

基于 Kinect 的动态手势识别算法改进与实现^①

李国友^② 孟 岩^③ 闫春玮 宋成全

(燕山大学电气工程学院 秦皇岛 066004)

摘 要 为解决隐马尔科夫模型(HMM)算法在动态手势识别中存在的准确率不高、容易受未定义手势的干扰、难以适应复杂背景等问题,本文提出了一种基于 Kinect V2 传感器改进的 HMM 动态手势识别方法。该方法进行手势分割后,以质心运动轨迹正切角的值进行均匀量化编码,通过设置概率阈值模型及编码的种类来排除未定义手势、进行动态手势识别,并对比不同实验环境下的识别效果。实验结果表明,改进后的 HMM 算法有效地排除了多种未定义手势,能够适应复杂背景和黑暗条件,而且能够提高对已定义手势的识别率。

关键词 Kinect V2 传感器, 动态手势识别, 改进隐马尔科夫模型(HMM), 未定义手势, 识别率

0 引 言

手势作为人们日常生活中的习惯交流方式,有着直观、自然的特性。因此,随着计算机技术的不断发展,基于机器视觉的手势识别技术也逐渐成为人机交互领域的研究热点。基于手势的人机交互方式的出现使用户不再局限于鼠标、键盘等传统的人机交互方式,而是一种类似于人与人的,更加自然流畅的交互方式^[1]。由于手势识别交互方式的自然性和灵活性,使其广泛应用于各个领域,例如医学图像、聋哑人辅助生活、远程交流、机器人操作、虚拟仿真、电子游戏及无人驾驶等领域^[2]。

手势识别根据手势运动状态主要分为静态手势识别和动态手势识别^[3]。动态手势识别相对静态手势识别而言,动态手势可以更直观、更方便地表达客户的用途,满足用户的需求^[4]。但在研究过程中也有一定的困难,比如特征复杂、分类困难等。近年来,主要的动态手势学习和识别方法有隐马尔科夫

模型(hidden Markov model, HMM)、动态时间规整(dynamic time warping, DTW)、最长公共子序列(longest common subsequence, LCSS)、K 近邻(K-nearest neighbor, KNN)^[5]等。其中动态时间规整、最长公共子序列和 K 近邻属于模板匹配的方法,隐马尔科夫模型是概率统计的方法^[6]。李凯等人^[7]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 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。Ruan 等人^[8]提出了一种通过失真度阈值和路径约束改进的 DTW 算法,该方法选取 8 个点作为手部特征,并且通过加权距离的方法建立手势的数学模型。动态时间规整法适用于处理特征匹配问题,但相比于 HMM,不能融合所有训练数据的特性,导致其对于模型的代表性不是很好。KNN 虽然可以选择具有代表性的轨迹特征作为模板,但是用户完成一个动作时仍会有较大差异。Nguyen-Dinh 使用最长公共子序列实现了实时的动态手势识别,同时将 LCSS 分别与 DTW 和 SVM 进行了比较,结果表

① 国家自然科学基金(F2012203111)和河北省高等学校科学技术研究青年基金(2011139)资助项目。
② 男,1972 年生,博士,教授;研究方向:工业控制,图像处理,机器视觉,智能控制;E-mail: lgyysu@163.com
③ 通信作者,E-mail: 15733548281@163.com
(收稿日期:2018-11-27)

明,基于 LCSS 的方法能大大降低多人标注引起的标注错误对动态手势识别的影响^[9]。但是每一类手势只能选用一个序列作为模板,而不同用户完成一个动作时只与一个模板对比识别是远远不够的。HMM 是基于当前状态和上一个状态相关的假设模型,符合动态手势识别的情况,在有大量的训练数据时能够取得较好的识别效果^[10]。如郭晓利等人^[11]提出了一种 HMM 与模糊神经网络(FNN)结合的模型来对动态手势进行分类识别,此算法比传统 HMM 模型要稳定,但是还没有应用到实际中,有待在实际环境中进一步检测。Saha 等人^[12]提出了一种特征描述子,用来描叙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,取得了良好的识别效果。本文主要选用 HMM 进行动态手势识别,以往的研究者们虽然对隐马尔科夫模型算法进行了很好的优化和改进,但缺少对未定义手势的判断和排除的研究。

本文中的研究方法主要改进有:一是在图像分割和预处理阶段,将 Kinect V2 传感器进行深度图像前景分割后,再利用基于 YCbCr 空间高斯肤色模型进行手部区域细分;二是计算手掌重心的坐标和水平夹角的正切值,量化成 12 个观察状态,得到观察序列来设置 HMM 模型参数;三是在识别过程中根据概率阈值和编码种类排除未定义手势。该方法对已经定义的向上、向下、向左、向右、圆圈和三角形手势的识别率高达 98% 以上,同时能够有效避免光照和复杂背景的干扰。

1 手势分割及质心轨迹跟踪

1.1 基于 Kinect 深度阈值的前景分割

前景分割就是将 Kinect 获取到的深度数据映射到彩色图像上,然后利用深度阈值过滤掉背景,最后通过深度与彩色的对应关系来得到前景的彩色图像^[13]。前景分割过程如图 1 所示。

如图 1 所示,当用户做手势动作时,用户的身体与手部不在同一个平面内,即身体所在位置与手部所在位置的深度是不相同的。而 Kinect 深度摄像头成像是有限距离限制的,图中用最大阈值表示 Kinect

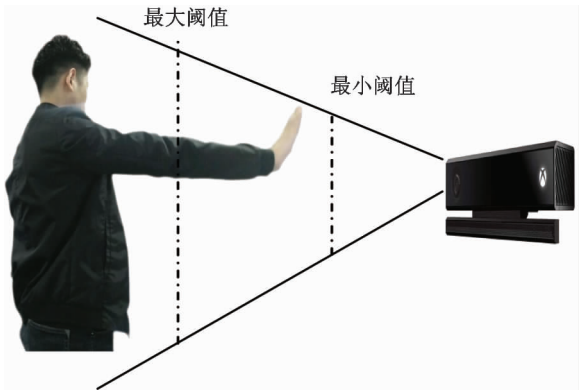


图 1 前景分割过程示意图

最远成像距离,用最小阈值表示 Kinect 最近成像距离。经过 2 次阈值分割,将大于最大阈值和小于最小阈值的部分从人体部分分割,只保留最小阈值和最大阈值之间的数据,这样可以消除人体、前景和背景部分,得到含有手臂部分的手势图像,如图 2 所示。

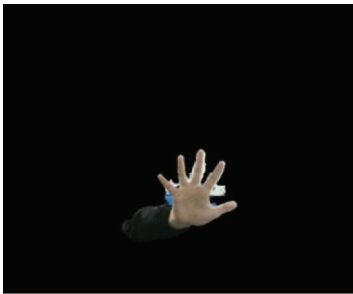


图 2 前景分割结果

1.2 基于 YCbCr 空间的高斯肤色模型手势分割

由图 2 可以看出,分割出的前景图像仍然有手臂部分和不同平面内颜色的影响,因此需要进行手部区域细分。图中肤色的区域比较集中,所以用高斯分布进行描述,为了利用肤色在色度空间的聚类性,选取 YCbCr 彩色空间进行肤色提取。具体肤色分割步骤如下。

1.2.1 肤色相似度计算

根据肤色在色度空间的高斯分布,把彩色图像中的每个像素从 RGB 色彩空间转换到 YCbCr 空间,然后计算该点属于皮肤区域的概率。即根据该点离高斯分布中心的远近来得到与肤色的相似度,将彩色图像转换为灰度图后,灰度图中每个像素的

灰度对应该点与肤色的相似度^[14],相似度的计算公式如下,其中 m 为均值, C 为协方差矩阵。

$$p(CbCr) = \exp[-0.5(x - m)^T C^{-1}(x - m)] \quad (1)$$

$$m = E(x) \quad (2)$$

$$x = (CbCr)^T \quad (3)$$

$$C = E(x - m)(x - m)^T \quad (4)$$

1.2.2 阈值分割

进行阈值分割的目的是将皮肤与背景分开,通常有固定阈值法和自适应阈值法等。本文中考虑到运算肤色与其他颜色区分明显,所以选用固定阈值法进行分割。正常黄种人的 Cr 分量大约在 140 ~ 175 之间, Cb 分量大约在 100 ~ 120 之间。

1.2.3 图像二值化处理

二值化的目的是将皮肤区域与其他区域更好地分割开,肤色区域用“1”(白色)表示,非肤色区域用“0”(黑色)表示,二值化后的手势分割图如图 3 所示。

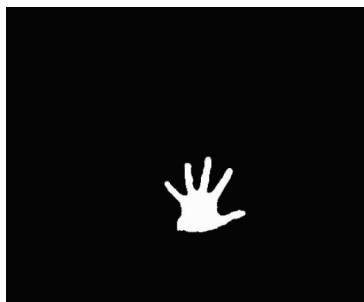


图 3 手势分割二值图

1.2.4 图像处理

由图 3 可以看出,手势二值图边缘存在许多锯齿噪声与凹洞,所以需要对图像进行相关形态学处理和关于噪声的滤波。

(1) 平滑滤波

图像平滑去噪处理作为一种常用的图像处理方式,有着模糊图像和消除图像噪音的作用。空间域的平滑滤波方法一般采用简单平均法进行,就是求邻近像元点的平均亮度值,选择的邻域的大小与平滑的效果直接相关。正常来说,邻域越大,图像的平滑处理效果越好,但如果邻域过大,边缘信息会严重损失,从而使输出的图像变得模糊^[15],因此需合理选择邻域的大小。常见的图像平滑滤波方法有均值

滤波、中值滤波、高斯滤波和双边滤波。其中均值滤波和高斯滤波属于线性滤波的方法,使用了邻域均值的算法,会导致目标边缘模糊。中值滤波和双边滤波属于非线性滤波的方法,均能很好地保护信号边。但是双边滤波处理时间较长,所以本文中采用中值滤波的方式进行图像平滑。

中值滤波计算的实质是一种邻域运算,和卷积运算相类似,但不同的是,中值滤波计算的不是加权求和,而是把邻域中的像素按灰度级进行排序和分组,然后选择该组的中间值作为输出像素值^[16]。用 1 个窗口 W 在图像上扫描、排序,用公式表示为

$$g(m, n) = \text{Median}\{f(m - k, n - l), (k, l) \in W\} \quad (5)$$

通常来说,窗口内像素数为奇数时,就会有 1 个中间像素。若窗内像素数为偶数时,则中值取中间两像素灰度的平均值,本文中选取的内核大小为 9。

(2) 形态学处理

常用的形态学操作包括膨胀、腐蚀、开运算、闭运算和形态学梯度等,考虑到本文中手势图像周围由于锯齿形成的凸起和凹洞,本文采用膨胀和腐蚀交替的形态学操作方式。膨胀的公式定义为

$$A \oplus B = \cup (A + b \mid b \in B) \quad (6)$$

表示图像 A 和卷积核 B 进行卷积计算,将计算出的核 B 覆盖区域的像素点最大值赋给参考点指定像素,从而扩大高亮区域^[17]。

腐蚀是膨胀的取反,将卷积核 B 与图像 A 进行卷积,将核 B 覆盖的区域最小值覆盖给参考点像素,使高亮区域收缩。本文采用的卷积核的参考点位于核中心的 3×3 矩阵,经过平滑处理和形态学处理后的手势二值图如图 4 所示。

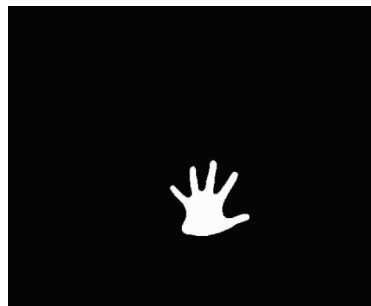


图 4 图像处理后的手势二值图

1.3 手势质心检测与追踪

要画出手势运动的轨迹,首先要得到手势轮廓的质心,质心获取过程如图 5 所示。

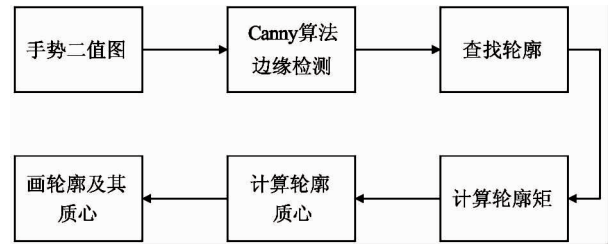


图 5 质心获取过程

对于边缘检测部分, OpenCV 中提供了 3 种算法用于图像的边缘检测,分别是 Canny 算法、Sobel 算法和 Laplacian 算法,其中 Canny 算法因具有低错误率、高定位性和最小响应的边缘检测标准而被很多人认为是边缘检测的最优算法。Canny 算法共分为 5 个步骤。

- (1) 使用高斯滤波器,以平滑图像,滤除噪声。
- (2) 计算图像中每个像素点的梯度强度和方向。
- (3) 应用非极大值抑制(non-maximum suppression),以消除边缘检测带来的杂散响应。
- (4) 应用双阈值(double-threshold)检测来确定真实的和潜在的边缘。
- (5) 通过抑制孤立的弱边缘最终完成边缘检测。

本文中利用 Canny 算法对手势的边缘检测图如图 6 所示。



图 6 手势边缘检测图

手势边缘检测后就可以查找图像中的物体轮廓并计算出轮廓矩和轮廓质心,文中查找轮廓使用的是 OpenCV 中的 FindContours() 函数,该函数检测到

的每个轮廓是以点向量的形式进行存储,即使用 point 类型的 vector 表示,方便下面轮廓矩和轮廓质心计算工作的进行。

对于轮廓矩的计算,在 OpenCV 中,可以很方便地计算多边形区域的 3 阶特征矩,OpenCV 中的矩主要包括空间矩、中心矩和中心归一化矩。其中对于目标物体灰度函数特征矩的公式定义如下:

$$m_{p,q} = \iint x^p y^q f(x, y) d_x d_y \tag{7}$$

如果是二值图像,那么 $f(x, y)$ 就表示为

$$mu_{ji} = \sum_{x,y} (array(x, y) \cdot x^j \cdot y^i) \tag{8}$$

中心矩的公式可表示为

$$mu_{ji} = \sum_{x,y} (array(x, y) \cdot (x - \bar{x})^j \cdot (y - \bar{y})^i) \tag{9}$$

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \tag{10}$$

其中, $(\bar{x}, \bar{y})$ 为轮廓的质心坐标, x, y 即为 vector 中 point 类型点向量的横纵坐标。质心检测结果如图 7、图 8 和图 9 所示。



图 7 质心检测结果

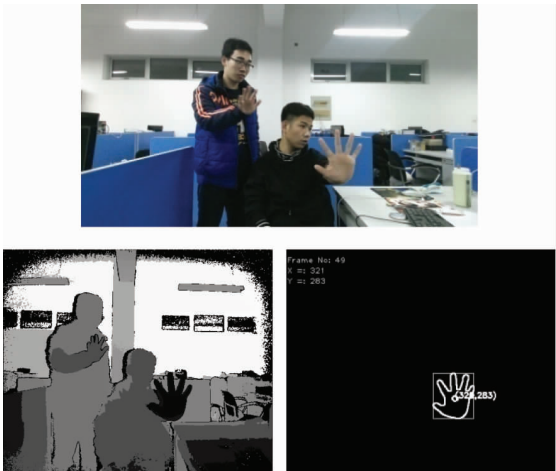


图 8 复杂背景下的质心检测结果



图9 黑暗条件下的质心检测结果

获取质心后,将质心坐标放入 vector 容器中,使用 OpenCV 中的 `cv::line()` 函数将相邻两点的坐标相连形成运动轨迹,如图 10 所示。



图10 质心运动轨迹

2 特征提取及量化编码

在 HMM 特征提取阶段主要有 3 种方法,分别是基于位置的特征提取、基于角度的特征提取和基于速度的特征提取。

(1) 位置特征

位置特征值指的是手势各个观察状态点与手势运动起始点之间的距离,距离 L_s 计算公式如下:

$$L_s = \sqrt{(x_t - x_0)^2 + (y_t - y_0)^2} \quad (11)$$

其中 $t = 1, 2, \dots, T$ 表示观察状态的长度, (x_0, y_0) 表示手势起始点。

(2) 角度特征

动态手势的方向角特征可以分为 3 种:

$$\begin{cases} \theta_1 = \arctan\left(\frac{y_{t+1} - y_t}{x_{t+1} - x_t}\right) \\ \theta_2 = \arctan\left(\frac{y_t - c_y}{x_t - c_x}\right) \\ \theta_3 = \arctan\left(\frac{y_t - y_0}{x_t - x_0}\right) \end{cases} \quad (12)$$

$$(c_x, c_y) = \frac{1}{n} \left(\sum_{i=1}^n x_i, \sum_{i=1}^n y_i \right) \quad (13)$$

其中 (c_x, c_y) 代表第 n 个观察状态点的中心位置, (x_0, y_0) 表示手势起始点坐标, (x_t, y_t) 表示当前观测状态的前一状态坐标, θ_1 表示当前观察状态点与前一观察状态点的方向角, θ_2 表示当前观测状态点与中心点方向角, θ_3 表示当前状态点与起始点方向角。

(3) 速度特征

速度特征是用当前观测状态和前一观测状态质心的欧几里德 (Euclidean) 距离表示的,计算公式如下:

$$L_s = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2} \quad (14)$$

位置特征需要计算每一个观察状态点与起始点的距离,比较耗时,速度特征不能够较好地描述手势轨迹的运动状态,总体来看角度特征最能表征手势轨迹的运动特征^[11]。大量实验表明,角度特征描述运动轨迹过程中,当前观察状态的角度特征与前一状态的方向角贡献最大,且该方向角特征只考虑当前帧和前一帧的运动状态,大大缩短了计算时间,所以本文中以此为特征进行手势运动轨迹特征提取。为了简化计算,提高运算速率,文中采用如图 11 所示的均匀量化编码方式将 360 度平均分成 12 份,得到的角度落在相应的区域,就会输出相应的区间值,这样观察序列就成了对这 12 个数据进行处理,在以后隐马尔科夫的识别中带来极大的方便。

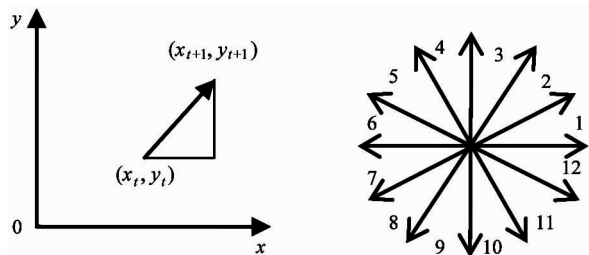


图11 方向角定义及量化编码图

3 手势训练与识别

3.1 隐马尔科夫概念及参数说明

隐马尔科夫模型(HMM)是一个具有双重随机过程的统计模型,是一个随机过程的齐次马尔科夫链,它描述了状态之间的转移规律,是不可观测的,称为状态过程^[12]。另一个堆积过程描述了状态与观察值之间的统计关系,是可观测的,称为观测过程。

在手势运动中每个轨迹序列点的方向状态可以看作是隐马尔科夫的一般随机过程,隐马尔科夫的完整参数可以用一个五元组^[18]表示 $\lambda = \{Y, W, A, B, \pi\}$, 一般情况下可以将 $\lambda = \{Y, W, A, B, \pi\}$ 简化为 $\lambda = \{A, B, \pi\}$ 。各参数说明如下:

(1) Y 代表 HMM 的隐状态个数,用 q_t 表示 t 时刻所处的状态, q_t 的取值范围 $\{S_1, S_2, \dots, S_Y\}$, 则状态序列记为 $Q = q_1, q_2, \dots, q_T$ 。

(2) W 代表 HMM 的观察值个数,用 o_t 表示 t 时刻所处的状态, o_t 的取值范围 $\{V_1, V_2, \dots, V_W\}$, 则状态序列记为 $O = o_1, o_2, \dots, o_T$ 。

(3) $\pi = \{\pi_1, \pi_2, \dots, \pi_Y\}$ 代表初始状态概率矩阵,即状态序列 Q 在初始时刻 $t = 1$ 时所处的状态 S_i 的概率^[19], 可以表示为: $\pi_i = P(q_i = S_i), 1 \leq i \leq Y$, 其中, π_i 满足 $\pi_i \geq 0$ 且 $\sum_{i=1}^Y \pi_i = 1$ 。

(4) $A = \{a_{ij}\}_{Y \times Y}$ 代表状态转移概率矩阵,某个时刻所处的状态 q_t 只与其对应前一时刻所处的状态 q_{t-1} 有关^[20], 所以 A 的元素可以表示为: $a_{ij} = P(q_{t+1} = S_i), 1 \leq i \leq Y, 1 \leq j \leq Y$, 其中, a_{ij} 满足 $\forall i \in [1, Y], a_{ij} \geq 0$ 且 $\sum_{j=1}^Y a_{ij} = 1$ 。

(5) $B = \{b_j(k)\}_{Y \times W}$ 代表观察概率矩阵,其元素可表示为: $b_j(k) = P(o_t = v_k | q_t = S_i), 1 \leq j \leq Y, 1 \leq k \leq W$, 其中, $b_j(k)$ 满足 $\forall i \in [1, Y],$ 有 $b_j(k) \geq 0$ 且 $\sum_{k=1}^W b_j(k) = 1$ 。

3.2 模型选定和初值设定

模型结构的确定和初值的选择对于隐马尔科夫的识别起到很关键的作用,对模型的结构添加适当的约束,在 Baum-Welch 重估时将得到与辨识结果

非常接近的模型参数^[21]。由第 2 节可知本文中选取了基于当前观察状态和前一观察状态的方向角,所以只与当前状态和当前状态的前一状态有关,则可以确定本次手势识别研究中的 HMM 模型。如图 12 所示, S_0 为起始状态, S_f 为终止状态, $S_1, S_{\dots}$ 代表动态手势中不同的观察状态。

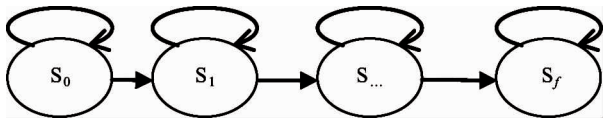


图 12 HMM 模型

此模型中共有 4 个变量需要定义,即初始状态 S_0 , 结束状态 S_f , 隐含状态个数 Y 和观察值个数 W 。在 1.2 节中进行高斯肤色模型分割时划分了 1 个色度通道,当没有手势进入时,背景全部为白色,带有肤色的手势进入时,背景变为黑色,手势区域变为白色,以此来判定是否有手势进入。一旦有手势进入并持续运动超过 10 帧,则判断为初始状态,开始提取并记录手势的轨迹特征,当手势由运动变为静止后持续 10 帧,则判断为手势的终止状态,当手势消失再次进入时,进行下一次的手势识别。理论上来说,对基于 HMM 的手势运动轨迹识别而言,隐含状态的个数越多,识别效果越好。但实验过程中发现,隐含状态数过多会造成采集的特征序列和训练的模型参数矩阵出现很多的 0, 出现数据冗余的现象,影响识别结果。经过测试,本文在满足所有动态手势特征不出现数据冗余的前提下,选择了最大的隐含状态函数,经实验测得这个最大隐含状态数为 6, 即 $Y = 6$ 。选取最大的隐含状态数虽然会耗费一些计算时间,但能够保证较好的识别效果。对于观察值个数,由第 2 节可知,本文中对手势质心运动轨迹的 12 个方向进行了均匀量化编码,所以可以确定每个隐含状态的观察值个数为 12, 即 $W = 12$ 。

下面开始对模型进行初始化,首先对第 1 个参数进行初始化,为了确保它是从第 1 个参数开始的,将其初始化为

$$\pi = (1, 0, 0, 0, 0, 0) \tag{15}$$

第 2 个参数是矩阵 A , 影响 A 的因素是轨迹的持续时间和隐含状态的数目^[22], 具体公式如下:

$$d = \frac{T}{Y} \quad (16)$$

$$a_{ij} = 1 - \frac{1}{d} \quad (17)$$

$$\mathbf{A} = \begin{pmatrix} a_{ij} & 1 - a_{ij} & 0 \\ 0 & a_{ij} & 1 - a_{ij} \\ 0 & 0 & 1 \end{pmatrix} \quad (18)$$

第3个参数是矩阵 $\mathbf{B}$, 因为隐马尔科夫的状态是离散的, 所以所有的矩阵 $\mathbf{B}$ 的不同状态的所有参数被初始化为相同的值。

$$\mathbf{B} = [b_{im}] \quad (19)$$

$$b_{im} = \frac{1}{W} \quad (20)$$

将各个值代入相应的式子中就会得到模型的初始值。

3.3 模型训练

在模型结构和初始化数值之后, 就开始训练隐马尔科夫模型, 训练的完整步骤如下:

(1) 按照上节方式对隐马尔科夫模型参数 $\lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$ 进行初始化, 然后新建1个与 λ 具有相同结构的参数 $\lambda^* = \{\mathbf{A}^*, \mathbf{B}^*, \boldsymbol{\pi}^*\}$, 所有元素都初始化为0^[23]。

(2) 将保存的编码序列值作为训练样本数据输入。

(3) 计算 $P(O | \lambda_n)$, 第一次 λ_n 初始化的参数, 以后就是上一次的值。

(4) 计算 Baum-Welch 算法的2个临时变量。

后验概率函数 $\gamma_t(i)$ 用前向变量 $\alpha_t(i)$ 和后向变量 $\beta_t(i)$ 可表示^[24]为

$$\begin{aligned} \gamma_t(i) &= P(q_t = S_i | O, \lambda) = \frac{P(q_t = S_i | O, \lambda)}{P(O | \lambda)} \\ &= \frac{\alpha_t(i) \times \beta_t(i)}{\sum_{i=1}^Y \alpha_t(i) \times \beta_t(i)} \end{aligned} \quad (21)$$

概率函数 $\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j, O | \lambda)$ 用前向变量 $\alpha_t(i)$ 和后向变量 $\beta_t(i)$ 可表示为

$$\begin{aligned} \xi_t(i, j) &= \frac{P(q_t = S_i, q_{t+1} = S_j, O | \lambda)}{P(O | \lambda)} \\ &= \frac{\alpha_t(i) \times a_{ij} \times b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^Y \sum_{j=1}^Y \alpha_t(i) \times a_{ij} \times b_j(o_{t+1}) \beta_{t+1}(j)} \end{aligned} \quad (22)$$

(5) 按照 Baum-Welch 算法对模型进行重估得到:

$$\bar{\pi} = P(q_1 = S_i) = \gamma_1(i) \quad (23)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (24)$$

$$\bar{b}_j(k) = \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \quad (25)$$

(6) 计算重估之后的参数的概率 $P(O | \bar{\lambda}_n)$, 将2次的值做差, 即 $P(O | \bar{\lambda}_n) - P(O | \lambda_n) < \varepsilon$ (ε 为设定好的收敛阈值), 如果满足, 则认为 $P(O | \bar{\lambda}_n)$ 收敛^[25], 可以认为此时是最接近样本的隐马尔科夫参数, 否则令 $\lambda = \bar{\lambda}$, 继续执行重新估计, 直到收敛为止。

(7) 重复步骤(2) ~ (6), 一直运算到所有的样本都输入。

(8) 对模型参数 $\lambda^* = \{\mathbf{A}^*, \mathbf{B}^*, \boldsymbol{\pi}^*\}$ 进行归一化处理, 即训练完成。

(9) 上述表示1个手势的隐马尔科夫模型参数训练完成, 完成之后需要进行拟定义的第2个手势训练, 重复步骤(1) ~ (8), 直到定义的手势完成(定义几个手势就会有几个模型)。

3.4 HMM 手势识别与改进

训练完成之后, 将实时采集的待辨识编码序列分别计算后与各个训练好的模型对比, 计算出概率最大值, 然后取最大概率所对应的手势作为识别结果。但有效手势中夹杂着无效手势, 会影响已定义手势的识别率, 产生这种现象的原因是无论做哪种手势, 都会判别为最大前向概率模型所对应的手势。为降低误检率以及排除无效的手势, 本文采用如下方法进行改进。

这里以向上和向左2个手势为例, 由图13中数据可以看出, 对 $P(O | \lambda_n)$ 取以10为底的对数后, 有且只有待识别手势的前向概率小于-10, 若所有手势的概率都大于-10, 则为未定义手势, 以此可以排除未定义手势的干扰, 改进的 HMM 算法流程图如图14所示。

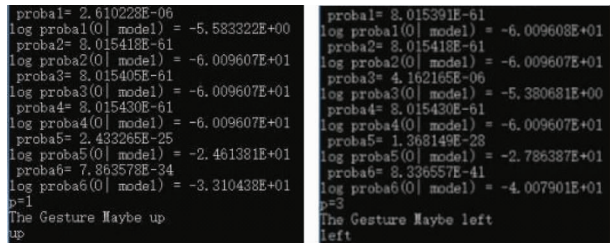


图 13 向上和向左手势时概率统计

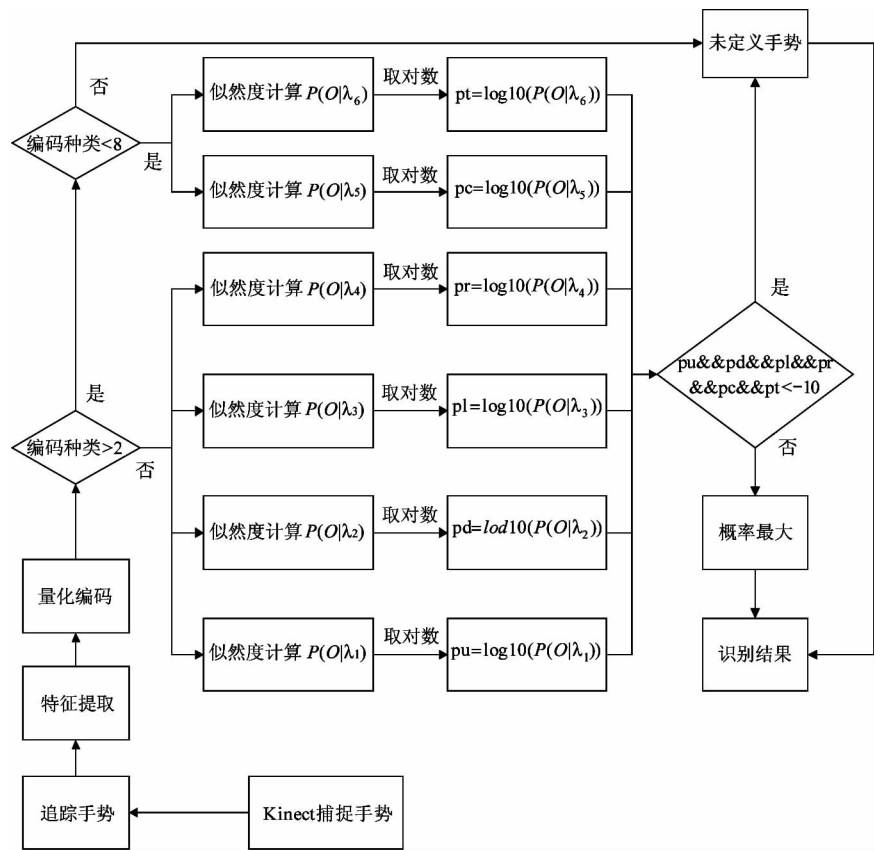


图 14 HMM 改进流程图

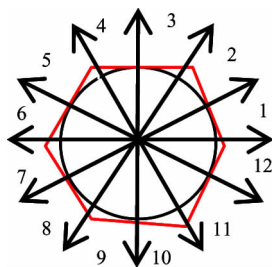


图 15 圆的编码示意图

4 实验

4.1 实验环境

本次实验中使用的处理器为 Intel (R) Core

由图 15 可知,圆形的手势运动轨迹编码种类大约在 6 个左右,由采集的圆形的编码特征序列可知其最大的编码种类数为 8 个,三角形的编码种类数为 3 个,所以改进的 HMM 算法流程图采用编码种类数是否小于 8 进行其他未定义手势的排除和定义手势的快速定位。

(TM) i5-4460 CPU @ 3.20 GHz, win10 64 位操作系统,基于 OpenCV 视觉库和 Kinect V2 开源数据库 Kinect SDK2.0,在 Visual Studio 2017 编程环境下使用 C++ 编程语言进行编程和算法改进。

4.2 实验结果

本文主要基于隐马尔科夫模型定义了上、下、左、右、圆圈和三角形 6 种手势,通过对隐马尔科夫模型识别方法的改进,对未定义手势进行了识别,提高了已定义手势的识别率,识别效果如图 16 和图 17 所示。

4.3 实验结果对比分析

4.3.1 改进后算法与改进前算法识别率对比

如图 17(a) 所示,改进前的 HMM 动态手势识别算法对未定义手势的识别率为 0%,也就是说,在识别过程中会将未定义手势识别成定义手势,从而影响已定义手势的识别率。产生这种现象的原因是无论做哪种手势,都会判别为概率模型所对应的概率最大的手势。本文通过对 HMM 算法的改进,有效地排除了未定义手势的干扰,降低了误检率,如图 17(b) 所示。表 1 为 HMM 手势识别方法改进前后的手势识别率对比,其中,HMM 代表改进前的 HMM 方法,IHMM 代表改进 HMM 方法,UD 代表未定义的无效手势。

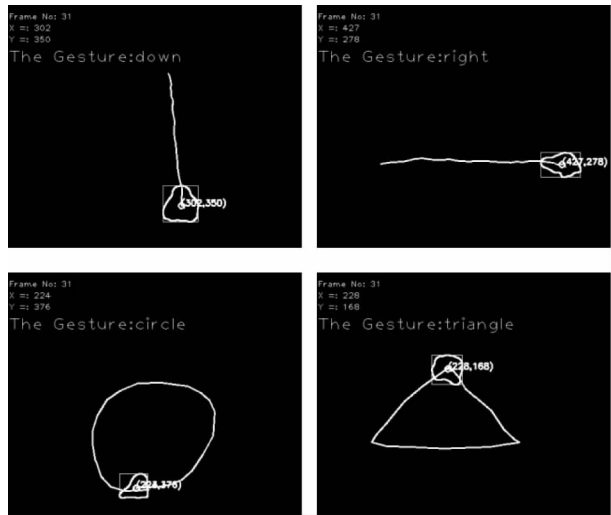


图 16 对已定义手势的识别

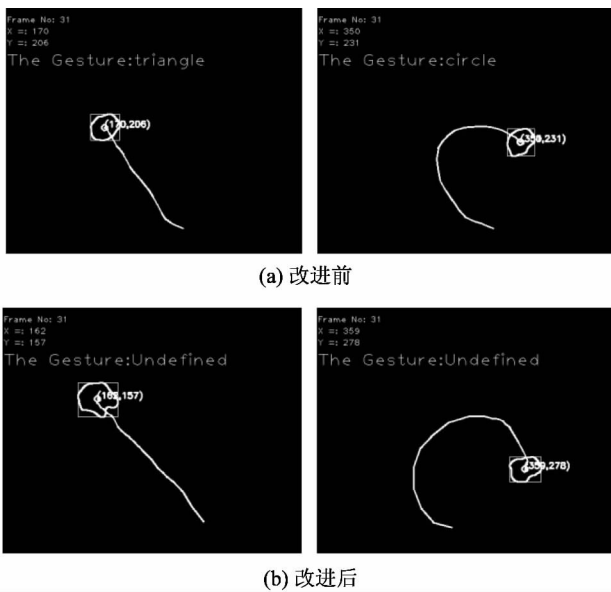


图 17 对未定义手势的识别

表 1 与改进前 HMM 算法对比实验

手势	实验次数	HMM 误检	IHMM 误检	HMM 识别率	IHMM 识别率
↑	50	2	0	96%	100%
↓	50	4	0	92%	100%
→	50	5	1	90%	98%
←	50	3	0	94%	100%
○	50	9	2	82%	96%
△	50	7	2	86%	96%
UD	50	50	1	0%	98%

从表 1 中可以看出,改进前的 HMM 方法对定义手势的平均识别率为 90.0%,而改进后的 HMM 方法对定义手势平均识别率已经达到 98.3%,提高了 8 个百分点。另外,改进前的 HMM 方法对未定义手势的识别率为 0.0%,而改进后的 HMM 方法对未定义手势识别率已经高达 98.0%。实验结果表明,改进的 HMM 方法有效地排除了未定义手势,提高了对训练手势的识别率,实验方法有效可行。

4.3.2 IHMM 在不同环境影响下的识别率对比

由表 2 可知,在黑暗条件和复杂环境的影响下,改进的算法对定义的 6 种手势和对未定义手势的识别率没有明显的浮动。实验结果表明,本文的手势识别算法能够很好地适应复杂背景和黑暗的环境。

5 结 论

本文利用 Kienct V2 的彩色数据和深度数据,通过获取手的质心位置,提取手势轨迹曲线上的切线角正切值特征,通过对 HMM 算法的改进,完成了对动态手势的识别和对未定义手势的排除,有效地降低了无效手势的误判率,使改进的隐马尔科夫识别方法的识别率达到 98% 以上,而且能够避免光照和复杂背景的影响。本文的数据是实际采集的,并不是针对特定的手势,因此在应用广泛性上有一定的优势,可以根据不同的需求来定义和训练手势,更好地满足用户人机交互的需求。由此可以得出,本文中改进的隐马尔科夫动态手势识别方法有效地减少了未定义手势的干扰,对动态手势识别有良好的实验效果且有一定的泛化能力。随着计算机技术的不断

表 2 环境影响对比实验

手势	光照条件		黑暗条件		复杂背景	
	误检次数	识别率	误检次数	识别率	误检次数	识别率
↑	0/50	100%	1/50	98%	0/50	100%
↓	0/50	100%	0/50	100%	0/50	100%
←	1/50	98%	1/50	98%	0/50	100%
→	0/50	100%	0/50	100%	0/50	100%
○	2/50	96%	3/50	94%	1/50	98%
△	2/50	96%	2/50	96%	2/50	96%
UD	1/50	98%	1/50	98%	0/50	100%

断发展,本文中的手势识别方法一定会在机器视觉和人机交互领域得到更好的应用。

参考文献

[1] 林伟强, 庄宝山. 动态手势轨迹跟踪与 HMM 模式识别[J]. 电脑知识与技术, 2014,10(32):7714-7718

[2] 吴晓雨, 杨成, 冯琦. 基于 Kinect 的手势识别算法研究及应用[J]. 计算机应用与软件, 2015, 32(7):173-176

[3] Guo X L, Yang T T. Gesture recognition based on HMM-FNN model using a Kinect[J]. *Journal on Multimodal User Interfaces*, 2016, 11(1):1-7

[4] Kumawat P, Khatri A, Nagaria B. Comparative analysis of offline handwriting recognition using invariant moments with HMM and combined SVM-HMM classifier[C]. In: 2013 International Conference on Communication Systems and Network Technologies (CSNT), Gwalior, India, 2013. 140-143

[5] 蒋海军, 谢钧, 段国仑, 等. 基于奇异值分解和隐 Markov 模型的动态手势识别[J]. 信息技术与网络安全, 2018, 37(1): 106-110

[6] Liu K, Chen C, Jafari R, et al. Multi-HMM classification for hand gesture recognition using two differing modality sensors[C]. In: 2014 IEEE Dallas Circuits and Systems Conference(DCAS), Richardson, USA, 2014. 1-4

[7] 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

[8] Ruan X G, Tian C Y. Dynamic gesture recognition based on improved DTW algorithm[C]. In: 2015 IEEE Inter-

national Conference on Mechatronics and Automation (ICMA), Beijing, China, 2015. 2134-2138

[9] Nguyen-Dinh L V, Calatroni A, Tröster G. Robust online gesture recognition with crowdsourced annotations [J]. *Journal of Machine Learning Research*, 2014, 15(1): 3187-3220

[10] 黄文静, 马力. 基于 Kinect 手势识别的研究与应用[J]. 电子设计工程, 2017(24):166-169

[11] 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

[12] Saha S, Lahiri R, Konar A, et al. HMM-based gesture recognition system using Kinect sensor for improvised human-computer interaction [C]. In: 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, USA, 2017. 2776-2783

[13] Ding I J, Chang Y J. HMM with improved feature extraction-based feature parameters for identity recognition of gesture command operators by using a sensed Kinect-data stream[J]. *Neurocomputing*, 2017, 262(1): 108-119

[14] 于胜举, 陈亚雄, 房森, 等. Kinect 与轨迹识别算法相结合在语音和手势识别方面的应用[J]. 中国科技博览, 2015(40):226-226

[15] Song Y, Gu Y, Wang P, et al. A Kinect based gesture recognition algorithm using GMM and HMM [C]. In: 2013 6th International Conference on Biomedical Engineering and Informatics, Hangzhou, China, 2013. 750-754

[16] 张乐, 戴广军, 朱凯, 等. 基于 Kinect 的动态手势识别交互算法[J]. 自动化应用, 2017(1):23-25

[17] Calin A D. Gesture recognition on Kinect time series data using dynamic time warping and hidden Markov models

- [C]. In: 2016 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Timisoara, Romania, 2016. 264-271
- [18] Chen S H, Hernawan A, Lee Y S, et al. Hand gesture recognition based on Bayesian sensing hidden Markov models and Bhattacharyya divergence [C]. In: 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 2017. 3535-3539
- [19] Kim Y J. 3D wrist gesture recognition using Three-Dimensional directional codes and HMM[J]. *International Journal of Multimedia and Ubiquitous Engineering*, 2015, 10(7):229-236
- [20] 杨萍萍. 基于 Kinect 的手势动作识别研究及其在虚拟仿真系统中的应用[D]. 南京:南京大学工程管理学院, 2018
- [21] Truong A, Zaharia T. Dynamic gesture recognition with Laban movement analysis and hidden Markov models [C]. In: Proceedings of the 33rd Computer Graphics International, Heraklion, Greece, 2017. 21-24
- [22] Yang W, Tao J, Xi C, et al. Sign language recognition system based on weighted hidden Markov model[C]. In: 2015 8th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 2016. 449-452
- [23] Kumar P, Gauba H, Roy P P, et al. Coupled HMM-based multi-sensor data fusion for sign language recognition[J]. *Pattern Recognition Letters*, 2017, 86(15):1-8
- [24] Zhou H, You M, Liu L, et al. Sequential data feature selection for human motion recognition via Markov blanket [J]. *Pattern Recognition Letters*, 2017, 86(15):18-25
- [25] Dinh D L, Lee S, Kim T S. Hand number gesture recognition using recognized hand parts in depth images[J]. *Multimedia Tools and Applications*, 2016, 75(2):1333-1348

Improvement and implementation of dynamic gesture recognition algorithm based on Kinect

Li Guoyou, Meng Yan, Yan Chunwei, Song Chengquan

(School of Electrical Engineering, Yanshan University, Qinhuangdao 066004)

Abstract

In order to solve the problem that the hidden Markov model (HMM) algorithm has low accuracy in dynamic gesture recognition, is easily interfered by undefined gestures, and is difficult to adapt to complex backgrounds, this paper proposes an improved HMM dynamic gesture recognition method based on Kinect V2 sensor. After the gesture segmentation, the method performs uniform quantization and coding with the value of the tangent angle of the centroid motion trajectory, sets the probability threshold model and the type of coding to exclude undefined gestures, performs dynamic gesture recognition, and compares the recognition effect under different experimental environments. The experimental results show that the improved HMM algorithm effectively eliminates many undefined gestures, can adapt to complex background and dark conditions, and can improve the recognition rate of already defined gestures.

Key words: Kinect V2 sensor, dynamic gesture recognition, improved hidden Markov model (HMM), undefined gesture, recognition rate