**版本信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本信息记录表** | | | | |
| 版本号 | 修订内容 | 修订人 | 修订日期 | 审核人 |
| 1.0 | 初版 | 党慧杰 | 2018.08.20 |  |
| 2.0 | 修订 | 党慧杰 | 2018.10.30 |  |
|  |  |  |  |  |

# 手势控制彩灯

## 【实验目的】

1、利用kinect的骨骼数据识别试手

2、根据手势控制彩灯

## 【实验原理】

Kinect采用TOF技术来获取被测物体的深度。红外发射器主动投射经调制的近红外光线，红外光线照到视野里的物体上就会发生反射，红外相机接收反射回来的红外线，采用TOF技术测量深度，计算光的时间差（通常是通过相位差来计算的），从而得到物体的深度（即物体到深度相机的距离）。

骨架追踪系统将检测到的深度图像数据转换到人体各个关节点坐标。该系统能同时追踪6个人，识别2个人的动作；每个人包含25个关节点数据，包括了X、Y、Z数据。Kinect采用分隔策略将人体从复杂的背景中区分出来，接下来Kinect会对深度图像进行评估（机器学习），来判断人体的不同部位。

**1. 向右摆手手势判别**

* Step1满足的条件：

1）左手手掌位于左手肘部前面，右手垂下（低于肩膀）

2）左手手掌高于臀关节

3）左手手掌位于左肩膀左侧

* Step2满足的条件：

1）左手手掌位于左手肘部前面，右手垂下（低于肩膀）

2）左手手掌高于臀关节

3）左手位于左肩膀右侧，但位于右肩膀左侧

* Step3满足的条件：

1）左手手掌位于左手肘部前面，右手垂下（低于肩膀）

2）左手手掌高于臀关节

3）左手手掌位于右肩膀右侧

**2. 向左摆手手势判别**

* Step1满足的条件：

1）右手手掌位于右手肘部前面，左手垂下（低于肩膀）

2）右手手掌高于臀关节

3）右手手掌位于右肩膀右侧

* Step2满足的条件：

1）右手手掌位于右手肘部前面，左手垂下（低于肩膀）

2）右手手掌高于臀关节

3）左手位于右肩膀左侧，但位于左肩膀右侧

* Step3 满足的条件：

1）右手手掌位于右手肘部前面，左手垂下（低于肩膀）

2）右手手掌高于臀关节

3）右手手掌位于左肩膀左侧

## 【实验环境】

开发环境

Windows 8或Windows 10系统、VS2012及以上版本、 Kinect设备、PC机必须具备USB3.0接口

编程语言

C/C++

## 【实验步骤】

### 一、打开项目

本实验在《人体骨骼及动作识别》实验的基础上进行。

1）打开文档同级目录‘KinectGesture’下的KinectDemo工程。如图所示：

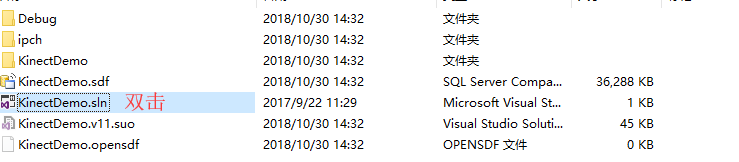


图 1

2）修改IDD\_BODY窗口

如果下图所示，打开IDD\_BODY窗口，修改窗体的名称为‘手势控制’。并在界面中添加如下按钮：ID为IDC\_BUTTON\_GET\_IP，名称为‘获取爱遥控IP’的按钮；ID为IDC\_BUTTON\_OPEN\_LAMP，名称为‘开灯’的按钮；ID为IDC\_BUTTON\_CLOSE\_LAMP，名称为‘关灯’的按钮；ID为IDC\_BUTTON\_SWITCH\_COLOR，名称为‘切换灯光颜色’的按钮。

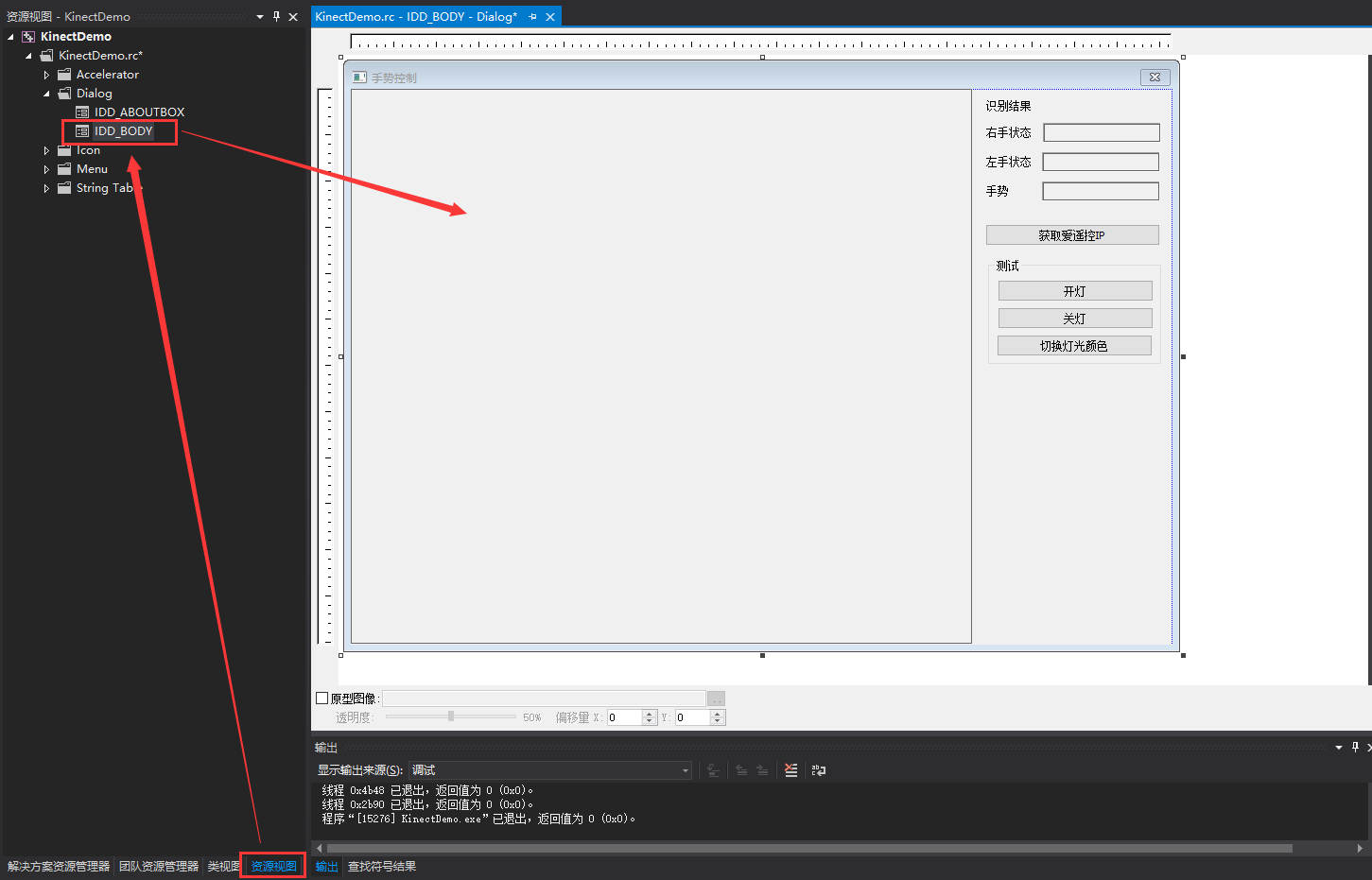


图 2

### 二、爱遥控的使用

将文档同级目录‘iyk’文件夹中的文件拷贝到工程目录/KinextDemo/中，并添加到工程中。其中IYKController负责向爱遥控设备发送数据；LightCode是已经学习好的彩灯控制的红外吗。

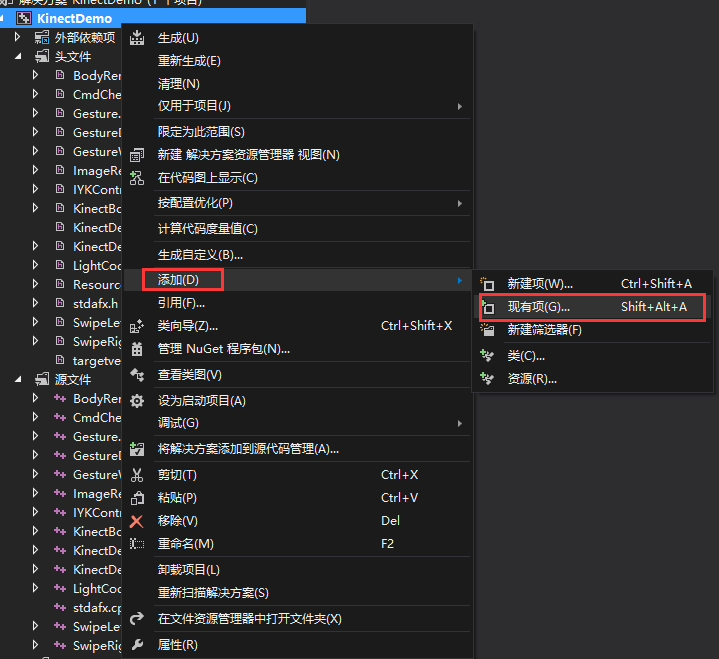


图 3

**1. 初始化爱遥控设备**

1）打开KinectBody.cpp，添加如下头文件：

#include "IYKController.h"

#include "LightCode.h"

2）在KinectBody.cpp中声明如下全局函数和变量：

// 灯光控制相关

// 切换灯光颜色

void ChangeLampColor(bool rightOrLeft);

// 颜色号

int nColorIndex = 0;

// 爱遥控设备控制类

IYKController \*pGlobalIYKController = NULL;

**2. 获取爱遥控设备IP**

打开KinectBody.cpp，在DlgProc函数中，添加按钮IDC\_BUTTON\_GET\_IP的点击事件的响应代码，在switch(message)添加如下分支：

case WM\_COMMAND:

wmId = LOWORD(wParam);

wmEvent = HIWORD(wParam);

switch (wmId)

{

// 获取爱遥控IP

case IDC\_BUTTON\_GET\_IP:

if (pGlobalIYKController == NULL)

{

pGlobalIYKController = new IYKController();

}

pGlobalIYKController->GetDeviceIP("IYK\_ZJJKQCZN");

if (pGlobalIYKController->IsDeviceAvailable())

{

pGlobalIYKController->InitUdpSocket();

SetDlgItemText(hWnd, IDC\_BUTTON\_GET\_IP, L"已获取爱遥控IP");

}

else

{

MessageBox(hWnd, L"获取IP失败", L"提示", MB\_OK);

}

break;

default:

return DefWindowProc(hWnd, message, wParam, lParam);

}

break;

注意：GetDeviceIP函数中传递的参数为爱遥控设备的名称，需要更加实际设备进行修改；爱遥控设备的配置和红外码学习可以参考文档《红外码学习》。

### 三、控制彩灯

彩灯控制的红外定义在LightCode.h中，包含：开、关、红、绿、蓝五个红外码。可以根据文档《红外码学习》添加更多的控制码。

**1. 打开灯**

开灯的红外码定义为Light\_On\_Code。通过IYKController发送该数据即可。在KinectBody.cpp的DlgProc函数中WM\_COMMAND分支中的switch(wmId)中添加如下分支代码：

case IDC\_BUTTON\_OPEN\_LAMP:

if (pGlobalIYKController != NULL)

{

pGlobalIYKController->SendCmdCode(Light\_On\_Code, LIGHT\_CODE\_LEN);

}

break;

**2. 关闭灯**

开灯的红外码定义为Light\_Off\_Code。同1，在switch(wmId)中添加如下分支代码：

case IDC\_BUTTON\_CLOSE\_LAMP:

if (pGlobalIYKController != NULL)

{

pGlobalIYKController->SendCmdCode(Light\_Off\_Code, LIGHT\_CODE\_LEN);

}

break;

**3. 切换灯光颜色**

实验中定义了三种颜色：Ligth\_Red\_Code、Light\_Green\_Code和Light\_Blue\_Code。

1）之前声明的ChangeLampColor函数，该函数根据是向左招手还是向右招手实现颜色的切换。向右摆手时灯的颜色以红、绿、蓝循环显示；向左摆手时灯的颜色以蓝、绿、红循环显示。实现代码如下：

void ChangeLampColor(bool rightOrLeft)

{

if (pGlobalIYKController)

{

switch (nColorIndex)

{

default:

case 0:

pGlobalIYKController->SendCmdCode(Light\_Red\_Code, LIGHT\_CODE\_LEN);

break;

case 1:

pGlobalIYKController->SendCmdCode(Light\_Green\_Code, LIGHT\_CODE\_LEN);

break;

case 2:

pGlobalIYKController->SendCmdCode(Light\_Blue\_Code, LIGHT\_CODE\_LEN);

break;

}

if (rightOrLeft)

{

nColorIndex ++;

if (nColorIndex >= LIGHT\_COLOR\_COUNT)

nColorIndex = 0;

}

else

{

nColorIndex --;

if (nColorIndex < 0)

{

nColorIndex = LIGHT\_COLOR\_COUNT;

}

}

}

}

2）同1，在switch(wmId)中添加如下分支代码：

case IDC\_BUTTON\_SWITCH\_COLOR:

ChangeLampColor(true);

break;

### 四、手势控制

**1. 手势控制开/闭**

规则如下：右手握拳时关闭彩灯；右手张开时打开彩灯。

在KinectBody.cpp的HandleBodyFrame函数快的最后添加如下代码：

// 手势开/关彩灯

static HandState lastRightHandleState = HandState\_Unknown;

hr = pBody->get\_HandRightState(&righHandState);

if (SUCCEEDED(hr) && righHandState != HandState\_Unknown && righHandState != lastRightHandleState)

{

if (righHandState == HandState\_Open)

{

if (pGlobalIYKController)

pGlobalIYKController->SendCmdCode(Light\_On\_Code, LIGHT\_CODE\_LEN);

}

else if (righHandState == HandState\_Closed)

{

if (pGlobalIYKController)

pGlobalIYKController->SendCmdCode(Light\_Off\_Code, LIGHT\_CODE\_LEN);

}

lastRightHandleState = righHandState;

}

**2. 手势切换彩灯颜色**

在上一节中已经实现了ChangeLampColor函数。现在只需要在左右摆手识别结果中调用该函数即可。

将手势识别结果回调函数修改如下：

void WINAPI OnGestureRecognized(GestureType type)

{

switch (type)

{

case GestureType::SwipeRight:

SetDlgItemText(hGlobalHwnd, IDC\_GESTURE, L"向右摆手");

ChangeLampColor(true);

break;

case GestureType::SwipeLeft:

SetDlgItemText(hGlobalHwnd, IDC\_GESTURE, L"向左摆手");

ChangeLampColor(false);

break;

default:

break;

}

}

红色标记部分为新添加代码。

### 五、编译运行

点击进行编译，如果编译出现错误，按照错误提示进行解决。

**1. 演示步骤**

1）配置爱遥控设备，可以参考文档《爱遥控配置》或《红外码学习》。

2）运行程序。

3）点击界面【手势控制】按钮，进入手势控制界面；

4）点击【获取爱遥空IP】按钮，当按钮文字更改为‘已获取爱遥控IP’表示获取成功，否则会弹出提示框。

5）点击测试区域的按钮，可以看到彩灯的变化。

6）当人站在Kinect设备的前面，并显示完整骨骼图时，举起左/右首做出百四的动作，可以看见彩灯颜色的切换。

## 【思考题】

### 选择题

题目1：下列说明中const char \*ptr；其中ptr应该是：（C）

A、指向字符常量的指针

B、指向字符的常量指针

C、指向字符串常量的指针

D、指向字符串的常量指针

题目2：KinectV2 能够追踪多少个人体骨骼关节点？(B)

A、24

B、25

C、26

D、27

### 2、简答题

const与define的区别？

1、编译器处理方式

const：编译时确定其值；

define：预处理时进行替换

2、类型检查

const：数据类型，编译时进行数据检查；

define：无类型，也不做类型检查

3、内存空间

const：在静态存储区储存，仅此一份；

define：在代码段区，每做一次替换就会进行一次拷贝

4、define可以用来防止重复定义，const不行