東南大學

毕业设计(论文)报告

题　目: 人体姿态与动作分析技术

学　　号:	08117125
姓　　名:	杜煜
学　　院:	自动化学院
专　　业:	机器人工程
指导教师:	夏思宇
起止日期:	2021.1-2021.6

东南大学毕业（设计）论文独创性声明

本人声明所呈交的毕业（设计）论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

论文作者签名：_____ 日期：_____ 年 _____ 月 _____ 日

东南大学毕业（设计）论文使用授权声明

东南大学有权保留本人所送交毕业（设计）论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权东南大学教务处办理。

论文作者签名：_____ 导师签名：_____

日期：_____ 年 _____ 月 _____ 日 日期：_____ 年 ____ 月 ____ 日

摘　要

动作分析评价作为专业运动训练中的一部分非常关键，但是由于需要专业知识、专人指导，成本非常高，对于大众运动健身等很难进行普及。研究计算机视觉技术进行运动分析可以大幅降低成本，使其得到普及，以避免居家健身或是康复运动中，由于动作不正确而导致的受伤。此外，大群体的青少年的体质筛查或是线上体育教学产生的大量动作视频等需要专业人士参与的大批量工作也可以使用运动分析系统进行处理。

本文应用计算机视觉技术，设计了一种基于单目 RGB 视频的动作评价方法，以动作视频作为输入，使用 OpenPose 作为人体骨架提取器，使用 ST-GCN 识别视频中动作所属类别，基于关节角度定义了一组人体姿态特征向量用以度量动作相似度，提出了一种基于帧间差分法的关键帧检测与基于 K-means 关键帧聚类的方法进行视频动作分割，实现对重复动作进行计数，并给出每一次动作的评价得分，以及评估重复动作完成的一致性。推导了一种弱刚性变换进行人体姿态配准，使得输入动作与模版动作的差异可以被直观地观察。

关键词：动作评价，姿态估计，动作分割，姿态配准

ABSTRACT

As a part of professional sports training, action analysis and evaluation is very important. However, due to the need for professional knowledge and professional guidance, the cost of action analysis and evaluation is very high, so it is difficult to popularize professional action analysis and evaluation. Research on computer vision technology for action analysis can greatly reduce costs and make it popular, so as to avoid injuries caused by incorrect movements in home fitness or rehabilitation exercises. In addition, large-scale work that requires professional participation, such as physical screening of large groups of young people or a large number of action videos generated by online physical education teaching, can also be processed by the action evaluation system.

This paper uses computer vision technology to design an action evaluation method based on monocular RGB video, taking action video as input, taking OpenPose as human skeleton extractor, using ST-GCN to identify the category of the action in the video. A set of human pose feature vectors based on key joint angles is defined to measure the similarity of actions. A key frame detection based on temporal difference method and a method based on K-means key frame clustering are proposed for video action segmentation to realize the counting of repetitive actions, compute the evaluation score of each action, and evaluate the consistency of the completion of repeated actions. A weak rigid transformation is derived for human pose registration, so that the difference between input actions and template actions can be observed intuitively.

KEY WORDS: Action evaluation, Pose estimation, Action Localization, Pose registration

目　　录

摘　　要 **I**

ABSTRACT **II**

目　　录 **IV**

第一章　绪论 **1**

1.1　课题背景及意义 1

1.2　研究现状 1

1.3　本文研究内容 3

第二章　人体姿态估计与动作识别 **5**

2.1　引言 5

2.2　OpenPose API 5

2.2.1　多人体过滤 5

2.2.2　关节骨架序列修补与平滑 6

2.3　ST-GCN API 7

2.4　本章小结 8

第三章　动作分割 **9**

3.1　引言 9

3.2　人体姿态特征向量定义与计算 9

3.3　静止关键帧提取 11

3.4　关键帧动作特征聚类和筛选配对 14

3.5　本章小结 17

第四章　动作评价 **18**

4.1　引言 18

4.2　图像动作评价 18

4.2.1　姿态配准 18

4.2.2　评价指标 19

4.3　视频动作评价 21

4.3.1 视频动态规整 21
4.3.2 评价指标 22
4.3.3 可视化输出结果说明 24
4.4 本章小结 25

第五章 实验结果与分析 **26**
5.1 关键帧提取 26
5.2 动作分割 27
5.3 视频预处理优化 28
5.4 系统运行时分析 29
5.5 本章小结 30

第六章 总结与展望 **31**
6.1 工作总结 31
6.2 工作展望 31
6.2.1 系统不足与改进方向 31
6.2.2 工程应用的完善方向 32

参考文献 **32**

致 谢 **36**

第一章　绪论

1.1　课题背景及意义

计算机视觉如今在生活中的应用无处不在，从刷脸支付到车站人脸身份核验。同时，如今的应用更要求计算机视觉能够无感地处理大批量人群数据，比如车站的流量监测，以及新冠疫情下的流动人群体温检测。

而动作分析评估作为专业运动训练中的一部分非常关键，但是由于需要专业知识、专人指导，成本非常高，对于大众运动健身等很难进行普及。研究计算机视觉技术进行运动分析可以大幅降低成本，使其得到普及，以避免居家健身或是康复运动中，由于动作不正确而导致的受伤。此外，现在在线体育教学等课程越发普及，大量学生需要上传每日锻炼视频与考试视频等，使用动作分析系统可以帮助老师快速处理学生上传视频，也可以让学生在线实时得到训练计划完成情况评估。

除了在专业动作分析系统中的应用，本课题中使用到的多种技术皆具有很大的社会效益。譬如动作识别技术，可用于智能安防、智能监控等场景，比如能够在加油站自动识别打电话的人并予以警告。再如动作相似度比较技术，在当前老龄化的大环境下，可用于老人跌倒检测等。视频动作分割则是视频理解中的热门研究领域，可以应用于视频网站的视频关键词检索，影片剧情高潮标识等。

早期的运动分析技术多使用 Kinect 等 RGB-D 深度相机进行人体骨架关节的提取，部署成本高，市场保有量低，仅能在专业场地进行使用。随着近几年，基于深度学习的人体姿态估计研究成果大量涌现，诸如 OpenPose[1]、AlphaPose[2]、HRNet[3] 等网络模型，使用单目 RGB 图像即可准确地获得 2D 人体骨架关节，这就有望将运动分析系统带到单目相机中。

通过当前的实验可以得出，选用合适的深度模型，使用配备 8G 内存及 4G 显存的 Nvidia 显卡的 PC 就可满足运行需要，而随着移动终端使用的独立 NPU 单元日益强大，以及大量深度学习框架对移动端 ARM 芯片的逐步支持，本系统将有很大空间迁移到移动设备上。使用智能手机、平板电脑等作为数据采集和计算单元，可作为大众居家健身的 AI 教练，其应用前景非常广阔。

1.2　研究现状

人体姿态估计研究概述

自深度学习方法被广泛采用以来，一系列旧时十分棘手的图像处理问题有了迅速的进

展。Lecun 等人[4] 提出卷积神经网络之后，它被迅速地运用到各类图像处理问题当中，并取得了丰厚的成绩，例如在 Wei 等人[5] 的工作中能看到卷积神经网络在人体姿态估计任务中的应用。何恺明等人[6] 提出的残差网络使得深度学习模型的深度与精度有了显著提升，基于此技术的研究成果，Newell 等人[7] 提出的 Stacked Hourglass 网络在单人体姿态估计任务的精度上有了质的飞跃，而它的基本模块也被后来关于人体姿态估计任务的研究广泛采用。多人体姿态估计任务更加棘手，目前大致分为两种解决方案，自顶向下和自底而上。自顶向下方案联合现有人体检测器与单人体姿态估计模型完成多人体姿态估计任务，Fang 等人[2] 的工作代表了当时的先进水平。自底向上方案中，Cao 等人[1] 提出了关节亲和场的方案，同时回归关节点置信图和肢体的朝向，解决了自顶向下方案中模型运行时间与场景中人体数量成正比的问题。Wang 等人[3] 并行连接高分辨率与低分辨率网络，在前向传播过程中始终保持着高分辨率，使得预测结果的精度有了进一步提高。

时序动作识别研究概述

动作识别模型可分为两个流派，一是使用 RGB 图像加密集光流作为输入，二是使用人体关节骨架序列作为输入。无论哪一流派的算法均可被分为基于手工设计特征和基于深度学习两种。早期的工作基本都基于手工设计特征，而至深度学习取得巨大成功以来，大量的研究方法转向深度学习。

基于图像及光流的方法首先需要计算光流场，经典的光流计算方法为 Lucas-Kanade 算法[8]，但其由于计算量过大，完全无法满足实时性的要求，随着 GPU 的大量应用，使用 CUDA 编译的 OpenCV 可以快速的提取光流，而 FlowNet[9] 则最早使用 CNN 解决了光流估计的问题。一般使用双流网络处理 RGB 帧与光流场来进行动作识别[10, 11]。C3D 网络[12] 则采用 3D 卷积直接处理 RGB 帧序列。TSN[13] 则设计了一个能对长时序建模的网络，通过时序稀疏采样和视频级的监督，能高效地使用整个动作视频进行学习。

基于骨架的方法相比之下，具有对光照和场景变化更为鲁棒的优点，同时由于现在已经出现了许多精确的深度采集设备以及人体姿态估计模型，获取 3D 或是 2D 的人体骨架的成本大幅降低，于是大量的基于关节骨架的动作识别算法相继被提出，Liu 等人[14] 使用 RNN 实现了端到端的动作识别模型，Yan 等人[15] 首次将图神经网络（GCN）应用在基于骨架的动作识别任务中，Liu 等人[16] 则进一步使用多尺度聚合提高了模型长时序建模的能力。训练动作识别模型常用的两个大型数据集为：$NTU\ RGB+D$ 120[17] 与 $Kinectics$[18]，$NTU\ RGB+D$ 提供了多种模态的动作数据，涵盖 120 种日常动作，数据模态包括 RGB-D 图像与 3D 骨架。$Kinectics$ 由 DeepMind 发布，其数据仅提供 YouTube 链接以及动作类型和视频中动作出现的时间段，Yan 等人在 ST-GCN 的仓库[19] 中则提供了从 $Kinectics$ 中提取出的骨架数据。

时序动作分割研究概述

如果使用图像任务类比动作视频理解任务，那么时序动作识别对应图像分类任务，识别视频中的动作类别对应识别图像中的物体类别，是一个分类任务。而时序动作分割任务则类似图像目标检测任务，不仅需要识别出动作的类别，还需要回归出动作开始与结束的边缘。当前动作分割的研究方法也如目标检测一样，分为两大类，一类是 two-stage 的方法，将分割提案与动作分类分为两步进行处理[20, 21]，另一类则是 one-stage 的方法，直接检测长视频中的动作实例[22-25]。RepNet[26] 则使用自相似矩阵实现了重复动作的自分割与计数。常用的动作分割数据集有 $THUMOS\ 14$[27] 与 $ActivityNet$[28]。

动作视频评价研究概述

当前动作视频普遍使用动态时间规整算法（DTW）进行时序对齐。Ji 等人[29] 的工作利用 DTW 算法来计算样本动作和测试动作之间的动作相似性程度。Jiang 等人[30] 的工作则进一步利用快速 DTW 方法自动确定所匹配动作片段并计算对齐后两个序列的距离。吴齐云等人[31] 则基于 Dijkstra 算法的思想对 DTW 算法进行改进，并使用 K-means 算法对动作进行评估。而人体骨架提取则多使用 Microsoft Kinect 作为采样设备，直接利用 Kinect SDK 提取和跟踪骨架关节点，Alexiadis 等人[32] 利用 Kinect 骨架跟踪技术设计表演者在演出时候舞蹈动作的评估系统。Wang 等人[33] 的工作则提出了时空融合模型（Spatial-Temporal Relation Module）对视频序列进行姿态估计，解决了快速运动过程中人体姿态估计与运动追踪的难题，并使用 SVM 对 Bad Pose 进行分类。

1.3 本文研究内容

围绕人体运动这一问题，本文主要研究人体动作视频的评价方法，其中涉及到几项技术的应用与方法的探索。能够识别视频中动作所属类别，对重复动作进行计数，并且将重复动作进行分割，给出每一分割的评价得分，以及评估重复动作完成的一致性（离散度）。并且在结果中，能够可视化地观察输入动作与模版动作在关节层面上的差异，帮助使用者更清晰地了解到自己与标准动作的差异之处，并进行改进。系统处理流程如图1.1所示。

系统可分为几个核心模块，分别为：骨架关键点提取、动作识别、动作视频分割、动作评价。

第二章中将介绍骨架关键点提取和动作识别，通过部署 OpenPose 人体姿态估计模型，使用单目相机进行 2D 人体关节骨架的提取，设计了一种插补算法以对 OpenPose 的关节估计缺失问题进行处理，添加了骨架序列平滑以抑止帧间骨架估计的抖动问题。而后将利用获得的人体骨架序列作为输入，部署了 ST-GCN 模型进行视频动作识别，识别的动作类别将用以自动加载对应动作评价配置文件。

第三章中将提出一种动作视频分割的方法。其中将使用第二章中获得的人体关节骨架计算人体核心关节角以组成姿态特征向量，并使用帧间差分法进行关键帧提取，并基于上述姿态特征向量对关键帧进行动作聚类与配对，最终达成动作分割的目标。

第四章中将介绍动作评价方法。首先将介绍图像动作评价方法，提出一种弱刚性变换方法进行姿态配准，并定义了一种基于位置偏差和关节角偏差的动作评价指标。其后将图像动作评价方法推广至视频动作评价，其中需要对视频进行分割后，对分割视频使用 DTW 算法进行时序规整，再对规整后的视频帧应用图像动作评价方法，最后提出一系列有关视频动作评价的评价指标。

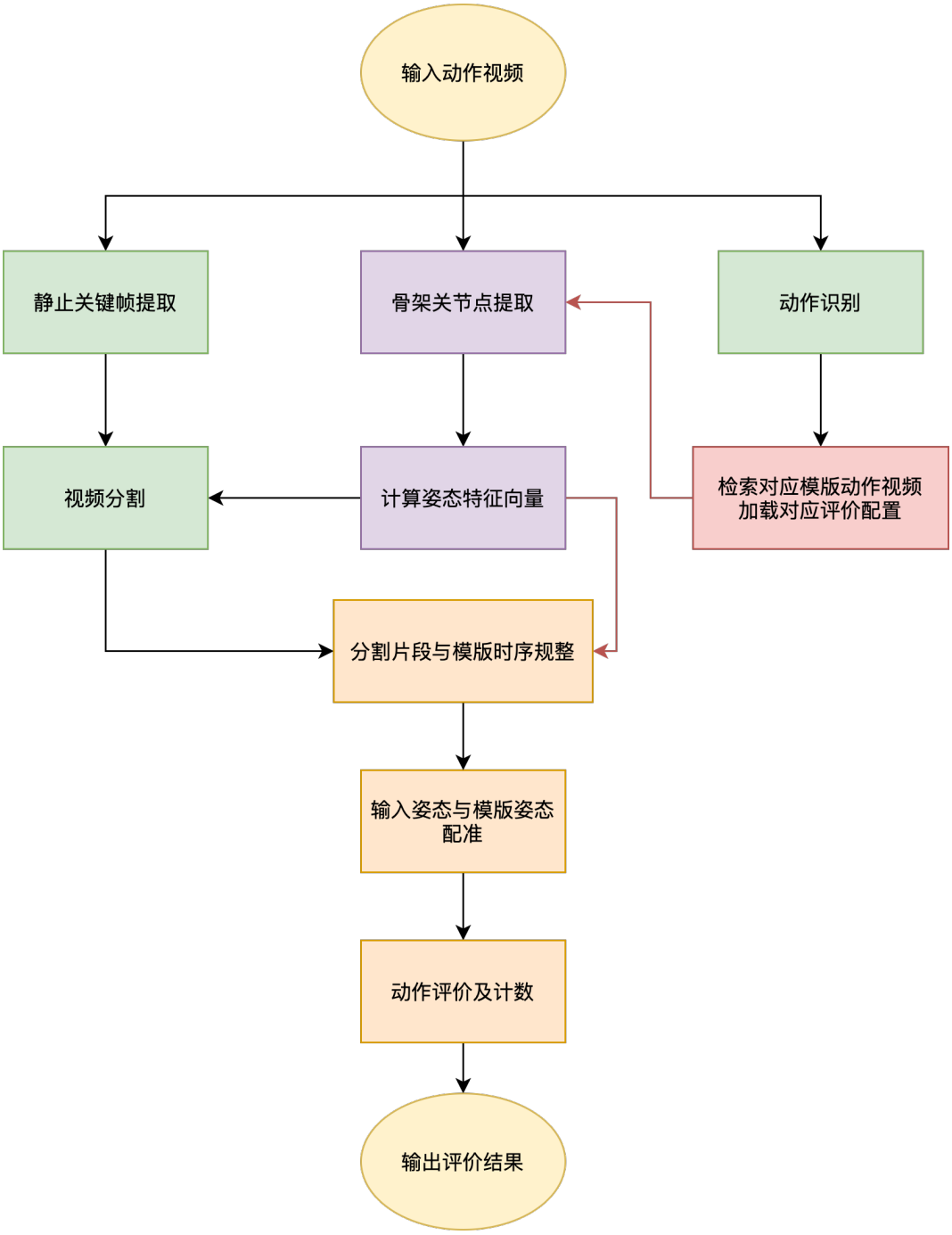


图 1.1: 系统框图

第二章　人体姿态估计与动作识别

2.1　引言

随着近几年计算机视觉的飞速发展，与人体相关的图像算法与深度学习模型已能够提取大量高层次的特征。对于动作评价任务而言，所需特征可分为两部分，一是知道动作是什么样，二是知道是什么动作，该如何评价。对于第一个部分，通过人体姿态估计模型获得的人体关节骨架可以对动作是什么样进行描述，人体姿态估计代表性的工作 CPM[34]、Stacked Hourglass[7]、OpenPose[1]、AlphaPose[2, 35, 36]、HR-Net[3] 等；对于第二个部分，则可以通过动作识别模型获得动作所属类别后，进行进一步的处理，动作识别模型可分为两个流派，一是使用 RGB 图像加密集光流作为输入，二是使用人体关节骨架序列作为输入。这里将选用较为成熟的 OpenPose 作为人体姿态估计模型，鉴于系统数据流中已出现关节骨架，就没有必要再浪费计算资源去获得昂贵的密集光流，选用 ST-GCN 这一图卷积网络作为动作识别模型。下面将分节介绍 OpenPose 与 ST-GCN 的部署、接口开发以及数据处理过程。

2.2　OpenPose API

这里使用 Cao 等人[37] 开源的 OpenPose 源码进行编译，配置 OpenPose Python API，在此基础上按照本设计需求进一步开发 OpenPose API，其提供视频估计与图像估计两个接口。图像估计接口输入图片矩阵，输出估计得到的人体关节矩阵；视频估计接口输入视频序列矩阵，输出估计得到的人体关节矩阵列表。

2.2.1　多人体过滤

动作视频中可能出现多个人，也有可能由于姿态估计模型的错误预测，在某些物体上产生”虚假“的人，对于动作评价系统而言，需要提取拍摄主体人物，并将多余的估计结果进行滤除。可行的过滤方法有两种：一是按照尺度排序，由于在动作视频的拍摄中，被拍摄的主体一定占据画面的主要部分，其骨架的尺度理应是最大的，分别计算估计所得骨架的长宽跨度，按此跨度排序，取最大的便是拍摄主体；二是从姿态估计模型中提取各个关节的置信度值，计算估计所得实例的各关节置信度之和，作为拍摄主体，由于尺度最大，遮挡最少，往往能够拥有最高的置信度，按此置信度和排序，取最大也能够实现主体的提取。

2.2.2 关节骨架序列修补与平滑

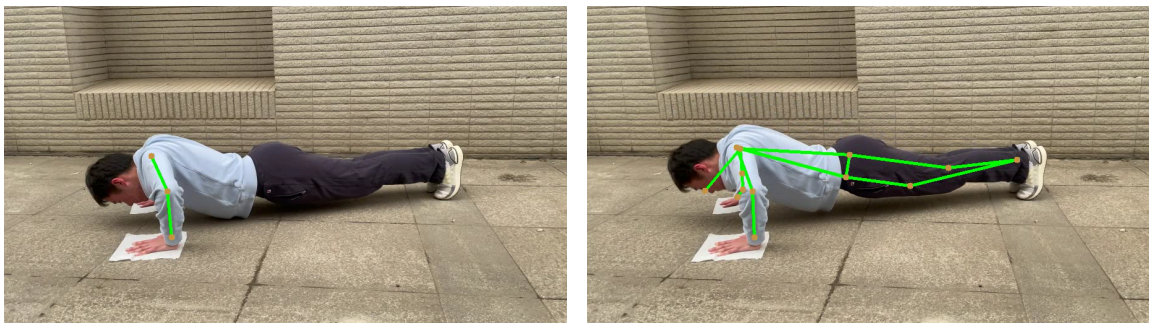
使用单目 RGB 进行人体姿态估计，性能会受制于物体遮挡与身体各部位的自遮挡等情况。在设计预期的场景中，当运动动作比较复杂时，常常会存在自遮挡，进而导致关节估计的缺失，正常来说由于遮挡导致的估计值缺失无伤大雅，但由于后续需要利用估计所得的关节骨架进一步计算姿态特征向量，大面积的关节估计值缺失（在某一特定动作下，多个关节反复缺失是非常常见的），就会对姿态的分类以及评价产生较大的影响。图2.1展示了一个较为极端的插补前后的差异。

考虑到，系统处理的是视频序列，人体在视频中的动作必然是连续可导的，因此可以利用当前帧前后的有效估计值对当前帧缺失的关节进行插补。一个简单的实现算法是使用使用线性插值，对所有缺失的估计使用有估计值的时刻进行插值，视频头部和尾部缺失的估计值将直接使用该关节第一个和最后一个有效估计值。

具体的算法如下，考虑关节 J_i，在帧 F_k 缺失估计值，其前一个和后一个有有效估计值的帧分别为 F_b 和 F_e，那么 J_i 在 F_k 处的插补估计值 (x_{ik}, y_{ik}) 如公式2.1所示。

$$\begin{bmatrix} x_{ik} \\ y_{ik} \end{bmatrix} = \begin{bmatrix} x_{ib} \\ y_{ib} \end{bmatrix} + \frac{i-b}{e-b} \times \begin{bmatrix} x_{ie} - x_{ib} \\ y_{ie} - y_{ib} \end{bmatrix}, \quad (b < i < e) \tag{2.1}$$

若关节 J_i，在帧 F_k 缺失估计值，并且无前向或后向有效估计值，则其直接使用后向或前向第一个有效估计值作为插补值；若关节 J_i 在整个序列中都无有效估计值，则不做任何插补。考虑到多次向前及向后遍历查找有效关节位置的计算效率十分低，代码实现中使用向量化的矩阵运算以提高计算效率，通过沿时间轴正向及逆向遍历个遍历一次，构建出起始图（Begin Map）、截止图（End Map）、计数图（Count Map），时间复杂度与空间复杂度均 $O(n)$。



(a) 插补前　　(b) 插补后

图 2.1: 关节插补前后对比

由于 Openpose 是针对单帧图像进行姿态估计，估计结果在时序上会发生抖动，包含大量噪声，会对后续处理中的姿态特征向量计算、特征聚类以及视频时序规整产生影响，因

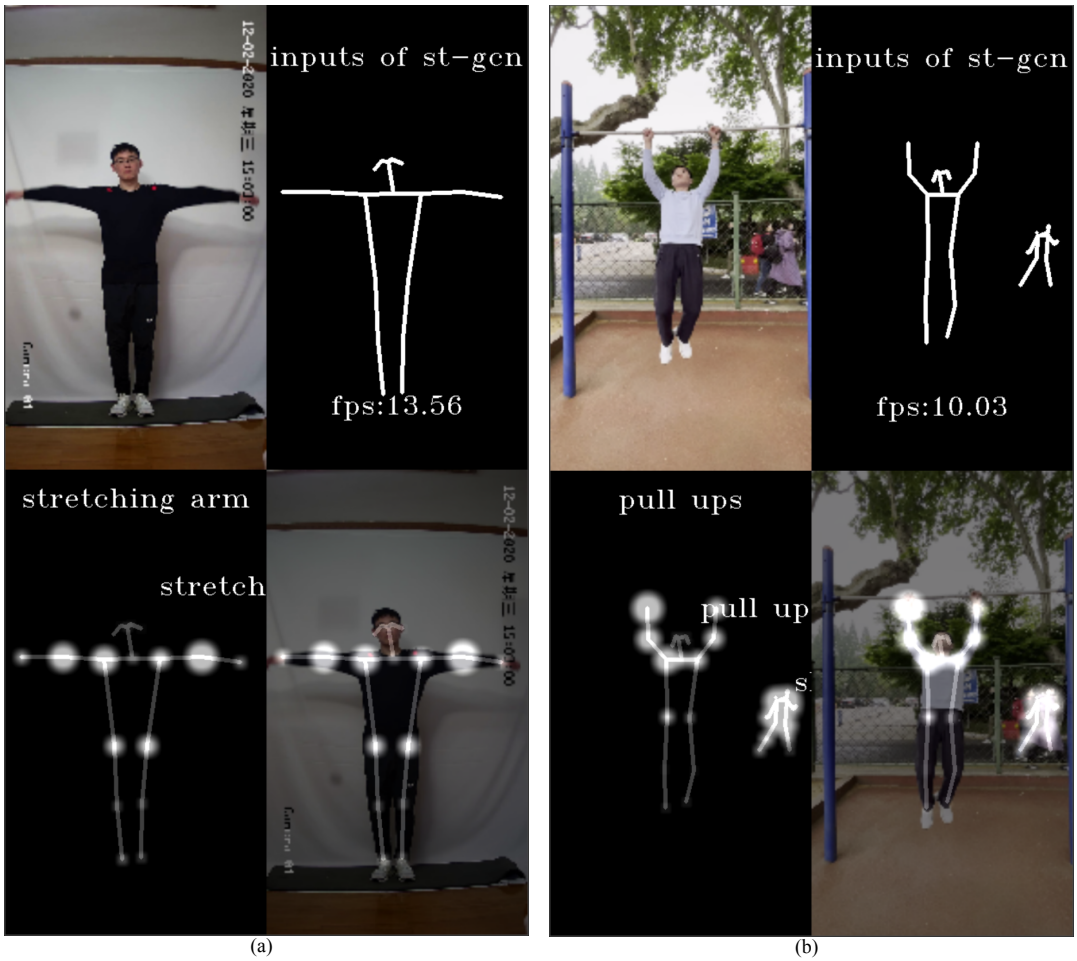
此需要对序列进行平滑，每一个关节点坐标在时序上都是一条曲线，因此可使用曲线平滑的方法，这里使用二次指数平滑的算法，计算方法如式2.2所示，其中 x_i 为测得第 i 个实际数值，s_i 为第 i 个平滑后的数值，t_i 为第 i 个平滑后的斜率值，α 控制历史数值和历史斜率对当前平滑数值的影响权重，β 控制历史斜率对当前平滑斜率的影响权重。

$$\begin{aligned} s_i &= \alpha x_i + (1-\alpha)\left(s_{i-1} + t_{i-1}\right) \\ t_i &= \beta\left(s_i - s_{i-1}\right) + (1-\beta) t_{i-1} \end{aligned} \tag{2.2}$$

2.3 ST-GCN API

这里使用 Yan 等人[15] 开源的 ST-GCN 源码与文档进行模型部署，并开发调用接口。动作识别接口输入关节矩阵列表，输出动作类别。

引入 ST-GCN 进行动作识别，目的是为了识别输入视频所属类别，从而自动检索对应的模版视频，以及自动加载所属动作的评价配置文件，关于评价配置的内容将在第3.4节和第四章中详细说明。图2.2为 ST-GCN Demo 在输入动作视频后，输出的识别结果。



(a) (b)

图 2.2: ST-GCN Demo 运行结果

2.4 本章小结

本章通过分析选取了 OpenPose 与 ST-GCN 两个深度学习模型，并开发了 API 分别完成人体姿态估计与动作识别两个任务。其中针对 OpenPose 输出的多个人体识别结果进行了主要人物提取，以及对估计关节点缺失和骨架抖动两点问题进行了改善。

第三章　动作分割

3.1　引言

在上一章中，通过部署 Openpose 模型与 ST-GCN 模型，已经获得了人体动作的高层次特征。本章中将对这些高层次特征进行进一步处理加工，以提取动作评价更核心关键的特征。具体来说，就是要就获得的关节骨架序列，进一步进行计算得到姿态特征向量，以及检测序列中每一周期动作的起始帧与结束帧的位置，以完成动作分割。

图3.1说明了动作视频分割的处理流程，本章将就除骨架关节点提取之外的部分进行详细阐述。

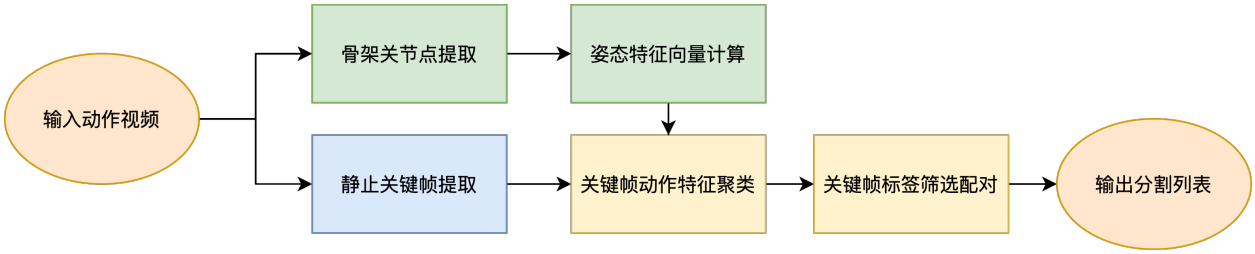
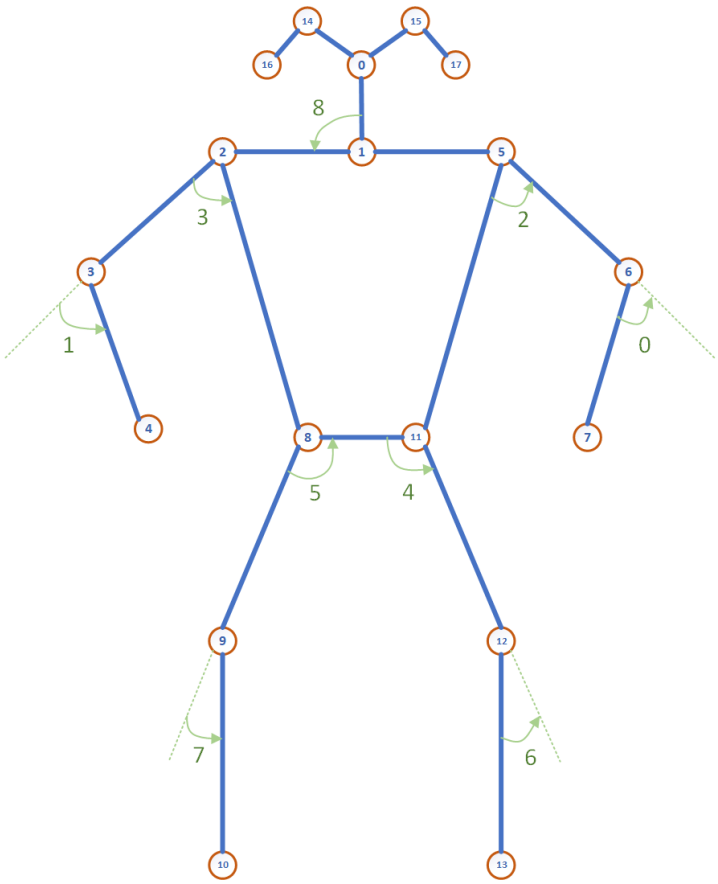


图 3.1: 动作分割流程框图

3.2　人体姿态特征向量定义与计算

骨架关节点是很好的一种降维抽象的人体姿态（二维、三维）空间描述方法，但是动作评价理应是位置及尺度不敏感的，换句话说，就是人体在图像（空间）中的位置及远近（大小）都不会影响他们的动作特征，只要他们做相同的动作，得到的评价就应当是一致的。这就要求算法应当将骨架关节点中的空间特征进一步去除，一个直观的想法是使用关节点计算关节角度，角度是位置不敏感的特征。于是这里定义姿态特征向量，由核心关节角组成，关节角由相连两向量夹角定义，向量的起点和终点均为关节点，逆时针方向为正，单位为弧度（rad），取值范围为 $(-\pi, \pi)$，以 $0°$ 为中心值，并且取值范围达到 2π，可以获得更稳定的关节角，避免角度越界产生的突变。

所谓核心关节角，即在运动过程中会大幅变化的关节夹角，像是躯干上的几个关节点所组成的关节角在运动过程中几乎不会变化，作为特征便不合适。这里最初定义的姿态特征向量由九个有向关节角组成，如图3.2所示，分别为：左肘、右肘、左肩、右键、左胯、右胯、左膝、右膝、颈部，方向为逆时针。受到[38] 的启发，发现人体的五个末端关节（头部、左手腕、右手腕、左脚踝、右脚踝）的相对位置关系，以及手腕与肩部、脚踝与胯部的角度关系也对姿态的相似度有着较大的影响，后续将这些特征也加入到特征向量中。



nose:	0
neck:	1
right_shoulder:	2
right_elbow:	3
right_wrist:	4
left_shoulder:	5
left_elbow:	6
left_wrist:	7
right_hip:	8
right_knee:	9
right_ankle:	10
left_hip:	11
left_knee:	12
left_ankle:	13
right_eye:	14
left_eye:	15
right_ear:	16
left_ear:	17

图 3.2: 骨架关节点与有向关节角定义，其中棕色圆圈为关节点，内部数字对应右侧关节名称；绿色弧线箭头与数字表示有向关节角定义的位置与顺序，方向为逆时针沿箭头方向。

姿态特征向量的计算的核心在于关节角度的计算，下面将介绍关节角度的计算方法，首先定义，$\vec{a} = J_i(x,y) - J_j(x,y)$，$\vec{b} = J_k(x,y) - J_l(x,y)$，其中 $i,j,k,l \in [0, N_J)$，N_J 为骨架中关节点的数量。角度的标量数值 θ_0，使用余弦定理可得：

$$\theta_0 = arccos(\frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|}) \tag{3.1}$$

取值范围为 $(0, \pi)$，角度方向则利用矢量叉积的正负进行判断，首先有叉积的坐标表达形式：

$$\vec{a} \times \vec{b} = |\vec{a}||\vec{b}|sin\theta = x_1 y_2 - x_2 y_1 \tag{3.2}$$

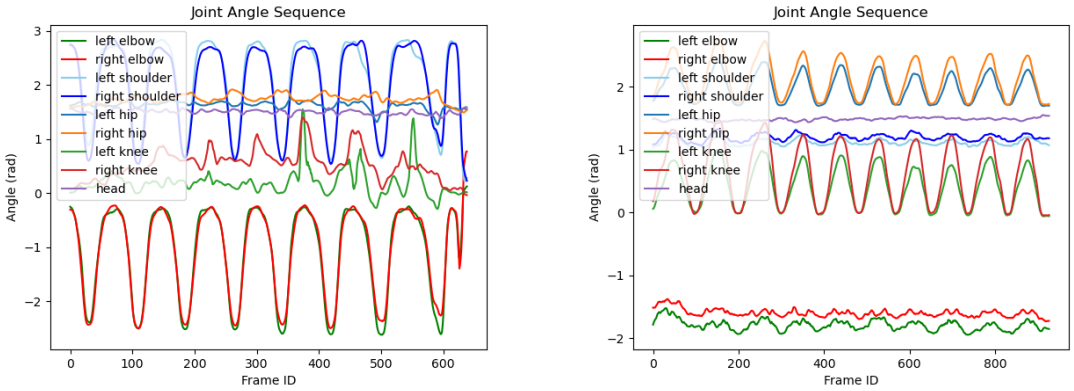
一般来说，上步由余弦定理计算出的角度大小在 $(0, \pi)$ 之间，在右手系中，$\vec{a} \times \vec{b} > 0$ 表示有向角度在 $(0, \pi)$ 之间即角度是逆时针旋转的；反之，有向角度则在 $(-\pi, 0)$ 之间，角度是顺时针旋转的，但是由于图像坐标系为左手系，这一结论则是相反的，于是有：

$$\begin{cases} anticlockwise, & \vec{a} \times \vec{b} < 0 \\ clockwise, & \vec{a} \times \vec{b} \geq 0 \end{cases} \tag{3.3}$$

由上述讨论可得最终角度：

$$\theta = \begin{cases} \theta_0, & \vec{a} \times \vec{b} < 0 \\ -\theta_0, & \vec{a} \times \vec{b} \geq 0 \end{cases} \tag{3.4}$$

由于这里计算的是关节角度序列，因此可以如第二章中对关节骨架序列处理过程一样，对角度序列进行插补与平滑，不过这里的角度插补则是进行角度异常检测，由于先前已经对关节骨架序列进行过插补，所以角度序列将不会有缺失，但是由于关节估计的抖动以及2D 姿态中可能存在的一些奇异位置，角度的计算可能会发生突变，这与常规认知的关节连续转动不符，因此，当检测到角度突变时，需对突变进行抑止。这里使用类似惯性的方法，设置当当前帧角度 θ_n 与上一帧角度 θ_{n-1} 差值 e_n 与上一组差值 e_{n-1} 的差超过 90° 时，替换当前帧角度为 $\theta_n = \theta_{n-1} + e_{n-1}$，及替换当前差值 $e_n = e_{n-1}$。在完成异常检测，得到关节角度序列后，使用 Savitzky-Golay 滤波器再一次进行平滑，设置窗口宽度为 7，多项式阶数为 1。以定义的 9 个角度为核心关节角作为姿态特征向量，记为 **A**，则得到的姿态特征向量序列数据形状（Shape）为 $(N_F \times 9)$，N_F 为视频帧数。图3.3(a)为经过平滑的引体向上关节角变化曲线，图3.3(b)为经过平滑的深蹲关节角变化曲线。



(a) 引体向上　　(b) 深蹲

图 3.3: 关节角变化曲线

3.3 静止关键帧提取

此处的关键帧指的是运动过程中，动作达到最大变化幅度的那一帧，包括起始动作。大多数的动作评价都会关注运动极限位置的情况，例如坐位体前屈关注手指向前达到的最远距离，或是引体向上关注下巴是否过杆，因此提取这一位置对于动作评价有很大的意义。

由于人的运动必然符合物理定律，改变运动方向必然存在一个减速到零，然后反向加速的过程，在这之中就会存在一个静止位置，就也就是运动到最极限的位置，这也就是为什么这里的关键帧被称为静止关键帧。同时由于摄像帧率可以根据人体动作快慢的不同来进行调整或者更换更高速的设备，因而这一静止位置可以稳定出现在很多连续帧中，为提取这一位置提供了很大的空间。

既然提到了静止位置检测，自然而然可以联想到运动检测，而运动检测的一种常用方

法便是帧间差分法。帧间差分法又分为两帧法和三帧法，其中两帧法直接计算相邻两帧的像素差值图，这一方法对运动物体的轮廓勾勒较为粗糙，而三帧法则计算相邻三帧中的两组相邻帧差值图并进行与运算，这样便得到了两组差值图的交集，得到了更为细腻的运动物体轮廓。

这里使用三帧法进行静止关键帧的提取，首先的步骤是计算视频帧间差分向量。差分向量具体的计算过程如下：

1. 保存当前帧，与当前帧的前两帧，分别计算相邻两帧的图像的差的绝对值，得到两张差值图 D_1, D_2；
2. 将 D_1, D_2 使用图像按位与运算，合成一张差值图 D，并将 D 转化为灰度图 D_{gray}；
3. 对 D_{gray} 进行二值化，并进行归一化，即得到 $0, 1$ 表示的差值图 D_{bool}；
4. 对 D_{bool} 求和，得到差分值；
5. 对每相邻三帧计算差分值，合并得到差分向量，差分向量的长度为视频帧数减 2；

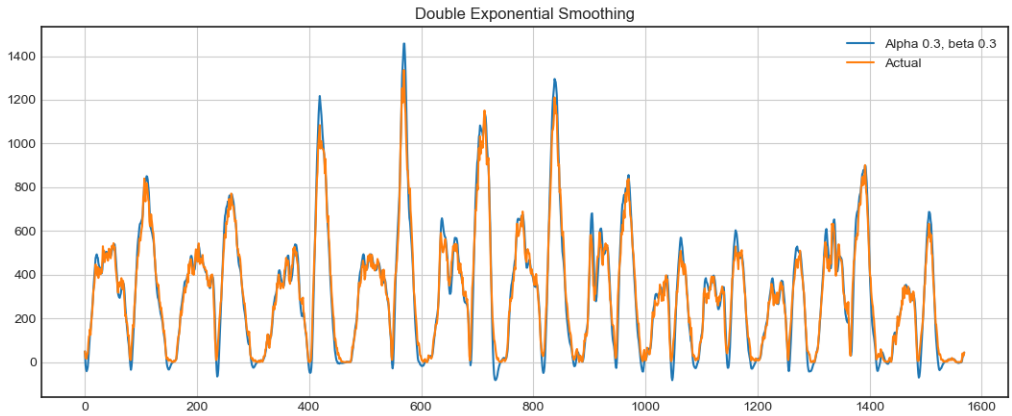


图 3.4: 多次深蹲动作差分向量。橙黄色曲线：原始差分值曲线；蓝色曲线：使用二次指数平滑法平滑后的差分值曲线（设置 $\alpha = 0.3$，$\beta = 3$）。

在绘制得到差分向量之后，可以发现其中非常显著的噪声，这对关键帧的提取非常不利，在这里也使用式2.2所定义的二次指数平滑法对差分向量进行平滑，平滑前后的曲线如图3.4所示。接着便需要从中得到静止关键帧的位置（索引值）了。所谓静止关键帧，顾名思义，便是帧间差分值很小的的帧，这一“很小”也是一个相对概念，视频差分值的绝对数值大小受到的影响包括：视频图像的分辨率、图中主体运动人物的尺度大小以及人物的运动幅度大小。图3.5(a) 中的蓝色垂直线条表示的便是差分向量的大小。而设计算法要获取的便是这些蓝色线条所勾勒出的波谷位置的索引值。但此时存在几个问题，首先当前并不知道视频动作存在几个相位（一个周期动作包含多少个不同的极限动作）、以及动作出现的次数；此

外，人体在运动过程中可能存在小幅度的速度抖动，因而产生许多局部极小值，也就是很多假“波谷”；最后便是，期望得到的静止关键帧是相互远离的，即得到不同动作相位及动作周期的关键帧，表现在差分向量图中，便是被大的波峰分割开的关键帧。

为处理这一任务，首先使用比例值来定义静止的阈值，这里设定整个视频的差分向量中的最大值的 0.2 倍作为静止帧的阈值，这样便在差分向量图中分割出了几块波谷的区域。此时，如果波谷的差分变化趋势为单调递减后单调递增，则其驻点便是动作的极限位置，即关键帧；如果波谷中仍存在一些小的起伏，对实际的动作位置也是影响不大，取波谷中的最小值同样可以较为精确的反应动作极限位置。利用差分向量提取关键帧的算法流程如算法1所述。

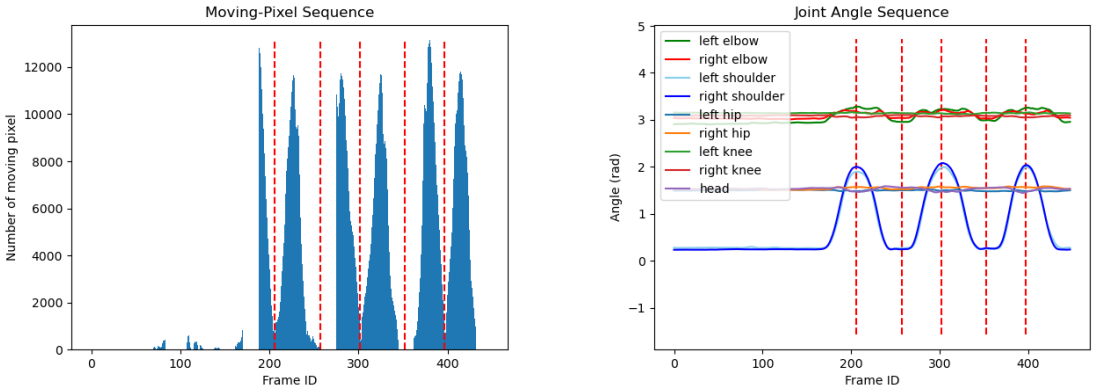
算法 1: 差分向量提取关键帧

Data: 静止状态符 is_static，静止起始位置符 $Begin$，静止结束位置符 End，当前索引差分值 δ_i，差分向量维数 N，静止阈值 η，差分向量 Δ；

Result: 静止关键帧索引 $idxs$；

$is\_static \leftarrow True$;
$Begin \leftarrow -1$;
$idxs \leftarrow list()$;
while $i \leq N$ **do**
  if $\delta_i < \eta$ **&&** $is\_static = False$ **then**
    $Begin \leftarrow i$;
    $is\_static \leftarrow True$;
  else if $\delta_i > \eta$ **&&** $is\_static = True$ **then**
    $End \leftarrow i$;
    $is\_static \leftarrow False$;
    if $Begin! = -1$ **then**
      $sliced \leftarrow \Delta[Begin : End]$;
      $arg = argsort(sliced)$;
      $idxs.append(arg + Begin)$;
      $Begin \leftarrow -1$;
    end
  end
end

以上是早期的算法步骤，而面对后续的视频分割需求时，需要给出每一个分割的起始和结束位置，因此需要将视频起始和结束的静止动作记录到关键帧集合中。于是添加视频顺序与倒序第一个差分值大于 0.5 倍阈值的帧定为动作开始帧与动作停止帧，这样可以避免视频开始后和结束前存在长时间无动作的部分被计入分割当中。上述关键帧提取算法只需要对差分向量进行一次遍历即可提取出所有关键帧，算法时间复杂度为 $O(n)$。图3.5(a)中的红线便是对这一差分向量提取的关键帧位置（不包含视频起始和结束动作）。图3.5(b)则是关键帧在关节角度曲线中所处位置的特征，可以清晰地看出关键帧均位于关节角度的极



(a) 帧间差分向量图　　(b) 关节角度曲线

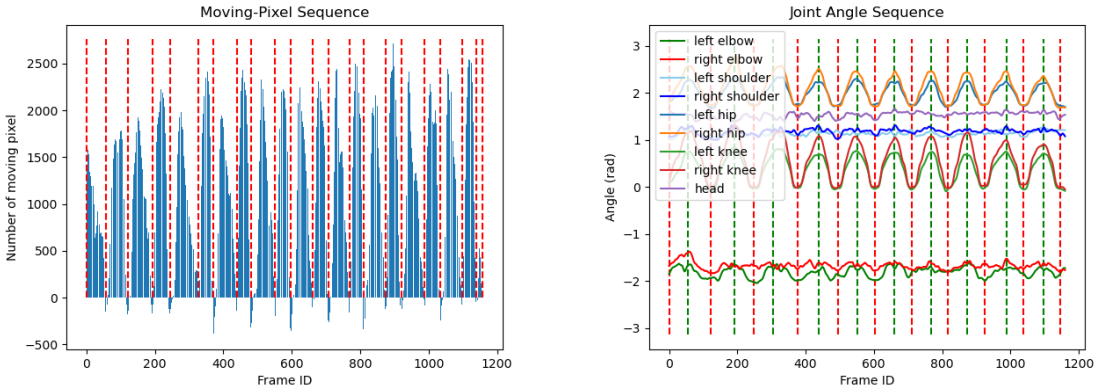
图 3.5: 关键帧提取结果，红色虚线所在横坐标为关键帧索引值。

值位置，即动作极限位置，符合设计预期。当动作变得更为复杂的时候，还是会存在多个距离非常近的重复的关键帧，这一问题将在第3.4节中进一步解决。

3.4 关键帧动作特征聚类和筛选配对

第3.3节中，已经完成了静止关键帧的提取，再进一步思考可以发现，这些动作极限位置天然地形成了每一个动作元的边界，类似于[25] 中使用的自底向上（bottom-up）方法，先给出动作边界的提案，然后对动作边界进行组合。那么如何将这些边界提案组合成包含起始结束的动作分割呢？这里便存在几个问题需要解决。一是，使用关键帧提取到的是动作中的极限位置，而这些极限位置的动作各有不同，一个标准动作周期中，可能包含多个动作元，需要识别出哪些是动作周期的边界动作，哪些又是动作周期的中间动作，即对动作进行分类；二是，关键帧中可能包含多个距离非常近的重复的关键帧，如图3.6(a)中，部分波谷出现两个关键帧，或是在某些“伪波谷”出现了关键帧，便会导致一个真实边界可能对应多个候选边界，需要评估每个边界的置信度，或者说是每个分割的置信度，然后筛选出高置信度的边界组合。

上述第一个问题，有两种可行的解决方式，一种是使用动作相似度比较的方法，在第四章中进行动作评价时也会继续使用这一概念，通过与给定模版动作关键帧进行特征距离计算，以特征距离大小来对关键帧进行分类，获得起始帧、结束帧、中间帧的类别标签，这一方法需要提前给出模版视频的关键帧动作特征。第二种则是使用聚类的方法，在一个重复多次的动作视频中，通过关键帧提取，可以获得多次周期变化的关键帧特征，使用聚类可以无监督的直接给不同位置的关键帧打上不同的标签，这一方法则需要知道动作中存在几个动作元。在这一系统中，最终选用的是第二种方法，由于不需要提前给出模版视频的关键帧动作特征，相比之下，添加新的动作类型进入系统则会更为方便，这也便是相比



(a) 帧间差分向量图　　(b) 关节曲线图

图 3.6: 十次深蹲动作对应的帧间差分图及关节曲线图。(a) 中未删除误检关键帧；(b) 中已删除误检关键帧，并且同一颜色虚线表示聚类后同为一类的关键帧。

深度学习方法，这一传统方法的优势，即无需提供大量的标注数据，众所周知，标注数据的成本还是相当高的。

现在开始详细说明动作聚类的实现方法。首先是聚类的方法，使用常见的 K-Means++ 算法[39]；第二是聚类中心的个数，前面在第2.3节中，已经获得了动作的类型，而根据动作的类型，系统会自动加载不同的配置文件，在配置文件中，就可以给定动作的动作元个数，即聚类中心个数，例如常见的动作，如深蹲、引体向上等均只包含两个动作元，也就是设置聚类中心个数为 2，而像是波比跳等动作，则包含更多的动作元，只需要在动作配置文件中设定动作元个数即可。第三，则是聚类特征的定义，在第3.2节中，计算了动作的特征向量，这一向量具有平面内的平移和旋转不变性，可以很好的描述动作姿态的特征，按照式3.5计算姿态特征向量和聚类中心的欧式距离，便可以对聚类中心进行迭代，其中 $\mathbf{A}$ 为特征向量，$\mathbf{A}_c$ 为当前迭代的聚类中心。

$$d = ||\mathbf{A} - \mathbf{A}_c|| \tag{3.5}$$

上述第二个问题，则是利用聚类的结果进行解决。在聚类完成后，每个关键帧会获得各自的标签，如果没有出现意外，这些标签应该是按照某一顺序循环出现，但有时会出现多个连续相同的标签，一种情况，是当测试者某几次动作非常不规范时，其连续采集到的几个关键帧动作可能都非常相似而被归为一类。对于这种情况，算法将利用时序信息自动判断并修改连续关键帧的标签，方法如算法2所述，图3.7使用 PCA 主成分分析的方法将姿态特征向量降至 2 维，从而可视化了关键帧聚类的分布情况，在图3.7(b)可以看到有些红点越过了分类分界线，这便是算法自动修改的关键帧标签，这些关键帧往往在分界线附近，由不标准的动作产生，不进行标签修改则会丢失这一次动作的分割；还有一种情况则如图3.6(a)中的冗余关键帧的情况，这些误检的冗余关键帧多是在抖动或是运动过程中出现的，其与标

算法 2: 关键帧标签调整

Data: 检测到的关键帧数量 N，聚类中心数 N_c，第 i 个关键帧的标签 L_i，第 i 个关键帧特征到第 j 个聚类中心的距离 d_{ij}，第 i 个关键帧对应聚类中心到第 j 个聚类中心的距离 d_{L_ij}，标签调整阈值 $\eta \in (0.5, 1.0)$;

Result: 修改后的关键帧标签 L;

```
i ← 0;
while i ≤ N do
    if L_i = L_{i-1} then
        j ← 0;
        d ← inf;
        while j ≤ N_c do
            if d_ij ≤ η d_{L_i j} && d_ij ≤ d then
                L_i ← j;
                d ← d_ij;
            end
            j ← j + 1;
        end
    end
    i ← i + 1;
end
```

准的极限位置动作距离肯定较远。一种合理的置信度判定可以通过计算关键帧与聚类中心的特征距离，在出现连续相同的关键帧标签时，选取该距离最小的帧作为最终关键帧，该删除关键帧的步骤应在修改关键帧后进行。有一点需要说明是的，由于系统处理的都是连续运动，上一个动作的结束帧与下一个动作的开始帧没有明显的分隔，于是直接将此两帧选为一帧。图3.6(b)展示了在关节曲线图中，使用上述方法去除了误检关键帧后，得到的关键帧标签，不同颜色的竖虚线表示不同的标签，由于深蹲动作的动作元只有蹲下、站起两个，因而标签也是两种。

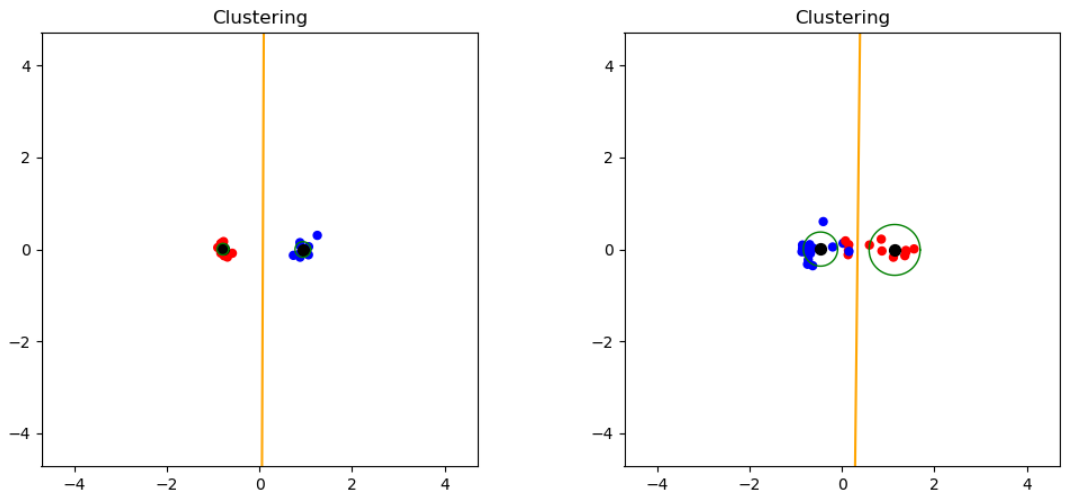
最终给出的动作分割是以列表的形式，对于一个 n 动作元的动作，其列表形式为:

$$[start, mid_i, end],\ i \in \{1, 2, ..., n-1\}$$

其元素为视频的帧索引值，由于聚类得到的标签是没有被赋予实际意义的，而动作从哪个动作元开始都是可以识别的，因此直接给定视频序列的第一个关键帧标签即为 $start$ 标签，end 与 $start$ 标签相同（周期动作的起始位置与结束位置相同），中间间隔的即为中间帧 mid_i。

综上所述，动作分割的流程可概述如下:

1. 帧间差分法提取关键帧;
2. 使用关键帧索引获取关键帧姿态特征向量，计算其与聚类中心的欧式距离作为距离度



(a) (b)

图 3.7: 经过 PCA 降维后的关键帧聚类可视化，其中橙色直线为聚类分界线，绿色圆半径为特征点到聚类中心的平均距离。

量；

3. K-means 聚类得到关键帧的姿态标签；
4. 连续相同标签的处理：首先结合时序信息判断是否需要修改关键帧类别标签，修改完成后仍然出现连续相同标签时，取特征距离聚类中心较小的关键帧；
5. 两个 $start$（end）标签所包围的区间即为一个动作周期，返回动作分割列表。

3.5 本章小结

本章首先定义和推导计算了人体运动核心关节角，并以此定义了人体姿态特征向量。随后使用帧间差分法进行了静止关键帧的提取，静止关键帧反映了运动中的极限位置，是每个动作元的边界位置。于是进一步的，使用静止关键帧对动作元进行分割，使用姿态特征向量的欧式距离作为度量，进行关键帧聚类，其中聚类中心个数通过动作识别获得的类别进行设定，得到各个动作元的起始与结束边界标签，并设计了一种标签自动修正及误检帧删除算法。最后，按照关键帧标签的顺序组合得到完整一次动作的分割。

第四章　动作评价

4.1　引言

动作评价是一个相对主观的任务，不像是图像质量评价等任务，已有比较完善的评价体系，动作评价这一任务，在我所调研的范围内，暂时没有找到一个比较通行的评价指标与性能指标。但鉴于此处使用与模版视频比较的方法进行动作评价，一种合理的思路是将动作评价任务转为相似度比较任务，而由相似度自然而然可以想到距离度量。先前在第3.4节中，已经利用特征向量的距离度量进行了关键帧动作聚类，这里的聚类换句话说，也就是按照相似度大小进行分类。

在本章中，将针对动作评价这一任务，提出更细致的相似度衡量与评价指标。首先在第4.2节中，将介绍单帧图像的动作评价方案，然后在第4.3节中将其推广至视频动作评价。

4.2　图像动作评价

图像动作评价将针对输入图像与模版图像的两个关节骨架进行研究，讨论基于关节骨架的评价指标设计。

4.2.1　姿态配准

在第3.2节中定义的姿态特征向量是一种具有平移旋转不变性的特征，但是人体关节点的相对空间位置还是对于动作评价有很大的意义。尤其是在硬件上使用的单目 RGB 相机只能实现 2D 的关节点估计，在计算关节角的时候会存在奇异的情况，比如在前平举动作时，肩部、肘部、腕部关节点几乎重合。这时使用仅使用关节角作为姿态特征不免发生很大的抖动和偏差，引入空间位置特征是对原有特征向量很好的一种补充和完善。

若需要使用关节点位置特征，一则必须消除其在图像中平移和旋转的问题，二则需要消除图像中人体远近以及个人身高臂展的差异。这两个问题可以通过对模版姿态进行空间变换，将其配准到输入姿态之上来解决。使用 OpenCV 中提供的仿射变换函数，只需要定义三个配准点，便可以非常简单地完成配准。但是仿射变换是非刚性的，虽然保持了骨架各肢体的长度比例不变，但是会导致人体的姿态发生形变，这便影响了关节角度的计算。刚性变换仅包含平移和旋转，可以保持肢体比例与关节角度均不变，但是无法处处理人体远近以及个人身高臂展的差异。

为此，这里提出一种弱刚性变换，其包含缩放、平移和旋转，定义缩放系数 a，旋转角

度 θ，平移向量 (p_x, p_y)，可以得到坐标变换的矩阵形式为式4.1，

$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} acos\theta & -asin\theta & p_x \\ asin\theta & acos\theta & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4.1}$$

其中 (x, y) 为变换前的点坐标，$(x*, y*)$ 为变换后的点坐标，有待求变换参数：a, θ, p_x, p_y，可知需要变换前后的两对对应点坐标来进行求解，设该对应点坐标为：(x_1, y_1)，(x_2, y_2)，(x_1^*, y_1^*)，(x_2^*, y_2^*)。解得上述变换参数如式4.2所示，为了表达简洁，令：$\Delta x = x_1 - x_2$，$\Delta y = y_1 - y_2$，$\Delta x^* = x_1^* - x_2^*$，$\Delta y^* = y_1^* - y_2^*$，

$$\begin{aligned} a &= \sqrt{\frac{\Delta x^{*2} + \Delta y^{*2}}{\Delta x^2 + \Delta y^2}} \\ cos\theta &= \frac{\Delta x \Delta x^* \pm \Delta y |\Delta y^*|}{a(\Delta x^2 + \Delta y^2)} \\ sin\theta &= \frac{-\Delta x^* \Delta y \pm \Delta x |\Delta y^*|}{a(\Delta x^2 + \Delta y^2)} \\ \begin{bmatrix} p_x \\ p_y \end{bmatrix} &= \begin{bmatrix} x_1^* \\ y_1^* \end{bmatrix} - \begin{bmatrix} acos\theta & -asin\theta \\ asin\theta & acos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \end{aligned} \tag{4.2}$$

其中 $sin\theta$, $cos\theta$ 中的 $\pm$ 依照平方和等于 1 约束，应取同号，而根据实验所得结果，应取 + 号。由上述推导说明可知，计算弱刚性变换矩阵需要两组对应关节点坐标，而这两组点应该能够反映其对应人体的尺度大小，能体现人体主要躯干的位置，并且在各种动作中应该是相对稳定的，一组比较好的候选组合是两肩的中点和两胯的中点，这两点的连线是人体的脊柱主干，能反映人体在图中的尺度，相比相对位置和角度大幅变化的手臂和腿部而言，其也非常稳定。图4.1为使用上述弱刚性变换配准结果。可以发现，在进行姿态配准后，带来的另外一个重要好处便是，可以直观地了解到系统输入的动作姿态与模版姿态的偏差所在，有助于使用者对自己的动作进行改进。

4.2.2 评价指标

在第4.2.1节中已经完成了姿态的配准，此时关节点的图像坐标已经消除了人体由于平移旋转以及远近等因素产生的偏差，可以用以进行动作的评价。此时，提出一种基于位置与角度偏差的动作评价指标，其中的位置偏差由式4.3定义，

$$err_{pos} = \frac{1}{D} \frac{1}{N_J} \sum_{i=1}^{N_J} ||\mathbf{J}_i - \mathbf{J}_{i\_ref}|| \tag{4.3}$$

式4.3中 $\mathbf{J}_i$ 和 $\mathbf{J}_{i\_ref}$ 分别为输入姿态和模版姿态中关节点 i 的坐标，N_J 则为参与评价的关节点个数，OpenPose 所使用的 COCO 数据集人体姿态使用 18 个关节点进行标注，头部包含鼻子、双眼、双耳五个关节点。由于动作评价并不涉及面部表情的变化，同时由于运动遮

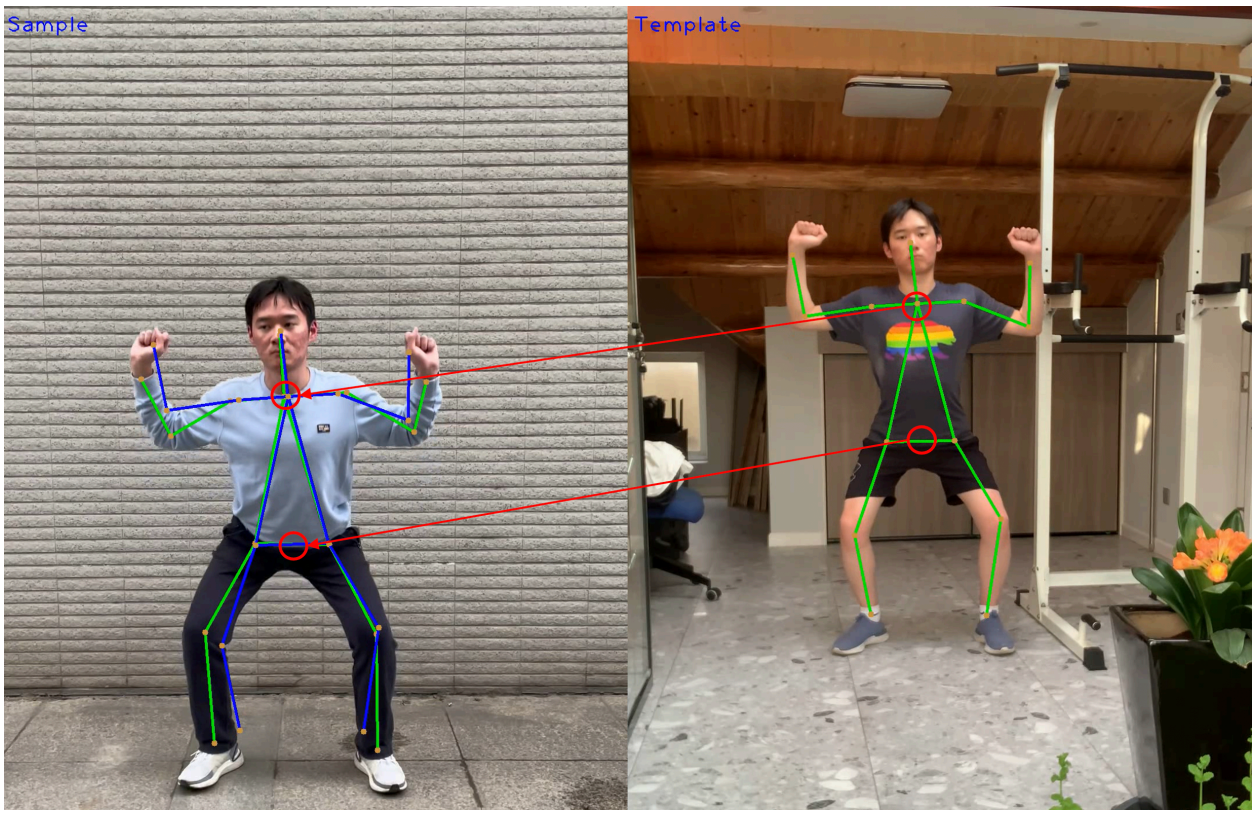


图 4.1: 姿态配准结果，左图中绿色骨架为图中人体姿态的估计结果，蓝色骨架为右图中人体骨架经变换配准后的结果，红圈标注的位置为两组配准点。

挡等原因，面部关节经常会缺失估计值，一般在进行动作评价时可以忽略面部关节点，仅保留鼻子来标识头部位置。式4.3中 D 则表示两肩中点到两胯中点的距离，用以对距离偏差进行正则化，由于人在图像中的尺度是未知的，而位置偏差应该是一个人体尺度的相对量，在第4.2.1节中已经说明两肩中点到两胯中点的距离，可以比较好的反应图中人体尺度，因而用以进行正则化。关节角偏差则由式4.4定义，

$$err_{angle} = \frac{1}{N_A}||\mathbf{A} - \mathbf{A}_{ref}|| \tag{4.4}$$

式4.3中 $\mathbf{A}$ 和 $\mathbf{A}_{ref}$ 分别为输入姿态和模版姿态中的关节角向量，即第3.2节中的定义的姿态特征向量，N_A 为姿态特征向量的维数。

最后则是相似度的计算，由于动作评价系统的输出是面向一般使用者的，这里需要将相似度的结果尽量直观化呈现，使用百分制对姿态进行打分，是使用者最普遍能接受的一种形式，于是在相似度计算时使用 e 指数将计算结果限制在 $(0, 1]$ 之间，并乘上 100 得到百分制的评价结果，计算方法如式4.5所示，

$$Score = 100 \times e^{-[(1-w)\times err_{pos} + w \times err_{angle}]} \tag{4.5}$$

其中 w 为关节夹角偏差所占权重，用以调整两部分偏差对评价结果的影响因数。虽然这一评价指标在数值上仍然存在一定主观因素，但是整体的评分变化趋势却是比较合理，e^{-x} 的

导数 $-e^{-x}$ 随着 x 的增大逐渐趋于 0，也就是说，随着姿态偏差的逐渐增大，评分的下降趋势逐渐减缓，符合人们的正常认知。

有一点值得说明的是，不同的动作所关注的身体部位不同，例如，引体向上关注下巴是否过杆，即动作最高点处头部是否高过两个手腕的位置，这一特点也可以反映在手臂部分的关节角中；深蹲则不关注手部的位置，重点关注膝盖和胯部的角度。因而在计算评分时，也应对不同关节给予不同的权重，这里继续利用第2.3节中动作识别的结果加载对应的动作配置文件，配置文件中可以设置不同关节角以及关节点的权重值。在实验中为了简单起见，例如引体向上的动作，将腿部关节角与关节点权重设为 0，而深蹲动作则将手部关节角与关节点权重设为 0。

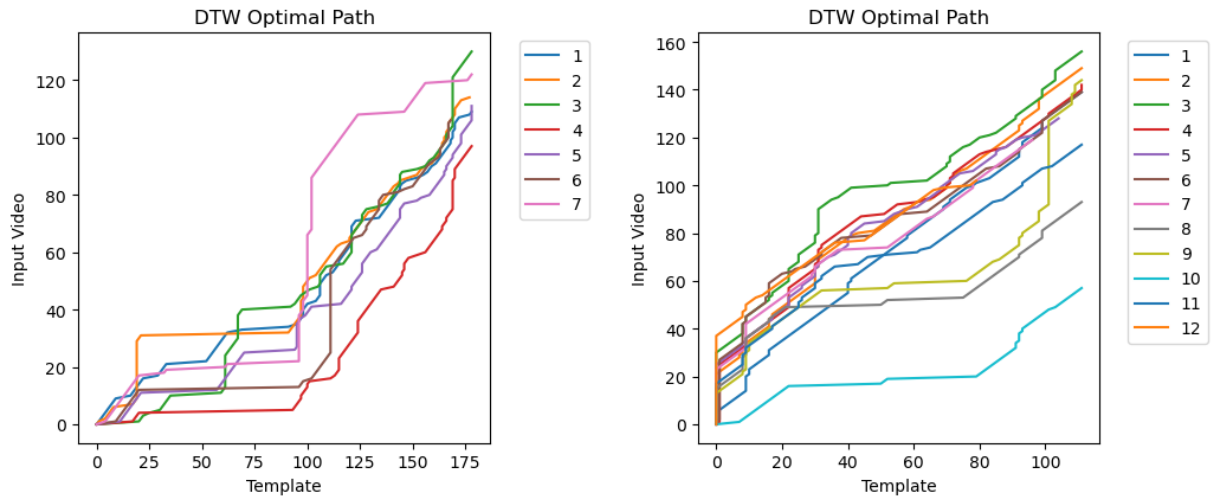
4.3 视频动作评价

视频即图像序列,每一帧的的动作评价都可以使用第4.2节中基于图像动作评价的方法,但是由于视频动作序列的长度不一，以及序列中不同节点动作的重要性不同，视频动作评价需要进行一些额外的处理。下面将分节介绍视频动作评价的一些处理方案与评价指标。

4.3.1 视频动态规整

由于输入视频的时间长度、包含动作的次数都是未知的，无法与模版视频做到帧与帧的一一匹配。利用第3.4节中获得的动作分割结果，可以将输入视频分为一个个单周期的动作序列，这样只需要使用同样单周期的模版视频进行匹配，就可以消除动作次数未知的问题。那么现在需要解决每一个动作分割与模版动作的序列长度不一致的问题。在自然语言处理系统中，由于同一个单词的发音可能长短不一，因而经常需要处理长度变化的序列的相似度问题，而常用的方法为动态时间规整（DTW）算法。该算法输入两组序列，采用一对多或是多对一的策略对序列中的每一个元素寻找匹配，使得两个序列的距离最小，这里又涉及到了距离的度量，序列的距离本质是元素距离的和，因此需要定义序列中元素的距离，即两帧间的距离。这里继续使用第3.4节中进行关键帧聚类所使用的距离度量方法，计算姿态特征向量的欧式距离。

DTW 算法是一种动态规划算法，得到的序列最佳匹配是以最佳路径的形式给出，由于存在一对多和多对一的情况，因此最佳路径的步数一定大于等于其中任意一个序列长度。图4.2给出了输入视频中各个动作分割使用 DTW 算法进行时序规整的最优路径，路径上的每一点横纵坐标对应匹配后的序列帧的索引值。若以模版索引为横坐标，输入视频为纵坐标，路径中斜率较低的部分反映出了这部分动作完成的较模版动作快，反之亦然，而路径轨迹越接近直线 $y = x$，速度上与模板动作越接近。而图中曲线簇的聚合度则和多次动作完成速度的一致性成正相关。



(a) 一组引体向上动作　　(b) 一组深蹲动作

图 4.2: 输入视频各分割 DTW 时序规整最优匹配路径，其中每一根曲线代表一个分割出的单周期动作，曲线的数量为视频中动作完成的周期数。

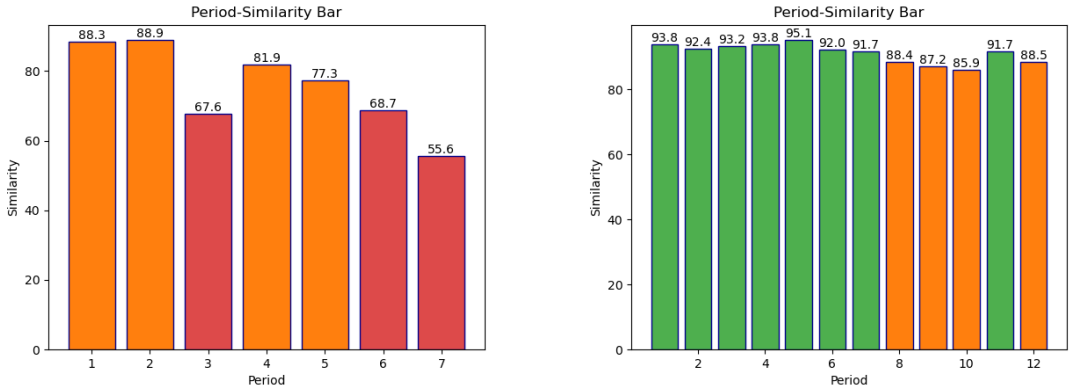
使用 DTW 进行动作序列规整，必须保证两个序列是相同的动作，在没有松弛的情况下，规整完的序列必须是首尾对齐的。在完成动作分割后，已经可以使用分割出的单周期动作与模版动作进行匹配，但是鉴于在进行关键帧提取时，已经获得了中间动作的索引值，事实上这里分割的已不仅仅是单周期动作，而是每一个动作元，因而使用这些动作元进行时序规整，将能够获得更精准的匹配。对于如引体向上或是深蹲这样的 2 动作元动作，对于每一个单周期分割，都利用其中间帧将动作分成两部分独立进行时序规整。

4.3.2 评价指标

在第4.3.1节中完成了单次动作的时序规整后，得到了帧-帧匹配的视频索引，此时对于每一对帧匹配，都可以应用第4.2节中说明的图像评价指标进行动作评价。考虑到视频当中不同位置的动作关键性并不相同，仍然以引体向上为例，人体在向上拉起的过程中，躯干是否垂直向下没有摆动，是衡量动作是否标准的一项指标，但是最终位置下巴是否过杆则相对更为重要，若是直接对所有视频帧进行平均计分得到单次动作的得分，关键帧处下巴过杆这一要求的重要性就被严重稀释。于是考虑对关键帧与过程帧（介于关键帧之间的视频帧）使用不同权值进行加权平均，计算过程如式4.6所示，

$$Score = \omega \times \frac{1}{C}\sum_{i=1}^{C} S_{key_i} + (1-\omega) \times \frac{1}{N-C}\sum_{i=1}^{N-C} S_{int_i} \tag{4.6}$$

其中 C 为当前动作的动作元个数，S_{key_i} 为当前动作关键帧的动作评分，使用式4.5计算得到，对于一个 C 动作元动作，会存在 $C+1$ 个关键帧，其中起始帧和结束帧由于姿态标签一致，并且当前动作周期的结束帧与下一周期的起始帧相同，因此起始帧与结束帧将求平均后再加入计算，以避免增大这两个关键帧的权重。N 则为这一分割序列的总帧数，S_{int_i} 为



(a) 一组引体向上动作　　(b) 一组深蹲动作

图 4.3: 输入视频各分割评分，柱状条根据得分自动设定颜色，得分从高至低颜色依次为：绿、橙、红。

过程帧的动作评分。ω 为关键帧占视频动作评分的权重。由式4.6计算得到的即为分割出的单次动作相似度评分，对所有分割求平均可以得到总的输入视频得分，图4.3展示两个输入视频每一次动作评分柱状图，其分别对应图4.2中的两个输入视频，从图4.2(b)与图4.3(b)可以看出，最优路径偏离直线 $y = x$ 越大的，往往动作评分也越低，不过这个并没有绝对的关系。除评分以外，对视频动作的分析还将给出如下几项结果。

动作计数

利用动作分割的数量可以直接获得视频中动作重复的次数以实现计数功能，为了在视频播放时显示合适的计数点，需要根据对于特定动作，设置人们直观认知的计数位置，例如引体向上，当人达到最高点时进行计数，在实现上可以通过某一关键帧的索引位置来确定计数点。

动作离散度

动作离散度表示同一输入视频中，不同次动作完成情况的一致性，离散度使用 $[0, 1)$ 的浮点数进行表示，离散度越接近 0，表示动作一致性越好。离散度的度量方法使用关键帧的特征距离，使用式3.5计算关键帧 i 到其聚类中心 C 的距离 d_i，然后使用式4.7计算离散度，

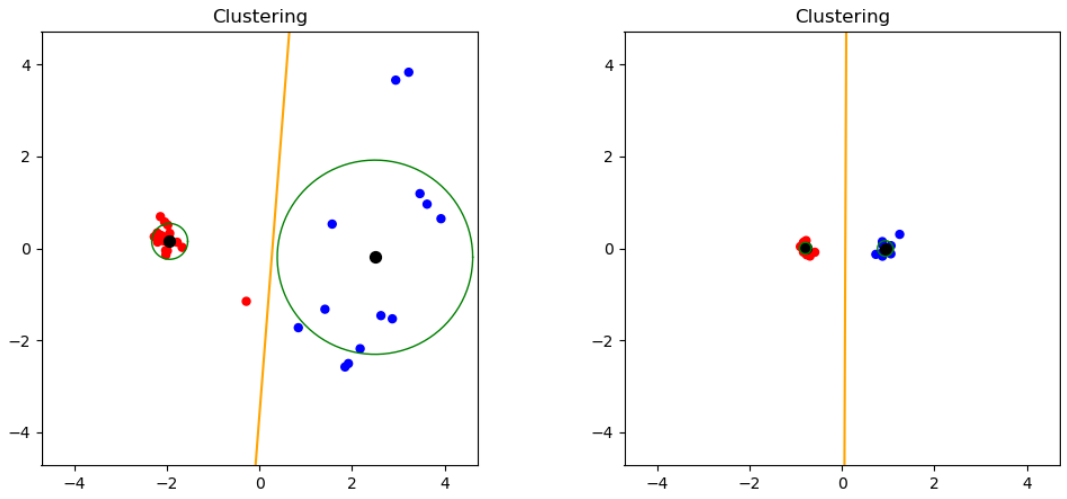
$$Dispersion = 1 - e^{-\frac{1}{N}\sum_{i=1}^{N} d_i} \tag{4.7}$$

其中 N 为关键帧数量。

(a) (b)

图 4.4: 评价结果显示样例

4.3.3 可视化输出结果说明



(a) (b)

图 4.5: 聚类离散度对比

图4.4(a)右侧视频画面显示的是输入的动作视频，绿色骨架是实时估计得的人体姿态，蓝色骨架是通过视频时序规整以及姿态配准获得的对应模版姿态，由此可以直观地比较输入姿态与模版姿态的差异之处，方便使用者进行姿态的纠正。图4.4左侧部分显示的是动作评价的一些计算结果，$Class$ 是动作识别的动作类别结果，$Dispersion$ 是多次重复动作的一致性评估结果，$Count$ 是动作计数的结果，$Score\ for\ current\ frame$ 为当前帧的动作评价得分，$Score\ for\ period\ i$ 是第 i 次动作周期的序列评价得分。图4.4(b)相比图4.4(a)，在最右侧增加了原始模板视频的显示，原始视频相比人体骨架包含更多信息，可以帮助使用者更直接地观察标准动作。图4.5(a)和图4.5(b)分别对应图4.4(a)和图4.4(b)中的视频，可以看出

聚类的特征点分布越分散，其离散度值越高。

4.4 本章小结

本章首先阐述了一种图像动作评价方法，将动作评价定义为输入动作与模版动作的相似度比较。首先推导了一种弱刚性变换矩阵，将模版姿态配准到输入姿态之上，而后设计了一种基于位置偏差和关节角偏差的评分计算方法，并且根据动作识别结果，按照动作类别对不同关节施加不同的权重。继而通过 DTW 算法进行视频时序规整，将图像动作评价推广至视频动作评价，使用动作分割标签完成动作计数功能，依照关键帧与聚类中心的距离平均值计算了多次动作完成的离散度。最后简要介绍了输出视频的理解方式。

第五章　实验结果与分析

5.1　关键帧提取

使用动作元较少的重复动作在静态图片中无法很好的反映关键帧提取的效果，这里采用一组动作元较多的动作分别使用帧间差分法与手工提取的方法获取关键帧进行对比。手工提取的标准是逐帧移动时间轴，当人体达到动作极限位置时标记为关键帧。通过图5.1的对比，可以基本认定帧间差分法提取到的关键帧能反映人体极限位置所在，用以作为动作分割和动作评价的基准是可靠的。

(a) 帧间差分法

(b) 手工提取

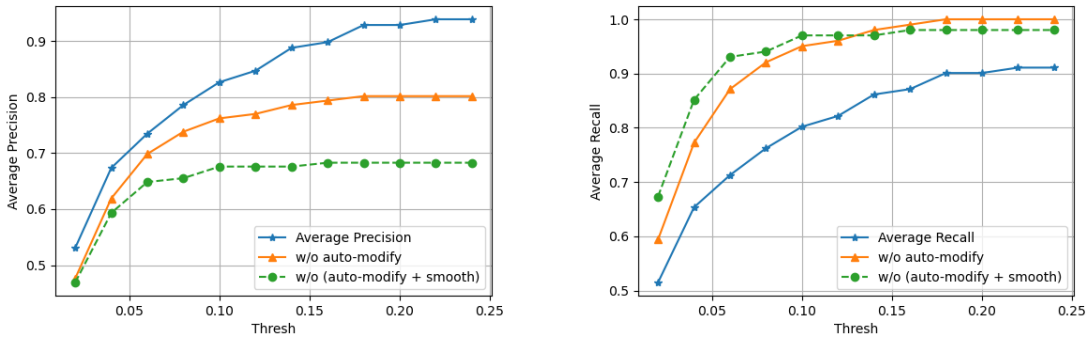
图 5.1: 帧间差分法提取的关键帧与手工提取的关键帧差异比较

为了更准确的评估关键帧提取算法的性能，本文手工标注了所有采集视频的关键帧，评估结果使用平均精度 AP(Average Precision) 与平均召回率 AR(Average Recall) 表示。首先一个问题是如何定义关键帧检测结果是否正确，这里设计一种评估指标，鉴于关键帧的提取是为后续视频分割服务的，使用提取得到关键帧 F_{ex} 与对应真值 F_{gt} 的偏差占整个动作分割周期 T 的比例 p 作为标准，所提取关键帧距离最近的真值将作为对应真值，分割取该对应真值相邻的两个真值关键帧形成的区间，即 $T = F_{gt+1} - F_{gt-1}$，若 F_{gt} 在视频边缘，只有一个相邻关键帧，则 $T = 2 \times (F_{gt+1} - F_{gt})$ 或 $T = 2 \times (F_{gt} - F_{gt-1})$，于是 $p = |F_{ex} - F_{gt}|/T$，当 p 小于设定阈值 $Thresh$ 时，认为该提取关键帧正确。设检测正确的关键帧数量为 N_{TP}，真值关键帧数量为 N_{gt}，提取到关键帧数量为 N_{ex}，那么精度 $P = N_{TP}/N_{ex}$，召回率 $R = N_{TP}/N_{gt}$。AP 与 AR 计算所有样本视频的平均，鉴于标注样本较少，将视频中的关键帧作为独立单

元参与计算，而非视频，计算如式5.1所示。

$$\begin{aligned} AP &= \frac{\sum N_{TP}}{\sum N_{ex}} \\ AR &= \frac{\sum N_{TP}}{\sum N_{gt}} \end{aligned} \tag{5.1}$$

图5.2反映了算法在不同阈值 $Thresh$ 下的关键帧提取性能，阈值设置为 $[0.02:0.02:0.25]$，同时还比较了不添加误检修正模块，以及同时不添加误检修正与二次指数平滑模块下的性能表现。可以看出添加两个模块的算法在精度上有着大幅提升，但在召回率上则有所下降，这是由于平滑算法减少了关键帧早期提案，而误检修正部分则删除一部分关键帧，多少会使得召回率下降。但高精度对于后续的动作分割则更为重要，大量的误检帧会产生很多的小视频分割片段，使得动作评价结果严重失真。



(a) (b)

图 5.2: 关键帧检测 AP 与 AR 随偏差阈值改变的变化曲线

5.2 动作分割

对于动作分割的性能评估，本文同样标注了样本视频的分割标签，按照动作元进行分割，判断分割是否正确则利用了类似于目标检测任务中的 IoU 指标，即计算分割结果 $[S_{ex}, E_{ex}]$ 与真值 $[S_{gt}, E_{gt}]$ 的交并比，计算方法如式5.2所示，图5.3反映了视频序列时间轴 IoU 的直观意义。当计算得到的 IOU 大于设置阈值时，判断分割正确，图5.4反映了动作分割的 AP 与 AR 随 IOU 阈值设置变化的曲线，IoU 阈值设置为 $[0.5:0.025:1.0]$，若设置 $IoU < 0.5$ 则可能出现一个分割同时覆盖两个分割而导致的重复计数。可以看出，添加了标签自动修正和指数平滑的的分割算法在 AR 上与没有添加的两者相比，性能有了大幅提升。最终，本文介绍的动作分割算法在 $IoU = 0.5$ 能达到 $AP = 92.39\%$ 和 $AR = 88.54\%$，以及在 $IoU = 0.75$ 能达到 $AP = 76.09\%$ 和 $AR = 72.92\%$。

$$IoU = \frac{min(E_{ex}, E_{gt}) - max(S_{ex}, S_{gt})}{max(E_{ex}, E_{gt}) - min(S_{ex}, S_{gt})} \tag{5.2}$$

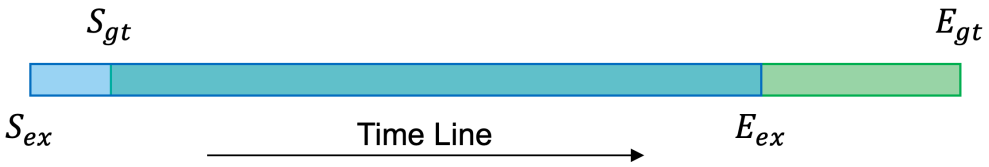
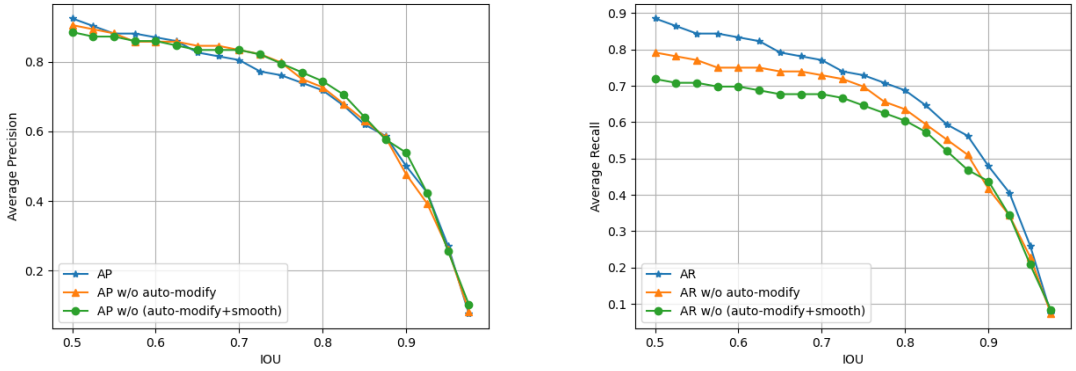


图 5.3: IoU 参数计算两个视频序列时间轴上的交集与并集的比值



(a) (b)

图 5.4: 动作分割 AP 与 AR 随 IOU 阈值改变的变化曲线

5.3 视频预处理优化

表 5.1: 系统开发及运行环境

名称	配置
操作系统	Windows 10 Professional
开发环境	Pycharm Professional
运行环境	Python 3.7 + CUDA 10.2 + Pytorch 1.7
CPU	Intel i9-9900K
RAM	32GB
GPU	Nvidia RTX 2080Ti

本节及第5.4节所有测试数据均基于表5.1所示的运行环境与一个 1572 帧 1080P 的样例视频，下简称样例视频。

目前系统运行速度的瓶颈在于 OpenPose 姿态估计的部分，为了提高调试效率以及在使用中方便使用者调整参数后尽快输出结果，系统将在完成姿态估计后保存提取得到的关节骨架，当再次输出此视频时，系统将优先检索是否存在已提取的骨架序列。

而进一步分析整体系统可知，目前系统中需读取三次视频原帧数据：关节骨架提取、帧间差分计算、输出结果渲染。表5.2中测试了视频的读取解码并进行缩放需要消耗的时间，可以发现这一操作将会消耗较大的计算资源和时间，考虑将视频经过预处理后加载在内存当中随时调用。表5.2中进一步比较了将视频存储为列表再转换为 numpy array 与预分配内

存直接保存为 numpy array 的代码实现，结果可以发现前者的内存占用显著增加，这是由于 python 列表其实是指针容器，内存地址是不连续的，转换为 numpy array 时需要重新分配一块整块的内存，复制数据时会产生双倍的内存占用。若是直接读取关节骨架，使用预加载视频的方式，样例视频总处理时间为 23 秒（其中还包含了编码保存输出视频的时间，约 5 秒），而无预加载则为 38 秒，处理时间缩短将近 40%。在内存空间允许的情况下，开启预加载设置能带来较大幅度的性能提升。

表 5.2: 视频预处理时间比较

样例视频	耗时	内存占用（缩放至 512 × 910）
仅解码读取（3 次）	22.715s	170 MB
解码读取先保存为 list 后转 ndarray	8.272s	3.3 GB
解码读取并直接保存为 ndarray	7.703s	1.9 GB

5.4 系统运行时分析

目前系统可以达到准实时的运行速度，主要的计算瓶颈在于 OpenPose 的姿态估计部分，其运行速度在 RTX 2080Ti 上大约为 15FPS 左右，并且经过测试，这一速度与输入视频的分辨率并无显著线性关系，降低输入分辨率并不能显著提高 OpenPose 的估计速度，对于视频文件（30FPS）的分析处理时间大约在视频时长的 2 倍左右。若是将系统应用在使用摄像头的实时系统上，由于从摄像头获取的均为当前时刻帧，并不会产生帧缓存堆积问题，可以在摄像头采集视频的同时进行姿态估计与帧间差分计算，采得的视频序列约为 15FPS 左右。而其余的动作分析部分则在完成视频采集后进行，主要由于大部分的动作信息需要获取全局上下文关系，但这部分的处理耗时相比 OpenPose 极短，，并不会产生很长的等待时间。

而系统的空间占用则分为两部分，内存占用与显存占用。显存占用为 OpenPose 及 ST-GCN 所使用，OpenPose 显存占用约为 2.5GB，ST-GCN 显存占用为 0.6GB，并且均与输入图像的分辨率无显著线性关系。内存占用则与视频分辨率及视频帧数成强正线性相关，占用主要部分是加载的完整的经过预处理的视频帧，这一部分内存使用主要是为了实现更快的运行速度，可以根据所使用计算机的内存资源进行调整。当处理样例视频时，系统的峰值内存占用约为 11.7GB，而当将同一段视频分辨率降至 512 × 910 时，系统的峰值内存占用相应的降至 3.7GB 左右，使用预加载视频的策略对于处理较短时长，较低分辨率的视频时，消耗时间上具有显著的优势，但是当视频体积增大时，将导致极大的内存占用，应关闭预加载设置。

5.5 本章小结

本章首先给出了关键帧提取与动作分割的一种性能评估指标，并对文中所介绍的算法性能在样本数据上进行了测试。然后介绍了一种视频预加载的优化方法，来提升小视频的处理速度。最后详细分析了系统运行时的时空资源消耗，用以帮助读者了解部署系统所需的计算配置。

第六章　总结与展望

6.1　工作总结

本设计完成了一个动作评价系统，以动作视频作为输入，能够识别视频中动作所属类别，对重复动作进行计数，并且将重复动作进行分割，给出每一分割的评价得分，以及评估重复动作完成的一致性（离散度）。此外，能够可视化地观察输入动作与模版动作在关节层面上的差异，帮助使用者更清晰地了解到自己与标准动作的差异之处。

其中，部署了 OpenPose 与 ST-GCN 模型分别进行关节骨架提取与动作识别，定义了一组人体姿态特征向量用以度量动作相似度，提出了一种基于帧间差分法的关键帧检测与基于 K-means 关键帧聚类的方法进行视频动作分割，推导了一种弱刚性变换进行人体姿态配准，设计了一系列基于图像和基于视频的动作评价指标。

6.2　工作展望

6.2.1　系统不足与改进方向

拍摄视角固定

当前使用单目 RGB 只能获取到 2D 的人体关节骨架，由于各关节及肢体位置均为投影结果，关节角度只有在同一视角拍摄的情况下才具有可比性。这一问题对于一个固定场景的动作评价系统倒是影响不大，但是对于大众用户居家使用等环境视角不确定的情况，则会大幅降低动作的评估有效性。这一问题最为简单的解决方法就是换用深度相机或者多目相机来获取 3D 的人体关节骨架，三维空间内的肢体与肢体夹角便不再随视角变化而改变，考虑到现在许多移动设备，如手机等，很多都已配备 TOF 相机或是激光雷达等深度探测传感器，改用深度相机进行系统设计仍然具有较大的市场应用空间。同时，本设计中所使用的大部分算法均可无缝迁移至新的 3D 系统，未来可能将应用 Kinect 等 RGB-D 相机进行实验验证算法可行性。

帧间差分法鲁棒性较差

帧间差分法检测人体运动关键帧的基础在于相机在拍摄过程中是稳定的，并且场景中没有其他的活动物体，这一问题同样对固定场景的动作评价系统影响不大，对于手持设备和户外使用情况影响非常大。一种改进方式是利用关节角度差分检测关键帧，先前已经实验过这一方法，目前设计的算法在满足相机稳定，场景无其他活动物体的情况下，关键帧的检测效果暂不如帧间差分法，因此暂时未部署，主要的原因在于，关节角度差分依赖于

姿态估计的结果，由于 OpenPose 获得关节骨架存在显著的抖动，以及很多突变的错误估计。还有一种方法可使用类似遗传算法的方法，通过迭代学习的方法寻找关节角度曲线中（如图3.3）人眼显著的波峰波谷位置，需要控制算法不会向全局最优进行优化，而是向多个局部最优进行优化。

无法实时输出动作分析结果

由于关键帧提取、聚类以及视频的动态规整均需要较大的上下文背景，甚至是全局上下文，因此这些分析都只能在获取完整的视频序列后才能够进行。

姿态估计无追踪

由于系统设计使用场景不包含多人，因此这里的追踪并非是指给每个检测到的人体分配轨迹 id，而是指利用先前的估计姿态，对当前帧的估计结果进行关节层面的追踪，以保障关节估计值不会发生大的抖动甚至是缺失。当前使用了较为简单的线性插补以及关节平滑等措施，没有部署更为鲁棒的追踪器。

资源占用较大

通过按照第5.4节中的说明调整内存使用设置或是改变输入视频分辨率，内存的占用量可以降到一个比较低的值，但是显存的占用量还是非常可观的，未来可能会在小显存机器上测试 OpenPose 显存占用是否有改善，若是没有，则需要考虑换用其他模型，或是使用 MobileNet 等主干网络以将系统部署到移动设备或是性能比较弱的计算设备中。

6.2.2 工程应用的完善方向

目前的系统属于一种动作评价算法验证的测试模型，使用完整的动作视频作为输入，尚未开发使用摄像头的实时运行模式。考虑工程实际应用，使用摄像头输入，则需要进行一些起始终止检测。在系统启动后，系统需要自动检测使用者是否开始进行动作录制，以及能够检测动作完成以自动停止录制。

下面叙述一种较为简易的部署方式。可以使用 YOLO 人体探测器以及帧间差分运动检测器，当探测器检测到人体，并且人体持续静止一段时间后，显示屏将显示倒计时提示使用者，当倒计时结束开始录制动作视频。当系统在录制动作视频时，如果再次满足检测到人体，并且人体静止达一定时间时，停止录制，并且删除视频最后人体静止的部分。为了减少使用者等待动作分析结果的时间，可以在录制过程中同时进行姿态估计、动作识别以及帧间差分这些不需要全局上下文的模块。

参考文献

[1] CAO Z, SIMON T, WEI S E, et al. Realtime multi-person 2d pose estimation using part affinity fields[C]//CVPR: vol. 1: 2. 2017: 7.

[2] FANG H S, XIE S, TAI Y W, et al. Rmpe: Regional multi-person pose estimation[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 2334-2343.

[3] WANG J, SUN K, CHENG T, et al. Deep high-resolution representation learning for visual recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2020.

[4] LECUN Y, BENGIO Y, et al. Convolutional networks for images, speech, and time series[J]. The handbook of brain theory and neural networks, 1998: 255-258.

[5] WEI S E, RAMAKRISHNA V, KANADE T, et al. Convolutional pose machines[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 4724-4732.

[6] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[7] NEWELL A, YANG K, DENG J. Stacked hourglass networks for human pose estimation[C]//European Conference on Computer Vision. 2016: 483-499.

[8] LUCAS B D, KANADE T, et al. An iterative image registration technique with an application to stereo vision[C]//Proceedings of the 7th international joint conference on Artificial intelligence: vol. 2. 1981: 674-679.

[9] DOSOVITSKIY A, FISCHER P, ILG E, et al. Flownet: Learning optical flow with convolutional networks[C]//Proceedings of the IEEE international conference on computer vision. 2015: 2758-2766.

[10] SIMONYAN K, ZISSERMAN A. Two-stream convolutional networks for action recognition in videos[J]. ArXiv preprint arXiv:1406.2199, 2014.

[11] FEICHTENHOFER C, PINZ A, ZISSERMAN A. Convolutional two-stream network fusion for video action recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 1933-1941.

[12] TRAN D, BOURDEV L, FERGUS R, et al. Learning spatiotemporal features with 3d convolutional networks[C]//Proceedings of the IEEE international conference on computer vision. 2015: 4489-4497.

[13] WANG L, XIONG Y, WANG Z, et al. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition[C]//ECCV. 2016.

[14] LIU J, SHAHROUDY A, XU D, et al. Spatio-temporal lstm with trust gates for 3d human action recognition[C]//European conference on computer vision. 2016: 816-833.

[15] YAN S, XIONG Y, LIN D. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition[C]//AAAI. 2018.

[16] LIU Z, ZHANG H, CHEN Z, et al. Disentangling and unifying graph convolutions for skeleton-based action recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 143-152.

[17] SHAHROUDY A, LIU J, NG T T, et al. Ntu rgb+ d: A large scale dataset for 3d human activity analysis[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 1010-1019.

[18] SMAIRA L, CARREIRA J, NOLAND E, et al. A Short Note on the Kinetics-700-2020 Human Action Dataset[J]. ArXiv preprint arXiv:2010.10864, 2020.

[19] YAN S, XIONG Y, WANG J, et al. St-gcn[Z]. https://github.com/yysijie/st-gcn. 2018.

[20] ZHAO Y, XIONG Y, WANG L, et al. Temporal action detection with structured segment networks[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 2914-2923.

[21] SINGH G, CUZZOLIN F. Untrimmed video classification for activity detection: submission to activitynet challenge[J]. ArXiv preprint arXiv:1607.01979, 2016.

[22] BUCH S, ESCORCIA V, GHANEM B, et al. End-to-end, single-stream temporal action detection in untrimmed videos[C]//Procedings of the British Machine Vision Conference 2017. 2019.

[23] LIN T, ZHAO X, SHOU Z. Single shot temporal action detection[C]//Proceedings of the 25th ACM international conference on Multimedia. 2017: 988-996.

[24] LIN T, ZHAO X, SU H, et al. Bsn: Boundary sensitive network for temporal action proposal generation[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 3-19.

[25] LIN T, LIU X, LI X, et al. Bmn: Boundary-matching network for temporal action proposal generation[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 3889-3898.

[26] DWIBEDI D, AYTAR Y, TOMPSON J, et al. Counting Out Time: Class Agnostic Video Repetition Counting in the Wild[C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020.

[27] JIANG Y G, LIU J, ROSHAN ZAMIR A, et al. THUMOS Challenge: Action Recognition with a Large Number of Classes[Z]. http://crcv.ucf.edu/THUMOS14/. 2014.

[28] FABIAN CABA HEILBRON B G, Victor Escorcia, NIEBLES J C. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 961-970.

[29] JI R, YAO H, SUN X. Actor-independent action search using spatiotemporal vocabulary with appearance hashing[J]. Pattern Recognition, 2011, 44(3): 624-638.

[30] JIANG Z, LIN Z, DAVIS L. Recognizing human actions by learning and matching shape-motion prototype trees[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(3): 533-547.

[31] 吴齐云. 基于 Kinect 的上肢运动康复交互系统研究[D]. 广东工业大学, 2016.

[32] ALEXIADIS D S, KELLY P, DARAS P, et al. Evaluating a dancer's performance using kinect-based skeleton tracking[C]//Proceedings of the 19th ACM international conference on Multimedia. 2011: 659-662.

[33] WANG J, QIU K, PENG H, et al. AI coach: Deep human pose estimation and analysis for personalized athletic training assistance[C]//Proceedings of the 27th ACM International Conference on Multimedia. 2019: 374-382.

[34] WEI S E, RAMAKRISHNA V, KANADE T, et al. Convolutional pose machines[C]//CVPR. 2016.

[35] LI J, WANG C, ZHU H, et al. CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark[J]. ArXiv preprint arXiv:1812.00324, 2018.

[36] XIU Y, LI J, WANG H, et al. Pose Flow: Efficient Online Pose Tracking[C]//BMVC. 2018.

[37] CAO Z, HIDALGO MARTINEZ G, SIMON T, et al. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.

[38] CHEN C, ZHUANG Y, NIE F, et al. Learning a 3D human pose distance metric from geometric pose descriptor[J]. IEEE Transactions on Visualization and Computer Graphics, 2010, 17(11): 1676-1689.

[39] 周志华. 机器学习[M/OL]. 清华大学出版社, 2016. https://books.google.com/books?id=j0G8nQAACAAJ.

致　　谢

衷心感谢导师夏思宇教授对我的帮助与指导。夏老师对待学生非常真诚和善，对待工作与学术一丝不苟，教导我认真勤奋、脚踏实地，鼓励我不断向前、砥砺前行，使我受益终生。感谢实验室的李成贤师兄与庄文林师兄给我提供的一些研究思路和研究方向，与他们探讨使我受益良多。感谢杭天恺学长提供的 LaTeX 毕业设计论文模板，省去了我大量的排版工作，帮助我高效地完成了论文的编写。感谢同班的赵永强同学帮助我拍摄了许多动作视频。感谢我的父母一直以来对我的精神和经济支持。