软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2012,23(9):2522−2532 [doi: 10.3724/SP.J.1001.2012.04239]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于自适应光标的图形用户界面输入效率优化*

喻 纯[1,2+], 史元春[1,2]

[1](清华大学 计算机科学与技术系,北京 100084)

[2](信息科学与技术国家实验室 普适计算研究部,北京 100084)

Optimizing Input Efficiency for Graphical User Interface Using Adaptive Cursors

YU Chun[1,2+], SHI Yuan-Chun[1,2]

[1](Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

[2](Pervasive Computing Division, Tsinghua National Laboratory for Information Science and Technology, Beijing 100084, China)

+ Corresponding author: E-mail: yc2pcg@gmail.com

Yu C, Shi YC. Optimizing input efficiency for graphical user interface using adaptive cursors. *Journal of Software*, 2012,23(9):2522−2532 (in Chinese). http://www.jos.org.cn/1000-9825/4239.htm

Abstract: The improvement of input efficiency of graphical user interfaces (GUIs) is an important issue in human computer interaction research. The existing researches include two aspects: pointing techniques and adaptive user interfaces: the formal manipulates either the visual representation or the controlling method of a cursor while the latter adjusts the layout of widgets. However, both approaches suffer drawbacks. This paper analyzes the operations on GUI and presents a quantitative model to evaluate the input efficiency of GUIs. Based on the metric, the study then proposes a novel approach, the adaptive cursor, which adaptively supports a predicted set of widgets with pointing techniques to enable fast access. This approach avoids the extra cognitive cost caused by adaptive user interfaces, which frequently adjust the widget layout. It also extends previous pointing techniques that should have applied only to a sparse layout. To evaluate its usability, the study realizes the adaptive cursor technique on Visual Studio, whose interface contains a considerate number of controls. Experimental result shows that the adaptive cursor can save up to 27.7% of pointing time.

Key words: input efficiency; graphical user interface; satellite cursor; adaptive user interface

摘 要: 提高图形用户界面(graphical user interface)的输入效率,是人机交互中的一项重要研究内容.已有的研究包括点击增强技术和自适应界面技术,前者改变光标的控制方式或呈现方式,后者改变界面上控件的位置布局,但两种技术都存在不足.通过分析界面操作,提出了图形用户界面输入效率的评价模型;然后,在此基础上提出一种人机交互效率优化技术:自适应光标.它以自适应的方式,有选择地对界面上用户可能访问的控件通过点击增强技术支持,实现快速访问.该方法既解决了以往的自适应界面技术因频繁调整控件布局而给用户带来额外认知成本的问题,也克服了点击增强技术仅适用于稀疏控件布局的限制.为了检验其可用性,在控件较多的 Visual Studio 上实现了自适应光标技术.实验结果表明,使用自适应光标技术可以将获取目标的时间缩短 27.7%,显著提高了图形用户界面的输入效率.

关键词: 输入效率;图形用户界面;卫星光标;自适应界面

* 基金项目: 国家自然科学基金(61003005); 高等学校博士学科点专项科研基金(20100002110052)
收稿时间: 2012-02-07; 定稿时间: 2012-04-10

中图法分类号: TP391 文献标识码: A

图形用户界面(graphical user interface,简称 GUI)是人机交互的主要界面形式,它以易于记忆的图形来形象地表示计算机系统的访问命令.图形用户界面的主要元素包括窗口(window)、图标(icon)、菜单(menu)和点击(pointing),所以也被称为 WMIP 界面.其中,点击是指通过鼠标、触摸板等指点设备控制屏幕上光标的移动,来获取屏幕上相应的图形元素的过程.如今,尽管新型的多模态输入界面(比如手势、触摸等)不断出现,但是 GUI 还将作为主要的人机界面形式而长期存在.因此,在人机交互领域,持续有大量研究工作致力于提高图形用户界面的输入效率.

图形用户界面上的输入操作有多种,比如拖拽操作、快捷键的设置等,但本文主要关注控件(widget)的点击操作.这是因为控件(比如菜单、工具条等)的使用代表了交互任务中的大部分操作.下文中,我们简称图形用户界面上点击操作的输入效率为界面输入效率或输入效率.

本文提出的自适应光标技术,扩展了作者 2010 年发表在 UIST 论文集上的点击增强技术、卫星光标(satellite cursor)[1].自适应光标技术仅对预测的用户可能访问的控件进行点击的增强支持,从而弥补了原有的卫星光标只适用于稀疏布局(界面上控件数量较少)的缺点.同时,相比于频繁改变控件布局的自适应界面技术,该方法并不改变控件的位置,而是调整光标的位置来缩短点击距离.这样的好处是:既避免了由于界面不断变化引入的认知负担,也简化了自适应策略的设计,使预测算法可以充分提高预测精度,而且不用受到调整频率的限制.用户实验证明,该方法可以显著地提高界面的输入效率.而且,由于自适应光标可以大幅度减少鼠标移动的距离,因而对于电脑人群中越来越多出现的“鼠标手”疾病也能起到预防作用.

本文首先回顾已有研究中增强 GUI 效率的技术和方法,然后提出界面输入效率的评价模型,接着详细阐述自适应光标技术的原理和设计,最后展示用户实验的结果.

1 GUI 效率优化的一般方法

1.1 点击增强

在人机交互理论中,完成点击操作的时间 T 由 Fitts' Law[2]来描述,现在通常使用的形式是

$$T = a + b \times \log\left(\frac{A}{W} + 1\right) \tag{1}$$

其中,A 表示点击的距离;W 表示目标的大小;a 和 b 是常数,不同的指点设备 a,b 的取值不同.

Fitts' Law 的有效性在指点设备的性能研究中得到了广泛的认同[3].由 Fitts' Law 可知,如果增大目标的大小 W 或者缩短点击距离 A,就可以缩短点击时间.基于此原理,已有若干种点击增强技术的研究,其中包括 Bubble Cursor[4],Drag-and-Pop[5],semantic pointing[6]和 satellite cursor[1]等.

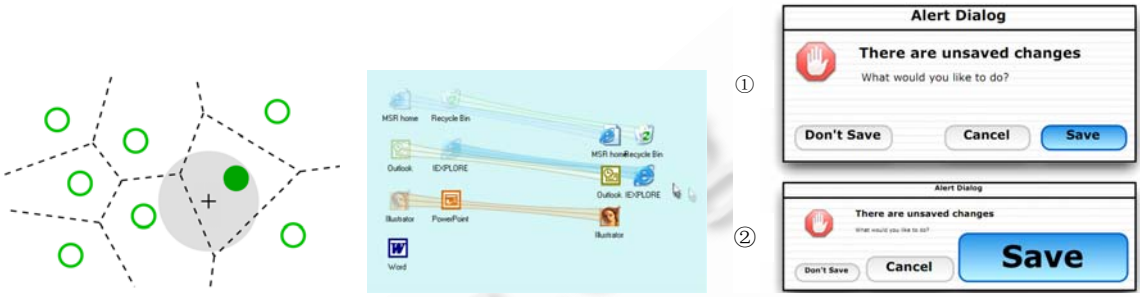
如图 1(a)所示,Bubble Cursor 将每个控件的点击响应区域扩大到了其附近的空白区域(原本不可点击的区域).用户在这些像素位置上的点击,相当于对与其距离最近的控件进行的点击.这是一种增大 W 的方法.如图 1(b)所示,Drag-and-Pop 是一种面向大幅面交互表面的用于获取远处控件的技术.当用户在拖拽某一个文件的图标时,系统自动判断可以用于打开该文件的应用程序,并将它们图标的副本自动移动到光标(用户所在位置)的附近区域.这是一种缩短 A 的方法.

Semantic pointing 则首次从显示域(visual space)和控制域(motor space)这两个空间来看待鼠标控制光标运动的问题.在显示域中,控件和光标之间的距离以其在屏幕上的距离来衡量,单位为像素;在控制域中,光标和控件之间的距离以用户控制鼠标获取控件需要移动的距离来计算,单位是毫米.两个域通过 C/D ratio 来关联,C/D ratio 的定义是用户为将光标移动 1 个单位像素,而需要移动鼠标的距离.C/D ratio 值越大,光标移动越慢;值越小,光标移动越快.Semantic pointing 的思想是在界面的不同区域关联不同的 C/D ratio,从而实现了在不改变显示域的情况下改变控件在控制域中的大小和位置.如图 1(c)所示,如果在情形①中的界面上设置“Save”按钮处的 C/D

ratio 偏大,而“Don’t Save”按钮和其他空白区域处的 C/D ratio 偏小,那么在控制域中则会出现图 1(c)情形②所示的情形.从原理上讲,这是一种混合的方法,因为它既增大了控件的面积,也缩小了点击距离.

卫星光标(satellite cursor)[1]是作者 2010 年提出的点击增强技术,它也是一种缩短点击距离的方法.对于它的介绍,请参见第 3.1 节.关于更多其他增强点击技术的介绍,见文献[3].

上述点击增强技术虽然都通过用户实验被证明可以缩短点击时间,但是它们的应用都有一个前提条件,即需要界面上有足够多的空白区域.否则,上述方法都不能发挥其提高输入效率的作用(比如,在绘图软件中每个像素点都可能被操作).但不幸的是,通常情况下,屏幕上很少存在空白区域,比如当 Office Word 最大化时,屏幕的中间大部分区域都是用户编辑文字的区域,而四周是菜单项、滚动条和工具栏.



(a) Bubble cursor (b) Drag and pop (c) Semantic pointing

Fig.1 Illusration of three pointing techniques

图 1 3 种点击增强技术示意图

1.2 自适应界面

从广义上讲,任何动态变化的界面都可以称为自适应界面(比如反映内容变化的列表、网页等).关于自适应界面技术的详细介绍和分类,可以参考 Findlater 等人[7]的讨论,本文主要关心的是针对图形用户界面上表征交互命令的控件的自适应界面技术.该技术根据交互历史预测用户未来可能访问的控件,然后动态改变界面的布局来提高界面输入效率.具有代表性的研究是 Split Menu[8].对于垂直方向的菜单,Split Menu 将其分为上下两个区域,将用户经常访问的菜单项的副本放置在上方区域,将默认的菜单项放置在下方区域.这样,对于经常访问的菜单选项,用户可以更快捷地获取.实验结果表明,该方法可以缩短用户选择常用菜单命令的时间.可是,虽然自适应界面被认为是实现界面个性化、提高交互效率的有效方法[9],但目前大多数的自适应界面只是在研究中出现而未能被商业软件采纳.这是因为,设计一个自适应界面技术面临诸多挑战,其中包括多方面的因素,如果某一个环节设计不当,就会影响整个自适应界面的可用性.

首先,自适应界面技术需要有一个预测模型来预测控件的访问,研究表明[10],其预测准确性对于自适应界面的性能有着重要的影响.此外,预测模型的可理解性也对用户是否会使用自适应界面有着重要影响:如果预测模型相对简单、易于理解,用户对系统自适应行为的预期会比较准确,有利于自适应界面的使用;反之,用户就有可能不使用自适应界面来进行交互.尽管预测模型的准确性和可理解性不是绝对的对立关系,但是在很多情况下,两者却体现出不相容的特点.这是因为,为了提高预测准确性,直接的方法就是增加用于预测的情景信息或者提高预测算法的复杂度,而这样势必会降低预测模型的可理解性.因此,预测算法的设计要兼顾两方面的因素.

第二,在设计一个自适应界面时,需要考虑自适应界面是通过可选方式(elective)还是强制方式(mandatory)呈现给用户.Split Menu 采用的是可选方式.用户可以自主地选择从菜单上部的自适应区域或者下部的默认区域来选择菜单项.这种设计可以较好地处理预测算法未能成功预测的情况.但是,它的缺点是用户需要付出额外的认知开销来发现系统是否成功做出了调整.相比之下,强制方式不提供用户选择的机会,因而不会增加用户选择界面的认知开销.但是,它在预测失败时,则有可能因为切换界面导致更多的开销.比如,在 Office 2000 中,用户展开菜单后首先出现的是一个包含较少项目的非完整菜单,其中的菜单项是系统根据用户模型预测的结果.用

户如果要访问其他菜单项,需要悬停鼠标若干秒,然后系统才会展开包含所有项目的完整菜单.该设计在随后的Office 版本中就被放弃了,这也反映了强制方式的问题.

最后,大多数的自适应界面是通过改变界面上控件的布局来提高效率的(比如 Split Menu).如此一来,控件布局就处于不断调整的状态,从而带来了额外的认知开销,使用户难以形成操作习惯.该问题被称为自适应界面的布局不稳定性(layout instability)[11].因此,自适应界面调整频率的设计也需要仔细对待.如果调整频率太快,界面无时不刻都是新的,用户需要花大量的时间来确定控件的位置;如果调整频率太慢,又不能及时地反应用户任务特点的变化.

总而言之,要设计一个有用的自适应界面需要综合考虑各方面的因素,单一地追求某一方面的性能(比如高的预测准确度)未必就能够提高自适应界面的可用性.同时,自适应界面的设计很依赖于具体的交互任务,因为其中涉及到的自适应原理和潜在的开销都是迥异的.另外,已有的自适应界面研究大多是针对菜单进行的.然而,在当前的操作系统中,按扭(button)、选择框(checkbox)等通用控件的使用也是无处不在的.与菜单不同的是,通用控件在显示屏幕上的位置更为随机.这样,改变控件布局的自适应方式有可能会引入更多的认知开销.因此,本文通过自适应光标来对通用控件进行自适应支持,在自适应方法和支持的目标控件类型两方面都是对自适应界面研究的补充.

2 GUI 输入效率的评价模型

在研究界面的点击效率时,Fitts' Law 的一个不足之处在于它是针对单次的点击操作而提出的.而实际情况下,交互任务是由一系列的点击操作构成的,涉及到多个控件的顺序点击操作.因此,Fitts' Law 并不能全面地评价输入效率.关于界面的输入效率,还有一个更为基础的方法:GOMS(goals, operators, methods, and selection rules)模型[12].对于交互任务,GOMS 模型明确地定义了任务的目标 G,基本的操作单元 O,用户完成一个任务可能采用的方法 M,以及用户在多种方法之间的选择策略 S.它将用户和应用的交互过程以基本操作的序列来表示.这样,完成交互任务所需要的时间则可以以任务中的完成各个单位步骤时间的总和来计算.GOMS 的限制是,在评价界面效率之前必须明确任务目标和操作流程.但是在实际中,用户的交互任务和操作流程不是固定的,而且多个任务还可能是交叉进行的,因此很难通过预先定义好的操作流程来评价界面效率.比如在撰写文章的过程中,用户可能切换到浏览器去查找资料,然后再回到编辑器继续编辑.面对这样复杂、随机的交互任务时,GOMS 就无法有效地对界面模型进行评价了.

为此,我们以随机序列的角度来看待交互任务中涉及到的控件,并提出基于概率的界面效率评价模型.该模型并不对交互任务做前提假设,而是通过控件的访问概率来描述任务,因而更适合于评价实际情况复杂任务的界面效率.沿用 GOMS 的思想,界面的效率以用户在界面上完成点击任务所需要的时间 T 来进行评价:时间越短,效率越高;时间越长,效率越低.在本研究问题中,操作的基本单元是单个控件的点击操作.这样,一个涉及 M 次控件点击的交互任务可以以随机变量序列($w_1,w_2,\ldots,w_M$)来表示,w_i 表示第 i 次点击操作涉及的控件.如果该序列中一共涉及 N 个不同的控件,那么我们则可以计算时间 T.下面分两步提出.

首先,我们给出一种简单的计算方式:

$$T=\sum_{i=1}^{N}p_i\times t_i \tag{2}$$

其中,p_i 表示 w_i 出现的频率,t_i 表示获取 w_i 所需的时间.

从公式(2)可以得出:经常使用的控件应该布局在易获取的位置.比如在 Windows 系统中,开始菜单通常放置在屏幕左下角.第 1.2 节中谈到的 Split Menu 也是基于此原理,将使用频率高的菜单项自动地调整到便于获取的位置.但是,公式(2)描述的完成点击任务的时间并不确切.这是因为根据 Fitts' Law,获取控件的时间开销并不直接由控件的位置决定,而是取决于获取控件之前光标所在的相对位置.

因此,更准确的描述如下:

$$T=\sum_{i=1}^{N}\sum_{j=1}^{N}p_{i,j}\times t_{i,j}=\sum_{i=1}^{N}p_i\times\sum_{j=1}^{N}p_{j|i}\times t_{i,j} \tag{3}$$

其中,$p_{i,j}$表示任务序列中 w_i 和 w_j 顺序先后出现的概率,$t_{i,j}$表示光标从控件 w_i 处出发获取控件 w_j 所需要的时间,$p_{j|i}$表示 w_j 出现在 w_i 后的概率.

以公式(3)为基础的输入效率评价方法,是本文提出的自适应光标技术的优化依据.从原理上讲,该方法要求图形用户界面在用户每次点击控件后,都根据后续控件的概率分布来重新呈现界面(比如改变控件位置或者光标位置),以充分地缩短时间 T.然而,以调整控件布局为基础的自适应方法要达到这一要求并不容易.首先,用户的交互任务是不断变化的,这意味着 $p_{j|i}$ 也是不断变化的.可是,自适应界面技术要求控件位置的调整频率不能太高,因而难以对用户交互任务的变化做出及时响应;其次,在不能频繁调整控件布局的限制下,如果要兼顾所有的控件,则对布局算法提出了巨大的挑战:由于屏幕资源是有限的,满足了一部分的控件布局要求,很容易会损害另外一部分控件布局要求.因此,很难保证每个控件和其后续可能访问的控件之间的距离都达到最优.总之,由于自适应界面技术本身的限制,使得以调整控件布局为基础的自适应方法不能充分地按照公式(3)所描述的目标来优化输入效率.而本文提出的自适应光标技术,并不需要改变界面上的控件布局,克服了调整频率受限的问题.因此,它的调整策略更加灵活,对于界面输入效率的优化也更为充分.

3 自适应光标

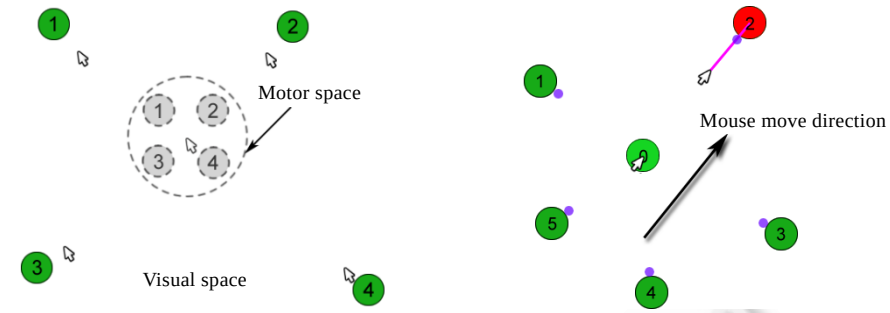
3.1 卫星光标

卫星光标(satellite cursor)[1]是一种增强点击操作的技术,它利用多个随鼠标同步移动的光标来优化点击过程.如图 2(a)中左图所示,对于屏幕上的每一个控件,系统会在其附近生成并关联一个与之唯一对应的光标.该光标总是处于控件的附近,就好像是控件的卫星一样.每个卫星光标只能用于点击与之关联的控件.该技术的一个主要特点是,系统保证在任意时刻最多只有一个卫星光标可以选中与其关联的控件.这样,当用户用卫星光标进行点击时,只需要关注目标控件,然后使用与之关联的卫星光标来点击即可,而不用担心其他的卫星光标也会选中另外的控件.卫星光标技术并不改变显示域中控件的大小和位置,而在控制域中将控件布局在了主光标的附近位置(如图 2(a)虚线圈中的阴影控件所示).主光标在控制域中对阴影控件的点击等价于在显示域中卫星光标对控件的点击.由于点击距离大大缩短了,所以点击的时间也被缩短了.

但是研究表明,卫星光标只适用于控件布局较稀疏的情况,其性能受到控件数量的限制.如果控件数量过多,卫星光标到其关联的控件的平均距离也会增大,从而使性能降低,而且多光标带来的视觉干扰问题也会显现出来.如图 2(b)所示,为了减轻这一影响,系统只显示那些朝着其关联控件移动的卫星光标.这样,大多数的卫星光标是隐藏着的,不会产生干扰.尽管如此,卫星光标技术还是难以直接应用在实际的交互任务中.这是因为在实际情况下,计算机屏幕上的每一个像素都是可以点击的.如果将每个像素都看作是可以单独点击的控件的话,那么可操作的控件数量则大大超出卫星光标可以支持的限度.正如第 1.1 节所讨论的,空白区域的缺乏是所有增大控件尺寸和缩短距离的点击增强技术不能在实际交互任务中发挥作用的重要原因.

3.2 自适应光标的工作原理

自适应光标技术扩展了卫星光标.其工作原理如图 3 所示,在用户交互过程中,系统自动监控界面上控件的点击事件,并预测用户下一步可能需要访问的控件.对于预测可能访问的控件,系统为其分配一个卫星光标;对于不在预测集合中的控件,系统则不进行支持.用户在使用该技术获取控件时,既可以选择使用默认光标,也可以选择使用卫星光标.通过有选择地对控件进行点击增强的支持,大幅减少了卫星光标支持的控件数量,避免了卫星光标在高密度的控件布局中性能受限的问题.如图 3 所示,下面针对点击事件的获取(UI events)、预测模型(predictive model)、自适应光标的生成(control)和模式选择(selection)进行介绍.



(a) 卫星光标在显示域和控制域中的对应情况 (b) 使用中,只显示朝着目标移动的卫星光标

Fig.2 Illustration of satellite cursor

图 2 卫星光标示意图

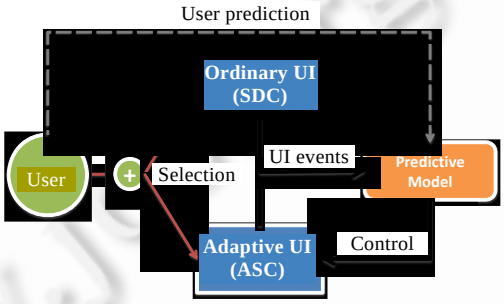


Fig.3 Working principle of the adatpive cursor technique

图 3 自适应光标技术的工作原理图

3.2.1 点击事件的获取

为了获取被点击控件的信息(包括控件的位置和大小),有 3 种方式可以完成:

1) 仅仅记录每次点击发生时光标在屏幕上或者相对于窗口的位置,并以该位置作为控件位置,其形状大小以固定直径(比如 24 个像素)的圆形表示.该方法的优点是实现简单,且不受系统 API 限制;其缺点是,如果控件布局随着窗口大小变化而变化(比如 Word 2010 中,如果窗口变小,Ribbon 区域的工具栏会从 2 行变成 3 行),那么当用户改变窗口大小后,控件的位置信息就不正确了;

2) 通过界面访问接口(accessibility API)来获取控件的详细信息,其中包括位置、名称和类型等.界面访问接口是操作系统用于让第三方的程序获得屏幕上其他应用界面上控件信息的编程接口(比如 Windows 的 UI automation[13]和 Max OS X 中的 AX API).该方法的优点是可以准确、快速地获得控件信息;缺点是,该接口不能解析用户自定义的控件;

3) 通过基于像素的反向工程技术[14]来提取界面上的控件位置和大小信息.该方法可以作为方式 2)的补充,它通过图像处理的方法分析屏幕截图来提取控件信息,可以用于解析用户自定义控件.但是,其缺点是图像识别算法比较耗资源,而且识别的结果也不能保证完全正确.

本文研究中,我们采取了方式 2)来捕获用户在 Visual Studio 上的进行点击事件,详见第 4.2 节.

3.2.2 预测模型的设计

如第 1.2 节中所讨论的,预测模型的准确度和可理解性直接影响着自适应技术的可用性.在本文研究中,预测问题可以描述为根据用户的交互历史(以控件序列表示)预测下一步用户可能会访问的控件的问题.由于卫星光标技术采用多光标的方案,使得预测算法可以预测一个集合而非某一个控件.这一特点有利于降低预测算法的复杂度和提高预测准确度.

基于上述分析,我们提出了一种对应于最优界面效率的预测算法:当系统检测到用户点击了某一控件后,自动搜索交互历史,将历史控件序列中所有曾经出现在该控件之后的控件视为用户有可能会再次访问,而仅对这些控件进行自适应支持.为了控制集合中控件数量的上限以保证卫星光标的正常工作,系统设置阈值 M=10.当预测的控件数量超过 M 时,系统仅仅支持最近使用的 M 个控件,这样就避免了控件数量过多的情况.

- 可理解性:该算法选择最近使用的控件进行支持(简称最近策略).优点就在于其简单性,易于让用户理解系统的工作原理.最近策略被广泛的运用在商业软件应用(比如 Office Word 里面的“最近使用的文件”)和研究[15]中;
- 预测准确度:多项研究[16,17]表明,交互任务中,用户对于控件和命令的使用符合最近性(recency effect).通过合适长度的最近使用列表,最近策略可以覆盖交互任务中绝大多数(>80%)的重访问.此外,研究数据[18]还表明,用户往往只用到界面上所有控件中的一个很小的子集.基于以上考虑我们认为,采用长度为 10 的最近列表来预测用户可能需要访问的控件,也会具有较高的准确度;
- 优化性能分析:如果将控件的访问序列视作对用户任务的充分描述的话(在没有其他辅助信息的情况下),那么自适应光标技术则是充分利用了交互任务的先验知识来对控件的点击距离进行了动态的优化.由公式(3)可知,其对应的界面输入效率可以达到最优.随着用户不断地点击控件,交互历史不断积累,对用户任务的描述也越来越准确.从原理上讲,该自适应策略类似于信息论中的编码过程,即将控件的点击距离按照其被访问的概率进行编码.

3.2.3 自适应光标的生成

自适应光标技术并不改变控件的位置,因而其调整频率也没有限制.系统可以在用户每次点击控件后都进行预测,并且重新设置卫星光标的数量和位置.由于卫星光标离与其关联的控件之间的距离都比较近,点击时间差别不大.因此,在确定卫星光标的位置时,并不按照控件被访问概率的大小来设置点击距离的长短,而是将它们等同对待.

3.2.4 模式区分

自适应光标以可选方式提供给用户:在预测准的时候,用户可以选择使用卫星光标或者默认光标来获取控件;在预测不准的时候,用户只能使用默认光标进行点击.可选方式保证自适应光标至少不会降低系统性能.为此,我们设计了通过鼠标中键来区分模式的操作方法,让用户指定以哪一种光标来点击的操作方法.使用卫星光标获取控件时,用户先移动鼠标将卫星光标移至目标控件之上,然后点击中键来触发点击.系统此时仅仅针对卫星光标进行点击测试.如果用户使用鼠标左键来点击,系统则将该操作理解为使用默认光标来进行点击.由于中键已逐渐成为鼠标的标准配置,使用中键来切换模式的方法在已有交互技术研究中也被采用过.

3.3 利益和开销分析

自适应光标为指点操作带来的好处是显而易见的,用户使用自适应光标不仅有可能缩短点击时间,而且还可以大幅度地缩短鼠标移动的距离.这一特点对于预防时下电脑人群中常见的“鼠标手”疾病(腕管综合症)是有好处的.研究表明,用户在移动鼠标的过程中,腕部承受的压力是静止状态下的 10 倍,缩短鼠标移动距离能减少这一疾病的发生概率.

但是,与使用普通光标点击不同,使用自适应光标时,用户需要提前判断控件是否被自适应光标支持,即是否在自适应光标的预测范围内.典型情况下,如果预测结果为“是”,那么用户会使用卫星光标来进行点击;如果预测结果为“否”,则会使用默认光标.但是,如果预测结果和实际情况相反,那么用户就需要改变点击方案,而花费更多的时间.因此,自适应光标技术可能在 3 个方面引入开销:

1) 用于发现目标是否被自适应支持所需要的认知开销;
2) 模式选择所需要的操作开销;
3) 当用户的判断和系统的自适应行为相反时,所需要的额外操作开销.

4 性能评测

决定自适应界面技术是否有用的关键因素在于技术本身带来的利益和开销之间的平衡关系.因此,本实验的考察点包括:

1) 用户是否会使用自适应光标来点击控件;
2) 如果使用的话,对交互效率的影响又是如何;
3) 使用开销是否会影响用户对自适应光标的使用.

为此,我们开展了受控的用户实验来研究其对于界面输入效率的影响.需要特别强调的是,自适应光标技术所支持的任务场景(缺乏空白区域的界面上通用控件的点击)是已有技术未能解决的:指点技术需要界面上有足够的空白区域,而已有自适应界面的研究是针对菜单进行的.因此,本实验仅以普通光标作为性能比较对象而不考虑这两类界面优化技术.

4.1 实验环境和实验人员

实验平台设置在 Intel 双核 2.93 GHz 的主机上,显示器尺寸为 24 寸,分辨率为 1920×1080.用户使用鼠标来操作电脑.主机运行 Windows 7 操作系统.我们从校园招聘了 8 名被试来参与实验,被试的年龄范围为 22 岁~27 岁,其中 2 名位是女性.所有被试均熟悉图形用户界面和鼠标操作.

如图 4 所示,我们在 Visual Studio 上实现了自适应光标技术.系统通过鼠标钩子获得点击消息,并在每次点击发生时利用 UI Automation 来获取被点击控件的大小和位置信息.支持的控件包括按钮、菜单项和树形栏目等.用户点击其他区域(比如代码编辑区域),系统不予处理.由于 Visual Studio 中不存在用户自定义控件,因此所有的控件都可以通过 UI Automation 来解析.在交互过程中,系统监控控件的访问情况,并在每次用户点击某一控件后,根据控件访问的历史序列来预测用户下一步可能要访问的控件集合,并为这些控件分配卫星光标.为了指示控件的自适应状态(是否被预测为可能访问并分配卫星光标),系统在预测的控件周围附着上虚线框来作为提示,并且当用户移动鼠标时,会以直线来表示卫星光标和其关联的控件之间的对应关系.用户通过点击鼠标中键或按住 Ctrl 键来使用卫星光标进行点击;否则,使用默认光标进行点击.

4.2 任务设计

实验任务要求用户在 Visual Studio 上分别使用普通光标(SDC)和自适应光标(ASC)来完成任务.实验任务中的点击序列来自于实际数据.我们通过记录用户使用 Visual Studio 的控件访问情况,来产生本实验中的点击序列,从而尽可能地保证受控实验的有效性.实际数据表明,用户在 Visual Studio 中使用到的控件个数通常在 6~10 的范围内.这和经验数据[18]相吻合,即通常只有少数功能会被用户使用.

点击序列对于每个用户是不同的,分别取自于实际数据中的某一部分.每个用户的实验任务又分为两个阶段,分别对应普通光标和自适应光标.每个阶段包括 4 个实验块,用户在每个实验块之间可以选择休息.每个实验块里面包含 50 次的点击操作.为了保证对比的公平性,点击序列在两种光标条件下是相同的.而且,普通光标和自适应光标被使用的先后顺序是平衡的:4 名用户先使用 ASC 完成实验,另外 4 名先使用 SDC.用户需要点击 400 次,平均花费 25 分钟来完成本实验.

4.3 实验过程

实验开始前,我们向参与实验的被试讲解实验的目的,并让他们进行若干轮的尝试性点击来熟悉实验界面.用户在该阶段可以比较两种模式选择的操作方法.实验过程中,每次的点击目标通过语音和提示框展现给用户,用户根据提示来点选控件.如果点选失败(选择了与提示不同的控件),系统会要求用户重新点击该控件.实验结束后,我们收集了用户对于自适应光标和普通光标的主观喜好度评价意见.

4.4 实验结果

- 整体性能和使用率

如图 5(a)所示,使用自适应光标完成单次点击所需要的平均时间(861.8ms)显著地小于使用普通光标的平

均时间(1192.3ms),方差分析的显著性水平为 F=764.2,p<0.0001.此外,由于卫星光标总是出现在目标的附件,点击的距离也极大地缩短了,只有使用普通光标的 14.6%.综合所有数据,卫星光标的使用率高达 93.6%,方差为 2.4%.上述数据表明,在所需目标被自适应支持时,用户会以很大地概率使用卫星光标来点击,而且点击效率有明显的提升.

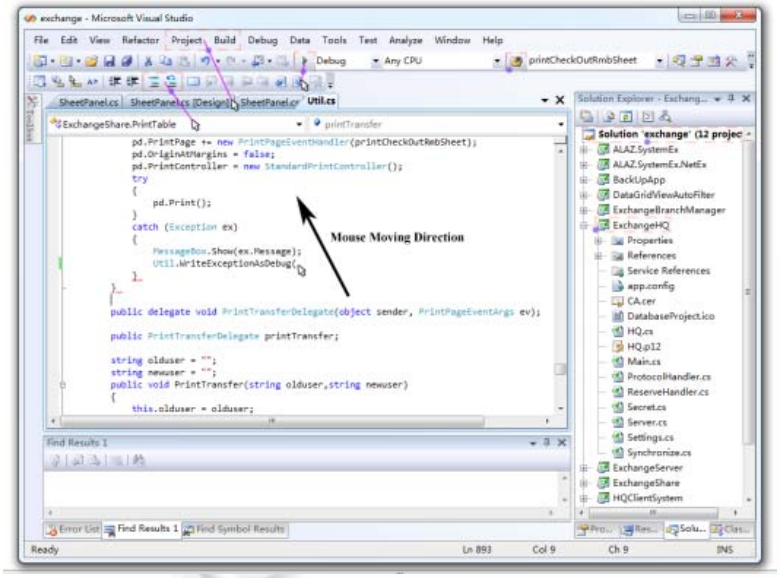


Fig.4 Usage Scenario when the adaptive satellite cursor applying to Visual Stuido

图 4 自适应光标技术应用于 Visual Studio 的情形

- 利益和开销

图 5(b)进一步展示了使用自适应光标带来的利益和开销之间的平衡关系.在实验条件为自适应光标的情况下,用户使用卫星光标的平均点击时间是 707.5ms,使用默认光标的平均点击时间是 1477.5ms.相比于实验条件为普通光标的情况,使用卫星光标使得点击效率提高了 40.7%,使用默认光标使得点击效率降低了 19.3%.卫星光标提高点击效率的原因容易理解,主要是因为鼠标移动的距离得到了大幅缩短(14.6%),而使用默认光标点击时间增大,则因为用户可能先查看目标是否被自适应光标支持,然后再决定使用哪种光标进行点击.这样,当目标没有被支持时,就产生了延时.

- 预测模型准确度和可理解性

在本实验中,由于任务中涉及的控件数量不超过卫星光标的最大数(10),因此访问过的控件当再次需要访问时都会被卫星光标支持,预测准确率达到 82%.实验结束后的调研结果也表明:基于最近策略的预测模型易于理解,用户可以轻松地辨别曾经访问过的控件.而且,提示性点缀(虚线框和连接线)也起到了强化用户对控件自适应状态的认识的作用.对于被卫星光标支持的控件,无论用户使用哪种光标进行点击,都会看到虚线框和连接线.因而,当下一次访问同一控件的时候,用户就更容易预测到它的状态.

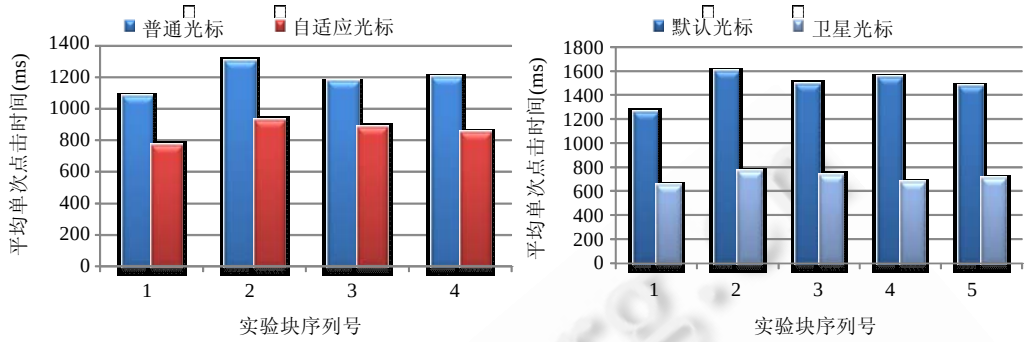
- 模式区分操作

用户对于鼠标中键的使用,在刚开始的尝试中也会有明显的不适应.但是当点击次数达到 20 次~30 次之后,用户便可以在两者之间自如地切换.用户调研结果表明,在熟悉了切换方式之后,用户不会感觉到中键引入的任何操作开销.我们认为,用户在用中键点击之前已经做好了使用卫星光标进行点击的心理准备,这与用户可以自如地使用鼠标左右键的原理类似.

- 主观评价

被试一致表示,他们更倾向于使用自适应光标来获取控件.而且相比于点击时间的缩短,用户更明显地感觉

到点击距离的缩短.此外,为了确定控件的自适应状态,用户并不完全依靠记忆,而是通过查看目标附近是否有卫星光标来判断.只有对于多次访问的控件,用户才会不假思索使用卫星光标来点击.而且,在实验任务中先使用自适应光标的被试都反映,在随后进行的普通光标任务中,会明显觉得普通光标很慢.我们认为,这是由于和使用自适应光标形成直接对比造成的.



(a) 对比实验中,使用普通光标和自适应光标的平均点击时间

(b) 界面在自适应光标支持的情况下,用户通过默认光标和卫星光标的平均点击时间

Fig.5 Results

图 5 实验结果

本实验证明了在预测准确度达到 80%的情况下,用户更喜欢使用自适应光标进行控件获取,并且指点效率可以得到显著的提高.也就是说,相对于操作和认知上的开销,自适应光标带来的利益会更多一些.在本文研究中,我们并没有针对预测准确度不高的任务来进行测试.这是因为之前的研究已经表明,短期内用户任务一般都集中在少数的应用程序上(比如用户可能在 Word 上写作 1 小时,其中只用到浏览器和 PDF 阅读器),而且对于控件的使用都呈现出“最近”性.这样,在多数情况下,基于最近策略的预测算法是可以覆盖用户 80%的访问需求的(本实验中的任务来自于实际交互数据,预测准确度为 82%).而我们也认为,在这类情况下,用户会有更好的心理准备来使用自适应光标,因为持续的访问可以强化用户对控件自适应状态的认识.对于用户总是需要访问新的控件的情况,用户依然可以使用普通光标来进行点击,而不会影响点击效率.

4.5 讨 论

本实验以 Visual Studio 为对象考察了自适应光标的性能.从界面上看,Visual Studio 代表了典型的界面形式(具有内容编辑区和菜单、工具栏区).从交互任务上看,用户只使用到了很有限的控件集合(6~10 个),这与已有研究中反映的控件使用情形相吻合(比如在 MS Word 中,仅仅 3.3%的一级命令会经常被用户访问[18]).因此我们认为,本实验的结果可以推广到普通的程序界面上.同时我们也看到,正是由于软件界面提供的控件集合远大于用户交互任务所需要的控件集合,自适应光标才有了发挥作用的空间.对于用户可能访问大量的控件的情形(比如网页中不断更新的超连接),自适应光标并不能起到作用.

5 结论和未来工作

本文提出的自适应光标技术,通过将卫星光标和自适应界面的思想结合起来,既解决了以往的自适应界面通过改变控件布局造成的认知成本增大的问题,也克服了卫星光标自身性能受制于控件数量的问题.本文首先提出了界面效率的定量评价模型,然后阐述了自适应光标的优化原理,并且对自适应光标的优化性能进行了分析.在用户交互任务以控件的访问序列来表示的前提下,自适应光标技术可以使输入效率达到最优.接着,本文详细介绍了自适应光标技术的交互设计细节.这些细节都是决定自适应光标是否可以真正提高界面效率的重要因素.其中,我们着重分析了自适应光标所带来的利益和开销.通过实验,我们证明了在预测准确度较高(本实

验中为 82%)的情况下,使用自适应光标确实可以提高界面效率,并且用户都更倾向于使用自适应光标来获取控件.

本研究的未来工作主要集中在如何将自适应光标集成入操作系统(比如 Windows),作为普适的点击增强技术来提高界面效率,并由此来考察预测模型对于实际交互任务的预测能力.这里面会涉及到如何更有效地获取用户点击事件和提取控件信息.我们认为,由于自适应光标技术不仅可以提高界面效率,还可以大幅度缩短人们移动鼠标的距离,因此对于长期工作在电脑前的用户有着重要意义.

References:

[1] Yu C, Shi YC, Balakrishnan R, Meng XL, Suo Y, Fan MM, Qin YQ. The satellite cursor: Achieving MAGIC pointing without gaze tracking using multiple cursors. In: Proc. of the UIST. 2010. 163−173.

[2] Fitts PM. The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology, 1954,47(6):381−391. [doi: 10.1037/h0055392]

[3] Balakrishnan R. “Beating”Fitts’ law: Virtual enhancements for pointing facilitation. Int’l Journal of Human-Computer Studies, 2004,61(6):857−874. http://dx.doi.org/10.1016/j.ijhcs.2004.09.002

[4] Grossman T, Balakrishnan R. The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor’s activation area. In: Proc. of the CHI. 2005. 281−290. [doi: 10.1145/1054972.1055012]

[5] Baudisch P, Cutrell E, Robbins D, Czerwinski M, Tandler P, Bederson B, Zierlinger A. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. In: Proc. of the Interact. 2003. 57−64.

[6] Blanch R, Guiard Y, Beaudouin-Lafon M. Semantic pointing: Improving target acquisition with control-display ratio adaptation. In: Proc. of the CHI. 2004. 519−525. [doi: 10.1145/985692.985758]

[7] Findlater L, Moffatt K, McGrenere J, Dawson J. Ephemeral adaptation: The use of gradual onset to improve menu selection performance. In: Proc. of the CHI. 2009. 1655−1664. [doi: 10.1145/1518701.1518956]

[8] Sears A, Shneiderman B. Split menus: Effectively using selection frequency to organize menus. ACM Trans. on Computer-Human Interaction, 1994,1(1):27−51. [doi: 10.1145/174630.174632]

[9] Findlater L, McGrenere J. A comparison of static, adaptive, and adaptable menus. In: Proc. of the CHI. 2004. 89−96. [doi: 10.1145/985692.985704]

[10] Gajos KZ, Everitt K, Tan DS, Czerwinski M, Weld DS. Predictability and accuracy in adaptive user interfaces. In: Proc. of the CHI. 2008. 1271−1274. [doi: 10.1145/1357054.1357252]

[11] Findlater L, Gajos KZ. Design space and evaluation challenges of adaptive graphical user interfaces. AI Magazine, 2009,30(4): 68−73.

[12] Card SK, Moran TP, Newell A. The Psychology of Human-Computer Interaction. London: Lawrence Erbaum Associates, 1983.

[13] UI automation overview. http://msdn.microsoft.com/en-us/library/ms747327.aspx

[14] Dixon M, Fogarty J. Prefab: Implementing advanced behaviors using pixel-based reverse engineering of interface structure. In: Proc. of the CHI. 2010. 1525−1534. [doi: 10.1145/1753326.1753554]

[15] Alexander J, Cockburn A, Fitchett S, Gutwin C, Greenberg S. Revisiting read wear: Analysis, design, and evaluation of a Footprints scrollbar. In: Proc. of the CHI. 2009. 1665−1674. [doi: 10.1145/1518701.1518957]

[16] Greenberg S, Witten IH. Supporting command reuse: Empirical foundations and principles. Int’l Journal of Man-Machine Studies, 1993,39(3):353−390. [doi: 10.1006/imms.1993.1065]

[17] Tauscher L, Greenberg S. Revisitation patterns in World Wide Web navigation. In: Proc. of the CHI. 1997. 399−406. [doi: 10.1145/258549.258816]

[18] McGrenere J, Moore G. Are we all in the same “Bloat”? In: Proc. of the Graphics Interface. 2000. 187−196.

喻纯(1984－),男,湖北荆州人,博士生,CCF 会员,主要研究领域为人机交互.

史元春(1967－),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为人机交互,普适计算,网络多媒体.