

基于 HTML5 的移动阅读内容展示方式

李青, 王晨升

(北京邮电大学自动化学院, 北京 100876)

5 **摘要:** 据“2014 中国数字阅读白皮书”的大数据显示, 在 2014 年底, 手机阅读用户接近 3 个
亿, 手机阅读明显已经成为手机网民在重拾碎片时间中的主要活动之一。目前的移动阅读平
台由 Native App 主导, Native App 生命周期短、占据用户设备内存资源、上架审核流程复杂、
10 版本更新慢的劣势日益显现。基于 HTML5 的移动阅读 Web App, 采用自适应的网页布局方
式, 自动识别设备屏幕宽度, 适配不同分辨率的设备填充文本内容, 对内容资源进行版式与
流式的排版展现。满足了用户打开移动设备浏览器就能进行阅读的需求, 提升阅读体验。

关键词: Web; Native App; Web App; 阅读

中图分类号: TP303.2

15 The Content Display Method of Mobile Reading Based on HTML5

LI Qing, WANG Chensheng

(Automation School, Beijing University of Posts and Telecommunications, Beijing 100876)

20 **Abstract:** According to the "2014 China Digital Reading white paper", at the end of 2014, the
mobile phone users to read nearly 3 hundred million, mobile phone reading has become one of the
main activities of mobile users in the time of repicking up the debris. The current mobile reading
platform is dominated by App Native, the App Native has a short life cycle, which has the
disadvantages of occupying user's device memory resource, the complexity of the audit process
and the slow version of the update. The Mobile reading Web App, which is based on HTML5,
25 adapt of different resolutions of the device to fill the text content. It meets the needs of the user to
open the mobile browser to read the mobile device to enhance the reading experience.

Key words: Web; Native App; Web App; reading

0 引言

30 随着 Wifi、4G 等网络技术的不断发展, 数据网络传输变得更加快捷与便利, 带动了移
动设备的智能化发展、智能移动设备应用的爆发性增长。在应用市场中, 移动阅读相关应用
市场占有量突出。目前的移动阅读平台由都 Native app 主导。而 Native App 应用的劣势也日
益显现。对于开发者而言, Native App 应用需要针对各种手机操作系统开发多个版本, 同时
Android 版本碎片化问题给开发带来了更大的阻力。随着 W3C 对 HTML5 规范的制定^[1], web
35 app 使用的网页语言在大部分操作系统和浏览器中都兼容, 有效节省了开发成本, 跨平台性、
版本迭代更新简单等巨大优势显现。基于 HTML5 的移动阅读应用使用户只需打开浏览器,
便能随时随地看书^[2]。

Web App 开发的重要特性是一套代码适配所有手机设备。但是, 移动端设备屏幕尺寸
碎片化严重。对于 Android 主流的分辨率就有 480x800、480x854、540x960、720x1280、
40 1080x1920。随着苹果公司推出 iPhone6 与 iPhone6 Plus, iPhone 系列分辨率有以下几种:
640x960、640x1136、750x1334、1242x2208。Ethan Marcotte 在 2010 提出了“自适应网页设
计”, 即自动识别设备屏幕宽度、做出相应网页调整。

作者简介: 李青 (1990-), 女, 在读研究生, 自动化与自动控制

通信联系人: 王晨升, 男, 副教授, 虚拟现实技术. E-mail: chenshengwang@gmail.com

数字阅读终端内容可分为纯图片与图文混排两种内容形式。内容展示过程中的滚屏效果需要表现的类似于纸质书籍阅读的效果，更贴近用户需求。本文主要针对版式排版的展示效果实现与流式排版展示效果实现加以说明。

1 用户触摸屏事件

1.1 Touch 事件处理机制

移动阅读要实现和纸质类书籍阅读类似的阅读体验，需要对用户触摸屏动作做出滚屏或者翻页的效果处理。处理过程即设备屏幕受到用户滑动屏幕后，页面试图区域随手指的移动而移动，手指抬起时根据移动方向判断用户进行上页或是下页的操作，自动移动切换到目标页面。

用户点按触摸屏设备，会触发 Touch 事件^[3]。Touch 事件包含了 Touchstart（手指触摸屏上的时候触发）、Touchmove（手指在屏幕上移动的时候触发）和 Touchend（手指从屏幕上拿起的时候触发）、Touchcancel（系统取消 touch 事件的时候触发）。在这一系列的 Touch 事件触发过程中，伴随有页面的移动，移动效果的处理过程可使用 css3 中的 translate（）方法实现。由于用户在触摸屏设备的屏幕上进行滑动或多点手势操作时往往会触发浏览器行为，例如滚动和缩放。为保证对用户触摸屏事件处理的可控性，需要在 html 文档的头部加上

```
<meta name="viewport" content="width=device-width,  
minimum-scale=1.0,maximum-scale=1.0user-scalable=no"/>  
禁止视图区域滚动和缩放。
```

1.2 Click 事件处理机制

点击事件的处理较容易处理。只需判断用户触摸屏事件位置，根据设定的阈值，判断用户想要实现的翻页效果，以切换页面前翻/后翻。本文将屏幕宽度的 1/3、2/3 位置值作为阈值，当点击事件发生在 1/3 位置左侧区域时，页面切换进入下一页，中间未知区域不做变化，当点击事件发生在 2/3 位置右侧区域时，页面切换至前一页。

2 页面切换效果实现方法的比对

2.1 Touch 事件与 translate()方法实现

对应的 html 文档层级关系如下：

```
<div class="wrap" id="viewBox" >           //视图区域  
    <div id="readFld">                     //内容填充区域  
        ... ..  
    </div>  
</div>
```

触摸屏检测到用户 touch 操作时，首先初始化数据，记录当前页面的触摸位置等信息，继续监听用户 Touchmove 操作，如果 Touchmove 未发生，恢复初始状态，页面不做处理。若发生了，则继续监听用户 Touchend 事件，Touchend 时间的触发表示用户操作结束，判断当前用户手势的相对位移，将该值与设定的阈值进行比较，超出阈值进行翻页处理，否则页面恢复到原始状态。Touchmove 与 Touchend 事件的触发过程中均有页面视图的平移效果，具体实现依赖于 translate()方法与相对位移的大小。

具体实现的流程图如图 1 所示:

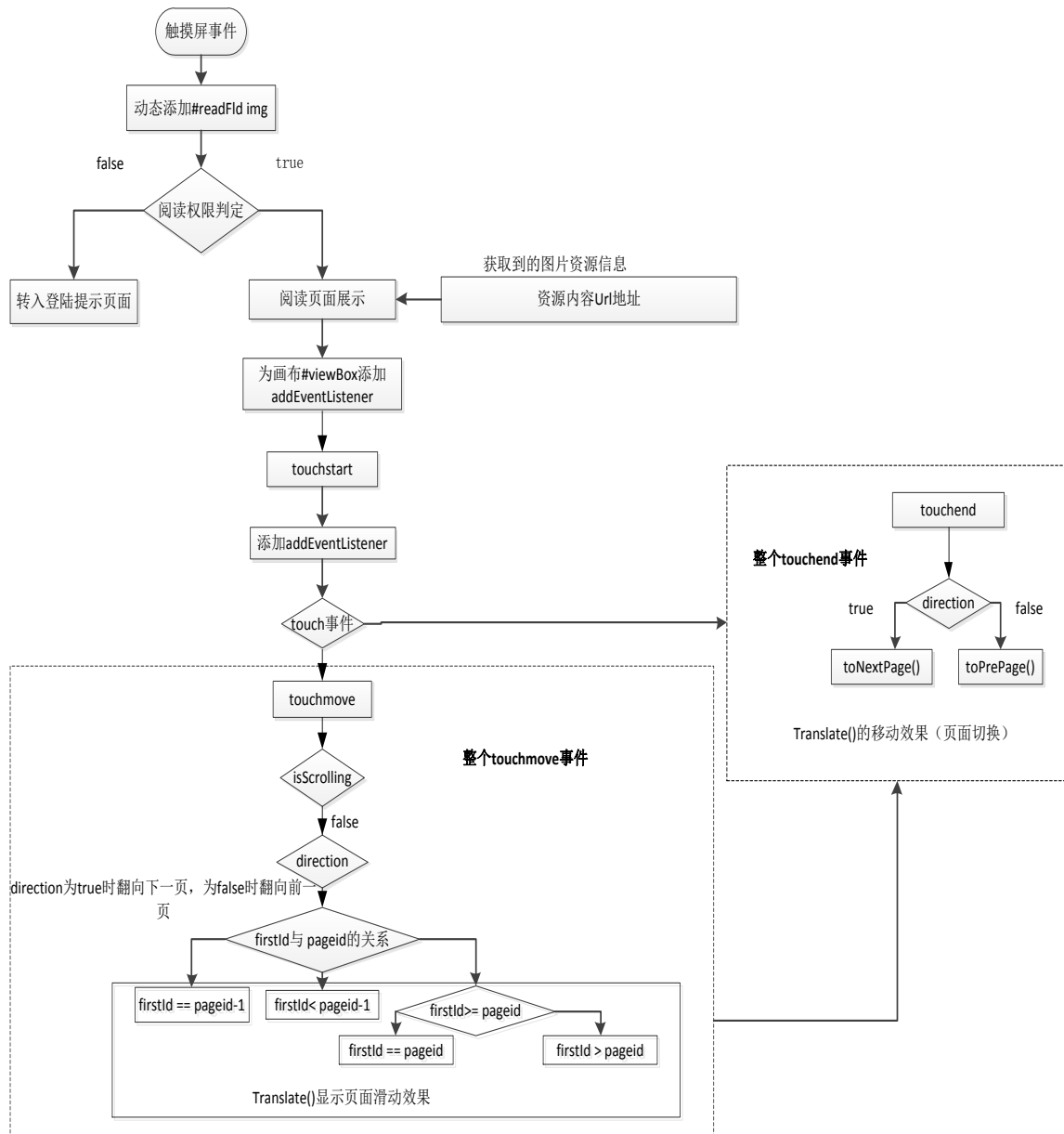


图 1 用户触摸屏事件翻页效果处理

Fig. 1 The processing of page-flipping because of the touches event

整个事件监听中, 手指拖动屏幕的滑屏效果的处理关键要素即为手指滑动的相对位移。在手指初始按压屏幕时即 Touchstart 事件触发时, 记录初始屏幕位置, $startX = event.touches[0].pageX, startY = event.touches[0].pageY$; 在拖动屏幕时触发 Touchmove 事件, 会产生新的位置值, 记录此刻屏幕位置, $nowX = event.touches[0].pageX, nowY = event.touches[0].pageY$; 首先应该根据水平与垂直位移的变化值判断用户行为, 是左右翻页, 还是想要上下滑动。由于是针对翻页的处理, 所以当用户表现出上下滑屏的倾向时, 不作处理。对于左右滑动, 依据相对位移的正负值判定方向, 将相对位移大小值传入 translate 方法, 展现滑屏效果。

95 2.2 借助 Swiper 插件实现

由于整个 touch 事件的处理基于用户行为的位移测算，过程复杂，容易出错。对于用户的快速滑屏行为处理较为棘手，会出现跳页等问题。Swiper 插件的引入，简化了 touch 事件处理过程且效果较为理想。Swiper 是一款轻量级的移动设备触控滑块 js 框架，主要用于移动端的网站、移动 web apps，native apps 和 hybrid apps。在 IOS、Android、WP8 系统都有着良好的用户体验。swiper 插件实际封装了 touch 事件对用户触屏事件的 translate 处理，实现了平滑、流畅的滑屏与切换功能，结合 Swiper 中回调函数的使用，可实现版式阅读的流畅翻页阅读效果。

Swiper 可以处理多个滑块内容，其中添加与删除滑块是常用到的方法。Swiper 对象的 prependSlide() 方法可以为新增滑块到滑块组的第一个位置，appendSlide() 方法将新增滑块到滑块组的最末位位置。对应的删除滑块方法为 removeSlide(index) 方法，index 为目标滑块在滑块组的索引值。

在使用 Swiper 进行阅读翻页效果处理时，实现要进行初始化。初始化过程规定页面滑块组中滑块个数确定视图区域的滑块索引值。为减少页面加载过程中用户等待的时长，提升加载效率，本文在初始化时为 Swiper 添加“左”“中”“右”三个 slide 页面，确定“中”为视图区域，为视图区域加载页面，页面 id=pageid。“左”“右”分别加载 id=pageid-1，id=pageid+1 页面。初始化如图 2 所示，其中着色区域为视图区域。

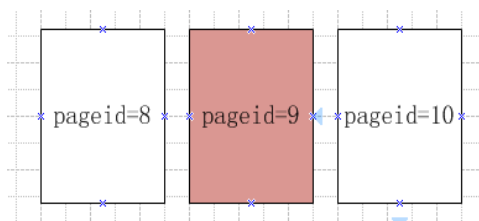


图 2 页面初始化处理
Fig.2 The processing of Page initialization

115 其中对应的 html 层级关系如下图所示：

```
<div class="swiper-container" id="swipeBox">  
  <div class="swiper-wrapper">  
    <div class="swiper-slide" id='pageId_' + (pageid-1) + "'><a...</a></div>  
120    <div class="swiper-slide" id='pageId_' + (pageid) + "'><a...</a></div>  
    <div class="swiper-slide" id='pageId_' + (pageid+1) + "'><a...</a></div>  
  </div>  
</div>
```

初始化设置 Swiper 的视图区域为第二个（中间位置）页面

```
125 <script language="javascript">  
    var mySwiper = new Swiper('.swiper-container',{  
        initialSlide :1  
    })  
</script>
```

130 在整个的处理过程中，首页与末页一直是处理的特殊临界点，在页面进行左翻、右翻的情况下分别进行特殊处理。视图区域进行翻页后的状态显示，即所要做的加载与删除操作处理过程如下，其中 a 着色区域为视图区域，b“←”为左滑，c“→”为右滑，d 用虚线框框起的

页面表示加载该页面，e 用×覆盖表示将页面做删除处理。

(1) 一般情况的处理（视图位置为第二页与倒数第二页之间的页面 pageid）

135 **A**滑动方向为←时，处理过程如图 3 所示，达到翻页阈值时，为 Swiper 末位添加 slide，首位删除 slide，视图区域变更。

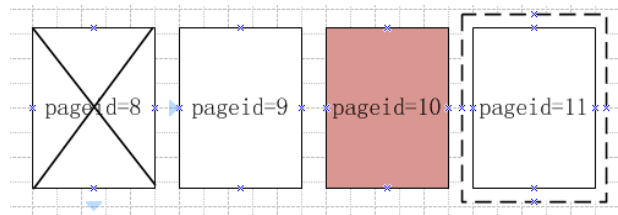


图 3 左滑效果
Fig.3 the effect of left sliding

140 **B** 滑动方向为→时，处理过程如图 4 所示，达到翻页阈值时，为 Swiper 首位添加 slide，末位删除 slide，视图区域变更。

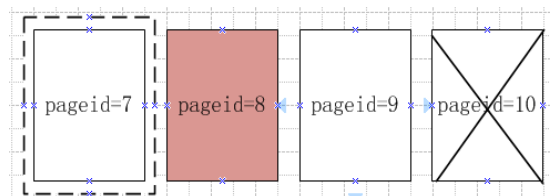


图 4 右滑效果
Fig.4 the effect of right sliding

(2) 特殊情况的处理

1. 第一页在视图区域

150 **A**滑动方向为←时，处理过程如图 5 所示滑动结束后，Swiper 只做末页添加处理，不做首位删除处理。

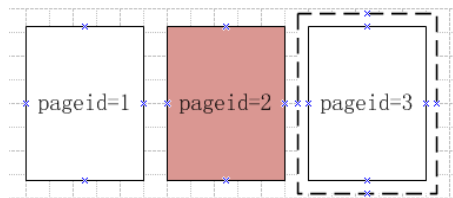


图 5 首页左滑
Fig.5 the left sliding of pageOne

155 **B** 滑动方向为→时，利用 Swiper 的回弹机制，不做处理。

2. 第二页在视图区域

A滑动方向为←时，与一般情况的左滑处理相同。

B 滑动方向为→时，处理过程如图 6 所示，不再为 Swiper 向前添加 slide 页面，对于首页，由于处于 slides 的第一个位置，滑动时候 Swiper 内置有滑动反弹机制，不用额外处理。

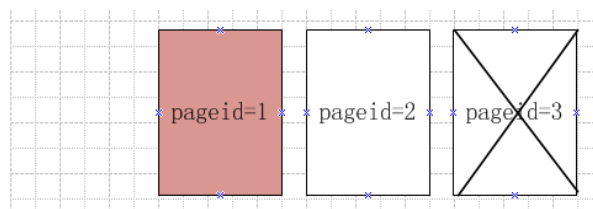


图 6 第二页右滑
Fig.6 the right sliding of pageSecond

3. 倒数第一页在视图区域

A 滑动方向为←时，利用 Swiper 的回弹机制，不做处理。

165 B 滑动方向为→时，处理过程如图 7 所示，由于倒数第一页之后无内容，所以不做末页删除处理，只在首位添加 slide。

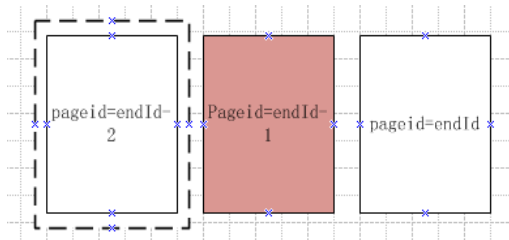


图 7 末页右滑

Fig.7 the right sliding of pageLast

170 4. 倒数第二页在视图区域

A 滑动方向为←时，处理过程如图 8 所示，滑动结束后，Swiper 末位不添加 slide 页面，只进行首位置删除操作。

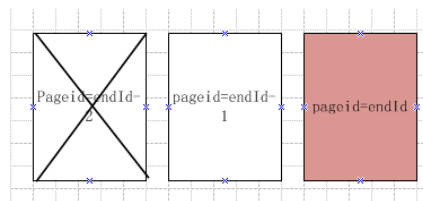


图 8 倒数第二页左滑

Fig.8 the left sliding of the last but one page

B 滑动方向为→时，处理过程与一般情况的右滑处理相同。

上述处理过程中，使用到的监听事件回调函数有以下几种，代码片段如下：

```
var mySwiper = new Swiper('.swiper-container',{ //创建 swiper
180     onSliderMove: function(swiper, event){ //监测用户滑屏事件
        slideEvent();
    },
    onTap: function(swiper){ //监测用户点击事件
        clickEvent();
185    },
    onSlideChangeStart: function(swiper){ //滑块释放时如果触发 slider 切换则执行
    },
    onSlideChangeEnd: function(swiper){ // slider 切换结束时执行
190    }
});
```

其中 onSlideChangeEnd 方法是关键。它表示视图区域的 slide 完全切换，即脱离视图区域时候进行的回调。在加载 slide 与删除 slide 的事件操作就是在这个回调函数中完成的，上述过程的实现均基于 onSlideChangeEnd 这个回调函数。

195 本小节主要提出了使用 Swiper 插件来实现的阅读展现方式，使开发者不用再关心页面展示过程中因为用户滑动而根据滑动距离而计算与实现滑动效果与内容加载，大大简化了复杂度。本小节提出了使用 Swiper 处理特殊页面时候的方式与方法，充分利用 Swiper 自身的

机制达到阅读方式的较好展示。

2.3 图文混排内容资源的特殊处理

对于内容资源为 html 标签元素包含的文本与图片内容资源^[5]考虑采用流式排版的处理方法。请求资源在前端的展示,需要首先在页面引入自定义样式 css 来限制图片大小在视图区域内,调整展示效果。对于文本字体大小,采用自适应页面规则,检测设备屏幕大小,根据不同屏幕分辨率设置不同大小^[6]。采用的方法即为媒体查询,与不同设备引入不同的 css 样式的方法。在使用 javascript 进行用户视图界面填充时,首先判断节点类型,按节点来填充。当填充内容不足或刚刚可以填满视图区域时,进行节点的顺次填放。当检测到填充内容超出视图区域时,对最后的节点进行分析,如果是图片类型的节点,就对图片进行下页填放。如果是段落节点,就将节点内容进行二分法拆分,将内容按字节进行填充,直到填满视图区域,多出内容下页填放。

3 结论

本文给出了基于 HTML5 的移动阅读内容展示方式^[7]。分析比较了采用原生的 javascript 代码实现的翻页阅读效果与借助 Swiper 插件实现的翻页阅读展现效果。Swiper 简化了用户触摸屏位移处理,利用基本的回调函数即可实现平滑过渡,效率高且不易出错。本文还对同时包含图片与文本的内容资源进行了流式排版处理,按节点在视图区域内填放内容,可达到适配各设备尺寸的自适应效果。

[参考文献](References)

- [1] 刘华星, 韩庚. HTML5--下一代 Web 开发标准研究[J]. 计算机技术与发展, 2011, 21 (8): 54-59.
- [2] 王兆, 胡锦. 从用户需求分析如何初步定义产品功能[J]. 艺术与设计(理论), 2011: 203-205
- [3] 黄竞. 基于 jQuery 框架的 Web 前端系统构建方法的研究与应用[D].北京: 北京邮电大学, 2012.
- [4] 黄永慧,陈程凯.HTML5 在移动应用开发上的应用前景[J]. 计算机技术与发展, 2013, (7): 207-210
- [5] 朱聚豹. 移动 Web 前端性能与图片优化的研究[D]. 武汉: 华中师范大学, 2013.
- [6] 戚琪. 移动平台网页自适应云转换机制研究[D].武汉: 华中科技大学, 2013
- [7] 刘斌.HTML5-未来网络应用的核心技术研究[J].自动化与仪器仪表, 2010, (4): 30-33