

# 一种智能手机上下文信息获取的代价模型及其应用

谌国风 孔俊俊 郭 耀 陈向群

(高可信软件技术教育部重点实验室 北京大学信息科学技术学院软件所 北京 100871)

**摘 要** 随着信息技术的发展和应用需求的增长,智能手机中嵌入了各种传感器和网络接口,它们是获取上下文信息,进而构建智能型移动应用的关键。尽管在智能手机中获取上下文信息的代价很大,但是这种代价却通常被移动应用开发者所忽视。提出了一个上下文信息获取的代价模型,它可对上下文信息获取的代价进行度量。设计并实现了一个上下文信息获取代价模型的测量工具 CRCTest,并对 Android 智能手机的代价模型进行了测量。基于测量得到的 Android 平台上下文信息获取的代价模型实现了一个应用实例,通过对比两种位置上下文信息生成方式的代价,说明了基于代价模型优化设计上下文获取的可行性。

**关键词** 智能手机,传感器,上下文信息获取,代价模型

**中图法分类号** TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.11.026

## Context Retrieval Cost Model on Smartphones and its Application

SHEN Guo-feng KONG Jun-jun GUO Yao CHEN Xiang-qun

(Key Laboratory of High Confidence Software Technologies (Ministry of Education) Institute of Software,  
School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

**Abstract** With the development of technology and increase of application requirements, a lot of sensors and network interfaces have been embedded in smartphones, which are a key contributor in acquiring context information and building smart mobile applications. Although the context retrieval cost on smartphone is significant, it is usually ignored by mobile application developers. This paper proposed a context retrieval cost model to analyze the cost of context information retrieval. We designed and implemented a context-retrieval-cost measuring tool, CRCTest, and measured the context retrieval cost model on an Android smartphone. Based on the measured cost model, we implemented an example application, and conducted some experiments to compare the cost of two approaches to acquiring location context. The result shows that it is feasible to optimize the context retrieval cost with the proposed context retrieval cost model.

**Keywords** Smartphone, Sensor, Context retrieval, Cost model

## 1 引言

在智能手机的发展过程中,为了支持更多的功能,嵌入到智能手机的传感器种类日益丰富;为了增强用户体验,智能手机传感器获取数据的频率和精度不断提高。这些都导致移动应用对于智能手机电池电量的消耗呈显著上升的趋势。然而,手机电池技术却没有取得太大的突破。在过去的功能手机时代,人们大约一星期为手机充电一次;而现在的智能手机时代,手机用户则平均每天都为手机充电一次。智能手机上更多、更高性能的硬件配置所带来的日益增长的能耗需求与增长缓慢的手机电池容量之间的矛盾,成为了目前智能手机发展所遇到的重要问题之一。

上下文信息(context)指与位置、时间、用户、物理环境、设备等相关的信息。在智能手机上使用非常广泛的两类信息是时间和位置(空间)。应用程序基于这些上下文信息来优化和

自适应自己的行为,更好地甚至主动地为用户提供服务,例如:基于位置的服务(LBS)。上下文信息通常是基于传感器产生的数据进行计算而获得的。我们把直接从传感器输出的值称为原始数据,而经过加工处理后获得的有语义的信息就是上下文信息。从传感器获取原始数据,然后使用特定算法处理原始数据而获得上下文信息的过程一般称为上下文信息获取。

上下文信息获取会造成较高的能耗。一方面,上下文信息获取需要访问传感器硬件,而使用这些硬件传感部件会导致手机功耗增加,特别是有些传感器硬件的功耗非常高,例如:摄像头和GPS等;另一方面,上下文信息获取需要持续性频繁访问传感器,从而导致高能耗。

针对智能手机上下文信息获取的高能耗问题,已经有研究工作提出了一些能耗优化的技术。已有的这类研究工作在度量智能手机上下文信息获取代价时一般仅考虑功耗开销,

到稿日期:2013-09-16 返修日期:2013-11-13 本文受国家自然科学基金(61121063, 61103026),国家重点基础研究发展计划(973)项目(2011CB302604),国家高技术研究发展计划(2011AA01A202)资助。

谌国风(1993-),男,硕士生,主要研究领域为物联网、普适计算,E-mail: shengf13@sei.pku.edu.cn;孔俊俊 男,博士生主要研究领域为物联网、普适计算;郭 耀 男,副教授,主要研究领域为操作系统、移动云计算;陈向群 女,教授,主要研究领域为操作系统、移动云计算。

而未考虑时间开销。本文提出的上下文信息获取的代价模型将这两种开销都考虑在内。

- 时间开销:即开始获取上下文信息到最终返回结果数据的时间延迟;

- 功耗开销:即获取上下文信息而导致的系统功耗增量。

相同的上下文信息可以用不同的算法使用不同的传感器原始数据生成,而不同的算法可能会导致差异明显的代价。例如,就位置信息而言,它既可以通过 GPS 定位获得,也可以通过蜂窝网络数据(GSM)、Wi-Fi 定位等方式获得。这些不同的获得途径导致的代价和上下文信息的精度存在着差别。假如应用请求位置信息,在满足应用要求的条件下,选择代价最小的实现方式来获取上下文信息并返回,就能够从整体上减小能耗开销从而延长手机的使用时间。事实上,相当多的应用并不需要或者并不能很好地利用它们所请求的上下文信息的精度和频率,一个仅需要十分钟才更新一次位置信息的应用可能每隔数秒就请求获得最新的位置信息,这样就造成了相当显著的能耗开销。为此,需要一个能够度量上下文信息获取代价的应用框架来提供上下文信息获取的低能耗的 API。

本文提出了一个上下文信息获取的代价模型及其测量方法,在 Android 智能手机上实现了一个代价模型的测量工具 CRCTest,并且使用 CRCTest 工具测量得出了 Google Nexus S 手机的上下文信息获取代价模型,最后通过一个应用实例说明了提出的上下文信息获取代价模型的可行性和应用价值。

## 2 相关工作

智能手机促进了移动传感技术的发展和應用。智能手机一般内嵌了丰富的传感器,移动应用可以借助这些传感器确定用户的状态、位置和环境信息等,从而增强用户体验。但是,移动传感,尤其是连续性的移动传感,会显著地增加系统能耗,例如 Lane 等人在文献[1]中给出的数据表明:连续性移动传感应用会导致手机待机时间(standby time)从 20 个小时左右下降到 6 个小时。移动传感应用的逐渐流行促使移动传感的能耗优化技术成为新的关注点。本文以智能手机平台传感 API(应用程序编程接口)的访问代价模型为基础来优化移动传感应用。

移动传感技术受到学术界的广泛关注。微软研究院的 Chu 等人甚至提出了基于传感器和上下文感知(context-awareness)的操作系统 CondOS<sup>[2]</sup>,使得上下文信息不但服务于应用程序而且用于优化操作系统服务。另有不少工作提出了基于智能手机的移动传感框架,把智能手机作为一个具有丰富传感能力的传感器来使用,例如:Funf<sup>[1]</sup> 开放传感框架、ODK 传感器集成框架<sup>[3]</sup>、Pogo 移动传感中间件<sup>[4]</sup>等。Lane 等人在文献[1]中讨论了智能手机移动传感的架构和机制,提出一个完整的智能手机传感模型的体系结构应当包含 3 个部分:感知部分、学习部分、通知和分享部分。该框架将移动传感和云计算结合起来,使得后两个部分既可以发生在智能手机端又可以发生在云计算服务端。

针对移动传感的高能耗问题,许多研究者已经提出了一些能耗优化技术。已有的研究思路和技术可以分为:基于低功耗传感硬件优化和基于软件算法优化。

基于硬件优化的例子有微软研究院的 LittleRock<sup>[5]</sup>,而更多的是基于软件算法优化的研究,如:SenseLess<sup>[6]</sup>、ACE<sup>[7]</sup>、QoINF<sup>[8]</sup>、GreenDroid<sup>[9]</sup>等。LittleRock 使用一种专用的低功耗硬件来卸载移动传感甚至传感数据处理,使得手机的主 CPU 有更多的时间处于睡眠状态。SenseLess 的基本思想是利用不同传感器的功耗特征来减小移动传感应用的能耗,通过使用低代价的传感器来替代或者减少高代价传感器的使用。ACE(Acquisitional Context Engine)是一个能够减少持续获取上下文信息代价的中间件。上下文感知应用可以向 ACE 直接请求所需求的上下文信息而不必关心如何获得,ACE 会以最优方式获得目标上下文信息的结果并返回给应用。ACE 实现能耗优化的主要技术是上下文信息(context)缓存和基于规则的推导。基于一些规则,可以通过低代价的上下文推导出目标上下文。文献[8]则基于多模态传感技术,提出了优化选择传感器集合的算法,使得用算法选择的传感器集生成上下文的代价更小。GreenDroid 虽然没有提出直接的移动传感优化技术,但是它能够自动地发现已有移动传感应用中由于传感器的不合理使用而导致的能耗问题。

本文的研究工作基于 SenseLess 的思想,但相对于 SenseLess 改进的地方有:提出并且度量了一个完整的 Android 传感 API 访问代价模型。此外,GreenDroid 也指出了现有的移动传感应用中存在的能耗问题与传感 API 的不当使用相关。因此,本文提出的以传感 API 访问代价为核心的代价模型不仅能够用于优化上下文生成算法,而且可以指导移动传感应用开发者更加合理地使用传感 API。

## 3 上下文信息获取的代价模型

上下文信息获取的过程中需要以传感器输出的原始数据作为输入执行特定的算法。执行算法所消耗的时间与输入参数和算法本身的复杂度相关,而且同一类型的上下文信息可以用多种不同的算法生成,因此,上下文信息生成算法的代价难以评估。但是,上下文信息获取过程中在特定平台上访问传感器的代价是相对确定的。一方面,特定平台上的传感器数量确定,另一方面,访问传感器的 API 导致的开销相对确定。通过相对确定的代价模型可以指导和优化上下文信息生成算法。

基于上述考虑,本文以访问特定平台传感器 API 的开销作为上下文信息获取的基本代价模型来评估移动平台上获取各个传感器数据的时间和能耗(或者功耗)开销,并且提供一个全面且细粒度的度量上下文信息获取的代价模型。该模型由两部分组成:

- E(Sensing API):访问传感器 API 的能耗或者功耗开销,可用毫焦(mJ)或毫瓦(mW)计量;
- T(Sensing API):访问传感器 API 的时延,可以用毫秒(ms)计量。

模型中的 Sensing API 是指具体的平台特定的传感 API(访问传感器的应用程序编程接口),应用程序或上下文信息生成算法可以通过这些 API 访问传感器输出的原始数据。

该代价模型对于上下文信息获取的意义在于:上下文信息生成算法的设计者可以明确地了解使用底层传感器 API

<sup>1)</sup> <http://www.funf.org/index.html>

的代价,进而从时间和能耗两个方面进行优化设计,最终能够从整体降低上下文信息获取的开销。

4 代价模型的测量工具

为了测量代价模型(第3节)中提出的 E(Sensing API)和 T(Sensing API),需要一个测量工具。该测量工具能够度量出特定移动平台中上下文信息获取的代价模型。我们选择目前最为流行的智能手机操作平台 Android 来设计并且实现了这样一个测量工具 CRCTest。其主要考虑的因素有:1)传感 API 的范围,即主要考虑哪些类型的传感器及其访问接口;2)测量访问传感 API 的时延和能耗(或功耗)的方法。

4.1 传感器和传感接口

在 Android 平台上获取上下文信息主要是通过查询各类内置的传感器和网络接口的输出值以及相关状态来实现的。Android 系统 API 支持的传感器类型已经超过 10 种,它们是上下文信息的重要来源。同时,使用网络接口也可以获得一些非常有用的上下文信息,例如访问 GPS 模块就可以获得位置信息,Wi-Fi 可以用于室内定位,而 RFID 和 NFC 可以用于获取物体或者用户的身份信息。

狭义上的传感器指 Android 系统 API 定义的传感器,包括加速度传感器、光线传感器等。在广义上,一切可以接收和获得外界信息的硬件设备都可以被称为传感器,因此智能手机中的相机、GPS 定位仪、无线网卡、麦克风、触摸屏等设备都可以被计入到传感器的范围之内。表 1 列举出了 Android 平台中主要的传感器类型和网络接口。

表 1 Android 平台主要的传感器和网络接口

| 传感器                 | 网络接口           |
|---------------------|----------------|
| 加速度 (Accelerometer) | Wi-Fi(WLAN)    |
| 磁场 (Magnetic field) | 蓝牙 (Bluetooth) |
| 方向 (Orientation)    | GSM,3G         |
| 螺旋仪 (Gyroscope)     | GPS            |
| 亮度 (Light)          | RFID           |
| 距离传感器 (Proximity)   | NFC            |
| 压力 (Pressure)       |                |
| 温度 (Temperature)    |                |
| 摄像头 (Camera)        |                |
| 声音 (Audio)          |                |

4.2 代价的测量方法

需要测量的代价是访问传感 API 的平均功耗和时延,即 E(Sensing API)和 T(Sensing API)。测量方法的基本思想是:在一段时间内持续性频繁地访问特定的传感 API(记为  $X_{api}$ ),并且记录系统增加的功耗  $P_{test}$  和总耗时  $T_{test}$ ,然后除以访问该传感 API 的总次数  $N_{test}$  就可以得到 E(Sensing API)和 T(Sensing API),如下所示:

$$E(X_{api})=(P_{test} * T_{test}) / N_{test}$$

$$T(X_{api})=T_{test} / N_{test}$$

实际上 Android 平台的传感器一般都是周期性访问的。 $T(X_{api})$ 就代表了访问  $X_{api}$  传感器的周期,在数值上等于采样频率的倒数。

传感 API 的参数与获取传感器输出的代价有关系。与代价有关的参数是:采样率、传感精度(accuracy)和传感器类型。一般来说,采样率越高,时延越小,但功耗越高。Android

系统定义的传感器类型一般有 4 种级别的采样频率:最快(Fastest)、游戏(Game)、正常(Normal)、用户界面(UI)。GPS 频度参数有高(High)、中(Medium)、低(Low);精度参数有低(Low)、精细(Fine)、粗(Coarse)等。这些参数都会影响 GPS 获取位置信息的代价。

4.3 测量工具的实现

代价模型的测量工具 CRCTest 是在 Android 平台上实现的。由于传感器之间各有差别,不同类别的传感器用不同的方法测量时延和功耗开销。

• 对于所有传感器(包括网络接口等),持续均匀地访问单个传感器,记录访问次数,并用外置电表记录功耗,通过计算得到访问该类传感器接口的功耗开销。

• 对于 Android 系统框架中传感器管理器(android.hardware.SensorManager)定义的所有传感器,其时间延迟即为发出获取数据请求到最新的数据返回之间的时间差。

• 对于无线网络接口、相机、麦克风等广义上的传感器,定义其时间开销为请求开启设备到设备就绪的时间差。

该测量工具需要一个灵敏度较高的外置电表配套工作,具体使用流程一般如下:

- 1)选择要测量的传感器类型、测量精度、频率等选项;
- 2)按“开始”按钮,测量工具开始监听和记录传感器数据;
- 3)按“结束”按钮,测量工具将记录写入日志文件并停止监听传感器数据。

5 Android 平台上下文信息获取的代价模型

5.1 测量的平台环境和设置

本文用基于上述代价模型的测量工具 CRCTest,对一个典型的 Android 手机平台 Google Nexus S(运行 Android 2.3.6操作系统)进行了测量,获得了一个实际的 Android 平台上下文信息获取的代价模型。

在代价模型的测量过程中,使用外置的 Power Monitor 电表设备实时地监测手机的电流和功耗。测试过程中,手机使用外部供电,电压恒定为 3.7V,屏幕保持常亮且亮度不变,测试某种传感器时无其他应用同时运行,无其他传感器同时开启。对于每一类传感器(及网络接口等)都连续地采集 5 分钟的传感器数据与 Power Monitor 电流数据。

5.2 Android 平台的代价模型

经过测量和计算,获得 Google Nexus S (Android 2.3.6)平台上下文信息获取的代价模型,表 2—表 4 列举了其 主要特征参数。其中,表 2 和表 3 分别展示了 Android 系统 API 定义的传感器的功耗开销和其他广义传感器的功耗开销,单位均为毫瓦(mW)。表 4 是几种常用的狭义传感器在最高采样频率和最低采样频率下的时间延迟,单位为毫秒(ms)。

表 2 6 种典型传感器的功耗开销<sup>1)</sup>

| 传感器   | 频率   | FASTEST | NORMAL | 功耗差别 |
|-------|------|---------|--------|------|
|       | (最高) | (正常)    |        | (mW) |
| 重力加速度 | 531  | 465     |        | 66   |
| 螺旋仪   | 761  | 510     |        | 211  |
| 旋转向量  | 508  | 466     |        | 42   |
| 磁场    | 490  | 441     |        | 49   |
| 方向    | 471  | 444     |        | 27   |
| 加速度   | 461  | 445     |        | 16   |

<sup>1)</sup> 注:测试时手机的基本功耗为约 432mW。

表3 摄像头、麦克风和网络接口导致的功耗开销

| 测试情境        | 手机总功耗(mW) | 传感器功耗(mW) |
|-------------|-----------|-----------|
| 手机闲置        | 432       |           |
| GPS 室外      | 572~630   | 140~198   |
| GPS 室内      | 657       | 225       |
| 麦克风录音       | 557~671   | 125~239   |
| 相机开启        | 722       | 290       |
| 3G 开启并闲置    | 534       | 102       |
| Wi-Fi 开启并闲置 | 471       | 39        |
| GPRS 开启并闲置  | 477       | 45        |

表4 访问传感器的平均时间延迟

| 传感器类型 | 频率 | FASTEST | NORMAL |
|-------|----|---------|--------|
|       |    | (最高)    | (正常)   |
| 加速度   |    | 20      | 156.53 |
| 重力加速度 |    | 19.57   | 156.5  |
| 线性加速度 |    | 19.57   | 156.52 |
| 螺旋仪   |    | 1       | 9.39   |
| 旋转向量  |    | 19.58   | 156.53 |
| 磁场    |    | 16.92   | 125.27 |
| 方向    |    | 16.9    | 125.35 |

由表2和表3可以得出一些结论,它们可用于优化上下文信息生成算法,例如:应该尽量选择低能耗的传感器。这些结论主要有:

- GPS模块、相机、麦克风和 Fastest 高频采样模式下的螺旋仪功耗很高;
- 加速度传感器、方向传感器、距离传感器即使在高频率的 Fastest 模式下,功耗开销不大;
- 加速度传感器、方向传感器等在高低采样频率下功耗差别不大;
- 处于室内的 GPS 设备没有信号,但是不断地搜寻卫星信号的功耗代价很高;
- 3G 设备即使在闲置状态下(即没有传输应用数据时)功耗依旧很高。

从表4可以看出,螺旋仪非常灵敏,可以达到很高的采样频率,即使是在正常模式下,也只有0.009s的延迟。但是,螺旋仪在高频模式(Fastest)下的能耗比正常模式多了49%。因此,在优化中可以考虑这一点,让螺旋仪尽量不在高频模式下工作,从而达到节省电量的目的。

将表2~表4这3个表格的分析结果综合起来,可以得到如下结论:1)GPS、麦克风、相机、螺旋仪传感器比较耗电,应当减少使用;2)加速度传感器高低精度下功耗差别很小,可以相对更多地在高精度下使用,而螺旋仪在高低精度下功耗差别很大,应当尽量在低精度下使用。

## 6 代价模型的应用实例及评估

本文实现了一个应用实例来验证提出的上下文信息获取代价模型的有效性,实际地检验和测试上下文信息获取能耗优化的效果。这个应用实例能够用两种不同的方式提供位置信息:

- 1)能够连续访问 GPS 获取用户位置;
- 2)能够结合 GPS 和加速度传感器来获取用户位置。

基于代价模型的应用实例可在通过结合加速度传感器和 GPS 来获取位置信息的同时减少整体能耗。由表2和表4对比可以看到,GPS和加速度传感器的功耗之间相差100多毫瓦,因此通过增加加速度传感器的使用并减少GPS的使用的

方法,应该能为位置信息的确定节省不少能耗。

定位方式1:GPS+加速度传感器。加速度传感器保持常开,采样频率为正常模式(Normal),在需要时才通过GPS获取位置信息。这种GPS结合加速度传感器获取位置信息的工作流程如下:首先,应用GPS获取位置信息,定义此时用户所处位置为原点并且暂时不再使用GPS;之后,应用对加速度传感器的值在水平方向(与地面水平)上进行积分,计算出用户当前位置与原点的距离,如果距离大于50米,则开启GPS获取位置信息,更新位置信息与原点位置。

定位方式2:仅GPS。连续访问GPS获取位置信息在本应用中的定义如下:设定最小更新距离为50,最小更新时间分别为0和60000,不断地通过GPS获取位置信息。

为了能够与测量得到的代价模型数据相对应,测试手机依旧选用Google Nexus S。此外,经过多次实验发现:加速度传感器设置在正常采样模式(Normal)下完全能够满足需求。当采样频率设置为最快(Fastest)、游戏(Game)和UI模式时,定位结果的精度虽然略有提升,但是与之而来的却是更大的CPU计算量。因此,应用实例中的加速度传感器在正常模式下工作。

两种定位方式的功耗开销(单位:mW)如图1所示。Android手机访问加速度传感器可以有4个采样频率选项,访问GPS可以有一个更新距离选项和一个更新时间选项(两者越小,更新频率越高)。图例中括号内的数据表明了应用实例中的参数选择,如GPS(0,0)表示要求Android系统以最高的频率和精度提供位置信息,GPS(0,50)表示要求Android系统以最高频率和最小更新距离为50米提供位置信息。从图1可以很轻易地看出,加速度传感器与GPS结合的方式相比仅仅使用GPS获取位置信息至少能够减少13%的功耗。

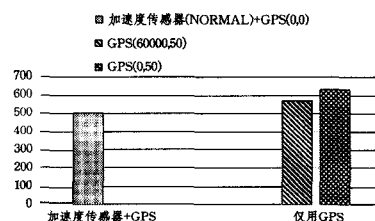


图1 两种定位方式的能耗对比

上述只是基于上下文信息获取代价模型的一个简单的应用示例,但是能够说明上下文信息获取代价优化还存在很大的空间,也存在着必要性和迫切性。

**结束语** 本文针对智能手机上下文信息获取的高能耗问题,提出了一个智能手机上下文信息获取代价的分析和评估模型。该模型考虑了应用程序获取上下文信息的时间开销和能耗开销,可以用E(Sensing API)和T(Sensing API)来表示。在Android移动平台上设计并实现了一个软件工具CRCTest来全面细粒度地测量特定Android平台的上下文信息获取代价模型。利用CRCTest工具和Power Monitor电表,在Google Nexus S智能手机上实际地测量出了一个代价模型,并得出一些结论:1)GPS、麦克风、相机、螺旋仪传感器比较耗电,应当减少使用;2)加速度传感器高低精度下功耗差别很小,可以相对更多地在高精度下使用。最后,本文基于Google Nexus S平台的代价模型实现了一个优化的应用实例来生成位置上下文信息。通过对比分析不同方式下的功耗开销,可以看出使用GPS和加速度传感器的组合能在满足应用

需求的同时降低系统的能耗。

## 参考文献

- [1] Lane N D, Miluzzo E, Lu H, et al. A Survey of Mobile Phone Sensing [J]. IEEE Communications Magazine, 2010, 48(9): 140-150
- [2] Chu D, Kansal A, Liu J, et al. Mobile Apps: It's Time to Move Up to CondOS [C]//Proceedings of the 13th Workshop on Hot Topics in Operating Systems. 2011: 16
- [3] Brunette W, Sodt R, Chaudhri R, et al. Open data kit sensors: a sensor integration framework for android at the application-level [C]//Proceedings of the 10th International Conference on Mobile Systems, Applications and Services. 2012: 351-364
- [4] Brouwers N, Pogo L K. A Middleware for Mobile Phone Sensing [C]//Proceedings of Middleware 2012. 2012: 21-40
- [5] Priyantha B, Lymberopoulos D, Liu J. LittleRock: Enabling Energy-Efficient Continuous Sensing on Mobile Phones [J]. IEEE Pervasive Computing, 2011, 10(2): 12-15
- [6] Ben A F, Phillips A, Henderson T. Less is more: energy-efficient mobile sensing with senseless [C]//Proceedings of the 1st ACM Workshop on Networking, Systems, and Applications for Mobile Handhelds (MobiHeld). 2009: 61-62
- [7] Nath S. ACE: exploiting correlation for energy-efficient and continuous context sensing [C]//Proceedings of the 10th International Conference on Mobile Systems, Applications and Services (MobiSys'12). 2012: 29-42
- [8] Roy N, Misra A, Julien C, et al. An energy-efficient quality adaptive framework for multi-modal sensor context recognition [C]//Proceedings of the 2011 IEEE International Conference on Pervasive Computing and Communications (PerCom'11). 2011: 63-73
- [9] Liu Ye-pang, Xu Chang, Cheung S C. Where Has My Battery Gone? Finding Sensor Related Energy Black Holes in Smartphone Applications [C]//Proceedings of the 11th IEEE International Conference on Pervasive Computing and Communications (PerCom'13). 2013: 2-10

(上接第 131 页)

计人员考虑了遥控器操作的特点,所设计的网页包含较少的上下页或滚动条操作元素,导致可优化的地方比较少,优化结果不明显。若应用中包含较多页面的翻页和滚动操作,我们方法的优化效果会更为明显。

### 5.2 网页元素分类查全率与查准率分析

我们通过手工检测发现在程序中有 6 个元素本可以参与优化调换,但被匹配算法过滤了,通过查看它们在代码中的设计与对比其显示尺寸,我们发现这些元素与它们应属的分类中的元素在代码中定义的大小并不相同,然而在显示的时候却显示出同样的大小,因为代码中定义的大小的差距过大,导致了它们被过滤掉,相对于我们查到的可调换的 82 个元素,可替换性评估方法的查全率为 93.2%。同时,我们对找出的可调换的 82 个元素进行手工检测,发现有 3 个元素没有可替换的对象,评估算法的查准率为 97.3%。

**结束语** 通过基于浏览代价和用户习惯的调优,我们发现对于实验所测试的互动应用界面而言,基于指令的交互方式和基于距离的交互方式产生的优化效果比较明显,而基于直接点击的交互方式的优化效果相对而言并不明显。这主要是由被测试的应用中能够符合基于直接点击交互方式优化的页面相对较少导致的。总体而言,从浏览代价的角度来看,利用用户访问习惯进行界面调优是具有明显效果的。在对网页元素分类时,我们发现有一些本应该归为同一类的元素并没有被归为同一类,这主要是由元素的代码设计与展示效果并不一致而导致的,我们认为今后可以结合实际展示效果来对页面元素进行分类;另一方面,交互性元素与非交互性元素之间的语义联系并没有得到识别,我们只是从展示形式和结构关系上对交互性元素和非交互性元素的联系进行了识别,在语义上的联系的识别也是值得后续研究的一部分。

## 参考文献

- [1] Brush A J, Lee B, Mahajan R, et al. Home Automation in the Wild: Challenges and Opportunities [C]//CHI'11 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York: ACM Press, 2011: 2115-2124
- [2] Kamilaris A, Pitsillides A, Trifa V. The Smart Home meets the Web of Things [C]//International Journal of Ad Hoc and Ubiquitous Computing 2011. Genève: Inderscience publishers, 2011: 145-154
- [3] Byun J-S, Jeon B, Noh J, et al. An intelligent self-adjusting sensor for smart home services based on ZigBee communications [J]. IEEE Transactions on Consumer Electronics, 2012, 58(3): 794-802
- [4] Hartmann M, Schreiber D. AUGUR: Interface Adaptation for Small Screen Devices [C]//Advances in Ubiquitous User Modeling Lecture Notes in Computer Science Volume 5830. Berlin: Springer-Verlag, 2009: 94-110
- [5] Funabiki N, Shimizu J, Isogai M, et al. An Extension of the Web-page Layout Optimization Method for Multimodal Browsing Sizes [C]//The 13th International Conference on Network-Based Information Systems (NBIS). 2010: 139-146
- [6] Nezhad H R M, Benatallah B, Martens A, et al. Semi-automated adaptation of service interactions [C]//WWW '07 Proceedings of the 16th international conference on World Wide Web Pages. 2007: 993-1002
- [7] Furche T, Grasso G, Kravchenko A, et al. Turn the Page: Automated Traversal of Paginated Websites [C]//Web Engineering Lecture Notes in Computer Science. 2012, 7387: 332-346
- [8] Agarwal R, Arya K V, Shashi S, et al. An Efficient Weighted Algorithm for Web Information Retrieval System [C]//International Conference on Computational Intelligence and Communication Networks (CICN). 2011: 126-131