

# 面向内容的三维模型数据库设计及其检索系统的实现<sup>\*</sup>

林金杰 韦 伟 杨育彬

(南京大学计算机软件新技术国家重点实验室 南京 210093)

**摘 要** 目前,对基于内容的三维模型检索的研究主要集中在检索过程所涉及到的算法方面,而对三维模型库的设计研究较少。为此,详细分析了面向内容的三维模型库的建立过程,重点论述三维模型库的组织结构和分类表的设计,提出一种三维模型分类的手工构造方法,开发了三维模型库的管理软件,并研制了三维模型检索的原型系统。其意义在于为三维模型检索研究提供一个结构合理、数据充分的三维模型库,从而为检索算法的研究奠定坚实的基础。

**关键词** 三维模型库,分类表,三维模型检索,面向内容

## Content-based Design and Implementation of 3D Model Database and Retrieval System

LIN Jin-jie WEI Wei YANG Yu-bin

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

**Abstract** Most of the current available content-based 3D model retrieval techniques mainly focus on feature extraction and similarity measurement, and seldom involve the proper design of 3D model databases. This paper presents a content-based design and creation of 3D model databases. Firstly, the paper proposes a novel 3D model database structure by introducing a classification hierarchy in the design of 3D model databases. On the basis of the proposed database design, a 3D model database management module, and a prototype system of content-based 3D model retrieval are implemented. The proposed database structure is capable of providing well-structured and rich information in 3D model databases, and facilitates the implementation of content-based 3D model retrieval system.

**Keywords** 3D model database, Classification hierarchy, 3D model retrieval, Content-based

## 1 引言

三维模型是一种比二维图像内容更为丰富真实、更加符合人类视觉特性的多媒体数据类型,随着计算机图形学的发展以及三维模型获取技术、图形硬件技术的不断进步,其应用越来越广泛。实现快速高效的基于内容的三维模型检索技术已成为国内外多媒体信息检索领域的一个研究热点。

国外在三维模型检索方面的研究起步较早,取得了较好的成果。最典型的是普林斯顿大学形状检索与分析实验室<sup>①</sup>提供的一套三维模型搜索引擎研究方案<sup>[1,2]</sup>,该检索系统基于文本、二维和三维形状及其组合进行查询。普林斯顿大学的基准三维模型库已成为目前研究的一种标准,被很多机构参考。近年来,我国的三维检索研究也发展迅速并取得很好的成绩。尤其,台湾大学通讯与多媒体实验室<sup>②</sup>研制的三维模型检索系统<sup>[3]</sup>,提供了一种基于文本关键字、可包含内部细节的二维草图或选择现有模型进行查询的研究方案。

套用 N. Wirth 关于程序的定义公式:算法+数据结构=程序,本文给出了三维模型检索系统的公式:检索算法+三维模型库+人机交互界面=三维模型检索系统。如果说人机界面是人的衣裳,那么检索算法就是人的灵魂,而三维模型库则

是人的躯体。三维模型库是三维模型检索系统不可或缺的基本的物质基础。此外,跟丰富的语料库在自然语言处理研究的作用一样,具有相当规模且包含类别较为全面的三维模型库是进行三维模型检索研究的基础。目前国内大部分的研究都是直接采用普林斯顿大学及台湾大学提供的三维模型库<sup>[4]</sup>。基于上述原因,又考虑到国内关于三维模型库的设计方面的研究内容较少,本文将探讨三维模型库的建立过程中非常重要的模型组织与分类问题,提出一种三维模型检索数据库的设计方法。在此基础上,本文还将探讨一个简单的三维模型检索原型系统的实现。

## 2 三维模型库的设计

### 2.1 三维模型库的设计内容

建立三维模型库不是简单地将三维模型收集起来存放在某一固定的存储位置,这个过程还涉及很多方面。

首先,需要考虑的是三维模型库的组织结构。由于三维模型文件具有众多的格式,那么,三维模型库是采用统一的文件格式还是兼容各种文件格式。对于前者,需要考虑文件格式的转换;对于后者,则需考虑如何整合不同的文件格式并为外部提供统一的访问接口。此外,模型文件在库中的存放方

<sup>\*</sup>国家自然科学基金专项基金资助项目(编号 60723003),国家自然科学基金资助项目(编号 60505008),江苏省自然科学基金创新人才(学术带头人)基金资助项目(编号 BK2007520)。林金杰 硕士研究生,主要研究方向为基于内容的视觉信息处理、三维模型检索、人工智能;韦 伟 硕士研究生,主要研究方向为三维模型检索、图像处理、计算机图形学;杨育彬 博士,主要研究方向为基于内容的视觉信息处理、人工智能、数据挖掘。

①<http://shape.cs.princeton.edu/search.html>

②<http://3d.csie.ntu.edu.tw/~dynamic/>

法也是一个需要细致考虑的问题。简单地将大量的模型文件堆放在一个文件目录下并非一个好的解决办法,因为模型文件可能会发生重名。总的来说,考虑模型库的组织问题,就是要研究模型文件如何存储在模型库中,从而使得外部应用程序可以便捷地访问。

其次,需要研究的是分类表的设计。三维模型库通常需要一个分类表来指明库中模型文件的分类情况。尽管目前该分类只能通过手工完成,但这项工作是有必要的。比如,PSB模型库是按照模型的自然属性进行人工分类的,以往的实验证明不同的分类方法会对检索效果的评价造成影响。在三维模型检索的研究中,召回率和准确率是两个对模型的检索结果进行评价的主要参数。计算这些参数都需要事先知道模型库中模型的分布情况,而这些信息可以使用分类表来描述。此外,三维模型检索中的某些算法,可能需要通过对已有的某类模型进行分类学习,从而得到该类模型的特征表示,这也需要事先知道模型文件的分类情况。分类表的设计有两个含义:一是分类文件格式的设计,二是按照语义概念设计类别的层次结构。

最后是关于模型库管理软件。模型库中包含大量的三维模型,在日常维护中需要检索和管理模型,通过人工在模型库中操作比较麻烦,因此需要提供一个模型库的管理软件。例如,由于三维模型通常都不是简单的模型文件的集合,因而新加入一个模型往往需要添加一系列的相关附加信息,这些操作如果通过人工操作将相当繁琐,最好是由模型库的管理软件来实现。

## 2.2 三维模型库的组织结构

### (1) 三维模型的组成

三维模型库中的基本构成单位是三维模型。在三维模型库中,一个三维模型通常由几何实体、缩略图和附加信息描述三方面的数据组成。存储三维模型几何实体数据的文件简称为三维模型文件。三维模型文件的格式众多,常见的有OBJ, WRL, 3DS和OFF等。缩略图通常是三维模型在某一平面的投影。用户检索后返回的结果集常常是相应三维模型缩略图的列表。用户由缩略图可以很直观地、迅速地辨别该三维模型所表现的事物。而附加信息描述通常指明了三维模型文件的一些附加信息,如源路径、文件格式等,或者是与模型建模相关的一些数据,如模型的坐标轴等信息。

考虑三维模型的组成,即是具体分析下面两个问题:

1) 三维模型文件的格式是否统一。由于三维模型文件的格式众多,为了便于日后的处理,我们常常希望模型具有统一的文件格式。那么这就需要考虑采用哪种文件格式作为标准的文件格式,是采用已有的某种文件格式还是自定义一种符合日后研究要求的文件格式。还需要考虑文件格式的转换问题。当然,模型库也可以不统一文件格式,那么如何将不同的文件格式整合起来,从而使外部程序可以方便地读取,便是一个需要精心考虑的问题。

2) 附加信息如何描述、如何存储。每个模型总会有一些附加的信息必须记录。这些附加信息可以分为两类:一类是指明模型文件的属性,如作者、源路径、文件格式等等,这些信息尽管对三维模型检索研究没有直接的意义,但是考虑到知识产权的缘故,应该如实记录下来,尤其是如果模型库采用了统一的文件格式,那么就更有必要在这些附加信息中说明原始模型文件的一些情况;另一类信息则是模型的统计信息,这些统计信息有可能是来自作者开发该模型时设定的一些参数

(如模型的坐标系等信息),也可能是为了降低运行时计算的复杂度预先计算出来的。在设计三维模型的组成时,我们就必须考虑这些附加信息具体包括哪些内容,一一列举出来。此后需要考虑的问题是如何存储这些信息。可以专门为一个模型设计一个附加信息的说明文件,也可以建立一个专门的关系数据库来记录这些信息。

### (2) 三维模型的命名规则与存储方法

前面已经指出,一个三维模型通常包含三个方面的数据,而这三方面的数据通常存储在不同的文件之中,即一个完整的三维模型是由多个文件来描述的。那么这些文件该如何存储?此外,这些文件有的是从外部收集得到,有的是模型库生成,那么这些文件如何命名?还有,当模型库具有相当的规模时,文件之间很可能发生重名现象,这个问题如何解决?

设想这样的一种解决办法:让同一模型对应的全部文件(三维模型文件、缩略图、附加信息说明文件)都具有相同的文件名,然后将这些文件全部存放在一个文件目录下。当模型发生重名现象时,在原模型名后加数字编号。为了使外部程序可以访问该目录下的模型,需要为模型库建立一张索引表,表明库中含有哪些模型以及它们的存储位置。

这并不是一个好的解决办法,缺点很明显。首先,当模型数量非常大时,模型重名现象很可能发生。尽管可以通过给模型名后增加数字编号的办法来解决,可是这样的过程相当麻烦。当给模型库增加新模型时,首先得检查有没有重名。若有的话,得先计算出当前已经编号到几,然后加1,作为模型文件名。其次,对于人而言,即使已经知道要检索的模型文件名,在这样含有成千上万个文件的目录中查找也是很麻烦的。此外,对计算机而言,这样的存储办法会使得文件查找速度慢。由于该文件目录中含有大量的目录项,导致系统查找一个指定的目录项可能会花费不少的时间。

因此,本文认为一个较为理想的解决办法是给每一个模型分配一个唯一的编号并且采用多级目录。具体而言,可以叙述为以下两点:

- 每一个模型对应一个目录,并且该目录以模型的编号为目录名。一个模型可能对应着若干个文件,这些文件都存放在该模型对应的目录中,其名称可以修改为模型编号,也可以保持原有的命名。由于三维模型文件原有的名称常常指明该模型所表现的事物,因此,如果入库时修改该文件名的话,则原文件名应该存储在附加信息的描述中。如果以文件的原始名称来存储的话,则需要建立一个索引,使得程序可以通过模型编号得到这些文件名。本文建议的解决办法是:在模型的目录中另外建立一个以模型编号为文件名的索引文件,该文件中对应有若干索引项,如三维模型文件名、缩略图文件名、附加信息说明文件名等;程序需要读取具体的模型数据时,首先访问该索引文件,再由该索引文件得到其它文件的名称。

- 若干个模型目录合并为一个目录。例如每100目录合并为一个目录,这样可以使得每一个目录下目录项都不至于太多,既方便了人的查找,也提高了系统的查找速度。

## 2.3 分类表的设计

模型库的分类表的主要作用是给出模型库中模型在概念上的组织关系,即指明哪些模型是一类的,模型库中都含有哪些类别的模型。在含有分类表的模型库中进行三维模型检索时,若能够计算出库中某个模型与查询模型精确匹配,则可直接将该模型所对应的类作为结果集返回,这样能够大大提高

系统检索的速度。

### (1) 分类文件格式的设计

分类表的主要内容有两个：一个列举类别，即指明库中有哪些类别的模型；二是给库中模型分类，使得库中的模型都有一个分类。基于这样的认识，本文设计了如下的分类文件格式（称为 NCF, NJU Classification file Format）：

```
NCF 版本号
# header classNums
  classID className otherName ...
...
# endheader
classID parentClassID modelNums
```

属于本类的模型编号（每个模型编号占用一行）

第一行定义本分类格式的版本号。header 部分是类别名称及其编号的映射表，所有的类别都必须在此处先声明。每一个类别编号可以对应多个名称，每个名称之间用空格隔开（这也隐含了对类别名称的要求：类别名称不能出现空格，对英语中常出现的含有空格的短语，本文建议的解决办法是将空格替换为下划线）。header 之后则是具体的模型分类。每一个类别的分类信息都是由这样的结构构成：首先是类别信息说明，紧接着就是属于该类的模型编号列表。类别信息说明定义为“classID parentClassID modelNums”，即“类别编号 父类别编号 类别包含的模型数”。这里定义，parentClassID = -1 时表示该类别位于最顶级别。

下面是一个分类文件的例子：

```
NCF 1
# header 3
  0 human man people 人类 人
  1 fruit 水果
  2 apple
# endheader
0 -1 1
  人模型 1 的编号
1 -1 0
2 1 1
  苹果模型 1 的编号
```

### (2) 分类的设计

所谓分类的设计是指建立一棵分类关系树，树中每一个结点代表一种类别，结点之间的关系是一种概念之间的层次关系。因为概念之间层次关系有些时候是模糊的，导致建立这样一棵完备的分类树成为一件相当困难的工作。从我们研究的实际出发，三维模型库中建立分类表的主要目的在于指定库中模型的分类情况，实际上对分类的精确度不必有太高的要求。因此本文提出一种基于“模型实例驱动”的分类表构造方法。“模型实例驱动”的含义是，分类的构造是在为模型指定分类时引发的。具体而言，初始时分类表可能是只给出了有限的分类，当为模型制定分类时，考虑当前的分类表是否已经满足了分类粒度的要求。不符合，则驱动分类的构造过程修改分类表。这样，在进行了若干次的操作后，就可以得到一张足以描述整个模型库模型分类的分类表。

具体的构造过程如下：

首先将分类树初始化。初始状态的分类树只包含有限的结点，这些结点的关系是我们比较容易确定的。如我们可以比较容易地把世界上的所有事物划分为生命体和非生命体两

类，生命体又可以分成动物、植物、人和微生物，而人又可以确切地分为男人和女人，那么我们可以将初始分类树建为：

```
根目录
|——生命体
  |——人类
    |——男人
    |——女人
  |——动物
  |——植物
  |——微生物
|——非生命体
```

当为模型指定分类时，根据当前分类树的状态可以有两种操作：当前分类树已经包含了较好描述模型分类情况的结点时，直接将该模型归入该结点代表的类别；当前分类树中结点的分类概念对模型而言粒度过粗，此时需要将该分类做进一步的划分。具体划分方法是：在该结点下插入若干个新的结点以及一个称为“未明分类”的结点，这些新结点组成了对父结点类别上的划分，将模型归入恰当的子结点中。新加入的子结点可能对模型而言还不够精确，可以重复这个过程。要注意的是，当一个模型如果被归入“未明分类”时，此时就把“未明分类”的名称修改为结点对应的类别名。

例如，在上面的初始树中加入一个自行车模型后，分类树中“非生命体”结点变为：

```
非生命体
|——交通工具
  |——非机动车
    |——自行车
    |——未明分类
  |——未明分类
|——未明分类
```

如现在需要在前面的基础上再加入一个小轿车，则为：

```
非生命体
|——交通工具
  |——非机动车
    |——自行车
    |——未明分类
  |——机动车
    |——小轿车
    |——未明分类
|——未明分类
```

显然上述的算法中，两个概念的包含关系有时候并不是很清楚，因此上面的构造过程有很多不精确的地方。但该构造方法的优点在于构造过程简单、由模型驱动分类。

### 2.4 三维模型库管理软件

三维模型库中通常包含大量的三维模型，在日常维护中需要检索和管理模型。通过手工在模型库中操作比较麻烦，设计一个模型库管理软件有助于简化化管理，提高效率。

#### 2.4.1 功能概述

该管理软件的功能具体可以分为下面 3 个方面。

- 模型导入功能：既可以导入单个模型，也可以导入一批模型。用户只需要简单地选择需要进行导入的目标模型，系统便可以自动为其分配编号，生成缩略图以及附加信息说明文件（或将这些说明信息写入一个特定的文件或数据库中）。

- 分类表编辑功能：由于目前对模型的自动分类技术还

没有出现,因而从实际出发,还是采用手工分类的办法,所以该软件应该允许用户便捷地创建、编辑分类表。

• 提供三维模型库的编程访问接口。由于模型库采用了较为复杂的组织结构,因此当程序需要对模型库进行访问时,本软件应该提供编程接口,让程序去访问模型库中的任意模型。

#### 2.4.2 系统框架

三维模型库与管理软件的关系如图 1 所示。

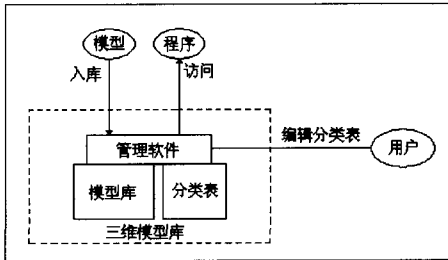
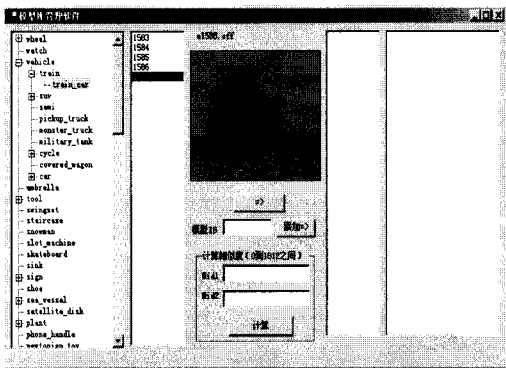


图 1 三维模型库结构图

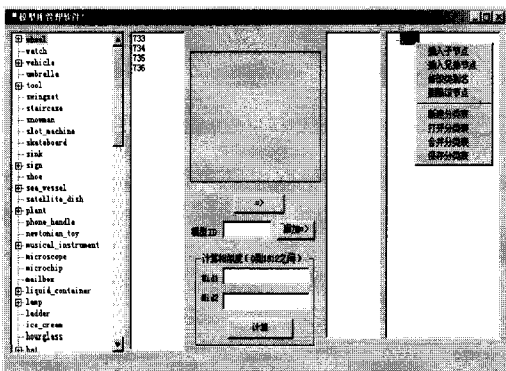
在上图中,三维模型库由模型库、分类表以及管理软件三大部分构成。模型库存储了三维模型数据,分类表给出了库中模型的逻辑组织关系,管理软件对模型库和分类表进行综合管理,为外部用户和应用程序提供访问接口。

#### 2.4.3 系统界面

本文基于微软.net 平台开发了一个模型库管理软件。系统主界面如图 2 所示。



(a) 模型库管理界面



(b) 分类表编辑界面

图 2 模型库管理软件主界面

整个界面可以分为左中右三大部分。

左边的部分是模型库的管理,由一个树形列表和一个普通列表构成。对模型库的操作通过右键的快捷菜单来实现,主要是导入新的模型入库和删除某个模型文件两种操作。导

入模型有两种方式:一是导入单个模型文件,二是导入整个文件夹中的模型文件。树形列表列出了整个库按照某种分类方法得到的树形分类结构。在树形列表中选择某一个类别后,旁边的列表便会显示该类别包含的模型的编号。选中模型后,便可以由中部看到模型的缩略图。双击列表中模型编号,便可以浏览相应的三维模型,参见图 2(a)。

软件的右半部分是分类表的编辑。在右边的树形列表中单击右键会出现操作菜单,如图 2(b)所示。

使用操作菜单可以建立分类表,然后用户可以从左边的模型列表中选择适当的模型(可以多选),点击中间的“=>”按钮,所选择的模型编号便会添加到右边的模型列表中。用户也可以在中部的“模型 ID”文本框中直接输入某一已知编号的模型(输入完成按回车可以浏览该三维模型),然后点击“添加=>”,从而将所填写的模型加入右边所选择的类别中。在操作列表中“合并分类表”功能是将现有的分类表跟另外的一个分类表进行合并。如果类别名称相同,则会自动归为同一类别。

中部提供了一个计算两个模型近似度的界面。用户输入两个模型编号后,点击“计算”,软件便会计算出两个模型的近似度并显示。

### 3 基于内容的三维模型检索原型系统实现

在三维模型库的基础上,本文设计并实现了一个简单的基于 B/S 结构的三维模型检索原型系统,作为前面工作成果的展示平台。

#### 3.1 系统结构及检索过程

原型系统的主要结构如图 3 所示。

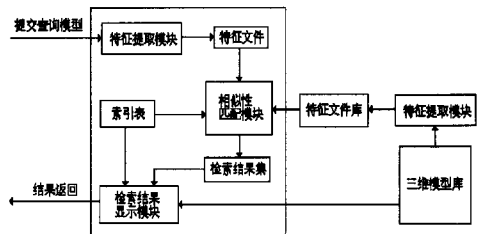


图 3 系统结构和检索过程示意图

在图 3 中,用户提交的查询模型上传至服务器后,首先经过特征提取模块提取三维模型的形状特征,存储为特征文件。然后,相似性匹配模块将该特征文件与服务器中的特征文件库(该特征文件库是系统脱机状态下通过事先调用特征提取模块对三维模型库中的三维模型进行特征提取后得到的)里的特征文件进行相似性计算,得到检索的结果集。这个检索结果集实际上只是三维模型的索引集,因此在检索结果显示模块,需要访问索引表和三维模型库,形成可供用户浏览的显示页面,最后将该结果页面返回给用户。

#### 3.2 核心模块概述

图 3 中,大方框内是原型系统的核心模块。核心模型主要包括:

##### (1) 特征提取模块

该模块的主要功能是提取三维模型的特征向量,生成特征描述文件。本文处理的三维模型实体均是采用三角形、多边形网格的网格表示法。网格表示法<sup>[5]</sup>能表示自然界各种各样复杂形体的表面,但难以描述模型内部信息,忽略了三维模型表面纹理等信息,因此适合于进行形状特征的分析与研究。

本文的检索系统采用基于三维模型形状轮廓的特征提取技术,利用在形状特征方面的球形谐函数(Spherical Harmonic Functions,又称球面调和函数)特征提取算法生成三维模型特征描述符。球面调和函数是一种球面坐标系的函数分析方法,又称单位球面上的二维傅立叶分析。基于函数分析的形状特征提取工作由来已久,Funkhouser 等人<sup>[6,7]</sup>就利用球形谐函数的不变特性,将三维空间分解为一系列具有不同半径的同心圆,然后分别为这些同心圆空间定义具有旋转不变性的频率函数,其集合构成三维模型的特征向量。

### (2)相似性匹配模块

通过特征提取模块,三维模型库的每一个三维模型都可以计算出一个特征文件。这些特征文件预先计算出来,存储在特定的位置(在本文中,这些特征文件存储在三维模型库中相应的模型目录下),构成一个概念上的特征文件库。相似性匹配模块将用户提交的三维模型经过特征提取算法计算得到的特征文件,与特征文件库中的每一个特征文件进行相似性计算,将相似程度在阈值范围内的模型编号作为结果集。本文使用简单的 Euclidean(欧几里德)<sup>[8]</sup>距离来计算两个三维模型的相似度,计算公式为:

$$D(X,Y)=\sqrt{\sum_{i=1}^n(x_i-y_i)^2}$$

### (3)检索结果显示模块

用户通过网页浏览器访问 Web 服务器,服务器通过检索系统搜索特征数据库,依据相似度生成结果页面返回给用户。由于检索结果集得到的是模型的编号集,因此需要本模块通过查找索引表和三维模型库,取得模型的缩略图。三维模型、缩略图和模型信息采用文件方式存放,依靠操作系统的文件管理功能管理。

## 3.3 原型系统界面

系统的界面主要包括用户的检索页面以及结果的显示页面,介绍分别如下:

### (1)主页面

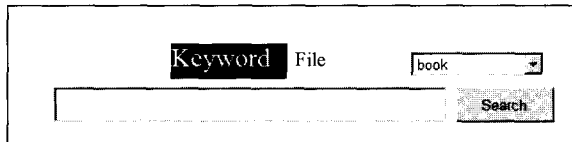


图 4 检索平台的主页面

主页面提供关键字和文件两种检索方式。

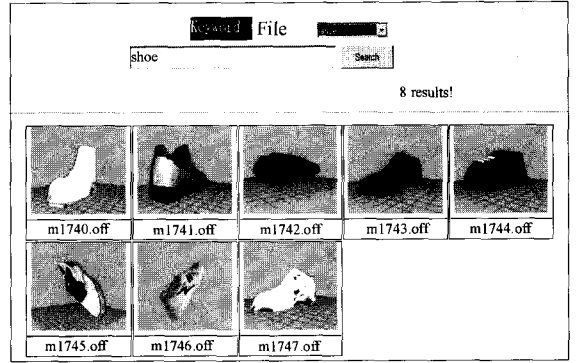
对于关键字查询方式,用户可以手工输入关键字,也可以从列表中选择关键字进行检索。文件检索方式则要求用户本地机上存有三维模型文件作为检索条件。

### (2)检索结果页面

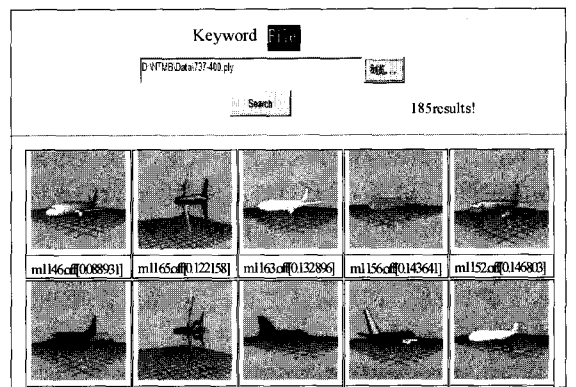
在图 5(a)中,用户输入了关键字“shoe”,系统通过将该关键字与关键字索引表匹配后得出用户检索的是鞋类模型。于是访问分类表,将 shoe 类的模型编号作为结果集传给结果显示模块。结果显示模块通过访问索引表和模型库,形成图 5(a)所示的检索结果页面。

图 5(b)中,用户提交了一个波音 737 的三维模型文件,要求检索与该模型形状相似的模型。系统将该文件首先上传至服务器,经过特征提取模块生成特征文件,然后与服务器中的三维模型的特征文件库进行相似性匹配,然后将近似程度在阈值以下的模型作为结果集返回。在本文中,相似度越高,计算出来的数值越小。如图 5(b)所示,本文三维模型库中

m1146 这个三维模型与用户提交的 737 模型近似度最高为 0.088931,故排在最前端。其它模型按照近似度依次排列。



(a)检索结果一



(b)检索结果二

图 5 检索结果

### (3)网页中的三维模型浏览

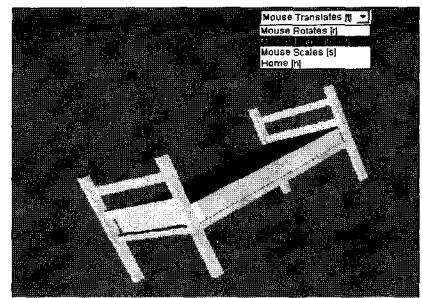


图 6 可浏览三维模型的 Java Applet 程序

该 Web 三维模型浏览程序提供了对三维模型的旋转、放大缩小和回复原始状态四种操作方式。操作方式的选择由菜单选项提供,请参见图 6。如用户选择“Mouse Rotates”方式后,当其在屏幕中移动鼠标时,三维模型就会发生旋转。

**结束语** 本文探讨了三维模型库建立过程中的一些关键细节,并设计和实现了一个三维模型库的管理软件,在此基础上研制了一个简单的基于内容的三维模型检索原型系统。

关于模型分类结构的设计,本文给出了一种手工的构造方法,但是通过手工对一个包含成千上万模型的模型库分类并不是一种高效的解决办法。更好的办法是能够实现模型的自动分类。而要实现模型的自动分类又跟三维模型的特征提

(下转第 291 页)

QTextEngine 类完成的。首先通过 QTextEngine 类静态成员函数 `static void bidiItemize (QTextEngine * engine, bool rightToLeft, int mode)` 按文本方向将混合文本分成 item, 并计算每个 item 的级数, 每个 item 内的文本的方向都是一致的。然后通过 QTextEngine 类的成员函数 `void QTextEngine::bidiReorder(int numItems, const Q_UINT8 * levels, int * visualOrder)` 进行 item 倒序操作, 这样就可以得到文本的显示顺序。然后再对明文进行自动选型, 计算每个 item 内字形大小和相对位置。最后计算每个 item 的绝对屏幕坐标位置, 按 item 的显示顺序, 将每个 item 依次显示在屏幕上。这样不论是对于段落向右对齐、从右到左书写的文本段, 还是段落向左对齐、从左向右书写的文本段, 处理的程序是一致的。不同的是每个 item 在屏幕上显示的坐标位置不同, 对于段落向右对齐的文本, 先按照向左对齐的模式计算它显示时的 X, Y 值, 再对 X 值进行一个固定的坐标变换, 使它右边界 (item 的轮廓是一个矩形) 靠文本书写区最右侧边界的距离与以前的 X 值一致, 这样显示出的效果就是靠右对齐。在 Qt 中 QTextEngine 类成员函数 `endLine` 进行每个 item 的坐标全定位。

**结束语** 我们在 Redhat Linux 9.0 的基础上, 按照上述方式对 Qt 库进行修改, 加入明文自动选形模块和左向文本书写模块, 重新编译 Qt 库, 同时在对对应目录下加入明文本地化数据库、输入法、字体以及界面翻译文件, 修改系统默认的本地化 Locale 变量, 就完成了维、哈、柯文本版 Linux 的开发, 它在保证了系统原有功能的基础上, 全面支持了明文的使用习惯。在使用 Qt 和 KDE 开发的图形程序中实现了明文自动选形、从右向左书写和民、汉文字混合编辑排版显示等功能。在图形方式下, 可以按照明文的使用习惯实现明文文件名、文件夹、搜索维文文件等操作系统的文本处理功能, 从而完成既定的设计目标。

文章论述的这些关键技术, 不仅很好地实现了 Linux 下维、哈、柯文的信息化处理, 而且对于与维、哈、柯文类似的阿拉伯语系文种的处理也具有普遍的指导意义。

(上接第 242 页)

取算法、相似性匹配算法甚至智能语义等都有密切的关系。我们下一步研究的一个主要目标就是实现模型的自动分类。

原型系统的检索方式中支持的关键字的个数有限, 而文件检索方式则要求用户本地机存在查询模型才可以进行模型检索。这些检索方式在有些情况下并不能很好地满足用户的检索要求。因此, 更加适宜用户使用的三维检索界面技术也是我们未来研究的内容之一。

## 参 考 文 献

[1] 杨育彬, 林琿. 基于内容的三维模型检索综述. 计算机学报, 2004, 27(10): 1297-1310

[2] Funkhouser T, Min P, Kazhdan M, et al. A search engine for 3D models. ACM Transactions on Graphics, 2003(22): 83-105

[3] Chen D, Tian X, Shen Y, et al. On Visual Similarity Based 3D Model Retrieval. Computer Graphics Forum (EUROGRAPH-

## 参 考 文 献

[1] Arabic Code Chart. <http://www.unicode.org/charts/PDF/U0600.pdf>, 2005, 1

[2] Arabic Presentation Forms - A. <http://www.unicode.org/charts/PDF/UFB50.pdf>, 2005, 1

[3] Arabic Presentation Forms - B. <http://www.unicode.org/charts/PDF/UFE70.pdf>, 2005, 1

[4] Arabic Shaping. <http://www.unicode.org//versions/Unicode4.0.0/ch08.pdf>, 2005, 1

[5] LineBreakingProperties. <http://www.unicode.org/reports/tr14.pdf>, 2005, 1

[6] Bishop A, Brown D, Meltzer D. Supporting multilanguage text layout and complex scripts with Windows 2000. <http://www.microsoft.com/typography/developers/uniscribe/intro.htm>, 2003, 12, 12

[7] OpenType Specification Version 1.4. <http://www.microsoft.com/typography/otspec/default.htm>, 2004

[8] 古丽拉·阿东别克, 米吉提·阿布力米提. 维吾尔语词切分方法初探. 中文信息学报, 2004, 18(6): 61-65

[9] 苏国平, 缪成, 夏国平. Linux 下维、哈、柯文多语种图形化处理平台的设计与实现. 中文信息学报, 2004, 28(4): 88-93

[10] 芮建武, 吴健, 孙玉芳. 国际化文字处理综述. 中文信息学报, 2006, 20(2): 87-93

[11] 靳箭明, 王华, 丁晓青. 维汉英混排文档识别. 电子与信息学报, 2006, 28(7): 1188-1191

[12] 缪成, 袁保社, 李莉. Linux 系统下开放式维、哈、柯、汉、英多语种混合输入法系统. 计算机应用, 2003, 23(11): 36-38

[13] 马宁, 于洪志. Linux 民文化技术. 西北民族大学学报: 自然科学版, 2005, 26(1): 58-63

[14] 卢有飞, 张伟, 等. 维文版 Office 设计中关键技术的研究与实现. 中文信息学报, 2007, 21(2): 112-116

[15] International Dr. 国际化软件开发. 申凤, 等译. 北京: 机械工业出版社, 2003

[16] 郑燕飞, 刘岩, 陈克非. Linux 文平台的实现关键技术及其发展方向研究[J]. 计算机工程, 2002, 28(1): 14-15

[17] 董军. 基于 Java 平台的维文版永中集成 Office 的设计[D]. 北京: 中国科学院研究生院, 2005

ICS'03), 2003, 22(3): 223-232

[4] 郑伯川, 彭维. 3D 模型检索技术综述. 计算机辅助设计与图形学学报, 2004, 16(7): 873-881

[5] Zaharia T, Preteux F. Shape-based retrieval of 3D mesh models. IEEE International Conference on Multimedia and Expo (ICME 2002), 2002(8)

[6] Kazhdan M, Funkhouser T, Rusinkiewicz S. Rotation invariant spherical harmonic representation of 3D shape descriptors // ACM SIGGRAPH'2003. 2003: 156-164

[7] Vranic D. An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres // IEEE International Conference on Image Processing (ICIP'2003). 2003(3): 757-760

[8] Veltkam P R. Shape matching: Similarity measures and algorithms. Shape Modeling International, 2001(5): 188-197