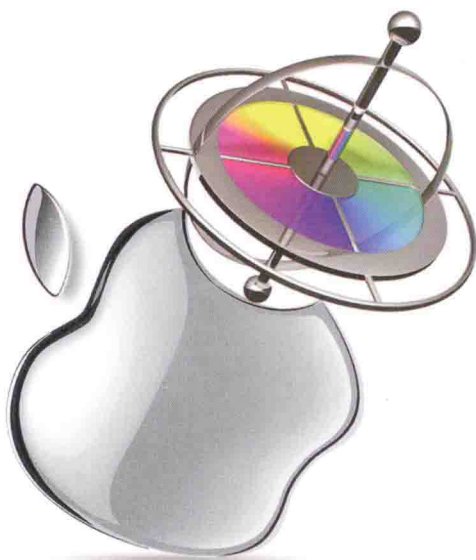


著名移动开发专家关东升倾力创作的iOS 7专业开发领衔之作，全面涵盖iOS 7传感器应用开发原理与项目实战的里程碑作品！

国内首本系统论述iOS 7传感器应用开发的专著！

清华

开发者书库



iOS Developing Insights: Sensors

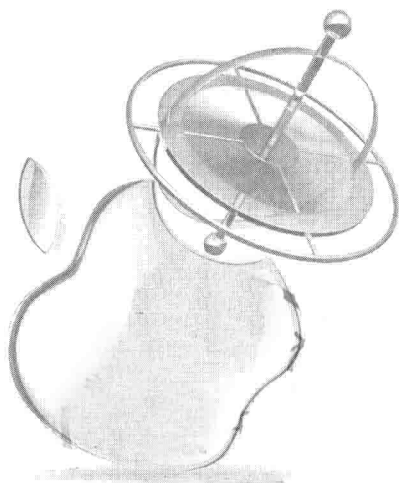
iOS传感器应用开发 最佳实践

关东升◎著

Guan Dongsheng

清华大学出版社





iOS Developing Insights: Sensors

iOS传感器应用开发 最佳实践

关东升◎著

Guan Dongsheng

清华大学出版社

内 容 简 介

本书采用基于 iOS 7 最新的 API,系统介绍了 iOS 设备主要的传感器应用开发的相关知识。本书共分为 9 章,分别从编程原理与开发实践两个角度论述了手势识别、加速度计、陀螺仪、指南针、磁力计、照相机、摄像头、音频设备访问、定位和蓝牙通信,以及 iOS 敏捷开发项目实战——增强现实版本铅锤应用等。

本书适合从事 iOS 开发,尤其是传感器应用开发的读者阅读使用。为便于读者快速动手实践,本书配套提供了所有实例的源代码。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

iOS 传感器应用开发最佳实践/关东升著.--北京:清华大学出版社,2014

清华开发者书库

ISBN 978-7-302-35115-3

I. ①i… II. ①关… III. ①传感器—应用开发 IV. ①TP212

中国版本图书馆 CIP 数据核字(2014)第 009222 号

责任编辑:盛东亮

封面设计:李召霞

责任校对:焦丽丽

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:186mm×240mm

印 张:14.25

字 数:321 千字

版 次:2014 年 3 月第 1 版

印 次:2014 年 3 月第 1 次印刷

印 数:1~2500

定 价:59.00 元

产品编号:056808-01

前言

FOREWORD

在 iOS 应用开发中有很多涉及硬件访问和传感器编程的知识,本书将全面介绍苹果 iOS 设备主要的传感器应用开发。本书是智捷 iOS 课堂团队编写的 iOS 系列丛书的一本,目的是使一个有 iOS 开发基础的程序员通过本书的学习,能够开发 iOS 硬件访问和传感器应用的程序。另外,对于没有 iOS 开发基础的读者,推荐先阅读我们已经出版的《iOS 开发指南——从零基础到 App Store 上线》(网站: <http://www.iosbook1.com>),这两本书都是我们编写的,知识体系衔接得很好。

经过几个月的努力,我们终于在 2014 年 1 月之前完成初稿,几个月来,智捷 iOS 课堂团队夜以继日,几乎推掉一切社交活动,我也推掉很多企业邀请去讲课的机会,每天工作 12 小时,不敢有任何的松懈,不敢有任何的模棱两可,只做一件事情——编写此书。每一张文字、每一张图片、每一个实例都是我们的呕心之作。

本书没有包括多媒体和游戏等知识,我们会在另外两本介绍 iOS 多媒体开发和游戏开发的书中介绍,具体进展请关注智捷 iOS 课堂官方网站 <http://www.51work6.com>。另外,大家如果对 iOS 网络和云端编程感兴趣,可以关注我们此前与清华大学出版社合作出版的《iOS 网络编程与云端应用最佳实践》(网站: <http://www.iosbook3.com>)。

关于本书网站

为了更好地为广大读者提供服务,在智捷 iOS 研究中心,我们专门为本书建立了一个网页 <http://www.51work6.cn/>,读者可以查看相关出版进度,并对书中内容发表评论,提出宝贵意见。

关于源代码

本书中包括了 100 多个完整的案例项目源代码,全部采用最新的 iOS 7 API 和操作界面,大家可以到本书网页下载。

勘误与支持

我们在网站 <http://www.51work6.cn/> 中建立了一个勘误专区,及时地把书中的问题、失误和纠正反馈给广大读者,如果您发现有什么问题,可以在网上留言,也可以发送电子邮件到 eorient@sina.com,我们会在第一时间回复您。您也可以在新浪微博中与我们

联系: @tony_关东升。

全书由关东升、董玉萍编写。感谢清华大学出版社的盛东亮责编给我们提供了宝贵的意见。感谢智捷 iOS 课堂团队的贾云龙参与内容讨论和审核,感谢赵大羽老师手绘了书中全部草图,并从专业的角度修改书中图片,力求更加真实、完美地奉献给广大读者。感谢我的家人容忍我的忙碌,以及对我的关心和照顾,使我能抽出这么多时间,投入全部精力专心编写此书。

由于时间仓促,书中难免存在不妥之处,请读者谅解,并提出宝贵意见。



2014 年 1 月于北京

目录

CONTENTS

第 1 章 开篇综述.....	1
1.1 本书结构	1
1.2 本书中的约定	1
1.2.1 实例代码约定.....	2
1.2.2 图示的约定.....	3
第 2 章 手势识别.....	6
2.1 手势种类	6
2.2 使用手势识别器	7
2.2.1 视图对象与手势识别.....	7
2.2.2 手势识别状态.....	8
2.2.3 检测 Tap	9
2.2.4 检测 Long Press	19
2.2.5 检测 Pan	24
2.2.6 检测 Swipe	28
2.2.7 检测 Rotation	32
2.2.8 检测 Pinch	36
2.3 触摸事件与手势识别.....	40
2.3.1 事件处理机制	40
2.3.2 响应者对象与响应链	41
2.3.3 触摸事件	42
2.3.4 手势识别	48
本章小结	55
第 3 章 加速度计与陀螺仪	56
3.1 加速度计.....	57
3.1.1 访问加速度计	57

3.1.2	实例：通过 Accelerometer 访问加速度计	58
3.1.3	实例：通过 CMMotionManager 访问加速度计	60
3.1.4	数据的“滤波”	63
3.1.5	实例：数据的“滤波”	65
3.1.6	感知设备方向	67
3.1.7	实例：感知设备方向	69
3.2	陀螺仪	73
3.2.1	访问陀螺仪	74
3.2.2	检测晃动设备	77
3.2.3	实例：检测晃动设备	78
	本章小结	80
第4章	指南针与磁力计	81
4.1	访问磁力计	81
4.1.1	访问导航方面的 API	82
4.1.2	磁力计与设备方向	82
4.1.3	实例：指南针应用	84
4.2	检测磁场	89
4.2.1	检测磁场 API	90
4.2.2	实例：特斯拉计应用	90
	本章小结	95
第5章	照相机与摄像头	96
5.1	iOS 设备摄像头	97
5.2	图像抓取	97
5.2.1	图像选择器	97
5.2.2	实例：照相机抓取图片	98
5.2.3	编辑图片	103
5.2.4	保存图片	104
5.2.5	添加照相机图层	106
5.3	视频捕获	109
5.3.1	实例：视频捕获	109
5.3.2	使用 UIImagePickerController 捕获	109
5.3.3	使用 AVFoundation 捕获	112
	本章小结	116

第 6 章 音频设备访问	117
6.1 音频 API 介绍	117
6.2 AVFoundation 框架实现音频录制与播放	118
6.2.1 音频播放	118
6.2.2 实例：音频播放	119
6.2.3 音频录制	122
6.2.4 实例：音频录制	123
本章小结	129
第 7 章 使用蓝牙	130
7.1 Game Kit 应用开发	130
7.1.1 对等结构网络	130
7.1.2 Game Kit API	131
7.1.3 实例：基于蓝牙对等网络通信	132
7.2 低功耗蓝牙	137
7.2.1 Core Bluetooth API	137
7.2.2 中心角色	137
7.2.3 外设角色	141
7.3 实例：低功耗蓝牙通信	144
7.3.1 中心角色编程实现	145
7.3.2 外设角色编程实现	152
7.4 使用 SensorTag 开发蓝牙低功耗应用	157
7.4.1 SensorTag 开发套件	158
7.4.2 实例：我的温度计	160
本章小结	171
第 8 章 定位	172
8.1 获得用户的位置	172
8.1.1 定位服务编程	173
8.1.2 实例：定位服务	174
8.1.3 距离过滤器和精度	175
8.1.4 启动和停止定位服务	176
8.1.5 获得位置信息	177
8.1.6 关于定位服务的测试	178
8.2 地理信息编码与反编码	181

8.2.1	地理信息反编码·····	181
8.2.2	实例：地理信息反编码·····	182
8.2.3	地理信息编码查询·····	184
8.2.4	实例：地理信息编码查询·····	184
8.3	微定位技术·····	186
8.3.1	地理围栏·····	186
8.3.2	iOS 7 中的 iBeacon 技术·····	186
8.3.3	实例：iBeacon 技术实现微定位·····	187
8.4	接近传感器·····	196
	本章小结·····	198
第 9 章	iOS 敏捷开发项目实战——增强现实版本铅锤应用 ·····	199
9.1	应用分析与设计·····	199
9.1.1	应用概述·····	199
9.1.2	需求分析·····	199
9.1.3	原型设计·····	201
9.2	iOS 敏捷开发·····	201
9.2.1	敏捷开发宣言·····	201
9.2.2	iOS 可以敏捷开发·····	202
9.2.3	iOS 敏捷开发一般过程·····	202
9.3	任务 1：创建应用并初始化工程·····	204
9.3.1	任务 1.1：创建 PlumbBob 工程·····	204
9.3.2	任务 1.2：设置状态栏·····	204
9.3.3	任务 1.3：设置设备支持方向·····	208
9.3.4	任务 1.4：添加铅锤图片·····	208
9.4	任务 2：铅锤功能·····	209
9.4.1	任务 2.1：添加铅锤 UIImageView 到主视图·····	209
9.4.2	任务 2.2：通过加速度计传感器控制铅锤·····	211
9.5	任务 3：增强现实功能·····	214
9.6	任务 4：关于我们·····	216
	本章小结·····	218

移动设备与 PC 的不同之处有很多,其中移动设备有很多传感器,这些传感器可以开发很多有意思的应用,可以使用麦克风获取音频,开发出一个测量风速的应用;也可以使用 GPS 定位开发出导航应用。本书将向读者介绍苹果 iOS 设备主要的传感器应用开发。

1.1 本书结构

本书向广大读者介绍苹果 iOS 设备主要的传感器应用开发的相关知识,其中包括手势识别、加速度计、陀螺仪、指南针、磁力计、照相机、摄像头、音频设备访问、定位和蓝牙通信等。

全书包括如下内容:

- iOS 中的手势识别,包括手势识别的种类、如何使用手势识别器以及手势识别与事件触发。
- 加速度计与陀螺仪的技术,涵盖加速度计与陀螺仪的访问与设备感知、设备检测。
- 指南针与磁力计,介绍了访问磁力计的 API 与检测磁场的 API,并通过指南针应用和特斯拉计应用详细讲解这两个传感器的使用。
- 照相机和摄像头,包括照片的捕捉与处理以及视频捕获的两种方式。
- 音频设备访问,首先介绍了音频设备,然后详细介绍音频 API,最后介绍音频的播放与录制。
- 定位,包括定位的 API、距离过滤器、精度定位以及地理信息编码与反编码。
- 蓝牙通信,包括对等结构网络和 Game Kit 框架以及低功耗蓝牙技术。
- 增强现实版的铅锤应用,总体介绍 iOS 传感器实战开发技巧以及 iOS 项目敏捷开发过程。

1.2 本书中的约定

为了方便读者使用本书,首先介绍本书中实例代码和图示的相关约定。

1.2.1 实例代码约定

本书作为一本介绍编程方面的书,书中有很多实例代码,下载本书代码并解压代码,会看到下面的目录结构:

```
├── ch02
│   ├── 2.2.3
│   │   ├── TapGestureRecognizer - 1
│   │   └── TapGestureRecognizer - 2
│   ├── 2.2.4
│   │   ├── LongPressGestureRecognizer - 1
│   │   └── LongPressGestureRecognizer - 2
│   ├── 2.2.5
│   │   ├── PanGestureRecognizer - 1
│   │   └── PanGestureRecognizer - 2
│   ├── 2.2.6
│   │   ├── SwipeGestureRecognizer - 1
│   │   └── SwipeGestureRecognizer - 2
│   ├── 2.2.7
│   │   ├── RotationGestureRecognizer - 1
│   │   └── RotationGestureRecognizer - 2
│   ├── 2.2.8
│   │   ├── PinchGestureRecognizer - 1
│   │   └── PinchGestureRecognizer - 2
│   ├── 2.3.3 EventInfo
│   │   └── EventInfo
│   └── 2.3.4
│       ├── PinchGestureRecognizer
│       └── TapGestureRecognizer
├── ch03
│   ├── 3.1.2 Accelerometer
│   ├── 3.1.3 MotionManager
│   ├── 3.1.5 Accelerometer
│   ├── 3.1.7 DeviceOrientation
│   ├── 3.2.1 Gyroscope
│   └── 3.2.3 MotionShake
├── ch04
│   ├── 4.1.3 Compass
│   └── 4.2.2
│       ├── Teslameter - 1
│       └── Teslameter - 2
├── ch05
│   ├── 5.2
│   │   └── 5.2.2 TakePicture
```

```

|   |   |--- 5.2.3 TakePicture
|   |   |--- 5.2.4 TakePicture
|   |   |--- 5.2.5 TakePicture
|   |   |--- 5.3
|   |   |--- 5.3.2 Ballon
|   |   |--- 5.3.3 Ballon
|--- ch06
|   |--- 6.2.2 MusicPlayer
|   |--- 6.2.4 AudioRecorder
|--- ch07
|   |--- 7.1 P2PGame
|   |--- 7.3
|   |   |--- BLECentral
|   |   |--- BLEPeripheral
|   |--- 7.4 Ambient Temperature
|--- ch08
|   |--- 8.1 WhereAmI
|   |--- 8.2.2 WhereAmI
|   |--- 8.2.4 WhereAmI
|   |--- 8.3
|   |   |--- APPClient
|   |   |--- BeaconDevice
|   |--- 8.4 ProximitySensorSample
|--- ch09
|   |--- PlumbBob

```

目录 ch02~ch09(代表第 2~9 章)的实例代码或一些资源文件,其中工程或工作空间的命名有如下几种形式:

- 二级目录标号,如:“8.2 P2PGame”说明是第 8 章第 8.2 节中使用的 P2PGame 工程实例。
- 三级目录标号,如:“3.1.2 Accelerometer”说明是第 3 章第 3.1.2 节中使用的 Accelerometer 工程实例。
- 没有标号情况下,由所在父目录说明是哪个章节的实例工程,如:PlumbBob 说明是在第 9 章中使用的。

1.2.2 图示的约定

为了更形象、有效地说明知识点或描述操作,本书添加了很多图示,下面对图示中的一些符号含义进行简要说明。

1. 图中的圈框

有时读者会看到如图 1-1 所示的圈框,圈框中的内容是选中的内容、或重点要说明的内容、需要引起注意的内容。

2. 图中的箭头

如图 1-1~图 1-3 所示,箭头用于说明用户的动作,一般箭尾是动作开始的地方,箭头指向动作结束的地方。如图 1-1 所示箭头代表界面的跳转。如图 1-2 所示使用实线箭头表示从对象库拖动控件到设计视图。如图 1-3 所示使用虚线箭头表示操作。



图 1-1 图中圈框

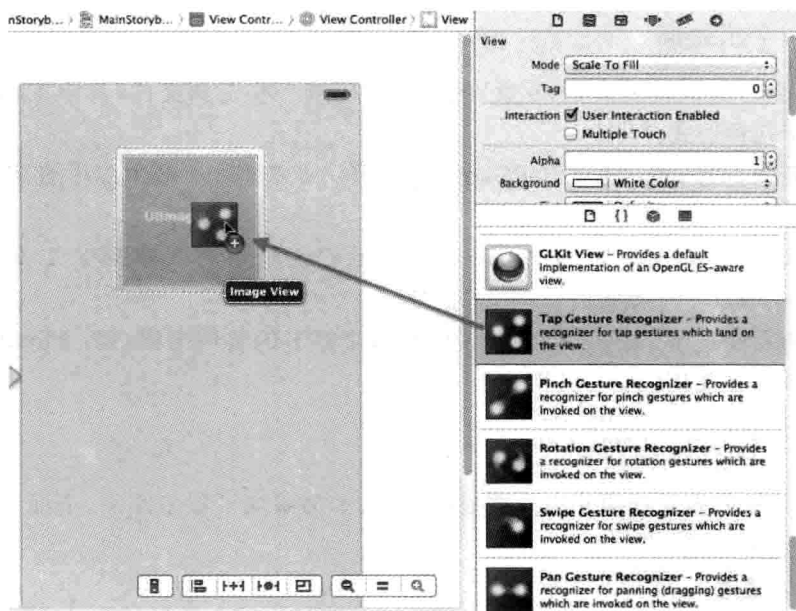


图 1-2 图中箭头 1

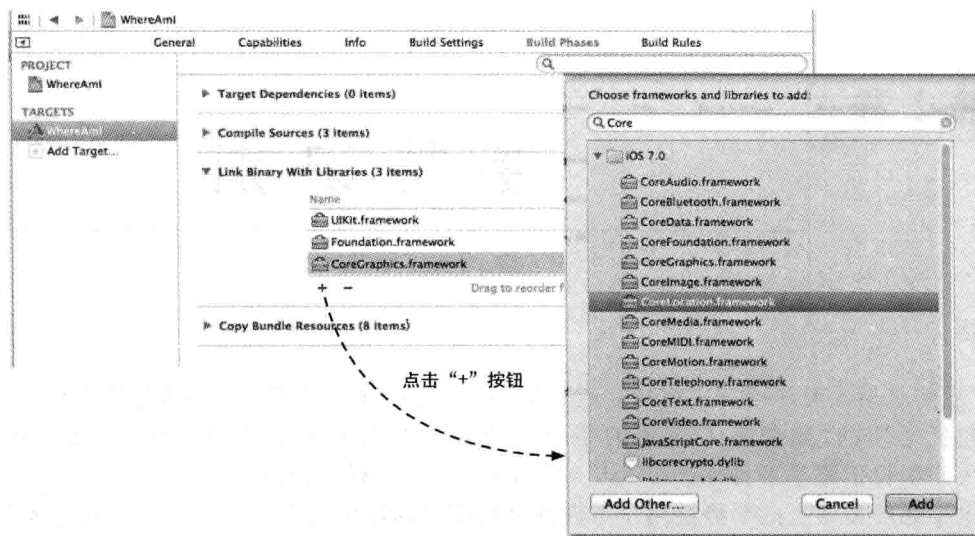


图 1-3 图中箭头 2

上述内容包含本书使用的大部分符号，一些个别符号会在使用的地方进行具体说明。

“手势是指人类用语言中枢建立起来的一套用手掌和手指位置、形状的特定语言系统。其中包括通用的,如聋哑人使用的手语。还有在特定情况下的该种系统,如海军陆战队。”——引自于维基百科(<http://zh.wikipedia.org/wiki/>)手势。

电子触屏设备上的手势是用户与设备进行交流的特定语言。作为设备不但能够识别这些手势,而且要能够为开发人员提供开发接口。

2.1 手势种类

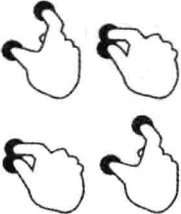
在 iOS 设备上有极其丰富的手势,理论上说手势的种类是没有限制的,可以开发出很多诡异的手势,但是用户是否会、用得不好就不得而知了。因此手势种类一般都是大众比较熟悉的几种。

在 iOS 设备中常用的手势有: Tap(点击)、Long Press(长按)、Pan(拖动)、Swipe(滑动)、Rotation(旋转)和 Pinch(手指的合拢和张开)等。这些手势如表 2-1 所示。

表 2-1 iOS 设备手势

手 势 名	手 势 图	说 明
Tap(点击)		选择、点击、碰触或连续碰触视图对象
Long Press(长按)		长时间按住屏幕上视图对象
Pan(拖动)		拖曳屏幕上的一个视图对象到不同的位置

续表

手 势 名	手 势 图	说 明
Swipe(滑动)		快速拖曳屏幕上的视图对象,然后突然停在
Rotation(旋转)		用两个手指按住屏幕上的视图对象,然后旋转
Pinch(手指的合拢和张开)		多个手指按住屏幕上的视图对象,然后合并或张开

2.2 使用手势识别器

在 iOS 设备上识别手势有两种实现方式:采用手势识别器(UIGestureRecognizer)和采用触摸事件(UITouch)识别。本节介绍采用手势识别器来实现手势识别。

手势识别器类 UIGestureRecognizer 是一个抽象类,它有 6 个具体实现类:

- UITapGestureRecognizer
- UIPinchGestureRecognizer
- UIRotationGestureRecognizer
- UISwipeGestureRecognizer
- UIPanGestureRecognizer
- UILongPressGestureRecognizer

从上面这几个类的命名可以看出与表 2-1 介绍的 6 种手势对应关系,如果这 6 种手势识别器不能满足要求,还可以直接继承 UIGestureRecognizer 实现特殊手势识别。

2.2.1 视图对象与手势识别

手势识别一定是发生在某一个视图对象上的,它可能是常用标签、按钮、图片等视图或者控件。要对视图对象进行手势识别,需要使用下面语句添加手势识别器:

```
[self.view addGestureRecognizer:gestureRecognizer];
```

其中,gestureRecognizer 是具体的手势识别器对象。

此外,针对视图对象还需要设置一些属性,主要有两个属性:

- `userInteractionEnabled`。开启或关闭用户事件。
- `multipleTouchEnabled`。设置是否接收多点触摸事件。

我们可以在程序代码中设置这两个属性,它们的设置通常是在视图控制器的 `viewDidLoad` 方法中完成的,实例代码如下:

```
-(void)viewDidLoad {
    self.view.multipleTouchEnabled = NO;
    self.view.userInteractionEnabled = YES;
    ...
}
```

当然也可以在 Interface Builder 中通过设计视图属性实现,在 Interface Builder 中选中要设置的视图对象,如图 2-1 所示,打开属性检查器,在 `View→Interaction` 属性中设置这两个属性。

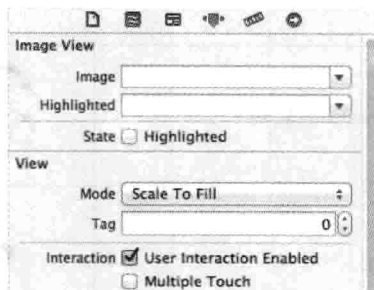


图 2-1 设置视图属性

2.2.2 手势识别状态

`UIGestureRecognizer` 类有一个 `state` 属性,它用来表示手势识别过程中的状态,手势识别的状态分为 7 个,这些状态是通过以下常量来定义的:

- `UIGestureRecognizerStatePossible` 手势尚未识别,它是默认状态。
- `UIGestureRecognizerStateBegan` 开始接收连续类型手势。
- `UIGestureRecognizerStateChanged` 接收连续类型手势状态变化。
- `UIGestureRecognizerStateEnded` 结束接收连续类型手势。
- `UIGestureRecognizerStateCancelled` 取消接收连续类型手势。
- `UIGestureRecognizerStateFailed` 离散类型的手势识别失败。
- `UIGestureRecognizerStateRecognized` 离散类型的手势识别成功。

手势分为连续类型的手势与离散类型的手势。连续类型的手势。如 `Pinch`(手指的合拢和张开),它在整个过程中连续产生多个触摸点,它的识别过程如图 2-2 所示,其中 `Changed` 状态可能会多次变化,最后有 `Ended`(结束)和 `Cancelled`(取消)两种状态。离散类型手势,只发生一次,如: `Tap`(点击)手势,如图 2-3 所示,识别过程只有两种状态:识别(`Recognized`)和失败(`Failed`)。

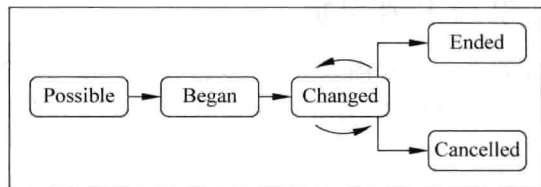


图 2-2 连续类型手势识别过程