

# 动捕数据到机械臂流程 踩坑一览

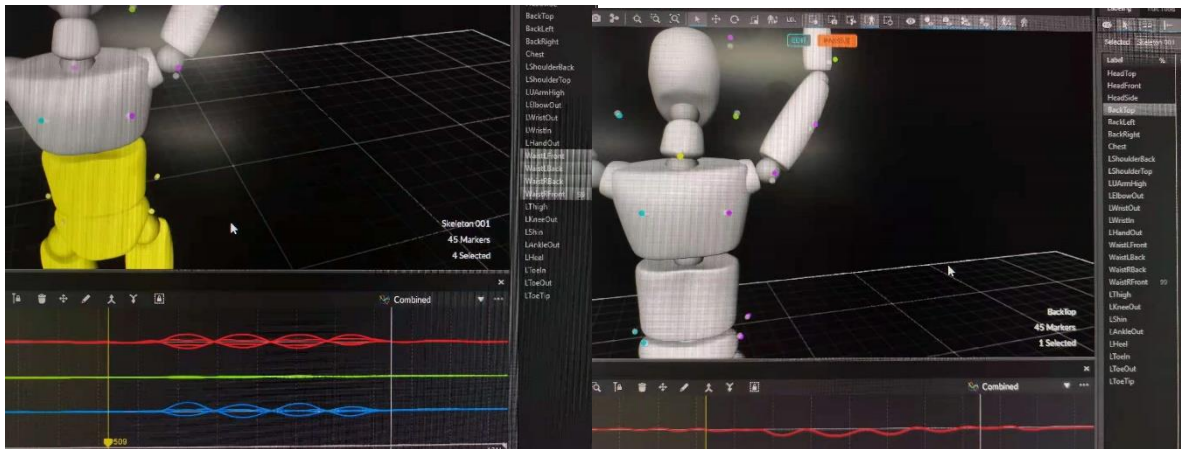
## Step 1: 贴点

一定要贴准！一定要贴准！一定要贴准！



主要是上半身的点，一定不能乱贴，尤其是：

- 1、胯部的四个点要水平（确定水平位置，否则直接裂开）
- 2、胸背部的中心点（确定坐标轴位置）
- 3、肘部关节（一定是关节移动不动的那个点）



## Step 2: 录动作 Motion Capture

目前是 41 个点的设置，用 T-Pose 生成后直接录制就可以了。录制的时候要注意不要动作做的太快，因为机器人的电位器速度最快约为每秒 360°，一秒之内关节最多转一圈。

角度限制：

转腰——左右各 90°

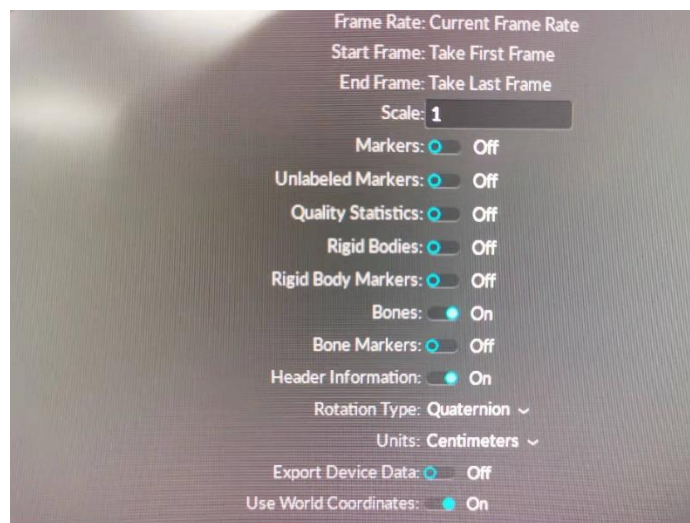
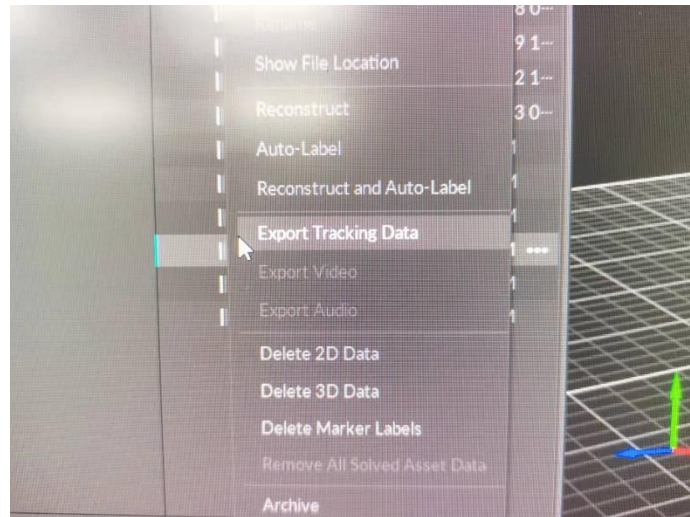
大臂张开——向上 120°

大臂前后——前后各 90°

转肘——约为 180°

以上均以录制时不穿模为最好

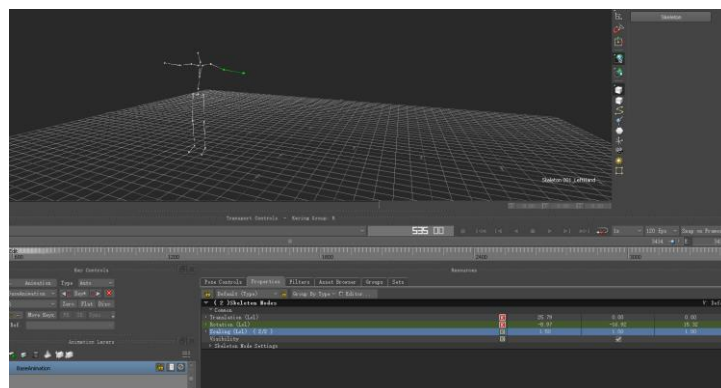




#### Step 4: 调整骨骼

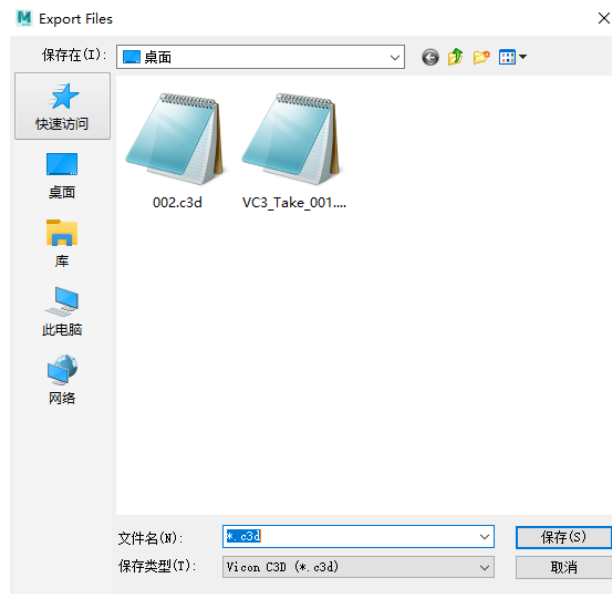
在这里有两种方法: 第一种是, 在录的时候直接通过 motive 实时发送给 motion builder, 在录的时候就做好重定向, 这种方法比较简单, 可以跳过 step4 至 step6。另一种就是录的时候没有接, 通过后期 motion builder 去手动的 retargeting 骨骼, 需要经历 step4 至 step6。

在 Motion Builder 里导入刚导出的文件, 一般是 fbx 格式的。然后选中需要的骨骼。点右下角的属性窗口, 在 Common 里选择 Scaling, 可以调整指定骨骼的长度。Motion builder 的使用方法在此不再赘述。



## Step 5 调整后的点云导出

File —> Motion File Export, 将所有的骨骼数据选中, 导出 Vicon c3d 文件, 命名。



## Step 6 安装 EZC3D

这种 c3d 文件不是很好打开, 需要下载 EZC3D, 下载和使用的方法参考他们的 github 网站即可。简单使用便可以读出来。

<https://github.com/pyomeca/ezc3d#python-3>

## EZC3D



EZC3D is an easy to use reader, modifier and writer for C3D format files. It is written in C++ with proper binders for Python and MATLAB/Octave scripting languages.

C3D (<http://c3d.org>) is a format specifically designed to store biomechanics data. Hence many biomechanics softwares can produce C3D files in order to share data. However, there is a lack in the biomechanics community of an easy to use, free and open source library to read, modify and write them as needed when it gets to the data analysis. There was at some point the BTK project (<https://github.com/Biomechanical-ToolKit/BTKCore>) that was targeting this goal, but the project is now obsolete.

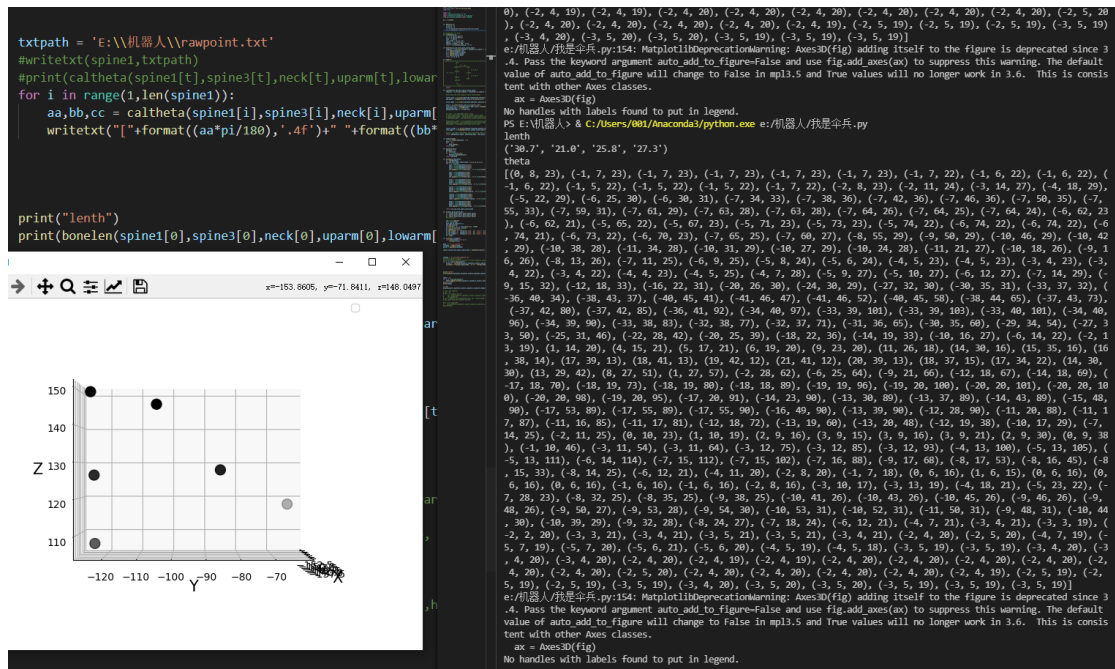
EZC3D addresses these issues. It offers a comprehensive and light API to read and write C3D files. The source code is written in C++ allowing to be compiled and used by higher level languages thanks to SWIG (<http://www.swig.org/>). Still, proper interface are written on top of the SWIG binder in order to facilitate the experience of the coders in their respective languages.

You can get the online version of the paper for EZC3D here: [JOSS 10.21105/joss.02911](https://doi.org/10.21105/joss.02911)

So, without further ado, let's begin C3Ding!

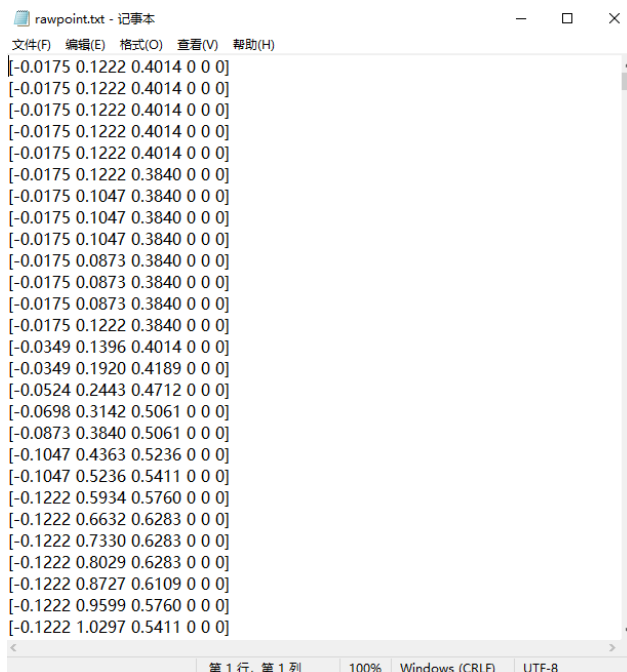






## Step 8 写入 txt 文件

写入 txt 文件“rawpoint”，是每一帧里电机的转角（弧度制），对应程序输出的数值（角度制）。

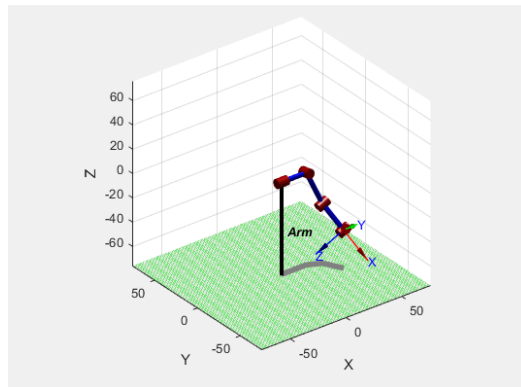


## Step 9 仿真

因为 python 里画图不是很好看，所以这里选择用 matlab 来输出视频看计算结果进行仿真。具体教程可以看我在知乎写的。

安装和调试 Matlab 工具箱 rvc-tools ——> <https://zhuanlan.zhihu.com/p/374909361>

Matlab 机器人工具箱基础代码解释 ——> <https://zhuanlan.zhihu.com/p/379449370>



matlab 程序一并上传了，只要把 txt 文件写入的矩阵，直接复制到 q 的这里赋值，直接点运行就可以看到视频。具体视频可以参考 demo。

```
q = [  
  
    [-0.0175  0.1222  0.4014  0  0  0]  
    [-0.0175  0.1222  0.4014  0  0  0]  
    [-0.0175  0.1222  0.4014  0  0  0]  
    [-0.0175  0.1222  0.4014  0  0  0]  
    [-0.0175  0.1222  0.4014  0  0  0]  
    [-0.0175  0.1222  0.3840  0  0  0]  
    [-0.0175  0.1047  0.3840  0  0  0]  
    [-0.0175  0.1047  0.3840  0  0  0]  
    [-0.0175  0.1047  0.3840  0  0  0]  
    [-0.0175  0.0873  0.3840  0  0  0]  
    [-0.0175  0.0873  0.3840  0  0  0]  
    [-0.0175  0.0873  0.3840  0  0  0]  
    [-0.0175  0.1222  0.3840  0  0  0]  
    [-0.0349  0.1396  0.4014  0  0  0]  
    [-0.0349  0.1920  0.4189  0  0  0]  
    [-0.0524  0.2443  0.4712  0  0  0]  
    [-0.0698  0.3142  0.5061  0  0  0]  
    [-0.0873  0.3840  0.5061  0  0  0]  
    [-0.1047  0.4363  0.5236  0  0  0]  
    [-0.1047  0.5236  0.5411  0  0  0]
```

## Step 10 整合数据、连接串口、驱动（之后的工作，便于端到端）

安装串口驱动程序，可以在 python 里面通过串口向机器人发送数据。这个程序目前没有，因为要根据机器人的驱动板去写。参考之前机器人头的板子，应该流程类似。之后可以写在一个程序里，实现端到端操控。

```
import serial, os  
import serial.tools.list_ports  
  
# 安装驱动软件，请安装串口驱动 gpl install serial  
# 测试驱动程序，请安装串口驱动 gpl install serial  
# 然后输入输入电机的角度，电机的角度不要超过限定值  
# 数据以逗号分隔，在数据中/电机测试角度  
  
# 串口驱动  
_link = "/dev/tty.usbserial-1430" # "/dev"  
_link = None  
_isopen = False  
_port_list = list()  
  
# 驱动串口驱动，通过串口驱动，应该安装和快速显示串口  
def checkserialist():  
    global _port_list  
    global _link  
    res = False  
    # 驱动串口驱动  
    _port_list = list(serial.tools.list_ports.comports())  
    if len(_port_list) <= 0:  
        temp = "没有串口/检查连接"  
        showMessage(temp)  
    else:  
        # 显示串口列表  
        for i in range(len(_port_list)):  
            temp = "No." + str(i) + " - " + str(_port_list[i])  
            showMessage(temp)  
        msg = "从列表选择连接串口(串口"+str(len(_port_list)-1)+")"  
        showMessage(msg)  
        # 从列表中选择，通过串口  
        selectis = list(input())  
        if selectis[0] in _port_list:  
            _link = _port_list[selectis]  
            _link = _port_list[0]  
            # 设置串口  
            _isopen = True  
        else:  
            showMessage("选择错误")  
    return res  
  
# 显示内容  
def showMessage(msg):  
    print(msg)  
  
# 打开串口  
def setserial(frc=115200):
```