



(12) 发明专利

(10) 授权公告号 CN 101067827 B

(45) 授权公告日 2010. 06. 09

(21) 申请号 200710118906. 3

Visual Communication and Image Processing

(22) 申请日 2007. 06. 14

2001 (VCIP 2001)4310 21.2001, 4310(21), 1-  
13.

(73) 专利权人 上海创图网络科技发展有限公司  
地址 200040 上海市静安区常熟路 88 号东  
艺大厦 402 室

审查员 覃婧婵

(72) 发明人 李欣

(74) 专利代理机构 北京市浩天知识产权代理事  
务所 11276  
代理人 许志勇

(51) Int. Cl.

G06F 17/30 (2006. 01)

G06T 15/00 (2006. 01)

(56) 对比文件

Cha Zhang, Jin Li. Interactive Browsing  
of 3D Environment over the Internet. SPIE

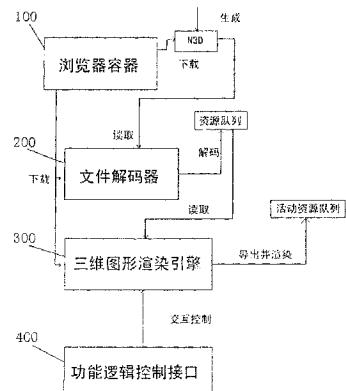
权利要求书 1 页 说明书 5 页 附图 2 页

(54) 发明名称

一种基于浏览器的三维网页实现方法

(57) 摘要

本发明提供一种基于浏览器的三维网页实现方法，采用如下的架构，该方法包括 4 个系统协同工作完成，分别为浏览器容器、文件解码器、三维图形渲染引擎和功能逻辑控制接口，基本流程包括以下步骤：a、采集转换模型数据，保存成一特定格式；b、利用浏览器容器动态加载文件解码器，三维图形渲染引擎以及三维网页文件；c、通过文件解码器将三维网页文件实时解码加入场景资源队列；d、启动三维图形渲染引擎读取资源队列实时生成活动资源队列并进行实时渲染；e、通过功能逻辑控制接口和浏览器实现互动。本发明采用这种方式提高了三维网页的互动性以及可操作性，同时减少三维文件的文件量，减少了不必要的资源开销。



1. 一种基于浏览器的三维网页实现方法，其特征在于，基本实现流程包括以下 5 个步骤：

- a、采集转换模型数据，采用动态哈夫曼编码进行数据的保存；
- b、利用浏览器容器动态加载文件解码器，三维图形渲染引擎以及三维网页文件；
- c、通过文件解码器将三维网页文件实时解码加入场景资源队列；
- d、启动三维图形渲染引擎读取场景资源队列实时生成活动资源队列并进行实时渲染；
- e、通过功能逻辑控制接口和浏览器实现互动。

2. 根据权利要求 1 所述的一种基于浏览器的三维网页实现方法，其特征在于，所述步骤 b 中采用浏览器容器的方法是利用在浏览器中嵌入一个体积非常小的容器控件实现按需要动态多线程的加载三维图形渲染引擎，文件解码器，三维网页文件。

3. 根据权利要求 1 所述的一种基于浏览器的三维网页实现方法，其特征在于，在 b 步骤进行的同时，在网页文件被加载的同时，c、d 两个步骤亦开始运行，拆分采用动态哈夫曼编码进行保存的数据，通过文件解码器进行实时地解码，并送达三维图形渲染引擎进行计算并输出。

4. 根据权利要求 1 所述的一种基于浏览器的三维网页实现方法，其特征在于，所述三维网页文件在渲染过程中，通过网页中的功能控制接口函数实现实时的浏览交互功能。

## 一种基于浏览器的三维网页实现方法

### 技术领域

[0001] 本发明涉及计算机科学技术领域，特别是涉及一种三维网页实现方法。

### 背景技术

[0002] 随着网络技术和计算机科学飞速的发展带动了互联网经济的爆炸式增长，现如今的互联网内容和形式都已经和5年前不能同日而语。在这个眼球经济的年代，谁可以以其独特的方式吸引眼球，谁便是互联网的主宰。

[0003] 如今的互联网内容提供商们采用各种各样的形式来表现其所需要展现的内容，图文，动画(flash)，视频等等。不过这些形式均存在局限性，人们生活的这个世界是三维空间化的，而在网络上看到内容的依然停留于平面的形式，三维网页技术的诞生为改变这种局面提供了可能性。

[0004] 三维网页技术的英文名称为 web3d，是虚拟现实技术在网络上的延伸，是基于网络的 3D 图形渲染技术。最早的 web3d 标准为虚拟现实造型语言 (VRML, Virtual Reality Modeling Language)，是由美国 SGI 公司开发的，它是因特网 (internet) 上基于万维网 (WWW, World Wide Web) 的具有交互性的虚拟现实建模语言，是超文本链接标示语言 (HTML, Hypertext Markup Language) 的三维模拟。但由于它是基于解释语言机制，因此运行效率不高，后来又相序出现了几种 web3d 技术，如瑞典 Cycore 公司的 Cult3D、美国 Viewpoint 公司的 Viewpoint、以及 Adobe 公司的 Shockwave3D 技术，国内目前在该领域还没有相关的同类技术。

[0005] Cult3D 基于 java 内核本身就决定了其在渲染效率上的天生不足；Viewpoint 使用的是软件渲染的方式虽然解决了对显示硬件的依赖，但却要耗费大量的 CPU 运算时间；Shockwave3D 由于是内置在 Adobe 公司的 Shockwave 引擎中，需要通过其自己定义的脚本语言 lingo 来进行控制，因此执行效率也是大打折扣，因此这几种技术对于大规模的三维化场景都存在运行效率低的问题，对图形硬件设备的要求比较高；同时，这几种引擎，其自身的引擎控件文件都相对较大，最小的是 Cult3D 的控件有 1.5M，Viewpoint 有 2.7M，Shockwave 需要 4M。下载插件便需要一段时间。同时运行的文件虽然其声称都进行了压缩，但是经过测试，对于比较大规模的场景压缩的比例并不高，而且也不太适合现在的网络环境。

[0006] 三维网页面对的受众群体是所有网民，也就是 internet 所延伸到每个地方，现在虽然计算机的硬件性能在不断提升，网络速度也在不断加快，但是三维化本身数据量就大，而且对硬件要求相对较高，目前的主流硬件环境和网络速度环境对其的支持还是有限。因此要在现有的硬件和网络环境下，要实现真正的模拟真实世界的三维互动内容，必须突破硬件渲染和网速的限制，开发出一种高压缩比，可以高速率传输的文件标准，同时可以利用有限硬件资源实现高速三维图形的逼真渲染技术。

## 发明内容

[0007] 针对以上问题，本发明的目的在于提供一种基于浏览器的三维网页实现方法，该方法克服了现有技术中场景文件大，浏览器引擎的插件文件大，以及运行效率低的问题。通过该方法实现的三维网页既适用于目前的准宽带的网络环境，同时又可以保证在如今普遍的计算机硬件上实现流畅的三维化效果。

[0008] 为了实现上述目标，本发明描述的一种基于浏览器的三维网页实现方法，采用如下的架构，该方法包括 4 个系统协同工作完成，分别为浏览器容器，文件解码器，三维图形渲染引擎和功能逻辑控制接口。基本的实现流程包括以下 5 个步骤：

- [0009] a、采集转换模型数据，采用动态哈夫曼编码进行数据的保存；
- [0010] b、利用浏览器容器动态加载文件解码器，三维图形渲染引擎以及三维网页文件；
- [0011] c、通过文件解码器将三维网页文件实时解码加入场景资源队列；
- [0012] d、启动三维图形渲染引擎读取场景资源队列实时生成活动资源队列并进行实时渲染；
- [0013] e、通过功能逻辑控制接口和浏览器实现互动。

[0014] 其中所述步骤 a 采集转换模型数据是将通用的但文件量较大的标准数据格式如 3dMax 文件格式转换成为网络浏览而特殊制定的文件结构，如本发明采用动态哈夫曼编码进行数据的保存，最终保存成 N3D（本发明定义的三维场景网页文件的后缀名）格式。

[0015] 所述步骤 b 中采用浏览器容器的方法不同于其他三维虚拟现实软件中，将整个软件打包在一个插件安装文件中的做法，而是利用在浏览器中嵌入一个体积非常小的容器控件实现按需要动态多线程的加载三维图形渲染，文件解码器，三维网页文件。该控件由于体积非常小，所以基本上在启动浏览器的同时即可加载，避免了向目前多数浏览器插件安装需要占用很长加载时间的不足，同时容器采用了多线程拆分传输的下载方式，缩短了引擎的安装过程以及网页文件加载时间。

[0016] 在 b 步骤进行的同时，在网页文件被加载的同时，c、d 两个步骤亦开始运行，拆分采用动态哈夫曼编码进行保存的数据，通过文件解码器进行实时地解码，并送达三维图形渲染引擎进行计算并输出，因此不需要等待整个网页文件下载完毕，即可在浏览器中实现对下载内容的浏览，进一步缩短了等待内容的时间，为浏览者提供更高效的浏览体验。三维内容在渲染过程中，通过网页中的功能控制接口函数实现实时的浏览交互功能。

[0017] 本发明采用这种方式突破了传统性将逻辑控制封装在加载的三维文件的方式（如 Viewpoint, shockwave 以及 cult3d），提高了三维网页的互动性以及可操作性，同时减少三维文件的文件量，减少了不必要的资源开销。

## 附图说明

- [0018] 图 1 是本发明提供的实现三维网页的架构示意图；
- [0019] 图 2 是描述本发明实现的 n3d 文件组织结构图；
- [0020] 图 3 是本发明中资源队列组织结构图。

## 具体实施方式

- [0021] 下面结合附图对本发明具体实施方式作进一步详细描述：
- [0022] 本发明描述的一种基于浏览器的三维网页实现方法，采用如下的架构，如图 1 所

示。该方法包括 4 个系统协同工作完成，分别为浏览器容器 100，文件解码器 200，三维图形渲染引擎 300 和功能逻辑控制接口 400。基本的实现流程包括以下步骤：

- [0023] a、采集转换模型数据，保存成 n3d 格式；
- [0024] b、利用浏览器容器 100 动态加载文件解码器 200，三维图形渲染引擎 300 以及三维网页文件；
- [0025] c、通过文件解码器 200 将三维网页文件实时解码加入场景资源队列；
- [0026] d、启动三维图形渲染引擎 300 读取资源队列实时生成活动资源队列并进行实时渲染；
- [0027] e、通过功能逻辑控制接口 400 和浏览器实现互动。

[0028] 其中步骤 a 中，所述采集转换模型数据的过程通过读取行业通用的模型数据并将其内容抽取，并重新表达成一套自定义的场景描述语言，该描述语言原理是将复杂的场景文件按场景、模型、顶点、材质、贴图进行特定的树状索引。其文件组织方式如图 2 所示，描述以场景 500 作为文件头以及根节点，场景节点下有 4 个子树，分别为模型树 510、材质树 520、顶点树 530、辅助动态内容信息树 540。每个树的树根都采用唯一的标识，以便在读取文件时实现快速的查询。模型树 510 下分索引模型树 511 和非索引模型树 512，索引模型树 511 记录所有被复制模型的索引号信息、模型名称信息、复制模型的索引号信息和空间变换信息，非索引模型树 512 记录模型的索引号信息、顶点索引号序列、材质索引号、贴图坐标。通过该种结构使有重复模型的文件中不再重复记录大量的模型信息，尤其是顶点索引号序列信息。一般的三维引擎处理模型时都会记录每个模型的所有顶点位置，这样处理在导入场景后会减少显卡的运算量，提高运算效率，但这样对于有存在共点信息的模型来说在文件上就会记录冗余的顶点信息，因此该发明采用将模型树 510 和顶点树 530 分开的方式，在模型树 510 中只记录顶点的索引号，在顶点树 530 中只记录了所有没有重复位置的顶点的索引号和坐标信息。材质树 520 分为材质 521 和贴图 522 两个子树，材质子树 521 记录材质索引号、材质名称以及贴图索引号，贴图子树 522 记录贴图的索引号信息、贴图名称以及贴图的图像信息，辅助动态内容信息树 540 记录帧关键帧的动画信息、电影贴图信息、特效信息、导航图、前景图、按钮以及外加控件信息，如调用动画、音频、视频等非三维格式的控制信息。文件在被保存的过程中采用基于视锥定位的排列方法，即先将场景中相机信息以一个特殊模型得方式记录在模型树中，接下来以默认相机浏览的起点为轴将空间分割，每 90 度为一个区域，顺时针旋转，依次将该区域的模型的索引号信息记录到模型树中，将模型的所有顶点使用递归的方式检查是否是重复顶点，如果不是重复顶点则在顶点信息树中记录顶点信息，并在模型树中为该模型索引号添加一个顶点索引号，如果是则只在模型树中添加顶点索引号信息。接下来将模型的材质索引号记录到模型树中，模型使用的贴图同样使用递归的方式检查是否在材质树中有重复的记录，如果没有则在材质树中添加一条记录。如果该模型使用到辅助信息如电影贴图则在辅助动态内容信息书中添加相应的记录。为了获得快速的数据压缩和解码速度，本发明采用了动态哈夫曼编码进行数据的保存，最终保存成 N3D（本发明定义的三维场景网页文件的后缀名）格式，该方法可以保证一次性遍历存储，同时具有理想的编解码速率。并将文件中 4 个树的根节点的编码后的偏移地址作为文件头保存。

[0029] 所述步骤 b 中，浏览器容器 100 提供了对整个三维网页文件系统的调度，浏览器容

器 100 的安装以插件压缩包的形式进行下载,文件的大小为 69k,利用 54k 的调制解调器接入网络的下载时间在 2 到 3 秒钟左右,在该浏览器容器 100 安装成功后,将驻留在系统中无需再次安装,浏览器容器 100 运行后,将根据网页中的具体需求,检查本地的三维引擎版本,如果本地没有运行过该版本的引擎,容器自动向服务器请求安装,并根据网络环境启动最多 10 个下载线程的多线程下载。当引擎文件下载完成后,浏览器容器 100 启动引擎自动完成在本地的安装,通过多线程下载控制,极大程度了利用了网络的带宽资源,缩短了用户在安装插件时的等待时间,使用户因为等待时间过长而对插件产生厌恶的问题迎刃而解。

[0030] 在引擎成功安装或引擎已经在本地安装后,调用网页脚本开启引擎的方法执行相应功能操作。当开启引擎的方法中调用导入三维网页内容文件的方法时,浏览器容器 100 启动文件解码器 200,并且从目标服务器下载三维网页内容。浏览器容器 100 和文件解码器 200 同时工作,浏览器容器 100 读取文件头,获得各子树的偏移地址,并同时启动 4 个下载线程从 4 个树的偏移地址下载文件并保存到 4 个缓存空间中。文件解码器 200 在下载的同时读取 4 个缓存空间的文件进行解码。如果服务器控件对多线程下载进行了限制,只允许使用单线程,浏览器容器 100 将采用分时轮转的方式,依次在一定时间片内下载 4 个树的内容并进行解码。解码的过程分两步,第一步遍历哈夫曼树读取每个子节点的数据,还原三维网页的场景内容。第二步是重新组织三维场景信息,创建临时的缓存文件,即场景资源队列,该队列按如下方法,如图 3 所示,仍然以场景 500 作为根节点,场景 500 下只分三个树,模型树 510,材质树 520 和辅助动态内容信息树 530,由于三维场景在渲染得每一帧都要将模型的顶点信息送入渲染管线进行计算,所以如果仍保留索引模型的信息将在每一帧都要在顶点列表中查找相应的顶点,这将消耗很大的系统资源以及事件,会降低渲染的效率,因此本发明在解码的过程中,在解码模型树 510 的线程获得模型信息后,即等待解码顶点树的线程获取所有该模型的顶点信息,如果是非索引模型则将其作为模型树中模型的属性填入到模型树中,如果是索引模型,则在读取空间变换信息后将其顶点进行空间变换后的值作为属性存入模型树的某个模型信息下。这样就可以保证在渲染过程中无需再进行大量的空间变换计算,直接向渲染器输出特定的顶点即可渲染即可。这样减小了硬件的压力,在低配置的硬件设备(集成或 TNT2 级别的显卡)上也可以保证每秒 25 帧的渲染效率。

[0031] 文件在解码的过程中,三维图形渲染引擎 300 已经开始工作,在其启动时,会为场景渲染创建一块场景活动资源队列。该队列动态的保存在视锥范围内的模型资源的信息。当解码器已经释放了一个模型的所有顶点信息时到场景资源队列时,三维渲染引擎 300 会进行基于视锥的可视性检查,如果该模型在视锥范围内则其信息将被插入活动资源队列,由渲染器便将其直接送入渲染管线进行输出。在浏览器中即可看到相应的模型出现。通过此方法场景实现了流式加载,当解码器释放的模型不在视锥范围内时则只讲起插入场景资源队列而不插入活动资源队列。在文件解码结束后,活动资源队列中即保存了所有视锥范围内的模型信息。浏览者也看到了整个场景。当浏览者进行漫游操作,视锥实时发生变化时,渲染器会自动刷新活动资源队列,插入新的进入视锥的模型信息,删除不在视锥范围的模型信息,该方法保证了显卡的高效运算,即指输出范围固定的看得见的场景,因此对于超大规模的场景,低配置显卡一样运行顺畅。

[0032] 本发明采用外部脚本的方式实现三维网页的交互功能,通过一组脚本函数实现对三维场景的各种控制功能。外部的脚本充分发挥了网页文件的开放性,同时脚本函数是执

行时才调用的,不需要占用系统资源,既缩小了网页文件的容量,又在运行时降低了系统的开销。提升了三维网页的运行效率。使浏览者可以更加流畅的体验三维网页所带来的突破传统的更佳贴近真实生活的内容。

[0033] 本发明实现了在浏览器中交互式的浏览丰富多彩的三维内容的需求,在效果及运算效率上达到了以下指标:

[0034] ◆网页的插件仅 69k,在 56kModem 接入网络的环境下安装插件仅需要 2 到 3 秒钟;

[0035] ◆三维渲染引擎和文件解码器文件量仅 363k,在 56k 调制解调器 (Modem) 接入网络的环境下安装插件仅需要 20 秒钟以内的时间;

[0036] ◆三维网页文件的压缩比的平均值可以达到 100 : 1,对于重复度高的大规模场景,其压缩比可以达到 200 : 1 以上;

[0037] ◆在 CPU :Celeron 1. 2G, 内存 :256M, 显卡 :Intel 845GVGraphics Controller(主板自带) 运行 100 万个带贴图的多边形场景,可以运行 30FPS 以上的渲染效率。

[0038] 另外再举例说明,在 3D Max 中建立 100 个材质相同的 segment 为 100 的球体面片,总面片数达到 160 万,其文件的容量为 69. 4M ;但采用本发明提供的方法将其保存为 N3D 文件后,其文件容量仅有 307K。

[0039] 本发明采用这种方式突破了传统性将逻辑控制封装在加载的三维文件的方式,提高了三维网页的互动性以及可操作性,同时减少三维文件的文件量,减少了不必要的资源开销。

[0040] 以上所述的实施例仅仅是对本发明的优选实施方式进行描述,并非对本发明的构思和范围进行限定,在不脱离本发明设计构思前提下,本领域中普通技术人员对本发明的技术方案作出的各种变型和改进,均应落入本发明的保护范围。

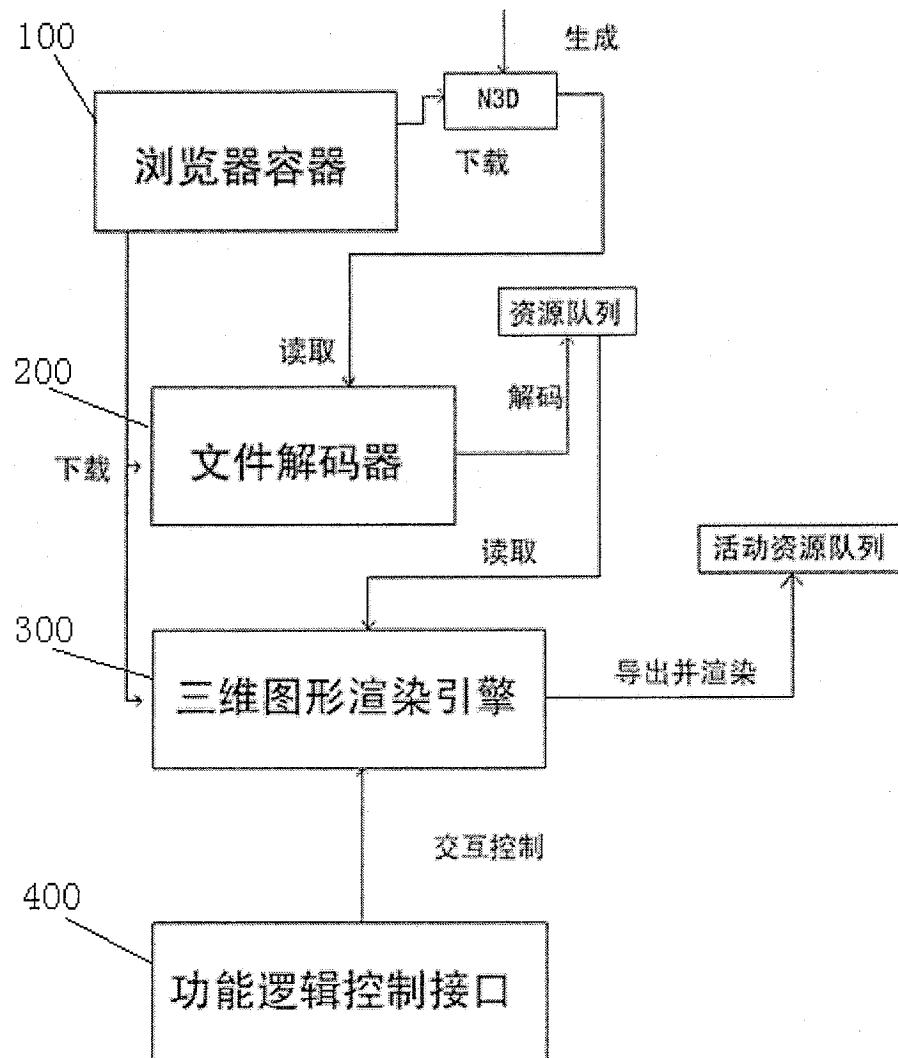


图 1

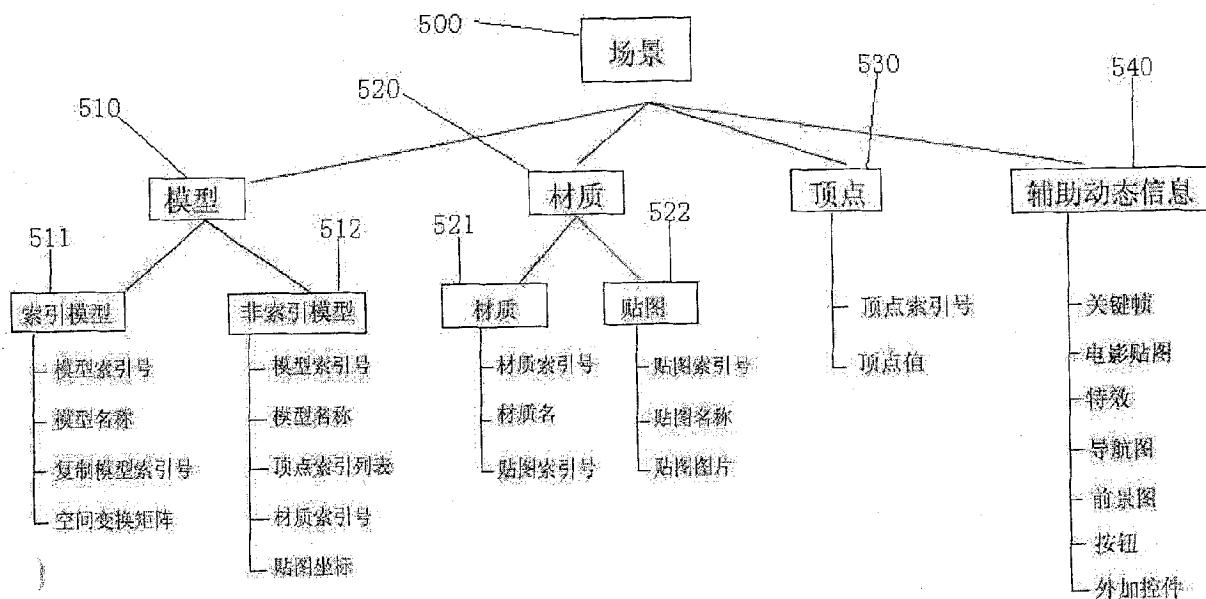


图 2

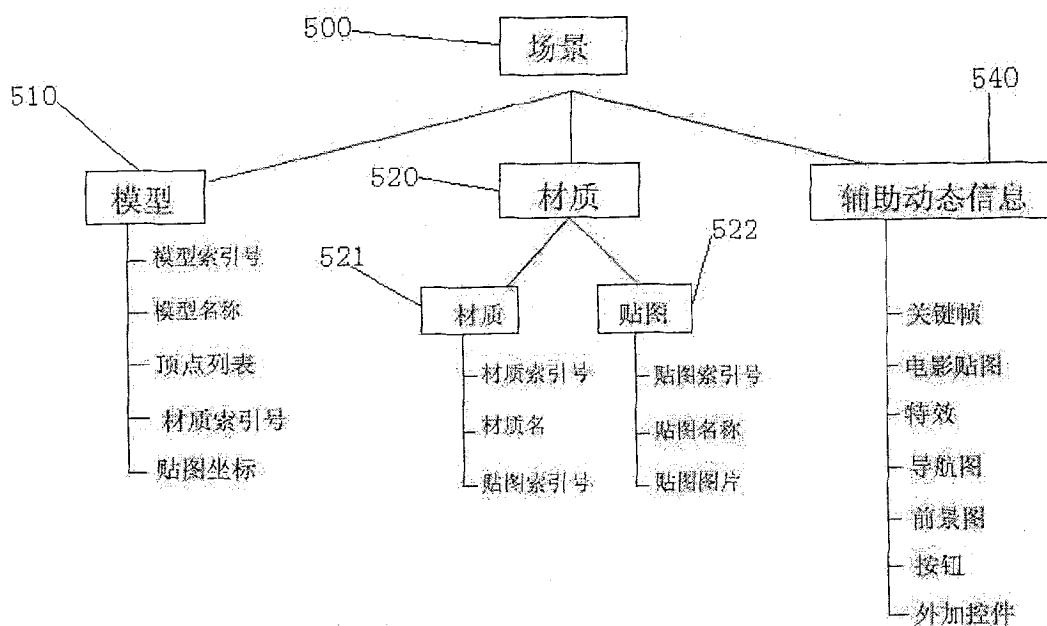


图 3