3D人体动作跟踪与对比

使用说明书

目录

[1. 引言 2](#_Toc14202714)

[1.1编写目的 2](#_Toc14202715)

[1.2项目背景 2](#_Toc14202716)

[1.3定义 2](#_Toc14202717)

[1.4参考资料 2](#_Toc14202718)

[2. 软件概述 2](#_Toc14202719)

[2.1功能 3](#_Toc14202720)

[2.1.1动作捕捉 3](#_Toc14202721)

[2.1.2姿态对比 3](#_Toc14202722)

[2.2性能 3](#_Toc14202723)

[2.2.1精确度 3](#_Toc14202724)

[2.2.2时间性能 3](#_Toc14202725)

[2.2.3灵活性 3](#_Toc14202726)

[3. 运行环境 3](#_Toc14202727)

[3.1硬件环境 3](#_Toc14202728)

[3.2软件环境 4](#_Toc14202729)

[4. 使用说明 4](#_Toc14202730)

[4.1安装和初始化 4](#_Toc14202731)

[4.2输入 4](#_Toc14202732)

[4.3输出 4](#_Toc14202733)

[4.3.1动作捕捉 4](#_Toc14202734)

[4.3.2姿态对比 4](#_Toc14202735)

[4.4求助查询 5](#_Toc14202736)

[5. 运行说明 5](#_Toc14202737)

# 引言

## 1.1编写目的

本文档的编写是为了用户更加清晰方便的使用本产品

## 1.2项目背景

本项目为数字媒体实训项目，目前，无论在游戏行业还是影视行业中，人物模型的动作制作仍是一项繁琐、昂贵的工作，或需要雇佣专业的人员的进行制作，或需要购买昂贵的专业设备进行人物的动作捕捉，这大大提高了制作人物模型动作的门槛，为此，我们决定制作一个低门槛的动作捕捉与动作文件输出的应用。以此为引，我们还考虑了在此基础上可拓展的功能：两个视频中的动作比较和纹理替换。

## 1.3定义

Vnect：RGB单摄像头3D重建算法，深度学习算法，可以从视频中提取人体3D坐标

Openpose：RGB单摄像头2D重建算法，深度学习算法，可以从视频中提取人体2D坐标

Baseline：深度学习算法，可以将人体的2D坐标转换为3D坐标

OpenMMD：Openpose与baseline的结合，可以从视频中提取人体2D坐标

Fbx：目前市场上主流的3D模型文件格式之一，可以在Unity与UE4引擎中直接使用

Vmd：动作序列文件格式，主要在MMD中使用

## 1.4参考资料

[1] Fbxsdk：<http://help.autodesk.com/view/FBX/2019/ENU/>

[2] 用python解析VMD格式读取：<https://www.jianshu.com/p/ae312fb53fc3?tdsourcetag=s_pcqq_aiomsg>

[3] OpenMMD：<https://github.com/peterljq/OpenMMD>

[4] Vnect: <http://gvv.mpi-inf.mpg.de/projects/VNect/>

# 软件概述

## 2.1功能

### 2.1.1动作捕捉

用户可以输入视频，选择使用的3D重构算法，选择输出的文件格式，然后处理得到相应的动作模型文件。

### 2.1.2姿态对比

用户可以输入两个视频，选择使用的3D重构算法，之后程序会对两个视频中的人体姿态进行对比，然后反映在opengl火柴人动画上，最后给出准确率。

## 2.2性能

### 2.2.1精确度

本产品使用现有的3D重建算法，精确度不高，处理之后的模型动作文件可能与原动作略有差别，或有些抖动。

### 2.2.2时间性能

本产品功能所使用的时间大部分是在Vnevt和OpenMMD上，OpenMMD算法使用NVIDIA GeForce GTX 1060显卡，处理1080p的视频，fps在5左右，Vnect算法使用NVIDIA GeForce GTX 860m显卡，处理1080p视频，fps在6左右。

Vnect算法速度较高于OpenMMD

### 2.2.3灵活性

本产品灵活性较差，仅有windows版本，需要用户自己配置相应的环境

# 运行环境

## 3.1硬件环境

无具体要求，建议用户使用NVIDIA显卡

## 3.2软件环境

Windows7/8/10/XP

若用户使用OpenMMD算法，需要用户自己配置python3环境，包括以下库：

h5py

opencv-python

matplotlib==3.0.0

tensorflow==1.13.1

pyqt5

# 使用说明

## 4.1安装和初始化

无需安装，直接打开3DPoseEstimation.exe即可

## 4.2输入

视频文件：视频分辨率，长度，不受限制，视频格式为MP4/avi

若输入多个视频，视频名称不要相同，视频名称不要带有中文

输出文件夹：选择输出的文件夹

## 4.3输出

### 4.3.1动作捕捉

根据用户的选择而产生不同的模型动作文件

若使用Vnect算法，则在用户选择的文件夹下生成与输入的视频名称相同的.json文件和.FBX/.vmd文件

若使用OpenMMD算法，则在用户选择的文件夹下生成与输入的视频名称相同文件夹，此文件夹下有OpenPose\_result文件夹，OpenPose\_result\_3d文件夹，openpose.avi输入视频经过openpose显示骨架的视频，输入的视频名称相同的.json文件和.FBX/.vmd文件

### 4.3.2姿态对比

无输出，对比的结果会显示在界面中

## 4.4求助查询

在启动界面，动作捕捉界面，姿态对比界面，右上角都有“？”标志，点击即可查看帮助

# 运行说明

启动3DPoseEstimation.exe后进入启动界面：



①：点击后进入帮助界面

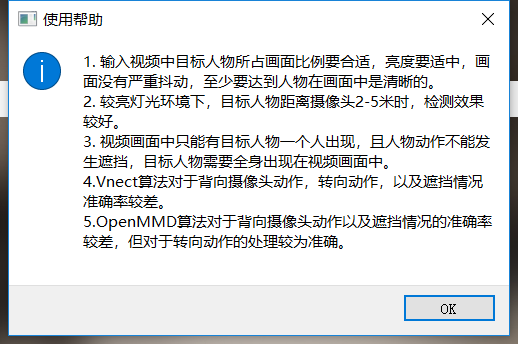
②：点击后最小化窗口

③：点击后退出程序

④：点击后进入动作捕捉界面

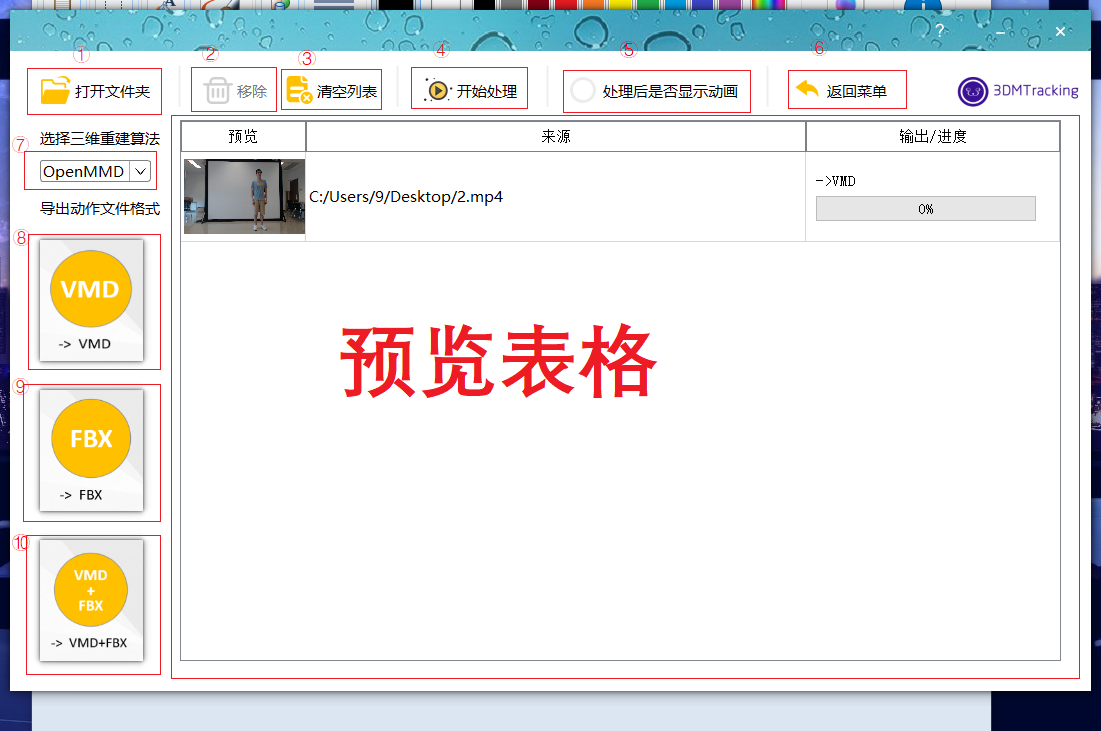
⑤：点击后进入姿态对比界面

帮助界面如下：



点击ok按钮或右上角x可以关闭本界面

动作捕捉界面：



①：点击后可以选择输出文件夹

②：点击预览表格中的视频，再点击②可以移除选中视频

③：点击后清空预览列表

④：选择完要处理的视频和输出路径后点击，开始处理

⑤：若选中，则在处理完视频后会播放opengl火柴人动画

⑥：点击返回启动界面

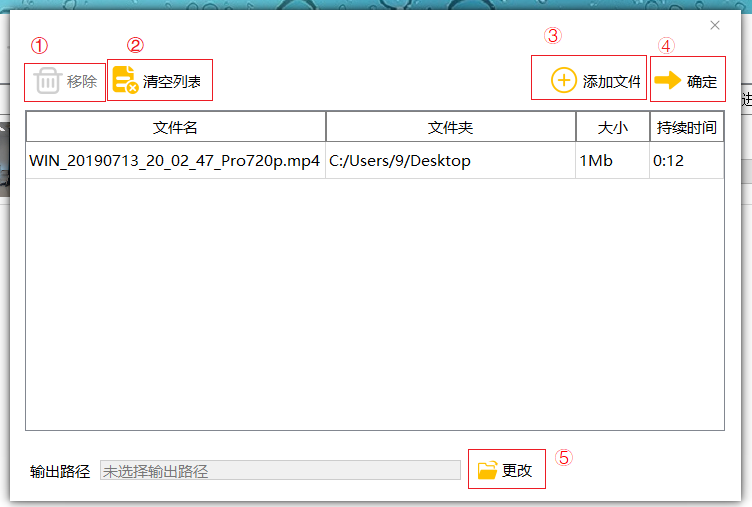
⑦：选择使用的算法

⑧：点击后进入视频选择界面，该界面下选择的视频将会导出成fbx格式

⑨：点击后进入视频选择界面，该界面下选择的视频将会导出成vmd格式

⑩：点击后进入视频选择界面，该界面下选择的视频将会导出成fbx格式和vmd格式

视频选择界面如下：



①：移除选中的视频

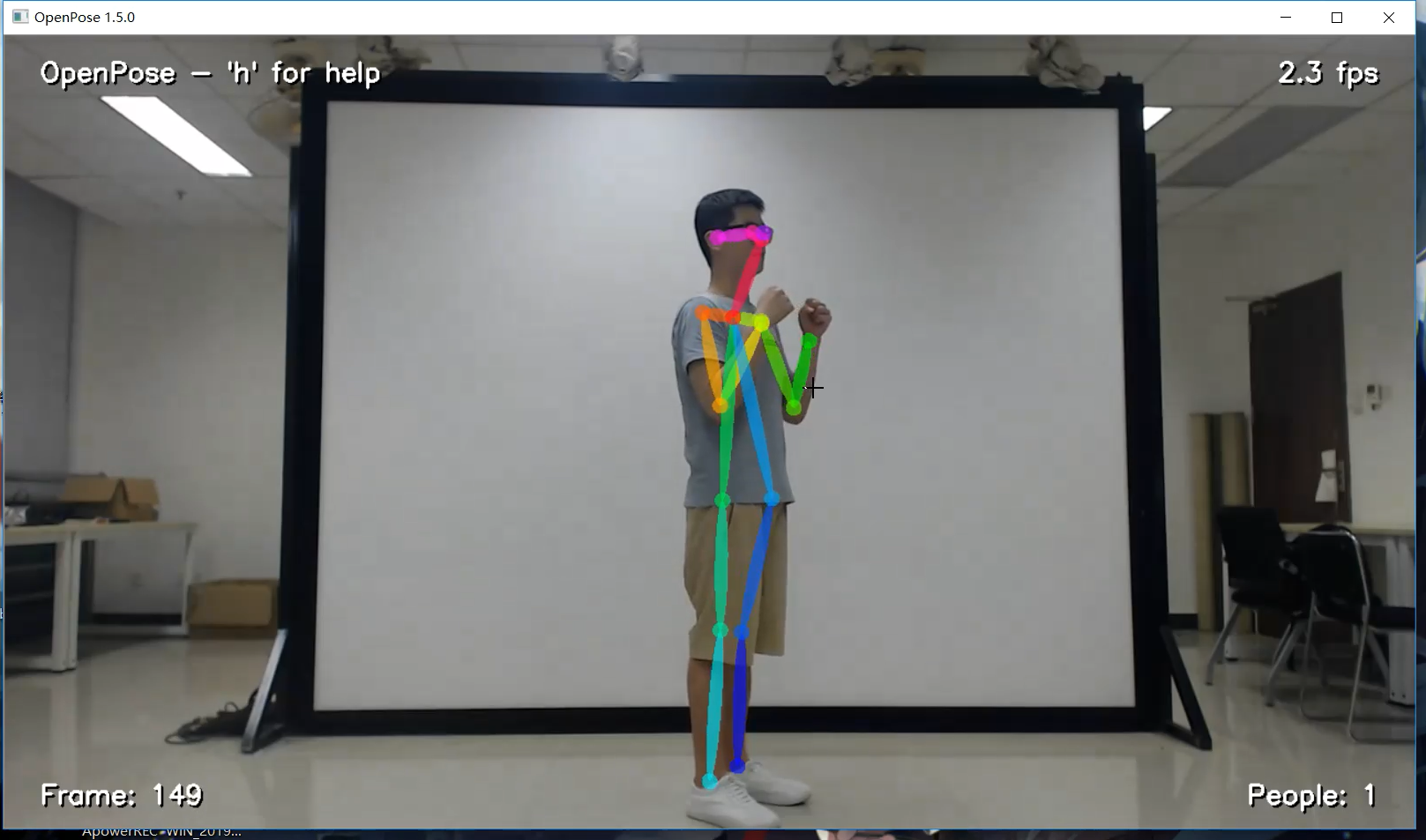
②：清空视频列表

③：点击后选择视频文件

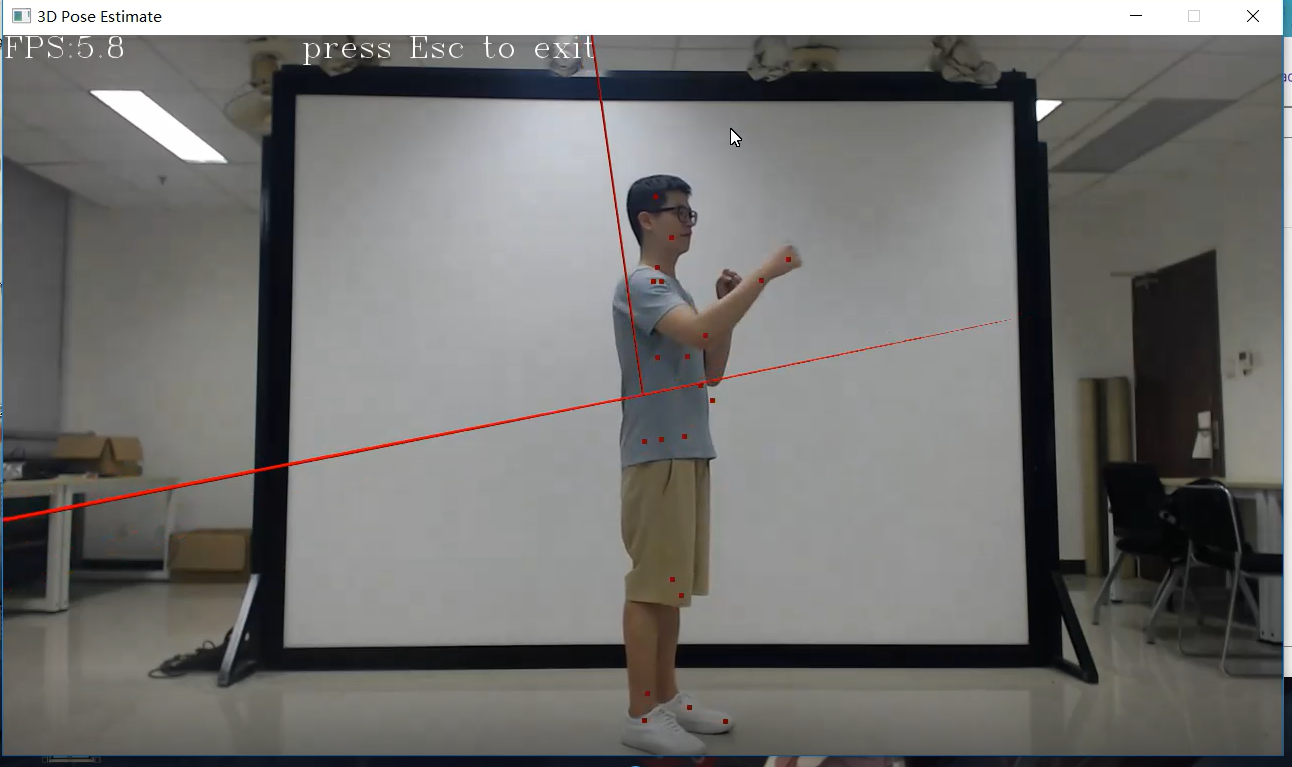
④：选择视频完毕关闭本窗口

⑤：在本界面下也可以修改输出路径，与动作捕捉界面中①相同

OpenMMD算法处理过程中会有所处理的视频显示：



Vnect算法处理视频过程中也会显示视频：



姿态对比界面：



①：选择使用的算法

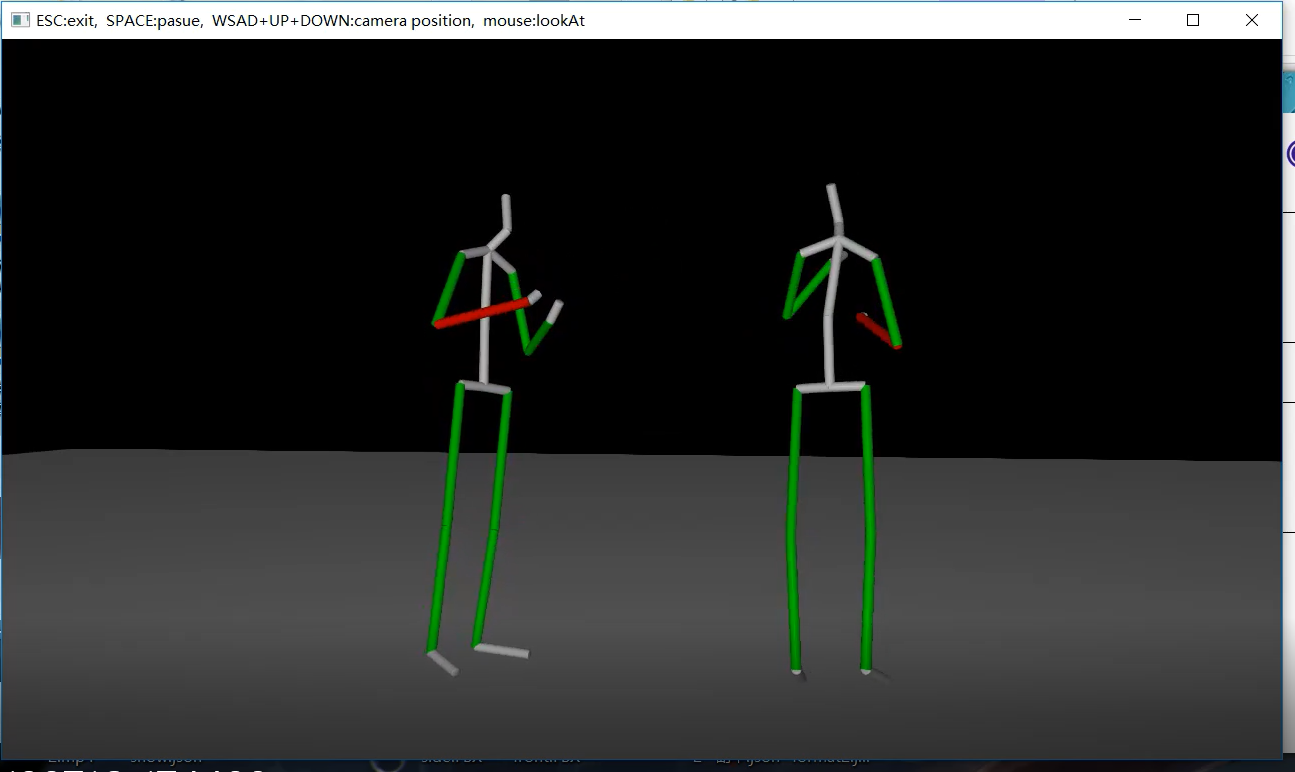
②：返回启动界面

③：选择视频

④：选择第二个视频

⑤：开始使用相应算法处理两个视频，过程中会显示相应算法处理视频的界面，处理完成后会有opengl火柴人动画播放，最后会有准确度显示在界面中

opengl火柴人动画界面如下：



按下空格可以暂停和开始播放，按下wsad↑↓可以控制相机前后左右上下移动，移动鼠标可以改变视角方向

注：在使用OpenMMD算法时，Openpose界面视频播放完后，会有一段时间的空白期，这个期间时baseline算法在跑，但是进度没有显示在界面中，此为目前已知bug，将在后续版本中更新。