showMeYourHand详细设计说明书

基于OpenGL与Leap Motion 手势识别的三维手重建

2013

贾唯秦

北京理工大学

2013/5/2

目录

[1 引言 2](#_Toc355355077)

[1.1 编写目的 2](#_Toc355355078)

[1.2 项目背景 2](#_Toc355355079)

[1.3 术语说明 2](#_Toc355355080)

[1.4 参考文献 2](#_Toc355355081)

[2 系统结构 3](#_Toc355355082)

[2.1 需求规定 3](#_Toc355355083)

[3 设计说明 3](#_Toc355355084)

[3.1 模型动画模块 3](#_Toc355355085)

[3.1.1 描述 3](#_Toc355355086)

[3.1.2 功能 3](#_Toc355355087)

[3.1.3 性能 3](#_Toc355355088)

[3.1.4 输入项 3](#_Toc355355089)

[3.1.5 输出项 3](#_Toc355355090)

[3.1.6 设计方法（算法） 3](#_Toc355355091)

[3.1.7 流程逻辑 3](#_Toc355355092)

[3.1.8 接口 3](#_Toc355355093)

[3.1.9 存储分配 3](#_Toc355355094)

[3.1.10 注释设计 3](#_Toc355355095)

[3.1.11 限制条件 4](#_Toc355355096)

[3.1.12 测试计划 4](#_Toc355355097)

[3.1.13 尚未解决的问题 4](#_Toc355355098)

[3.2 LeapMotion手势识别模块 4](#_Toc355355099)

# 引言

## 编写目的

由概要设计阶段得出的系统的物理配置，处理流程和系统的数据结构、接口设计，本文档主要对项目编码以及算法进行设计。

本文档的预期读者是编码及测试人员。

## 项目背景

Leap Motion提供了一种全新的交互方式和交互形式，但是官方提供的SDK中只提供了手指尖端位置、手掌中心位置、手掌曲率球半径等信息，没有完全实现手的模型重建；本项目基于OpenGL图形引擎，根据Leap Motion提供的跟踪数据帧，实现完整的人手形态实时重建与同步，并使虚拟三维手可与场景中的物体进行交互。

本项目是北京理工大学本科生贾唯秦的毕业设计作品，在中国科学院软件研究所研究院张凤军老师、北京理工大学软件学院刘来旸老师指导下完成。

## 术语说明

**表格 1 术语说明**

|  |  |
| --- | --- |
| 名词 | 解释 |
| Leap Motion | 一款新型人机交互设备，由硬件及其配套的软件组成 |
| OpenGL | 三维图形图像库 |
|  |  |

## 参考文献

# 系统结构

　　[给出系统的结构框图，包括软件结构、硬件结构框图。用一系列图表列出系统内的每个模块的名称、标识符和它们之间的层次结构关系。]

## 需求规定

# 设计说明

## 模型动画模块

* + 1. 描述

载入ms3d格式的模型并实现动画控制

* + 1. 功能
    2. 性能
    3. 输入项

模型数据文件

矩阵

Leap数据帧

* + 1. 输出项

模型变换矩阵

* + 1. 设计方法（算法描述清楚即可）

1. 有关动画的数据结构定义

关节：初始位移，旋转变换

四个矩阵：初始局部matStaticLocal，初始全局matStaticGlobal，当前局部matCurrentLocal，当前全局matCurrentGlobal。

初始局部和初始全局始终不变，当前局部和全局都是根据Leap的刷新帧变化的。

顶点：初始全局坐标系坐标，一个骨骼索引

1. 读取ms3d模型文件

以NeHe Lesson 31读取顶点、材质等静态数据为基础，在研究.ms3d文件格式后读取文件中后续的动态信息，包括KeyFrame和Joint，本程序中只有一帧

以二进制流方式读取文件，设置一个指针Pptr指向要读取的数据块的头部，根据数据块的大小移动指针并将数据读取到对应大小的数据结构中。

1. 骨骼的初始化及动画原理
2. 构造关节的初始局部矩阵。根据模型文件中关节的初始位置(m\_translation[3] = {α，β，γ})和初始旋转(m\_rotation[3])信息可以求出关节的初始局部矩阵matStaticLocal，这里实际上是要构造一个仿射变换矩阵，根据计算机图形学的知识，设定刚体取向的旋转矩阵 [\mathbf{R}] 是由下面三个基本旋转矩阵合成的：

[\mathbf{R}] = \begin{bmatrix}
\cos \gamma & \sin \gamma & 0 \\
-\sin \gamma & \cos \gamma & 0 \\
0 & 0 & 1 \end{bmatrix} \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \beta & \sin \beta \\
0 & -\sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix}
\cos \alpha & \sin \alpha & 0 \\
-\sin \alpha & \cos \alpha & 0 \\
0 & 0 & 1 \end{bmatrix}

单独分开作用，每个矩阵各自代表绕着其转动轴的旋转；但是，当它们照次序相乘，最里面的(最右的) 矩阵代表绕着 z 轴的旋转。最外面的(最左的) 矩阵代表绕着 Z 轴的旋转。在中间的矩阵代表绕着交点线的旋转。

[\mathbf{R}] = \begin{bmatrix}
\cos\alpha\cos\gamma-\cos\beta\sin\alpha\sin\gamma & \sin\alpha\cos\gamma+\cos\beta\cos\alpha\sin\gamma   & \sin\beta\sin\gamma
\\-\cos\alpha\sin\gamma-\cos\beta\sin\alpha\cos\gamma & -\sin\alpha\sin\gamma+\cos\beta\cos\alpha\cos\gamma & \sin\beta\cos\gamma 
\\ \sin\beta\sin\alpha & -\sin\beta\cos\alpha & \cos\beta 
\end{bmatrix}

[\mathbf{R}] 的逆矩阵是：

[\mathbf{R}]^{-1}= \begin{bmatrix}
\cos\alpha\cos\gamma-\cos\beta\sin\alpha\sin\gamma & -\cos\alpha\sin\gamma-\cos\beta\sin\alpha\cos\gamma   &   \sin\beta\sin\alpha
\\ \sin\alpha\cos\gamma+\cos\beta\cos\alpha\sin\gamma & -\sin\alpha\sin\gamma+\cos\beta\cos\alpha\cos\gamma & -\sin\beta\cos\alpha
\\ \sin\beta\sin\gamma & \sin\beta\cos\gamma  & \cos\beta 
\end{bmatrix}

1. 求关节的初始全局矩阵。根关节的初始全局就是初始局部矩阵；子关节的初始局部矩阵就是其相对于父节点的变换矩阵。求子关节初始全局矩阵matStaticLocal的变换公式如下：

Joint->matStaticGlobal =

ParentJoint->matStaticGlobal \* Joint->matStaticLocal;

（求全部关节的初始全局矩阵。根据上面的变换公式，遍历一遍关节数组，就能求得关节结构中的两个初始矩阵，也就可以将关节摆放到准确的初始位置。）

1. 求当前局部矩阵。Leap动作帧提供了每根手指的DIP关节的位移和旋转信息，根据人手生理学屈伸约束以及逆向运动解析学，可逐级逆向推导出每根手指的PIP关节、MP关节的位移和旋转信息，进而可重构出每个关节的当前帧变换矩阵matAnimate，与初始局部矩阵相乘后即为当前局部矩阵。求当前局部矩阵的变换公式如下：

Joint->matCurrentLocal = Joint->matStaticLocal \* matAnimate;

1. 求当前的全局矩阵。父节点的当前全局与该节点的当前局部矩阵相乘即为该节点当前的全局矩阵：

Joint->matCurrentGlobal =

ParentJoint->matCurrentGlobal \* Joint->matCurrentLocal;

1. 顶点绑定
2. 顶点的信息包含初始的全局坐标和骨骼索引。（存疑点1：为何模型文件的顶点信息中只含有一个骨骼索引？因为我绑定的时候没有添加权重。程序中先只考虑单个骨骼对顶点的影响，模拟效果不好的话，再重新学习绑定技术）
3. 求顶点相对于关节的初始相对坐标。对于顶点对应的关节，先对该关节的初始全局矩阵求逆，再应用Leap ::Matrix类提供的向量变换公式，即得到顶点相对于该关节的初始相对坐标。在后续的更新帧中可以用到，求顶点相对坐标的公式如下：（存疑点2：推导需要验证）

Joint-> matStaticGlobal-1 = Joint-> matStaticGlobal. rigidInverse(); (Leap API提供的函数)

Vertex->vec4Relative = vec4StaticGlobal \* Joint->matStaticGlobal-1（初始全局矩阵的逆）

分为两步：

旋转: matStaticGlobal-1 .transformDirection(vec4StaticGlobal);

平移: matStaticGlobal-1 .transformPoint(vec4StaticGlobal);

存疑点3：每隔一个顶点，就有一个joint1的顶点，其boneID是0，由于其初始全局矩阵是单位矩阵，因此

1. 求动作帧中的顶点全局坐标。当程序获得Leap的更新帧数据后，将初始相对位置坐标向量与节点的当前全局矩阵相乘即得到顶点的当前全局坐标。（这是一个相对位置的概念，就是说初始时，模型文件提供了一个顶点相对于节点的相对位置坐标，这个相对坐标可以通过逆运算得到）求动作帧中顶点的全局坐标的公式如下：

Vertex->vec4CurrentGlobal = Vertex->vec4Relative \* Joint->matCurrentGlobal

（对于每个顶点的关节，需要进行两次矩阵运算，所得到的矩阵是关节的初始全局矩阵和当前全局矩阵，步骤三和步骤四在OpenGL中对应于模型变换的概念）

1. 利用Leap获取DIP关节的旋转和平移信息
2. 求PIP，MP关节的旋转和平移信息
3. 更新节点的绝对变换
   * 1. 流程逻辑
     2. 接口
     3. 存储分配
     4. 注释设计
     5. 限制条件
     6. 测试计划
     7. 尚未解决的问题

## LeapMotion手势识别模块