**Kinect中的骨骼坐标的实时提取**

1. **开发准备。**

**Visual studio 2013**

**Kinect**

**操作系统：**

**win10或者win8.1**

**环境配置:**

**Kinect for Windows SDK v2.0**

**Opencv2.4.9**

**环境配置的搭建：**

**至于怎样进行环境的搭建，我在这里就不进行详细的说明了，你可以在百度上搜索**

**win8.1 + VS2013 + Opencv2.4.9 + Kinect for Windows SDK v1.8开发环境配置这些关键字或者搜索一下这个网址：**

[**http://demo.netfoucs.com/kh1445291129/article/details/37879119**](http://demo.netfoucs.com/kh1445291129/article/details/37879119)

**来进行环境配置的搭建。PS：如果你是按我给你的这个网址进行环境搭建的话，有一个要注意的地方，它里面有一个地方时连接器->输入->附加依赖项->编辑下叫你复制的那一些文字，最后一个是Kinect10.lib的，因为这个网址的是用Kinect for Windows SDK v1.8开发的，所以这里就是Kinect10.lib，因为我用的是Kinect for Windows SDK v2.0开发的，所以这里你复制进去的时候应该是kinect20.lib到里面，而不是Kinect10.lib，否则，你进行编译的时候，会出现编译出错的问题。**

**这一阶段我觉得，我们不能操之过急，要慢慢来，特别是在环境配置的时候，按步骤，耐心的一步一步的做下去，编译出错了，然后回来查找错误，不行就直接重新再配一次，直到成功。**

1. **研究目的。**

**目的：通过Kinect来进行实时的控制Uarm机械臂。**

**控制方式：以右手的关节点的坐标作为Uarm机械臂的末端来进行控制，一开始把第一次获取右手坐标作为原点，此后获取的坐标与第一个坐标进行x、y、z进行坐标的作差，从而形成新的控制坐标。**

**对于Uarm机械臂的吸放，我们以Kinect捕捉到左手的手势来控制Uarm机械臂的吸放，左手的张开手势就是Uarm的吸，紧握就是Uarm的放。**

**对于程序的运行结束，我们以同样的道理来利用右手的手势来控制，当我们的右手的手势是张开的时候，程序就会结束。**

**因此，下面我就从两方面进行研究：1、右手坐标的实时提取；2、左右手的手势的实时提取。**

1. **右手坐标的实时提取。**

**1、首先初始化设备：**

**HRESULT hr = myKinect.InitializeDefaultSensor();//初始化设备;**

**初始化设备会调用这个函数：InitializeDefaultSensor()；**

**这个函数实现的功能有：**

**//搜索Kinect设备**

**hr = GetDefaultKinectSensor(&m\_pKinectSensor);**

**如果找不到设备，则初始化不成功，则进行不了后面的步骤；倘若找的的Kinect设备，则做下面的程序：**

**if (m\_pKinectSensor)**

**{**

**IBodyFrameSource\* pBodyFrameSource = NULL;//读取骨架;**

**IdepthFrameSource\* pDepthFrameSource = NULL;//读取深度信息;**

**IbodyIndexFrameSource\* pBodyIndexFrameSource = NULL;//读取背景二值图**

**//打开Kinect设备**

**hr = m\_pKinectSensor->Open();**

**//坐标的映射;**

**if (SUCCEEDED(hr))**

**{**

**hr = m\_pKinectSensor->get\_CoordinateMapper(&m\_pCoordinateMapper);**

**}**

**//身体骨架;**

**if (SUCCEEDED(hr))**

**{**

**hr = m\_pKinectSensor->get\_BodyFrameSource(&pBodyFrameSource);**

**}**

**//读取身体骨架;**

**if (SUCCEEDED(hr))**

**{**

**hr = pBodyFrameSource->OpenReader(&m\_pBodyFrameReader);**

**}**

**//深度frame;**

**if (SUCCEEDED(hr))**

**{**

**hr = pDepthFrameSource->OpenReader(&m\_pDepthFrameReader);**

**}**

**//身体二值frame;**

**if (SUCCEEDED(hr))**

**{**

**hr = m\_pKinectSensor->get\_BodyIndexFrameSource(&pBodyIndexFrameSource);**

**}**

**if (SUCCEEDED(hr))**

**{**

**hr = pBodyIndexFrameSouce->OpenReader(&m\_pBodyIndexFrameReader);**

**}**

**/\***

**下面还有一些代码，我只是写了一些重要的代码，主要是通过初始化设备，查找Kinect设备，并且打开Kinect设备，开始调用里面的函数，身体骨架，读取身体骨架，深度值frame，身体二值frame等等，来获取数据。**

**\*/**

**}**

**2、初始化完成后，因为实时的，所以要时刻进行刷新，这样才能将前一次的信息清空掉，执行下一次的数据的获取，因此下面主要是调用CBodyBasics::Update()函数。**

**原本函数是这样的：void CbodyBasics::Update()**

**现在我们将其改成：double\* CbodyBasics::Update(double \*array)**

**设成指针函数，将我们想要的结果存储在一个指针数组中，然后将其指向到主函数中去，从而实现数据的输出。**

**/\***

**写主要的程序，没有将全部程序放上去。**

**\*/**

**double\* CbodyBasics::Update(double \*array)**

**{**

**//每次先清空SkeletonImg**

**skeletonImg.steTo(0);**

**//检查Kinect是否丢失，若丢失则不做操作.**

**if (!m\_pBodyFramReader)**

**{**

**exit (1);**

**}**

**IbodyFream\* pBodyFrame = NULL;//骨架信息**

**IdepthFrame\* pDepthFrame = NULL;//深度信息**

**IbodyIndexFrame\* pBodyIndexFrame = NULL;//背景二值图;**

**//记录每次操作的成功与否**

**HRESULT hr = S\_OK;**

**//----------------------------获取深度数据并显示---------------------------------**

**if (SUCCEEDED(hr))**

**{**

**hr = m\_pDepthFrameReader->AcquireLatesetFrame(&pDepthFrame);//获取深度数据;**

**}**

**·**

**·**

**·**

**SafeRelease(pDepthFrame);//必须要释放，否则之后无法获得新的f rame数据;**

**//--------------------------------获取骨架并显示-----------------------------------**

**if (SUCCEEDED(hr))**

**{**

**hr = m\_pBodyFrameReader->AcquireLateseFrame(&pBodyFrame);//获取骨架信息;**

**}**

**if (SUCCEEDED(hr))**

**{**

**Ibody\* ppBodies[BODY\_COUNT] = { 0 };//每一Ibody可以追踪一个人，总共可以追踪六个人;**

**if (SUCCEEDED(hr))**

**{**

**//把Kinect追踪到的人的信息，分别存到每一个Ibody中;**

**hr = pBodyFrame->GetAndRefreshBodyData(\_countof(ppBodies),ppBodies);//\_countof(ppBodies)计算个数;**

**}**

**if (SUCCEEDED(hr))**

**{**

**HRESULT hr；**

**//对于每一个Ibody**

**for (int i = 0 ; i < BODY\_COUNT; ++i)**

**{**

**IBody\* pBody = ppBodies[i];**

**if (pBody)**

**{**

**BOOLEAN bTracked = false;**

**hr = pBody->get\_IsTracked(&bTracked);**

**if (SUCCEEDED(hr) && bTracked)**

**{**

**Joint joints[JointType\_Count];//存储关节点类;**

**HandState leftHandState = HandState\_UnKnown;//左手状态;**

**HandState rightHandState = HandState\_Unknown;//右手状态;**

**//获取左右状态;**

**pBody->get\_HandLeftState(&leftHandState);**

**pBody->get\_HandrightState(&rightHandState);**

**//存储深度坐标系中的关节点位置**

**DepthSpacePoint \*depthSpacePosition = new EdpthSpacePoint[\_countof(joints)];**

**//获得关节点类**

**hr = pBody->GetJoints(\_countof(joints),joints)’**

**if (SUCCEEDED(hr))**

**{**

**for (int j = 0 ; j < \_countof(jonits); ++j)**

**{**

**//将关节点坐标从摄像机坐标系转到深度坐标系中**

**m\_pCoordinateMapper->MapCameraPointToDepthSpace(joints[j].Position,&depthSpacePosition[j]);**

**}**

**/\***

**到此为止，现在已经将所有的关节点的位置坐标存储在joints[i].Position中只要将我们需要的坐标的值存储到我们新建的那个数组中即可**

**\*/**

**/\***

**这里有一个关节点位置的字符串数组：**

**std::string The\_Position[25];//定义关节点位置**

**有这样一个函数：**

**void ini\_OutputToFile()**

**{**

**char a[][30] =**

**{**

**“1 Spine\_Base: “,//脊柱底部**

**“2 Spine\_Mid:”,//脊柱·中部**

**·**

**·**

**“12 Hand\_Right:”,//右手**

**·**

**·**

**·**

**“25 Thumb\_Right:”,//右拇指**

**}**

**for (int i = 0 ; i < 25 ; i++)**

**The\_Position[i] = a[i];**

**}**

**因此，我们要的坐标是右手坐标，即：**

**joints[11].Position;**

**\*/**

**//因此做一下处理**

**static int h = 1;//获取第一次右手坐标的条件；**

**static double a , b ,c , d , e ,f;**

**for (int i = 0 ; i < 25 ; ++i)**

**{**

**if (i % 25 == 11)**

**{**

**if (h == 11)**

**{**

**a = joints[i].Position.X;**

**b = joints[i].Position.Y;**

**c = joints[i].Position.Z;**

**//倘若已经获取到坐标则值不为NULL，否则，跳出，直到获取到第一次数据为止**

**if (a == NULL && b == NULL && c == NULL) break;**

**h++;**

**}**

**else if (h > 1 )**

**{**

**d = joints[i].Position.X;**

**e = joints[i].Position.Y;**

**f = joints[i].Position.Z;**

**}**

**}**

**}**

**//用一个数组来存储对应的x、y、z坐标**

**array[0] = d – a;**

**array[1] = e – b;**

**array[2] = f – c;**

**·**

**·**

**·**

**//----------------hand state left-----------------------**

**DrawHandState(depthSpacePosition[JointType\_HandLeft],leftHandState);**

**DrawHandState(depthSpacePosition[JointTyep\_HandRight],rightHandState);**

**//将左右手的手势存储到数组中;**

**array[3] = rightHandState;//右手手势;**

**array[4] = leftHandState;//左手手势;**

**·**

**·**

**·**

**·**

**·**

**}**

**}**

**}**

**}**

**}**

**}**

**}**

**以上就是右手的坐标的实时提取。**

1. **左右手手势的实时提取。**

**其实在三过程里就已经将左右手手势的坐标给实时提取出来了，其中手势有如下值：**

**typedef enum \_HandState HandState:**

**enum \_HandState**

**{**

**HandState\_Unkonwn = 0,**

**HandState\_NotTracked = 1,**

**HandState\_Open = 2,**

**HandState\_Closed = 3,**

**HandState\_Lasso = 4**

**};**

**//手势识别的代码如下：**

**//-----------------------------------hand state left---------------------------**

**DrawHandState(depthSpacePosition[JointType\_HandLeft],leftHandState);**

**DrawHandState(depthSpacePosition[JointTyep\_HandRight],rightHandState);**

**//将左右手的手势存储到数组中;**

**array[3] = rightHandState;//右手手势;**

**array[4] = leftHandState;//左手手势;**

**到此左右手的手势实时提取已经完成了。**

1. **一些数据的处理。**

**在主函数中我们要将从Update()函数中获取的相应数据放到主函数中，因此定义一个指针数组，将数值整出来，如下：**

**double \*array1 = new double[5];//先进行定义;**

**if (SUCCEEDED(hr)) //初始化是否成功**

**{**

**while (1)**

**{**

**array1 = myKinect.Update(array1);//将值进行存放;**

**cout << array1[0] <<” “;**

**cout << array1[1] << “ “;**

**cout << array1[2] << “ “;**

**cout << array1[3] << “ “;**

**cout << array1[4] << endl;**

**//这样就已经将右手的坐标给提取出来了，还有左右手手势的状态也给整出来了**

**//右手来控制跳出循环;**

**if (array[3] == 2)//右手手势是张开的时候就能跳出循环了**

**{**

**break;**

**}**

**}**

**free (array1);//要将内存及时给释放掉；**

**}**

**else**

**{**

**cout << “Kinect initialization failed!” << endl;**

**}**

**最后，整个内容过程已经完成了，谢谢！**