417155878)

[3.1.1 Kinect深度传感器 16](#_Toc417155879)

[3.1.2 阈值的确定 17](#_Toc417155880)

[3.1.3 K-Means分割手势区域 18](#_Toc417155881)

[3.2 手势轮廓预处理 19](#_Toc417155882)

[3.2.1 手势轮廓提取 19](#_Toc417155883)

[3.2.2 手势轮廓近似 20](#_Toc417155884)

[3.2.3 手势轮廓的最小凸包 21](#_Toc417155885)

[3.3 FT-GB算法提取手指点 21](#_Toc417155886)

[3.3.1 FT-GB算法 21](#_Toc417155887)

[3.3.2 A\_Thres参数的设定 22](#_Toc417155888)

[3.4 静态手势的识别 23](#_Toc417155889)

[3.4.1 静态手势的定义 23](#_Toc417155890)

[3.4.2 静态手势的识别 24](#_Toc417155891)

[3.5 本章小结 25](#_Toc417155892)

[第4章 动态手势识别 26](#_Toc417155893)

[4.1 抓取手势 26](#_Toc417155894)

[4.2 光标移动手势 27](#_Toc417155895)

[4.3 轨迹书写手势 28](#_Toc417155896)

[4.4 空间手写轨迹识别 29](#_Toc417155897)

[4.4.1 轨迹特征 31](#_Toc417155898)

[4.4.2 DTW算法 31](#_Toc417155899)

[4.4.3 基于位置相似度权重的基距离 32](#_Toc417155900)

[4.5 本章小结 34](#_Toc417155901)

[第5章 系统硬件及软件环境 35](#_Toc417155902)

[5.1 实验软硬件环境 35](#_Toc417155903)

[5.3.1 硬件环境 35](#_Toc417155904)

[5.3.2 软件环境 35](#_Toc417155905)

[5.3.3 软件架构 35](#_Toc417155906)

[5.2 手指点检测 36](#_Toc417155907)

[5.3 静态手势识别 37](#_Toc417155908)

[5.4 动态手势识别 38](#_Toc417155909)

[5.5 空间手写轨迹的识别 38](#_Toc417155910)

[5.6 本章小结 39](#_Toc417155911)

[结论和展望 40](#_Toc417155912)

[结论 40](#_Toc417155913)

[改进与展望 41](#_Toc417155914)

[致谢 42](#_Toc417155915)

[参考文献 43](#_Toc417155916)

1. 绪论
2. 课题研究的意义

随着计算机软硬件技术飞速发展，“客厅娱乐”已逐步替代传统的家庭娱乐模式成为主流。智能电视【引用网页】在“客厅娱乐”的过程中扮演着主要角色。现今的智能电视，除了传统的电视功能外，还能提供玩游戏、检索观看网络视频、浏览网页等传统电视不具备的新功能。

这些新功能对智能电视和用户之间的人机交互技术提出了新的要求。比如选择【PPT多级列表中的节目】，【PPT浏览排版复杂的网页时】，用户通过传统电视的遥控器选择所需内容的操作过程非常繁琐。由于传统电视的遥控器通过精简按键数量，以牺牲输入效率为代价来避免用户（尤其是老年用户）对于复杂按键操作的困惑[1]，这对于在网页上输入文字信息、通过节目名称查找等常用功能来讲，无疑造成了极大的不便。



图1-1 传统电视遥控器（左） 迈乐F10-Pro全键盘控制器（右）

鉴于这种情况，诸如乐视【引用网页】采用了在控制器上植入触摸板，并且在电视内置操作系统中添加虚拟键盘的方式（如图1-2所示），模拟用户和PC的交互方式。这种方法虽然较传统控制器便利性上有提高，但是考虑到电视环境的交互特点，用户仍旧无法便利地使用[1]。



图1-2 乐视TV遥控器

与键盘、鼠标、触摸显示屏等传统人机交互方式所不同的是，日常生活中，脸部表情、身体姿势、手势等等则是传递信息的主要方式。特别是基于手势的人机交互技术，使得交互过程更加自然、便捷。比如通过识别静态手势来识别手语[ 5， 9～ 14]。利用手部动作信息来拓展静态手势的表达能力也已经有不少尝试，比如（1）使用手指控制计算机图形界面的鼠标指针[ 8，15，16]；（2）利用手指来对虚拟物体进行控制[ 17～ 20] （3）操作桌面应用程序，如WWW 浏览器导航[ 22， 8] 、屏幕操纵控制[ 23] 、游戏控制[ 24] 等；（4）手指绘画[ 8， 17， 24]等。上述研究成果为基于手势的人机交互技术能否满足智能电视和用户之间的交互需求验证了可行性。

虽然基于手势的人机交互技术已经发展了很多年，但是在2010年之前，由于手势数据采集设备的诸多限制，这种交互技术的研究一直停留在实验阶段。此外，尽管针对手势在人机交互领域的应用型研究已经有了大量工作，但是，针对智能电视这个特殊应用场景的应用性研究还是很缺乏。

2010年，Microsoft推出了基于红外线深度传感技术的Kinect系列体感设备以及配套的Kinect for Windows SDK开发套件。通过这套软硬件，可以实时获取人体各个部位的空间位置信息，从而识别人体的手势动作，实现人机交互。

Kinect系列产品为手势识别提供了有力的帮助，达到了不错的效果【参考文献】。同时，由于Kinect系列产品面向的就是“客厅娱乐”这个应用场景，而Kinect for Windows SDK开发套件对手势数据的采集提供了良好的支持【可以提供参考】，但并没有直接实现具体的手势识别功能，因此，本文的研究工作，是以与智能电视进行交互为应用场景，针对常用操作预先定义【一组手势】，然后以Kinect系列产品提供的手势数据为基础，对【预先定义的手势】进行识别，从而实现与智能电视的交互。本文的意义，就在于对基于手势的人机交互技术在智能电视领域的应用做出一些探索。

1. Kinect体感技术简介

新一代“体感技术”的亮点，在于用户使用支持体感技术的游戏、应用程序时，可以直接将身体各个部分（比如手、胳膊）作为“控制器”与虚拟场景中的物品进行互动，而无需使用或者穿戴任何其他的控制设备。自从2006年任天堂推出 Wii，体感游戏受到众多玩家的欢迎。各大厂商也紧随其后，推出了各式各样支持体感技术的智能设备及相关配套技术。比如在2010 年，Microsoft 发布了旗下第一代体感感应套件 Kinect for Xbox。用户使用这套设备时，无需使用任何外接控制器，便可实现体感控制的效果。在技术上，Kinect 同时使用普通RGB摄像头和深度信息摄像头来采集用户动作的视频流。Kinect可以捕捉人体在各种姿态下，身体各部分的空间位置数据，同时不受任何可见光光照变化的影响。因此，使用Kinect采集得到数据拥有良好的鲁棒性。

### Kinect原理

本文所使用的Kinect For Windows与Kinect For Xbox同属于Microsoft研制开发的Kinect系列中的第一代体感产品。Kinect系列产品均是由一个普通RGB摄像头、一对深度信息摄像头、一组麦克风阵列及一个转动马达组合构成，如图1-3所示。上述传感器使得Kinect具备时动态拙捉、影像辨识、语音识别和多人群体互动等多种功能。



图1-3 Kinect 结构图

每个时刻，Kinect可以同时采集3 种信息，分别是RGB彩色图像、深度图像和声音信号。Kinect机身上有 3 个摄像头，从左到右（面向摄像头）依次是红外发射器、普通RGB摄像头、和红外线CMOS深度信息摄像头。Kinect主要通过深度信息摄像头来记录和检测用户的行为。Kinect 还配备了追焦技术，底座马达会随着当前追踪物体的移动而自动调整对焦。Kinect 也有一套内置的阵列麦克风系统。多组麦克风同时收音，通过比对消除杂音，为后续的Kinect语音识别功能提供清晰的数据。

在Kinect系列中，Kinect For Windows与Kinect For Xbox的唯一区别在于Kinect For Windows具有近距离模式（Near Mode），可以获取0.4~0.8米范围内清晰稳定的深度图像。而默认模式下，两者都可以获得0.8~4米范围内的稳定深度数据。

### Kinect关键技术

【语言不通畅：Kinect核心技术：获取“视野范围”内各点景深，即到摄像机平面的空间距离，并通过此结合摄像头之间的距离，计算各个点的3维坐标】。视野范围内的所有点构成深度图像矩阵（又称深度图像帧），图像矩阵每个像素点的值即为空间内该点处的深度值。每个深度值由16bit数据表示，其中13位用来记录深度数据，也就是说，深度数据的分度值共有2048级，每一级的表示1毫米。

Kinect配套的Kinect for Windows SDK开发套件提供了包含人体20个骨豁点空间位置信息的人体骨豁模型，如图所示。根据Kinect for Windows SDK实时提供的人体骨骼模型，在某一时刻可以获取骨豁节点的位置，进而得到骨豁节点之间的夹角和相对位置。若在连续的一段时间内，则可以获得骨骼节点的运动向量。

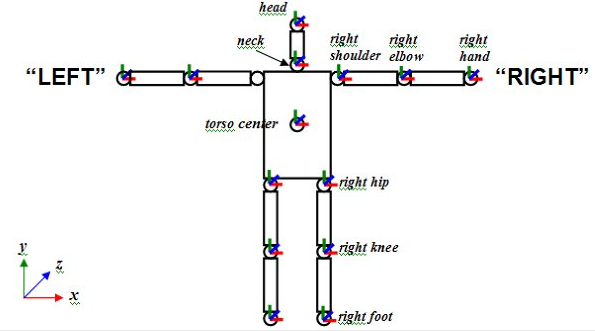


图1-4 Kinect骨骼模型

这些信息可以用作姿势识别和手势识别。由于姿势识别的对象主要是静态姿势，因此，可以利用Kinect for Windows SDK提供的角度、节点相对位置等信息作为识别依据。手势识别则主要分为静态手势识别与动态手势识别两类。支持向量机算法、人工神经网络算法、模板匹配算法等算法常用来识别静态手势。而由于常用的动态手势可以简化表示为一条由位置随着时间变化且长度不定的位置点序列构成的轨迹，因此，常用隐马尔可夫模型（HMM）和 DTW匹配算法等【文献：陈静[9]】来识别动态手势。

此外，Kinect for Windows SDK开发套件还提供了语音识别、人脸检测等关键技术。

【这些技术结合起来，可以实现“虚拟课堂”，“人机交互”等许多功能】

### Kinect局限性

Kinect在捕捉人体骨骼模型，识别手势、动作等方面也有一定的局限性，主要有以下几个方面：

（1）“体感技术”对运行系统的配置要求：Kinect For Windows SDK套件实现的“体感技术”相关算法在运行时产生巨大的数据计算量。如果追求算法的实时性，运行系统的CPU需要具备较高的数据处理性能。为了保证流畅性，Microsoft官方推荐配置为双核2.66GHz的CPU。

（2）骨骼模型的鲁棒性：如果摄像头的视野范围内出现与人体有相似结构的物体，那么这些物体会被误识别为人体骨骼，出现在骨骼模型的相应位置，干扰正常骨骼模型的构建。

（3）多人互动的人数限制：虽然通过Kinect For Windows SDK的算法可以检测到至多6个用户，但是追踪时至多支持2个用户。

（4）Kinect的精度：虽然Kinect系列产品可稳定获取的深度图像信息的范围在0.4~4米之间，但是随着距离的增加，Kinect的深度传感器的灵敏度逐渐降低。Kinect的深度值随距离变化的关系如图所示。

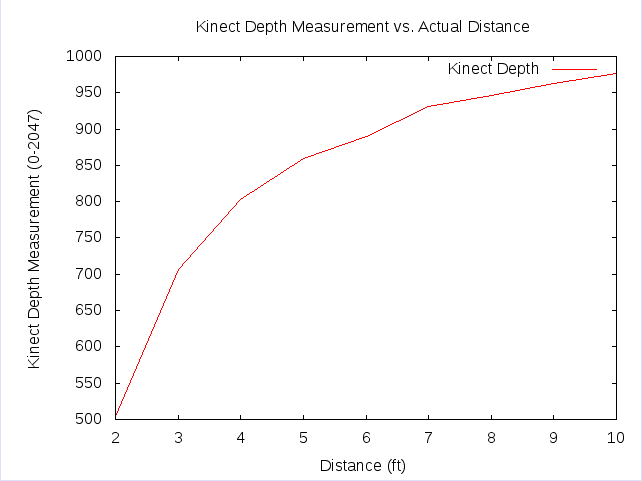


图1-5 深度值与距离的关系

（5）Kinect的视野范围有限：Kinect配备的 Moving Touch传动马达只能上下旋转，因此Kinect的摄像头只能拍摄底座正前方一定范围内的信息。

1. 国内外研究现状

### 人机交互

中文的“人机交互”在英语中有三个对应概念：一个是Human Machine Interaction (人一机器交互)'另一个是 Human Computer Interaction (人一计算机交互)，第三个是Human Robot Interaction (人一机器人交互)。本文的“人机交互”指的是第一个概念，即人与机器之间的互动。人机交互的目标是用户可以高效地对机器进行控制，机器同步地将交互结果返回给用户，给用户的下一步操作提供依据。用户通过一种叫做用户接口（User-Interface）的控制器和机器进行交互。用户接口设计的原则是更加简单、高效、并且用户友好。

人机交互发展至今日，用户接口的变革主要经历了以下几个阶段[9]：

（1）早期的手工作业。当时的计算机只能由设计者本人来使用,设计通过手工操作和二进制代码的方法来适应十分笨拙的大型计算机。

（2）命令行接口（Command Line Interface，CUI）。在这一阶段,程序员处理计算机的语言是作业语言或交互命令语言。这个阶段虽然要记忆许多命令,但已经可以稍微方便地调试程序和了解执行情况。

（3）图形用户界面（Graphical User Interface，GUI）。

（4）自然用户界面（Natural User Interface，NUI）。

【近年来，随着NUI概念的提出，越来越多的用户界面的研究和设计朝着利用用户的日常行为（如语言、手写、姿势、视线、表情等），通过不可见的、高效的用户接口与设备进行交互，提高人机交互的效率和自然性的方向发展。通过人脸识别、虹膜识别、人体姿势识别、手势识别、语音识别等技术，用户可以不操作任何的控制器，而仅仅通过脸部表情、眼神、身体姿势、手势的变化以及声控命令来和设备进行交互。【这方面的研究有：NUI + speech recognition，posture estimation，虹膜等】。在所有的这些日常行为中，手势作为更方便常用（相较于身体姿势、眼神、视线），表达力更加丰富（相较于虹膜、表情）的交互方式，同时随着语音识别技术的日臻完善，越来越多的研究开始着重于手势识别的研究。

### 手势识别

手势是一种自然而直观的交流方式，符合人类日常习惯。在日常生活中人们之间的交流通常会辅以手势来传达一些信息或表达某种特定的意图。

手势识别技术的早期研究主要集中于通过专用的硬件设备来采集手势信息。例如数据手套，可以获取用户在做手势时，手的位置、手指的伸展弯曲程度等信息。1993 年 B．Thamas 等人做的自由手遥控目标的系统是凭借数据手套作为输入的媒介，

在数据手套发展的同时，由于不满其需要使用者佩戴手套的诸多限制，越来越多的研究者逐渐转移到通过其他方式采集手势信息的方向上。比如标记手势的研究，研究者通过在手上作标记，例如在手腕和手指处贴上或画上特殊颜色的圆点，用来识别手势[11]。【也有加速度传感器】

但是，更多的研究者将注意力集中到仅仅通过自然手的视觉图像信息获取手势信息的研究上。1994 年，高文等人提出了一种静态复杂背景中手势目标的捕获与识别，1995 年又提出了动态复杂背景中手势目标的捕获与识别方法。1999 年，常红等人提出了基于计算机视觉技术的手形手位跟踪方法。2002年邹伟提出了一种基于双信息源的人手空间跟踪方法(利用视觉输入提取手的平面位置信息，借助于安装于肘部的弯曲传感器获取手臂的弯曲角度，根据所建模型由二者通过模糊计算确定手的深度信息)[13]。在 2011 年，ZhouRen 等提出基于Kinect 利用 FEMD（finger-earth mover's distance）算法实现稳定的静态手势识别，该算法需要大量的数据进行训练[23]。

基于视觉的手势识别是实现新一代人机交互所不可缺少的一项关键技术.然而， 由于手势本身具有的多样性、多义性以及时间和空间上的差异性等特点，加之人手是复杂变形体以及视觉本身容易受到背景冗杂等因素的干扰而并不鲁棒，因此基于视觉的手势识别仍旧是一个极富挑战性的研究课题。

### 动态手势识别

手势包括动态手势和静态手势。包含人体上肢（手臂、胳膊）协同手指、手掌运动的行为可以看做是一种动态手势。由于常用的动态手势可以简化表示为手势区域多个点的空间位置随着时间变化形成的多条不定长轨迹，所以动态手势识别问题又可以看作是轨迹识别问题。

Zhang等人通过手上佩戴的加速度传感器采样XYZ三个方向的加速度值作为手写的阿拉伯数字轨迹特征，然后使用隐马尔可夫模型对加速度特征序列进行训练和识别，最后在420个样本的数据集上取得了94.3%的识别率[2]。Amma等人通过采集手势动作过程中加速度值大小、角速度等信息作为特征序列，然后使用隐马尔可夫模型对特征序列进行解码，实现对手写的字母的识别，并通过语言模型将字母序列还原成单词和句子。在实验收集的8000个单词的语料库上，取得了89%的单词识别率[3]。

国内研究者中，[邹节华](http://www.cnki.net/kcms/detail/search.aspx?dbcode=CMFD&sfield=au&skey=%e9%82%b9%e8%8a%82%e5%8d%8e&code=28510588;)等人通过对单个摄像头采集得到的数据采用运动检测、肤色提取手势区域，然后采用Mean Shift跟踪算法获取手势轨迹，并采用隐马尔可夫模型对手势进行识别[4]。 [张毅](http://www.cnki.net/kcms/detail/search.aspx?dbcode=CJFQ&sfield=au&skey=%e5%bc%a0%e6%af%85&code=10570313;27926460;11420285;27926461;)等人使用OpenNI的手部分析模块，获得手心位置的轨迹，并利用隐马尔可夫模型训练有效的轨迹样本并实现轨迹的识别，从而通过不同的手势控制智能轮椅的运动[5]。王松林等人通过Kinect采集和分割手势图像，通过K曲率算法识别静态手势，并将静态手势结合DTW算法识别预先定义的6个简单的动态手势（前进后退等）控制机器人的行动[6]。

1. 研究内容和组织结构

本文首先对手势识别技术进行了全面回顾【，并对智能电视的普及对人机交互技术提出的新要求给予了阐述，并以此为应用场景，尝试通过手势识别技术满足这些新需求】。本文沿用之前的研究方法，将手势分为静态手势和动态手势2类，分别对这2部分手势的识别方法进行了实验和研究，并对动态手势识别时选用的手语特征提取方法进行了改进，【通过Kinect对手势进行采集，】实现了基于手势的人机交互功能。

本文的内容组织结构如下：

第一章绪论。对本文的背景及意义进行了详细的说明，介绍了基于手势的人际交互技术随着智能电视的普及所产生的意义和研究相关技术的必要性，以及研究的发展方向。同时对国内外相关工作的现状进行了阐述。

第二章【手势识别技术概述】。

第三章【基于深度图像的静态手势识别】。。

第四章【基于DTW算法的动态手势识别】。。

第五章【系统实现】。。对前述的技术进行了实验验证，介绍了手势识别系统的【架构】，并对实验结果进行了总结和分析。

第六章【】。。

1. 手势识别技术概述

手势是指手指、手掌或者手掌连同手臂产生的各种动作或者姿势，作为一种表达思想或感情的肢体运动，它能够与计算机更为直接、自然地交互。根据手势的时变性可以将手势分为静态手势和动态手势。静态手势是指静止不动的手做出特殊形状或姿势，即通过手的形状或姿势来表明操作者的意图。动态手势是指运动的手势，它是由一段时间范围内的一系列静态手势所构成，其特征就是手势是随时间变化的。

针对手势识别的研究方向有两种：一种是以数据手套为基础的手势识别技术，另一种是以计算机视觉为基础的自然手势识别技术。考虑到人手形变的高维度性和视觉问题本身的不确定性，基于视觉的手势识别一直是一项很有挑战性的研究课题【46】。

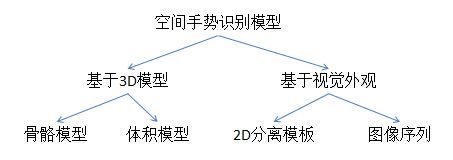


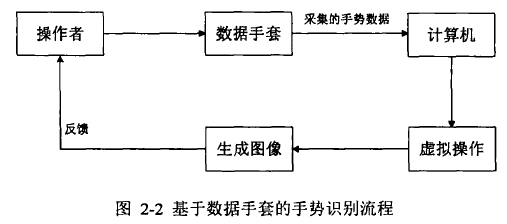
图2-1 手势识别技术分类

下面将着重介绍手语识别中常用的特征提取和识别算法，以及本文中涉及到的相关的概念和技术。

1. 基于数据手套的手势识别

基于数据手套的手势识别方法主要是通过数据手套内的传感器系统采集数据以判断手指的弯曲程度、关节的角度等,将之转化为传感信号输入到计算机中。数据手套如图2-1所示,它的类型有很多,不同类型的数据手套的性能和功能差别也很大。简单的数据手套只用到几个传感器来测量手指的曲率或者手的移动等,复杂的数据手套则用多个传感器来测量手的各种姿势,有的还带有触觉反馈等功能【王松林47】。

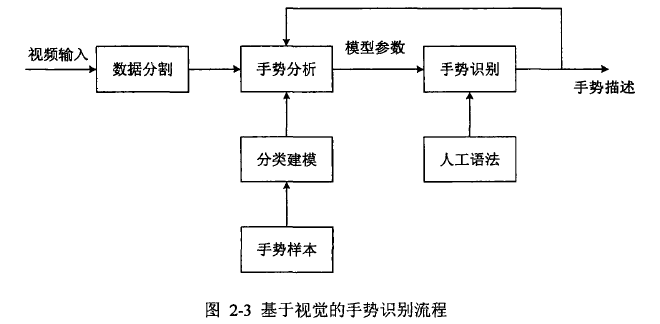
基于数据手套的手势识别过程一般如图2-2所示,当操作者按照自己的意图,做出一定的动作时,数据手套就会实时地收集代表手势的数据,例如手的位置、形状、手指各关节的弯曲程度等。然后将这些信息转化为传感信号发送给计算机, 计算机分析并处理这些传感信号,确定手势并据此跟随着操作者的手做相应动作,如抓取、释放、移动某物体,并且生成实时的图像,能够让操作者观察到,以便操作者修正自己的手势或动作。



1. 基于视觉的手势识别

以计算机视觉为基础的自然手势识别研究,就是通过一个或者多个摄像头釆“ 集图像或视频,然后利用视频图像处理方法分析识别手势信息。其特点是对人的自然手不附带任何条件,是一种非接触式手势识别,这样使得人机交互更加趋于自然化。手势识别中用到的摄像头可以是普通摄像头,也可以是深度摄像头,如微软公司发布的Kinect体感设备。

基于视觉的手势识别流程如图2-3所示。首先,通过一个或多个摄像机获取视频数据流信息。如果存在手势,则需要把该手势从视频信号中分割出来,然后选择己经训练好的手势样本模型进行手势分析,在分析过程中提取手势的特征值,针对这些特征值信息采用一些人工语法进行识别。最后生成手势描述,系统将这些描述转化具体操作由计算机去完成。



### 静态手势识别

静态手势识别指的是根据图像中手的形状和姿势判断手势所属的类别。静态手势识别的过程主要包括手势分割、手势特征提取和手势识别3个步骤。常用的识别算法有神经网络算法、模板匹配算法和检测指尖数量算法等。识别流程图如图所示：

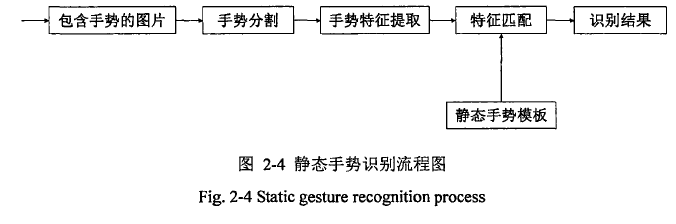


图2-4

1. 模板匹配法

模板匹配法在数字图像处理中使用的很广泛，是模式识别领域比较简单同时比较重要的算法。模板匹配法就是根据一幅图像中的对象，计算它与另一幅或多幅图像中对象的相似度，从而找到相匹配的目标图像 [23]。

模板匹配法的优点就是模板建立起来简单方便，适用于识别的目标较少的情况。计算简单、速度快的特点,在静态手势识别方面应用较多,例如通过计算预处理后的输入图像与标准模板库中图像的Hausdorff距离来判断手势识别的结果[49]但是当识别的目标较多时，识别率较低。

1. 人工神经网络

人工神经网络是一种模仿人类的大脑神经网络的结构和信息处理方式的模式识别算法。它将大脑的神经网络用数学模型表示出来，通过一个个简单的神经元之间相互连接、彼此传输信息、相互作用产生复杂的神经网络。这些神经元在不同的情况会处于不同的状态，并且是非线性变化。它们的这种结构使信息离散的存储起来。当神经网络局部出现问题时，其他部分并不受到影响，提高了算法的鲁棒性。同时，人工神经网络算法是按层分布的，每层独立的分部若干个神经元，它们之间可同步工作，使信息能够并行协同处理。

在神经网络中，算法的学习能力并不是由局部的神经元决定的，它取决于各个神经元间的相互作用。当神经网络中有大量的神经元时，其性能和功能也随之变得强大起来，能够处理多种复杂的问题。

不仅如此，神经网络的神经元之间连接强度会通过使用的权值决定，随着对算法的不断训练和优化，相应的权值同样会不断的调整，从而提高算法的灵活性。

1. 检测指尖数量【增加K曲率算法描述，在手势轮廓近似时，需要进行比较】

手指指尖的定位是更为通用和可靠的识别方法,通常一些静态手势可以通过检测指尖的个数、方向和位置等识别出来。常用的方法有基于区域分析的指尖检测和基于轮廓的指尖检测。

基于区域的方法利用的是手势形状的全部区域,其区域特征参数是针对区域内所有像素点计算的。简单的区域参数有区域面积、圆形度、矩形度、凹度等,比较复杂的参数有7个不变矩和Zemike矩等[59]。

基于轮廓的方法主要利用手势形状的外部边缘,通过对边界特征点的描述来获取目标区域的形状参数,其流程如图2-6所示。这些特征点能够代表曲线特征,反映目标轮廓曲线的形状。所以特征点是判断图像的关键,而提取特征点的关键在于边缘检测。常用来描述边缘特征的有曲率、几何常量和傅里叶算子等。

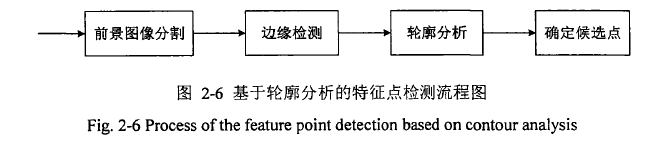
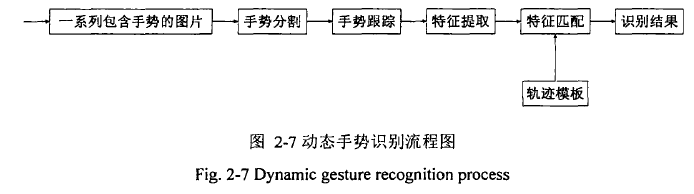


图2-5

### 动态手势识别

动态手势是指手的形状或位置能随时间发生变化的手势,它能够更加丰富和准确地表达操作者的意图。动态手势的识别与静态手势识别的不同之处在于它是指对一系列连续的手势动作进行识别,包括手的形变、手的运动轨迹识别,一般的流程如图2-7所示。

图2-6

动态手势识别常用的算法主要分为基于统计、基于模板、基于语法三种方法。

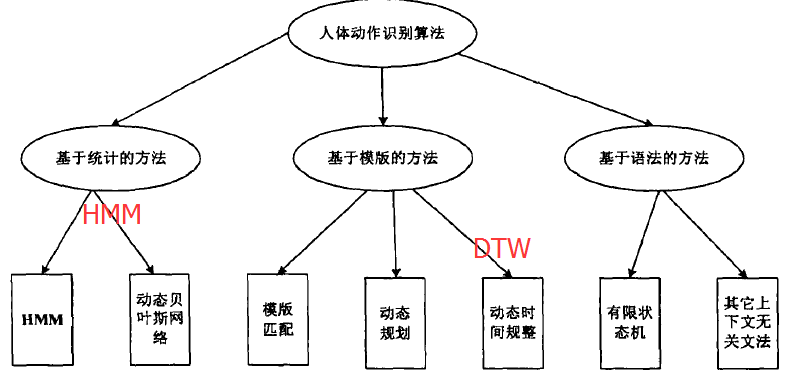


图2-7 动态手势识别常用算法

（1）基于统计的方法

目前手势识别主要釆用基于统计的方法,常用的方法有DBN和HMM。DBN具有灵活性和较强的扩展性,能够有效地融合推理多信息、识别多物体动态系统,主要用在复杂的、多目标动态系统的分析和识别上。由于DBN复杂的模板结构,以及过多的配置参数,所以在实际中很少直接应用于手势识别。HMM已经在手势识别方面有了比较广泛的应用,例如J.Schlenzig等人在处理基于视觉的手势识别中,为每个手势都定义了一个HMM[56]。该研究表明了HMM在连续手势识别中应用的能力。但是隐马尔可夫模型的训练计算过程复杂,不适用与大规模灵活多变的手势识别场景。同时,对于每一个新增的手势都需要设计对应的模型,并从大量训练样本中得出模型参数,不够灵活。

（2）基于模板的方法

基于模板的方法主要包含动态时空规整、动态规划算法和模板匹配方法。模板匹配算法是其中最为简单的实现,即将已经建好的模板与每个动作的特征数据进行比较,根据两者的相似度来判别手势。计算代价低是模板匹配算法的优点,其缺点是容易受到时间间隔变化的影响[16]。

DTW是在动态规划基础上发展起来的一种很好的非线性时间规整方法。它具有非线性归一化效果的模式匹配算法,釆用对特征信号进行伸长或缩短直至与标准模式的长度一致,以使得与模板更好的匹配。其优点是步骤简单,计算量少,对于单一动作序列的识别比较出色。缺点是只能做些简单固定的识别,同时无法真正反映动态系统[17]。

动态规划算法研究的是一个多阶段最优化问题。在算法实施的过程中,时间对样本模板和待测模板不产生影响,当待测模板中每个时刻的特征能够和样本模板中任意时刻的特征匹配时,可搜索出两个模板最优的匹配路径。动态规划算法的主要缺点是需要将样本模板和训练样本进行一一匹配,容易受到噪声影响,并且计算量会随着训练样本的增多而增加[18]

1. OpenCV简介

在进行实现手势识别相关算法时，通常使用OpenCV提供的一些基础算法。由于本文的实验环境基于WPF编程框架，因此，本文采用了对OpenCV使用C#进行封装的【EmguCV】。

OpenCV全称为Open Source Computer Vision Library，由Intel微处理器研究实验室于1999年建立，是一个开源、跨平台的计算机视觉库。它包含300多个用于图像和视频分析的算法开源库，最初的目标在于建立一个简单易用的计算机视觉框架，帮助广大研发工作群体用于计算机视觉领域的开发。它采用优化的C代码编写，并且充分利用了多核处理器的优点，加快了设计目标的执行速度，提高了系统实时处理能力。[29]

OpenCV应用在许多计算机视觉的相关领域。包括计算机视觉监控、信息安全、网络图像和视频、游戏界面、无人飞行器等。由于计算机视觉很多涉及到模式识别算法、机器学习等内容，因此OpenCV提供了机器学习库MLL(Machine Learning Library)来方便开发者进行模式分类等操作。该机器学习库主要涉及到模式识别和聚类两方面，其开发领域十分广泛和灵活，不仅包括计算机视觉，还可以应用于其他使用机器学习的场合。

通过OpenCV开发的程序不需要更改代码，可以在多个平台间使用，如Windows、Unix、Linux、MacOSX等。同时，OpenCV提供了不同语言的接口，如Python、MATLAB、Ruby等，能够与不同的软件兼容，使其涉及的图像处理和模式识别相关的算法更加全面。不仅如此，通过OpenCV开发的程序在调试运行时被编译为二进制可执行文件，使程序的运行时间更短，提高了程序的实时性。

OpenCV的主体一共分为5部分，包括HighGUI、CxCore、Cv、ML、CxAux。其中，HighGUI包含用户交互及图像和视频的输入输出函数；CxCore是OpenCV最重要的一个库，它提供了最基本的数据结构和函数，包括数组、矩阵等的基本运算，出错处理函数等。核心函数库Cv包含了图像处理方法和一些计算机视觉算法。ML是机器学习库，包含大多数的函数和工具用于统计分类和聚类。CxAux是用来存储即将淘汰的和最新更新的函数和算法。OpenCV其中四个的基本结构如图2-7所示[29]。



图2-8EmguCV结构图

1. 智能电视交互场景下的手势
2. 本文中，静态手势和动态手势之间的区别和联系。
3. 根据智能电视的场景，确定本文需要涉及的静态和动态手势。



图2-8 应用选择界面



图2-9 文字输入



图2-10 电视参数设置界面

1. 本章小结

本章首先根据手势识别研究方向的分类,分别介绍两种不同的手势识别,对静态手势与动态手势识别的一般流程进行了综述。最后调研了常用的静态手势与动态手势识别算法,并进行了概括与总结。

1. 基于深度图像的静态手势识别

本章描述了一套能够从深度图像中提取手势区域，并在此基础上通过识别手指点个数，完成静态手势识别的算法框架。基于Kinect的静态手势识别的算法流程如图3-1所示。

为了识别静态手势，我们首先要从深度图像中，提取出手势区域。本文考虑到算法通用性，没有使用Windows Kinect SDK自带的骨骼模型来定位手的位置，而是采用深度值阈值限定在Kinect摄像平面前的指定区域提取手势区域。由于得到的手势区域包含的像素点数量太多，不可能逐一判断所有的像素点是否是手指点，因此，本文采用手势轮廓对手势区域进行近似描述，并且在手势轮廓上寻找手指点。根据手指点的数量，划分静态手势。



图3-1 静态手势识别算法流程

1. 基于阈值的手势分割

通过Kinect深度摄像头得到的深度图像，除了手势，还有身体的其他部分、周围的物品等无关的背景冗杂。为了准确地识别手势，我们需要利用Kinect深度传感器的特性，去除这些冗杂区域，从而获取比较精确的手势区域。如图所示。

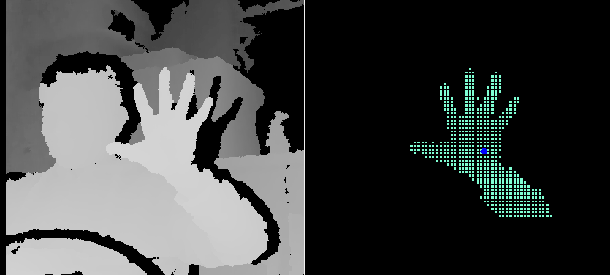


图3-2 深度图像（左）和分割后的手势区域（右）

1. Kinect深度传感器

通过实验发现，当用户与Kinect距离过远时，单位面积内，Kinect深度传感器投射到手指上的定位点太少，导致深度图像中，手势区域变得非常模糊。这种现象会严重影响后续的指尖识别算法的准确率。如图3-3所示。

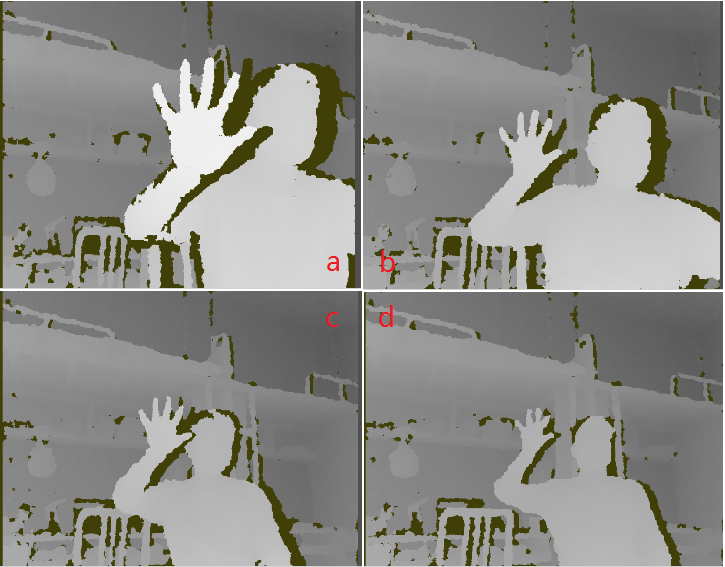


图3-3手势区域的清晰度与距离的关系

（a~d：0.4m、0.8m、1.0m、1.2m）

当用户与Kinect距离太近时，手势区域所占据整个深度图像比例过大，无法为后续的动态手势识别（尤其是书写轨迹识别）提供足够的动作空间。

因此，需要通过实验确定恰当的距离范围，使得手势足够清晰的同时，保留一定的动作空间。

1. 阈值的确定

本节通过实验，确定手指点识别率随距离的变化趋势，借助此确定手势区域的深度值阈值的最优取值范围。

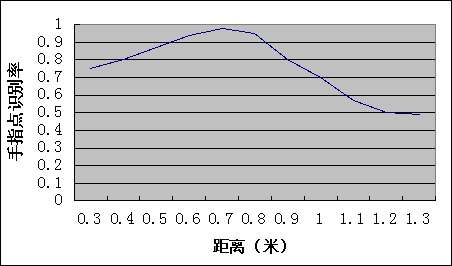


图3-4 手指点识别率与距离的关系

手指点识别率与距离的关系如图所示。通过实验发现：

1. 当距离在0.3~0.7米时，识别率逐步提高。在0.7米之后，识别率逐步降低。
2. 当距离在0.6~0.8的区间内时，平均识别率在94%以上。
3. 当距离超过0.8米时，识别率急剧下降。在0.9米的识别率为80%。

考虑到用户在作动态手势时，手与Kinect的距离会因为手的动作，发生一定的变化，因此，为了既满足较高的手指点识别率，又保留足够的距离变动空间，确定将0.6~0.8米为作为深度值阈值范围。本文将这个空间称之为“手势动作空间”，如图3-5所示。

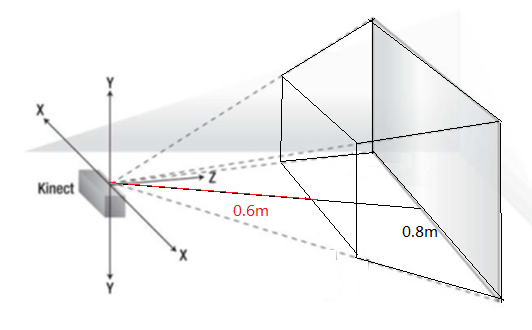


图3-5 手势动作空间（黑色实线框）示意图

1. K-Means分割手势区域

通过3.1.2中实验得到的阈值范围，我们将深度图像中，深度值在阈值范围内的像素点提取出来，作为手势区域的候选像素点。这些候选像素点中，仍然包含一些虽然在距离阈值范围内，但不属于手势区域的像素点，比如一些背景中冗杂的像素点、噪声点等等。

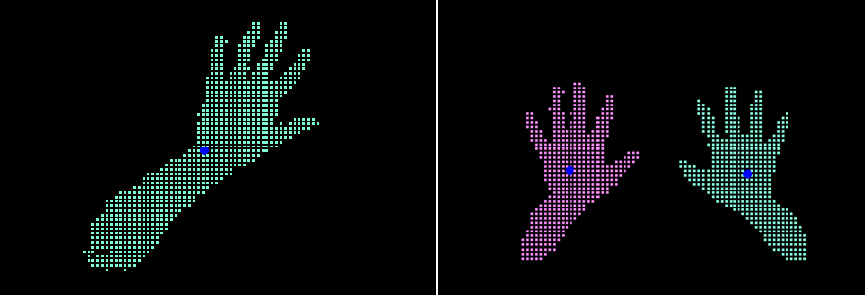


图3-6 K-Means分割手势（左：Cluster Count=1，右Cluster Count=2）

我们通过K-Means算法，对属于手势区域的像素点进行聚类。K-Means算法有如下优点：

1. 通过设置类个数参数，可以支持深度图像中出现1个或多个手势。
2. 聚类过程可以去除背景中冗杂的孤立像素点，只保留区域内连通、深度值相近的像素点。

K-Means算法实现的过程中，需要实验确定的参数如下：

1. Cluster Count：需要聚合的类的总数量。当只深度图像中只有一个手势时，设置成1即可。当出现多个手势时，只需要设置成需要支持的手势数量，就能按照设置生成同等数量的类，如图3-6所示。
2. Minimum/Maximum Points For Valid Cluster：每个类所拥有的最小/最大像素点的数量。因为每个用户的手势区域的大小（以像素个数衡量）是有上下界的。设定合理的最大、最小值，我们可以排除掉背景中一些明显不属于手势区域的冗杂区域。
3. Maximum Cluster Center Distances：类中所有像素点到类中心的最大距离（三维）。由于用户手的大小是定值，因此手上各点到手势区域中心的距离具有确定的最大值，通过实验设定合理的最大距离值，我们可以除掉一些不属于手势区域的像素点。

通过实验，我们确定各个参数的取值如表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| Cluster Count | Points For Valid Cluster | | Maximum Cluster Center Distances |
| Minimum | Maximum |
| 1 | 100 | 1000 | 120 |

表3-1 K-Means参数设置

1. 手势轮廓预处理

在手势分割完成之后，为了迎合后续的手指识别算法的要求，需要对手势轮廓进行预处理。手势轮廓是手势区域与背景区域的边界点组合起来形成的一条封闭曲线。通过预处理，可以得到手指识别算法要求的凸包；同时，降低手指识别算法的计算量，提高算法的实时性。

1. 手势轮廓提取

轮廓提取的主要原理是设定严格的检测标准，在手势区域内查找符合边界点条件的像素点，然后根据这些边缘点的特征寻找相似的点，最后依次连接起来形成轮廓图像。

本文使用边界跟随算法【见StickyNotes】提取手势区域的边界。EmguCV的FindContours函数实现了【这个算法】，因此使用该函数来查找手势轮廓。

通过实验发现，手势轮廓提取的最优结果是，只提取最外层的轮廓和构成这个轮廓的水平、垂直、对角线直线段的端点。这样，构成轮廓的点的数量被减少，更方便后续的处理。提取得到的手势轮廓如图所示。

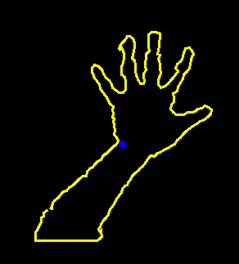


图3-7 手势区域的轮廓

1. 手势轮廓近似

实验统计表明，在手势分割之后，不同手势轮廓边界点的平均数量在【伪造60~100】。后续的手指点识别算法需要对边界点逐个判定，计算量大。在保证手指识别准确率的前提下，为了进一步减少计算量，使用近似多边形描述手势轮廓，本文使用EmguCV库的ApproxPoly函数对手势轮廓进行处理，如图所示。

近似多边形上的点的数量只有手势轮廓边界点数量的20%左右。通过观察可知，处理得到的近似多边形的角点【A’B’C’D’E’】与图3-7中的手指点ABCDE在位置上的偏差很少，因此，可以近似地认为是手指点。另外，FGH虽然也是多边形的角点，但从位置上看，显然不是手指点。后续的手指点算法可以区分这样的角点，具体会在3.3.2节手指点识别算法中讨论。综上所述，对手势轮廓采用近似多边形进行描述，对手指点的识别不会产生影响，同时，减少了计算量，提高了实时性。

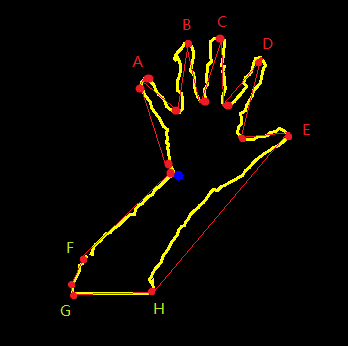


图3-8 手势轮廓的近似多边形

这种方法的另一个好处是，在对边界点逐个判定时，可以减少少数异常的边界点所带来的干扰。

1. 手势轮廓的最小凸包

因为在第4章动态手势识别的过程中，需要获取手势轮廓的最小凸包，因此，在得到了手势轮廓的近似多边形之后，本文借助Graham Scan算法【算法引用】，计算近似多边形的最小凸包。EmguCV的ConvexHul函数已经实现了该算法，因此我们使用这个函数获取手势轮廓似多边形的最小凸包，如图所示。

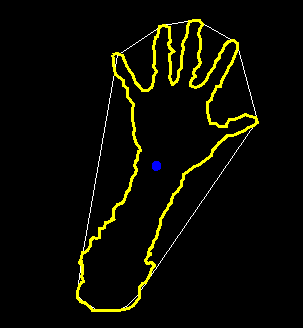


图3-9 手势轮廓的最小凸包

1. FT-GB算法提取手指点

在对手指轮廓进行预处理之后，可以得到了手势区域的近似多边形。通过观察可知，所有近似手指点都是近似多边形边界上的角点。同时，边界上也有一些非手指点的角点。他们之间的区别在于FT-GB算法正是利用了这一个特点，通过手型的几何特征，判断最小凸包上的各个点是否属于手指点。

1. FT-GB算法

假设是构成最小凸包的M个点，是近似多变形上的N个角点。显然，。

FT-GB算法的具体描述如下：

1. 对于所有的，我们在C中找到相邻的两个点；
2. 计算和之间的夹角α（下面称指尖点α参数），如图所示。

（3）如果 α≤ A\_Thres，那么，判定为指尖。

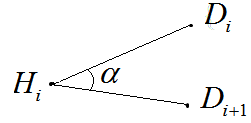


图3-10 指尖点α参数示意图

按照如上所述的算法步骤，可以得到所有的手指点。这种方法的优点在于：

1. 只要手指所在平面平行于摄像平面，即使发生旋转，α也能够保持不变。
2. 相比于其他的手指检测方法【文献】，这种方法只需要对A\_Thres参数进行假设或者实验设定。
3. 经过实验观测，对于同一个用户的不同手指，A\_Thres参数可以认为保持不变。

因此，这种算法是易用的、具有一定的鲁棒性。

对于3.2.2提出的如何区分非手指点的角点这个问题，可以看到图3-8中，和的角度是比手指点大很多的，因此，只要设定合理的A\_Thres值，使得的手指点的α小于A\_Thres，而非手指点的α大于A\_Thres，便可以保证只有真正的手指点得到保留。

1. A\_Thres参数的设定

FT-GB算法中有一个很重要的变量α。直观地来讲，α是手指根两端，与指尖中心的连线的夹角。两个可能因素会影响α：

1. 对于不同用户，手指的粗细和长度会发生变化。
2. 对于同一个用户，手指发生旋转。

对于第1个因素，本文采集和测量了5位用户【用户改为测试对象】的各根手指的α值，发现5位用户的α值并没有明显的差异。因此，本文假设第1个因素不影响α的选取。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 用户1 | 用户2 | 用户3 | 用户3 | 用户3 |
| 大拇指 | 20° | 21° | 20° | 20° | 19° |
| 食指 | 18° | 18° | 18° | 18° | 19° |
| 中指 | 18° | 18° | 18° | 18° | 18° |
| 无名指 | 18° | 18° | 18° | 18° | 18° |
| 小拇指 | 18° | 19° | 18° | 18° | 18° |

表3-2 用户手指点α参数调查

由于第2个因素涉及的情况复杂，本文没有对这个因素进行讨论，而是假设手指所在的直线始终与摄像平面平行。这样，由于手指可以近似看作是圆柱体，无论手掌如何旋转，α都基本保持不变。

一方面，如若阈值参数A\_Thres设置得太大,可能会将手腕处那一段（可以构成一个钝角）误认为为指尖；另一方面，当手指较短时或者由于离Kinect较远，造成截取图片中手指较短时，测出的角度会很大，如若A\_thres度太小，就无法识别出指尖。

通过实验，我们最终确定A\_Thres的取值为30。

1. 静态手势的识别

通过FT-GB算法，可以得到每一帧深度图像中的手指点个数。本节首先定义静态手势，接着介绍了如何通过FT-GB算法识别静态手势。

1. 静态手势的定义

本研究工作预定义了如下几个静态手势：

* 1. 停止手势：如图所示。表示一个静态/动态手势已经做完。保持手势期间，系统不识别任何的手势。

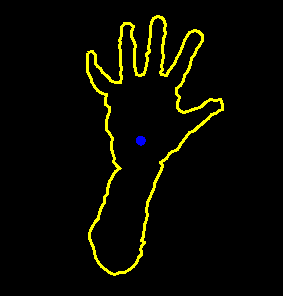


图3-11停止手势

* 1. 光标移动手势 / 轨迹书写手势（静态）：如图所示。
     1. 当处于非书写状态时，表示控制GUI的光标，通过手的移动，控制光标的移动。保持手势期间，始终对GUI的光标保持控制。
     2. 当处于书写状态时，表示控制了虚拟书写笔触，通过手的移动，控制笔触的移动，生成书写轨迹。保持手势期间，始终对虚拟书写笔触保持控制。

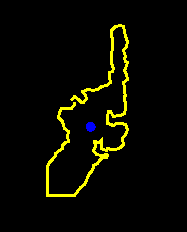


图3-12光标移动/轨迹书写手势

* 1. 抓取-开始手势：如图所示。表示进入抓取等待状态，准备抓取拖动GUI滚动条滑块。保持手势期间，维持抓取等待状态。

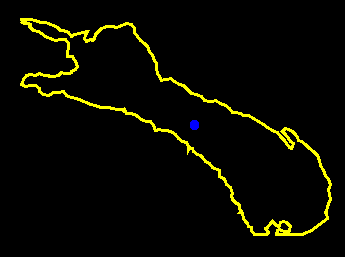


图3-13抓取-开始手势

* 1. 抓取-拖动手势：如图所示。表示已经抓取GUI滚动条滑块，正在沿着滚动条拖动滑块。

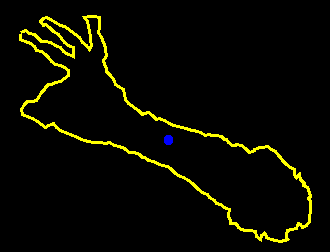


图3-14 抓取-拖动手势

1. 静态手势的识别

通过之前的算法步骤，各个静态手势的手指点示意图如图3-15所示。由于本研究工作中的定义的手势数量只有4个，因此，只通过手指点的数量识别不同的手势。除了手指点的数量，还可以结合手势轮廓，对仅仅通过手指点数量无法区分的手势进行识别。

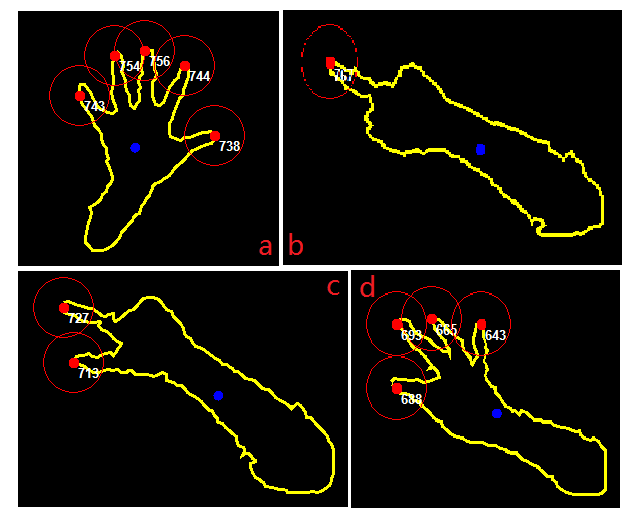


图3-15 不同静态手势的手指点

（a~d：静态手势（1）~（4））

1. 本章小结

本章首先通过实验，确定能够为手指点检测算法提供有效手势区域的深度值范围；接着，利用K-Means算法对深度值范围内的深度图像区域进行手势分割；其次，使用EmguCV的轮廓提取和近似多边形算法，对手势轮廓使用近似多边形进行描述，并使用Graham Scan算法，求出近似多变形的最小凸包。最后，借助FT-GB算法，逐个判断近似多边形的边界点是否为指尖点，然后根据指尖的数量识别定义的静态手势。

1. 动态手势识别

动态手势由静态手势组合用户动作序列组合而成。比如点击手势，是由点击静态手势加上“点击”动作构成。本研究工作主要实现了抓取、光标移动、英文字母书写这3个基础功能的动态手势。本章首先介绍了动态手势的功能、定义（动作序列）以及相应识别算法。本章余下的部分着重介绍了英文字母凌空书写手势的识别算法。为了提高书写轨迹的识别率，本文提出了一种针对DTW算法进行改进的，基于位置相似性的轨迹特征。

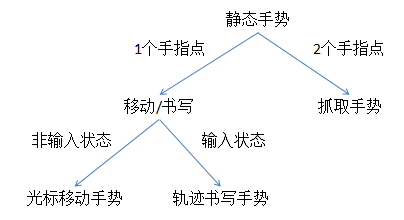


图4-1 动态手势判定决策树

1. 抓取手势

抓取手势的功能是抓取用户界面的滚动条、滑动块等GUI控件。抓取手势的动作序列如图所示。

手势序列由开始、拖动、结束3个动作构成：

* 1. 开始动作：手势保持“抓取-开始”静态手势。
  2. 拖动动作：首先，2指合拢呈“抓取-拖动”静态手势；接着，水平左右平移。
  3. 结束动作：展开5指，手势保持“停止”静态手势。



图4-2 抓取手势动作序列

抓取手势的识别算法的流程图如图所示。识别算法具体描述如下：

1. 输入静态手势，判断手势是否为“抓取-开始”手势。
2. 是，则进入“抓取-开始”状态。之后，等待下一帧的输入。
3. 否，则跳转至（2）。
4. 判断手势是否为“抓取-拖动”手势。
5. 是，则跳转至（3）。
6. 否，则等待下一帧的输入。
7. 判断当前是否处于“抓取-开始”状态。
8. 是，则进行如下操作：
9. 记录当前手指点位置；
10. 计算与上一个记录点的距离差；
11. 将距离差输入到GUI控制模块。
12. 否，则等待下一帧的输入。



图4-3 算法流程图

1. 光标移动手势

光标移动手势的功能是移动GUI的光标到指定位置。光标移动的动作序列如图所示。

光标移动手势序列由起始动作、移动动作、停止动作3部分构成。

* 1. 起始动作：手势呈“光标移动”静态手势。
  2. 移动动作：手势保持“光标移动”手势，在【阈值空间内】，手掌平行于摄像机平面，朝任意方向移动。
  3. 停止动作：展开5指，手势保持“停止”静态手势。



图4-4 光标移动手势动作序列

光标移动手势的识别算法的流程图如图所示。识别算法具体描述如下：

1. 输入静态手势，判断手势是否为“光标移动”手势。
2. 是，则跳转至（2）。
3. 否，等待下一帧的输入。
4. 判断当前是否处于“光标移动”状态。
5. 是，则进行如下操作：
6. 记录当前手指点位置；
7. 计算与上一个记录点的沿着X和Y方向的位移；
8. 将位移输入到GUI控制模块。
9. 否，则进入“光标移动”状态。等待下一帧的输入。

【 】

图4-5 识别算法流程图

1. 轨迹书写手势

轨迹书写手势的功能是凌空书写26个大写英文字母的空间手写轨迹。

轨迹书写手势的手势序列由起始动作、书写动作和停止动作3部分构成。

* 1. 起始动作：手势呈“轨迹书写”静态手势。
  2. 书写动作：手势保持“轨迹书写”手势，在【阈值空间内】，手掌平行于摄像机平面，连笔书写。
  3. 停止动作：展开5指，手势保持“停止”静态手势。



图4-6 轨迹书写手势动作序列

轨迹书写手势的识别算法的流程图如图所示。识别算法具体描述如下：

1. 输入静态手势，判断手势是否为“光标移动”手势。
2. 是，则跳转至（2）。
3. 否，等待下一帧的输入。
4. 判断当前是否处于“光标移动”状态。
5. 是，则进行如下操作：
6. 记录当前手指点位置；
7. 计算与上一个记录点的沿着X和Y方向的位移；
8. 将位移输入到GUI控制模块。
9. 否，则进入“光标移动”状态。等待下一帧的输入。

【 】

图4-7算法流程图

在对轨迹书写手势进行识别之后，通过对手势唯一的手指点进行跟踪，得到了手指点的空间手写轨迹。空间手写轨迹可以看作是手指点位置信息的时间序列。对位置信息进行处理，可以提取得到任意时刻的速度、加速度等信息，以此作为轨迹的特征序列。借助特征序列，可以实现对手写轨迹的识别。

本章的余下章节将会对空间手写轨迹的识别做详细的讨论。

1. 空间手写轨迹识别

手写轨迹识别问题，既可以看作是平面手写识别问题，也可以看作是空间手写轨迹的识别问题。

平面手写识别的研究中，研究者采用了笔触位置、笔画的斜率、方向等特征来描述平面的手写笔迹，然后结合神经网络算法进行识别。Jaeger等人开发的NPen++手写识别引擎多状态时间延迟神经网络（Multi-State Time Delay Neural Network）并对轨迹采集了11种特征，对5000词汇量的语料库达到了96%的识别率。

空间手写轨迹识别的研究中，研究者分别采用了方向角特征、加速度特征、时域（RF、WPD）特征与频域（FFT）特征融合技术来描述手的运动轨迹，并通过各种HMM分类器、SVM分类器、神经网络分类器对数字0-9的手写轨迹进行分类识别，取得了85%~94%的识别率。这些研究表明，轨迹特征和分类器的组合对识别率有关键影响。金连文等人利用普通RGB摄像头捕捉人手指的运动轨迹，通过【是不是手指跟踪及检测算法】，恢复出手指虚拟“书写”文字的二维图像数据，再进行识别输出。实验结果表明，在稳定光照背景下，该系统对阿拉伯数字、英文字母的识别率能达到95%以上。2014年，Vikram等人通过Leap Motion体感设备，采集了手指的空间手写轨迹，并且通过DTW分类器对单个英文字母进行识别。

两类问题的共同之处在于，都是从轨迹提取出特征序列，通过分类算法进行分类识别。但是，相比平面手写识别，本文实现的空间手写识别没有提笔/落笔的动作，因此，书写轨迹是连续无停顿的。平面手写识别中许多基于笔画的特征无法应用到空间手写识别的场景中。

另一方面，局部特征中轨迹上某点的斜率、书写方向、书写速度等基于采样点的特征，仍然可以描述空间手写轨迹。因此，本文采用基于采样点特征描述空间手写轨迹。

如图所示，为了对特征序列进行分类识别，常见的做法【引用参考文献】是将输入的轨迹的特征序列与轨迹类模板的特征序列进行匹配，根据相似度的高低，将轨迹归类为与之相似度最高的模板所属的类。因为书写者完成轨迹输入的时间长短不同，所以即使是同一个字母，其轨迹特征序列也不尽等长。针对这个特点，本文采用DTW算法对轨迹进行分类。

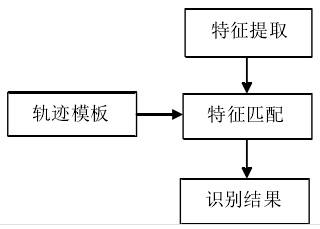


图4-8

在重复书写同一个轨迹的过程中，有很多因素会造成轨迹特征序列的变化：

1. 加速度仪、触摸屏等设备采样率高（50-200Hz），信号数据中夹杂噪声。应攀等人采用频域和时域特征融合技术，使用WPD/RF+FFT特征，减少噪声对分类的影响。
2. 书写过程中，书写速度会造成轨迹输入的时间长短的变化，从而导致特征序列的长度发生变化。
3. 用户面对Kinect的姿势，手距离Kinect的远近也会直接造成轨迹发生旋转、平移、空间尺寸的变化。比如，用户躺着面对摄像机会造成轨迹在摄像机平面内的旋转。用户以坐姿或者站姿面向Kinect时，会造成轨迹所在平面以人体为旋转轴发生旋转。

对于（1），对于Kinect来讲，由于摄像头采样率很低（25Hz），因此不会出现上述情况。对于（2），借助DTW算法，可以很好地解决这个问题。

对于（3），首先，本文采用的采样点的特征具有平移不变性；其次，不论距离的远近，在“阈值范围内”可以精确地获得手的空间位置信息，轨迹空间尺寸的变化不会影响加速度特征。

对于旋转变化的情况，本文做出“用户以**站姿**正面面对摄像机平面”的假设。

1. 轨迹特征

本文采用速度和加速度特征作为识别结果。

1. DTW算法

假设时间序列，。DTW算法通过计算X、Y之间的DTW距离，判断时间序列的差异度。DTW距离值越大，则差异度越高；反之，则越低。DTW距离的定义如下：

（4-1）

其中，是之间的基距离，又可表示为：

 （4-2）

如果是标量，那么可表示为。如果是矢量，那么基距离的计算方法参考【OCR识别的survey】。

由于本研究工作中，采用了采样点的加速度作为轨迹特征，加速度是矢量，因此，在计算加速度特征的基距离时，本文采用余弦距离【引用论文】。

###求解DTW的过程。

###精确计算DTW距离，提高区分度

在分类时，只需要逐个计算测试样本与每个轨迹类模板之间的DTW距离，根据计算结果，将测试样本归为与轨迹类模板的DTW距离最小的类。为了达到更高的区分度，即需要保证：

1. 增加不同类轨迹的特征序列的DTW距离。
2. 减少同类轨迹的特征序列的DTW距离。

###另一方面，从求解DTW距离的过程可以看到，由于没有对轨迹特征序列进行对齐，会出现时间序列上两个根本不在同一个位置的采样点的特征计算基距离。实际上，由于位置不同，这两个点是不具有可比性的。基于公式（4-2）的DTW距离计算方法不够精确。

为了修正，DTW算法引入了“窗口”这个概念，通过窗口的限制，只计算在时间上相邻的特征向量的基距离。这样，对于同一类轨迹来说，时间上相邻的采样点在空间上。但是，对于不同的轨迹来讲，时间上相邻的采样点在空间上没有任何必然的联系。

由于书写速度会发生变化，即使算法中加入了窗口的限制，也不能排除窗口内的采样点在位置上不相关所造成的影响：

1. 同一个轨迹类内，不相似的位置上有相似的特征向量，DTW距离错误增加。
2. 不同轨迹类内，相似位置，DTW距离错误减少。

这两种情况，都将导致类间的区分度的降低，影响最后的识别结果。根本原因在于没有判别采样点是否在轨迹上的相似位置。

1. 基于位置相似度权重的基距离

为了避免上一节中DTW算法的弊端，在计算基距离时，增加“位置相似度权重”：

（4-3）

其中，表示由采样点共同决定的位置相似度权重。

#### 位置相似度

位置相似度，指的是不同轨迹的特征序列的2个采样点的相对位置的临近程度。

相对位置，指的是任意采样点相对于轨迹起始点的空间位置。

由于空间手写轨迹的采样点不能保证都在同一个平面上，因此，下面先以平面轨迹介绍相对位置的基本概念，再推广到三维的空间手写轨迹。

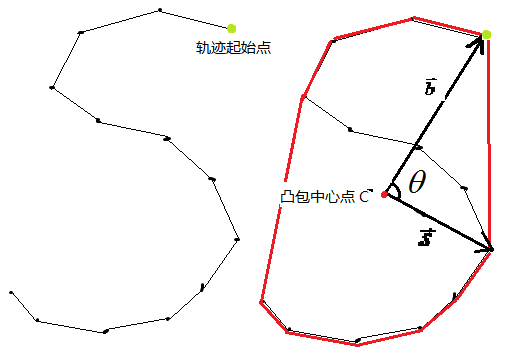


图4-

如图所示。C是平面轨迹的凸包中心点。是C与轨迹起始点B的连线的矢量。是以C为起点，指向轨迹上除了起始点之外的任意采样点S的矢量。是的长度，即采样点到C的空间距离。是和之间的夹角。对于平面轨迹，采样点的相对位置由和2个参数共同描述，即：

 （4-3）

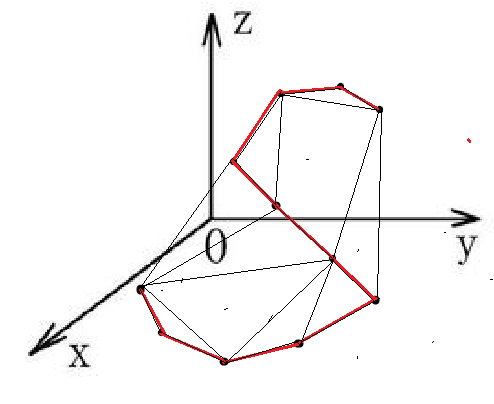


图4-10最小凸包

对于空间手写轨迹（三维），可以计算轨迹的三维凸包，如图所示。只是采样点的相对位置要复杂一些，具体表述为：

 （4-4）

其中，分别是和在YZ，XZ，XY平面内的投影之间的夹角。

假设三维凸包是由轨迹的部分采样点集合构成。那么，对于任意且，C的表达式如下：

 （4-）

求得凸包中心点C之后，下一步便可以计算和，进一步计算相对位置的各个分量。

#### 位置相似度权重的计算

假设分别是轨迹特征序列上任意的两个采样点的特征向量。分别表示的相对位置。位置相似度权重的定义如下：

 （4-5）

其中，表示的分量的近似程度。表示的分量的近似程度。是通过实验确定的常量。

1. 的计算

 （4-6）

其中分别表示的相对位置P的分量。以此类推，分别有：

 （4-7）

 （4-8）

1. 的计算

 （4-9）

其中，分别表示的分量。

1. 取倒数

都是的值，因此，。对于同一类轨迹的2个不同的特征序列（比如测试样本和轨迹类模板），，而对于不同轨迹的特征序列，大多数情况下。我们希望的是同一类轨迹的DTW距离足够小，而“放大”不同轨迹的DTW距离，因此，通过取倒数来达到效果。

1. 参数

当时，会趋向于一个很大的正实数。在使用计算机的基础数据类型表示时，很有可能溢出。因此，通过参数限定分母的下界：。参数通过实验得到，一般取值为0.1。

1. 本章小结

本章首先介绍了【抓取、光标移动、轨迹书写】手势的功能、定义和识别算法。本文通过DTW算法，对【不通畅：轨迹书写手势生成的大写英文字母轨迹】进行识别。针对DTW算法在计算采样点特征向量的基距离时，没有考虑采样点在轨迹上的相对位置的相关度的缺陷，本章余下部分讨论了通过增加位置相关度权重，提高基距离计算的精确性，并且提高不同类轨迹之间的DTW距离的差异度的方法，优化DTW算法的分类效果。

1. 系统硬件及软件环境
2. 实验软硬件环境

本文第3章中,对基于Kinect的静态手势识别方法做了详细的分析,第4章接着给出了基于Kinect的动态手势识别方法,两者结合即可完成基于Kinect的手势识别。本章主要是进行了基于Kinect的手势识别软件的具体实现,并测试手势识别的结果。

1. 硬件环境

PC主机：华硕笔记本电脑，其型号为X42JZ，CPU为Intel(R) Core(TM) i3 480M，主频为2.13 GHz。内存为2.00 GB。显卡为ADM Radeon HD 6470M，显存为512M，支持DirectX11。

Kinect for Windows v1：由Microsoft研制发布，可以采集RGB和深度数据。水平视角为57度，垂直视角为43度，机身转动范围为±27度，传感深度范围为0.4~3.5米，有效工作距离为5米。相比于Kinect for Xbox，支持近景模式（Near Mode）。

1. 软件环境

操作系统：Windows7 64位 旗舰版

编程环境：Microsoft Visual Studio 2013 个人版

Kinect设备驱动和开发套件：Kinect for Windows SDK 1.8

第三方库：图像处理：OpenCV 2.4.10

1. 软件架构

软件系统的架构图如图5-1所示。

Kinect for Windows SDK负责向手势识别系统提供深度图像数据。手势识别系统则分为数据处理模块和数据显示模块。数据处理模块，负责逐过程处理输入的深度图像得到手势区域、手指点等数据。数据绘制模块，负责将逐过程得到的数据通过可视化的方式反馈给用户。比如，将手势区域的轮廓绘制出来，通过红圈标志出手指点。

当识别出具体的动态手势之后，手势识别系统会记录动态手势所表达的控制数据。在一定时刻，控制数据会发送给智能电视的用户接口，控制智能电视的行为。比如，手势识别系统会记录光标移动手势控制的光标移动的位移，在每一帧处理完之后，发送给智能电视，控制光标的移动。

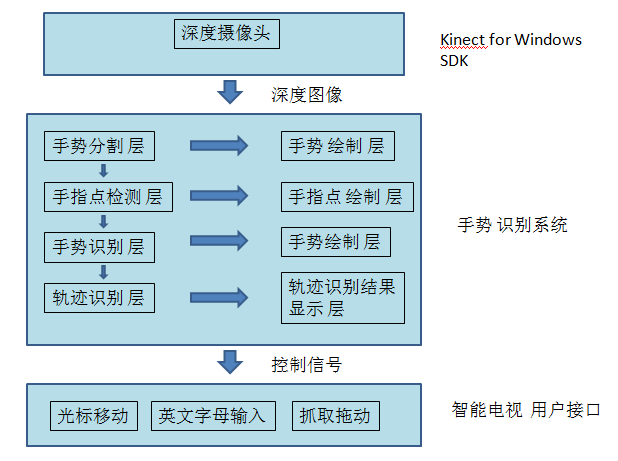


图5-1 软件系统架构图

1. 手指点检测

首先对FT-GB算法的准确性进行了测试。测试分成2个阶段：

第1个阶段，测试不同手指的单个手指点的识别率。手臂平行于摄像机平面，从大拇指开始，轮流伸出1根手指，并且以手臂为转轴，手掌按照一定的转动角度，保持2秒。每个角度重复10次。统计手指点的识别率。

测试结果如表所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 45° | 90° | 135° | 180°（0°） |
| 大拇指 | 90% | 90% | 90% | 90% |
| 食指 | 90% | 90% | 90% | 90% |
| 中指 | 90% | 90% | 90% | 90% |
| 无名指 | 90% | 90% | 90% | 90% |
| 小拇指 | 90% | 90% | 90% | 90% |

表5-1 不同角度各手指识别率

第2个阶段，测试多个手指点的识别率。手臂平行于Kinect摄像平面，从大拇指开始，轮流伸出1~5根手指，并且以手臂为转轴，手掌按照一定的转动角度，保持2秒。每个角度重复10次。统计手指点的识别率。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 30° | 60° | 120° | 180°（0°） |
| 1根手指 | 90% | 90% | 90% | 90% |
| 2根手指 | 90% | 90% | 90% | 90% |
| 3根手指 | 90% | 70% | 90 | 90% |
| 4根手指 | 90% | 50% | 90% | 90% |
| 5根手指 | 100% | 40% | 100% | 100% |

表5-2 不同角度不同数量手指识别率

通过分析实验数据，可以看到：

* 1. 只有一个手指点时，不同手指的识别率在不同角度下都经近似为90%。
  2. 当出现多个手指点时，
     1. 如果手指点个数少于3个时，由于阶段2的实验2角度中没有发生遮挡，因此，多个手指点的识别率和单个手指点一致。
     2. 当手指点的个数多余3个时，在60°这个角度发生了遮挡，因此，识别率降低。而且，当手指个数越多，遮挡部分更加多，识别率越低。
  3. 在没有遮挡干扰的情况下，单个手指点的识别率和多个手指点的识别率几乎是相同的。这是因为FT-GB算法利用的是手指的几何性质，不受手指数量的影响。

本文中的静态手势分别有1、2、5个手指点，而且做手势时，手掌平行于Kinect摄像平面的。基于手指点检测的实验结果，静态手势可以取得90%以上的识别率。因此，下一节将会进行手势识别的测试。

1. 静态手势识别

实验测试3.4节定义的静态手势的识别率。实验时，手掌平面平行于Kinect摄像平面，做出手势后保持静止，持续5秒。实验对每个静态手势采集了50个手势样本。测试结果如表所示。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 实验次数 | 识别次数 | 识别率 |
| 停止手势 | 50 | 49 | 98% |
| 抓取-开始手势 | 50 | 48 | 96% |
| 抓取-拖动手势 | 50 | 45 | 90% |
| 光标移动手势 | 50 | 47 | 94% |
| 轨迹书写手势 | 50 | 47 | 94% |

表5-3 静态手势识别率

分析实验数据，可以看到：

* + 1. 静态手势的识别率与动态手势的识别率基本保持一致。
    2. 抓取拖动手势的2个捏合成1个的手指点可以被FT-GB算法准确地识别。

在本文定义的所有静态手势的识别率达到90%以上的情况下，下一节对动态手势识别进行实验。

1. 动态手势识别

实验测试4.2~4.3节定义的动态手势的识别率。实验时，手掌平面平行于Kinect摄像平面。每个动态手势重复做50次。测试结果如表所示。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 实验次数 | 识别次数 | 识别率 |
| 抓取手势 | 50 | 49 | 98% |
| 光标移动手势 | 50 | 48 | 96% |
| 轨迹书写手势 | 50 | 46 | 92% |

表5-4 动态手势识别率

通过实验数据可以看到，动态手势的识别非常稳定和准确。静态手势在添加了动作序列之后，由于FT-GB算法的优越性能，识别率并未受到影响。

1. 空间手写轨迹的识别

实验测试A-Z共26个大写英文字母。实验分为训练和测试2个阶段。首先，在训练阶段，测试者通过轨迹书写手势依次输入26个大写英文字母的轨迹作为轨迹模板。接着，在测试阶段，测试者书写同一个字母100次作为测试样本。

表5-5是采用了传统加速度特征和DTW算法进行识别的实验结果。表5-6是加速度特征结合4.4.3中介绍的位置相似度权重的实验结果。

|  |  |  |  |
| --- | --- | --- | --- |
| 手势名称 | 实验次数 | 识别次数 | 准确率 |
| A | 100 | 89 | 89% |
| B | 100 | 87 | 87% |
| C | 100 | 87 | 87% |
| D | 100 | 90 | 90% |
| E | 100 | 92 | 92% |
| F | 100 | 89 | 89% |
| G | 100 | 87 | 87% |
| H | 100 | 87 | 87% |
| I | 100 | 90 | 90% |
| J | 100 | 92 | 92% |
| K | 100 | 89 | 89% |
| L | 100 | 87 | 87% |
| M | 100 | 87 | 87% |
| N | 100 | 90 | 90% |
| O | 100 | 92 | 92% |
| P | 100 | 89 | 89% |
| Q | 100 | 87 | 87% |
| R | 100 | 87 | 87% |
| S | 100 | 90 | 90% |
| T | 100 | 92 | 92% |
| U | 100 | 90 | 90% |
| V | 100 | 92 | 92% |
| W | 100 | 89 | 89% |
| X | 100 | 87 | 87% |
| Y | 100 | 87 | 87% |
| Z | 100 | 90 | 90% |

表5-5未使用位置相关权重时的识别率

|  |  |  |  |
| --- | --- | --- | --- |
| 手势名称 | 实验次数 | 识别次数 | 准确率 |
| A | 100 | 89 | 89% |
| B | 100 | 87 | 87% |
| C | 100 | 87 | 87% |
| D | 100 | 90 | 90% |
| E | 100 | 92 | 92% |
| F | 100 | 89 | 89% |
| G | 100 | 87 | 87% |
| H | 100 | 87 | 87% |
| I | 100 | 90 | 90% |
| J | 100 | 92 | 92% |
| K | 100 | 89 | 89% |
| L | 100 | 87 | 87% |
| M | 100 | 87 | 87% |
| N | 100 | 90 | 90% |
| O | 100 | 92 | 92% |
| P | 100 | 89 | 89% |
| Q | 100 | 87 | 87% |
| R | 100 | 87 | 87% |
| S | 100 | 90 | 90% |
| T | 100 | 92 | 92% |
| U | 100 | 90 | 90% |
| V | 100 | 92 | 92% |
| W | 100 | 89 | 89% |
| X | 100 | 87 | 87% |
| Y | 100 | 87 | 87% |
| Z | 100 | 90 | 90% |

表5-6 增加位置相关权重后的识别率

分析实验数据可以看到：

* 1. 采用位置相关权重特征的平均识别率（95%）相比于传统特征的平均识别率（90%）高5%；
  2. 采用了位置相关权重之后，对于英语中出现频率高的字母（etaoinshrdlu，按照出现的频率排列），除了E、F、O、U之外，识别率都在90%以上。
  3. 字母E和F、O和Q、U和V的相互误识别率比较高。原因在于这几个字母的轨迹比较相近，轨迹特征序列比较相似。

1. 本章小结

本章首先介绍了本研究工作基于的软硬件实验环境，以及研究工作中开发的软件系统的软件架构。接着对手指点检测算法、静态手势识别、动态手势识别的识别率进行了实验。实验结果表明基于FT-GB手指点识别算法使得基于手指点数量和位置检测的静态手势识别、动态手势识别取得了【90%以上】的识别率。在此基础上，本文通过DTW算法对轨迹书写手势输入的26个大写英文字母的轨迹进行了识别，并且比较了增加位置相关度权重前后识别率的变化。实验表明，位置相关度权重可以对识别率有5%的提高。

结论和展望

## 结论

将基于手势的新型人机交互技术与智能电视的图形用户界面进行交互，

可以（1）摆脱对传统的遥控器的依赖，（2）使得用户仅通过日常生活中常见的手势，就可以实现光标移动、滚动条抓取拖动、以及英文字母的输入等在使用智能电视时常见的操作，（3）不仅提供了较高的效率和便捷度，增强了智能电视的用户体验，还避免了用户学习使用传统遥控器时花费的学习时间，（4）体现“用户即控制器”的自然人机交互的理念。配备了深度传感技术的Kinect系列体感设备可以为手势识别技术提供较为可靠的手部各部分的空间位置信息，为准确地描述手的轮廓、手指空间位置等信息提供了可靠基础。本研究工作正是借助Kinect在深度传感技术上的特点，实现了前述的手势控制，并取得较高的稳定性和准确率。

本文的工作主要包括以下几点：

* + 1. 陈述了人机交互的历史和最新发展趋势，介绍了基于手势识别的人际交互技术的人际交互技术的特点和研究现状和成果，着重介绍了动态手势识别的最新进展。
    2. 利用Kinect深度摄像头获取包含手势区域的深度图像，提出了使用K-Means聚类算法分割手势区域，并借鉴Maisto等人[7] 的方法，提取手势轮廓，并采用近似多边形，近似描述手势轮廓。为了获取准确有效的手势区域，本文通过实验确定了手距离Kinect摄像头平面的最优距离范围。
    3. 使用FT-GB算法，逐个判断手势轮廓上的边界点是否为手指点，通过手指点的数量，实现静态手势识别。通过实验，确定了FT-GB的A\_Thres参数。
    4. 基于静态手势识别的成果，对本文定义的抓取、光标移动、轨迹书写这3个动态手势进行识别。实验表明，基于静态手势和自定义动作序列组合的动态手势具备较高的识别率，并且，静态手势和动作序列可以根据用户习惯，被用户重新组合，因此，具有良好的扩展性。
    5. 采用DTW算法，对动态手势输入的大写英文字母轨迹进行了识别。针对DTW算法的缺点，本文提出在计算基距离时，利用位置相似度权重保证与同类轨迹的DTW距离准确度的同时，扩大不同轨迹类模板与测试样本的DTW距离的差异度，提高分类的准确率。

本文的创新点主要包括以下两点：

* + 1. 针对空间轨迹识别的应用场景，本文提出了空间位置相似度权重来改进DTW算法在计算基距离时，没有考虑采样点在空间位置上的相关性的缺陷，提高了DTW算法的性能。
    2. 将基于Kinect的手势识别技术应用到智能电视的【图形用户界面】的交互中，对于基于手势的人机交互技术在智能电视领域的应用做出一些探索。

## 改进与展望

虽然本研究工作通过实验，展示了手势识别技术的令人期待的效果，但是，这项技术要投入实际应用还有许多需要改进的地方：

* 1. 从手势采集设备来讲，由于Kinect的摄像头是基于视觉信息的，在操作过程中，一些角度下（比如手掌垂直于摄像机平面），一些手势（比如5指并拢），手指会发生相互遮挡，导致采集得到的手势深度数据缺乏足够的手势信息，影响后续的算法流程。
  2. 在手势区域分割时，仅仅通过阈值从深度图像中筛选手势区域像素点是无法满足复杂场景下的交互需求的，因为此时可能会有杂物处于【手势舞动却与内】，被误识别为手势。因此，可以考虑佩戴指环来追踪手的空间位置，提高手势分割的准确度。
  3. 基于手指轮廓的静态手势识别虽然在伸直的手指上取得了很好的效果，但是并不适用于手指弯曲、指向摄像头的情况。这些情况在构建复杂手势时，必然会出现。这是一个重要的问题。

尽管还有如上诸多亟待改进之处，但总的说来，将手势识别技术应用到“客厅娱乐”这个日常生活中的重要场合是继基于Kinect系列产品的人体体感技术问世之后的必然趋势。

# 致谢

历时一年多的论文终于完成了，在这一年里，从学习资料的搜集到具体的分析与实现，这期间遇到了许多困难和障碍，在完成之际，本人向所有关心我、帮助我、支持我的老师和朋友们道一声最诚挚的感谢。

尤其要感谢我研究生阶段的导师景红教授，景老师在我完成论文阶段，为我提供了良好的科研氛围，每当研究过程中遇到问题，老师都尽力帮我解决。对于研究中出现的不足之处，老师会及时的指出并帮助我改正，保证了我顺利的完成了课堂研究工作。除了在学业上，在生活中老师也给及我无微不至的关怀。每当我遇到挫折时，老师都如母亲般的给予我关爱和鼓励，同时为我提出了宝贵的意见，帮助我克服困难。在此，向景红老师表达我最崇高的敬意。

感谢我的周围的同学，在我写论文遇到困难时，你们给予我无私的帮助，帮助我克服一个又一个阻碍。同时，当我在生活中遇到挫折，需要帮助的时候，你们也无时无刻不在我身边，帮助我安慰我。感觉你们在我研究生阶段一直陪在我身边。

感谢我的父母，感谢父母辛勤把我养育成人，默默的无私的为我付出，陪伴着我从一个幼儿长大成人。我能从一个懵懂无知的少年，成长成为一名研究生，这些与父母对我的爱是分不开的，他们陪伴这我面对人生一个又抉择，度过一个又一个困难，分享一个又一个喜悦，是他们给予我不断学习进取的动力。

最后，衷心感谢对我论文进行评阅和指导的各位专家和教授。

# 参考文献

1. 李玉朵.基于SVM的人脸表情识别研究.河北工程大学.2012年5月
2. http://viml.nchc.org.tw/blog/sub\_class.php?SUB\_ID=1&CLASS\_ID=1

**References:**

[1]. 臧圣国, 黄俊与王沁莹, NUI交互体系对智能电视发展的探讨. 有线电视技术, 2014(02): 第99-103页.

[2]. Handwritten Character Recognition Using Orientation Quantization Based on 3D Accelerometer.

[3]. Airwriting: Hands-free Mobile Text Input by Spotting and Continuous Recognition of 3d-Space Handwriting with Inertial Sensors.

[4]. 邹节华, 基于单目视觉的动态手势轨迹识别系统研究, 2012, 西安电子科技大学. 第 75页.

[5]. 张毅等, 基于Kinect深度图像信息的手势轨迹识别及应用. 计算机应用研究, 2012(09): 第3547-3550页.

[6]. 王松林, 基于Kinect的手势识别与机器人控制技术研究, 2014, 北京交通大学. 第 82页.

[7]. Maisto, M., et al., An Accurate Algorithm for the Identification of Fingertips Using an RGB-D Camera. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2013. 3(2): p. 272-283.