



(12)发明专利

(10)授权公告号 CN 103677592 B

(45)授权公告日 2018.01.05

(21)申请号 201210321532.6

(22)申请日 2012.09.03

(65)同一申请的已公布的文献号  
申请公布号 CN 103677592 A

(43)申请公布日 2014.03.26

(73)专利权人 腾讯科技(深圳)有限公司  
地址 518044 广东省深圳市福田区振兴路  
赛格科技园2栋东403室

(72)发明人 刘明军

(74)专利代理机构 广州华进联合专利商标代理  
有限公司 44224  
代理人 何平 曾旻辉

(51)Int.Cl.  
G06F 3/0488(2013.01)

(56)对比文件

CN 101052939 A,2007.10.10,  
CN 102629166 A,2012.08.08,  
CN 101819498 A,2010.09.01,  
TW 200715192 A,2007.04.16,  
US 2007273668 A1,2007.11.29,

审查员 于芝枝

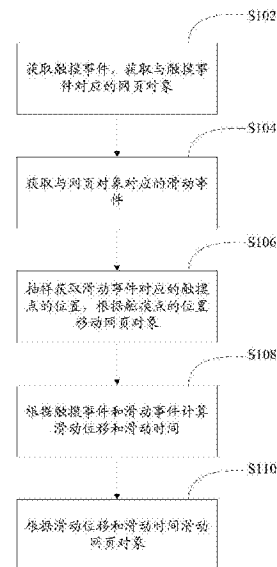
权利要求书2页 说明书10页 附图2页

(54)发明名称

网页内滑动显示网页对象的方法及装置

(57)摘要

一种网页内滑动显示网页对象的方法,包括:获取触摸事件,获取与所述触摸事件对应的网页对象;获取与所述网页对象对应的滑动事件;抽样获取所述滑动事件对应的触摸点的位置,根据所述触摸点的位置移动所述网页对象;根据所述触摸事件和所述滑动事件计算滑动位移和滑动时间;根据所述滑动位移和滑动时间滑动所述网页对象。此外,还提供了一种网页内滑动显示网页对象的装置。上述网页内滑动显示网页对象的方法和装置可以提高用户操作的便利性。



1. 一种网页内滑动显示网页对象的方法,包括:
  - 获取触摸事件,获取与所述触摸事件对应的网页对象;
  - 获取与所述网页对象对应的滑动事件;
  - 抽样获取所述滑动事件对应的触摸点的位置,根据所述触摸点的位置移动所述网页对象;
  - 根据所述触摸事件和所述滑动事件计算滑动位移和滑动时间;
  - 根据所述滑动位移和滑动时间滑动所述网页对象;
  - 所述根据所述滑动位移和滑动时间滑动所述网页对象的步骤为:
    - 根据所述滑动位移和滑动时间计算滑动系数;
    - 根据所述滑动位移和所述滑动系数计算所述网页对象的期望位移;所述期望位移等于所述滑动位移和所述滑动系数的乘积;
    - 根据所述期望位移滑动所述网页对象。
2. 根据权利要求1所述的网页内滑动显示网页对象的方法,其特征在于,所述根据所述触摸事件和所述滑动事件计算滑动位移和滑动时间的步骤为:
  - 获取所述触摸事件对应的起始触摸点的位置以及起始触摸点对应的的时间戳;
  - 获取所述滑动事件对应的终止触摸点的位置以及终止触摸点对应的的时间戳;
  - 根据所述起始触摸点和终止触摸点的位置计算所述滑动位移,根据所述起始触摸点和终止触摸点对应的的时间戳计算所述滑动时间。
3. 根据权利要求1所述的网页内滑动显示网页对象的方法,其特征在于,所述根据所述期望位移滑动所述网页对象的步骤为:
  - 获取所述期望位移对应的预设的阈值区间;
  - 获取所述阈值区间对应的延时参数;
  - 根据所述期望位移和所述延时参数通过调用内置浏览器函数滑动所述网页对象。
4. 根据权利要求1所述的网页内滑动显示网页对象的方法,其特征在于,所述方法还包括:
  - 获取所述网页对象对应的可视窗口;
  - 所述根据期望位移滑动网页对象的步骤之前还包括:
    - 获取所述网页对象的位置;
    - 根据所述期望位移获取对应的可视窗口的边界位置;
    - 判断所述可视窗口的边界位置与所述网页对象的位置之差是否小于或等于所述期望位移,若是,则根据所述网页对象的位置和所述可视窗口的边界位置更新期望位移。
5. 根据权利要求4所述的网页内滑动显示网页对象的方法,其特征在于,所述可视窗口还包括内嵌或外延的边界缓冲区;
  - 所述根据所述触摸点的位置移动所述网页对象的步骤为:
    - 获取所述网页对象的位置;
    - 判断所述网页对象的位置是否超出所述可视窗口或者所述边界缓冲区的边缘,若是,则停止移动所述网页对象;否则,继续执行所述根据所述触摸点的位置移动所述网页对象的步骤。
6. 根据权利要求5所述的网页内滑动显示网页对象的方法,其特征在于,所述根据所述

触摸点的位置移动所述网页对象的步骤之后还包括：

获取所述网页对象的位置；

判断所述网页对象的位置是否位于所述边界缓冲区中，若是，则滑动所述网页对象至所述边界缓冲区的边缘；否则，继续执行所述根据所述触摸事件和所述滑动事件计算滑动位移和滑动时间的步骤。

7. 一种网页内滑动显示网页对象的装置，其特征在于，包括：

触摸事件获取模块，用于获取触摸事件，获取与所述触摸事件对应的网页对象；

滑动事件获取模块，用于获取与所述网页对象对应的滑动事件；

同步滑动模块，用于抽样获取所述滑动事件对应的触摸点的位置，根据所述触摸点的位置移动所述网页对象；

参数计算模块，用于根据所述触摸事件和所述滑动事件计算滑动位移和滑动时间；

惯性滑动模块，用于根据所述滑动位移和滑动时间滑动所述网页对象；

所述惯性滑动模块还用于根据所述滑动位移和滑动时间计算滑动系数，根据所述滑动位移和所述滑动系数计算所述网页对象的期望位移，根据所述期望位移滑动所述网页对象；所述期望位移等于所述滑动位移和所述滑动系数的乘积。

8. 根据权利要求7所述的网页内滑动显示网页对象的装置，其特征在于，所述参数计算模块还用于获取所述触摸事件对应的起始触摸点的位置以及起始触摸点对应的的时间戳，获取所述滑动事件对应的终止触摸点的位置以及终止触摸点对应的的时间戳，根据所述起始触摸点和终止触摸点的位置计算所述滑动位移，根据所述起始触摸点和终止触摸点对应的的时间戳计算所述滑动时间。

9. 根据权利要求7所述的网页内滑动显示网页对象的装置，其特征在于，所述惯性滑动模块还用于根据所述滑动位移和滑动时间计算滑动系数，根据所述滑动位移和所述滑动系数计算所述网页对象的期望位移，根据所述期望位移滑动所述网页对象。

10. 根据权利要求7所述的网页内滑动显示网页对象的装置，其特征在于，所述装置还包括可视窗口获取模块，用于获取所述网页对象对应的可视窗口；

所述惯性滑动模块还用于获取所述网页对象的位置，根据所述期望位移获取对应的可视窗口的边界位置，在所述可视窗口的边界位置与所述网页对象的位置之差小于或等于所述期望位移时，根据所述网页对象的位置和所述可视窗口的边界位置更新期望位移。

11. 根据权利要求10所述的网页内滑动显示网页对象的装置，其特征在于，所述可视窗口还包括内嵌或外延的边界缓冲区；

所述同步滑动模块还用于获取所述网页对象的位置，在所述网页对象的位置超出所述可视窗口或者所述边界缓冲区的边缘时，停止移动所述网页对象。

12. 根据权利要求11所述的网页内滑动显示网页对象的装置，其特征在于，所述装置还包括边界处理模块，用于获取所述网页对象的位置，在所述网页对象的位置位于所述边界缓冲区内时，滑动所述网页对象至所述边界缓冲区的边缘。

## 网页内滑动显示网页对象的方法及装置

### 技术领域

[0001] 本发明涉及计算机技术领域,特别是涉及一种网页内滑动显示网页对象的方法及装置。

### 背景技术

[0002] 随着触摸屏技术和移动终端技术的发展,智能移动设备已经成为当今市场上的主流,包括智能手机、掌上电脑、平板电脑、电子书等。用户可通过智能移动设备上自带的移动版浏览器在任何有无线网络覆盖的区域自由地浏览网页,使得人与人之间的沟通更加方便。

[0003] 传统技术中,用户在使用移动版浏览器上网时,可通过在触摸屏上滑动触摸来实现网页的整体滑动。网页中还包含网页对象,如图片、文本块、表单等。网页对象可在其所处的可视窗口(如由frame、div、span、input等标签定义的网页显示区域)中滑动。

[0004] 例如,可在网页中通过div标签定义可视窗口,设置div标签的overflow属性为auto,当该div标签中包含的网页对象的大小大于该div的可视窗口的大小时,即通过添加滚动条来显示该网页对象。用户可通过对该网页对象滑动触摸来在可视窗口中滑动显示该网页对象的全部内容。

[0005] 然而,传统技术中,用户在滑动触摸网页对象时,该网页对象只能跟随触摸点的移动而移动,也就是说网页对象的最大滑动距离(考虑边界不能滑的情况)与触摸点的移动距离相同。而由于可视窗口的大小有限,因此,在网页对象的大小较大时,用户需要滑动多次才能将网页对象滑到指定的位置来展示相应的内容,从而使得用户操作的便利性不足。

### 发明内容

[0006] 基于此,有必要提供一种能提高用户操作的便利性的网页内滑动显示网页对象的方法。

[0007] 一种网页内滑动显示网页对象的方法,包括:

[0008] 获取触摸事件,获取与所述触摸事件对应的网页对象;

[0009] 获取与所述网页对象对应的滑动事件;

[0010] 抽样获取所述滑动事件对应的触摸点的位置,根据所述触摸点的位置移动所述网页对象;

[0011] 根据所述触摸事件和所述滑动事件计算滑动位移和滑动时间;

[0012] 根据所述滑动位移和滑动时间滑动所述网页对象。

[0013] 此外,还有必要提供一种能提高用户操作的便利性的网页内滑动显示网页对象的装置。

[0014] 一种网页内滑动显示网页对象的装置,包括:

[0015] 触摸事件获取模块,用于获取触摸事件,获取与所述触摸事件对应的网页对象;

[0016] 滑动事件获取模块,用于获取与所述网页对象对应的滑动事件;

[0017] 同步滑动模块,用于抽样获取所述滑动事件对应的触摸点的位置,根据所述触摸点的位置移动所述网页对象;

[0018] 参数计算模块,用于根据所述触摸事件和所述滑动事件计算滑动位移和滑动时间;

[0019] 惯性滑动模块,用于根据所述滑动位移和滑动时间滑动所述网页对象。

[0020] 上述网页内滑动显示网页对象的方法和装置,先获取网页对象的触摸事件和滑动事件,并根据滑动事件对应的触摸点移动网页对象,然后在同步移动网页对象之后还根据触摸事件和滑动事件计算滑动位移和滑动时间,并根据滑动位移和滑动时间继续滑动网页对象。用户在滑动网页对象时,可使网页对象获得比滑动位移更大的移动距离,且可通过控制滑动操作的速度来控制网页对象的移动距离。与传统技术相比,用户只需滑动较少的次数即可将网页对象滑动到特定的位置,从而提高了操作的便利性。

## 附图说明

[0021] 图1为一个实施例中网页内滑动显示网页对象的方法的流程图;

[0022] 图2为一个实施例中网页内滑动显示网页对象的装置的结构示意图;

[0023] 图3为另一个实施例中网页内滑动显示网页对象的装置的结构示意图。

## 具体实施方式

[0024] 在一个实施例中,如图1所示,一种网页内滑动显示网页对象的方法,包括:

[0025] 步骤S102,获取触摸事件,获取与触摸事件对应的网页对象。

[0026] 触摸事件即触摸屏检测到的触摸操作时触发的事件。当用户用手指接触触摸屏表面时,即会触发触摸事件。触摸事件可与终端设备上当前运行的应用程序绑定。例如,若用户在网页上的某个图片对象所处的可视窗口内进行按压时,浏览器可根据触摸点在触摸屏上的位置来定位到该图片对象,并将该图片对象与该触摸事件绑定,然后抛出该触摸事件由相应的事件处理函数处理。

[0027] 在本实施例中,可通过脚本函数(例如,由Javascript语言定义的脚本函数)获取触摸事件,提取触摸事件中包含的网页对象标识,根据提取得到的网页对象标识获取网页对象。

[0028] 步骤S104,获取与网页对象对应的滑动事件。

[0029] 滑动事件即触摸屏检测到触摸点的位置连续变化时触发的事件。滑动事件可与终端设备上当前运行的应用程序绑定。例如,当用户在网页上的某个图片对象所处的可视窗口内滑动时,浏览器可根据触摸点的位置定位到该图片对象,并将该图片对象与该滑动事件绑定,然后抛出滑动事件由相应的事件处理函数处理。

[0030] 需要说明的是,在滑动过程中,若触摸点的位置移出网页对象A所处的可视窗口进入网页对象B所处的可视窗口,则网页对象A停止抛出滑动事件。而当触摸点进入网页对象B所处的可视窗口时,触发与网页对象B对应的触摸事件,当触摸点继续在网页对象B所处的可视窗口内滑动时,触发与网页对象B对应的滑动事件。

[0031] 在滑动过程中,若触摸点在网页对象A所处的可视窗口内离开触摸屏,则网页对象A停止抛出滑动事件。

[0032] 在获取到与网页对象对应的触摸事件后,若继续获取到与该网页对象对应的滑动事件,则表示用户在接触触摸屏后紧接着执行了滑动操作;若在获取到网页对象的触摸事件之后检测到该网页对象的触摸停止事件,则表示,本次输入中,用户未在触摸屏上进行滑动操作,而只是在触摸屏上进行了点击操作。例如,网页对象中可能包括链接、按钮等。用户可通过对链接进行点击操作来触发页面跳转指令。

[0033] 步骤S106,抽样获取滑动事件对应的触摸点的位置,根据触摸点的位置移动网页对象。

[0034] 在一个实施例中,滑动事件对应的触摸点的位置可通过像素坐标表示,例如,可通过`top:10px,left:20px`表示触摸点的位置。在本实施例中,根据触摸点的位置移动网页对象的步骤可具体为:根据抽样获取的触摸点的像素坐标计算触摸点位置变化的像素增量,根据该像素增量设置网页对象的位置。

[0035] 例如,若抽样获取的触摸点的像素坐标的变化增量为`top:2px`,则可在网页对象的CSS(Cascading Style Sheet,层叠样式表,用于定义网页对象的样式)属性中将`top`属性值增加`2px`。浏览器中设置有CSS属性修改监听器,当网页对象的`top`属性改变时,自动在可视窗口中根据更新后的`top`属性调整网页对象的位置。由于抽样时间较短,因此整体上就展现出网页对象跟随触摸点同步移动的效果。

[0036] 步骤S108,根据触摸事件和滑动事件计算滑动位移和滑动时间。

[0037] 滑动位移即触摸点在网页对象所处的可视窗口内滑动的位移。滑动时间即由触摸事件被触发开始,到滑动事件结束的时间长度。

[0038] 在一个实施例中,根据触摸事件和滑动事件计算滑动位移和滑动时间的步骤为:

[0039] 获取触摸事件对应的起始触摸点的位置以及起始触摸点对应的的时间戳,获取滑动事件对应的终止触摸点的位置以及终止触摸点对应的的时间戳,根据起始触摸点和终止触摸点的位置计算滑动位移,根据起始触摸点和终止触摸点对应的的时间戳计算滑动时间。

[0040] 起始触摸点即网页对象的可视窗口内触发触摸事件时对应的触摸点。终止触摸点即为滑动操作在离开网页对象的可视窗口时的触摸点。终止触摸点可以是滑出可视窗口时的触摸点,也可以是在可视窗口内离开触摸屏时的触摸点。

[0041] 起始触摸点和终止触摸点的位置可通过像素坐标表示。滑动位移可通过水平和垂直方向上的像素坐标表示。例如,滑动位移可表示为`top:-10px,left:5px`,即意味着水平方向向右(相对于正常显示的网页)滑动了`5px`,垂直方向向上滑动了`10px`。

[0042] 步骤S110,根据滑动位移和滑动时间滑动网页对象。

[0043] 需要说明的是,根据滑动位移和滑动时间滑动网页对象的步骤在执行时,触摸点已离开触摸屏。也就是说,用户通过滑动触摸移动网页对象之后,网页对象还能够以“惯性”继续滑动相应的距离。

[0044] 在一个实施例中,根据滑动位移和滑动时间滑动网页对象的步骤为:

[0045] 根据滑动位移和滑动时间计算滑动系数,根据滑动位移和滑动系数计算网页对象的期望位移,根据期望位移滑动网页对象。

[0046] 滑动系数用于表示触摸点在触摸屏上的滑动速度。在一个实施例中,可根据公式:

[0047] 
$$V = a \times |\text{distance}| \div \text{mtime}$$

$$[0048] \quad F = \begin{cases} V^c (V \leq b) \\ V^d (V > b) \end{cases}$$

[0049] 计算滑动系数。其中，V为中间结果，表示滑动速度；a为预设的比例系数；distance为滑动位移；mtime为滑动时间；b为预设的门限系数，c和d分别为预设的幂系数；F为滑动系数。

[0050] 也就是说，当滑动速度V小于等于门限系数b时，通过 $V^c$ 计算滑动系数F；当滑动速度V大于门限系数b时，通过 $V^d$ 计算滑动系数F。

[0051] 期望位移即触摸点在离开网页对象的可视窗口后，网页对象会继续滑动的位移。

[0052] 在一个实施例中，可根据公式：

$$[0053] \quad S = \text{distance} \times F$$

[0054] 计算期望位移。其中S为期望位移，distance为滑动位移，F为滑动系数。也就是说，触摸点滑的越快，则期望位移越大，即滑的越远。

[0055] 需要说明的是，若滑动位移既包含水平方向的像素点变化又包含垂直方向的像素点变化时，可在水平方向和垂直方向分别计算期望位移，即滑动位移的水平分量影响网页对象在水平方向上的滑动，滑动位移的垂直分量影响网页对象在垂直方向上的滑动。

[0056] 在一个实施例中，还可获取网页对象对应的可视窗口。根据期望位移滑动网页对象的步骤之前还包括：

[0057] 获取网页对象的位置，根据期望位移获取对应的可视窗口的边界位置，判断可视窗口的边界位置与网页对象的位置之差是否小于或等于期望位移，若是，则根据网页对象的位置和可视窗口的边界位置更新期望位移。

[0058] 根据期望位移获取对应的可视窗口的边界位置的步骤可具体为：获取期望位移的方向，根据期望位移的方向获取可视窗口在该方向上的边界位置。

[0059] 判断可视窗口的边界位置与网页对象的位置之差是否大于期望位移的步骤即为判断期望位移是否大于可视窗口内的可滑动距离，例如，若网页对象的位置在触摸点离开触摸屏时已经位于可视窗口的底部，则期望位移大于可视窗口内的可滑动距离。

[0060] 若期望位移大于网页对象的位置和可视窗口的边界位置的差值，则可根据该差值更新期望位移，并根据更新后的期望位移滑动网页对象。也就是说，当触摸滑动的速度较快，而可视窗口中剩余的可滑动距离不够时，将网页对象滑到可视窗口的边界处。

[0061] 在本实施例中，进一步的，根据期望位移滑动网页对象的步骤为：

[0062] 获取期望位移对应的预设的阈值区间，获取阈值区间对应的延时参数，根据期望位移和延时参数通过调用内置浏览器函数滑动网页对象。

[0063] 可预先设置阈值区间，并设置与阈值区间对应的延时参数值。例如，可设置0至100px的阈值区间对应的延时参数为100ms，100至300px的阈值区间对应的延时参数为200ms，高于300px的阈值区间对应的延时参数为300ms。在通过计算得到期望位移后，可根据期望位移的大小判断期望位移所处阈值区间，并获取该阈值区间对应的延迟参数。

[0064] 在本实施例中，根据期望位移和延时参数通过调用内置浏览器函数滑动网页对象的步骤可具体为：

[0065] 根据延迟参数设置网页对象的CSS属性的-webkit-transition属性；

[0066] 根据期望位移设置网页对象的CSS属性的top属性和/或left属性;

[0067] 通过浏览器的CSS事件处理函数设置网页对象的位置。

[0068] 网页对象的CSS属性中的-webkit-transition为过渡属性,用于使得网页对象的其他属性的变化由阶梯形的瞬间变化更改为渐进式的逐渐变化。当根据延迟参数设置网页对象的CSS属性的-webkit-transition属性后,再将网页对象的位置设置到目标位置时,网页对象不会立即出现在目标位置,而是逐渐地移动到目标位置。

[0069] 例如,若根据延迟参数将网页对象的CSS属性的-webkit-transition值设置为delay 200ms,则根据期望位移设置网页对象的CSS属性的top属性和/或left属性后,网页对象不会立即移动到该top属性值和/或left属性值对应的位置,而会经过200ms的延迟逐渐移动到该位置。

[0070] 通过设置网页对象的CSS属性来滑动网页对象,可以使网页对象的滑动附带减速滑动的动画效果,从而具有较好的用户体验。

[0071] 在另一个实施例中,还可根据期望位移和延时参数通过脚本函数滑动网页对象。例如,基于Javascript语言的JQuery框架中提供了移动网页对象的脚本函数,可通过调用该脚本函数,并将期望位移和延时参数作为参数传递给该脚本函数来滑动网页。

[0072] 在一个实施例中,可视窗口还包括内嵌或外延的边界缓冲区。根据触摸点的位置移动网页对象的步骤可具体为:

[0073] 获取网页对象的位置,判断网页对象的位置是否超出可视窗口或者边界缓冲区的边缘,若是,则停止移动网页对象;否则,继续执行根据触摸点的位置移动网页对象的步骤。

[0074] 边界缓冲区可通过CSS设置,即可视窗口的留白区域,其宽度可由CSS属性中的margin或padding属性值决定。边界缓冲区内不显示网页对象,但在边界缓冲区内触摸和滑动操作仍能触发网页对象相应的事件。当触摸滑动操作将网页对象滑至可视窗口的边缘时边界缓冲区时,可继续移动网页对象的位置至边界缓冲区中,但进入边界缓冲区的网页对象的部分为不可见,即被遮挡。

[0075] 可获取网页对象的位置、可视窗口的边界位置以及边界缓冲区的宽度,通过判断网页对象的位置与可视窗口的边界位置的差值是否小于或等于边界缓冲区的宽度判断网页对象的位置是否超出可视窗口或者边界缓冲区的边缘。

[0076] 也就是说,当网页对象的位置处于边界缓冲区的边缘时,不再移动网页对象,使得网页对象只能进入边界缓冲区一部分,从而可在触摸操作将网页对象滑到可视窗口边缘时造成“滑不动”的效果。

[0077] 进一步的,根据触摸事件和滑动事件计算滑动位移和滑动时间的步骤之后还包括:

[0078] 获取网页对象的位置;判断网页对象的位置是否位于边界缓冲区中,若是,则滑动网页对象至可视窗口内的边界缓冲区的边缘;否则,继续执行根据触摸事件和滑动事件计算滑动位移和滑动时间的步骤。

[0079] 根据触摸事件和滑动事件计算滑动位移和滑动时间的步骤之后,触摸滑动操作结束,触摸点离开网页对象对应的可视窗口。若此时获取到的网页对象的位置位于边界缓冲区内,则将网页对象滑至可视窗口内边界缓冲区的边缘,且停止后续的滑动网页对象的操作;若网页对象的位置不在边界缓冲区中,则计算滑动位移和滑动时间,用于继续滑动网页



对象。

[0080] 在本实施例中,滑动网页对象至可视窗口内的边界缓冲区的边缘的步骤可具体为:

[0081] 根据预设的弹起参数设置网页对象的CSS属性的-webkit-transition值;获取可视窗口内的边界缓冲区的边缘的位置;根据获取的可视窗口内的边界缓冲区的边缘的位置设置网页对象的CSS属性值;通过CSS事件处理函数滑动网页对象至可视窗口内的边界缓冲区的边缘。

[0082] 也就是说,当停止滑动触摸或滑出网页对象对应的可视窗口时,若网页对象的位置停留在可视窗口的边缘缓冲区中时,可将网页对象移动至可视窗口边缘全部显示,移动的速度和动画效果由预设的弹起参数决定,从而给用户带来更好的用户体验。

[0083] 在一个实施例中,如图2所示,一种网页内滑动显示网页对象的装置,包括:触摸事件获取模块102、滑动事件获取模块104、同步滑动模块106、参数计算模块108以及惯性滑动模块110,其中:

[0084] 触摸事件获取模块102,用于获取触摸事件,获取与触摸事件对应的网页对象。

[0085] 触摸事件即触摸屏检测到的触摸操作时触发的事件。当用户用手指接触触摸屏表面时,即会触发触摸事件。触摸事件可与终端设备上当前运行的应用程序绑定。例如,若用户在网页上的某个图片对象所处的可视窗口内进行按压时,浏览器可根据触摸点在触摸屏上所处的位置来定位到该图片对象,并将该图片对象与该触摸事件绑定,然后抛出该触摸事件由相应的事件处理函数处理。

[0086] 在本实施例中,触摸事件获取模块102可用于通过脚本函数(例如,由Javascript语言定义的脚本函数)获取触摸事件,提取触摸事件中包含的网页对象标识,根据提取得到的网页对象标识获取网页对象。

[0087] 滑动事件获取模块104,用于获取与网页对象对应的滑动事件。

[0088] 滑动事件即触摸屏检测到触摸点的位置连续变化时触发的事件。滑动事件可与终端设备上当前运行的应用程序绑定。例如,当用户在网页上的某个图片对象所处的可视窗口内滑动时,浏览器可根据触摸点的位置定位到该图片对象,并将该图片对象与该滑动事件绑定,然后抛出滑动事件由相应的事件处理函数处理。

[0089] 需要说明的是,在滑动过程中,若触摸点的位置移出网页对象A所处的可视窗口进入网页对象B所处的可视窗口,则网页对象A停止抛出滑动事件。而当触摸点进入网页对象B所处的可视窗口时,触发与网页对象B对应的触摸事件,当触摸点继续在网页对象B所处的可视窗口内滑动时,触发与网页对象B对应的滑动事件。

[0090] 在滑动过程中,若触摸点在网页对象A所处的可视窗口内离开触摸屏,则网页对象A停止抛出滑动事件。

[0091] 在触摸事件获取模块102获取到与网页对象对应的触摸事件后,若滑动事件获取模块104继续获取到与该网页对象对应的滑动事件,则表示用户在接触触摸屏后紧接着执行了滑动操作;若在获取到网页对象的触摸事件之后检测到该网页对象的触摸停止事件,则表示,本次输入中,用户未在触摸屏上进行滑动操作,而只是在触摸屏上进行了点击操作。例如,网页对象中可能包括链接、按钮等。用户可通过对链接进行点击操作来触发页面跳转指令。

[0092] 同步滑动模块106,用于抽样获取滑动事件对应的触摸点的位置,根据触摸点的位置移动网页对象。

[0093] 在一个实施例中,滑动事件对应的触摸点的位置可通过像素坐标表示,例如,可通过top:10px,left:20px表示触摸点的位置。在本实施例中,同步滑动模块106可用于根据抽样获取的触摸点的像素坐标计算触摸点位置变化的像素增量,根据该像素增量设置网页对象的位置。

[0094] 例如,若抽样获取的触摸点的像素坐标的变化增量为top:2px,则可在网页对象的CSS(Cascading Style Sheet,层叠样式表,用于定义网页对象的样式)属性中将top属性值增加2px。浏览器中设置有CSS属性修改监听器,当网页对象的top属性改变时,自动在可视窗口中根据更新后的top属性调整网页对象的位置。由于抽样时间较短,因此整体上就展现出网页对象跟随触摸点同步移动的效果。

[0095] 参数计算模块108,用于根据触摸事件和滑动事件计算滑动位移和滑动时间。

[0096] 滑动位移即触摸点在网页对象所处的可视窗口内滑动的位移。滑动时间即由触摸事件被触发开始,到滑动事件结束的时间长度。

[0097] 在一个实施例中,参数计算模块108可用于获取触摸事件对应的起始触摸点的位置以及起始触摸点对应的的时间戳,获取滑动事件对应的终止触摸点的位置以及终止触摸点对应的的时间戳,根据起始触摸点和终止触摸点的位置计算滑动位移,根据起始触摸点和终止触摸点对应的的时间戳计算滑动时间。

[0098] 起始触摸点即网页对象的可视窗口内触发触摸事件时对应的触摸点。终止触摸点即为滑动操作在离开网页对象的可视窗口时的触摸点。终止触摸点可以是滑出可视窗口时的触摸点,也可以是在可视窗口内离开触摸屏的触摸点。

[0099] 起始触摸点和终止触摸点的位置可通过像素坐标表示。滑动位移可通过水平和垂直方向上的像素坐标表示。例如,滑动位移可表示为top:-10px,left:5px,即意味着水平方向向右(相对于正常显示的网页)滑动了5px,垂直方向向上滑动了10px。

[0100] 惯性滑动模块110,用于根据滑动位移和滑动时间滑动网页对象。

[0101] 需要说明的是,同步滑动模块106用于使网页对象跟随触摸点滑动,惯性滑动模块110用于在滑动操作结束,触摸点已离开触摸屏后继续滑动网页对象。也就是说,惯性滑动模块110可用于使网页对象在跟随触摸点滑动之后,还能够以“惯性”继续滑动相应的距离。

[0102] 在一个实施例中,惯性滑动模块110可用于根据滑动位移和滑动时间计算滑动系数,根据滑动位移和滑动系数计算网页对象的期望位移,根据期望位移滑动网页对象。

[0103] 滑动系数用于表示触摸点在触摸屏上的滑动速度。在一个实施例中,惯性滑动模块110可用于根据公式:

$$[0104] \quad V = a \times |\text{distance}| \div \text{mtime}$$

$$[0105] \quad F = \begin{cases} V^c (V \leq b) \\ V^d (V > b) \end{cases}$$

[0106] 计算滑动系数。其中,V为中间结果,表示滑动速度;a为预设的比例系数;distance为滑动位移;mtime为滑动时间;b为预设的门限系数,c和d分别为预设的幂系数;F为滑动系数。

[0107] 也就是说,当滑动速度 $V$ 小于等于门限系数 $b$ 时,惯性滑动模块110可用于通过 $V^c$ 计算滑动系数 $F$ ;当滑动速度 $V$ 大于门限系数 $b$ 时,惯性滑动模块110可用于通过 $V^d$ 计算滑动系数 $F$ 。

[0108] 期望位移即触摸点在离开网页对象的可视窗口后,网页对象会继续滑动的位移。

[0109] 在一个实施例中,惯性滑动模块110可用于根据公式:

[0110]  $S = \text{distance} \times F$

[0111] 计算期望位移。其中 $S$ 为期望位移, $\text{distance}$ 为滑动位移, $F$ 为滑动系数。也就是说,触摸点滑的越快,则期望位移越大,即滑的越远。

[0112] 需要说明的是,若滑动位移既包含水平方向的像素点变化又包含垂直方向的像素点变化时,惯性滑动模块110可用于在水平方向和垂直方向分别计算期望位移,即滑动位移的水平分量影响网页对象在水平方向上的滑动,滑动位移的垂直分量影响网页对象在垂直方向上的滑动。

[0113] 在一个实施例中,如图3所示,还包括可视窗口获取模块112,用于获取网页对象对应的可视窗口。

[0114] 惯性滑动模块110还可用于获取网页对象的位置,根据期望位移获取对应的可视窗口的边界位置,在可视窗口的边界位置与网页对象的位置之差小于或等于期望位移时,根据网页对象的位置和可视窗口的边界位置更新期望位移。

[0115] 惯性滑动模块110还可用于获取期望位移的方向,根据期望位移的方向获取可视窗口在该方向上的边界位置。

[0116] 可视窗口的边界位置与网页对象的位置之差大于期望位移即为期望位移大于可视窗口内的可滑动距离,例如,若网页对象的位置在触摸点离开触摸屏时已经位于可视窗口的底部,则期望位移大于可视窗口内的可滑动距离。

[0117] 若期望位移大于网页对象的位置和可视窗口的边界位置的差值,则惯性滑动模块110可用于根据该差值更新期望位移,并根据更新后的期望位移滑动网页对象。也就是说,当触摸滑动的速度较快,而可视窗口中剩余的可滑动距离不够时,将网页对象滑到可视窗口的边界处。

[0118] 在本实施例中,进一步的,惯性滑动模块110还可用于获取期望位移对应的预设的阈值区间,获取阈值区间对应的延时参数,根据期望位移和延时参数通过调用内置浏览器函数滑动网页对象。

[0119] 可预先设置阈值区间,并设置与阈值区间对应的延时参数值。例如,可设置0至100px的阈值区间对应的延时参数为100ms,100至300px的阈值区间对应的延时参数为200ms,高于300px的阈值区间对应的延时参数为300ms。惯性滑动模块110还可用于在通过计算得到期望位移后,可根据期望位移的大小判断期望位移所处阈值区间,并获取该阈值区间对应的延迟参数。

[0120] 在本实施例中,惯性滑动模块110还可用于根据延迟参数设置网页对象的CSS属性的`-webkit-transition`属性,根据期望位移设置网页对象的CSS属性的`top`属性和/或`left`属性,通过浏览器的CSS事件处理函数设置网页对象的位置。

[0121] 网页对象的CSS属性中的`-webkit-transition`为过渡属性,用于使得网页对象的其他属性的变化由阶梯形的瞬间变化更改为渐进式的逐渐变化。当根据延迟参数设置网页

对象的CSS属性的`-webkit-transition`属性后,再将网页对象的位置设置到目标位置时,网页对象不会立即出现在目标位置,而是逐渐地移动到目标位置。

[0122] 例如,若根据延迟参数将网页对象的CSS属性的`-webkit-transition`值设置为`delay 200ms`,则根据期望位移设置网页对象的CSS属性的`top`属性和/或`left`属性后,网页对象不会立即移动到该`top`属性值和/或`left`属性值对应的位置,而会经过200ms的延迟逐渐移动到该位置。

[0123] 通过设置网页对象的CSS属性来滑动网页对象,可以使网页对象的滑动附带减速滑动的动画效果,从而具有较好的用户体验。

[0124] 在另一个实施例中,惯性滑动模块110还可用于根据期望位移和延时参数通过脚本函数滑动网页对象。例如,基于Javascript语言的jQuery框架中提供了移动网页对象的脚本函数,惯性滑动模块110还可用于通过调用该脚本函数,并将期望位移和延时参数作为参数传递给该脚本函数来滑动网页。

[0125] 在一个实施例中,可视窗口还包括内嵌或外延的边界缓冲区。同步滑动模块106还可用于获取网页对象的位置,在网页对象的位置超出可视窗口或者边界缓冲区的边缘时,停止移动网页对象。

[0126] 边界缓冲区可通过CSS设置,即可视窗口的留白区域,其宽度可由CSS属性中的`margin`或`padding`属性值决定。边界缓冲区内不显示网页对象,但在边界缓冲区内触摸和滑动操作仍能触发网页对象相应的事件。当触摸滑动操作将网页对象滑至可视窗口的边缘时边界缓冲区时,可继续移动网页对象的位置至边界缓冲区中,但进入边界缓冲区的网页对象的部分为不可见,即被遮挡。

[0127] 同步滑动模块106还可用于获取网页对象的位置、可视窗口的边界位置以及边界缓冲区的宽度,通过判断网页对象的位置与可视窗口的边界位置的差值是否小于或等于边界缓冲区的宽度判断网页对象的位置是否超出可视窗口或者边界缓冲区的边缘。

[0128] 也就是说,当网页对象的位置处于边界缓冲区的边缘时,不再移动网页对象,使得网页对象只能进入边界缓冲区一部分,从而可在触摸操作将网页对象滑到可视窗口边缘时造成“滑不动”的效果。

[0129] 进一步的,如图3所示,网页内滑动显示网页对象的装置还包括边界处理模块114,用于获取所述网页对象的位置,在所述网页对象的位置位于所述边界缓冲区内时,滑动所述网页对象至所述边界缓冲区的边缘。

[0130] 同步滑动模块106将网页对象跟随触摸点滑动之后,触摸滑动操作结束,触摸点离开网页对象对应的可视窗口。若此时边界处理模块114获取到的网页对象的位置位于边界缓冲区内,则边界处理模块114可用于将网页对象滑至可视窗口内边界缓冲区的边缘,且停止后续的滑动网页对象的操作;若网页对象的位置不在边界缓冲区中,则参数计算模块108用于计算滑动位移和滑动时间,用于继续滑动网页对象。

[0131] 在本实施例中,边界处理模块114可用于根据预设的弹起参数设置网页对象的CSS属性的`-webkit-transition`值;获取可视窗口内的边界缓冲区的边缘的位置;根据获取的可视窗口内的边界缓冲区的边缘的位置设置网页对象的CSS属性值;通过CSS事件处理函数滑动网页对象至可视窗口内的边界缓冲区的边缘。

[0132] 也就是说,当停止滑动触摸或滑出网页对象对应的可视窗口时,若网页对象的位

置停留在可视窗口的边缘缓冲区中时,可将网页对象移动至可视窗口边缘全部显示,移动的速度和动画效果由预设的弹起参数决定,从而给用户带来更好的用户体验。

[0133] 上述网页内滑动显示网页对象的方法和装置,先获取网页对象的触摸事件和滑动事件,并根据滑动事件对应的触摸点移动网页对象,然后在同步移动网页对象之后还根据触摸事件和滑动事件计算滑动位移和滑动时间,并根据滑动位移和滑动时间继续滑动网页对象。用户在滑动网页对象时,可使网页对象获得比滑动位移更大的移动距离,且可通过控制滑动操作的速度来控制网页对象的移动距离。与传统技术相比,用户只需滑动较少的次数即可将网页对象滑动到特定的位置,从而提高了操作的便利性。

[0134] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,所述的程序可存储于一计算机可读取存储介质中,该程序在执行时,可包括如上述各方法的实施例的流程。其中,所述的存储介质可为磁碟、光盘、只读存储记忆体(Read-Only Memory,ROM)或随机存储记忆体(Random Access Memory,RAM)等。

[0135] 以上所述实施例仅表达了本发明的几种实施方式,其描述较为具体和详细,但不能因此而理解为对本发明专利范围的限制。应当指出的是,对于本领域的普通技术人员来说,在不脱离本发明构思的前提下,还可以做出若干变形和改进,这些都属于本发明的保护范围。因此,本发明专利的保护范围应以所附权利要求为准。

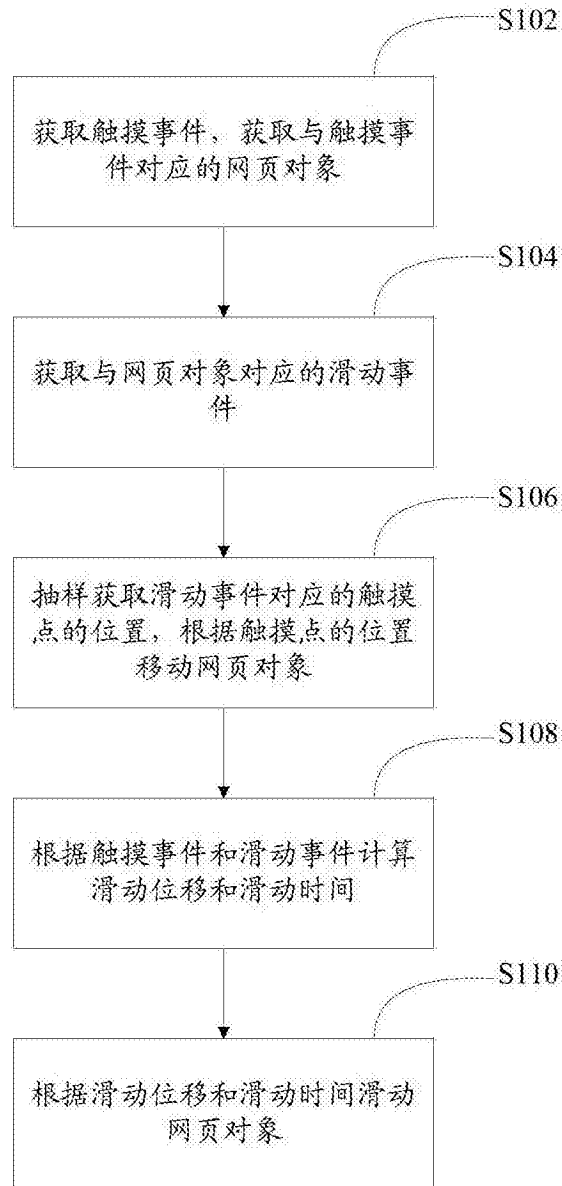


图1

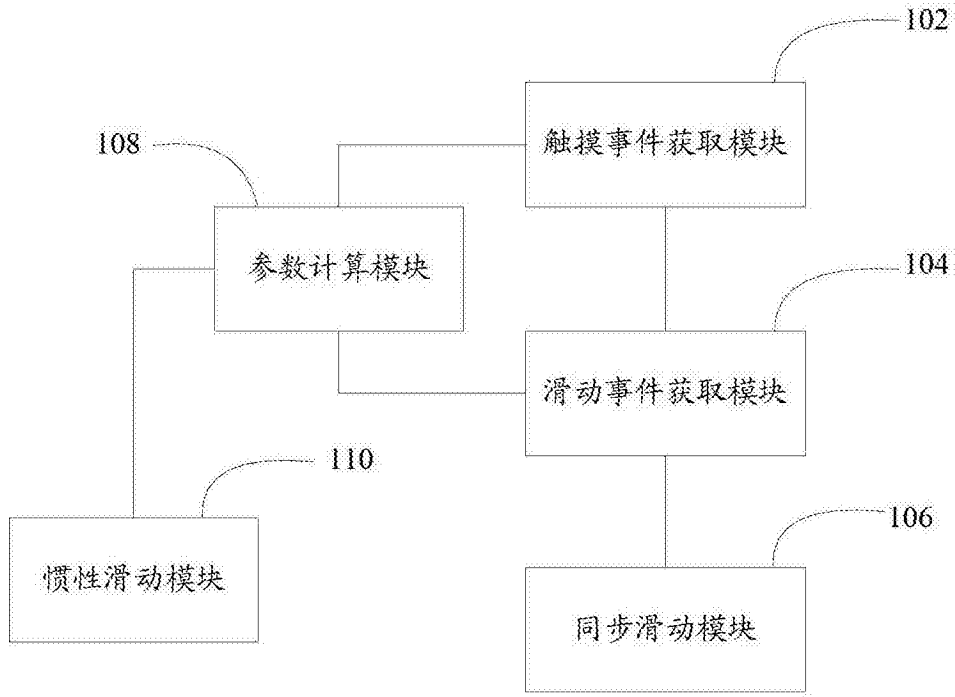


图2

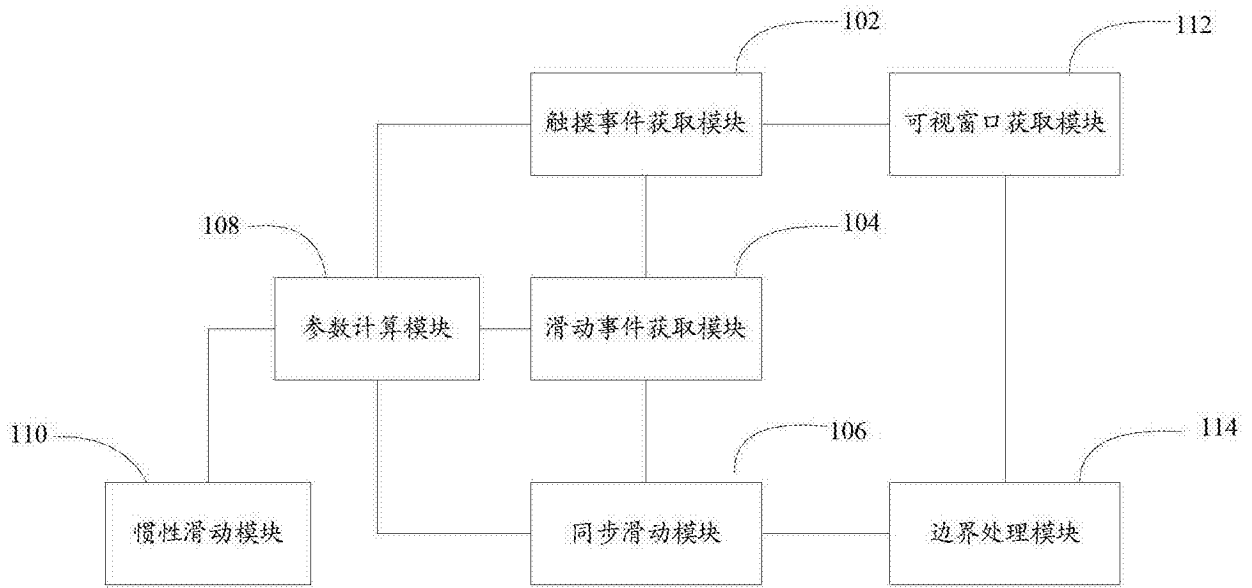


图3