

# 基于实例和约束的三维场景概念设计和系统实现<sup>\*</sup>)

江 澜<sup>1</sup> 岳小莉<sup>2</sup> 冯柏岚<sup>2,3</sup> 李良军<sup>1</sup> 曹存根<sup>2</sup>

(首都师范大学 & 中科院计算所联合实验室 北京 100037)<sup>1</sup>

(中国科学院计算技术研究所 中科院智能信息处理重点实验室 北京 100080)<sup>2</sup>

(中国科学院研究生院)<sup>3</sup>

**摘 要** 3D 场景设计是 VR 建模中最重要也最耗人力的工作之一,如何解决该问题是一项重要的课题。本文提出一种新的基于实例和约束的三维场景概念设计方法。首先,根据所获取的场景实例中所蕴涵的不同层次的语义信息,对其进行场景元素过滤和场景语义标注,并由此构建三维场景实例库。相应地,对场景中可能出现的元素类别进行分类和语义标注,并建立场景元素库。接着,应用基于实例的方法,并结合场景设计过程中所涉及到的知识(约束),给出三维场景概念设计的一般方法;最后以一个实例对该方法做进一步的说明。研究表明,该方法符合人们对场景设计的一般过程。此外,由于场景实例和场景元素具有丰富的语义信息,使得所生成的场景除了常规的可视效果,还可以直接应用于虚拟现实环境以及三维动画等数字内容制作中。

**关键词** 场景实例,语义约束,场景概念模型

## A System for Case-and Constraint-based Conceptual Design Method for 3D Scenes

JIANG Lan<sup>1</sup> YUE Xiao-Li<sup>2</sup> FENG Bai-Lan<sup>2,3</sup> LI Liang-Jun<sup>1</sup> CAO Cun-Gen<sup>2</sup>

(Unite Academe of ICT and CNU Beijing, Beijing 100037)<sup>1</sup>

(Institute of Computing Technology, Chinese Academy of Sciences, Key Laboratory of Intelligent Information Processing, Beijing 100080)<sup>2</sup>

(Graduate School of the Chinese Academy of Sciences)<sup>3</sup>

**Abstract** 3D scene design is one of the most important yet labor-intensive tasks in virtual world modeling. This paper proposes a new case- and constraint-based method of conceptual 3D scene design. First, we construct a 3D scene case-based by filtrating and marking the scene through several levels of semantic information, and construct a 3D scene element base by classifying and marking pre-modeled scene elements. And then, we present a conceptual design method for 3D scenes in which scene constraints are incorporated. Finally, we illustrate this method through an example. As experiments indicate, our method accords with the human scene design process. Furthermore, scenes which are generate could be used in the virtual reality environment and the figure content such as 3D animation because of the plenty of semantic information which is included in scene cases and scene element cases.

**Keywords** Scene case, Semantic constraint, Scene concept model

## 1 引言

在影视动画和游戏制作中,场景设计提供剧中人物的活动环境,其制作技术的好坏,直接关系到影片的造型质量。场景设计的基本内容包括基本场景的搭建以及场景中的道具选择和布局。传统场景设计的一般过程是:场景设计之初,在大脑中想象或是寻找一个抽象的、粗粒度的场景,即场景原型;然后将粗粒度的场景原型转化为细粒度的与剧情相匹配的真实场景;最后,在真实世界中对该场景进行搭建或图示化。近年来,随着计算机技术的飞速发展,人们可以更多地借助计算机来帮助完成大量繁重的手工绘制工作。与传统做法相比较,除了可以实现设计过程中的多次反复修改,设计结果可以被重用和共享,在大幅度提高设计效率的同时降低了制作成本。尽管现有不少系统提供了很多平台和工具来帮助专业人员完成上述工作,但本质上,它是将场景设计的载体从传统的工作平台转移到了计算机上,在制作流程上没有质的变化,即

在计算机上,大量的时间仍然要耗费在一些常规的、无需创意的工作上。自动生成一个符合故事内容基本要求(不包括创意要求)的场景已经成为一项迫切的需要。

场景自动生成从最早的基于几何约束的空间自动布局到基于语义的场景构建、从人工干涉到系统自动生成,关于场景自动生成的研究已走过二十多年的历程,取得了许多有益的成果。从技术和功能实现的角度来划分,目前已有的研究可以划分为两个主要的方向:基于约束的自动空间布局和基于知识的从文本到动画的自动转换。限于篇幅,以下仅列出与本文直接相关的几项有代表性的工作。

(1)基于几何约束和语义信息的场景自动布局。这类系统主要研究如何在一个有限空间域中实现内部物体的自动放置与合理布局。有文献可查的这类研究较多<sup>[1~7]</sup>,这里以更具代表性的系统 CAPS 为例加以说明。CAPS(Constraint-based Automatic Placement for Scene Composition)<sup>[8]</sup>由 Toronto 大学的 Xu Ken 等人研制。该系统提出一种基于约束

<sup>\*</sup>)本文工作得到自然科学基金的资助( # 60273019、60573064、60573063 和 60496326)和国家重点基础研究发展计划 2003CB317008 和 G1999023701 的资助。

的自动空间布局方法,即基于虚拟物理世界的物体空间几何约束以及基于真实世界中物体之间位置关系的语义约束,自动引导物体在空间中的放置,便于用户快速、方便并合理地复杂场景进行布局。实例表明,该系统可以帮助用户在不到10min的时间内布置一个内含300个物体的复杂空间。

(2)文本-场景的自动生成。这类系统<sup>[9~12,15]</sup>以自然语言的文本为输入,旨在通过对文本的语法分析和语义理解将文本的内容自动转化为二维图片或三维场景。具代表性的系统有WordsEye、CarSim和Swan等。

WordsEye<sup>[13]</sup>系统由AT&T实验室的Bob Coyne和Richard Sproat研究。以自然语言的文本为输入,首先对句子进行标识并分析,将句子转换为一些具有独立结构的语义单元,同时给出语义表达;其次,使用描述规则将相关的语义表达转换为的一组底层的描述子(可用来表达三维物体的姿态、空间关系、色彩等);接下来,应用传导规则消解场景元素之间可能存在的约束冲突并添加隐式约束。经过上述步骤,最后得到的描述子可用来构造与文本内容相对应的三维物体并完成最终的场景。

CarSim系统<sup>[14]</sup>由法国GREYC实验室的Sylvain Dupuy等人研制,借鉴了WordsEye的研究成果。该系统以汽车交通事故过程的可视化为研究对象,以自然语言(法语)描述的事件过程为输入,输出模拟该事件过程的动画。在信息的处理上,首先从文本中抽取出静态信息,即事件发生时所处环境的信息(本例中指道路以及路旁的树的信息);再进一步抽取事件中的动态信息(如汽车碰撞事件中的当事人、目击者、相撞的两辆汽车等)。该系统包含两个模块:语义模块和可视化模块。语义模块的功能是通过对自然语言的理解抽取文本中与时空相关的描述信息,将自然语言描述转化为一种形式化的中间语言;可视化模块则负责把形式化的语义转化为三维动画。

Swan<sup>[17]</sup>由中国科学院计算技术研究所的陆汝钤院士于20世纪80年代末开始的一个尝试,试图在计算机上模拟传统动画生成的全过程。该系统侧重于对文本的分析和理解,在场景动画的具体生成方面则没有具体说明。从实现的效果来看,场景部分是以类似于舞台剧的二维背景形式出现的,缺乏真实感,难以实现人物与场景元素的交互。

总结现有的工作,不难发现,目前的工作主要存在以下几点不足:(1)必须对场景中的元素进行完整的描述,即系统不能生成描述中未出现的场景元素;(2)在场景构建过程中采用的是“从无到有”的构建方法,在对有代表性的场景的重用和共享方面涉及不多;(3)在场景构建中,较少考虑其中可能出现的行为(动画创作中场景设计的一个显著需求),即融合行为的场景设计。

由此,我们提出一种基于实例和约束的场景概念设计方法。本文第2节介绍算法设计过程中所涉及到的基本概念及算法中用到的场景元素库、场景实例库和场景语义约束,第3节给出基于实例和约束的场景概念设计的算法流程,第4节通过一个实例对该方法的系统实现做了进一步说明,最后对我们的工作进行了总结和展望。

## 2 场景元素库、场景实例库和场景语义约束

### 2.1 基本概念

场景:本文研究中的场景是指一个封闭的空间,即内景,包括卧室、客厅、办公室等。一个场景由一个二元组来描述

$Scene = \langle SE, R_{SE} \rangle$ 。其中SE是场景元素的集合, $R_{SE}$ 是SE上的关系。场景元素是指构成场景的实体,例如墙壁、桌子、椅子、餐具等都是场景元素。场景元素的关系包括组成关系、支撑关系、伴随关系和其它的参照关系(如位置关系、朝向关系等空间关系)。

场景概念模型(Scene Concept Model):场景的结构可以借助场景概念模型来表示,由一棵树来描述。场景概念模型在一个高层次上表述了场景中场景元素的行为以及各场景元素间的关系<sup>[16]</sup>。模型中的每个节点代表了一个场景元素,根节点为场景,如图1所示。方便起见,在下文中,使用忽略了参照关系的树状结构图来表示场景概念模型,而参照关系将在布局中表示。

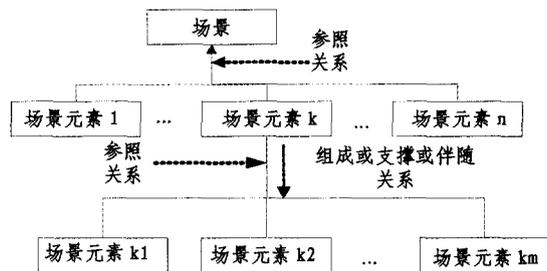


图1 场景概念模型

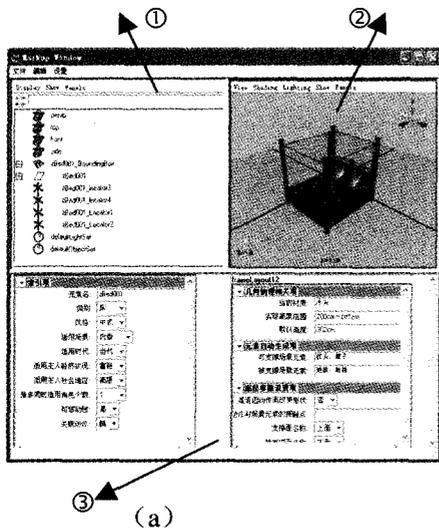
场景概念设计:就设计层次而言,场景设计可以分为功能设计和个性化要素设计。其中功能设计的内容包括场景布局、场景元素的类别选取以及恰当的色彩配置等;个性化设计的内容则是与场景主人信息相匹配的个性化场景元素的补充。从重用性和共享性来看,功能设计的内容通常可以从大量的设计实例中抽象出来并固化,作为新设计的参照。基于此,概念设计的主要内容即是基于已有实例并结合目标场景的相关信息完成满足主要功能并符合基本语义约束的新场景。

目前,我们获取实例的主要来源包括一些从事室内设计的公司以及影视、动画、游戏等特效制作公司。在长期的实际工作中,已经积累了大量形式多样的场景实例,包括二维和三维的。在我们的实例库中,主要选取的是有代表性的三维场景实例。

通常,初始的场景实例所包含的内容非常丰富,而从可重用的角度来看,需要对其内容进行过滤,仅保留最具共享性的内容。通过筛选,保留的内容应包括三类:场景属性、场景中与功能相吻合的主要的场景元素(包括场景元素的色彩、样式信息)以及由其组成的场景概念模型和对应的场景布局信息。并且,为了便于实例的后续应用,相关的场景元素的属性信息被保留,置为缺省值。基于此,我们对所获取的场景实例和场景元素进行信息过滤和加工,其结果放入资源库中,作为候选。

### 2.2 场景元素标注

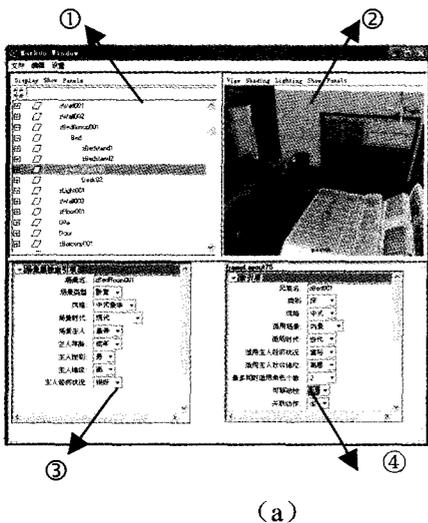
场景元素是场景的一部分,而对场景的所有操作最终都会转化到对场景元素的操作。因此,为了更加方便、有效地对场景元素进行操作,对场景元素进行标注是必不可少的。除了对场景元素的几何信息和外观信息进行标注外,还需要考虑场景元素本身所包含的语义信息(如类别、相关联的场景元素等)以及与角色之间的可能的交互信息。图2(a)所示是场景元素的标注界面,(b)为场景元素部分标注内容。



**场景元素标注内容 (部分):**  
 元素名: zBed001  
 类别: 双人床  
 适用场景: 卧室  
 适用时代: 现代  
 适用主人经济状况: 小康  
 适用主人社会地位: 社会中层  
 关联动作: 坐 (x,zBed001), 躺 (x,zBed001)  
 当前材质: 木头  
 实际高度范围: 200cm—205cm  
 可被支撑场景元素: 地板、地毯

①场景元素节点显示区,②场景元素实例显示区,③场景元素信息标注区。

图 2 场景元素实例的标注



**场景属性标注内容 (部分):**  
 场景名: zBedRoom001  
 场景类型: 卧室  
 风格: 中式豪华  
 场景时代: 现代  
 主人年龄: 成年  
 主人性别: 男性  
 主人经济状况: 好  
 主人社会地位: 高

说明: 场景的一些属性值是可以被场景元素继承的,例如场景主人的一些特性,具体可参看场景元素标注内容。

①场景元素节点显示区,②场景实例显示区,③场景信息标注区,④场景元素信息标。

图 3 场景实例的标注

### 2.3 场景实例标注

#### (1) 场景属性标注

初始场景实例往往包含许多无序的、非必要的信息,需要对其进行筛选和再组织,并且对场景属性进行语义描述,而场景里的重要场景元素也要按照场景元素的标注方法进行标注。图 3(a)为场景标注界面,(b)为场景属性的部分标注内容。

#### (2) 场景概念模型的二维布局获取

文[16]中介绍了一种图导向概念模型来描述场景中元素与元素的关系,此模型侧重于场景元素的显式的组成关系(如桌子是由桌面和桌腿构成),忽略了场景中另外两种普遍存在的关系:支撑关系(如桌子上支撑着一本书)和伴随关系(如桌子旁边都伴随着椅子)。本文将这两种关系补充进来,对该模型加以扩充。扩充后的模型具有以下几项优点:

- 可以进行一些场景推理。例如,当场景中出现一本书时,那么这本书必然有支撑它的场景元素存在,由此可以推出支撑它的元素:桌子,同时由这种支撑关系推出书与桌子存在

以下的几何约束:书的底面与桌子的顶面是平行的,并且两个面的距离为 0;

- 可以方便场景动态变化时的管理。例如,把书作为桌子的水平支撑元素,则当桌子运动时,书就会自动跟着桌子一起运动。

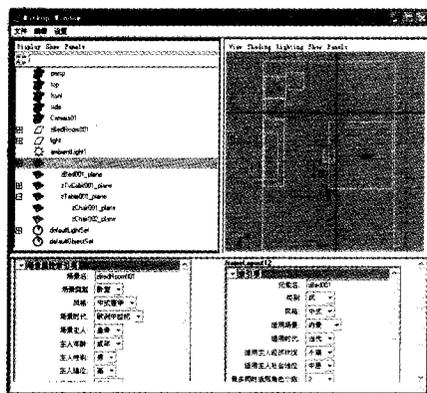
本文的研究中,场景概念模型所蕴涵的组成、支撑、伴随关系是通过语义约束来完成的,而参考关系则是通过几何约束来实现的,它们共同反映场景中的布局情况。在场景概念设计中,场景的二维布局是设计的重点。图 4(a)是对图 3 的场景的二维布局信息进行抽取的界面,(b)为场景概念模型示意图。

#### (3) 场景色彩配置

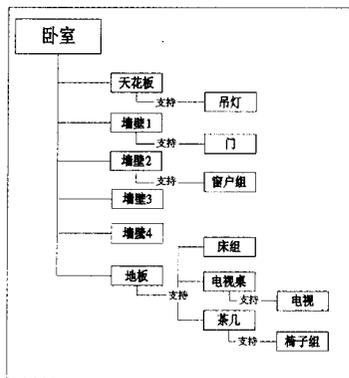
场景的色彩配置是指场景中主要元素或大面积元素的色彩信息,它是表现场景风格的一个重要因素,并能够从侧面反映场景主人的相关信息(如爱好、性格、经济状况、年龄和性别等)。如果用基于规则的方法很难对场景的色彩配置方法进行穷尽,因此我们在对场景实例进行获取和信息加工时,会有

意识地选取和保留有重用价值的场景色彩配置信息,以期为新场景的设计提供相关的借鉴和共享。表 1 给出了图 4 所示

场景的配色方案。



(a)



(b)

①场景概念模型树状结构显示区,②场景二维布局显示区,③场景信息标注区,④场景元素信息标注区。

图 4 场景二维布局界面

表 1 场景实例配色方案

场景风格	场景主人特性	场景元素类别	颜色	RGB 值
现代	家庭富裕,成年男性	墙壁	亮白	(0.851,0.839,0.812)
		地板	棕红	(0.369,0.224,0.153)
		床	白色	(0.773,0.761,0.729)

表 2 空间几何约束

约束类别	谓词举例	说明
单个物体的定位	位于坐标点((场景元素名),(<实数>,<实数>,<实数>))	指明一个场景元素的中心所处的位置
	朝向方向((场景元素名),<正面 背面 顶面 底面 左面 右面>,<方位词> <实数>,<实数>))	指明一个场景元素的绑定盒的某一面的朝向
两个物体的定位	位于((场景元素名),<场景元素名>,相邻)	第一个场景元素在第二个场景的旁边
三个及以上物体之间的位置关系	间距离((场景元素名),<距离>,<中间场景元素名>,<距离>,<场景元素名>))	一个场景元素的几何中心到中间场景元素的几何中心的距离以及另一个场景元素到中间场景元素的几何中心的距离,前提是这三个场景元素的几何中心在一条直线上。

表 3 语义约束

约束类别	谓词举例	说明
场景属性约束	场景属性((场景名),场景类型,<室内 室外 卧室 餐厅 ...>))	指明一个场景的类型是什么
	场景属性((场景名),场景风格,<中式豪华 西式豪华 中式普通 西式普通>))	指明一个场景的风格是什么
	场景属性((场景名),场景元素,<场景元素名>))	指明一个场景里具有的场景元素
场景元素属性约束	场景元素属性((场景元素名),类型,<家具 照明设备 装饰品 采暖设备 ...>))	指明一个场景元素的类型是什么
	场景元素属性((场景元素名),颜色·HSL,<色相值:0-255>,<饱和度:0-100>,<亮度值:0-100>))	指明一个场景元素的颜色 HSL 值分别是什么

## 2.4 约束

一个合理有效的场景设计必须满足相关的约束。本文将约束分为空间几何约束和语义约束。空间几何约束主要是用

于场景的布局,而语义约束则服务于场景属性、场景元素属性等内容的描述。本文采用谓词的形式来表示这些约束,具体内容参见表 2 和表 3。此外,在场景概念设计过程中,会用到

大量与设计相关的常识和经验类知识,这些内容我们通过规则加以表示,如表 4 所示。

### 3 场景概念设计方法

以上对场景实例和约束做了基本的介绍,图 5 给出了基于实例和约束的场景概念设计的基本方法以及输入输出的说明。

输入: Scene · Attribute() 为场景属性,包括场景类型和场景风格;  
 Scene · LWH() 为场景的实际长、宽、高;  
 Element\_Set() 为场景元素集合,包括场景元素的类别和场景元素的属性;  
 EleRel\_Set() 为场景元素关系集合;  
 Rule\_Set() 为布局过程中所涉及的约束和规则集合。  
 输出: Output(目标场景概念模型)  
 算法流程:  
 Step1: SC=Scene · C(场景类型), ST=Scene · T(场景风格);  
 Step2: 根据 Element\_Set() 和 EleRel\_Set() 构建场景的初始场景概念模型 OSCM;

Step3: 根据 SC 和 ST 在场景实例库 SCB(Scene Case Base)中搜索相似实例,如果有, cn=搜索到的相似实例个数,转 Step4; 否则,目标概念模型 GSCM=OSCM,转 Step11;  
 Step4: 如果 cn>1,转 Step5; 否则,转 Step6;  
 Step5: 按照场景属性的优先序,对候选实例进行相似性深度搜索,找到相似度最大的实例 Case<sub>i</sub>;  
 Step6: 获取场景实例 Case<sub>i</sub> 中的场景概念模型 Case<sub>i</sub> · SCM;  
 Step7: 将 OSCM 和 Case<sub>i</sub> · SCM 进行差异性比较,并根据比较的结果对实例中的元素类别和元素间的位置关系进行调整,得到新的场景概念模型 NSCM;  
 Step8: 根据 Rule\_Set() 中的规则对 NSCM 进行验证,如果满足规则,转 Step10; 否则,转 Step9;  
 Step9: 对存在于场景概念模型与规则之间的冲突进行消解;  
 Step10: 目标场景概念模型 GSCM=NSCM;  
 Step11: 根据场景概念模型中的场景元素类别在场景元素素材库中搜索出相关的场景元素(通常有多个符合条件的元素,进一步的选择需要综合考虑场景元素的材质和色彩);  
 Step12: 根据 GSCM 对场景元素进行放置;  
 Step13: 根据 Scene · LWH() 和 Rule\_Set() 的相关规则(包括与角色的交互信息)对场景元素的大小进行缩放;  
 Step14: 结束。

表 4 规则分类及举例

规则类型	规则举例
场景元素属性规则	场景属性 (s, 场景风格, 中式豪华) → 场景属性 (s, 场景元素, se) ~ 场景元素属性 (se, 场景元素风格, 中式豪华)
布局规则	场景属性 (s, 场景类型, 内景) ~ 场景属性 (s, 场景元素, se) ~ 场景元素属性 (se, 重量, 大) → 位于 (se, <墙>, 相邻)
色彩配置规则	场景属性 (s, 场景类型, 卧室) ~ 场景属性 (s, 风格, 中式豪华) → 场景属性 (s, 场景元素, se) ~ 场景元素属性 (se, 颜色 · HSL, <饱和度: 高, 亮度: 高>))

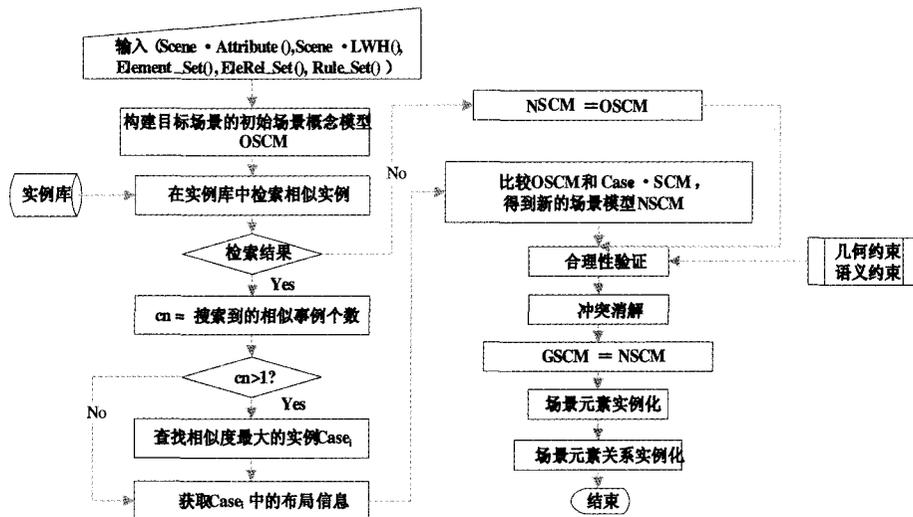


图 5 基于实例和约束的场景概念设计流程

### 4 系统实现和例子

借助上面提出的算法以及所构建的相关资源,我们设计了一个场景概念设计的实验系统,系统结构如图 6 所示。在获取用户对场景的设计需求后,基于实例和约束对场景进行

概念设计;进一步,对生成的场景模型进行评价,视评价结果对场景进行再设计或直接输出目标场景的概念模型;最后,利用三维建模工具输出三维目标场景。整个设计过程的实现在 Pentium 4 3000HZ, 2G Memory, windowsXP, Maya7.0 环境下进行。

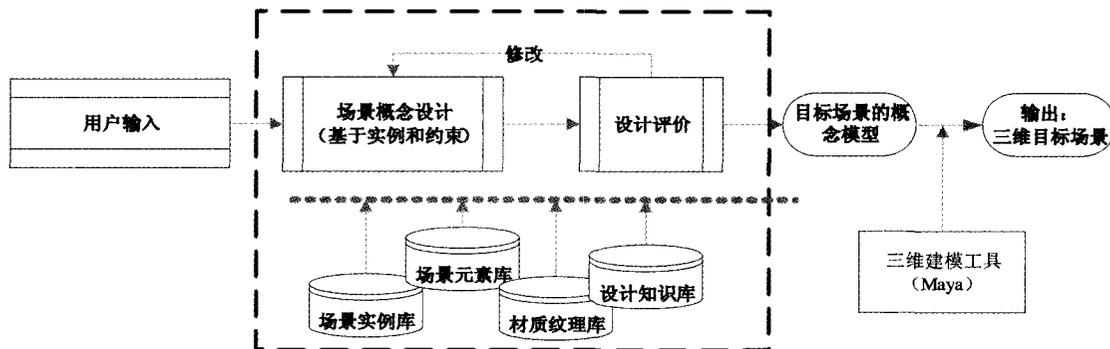


图 6 场景概念设计系统结构图

在实例的选取上,我们以《皇帝的新装》中皇帝的卧室为目标场景。参照图 5 的设计流程,以下对该场景的设计过程做进一步的说明。

(1)输入

• Scene • Attribute():〈场景类型:卧室〉,〈场景主人:皇帝 1〉,〈场景风格:西式豪华〉,〈场景时代:欧洲中世纪〉,〈主人年龄:成年〉,〈主人性别:男〉,〈主人地位:高〉,〈主人经济状况:很好〉

• Scene • LWH():(10,6,4)

• Rule\_Sets():大物靠墙放,同一地方不能同时出现两个物体,床的正面朝向墙壁 2。

• 场景元素集合和场景元素关系集合用图 7 所示场景概念模型来示。

(2)相似实例选取

通过对场景属性(这里选取场景类别、主人信息、风格等几个属性)进行检索和相似性比较,检索出图 3 中所示的场景为实例库中与目标场景最相似的场景。

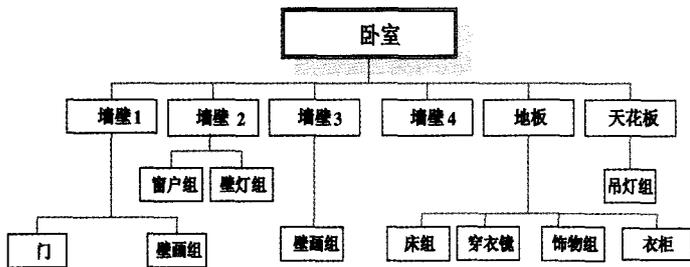


图 7 目标场景的初始场景概念模型示意图

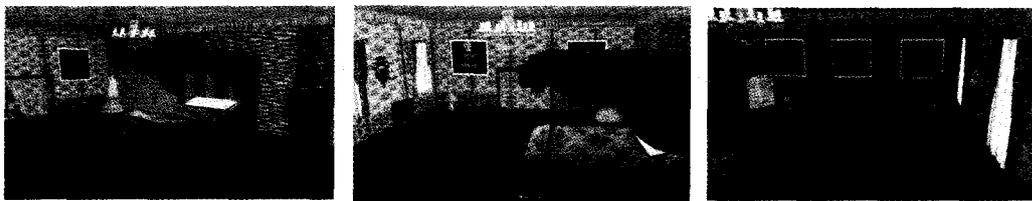
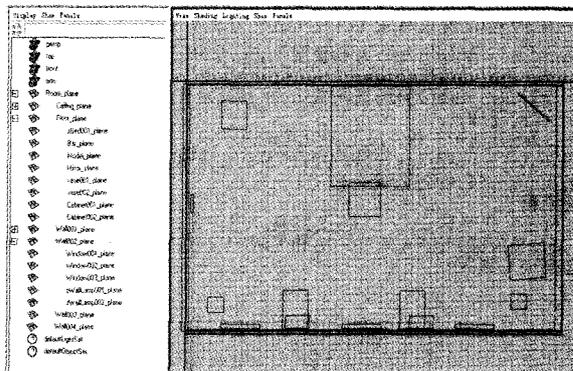


图 9 目标场景——皇帝卧室图例

(3)获取场景布局信息

通过对场景实例的概念模型(见图 4)和目标场景的概念模型比较,并根据比较结果对布局进行修改,再进行冲突消解,使得布局满足相关的规则,最后得到目标概念模型,如图 8 所示。



①场景树状结构显示区,②场景二维布局显示区。

图 8 目标场景的概念模型

(4)最终场景设计效果图

根据概念模型的层次关系,依次在二维布局图中对场景元素进行实例化,并根据实例的色彩配置方案进行色彩配置,最终得到的效果图如图 9。

**结论与展望** 近年来,基于实例的设计方法得到了较为广泛的应用。较之于已有方法,本文的创新点主要体现在以下几个方面:

• 针对场景实例中所蕴涵的多种信息,采用分层描述的方法获取和存储相关内容,以保证这些信息的后续重用和共享。

• 多种类型约束和相关知识的使用,除了保证场景几何空间布局上的正确性外,更可以满足场景设计结果合理性、有效性以及多样性。

• 与传统的场景设计方法相比较,场景概念设计将设计重点放在可以表征场景功能和风格的有限内容上,从而使得设计时间大大缩短,并且因其所具有的典型性,使得设计结果具有最大程度的可重用性。

初期实验表明,应用本文所述方法在场景生成的效率和效果上,均可以取得较为满意的结果。进一步的工作主要包括场景评价标准的量化和资源库的扩充。

场景评价标准量化:目前我们仅给出了场景评价的相关标准,还需要对其进行量化处理,以期实现场景的自动评价。

资源库扩充:较大规模的资源库的构建是场景设计结果多

样性的重要保证,在现有场景实例库、场景元素库、材质纹理库和设计知识库的基本构架的基础上,我们将进一步扩大资源的规模,并将基于本体的思想对其做进一步的改进和完善。

参考文献

- 1 Nielson G M, Olson D R. Direct Manipulation Techniques for 3D Objects Using 2D Location Devices. In: Workshop on Interactive 3D Graphics, 1986. 175~182
- 2 Houde S. Iterative Design of an Interface for Easy 3-D Direct Manipulation. In: ACM SIGCHI, 1992. 135~142
- 3 Venolia D. Facile 3D Direct Manipulation. In: ACM SIGCHI, 1993. 31~36
- 4 Borning A, Marriott K, Stuckey P, et al. Solving Linear Arithmetic Constraints for User Interface Applications. In: ACM Symposium on User Interface Software and Technology (UIST), 1997. 87~96
- 5 Smith G, Salzman T, Stuerzlinger W. 3D Scene Manipulation with 2D Devices and Constraints. In: Graphics Interface, 2000. 135~142

(下转第 249 页)

过 Zigzag 扫描, IQ-IDCT 处理, 最后得到一个宏块结果。对于帧内宏块存放的就是解码后的结果, 对于帧间存放的是纹理残差值。

同时将变长解码的码表, 逆 zigzag 扫描表存放到片内, 在片内还设置了全局变量, 用来保存 VOP、VOL 的头信息。表 2 给出了片内 RAM 主要分配。

表 2 片内 RAM 缓冲区主要分配

全局变量	所占空间/Byte
变长解码表(VLD)	4906
Zigzag 扫描表	192
VOL、VOP 头信息	108
解码输出缓存区(一个宏块行)	8448
重建宏块数据缓存区	384
运动补偿参考块缓存区	289
IDCT 反变换缓存区	896
宏块纹理解码缓存区	384
输入压缩码流缓存区	15K
合计	45.3K

#### 4 实验结果

我们的实验平台是使用 Freescale 公司 i. 250 芯片的 GSM 手机, 经过上述优化, 对 Sub-QCIF(128×96) 大小的 MPEG-4 视频文件进行了播放速度测试, 结果如表 3。

表 3 实验结果

视频序列	运动强度	播放帧数(f)	播放时间(ms)	解码速度(f/s)
序列 1	中等	974	63169	15.42
序列 2	低	1018	62970	16.17
序列 3	高	487	35222	13.83
序列 4	高	149	12922	11.45
序列 5	中等	1207	67767	17.81
序列 6	高	487	35309	13.79

实验结果表明, 经过优化的视频解码器在主频为 52MHz 的 ARM7 嵌入式处理器核上, 解码速率基本达到了实时效果。在基于低端嵌入式处理器(freescale 公司的 i. 250 芯片)平台下所设计的最终手机产品中, 实现了低分辨率的嵌入式 MPEG-4 视频实时解码。

通过与 2006 年 Philips 公司所公布的嵌入式播放器 Obigo Media Player 相比(注: Philips 公司所采用的硬件是 ARM7TDMI, 52 MHz, 256k RAM, Sub-QCIF MPEG-4 decoding in>10fps), 在处理能力基本相近的处理器平台上, 本文所设计的系统其解码速度已经达到甚至超过了 Obigo Media Player 的处理水平。

#### 参考文献

- ISO/IEC. MPEG-4 Video Verification Model version 18. 0. MPEG N3908, 2001
- 钟玉琢, 王琪, 贺玉文. 基于对象的多媒体数据压缩编码国际标准—MPEG-4 及其校验模型. 科学出版社, 2000
- Talluri R. Error Resilient Video Coding in the ISO MPEG-4 Standard. IEEE Communications Magazine, 1998, 36(6): 112~119
- He Yuwen. A Platform-based MPEG-4 Advanced Video Coding (AVC) Decoder with Block Level Pipelining. PCM2003; 15-18, Singapore. December 2003
- Sloss A N, Symes D, Wright C. ARM System Developer's Guide: Designing and Optimizing System Software. 北京航空航天大学出版社, 2002
- 田纲, 胡瑞敏, 王中元, 等. Trimedia 平台 MPEG4 编码器优化策略. 计算机工程与应用, 2006(36): 78~81
- 顾梅花, 张太镒. 基于 ARM 的 MPEG4 视频解码器. ARM 应用技术论文大奖赛
- Kingston. Interactive Manipulation of Articulated Objects with Geometry Awareness. In: Proceedings of the 1999 Conference on Graphics Interface, May 1999. 592~598
- Coyne B, Sproat R. Wordseye: An Automatic Text-to-Scene Conversion System. In: ACM SIGGRAPH, 2001. 487~496
- Dupuy S, Egges A, Legendre V, et al. Generating a 3D Simulation of a Car Accident from a Written Description in Natural Language: the CarSim System. In: Proc. ACL Workshop on Temporal and Spatial Information Processing, 2001. 1~8
- Tappan D. Knowledge-based Spatial Reasoning for Automated Scene Generation from Text Descriptions: [Ph D dissertation, New]. Mexico State University, 2004
- Bernier F, Boivin E, Laurendeau D, et al. Conceptual Models for Describing Virtual Worlds. In: WSCG (Posters), 2004. 25~28
- 陆汝钤, 张松懋. 从故事到动画片. 自动化学报, 2002, 28: 321~348
- Salzman T, Smith G, Stuerzlinger W. Constraint Based 3D Scene Construction; [Master dissertation, York University]. 2000
- Stuerzlinger W, Smith G. Efficient Manipulation of Object Groups in Virtual Environments. IEEE Virtual Reality, 2002
- Xu Ken, Stewart J, Fiume E. Constraint-based Automatic Placement for Scene Composition. In: Graphics Interface, 2002. 25~34
- Ehret B D, Microsystems S. Learning Where to Look: Location Learning in Graphical User Interface. In: Spatial Cognition Minneapolis, Minnesota, USA, 2002. 20~25
- Adorni G, Di Manzo M, Giunchiglia F. Natural Language Driven Image Generation. COLING 84, 1984. 495~500
- Clay S R, Wilhelms J. Put: Language-based Interactive Manipulation of Objects. In: IEEE Computer Graphics and Applications, March 1996. 31~39

(上接第 232 页)