



(12)发明专利

(10)授权公告号 CN 107871338 B

(45)授权公告日 2019.12.03

(21)申请号 201610856078.2

G06T 15/50(2011.01)

(22)申请日 2016.09.27

审查员 刘芬

(65)同一申请的已公布的文献号

申请公布号 CN 107871338 A

(43)申请公布日 2018.04.03

(73)专利权人 重庆完美空间科技有限公司

地址 400043 重庆市渝中区瑞天路56-2号
19层2单元

(72)发明人 向海明 张程

(74)专利代理机构 重庆中之信知识产权代理事

务所(普通合伙) 50213

代理人 张景根

(51)Int.Cl.

G06T 15/00(2011.01)

G06T 15/04(2011.01)

权利要求书3页 说明书8页 附图2页

(54)发明名称

基于场景装饰的实时交互渲染方法

(57)摘要

本发明提供了一种基于场景装饰的实时交互渲染方法,包括以下步骤:构建后模型导出、模型预处理、模型场景化、关键模型标注、模型数据流传输、WebGL流式渲染与本地缓存;在空间环境中心放置一拍摄镜头,拍摄镜头依次以30度步进作为拍摄点在空间环境内360度旋转连续拍摄出多个场景图片,将拍摄出每个场景图片进行保存;对于多个场景图片采用上下文的多核SVM场景识别算法;采用标准的Web流传输方式;先传输配置文件,然后整个空间环境材质信息和光源数据信息,再次依据部件关键性从大到小传输空间环境中各部件模型及次要模型;接下来传送该初始场景相邻场景数据,最后传送剩余数据。以上方法显著减少实时渲染的等待时间;提高实时渲染的效率和效果。



1. 一种基于场景装饰的实时交互渲染方法,其特征在于,包括以下步骤:

S1、模型导出,将空间环境及其内部各个部件建模、贴材质和渲染效果预览后的3D模型导出;首先导出整个3D模型所在的整个文件,然后分别导出空间环境中的各个部件模型和材质图片,同时每个部件模型的坐标原点定位于其自身;

S2、模型预处理,将步骤S1导出的各个部件模型中的各类信息分别单独存储在数据库及配置文件中;

S3、模型场景化,在空间环境中心放置一个拍摄镜头,拍摄镜头位置固定,依次以30度旋转步进在空间环境内360度旋转连续拍摄出多个场景图片,拍摄出每个场景图片进行保存;

对于多个场景图片采用上下文的多核SVM场景识别算法,该算法以0度视角保存的场景图片作为初始场景,使每个场景图片存在显式的上下文关系,通过马尔可夫模型建立起多个场景图片的上下文关系;再通过整个场景中上下文的贝叶斯分类算法提取不同角度拍摄的场景中的部件并进行识别和标记;

S4、模型标注,将步骤S3中识别的部件与步骤S1中导出的独立部件模型之间建立对应关系,再将提取的部件与独立部件模型比对,从而得出重复部件并在关键的独立部件模型上进行模型标注;

S5、模型数据流传输,采用标准的Web流传输方式;

用户通过web发送对整个空间环境的3D渲染预览请求,服务端调出该空间环境的所有模型文件和该空间环境的配置文件集;然后通过网络传送到前端;

先传输配置文件,然后传输整个空间环境材质信息和光源数据信息,再次依据部件关键性从大到小传输空间环境中各部件模型及次要模型;接下来再传送该初始场景相邻场景数据,最后传送剩余数据;

S6、WebGL流式渲染与本地缓存

Web前端接收到配置文件数据和材质数据后,首先通过3D渲染代码在前端重构空间环境;接下来渲染初始场景的关键模型和次要模型及初始模型邻接的其它场景角落;然后,WebGL流式渲染开始初始场景细节化和精细化;后台继续接受模型数据缓存。

2. 根据权利要求1中所述的基于场景装饰的实时交互渲染方法,其特征在于:所述的多个场景的上下文关系在建立时,先根据场景类别和图像特征建立一阶马尔可夫公式: $P(Q_t | Q_{1:t-1}) = P(Q_t | Q_{t-1})$;其中 Q_t 和 x_t 分别代表 t 角度时图像对应的场景类别和特征, Q_t 和 Q_{t-1} 之间是相差 30° 的镜头视角;

根据以上关系建立了多个场景上下文关系公式:

$$P(Q_t = q | x_{1:t}) \propto P(x_t | Q_t = q) P(Q_t = q | x_{1:t-1}) = P(x_t | Q_t = q) \sum_{q'} A(q', q) p(Q_{t-1} = q' | x_{1:t-1})$$

其中, $\sum_{q'} A(q', q) p(Q_{t-1} = q' | x_{1:t-1})$ 是马尔可夫概率矩阵, $P(x_t | Q_t)$ 是观测率。

3. 根据权利要求2所述的基于场景装饰的实时交互渲染方法,其特征在于:所述场景中部件的识别采用基于图像分割的物体识别算法。

4. 根据权利要求3所述的基于场景装饰的实时交互渲染方法,其特征在于:所述的基于图像分割的物体识别算法;其中,图像的像素集合为 x ,所有相邻像素的集合为 N ,设 $C =$

$(C_1, \dots, C_t, \dots, C_x)$ 是该图像的一个分割, $C_i = x_B$ 表示第 i 个像素属于背景, $C_i = x_F$ 表示第 i 个像素属于前景; 该图像分割的总势能 $E_{(x)}$ 、单元势能 $U_{(x)}$ 、点对势能 $V_{(x)}$ 分别表述如下:

$$E_{(x)} = U_{(x)} + \lambda V_{(x)}$$

$$U_{(x)} = \sum_{i \in x} \varphi_i(x_i)$$

$$V_{(x)} = \sum_{(i,j) \in N} V_{(i,j)} \sigma(x_i, x_j)$$

$$\sigma(x_i, x_j) = \begin{cases} 1 & x_i \neq x_j \\ 0 & x_i = x_j \end{cases}$$

其中, $U_{(x)}$ 代表单元势能, 每个像素 i 分割为 x_i 所花费的代价; $V_{(x)}$ 代表点对势能, 反映像素集中任意两个相邻像素 i 和 j , 当分割不连续, 即 $x_i \neq x_j$ 时所花费的代价; i 和 j 越相似, 则 $V_{(i,j)}$ 越小; 反之, 则 $V_{(i,j)}$ 越大。

5. 根据权利要求4所述的基于场景装饰的实时交互渲染方法, 其特征在于: 所述的基于图像分割的物体识别算法在进行图像分割时, 采用以下方法:

a. 初始化, 取整图区域作为初始区域, 生成第一个初始分割, 其前景部分为未知, 背景像素点为背景;

b. 前景和背景通过 K -均值的高斯混合建模, 构建一个图用以找到一致的前景和背景像素点;

c. 定义背景控制阈值和最大循环次数, 再逐步减小方框区域直至识别出部件。

6. 根据权利要求5所述的基于场景装饰的实时交互渲染方法, 其特征在于: 所述的重复部件标注, 采用以下方法:

a. 当两个部件存在上下关系时, 下部部件为重复部件;

b. 当一部件在两个场景甚至多个连续场景中出现时, 该部件是重复部件, 在不同场景中出现频率越高, 则关键值越大;

c. 部件的关键值具有场景标签, 即 K_i^m 代表部件 m 在第 i 个场景中的关键值。

7. 根据权利要求6所述的基于场景装饰的实时交互渲染方法, 其特征在于: 将步骤S3识别的部件图片进一步处理使获得部件图片与空间环境内单独部件匹配, 以自动识别部件的具体品名, 首先对每个具体部件建立视觉直方图, 然后对步骤S3中获得的处理后的图像也建立视觉直方图; 最后再通过比对判断步骤S3中提取的部件图像确定具体部件。

8. 根据权利要求7所述的基于场景装饰的实时交互渲染方法, 其特征在于: 所述特征采用以下方法表示整幅图像:

(1) 首先定义全局图像的BoW特征;

(2) 将图像按照空间金字塔形式划分为子图像并计算BoW特征;

(3) 最后拼接金字塔各层的所有BoW特征来表示整幅图像;

设定图像数据集 $D = D_{\text{train}} \cup D_{\text{test}}$, 其中图像子集 D_{train} 用来训练分类集, 图像子集 D_{test} 用作测试, 通过规则网格法进行特征检测, 用均匀间隔水平和垂直的直线划分图像, 对形成的局部区域进行特征检测; 完成特征检测之后, 图像被抽象成为了若干个局部区域, 并对形成的局部区域加以特征描述, 将其转换为数值向量; 每幅图像用维数相同的向量的集合来表

示；

对于图像子集 $D_{voc} \subseteq D_{train}$ ，经特征检测和描述后得到一个描述符的集合，记作 S^m ；随后，基于欧氏距离，利用K-Means算法把集合 S^m 聚成一个 V^m 类，每个聚类中心相当于一个视觉单词，而所有的聚类中心构成一个视觉字典，记作 W^m ；然后计算训练图像的金字塔单词直方图；

$x_n^m = (x_{nlc}^m | l=1, \dots, D^m; c=1, \dots, 2^{2(l-1)})$ ，其中 x_n^m 的维数大小是 $L^m = V^m \times \sum_{l=1}^{D^m} 2^{2(l-1)}$ ； x_I^m 和 x_J^m 分别表示图像I和图像J第m通道的特征，则通过直方图交叉距离计算图像I和图像J的相似度 $d_{hi}(x_I^m, x_J^m)$ ；

$d_{hi}(x_I^m, x_J^m) = \sum_{i=1}^{L^m} \min[x_I^m(i), x_J^m(i)]$ ，其中， $L^m = V^m \times \sum_{l=1}^{D^m} 2^{2(l-1)}$ 是特征向量 x_I^m 或 x_J^m 的长度， V^m 和 D^m 分别表示第m通道字典的大小和金字塔的深度。

基于场景装饰的实时交互渲染方法

技术领域

[0001] 本发明涉及3D场景渲染技术领域,特别涉及一种基于场景装饰的实时交互渲染方法。

背景技术

[0002] 随着计算机技术和多媒体技术的迅猛发展,图像处理与展示已经从2D时代跨越到了3D时代。3D电影、虚拟现实、3D游戏等,都是3D技术在人们日常生活环境中的广泛应用。而在家装领域,人们也从传统的2D效果图迈向了3D效果图,越来越多的装修软件和装修企业,都开始应用3D渲染技术,来给用户展现一个更为真实的装修后体验。

[0003] 在这些3D应用中,技术实现方式主要还是通过渲染服务器,依托主流的渲染软件,实现对3D模型的渲染,然后再将渲染效果以视频、图片或者多张图片组合后通过类似three.js技术的方式予以Web展现。即使在这样的Web展现过程中,虽然实现了3D交互效果,但其交互主要是对于360度角度上的交互,而无法实现径深路径上的交互,也不能解决实时局部放大、缩小等高级功能。因此,这样的3D实时交互,并不是真正意义的3D实时渲染。随着HTML5技术的进一步推广,WebGL技术的提出,解决了3D实时渲染的Web实现技术展现问题。但是,按照WebGL技术的实现方案,3D交互依赖obj文件和材质渲染,如果obj文件过大,则在网络传输上要花费较多的预下载时间,基于Web的3D实时渲染等待时间过长。

[0004] 3D渲染大体上分为两个关键过程,即建模过程和渲染过程。建模负责创建描述渲染场景中特定对象,并且尽可能的符合部件对象的几何形状和外观。渲染过程负责通过输入的模型数据确定最终图像的像素颜色信息。最终形成外部可见可操作的3D场景以及场景中的部件。在建模方法中,比较常见的建模方式是通过Sketchup软件或3DMax软件完成3D模型的Obj文件制作,而这样的Obj文件往往很大(包括其配套的mtl文件),虽然当前国内的网络带宽已经很快(在10M以上),但对于一个部件模型就超过80M,整个场景模型量超过20个3D环境,10M+的网络环境仍然无法满足实时交互渲染的需要,如何减小模型数据量,满足模型渲染数据的实时传输,降低渲染等待时间,是当前3D实时渲染技术及应用中都面临的问题和研发的重点。

发明内容

[0005] 针对现有技术中所存在的不足,本发明的目的在于提供一种基于场景装饰实时交互渲染技术,通过步骤模型预处理、模型场景化、重复部件标记、模型数据流传输,解决了在家居场景中3D实时交互渲染时间长的技术问题,大大降低了渲染时延,提高了3D交互体验感,取得了较好的效果。

[0006] 为实现上述目的,本发明采用了如下的技术方案:

[0007] 一种基于场景装饰的实时交互渲染方法,包括以下步骤:

[0008] S1、模型导出,将空间环境及其内部各个部件建模、贴材质和渲染效果预览后的3D模型导出;首先导出整个3D模型所在的整个文件,然后分别导出空间环境中的各个部件模

型和材质图片,同时每个部件模型的坐标原点定位于其自身;

[0009] S2、模型预处理,将步骤S1导出的各个部件模型中的各类信息分别单独存储在数据库及配置文件中;

[0010] S3、模型场景化,在空间环境中心放置一个拍摄镜头,拍摄镜头位置固定,依次以30度旋转步进在空间环境内360度旋转连续拍摄出多个场景图片,拍摄出每个场景图片进行保存;

[0011] 对于多个场景图片采用上下文的多核SVM场景识别算法,该算法以0度视角保存的场景图片作为初始场景,使每个场景图片存在显式的上下文关系,通过马尔可夫模型建立起多个场景图片的上下文关系;再通过整个场景中上下文的贝叶斯分类算法提取不同角度拍摄的场景中的部件并进行识别和标记;

[0012] S4、模型标注,将步骤S3中识别的部件与步骤S1中导出的独立部件模型之间建立对应关系,再将提取的部件与独立部件模型比对,从而得出重复部件并在关键的独立部件模型上进行模型标注;

[0013] S5、模型数据流传输,采用标准的Web流传输方式;

[0014] 用户通过web发送对整个空间环境的3D渲染预览请求,服务端调出该空间环境的所有模型文件和该空间环境的配置文件集;然后通过网络传送到前端;

[0015] 先传输配置文件,然后传输整个空间环境材质信息和光源数据信息,再次依据部件关键性从大到小传输空间环境中各部件模型及次要模型;接下来再传送该初始场景相邻场景数据,最后传送剩余数据;

[0016] S6、WebGL流式渲染与本地缓存

[0017] Web前端接收到配置文件数据和材质数据后,首先通过3D渲染代码在前端重构空间环境;接下来渲染初始场景的关键模型和次要模型及初始模型邻接的其它场景角落;然后,WebGL流式渲染开始初始场景细节化和精细化;后台继续接受模型数据缓存。

[0018] 其中,所述的多个场景的上下文关系在建立时,先根据场景类别和图像特征建立一阶马尔可夫公式: $P(Q_t | Q_{1:t-1}) = P(Q_t | Q_{t-1})$;其中 Q_t 和 x_t 分别代表 t 角度时图像对应的场景类别和特征, Q_t 和 Q_{t-1} 之间是相差 30° 的镜头视角;

[0019] 根据以上关系建立了多个场景上下文关系公式:

$$[0020] \quad P(Q_t = q | x_{1:t}) \propto P(x_t | Q_t = q) P(Q_t = q | x_{1:t-1}) = P(x_t | Q_t = q) \sum_{q'} A(q', q) p(Q_{t-1} = q' | x_{1:t-1})$$

[0021] 其中, $\sum_{q'} A(q', q) P(Q_{t-1} = q' | x_{1:t-1})$ 是马尔可夫概率矩阵, $P(x_t | Q_t)$ 是观测率。

[0022] 具体地,所述场景中部件的识别采用基于图像分割的物体识别算法。

[0023] 进一步地,所述的基于图像分割的物体识别算法;其中,图像的像素集合为 x ,所有相邻像素的集合为 N ,设 $C = (C_1, \dots, C_t, \dots, C_x)$ 是该图像的一个分割, $C_i = x_B$ 表示第 i 个像素属于背景, $C_i = x_F$ 表示第 i 个像素属于前景;该图像分割的总势能 $E_{(x)}$ 、单元势能 $U_{(x)}$ 、点对势能 $V_{(x)}$ 分别表述如下:

$$[0024] \quad E_{(x)} = U_{(x)} + \lambda V_{(x)}$$

$$[0025] \quad U_{(x)} = \sum_{i \in x} \varphi_i(x_i)$$

$$[0026] \quad V_{(x)} = \sum_{(i,j) \in N} V_{(i,j)} \sigma(x_i, x_j)$$

$$[0027] \quad \sigma(x_i, x_j) = \begin{cases} 1 & x_i \neq x_j \\ 0 & x_i = x_j \end{cases}$$

[0028] 其中, $U_{(x)}$ 代表单元势能, 每个像素 i 分割为 x_i 所花费的代价; $V_{(x)}$ 代表点对势能, 反映像素集中任意两个相邻像素 i 和 j , 当分割不连续, 即 $x_i \neq x_j$ 时所花费的代价; i 和 j 越相似, 则 $V_{(i,j)}$ 越小; 反之, 则 $V_{(i,j)}$ 越大。

[0029] 再有, 所述的基于图像分割的物体识别算法在进行图像分割时, 分割步骤如下:

[0030] a. 初始化, 取整图区域作为初始区域, 生成第一个初始分割, 其前景部分为未知, 背景像素点为背景;

[0031] b. 前景和背景通过 K -均值的高斯混合建模, 构建一个图用以找到一致的前景和背景像素点:

[0032] c. 定义背景控制阈值和最大循环次数, 再逐步减小方框区域直至识别出部件。

[0033] 进一步地, 所述的重复部件标记, 采用以下方法:

[0034] a. 当两个部件存在上下关系时, 下部部件为重复部件;

[0035] b. 当一部件在两个场景甚至多个连续场景中出现时, 该部件是重复部件, 在不同场景中出现频率越高, 则关键值越大;

[0036] c. 部件的关键值具有场景标签, 即 K_i^m 代表部件 m 在第 i 个场景中的关键值。

[0037] 更进一步地, 将步骤 S3 识别的部件图片进一步处理使获得部件图片与空间环境内单独部件匹配, 以自动识别部件的具体品名, 首先对每个具体部件建立视觉直方图, 然后对步骤 S3 中获得的处理后的图像也建立视觉直方图; 最后再通过比对判断步骤 S3 中提取的部件图像确定具体部件。

[0038] 更进一步地, 所述特征采用以下方法表示整幅图像:

[0039] (1) 首先定义全局图像的 BoW 特征;

[0040] (2) 将图像按照空间金字塔形式划分为子图像并计算 BoW 特征;

[0041] (3) 最后拼接金字塔各层的所有 BoW 特征来表示整幅图像。

[0042] 设定图像数据集 $D = D_{\text{train}} \cup D_{\text{test}}$, 其中图像子集 D_{train} 用来训练分类集, 图像子集 D_{test} 用作测试, 通过规则网格法进行特征检测, 用均匀间隔水平和垂直的直线划分图像, 对形成的局部区域进行特征检测; 完成特征检测之后, 图像被抽象成为了若干个局部区域, 并对形成的局部区域加以特征描述, 将其转换为数值向量; 每幅图像可以用维数相同的向量的集合来表示;

[0043] 对于图像子集 $D_{\text{voc}} \subseteq D_{\text{train}}$, 经特征检测和描述后得到一个描述符的集合, 记作 S^m ; 随后, 基于欧氏距离, 利用 K -Means 算法把集合 S^m 聚成一个 V^m 类, 每个聚类中心相当于一个视觉单词, 而所有的聚类中心构成一个视觉字典, 记作 W^m ; 然后计算训练图像的金字塔单词直方图;

$$[0044] \quad x_n^m = (x_{nlc}^m | l=1, \dots, D^m; c=1, \dots, 2^{2(l-1)}), \text{ 其中 } x_n^m \text{ 的维数大小是 } L^m = V^m \times \sum_{l=1}^{D^m} 2^{2(l-1)};$$

x_I^m 和 x_J^m 分别表示图像I和图像J第m通道的特征,则可以通过直方图交叉距离计算图像I和图像J的相似度 $d_{hi}(x_I^m, x_J^m)$;

[0045] $d_{hi}(x_I^m, x_J^m) = \sum_{i=1}^{L^m} \min[x_I^m(i), x_J^m(i)]$, 其中, $L^m = V^m \times \sum_{i=1}^{D^m} 2^{2(i-1)}$ 是特征向量 x_I^m 或 x_J^m 的长度, V^m 和 D^m 分别表示第m通道字典的大小和金字塔的深度。

[0046] 相比于现有技术,本发明具有如下有益效果:

[0047] 1、在导出时,每个部件模型的原点定位于其自身,环境模型中保存每个部件模型在该环境中的坐标信息,环境模型单独定位原点。这一模型导出的方法确保了远端实时重建环境时的部件定位准确,同时,也实现了部件模型的独立,提高了同一部件在不同环境中的应用可能性;同时,多核SVM场景识别算法,实现了空间中场景的定义、上下文关系的建立及场景中部件的自动识别和提取;

[0048] 重复部件标记采用改进的图形切算法,将得到的抽象图形与单独部件建立对应关系,并判断是否为重复部件;根据流式数据传输优先顺序和WebGL实时渲染优先顺序,关键性越高的部件提前渲染,对于减小Web实时渲染的延迟。

[0049] 4、优先传送配置文件,然后是材质数据和光源数据信息,再次传送初始场景中的关键模型以及其他模型,再传送该初始场景相邻场景数据,最后传送剩余数据。采用这一方式,可以显著减少实时渲染的等待时间;用延迟着色技术,提高实时渲染的效率和效果。

附图说明

[0050] 图1为本发明的流程图;

[0051] 图2为本发明导出模型的流程图。

具体实施方式

[0052] 本发明的为实现上述目的,本发明采用了如下的技术方案:

[0053] 参见图1、图2;本发明提出一种基于场景装饰的实时交互渲染方法,包括以下步骤:

[0054] S1、将完成的空间环境及其内部各个部件建模、贴材质和渲染效果预览后的3D模型导出;首先导出整个3D模型所在的整个文件,然后分别导出空间环境中的各个部件模型和材质图片,在导出时每个部件的原点定位于自身。

[0055] 如图2所示,在客厅家居设计与3D实时展现中,使用三维软件,比如3DMax制作客厅环境以及客厅中的每个家具模型,在3DMax软件中查看整个客厅环境的家具实景效果。使用3DMax软件分别导出客厅中的每个家具模型以及整个客厅环境;当用3Dmax导出时,首先导出的是整个3Dmax的文件,然后分别导出该空间环境中的各个部件,包括(1)地面obj模型和材质图片(2)天蓬obj模型和材质图片(3)房间内各个墙体的obj模型和材质图片(4)房间内所有家具的obj模型和材质图片。本发明在3DMax模型导出时,每个部件模型的原点定位于其自身,空间环境模型中保存每个部件模型在该环境中的坐标信息,空间环境模型单独定位原点;将原点根据家具特点设置到该家具的特定位置,不是整个客厅环境中的统一原点位置。比如,沙发模型的原点坐标位于沙发模型的背侧中间;茶几模型的原点坐标位于茶几模型的左下角。该导出方法确保了远端实时重建环境时部件定位准确,同时,也实现了部件

模型的独立,提高了同一部件在不同环境中的应用可能性。

[0056] S2、模型预处理,将步骤S1导出的空间环境中的各个部件模型中的各类信息分别单独提取出来存储在数据库及配置文件中。

[0057] 3DMax对整套模型进行预处理,从客厅环境模型中读取每个家具模型在客厅环境中的坐标位置,从客厅环境模型中分析出每面墙模型信息和坐标信息,对于两面墙交界区域进行标记,从客厅环境模型中读取地板模型信息和天花板模型信息。上述信息不光存储在数据库中,同时也存储在配置文件中。该步骤本发明中的特点在于:(1)实现了从3DMax模型中读取信息,识别模型组件;(2)实现了对于模型组件在3DMax模型中的具体坐标的识别和读取;(3)识别了各个墙体交接处存在的家具组件(这些家具组件在实时交互渲染时会成为移动交互过程中的重要节点,因此,其也是下一步的重复部件之一)。

[0058] S3、模型场景化,模型场景化,在空间环境中心放置一个拍摄镜头,拍摄镜头位置固定,依次以30度旋转步进在空间环境内360度旋转连续拍摄出多个场景图片,拍摄出每个场景图片进行保存。

[0059] 对于多个场景图片采用上下文的多核SVM场景识别算法,该算法以0度视角保存的场景图片作为初始场景,使每个场景图片存在显式的上下文关系,通过马尔可夫模型建立起多个场景图片的上下文关系;再通过整个场景中上下文的贝叶斯提取进行场景中部件并进行识别和标记。

[0060] 在空间环境中心放置一镜头,镜头以30度偏移为间距旋转进行取景保存12张场景图片,基于这12张图片本发明采用基于上下文的多核SVM场景识别算法,该算法首先定义0度视角保存的房间图片为初始场景;由于本发明应用的特定性及镜头30度偏移使得12张图片存在显示上的上下文关系,即可通过马尔可夫模型建立图像的上下文关系并进行识别;然后再通过整个场景中上下文的贝叶斯分类算法提取进行场景中部件识别和标记;多核SVM场景识别算法,首先,上下文关系的建立,由于镜头以30度偏移为间距旋转,而整个房间的布局在镜头360度旋转过程中并没有任何的改变,因此,每30度的偏移获取的场景图像之间存在关联,可以视为隐马尔可夫模型中一个状态到另一个状态的转换,进而可以将图像所属场景类别视为潜在变量,将图像特征向量视为观测变量,进而建立一阶马尔可夫公式: $P(Q_t | Q_{1:t-1}) = P(Q_t | Q_{t-1})$;其中 Q_t 和 x_t 分别代表 t 角度时图像对应的场景类别和特征, Q_t 和 Q_{t-1} 之间是相差 30° 的镜头视角;

[0061] 根据以上关系建立了多个场景上下文关系公式:

$$[0062] \quad P(Q_t = q | x_{1:t}) \propto P(x_t | Q_t = q) P(Q_t = q | x_{1:t-1}) = P(x_t | Q_t = q) \sum_{q'} A(q', q) p(Q_{t-1} = q' | x_{1:t-1})$$

[0063] 其中, $\sum_{q'} A(q', q) P(Q_{t-1} = q' | x_{1:t-1})$ 是马尔可夫概率矩阵, $P(x_t | Q_t)$ 是观测率。通过求取 P 的最大值 q ,则可以获得该图像对应的场景。

[0064] 通过以上算法,实现房屋空间中的场景定义,以房屋中心的镜头旋转30度为一个渲染来获取场景;同时,建立场景中上下文关系;并且实现场景中部件的自动识别和提取。

[0065] 另外,所述的基于图像分割的物体识别算法;其中,图像的像素集合为 x ,所有相邻像素的集合为 N ,设 $C = (C_1, \dots, C_t, \dots, C_x)$ 是该图像的一个分割, $C_i = x_B$ 表示第 i 个像素属于背景, $C_i = x_F$ 表示第 i 个像素属于前景;该图像分割的总势能 $E_{(x)}$ 、单元势能 $U_{(x)}$ 、点对势能 $V_{(x)}$

分别表述如下：

$$[0066] \quad E_{(x)} = U_{(x)} + \lambda V_{(x)}$$

$$[0067] \quad U_{(x)} = \sum_{i \in x} \varphi_i(x_i)$$

$$[0068] \quad V_{(x)} = \sum_{(i,j) \in N} V_{(i,j)} \sigma(x_i, x_j)$$

$$[0069] \quad \sigma(x_i, x_j) = \begin{cases} 1 & x_i \neq x_j \\ 0 & x_i = x_j \end{cases}$$

[0070] 其中, $U_{(x)}$ 代表单元势能, 每个像素 i 分割为 x_i 所花费的代价; $V_{(x)}$ 代表点对势能, 反映像素集中任意两个相邻像素 i 和 j , 当分割不连续, 即 $x_i \neq x_j$ 时所花费的代价; i 和 j 越相似, 则 $V_{(i,j)}$ 越小; 反之, 则 $V_{(i,j)}$ 越大。

[0071] 所述的基于图像分割的物体识别算法在进行图像分割时, 分割步骤如下:

[0072] a. 初始化, 取整图区域作为初始区域, 生成第一个初始分割, 其前景部分为未知, 背景像素点为背景;

[0073] b. 前景和背景通过 K -均值的高斯混合建模, 构建一个图用以找到一致的前景和背景像素点;

[0074] c. 定义背景控制阈值和最大循环次数, 再逐步减小方框区域直至识别出部件。

[0075] S4、关键模型标注, 将步骤S3中识别的部件自动与步骤S1中导出的部件建立关联关系, 并在步骤S1中导出的部件基础上进行关键模型标注;

[0076] 这一算法解决的是从第2步提取的部件, 怎样自动识别为该场景中的某个具体部件。因为步骤2得到的是一个相对抽象的图像, 而其需要与某个单独部件建立对应关系后, 才能够根据规则判定该单独部件是否为重复部件。因此, 建立对应关系的算法 (即部件标注算法) 很重要。

[0077] 根据步骤S3的算法, 提取出的部件其实对应了各自独立的家具部件, 下面再根据提取的部件, 以及单独家具模型的部件比对, 从而比较出重复部件。经过反复实验, 本发明在标注重复部件时采用以下方法:

[0078] a. 当两个部件存在上下关系时, 下部部件为重复部件;

[0079] b. 当一部件在两个场景甚至多个连续场景中出现时, 该部件是重复部件, 在不同场景中出现频率越高, 则关键值越大;

[0080] c. 部件的关键值具有场景标签, 即 K_i^m 代表部件 m 在第 i 个场景中的关键值。

[0081] 所述特征采用以下方法表示整幅图像:

[0082] (1) 首先定义全局图像的BoW特征;

[0083] (2) 将图像按照空间金字塔形式划分为子图像并计算BoW特征;

[0084] (3) 最后拼接金字塔各层的所有BoW特征来表示整幅图像。

[0085] 设定图像数据集 $D = D_{\text{train}} \cup D_{\text{test}}$, 其中图像子集 D_{train} 用来训练分类集, 图像子集 D_{test} 用作测试, 通过规则网格法进行特征检测, 用均匀间隔水平和垂直的直线划分图像, 对形成的局部区域进行特征检测; 完成特征检测之后, 图像被抽象成为了若干个局部区域, 并对形成的局部区域加以特征描述, 将其转换为数值向量; 每幅图像可以用维数相同的向量的集合来表示;

[0086] 对于图像子集 $D_{voc} \subseteq D_{train}$, 经特征检测和描述后得到一个描述符的集合, 记作 S^m ; 随后, 基于欧氏距离, 利用K-Means算法把集合 S^m 聚成一个 V^m 类, 每个聚类中心相当于一个视觉单词, 而所有的聚类中心构成一个视觉字典, 记作 W^m ; 然后计算训练图像的金字塔单词直方图;

[0087] $x_n^m = (x_{nlc}^m | l=1, \dots, D^m; c=1, \dots, 2^{2(l-1)})$, 其中 x_n^m 的维数大小是 $L^m = V^m \times \sum_{l=1}^{D^m} 2^{2(l-1)}$; x_I^m 和 x_J^m 分别表示图像I和图像J第m通道的特征, 则可以通过直方图交叉距离计算图像I和图像J的相似度 $d_{hi}(x_I^m, x_J^m)$;

[0088] $d_{hi}(x_I^m, x_J^m) = \sum_{i=1}^{L^m} \min[x_I^m(i), x_J^m(i)]$, 其中, $L^m = V^m \times \sum_{l=1}^{D^m} 2^{2(l-1)}$ 是特征向量 x_I^m 或 x_J^m 的长度, V^m 和 D^m 分别表示第m通道字典的大小和金字塔的深度。

[0089] 在客厅环境中, 主要的模型是各个家具部件, 各面墙的材质以及地板和天花板的材质。对各个家具部件, 虽然都可能是客厅环境中不可或缺的, 但是其重要性还是存在差异。比如, 茶几上放了一套茶具。这是两个不同的家具部件, 但茶几是关键模型, 而茶具不是。因为茶几的位置、高度都会直接影响到茶具。而茶具的不同, 对茶几的环境因素没有影响; 所以, 在客厅环境中, 需要标注出关键模型, 这些关键模型在未来远端Web重建还原时, 需要优先考虑。

[0090] S5、模型数据流传输, 采用标准的Web流传输方式

[0091] 用户通过web发送对整个空间环境的3D渲染预览请求, 服务端调出该空间环境的所有模型文件和该空间环境的配置文件集; 然后通过网络传送到前端。

[0092] 先传输配置文件, 然后整个空间环境材质信息和光源数据信息, 再次依据部件关键性从大到小传输空间环境中各部件模型及次要模型; 接下来传送该初始场景相邻场景数据, 最后传送剩余数据。

[0093] 当用户通过Web发送对该房间的3D渲染预览请求后, 服务端调出该客厅的所有模型文件和该房间的配置数据集。然后通过网络传送到前端, 供前端WebGL渲染时调用。但是, 由于一个客厅房间的所有模型文件很大(通常在800M以上), 如果不加限制的任其自由传送, 则在带宽较小时Web渲染的时延较长。因此, 本发明制定了数据流传输的规则: 优先传送配置文件, 然后是四面墙的材质数据和两路光源数据信息, 再次传送初始场景中的关键模型以及其他模型, 接下来传送该初始场景相邻场景数据, 最后传送剩余数据。

[0094] 本发明制定了数据流传输的规则, 即优先传送配置文件, 然后是4面墙的材质数据和2路光源数据信息, 再次传送初始场景中的关键模型, 以及其他模型, 接下来传送该初始场景相邻场景数据, 最后传送剩余数据。采用这一方式, 可以显著减少实时渲染的等待时间。

[0095] S6、WebGL流式渲染与本地缓存,

[0096] Web前端接收到配置文件数据和材质数据后, 首先通过3D渲染代码在前端重构了空间环境; 接下来渲染初始场景的关键模型和次要模型及初始模型邻接的其他其它场景角落; 然后, WebGL流式渲染会开始初始场景细节化和精细化; 后续传输的模型数据在后台继续接受缓存。

[0097] Web前端接收到配置文件数据和4面墙的材质数据后, 首先通过3D渲染代码在前端

重构了客厅的4面墙和天花板以及地面(含材质);接下来渲染初始场景的关键模型和次要模型。以及初始模型邻接的场景角落。然后,WebGL流式渲染会开始初始场景细节化(后台继续接收缓存模型数据)和精细化。在此过程中,将用到延迟着色技术,来提高实时渲染的效率和效果。

[0098] 本发明结合家居设计的具体应用以分场景、分模型、分优先次序的方式,在客户端重构场景,通过优先级控制的网络数据传输,解决了实时交互渲染的等待时延问题。

[0099] 以上所述仅为本发明的较佳实施例而已,并不用以限制本发明,凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。



图1

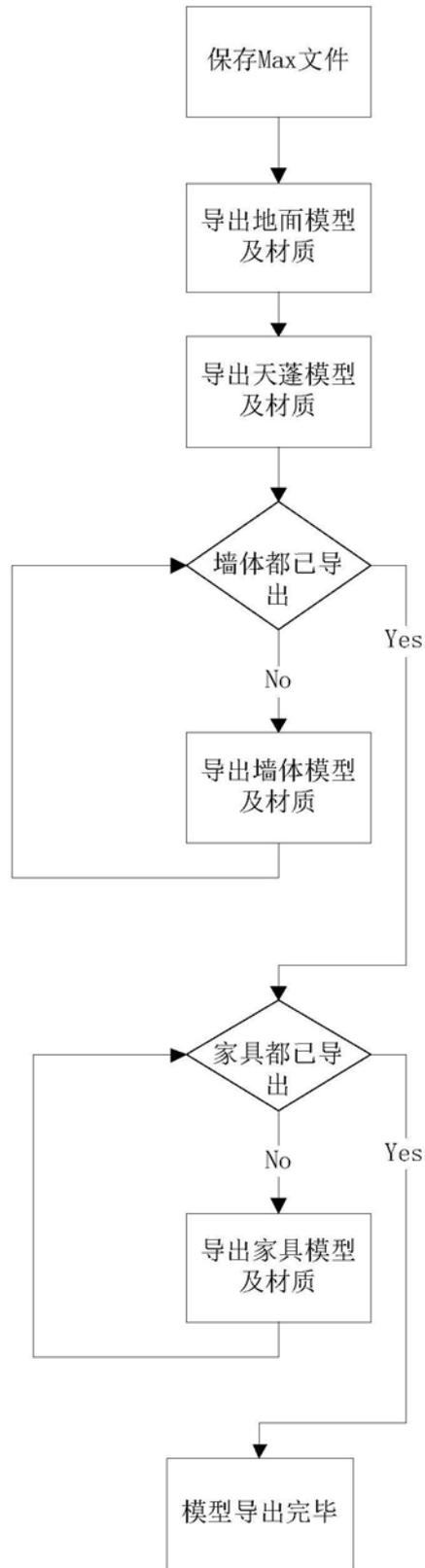


图2