

**本科毕业设计（论文）**

**深度相机网格化**

**学 院 计算机学院**

**专 业 计算机科学与技术**

**年级班别 计科2班**

**学 号 3113005848**

**学生姓名 魏永倬**

**指导教师 杨卓**

**2017 年5月**

**面向康复训练的随访数据管理系统的设计与实现**

**魏**

**计算机学院**

**摘 要**

随着计算机的快速发展和日益普及，各种软件系统遍及我们生活的每个角 落，为我们的学习和工作带来极大的方便。医院对神经损伤、骨骼损伤、肌肉损伤等这些需要康复训练的患者，会进行一些周期性的随访，了解患者的康复情况，并跟据这些随访信息的反馈，医生做出一些适当的康复训练调整，以便于让患者得到更好的康复效果。本设计主要针对上述信息设计出面向康复训练的随访数据管理系统。系统是用JSP制作的前端页面，SSH框架搭建的后台，MySql储存信息，以web的显示效果展示。系统中医生可以对患者进行随访和医疗，可以选择随访方式对患者进行随访，随访的时候还可以上传图片来得到一些随访的信息，患者可以评价医生的医疗效果，即时的更新医生的得分。有效地解决了康复训练的随访问题。

**关键字**：康复训练、随访、运动医疗

**Abstract**

With the rapid development of computers and the growing popularity of various software systems throughout every corner of our lives, it brings great convenience for our study and work.These patients require hospital rehabilitation nerve damage, bone damage, muscle injury, does some periodic follow-up, to understand the patient's rehabilitation and follow-up with them, according to the feedback information, physicians make some appropriate adjustments rehabilitation in order to enable patients to get better rehabilitation.This design is mainly designed for the above information for the rehabilitation of follow-up data management system. JSP system is made the front page, SSH framework set up backstage, MySql store information to display web display.System doctor can follow-up the patient and health care, you can choose the way of follow-up patients were followed up when you can also upload pictures to get some follow-up information, patients can evaluate the doctor's medical effects, real-time score updates physician.Effectively solve the problem of follow-up rehabilitation.

**Key words:** rehabilitation training, follow-up, sports medicine

**目 录**

[**1 绪论** 1](#_Toc481921613)

[**1.1题目背景及目的** 1](#_Toc481921614)

[**1.2国内外研究前景** 1](#_Toc481921615)

[**1.3论文构成及研究内容** 1](#_Toc481921616)

[**2 可行性分析** 2](#_Toc481921617)

[**2.1技术可行性分析** 2](#_Toc481921618)

[**2.2经济可行性分析** 2](#_Toc481921619)

[**2.3社会意义可行性分析** 2](#_Toc481921620)

[**3 开发工具和技术** 3](#_Toc481921621)

[**3.1 运行环境** 3](#_Toc481921622)

[3.1.1 硬件环境 3](#_Toc481921623)

[3.1.2 软件环境 3](#_Toc481921624)

[**3.2 开发环境** 3](#_Toc481921625)

[3.2.1 硬件配置 3](#_Toc481921626)

[3.2.2 软件配置 3](#_Toc481921627)

[**3.3 开发软件以及第三方库** 3](#_Toc481921628)

[3.3.1 Visual Studio 2015 3](#_Toc481921629)

[3.3.1 Qt 5.8.0 3](#_Toc481921630)

[3.3.1 OpenCV 3.1.0 3](#_Toc481921631)

[3.3.1 OpenGL 3.3 3](#_Toc481921632)

[3.3.1 FFmpeg 3.2.4 3](#_Toc481921633)

[3.3.1 Protobuf 3.0.0 3](#_Toc481921634)

[3.3.1 cmake 3.7.1 3](#_Toc481921635)

[**4 Kinect数据介绍** 4](#_Toc481921636)

[**4.1 彩图数据** 4](#_Toc481921637)

[**4.2 深度数据** 4](#_Toc481921638)

[**4.3 骨骼数据** 5](#_Toc481921639)

[**5 概要设计** 7](#_Toc481921640)

[**5.1计算机配置** 7](#_Toc481921641)

[5.1.1在电脑上配置java的运行环境JDK 7](#_Toc481921642)

[5.1.2安装数据库MySql 7](#_Toc481921643)

[5.1.3准备好开发工具Eclipse 或者My Eclipse 7](#_Toc481921644)

[5.1.4下载web插件服务器，如tomcat等 7](#_Toc481921645)

[5.1.5下载开发时要用到的jar包 7](#_Toc481921646)

[**5.2系统模块结构设计** 7](#_Toc481921647)

[5.2.1模块图 7](#_Toc481921648)

[5.2.2各模块信息 7](#_Toc481921649)

[**5.3传输控制协议设计** 7](#_Toc481921650)

[5.3.1 传输控制协议的协议头 7](#_Toc481921651)

[5.3.3 连接协议 8](#_Toc481921652)

[5.3.4 骨骼,三维点云协议 8](#_Toc481921653)

[**5.4传输Kinect数据协议设计** 9](#_Toc481921654)

[5.4.1 **客户端到服务端（仅在建立连接后发送一次）** 9](#_Toc481921655)

[协议头（和控制套接字用的是同一类型的头部） 9](#_Toc481921656)

[5.4.2 **服务端到客户端** 10](#_Toc481921657)

[**5.5设计模式** 11](#_Toc481921658)

[5.4.1 MVC设计模式 11](#_Toc481921659)

[5.4.2 DAO设计模式 11](#_Toc481921660)

[5.4.3 工厂模式 11](#_Toc481921661)

[5.4.4单例模式 11](#_Toc481921662)

[**6 详细设计** 11](#_Toc481921663)

[**6.1功能模块的实现** 11](#_Toc481921664)

[6.1.1注册功能 11](#_Toc481921665)

[6.1.2登录功能 11](#_Toc481921666)

[6.1.3个人信息修改功能 11](#_Toc481921667)

[6.1.4求诊功能 11](#_Toc481921668)

[6.1.5评价功能 11](#_Toc481921669)

[6.1.6治疗功能 11](#_Toc481921670)

[6.1.7随访功能 11](#_Toc481921671)

[**6.2 SSH设计思路** 11](#_Toc481921672)

[6.2.1 SSH代码块的实现流程 11](#_Toc481921673)

[**7 维护与测试** 12](#_Toc481921674)

[**7.1系统曾出现的疑难杂症** 12](#_Toc481921675)

[**7.2运行截图** 12](#_Toc481921676)

[**7**.2.1 程序启动 12](#_Toc481921677)

[**7**.2.2 设备列表刷新 13](#_Toc481921678)

[**7**.2.3 设备数据显示 13](#_Toc481921679)

[**结论** 16](#_Toc481921680)

[**参考文献** 17](#_Toc481921681)

[**致谢** 18](#_Toc481921682)

**1 绪论**

**1.1题目背景及目的**

**1.2国内外研究前景**

。

**1.3论文构成及研究内容**

**2 可行性分析**

**2.1技术可行性分析**

**2.2经济可行性分析**

**2.3社会意义可行性分析**

**3 开发工具和技术**

**3.1 运行环境**

3.1.1 硬件环境

GPU：主频1.8GHz以上

内存：1G

硬盘：可用空间10G以上

显卡：支持OpenGL 3.3以上

3.1.2 软件环境

运行操作系统：Win7/win10 64bit

**3.2 开发环境**

3.2.1 硬件配置

GPU：主频2.6GHz以上

内存：4G以上

硬盘：可用空间50G以上

显卡：支持OpenGL 3.3以上

3.2.2 软件配置

运行操作系统：Win7/win10 64bit

**3.3 开发软件以及第三方库**

3.3.1 Visual Studio 2015

3.3.1 Qt 5.8.0

3.3.1 OpenCV 3.1.0

3.3.1 OpenGL 3.3

3.3.1 FFmpeg 3.2.4

3.3.1 Protobuf 3.0.0

3.3.1 cmake 3.7.1

**4 Kinect数据介绍**

**4.1 彩图数据**

作为一款深度相机，Kinect拥有最基础的RGB摄像头，可以获取到一般摄像机可以获取到的彩色图像。



图4-1 普通彩图数据

**4.2 深度数据**

Kinect设备和普通摄像机不同的在于它拥有一个特殊的功能，那就是三维数据的获取。此设备除了包含RGB摄像头外，还包含了一个红外发射器及红外接收器，如下图所示。



图4-2 Kinect的设备结构

通过这两个组件，Kinect无须通过运用算法对获取到的彩图进行分析得到深度图，而是直接获取到深度图信息，从而大幅度提升了实时性和准确性。众所周知，实时视频或实时电脑游戏对实时性和准确性要求是非常之高的，而Kinect设备则很好的提升这一性能。

深度图即是每个像素点都有对应一个深度数值的图像，使用灰度图来表示深度数据，可得到如下图像，颜色越深代表该点越远离Kinect设备，否则相反。（如果深度中有人体数据，Kinect SDK for Window会使用一个彩色颜色空间中的单色来表示人形）



图4-3 灰度图表示的深度数据

**4.3 骨骼数据**

Kinect SDK for Window 提供了三维骨骼数据结构体格式如下：

typedef struct \_NUI\_SKELETON\_FRAME {

   LARGE\_INTEGER liTimeStamp;

   DWORD dwFrameNumber;

   DWORD dwFlags;

   Vector4 vFloorClipPlane;

   Vector4 vNormalToGravity;

   NUI\_SKELETON\_DATA SkeletonData[NUI\_SKELETON\_COUNT];

} NUI\_SKELETON\_FRAME;

一个Kinect设备支持捕捉6个人，而精确到骨骼信息的最多是2人，每个人记录20个骨骼部位的三维信息。骨骼位置如下图：

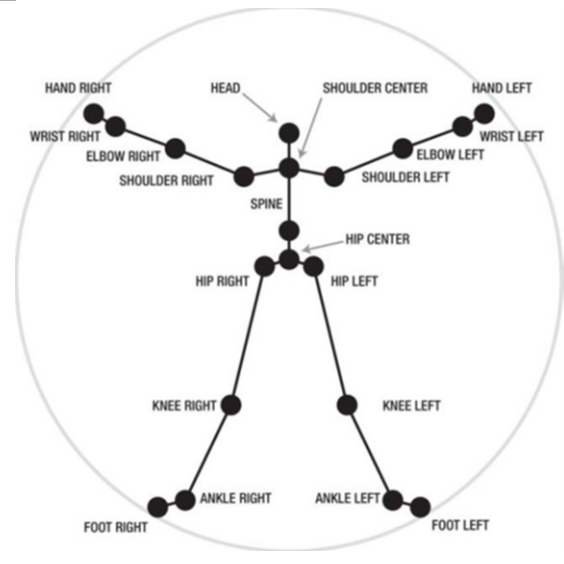


图4-4-1三维骨骼数据追踪点

从NUI\_SKELETON\_FRAME结构体中获取需要的数据—点坐标信息以及点之间的连接信息，并将这些数据转换多段折线数据，如下图所示。

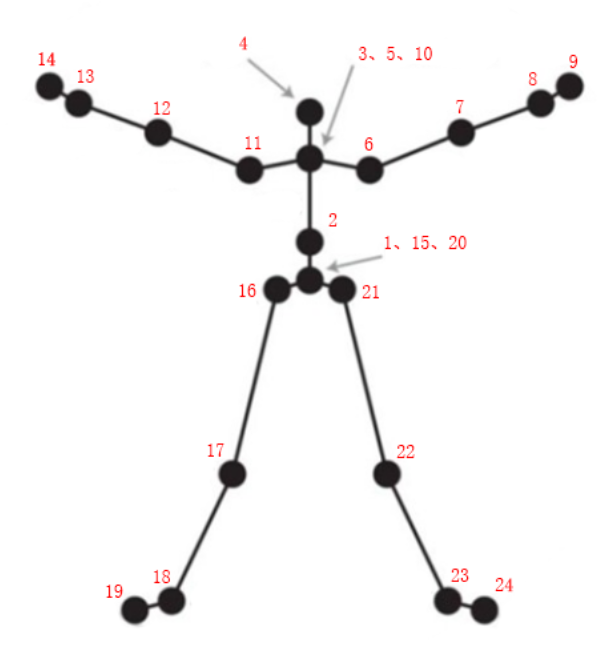


图4-4-2骨骼数据五段折线图

人体骨骼可以分为五段折线，分别为以下五段：点1、2、3、4；点5、6、7、8、9；点10、11、12、13、14；点15、16、17、18、19；点20、21、22、23、24。其中3、5、10是同一点，1、15、20则是另外的一个同一点。而需要从服务器传输到客户端的骨骼数据就是这5段折线。视Kinect工作的实际环境情况，从获取到的数据转换为5段折线，这些折线中的点数可能不严格如上图所示，可能会少一两个点，但并不影响这5段折线组合成一个人体骨骼。

**5 概要设计**

**5.1计算机配置**

5.1.1在电脑上配置java的运行环境JDK

5.1.2安装数据库MySql

5.1.3准备好开发工具Eclipse 或者My Eclipse

5.1.4下载web插件服务器，如tomcat等

5.1.5下载开发时要用到的jar包

**5.2系统模块结构设计**

5.2.1模块图

5.2.2各模块信息

**5.3传输控制协议设计**

5.3.1 传输控制协议的协议头

传输内容使用Protobuf封装、解析。由于TCP是面向字节流的，所以，需要加一个头部进行标记分包；而且，需要标记指令类型，使通讯双方依据不同指令类型而做出不同响应；最后，还需要包含一些版本信息、包体新旧判断、包体大小（传输内容的大小）。

根据以上要求，头部包含以下信息：

1. 指令类型：针对请求类型划分。
2. 指令号：划分某一指令类型的多个指令。
3. 序列号：用于判断指令是否过期的。
4. 版本号：标记指令版本号，而使用不同版本封装、解析方式。
5. 包体长度：用于对TCP数据分包。

数据类型要求：

1. 传输使用大端字节流
2. char类型的长度为一个字节
3. int32、uint32类型的长度为4个字节

|  |  |
| --- | --- |
| 意义 | 数据类型 |
| 指令类型 | char |
| 指令号 | char |
| 序列号 | int32 |
| 版本号 | char |
| 包体长度 | int32 |
| 包体长度 | uint32 |
| 包体(Protobuf封装的指令内容) | |

5.3.2 保活协议

考虑到网络各种异常情况，如丢包、抖动、延时等。设计了应用层的保活协议，防止连接断开后，客户端、服务端仍当连接存在，继续传输。

|  |  |  |  |
| --- | --- | --- | --- |
| 功能 | 指令类型 | 指令号 | 内容 |
| 客户端主动保活 | 99 | 1 |  |
| 服务器回应客户端的保活 | 99 | 100 |  |

5.3.3 连接协议

|  |  |  |  |
| --- | --- | --- | --- |
| 功能 | 指令类型 | 指令号 | 内容 |
| 请求连接 | 1 | 1 | {  hostname:”string”  } |
| 退出连接 | 1 | 2 |  |
| 请求连接回复 | 1 | 100 | {  resultType:“int32”  //1成功，-100失败  failReason:“string”  guid:”string”  colorPort:“int32”//端口号  depthPort:”int32”  skelePort:”int32”  } |
| 请求设备列表 | 1 | 3 |  |
| 回复设备列表 | 1 | 101 | {  deviceList: “string”  Separator: “string”  } |

5.3.4 骨骼,三维点云协议

|  |  |  |  |
| --- | --- | --- | --- |
| 功能 | 指令类型 | 指令号 | 内容 |
| 开始请求 | 2 | 1 | {  deviceName:”string”  } |
| 回复开始请求 | 2 | 100 | {  deviceName:”string”  resultType:“int32”  //1成功，-100失败  failReason:“string”  } |
| 服务器主动结束 | 2 | 101 | {  deviceName:”string”  Type:”int32” // 1:socket 断开 ，2:Kinect设备断开  reason:”string” // 断开连接的任意附加字符串信息  }备注：设备断开后将发生此事件 |
| 结束请求 | 2 | 2 | {  deviceName:”string”  reason:”string”// 断开连接的任意附加字符串信息  } |

备注:

1. 一个客户端仅对应一个控制套接字，此外客户端是个单例，也就是每台PC上只允许同时运行一个客户端程序
2. 每个控制套接字可以请求多个设备的数据，但不允许同时请求同一设备的数据两次或多次。亦就是说，如果一个客户端成功请求了某一设备的数据，即不能再次请求该设备的数据，直到该设备的数据传输套接字全部断开。即使由于未知错误发生这种重复请求，该连接请求也会被服务器忽略，不予受理。
3. 客户端向服务器结束请求时发送设备名字，服务器仅断开该设备的所有数据套接字（颜色、深度、骨骼），而不会影响该客户端与服务器的其他设备的数据传输。
4. 在控制套接字上，客户端主动请求退出连接或者未知错误断开了与服务器的连接，服务器会断开该客户端所有设备的数据套接字连接。
5. GUID由服务器产生，客户端的数据套接字返回该GUID。服务器端用以关联控制套接字 和 该控制套接字对应的所有数据套接字。

**5.4传输Kinect数据协议设计**

5.4.1 **客户端到服务端（仅在建立连接后发送一次）**

协议头（和控制套接字用的是同一类型的头部）

|  |  |
| --- | --- |
| 意义 | 数据类型 |
| 指令类型 | char |
| 指令号 | char |
| 序列号 | int32 |
| 版本号 | char |
| 包体长度 | uint32 |
| 包体长度 | uint32 |
| 包体 | |

数据信道 （数据套接字）

|  |  |  |  |
| --- | --- | --- | --- |
| 功能 | 指令类型 | 指令号 | 内容 |
| 由数据传输socket发送到服务器端(用于关联控制信道和数据信道套接字) | 3 | 1 | {  deviceName:”string”  guid:”string”  } |

5.4.2 **服务端到客户端**

协议帧（彩色、深度）

|  |  |
| --- | --- |
| **意义** | **数据类型** |
| 帧编号 | uint32 |
| 数据类型 | uint32 |
| 包体长度 | uint32 |
| 包体  H264数据流 | |

协议帧（骨骼）

|  |  |
| --- | --- |
| **意义** | **数据类型** |
| 帧编号 | uint32 |
| 数据类型 | uint32 |
| 包体长度 | uint32 |
| 包体   |  |  | | --- | --- | | 宽度 | short | | 高度 | short | | 线段数(N) | char | | 每段点个数 | char[N] | | 点数组 | ushort[N\*M] | | |

**5.5设计模式**

5.4.1 MVC设计模式

5.4.2 DAO设计模式

5.4.3 工厂模式

5.4.4单例模式

**6 详细设计**

**6.1功能模块的实现**

6.1.1注册功能

6.1.2登录功能

6.1.3个人信息修改功能

6.1.4求诊功能

6.1.5评价功能

6.1.6治疗功能

6.1.7随访功能

**6.2 SSH设计思路**

6.2.1 SSH代码块的实现流程

**7 维护与测试**

**7.1系统曾出现的疑难杂症**

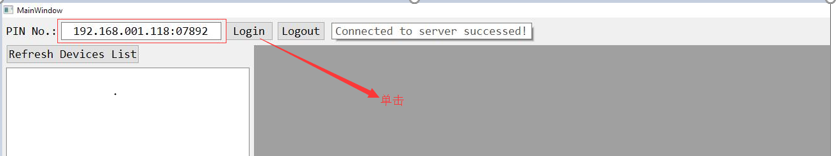
**表**7.1 **系统出现的错误整理**

|  |  |  |
| --- | --- | --- |
| **问题** | **问题说明** | **解决程度** |
| 页面注册提交错误 | 用户在注册数据类型的填错，导致页面和后台无法交互。 | 已解决 |
| 登陆跳转过程中session混乱现象 | 不同的用户同时登录系统时出现用户名相同的现象，个人信息相同等现象。 | 已解决 |
| 页面兼容性问题 | 页面没经过兼容性处理，建议使用流行的浏览器访问，例如IE8+，chrome，opera等（兼容性问题还没有处理）。 | 没解决 |
| 数据库写入错误，页面返回信息提示，后台不会报错 | 修改信息的时候因为级联的关系，导致同时操作几张表是出现不同程度的错误。 | 已解决 |
| 退出后，系统回退刷新，session未清楚 | 账号已经退出的情况下，再次刷新页面session遗留，没有做到退出的效果 | 已解决 |
| Jar包版本问题，出现的后台内部错误 | 两个jar包，因为版本不同，出现后台混乱的错误，单个jar包支持的框架不一样。 | 已解决 |
| Tomcat因为不明原因会产生奔溃现象 | 本来tomcat是可用的，经过一段时间，发布进去的的工程无法运行。 | 已解决 |

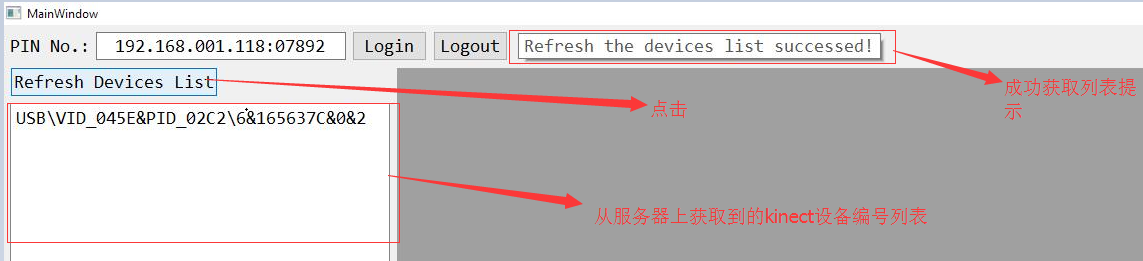
**7.2运行截图**

7.2.1 程序启动

1. 打开程序，在Pin No.的输入框输入服务器方提供的ip:port或者pin码，然后点击Login按钮连接到服务器。如下图所示。若成功连接到服务器上，软件的右上方会有对应提示。

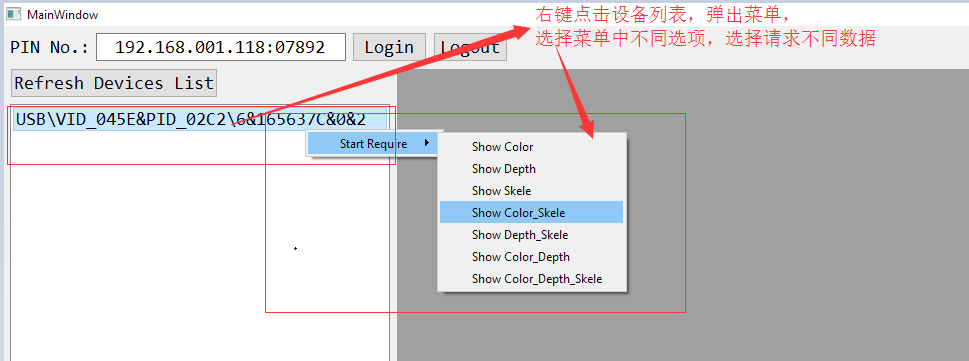


1. 点击刷新设备列表按钮，获取kinect设备编号列表。服务器将返回设备的列表。



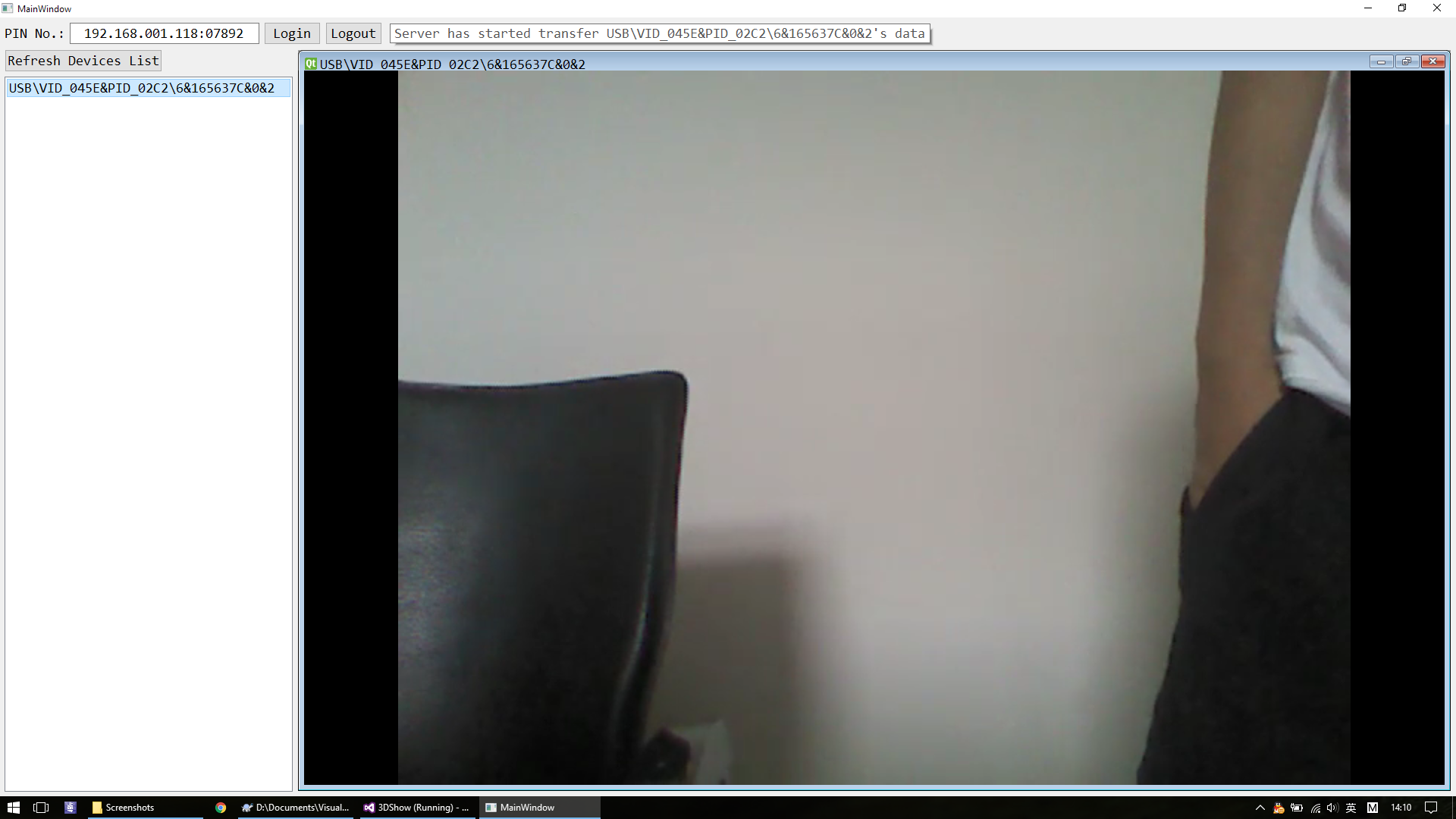
7.2.2 设备列表刷新

选择不同设备，向服务器请求各种数据



7.2.3 设备数据显示

1. 彩色数据



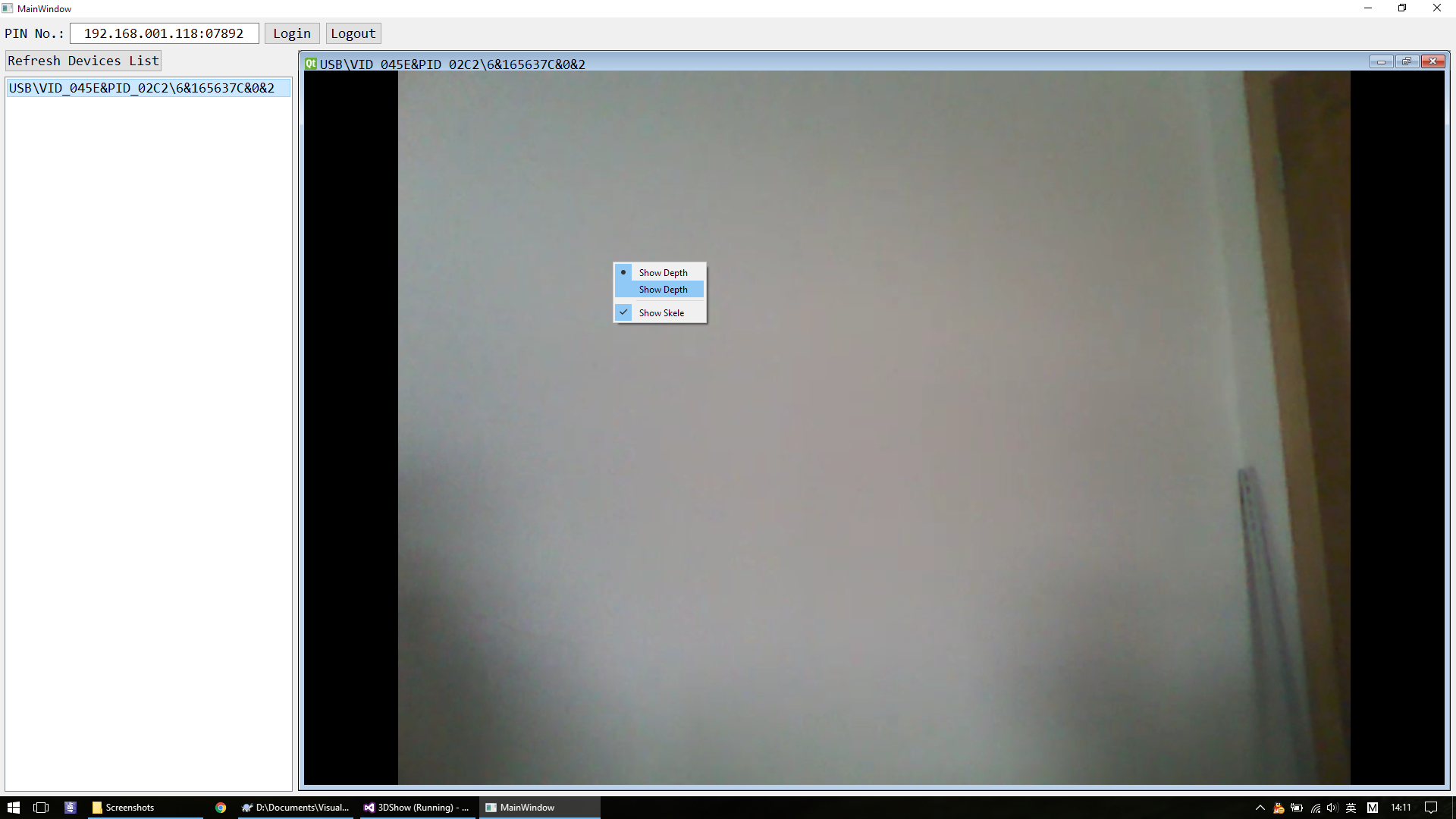
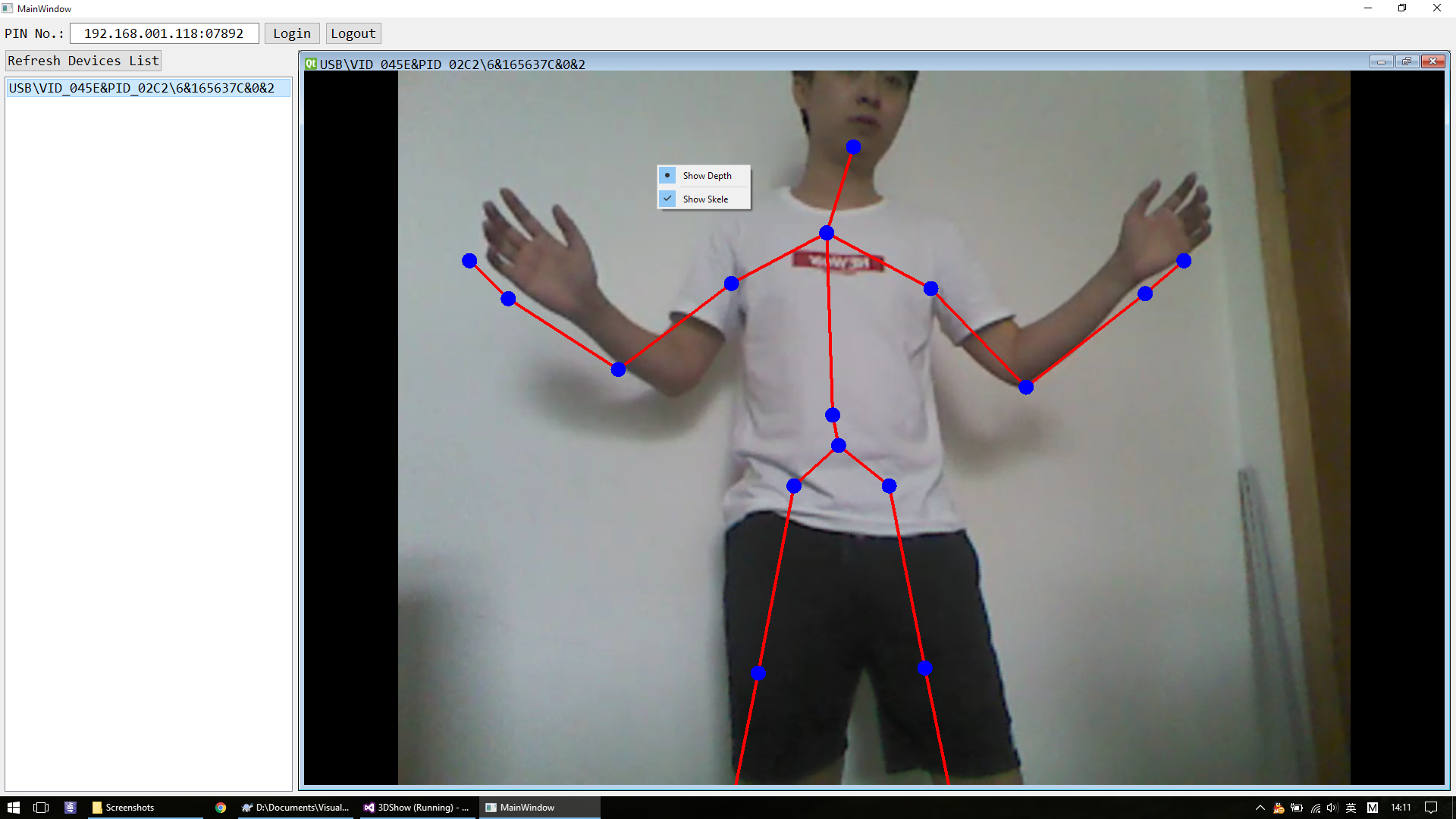
1. 深度数据



1. 骨骼数据



1. 多种数据同时显示在同一窗口



**结论**

**参考文献**

1. 谷钰，杨艳斌，王泽生.Eclipse插件体系结构的研究[J].电脑知识与技术，2009-11-05：17-20.
2. 刘磊，朱敏，徐彦军.基于Struts框架的Java Web应用开发技术[J].计算机应用与软件，2007-05：1-10.

**致谢**

在此次毕业设计的过程中杨卓老师和战教授不辞幸苦的督促我们完成毕业设计，从选题到设计再到最后的成果老师时刻都在督促我们认真完成毕业设计，还有想法让我们参加几个比赛，但是由于我们的能力有限和付出的可能有点少辜负了老师，感到法非常的抱歉。但是我们还是非常的感谢指导老师为我们的付出，感谢老师不舍余力的对我们因材施教，在此表示由衷的谢意。也感谢同学们不厌其烦的帮助，感谢小组成员的帮助,他们帮助了我很多，曾经失落的时候鼓励过我，激励过我。我再次感谢再此过程中帮助过我的老师，同学。