

多点触摸手势分析及识别算法的研究

王晓庆^{1,2} 陈戈² 王栋^{2,3} 王春⁴

(中国海洋大学信息科学与工程学院电子工程系 青岛 266100)¹

(中国海洋大学海洋信息技术教育部工程研究中心 青岛 266100)²

(中国海洋大学图书馆 青岛 266100)³ (化学品安全控制国家重点实验室 青岛 266071)⁴

摘要 针对目前多点触控系统缺乏对触摸手势含义的理想描述和解析,提出了一种通用的多点触摸手势分析与设计框架,研究了高性能的算法合理解析并优化多点触控指令,使其更符合用户的思维与认知。设置触点位移和时间函数双阈值来提高触点识别的精确性,防止突增杂点的误判断,减缓过快操作产生的抖动;采用 RBF 神经网络模型解决动态手势识别的问题,并引入基于欧氏距离函数的聚类统计量作为网络的特征参数,大大提高了多点触摸手势识别的效率和精确度。

关键词 多点触摸,触摸手势,RBF 神经网络,欧氏距离,双阈值

中图分类号 TP391 文献标识码 A

Research on Multi-touch Gesture Analysis and Recognition Algorithm

WANG Xiao-qing^{1,2} CHEN Ge² WANG Dong^{2,3} WANG Chun⁴

(Department of Electronic Engineering, College of Information Science & Engineering, Ocean University of China, Qingdao 266100, China)¹

(Engineering Research Center of Marine Information Technology, Ministry of Education, Ocean University of China, Qingdao 266100, China)²

(Library, Ocean University of China, Qingdao 266100, China)³

(National Registration Center for Chemicals, SAWS, Qingdao 266071, China)⁴

Abstract Currently, multi-touch gestures on the touch system lack of an ideal description and analysis. Research on Multi-touch gesture analysis and recognition algorithm presents a general multi-touch gesture analysis and design framework, rational analysis of high performance algorithms and optimizing multi-touch commands. Contact displacement and time to set the threshold to improve the dual function of identifying the accuracy of contact, to prevent the sudden increase in the noise of false judgments, slow down fast operation jitter; using RBF neural network model to solve the problem of dynamic gesture recognition, and the introduction of Euclidean distance-based clustering statistics as a function of the characteristic parameters of the network, greatly improving the multi-touch gesture recognition efficiency and accuracy.

Keywords Multi-touch, Touch gesture, RBF neural networks, Euclidean distance, Dual-threshold

1 引言

近年来,随着苹果手机的广泛应用以及 windows 7 操作系统对多点触控技术^[1]的支持,多点触控技术发展十分迅速。从手机、GPS 定位导航仪、平板电脑等中小尺寸的个人数码便携产品到会展、博物馆、互动会议、商业店面等大尺寸商用显示和触摸产品,多点触控技术这一时尚、前沿的人机交互技术得到广泛的关注和应用。

目前支持多点触摸的系统,缺乏对触摸手势含义的理想描述和解析,多点操作指令解析大都是在程序代码内部加以

判断,这就带来一系列问题:如各个程序对于多点指令判断思路混乱;运行不同的多点程序时需要运行不同的轨迹诠释驱动软件;同时运行多个多点程序时容易产生指令误判断;多点触控操作的精确性、防抖、防误判断效果不理想。

本文提出一种通用的多点触摸手势分析与设计框架,研究了高性能的算法合理解析并优化多点触控指令,使其更符合用户的思维与认知。交互设计者能够深入了解多点触摸平台中所有可能的手势,从中选择出对其有用的手势为应用系统服务,大大提高了多点触控手势指令的统一性、规范性和普适性。高性能的智能算法合理解析多点手势指令,并对其进

本文受科学技术部科技人员服务企业行动项目(2009GJA00047),国家重点实验室开放研究基金项目(SKL-014)资助。

王晓庆(1986-),女,硕士生,主要研究方向为多媒体通信网络与系统设计。

行拓展和优化,提高应用程序与系统的兼容性。

2 多点触控系统人机交互手势定义

触摸手势动作,是触摸交互过程中根据单手或双手手指在触摸平台表面的触摸状态、触点位置或触点相对位移特征加以区分的有特定含义的触摸动作^[2]。

多点触摸系统通过对原始触摸图像抓帧、二值化、提取外部轮廓、判断并定位触点之后获得触点信息,这一过程称为触点检测跟踪^[3],如图 1 所示。

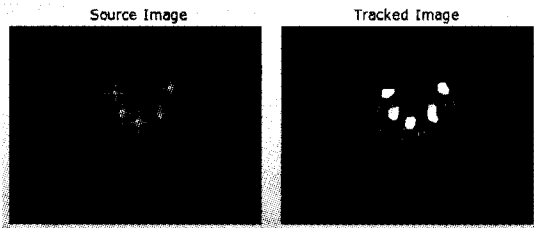


图 1 系统根据原始图像检测生成触点

多点触摸系统在检测和定位出多个触点后,分配给每个触点唯一的 ID 号,并对触点进行跟踪,记录每个触点的坐标变化 (x, y) 及触点的生命周期 Time,这一过程称为触点追踪。识别出的触点信息被封装成遵循 TUIO 格式的 XML 数据包,该数据包的格式如图 2 所示。

```

- <OSCPACKET ADDRESS="127.0.0.1" PORT="1172" TIME="1">
- <MESSAGE NAME="/tuio/2Dcur">
  <ARGUMENT TYPE="s" VALUE="set" />
  <ARGUMENT TYPE="f" VALUE="10158" />
  <ARGUMENT TYPE="f" VALUE="0.40423685" />
  <ARGUMENT TYPE="f" VALUE="0.40103334" />
  <ARGUMENT TYPE="f" VALUE="0.0" />
  <ARGUMENT TYPE="f" VALUE="-0.0" />
  <ARGUMENT TYPE="f" VALUE="0.0" />
  <ARGUMENT TYPE="f" VALUE="25.0" />
</MESSAGE>
- <MESSAGE NAME="/tuio/2Dcur">
  <ARGUMENT TYPE="s" VALUE="alive" />
  <ARGUMENT TYPE="f" VALUE="10158" />
</MESSAGE>
- <MESSAGE NAME="/tuio/2Dcur">
  <ARGUMENT TYPE="s" VALUE="fseq" />
  <ARGUMENT TYPE="f" VALUE="53128" />
</MESSAGE>
</OSCPACKET>

```

图 2 系统对触点追踪并记录数据信息

完成这两个过程后,系统解析触摸手势含义并做出响应,从而实现基于触摸手势的自由交互。

2.1 触摸手势元动作

触摸手势元动作^[4]是构成多点触摸手势的基本单元。由于所有的触摸手势动作都是在触摸屏幕这一二维平面上实现的,因此多点触摸手势动作是一系列二自由度的手势动作。触摸手势元动作是对单触摸点的描述,其可以由触点检测和触点追踪两个过程中的 4 个属性构成的数据结构来表示:

$$\text{Point} = \{ \text{ID}, \text{State}, \text{Coordinate}, \text{Time} \}$$

其中 $\text{ID} = \{ 1, 2, \dots \}$, 表示识别触点的身份识别号,对于每个触点只有唯一的 ID 号与其对应,这就使得连续记录该触点的轨迹和时间参数成为可能; $\text{State} = \{ 0, 1 \}$ 表示触摸点相对于

屏幕的状态, $\text{State} = 0$ 表示接触, $\text{State} = 1$ 表示未接触; $\text{Coordinate} = P(x, y)$, 表示触摸点的坐标, $\text{State} = 0$ 即接触状态时系统记录触点的坐标及其变化信息; Time 表示触点的生命周期,即从开始接触($\text{State} = 0$)到触摸结束($\text{State} = 1$)的时间,可以通过检测系统时间得到。

触摸手势由一个或者多个手势元动作构成,每个触摸手势可以由 5 个属性构成的数据结构来表示:

$$\text{Gesture} = \{ \text{Number}, \text{ID}, \text{State}, \text{Coordinate}, \text{Time} \}$$

其中, $\text{Number} = \{ 1, 2, \dots \}$ 代表当前系统检测到的触摸点的个数; ID 表示触点识别号; State 表示触点接触状态; Coordinate 表示触点坐标; Time 表示手势持续时间。

2.2 多点触摸手势的分类及描述

通过对触摸手势元动作分析不难发现,触摸手势动作实际上是触摸元动作按照时间顺序排列组合而成的。基于触点轨迹变化触摸手势可以分为:静态手势和动态手势。基于触摸点个数触摸手势可以分为:单点触摸手势和多点触摸手势,本文中把单点触摸手势作为多点触摸手势的一个特例进行分析。基于运动方式的不同,触摸手势可以分为:点击手势和笔画手势。静态手势和动态手势集分别如表 1 和表 2 所列。

表 1 静态手势集

编号	触点	手势	操作语义
1	1	单指点击	鼠标左键单击、选取对象
2	2	二指点击	鼠标左键双击
3	3	三指点击	鼠标右键单击
4	4	四指点击	鼠标右键双击
5	5	五指长按	刷新页面

表 2 动态手势集

编号	触点	手势	操作语义
1	1	单指触摸沿直线拖动	推动选取对象
2	1	单指画对号	确认操作
3	1	单指画圈	选中多个对象
4	2	两指远离	放大对象
5	2	两指靠近	缩小对象
6	2	两指旋转	旋转对象
7	3	三指向下拖动	当前程序常用导航
8	3	两指静态触摸+单指水平拖动	3D 场景变换水平视角
9	3	两指静态触摸+单指上下拖动	3D 场景变换俯仰视角
10	4	三指静态触摸+单指点击	复选
11	4	三指静态触摸+单指沿直线推动	复选组整体缩放
12	4	三指静态触摸+单指画弧	复选组整体旋转
13	5	五指相互靠近	显示桌面

2.3 多点触摸手势建模

触摸手势元动作模型如图 3 所示,表示一触摸点轨迹从 A 位置移动到 B 位置的过程中系统记录下该点的属性数据。其中 i 代表该触摸点是系统当前检测到的第 i 个触摸点; (x_{i0}, y_{i0}) 表示这一元动作的触摸起始坐标, (x_{in}, y_{in}) 表示触摸结束坐标; (x_{ij}, y_{ij}) 表示这一元动作的一系列轨迹变化。

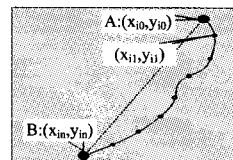


图 3 触摸手势元动作模型

为了后面的触摸手势学习和识别的需要,在触摸轨迹坐标序列中选取确定数目的代表坐标点,比如 17 个。如果轨迹坐标序列集合的坐标个数超过 17 个,则通过查找距离最小的两个点,用它们的中点取代它们。重复执行,直到坐标点个数为 17 个。根据这 17 个代表点可以得到 16 个二维向量,并对它们进行归一化处理,从而得到该手势元动作的归一化特征向量,将其作为识别算法的输入。

一个或多个触摸手势元动作按照时间顺序排列组合形成触摸手势,有 m 个触点的触摸手势特征矩阵为:

$$\begin{bmatrix} ID_1 & State_1 & (x_{10}, y_{10}) & \cdots & (x_{1n}, y_{1n}) & Time_1 \\ ID_2 & State_2 & (x_{20}, y_{20}) & \cdots & (x_{2n}, y_{2n}) & Time_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ ID_m & State_m & (x_{m0}, y_{m0}) & \cdots & (x_{mn}, y_{mn}) & Time_m \end{bmatrix}$$

3 多点触摸手势的识别方法

为保证多点触控系统对触摸手势指令识别的正确率和效率,识别的方法需基于上述定义的触摸手势的特点进行设计。如图 2 触点信息封装的 TUIO 数据包中,包含了计算机的 IP 地址 OSCPACKET ADDRESS、通信端口号 PORT、时间戳 TIME、触点的 ID 号、坐标等。因此只要解析这些多点数据包,即可获得所有的触点信息。

3.1 静态手势识别

3.1.1 识别方法

涉及静态触摸手势的识别时,传统的识别方法是:先判断触摸点的个数,进行触点个数的分类;再通过查找手势集合数据库中与其相匹配的静态触摸手势,确定是何种手势^[5]。

但是在实际应用中,这种识别方法的识别效率并不是十分理想。经过测试主要存在两方面的问题:一是由于光线等原因触摸屏突增杂点也被系统记录为有效触点,从而产生误判断;二是触摸手势操作过快时容易产生触点抖动,以致系统误判或丢失触点。

3.1.2 改进后的识别方法

改进的方法是对每个触点添加触点位移和时间间隔检测,并设定阈值函数,其流程如图 4 所示。

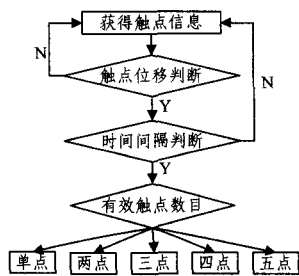


图 4 改进后的静态手势识别流程

触点位移阈值函数:

$$\lambda_1 = f(x, y) = \begin{cases} 0, & \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} \geq a \\ 1, & \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} < a \end{cases}$$

时间间隔阈值函数:

$$\lambda_2 = f(t) = \begin{cases} 0, & t < a \\ 1, & t \geq a \end{cases}$$

经过改进后,静态触摸手势识别需要输入达到设置的门限值时才会启动激活函数,从而被判别为有效的触点;阶跃函数作为激活函数,可防止突增杂点的误判断,减缓过快操作产生的抖动;设定触点位移和时间函数双阈值能够提高识别的精确性。

3.2 动态手势识别

相比静态手势的识别,多点触控动态手势的识别方法则要复杂得多。本文重点讨论基于径向基函数网络的多点触摸手势的识别算法及其改进。

3.2.1 RBF 神经网络的结构

径向基函数神经网络(RBF 网)模仿生物神经元的“近兴奋远抑制”的功能,是一种典型的局部逼近网络,可以逼近任意的非线性函数^[6]。其结构一般分为 3 层,如图 5 所示^[7]。

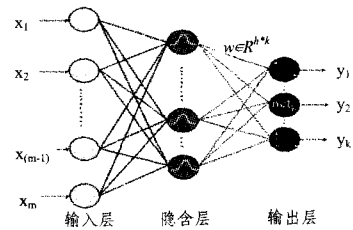


图 5 RBF 网络结构

图 5 为 $m-h-k$ 结构的 RBF 网,网络具有 m 个输入, h 个隐节点, k 个输出。其中 $x = (x_1, x_2, \dots, x_m)^T \in R^m$ 为网络输入矢量, $w \in R^{h \times k}$ 为输出权矩阵, $y = (y_1, y_2, \dots, y_k)^T$ 为网络输出, φ_i 为第 i 个隐节点的激活函数。

隐节点的基函数采用欧氏距离函数^[8]:

$$D_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2}$$

径向基函数采用 Gaussian 函数:

$$\varphi(\text{net}_i \| x - c_i \|) = \exp\left(-\frac{r^2}{2\sigma^2}\right)$$

式中, σ 是基函数的扩展常数,当中心由训练数据确定后, RBF 的宽度可由 σ 确定。设 d 是样本的最大距离, M 是样本的数目,采用固定法确定宽度^[9]:

$$\sigma = \frac{d_{\max}}{\sqrt{2M}}$$

RBF 网的第 r 个输出可表示为:

$$y_k = \sum_{i=1}^h w_i \varphi_i(\| x - c_i \|)$$

3.2.2 RBF 神经网络的训练

设定训练样本集: $X = [X_1, X_2, \dots, X_N]^T$

任意样本: $X_k = [x_{k1}, x_{k2}, \dots, x_{km}]$

实际输出: $Y_k = [y_{k1}, y_{k2}, \dots, y_{kj}]$; $y_{kj}(X_k) = \sum_{i=1}^h w_{ij} \varphi_i(X_k, C_i)$

期望输出: $D_k = [d_{k1}, d_{k2}, \dots, d_{kj}]$

令 $y(x_i) = w_1 \varphi_1(\| x_i - c_1 \|) + \dots + w_{m1} \varphi_{m1}(\| x_i - c_{m1} \|) = d_i$

以矩阵形式表达:

$$[\varphi_1(\|x_i - c_1\|) \cdots \varphi_{m_1}(\|x_i - c_{m_1}\|)] [\omega_1 \cdots \omega_{m_1}]^T = d_i$$

对于所有的样本得到:

$$\begin{bmatrix} \varphi_1(\|x_1 - c_1\|) \cdots \varphi_{m_1}(\|x_1 - c_{m_1}\|) \\ \vdots \\ \varphi_1(\|x_N - c_1\|) \cdots \varphi_{m_1}(\|x_N - c_{m_1}\|) \end{bmatrix} [\omega_1 \cdots \omega_{m_1}]^T =$$

$$[d_1 \cdots d_N]^T$$

令

$$\Phi = \begin{bmatrix} \varphi_1(\|x_1 - c_1\|) \cdots \varphi_{m_1}(\|x_1 - c_{m_1}\|) \\ \vdots \\ \varphi_1(\|x_N - c_1\|) \cdots \varphi_{m_1}(\|x_N - c_{m_1}\|) \end{bmatrix}$$

$$W = \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_{m_1} \end{bmatrix} \quad D = \begin{bmatrix} d_1 \\ \vdots \\ d_N \end{bmatrix}$$

得到 $\Phi W = D$ 。若 Φ 可逆, 则得到 $W = \Phi^{-1} D$ 。

3.2.3 手势的识别

首先根据上述建立输入动态手势的模型, 将其向量集合作为输入向量输入。经过训练好的 RBF 神经网络的运算即可实现触摸手势识别。如果神经网络输出向量与第 k 种手势的标准输出向量越接近, 说明输入手势越接近第 k 种手势。例如 RBF 神经网络的输出为 (0.0121, 0.1810, 0.1667, 0.1830, 0.2130, 0.2817, 0.0386, 0.2585, 0.2410, 0.8936, 0.2422, 0.2510, 0.1164), 最大的分量为第 10 个分量 0.8936, 可判断为第 11 种手势。

4 测试

4.1 普适性测试

针对提出的通用多点触摸手势的描述和解析框架进行普适性对比测试, 如表 3 所列。

表 3 不规范手势解析

运行软件	辅助软件
基于 Touchlib 的多点触控程序	Community Core Vision
单点不涉及轨迹识别程序	Community Core Vision 鼠标辅助程序(TUIO Mouse)
单点涉及轨迹识别程序 (C/C++, java, C#)	Community Core Vision 手势解析程序(VR Driver) 鼠标辅助程序(TUIO Mouse)
多点涉及轨迹识别	Community Core Vision 手势解析程序(VR Driver)

规范化手势解析

运行软件	辅助软件
基于 Touchlib 的多点触控程序	Community Core Vision
点不涉及轨迹识别程序	Community Core Vision
单点轨迹识别程序	Community Core Vision
多点轨迹识别	Community Core Vision

4.2 手势训练耗时间测试

表 4 训练耗时

样本数	耗时/s
8	0.69
16	1.12
32	3.42

4.3 识别训练测试

表 5 识别测试结果

静态手势编号	平均识别时间/s	正确率
1	0.371	96.2
2	0.403	92.0
3	0.375	91.7
4	0.394	89.3
5	0.546	90.6

动态手势编号	平均识别时间/s	正确率
1	0.125	96.2
2	0.176	97.0
3	0.134	91.7
4	0.230	95.0
5	0.234	94.3
6	0.208	95.4
7	0.276	100
8	0.240	92.5
9	0.262	93.6
10	0.328	97.8
11	0.310	98.5
12	0.329	89.5
13	0.134	99.7

4.4 测试结果分析

从实验结果来看, 所提出的触摸手势识别方法识别准确度高、识别速度快、运行辅助软件少、普适性好。

本文采用 RBF 神经网络算法具有并行分布处理、非线性映射、所需特征参数简单明确的特性, 网络资源和时间消耗比较少, 具有较高的识别准确率。本文定义的手势特征明确、参数简单, 因此识别速度快。

结束语 本文对基于触摸手势元动作的多点触摸手势特性进行分析, 对 18 个常用的多点触摸手势进行规范化的描述, 手势特征明确, 并根据其特点提出静态手势识别和动态手势识别方法。

本文创新点: 1) 提出了一种设置触点位移和时间函数双阈值的方法来提高触点识别的精确性, 防止突增杂点的误判断, 减缓过快操作产生的抖动; 2) 采用 RBF 神经网络模型解决动态手势识别的问题, 并引入基于欧氏距离函数的聚类统计量作为网络的特征参数, 大大提高了多点触摸手势识别的效率和精确度。

参考文献

- [1] Jefferson Y H. Low-cost multi-touch sensing through frustrated total internal reflection [C]// Proceedings of the 18th Annual ACM Symposium On User Interface Software and Technology. New York: ACM, 2005: 315-319
- [2] Buxton W, Myers B. A study in two handed input [C]// Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York, USA: ACM, 1986: 321-326
- [3] 齐婷. 基于视觉的多点触控系统关键技术研究述[D]. 昆明: 理工大学, 2011
- [4] 张国华, 衡祥安, 老松杨, 等. 基于多点触摸的交互手势分析与设计[J]. 计算机应用研究, 2010, 27(5): 1737-1739
- [5] 凌云翔, 张国华, 李锐, 等. 基于多点触摸的自然手势识别方法的研究[J]. 国防科技大学学报, 2010, 32(1): 4-5
- [6] 施彦, 韩力群, 廉小亲. 神经网络设计方法与实例分析[M]. 北京: 北京邮电大学出版社, 2009: 41-43
- [7] 韩力群. 人工神经网络教程[M]. 北京: 北京邮电学院出版社, 2006
- [8] 孙丹. 径向基函数神经网络算法改进及其在现金流预测中的应用[D]. 吉林: 吉林大学, 2010
- [9] 张瑾. 基于蚁群聚类优化的 RBF 神经网络软测量应用[D]. 沈阳: 东北大学, 2008