



(12) 发明专利

(10) 授权公告号 CN 112395437 B

(45) 授权公告日 2024. 04. 30

(21) 申请号 201910759289.8

G06F 16/957 (2019.01)

(22) 申请日 2019.08.16

G06F 16/958 (2019.01)

(65) 同一申请的已公布的文献号

申请公布号 CN 112395437 A

(56) 对比文件

CN 106682176 A, 2017.05.17

CN 107038044 A, 2017.08.11

CN 107784086 A, 2018.03.09

CN 109218354 A, 2019.01.15

CN 109299194 A, 2019.02.01

CN 109889608 A, 2019.06.14

US 7739655 B1, 2010.06.15

(43) 申请公布日 2021.02.23

(73) 专利权人 北京百度网讯科技有限公司

地址 100085 北京市海淀区上地十街10号

百度大厦2层

(72) 发明人 孙学青 郭武彪 胡慈娇 袁振坤

审查员 任甜甜

(74) 专利代理机构 北京品源专利代理有限公司

11332

专利代理师 孟金喆

(51) Int. Cl.

G06F 16/51 (2019.01)

G06F 16/955 (2019.01)

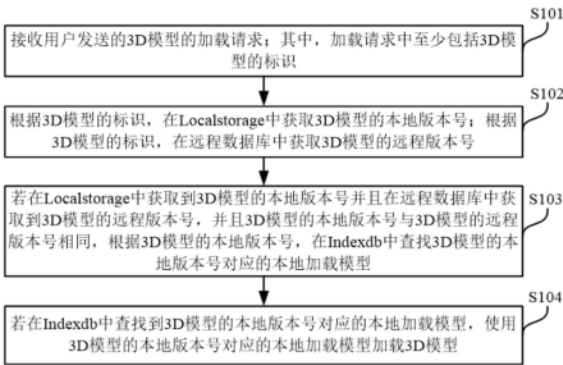
权利要求书3页 说明书15页 附图4页

(54) 发明名称

一种3D模型的加载方法、装置、电子设备及存储介质

(57) 摘要

本发明实施例公开了一种3D模型的加载方法、装置、电子设备及存储介质。所述方法包括：接收用户发送的3D模型的加载请求；根据加载请求中的3D模型的标识，在Localstorage中获取3D模型的本地版本号；根据3D模型的标识在远程数据库中获取3D模型的远程版本号；若在Localstorage中获取到3D模型的本地版本号并且在远程数据库中获取到3D模型的远程版本号，并且3D模型的本地版本号与3D模型的远程版本号相同，根据3D模型的本地版本号在Indexdb中查找3D模型的本地版本号对应的本地加载模型；若未查找到，使用3D模型的本地版本号对应的本地加载模型加载3D模型。不仅可以提高3D模型的加载速度，减少用户等待时间，而且还可以节约远程数据库带宽，降低3D模型的加载复杂度。



1. 一种3D模型的加载方法,其特征在于,所述方法包括:

接收用户发送的3D模型的加载请求;其中,所述加载请求中至少包括所述3D模型的标识;

根据所述3D模型的标识,在本地存储Localstorage中获取所述3D模型的本地版本号;根据所述3D模型的标识,在远程数据库中获取所述3D模型的远程版本号;

若在所述Localstorage中获取到所述3D模型的本地版本号并且在所述远程数据库中获取到所述3D模型的远程版本号,并且所述3D模型的本地版本号与所述3D模型的远程版本号相同,根据所述3D模型的本地版本号,在本地数据库Indexdb中查找所述3D模型的本地版本号对应的本地加载模型;

若在所述Indexdb中查找到所述3D模型的本地版本号对应的本地加载模型,使用所述3D模型的本地版本号对应的本地加载模型加载所述3D模型;

若所述3D模型的本地版本号与所述3D模型的远程版本号不相同,根据所述3D模型的本地版本号,在所述Indexdb中查找所述3D模型的本地版本号对应的本地加载模型;若在所述Indexdb中查找到所述3D模型的本地版本号对应的本地加载模型,使用所述3D模型的本地版本号对应的本地加载模型加载所述3D模型;

根据所述3D模型的远程版本号,在所述远程数据库中查找所述3D模型的远程版本号对应的远程加载模型;若在所述远程数据库中查找到所述3D模型的远程版本号对应的远程加载模型,使用所述3D模型的远程版本号对应的远程加载模型重新加载所述3D模型;

在所述Localstorage中添加所述3D模型的远程版本号;在所述Indexdb中添加所述3D模型的远程版本号对应的远程加载模型;并在所述Localstorage中删除所述3D模型的本地版本号;在所述Indexdb中删除所述3D模型的本地版本号对应的本地加载模型。

2. 根据权利要求1所述的方法,其特征在于,所述根据所述3D模型的标识,在Localstorage中获取所述3D模型的本地版本号,包括:

根据所述3D模型的标识,在所述Localstorage中查找所述3D模型的本地索引;

若在所述Localstorage中查找到所述3D模型的本地索引,根据所述3D模型的本地索引,在所述Localstorage中获取所述3D模型的本地版本号。

3. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

若所述3D模型的本地版本号与所述3D模型的远程版本号不相同,根据所述3D模型的远程版本号,在所述远程数据库中查找所述3D模型的远程版本号对应的远程加载模型;若在所述远程数据库中查找到所述3D模型的远程版本号对应的远程加载模型,使用所述3D模型的远程版本号对应的远程加载模型加载所述3D模型;

在所述Localstorage中添加所述3D模型的远程版本号;在所述Indexdb中添加所述3D模型的远程版本号对应的远程加载模型;并在所述Localstorage中删除所述3D模型的本地版本号;在所述Indexdb中删除所述3D模型的本地版本号对应的本地加载模型。

4. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

若在所述Localstorage中未获取到所述3D模型的本地版本号,而在所述远程数据库中获取到所述3D模型的远程版本号,使用所述3D模型的远程版本号对应的远程加载模型加载所述3D模型;在所述Localstorage中添加所述3D模型的远程版本号;在所述Indexdb中添加所述3D模型的远程版本号对应的远程加载模型;

若在上述Localstorage中获取到所述3D模型的本地版本号,而在所述远程数据库中未获取到所述3D模型的远程版本号,在上述Localstorage中删除所述3D模型的本地版本号;在上述Indexdb中删除所述3D模型的本地版本号对应的本地加载模型。

5. 一种3D模型的加载装置,其特征在于,所述装置包括:接收模块、获取模块、查找模块和加载模块;其中,

所述接收模块,用于接收用户发送的3D模型的加载请求;其中,所述加载请求中至少包括所述3D模型的标识;

所述获取模块,用于根据所述3D模型的标识,在本地存储Localstorage中获取所述3D模型的本地版本号;根据所述3D模型的标识,在远程数据库中获取所述3D模型的远程版本号;

所述查找模块,用于若在上述Localstorage中获取到所述3D模型的本地版本号并且在所述远程数据库中获取到所述3D模型的远程版本号,并且所述3D模型的本地版本号与所述3D模型的远程版本号相同,根据所述3D模型的本地版本号,在本地数据库Indexdb中查找所述3D模型的本地版本号对应的本地加载模型;

所述加载模块,用于若在上述Indexdb中查找到所述3D模型的本地版本号对应的本地加载模型,使用所述3D模型的本地版本号对应的本地加载模型加载所述3D模型;

所述查找模块,还用于若所述3D模型的本地版本号与所述3D模型的远程版本号不相同,根据所述3D模型的本地版本号,在上述Indexdb中查找所述3D模型的本地版本号对应的本地加载模型;

所述加载模块,还用于若在上述Indexdb中查找到所述3D模型的本地版本号对应的本地加载模型,使用所述3D模型的本地版本号对应的本地加载模型加载所述3D模型;

所述查找模块,还用于根据所述3D模型的远程版本号,在上述远程数据库中查找所述3D模型的远程版本号对应的远程加载模型;

所述加载模块,还用于若在上述远程数据库中查找到所述3D模型的远程版本号对应的远程加载模型,使用所述3D模型的远程版本号对应的远程加载模型重新加载所述3D模型;在上述Localstorage中添加所述3D模型的远程版本号;在上述Indexdb中添加所述3D模型的远程版本号对应的远程加载模型;并在上述Localstorage中删除所述3D模型的本地版本号;在上述Indexdb中删除所述3D模型的本地版本号对应的本地加载模型。

6. 根据权利要求5所述的装置,其特征在于,所述获取模块包括:查找子模块和获取子模块;其中,

所述查找子模块,用于根据所述3D模型的标识,在上述Localstorage中查找所述3D模型的本地索引;

所述获取子模块,用于若在上述Localstorage中查找到所述3D模型的本地索引,根据所述3D模型的本地索引,在上述Localstorage中获取所述3D模型的本地版本号。

7. 根据权利要求5所述的装置,其特征在于:

所述查找模块,还用于若所述3D模型的本地版本号与所述3D模型的远程版本号不相同,根据所述3D模型的远程版本号,在上述远程数据库中查找所述3D模型的远程版本号对应的远程加载模型;

所述加载模块,还用于若在上述远程数据库中查找到所述3D模型的远程版本号对应的

远程加载模型,使用所述3D模型的远程版本号对应的远程加载模型加载所述3D模型;在所述Localstorage中添加所述3D模型的远程版本号;在所述Indexdb中添加所述3D模型的远程版本号对应的远程加载模型;并在所述Localstorage中删除所述3D模型的本地版本号;在所述Indexdb中删除所述3D模型的本地版本号对应的本地加载模型。

8.根据权利要求5所述的装置,其特征在于:

所述加载模块,还用于若在所述Localstorage中未获取到所述3D模型的本地版本号,而在所述远程数据库中获取到所述3D模型的远程版本号,使用所述3D模型的远程版本号对应的远程加载模型加载所述3D模型;在所述Localstorage中添加所述3D模型的远程版本号;在所述Indexdb中添加所述3D模型的远程版本号对应的远程加载模型;若在所述Localstorage中获取到所述3D模型的本地版本号,而在所述远程数据库中未获取到所述3D模型的远程版本号,在所述Localstorage中删除所述3D模型的本地版本号;在所述Indexdb中删除所述3D模型的本地版本号对应的本地加载模型。

9.一种电子设备,其特征在于,包括:

一个或多个处理器;

存储器,用于存储一个或多个程序,

当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现如权利要求1至4中任一项所述的3D模型的加载方法。

10.一种存储介质,其上存储有计算机程序,其特征在于,该程序被处理器执行时实现如权利要求1至4中任一项所述的3D模型的加载方法。

一种3D模型的加载方法、装置、电子设备及存储介质

技术领域

[0001] 本发明实施例涉及互联网技术领域,尤其涉及一种3D模型的加载方法、装置、电子设备及存储介质。

背景技术

[0002] 3D模型是一种立体的三维展示技术,可以实现在Web浏览器上展现3D的对象。3D模型技术被人们越来越重视,并且用于各类行业。但是3D模型的加载速度远远慢于普通网页,原因在于每一个3D模型对应于一个庞大的加载模型。模型的加载时间通常比普通网页中各类文件如代码,图片等总和还多,尤其是BIM,网页游戏,模型商店。

[0003] 在实现现有技术的过程中,发明人发现3D模型通常很少更换,绝大部分情况是,用户打开同样的网站,通过远程数据库重复加载模型;在大量模型的场景下,大量的重复加载不仅会造成用户等待时间延长,而且还会浪费远程数据库的带宽。

发明内容

[0004] 有鉴于此,本发明实施例提供一种3D模型的加载方法、装置、电子设备及存储介质,不仅可以提高3D模型的加载速度,减少用户等待时间,而且还可以节约远程数据库带宽,降低3D模型的加载复杂度。

[0005] 第一方面,本发明实施例提供了一种3D模型的加载方法,所述方法包括:

[0006] 接收用户发送的3D模型的加载请求;其中,所述加载请求中至少包括所述3D模型的标识;

[0007] 根据所述3D模型的标识,在本地存储Localstorage中获取所述3D模型的本地版本号;根据所述3D模型的标识,在远程数据库中获取所述3D模型的远程版本号;

[0008] 若在所述Localstorage中获取到所述3D模型的本地版本号并且在所述远程数据库中获取到所述3D模型的远程版本号,并且所述3D模型的本地版本号与所述3D模型的远程版本号相同,根据所述3D模型的本地版本号,在本地数据库Indexdb中查找所述3D模型的本地版本号对应的本地加载模型;

[0009] 若在所述Indexdb中查找到所述3D模型的本地版本号对应的本地加载模型,使用所述3D模型的本地版本号对应的本地加载模型加载所述3D模型。

[0010] 在上述实施例中,所述根据所述3D模型的标识,在Localstorage中获取所述3D模型的本地版本号,包括:

[0011] 根据所述3D模型的标识,在所述Localstorage中查找所述3D模型的本地索引;

[0012] 若在所述Localstorage中查找到所述3D模型的本地索引,根据所述3D模型的本地索引,在所述Localstorage中获取所述3D模型的本地版本号。

[0013] 在上述实施例中,所述方法还包括:

[0014] 若所述3D模型的本地版本号与所述3D模型的远程版本号不相同,根据所述3D模型的本地版本号,在所述Indexdb中查找所述3D模型的本地版本号对应的本地加载模型;若在

所述Indexdb中查找到所述3D模型的本地版本号对应的本地加载模型,使用所述3D模型的本地版本号对应的本地加载模型加载所述3D模型;

[0015] 根据所述3D模型的远程版本号,在所述远程数据库中查找所述3D模型的远程版本号对应的远程加载模型;若在所述远程数据库中查找到所述3D模型的远程版本号对应的远程加载模型,使用所述3D模型的远程版本号对应的远程加载模型重新加载所述3D模型;

[0016] 在所述Localstorage中添加所述3D模型的远程版本号;在所述Indexdb中添加所述3D模型的远程版本号对应的远程加载模型;并在所述Localstorage中删除所述3D模型的本地版本号;在所述Indexdb中删除所述3D模型的本地版本号对应的本地加载模型。

[0017] 在上述实施例中,所述方法还包括:

[0018] 若所述3D模型的本地版本号与所述3D模型的远程版本号不相同,根据所述3D模型的远程版本号,在所述远程数据库中查找所述3D模型的远程版本号对应的远程加载模型;若在所述远程数据库中查找到所述3D模型的远程版本号对应的远程加载模型,使用所述3D模型的远程版本号对应的远程加载模型加载所述3D模型;

[0019] 在所述Localstorage中添加所述3D模型的远程版本号;在所述Indexdb中添加所述3D模型的远程版本号对应的远程加载模型;并在所述Localstorage中删除所述3D模型的本地版本号;在所述Indexdb中删除所述3D模型的本地版本号对应的本地加载模型。

[0020] 在上述实施例中,所述方法还包括:

[0021] 若在所述Localstorage中未获取到所述3D模型的本地版本号,而在所述远程数据库中获取到所述3D模型的远程版本号,使用所述3D模型的远程版本号对应的远程加载模型加载所述3D模型;在所述Localstorage中添加所述3D模型的远程版本号;在所述Indexdb中添加所述3D模型的远程版本号对应的远程加载模型;

[0022] 若在所述Localstorage中获取到所述3D模型的本地版本号,而在所述远程数据库中未获取到所述3D模型的远程版本号,在所述Localstorage中删除所述3D模型的本地版本号;在所述Indexdb中删除所述3D模型的本地版本号对应的本地加载模型。

[0023] 第二方面,本发明实施例提供了一种3D模型的加载装置,所述装置包括:接收模块、获取模块、查找模块和加载模块;其中,

[0024] 所述接收模块,用于接收用户发送的3D模型的加载请求;其中,所述加载请求中至少包括所述3D模型的标识;

[0025] 所述获取模块,用于根据所述3D模型的标识,在本地存储Localstorage中获取所述3D模型的本地版本号;根据所述3D模型的标识,在远程数据库中获取所述3D模型的远程版本号;

[0026] 所述查找模块,用于若在所述Localstorage中获取到所述3D模型的本地版本号并且在所述远程数据库中获取到所述3D模型的远程版本号,并且所述3D模型的本地版本号与所述3D模型的远程版本号相同,根据所述3D模型的本地版本号,在本地数据库Indexdb中查找所述3D模型的本地版本号对应的本地加载模型;

[0027] 所述加载模块,用于若在所述Indexdb中查找到所述3D模型的本地版本号对应的本地加载模型,使用所述3D模型的本地版本号对应的本地加载模型加载所述3D模型。

[0028] 在上述实施例中,所述获取模块包括:查找子模块和获取子模块;其中,

[0029] 所述查找子模块,用于根据所述3D模型的标识,在所述Localstorage中查找所述

3D模型的本地索引；

[0030] 所述获取子模块,用于若在上述Localstorage中查找到所述3D模型的本地索引,根据所述3D模型的本地索引,在上述Localstorage中获取所述3D模型的本地版本号。

[0031] 在上述实施例中,所述查找模块,还用于若所述3D模型的本地版本号与所述3D模型的远程版本号不相同,根据所述3D模型的本地版本号,在上述Indexdb中查找所述3D模型的本地版本号对应的本地加载模型；

[0032] 所述加载模块,还用于若在上述Indexdb中查找到所述3D模型的本地版本号对应的本地加载模型,使用所述3D模型的本地版本号对应的本地加载模型加载所述3D模型；

[0033] 所述查找模块,还用于根据所述3D模型的远程版本号,在上述远程数据库中查找所述3D模型的远程版本号对应的远程加载模型；

[0034] 所述加载模块,还用于若在上述远程数据库中查找到所述3D模型的远程版本号对应的远程加载模型,使用所述3D模型的远程版本号对应的远程加载模型重新加载所述3D模型；在上述Localstorage中添加所述3D模型的远程版本号；在上述Indexdb中添加所述3D模型的远程版本号对应的远程加载模型；并在上述Localstorage中删除所述3D模型的本地版本号；在上述Indexdb中删除所述3D模型的本地版本号对应的本地加载模型。

[0035] 在上述实施例中,所述查找模块,还用于若所述3D模型的本地版本号与所述3D模型的远程版本号不相同,根据所述3D模型的远程版本号,在上述远程数据库中查找所述3D模型的远程版本号对应的远程加载模型；

[0036] 所述加载模块,还用于若在上述远程数据库中查找到所述3D模型的远程版本号对应的远程加载模型,使用所述3D模型的远程版本号对应的远程加载模型加载所述3D模型；在上述Localstorage中添加所述3D模型的远程版本号；在上述Indexdb中添加所述3D模型的远程版本号对应的远程加载模型；并在上述Localstorage中删除所述3D模型的本地版本号；在上述Indexdb中删除所述3D模型的本地版本号对应的本地加载模型。

[0037] 在上述实施例中,所述加载模块,还用于若在上述Localstorage中未获取到所述3D模型的本地版本号,而在上述远程数据库中获取到所述3D模型的远程版本号,使用所述3D模型的远程版本号对应的远程加载模型加载所述3D模型；在上述Localstorage中添加所述3D模型的远程版本号；在上述Indexdb中添加所述3D模型的远程版本号对应的远程加载模型；若在上述Localstorage中获取到所述3D模型的本地版本号,而在上述远程数据库中未获取到所述3D模型的远程版本号,在上述Localstorage中删除所述3D模型的本地版本号；在上述Indexdb中删除所述3D模型的本地版本号对应的本地加载模型。

[0038] 第三方面,本发明实施例提供了一种电子设备,包括:

[0039] 一个或多个处理器；

[0040] 存储器,用于存储一个或多个程序,

[0041] 当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现本发明任意实施例所述的3D模型的加载方法。

[0042] 第四方面,本发明实施例提供了一种存储介质,其上存储有计算机程序,该程序被处理器执行时实现本发明任意实施例所述的3D模型的加载方法。

[0043] 本发明实施例提出了一种3D模型的加载方法、装置、电子设备及存储介质,先接收用户发送的3D模型的加载请求；然后根据加载请求中的3D模型的标识,在Localstorage中

获取3D模型的本地版本号;根据3D模型的标识,在远程数据库中获取3D模型的远程版本号;若在Localstorage中获取到3D模型的本地版本号并且在远程数据库中获取到3D模型的远程版本号,并且3D模型的本地版本号与3D模型的远程版本号相同,根据3D模型的本地版本号,在Indexdb中查找3D模型的本地版本号对应的本地加载模型;若在Indexdb中查找到3D模型的本地版本号对应的本地加载模型,使用3D模型的本地版本号对应的本地加载模型加载3D模型。也就是说,在本发明的技术方案中,可以在Localstorage中获取到3D模型的本地版本号并且可以在远程数据库中获取到3D模型的远程版本号,并且3D模型的本地版本号与3D模型的远程版本号相同时,可以使用查找到的3D模型的本地版本号对应的本地加载模型加载3D模型。而在现有的3D模型的加载方法中,用户打开同样的网站,通过远程数据库重复加载模型;在大量模型的场景下,大量的重复加载不仅会造成用户等待时间延长,而且还会浪费远程数据库的带宽。因此,和现有技术相比,本发明实施例提出的3D模型的加载方法、装置、电子设备及存储介质,不仅可以提高3D模型的加载速度,减少用户等待时间,而且还可以节约远程数据库带宽,降低3D模型的加载复杂度;并且,本发明实施例的技术方案实现简单方便、便于普及,适用范围更广。

附图说明

- [0044] 图1为本发明实施例一提供的3D模型的加载方法的流程示意图;
- [0045] 图2为本发明实施例二提供的3D模型的加载方法的流程示意图;
- [0046] 图3为本发明实施例三提供的3D模型的加载方法的流程示意图;
- [0047] 图4为本发明实施例四提供的3D模型的加载方法的流程示意图;
- [0048] 图5为本发明实施例五提供的3D模型的加载装置的第一结构示意图;
- [0049] 图6为本发明实施例五提供的3D模型的加载装置的第二结构示意图;
- [0050] 图7为本发明实施例六提供的电子设备的结构示意图。

具体实施方式

[0051] 下面结合附图和实施例对本发明作进一步的详细说明。可以理解的是,此处所描述的具体实施例仅仅用于解释本发明,而非对本发明的限定。另外还需要说明的是,为了便于描述,附图中仅示出了与本发明相关的部分而非全部内容。

[0052] 实施例一

[0053] 图1为本发明实施例一提供的3D模型的加载方法的流程示意图,该方法可以由3D模型的加载装置或者电子设备来执行,该装置或者电子设备可以由软件和/或硬件的方式实现,该装置或者电子设备可以集成在任何具有网络通信功能的智能设备中。如图1所示,3D模型的加载方法可以包括以下步骤:

[0054] S101、接收用户发送的3D模型的加载请求;其中,加载请求中至少包括3D模型的标识。

[0055] 在本发明的具体实施例中,电子设备可以接收用户发送的3D模型的加载请求;其中,加载请求中至少包括3D模型的标识。具体地,3D模型的标识可以是该模型的名称,也可以是该模型的编号,在此不做限定。

[0056] S102、根据3D模型的标识,在Localstorage中获取3D模型的本地版本号;根据3D模

型的标识,在远程数据库中获取3D模型的远程版本号。

[0057] 在本发明的具体实施例中,电子设备可以根据3D模型的标识,在Localstorage中获取3D模型的本地版本号;根据3D模型的标识,在远程数据库中获取3D模型的远程版本号。具体地,Localstorage可以预先保存各个3D模型的本地索引、本地版本号以及URL;在本步骤中,电子设备可以根据3D模型的标识,在Localstorage中获取3D模型的本地索引、本地版本号和URL。此外,远程数据库还可以预先保存各个3D模型的远程索引、远程版本号以及URL;在本步骤中,电子设备可以根据3D模型的标识,在远程数据库中获取3D模型的远程索引、远程版本号和URL。

[0058] S103、若在Localstorage中获取到3D模型的本地版本号并且在远程数据库中获取到3D模型的远程版本号,并且3D模型的本地版本号与3D模型的远程版本号相同,根据3D模型的本地版本号,在Indexdb中查找3D模型的本地版本号对应的本地加载模型。

[0059] 在本发明的具体实施例中,若在Localstorage中获取到3D模型的本地版本号并且在远程数据库中获取到3D模型的远程版本号,并且3D模型的本地版本号与3D模型的远程版本号相同,电子设备可以根据3D模型的本地版本号,在Indexdb中查找3D模型的本地版本号对应的本地加载模型。具体地,若在Localstorage中未获取到3D模型的本地版本号,而在远程数据库中获取到3D模型的远程版本号,电子设备可以使用3D模型的远程版本号对应的远程加载模型加载3D模型;同时在Localstorage中添加3D模型的远程版本号;在Indexdb中添加3D模型的远程版本号对应的远程加载模型;若在Localstorage中可以获取到3D模型的本地版本号,而在远程数据库中未获取到3D模型的远程版本号,电子设备在Localstorage中删除3D模型的本地版本号;在Indexdb中删除3D模型的本地版本号对应的本地加载模型,此时电子设备可以结束3D模型的加载流程。

[0060] 此外,若3D模型的本地版本号与3D模型的远程版本号不相同,电子设备可以根据3D模型的本地版本号,在Indexdb中查找3D模型的本地版本号对应的本地加载模型;若在Indexdb中查找到3D模型的本地版本号对应的本地加载模型,电子设备可以先使用3D模型的本地版本号对应的本地加载模型加载3D模型;然后根据3D模型的远程版本号,在远程数据库中查找3D模型的远程版本号对应的远程加载模型;若在远程数据库中查找到3D模型的远程版本号对应的远程加载模型,可以再使用3D模型的远程版本号对应的远程加载模型重新加载3D模型;同时在Indexdb中添加3D模型的远程版本号对应的远程加载模型;并在Indexdb中删除3D模型的本地版本号对应的本地加载模型。这样,用户可以先体验到旧版本的3D模型,然后将新版本的3D模型更新旧版本的3D模型,从而可以节省用户的等待时间。

[0061] 可选地,若3D模型的本地版本号与3D模型的远程版本号不相同,电子设备可以根据3D模型的远程版本号,在远程数据库中查找3D模型的远程版本号对应的远程加载模型;若在远程数据库中查找到3D模型的远程版本号对应的远程加载模型,电子设备可以使用3D模型的远程版本号对应的远程加载模型加载3D模型;同时在Indexdb中添加3D模型的远程版本号对应的远程加载模型;并在Indexdb中删除3D模型的本地版本号对应的本地加载模型。这样,无需先加载旧版本的3D模型,可以简化加载流程,节约系统带宽。

[0062] S104、若在Indexdb中查找到3D模型的本地版本号对应的本地加载模型,使用3D模型的本地版本号对应的本地加载模型加载3D模型。

[0063] 在本发明的具体实施例中,若在Indexdb中查找到3D模型的本地版本号对应的本

地加载模型,电子设备可以使用3D模型的本地版本号对应的本地加载模型加载3D模型;若在Indexdb中未查找到3D模型的本地版本号对应的本地加载模型,电子设备可以再根据3D模型的远程版本号,在远程数据库中查找3D模型的远程版本号对应的远程加载模型;若在远程数据库中查找到3D模型的远程版本号对应的远程加载模型,电子设备可以使用3D模型的远程版本号对应的远程加载模型加载3D模型;若在远程数据库中未查找到3D模型的远程版本号对应的远程加载模型,电子设备可以结束3D模型的加载流程。

[0064] 本发明实施例可以解决如下四个技术问题:1) 在第二次和第二次以后的打开网页时,可提高3D模型的加载速度;2) 如何将缓存变成人为可控,可预测,可修改;3) 如何摆脱开发所使用的引擎限制,提供一套统一的解决方案来应对使用不同技术的优化;4) 如何做到资源服务器宕机,网页也可以正常使用。具体地,针对第一个技术问题,一开始接收到用户发送的3D模型的加载请求后,页面会去请求远程数据库的后端接口,获取到该页面的所有3D模型的远程索引、远程版本号和URL。但是在获取到这些数据后并不是直接通过URL就去加载3D模型;而是从LocalStorage里面获取出历史数据。LocalStorage里面存储着以前用户打开该网页时所保存的所有3D模型的本地索引、本地版本号和URL,这些信息表示为一个json字符串,该字符串里面的Data是个数组,用于记录每个3D模型的本地索引、本地版本号和URL。然后可以将从LocalStorage获取到的本地信息和从远程数据库获取到的远程信息进行对比,这样就可以知道哪些3D模型在新版本里面不需要了,哪些3D模型有更新;这样可以删除不需要的3D模型,重新加载更新的3D模型。对于一个的复杂场景,例如游戏或者BIM,使用这种方法可以非常显著的提高第二次及第二次以后的游戏加载速度。因为3D模型的更新是一个低频率的现象,例如,游戏版本升级添加新的角色模型,那么玩家在升级的那天进入页面,则只需要下载更新的3D模型。对于简单的页面场景,例如,模型商店,可以实现在数秒内打开。用户如果多次访问一个3D模型的展示页面,后续的页面加载不会再次加载该模型,从而大幅度增加用户体验。针对第二个技术问题,由于使用LocalStorage和Indexdb,这些离线存储技术是浏览器提供API可以人为控制,因此开发者可以人为控制储存的数据量。例如模型商店,用户对于某个模型的查看次数是少量的,用户很可能存在进入模型渲染页面只是看一下然后就换一个模型,一次性看上百个3D模型,3D模型的本地储存数量会随着用户的操作而线性增加。那么久没有必要上百个3D模型都永久储存于用户电脑,通过控制LocalStorage限制一个模型离线储存的上限数量,把超出数量的模型从Indexdb删除,就能节省用户的本地磁盘占用。开发者也可以根据系统的业务情况做自定义的剔除策略,例如使用LRU算法把使用频率最少的模型优先剔除。而对于游戏场景,用户对某一模型的使用次数是相当频繁的,但是模型的总体体积不会改变(除非游戏版本升级),因此对于这类场景则无需限制离线存储模型数量上限。开发者就可以针对不同的业务场景进行人为的控制和调整模型的本地储存数量。针对第三个技术问题,由于整个流程都不会涉及各类引擎使用的API,3D模型的本质就是纯粹的数据,因此使用二进制的方式存储在Indexdb会比使用json格式储存在Indexdb更合适。因为对于前端WebGL渲染引擎(Three.js, Babylon, PlayCanvas等),不同的引擎加载解析模型最后形成json格式的数据结构都不一样。如果使用json则需要为各种引擎都提供各自的储存解析方案。而使用二进制则所有引擎都是通用的,在使用时再动态的去解析二进制为各自的json格式。另一方面二进制格式占用的磁盘体积远远小于json格式。针对第四个技术问题,对于一个WebGL应用,为了提高加载速度减

少服务器的压力,很多架构方案会将3D模型资源存放在CDN服务器。但是CDN服务器的可靠性也不是100%,各类资源服务器出现故障造成服务无法使用的情况也会时常发生。一旦资源服务器宕机了,那么对于WebGL的网页来说,就无法加载任何模型造成整个页面无法使用。但是使用本发明实施例提供的技术方案,由于indexdb存储有历史模型的资源,即便版本不是最新的,也可以先把旧的所有模型全部渲染出来给用户使用。这样用户甚至觉察不到资源服务器宕机的情况,从而即便服务器故障依旧保持着网页可以正常使用;系统的可用性也大幅度提高。

[0065] 本发明实施例提出的3D模型的加载方法,先接收用户发送的3D模型的加载请求;然后根据加载请求中的3D模型的标识,在Localstorage中获取3D模型的本地版本号;根据3D模型的标识,在远程数据库中获取3D模型的远程版本号;若在Localstorage中获取到3D模型的本地版本号并且在远程数据库中获取到3D模型的远程版本号,并且3D模型的本地版本号与3D模型的远程版本号相同,根据3D模型的本地版本号,在Indexdb中查找3D模型的本地版本号对应的本地加载模型;若在Indexdb中查找到3D模型的本地版本号对应的本地加载模型,使用3D模型的本地版本号对应的本地加载模型加载3D模型。也就是说,在本发明的技术方案中,可以在Localstorage中获取到3D模型的本地版本号并且可以在远程数据库中获取到3D模型的远程版本号,并且3D模型的本地版本号与3D模型的远程版本号相同时,可以使用查找到的3D模型的本地版本号对应的本地加载模型加载3D模型。而在现有的3D模型的加载方法中,用户打开同样的网站,通过远程数据库重复加载模型;在大量模型的场景下,大量的重复加载不仅会造成用户等待时间延长,而且还会浪费远程数据库的带宽。因此,和现有技术相比,本发明实施例提出的3D模型的加载方法,不仅可以提高3D模型的加载速度,减少用户等待时间,而且还可以节约远程数据库带宽,降低3D模型的加载复杂度;并且,本发明实施例的技术方案实现简单方便、便于普及,适用范围更广。

[0066] 实施例二

[0067] 图2为本发明实施例二提供的3D模型的加载方法的流程示意图。如图2所示,3D模型的加载方法可以包括以下步骤:

[0068] S201、接收用户发送的3D模型的加载请求;其中,加载请求中至少包括3D模型的标识。

[0069] 在本发明的具体实施例中,电子设备可以接收用户发送的3D模型的加载请求;其中,加载请求中至少包括3D模型的标识。具体地,3D模型的标识可以是该模型的名称,也可以是该模型的编号,在此不做限定。

[0070] S202、根据3D模型的标识,在Localstorage中查找3D模型的本地索引。

[0071] 在本发明的具体实施例中,根据3D模型的标识,在Localstorage中查找3D模型的本地索引。具体地,电子设备可以在Localstorage中预先保存3D模型的标识与本地索引的对应关系;在本步骤中,电子设备可以根据3D模型的标识,在Localstorage中查找3D模型的本地索引。

[0072] S203、若在Localstorage中查找到3D模型的本地索引,根据3D模型的本地索引,在Localstorage中获取3D模型的本地版本号。

[0073] 在本发明的具体实施例中,若在Localstorage中查找到3D模型的本地索引,电子设备可以根据3D模型的本地索引,在Localstorage中获取3D模型的本地版本号。具体地,电

子设备还可以在Localstorage中预先保存3D模型的本地索引与本地版本号的关系;在本步骤中,电子设备可以根据3D模型的本地索引,在Localstorage中获取3D模型的本地版本号。此外,若在Localstorage中未查找到3D模型的本地索引,电子设备可以根据3D模型的标识,在远程数据库中获取3D模型的远程版本号;若在远程数据库中获取到3D模型的远程版本号,电子设备可以在远程数据库中查找3D模型的远程版本号对应的远程加载模型;若在远程数据库中查找到3D模型的远程版本号对应的远程加载模型,电子设备可以使用3D模型的远程版本号对应的远程加载模型加载3D模型。

[0074] S204、根据3D模型的标识,在远程数据库中获取3D模型的远程版本号。

[0075] 在本发明的具体实施例中,电子设备可以根据3D模型的标识,在远程数据库中获取3D模型的远程版本号。具体地,电子设备可以在远程数据库中预先保存3D模型的标识与远程索引的对应关系;在本步骤中,电子设备可以根据3D模型的标识,在远程数据库中查找3D模型的远程索引。

[0076] S205、若在Localstorage中获取到3D模型的本地版本号并且在远程数据库中获取到3D模型的远程版本号,并且3D模型的本地版本号与3D模型的远程版本号相同,根据3D模型的本地版本号,在Indexdb中查找3D模型的本地版本号对应的本地加载模型。

[0077] 在本发明的具体实施例中,若在Localstorage中获取到3D模型的本地版本号并且在远程数据库中获取到3D模型的远程版本号,并且3D模型的本地版本号与3D模型的远程版本号相同,电子设备可以根据3D模型的本地版本号,在Indexdb中查找3D模型的本地版本号对应的本地加载模型。具体地,若在Localstorage中未获取到3D模型的本地版本号,而在远程数据库中获取到3D模型的远程版本号,电子设备可以使用3D模型的远程版本号对应的远程加载模型加载3D模型;同时在Localstorage中添加3D模型的远程版本号;在Indexdb中添加3D模型的远程版本号对应的远程加载模型;若在Localstorage中获取到3D模型的本地版本号,而在远程数据库中未获取到3D模型的远程版本号,电子设备在Localstorage中删除3D模型的本地版本号;在Indexdb中删除3D模型的本地版本号对应的本地加载模型,此时电子设备可以结束3D模型的加载流程。

[0078] 此外,若3D模型的本地版本号与3D模型的远程版本号不相同,电子设备可以根据3D模型的本地版本号,在Indexdb中查找3D模型的本地版本号对应的本地加载模型;若在Indexdb中查找到3D模型的本地版本号对应的本地加载模型,电子设备可以先使用3D模型的本地版本号对应的本地加载模型加载3D模型;然后根据3D模型的远程版本号,在远程数据库中查找3D模型的远程版本号对应的远程加载模型;若在远程数据库中查找到3D模型的远程版本号对应的远程加载模型,可以再使用3D模型的远程版本号对应的远程加载模型重新加载3D模型;同时在Indexdb中添加3D模型的远程版本号对应的远程加载模型;并在Indexdb中删除3D模型的本地版本号对应的本地加载模型。这样,用户可以先体验到旧版本的3D模型,然后将新版本的3D模型更新旧版本的3D模型,从而可以节省用户的等待时间。

[0079] 可选地,若3D模型的本地版本号与3D模型的远程版本号不相同,电子设备可以根据3D模型的远程版本号,在远程数据库中查找3D模型的远程版本号对应的远程加载模型;若在远程数据库中查找到3D模型的远程版本号对应的远程加载模型,电子设备可以使用3D模型的远程版本号对应的远程加载模型加载3D模型;同时在Indexdb中添加3D模型的远程版本号对应的远程加载模型;并在Indexdb中删除3D模型的本地版本号对应的本地加载模型。

这样,无需先加载旧版本的3D模型,可以简化加载流程,节约系统带宽。

[0080] S206、若在Indexdb中查找到3D模型的本地版本号对应的本地加载模型,使用3D模型的本地版本号对应的本地加载模型加载3D模型。

[0081] 在本发明的具体实施例中,若在Indexdb中查找到3D模型的本地版本号对应的本地加载模型,电子设备可以使用3D模型的本地版本号对应的本地加载模型加载3D模型。若在Indexdb中未查找到3D模型的本地版本号对应的本地加载模型,电子设备可以再根据3D模型的远程版本号,在远程数据库中查找3D模型的远程版本号对应的远程加载模型;若在远程数据库中查找到3D模型的远程版本号对应的远程加载模型,电子设备可以使用3D模型的远程版本号对应的远程加载模型加载3D模型;若在远程数据库中未查找到3D模型的远程版本号对应的远程加载模型,电子设备可以结束3D模型的加载流程。

[0082] 本发明实施例提出的3D模型的加载方法,先接收用户发送的3D模型的加载请求;然后根据加载请求中的3D模型的标识,在Localstorage中获取3D模型的本地版本号;根据3D模型的标识,在远程数据库中获取3D模型的远程版本号;若在Localstorage中获取到3D模型的本地版本号并且在远程数据库中获取到3D模型的远程版本号,并且3D模型的本地版本号与3D模型的远程版本号相同,根据3D模型的本地版本号,在Indexdb中查找3D模型的本地版本号对应的本地加载模型;若在Indexdb中查找到3D模型的本地版本号对应的本地加载模型,使用3D模型的本地版本号对应的本地加载模型加载3D模型。也就是说,在本发明的技术方案中,可以在Localstorage中获取到3D模型的本地版本号并且可以在远程数据库中获取到3D模型的远程版本号,并且3D模型的本地版本号与3D模型的远程版本号相同时,可以使用查找到的3D模型的本地版本号对应的本地加载模型加载3D模型。而在现有的3D模型的加载方法中,用户打开同样的网站,通过远程数据库重复加载模型;在大量模型的场景下,大量的重复加载不仅会造成用户等待时间延长,而且还会浪费远程数据库的带宽。因此,和现有技术相比,本发明实施例提出的3D模型的加载方法,不仅可以提高3D模型的加载速度,减少用户等待时间,而且还可以节约远程数据库带宽,降低3D模型的加载复杂度;并且,本发明实施例的技术方案实现简单方便、便于普及,适用范围更广。

[0083] 实施例三

[0084] 图3为本发明实施例三提供的3D模型的加载方法的流程示意图。如图3所示,3D模型的加载方法可以包括以下步骤:

[0085] S301、接收用户发送的3D模型的加载请求;其中,加载请求中至少包括3D模型的标识。

[0086] 在本发明的具体实施例中,电子设备可以接收用户发送的3D模型的加载请求;其中,加载请求中至少包括3D模型的标识。具体地,3D模型的标识可以是该模型的名称,也可以是该模型的编号,在此不做限定。

[0087] S302、根据3D模型的标识,在Localstorage中获取3D模型的本地版本号;根据3D模型的标识,在远程数据库中获取3D模型的远程版本号。

[0088] 在本发明的具体实施例中,电子设备可以根据3D模型的标识,在Localstorage中获取3D模型的本地版本号;根据3D模型的标识,在远程数据库中获取3D模型的远程版本号。具体地,Localstorage可以预先保存各个3D模型的本地索引、本地版本号以及URL;在本步骤中,电子设备可以根据3D模型的标识,在Localstorage中获取3D模型的本地索引、本地版

本号和URL。此外,远程数据库还可以预先保存各个3D模型的远程索引、远程版本号以及URL;在本步骤中,电子设备可以根据3D模型的标识,在远程数据库中获取3D模型的远程索引、远程版本号和URL。

[0089] S303、若在Localstorage中未获取到3D模型的本地版本号,而在远程数据库中获取到3D模型的远程版本号,使用3D模型的远程版本号对应的远程加载模型加载3D模型。

[0090] 在本发明的具体实施例中,若在Localstorage中未获取到3D模型的本地版本号,而在远程数据库中获取到3D模型的远程版本号,电子设备可以使用3D模型的远程版本号对应的远程加载模型加载3D模型。

[0091] S304、在Localstorage中添加3D模型的远程版本号;在Indexdb中添加3D模型的远程版本号对应的远程加载模型;并在Localstorage中删除3D模型的本地版本号;在Indexdb中删除3D模型的本地版本号对应的本地加载模型。

[0092] 在本发明的具体实施例中,电子设备在使用3D模型的远程版本号对应的远程加载模型加载3D模型之后,可以在Localstorage中添加3D模型的远程版本号;在Indexdb中添加3D模型的远程版本号对应的远程加载模型;并在Localstorage中删除3D模型的本地版本号;在Indexdb中删除3D模型的本地版本号对应的本地加载模型。具体地,电子设备可以将3D模型的远程版本号添加到Localstorage中,并为该版本号建立本地索引;将3D模型的数据添加到Indexdb中;同时在Localstorage中删除3D模型的本地版本号;在Indexdb中删除3D模型的本地版本号对应的本地加载模型。

[0093] 本发明实施例提出的3D模型的加载方法,1)不依赖具体的前端WebGL引擎:无论是自研引擎,还是使用诸如Three.js,Babylon等都是完全一致的解决方案;2)网页加载速度显著提高,服务器带宽流量显著节约:在BIM系统中加载20个3D模型,总体积约120M,在首次加载完成之后,对3D模型的本地版本信息进行2条数据变更。在使用本发明实施例的情况下,浏览器只重新加载了变更版本信息的2个3D模型。在不使用本发明的情况下,浏览器重新加载了20个3D模型。用户等待时间缩短为1/10。同样服务器的带宽消耗也变成了1/10。3)能够做到缓存可控:通过在chrome里面测试,编写10多个Demo页面。每个页面都有较多数量体积较小的模型。将一个Demo页面的模型总体积控制在20M左右。如果采用本发明提供的技术方案,则所有Demo在第一次打开之后,再次刷新页面打开,则无任何模型加载,网页直接渲染模型。在不使用本发明的情况下,只靠浏览器自动缓存,会发现第一个demo如果多次打开,则浏览器会缓存里面的各个模型。当不断的切换多个demo,会发现浏览器自动清除一部分之前Demo的模型。如果再次打开以前打开的页面,则浏览器仍然重新加载模型。而究竟哪些模型会被重新加载,哪些不会被重新加载,不受人控制,存在大量的不确定性因素,不确定性因素也会在中无形中对项目的稳定性产生影响,很多偶然出现的Bug也会诞生。而且浏览器种类很多,不同浏览器缓存的时机,自动清理缓存的时机和具体实现都会有差异,造成用户在使用不同浏览器的情况下,可能会产生体验效果不一致情况;4)支持多版本替换渲染:如果一个系统3D模型有更新操作,那么如果这个模型在未加载完成的时候是无法进行渲染的,这样在用户体验上就会造成这段时间里页面是空的,造成体验上的缺陷;而通常3D模型的更新是为了优化模型,例如减小模型体积,或者模型做的更精致,那么如果采用本发明提供的技术方案,则网页在新版本的模型未加载完成时,可以去渲染旧版本模型,这样在用户体验上,整个网页在一开始进去都有东西存在,在用户体验上有了提高;5)不会对前端

业务层造成复杂性的增加:本发明虽然需要用到前端的很多技术以及前后端的通信,这必定在整个项目上增加了一定的复杂度。但是实际情况是这个复杂度只是在引擎层面增加复杂度,而完全能做到前端业务开发层面无任何复杂度变化。甚至可以说对于业务开发的程序员是毫无感知的;例如,使用Three.js或者Babylon,只需要前端架构师使用Proxy对模型的Loader进行代理,预先进行版本列表的对比,对发生更新的3D模型进行请求即可;对于业务开发程序员来说,无需修改任何代码就做到了页面的优化;对于前端架构师来说,本发明无需性能高昂的算法计算并且实现难度不高,从而非常有利于项目实际使用;6) 支持多站点管理:浏览器的缓存机制会造成如果多个不同的站点都有好多数据需要被缓存,那么在浏览器缓存达到一定体积之后自动清理一部分页面历史较久的缓存。而使用本发明提供的技术方案,不同站点都是独立的,原因在于使用Localstorage去储存索引与版本数据。这样浏览器的每个页面模型信息都是独立的,不会产生任何相互干扰;7) 大幅提高网站的可用性:由于WebGL网页的资源可能会在各种cdn服务器或者公司自己的资源服务器,而服务器是存在宕机的概率。一旦资源服务器宕机就会造成网页无法渲染任何东西造成页面无法使用;本发明技术方案保证了用户浏览器里indexdb存有副本,可以直接使用本地副本让网站继续正常可使用;8) 减少企业的服务器成本:用户每次请求都会消耗服务器的资源。而对于像游戏类的网页,模型消耗极大的服务器资源。通过本发明,用户几乎不需要重复请求下载重复的模型资源,从而节省了大量请求次数和网络带宽压力,一台服务器就可以为更多的用户进行服务,这样就降低了企业的成本。

[0094] 实施例四

[0095] 图4为本发明实施例四提供的3D模型的加载方法的流程示意图。如图4所示,3D模型的加载方法可以包括以下步骤:

[0096] S401、接收用户发送的3D模型的加载请求;其中,加载请求中至少包括3D模型的标识。

[0097] 在本发明的具体实施例中,电子设备可以接收用户发送的3D模型的加载请求;其中,加载请求中至少包括3D模型的标识。具体地,3D模型的标识可以是该模型的名称,也可以是该模型的编号,在此不做限定。

[0098] S402、根据3D模型的标识,在本地存储Localstorage中获取3D模型的本地版本号;根据3D模型的标识,在远程数据库中获取3D模型的远程版本号。

[0099] 在本发明的具体实施例中,电子设备可以根据3D模型的标识,在Localstorage中获取3D模型的本地版本号;根据3D模型的标识,在远程数据库中获取3D模型的远程版本号。具体地,Localstorage可以预先保存各个3D模型的本地索引、本地版本号以及URL;在本步骤中,电子设备可以根据3D模型的标识,在Localstorage中获取3D模型的本地索引、本地版本号和URL。此外,远程数据库还可以预先保存各个3D模型的远程索引、远程版本号以及URL;在本步骤中,电子设备可以根据3D模型的标识,在远程数据库中获取3D模型的远程索引、远程版本号和URL。

[0100] S403、若在Localstorage中获取到3D模型的本地版本号,而在远程数据库中未获取到3D模型的远程版本号,在Localstorage中删除3D模型的本地版本号;在Indexdb中删除3D模型的本地版本号对应的本地加载模型。

[0101] 在本发明的具体实施例中,若在Localstorage中获取到3D模型的本地版本号,而

在远程数据库中未获取到3D模型的远程版本号,电子设备可以在Localstorage中删除3D模型的本地版本号;在Indexdb中删除3D模型的本地版本号对应的本地加载模型。具体地,电子设备可以先在Localstorage中删除3D模型的本地版本号;根据3D模型的本地版本号在Indexdb中查找3D模型的本地版本号对应的本地加载模型;若在Indexdb中查找3D模型的本地版本号对应的本地加载模型;在Indexdb中删除3D模型的本地版本号对应的本地加载模型;若在Indexdb中未查找3D模型的本地版本号对应的本地加载模型,结束3D模型的加载流程。

[0102] 本发明实施例提出的3D模型的加载方法,若在Localstorage中未获取到3D模型的本地版本号,而在远程数据库中获取到3D模型的远程版本号,使用3D模型的远程版本号对应的远程加载模型加载3D模型,可以使用3D模型的远程版本号对应的远程加载模型加载3D模型;若在Localstorage中获取到3D模型的本地版本号,而在远程数据库中未获取到3D模型的远程版本号,则结束3D模型的加载流程。和现有技术相比,本发明实施例提出的3D模型的加载方法,不仅可以提高3D模型的加载速度,减少用户等待时间,而且还可以节约远程数据库带宽,降低3D模型的加载复杂度;并且,本发明实施例的技术方案实现简单方便、便于普及,适用范围更广。

[0103] 实施例五

[0104] 图5为本发明实施例五提供的3D模型的加载装置的第一结构示意图。如图5所示,本发明实施例所述的3D模型的加载装置可以包括:接收模块501、获取模块502、查找模块503和加载模块504;其中,

[0105] 所述接收模块501,用于接收用户发送的3D模型的加载请求;其中,所述加载请求中至少包括所述3D模型的标识;

[0106] 所述获取模块502,用于根据所述3D模型的标识,在本地存储Localstorage中获取所述3D模型的本地版本号;根据所述3D模型的标识,在远程数据库中获取所述3D模型的远程版本号;

[0107] 所述查找模块503,用于若在所述Localstorage中获取到所述3D模型的本地版本号并且在所述远程数据库中获取到所述3D模型的远程版本号,并且所述3D模型的本地版本号与所述3D模型的远程版本号相同,根据所述3D模型的本地版本号,在本地数据库Indexdb中查找所述3D模型的本地版本号对应的本地加载模型;

[0108] 所述加载模块504,用于若在所述Indexdb中查找到所述3D模型的本地版本号对应的本地加载模型,使用所述3D模型的本地版本号对应的本地加载模型加载所述3D模型。

[0109] 图6为本发明实施例五提供的3D模型的加载装置的第二结构示意图。如图6所示,所述获取模块502包括:查找子模块5021和获取子模块5022;其中,

[0110] 所述查找子模块5021,用于根据所述3D模型的标识,在所述Localstorage中查找所述3D模型的本地索引;

[0111] 所述获取子模块5022,用于若在所述Localstorage中查找到所述3D模型的本地索引,根据所述3D模型的本地索引,在所述Localstorage中获取所述3D模型的本地版本号。

[0112] 进一步的,所述查找模块503,还用于若所述3D模型的本地版本号与所述3D模型的远程版本号不相同,根据所述3D模型的本地版本号,在所述Indexdb中查找所述3D模型的本地版本号对应的本地加载模型;

[0113] 所述加载模块504,还用于若在所述Indexdb中查找到所述3D模型的本地版本号对应的本地加载模型,使用所述3D模型的本地版本号对应的本地加载模型加载所述3D模型;

[0114] 所述查找模块503,还用于根据所述3D模型的远程版本号,在所述远程数据库中查找所述3D模型的远程版本号对应的远程加载模型;

[0115] 所述加载模块504,还用于若在所述远程数据库中查找到所述3D模型的远程版本号对应的远程加载模型,使用所述3D模型的远程版本号对应的远程加载模型重新加载所述3D模型;在所述Localstorage中添加所述3D模型的远程版本号;在所述Indexdb中添加所述3D模型的远程版本号对应的远程加载模型;并在所述Localstorage中删除所述3D模型的本地版本号;在所述Indexdb中删除所述3D模型的本地版本号对应的本地加载模型。

[0116] 进一步的,所述查找模块503,还用于若所述3D模型的本地版本号与所述3D模型的远程版本号不相同,根据所述3D模型的远程版本号,在所述远程数据库中查找所述3D模型的远程版本号对应的远程加载模型;

[0117] 所述加载模块504,还用于若在所述远程数据库中查找到所述3D模型的远程版本号对应的远程加载模型,使用所述3D模型的远程版本号对应的远程加载模型加载所述3D模型;在所述Localstorage中添加所述3D模型的远程版本号;在所述Indexdb中添加所述3D模型的远程版本号对应的远程加载模型;并在所述Localstorage中删除所述3D模型的本地版本号;在所述Indexdb中删除所述3D模型的本地版本号对应的本地加载模型。

[0118] 进一步的,所述加载模块504,还用于若在所述Localstorage中未获取到所述3D模型的本地版本号,而在所述远程数据库中获取到所述3D模型的远程版本号,使用所述3D模型的远程版本号对应的远程加载模型加载所述3D模型;在所述Localstorage中添加所述3D模型的远程版本号;在所述Indexdb中添加所述3D模型的远程版本号对应的远程加载模型;若在所述Localstorage中获取到所述3D模型的本地版本号,而在所述远程数据库中未获取到所述3D模型的远程版本号,在所述Localstorage中删除所述3D模型的本地版本号;在所述Indexdb中删除所述3D模型的本地版本号对应的本地加载模型。

[0119] 上述3D模型的加载装置可执行本发明任意实施例所提供的方法,具备执行方法相应的功能模块和有益效果。未在本实施例中详尽描述的技术细节,可参见本发明任意实施例提供的3D模型的加载方法。

[0120] 实施例六

[0121] 图7为本发明实施例六提供的电子设备的结构示意图。图7示出了适于用来实现本发明实施方式的示例性电子设备的框图。图7显示的电子设备12仅仅是一个示例,不应对本发明实施例的功能和使用范围带来任何限制。

[0122] 如图7所示,电子设备12以通用计算设备的形式表现。电子设备12的组件可以包括但不限于:一个或者多个处理器或者处理单元16,系统存储器28,连接不同系统组件(包括系统存储器28和处理单元16)的总线18。

[0123] 总线18表示几类总线结构中的一种或多种,包括存储器总线或者存储器控制器,外围总线,图形加速端口,处理器或者使用多种总线结构中的任意总线结构的局域总线。举例来说,这些体系结构包括但不限于工业标准体系结构(ISA)总线,微通道体系结构(MAC)总线,增强型ISA总线、视频电子标准协会(VESA)局域总线以及外围组件互连(PCI)总线。

[0124] 电子设备12典型地包括多种计算机系统可读介质。这些介质可以是任何能够被电

子设备12访问的可用介质,包括易失性和非易失性介质,可移动的和不可移动的介质。

[0125] 系统存储器28可以包括易失性存储器形式的计算机系统可读介质,例如随机存取存储器(RAM)30和/或高速缓存存储器32。电子设备12可以进一步包括其它可移动/不可移动的、易失性/非易失性计算机系统存储介质。仅作为举例,存储系统34可以用于读写不可移动的、非易失性磁介质(图7未显示,通常称为“硬盘驱动器”)。尽管图7中未示出,可以提供用于对可移动非易失性磁盘(例如“软盘”)读写的磁盘驱动器,以及对可移动非易失性光盘(例如CD-ROM,DVD-ROM或者其它光介质)读写的光盘驱动器。在这些情况下,每个驱动器可以通过一个或者多个数据介质接口与总线18相连。存储器28可以包括至少一个程序产品,该程序产品具有一组(例如至少一个)程序模块,这些程序模块被配置以执行本发明各实施例的功能。

[0126] 具有一组(至少一个)程序模块42的程序/实用工具40,可以存储在例如存储器28中,这样的程序模块42包括但不限于操作系统、一个或者多个应用程序、其它程序模块以及程序数据,这些示例中的每一个或某种组合中可能包括网络环境的实现。程序模块42通常执行本发明所描述的实施例中的功能和/或方法。

[0127] 电子设备12也可以与一个或多个外部设备14(例如键盘、指向设备、显示器24等)通信,还可与一个或者多个使得用户能与该电子设备12交互的设备通信,和/或与使得该电子设备12能与一个或多个其它计算设备进行通信的任何设备(例如网卡,调制解调器等等)通信。这种通信可以通过输入/输出(I/O)接口22进行。并且,电子设备12还可以通过网络适配器20与一个或者多个网络(例如局域网(LAN),广域网(WAN)和/或公共网络,例如因特网)通信。如图7所示,网络适配器20通过总线18与电子设备12的其它模块通信。应当明白,尽管图7中未示出,可以结合电子设备12使用其它硬件和/或软件模块,包括但不限于:微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、RAID系统、磁带驱动器以及数据备份存储系统等。

[0128] 处理单元16通过运行存储在系统存储器28中的程序,从而执行各种功能应用以及数据处理,例如实现本发明实施例所提供的3D模型的加载方法。

[0129] 实施例七

[0130] 本发明实施例七提供了一种计算机存储介质。

[0131] 本发明实施例的计算机可读存储介质,可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是一——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0132] 计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可

读存储介质以外的任何计算机可读介质,该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

[0133] 计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括——但不限于无线、电线、光缆、RF等等,或者上述的任意合适的组合。

[0134] 可以以一种或多种程序设计语言或其组合来编写用于执行本发明操作的计算机程序代码,所述程序设计语言包括面向对象的程序设计语言——诸如Java、Smalltalk、C++,还包括常规的过程式程序设计语言——诸如“C”语言或类似的设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络——包括局域网(LAN)或广域网(WAN)——连接到用户计算机,或者,可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0135] 注意,上述仅为本发明的较佳实施例及所运用技术原理。本领域技术人员会理解,本发明不限于这里所述的特定实施例,对本领域技术人员来说能够进行各种明显的变化、重新调整和替代而不会脱离本发明的保护范围。因此,虽然通过以上实施例对本发明进行了较为详细的说明,但是本发明不仅仅限于以上实施例,在不脱离本发明构思的情况下,还可以包括更多其他等效实施例,而本发明的范围由所附的权利要求范围决定。

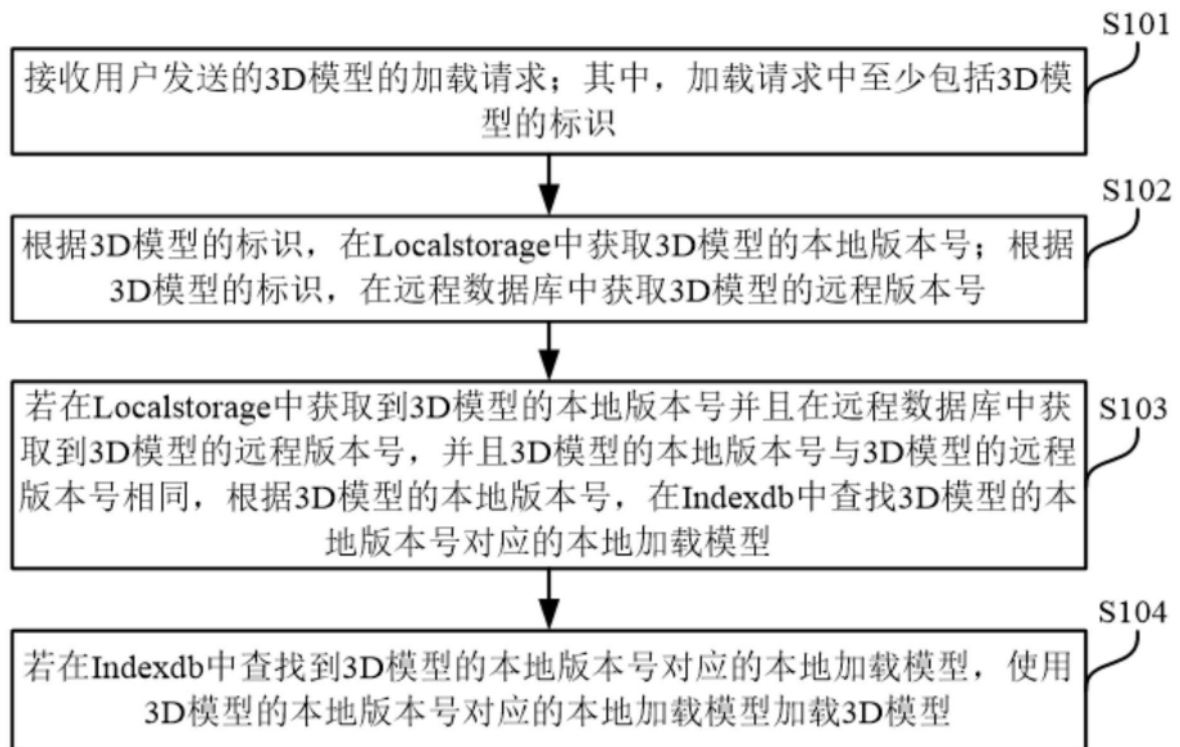


图1

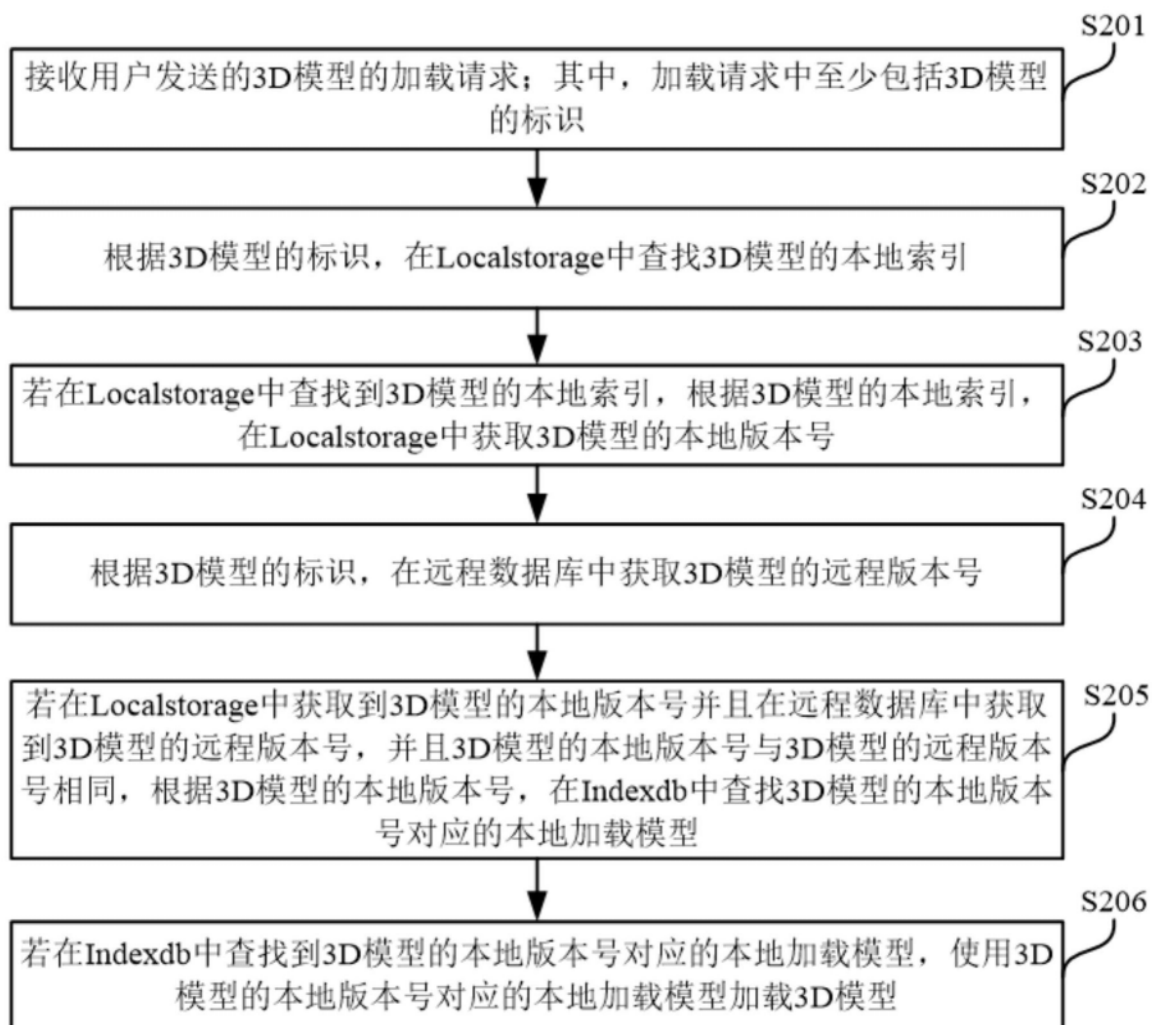


图2

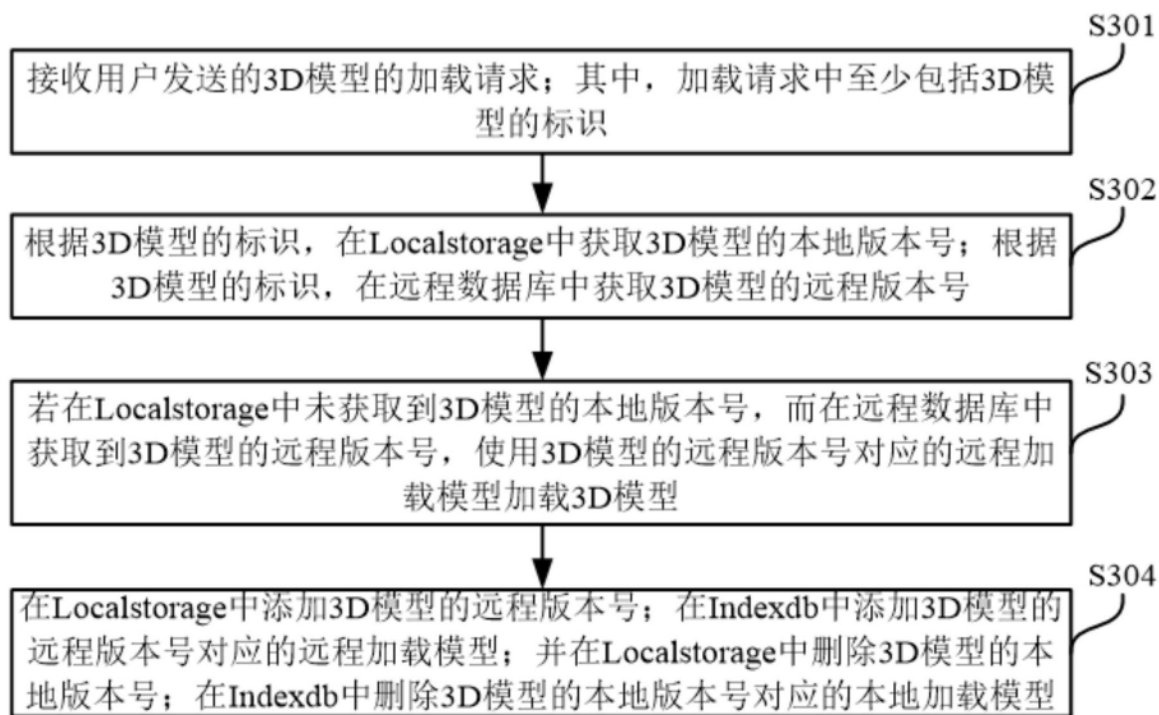


图3

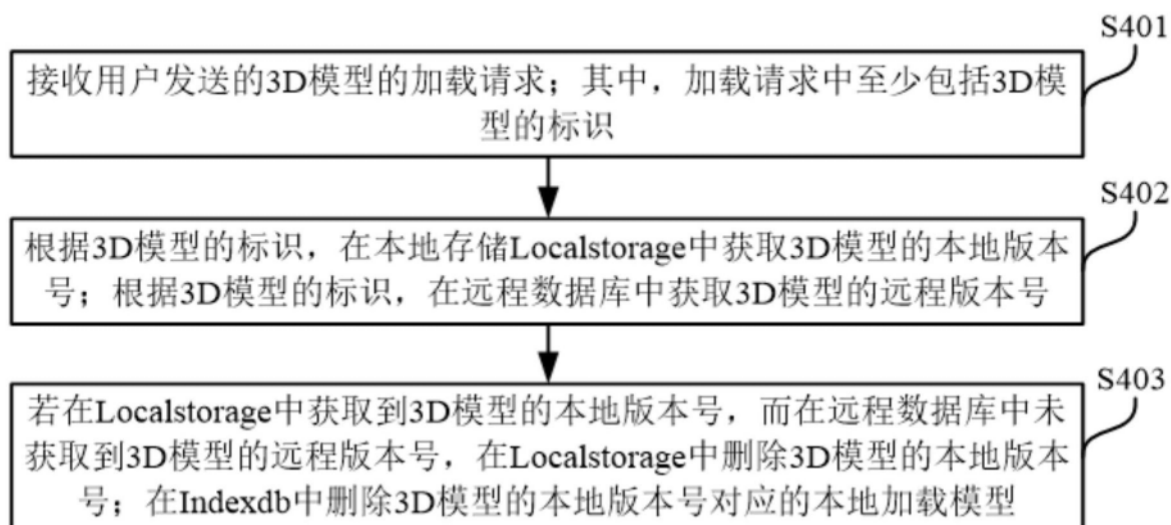


图4



图5

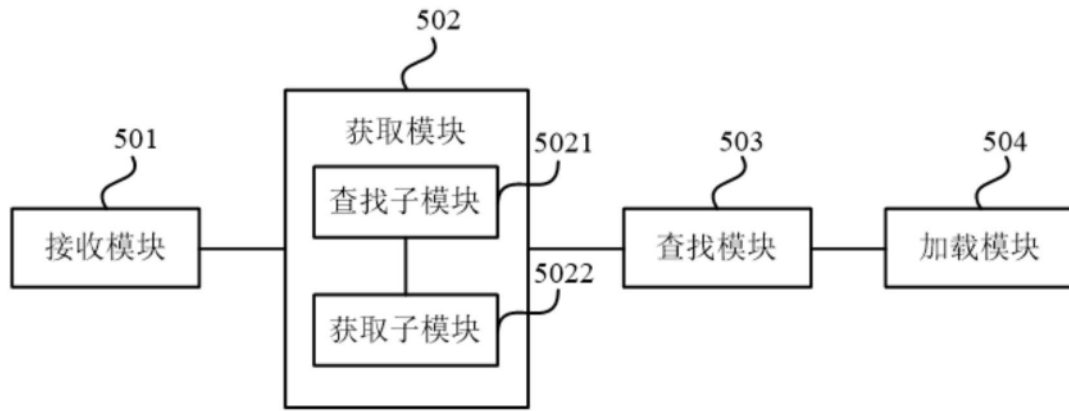


图6

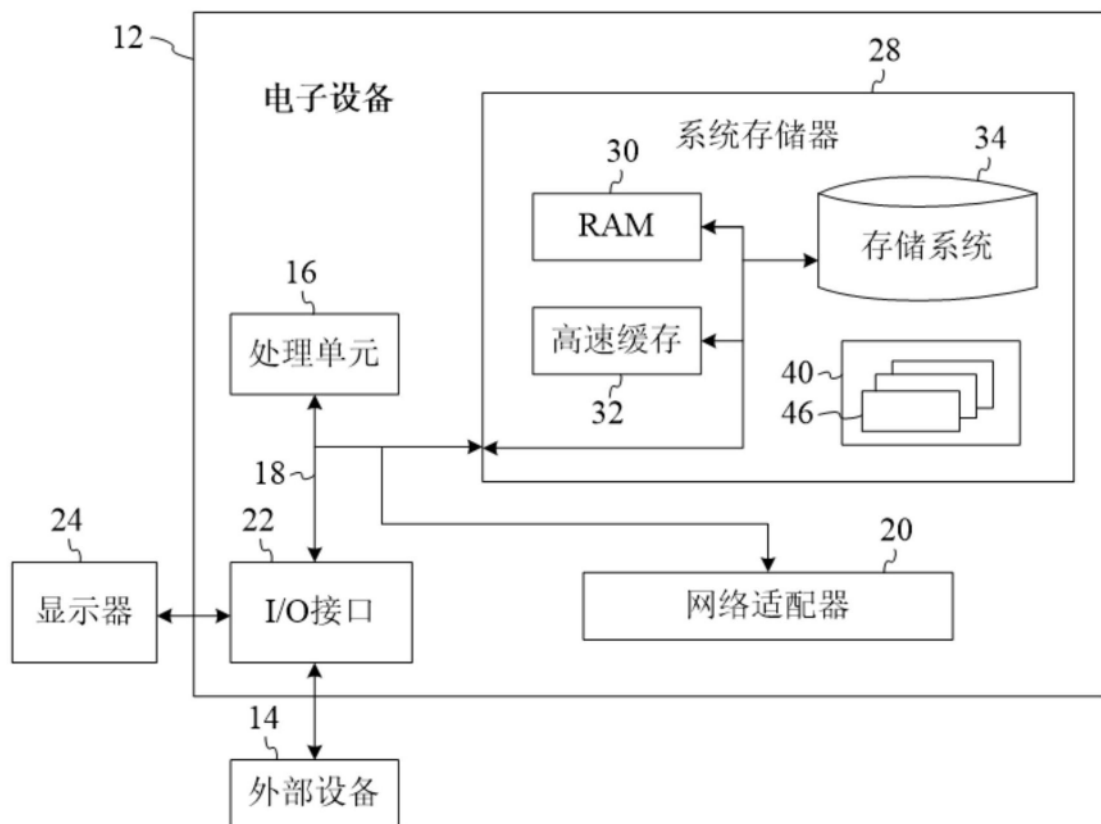


图7