图片包含 图表

描述已自动生成

卡通人物

低可信度描述已自动生成

**数字化设计与制造大作业**

|  |  |
| --- | --- |
| **学 院** | 精工书院 |
| **班 级** | 03321901 |
| **姓 名** | 冯恒阳、朱晗琪、方俊现 |

日 期： 2022 年 5 月 7 日

目录

[第一章 项目介绍 3](#_Toc103028642)

[第二章 原理简介 3](#_Toc103028643)

[2.1 直纹面 3](#_Toc103028644)

[2.2 贝塞尔曲面 4](#_Toc103028645)

[2.3 曲面拼接 4](#_Toc103028646)

[2.4 几何变换原理 5](#_Toc103028647)

[第三章 代码实现 6](#_Toc103028648)

[3.1 *control\_point*类 6](#_Toc103028649)

[3.2 *control\_mesh*类 6](#_Toc103028650)

[3.3 *control\_joint*类 7](#_Toc103028651)

[3.4 曲面生成、动画制作 8](#_Toc103028652)

[第四章 心得体会 8](#_Toc103028653)

[第五章 代码 8](#_Toc103028654)

[5.1 main.m 8](#_Toc103028655)

[5.2 *control\_point.m* 9](#_Toc103028656)

[5.3 *control\_mesh.m* 9](#_Toc103028657)

[5.4 *control\_joint.m* 15](#_Toc103028658)

# 第一章 项目介绍

项目的灵感来源于一位动漫角色——初音未来。在互联网上出现过很多有关于此动漫角色的动作的视频。因此，我们希望在Matlab中实现类似的动作。



在项目中，我们使用了200多个贝塞尔曲面，另外还有一部分直纹面、旋转曲面、扫掠曲面等多个曲面。由于涉及的控制点的数量高达5000。我们学习了人体骨骼的思想，将每一个控制点归属在每一个骨骼上，通过骨骼的变换带动控制点的变换，从而带动人体的动作。通过Matlab面向对象的编程，形成了十分具有逻辑的结构层次。由于模型十分的复杂，我们通过犀牛软件将模型进行了面片的划分。合理的使用了贝塞尔曲面等一系列曲面表达了所需要的面片。

我们三个人中，朱晗琪、方俊现同学对于整个模型的划分进行了整理，按照一定的格式获取了所有曲面的控制点。冯恒阳同学主要编写了Matlab代码，定义了三个class类，提供了生成曲面的接口，并且描述了控制点变换的函数，以便于后期的对于动画的调整。最后一同进行了动画的制作。

# 第二章 原理简介

## 2.1 直纹面

首先将两条曲线进行参数化，得到和，通过将上述两直线进行线性组合可以得到



其中

## 2.2 贝塞尔曲面

次的贝塞尔曲面是由一条正交的次贝塞尔曲线和一条正交的次贝塞尔曲线组成，具体公式如下：



其中和是*Bernstein*基函数,是控制点位置矢量，和是多项式的次数，*Bernstein*基函数的表达式具体如下所示：



对于双三次贝塞尔曲面有可以将上述表达式转换为矩阵表达式



经化简可以得到



## 2.3 曲面拼接

在模型中，我们广泛运用了G0和C0连续，保证了曲面的连续性，在关键的场合，我们也使用了G1和C1连续。其中G0和C0连续只是曲面的两条线段的连续，并没有对其导函数有更进一步的要求，而C1连续要求曲面的两端的导函数也要相同，G1连续则要求每组三个控制点共线，且线段的长度比相同。

## 2.4 几何变换原理

在本模型当中，为了实现某一部位的移动，需要通过几何变换的方式，对于图形的控制点的坐标进行变换。因为图形是由控制点坐标、控制点之间的拓扑关系以及组成图形的表达模型所决定的，对于一个具体的图形，其拓扑关系和表达模型是固定的，对其进行几何变换是改变控制点坐标。

我们需要将点的坐标写成齐次坐标矩阵



通过如下的变换公式对于控制点进行变换



其中为变换前图形的控制点齐次坐标矩阵，为变换后图形的控制点齐次坐标矩阵，为变换矩阵。如下为变换矩阵对应的变换变换形式。

1. 比例变换



其中分别是方向的比例因子。

1. 平移变换



其中分别是方向的平移量。

1. 旋转变换
   1. 绕轴旋转的旋转变换



* 1. 绕轴旋转的旋转变换



* 1. 绕轴旋转的旋转变换



# 第三章 代码实现

为了方便代码的管理，我们使用了Matlab面向对象的编程来实现，为了方便展示代码的数据类型，使用C++来描述类中的详细数据。

## 3.1 *control\_point*类

*control\_point*代表了我们模型中的一个控制点这个小单元，是最为底层的一个类，我们认为对于一个控制点我们需要四个变量对其进行定义，其中*m\_x*，*m\_y*，*m\_z*分别代表了控制点在世界坐标系的的坐标。

在这个类当中还有一个*m\_level*代表了此控制点的属性。我们在进行代码编写的时候，考虑了人体运动的一些常识，当我们需要进行一些运动，必然是因为骨骼带动皮肤进行运动，我们通过*m\_level*这个属性来表述控制点归属于哪个骨骼。由于时间有限，我们仅仅定义了属于身体、左臂、右臂和头四种类型的控制点。当进行坐标变换的时候，我们便可以通过识别这些控制点的属性，有选择的尽心控制。如下是对于*control\_point*类的定义：

1. **class** control\_point{
2. **double** m\_x;
3. **double** m\_y;
4. **double** m\_z;
5. **int** m\_level;
6. };

## 3.2 *control\_mesh*类

*control\_mesh*类是模型中的第二级管理，主要掌管了对于一个面片的控制。在*control\_mesh*类中我们添加了此种类的方法，用于生成面片的三维模型，我们定义此方法为*make\_mesh*。在面片中，我们认为只需要包括*control\_point*类在内的三种变量便可以对于一个面片进行准确的描述。其中m\_type表示了此面片的种类，以便于在执行make\_mesh方法的过程中，识别应该调用怎样的程序生成对应的网格。如下是对于*control\_mesh*类的定义：

1. **class** control\_mesh{
2. **string** m\_type;
3. **string** m\_color;
4. **control\_point** m\_point[16];
5. **void** make\_mesh();
6. };

## 3.3 *control\_joint*类

*control\_joint*类作为整个模型的最高级别管理，对于整个*control\_mesh*类进行管理，*control\_joint*类中定义了与骨骼和模型本身相关，我们认为只需要5个变量便可以表示整个模型，其中除了*m\_mesh*变量中记录了模型所有的*control\_mesh*类，还有其他的四个量，分别记录了模型本身在世界坐标系的坐标，和左肩膀、右肩膀以及头关节的坐标，为坐标的变换提供原点。*control\_joint*类中还存在三个方法作用域*control\_joint*类。这三个方法通过遍历的方式，对于所有需要控制的控制点进行坐标变换。如下为*control\_joint*类的定义，其中对于*move\_world,move\_head,move\_shoulder\_r,move\_shoulder\_l*的方法所对应的*m\_level*写在注释当中:

1. **class** control\_joint{
2. **double** m\_world[3];//
3. **double** m\_head[3];//level = 1
4. **double** m\_shoulder\_l[3];//level = 2
5. **double** m\_shoulder\_r[3];//level = 3
6. **control\_mesh** mesh[286];
7. **void** move\_world(**double** x,**double** y,**double** z,**double** alpha,**double** beta,**double** gammar);
8. **void** move\_head(**double** x,**double** y,**double** z,**double** alpha,**double** beta,**double** gammar);
9. **void** move\_shoulder\_l(**double** x,**double** y,**double** z,**double** alpha,**double** beta,**double** gammar);
10. **void** move\_shoulder\_r(**double** x,**double** y,**double** z,**double** alpha,**double** beta,**double** gammar);
11. };

## 3.4 曲面生成、动画制作

通过对于*control\_mesh*类中的所有曲面进行遍历，便可以得到图像，通过*control\_joint*类中*move*命令便可以对于我们期望的控制点进行变换。通过每一次的循环，我们将图片保存下来，通过Matlab自带的*VideoWriter*函数将所有的图片变成*avi*格式的视频。通过*Adobe Premiere*进行进一步的剪辑，便可以得到一个活动的模型。

# 第四章 心得体会

通过这次大作业，加深了我们对于贝塞尔曲面等各种曲面的认识，同时对于曲面与曲面之间的关系有了更加深刻的认识。通过程序的编写，大大的提高了自己的编程水平，认为自己对于Matlab语言的掌握更加深刻，对于面向对象的语言有了更进一步的了解。

# 第五章 代码

## 5.1 *main.m*

%% Copyright (C) 2022:

% - Hengyang Feng <hengyang9311@163.com>

% - Hanqi Zhu <277577697@qq.com>

% - Junxian Fang<bang990104@naver.com>

% All Rights Reserved.

%% main

initial;

scrsz = get(0,'ScreenSize');

figure1=figure('Position',[0 0 1920 1080]);

v = VideoWriter('fufu.avi','Uncompressed AVI');

v.FrameRate = 60;

open(v);

for j = 1: 769

hold on;

view(73,18);

xlim([-3 5]);ylim([-3 3]);zlim([-3 3]);

axis image;axis off;

for i = 1:size(joint.m\_mesh,2)

[Sx,Sy,Sz] = joint.m\_mesh(i).make\_mesh();

end

joint.move\_head(0,0,0,head(j),0,0);

joint.move\_shoulder\_l(0,0,0,shoulder\_l(1,j),0,shoulder\_l(2,j));

joint.move\_shoulder\_r(0,0,0,shoulder\_r(1,j),0,shoulder\_r(2,j));

F = getframe(gcf);

writeVideo(v,F);

clf;

end

close(v);

close all;

hold off;

## 5.2 *control\_point.m*

% Copyright (C) 2022:

% - Hengyang Feng <hengyang9311@163.com>

% - Hanqi Zhu <277577697@qq.com>

% - Junxian Fang<bang990104@naver.com>

% All Rights Reserved.

classdef control\_point < handle

properties

m\_x,

m\_y,

m\_z,

m\_level

end

methods

function obj = control\_point(x,y,z,level)

obj.m\_y = x;

obj.m\_z = y;

obj.m\_x = z;

obj.m\_level = level;

end

end

end

## 5.3 *control\_mesh.m*

% Copyright (C) 2022:

% - Hengyang Feng <hengyang9311@163.com>

% - Hanqi Zhu <277577697@qq.com>

% - Junxian Fang<bang990104@naver.com>

% All Rights Reserved.

classdef control\_mesh < handle

properties

m\_type,

m\_m,

m\_n,

m\_point

m\_color

end

methods

function obj = control\_mesh(type,point,color)

obj.m\_m = 3;

obj.m\_n = 3;

obj.m\_type = type;

obj.m\_color = color;

obj.m\_point = point;

if type == "Bezier32"

obj.m\_n = 2;

elseif type == "Bezier22"

obj.m\_m = 2;obj.m\_n = 2;

end

end

function [Sx,Sy,Sz] = make\_mesh(obj)

if obj.m\_type == "Bezier33"

px = reshape([obj.m\_point(1).m\_x

obj.m\_point(2).m\_x

obj.m\_point(3).m\_x

obj.m\_point(4).m\_x

obj.m\_point(5).m\_x

obj.m\_point(6).m\_x

obj.m\_point(7).m\_x

obj.m\_point(8).m\_x

obj.m\_point(9).m\_x

obj.m\_point(10).m\_x

obj.m\_point(11).m\_x

obj.m\_point(12).m\_x

obj.m\_point(13).m\_x

obj.m\_point(14).m\_x

obj.m\_point(15).m\_x

obj.m\_point(16).m\_x],4,4)';

py = reshape([obj.m\_point(1).m\_y

obj.m\_point(2).m\_y

obj.m\_point(3).m\_y

obj.m\_point(4).m\_y

obj.m\_point(5).m\_y

obj.m\_point(6).m\_y

obj.m\_point(7).m\_y

obj.m\_point(8).m\_y

obj.m\_point(9).m\_y

obj.m\_point(10).m\_y

obj.m\_point(11).m\_y

obj.m\_point(12).m\_y

obj.m\_point(13).m\_y

obj.m\_point(14).m\_y

obj.m\_point(15).m\_y

obj.m\_point(16).m\_y],4,4)';

pz = reshape([obj.m\_point(1).m\_z

obj.m\_point(2).m\_z

obj.m\_point(3).m\_z

obj.m\_point(4).m\_z

obj.m\_point(5).m\_z

obj.m\_point(6).m\_z

obj.m\_point(7).m\_z

obj.m\_point(8).m\_z

obj.m\_point(9).m\_z

obj.m\_point(10).m\_z

obj.m\_point(11).m\_z

obj.m\_point(12).m\_z

obj.m\_point(13).m\_z

obj.m\_point(14).m\_z

obj.m\_point(15).m\_z

obj.m\_point(16).m\_z],4,4)';

Rm = 20;Rn = 20;

u = linspace(0,1,Rm)';

w = linspace(0,1,Rn)';

Bu = zeros([Rm , obj.m\_m + 1]);

Bw = zeros([Rn , obj.m\_n + 1]);

for i = 0:obj.m\_m

Bu(:,i+1) = prod(i+1:obj.m\_m)/prod(1:obj.m\_m-i)\*u.^i.\*(1-u).^(obj.m\_m-i);

end

for i = 0:obj.m\_n

Bw(:,i+1) = prod(i+1:obj.m\_n)/prod(1:obj.m\_n-i)\*w.^i.\*(1-w).^(obj.m\_n-i);

end

Sx = Bu \* px \* Bw';

Sy = Bu \* py \* Bw';

Sz = Bu \* pz \* Bw';

s = surf(Sx,Sy,Sz);

s.LineStyle = "none";

s.FaceColor = obj.m\_color;

elseif obj.m\_type == "Bezier32"

px = reshape([obj.m\_point(1).m\_x

obj.m\_point(2).m\_x

obj.m\_point(3).m\_x

obj.m\_point(4).m\_x

obj.m\_point(5).m\_x

obj.m\_point(6).m\_x

obj.m\_point(7).m\_x

obj.m\_point(8).m\_x

obj.m\_point(9).m\_x

obj.m\_point(10).m\_x

obj.m\_point(11).m\_x

obj.m\_point(12).m\_x],3,4)';

py = reshape([obj.m\_point(1).m\_y

obj.m\_point(2).m\_y

obj.m\_point(3).m\_y

obj.m\_point(4).m\_y

obj.m\_point(5).m\_y

obj.m\_point(6).m\_y

obj.m\_point(7).m\_y

obj.m\_point(8).m\_y

obj.m\_point(9).m\_y

obj.m\_point(10).m\_y

obj.m\_point(11).m\_y

obj.m\_point(12).m\_y],3,4)';

pz = reshape([obj.m\_point(1).m\_z

obj.m\_point(2).m\_z

obj.m\_point(3).m\_z

obj.m\_point(4).m\_z

obj.m\_point(5).m\_z

obj.m\_point(6).m\_z

obj.m\_point(7).m\_z

obj.m\_point(8).m\_z

obj.m\_point(9).m\_z

obj.m\_point(10).m\_z

obj.m\_point(11).m\_z

obj.m\_point(12).m\_z],3,4)';

Rm = 20;Rn = 20;

u = linspace(0,1,Rm)';

w = linspace(0,1,Rn)';

Bu = zeros([Rm , obj.m\_m + 1]);

Bw = zeros([Rn , obj.m\_n + 1]);

for i = 0:obj.m\_m

Bu(:,i+1) = prod(i + 1:obj.m\_m) / prod(1:obj.m\_m - i) \* u .^ i .\* (1 - u) .^ (obj.m\_m - i);

end

for i = 0:obj.m\_n

Bw(:,i+1) = prod(i + 1:obj.m\_n) / prod(1:obj.m\_n - i) \* w .^ i .\* (1 - w) .^ (obj.m\_n - i);

end

Sx = Bu \* px \* Bw';

Sy = Bu \* py \* Bw';

Sz = Bu \* pz \* Bw';

s = surf(Sx,Sy,Sz);

s.LineStyle = "none";

s.FaceColor = obj.m\_color;

elseif obj.m\_type == "Bezier22"

px = reshape([obj.m\_point(1).m\_x

obj.m\_point(2).m\_x

obj.m\_point(3).m\_x

obj.m\_point(4).m\_x

obj.m\_point(5).m\_x

obj.m\_point(6).m\_x

obj.m\_point(7).m\_x

obj.m\_point(8).m\_x

obj.m\_point(9).m\_x],3,3)';

py = reshape([obj.m\_point(1).m\_y

obj.m\_point(2).m\_y

obj.m\_point(3).m\_y

obj.m\_point(4).m\_y

obj.m\_point(5).m\_y

obj.m\_point(6).m\_y

obj.m\_point(7).m\_y

obj.m\_point(8).m\_y

obj.m\_point(9).m\_y],3,3)';

pz = reshape([obj.m\_point(1).m\_z

obj.m\_point(2).m\_z

obj.m\_point(3).m\_z

obj.m\_point(4).m\_z

obj.m\_point(5).m\_z

obj.m\_point(6).m\_z

obj.m\_point(7).m\_z

obj.m\_point(8).m\_z

obj.m\_point(9).m\_z],3,3)';

Rm = 20;Rn = 20;

u = linspace(0,1,Rm)';

w = linspace(0,1,Rn)';

Bu = zeros([Rm , obj.m\_m + 1]);

Bw = zeros([Rn , obj.m\_n + 1]);

for i = 0:obj.m\_m

Bu(:,i+1) = prod(i + 1:obj.m\_m) / prod(1:obj.m\_m - i) \* u .^ i .\* (1 - u) .^ (obj.m\_m - i);

end

for i = 0:obj.m\_n

Bw(:,i+1) = prod(i + 1:obj.m\_n) / prod(1:obj.m\_n - i) \* w .^ i .\* (1 - w) .^ (obj.m\_n - i);

end

Sx = Bu \* px \* Bw';

Sy = Bu \* py \* Bw';

Sz = Bu \* pz \* Bw';

s = surf(Sx,Sy,Sz);

s.LineStyle = "none";

s.FaceColor = obj.m\_color;

elseif obj.m\_type == "circle"

Sx = zeros([100,101]);Sy = zeros([100,101]);Sz = zeros([100,101]);

o\_1\_x = (obj.m\_point(1).m\_x + obj.m\_point(3).m\_x) / 2;

o\_1\_y = (obj.m\_point(1).m\_y + obj.m\_point(3).m\_y) / 2;

o\_1\_z = (obj.m\_point(1).m\_z + obj.m\_point(3).m\_z) / 2;

o\_2\_x = (obj.m\_point(4).m\_x + obj.m\_point(6).m\_x) / 2;

o\_2\_y = (obj.m\_point(4).m\_y + obj.m\_point(6).m\_y) / 2;

o\_2\_z = (obj.m\_point(4).m\_z + obj.m\_point(6).m\_z) / 2;

a\_1\_x = obj.m\_point(2).m\_x - o\_1\_x;

a\_1\_y = obj.m\_point(2).m\_y - o\_1\_y;

a\_1\_z = obj.m\_point(2).m\_z - o\_1\_z;

b\_1\_x = obj.m\_point(1).m\_x - o\_1\_x;

b\_1\_y = obj.m\_point(1).m\_y - o\_1\_y;

b\_1\_z = obj.m\_point(1).m\_z - o\_1\_z;

a\_2\_x = obj.m\_point(5).m\_x - o\_2\_x;

a\_2\_y = obj.m\_point(5).m\_y - o\_2\_y;

a\_2\_z = obj.m\_point(5).m\_z - o\_2\_z;

b\_2\_x = obj.m\_point(4).m\_x - o\_2\_x;

b\_2\_y = obj.m\_point(4).m\_y - o\_2\_y;

b\_2\_z = obj.m\_point(4).m\_z - o\_2\_z;

u = linspace(0,1,100);

v = 0 : 2 \* pi / 100 : 2 \* pi;

for i = 1 : 100

o\_x = u(i) \* o\_1\_x + (1 - u(i)) \* o\_2\_x;

o\_y = u(i) \* o\_1\_y + (1 - u(i)) \* o\_2\_y;

o\_z = u(i) \* o\_1\_z + (1 - u(i)) \* o\_2\_z;

a\_x = u(i) \* a\_1\_x + (1 - u(i)) \* a\_2\_x;

a\_y = u(i) \* a\_1\_y + (1 - u(i)) \* a\_2\_y;

a\_z = u(i) \* a\_1\_z + (1 - u(i)) \* a\_2\_z;

b\_x = u(i) \* b\_1\_x + (1 - u(i)) \* b\_2\_x;

b\_y = u(i) \* b\_1\_y + (1 - u(i)) \* b\_2\_y;

b\_z = u(i) \* b\_1\_z + (1 - u(i)) \* b\_2\_z;

for j = 1 : 101

Sx(i,j) = o\_x + a\_x \* sin(v(j)) + b\_x \* cos(v(j));

Sy(i,j) = o\_y + a\_y \* sin(v(j)) + b\_y \* cos(v(j));

Sz(i,j) = o\_z + a\_z \* sin(v(j)) + b\_z \* cos(v(j));

end

end

s = surf(Sx,Sy,Sz);

s.LineStyle = "none";

s.FaceColor = obj.m\_color;

elseif obj.m\_type == "line"

Rm = 20;Rn = 20;

u = linspace(0,1,Rm);

w = linspace(0,1,Rn);

Sx = zeros([Rm,Rn]);Sy = zeros([Rm,Rn]);Sz = zeros([Rm,Rn]);

for i = 1:Rm

for j = 1:Rn

Pux = obj.m\_point(1).m\_x \* (1 - u(i)) + obj.m\_point(2).m\_x \* u(i);

Qux = obj.m\_point(3).m\_x \* (1 - u(i)) + obj.m\_point(4).m\_x \* u(i);

Sx(i,j) = Pux \* (1 - w(j)) + Qux \* w(j);

Puy = obj.m\_point(1).m\_y \* (1 - u(i)) + obj.m\_point(2).m\_y \* u(i);

Quy = obj.m\_point(3).m\_y \* (1 - u(i)) + obj.m\_point(4).m\_y \* u(i);

Sy(i,j) = Puy \* (1 - w(j)) + Quy \* w(j);

Puz = obj.m\_point(1).m\_z \* (1 - u(i)) + obj.m\_point(2).m\_z \* u(i);

Quz = obj.m\_point(3).m\_z \* (1 - u(i)) + obj.m\_point(4).m\_z \* u(i);

Sz(i,j) = Puz \* (1 - w(j)) + Quz \* w(j);

end

end

s = surf(Sx,Sy,Sz);

s.LineStyle = "none";

s.FaceColor = obj.m\_color;

elseif obj.m\_type == "tui"

px = [obj.m\_point(1).m\_y

obj.m\_point(2).m\_y

obj.m\_point(3).m\_y

obj.m\_point(4).m\_y

obj.m\_point(5).m\_y

obj.m\_point(6).m\_y

obj.m\_point(7).m\_y

obj.m\_point(8).m\_y

obj.m\_point(9).m\_y

obj.m\_point(10).m\_y

obj.m\_point(11).m\_y

obj.m\_point(12).m\_y

obj.m\_point(13).m\_y

obj.m\_point(14).m\_y

obj.m\_point(15).m\_y];

py = [obj.m\_point(1).m\_z

obj.m\_point(2).m\_z

obj.m\_point(3).m\_z

obj.m\_point(4).m\_z

obj.m\_point(5).m\_z

obj.m\_point(6).m\_z

obj.m\_point(7).m\_z

obj.m\_point(8).m\_z

obj.m\_point(9).m\_z

obj.m\_point(10).m\_z

obj.m\_point(11).m\_z

obj.m\_point(12).m\_z

obj.m\_point(13).m\_z

obj.m\_point(14).m\_z

obj.m\_point(15).m\_z];

m = 14;Rm = 50;Bu = zeros([Rm , m + 1]);

u = linspace(0,1,Rm);v = 0 : 2 \* pi / 100 : 2 \* pi;

Sx = zeros([Rm,101]);Sy = zeros([Rm,101]);Sz = zeros([Rm,101]);

for i = 0 : m

Bu(:,i + 1) = prod(i + 1:m) / prod(1:m - i) \* u .^ i .\* (1 - u) .^ (m - i);

end

Dx = Bu \* px;Dy = Bu \* py;

for i = 1 : Rm

r = abs(Dx(i) - obj.m\_point(15).m\_y);

for j = 1:101

Sx(i,j) = obj.m\_point(15).m\_y + r \* cos(v(j));

Sy(i,j) = Dy(i);

Sz(i,j) = r \* sin(v(j));

end

end

s = surf(Sz,Sx,Sy);

s.LineStyle = "none";

s.FaceColor = obj.m\_color;

end

end

end

end

## 5.4 *control\_joint.m*

% Copyright (C) 2022:

% - Hengyang Feng <hengyang9311@163.com>

% - Hanqi Zhu <277577697@qq.com>

% - Junxian Fang<bang990104@naver.com>

% All Rights Reserved.

classdef control\_joint < handle

properties

m\_world,%double[3]

m\_head,%double[3]

m\_shoulder\_l,%double[3]

m\_shoulder\_r,%double[3]

m\_mesh,%control\_mesh

m\_a;%constant

end

methods

function obj = control\_joint(m\_mesh)

obj.m\_world = [0 0 0];

obj.m\_head = [0.006061,-3.55271e-15,2.42653];

obj.m\_shoulder\_l = [0,0.469285,2.14514];

obj.m\_shoulder\_r = [0,-0.469285,2.14514];

obj.m\_mesh = m\_mesh;

obj.m\_a = 0.5;

end

function move\_world(obj,x,y,z,alpha,beta,gammar)

for i = 1:size(obj.m\_mesh,2)

for j = 1:16

n = [obj.m\_mesh(i).m\_point(j).m\_x

obj.m\_mesh(i).m\_point(j).m\_y

obj.m\_mesh(i).m\_point(j).m\_z

1]';

T1 = [1 0 0 0;

0 1 0 0;

0 0 1 0;

-obj.m\_world(1) -obj.m\_world(2) -obj.m\_world(3) 1];

T2 = [1 0 0 0;

0 1 0 0;

0 0 1 0;

x y z 1;];

T3 = [1 0 0 0;

0 cos(alpha) sin(alpha) 0;

0 -sin(alpha) cos(alpha) 0;

0 0 0 1];

T4 = [cos(beta) 0 -sin(beta) 0;

0 1 0 0;

sin(beta) 0 cos(beta) 0;

0 0 0 1];

T5 = [cos(gammar) sin(gammar) 0 0;

-sin(gammar) cos(gammar) 0 0;

0 0 1 0;

0 0 0 1;];

T6 = [1 0 0 0;

0 1 0 0;

0 0 1 0;

obj.m\_world(1) obj.m\_world(2) obj.m\_world(3) 1];

temp = n \* T1 \* T2 \* T3 \* T4 \* T5 \* T6;

obj.m\_mesh(i).m\_point(j).m\_x = temp(1);

obj.m\_mesh(i).m\_point(j).m\_y = temp(2);

obj.m\_mesh(i).m\_point(j).m\_z = temp(3);

end

end

obj.m\_world(1) = obj.m\_world(1) + x;

obj.m\_world(2) = obj.m\_world(2) + y;

obj.m\_world(3) = obj.m\_world(3) + z;

obj.m\_head(1) = obj.m\_head(1) + x;

obj.m\_head(2) = obj.m\_head(2) + y;

obj.m\_head(3) = obj.m\_head(3) + z;

obj.m\_shoulder\_l(1) = obj.m\_shoulder\_l(1) + x;

obj.m\_shoulder\_l(2) = obj.m\_shoulder\_l(2) + y;

obj.m\_shoulder\_l(3) = obj.m\_shoulder\_l(3) + z;

obj.m\_shoulder\_r(1) = obj.m\_shoulder\_r(1) + x;

obj.m\_shoulder\_r(2) = obj.m\_shoulder\_r(2) + y;

obj.m\_shoulder\_r(3) = obj.m\_shoulder\_r(3) + z;

end

function move\_head(obj,x,y,z,alpha,beta,gammar)

for i = 1:size(obj.m\_mesh,2)

for j = 1:16

if obj.m\_mesh(i).m\_point(j).m\_level == 1

n = [obj.m\_mesh(i).m\_point(j).m\_x

obj.m\_mesh(i).m\_point(j).m\_y

obj.m\_mesh(i).m\_point(j).m\_z

1]';

T1 = [1 0 0 0;

0 1 0 0;

0 0 1 0;

-obj.m\_head(1) -obj.m\_head(2) -obj.m\_head(3) 1];

T2 = [1 0 0 0;

0 1 0 0;

0 0 1 0;

x y z 1;];

T3 = [1 0 0 0;

0 cos(alpha) sin(alpha) 0;

0 -sin(alpha) cos(alpha) 0;

0 0 0 1];

T4 = [cos(beta) 0 -sin(beta) 0;

0 1 0 0;

sin(beta) 0 cos(beta) 0;

0 0 0 1];

T5 = [cos(gammar) sin(gammar) 0 0;

-sin(gammar) cos(gammar) 0 0;

0 0 1 0;

0 0 0 1;];

T6 = [1 0 0 0;

0 1 0 0;

0 0 1 0;

obj.m\_head(1) obj.m\_head(2) obj.m\_head(3) 1];

temp = n \* T1 \* T2 \* T3 \* T4 \* T5 \* T6;

obj.m\_mesh(i).m\_point(j).m\_x = temp(1);

obj.m\_mesh(i).m\_point(j).m\_y = temp(2);

obj.m\_mesh(i).m\_point(j).m\_z = temp(3);

else

continue

end

end

end

obj.m\_head(1) = obj.m\_head(1) + x;

obj.m\_head(2) = obj.m\_head(2) + y;

obj.m\_head(3) = obj.m\_head(3) + z;

end

function move\_shoulder\_l(obj,x,y,z,alpha,beta,gammar)

for i = 1:size(obj.m\_mesh,2)

for j = 1:16

if obj.m\_mesh(i).m\_point(j).m\_level == 2

n = [obj.m\_mesh(i).m\_point(j).m\_x

obj.m\_mesh(i).m\_point(j).m\_y

obj.m\_mesh(i).m\_point(j).m\_z

1]';

T1 = [1 0 0 0;

0 1 0 0;

0 0 1 0;

-obj.m\_shoulder\_l(1) -obj.m\_shoulder\_l(2) -obj.m\_shoulder\_l(3) 1];

T2 = [1 0 0 0;

0 1 0 0;

0 0 1 0;

x y z 1;];

T3 = [1 0 0 0;

0 cos(alpha) sin(alpha) 0;

0 -sin(alpha) cos(alpha) 0;

0 0 0 1];

T4 = [cos(beta) 0 -sin(beta) 0;

0 1 0 0;

sin(beta) 0 cos(beta) 0;

0 0 0 1];

T5 = [cos(gammar) sin(gammar) 0 0;

-sin(gammar) cos(gammar) 0 0;

0 0 1 0;

0 0 0 1;];

T6 = [1 0 0 0;

0 1 0 0;

0 0 1 0;

obj.m\_shoulder\_l(1) obj.m\_shoulder\_l(2) obj.m\_shoulder\_l(3) 1];

temp = n \* T1 \* T2 \* T3 \* T4 \* T5 \* T6;

obj.m\_mesh(i).m\_point(j).m\_x = temp(1);

obj.m\_mesh(i).m\_point(j).m\_y = temp(2);

obj.m\_mesh(i).m\_point(j).m\_z = temp(3);

else

continue

end

end

end

obj.m\_shoulder\_l(1) = obj.m\_shoulder\_l(1) + x;

obj.m\_shoulder\_l(2) = obj.m\_shoulder\_l(2) + y;

obj.m\_shoulder\_l(3) = obj.m\_shoulder\_l(3) + z;

end

function move\_shoulder\_r(obj,x,y,z,alpha,beta,gammar)

for i = 1:size(obj.m\_mesh,2)

for j = 1:16

if obj.m\_mesh(i).m\_point(j).m\_level == 3 || obj.m\_mesh(i).m\_point(j).m\_level == 31

n = [obj.m\_mesh(i).m\_point(j).m\_x

obj.m\_mesh(i).m\_point(j).m\_y

obj.m\_mesh(i).m\_point(j).m\_z

1]';

T1 = [1 0 0 0;

0 1 0 0;

0 0 1 0;

-obj.m\_shoulder\_r(1) -obj.m\_shoulder\_r(2) -obj.m\_shoulder\_r(3) 1];

T2 = [1 0 0 0;

0 1 0 0;

0 0 1 0;

x y z 1;];

T3 = [1 0 0 0;

0 cos(alpha) sin(alpha) 0;

0 -sin(alpha) cos(alpha) 0;

0 0 0 1];

T4 = [cos(beta) 0 -sin(beta) 0;

0 1 0 0;

sin(beta) 0 cos(beta) 0;

0 0 0 1];

T5 = [cos(gammar) sin(gammar) 0 0;

-sin(gammar) cos(gammar) 0 0;

0 0 1 0;

0 0 0 1;];

T6 = [1 0 0 0;

0 1 0 0;

0 0 1 0;

obj.m\_shoulder\_r(1) obj.m\_shoulder\_r(2) obj.m\_shoulder\_r(3) 1];

temp = n \* T1 \* T2 \* T3 \* T4 \* T5 \* T6;

obj.m\_mesh(i).m\_point(j).m\_x = temp(1);

obj.m\_mesh(i).m\_point(j).m\_y = temp(2);

obj.m\_mesh(i).m\_point(j).m\_z = temp(3);

else

continue

end

end

end

obj.m\_shoulder\_r(1) = obj.m\_shoulder\_r(1) + x;

obj.m\_shoulder\_r(2) = obj.m\_shoulder\_r(2) + y;

obj.m\_shoulder\_r(3) = obj.m\_shoulder\_r(3) + z;

end

end

end