

(12) 发明专利申请

(10) 申请公布号 CN 102156713 A

(43) 申请公布日 2011.08.17

(21) 申请号 201110059559.8

(22) 申请日 2011.03.11

(71) 申请人 上海全景数字技术有限公司
地址 200083 上海市虹口区广纪路 333 号

(72) 发明人 孙宏明

(74) 专利代理机构 上海专利商标事务所有限公司 31100

代理人 骆希聪

(51) Int. Cl.

G06F 17/30(2006.01)

G09F 27/00(2006.01)

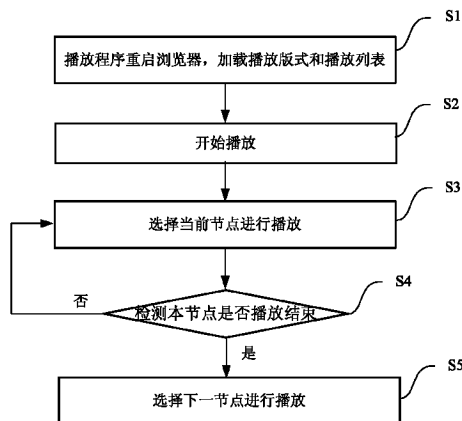
权利要求书 1 页 说明书 9 页 附图 6 页

(54) 发明名称

基于嵌入式设备的多媒体播放方法

(57) 摘要

本发明涉及提出一种基于嵌入式设备的多媒体播放方法,包括以下步骤:加载一播放版式和一播放列表到嵌入式设备的浏览器中,该播放版式是按照一定排版效果预先定制的静态网页,该播放列表定义该媒体资料的播放时间、资源内容以及播放属性;利用该播放列表,根据当前时间对应的播放节点位置,选择媒体资料中的节点进行播放;判断当前节点是否播放完毕;若当前节点播放完毕,选择下一节点进行播放;以及根据该播放版式配置播放的媒体资料在该浏览器中的布局。



1. 一种基于嵌入式设备的多媒体播放方法,包括以下步骤:

加载一播放版式和一播放列表到嵌入式设备的浏览器中,该播放版式是按照一定排版效果预先定制的静态网页,该播放列表定义该媒体资料的播放时间、资源内容以及播放属性;

利用该播放列表,根据当前时间对应的播放节点位置,选择媒体资料中的节点进行播放;

判断当前节点是否播放完毕;

若当前节点播放完毕,选择下一节点进行播放;以及

根据该播放版式配置播放的媒体资料在该浏览器中的布局。

2. 如权利要求 1 所述的方法,其特征在于,还包括:

监听一服务器端是否有更换版式消息;

如果监听到该更换版式消息,则从一服务器端接收播放版式、播放列表和媒体资料到嵌入式设备的本地硬盘,其中该媒体资料的下载与该播放版式和播放列表的下载是相互独立的。

3. 如权利要求 1 所述的方法,其特征在于,还包括:

监听一服务器端是否有插播消息;

如果监听到该插播消息,则从一服务器端接收播放版式、播放列表和媒体资料到嵌入式设备的本地硬盘,其中该媒体资料的下载与该播放版式和播放列表的下载是相互独立的。

4. 如权利要求 3 所述的方法,其特征在于,还包括:

当该播放版式、播放列表和媒体资料下载完毕时,加载插播的播放列表,并且当插播的时间到来时,利用插播的播放列表,根据当前时间对应的插播节点位置,选择媒体资料中的节点进行插播;

判断当前节点是否插播完毕;

若当前节点插播完毕,判断插播是否结束;

如果插播未结束,则选择下一插播节点进行插播;

如果插播结束,则加载原始播放列表进行播放。

5. 如权利要求 1 所述的方法,其特征在于,所述媒体资料包括以下的至少部分:图片、文字、音乐、以及视频。

6. 如权利要求 1 所述的方法,其特征在于,该嵌入式设备为机顶盒。

基于嵌入式设备的多媒体播放方法

技术领域

[0001] 本发明涉及在多媒体信息发布设备的多媒体播放,尤其是涉及一种基于嵌入式设备的多媒体播放方法。

背景技术

[0002] 当前,图文并茂、声画合一的多媒体信息已成为现代社会强有力的信息传播媒介。多媒体信息发布设备作为信息发布的载体,通常工作在环境恶劣的公共场合,而且需要长时间稳定运行。目前已有的多媒体信息发布技术大多采用标准的 PC 机浏览器,通过 IP 网络访问 Web 服务的方式实现。该实现方式往往需要较高的硬件配置,并且需要实时与服务器交换数据,无法满足低功耗、稳定运行等要求。

发明内容

[0003] 本发明所要解决的技术问题是提供一种基于嵌入式设备的多媒体播放方法。

[0004] 本发明为解决上述技术问题而采用的技术方案是提出一种基于嵌入式设备的多媒体播放方法,包括以下步骤:

[0005] 加载一播放版式和一播放列表到嵌入式设备的浏览器中,该播放版式是按照一定排版效果预先定制的静态网页,该播放列表定义该媒体资料的播放时间、资源内容以及播放属性;

[0006] 利用该播放列表,根据当前时间对应的播放节点位置,选择媒体资料中的节点进行播放;

[0007] 判断当前节点是否播放完毕;

[0008] 若当前节点播放完毕,选择下一节点进行播放;以及

[0009] 根据该播放版式配置播放的媒体资料在该浏览器中的布局。

[0010] 在本发明的一实施例中,上述方法还包括:监听一服务器端是否有更换版式消息;如果监听到该更换版式消息,则从一服务器端接收播放版式、播放列表和媒体资料到嵌入式设备的本地硬盘,其中该媒体资料的下载与该播放版式和播放列表的下载是相互独立的。

[0011] 在本发明的一实施例中,上述方法还包括:监听一服务器端是否有插播消息;如果监听到该插播消息,则从一服务器端接收播放版式、播放列表和媒体资料到嵌入式设备的本地硬盘,其中该媒体资料的下载与该播放版式和播放列表的下载是相互独立的。

[0012] 在本发明的一实施例中,上述方法还包括:当该播放版式、播放列表和媒体资料下载完毕时,加载插播的播放列表,并且当插播的时间到来时,利用插播的播放列表,根据当前时间对应的插播节点位置,选择媒体资料中的节点进行插播;判断当前节点是否插播完毕;若当前节点插播完毕,判断插播是否结束;如果插播未结束,则选择下一插播节点进行插播;如果插播结束,则加载原始播放列表进行播放。

[0013] 在本发明的一实施例中,所述媒体资料包括以下的至少部分:图片、文字、音乐、以

及视频。

[0014] 在本发明的一实施例中,该嵌入式设备为机顶盒。

[0015] 本发明基于嵌入式设备进行开发,采用嵌入式浏览器技术,通过加载定制的网页模版、播放列表进行图片、文字、视频、声音的混合播放,同时可通过私有脚本定义对媒资进行插播、重播,版式更换等控制。相比传统的实时播放方法,本发明的代码量小,可满足低功耗、长时间稳定运行等需求。

附图说明

[0016] 为了让本发明的上述目的、特征和优点能更明显易懂,以下结合附图对本发明的具体实施方式作详细说明,其中:

[0017] 图 1 示出本发明的示例性实施环境。

[0018] 图 2 示出本发明一实施例的终端系统架构图。

[0019] 图 3 示出本发明一实施例的版式示意图。

[0020] 图 4 示出本发明一实施例的播放流程图。

[0021] 图 5 示出本发明一实施例的版式更换流程图。

[0022] 图 6 示出本发明一实施例的列表插播流程图。

具体实施方式

[0023] 图 1 示出本发明的示例性实施环境。参照图 1 所示,实施环境中包含多个服务器,例如数据库服务器 110、web 应用服务器 120、下载服务器 130、后台服务器 140 以及接口服务器 150,这些服务器由集群管理服务器 100 进行管理。在本发明的实施例中,各个服务器可以由各自独立的计算机或者计算机组构建,也可以构建在一个计算机或计算机组中。集群管理服务器 100 连接到网络 200。该网络 200 可以是专有网络,或者是公有的互联网。各个地理上分散的场所,例如办公大楼、小区、超市、银行、地铁、机场,营业厅等场所配置有网络设备以连接到网络 200。在这些场所中布置有多媒体信息发布设备。在本实施例中,多媒体信息发布设备使用机顶盒 300 作为播放设备,并可搭配诸如显示屏 402、触摸屏 404、音频输出设备 406 等视频或音频输出设备 400。

[0024] 图 2 示出本发明一实施例的终端系统架构图。参照图 2 所示,机顶盒 300 是嵌入式设备,其包含处理器 302、内存 304、音视频解码单元 306、硬盘 308、通信单元 310、以及非易失性存储器 312。处理器 302 可从非易失性存储器 312 中载入并执行程序,例如图未示的系统程序、以及为实现播放所需的浏览器 322、播放程序 324 和监控程序 326。在执行多媒体播放时,播放程序从硬盘 308 中获取音频、视频等多媒体资料,并交由音视频解码单元 306 进行解码,然后经通信单元 310 输送给诸如显示器等外部设备进行播放。

[0025] 在本实施例中,机顶盒 300 采用嵌入式浏览器技术,通过加载定制的网页模版进行图片、文字、视频、声音等媒体的混合播放。并且,机顶盒 300 可通过私有脚本定义对媒体资料进行插播、重播、版式更换等控制。这一方案的优势是代码量小,可满足低功耗、长时间稳定运行等需求。

[0026] 在此,播放程序 324 和监控程序 326 均可嵌入浏览器 322 中执行。

[0027] 网页模板包括播放版式、媒体资料以及播放列表三部分元素。播放版式是按照一

定排版效果由服务器端预先定制的静态网页。播放列表为对应于各种媒体资料的 XML 列表文件。媒体资料可为图片、文字、视频、声音等形式。这三部分元素包含播放所需的资源和控制信息,其具体描述将在后文展开。三部分元素的获取是相互独立的,尤其是,耗费带宽的媒体资料可独立获取,而依靠播放列表进行播放控制,这使得机顶盒 300 的播放不依赖于媒体资料的实时传输。举例来说,可以使用 P2P 网络传输将图片、文字、声音、视频等媒体资料推送至机顶盒 300 的硬盘 308 后,由监控程序 326 下载播放版式、播放列表等资源,并向播放程序 324 发送播放模板指令。播放程序 324 通过浏览器 322 加载网页模板,按照播放版式配置媒体资料的布局,并将各种媒体资料按照播放列表进行播放。

[0028] 图 3 示出一个示例性的版式示意图,参照图 3 所示,这一示例性的版式 30 包括文字区 31、视频区 32、图片区 33a、33b、以及滚动文字区 34。本领域技术人员可以理解,版式可以根据具体需要进行不同的布局设计。

[0029] 下面列出这一版式 30 的示范性代码:

[0030]

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<script type="text/javascript">
var STATIC_RELEASE_PATH = "/mnt/hd/playlist_all/";

var STATIC_PLAYLIST_PHOTO=STATIC_RELEASE_PATH+ "xml/playlist_
photo.xml";
var STATIC_PLAYLIST_TEXT=STATIC_RELEASE_PATH+ "xml/playlist_te
xt.xml";
var STATIC_PLAYLIST_VIDEO=STATIC_RELEASE_PATH+ "xml/playlist_
video.xml";
var STATIC_PLAYLIST_AUDIO=STATIC_RELEASE_PATH+ "xml/playlist_
audio.xml";
.....
</script>
</head>
<body background="images/bg_main.gif" bgcolor="#ffffff" onload="init();"
[0031]
```

```
onstbnotify="testnotify();">
```

```
<!--文字区 -->
```

```
<div style="position:absolute; left:74px; top:25px; width:600px; height:50px;
z-index:1">
```

```
最新时事新闻
```

```
</div>
```

```
<!--视频区 -->
```

```
<div style="position:absolute; left:59px; top:126px; width:496px; height:317px;
z-index:1">
```

```
<object id="tv_hd" type="tv" width="496" height="317">
```

```
<param id="tv_hd_filename" name="filename"
```

```
value="file:///mnt/hd/asset/video/haibao.ts">
```

```
<param name="chanswitch" value="no">
```

```
<param name="mute" value="0">
```

```
<param name="volume" value="0">
```

```
<param name="adjustvolume" value="1">
```

```
<param name="fullscreen" value="0">
```

```
<param name="timehidden" value="60">
```

```
<param name="playcontrol" value="play">
```

```
<param name="cleararea" value="yes">
```

```
<param name="x" value="59">
```

```
<param name="y" value="126">
```

```
<param name="decode" value="0">
```

```
<param name="play" value="video">
```

```
</object>
```

```
</div>
```

```
[0032]
```

<!--音乐 -->

```
<object id="mp3" classid="clsid:22D6F312-B0F6-11D0-94AB-0080C74C7E95"
type="mp3">
```

```
<param id="mp3_filename" name="filename"
value="file:///mnt/hd/xml/jimozaichangge.mp3">
```

```
<param name="autostart" value="1">
```

```
<param name="playcount" value="0">
```

```
<param name="showcontrols" value="0">
```

```
<param name="showdisplay" value="0">
```

```
<param name="duration" value="4000">
```

```
<param name="decode" value="1">
```

```
<param name="x" value="0">
```

```
<param name="y" value="0">
```

```
<param name="play" value="audio">
```

```
<param name="volume" value="100">
```

```
</object>
```

<!--图片区-->

```
<div style="position:absolute; left:617px; top:95px; width:270px; height:170px;
z-index:1">
```

```

```

```
</div>
```

<!--图片区-->

```
<div style="position:absolute; left:617px; top:278px; width:270px; height:170px;
z-index:1">
```

```

```

```
</div>
```

[0033]

```
<!-- 滚动文字 -->
<div style="position:absolute; left:55px; top:475px; width:845px; height:50px;
z-index:1">
  <marquee id="asset_marquee" name="asset_marquee" direction="left"
scrollamount="4" behavior="scroll" scrolldelay="100" width="845" height="30"
bgcolor="#724849"><font color="#3D0202">版式一 滚动文字 版式一 滚动文字
</font></marquee>
</div>

</body>
</html>
```

[0034] 媒体资料包括图片、文字、声音、视频等形式,通过不同的标签定义,嵌入播放版式中。图片格式包括但不限于 JPG, GIF。音乐格式例如为 mp3,视频格式例如为 ts。

[0035] 播放列表中定义了各种媒体资料的播放时间、资源内容、播放属性等。浏览器通过 JavaScript 脚本加载播放列表文件,并通过定时器控制各种媒体资料的起始播放时间。

[0036] 下面给出图片播放列表 XML 文件的示例:

[0037]

```
<?xml version="1.0"?>
<list list-id="1001" list-name="playlist_photo" list-type="playlist" file-createtim
e="2010-05-19 23:50:00">
  <elements>
    <element-strategy element-strategy-id="100001" element-id="img_1" ele
ment-strategy-type="time" element-type="photo">
      <assets>
        <asset-strategy asset-strategy-id="0" asset-id="10003" asset-file-u
rl="../../asset/photo/1-3.jpg" play-start-time="00:00:00" play-end-time="02:59:59" pla
y-order="1" display-long="" description="" />
```

[0038]


```
<asset-strategy asset-strategy-id="1" asset-id="10004" asset-file-ur
l="../../asset/photo/1-4.jpg" play-start-time="03:00:00" play-end-time="05:59:59" pla
y-order="2" display-long="" description="" />
```

```
</assets>
```

```
</element-strategy>
```

```
<element-strategy element-strategy-id="100002" element-id="img_2" ele
ment-strategy-type="time" element-type="photo" description="">
```

```
<assets>
```

```
<asset-strategy asset-strategy-id="6" asset-id="10004" asset-file-u
rl="../../asset/photo/1-4.jpg" play-start-time="18:00:00" play-end-time="20:59:59" pla
y-order="7" display-long="" description="" />
```

```
<asset-strategy asset-strategy-id="7" asset-id="10003" asset-file-u
rl="../../asset/photo/1-3.jpg" play-start-time="21:00:00" play-end-time="23:59:59" pla
y-order="8" display-long="" description="" />
```

```
</assets>
```

```
</element-strategy>
```

```
</elements>
```

```
</list>
```

[0039] 文字、音乐、视频等播放列表与图片播放列表类似，仅某些属性不同。

[0040] 硬盘 308 可预置默认的模板，包括播放版式、播放列表和媒体资料等。系统启动时，监控程序 326 向播放程序 324 发送默认播放指令。播放程序 324 加载硬盘 308 上的默认模板，进行播放。

[0041] 图 4 示出本发明一实施例的播放流程图。如图 4 所示，播放流程如下：

[0042] 在步骤 S1，播放程序 224 启动浏览器 222，加载播放版式和播放列表。这些播放版式和播放列表可为硬盘预置的，也可为之后从服务器端下载的。

[0043] 在步骤 S2，浏览器 222 运行 JavaScript 脚本，开始进行播放。

[0044] 在播放过程中，通过加载的 XML 播放列表，根据当前时间对应的媒体资料的播放节点位置，选择节点进行播放。如在步骤 S3，选择当前节点进行播放。以前述的播放列表的示范例而言，有两个时间节点，分别为：

```
[0045] <asset-strategy asset-strategy-id = " 0 " asset-id
= " 10003 " asset-file-url = " ../../asset/photo/1-3.jpg " play-start-time
= " 00:00:00 " play-end-time = " 02:59:59 " play-order = " 1 " display-long
```

```
= " " description = " " />
```

[0046] 和

```
[0047] <asset-strategy asset-strategy-id = " 1 " asset-id  
= " 10004 " asset-file-url = " ../../asset/photo/1-4.jpg " play-start-time  
= " 03:00:00 " play-end-time = " 05:59:59 " play-order = " 2 " display-long  
= " " description = " " />
```

[0048] 在步骤 S4, 浏览器 222 通过 JavaScript 脚本定时检测本节点是否播放结束, 如播放完毕则转至步骤 S5, 将用于播放的 DOM 结构中的当前节点内容更换为下一节点内容, 开始播放下一节点。如未播放结束则返回步骤 S3, 继续播放当前节点。

[0049] 当服务器端有新的内容要播放时, 其传输媒体资料, 并按需要传输播放版式和播放列表给机顶盒 300。

[0050] 图 5 示出本发明一实施例的版式更换流程图。如图 5 所示, 版式更换的主要流程如下:

[0051] 在步骤 S11, 监控程序 224 监听管理服务器端指令。当在步骤 S12, 监控程序 226 接收到更换版式消息后, 等待接收数据。在步骤 S13, 管理服务器端下发播放版式、播放列表和相关资源文件, 监控程序 226 接收播放版式、播放列表和媒体资料, 并保存至硬盘 208。

[0052] 在步骤 S14, 当监控程序 226 下载全部资源后, 向播放程序 224 发送更换版式命令。

[0053] 在步骤 S15, 播放程序 224 重启浏览器 222, 加载步骤 S15 中下载的播放版式和播放列表。

[0054] 在步骤 S16, 浏览器 222 运行 JavaScript 脚本, 开始进行播放。

[0055] 在播放过程中, 通过加载的 XML 播放列表, 根据当前时间对应的媒体资料的播放节点位置, 选择节点进行播放。如在步骤 S17, 选择当前节点进行播放。以前述的播放列表的示范例而言, 有两个时间节点, 分别为:

```
[0056] <asset-strategy asset-strategy-id = " 0 " asset-id  
= " 10003 " asset-file-url = " ../../asset/photo/1-3.jpg " play-start-time  
= " 00:00:00 " play-end-time = " 02:59:59 " play-order = " 1 " display-long  
= " " description = " " />
```

[0057] 和

```
[0058] <asset-strategy asset-strategy-id = " 1 " asset-id  
= " 10004 " asset-file-url = " ../../asset/photo/1-4.jpg " play-start-time  
= " 03:00:00 " play-end-time = " 05:59:59 " play-order = " 2 " display-long  
= " " description = " " />
```

[0059] 在步骤 S18, 浏览器 222 通过 JavaScript 脚本定时检测本节点是否播放结束, 如播放完毕则转至步骤 S19, 将用于播放的 DOM 结构中的当前节点内容更换为下一节点内容, 开始播放下一节点。如未播放结束则返回步骤 S17, 继续播放当前节点。

[0060] 图 6 示出本发明一实施例的列表插播流程图。如图 6 所示, 列表插播的主要流程如下:

[0061] 在步骤 S31, 监控程序 226 监听管理服务器端指令, 当于步骤 S32 接收到插播消息后, 等待接收数据。在步骤 S33, 服务器端下发播放版式、播放列表和相关媒体资料, 监控程

序 226 接收播放版式、播放列表和相关媒体资料,并保存到硬盘 208。

[0062] 在本发明的实施例中,插播的媒体文件主要为本地文件,以防止网络传输速度影响插播时间。当然,插播的媒体文件可包含包含刚下载的文件。

[0063] 在步骤 S34,监控程序 226 下载全部资源后,向播放程序 224 发送插播命令。

[0064] 在步骤 S35,播放程序 224 通过 JavaScript 监听接口,监测到插播命令后,通过 JavaScript 脚本,加载插播的 XML 播放列表,转入步骤 S36。在此,JavaScript 脚本根据插播列表时间判断当前是否进行插播。如果进行插播则转入步骤 S37,获取对应当前时间的当前节点内容,进行播放。如果不进行插播则继续播放原始播放列表,转入步骤 S42。

[0065] 开始插播后,在步骤 S38,JavaScript 定时检测当前节点是否播放完毕,如完毕则转入步骤 S39,JavaScript 判断插播是否结束,如果不是,则进入步骤 S40,将当前用于播放的 DOM 中的内容替换为下一插播节点内容,转入步骤 S37,开始播放下一节点。

[0066] 插播列表只播放一次,如插播列表播放完毕,则转入步骤 S41,加载原始播放列表,继续播放插播前的节点,转入原始列表播放流程 S42、S43。

[0067] 本发明基于嵌入式浏览器技术,媒体资源下载后保存在本地硬盘,可以保证播出质量,媒资的发放由管理服务器高效安全的进行控制和管理,同时能够对终端进行监控,从而提高了信息发布业务的效率。

[0068] 虽然本发明已以较佳实施例揭示如上,然其并非用以限定本发明,任何本领域技术人员,在不脱离本发明的精神和范围内,当可作些许的修改和完善,因此本发明的保护范围当以权利要求书所界定的为准。

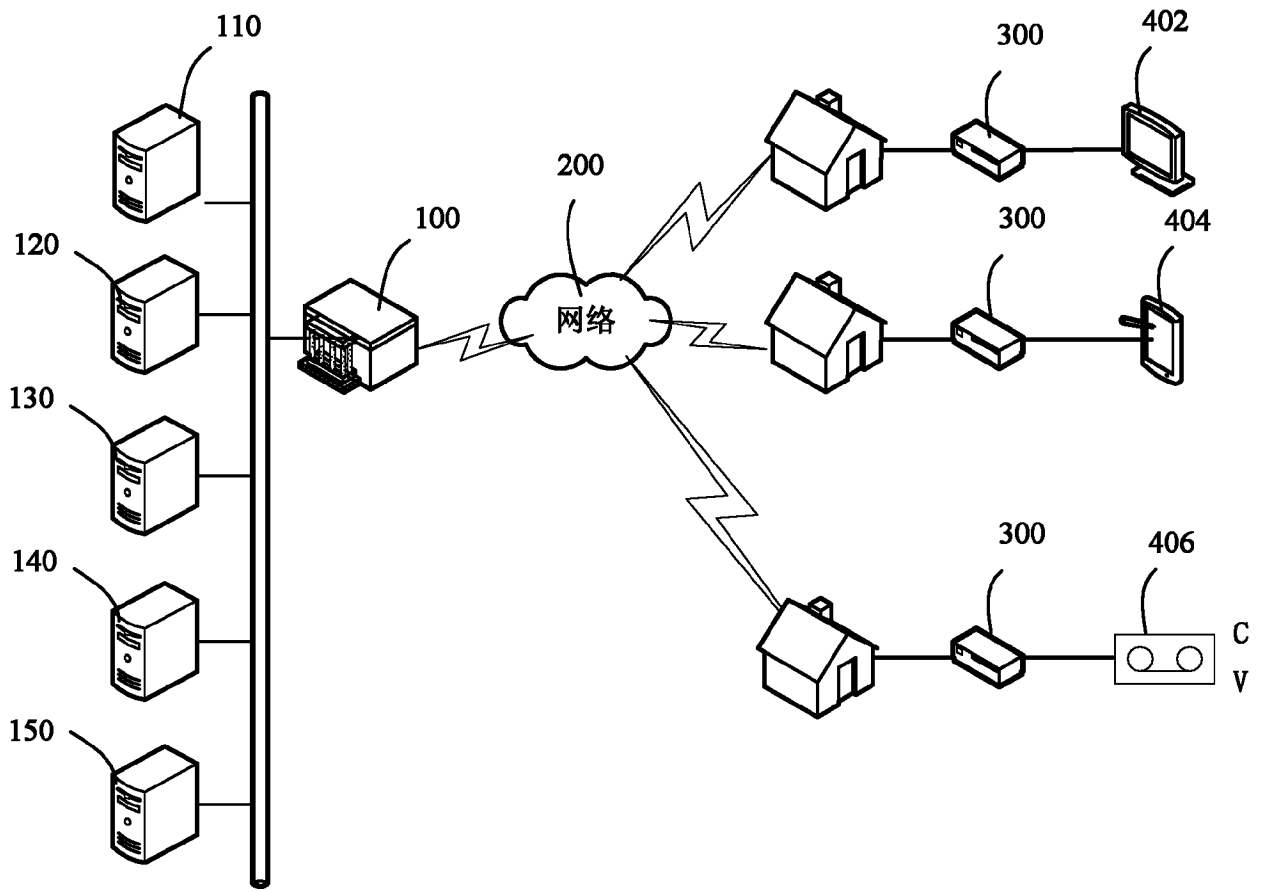


图 1

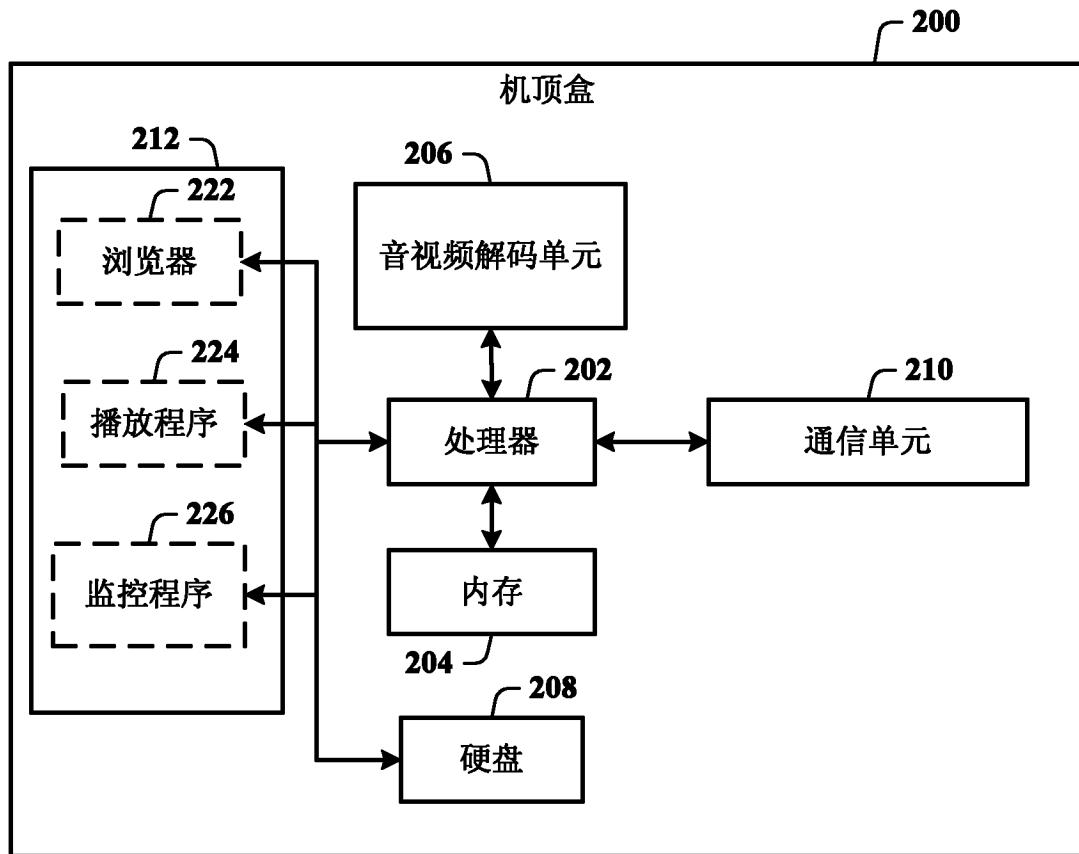


图 2

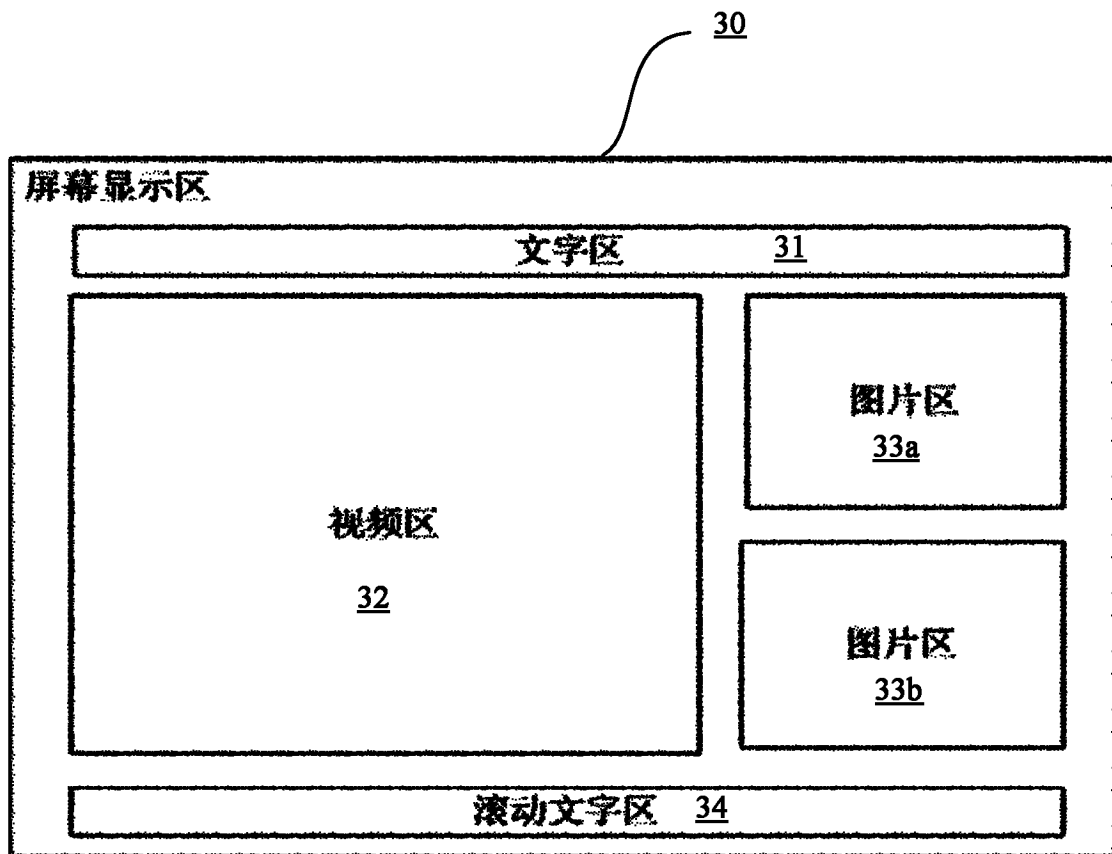


图 3

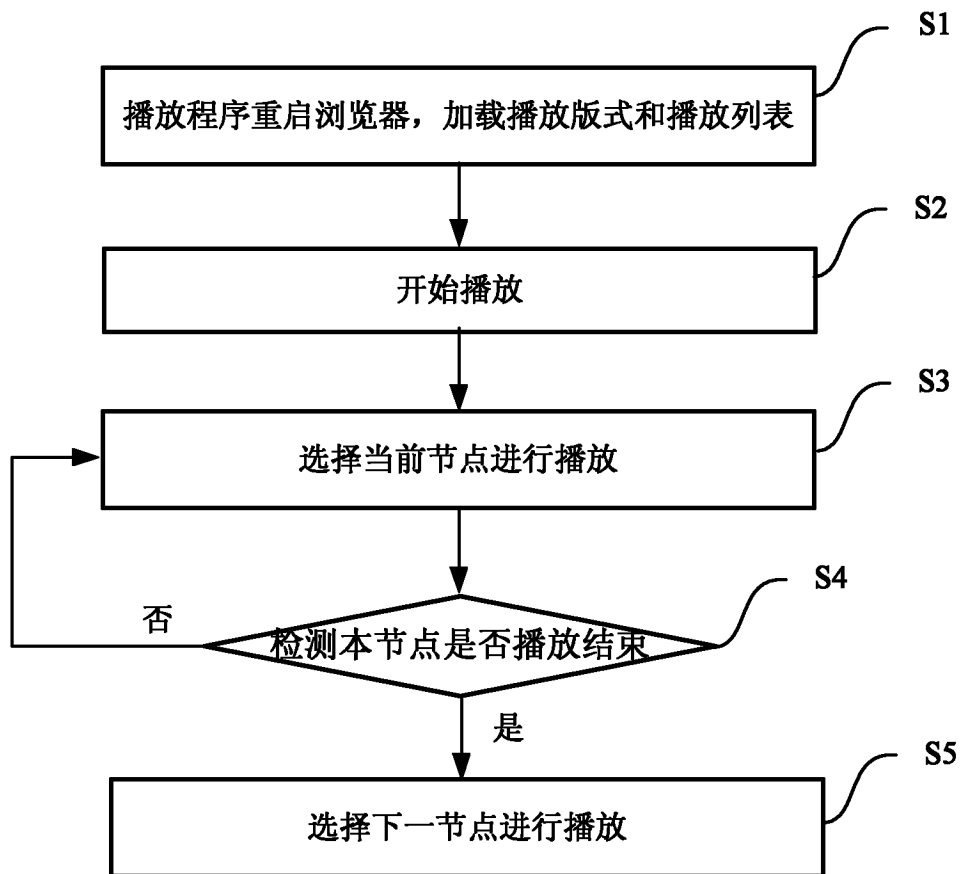


图 4

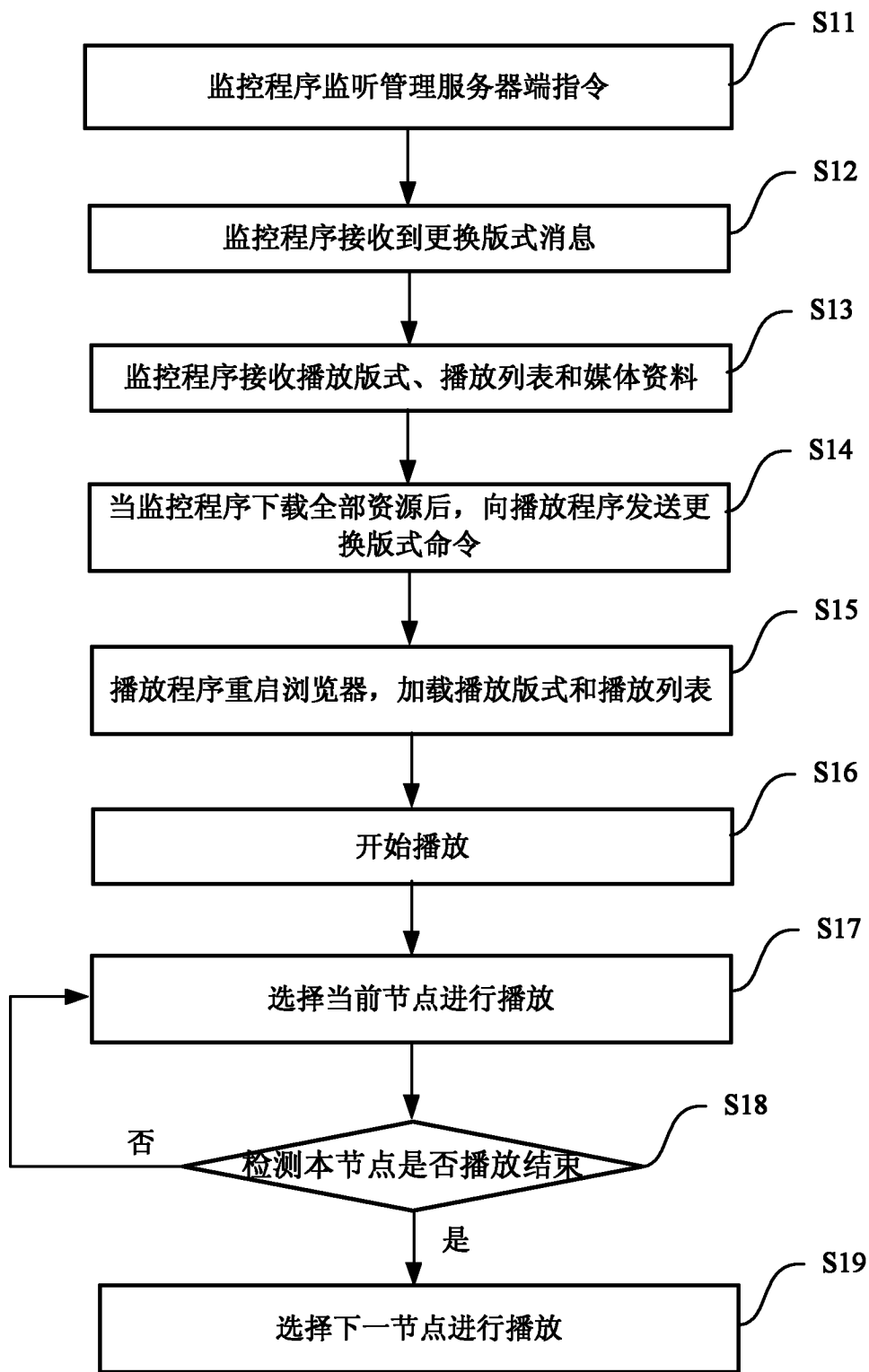


图 5

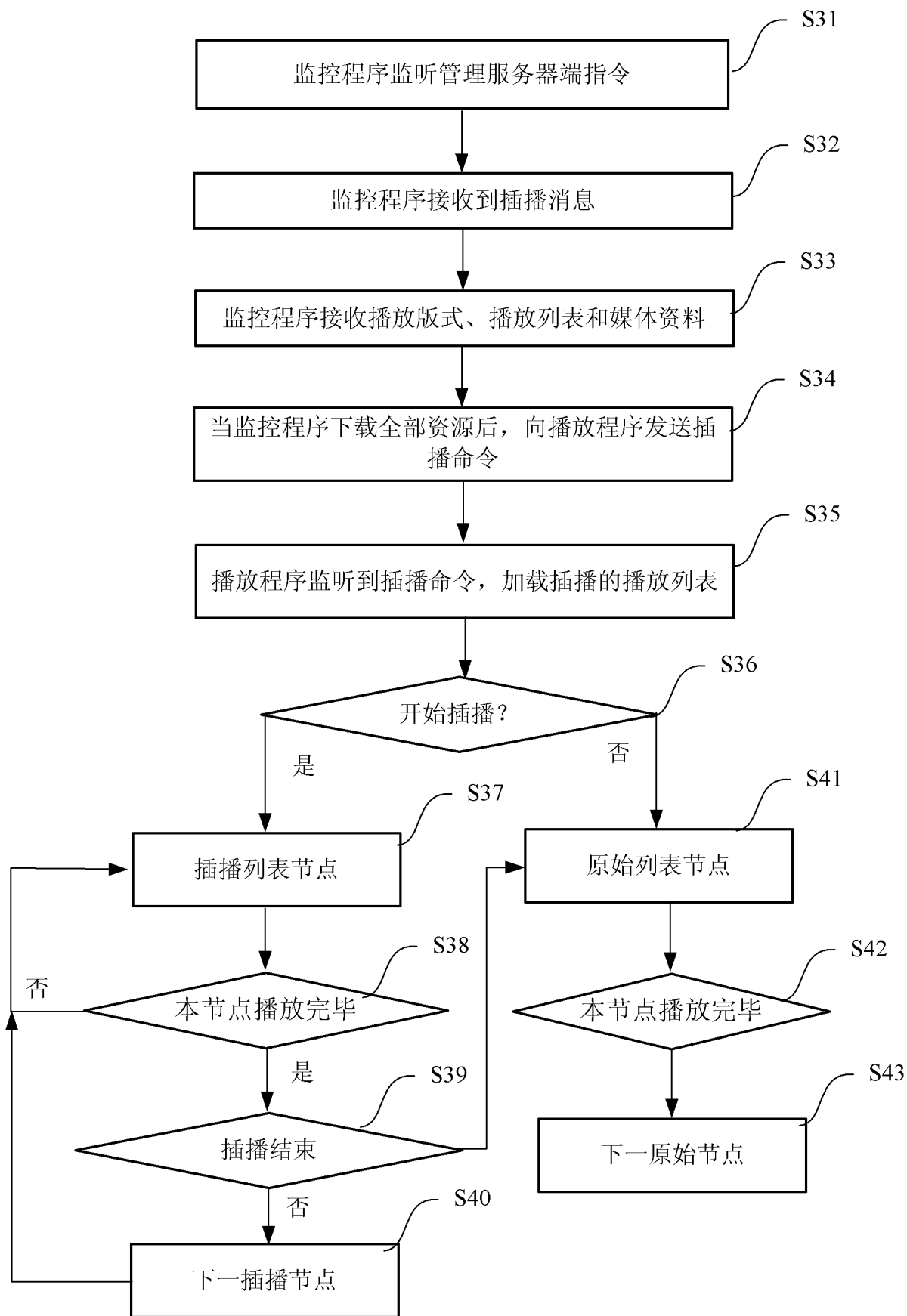


图 6