

KinectV2 Sensor Real-Time Dynamic Gesture Recognition Algorithm

Xidong Liu¹, Yan Meng¹, Guoyou Li², Lei Tang², Jiadong Guo²

¹College of Biological and Environmental Engineering, Tianjin Vocational Institute University, Tianjing

²Shanghai Institute of Aerospace Technology, Shanghai

Email: 15733548281@163.com

Received: Feb. 15th, 2020; accepted: Mar. 3rd, 2020; published: Mar. 10th, 2020

Abstract

In order to solve the problems of low generalization of features extracted by the dynamic gesture action recognition algorithm, poor real-time recognition and low recognition rate, this paper proposes a template matching gesture recognition method based on KinectV2 sensor. This method uses Kinect Studio and Visual Gesture Builder to build a gesture action database, extracts the angle between key bones of the human body as a one-dimensional feature vector that changes with time, and proposes Weighted DTW gesture motion recognition algorithm based on the dynamic time warping (DTW) algorithm with the help of dynamic planning ideas. Through the improvement of the DTW algorithm, the regular path converges faster, the real-time and accuracy of gesture motion recognition is improved, and the real-time and validity verification experiments are carried out in the Unity 3D virtual simulation experiment platform for simple human-computer interaction.

Keywords

Kinectv2 Sensor, Gesture Recognition, Skeletal Angle Characteristics, Human Computer Interaction

KinectV2传感器实时动态手势识别算法

刘希东¹, 孟岩¹, 李国友¹, 唐磊², 郭佳栋²

¹天津职业大学生物与环境工程学院, 天津

²上海航天技术研究院, 上海

Email: 15733548281@163.com

收稿日期: 2020年2月15日; 录用日期: 2020年3月3日; 发布日期: 2020年3月10日

摘要

为解决动态手势动作识别算法提取的特征泛化程度低, 识别实时性差和识别率不高的问题, 本文提出了

一种基于KinectV2传感器的模板匹配手势识别方法。该方法利用Kinect Studio和Visual Gesture Builder建立手势动作数据库,提取人体关键骨骼间的夹角作为随时间变化的一维特征向量,借助动态规划思想,在动态时间规整(DTW)算法的基础上提出了加权DTW手势动作识别算法。通过对DTW算法的改进,使其规整路径更快的收敛,提高了手势动作识别的实时性和准确性,并在Unity 3D虚拟仿真实验平台中进行了实时性和有效性的验证实验,同时实现了简单的人机交互。

关键词

KinectV2传感器, 手势识别, 骨骼角度特征, 人机交互

Copyright © 2020 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

现代石油工业发展迅猛,规模不断增大,人们对其工艺水平要求越来越严格。但在实际生产中,随着石油产业规模的加大,往往会带来爆炸、有毒物质泄露等安全因素,因此,可引入手势识别技术,使用手势对单体化工设备进行放缩,旋转等操作,使一线生产工人对设备外观等有更直观的了解,此外,还可以在虚拟现实平台进行事故演习时控制灭火器,阀门旋转等来模拟事故现场应急方案。从而为石化装置生产一线人员提供一个真实感官效果、状态信息明确、操控方便的全新工作环境[1]。

手势识别技术在半个世纪前就开始有所体现,在2010年11月,随着微软公司推出Kinect体感交互设备,手势交互技术迅速被众多研究人员视为研究热点,尤其是其在骨骼追踪,预测及人体定位有着非常好的准确度,可以说是人机交互的一个革命性产品,随着Kinect for Windows SDK的推出,研究人员可以更加方便的获取用户的骨骼位置信息等,可以更加理解使用者的手势意义,从而可以帮助用户完成更好的人机交互体验。

通过研究发现,语音识别算法和手势识别算法基本形同,主要有矢量量化,轨迹追踪和模板匹配三种方法[2],因为矢量化的识别算法识别效率比较低,现在已经基本不再研究,所以近年来的手势识别算法主要是以隐马尔科夫模型(HMM)为代表的轨迹追踪算法和以动态时间规整模型(DTW)为代表的模板匹配算法。郭晓利等人[3]提出了一种改进的HMM模型算法,该算法与模糊神经网络相结合,比原HMM模型要稳定,但是还没有应用到实际中,还有待在实际环境中进一步检测。Saha S等人[4]提出了一种特征描述子用来描述单个时间序列的扭曲信息,结合HMM模型来进行手势识别,该方法对12种不同的手势进行了测试,虽然准确度较高,但由于该方法模型较为复杂,所以实时性较差。李凯等人[5]提出了一种改进的动态时间规整算法,该方法将获取到的Kinect的骨骼点坐标及手型数据结合,构造了矢量特征来描述手的运动轨迹,实现了手势的快速匹配,但未考虑到每个人因骨骼坐标位置不同而对识别结果带来的影响;Ruan X等人[6]提出了一种通过失真度阈值和路径约束来改进DTW的算法,该方法通过选取8个点作为手部特征建立手势的数学模型,实验结果表明,该方法无论是在识别率还是鲁棒性方便,都取得了很好的实验效果。综上所述,HMM和DTW算法在手势明显的情况下,分类效果差别不大,通常情况下,HMM算法的识别率是要稍高于DTW算法,但是HMM方法模型复杂,计算量较大,因此实时性不如DTW算法,很难在实际生活中应用。

本文针对提取骨骼坐标位置特征时,因个人骨骼坐标位置不同造成的识别效率低,实时性差的缺陷,

提出了一种基于 KinectV2 传感器骨骼数据的改进 DTW 模板匹配手势识别算法。该算法通过提取人体关键骨骼间的夹角特征，改善了因每个人骨骼位置不同造成的对手势动作特征描述不准确的缺点；通过对 DTW 算法进行改进，提高了计算速率，使动态规整路径更快的收敛。本文的算法不受骨骼坐标位置差异影响，能够使路径更快的收敛，既满足手势识别的准确性，又满足实时性。

2. Kinect 骨骼跟踪技术

Kinect SDK 2.0 中提供了人体的各个骨骼关节的空间坐标位置分布信息及相对位置的变化，骨骼节点的数据类型以骨骼帧的形式表示，每一帧都是由 25 个骨骼节点组成的 JointType 类型的集合[7]，人体各部位节点位置标注图和编号如图 1 所示。根据 KinectV2 追踪骨骼数据的状态在 Kinect SDK 2.0 中定义了三种骨骼节点跟踪情况：能够进行跟踪(Tracked)、不能够进行跟踪(Not Tracked)、通过运动状态进行推断跟踪(Infered) [8]。

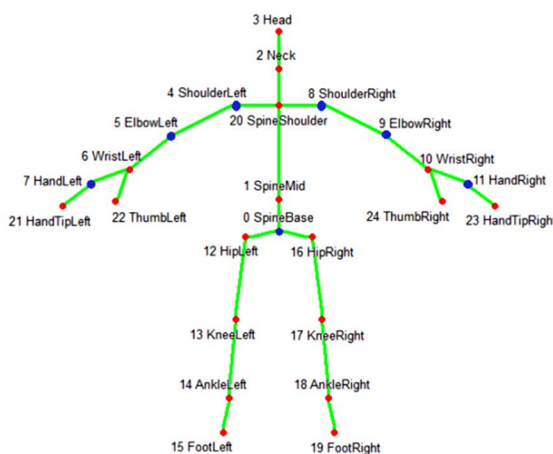


Figure 1. Contrast map of human skeleton node
图 1. 人体骨架节点对照图

3. 建立手势样本库

本文在建立手势样本库过程中主要分为录取手势，制作手势正负样本和通过程序代码调用手势样本库三个步骤，具体步骤的工具和如图 2 所示：

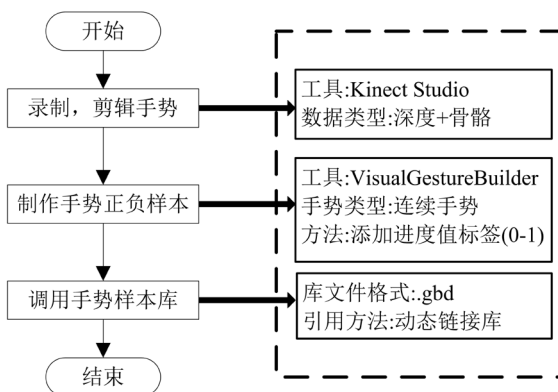


Figure 2. Gesture sample library building flowchart
图 2. 手势样本库建立流程图

4. 手势运动特征提取

如何提取低计算量, 低复杂度的特征并且找到统一的动作特征来对人体动作进行识别, 是手势识别过程中的关键问题。以往的研究者们在进行骨骼动作的识别的时候, 主要提取位置特征, 角度特征或者速度特征来描述手势的运动[9], 然而对于不同身高, 不同体型的人做同一动作或者同一个人做不同动作的过程中, 由于每个人的骨骼关节位置信息有一定的差异, 运动速度也不尽相同[10], 均不能统一而准确的描述手势的运动, 但是即使每个人的骨骼坐标位置和运动速度不同, 在运动过程中, 关节间的夹角特征并没有发生明显变化, 基于这种现状, 本文中提出了将手势运动过程中的骨骼间的向量夹角作为手势特征描述, 并结合动态时间规整(DTW)算法进行模板匹配, 夹角特征提取过程如下。

4.1. 人体骨骼节点坐标获取

如图3所示, 以求取右手手腕, 右肩和颈下脊椎之间的夹角为例, 其中, WR 代表右手手腕关节点, SR 代表右肩关节点, SS 代表颈下脊椎关节点, 要想求它们之间的空间向量的夹角, 则要求其关节点的空间坐标位置。

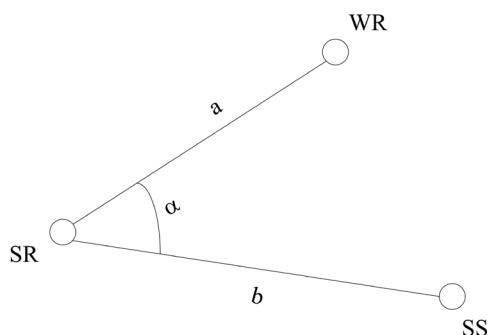


Figure 3. Diagram of joint angle
图3. 关节夹角示意图

虽然 KinectV2 能够得到人体各骨骼节点图像, 但是却不能够直接获取其各个节点的空间坐标位置, 所以我们需要通过编程的方式来采集需要的关节点坐标, 如果将 KinectV2 捕捉的 25 个关节点全部用来做人体运动的特征分析, 则会增加本系统的计算量和复杂度, 从而影响系统的实时性, 考虑到文中所识别的手势只用到了人体上半身的 8 个关键骨骼点, 所以只对上半身的 8 个骨骼节点坐标进行提取, 由于篇幅原因, 表 1 中只列出了其中一组关键骨骼节点的空间位置坐标。

Table 1. Coordinate positions of key skeletal nodes

表 1. 各关键骨骼节点坐标位置

空间坐标位置	Point.x	Point.y	Point.z
ShoulderRight	288	217	1.53865
ElbowRight	311	262	1.42707
WirstRight	336	244	1.21293
SpinShoulder	243	202	1.55516
SpinWid	242	255	1.50976
ShoulderLeft	190	210	1.53874
ElbowLeft	177	266	1.47452
WirstLeft	172	316	1.30465

4.2. 夹角特征提取

如图 3, 假设 WR 关节点的坐标为 (x_1, y_1, z_1) , SR 关节点的坐标为 (x_2, y_2, z_2) , SS 的关节点坐标为 (x_3, y_3, z_3) , 则向量 $a = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$, 向量 $b = (x_3 - x_2, y_3 - y_2, z_3 - z_2)$, 设 $\langle a, b \rangle$ 之间夹角为 α , 则有:

$$\cos \alpha = \frac{a \cdot b}{|a||b|} \quad (1)$$

$$a \cdot b = (x_2 - x_1)(x_3 - x_2) + (y_2 - y_1)(y_3 - y_2) + (z_2 - z_1)(z_3 - z_2) \quad (2)$$

$$|a| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (3)$$

$$|b| = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2} \quad (4)$$

将上述中的式(2), (3), (4)代入式(1) 即可求出三个关键骨骼关节点两两连成的向量的夹角。如图 4 所示, 使用上述方法分别测出了某一帧右手手腕, 右肘, 右肩之间的夹角和颈下脊椎, 右肩, 右手肘之间的夹角。

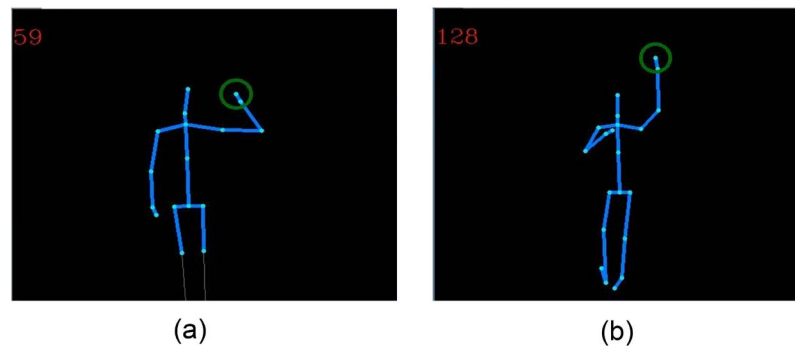


Figure 4. The angle between skeletons: (a) Right wrist, right elbow and right shoulder; (b) Lower cervical spine, right shoulder and right elbow

图 4. 不同骨骼间夹角角度: (a)右手手腕, 右手肘和右肩的夹角; (b)颈下脊椎, 右肩, 右手肘的夹角

5. DTW 算法原理介绍及实验分析

动态时间规整(DTW)算法是一种模板匹配方法, 最开始用于处理语音序列长短不一的模式匹配问题, 由于手势序列也是长短不一的, 因此 DTW 算法也成为手势识别中常用的一种方法。

动态时间规整算法实际上是一个典型的优化问题, 即利用距离函数计算出的距离值表示两个模板或者动作序列之间的相似度。如果两模板长度相等, 则可以直接计算出这两个模板序列的距离来进行匹配识别; 如果长度不相等, 则需要将它们的时间轴上对齐, 这时引入一个矩阵 A , 用该矩阵里面的每一个元素 (i, j) 表示两模板元素之间的距离 $d(L_i, C_j)$, 这个距离就代表模板 L 和模板 C 的相似度。

本次实验中使用的欧氏距离函数计算方法, 因为它可以计算任何维度空间内的两个点之间的距离, 如在二维平面空间里, 两点之间的距离公式可以表示为:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (5)$$

在三维立体空间里, 任意两点之间的距离公式可以表示为:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (6)$$

基于欧式距离的最短距离计算共有三个约束条件[10]:

- 1) 边界性: 规划的路径必须是从引入的矩阵的左下角开始, 以右上角结束。
- 2) 连续性: 路径上的点与相邻的点必须对齐, 这样做的目的是为了保证计算到L和C中的每一个点。
- 3) 单调性: 路径上的点相对时间轴必须呈递增状态。

按照上述约束条件可以推出, 对于A中的每一个元素而言, 有且只有三个元素与之相连, 元素的节点分布图如图5所示:

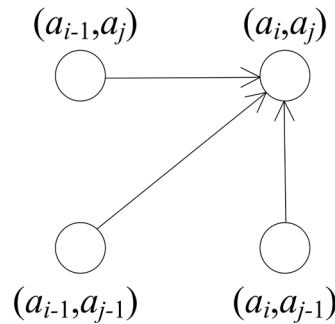


Figure 5. Element node distribution map
图5. 元素节点分布图

这里我们定义一个累积距离 $s(i, j)$, 对L和C这两个模板序列从起点(1, 1)点进行计算, 累计之前所有的点计算的距离, 即累计点 l_i 与点 g_j 之间的距离,

再加上其周围与点 (i, j) 距离最小的那个点之间的距离, 如式(7)所示。当匹配到达终点 (m, n) 的时候, 距离值 $s(i, j)$ 则代表这两个时间序列的相似度。

$$s(i, j) = d(L_i, C_j) + \min \begin{cases} s(i-1, j) \\ s(i, j-1) \\ s(i-1, j-1) \end{cases} \quad (7)$$

累计距离和相似度成反比, 累计距离越小, 两个动作模板之间的相似度越高, 累计距离越大, 两个动作模板之间的相似度越低[11]。在这里, 为了更好的理解 DTW 算法, 我们举例进行详细说明, 如图9所示, 分别为长度为5的标准模板{A, B, C, D, E}和长度为4的测试模板为{F, G, H, I}, 模板中各元素距离已经给出, 为了找出这两条模板最短的匹配路径, 现在假设该模板满足下面的前提: 点 $(i-1, j)$ 、 $(i-1, j-1)$ 、 $(i, j-1)$ 作为起点, 点 (i, j) 作为与之相连的点, 如果从 $(i-1, j)$ 到 (i, j) 或者从 $(i, j-1)$ 到 (i, j) , 则距离为 $d(i, j)$, 如果是从 $(i-1, j-1)$ 到 (i, j) , 距离为 $2d(i, j)$, 式(8)表示其前提条件为:

$$g(i, j) = \min \begin{cases} g(i-1, j) + d(i, j) \\ g(i-1, j-1) + 2d(i, j) \\ g(i, j-1) + d(i, j) \end{cases} \quad (8)$$

我们假设 $g(0,0) = 0$, 则 $g(1,1) = g(0,0) + 2d(1,1) = 4$; 而 $g(2,2)$ 的计算分三条路径比较: 如果将 $g(1,2)$ 作为起点 $g(2,2) = g(1,2) + d(2,2) = 9 + 3 = 12$; 如果将 $g(1,1)$ 作为起点, $g(2,2) = g(1,1) + 2d(2,2) = 4 + 6 = 10$; 如果将 $g(2,1)$ 作为起点, $g(2,2) = g(2,1) + d(2,2) = 8 + 3 = 11$ 。

根据上述距离值计算, 取最小值, $g(2,2) = 10$; 以此类推, 矩阵A中两个模板的最优匹配路径经过的格点为(1, 1), (2, 1), (2, 2), (2, 3), (3, 4), (4, 4), (4, 5), 如图6红线所示:

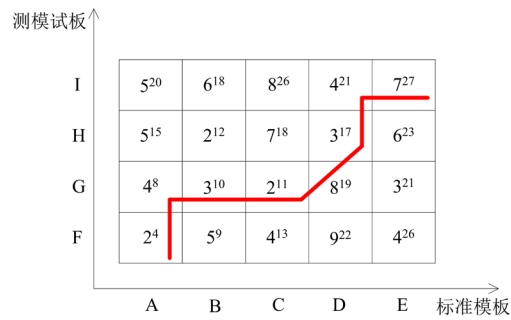


Figure 6. Optimal matching path
图 6. 最优匹配路径

本次实验中采用的动态时间规整算法对手势进行识别的具体设计步骤如图 7 所示:

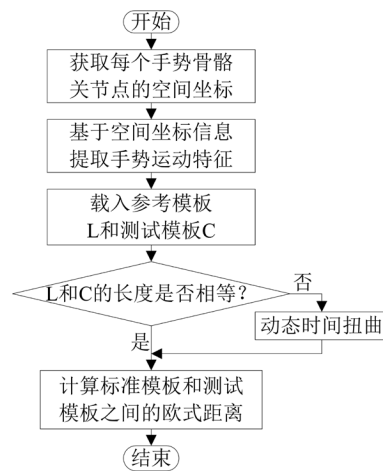


Figure 7. DTW algorithm flow
图 7. DTW 算法流程

这里以左滑手势为例, 利用 Visual Studio 和 Matlab 混合编程环境进行分析, 这个手势主要是基于颈下脊椎, 左肩, 右肘这三个关键骨骼节点间的夹角进行判断, 将这三个关键骨骼点每一帧形成的夹角作为一组关节角度向量来进行左滑动动作的描述, 测试者在左滑手势时的关节角度变化的曲线如图 8 所示:

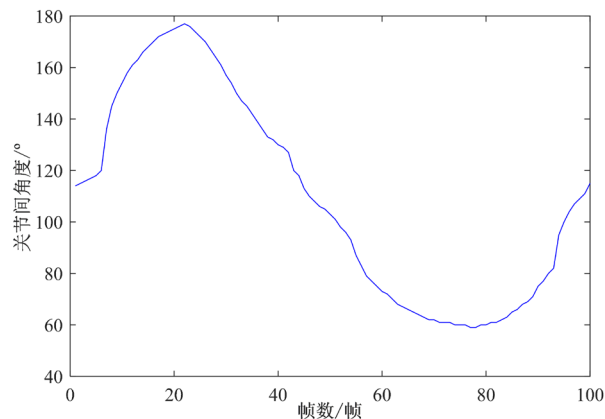


Figure 8. Angle change curve of left sliding gesture joint of tester
图 8. 测试者左滑手势关节角度变化曲线

从图 8 中可以看出, 关节间夹角的角度变化是一条随时间的变化而不断变化的曲线, 可以反映出左滑这个动作的手势特征, 角度的变化可以用关于时间的一维向量表示。

由于每个人的身高, 体型, 运动速度的不同导致手势完成的时间和每一帧的角度变化范围也不相同, 虽然角度变化曲线的起伏程度不完全相同, 但是从整体上来看, 曲线的基本形状确是一样的, 在进行模板匹配时, DTW 会将长短不一的模板序列在时间轴上进行扭曲对齐, 两模板序列关于时间轴的扭曲对齐实验仿真结果如图 9 所示。

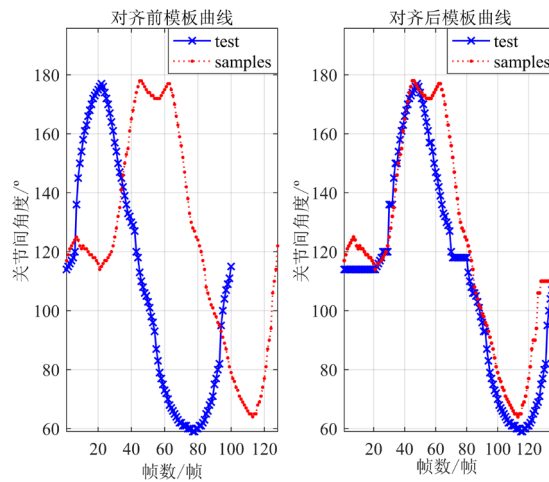


Figure 9. Time distortion alignment experiment in template matching
图 9. 模板匹配过程中时间扭曲对齐实验

经过上述实验步骤后, 就可以寻找两模板之间的最优匹配路径, 这里还是以左滑手势为例, samples 为时间序列为 128 帧的标准模板, test 为时间序列为 100 帧的测试模板, 因为参考模板和测试模板的长度不一致, 所以进行了时间序列的扭曲, 将长短不一的时间序列特征进行了对齐, 如图 9 所示。将时间序列进行对齐后, 计算两模板的最优匹配路径和欧式距离, 进而计算累计距离, 当累计距离越小时, 则表明两模板的相似度越高, 左滑手势的测试模板和标准模板的最优路径匹配结果如图 10 所示:

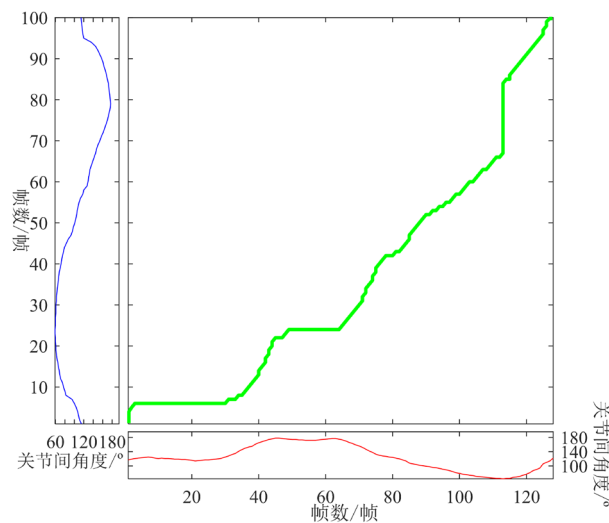


Figure 10. Optimal matching path result of left-slip gesture
图 10. 左滑手势最优匹配路径结果

6. DTW 算法改进

由图 11 所示, 左滑手势两模板的规整路径是弯曲上升的一条折线, 而理想状态下的规整路径应该是一条接近对角线, 接近直线的路径。变为图 11 所示路径的原因是随着样本数量和计算量的增加, 识别精度随之降低, 导致其动态时间规整路径无法趋向于对角线, 为提高手势动作的识别精度和实时性, 我们对 DTW 算法做出如下改进:

规整路径的累计值如式(7)可以计算得出, 主要用来衡量和计算两条时间序列的相似度, 为了使规整路径更加趋向于对角线, 文中进行了如下改进:

$$s(i, j) = d(L_i, C_j) + \min \left\{ \begin{array}{l} \sigma_1 s(i-1, j) \\ \sigma_2 s(i, j-1) \\ \sigma_3 s(i-1, j-1) \end{array} \right\} \quad (9)$$

其中, $\sigma_1, \sigma_2, \sigma_3$ 为改进 DTW 算法的优化系数, 根据图 7 所示, 要想使路径趋近与对角线, 则应使 $s(i-1, j-1)$ 大于 $s(i-1, j)$ 和 $s(i, j-1)$, 理想状态下, 假设当 $s(i-1, j)$ 和 $s(i, j-1)$ 均为 1 时, 则 $s(i-1, j-1)$ 应为 $\sqrt{2}$, 因此若想使路径尽可能的趋向对角线的概率变大, 应使 $s(i-1, j)$ 和 $s(i, j-1)$ 中的最大值大于 $\sqrt{2}s(i-1, j-1)$, 所以在无法比较 $s(i-1, j)$ 和 $s(i, j-1)$ 的前提下, 分别对 σ_1 和 σ_2 取大于 $\sqrt{2}$ 小数倍, 所以本文中的优化系数 $\sigma_1, \sigma_2, \sigma_3$ 的值分别为 1.5, 1.5, 1。为验证改进 DTW 算法的有效性和准确性, 文中针对左滑手势时间序列的规整路径进行了 DTW 算法改进前后的实验对比, 如图 11 所示:

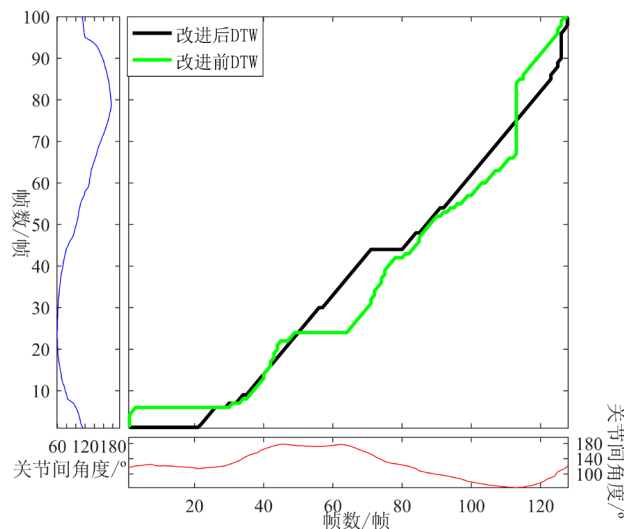


Figure 11. Comparisons of regular path before and after improvement of DTW left-sliding gesture
图 11. 左滑手势 DTW 算法改进前后规整路径对比结果

由图 11 所示, 改进后的 DTW 算法比改进前的 DTW 算法更接近于对角线, 相对于改进前的 DTW 算法, 改进后的 DTW 算法计算量较少, 用来存储数据的空间需要的也更少, 能够更快的收敛, 有效的提升了识别算法的效率, 提高了手势识别的准确性和实时性。

7. Unity3D 虚拟现实平台实时交互实验验证及结果分析

7.1. Unity3D 虚拟显示平台实时交互实验

实验中共定义了上滑, 下滑, 左滑, 右滑, 左转, 右转, 放大, 缩小这 8 种手势, 为了验证本文中

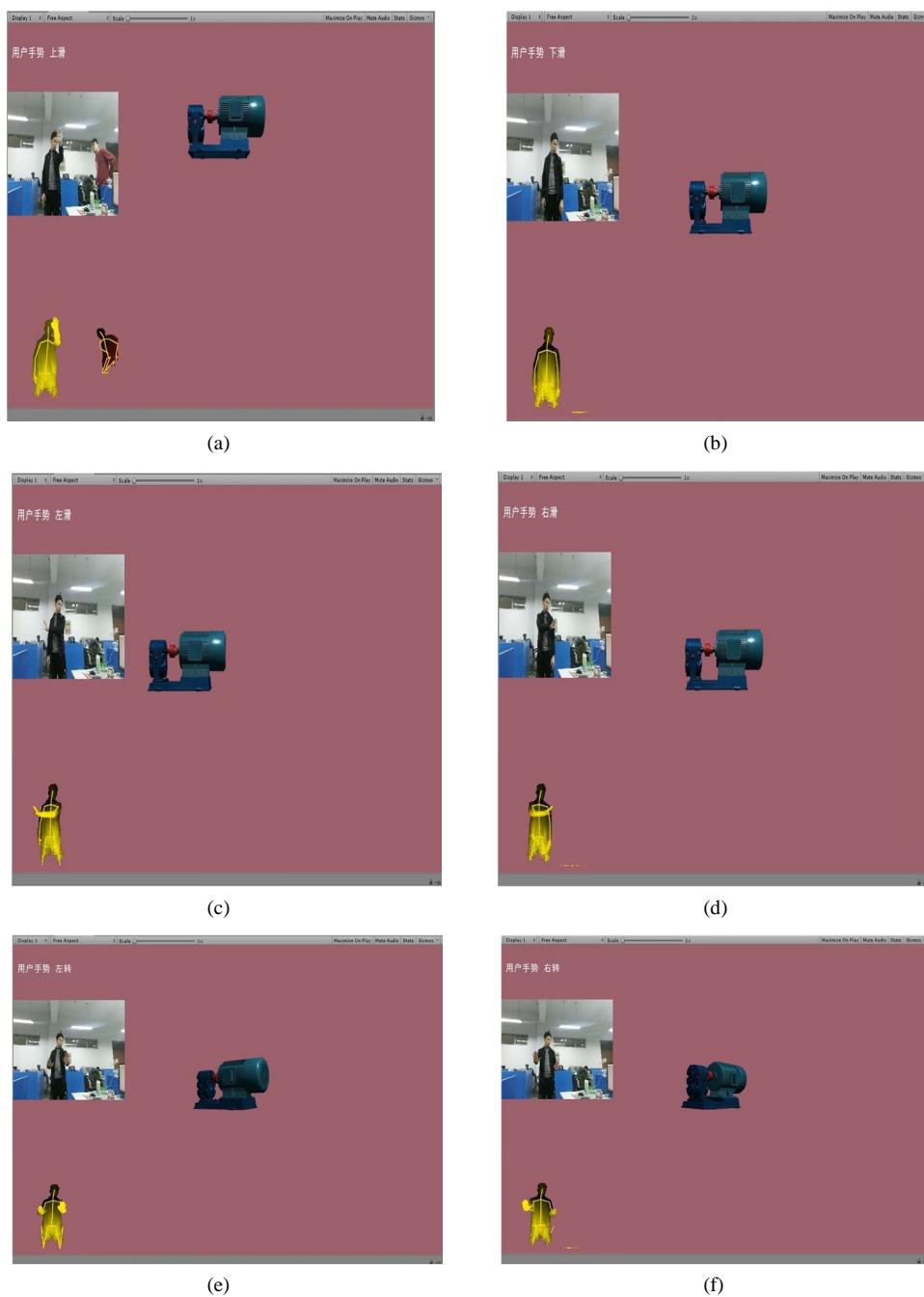
手势的实时性和有效性，将定义和训练好的手势动作向虚拟现实中的单体设备发出控制命令，使单体设备响应动作，从而完成与虚拟现实场景的实时交互，手势控制指令与动作响应对照表如表 2 所示。

Table 2. Control instruction and action response table

表 2. 控制指令与动作响应表

控制指令	上滑	下滑	左滑	右滑	左转	右转	缩小	放大
动作响应	向上移动	向下移动	向左移动	向右移动	顺时针转动	逆时针转动	缩小视角	放大视角

虚拟场景中的单体设备对手势控制指令的响应效果如图 12 所示：



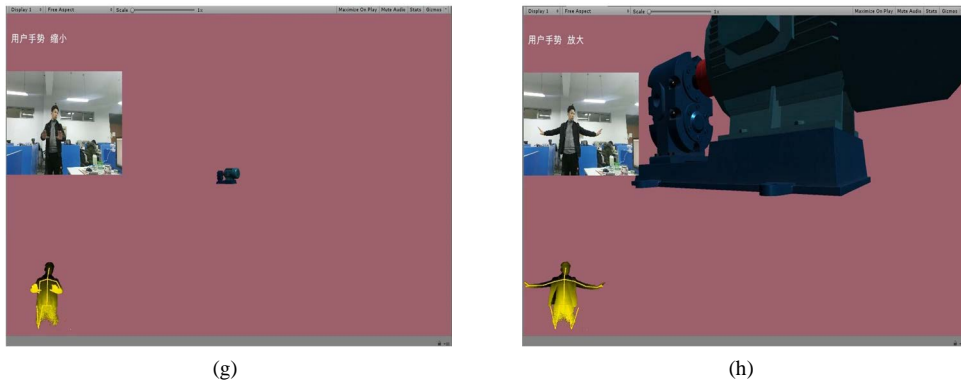


Figure 12. Comparisons of regular path before and after improvement of DTW left-sliding gesture. (a) Swipe up gesture response; (b) Swipe down gesture response; (c) Swipe left gesture response; (d) Swipe right gesture response; (e) Turn left gesture response; (f) Turn right gesture response; (g) Zoom out gesture response; (h) Zoom in gesture response

图 12. 左滑手势 DTW 算法改进前后规整路径对比结果。(a) 上滑手势响应; (b) 下滑手势响应; (c) 左滑手势响应; (d) 右滑手势响应; (e) 左转手势响应; (f) 右转手势响应; (g) 缩小手势响应; (h) 放大手势响应

7.2. 实验结果分析

这种通过 Kinect 自己建立数据库的方法，能够训练多种复杂的连续手势，能满足用户的各种手势需求，使用户进行人机交互时更加方便，自然，本文通过提取关键骨骼点间的角度序列作为特征，利用提出的加权 DTW 算法进行分类和识别，提高了识别率的准确性和实时性，各个手势的识别率如表 3 所示，与黄东方等人[12]，薛俊杰等人[13]和阮晓钢等人[14]的研究算法对比结果如图 13 所示。

Table 3. Gesture recognition rate

表 3. 各手势识别率

手势	实验次数	误检数	识别率
上滑	50	3	94%
下滑	50	6	88%
左滑	50	5	90%
右滑	50	2	96%
左转	50	1	98%
右转	50	3	94%
缩小	50	1	98%
放大	50	4	92%

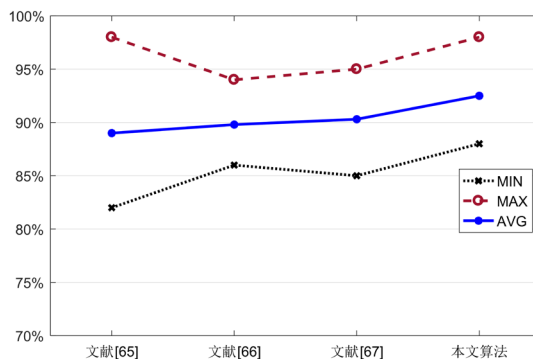


Figure 13. Comparison of recognition rate with other literature algorithms

图 13. 与其他文献算法识别率对比

为验证文中改进 DTW 算法的实时性, 文中选择对其中 6 种手势分别采用了 HMM 算法, 矢量特征+DTW 算法和多项式时间算法+DTW 算法进行手势处理时间的对比, 每种手势测试 50 次, 然后计算每种手势每次的平均处理时间, 单位为 ms/次, 对比结果如表 4 所示:

Table 4. Real-time comparison of algorithms

表 4. 各算法实时性对比

平均处理时间/ms	HMM 算法	矢量特征+DTW 算法	多项式时间算法+DTW 算法	本文改进 DTW 算法
上滑	242 ms	182ms	103 ms	29 ms
下滑	217 ms	165 ms	112 ms	23 ms
左滑	226 ms	174 ms	104 ms	48 ms
右滑	251 ms	159 ms	126 ms	34 ms
缩小	302 ms	201 ms	137 ms	54 ms
放大	281ms	194 ms	142ms	43 ms

8. 结论

由表 3 和图 13 可知, 本文中对定义的手势的平均识别率为 92.5%, 本文算法的平均识别率与黄东方等人的算法识别率相比高 4.5 个百分点, 高于贺霄琛等人的算法 2.7 个百分点, 高于阮晓钢等的算法 2.2 个百分点, 由图 13 可知, 本文中的算法识别率高于其他参考文献算法的识别率, 由表 4 可以看出, 本文中的改进 DTW 算法对每个手势的处理时间远远少于其他各算法, 有较好的实时性。通过以上各项试验数据和交互实验结果可以看出本文中的研究方法能够取得良好的实验效果, 实验方案有效可行。

基金项目

天津职业大学科学研究基金项目(20121101); 国家自然科学基金(F2012203111)资助项目; 河北省高等学校科学技术研究青年基金项目(2011139)。

参考文献

- [1] 刘刚, 袁纪武, 李磊, 等. 基于增强现实的石化 DCS 人机交互方法研究[J]. 计算机与应用化学, 2013, 30(12): 1431-1434.
- [2] Fujino, M. and Zin, T.T. (2016) Action Recognition System with the Microsoft KinectV2 Using a Hidden Markov Model. 2016 *Third International Conference on Computing Measurement Control and Sensor Network (CMCSN)*, Matsue, Japan, 20-22 May 2016, 118-121. <https://doi.org/10.1109/CMCSN.2016.49>
- [3] 郭晓利, 杨婷婷, 张雅超. 基于 Kinect 深度信息的动态手势识别[J]. 东北电力大学学报, 2016, 36(2): 90-94.
- [4] Saha, S., Lahiri, R., Konar, A., et al. (2017) HMM-Based Gesture Recognition System Using Kinect Sensor for Improved Human-Computer Interaction. *International Joint Conference on Neural Networks*, Anchorage, AK, USA, 14-19 May 2017, 2776-2783. <https://doi.org/10.1109/IJCNN.2017.7966198>
- [5] 李凯, 王永雄, 孙一品. 一种改进的 DTW 动态手势识别方法[J]. 小型微型计算机系统, 2016, 37(7): 1600-1603.
- [6] Ruan, X. and Tian, C. (2015) Dynamic Gesture Recognition Based on Improved DTW Algorithm. *IEEE International Conference on Mechatronics & Automation*, Beijing, China, 2-5 August 2015, 2134-2138. <https://doi.org/10.1109/ICMA.2015.7237816>
- [7] Biswas, K.K. and Basu, S.K. (2012) Gesture Recognition Using Microsoft Kinect®. *The 5th International Conference on Automation*, 6-8 December 2011, 100-103. <https://doi.org/10.1109/ICARA.2011.6144864>
- [8] Fornaser, A., Tomasin, P., De Cecco, M., et al. (2017) Automatic Graph Based Spatiotemporal Extrinsic Calibration of Multiple Kinect V2 ToF Cameras. *Robotics & Autonomous Systems*, **98**, 105-125. <https://doi.org/10.1016/j.robot.2017.09.007>

- [9] Raheja, J.L., Minhas, M., Prashanth, D., *et al.* (2015) Robust Gesture Recognition Using Kinect: A Comparison Between DTW and HMM. *Opti—International Journal for Light and Electron Optics*, **126**, 1098-1104.
<https://doi.org/10.1016/j.ijleo.2015.02.043>
- [10] Shotton, J., Girshick, R., Fitzgibbon, A., *et al.* (2013) Efficient Human Pose Estimation from Single Depth Images. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **35**, 2821-2840.
<https://doi.org/10.1109/TPAMI.2012.241>
- [11] Zhao, Y., Liu, Y., Dong, M., *et al.* (2016) Multi-Feature Gesture Recognition Based on Kinect. *IEEE International Conference on Cyber Technology in Automation*, Chengdu, China, 19-22 June 2016, 392-396.
<https://doi.org/10.1109/CYBER.2016.7574856>
- [12] 黄东方, 杨晶东. 基于改进 ND-DTW 算法的动态手势识别[J]. 电子科技, 2017, 30(3): 37-40.
- [13] 贺霄琛, 韩燮, 李顺增. 改进的 LB 算法在动态手势识别中的应用[J]. 微电子学与计算机, 2016, 33(4): 55-59.
- [14] 阮晓钢, 田重阳, 李望博. 基于 Kinect 视觉功能的机器人控制方法[J]. 北京工业大学学报, 2016, 42(4): 487-491.