

基于手势帧序列提取的自适应 实时手势分类算法

林 琳^{1,2}, 陈雨欣¹, 仉威至^{1,2}

(1. 吉林大学 通信工程学院, 长春 130021; 2. 吉林大学 吉林省智能信号识别和设备工程技术研究中心, 长春 130021)

摘 要: 为了提高手势识别算法对实时手势的处理效率, 提出了一种基于手势帧序列提取的自适应实时分类算法。该算法对实时提取的当前帧图像数据进行帧序列延展, 与顺序提取的多帧序列共同完成分类任务, 并进行联合判别, 可以同时识别静态手势和动态手势, 快速获取实时识别结果。本文算法占用较少的计算资源缩短识别时间, 解决了手势识别的延迟问题, 提高了实时性。通过手势识别实时性实验测试, 本文算法使手势平均识别时间控制在 0.4 s 以内。

关键词: 通信与信息系统; 手势实时识别; 人机交互; 帧序列提取

中图分类号: TP391.41 **文献标志码:** A **文章编号:** 1671-5497(2025)09-3042-07

DOI: 10.13229/j.cnki.jdxbgxb.20240529

Adaptive real-time gesture classification algorithm based on gesture frame sequence extraction

LIN Lin^{1,2}, CHEN Yu-xin¹, NAI Wei-zhi^{1,2}

(1. College of Communication Engineering, Jilin University, Changchun 130021, China; 2. Intelligent Signal Identification Equipment Engineering and Technology Research Center of Jilin Province, Jilin University, Changchun 130021, China)

Abstract: In order to improve the processing efficiency of the gesture recognition algorithm for real-time gestures, an adaptive real-time classification algorithm based on gesture frame sequence extraction was proposed. The algorithm extends the frame sequence of the current frame of image data extracted in real-time, completes the classification task with the sequentially extracted multi-frame sequences and performs the joint discriminant, which can recognize static and dynamic gestures at the same time, and obtains the real-time recognition results quickly. The proposed algorithm accupies less computational resources to shortens the recognition time, solves the delay problem of gesture recognition, and improves the real-time performance. Through the gesture recognition real-time performance experimental test, the proposed algorithm controls the average recognition time of gesture within 0.4 s.

Key words: communications and information systems; real-time gesture recognition; human-computer interaction; frame sequence extraction

收稿日期: 2024-05-14.

基金项目: 国家自然科学基金重点项目(62332003).

作者简介: 林琳(1979-), 女, 副教授, 博士. 研究方向: 模式识别. E-mail: lin_lin@jlu.edu.cn

0 引言

人机交互在当今信息时代必不可少,而使用手势取代传统方式控制计算机,能够使人与机器之间的交互更智能、更自然。手势交互有多种应用场景,如游戏控制、智能家居、手语实时翻译等^[1,2],动态手势和静态手势在这些应用场景中能够实现不同的交互效果。手势识别借助机器学习进行分类,机器学习方法分为深度学习方法和传统方法。采用图像分割等预处理提取有效的手势信息,通过深度学习对其进行识别,这是目前手势识别最常见的实现方法^[3,4],但这类方法对运算资源要求较高,将算法集成至嵌入式系统中的难度较大。

传统机器学习方法包括k最近邻(k-nearest neighbor, KNN)算法、决策树和支持向量机等。相比深度学习方法,使用传统机器学习方法进行手势分类需要更少的计算资源和数据进行训练,但使用这种方法的主要局限性是较长的运行时间和延迟响应^[5]。由于手势识别的延迟会影响实际手势识别系统中用户的体验性^[6],因此系统延迟是手势识别应用中亟待解决的问题。兼顾静态手势和动态手势的识别并降低算法延迟以满足系统的实时性要求,对实际的手势识别系统十分必要。

提高手势识别实时性的常用方法包括硬件加速^[7]、多线程并行处理^[8]和基于视频关键帧提取的手势识别方式^[9,10]。其中,基于视频关键帧提取的手势识别方式在视频流或者长时间录制的视频中,只需对关键帧进行分析,可节省大量计算资源和运行时间,有助于系统快速地处理手势识别任务。相比硬件加速,该方法具有低成本的优势;相比多线程并行处理,更能从根本上解决对计算资源的占用。该方法通过单目或者多目摄像头捕获使用者的手势视频,以视频中局部信息点作为关键特征提取关键帧,并将视频帧中最具代表性的帧作为关键帧进行手势识别。为了减少计算量,针对重复性视频信息,可使用特征匹配法找到视频信息中目标特征相同的两幅帧图像^[11],从而得到一段完整帧序列中的“首-尾”图像。基于视频关键帧提取的手势识别方式中,多数基于帧序列匹配的算法均以顺序提取的包含手势信息的帧序列为基础,与训练集中的样本进行匹配,从而得到识别结果。这种方式需等待算法中所有帧数提取完毕,才能进行识别任务,而在实际手势识别系

统中,等待帧序列提取的过程将占用一定时间,造成实时识别的延迟。

针对上述算法中存在的手势识别延迟问题,本文的目的是当做出手势时,直接对当前帧图像数据进行特征提取,将该帧作为手势分类的“关键帧”,通过对实时提取的当前帧手势特征向量以复制的方式进行延展,构造多帧手势序列。相比现有方法,本文方法无需等待全部帧序列提取完毕就能得到识别结果,且在选择用于识别的帧序列时,占用的计算资源更少、实时性更强。在对关键帧进行手势分类时,为了防止关键帧信息不完整导致识别率降低,并行地对顺序提取的多帧序列进行分类,并重点将当前帧图像信息快速理后得到的分类结果与顺序提取多帧序列的分类结果进行联合判断,从而达到快速计算和准确识别静态、动态手势的效果。

本文利用图像深度数据,以基于随机线段的手势特征提取算法为基础,通过提取手势帧序列,从有限帧序列数据中获得充分图像信息,即时完成对静态和动态手势的分类,并根据实时手势识别场景下的响应时间测试验证算法的有效性。

1 基于随机线段的实时手势轨迹构建算法

基于随机线段的实时手势轨迹构建算法,从随机定位线段上的像素中提取特征,不仅计算速度较快,还具有尺度不变性和旋转不变性。

该算法首先通过深度相机获取实时手势深度图像,然后在手势识别程序中载入并读取预先生成的 M 条随机线段数据,经算法计算特征函数值,并构建手势动作轨迹。算法框图如图1所示。

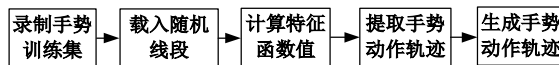


图1 基于随机线段的实时手势轨迹构建算法框图

Fig. 1 Algorithm diagram for real-time gesture trajectory construction based on random line segments

1.1 随机线段特征函数

本文使用的手势识别算法借助随机线段确定手势特征^[12],原因在于随机线段特征跟随手的姿态进行计算,因此具有尺度不变性和旋转不变性,且不受手到摄像机距离的影响。

该算法将 M 条不同长度的随机线段散布在每一帧深度图像的手部区域,以手掌中心为原点、手的轴线为 x 轴,以手腕到距离手部最远处的距离像素值为 1 个单位长度建立坐标系。每条线段通过 4 个参数 $(\epsilon, \beta, \gamma, \delta)$ 进行参数化,并根据每一帧中手的大小和方向进行缩放和旋转。线段在上述坐标系中的中心坐标为 (ϵ, β) , 其中 $\epsilon, \beta \in [-0.5, 0.5]$; 坐标系 x 轴与线段方向的夹角定义为 $\gamma, \gamma \in (-90^\circ, 90^\circ)$; 线段的长度定义为 $\delta, \delta \in (0, 2.8)$ 。深度相机获取的一张深度图像像素用 $I(x)$ 表示,其中的一条随机线段用 $L(\epsilon, \beta, \gamma, \delta)$ 表示,这条线段覆盖深度图像上的所有像素用集合 $A(L)$ 表示。当线段的方向更接近于图像的水平方向时,对应图像上每一列只取一个像素,竖直方向同理。手势部分的像素称为前景像素,用集合 $S(L)$ 表示。为了描述随机线段与前景像素的关联,并为随机线段赋值,每条线段均用 4 个特征函数值中的一个表示。

第 1 种特征函数定义为该线段覆盖的前景像素占有所有覆盖像素的比例,即:

$$f_{(1)}(I; L) = \frac{|S(L)|}{|A(L)|} \quad (1)$$

第 2 种特征函数定义为该线段覆盖的前景像素的平均深度值,即:

$$f_{(2)}(I; L) = \frac{1}{|S(L)|} \sum_{x \in S} D(x) \quad (2)$$

式中: $D(x)$ 为原始深度图像 $I(x)$ 的一个简单变换。

设 I_a 为前景像素的平均深度值,则 $D(x)$ 为将 $I(x)$ 的值域 $[I_a - 100, I_a + 100]$ 映射到 $[0, 1]$ 后的函数,即:

$$D(x) = \frac{I(x) - I_a + 100}{200} \quad (3)$$

式中: $I(x)$ 的单位是毫米。

用 $\hat{x}$ 表示线段 L 上深度值最小的点,即:

$$\hat{x} = \arg \min_{x \in S(L)} D(x) \quad (4)$$

第 3 种特征函数表示该点在线段上的位置,即:

$$f_{(3)}(I; L) = \frac{|R(L)|}{|A(L)|} \quad (5)$$

式中: $R(L)$ 为 L 上 x 坐标值 $< \hat{x}$ 的点,即:

$$R(L) = \{x \in A(L) | x_1 < \hat{x}_1\} \quad (6)$$

式中: x_1 为 $\hat{x}$ 的 x 轴坐标值。由于 γ 的取值不包括 $\pm 90^\circ$, 因此线段 L 不会垂直于 y 坐标轴。

第 4 种特征函数用于度量线段上前景区域深度值的波动性,其定义为沿着该线段的深度值绝对变化值的总和,即:

$$f_{(4)}(I; L) = \int_L \left| \frac{dD(x)}{dl} \right| dl \quad (7)$$

第 4 种特征函数可以用线段上相邻前景像素的深度差绝对值之和进行估算,即:

$$f_{(4)}(I; L) = \sum_{(x_1, x_2) \in N} |D(x_1) - D(x_2)| \quad (8)$$

式中: $N = \{(x_1, x_2) | x_1, x_2 \in S, x_1 \text{ neighbors } x_2\}$, 当随机线段被载入深度图像后,每条线段对应一个特征函数值。当手面向摄像机前后移动时, $D(x)$ 原理上不会发生改变,故这 4 个特征函数值原理上也不会发生改变,因为所有的前景像素在计算 4 种特征函数值时起到均等的作用,所以图像上轻微的噪声对特征函数值的影响较小。

1.2 手势动作轨迹特征提取

1.2.1 手势动作轨迹构建

(1) 当前帧手势动作轨迹构建

提取实时获取的深度视频当前帧的手势特征向量,并对该手势特征向量进行延展,即将当前第 s 帧的特征向量 $\mathbf{p}^{(s)} = [f_1^{(s)}, f_2^{(s)}, \dots, f_M^{(s)}]$ 进行复制,得到与样本帧数相同的 N 帧特征向量,用这些向量构建一条动作轨迹 $T_0 = [\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots, \mathbf{p}^{(N)}]$, 其中特征向量 $\mathbf{p}^{(s)}$ 的个数为 N 个。构建过程如图 2 所示。

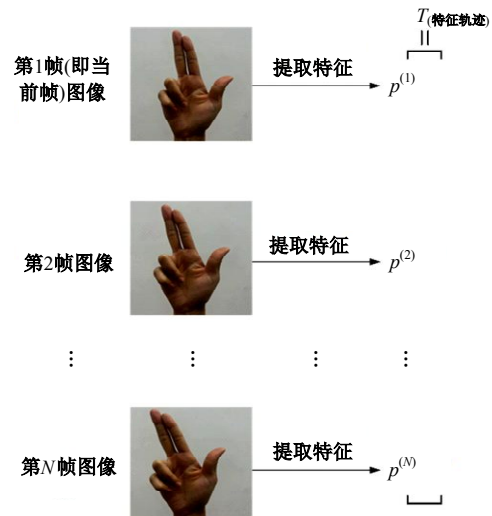


图 2 手势特征轨迹构建示意图

Fig. 2 Schematic diagram of gesture feature trajectory construction

(2) N 帧手势动作轨迹构建

顺序提取实时获取的深度视频帧序列中包括(1)中当前帧在内的共 N 帧手势特征向量,并构成动作轨迹 $T_{00}=[p^{(1)}, p^{(2)}, \dots, p^{(N)}]$ 。

1.2.2 动作轨迹特征

手势动作轨迹使用以下3种特征表示:

第1种特征是三维距离特征,用于描述深度视频帧序列中动作轨迹上的特征点在三维空间中的距离,可以表示出手势的外观差异。 T_a 和 T_b 两个序列在第 n 帧特征向量 $p_a^{(n)}$ 和 $p_b^{(n)}$ 的距离定义为:

$$d_1(p_a^{(n)}, p_b^{(n)}) = \frac{1}{M} \sum_{m=1}^M (|f_{m,a}^{(n)} - f_{m,b}^{(n)}|) \quad (9)$$

式中: $f_{m,a}^{(n)}$ 和 $f_{m,b}^{(n)}$ 分别为特征向量 $p_a^{(n)}$ 和 $p_b^{(n)}$ 在 M 个数值中的第 m 个值。

第2种特征是运动长度特征,可以表示出手势在空间中随时间变换的位置,其计算公式为:

$$L(T) = \frac{1}{N-1} \sum_{n=1}^{N-1} \sqrt{\sum_{m=1}^M (f_m^{(n)} - f_m^{(n+1)})^2} \quad (10)$$

第3种特征是速度特征,通过计算手势运动速度得到。

2 手势样本阈值计算

通过计算机键入分类标签,并录制每个分类标签对应的多个深度视频帧序列作为训练集。每个深度视频帧序列包含 N 帧,为一个样本,分类标签相同的样本为同类样本。对训练集进行动作轨迹构建后,计算手势样本阈值。

手势标签 $\text{Label}(T_i)$ 是采集手势样本时使用计算机键入的分类标签,包含阿拉伯数字0~9,其中 T_i 为样本的手势序列。手势标签对应的阈值为 $\text{Th}_q(\text{Label}(T_i))$,其计算方法为:计算训练集中所有标签相同的样本两两之间的3种特征差值的中位数,并乘以阈值系数 α ,即:

$$\text{Th}_q(\text{Label}(T_i)) = \alpha \cdot \text{median} D_q(T_a, T_b) \quad (11)$$

式中: $D_q(T_a, T_b)$, $q \in \{1, 2, 3\}$ 分别为三维距离特征差、运动长度特征差、速度特征差,分别用于衡量两个深度视频手势序列间动作轨迹的三维距离特征、运动长度、速度差异程度。

三维距离特征差为两个特征向量之间的欧氏距离,即:

$$D_1(T_a, T_b) = \frac{1}{N} \sum_{n=1}^N d_1(p_a^{(n)}, p_b^{(n)}) \quad (12)$$

式中:序列 T_a 和 T_b 为需进行特征差计算的两个手

势动作轨迹。

运动长度特征差计算公式为:

$$D_2(T_a, T_b) = \frac{L(T_L)}{L(T_S)} \quad (13)$$

式中: T_L 和 T_S 分别为动作轨迹 T_a 和 T_b 中长度特征值较大和较小的轨迹。

速度特征差计算公式为:

$$D_3(T_a, T_b) = \frac{1}{N} \sum_{n=1}^N d_3(v_a^{(n)}, v_b^{(n)}) \quad (14)$$

式中: $v_a^{(n)}$ 和 $v_b^{(n)}$ 分别为手势动作轨迹 T_a 和 T_b 在第 n 帧中的移动速度。

3 基于帧序列提取的自适应手势分类算法

基于帧序列提取的自适应手势分类算法,借助手势样本阈值和手势样本特征差实时对手势进行分类并输出识别结果。该算法根据手势轨迹 T_0 和 T_{00} 在手势训练集中的样本匹配情况,调整系统识别结果的输出,因此具有一定的自适应性。

3.1 手势动作轨迹 T_0 分类

该算法根据获取的实时手势图像,用1.2.1小节中基于当前帧手势特征向量构建的手势动作轨迹 T_0 ,按式(12)~式(14)计算样本特征差 $D_q(T_0, T_1)$, $q = \{1, 2, 3\}$ 。其中,序列 T_1 为实时测试前采集的训练集中样本的手势动作轨迹。当特征差 $D_q(T_0, T_1)$, $q = \{1, 2, 3\}$ 满足式(15)~式(17)的条件时,判定手势动作轨迹 T_0 与样本的手势动作轨迹 T_1 匹配。

$$D_1(T_0, T_1) < \text{Th}_1(\text{label}(T_1)) \quad (15)$$

$$D_2(T_0, T_1) < \text{Th}_2(\text{label}(T_1)) \quad (16)$$

$$D_3(T_0, T_1) < \text{Th}_3(\text{label}(T_1)) \quad (17)$$

式中: $\text{label}(T_1)$ 为分类标签。每个手势标签对应的阈值 $\text{Th}_q(\text{label}(T_1))$ 通过该分类标签下的样本按式(11)计算得到。

在实时识别时,每次深度相机捕获到最新一帧后,按式(15)~式(17)将构建的动作轨迹对应序列与训练集中的所有样本深度视频帧序列进行比较,若存在一个或多个样本序列与该序列匹配,则将对应的 $1/D(T_0, T_1)$ 作为每个匹配样本的分数,使用KNN算法输出累积分数最大的标签作为分类结果1。其中:

$$D(T_0, T_1) = 0.9D_1(T_0, T_1) + 0.05D_2(T_0, T_1) + 0.05D_3(T_0, T_1) \quad (18)$$

式中: $D(T_0, T_1)$ 为序列 T_0 与样本序列 T_1 的三维距离特征差、运动长度特征差、速度特征差的加权平均值。

3.2 手势动作轨迹 T_{00} 分类

对按照顺序实时获取的 N 帧特征向量构建的动作轨迹 T_{00} ,按式(12)~式(14)计算样本特征差 $D_q(T_{00}, T_2)$, $q=\{1, 2, 3\}$ 。其中,序列 T_2 为实时测试前采集的训练集中样本手势动作轨迹。当特征差 $D_q(T_{00}, T_2)$, $q=\{1, 2, 3\}$ 满足式(19)~式(21)的条件时,判定轨迹 T_{00} 与样本轨迹 T_2 匹配。

$$D_1(T_{00}, T_2) < Th_1(\text{label}(T_2)) \quad (19)$$

$$D_2(T_{00}, T_2) < Th_2(\text{label}(T_2)) \quad (20)$$

$$D_3(T_{00}, T_2) < Th_3(\text{label}(T_2)) \quad (21)$$

式中: $\text{label}(T_2)$ 为分类标签。每个手势标签对应的阈值 $Th_q(\text{label}(T_2))$ 通过该分类标签下的样本按式(11)计算得到。

在实时识别时,按式(19)~式(21)将最近 N 个深度视频帧序列与训练集中的所有样本深度视频帧序列进行比较,若存在一个或多个样本序列与该序列匹配,则将对应的 $1/D(T_{00}, T_2)$ 作为每个匹配样本的分数,使用KNN算法输出累积分数最大的标签作为分类结果2。其中:

$$D(T_{00}, T_2) = 0.9D_1(T_{00}, T_2) + 0.05D_2(T_{00}, T_2) + 0.05D_3(T_{00}, T_2) \quad (22)$$

式中: $D(T_{00}, T_2)$ 为序列 T_{00} 与样本序列 T_2 的三维距离特征差、运动长度特征差、速度特征差的加权平均值。

3.3 自适应手势分类规则

自适应手势分类算法根据序列 T_0 和 T_{00} 在手势训练集中的样本匹配实际情况进行判别,使系统输出的识别结果与实时手势的相似程度更高,根据匹配情况分为以下3种情况:

(1)若序列 T_0 和 T_{00} 在手势训练集中均没有匹配的序列,则算法不输出分类结果。

(2)若仅有 T_0 有匹配的序列 T_1 ,而 T_{00} 没有匹配的序列,则将分类结果1作为识别结果输出;若仅有 T_{00} 有匹配的序列 T_2 ,而 T_0 没有匹配的序列,则将分类结果2作为识别结果输出。

(3)若序列 T_0 和 T_{00} 在手势训练集中均有匹配的序列,则将两个分类结果进行比较,将与样本相差更小的分类结果作为最终手势分类结果输出。当 $D(T_0, T_1) \leq D(T_{00}, T_2)$ 时,输出分类结果1;否则,输出分类结果2。

4 仿真实验

4.1 实验条件

本实验均在由Intel RealSense D435深度相机和计算机构成的实验环境中完成。本实验使用的计算机系统为Windows 10,搭载Intel(R)Core(TM) i7-7700处理器,CPU主频为3.60 GHz。

由一名测试者A学习使用如图3所示10种手势,按照从左到右、从上到下的顺序为手势分配0~9号标签。由于本文算法主要针对静态手势识别延迟进行设计,因此仅选用一种动态手势进行算法实验,其中5号手势为左右摇摆的动态手势,其余均为静态手势。

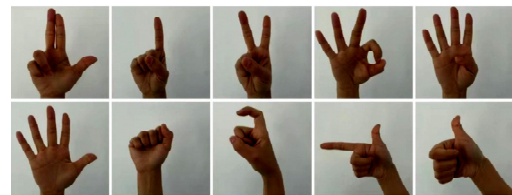


图3 0~9号手势图

Fig. 3 Hand signals 0~9

每种手势在手掌竖直状态下,在手掌根部水平线以上的范围内自由变换角度,每种手势重复15次,共获得150个深度视频作为本实验的训练集。本实验使用1 000条随机线段^[13]用于手势特征提取,每个样本的帧序列长度为30帧。在这种条件下进行手势识别实验,手势识别效果更好。

4.2 阈值系数 α 对识别率的影响

在0.5~2.5范围内分别取 α 值为0.5、0.7、0.9、1.1、1.3、1.5、1.7、1.9、2.1、2.3、2.5。本实验令测试者在一段时间内做出手势,以测试基于手势帧序列提取的自适应实时分类算法^[14]。在不同阈值系数下,由B、C、D、E、F 5名测试者分别在3 min内按照任意顺序和次数做出10种手势,计算每位测试者的手势识别率 R :

$$R = \frac{T}{T + W_1 + W_2} \quad (23)$$

式中: T 为被系统正确识别的手势个数; W_1 为做出手势未被正确识别的手势数量; W_2 为没有做出任何手势时系统的误报个数。

计算5名测试者手势识别率的平均值,作为该阈值系数下的手势识别率,绘制如图4所示的手势识别率随阈值系数变换的曲线。

根据实验结果观测,当阈值系数值 <1.5 时,系统的错误更多来源于做出手势却没有匹配到任

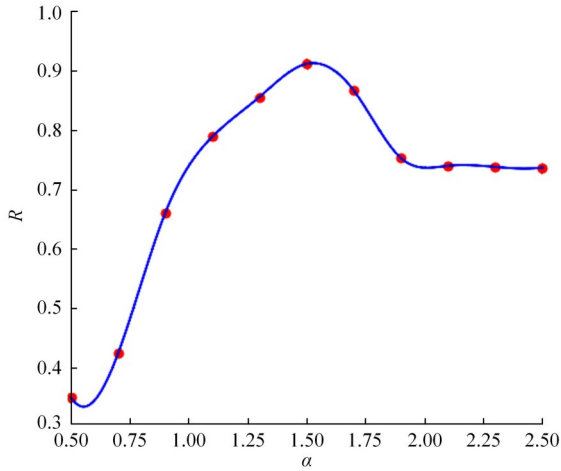


图4 手势识别率随阈值系数变换曲线图

Fig. 4 Gesture recognition rate with threshold coefficient transformation plot

何样本,故而没有输出识别结果;当阈值系数值 >1.5 时,系统的错误更多来源于没有做出手势却输出了识别结果,做出手势时的识别结果大多数为正确。由于本实验采用的方式是让测试者在固定时间内做出多个手势,并未规定需要做出手势的总数量,而5名测试者都倾向于在规定时间内做出尽量多的手势,导致没有做出手势的时间较短,因此当阈值系数值 >1.5 时,手势识别率降低至一定值时趋于稳定。由曲线变换趋势可以发现,当阈值系数值在1.5附近时,算法的识别率最高,为91.2%。此时,系统既没有出现做出手势却没有匹配到任何样本的情况,也较少出现没有做出手势却输出了识别结果的情况,因此,本文将算法的阈值系数值确定为1.5。此时,5号动态手势的识别率为92.1%,其余静态手势的识别率为90.3%,系统实现了对静态手势和动态手势的同时识别。

4.3 算法实时性测试

为验证自适应分类算法对手势识别实时性能的提升效果,本节设计实验对自适应算法和非自适应算法的手势识别响应时间进行测试及对比,两种算法示意图如图5所示。其中,非自适应算法是指通过实时获取的30帧图像数据,仅用1.2.1小节中构建的动作轨迹 $T_{00}=[p^{(1)}, p^{(2)}, \dots, p^{(30)}]$ 按照3.2节中的分类方法完成手势识别的算法,该算法以分类结果2作为最终识别结果;自适应算法则是联合当前帧图像数据和实时获取的30帧图像数据,分别构建包含30个 $p^{(s)}$ 的动作轨迹 $T_0=[p^{(s)}, p^{(s)}, \dots, p^{(s)}]$ 和动作轨迹 $T_{00}=[p^{(1)}, p^{(2)}, \dots, p^{(30)}]$ 进行手势识别的算法,并按照3.3节

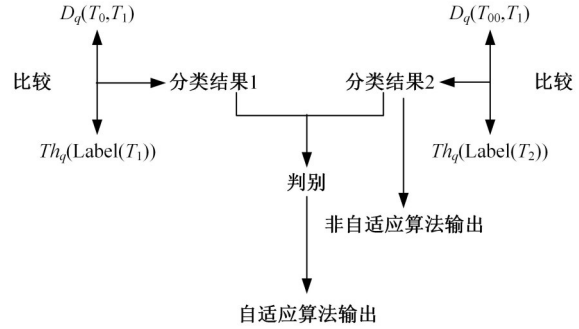


图5 自适应算法及非自适应算法示意图

Fig. 5 Schematic diagram of adaptive and non-adaptive algorithms

的自适应手势分类规则对分类结果1和分类结果2进行判别,得到最终识别结果。

测试方法如下:让一名测试者做出10种手势,每种手势做出1次,整个过程重复10次,使用测试相机录下手势识别过程,将测试相机的录像拷贝到PC上,用Avidemux软件逐帧检查该录像,分别记录用户做完10种手势动作的视频时间 t_1 和屏幕开始显示识别结果标签的视频时间 t_2 ,则 t_2-t_1 为识别响应时间。例如,做出某手势的时间在视频的00:00:00.766处,程序识别出结果并显示在屏幕上的时间在视频的00:00:01.168处,如图6、图7所示。两个时间点做差所得402 ms即本



(a) 做出手势的视频帧时间戳

(b) 显示手势识别结果的视频帧时间戳

图6 做出手势和显示手势识别结果的视频帧时间戳

Fig. 6 Timestamp of the video frame in which the gesture was made and showing gesture recognition results

次识别响应时间,取系统对每种手势响应时间平均值作为10种手势中每种手势的识别响应时间。

对两种算法进行上述测试,测试结果如表1所示,自适应算法的平均识别响应时间为0.38 s,响应性能较非自适应算法提升了约72.3%。

以上实验结果说明,本文算法通过自适应手势分类规则,对当前帧手势特征向量复制及构建的动作轨迹 T_0 进行处理,使手势识别实时性增强,有效缩短了识别时间。

表 1 10 种手势识别响应时间表

Table 1 Ten gesture recognition response schedules

手势 标签	非自适应算法识 别响应时间/s	自适应算法识别 响应时间/s	识别响应时间缩 短比率/%
0	1.33	0.40	69.9
1	1.38	0.37	73.2
2	1.37	0.33	75.9
3	1.40	0.37	73.6
4	1.51	0.47	68.9
5	1.50	0.49	67.3
6	1.20	0.30	75.0
7	1.34	0.43	67.9
8	1.35	0.40	70.4
9	1.21	0.23	81.0

5 结束语

基于手势帧序列提取的自适应实时分类算法通过对实时提取的当前帧手势特征向量进行延展,并与顺序提取的多帧序列进行联合判别,实现了对静态手势和动态手势的识别。实验结果显示,该算法对这两种手势的识别率分别达到 90.3% 和 92.1%。该算法充分利用了当前帧的手势信息,无需在每次做出手势时等待全部帧的手势序列提取完毕就能得到手势识别结果,将平均识别响应时间缩短至 0.38 s,解决了手势识别过程中的延迟问题,增强了手势识别算法的实时性。

参考文献:

- [1] 解迎刚, 王全. 基于视觉的动态手势识别研究综述[J]. 计算机工程与应用, 2021, 57(22): 68-77.
Xie Ying-gang, Wang Quan. Summary of dynamic gesture recognition based on vision[J]. Computer Engineering and Applications, 2021, 57(22): 68-77.
- [2] Ahmed S, Cho S H. Hand gesture recognition using an IR-UWB radar with an inception module-based classifier[J]. Sensors, 2020, 20(2): No. 564.
- [3] Ma M, Chen Z X, Wu J. A Recognition Method of Hand Gesture with CNN-SVM Model[M]. Heidelberg: Springer, 2016.
- [4] Tan Y S, Lim K M, Lee C P. Hand gesture recognition via enhanced densely connected convolutional neural network[J]. Expert Systems with Applications, 2021, 175: No. 114797.
- [5] Witchuda T, Wiranata A, Maeda S, et al. Reservoir computing model for human hand locomotion signal classification[J]. IEEE Access, 2023, 11: 19591-19601.
- [6] Yeh S C, Wu E H K, Lee Y R, et al. User experience of virtual-reality interactive interfaces: a comparison between hand gesture recognition and joystick control for XRSPACE MANOVA[J]. Applied Sciences, 2022, 12(23): No. 12230.
- [7] 赵忠宇, 高博, 龙行锐, 等. 实时 FPGA 手势识别算法的设计[J]. 电子测量技术, 2020, 43(20): 89-92.
Zhao Zhong-yu, Gao Bo, Long Xing-rui, et al. Design of real-time figure recognition algorithm based FPGA[J]. Electronic Measurement Technology, 2020, 43(20): 89-92.
- [8] 杨建华, 李正, 赵好, 等. 基于肌电信号的嵌入式手势识别系统设计[J]. 自动化与仪表, 2021, 36(12): 62-66.
- [9] Li X, Zhao B, Lu X. Key frame extraction in the summary space[J]. IEEE Transactions on Cybernetics, 2017, 48(6): 1-12.
- [10] Gharbi H, Bahroun S, Zagrouba E. Key frame extraction for video summarization using local description and repeatability graph clustering[J]. Signal, Image and Video Processing, 2019, 13(3): 507-515.
- [11] 邵进达, 杨帅, 程琳. 改进 SIFT 算法结合两级特征匹配的无人机图像匹配算法[J]. 计算机科学, 2019, 46(6): 316-321.
Shao Jin-da, Yang Shuai, Cheng Lin. UAV image matching algorithm based on improved SIFT algorithm and two-stage feature matching[J]. Computer Science, 2019, 46(6): 316-321.
- [12] Nai W Z, Liu Y, Rempel D, et al. Fast hand posture classification using depth features extracted from random line segments[J]. Pattern Recognition, 2017, 65: 1-10.
- [13] Nai W, Liu Y, Wang Q, et al. Learning end-user customized mid-air hand gestures using a depth image sensor[J]. IEEE Sensors Journal, 2022, 22(17): 16994-17004.
- [14] Maqueda A I, del-Blanco C R, Jaureguizar F, et al. Human-computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns[J]. Computer Vision and Image Understand, 2015, 141: 126-137.