



(12)发明专利申请

(10)申请公布号 CN 106295463 A

(43)申请公布日 2017.01.04

(21)申请号 201510249327.7

(22)申请日 2015.05.15

(71)申请人 济南大学

地址 250022 山东省济南市市中区南辛庄
西路336号

(72)发明人 冯志全

(74)专利代理机构 济南泉城专利商标事务所
37218

代理人 肖健

(51)Int.Cl.

G06K 9/00(2006.01)

权利要求书1页 说明书5页

(54)发明名称

一种基于特征值的手势识别方法

(57)摘要

本发明的一种基于特征值的手势识别方法，首先构建手势数据库，选取 m 种手势，每种手势选取 n 幅手势图像，找出每幅手势图像中的四个特征值，存储在手势数据库中；然后在视频流中连续选取 K 帧待识别手势图像，采用相同的方法实时计算出每帧待识别图像的四个特征值进行比对。本发明的有益效果是：改善手势识别的识别率，并且对于手势的旋转、缩放和平移可以进行有效识别。

1. 一种基于特征值的手势识别方法,其特征在于,包括以下步骤:

步骤 1, 构建手势对比库, 选取 m 种手势, 每种手势选取 n 幅手势图像, 每幅手势图像对应关联一组模型数据, 每组模型数据包括四个特征值 $F1$ 、 $F2$ 、 $F3$ 、 $F4$;

其中 $F1$ 表示手势轮廓上各点到手势重心的平均距离除以手势外接圆半径的值, 其计算公式为: $F1 = [(\sum_{i=1}^S \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}) / S] / R$;

$F2$ 表示手势轮廓上距离手势重心的 M 个最大距离平均值除以外接圆半径的值, 其计算公式为: $F2 = [(\sum_{i=1}^M D_i) / M] / R$;

$F3$ 表示手势轮廓上距离手势重心的 N 个最小距离平均值除以外接圆半径的值, 其计算公式为: $F3 = [(\sum_{i=1}^N d_i) / N] / R$;

$F4$ 表示手势轮廓上距离手势重心的 M 个最大距离之和与手势轮廓上距离手势重心的 N 个最小距离之和的差值除以外接圆半径的值, 其计算公式为: $F4 = (\sum_{i=1}^M D_i - \sum_{i=1}^N d_i) / R$;

其中, (x_0, y_0) 是手势重心位置坐标, D_i 、 d_i 表示手势轮廓上每个点到重心 (x_0, y_0) 的欧式距离, 把 S 个欧式距离值按照由大到小排序, 得到序列 $(D_1, D_2, \dots, D_M, \dots, D_S)$; R 是手势外接圆半径;

步骤 2, 从视频流中连续选取 K 帧待识别手势图像, 并采用与步骤 1 相同的方法实时计算出每帧被识别手势图像对应关联的四个特征值 $F1$ 、 $F2$ 、 $F3$ 、 $F4$, 并将每帧被识别手势图像对应关联的四个特征值 $F1$ 、 $F2$ 、 $F3$ 、 $F4$ 保存在一个对比数组中;

步骤 3, 将步骤 2 中每帧待识别手势图像对应的对比数组分别与手势对比库内 $m \times n$ 幅手势图像对应的模型数据一一计算欧氏距离, 每帧待识别手势图像对应得到 $m \times n$ 个欧氏距离, 从中选出与该对比数组欧氏距离最小的一组模型数据, 并记录该组模型数据对应存储在手势对比库的那幅手势图像以及该最小的欧氏距离;

步骤 4, 采用步骤 3 的方法, 分别找出与 K 帧待识别手势图像对应的 K 个最小的欧氏距离, 然后在这 K 个最小的欧氏距离中再选取一个最小的欧氏距离 F , 找出与该欧氏距离值 F 对应的模型数据, 并最终将该模型数据对应存储在手势对比库的那张手势图像作为最终识别结果输出。

2. 根据权利要求 1 所述基于特征值的手势识别方法, 其特征在于: 在步骤 1 中, $m = 5$ 、 $n = 10$; 即选取 5 种手势, 每种手势选取 10 幅手势图像。

3. 根据权利要求 1 所述基于特征值的手势识别方法, 其特征在于: 在步骤 2 中, 从视频流的中第 10 帧开始直到第 30 帧结束, 连续选取 20 帧待识别手势图像。

一种基于特征值的手势识别方法

技术领域

[0001] 本发明涉及一种基于特征值的手势识别方法。

背景技术

[0002] 随着计算机的发展与应用,人与计算机的交互越来越密切。手势交互在人机交互领域内的重要性更是越发突出,其中基于视觉的手势识别技术已成为手势交互领域内的一个研究热点。从手势的运动特点出发,基于视觉的手势识别可分为动态手势识别和静态手势识别两大类。动态手势可以定义为手运动的轨迹,是指随时间的变化手势的形状和位置也发生着相应的改变。因此,识别效果容易受到手势轮廓、时空位置、移动速率等因素的影响。静态手势可以认为是动态手势中某一个时间点下的特例,是指不随时间变化而变化的手势,其识别效果与手势的轮廓、形状、纹理等相关。由于静态手势是动态手势变化过程中的某一状态,是手势识别的核心工作。

[0003] Haitham H 等提出一种利用神经网络进行手势识别的方法,该方法通过边缘检测提取样本几何矩特征和轮廓特征,然后利用神经网络对手势进行识别,但是该方法对静态手势的识别较低。姚远等利用深度像素信息设计了一个基于视觉的手势交互桌面应用开发框架,采用 RGB-D 信息作为数据输入,同时利用 3D 手形轮廓来进行手势匹配,该方法能够实现手势的快速匹配,但是识别时容易受到手臂的影响。Li 等提出了一种利用分层弹性图匹配 (HEGM) 的手势识别方法,用 Boosting 算法来确定一个给定的图形的层次结构,利用梯度方向直方图 (HOG) 提取视觉特征。该方法的手势识别率高达 99.85%。Padam 等提出一种基于几何标准化和 Krawtchouk 矩特征的静态手势识别算法,根据手势测量方法把手从手和前臂区域中提取出来,再用 Krawtchouk 矩特征和最小距离分类器把手势轮廓进行归一化。该方法对小的训练样本集能够进行很好的识别。

发明内容

[0004] 为解决以上技术上的不足,本发明提供了一种识别率高的基于特征值的手势识别方法。

[0005] 本发明是通过以下措施实现的:

[0006] 本发明的一种基于特征值的手势识别方法,包括以下步骤:

[0007] 步骤 1,构建手势对比库,选取 m 种手势,每种手势选取 n 幅手势图像,每幅手势图像对应关联一组模型数据,每组模型数据包括四个特征值 F1、F2、F3、F4;

[0008] 其中 F1 表示手势轮廓上各点到手势重心的平均距离除以手势外接圆半径的值,

其计算公式为:
$$F1 = [(\sum_{i=1}^S \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}) / S] / R;$$

[0009] F2 表示手势轮廓上距离手势重心的 M 个最大距离平均值除以外接圆半径的值,其

计算公式为:
$$F2 = [(\sum_{i=1}^M D_i) / M] / R;$$

[0010] F3 表示手势轮廓上距离手势重心的 N 个最小距离平均值除以外接圆半径的值, 其计算公式为:

$$F3 = [(\sum_{i=1}^N d_i) / N] / R;$$

[0011] F4 表示手势轮廓上距离手势重心的 M 个最大距离之和与手势轮廓上距离手势重心的 N 个最小距离之和的差值除以外接圆半径的值, 其计算公式为:

$$F4 = (\sum_{i=1}^M D_i - \sum_{i=1}^N d_i) / R;$$

[0012] 其中, (x_0, y_0) 是手势重心位置坐标, D_i 、 d_i 表示手势轮廓上每个点到重心 (x_0, y_0) 的欧式距离, 把 S 个欧式距离值按照由大到小排序, 得到序列 $(D_1, D_2, \dots, D_M, \dots, D_S)$; R 是手势外接圆半径;

[0013] 步骤 2, 从视频流中连续选取 K 帧待识别手势图像, 并采用与步骤 1 相同的方法实时计算出每帧被识别手势图像对应关联的四个特征值 F1、F2、F3、F4, 并将每帧被识别手势图像对应关联的四个特征值 F1、F2、F3、F4 保存在一个对比数组中;

[0014] 步骤 3, 将步骤 2 中每帧待识别手势图像对应的对比数组分别与手势对比库内 $m \times n$ 幅手势图像对应的模型数据一一计算欧氏距离, 每帧待识别手势图像对应得到 $m \times n$ 个欧氏距离, 从中选出与该对比数组欧氏距离最小的一组模型数据, 并记录该组模型数据对应存储在手势对比库的那幅手势图像以及该最小的欧氏距离;

[0015] 步骤 4, 采用步骤 3 的方法, 分别找出与 K 帧待识别手势图像对应的 K 个最小的欧氏距离, 然后在这 K 个最小的欧氏距离中再选取一个最小的欧氏距离 F, 找出与该欧氏距离值 F 对应的模型数据, 并最后将该模型数据对应存储在手势对比库的那张手势图像作为最终识别结果输出。

[0016] 在步骤 1 中, $m = 5$ 、 $n = 10$; 即选取 5 种手势, 每种手势选取 10 幅手势图像。

[0017] 在步骤 2 中, 从视频流的中第 10 帧开始直到第 30 帧结束, 连续选取 20 帧待识别手势图像。

[0018] 本发明的有益效果是: 改善手势识别的识别率, 并且对于手势的旋转、缩放和平移可以进行有效识别。

具体实施方式

[0019] 本发明的一种基于特征值的手势识别方法, 包括以下步骤:

[0020] 首先构建手势对比库。

[0021] 选取 m 种手势, 每种手势选取 n 幅手势图像, $m = 5$ 、 $n = 10$; 即选取 5 种手势, 每种手势选取 10 幅手势图像。每幅手势图像对应关联一组模型数据, 每组模型数据包括四个特征值 F1、F2、F3、F4, 其中, 其中 F1 表示手势轮廓上各点到手势重心的平均距离除以外接圆半径的值, 其计算公式为:

$$F1 = [(\sum_{i=1}^S \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}) / S] / R;$$

[0022] F2 表示手势轮廓上距离手势重心的 M 个最大距离平均值除以外接圆半径的值, 其计算公式为:

$$F2 = [(\sum_{i=1}^M D_i) / M] / R;$$

[0023] F3 表示手势轮廓上距离手势重心的 N 个最小距离平均值除以外接圆半径的值,其

$$F3 = [(\sum_{i=1}^N d_i) / N] / R;$$

[0024] F4 表示手势轮廓上距离手势重心的 M 个最大距离之和与手势轮廓上距离手势重心的 N 个最小距离之和的差值除以外接圆半径的值,其计算公式为:

$$F4 = (\sum_{i=1}^M D_i - \sum_{i=1}^N d_i) / R;$$

[0025] 其中, (x_0, y_0) 是手势重心位置坐标, D_i 、 d_i 表示手势轮廓上每个点到重心 (x_0, y_0) 的欧式距离,把 S 个欧式距离值按照由大到小排序,得到序列 $(D_1, D_2, \dots, D_M, \dots, D_S)$; R 是手势外接圆半径。

[0026] 对选取的五种手势,每种手势取 10 张手势图片,对每张图片求以上四个特征值,则一种手势就是一个 10×4 的二维数组:每行代表一张图片求取的 4 个特征值,每列代表一种特征值的 10 张图片的不同值。由此形成 5 个不同的手势文件,这五个文件一起构成手势对比库。库里共有 50 个特征值序列,每 10 个是一种手势。

[0027] 从视频流中连续选取 K 帧待识别手势图像,从视频流的中第 10 帧开始直到第 30 帧结束,连续选取 20 帧待识别手势图像。并采用与上述相同的方法实时计算出每帧被识别手势图像对应关联的四个特征值 F1、F2、F3、F4,并将每帧被识别手势图像对应关联的四个特征值 F1、F2、F3、F4 保存在一个对比数组中;将每帧待识别手势图像对应的对比数组分别与手势对比库内 $m \times n$ 幅手势图像对应的模型数据一一计算欧氏距离,每帧待识别手势图像对应得到 $m \times n$ 个欧氏距离,从中选出与该对比数组欧氏距离最小的一组模型数据,并记录该组模型数据对应存储在手势对比库的那幅手势图像以及该最小的欧氏距离;

[0028] 采用上述的方法,分别找出与 K 帧待识别手势图像对应的 K 个最小的欧氏距离,然后在这 K 个最小的欧氏距离中再选取一个最小的欧氏距离 F,找出与该欧氏距离值 F 对应的模型数据,并最后将该模型数据对应存储在手势对比库的那张手势图像作为最终识别结果输出。

[0029] 其具体的流程为:

[0030] (1) 初始化数据,并读入 5 种手势模型的比较数据。数据放在程序文件夹的“PPT\模型库\实心\3”下,包括 6 个文件,5 个手势文件和 1 个说明文件。说明文件给出了 M 和 N 的值,5 个手势文件每一个都是 10×4 的二维数组,每一行代表一张此手势图片求取的 4 个特征值序列。

[0031] (2) 由手势轮廓算法提取出的第 K 帧手势轮廓图像,从第 10 帧开始进行手势识别。选取从 10 帧开始主要是为了等待手势轮廓提取的比较稳定。

[0032] (3) 依次求取手势的重心→手势的外接圆半径→手势的所有边缘点→手势轮廓上各点到重心的距离和手势轮廓上各点到重心的平均距离→求取特征值 A→对手势轮廓上各点到重心的距离进行排序→求取特征值。

[0033] (4) 把求得的 4 个特征值与库里的 50 组特征值序列分别进行欧式距离比较,求出最小的欧氏距离,保存第 10 帧的最小欧式距离以及对应的手势序号。

[0034] (5) 若当前为第 11 帧,按以上步骤的 (2)、(3)、(4) 进行处理,并把当前帧所获得的最小欧式距离与保存的最小欧式距离进行比较,保存两者中最小的欧式距离以及对应的

手势序号。11 帧的后续帧与第 11 帧的处理方法相同。

[0035] (6) 若当前帧小于 30 帧则按照步骤 (2)、(3)、(4) 进行。

[0036] (7) 若当前帧为第 30 帧,对最小欧式距离所对应的手势序号进行判断:为“0”,手势为“包袱”;为“1”,手势为“剪刀”;为“2”,手势为“1 指”(竖大拇指);为“3”,手势为“ok”;为“4”,手势为“拳头”。

[0037] (8) 识别结束,退出。

[0038] 函数介绍

[0039] 1. void IdentifyGesture(BYTE*CopyBuffer, int width, int height, int FraNO, double Model0[10][4], double Model1[10][4], double Model2[10][4], double Model3[10][4], double Model4[10][4], double saveLastMin[2], double weight[4]);

[0040] 函数功能:实现特征值欧式距离对比,获取最小特征值,输出识别结果。输入:手势图像的首地址 CopyBuffer,手势图像的宽 width、高 height,帧数 FraNO,手势包袱的特征值 Model0[10][4],手势剪刀的特征值 Model1[10][4],手势 1 指的特征值 Model2[10][4],手势 ok 的特征值 Model3[10][4],手势拳头的特征值 Model4[10][4],保存最小的欧式距离和对应手势序号 saveLastMin[2] (saveLastMin[0]:最小的欧式距离, saveLastMin[1]:对应手势序号),权值 weight[4]。

[0041] 输出:以前帧的最小欧式距离和对应的手势序号 saveLastMin[2],手势识别结果。

[0042] 具体步骤:

[0043] ①调用函数 GetGestureFeature,求取当前帧图像的特征值。

[0044] ②把求得的 4 个特征值与库里的 50 组特征值序列分别进行欧式距离比较,求出最小的欧氏距离,保存第 10 帧的最小欧式距离以及对应的手势序号。

[0045] ③若当前为第 11 帧,把当前帧图像所获得的最小欧式距离与保存的最小欧式距离进行比较,保存两者中最小的欧式距离以及对应的手势序号。11 帧的后续帧与第 11 帧的处理方法相同。

[0046] ④若当前帧为第 30 帧,对最小欧式距离所对应的手势序号进行判断:为“0”,手势为“包袱”;为“1”,手势为“剪刀”;为“2”,手势为“1 指”(竖大拇指);为“3”,手势为“ok”;为“4”,手势为“拳头”。

[0047] 2. void GetGestureFeature(BYTE*pDib, int Width, int Height, int sortM, int sortN, double feature[4]);

[0048] 函数功能:求取当前帧图像的四个特征值。

[0049] 输入:手势图像的首地址 pDib,手势图像的宽 Width、高 Height,M 的值 sortM,N 的值 sortN,当前帧图像的四个特征值 feature[4]。

[0050] 输出:当前帧图像的四个特征值。

[0051] 具体步骤:

[0052] ①求手势的重心。

[0053] ②求手势的外接圆半径。

[0054] ③求手势的所有边缘点。

[0055] ④求手势轮廓上各点到重心的距离和手势轮廓上各点到重心的平均距离。

[0056] ⑤求取特征值 A。

[0057] ⑥对手势轮廓上各点到重心的距离进行排序。

[0058] ⑦求取特征值 B、C、D。

[0059] 3.int EdgeDetection(D2POINT edgepoint[], unsigned char*pDib, int Width, int Height) ;

[0060] 函数功能 :求取边缘点的坐标和总个数。

[0061] 输入 :边缘点的结构体数组 edgepoint[], 手势图像的首地址 pDib, 手势图像的宽 Width、高 Height。

[0062] 输出 :边缘点的坐标和总个数。

[0063] 具体步骤 :

[0064] ①图像的一行字节数。

[0065] ②对图像的像素点从上到下, 从左至右进行扫描。判断每个扫描点的 B、G、R 三基色。

[0066] ③若当前点为黑色 (背景点, 手势上的点不为黑色), 扫描下一个点。

[0067] ④若当前点不为黑色 ($b \neq 0 \&\&g \neq 0 \&\&r \neq 0$) (背景点, 手势上的点不为黑色), 对该点周围的 8 个点分别进行判断, 若这 8 个点中有 2 到 7 个黑色点, 则该点为手势边缘点, 记录该点的坐标, 边缘点个数 +1。

[0068] ⑤返回边缘点总个数。

[0069] 以上所述仅是本专利的优选实施方式, 应当指出, 对于本技术领域的普通技术人员来说, 在不脱离本专利技术原理的前提下, 还可以做出若干改进和替换, 这些改进和替换也应视为本专利的保护范围。