

一种基于浏览代价优化的电视网页重构方法

龙永浩^{1,3} 王佳¹ 陈湘萍^{2,3} 利开园^{1,3} 欧阳春霞¹

(中山大学信息科学与技术学院 数字家庭互动应用国家地方联合工程实验室 广州 510006)¹

(中山大学先进技术研究院 国家数字家庭工程技术研究中心 广州 510006)²

(中山大学深圳研究院 深圳 518057)³

摘要 随着智能家庭概念的发展与推广,出现了大量面向电视机用户的 Web 应用。由于用户与电视机之间通过遥控的方式进行交互,一方面使得网页浏览代价受页面元素排布的影响较大,而另一方面,电视机网页元素类型和数量少的特点,使得对其进行自动优化成为可能。考虑到包括按键式遥控器、手势、触屏等电视机交互方式,根据不同的交互方式的特点提出了页面浏览代价的计算方法;考虑到元素类型、位置、大小、内容关键字、网页结构等信息,提出了一种页面元素可替换性评估方法;在这两种方法的基础上,基于网页浏览代价全局最小化的原则,提出了根据页面的访问情况动态对网页进行自动重构的方法。在一个包含 116 个页面的电视机 Web 应用中采用该方法进行的实验验证了方法的正确性和有效性。

关键词 网页重构,浏览代价,可互换性评估,网页优化

中图分类号 TP311.5 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.11.025

Automatic Refactoring of TV Webpage for Optimizing Cost of Browsing

LONG Yong-hao^{1,3} WANG Jia¹ CHEN Xiang-ping^{2,3} LI Kai-yuan^{1,3} OUYANG Chun-xia¹

(School of Information Science and Technology, State-Province Joint Laboratory of Digital Home Interactive Applications, Sun Yat-sen University, Guangzhou 510006, China)¹

(Institute of Advanced Technology, Sun Yat-sen University, National Engineering Research Center of Digital Life, Guangzhou 510006, China)²

(Research Institute of Sun Yat-sen University in Shenzhen, Shenzhen 518057, China)³

Abstract With the development of the smart home, a great number of TV-user oriented Web applications are developed. Since the users interact with the television by remoter, the browsing cost of the webpage is affected by its layouts. What's more, the simplifying of the elements on the webpage makes the automatic optimizing possible. In this paper, we proposed an algorithm to calculate the browsing cost for different kinds of interactions. Considering the elements' types, positions, sizes, and structures, we proposed an interchangeability estimate algorithm to check out whether the two elements can change their positions with each other. On the basis of the two mentioned methods, we proposed an automatic webpage refactoring which gives each element a weight from the visits of the users and refactoring the webpage in order to reduce the total browsing costs. The technique was tested on a website which consists 116 pages to test the correction and significance of the method.

Keywords Webpage refactoring, Browsing cost, Interchangeability estimate, Webpage optimizing

1 引言

随着物联网技术的发展,智能家庭^[1,2]这一概念被广泛推广,出现了很多为家庭设备开发的软件应用。Web 网站应用的开发者也开始将传统 Web 页面移植到电视机屏幕,出现了大量面向电视用户的 Web 应用^[3]。在本文的前期调研中,我们对 190 个电视网页和普通网页进行了统计对比,每个电

视网页中包含的元素平均数量为 Web 页面的 1/20,其中带超链接的元素平均数量为 Web 页面的 1/100;相对于传统网页,电视机的网页有着元素数量少、元素类型简单的特点,这使得自动电视网页优化成为可能。

由于电视机与用户之间的距离较远,电视机上的页面交互方式主要通过遥控来交互。目前,主要的交互方式包括:按键遥控器、手势识别、触屏遥控器等。用户在浏览页面时,发

到稿日期:2013-09-16 返修日期:2013-11-16 本文受国家自然科学基金(61100002),广东省自然科学基金(S2011040005180),深圳市生物、互联网、新能源、新材料产业发展专项资金项目(JC201104220324A),中央高校基本科研业务费专项资金,国家科技支撑计划(2012BAH75F03)资助。

龙永浩(1990—),男,硕士,主要研究方向为软件工程、人机交互,E-mail:qoolyh@gmail.com;王佳(1989—),男,硕士,主要研究方向为软件工程、人机交互;陈湘萍(1981—),女,博士,助理研究员,主要研究方向为软件工程、软件分析、软件体系结构,E-mail:chenxp8@mail.sysu.edu.cn;利开园(1989—),男,硕士,主要研究方向为软件工程、大数据分析;欧阳春霞(1991—),女,主要研究方向为软件工程。

出一系列指令,通过遥控器上的按键、手势交互,或触屏上滑动操作等移动焦点或者跳转页面。与鼠标的交互方式不同,大部分的电视机交互方式在单个网页中移动焦点到目标元素的过程需要用户额外的操作。例如,使用按钮遥控器浏览一个包含 10 行标题的新闻页面时,需要点击 10 次才能将焦点移动到第 10 个新闻上。因此,页中元素的位置排布成为影响电视机用户网页浏览代价的重要因素。

传统的针对页面布局的优化方法主要是针对导航元素的布局优化^[4],或是将传统页面移植到不同的显示设备上的布局优化^[5,6],页面的布局优化相对来说比较匮乏。这主要是因为传统网页结构复杂,元素的识别和调整难以自动化。由于电视机上交互方式的特点,不同网页所需的浏览代价可能会相差很大,基于页面排布的优化变得尤为重要。而另一方面,电视上的网页页面结构相对简单,模式也较为简单,这使得对网页的自动重构成为可能。

针对电视网页的特点,本文面向电视机用户,提出一种基于浏览代价优化的网页重构方法。该方法根据不同的交互方式的特点,计算每个页面元素的浏览代价;考虑元素类型、位置、大小、内容、网页结构等信息,提出一种元素可替换性评估方法;在此基础上,该方法基于网页浏览代价最小化的原则,根据页面的访问情况,动态对网页进行自动重构。

我们对一个包含 116 个页面的电视机互动应用程序进行实验,以验证方法的有效性和正确性。通过对采集到的 4 组不同的用户访问记录的分析,我们的方法对于以命令实现焦点跳转、以拖动距离实现焦点跳转和直接触屏点击的 3 种浏览方式,分别能够平均减少用户 27.1%,27.9%和 6%的浏览代价。其中,对元素的可替换性评估方法的查全率为 93.2%,查准率为 97.3%。

2 相关工作

在提升网页交互性能方面,已有的工作关注对页面元素重要性的评估,并在此基础上预测用户行为或者进行网页优化。Agarwal 等人^[8]针对网页元素的权值计算给出了一种结合点击数、访问时间、鼠标停留综合计算代价的方法。我们采用了上述方法中将用户的访问频率作为元素的权值的计算方式,结合电视机页面浏览代价的特点对网页进行了结构调整。

已有网页优化方法主要关注页面的自适应调整,特别是针对小屏幕设备的页面自适应调整^[4-6],Hartmann 等人^[4]将整个网站按页面结构分解成树,每个树节点代表一个页面,并通过统计用户的访问记录对每个节点标明频度值,通过他们提出的 FxL* 算法能够很好地预测出用户针对当前页面下一步最有可能访问的页面;文献^[5,6]侧重于屏幕分辨率的识别,文献^[5]通过一种引用模型架构的方法来实现对移动设备的自适应,而文献^[6]主要研究屏幕分辨率识别的错误匹配的矫正。受到小屏幕设备的影响,原始页面的元素不能全部显示到设备上,所以这些方法将排除大部分非交互性元素。实际上有些非交互性元素与交互性元素是有联系的,不应该将它们都过滤。电视机页面与普通页面和手机网页上的不同在于很多情况下页面中的元素都是比较简单的,元素与元素之间的联系比较明朗。我们的方法根据电视机页面特性来关注网页的自动优化,并考虑了部分非交互性元素与交互性元素

的匹配,其中元素的相似性评估方面借鉴了文献^[7]中部分网页元素的通用特征。

3 网页元素代价计算与分类

3.1 浏览代价计算

由于用户与电视机的交互方式不同,相应的产生的浏览代价的计算方式也不同;按键式遥控器的浏览代价主要在按键数上,手势交互的浏览代价主要在元素与焦点的距离或手势数上,触屏交互的浏览代价主要由元素与焦点的距离或元素所在页面与起点所在页面的页面数决定。我们发现这几种交互方式从影响浏览代价的计算角度来看可以分为以指令数为浏览代价的交互、以元素距离为浏览代价的交互,以及以页面数为浏览代价的交互。

最小代价的计算首先需要将整个网站结构分解成结构图(见图 1),其中,节点代表的是每个页面,节点中的数字代表页面的 ID,节点间的连接线代表跳转,上面的数字代表在上个页面中从起始元素移动到下个页面的链接元素所需的浏览代价。之后利用 Dijkstra 算法求出最小代价。

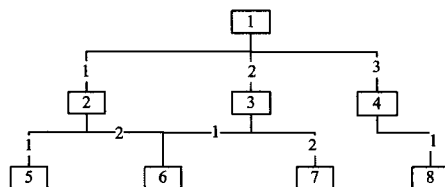


图 1 网站结构图

基于指令操作的交互的浏览代价

基于指令操作的交互方式中,用户通过遥控器输入指令控制电视机焦点位置以及页面跳转。这种交互方式中每种操作的代价一样,因此浏览代价由指令数决定。用户在浏览过程中,可能由于各种原因而产生无效操作,例如:用户将焦点跳到左边后,又将其跳回右边。这些无效操作不在本文优化的范围内。页面的浏览代价定义为:在初始化后,用户通过发出指令达到目标元素的过程中所需的最小指令数。

基于距离的交互代价计算

除了通过手势交互或触屏交互操控电视页面,还有一种交互方法是通过移动指针到目标区域然后点击进入的方式。在这种交互方式下,我们认为浏览代价的主要部分在于指针移动的路径长度,即目标点与起点之间的最小距离,因而其与基于指令的交互代价的计算主要在于单个页面内的代价计算。为了能够与指令操作交互下的数据进行对比,我们将距离通过如下公式变换成浏览代价:

$$browseCost_i = 1 + \max(\frac{d_i - d_0}{d_0}, 0)$$

其中, d_i 为目标点与起点的直线距离, d_0 为整个网站中的元素间距的平均值,元素间距指的是相同的 X 轴或 Y 轴且中间没有任何元素的两个元素之间的距离。通过这种变换,网站结构图上边的数值即为相应的 $browseCost_i$ 。

直接点击操作的代价计算

基于直接点击操作的交互方式主要存在于触屏交互中,其浏览代价的特点是一个页面中的元素的浏览代价消耗是相同的,如果在页面中存在上下页或滚动条的情况,则以一个页面尺寸为基准来计算目标元素与起点之间的距离。类似地,

浏览代价的变化只在页内的浏览代价上体现,网站结构图上的边的数值即为相应的步数。

通过上述的步数计算后,更新原有的网站结构图上各个连线上的权值,之后根据 Dijkstra 算法即可求出从起始页面到目标页面的最小浏览代价。

3.2 元素可互换性评估

对于网页上的元素,我们将其定义为一个五元组:

$$element = \langle type, keyword, size, position, structure \rangle$$

其中, $type$ 为元素类型,用以识别待优化元素。在电视互动应用中,页面元素可以分为交互性元素和非交互性元素。交互性元素指元素的类型属于链接型,主要类型为 link。用户浏览时,其页面焦点只在交互性元素之间移动。非交互性元素即不会与用户产生交互的元素,在我们分析的系统中,元素类型不为 link 的元素都是非交互性元素。

$keyword$ 为元素的内容关键字,用以过滤特殊的功能性元素。 $size$ 为元素的尺寸,用来比较元素的大小, $size$ 由元素的上下左右坐标值来描述, $size = \langle top, bottom, left, right \rangle$, 单位是 $pixel$ 。 $position$ 为元素的位置,由其左上角的绝对坐标值表示, $position = \langle x, y \rangle$, 其中 x 为横坐标, y 为纵坐标。 $structure$ 为其所属的 HTML 结构。

我们对元素的调整过程中只对交互性元素进行了评估。但在一个页面中,有些交互性元素可能与一些非交互性元素有联系;同时,页面中的元素即使都是交互性元素,它们也有可能分属于不同的部分,例如 A 元素是一条新闻而 B 元素是一条广告,它们之间并不可互换位置。种种原因使得页面中的元素不能随意地互换位置;因此,我们提出了在一个页面中对两个元素间进行可互换性评估分析,通过评估的得分将可以互换位置的元素归为一类,对每一类元素使用相应的调换策略。我们通过表 1 所列条件对两个元素的可互换性进行评估。

表 1 两个元素的可互换性评估条件

条件名称	描述	条件类型
类型	判断元素的类型是否是待优化类型,如果不是则过滤掉	过滤型
关键字	判断元素是否为特殊的功能性元素,如果是则过滤掉	过滤型
邻居	判断两个元素是否含有非交互性邻居,邻居的相似度越高,可互换性越高	评估型
大小	比较元素与另一个元素所在空间的长与宽,元素的大小差距越小,可互换性越高	评估型
HTML 结构	比较两个元素所在 HTML 结构树中节点的距离,距离越近可互换性越高	评估型
位置	比较两个元素的相对位置以及两者中间是否含障碍物,两个元素距离越近,障碍物数越少,其可互换性越高	评估型

3.2.1 过滤型评估条件

对于一个页面,其包含的元素为集合 $Element$ 。过滤性评估条件将对 $Element$ 进行过滤,获得存在调优可能性的元素集合 $Element_{inBeOpt}$ 。

元素的类型:如前文所言,我们只考虑交互性的元素,非 link 型的元素是暂不考虑的,即:

$$E_{inBeOpt} = \{e | e.type = link, e \in E\}$$

元素的内容:在一个页面中,有一些元素具有特殊功能,如 'back'、'next'、'prev' 等。这类元素的排布通常与网页风

格相关,不参与优化。这类元素的特征是在 $html$ 的定义中包含关键字,如 'back'、'next'、'prev' 等。

$$E_{inBeOpt} = E - \{e | e.keyword = 'back' \parallel 'next' \parallel 'prev' \parallel \dots\}$$

如果两个待优化元素通过了上述的两个判定条件,我们认为这两个元素具有了互换位置的可能,并将通过下述几个判定条件来评估两个元素互换位置的可行性和调换的策略。

3.2.2 评估条件

对于 $E_{inBeOpt}$ 中的两个元素 a 和 b ,以下评估条件将用于评估两个元素之间可以互换的可能性。

元素的邻居:对于两个元素 $a(x_a, y_a)$ 和 $b(x_b, y_b)$, a 和 b 在同一 X 轴(或者 Y 轴)上,若两者间没有其它元素,则 a 和 b 是 X (或 Y)上的邻居,记为 $neighbor(a, b) (\{e | e \in E_{inBeOpt} \wedge ((y_b < e.position.y < y_a) \vee (y_a < e.position.y < y_b))\} = \emptyset) \wedge (x_a = x_b) \vee ((y_a = y_b) \wedge \{e | e \in E_{inBeOpt} \wedge ((x_b < e.position.x < x_a) \vee (x_a < e.position.x < x_b))\} = \emptyset) \rightarrow neighbor(a, b)$ 。

两个元素 a, b 在互换位置时有可能出现 a, b 与非交互性元素有联系的情况,对于这种情况, a, b 与非交互性元素是互为邻居关系的,因此在考虑两个元素 a, b 是否能互换位置时,我们需要查看两个元素是否含有非交互性元素;如果两者均有非交互性元素的邻居,则对比两者的邻居(包括相对于 a, b 的相对位置,两个邻居元素的相对位置、大小、类型)。如果两个元素相似,则将两者合为整体再对比邻居,直到找不到相似的邻居为止。

算法 1 比较两个元素 a, b 的邻居集的相似度

输入:两个待比较的元素 a, b

输出:两个元素的相似邻居集

说明: $checkNeighbors(e, x, y, tmpSet, num)$ 用于查找元素 e 同 x 轴或 y 轴(当 $x=1$ 或 $y=1$ 时)上的元素,并将它们按坐标从小到大放入数组 $tmpset[]$ 中, num 表示数组中的元素数。

$compare(set_1, set_2)$ 用于比较并找出两个集合的相似部分。

$swap(a, b)$ 用于将 a, b 两个数的值互换。

START:

IF($a.position.x = b.position.x$)

THEN $xAris = 1$;

ELSE IF($a.position.y = b.position.y$)

THEN $yAris = 1$;

ELSE{

$checkNeighbors(a, 1, 0, tmpSet1, num1)$;

$checkNeighbors(a, 0, 1, tmpSet2, num2)$;

$(num1 > num2) ? (xAris = 1) : (yAris = 1)$;

$checkNeighbors(a, xAris, yAris, neighborSet_a, numA)$;

$checkNeighbors(b, xAris, yAris, neighborSet_b, numB)$;

$commonSet = compare(neighborSet_a, neighborSet_b)$;

IF($commonSet = NULL$) {

$swap(xAris, yAris)$;

$checkNeighbors(a, xAris, yAris, neighborSet_a, numA)$;

$checkNeighbors(b, xAris, yAris, neighborSet_b, numB)$;

$commonSet = compare(neighborSet_a, neighborSet_b)$;

RETURN $commonSet$;

END

对于带有邻居集的情况,我们认为元素移动位置时,其邻居集应跟着一起整体移动,因为这些邻居集可能与元素具有语义上或逻辑上的联系。如果两个元素均有邻居集,则从大

小、类型、相对位置上对比邻居集,根据相似度进行评分,如果没有匹配的邻居集,则在该条件下的得分为0。

元素的大小:在考虑两个元素调换后是否会影响页面视觉效果时,我们需要计算两个元素的大小差距。我们将一个元素本身的大小与另一个元素所在空间的大小的对比作为大小差距。元素的空间定义为以该元素为中心点往上下左右4个方向延伸形成的最大的不包含其它元素的对称矩形空间。一个元素能够与另一元素互换的前提是该元素大小不大于另一元素的空间大小,否则不能互换。如果一个元素与另一元素所在空间的大小差距在5px以内,我们认为该元素在大小判断条件下的得分为满分,其后差距每增加1px,分数减少1分,直至0分为止。

元素的HTML结构:元素的HTML结构指的是元素定义在HTML文件中的容器型标签,如<div>、<tr>等,如前文所述,在元素调换时可能出现两元素分属不同类别而不可调换的情况。通过对各个网站的元素统计发现,同类元素一般会定义在一个相同的结构中,这个结构一般是元素在DOM树上向上一级的结构,多数为<div>。我们根据两个元素的结构关系可以进行打判别:若两个元素同属一个结构,两者的得分是满分,若均在<div>结构中,则对比两者所在DOM树中节点的距离,距离越远得分越少,其它情况均为0分。

元素的位置关系:如果两个元素具有较规律的位置关系如同X轴或Y轴,我们认为这两个元素互换的可能性高于它们不同轴的情况。在两个元素同轴的情况下,若两元素之间有且只有同类等大的元素,则两者属于同类的概率高一些,相应的得分会比较高;如果两个同轴元素间含有不同类型或不等大小的其他元素,则将这些“障碍”作为减少两者的得分的因素。在两个元素不同轴的情况下,我们主要查看两个点形成的矩形区域中元素的类型与数目,若矩形区域中元素的类型和大小均与两个元素相等,则得分较高;相应地,如果有“障碍”元素,则会减少得分。

4 页面调优

4.1 确定元素的调换范围

通过3.2节可互换性评估后,我们根据两个元素在前述各判定条件下的符合程度给出一个总体分数来描述。如果两个元素在某个条件下的得分过低(即低于规定阈值),我们认为这两个元素不能调换位置。如果两个元素在各个条件下的得分都高于阈值,我们给每个条件一个权值w表示其在判断两个元素互换的可行性时的重要性。两个元素在各个条件的判别后会得到一个总分。之后根据页面中的元素两两之间的得分对所有元素进行分类,每个分类中的元素均可互换。

4.2 根据调换范围对元素进行调优

通过前述的方法对一个页面中的元素进行分类后,对于每个分类中的元素,我们将其按照访问频率从高到低进行排序。同时,将这些元素的位置按照浏览代价进行从小到大的排序。通过修改定义元素信息的CSS文件中的元素位置信息,将访问频率越高的元素放到浏览代价越低的位置上,重新生成HTML页面,完成调优。

5 实验数据与分析

我们采用一个电视机上的互动应用作为实验程序,该应

用包含116个页面,共计493个元素。通过firefox浏览器插件记录用户的访问记录,根据用户的访问记录和浏览方式计算元素的浏览代价和权重并予以优化,将得出的优化页面再次使用用户原来的浏览记录计算总的浏览代价,将其与用户原本的总浏览代价进行对比分析。

5.1 浏览代价优化对比分析

我们随机抽取了4个用户在一段时间内使用该电视机Web应用的浏览记录,并计算其浏览代价。浏览代价记为B_N。同时我们计算了3种不同的交互方式所产生的浏览代价。之后,使用本文方法对网页进行自动重构。对重构后的网页,使用用户的浏览记录计算其浏览代价。优化后的网页浏览代价记为A_N。我们将交互的提升效果记为:

improvement=(B_N-A_N)/B_N

对于每一种交互情况,对比其在不同浏览方式下的优化情况。表2是根据4个用户浏览记录计算得到的使用遥控器按键交互情况下的优化情况对比。从数据中可以看到,我们的方法在不同浏览过程中优化的效果较为稳定。

表2 通过遥控器按键交互的优化对比

	B _N	A _N	Improvement
Case1	433	333	23.0%
Case2	826	601	27.2%
Case3	1375	1026	25.4%
Case4	1329	1412	27.7%
Avg	1156	843	27.0%

我们对3种主要的浏览代价计算方式进行了步数统计与优化对比(见表3)。

表3 各种交互方式的平均优化效果对比

交互方式	Improvement
基于指令	27.1%
基于距离	27.9%
直接点击	6.0%

图2中展示了4个用户浏览记录在3种浏览代价计算方式下的优化效果。从图中可以看到,交互方式是影响浏览代价的重要因素。触摸屏设备等通过直接点击焦点的交互方式,使其浏览代价较少。其他交互方式在经过优化后,其浏览代价仍然远高于直接点击的交互方式。但是由于设备造价等因素,目前电视机交互方式仍然是以按键式遥控器为主。

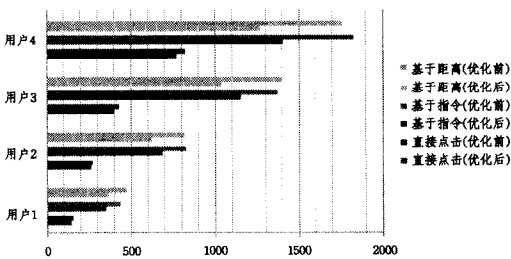


图2 优化结果统计图

表3对比了3种交互方式在不同用户浏览记录下的页面优化效果。其中,对于基于指令和基于距离的产生浏览代价的交互方式,其平均的浏览代价减少率都高于27%,效果较为明显。而对于直接点击产生浏览代价的交互方式,其平均的浏览代价减少率为6%。这是由于我们选取的电视机Web应用本身是针对电视机用户使用遥控器设计的。该应用的设

(下转第136页)

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