

西南民族大学

本科生毕业设计(论文)

题目：智能手表的非接触式备择交互模式的设计

Designing Alternative Contact-free Control Modalities for Smart Watches

教学单位 计算机科学与技术

姓 名 欧长坤

学 号 201231102123

年 级 2012

专 业 计算机科学与技术

导 师 陈雅茜

职 称 副教授

联合导师 Andreas Butz

职 称 Professor

2016 年 4 月 3 日

目录

摘要	2
Abstract	3
第一章 概述	4
1.1 目的、背景及意义	4
1.2 相关工作	4
第二章 手表上交互	5
2.1 传统交互	5
2.1.1 点击	5
2.1.2 滑动	5
2.2 特有交互	5
2.2.1 Digital Crown	5
2.2.2 Force Touch	5
2.2.3 Taptic Engine	6
2.3 其他交互	6
2.3.1 侧面按钮	6
2.3.2 常规手势	6
2.3.3 语音控制	6
第三章 非接触备择设计	7
3.1 相关工作	7
3.2 技术可行性	7
3.3 交互方法	7
3.3.1 点击	7
3.3.2 滑动	7
3.3.3 Digital Crown	7
3.3.4 Force Touch 仿真	7
3.3.5 其他	7
第四章 系统设计	8
4.1 总体设计	8
4.2 框架选型	8
4.2.1 LeapMotion 与 LeapJS	8
4.2.2 watchOS 与 WatchConnectivity	8
4.3 架构设计	8

4.3.1	通信架构	8
4.3.2		8
4.4	演示程序	8
第五章	编码实现	10
5.1	环境搭建	10
5.1.1	配置本地 LeapMotion 环境	10
5.1.2	配置 watchOS 的网络访问	11
5.2	服务端编码	11
5.2.1	Tap 手势识别	11
5.2.2	Swipe 手势识别	11
5.3	手表端编码	11
第六章	用户调研与测试	12
6.1	用户调研	12
6.2	测试方法	12
第七章	进一步工作	13
7.1	缺陷	13
7.2	改进方向	13
	参考文献	14
	致谢	18

摘要

本文设计了一套用于智能手表上的择备交互模式，能够释放双手。

关键词 智能手表; 手势交互

Abstract

This paper try to introduce a alternative interaction on smart watches, which can be contact-free.

Keywords Smart Watches; Gesture Interaction

第一章 概述

1.1 目的、背景及意义

1.2 相关工作

第二章 手表上交互

在 Apple Watch 中，由于触摸屏的存在，大部分手表中的交互方式沿袭自带有触摸屏的智能手机。为了对手表中的交互方式进行备择设计，我们必须分析并且明确在 Apple Watch 中现存的交互方式及其优缺点。

2.1 传统交互

2.1.1 点击

2.1.2 滑动

2.2 特有交互

2.2.1 Digital Crown

Digital Crown 是苹果公司在 Apple Watch 上推出的一个全新的交互技术，苹果公司在退出此项交互的技术时，将其对比了在人机交互历史中的两个革命性的交互技术：鼠标和触摸屏，这意味着苹果公司认为，Digital Crown 是一项在手表上的革命性交互方式。

这种交互方式利用了传统手表的时钟旋钮在功能可见性上的不足。在传统手表中，时钟旋钮在普通状态下不具备任何功能，只有当旋钮被从里向外拉出时，才具备调节时间的功能，这一装置在大部分时间里都不能发挥自身的作用，是一个典型的需求驱动型设计，并没有仔细考虑过其自身的存在方式，只是习惯性沿用。

而在 Apple Watch 上，信息呈现的方式以流式进行总想展示，所有呈现的内容被限制在一个宽度固定、纵向可伸缩的屏幕区域里。这时，Digital Crown 便能发挥其旋钮的功能。当产生旋转时，内容在竖直方向上进行移动，从而呈现更多的内容；并且，在交互情景发生变化时，Digital Crown 能够表达出不同的交互指令，例如在影月播放界面时，Digital Crown 的旋转能够调节播放音乐的音量。

2.2.2 Force Touch

Force Touch 这项交互技术首次在民用消费品中出现，在学术界中，对触摸的感知被研究了多年，***等一系列文献研究了触觉感知如何在触摸屏上进行增强，包括感觉反馈、触摸面积的测量、触摸力度等等，而 Force Touch 就是触摸力度的实际体现。

Force Touch 一共将触摸行为分为了两个等级，第一触摸等级就是传统意义上的触摸行为，手指轻触屏幕时即可被感知；第二触摸等级就是 Force Touch，这时需要用户将触摸屏幕的力度提升到一个级别后，系统才会进行响应，进一步处理交互。

2.2.3 Taptic Engine

2.3 其他交互

2.3.1 侧面按钮

2.3.2 常规手势

2.3.3 语音控制

第三章 非接触备择设计

3.1 相关工作

3.2 技术可行性

3.3 交互方法

3.3.1 点击

3.3.2 滑动

3.3.3 Digital Crown

3.3.4 Force Touch 仿真

对 Force Touch 进行仿真, 我们需要定义两个常量: DELAY 表示触发 Force Touch 的时间延时。DURATION 表示 Force Touch 从最小值到最大值的持续时间。其中 DELAY 的值为 200, DURATION 则为 1000, 两者单位为毫秒 (ms)。

设在一次按压中的按压时间为 t_{press} , 则 Force Touch 可以使用公式 3.1 进行模拟:

$$v_F = \begin{cases} \frac{t_{\text{press}} - \text{DELAY}}{\text{DURATION}} & \text{if } t_{\text{press}} - \text{DELAY} < \text{DURATION} \\ 1 & \text{Otherwise} \end{cases} \quad (3.1)$$

3.3.5 其他

第四章 系统设计

4.1 总体设计

4.2 框架选型

4.2.1 LeapMotion 与 LeapJS

4.2.2 watchOS 与 WatchConnectivity

4.3 架构设计

4.3.1 通信架构

watchOS 从 2.0 开始从 iOS App Extension 中剥离开来，将 Watch App 部分全部移至 watchOS 端，这时这部分代码在手表端具备了可执行的权限，因此将 watchOS 从 1.0 中的单向接收 iOS 端的系统级的通信，转变为第三方 App 执行管理 [1]，如图 4.1 所示。这使其与外界及时通信成为了可能。

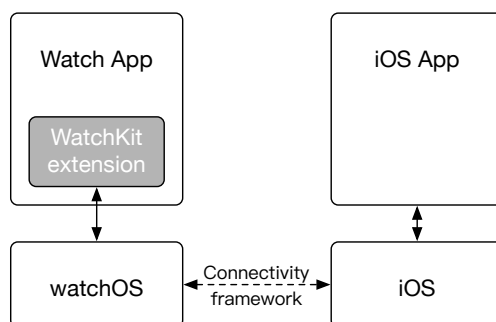


图 4.1: Watch App、WatchKit 扩展和 iOS App 之间的联系

然而，即便如此在 watchOS 上的网络访问能力依然十分有限，在 watchOS 2 中¹，Apple Watch 只能在和与其配对的 iPhone 失去连接，且同时处于已保存的 Wi-Fi 网络覆盖范围内时，才能独立使用 NSURLSession 访问网络，条件十分苛刻。

鉴于以上考虑，本文对从服务端到客户端的通信架构设计如图 4.2 所示。

其中，watchOS 将 iOS 端作为与服务器通信的桥梁，处理性能及其有限的 watchOS 端仅负责对通信内容的呈献，性能稍强的 iOS 端对服务端消息进行筛选与加工，而服务端则对 LeapMotion 原始数据进行分析，并封装其分析结果后与 iOS 端进行通信。为此，我们需要设计在 watchOS 和 iOS 之间、iOS 与服务端之间设计相关的通信协议。

4.3.2

4.4 演示程序

¹ 本文写成时的 watchOS 版本为 2.2。

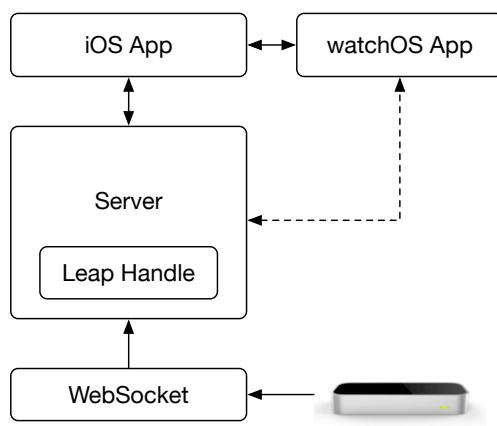


图 4.2: 通信架构: watchOS 不直接与服务器进行通信, 而是将 iOS 端作为与服务器通信的桥梁

第五章 编码实现

5.1 环境搭建

无论是手表端的还是服务端，都存在框架依赖，因此环境搭建不可避免。在本项目中，服务端使用 NodeJS 进行编码，因此对于 NodeJS 相关的基本环境，如 Node 本体，NPM 包管理等常见工具的环境限于篇幅在本文中略去，这里主要介绍运行本平台最重要第三方 Leap Motion 环境；而在手表端中，虽然我们不依赖其他第三方框架，但由于 watchOS 自身的限制?? 在 watchOS 2 中我们需要使用 WatchConnectivity 框架与 iOS 应用本体进行数据通信，而又由于 iOS 系统本身限制（iOS 9 及以上）强制要求应用必须与 HTTPS 服务器进行通信，因此这里介绍在 iOS 9 中与 HTTP 服务器通信的配置方法。

5.1.1 配置本地 LeapMotion 环境

LeapMotion 提供了在 Mac OS X 中的开发环境，并且提供了各种不同的开发语言

1. 安装

2. 环境配置

受到 LeapMotion 自身的限制 [2]，WebSocket 服务并非默认的向非本地访问开放，因此需要将 Leap 配置启用非本地客户端连接。

这需要对 LeapMotion 的配置文件进行修改。在修改配置之前，需要关闭 LeapMotion 的相关服务。在 Mac 中，使用下面的命令关闭 LeapMotion 的守护进程：

```
1 sudo launchctl unload /Library/LaunchDaemons/com.leapmotion.leapd.plist
```

接下来我们需要修改 LeapMotion 的配置文件，根据 LeapMotion 的官方文档显示，Leap 包含两个不同的配置，其中控制面板配置的优先级最高，因此我们需要下面这个目录下：

```
1 $HOME/Library/Application\ Support/Leap\ Motion
```

找到 config.json 的修改配置，编辑 config.json 文件，并在 configuration 字段中的任意位置添加一条：

```
1 "websockets_allow_remote": true
```

最终得到：

```
1 {
2   "configuration": {
3     "websockets_allow_remote": true,
4     "background_app_mode": 2,
5     "images_mode": 2,
6     "interaction_box_auto": true,
7     "power_saving_adapter": true,
8     "robust_mode_enabled": false,
9     "tracking_tool_enabled": true
10  }
11 }
```

保存，退出，重新启动 LeapMotion 服务：

```
1 sudo launchctl load /Library/LaunchDaemons/com.leapmotion.leapd.plist
```

5.1.2 配置 watchOS 的网络访问

5.2 服务端编码

实现一个端口为 10086 的 HTTP 服务器是第一步：

```
1 var http = require('http');
2 function handler (req, res) {
3     res.writeHead(200);
4 }
5 http.createServer(handler).listen(10086);
6 console.log("Server running at http://localhost:10086")
```

下面我们来关注服务端中对 LeapMotion 手势的关键处理。

5.2.1 Tap 手势识别

5.2.2 Swipe 手势识别

5.3 手表端编码

第六章 用户调研与测试

6.1 用户调研

6.2 测试方法

第七章 进一步工作

7.1 缺陷

讲解技术上的困难。

7.2 改进方向

讲解可能的改进方向。

参考文献

- [1] Apple Inc. App Programming Guide for watchOS [EB/OL]. [2016-04-02]. <https://developer.apple.com/library/watchos/documentation/General/Conceptual/WatchKitProgrammingGuide/>.
- [2] Leap Motion Inc. Leap Python SDK Documentation [EB/OL]. [2016-03-17]. <https://developer.leapmotion.com/documentation/python/index.html>.
- [3] Bi X, Li Y, Zhai S. FFitts Law: Modeling Finger Touch with Fitts' Law [C/OL]. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York, NY, USA, 2013: 1363–1372. <http://doi.acm.org/10.1145/2470654.2466180>.
- [4] Yang Y, Chae S, Shim J, et al. EMG Sensor-based Two-Hand Smart Watch Interaction [C/OL]. In Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology. New York, NY, USA, 2015: 73–74. <http://doi.acm.org/10.1145/2815585.2815724>.
- [5] Vatavu R-D, Zaiti I-A. Leap Gestures for TV: Insights from an Elicitation Study [C/OL]. In Proceedings of the 2014 ACM International Conference on Interactive Experiences for TV and Online Video. New York, NY, USA, 2014: 131–138. <http://doi.acm.org/10.1145/2602299.2602316>.
- [6] Perrault S T, Lecolinet E, Eagan J, et al. Watchit: Simple Gestures and Eyes-free Interaction for Wrist-watches and Bracelets [C/OL]. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York, NY, USA, 2013: 1451–1460. <http://doi.acm.org/10.1145/2470654.2466192>.
- [7] Yang F, Li S, Huang R, et al. MagicWatch: Interacting & Segueing [C/OL]. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication. New York, NY, USA, 2014: 315–318. <http://doi.acm.org/10.1145/2638728.2638848>.
- [8] Chen X A, Grossman T, Wigdor D J, et al. Duet: Exploring Joint Interactions on a Smart Phone and a Smart Watch [C/OL]. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York, NY, USA, 2014: 159–168. <http://doi.acm.org/10.1145/2556288.2556955>.
- [9] Ogata M, Imai M. SkinWatch: Skin Gesture Interaction for Smart Watch [C/OL]. In Proceedings of the 6th Augmented Human International Conference. New York, NY, USA, 2015: 21–24. <http://doi.acm.org/10.1145/2735711.2735830>.
- [10] Knibbe J, Martinez Plasencia D, Bainbridge C, et al. Extending Interaction for Smart Watches: Enabling Bimanual Around Device Control [C/OL]. In CHI '14 Extended Abstracts on Human Factors in Computing Systems. New York, NY, USA, 2014: 1891–1896. <http://doi.acm.org/10.1145/2559206.2581315>.
- [11] Loclair C, Gustafson S, Baudisch P. PinchWatch: a wearable device for one-handed microinteractions [C]. In MobileHCI Workshop on Ensembles of On-Body Devices. 2010.
- [12] Kim J, He J, Lyons K, et al. The gesture watch: A wireless contact-free gesture based wrist interface [C]. In Wearable Computers, 2007 11th IEEE International Symposium on. 2007: 15–22.

- [13] Bailly G, Müller J, Rohs M, et al. ShoeSense: A New Perspective on Gestural Interaction and Wearable Applications [C/OL]. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York, NY, USA, 2012: 1239–1248. <http://doi.acm.org/10.1145/2207676.2208576>.
- [14] Kerber F, Schardt P, Löchtefeld M. WristRotate: A Personalized Motion Gesture Delimiter for Wrist-worn Devices [C/OL]. In Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia. New York, NY, USA, 2015: 218–222. <http://doi.acm.org/10.1145/2836041.2836063>.
- [15] Kratz S, Rohs M. Hoverflow: Exploring Around-device Interaction with IR Distance Sensors [C/OL]. In Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services. New York, NY, USA, 2009: 42:1–42:4. <http://doi.acm.org/10.1145/1613858.1613912>.
- [16] Lv Z, Feng S, Feng L, et al. Extending touch-less interaction on vision based wearable device [C]. In Virtual Reality (VR), 2015 IEEE. 2015: 231–232.
- [17] Vogel D, Baudisch P. Shift: A Technique for Operating Pen-based Interfaces Using Touch [C/OL]. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York, NY, USA, 2007: 657–666. <http://doi.acm.org/10.1145/1240624.1240727>.
- [18] 侯文君, 吴春京. 基于数据分析的智能手表手势直觉化交互研究 [J]. 包装工程. 2015, 36 (22): 13–16.
- [19] 程时伟. 基于上下文感知的移动设备自适应用户界面设计研究 [D]. [S. l.]: 浙江大学, 2009.
- [20] 扶爱名. 基于可穿戴计算的设备维护辅助系统自适应用户界面研究 [D]. [S. l.]: 电子科技大学, 2006.
- [21] 傅泉俊. 穿戴式体感交互平台的研究与实现 [D]. [S. l.]: 电子科技大学, 2015.
- [22] 朱子健. 情境因素对智能手表使用行为影响研究 [J]. 中国新技术新产品. 2015 (7): 10–12.
- [23] 刘珩. 智能手表交互设计研究 [J]. 科技与创新. 2015 (8): 73–73.
- [24] 孙效华, 冯泽西. 可穿戴设备交互设计研究 [J]. 装饰. 2014 (2): 28–33.
- [25] 寇利敏, 常交法, 魏玲. 智能可穿戴设备专利分析 [J]. 电声技术. 2015 (39): 7–9.
- [26] 董士海. 人机交互的进展及面临的挑战 [J]. 计算机辅助设计与图形学学报. 2004, 16 (1): 1–13.
- [27] 岳玮宁, 董士海, 王悦, et al. 普适计算的人机交互框架研究 [J]. 计算机学报. 2004, 27 (12): 1657–1664.
- [28] Hudson S E, Mankoff J. Concepts, Values, and Methods for Technical Human–Computer Interaction Research [M] // Judith S Olson W A K. Ways of Knowing in HCI. Springer, 2014: 2014: 69–93.
- [29] Billingham M, Buxton B. Gesture based interaction [M] // Microsoft. Haptic input. 2011, 2011:.
- [30] Jiang H, Chen X, Zhang S, et al. Software for Wearable Devices: Challenges and Opportunities [J/OL]. CoRR. 2015, abs/1504.00747. <http://arxiv.org/abs/1504.00747>.
- [31] 李清水, 方志刚, 沈模卫, et al. 手势识别技术及其在人机交互中的应用 [J]. 人类工效学. 2002, 8 (1): 27–29.

- [32] 陈雅茜, 欧长坤, 郭阳. 基于单目视觉和简单手势的空间交互技术 [J]. 西南民族大学学报: 自然科学版. 2014, 40 (6): 871–876.
- [33] 刘平和. 基于手指的人机交互系统研究与实现 [D]. [S. l.]: 西北师范大学, 2014.
- [34] 狄海进. 基于三维视觉的手势跟踪及人机交互中的应用 [D]. [S. l.]: 南京大学, 2011.
- [35] Wobbrock J O, Wilson A D, Li Y. Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes [C]. In Proceedings of the 20th annual ACM symposium on User interface software and technology. 2007: 159–168.
- [36] Anthony L, Wobbrock J O. A lightweight multistroke recognizer for user interface prototypes [C]. In Proceedings of Graphics Interface 2010. 2010: 245–252.
- [37] 徐崇斌, 周明全, 沈俊辰, et al. 一种基于 Leap Motion 的直观体交互技术 [J]. 电子与信息学报. 2015, 37 (2): 353–359.
- [38] 潘佳佳, 徐昆. 基于 Leap Motion 的三维自由手势操作 [J]. 中国科技论文. 2015 (2): 207–212.
- [39] 胡弘, 晁建刚, 杨进, et al. Leap Motion 关键点模型手姿态估计方法 [J]. 计算机辅助设计与图形学学报. 2015, 27 (7): 1211–1216.
- [40] Weichert F, Bachmann D, Rudak B, et al. Analysis of the Accuracy and Robustness of the Leap Motion Controller [J]. Sensors. 2013, 13: 6380–6393.
- [41] Marin G, Dominio F, Zanuttigh P. Hand gesture recognition with leap motion and kinect devices [C]. In Image Processing (ICIP), 2014 IEEE International Conference on. 2014: 1565–1569.
- [42] Garber L. Gestural technology: Moving interfaces in a new direction [technology news] [J]. Computer. 2013, 46 (10): 22–25.
- [43] Zaiti I-A, Pentiu Ş-G, Vatau R-D. On free-hand TV control: experimental results on user-elicited gestures with Leap Motion [J]. Personal and Ubiquitous Computing. 2015, 19 (5-6): 821–838.
- [44] Du G, Zhang P, Liu X. Markerless Human-Manipulator Interface Using Leap Motion with Interval Kalman Filter and Improved Particle Filter [J]. IEEE Transactions on Industrial Informatics. 2016, PP (99): 1–1.
- [45] Huxpro. Forcify [EB/OL]. [2016-03-16]. <https://github.com/huxpro/forcify>.
- [46] Apple Inc. Apple Watch Human Interface Guidelines [EB/OL]. [2016-03-17]. <https://developer.apple.com/watch/human-interface-guidelines/>.
- [47] Apple Inc. Watch Connectivity Framework Reference [EB/OL]. [2016-04-02]. https://developer.apple.com/library/watchos/documentation/WatchConnectivity/Reference/WatchConnectivity_framework/.
- [48] Apple Inc. The Swift Programming Language(Swift 2.2 Prerelease) [M/OL]. Apple Inc., 2016. <https://itunes.apple.com/cn/book/swift-programming-language/id1002622538>.

- [49] Apple Inc. Using Swift with Cocoa and Objective-C(Swift 2.2 Prerelease) [M/OL]. Apple Inc., 2016.
<https://itunes.apple.com/cn/book/using-swift-cocoa-objective/id1002624212>.

致谢

感谢亲人感谢朋友。