

基于 Kinect 的指尖检测与手势识别方法

谈家谱, 徐文胜*

(北京交通大学 机械与电子控制工程学院, 北京 100044)

(*通信作者电子邮箱 wshxu@bjtu.edu.cn)

摘要:针对基于视频的弯曲指尖点识别难、识别率不高的问题,提出一种基于深度信息、骨骼信息和彩色信息的手势识别方法。该方法首先利用 Kinect 相机的深度信息和骨骼信息初步快速判定手势在彩色图像中所在的区域,在该区域运用 YCrCb 肤色模型分割出手势区域;然后计算手势轮廓点到掌心点的距离并生成距离曲线,设定曲线波峰与波谷的比值参数来判定指尖点;最后结合弯曲指尖点特征和最大内轮廓面积特征识别出常用的 12 个手势。实验结果验证阶段邀请了 6 位实验者在相对稳定的光照环境条件下来验证提出的方法,每个手势被实验 120 次,12 种手势的平均识别率达到了 97.92%。实验结果表明,该方法能快速定位手势并准确地识别出常用的 12 种手势,且识别率较高。

关键词:Kinect; 肤色模型; 开源计算机视觉库; 指尖检测; 手势识别

中图分类号: TP391.41 **文献标志码:** A

Fingertip detection and gesture recognition method based on Kinect

TAN Jiapu, XU Wensheng*

(College of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China)

Abstract: The recognition of bending finger points based on video is difficult and recognition rate is not high. Focusing on the issue, a hand gesture recognition method based on depth image, skeleton image and color image information was proposed. Firstly, the gesture in color image area was initially determined rapidly by using depth information and skeleton information of Kinect, and the hand posture region was extracted from the gesture area by using the YCrCb color model. Then the distances between the gesture contour points and the palm point were calculated to generate the distance curve, and the ratio of curve peaks to troughs was set up to obtain finger point. Finally, commonly used 12 gestures were identified by combining bending finger point features and the maximum amount of contour area. Six experimenters were invited to validate the proposed method in experimental results verification phase. Every gesture was experimented 120 times under the condition of relatively stable light environment and the recognition rate of 12 gestures was 97.92% on average. The experimental results show that the proposed method can quickly location gestures and accurately recognize the commonly used 12 kinds of hand gestures, and the recognition rate is high.

Key words: Kinect; skin color model; Open Source Computer Vision Library (OPENCV); finger detection; gesture recognition

0 引言

随着计算机视觉的发展,基于视觉的手势识别方法已成为人机交互领域的研究热点^[1]。但由于手势应用的场景变化较大,正确高效地从计算机视觉捕获的图像中提取出手势有一定的难度,而提取出的手势效果好坏将直接影响手势识别的结果。近年来,学者们对于基于视觉的手势识别进行了研究。如基于颜色手套的手势识别方法^[2],该方法虽然降低了手势提取的难度,但同时也使得识别变得不够自然。为了使得手势识别变得更加自然、方便直接,一些研究者开始采用基于肤色模型^[3-4]的方法来对手势进行识别。但由于肤色模型受到光照以及类肤色背景的影响,手势提取的效果得不到保证。部分研究人员采用神经网络法^[5]、迭代算法(AdaBoost)^[6]、隐马尔可夫模型(Hidden Markov Model, HMM)^[7]和支持向量机(Support Vector Machine, SVM)算法^[8]对静态手势进行识别,识别率普遍在 90% 以上,但这些方法

前期都需要采集正负样本,采集量上百或上千张,同时需要对正负样本进行训练,要求训练用的计算机性能高,且训练周期长,给研究增加了一定的难度。随着 2010 年微软推出的 Kinect 以及其他种类深度相机的普及,越来越多的研究者通过利用带有深度信息的新相机来消除肤色模型分割的缺点^[9]。文献[10]利用 Kinect 的深度信息以及基于像素分类的方法进行指尖点检测并对手势抓取和释放进行识别,平均识别率达到 96.96%;但缺乏对具有像手势 9 这样弯曲指尖点检测的研究。为了规避人们日常生活中常用的手势 9 的弯曲特征,一些研究者定义指尖点特征明显的手势作为常用手势 9 的替代^[11],这样的确能减少手势 9 的识别难度;但是降低了人们的手势体验,变得不自然。文献[12]通过基于 Kinect 深度信息和骨骼信息相结合利用 K 曲率算法检测手指指尖点,平均识别率为 93%;但只考虑了指尖特征明显的简单手势 1~5 和手势拳头的识别,对于像手势 0 指尖特征与手势拳头指尖特征相似的检测就显得不足。有些研究者为了降

收稿日期:2014-12-25;修回日期:2015-04-01。 基金项目:国家自然科学基金资助项目(51175033)。

作者简介:谈家谱(1992-),男,江西南昌人,硕士研究生,主要研究方向:模式识别、图像处理; 徐文胜(1970-),男,湖北鄂州人,副教授,博士,主要研究方向:云制造、人机交互。

低识别难度,将手势拳头作为手势0的识别^[13]。目前国外借助 Kinect 设备已经研制出老年护理机器人,但该护理机器人只是通过对简单的一些人体姿势和人体动作进行识别,没有对手势进行识别。2013 年 Leap 公司推出了一款 Leap Motion 体感器,该设备侧重于对动态手势的识别,官方宣称该设备的工作精度可以达到 0.01 mm,高精度使得一些研究者开始借助该设备研究手势识别。文献[14]利用 Leap Motion 基于 SVM 算法对数字 0~9 的手势轨迹进行识别,平均识别率为 91%。文献[15]利用 Leap Motion 基于 HMM 算法对数字 0~9 的手势轨迹的平均识别率为 98.25%。尽管利用 Leap Motion 对手势的识别率较高,但是该设备的检测范围大体在传感器上方 25~600 mm (Kinect 深度值最大为 4096 mm),使得该设备适用条件苛刻,人们做手势时必须在该传感器可测的小范围内。用 Kinect 时则只需站在该相机前 4096 mm 范围内(微软建议在利用深度数据开发中使用 1220~3810 mm 范围内的值),相比 Leap Motion 来说使用更方便,限制条件更少。文献[16]通过用 Leap Motion 和 Kinect 两种设备作手势识别,对比发现前者提供的深度数据有限但精度高,后者则有一个完整的深度数据。两种设备各有优缺点,但是对于手势识别来说,需要的是对手势动作的整体把握,深度数据精度方面要求不是很严格。

针对上述问题,本文提出了一种基于 Kinect 深度信息和彩色信息的手势识别方法,可以检测出常用手势 9 的弯曲特征。该方法首先利用 Kinect 的骨骼和深度信息快速分割出手势,确定手势在图中的区域,将深度图像中的手势坐标点——对应到彩色坐标空间中,对彩色坐标空间中的手势区域进行膨胀,膨胀后的区域即为彩色空间中手势的背景区域,这一步骤减少了包含手势的背景区域面积。在该背景区域内基于 YCrCb 肤色模型分割出有效的肤色区域。由于背景面积的减少,可以适当地扩大肤色区域的判定阈值区间,减少光照变化的影响。提取分割出的手势轮廓点并与掌心点做距离直方图,利用人手指尖和两相邻指尖点间的凹点与掌心点的距离比值这一特征检测出指尖点候选点,接着提取手势其他的表现特征(像内轮廓、指尖点之间的夹角、弯曲特征)并分类,从而实现手势识别。

1 手势提取

在手势识别过程中,准确快速地提取手势是实现对手势高效识别的前提。本文利用微软 2010 年发布的 Kinect 传感器,结合开源计算机视觉库 (Open Source Computer Vision Library, OPENCV) 库来采集和处理包含手部区域的彩色数据、深度数据以及骨骼数据^[17],利用深度数据和骨骼数据大致定位包含手部区域的彩色感兴趣区域 (Region of Interest, ROI),结合 YCrCb 颜色空间对 ROI 区域进行肤色检测并提取面积最大的肤色区域,即手的区域。

1.1 包含手部的 ROI 提取

通过 Kinect 传感器采集彩色图像、深度图像以及骨骼数据,将深度图像每个像素点的深度值存储到二维数组 $distance$ 中。Kinect 传感器的骨骼追踪引擎能够跟踪和获取每个用户的近 20 个点或者关节信息。

提取右手掌关节三维坐标 Z 方向的值 z 进行以下判定^[18-19]:

If $(z - 100 < distance[i][j] < z + 50 \ \&\& \ distance[i][j] < 1200)$
 $distance[i][j] = 1;$
 Else $distance[i][j] = 0;$

将 $distance$ 二维数组与深度图进行映射,如果 $distance[i][j] = 1$,则深度图中对应坐标点的像素值取为 0xff。如果 $distance[i][j] = 0$,则深度图中对应的像素值取为 0x00。将映射处理后的深度坐标空间中的手势坐标点——转化到彩色坐标空间中,将转化后的手势进行膨胀处理得到包含手势的区域 ROI,从彩色图片中相应的 ROI 区域进行显示,其他区域设为 0x00。整个过程如图 1 所示(图 1(a)为深度手部区域图,图 1(b)为将深度坐标空间中的图 1(a)转化过来的彩色坐标空间中的手部区域图,图 1(c)为图 1(b)膨胀之后的 ROI 区域,图 1(d)为在 ROI 区域显示的彩色图)。



图1 包含手部的 ROI 区域提取

1.2 基于 YCrCb 的肤色提取

肤色是人体表面最为显著的特征之一,是人脸及手部区别于其他部分的重要特征。肤色检测的效果将直接影响手势识别的结果。目前肤色检测的颜色空间主要有 HSV、YUV、YCrCb、HIS 以及归一化的 RGB 颜色空间。在这些颜色空间中,YCrCb 颜色空间具有与人的视觉感知一致性,能够有效地从图像中分割出肤色区域。彩色分量 Y、Cr、Cb 由 R、G、B 分量经过线性变换得到,转换公式为:

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.257 & 0.564 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

由于 ROI 区域中手势区域面积占比较大,因此可以适当扩大肤色的判定阈值区间。经过大量测试实验,发现 $136 \leq Cr \leq 170, 70 \leq Cb \leq 127$ 时,肤色分割出的效果较好。本文通过基于 YCrCb 对 ROI 区域的肤色进行提取,对提取出的肤色图进行腐蚀、平滑和膨胀处理。为了之后指尖检测的方便,利用 OpenCV 中的 cvFindContours 函数寻找出肤色分割后最大的肤色轮廓区域,排除一些肤色检测小区域的干扰,将分割出的手势轮廓区域像素值设为 0xff,其他区域设为 0x00。整个手势的提取过程如图 2 所示(图 2(a)为 ROI 区域基于 YCrCb 肤色分割的初步处理,图 2(b)为基于图 2(a)的腐蚀操作,图 2(c)为基于图 2(b)的平滑处理,图 2(d)为基于图 2(c)的膨胀处理,图 2(e)为利用 OpenCV 找出的最大轮廓区域即手势)。由图 2(e)可知轮廓相对于图 1(a)轮廓,边缘变得更平滑了,减少了锯齿状对后期指尖点检测的干扰。

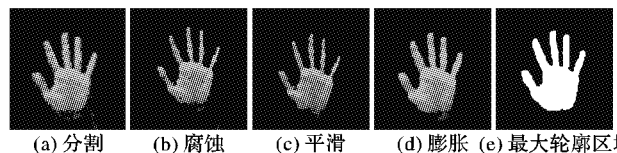


图2 手势的提取处理过程

2 指尖点检测

数字手势最明显的特征是手的指尖点,所以指尖点检测算法的好坏将直接影响后期手势识别的结果。有些研究者利用凸包算法来检测指尖点,该算法对于伸直的指尖点的检测没有问题,但是凸包算法不能准确地检测出手势 9 的弯曲指尖点。为解决手势 9 弯曲指尖点检测问题,本文利用分割出

的手势进行掌心点计算以及对手势轮廓点进行顺时针提取,通过计算手势轮廓点与掌心点的距离并生成距离曲线,找出距离曲线中的波峰波谷点。通过手掌骨骼可以发现人类伸直后的指尖点和两相邻指尖点间凹处与掌心点距离的比值变化不大。利用上述这一特性,设定比例阈值找出指尖点候选点,再结合不同手势的其他特征判定该候选点是否是指尖点。

2.1 掌心点计算

微软 Kinect 获取的骨骼信息提供了左右手的掌心点坐标,但该掌心点坐标并不是很准确,而且也不是很稳定,因此本文舍弃了骨骼信息提供的掌心点。掌心点的计算可以直接对图 2(e) 中手势区域利用求几何中心的计算公式来得到,但由于手指区域的影响,几何中心与掌心点的偏差较严重。因此,为减少手指区域对掌心点计算的影响,本文在计算掌心点时引入权重系数。由图 3 可以看出手势 5 在不同的水平线前后像素值变化的次数不同,手掌区域变化的次数为 2,手指区域变化次数则为 4、6、8、10 等。因此可以设定变化次数为 6、8 和 10 的区域像素值为零,利用 cvFindContours 函数重新获得 ROI 区域内轮廓最大的区域。在该区域内对变化次数为 4 的区域坐标在进行掌心点计算时引入权重系数 1/3,变化次数为 2 的区域坐标在进行掌心点计算时引入权重系数 2/3,减少手指区域对掌心点的影响。利用几何中心的计算公式来求掌心点,如果像素值为 0x00,则该像素点在掌心点计算中不起作用。如果像素值为 0xff,掌心点计算如式(1):

$$\begin{cases} Avr_x = w_1 * \sum_{i=1}^{n_1} x_{1i}/n_1 + w_2 * \sum_{i=1}^{n_2} x_{2i}/n_2 \\ Avr_y = w_1 * \sum_{i=1}^{n_1} y_{1i}/n_1 + w_2 * \sum_{i=1}^{n_2} y_{2i}/n_2 \end{cases} \quad (1)$$

其中: Avr_x 为求得的掌心点的横坐标; Avr_y 为求得的掌心点的纵坐标; w_1 的值为 1/3; x_{1i} 是行变化次数为 4 的横坐标; y_{1i} 是行变化次数为 4 的纵坐标; n_1 是行变化次数为 4、像素值为 0xff 的总数量; w_2 的值为 2/3; x_{2i} 是行变化次数为 2 的横坐标; y_{2i} 是行变化次数为 2 的纵坐标; n_2 是行变化次数为 2、像素值为 0xff 的总数量。(Avr_x, Avr_y) 为所求的掌心点坐标。

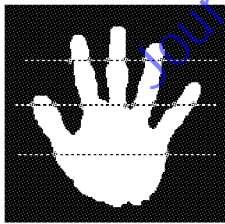


图3 手势区域像素值变化次数示意图

掌心点求取的图像处理过程如图 4 所示(图 4(a) 为手势图,图 4(b) 为设置变化次数大于 4 的行像素值为 0x00,图 4(c) 为利用 OpenCV 求轮廓最大区域并对该区域求掌心点)。

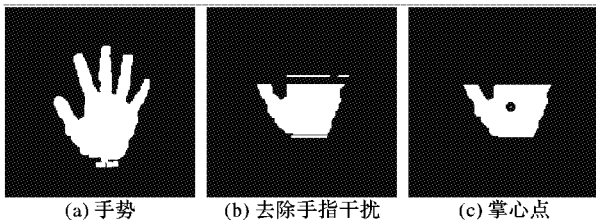


图4 手势掌心点求取过程

2.2 手势轮廓点提取

要提取手势轮廓点,首先需要从手势区域中分离出轮廓

边缘点以及内部点。利用十字型窗口遍历 ROI 区域的像素点,判断处于该十字型窗口的像素值是否都为 0xff。判断结果为真则为内部点并去除,假则为轮廓候选点。遍历循环结束后 ROI 区域只剩下轮廓候选点。但是为实现顺时针提取轮廓点,需要判定像素值为 0xff 的三个连续点是否构成了直角,是则消除该伪轮廓点,并用 3*3 的窗口去判定整个窗口内是否只有两个像素值为 0xff,是则去除该毛刺点,防止有毛刺点使得程序崩溃。使得最后得出的轮廓可以满足以下条件:遍历循环 ROI 区域像素值,当判定像素值为 0xff 时,以此像素为中心做 3*3 的窗口,可以发现窗口始终只有 3 个像素值为 0xff,这有利于后面的顺时针轮廓点提取。上述步骤的大致过程如图 5 所示。

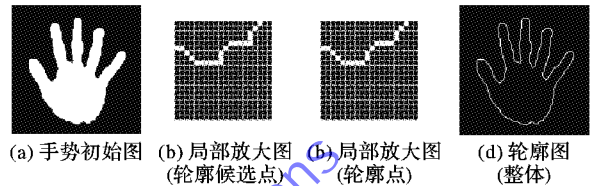


图5 获取轮廓图像的过程

通过掌心点坐标计算出图像中同一行从左到右第一个像素值为 0xff 的坐标作为起始坐标 $start1$ 存入轮廓点数组中,同一行从右到左第一个像素值为 0xff 的坐标作为终点坐标 end 。求取步骤如下所示:

- 1) 以起始坐标点 $start1$ 作为 3*3 窗口的中心,从 3*3 窗口左上角绕着 $start$ 中心点顺时针循环查看第一个像素值 0xff 的点作为坐标 $start2$ 存入轮廓点数组中;
- 2) 然后以 $start2$ 为中心坐标查看 3*3 窗口中像素值为 0xff 且点坐标不为 $start1$ 的坐标 $start3$ 存入轮廓点数组中;
- 3) 接着以上 $start3$ 为中心查看 3*3 窗口中像素值为 0xff 且坐标不为 $start2$ 的坐标 $start4$ 存入轮廓点数组中。

按如上步骤循环直到存入轮廓点数组的坐标点为 end 点,将轮廓点顺时针提取出来。接着轮廓点每隔 4 个点重新显示在图中,增加轮廓边缘的平滑度,减少后期处理的计算量。如图 6 所示。

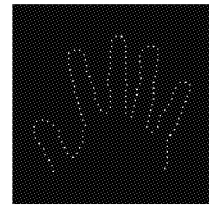


图6 手势 5 轮廓点图

2.3 候选指尖点检测

获得轮廓点后,按顺序计算每个轮廓点与掌心点的距离并生成距离直方图,利用波峰点上方的 3*5 窗口的像素值都为 0x00 以及波谷点下方 3*5 窗口的像素值为 0xff 的特点计算直方图的波峰点和波谷点(将 $start1$ 点和 end 点判定为波谷点)。此时的波峰点和波谷点数据还不适合直接用来做指尖点判断,需要将数据经过下面的处理:

- 1) 对求取出来的波峰点集进行局部波峰点抑制,当两个波峰点的横坐标相差在 15 个像素内,舍弃距离掌心点相对小的波峰点,将剩下的波峰点按横坐标从小到大排序。
- 2) 对求取出来的波谷点集按横坐标从小到大排序,检测每两个相邻的波谷点之间是否有波峰点,如果没有波峰点,则舍弃距离掌心点相对大的波谷点。将检测完的波谷点集按横

坐标从小到大排序。

3) 检测每两个相邻的波峰点之间是否有波谷点, 如果没有波谷点, 则舍弃距离掌心点相对小的波峰点, 将剩下的波峰点按横坐标从小到大排序。

经过以上三步的处理之后, 每相邻两个波谷点之间都有一个波峰点, 每相邻两个波峰点之间都有一个波谷点。设置判定条件波峰点距掌心点的距离比上两相邻波谷点距掌心点的距离大于 $threshold1$ 时 (经实验测定, 阈值设为 1.5 时判定效果较好), 则为指尖候选点。对于手势 5, 指尖候选点即为指尖点。将手势 5 检测出的距离曲线数据、波峰点数据和波谷点数据输出用 Matlab 显示, 如图 7 所示。

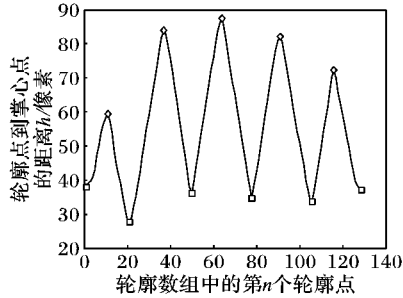


图7 手势5的距离曲线以及波峰波谷点检测

3 手势识别

不同的手势所具有的特征是不一样的, 通过检测这些特征可以分辨出不同的手势。本文所要识别的手势如图 8 所示。将要识别的手势的距离曲线数据输出用 Matlab 显示如图 9 所示。

由图 8 可以发现手势 0 和手势 OK 都具有内轮廓特征, 手势 2、6 和 8 都具有两个指尖点特征, 手势 1、7、9 都具有一个指尖点特征, 手势 3、4、5 分别具有比较明显的指尖点特征 (分别为 3 个, 4 个, 5 个), 手势拳头没有指尖点。上述特征结合图 9 距离曲线的指尖特征可以将这 12 种手势区分开来。12 种手势所具有的不同特征如表 1 所示。

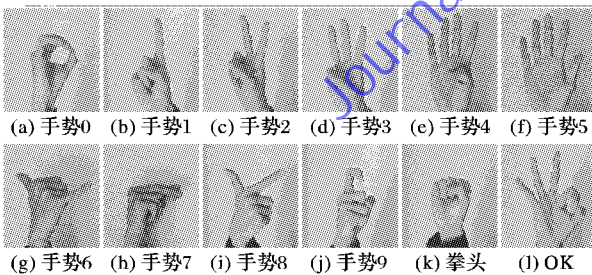
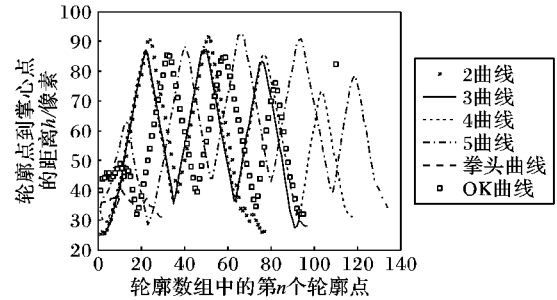


图8 所要识别的12种手势

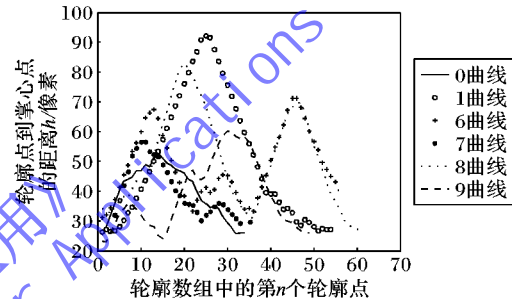
手势识别程序中的手势判定步骤如下:

- 1) 对 Kinect 提取出的三种数据进行处理, 获得手势最大内轮廓面积特征、指尖点特征。
- 2) 判定最大内轮廓面积是否大于设定的阈值 150 个像素点, 如果大于设定的阈值, 则转入步骤 3); 否则转入步骤 4)。
- 3) 判定指尖点特征, 如果指尖点数为 3, 则为手势 OK, 当前手势识别完成, 转入步骤 1) 循环进行手势识别; 如果指尖点数为 0, 则为手势 0, 当前手势识别完成, 转入步骤 1) 循环进行手势识别; 如果指尖点数不为 3 和 0, 则该手势不在识别范围内, 转入步骤 1) 循环进行手势识别。
- 4) 判定指尖点特征, 如果指尖点数为 0, 则为手势拳头, 当前手势识别完成, 转入步骤 1) 循环进行手势识别; 如果指

尖点数为 1, 则转入步骤 5); 如果指尖点数为 2, 则转入步骤 6); 如果指尖点数为 3, 则为手势 3, 当前手势识别完成, 转入步骤 1) 循环进行手势识别; 如果指尖点数为 4, 则为手势 4, 当前手势识别完成, 转入步骤 1) 循环进行手势识别; 如果指尖点数为 5, 则为手势 5, 当前手势识别完成, 转入步骤 1) 循环进行手势识别。



(a) 手势2、3、4、5、拳头、OK手势轮廓点到掌心点的距离曲线



(b) 手势0、1、6、7、8、9手势轮廓点到掌心点的距离曲线

图9 不同手势轮廓点到掌心点的距离曲线

表1 不同手势所具有的不同特征

手势	特征
0	内轮廓面积大于 150 个像素点, 通过候选指尖点与掌心点连线过内轮廓则舍弃处理后无指尖点
1	候选指尖点即为指尖点, 只有 1 个指尖点, 指尖点与掌心点的像素点距离大于 2.4 倍的相邻波谷点与掌心点的像素点距离 (阈值参数为多次实验得来), 无弯曲特征
2	候选指尖点即为指尖点, 有 2 个指尖点, 两指尖与掌心点之间的夹角小于 60°
3	候选指尖点即为指尖点, 有 3 个指尖点
4	候选指尖点即为指尖点, 有 4 个指尖点
5	候选指尖点即为指尖点, 有 5 个指尖点
6	候选指尖点即为指尖点, 有 2 个指尖点, 两指尖与掌心点之间的夹角大于 60° , 距离曲线中两指尖点之间有波峰
7	候选指尖点即为指尖点, 只有 1 个指尖点, 指尖点与掌心点的像素点距离小于 2.4 倍的相邻波谷点与掌心点的像素点距离 (阈值参数为多次实验得来), 无弯曲特征
8	候选指尖点即为指尖点, 有 2 个指尖点, 两指尖与掌心点之间的夹角大于 60° , 距离曲线中两指尖点之间没有波峰
9	有弯曲特征, 候选指尖点只有 1 个, 候选指尖点不为指尖点
拳头	候选指尖点即为指尖点, 指尖点数为 0
OK	内轮廓面积大于 150 个像素点, 通过候选指尖点与掌心点连线过内轮廓则舍弃处理后有三个指尖点

5) 检测手势的弯曲特征, 如果手势有弯曲特征, 则为手势 9, 当前手势识别完成, 转入步骤 1) 循环进行手势识别。如果手势无弯曲特征且指尖点与掌心点的像素点距离大于 2.4

倍的相邻波谷点与掌心点的像素点距离,则为手势 1,当前手势识别完成,转入步骤 1)循环进行手势识别;如果手势无弯曲特征且指尖点与掌心点的像素点距离小于 2.4 倍的相邻波谷点与掌心点的像素点距离,则为手势 7,当前手势识别完成,转入步骤 1)循环进行手势识别。

6)借助掌心点,判定 2 个指尖点的夹角以及轮廓曲线中两指尖点指尖是否有凸起。如果夹角小于 60° 时,则为手势 2,当前手势识别完成,转入步骤 1)循环进行手势识别;如果夹角大于 60° 且两指尖点之间有凸起,则为手势 6,当前手势识别完成,转入步骤 1)循环进行手势识别;如果夹角大于 60° 且两指尖点之间无凸起,则为手势 8,当前手势识别完成,转入步骤 1)循环进行手势识别。

由表 1 可以发现手势 1、7 和手势 9 相差一个弯曲特征以及手势 9 的候选指尖点并不是手势 9 真正的指尖点。因此需要判定手势弯曲特征并求出手势 9 的指尖点,其判定以及求取步骤如下:

1)当候选指尖点为 1 时,将手势轮廓点用直线连接起来,直线像素为 0xff,其余像素为 0x00;

2)在 $finger.y$ 和 $(scenter.y + finger.y) * 1/2$ 行内从左到右计算相邻像素点数值的变化次数,变化次数达到 8 时则手势有弯曲特征($finger.y$ 为候选指尖点 y 坐标, $scenter.y$ 为掌心点 y 坐标);

3)在步骤 2)设定的行内检测相邻像素点数值变化次数为 6 的行,寻找该行从左到右像素点数值第一次变化的点坐标,计算轮廓点与该点的距离,距离最小的轮廓点即为所求的指尖点。

弯曲特征以及手势 9 指尖点的求取过程如图 10 所示,所有手势检测出来的指尖点如图 11 所示。利用 `CvPutText` 函数在图像中输出手势识别结果(`CvPutText` 函数不支持汉字,只能用英文显示),手势 5 识别结果如图 12 所示。

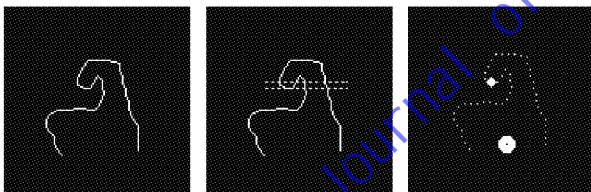


图 10 弯曲特征判断和手势 9 指尖提取

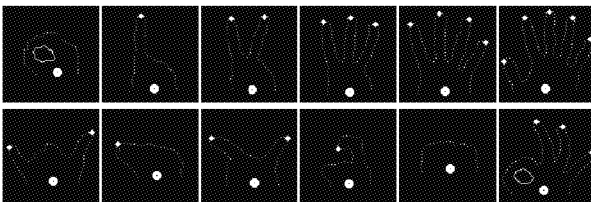


图 11 需要识别的全部手势指尖点(中间的圆点为掌心点)



图 12 手势 5 的识别结果

4 实验结果与分析

本实验使用的计算机处理器为 Intel Core i3 CPU M350 @ 2.27 GHz, 2.00 GB 内存,摄像头为第一代的 Kinect XBOX 摄像机。利用 Microsoft Visual Studio 2010 作为系统开发平台,并结合 OpenCV 和开放自然交互 (Open Natural Interaction, OpenNI) 完成了本文算法的实现。

为了检验本文算法的准确性和实时性,邀请了 6 位实验者在环境光照相对稳定的条件下对 12 个常见手势进行实验。每个手势总共被实验了 120 次,测试结果如表 2 所示。

表 2 手势识别结果

手势	实验次数	正确识别次数	识别率/%
0	120	112	93.3
1	120	119	99.2
2	120	120	100
3	120	120	100
4	120	120	100
5	120	120	100
6	120	120	100
7	120	110	91.7
8	120	120	100
9	120	114	95
拳头	120	120	100
OK	120	115	95.8

从表 2 可以看出,该算法对手势 2、3、4、5、6、8 和手势“拳头”的识别结果达到了 100%,手势 0、1、7、9、手势“OK”的识别结果达到了 90% 以上。12 种手势的平均识别率为 97.92%。文献[11]提出一种基于手指角度特征的手势识别算法,以指尖点到手掌中心的连线构成手势骨架,计算手指间角度,以角度的大小和指尖数量进行分类。文献[15]利用 Leap Motion 设备基于支持向量机 (SVM) 算法来识别数字手势轨迹 1~9。尽管文献[11]和本文的算法是识别静态手势,文献[15]是识别手势轨迹,但目的都是识别手势 1~9。因此本文将这三种算法对相同手势的识别结果进行对比,其结果如表 3 所示。

表 3 三种算法对比

手势 数字	识别率/%		
	文献[11]算法	文献[15]算法	本文算法
1	98	96.25	99.2
2	96	100	100
3	96	98.75	100
4	96	100	100
5	97	100	100
6	97	100	100
7	95	100	91.7
8	98	100	100
9	98	93.75	95

由表 3 可知本文算法对手势 1~9 的平均识别率 98.43% 高于文献[11]的平均识别率 96.78%,略低于借助专业设备 Leap Motion 文献[15]的平均识别率 98.75%。文献[11]中仅通过判断手指间的角度大小以及指尖点来识别手势,定义的手势 7 和手势 9 利用指尖点特征明显的手势来代替日常生活中使用的手势 7 和手势 9,减少了识别难度。本文算法则

通过增加对最大内轮廓面积特征以及弯曲特征的识别使得手势识别更准确,识别率更高。文献[15]使用隐马尔可夫(HMM)算法对手势进行识别,尽管识别率略高于本文算法,但该算法前期需要对大量的样本进行训练,对计算机性能要求高。而本文算法则不需要在前期对大量的样本进行训练,对于常用的手势1~9的识别率也能达到98%以上。

5 结语

本文通过 Kinect 的深度数据和骨骼数据快速定位并缩小手势所在的区域,减少后续图像处理的时间,在缩小后的手势区域中利用 YCrCb 模型进行肤色分割,得到手势图。再利用文中所述算法步骤获得掌心点以上的轮廓点图以及轮廓点与掌心点之间的距离曲线。最后利用距离曲线特征、内轮廓特征以及指尖夹角特征来识别出这 12 种手势。通过实验者所做的手势实验结果可以发现,在环境光照相对稳定的条件下,该算法能准确识别手势 2、3、4、5、6、8 和手势拳头。手势 7、手势 1、手势 OK 和手势 0 的平均识别率也达到了 95.66%。结果表明,该算法对手势的识别率较高,能够较好地应用于人机交互、面向机器人的手势控制等应用中。

由于本实验是在光照相对稳定以及手势离 Kinect 距离不变的条件下进行的,因此后期将在不同的光照和距离条件下进行实验,确定不同条件下的阈值,实现算法在不同条件下阈值的自适应。

参考文献:

- [1] ZHU Y, YANG Z, YUAN B. Vision based hand gesture recognition [C]// ICSS 2013: Proceedings of the 2013 International Conference on Service Sciences. Washington, DC: IEEE Computer Society, 2013: 260-265.
- [2] MAZUMDAR D, TALUKDAR A K, SARMA K K. Gloved and free hand tracking based hand gesture recognition [C]// ICETACS 2013: Proceedings of the 2013 1st International Conference on Emerging Trends and Applications in Computer Science. Piscataway: IEEE, 2013: 197-202.
- [3] YANG L, SHUI Y, YAO L. Static gesture detection based on combination of skin color segmentation and hand shape matching algorithm [J]. Information and Communications, 2013(4): 4-6. (杨磊, 隋云衡, 姚立虎. 结合肤色分割与手形匹配算法的静态手势检测[J]. 信息通信, 2013(4): 4-6.)
- [4] CHOUDHURY A, TALUKDAR A K, SARMA K K. A novel hand segmentation method for multiple-hand gesture recognition system under complex background [C]// SPIN 2014: Proceedings of the 2014 International Conference on Signal Processing and Integrated Networks. Piscataway: IEEE, 2014: 136-140.
- [5] HASAN H, ABDULKAREEM S. Static hand gesture recognition using neural networks [J]. Artificial Intelligence Review, 2014, 41(2): 147-181.
- [6] WANG K, YU H, ZHANG P. Real-time gesture recognition based on AdaBoost algorithm and optical flow matching [J]. Microelectronics and Computer, 2012, 29(4): 138-141. (王凯, 于鸿洋, 张萍. 基于 AdaBoost 算法和光流匹配的实时手势识别[J]. 微电子学与计算机, 2012, 29(4): 138-141.)
- [7] VIRRIU R, MIRONICAL L, GORAS B. Background invariant static hand gesture recognition based on hidden Markov models [C]// ISSCS 2013: Proceedings of the 2013 International Symposium on Signals, Circuits and Systems. Piscataway: IEEE, 2013: 1-4.
- [8] REN Y, GU C. Hand gesture recognition based on HOG characters and SVM [J]. Bulletin of Science and Technology, 2011, 27(2): 211-214. (任彧, 顾成成. 基于 HOG 特征和 SVM 的手势识别[J]. 科技通报, 2011, 27(2): 211-214.)
- [9] LI Y. Hand gesture recognition using Kinect [C]// ICSESS 2012: Proceedings of 2012 IEEE 3rd International Conference on Software Engineering and Service Science. Piscataway: IEEE, 2012: 196-199.
- [10] YAN H, ZHANG M, TONG J, et al. Real time robust multi-fingertips tracking in 3D space using Kinect [J]. Journal of Computer-Aided Design and Computer Graphics, 2013, 25(12): 1801-1809. (晏浩, 张明敏, 童晶, 等. 基于 Kinect 的实时稳定的三维多手指跟踪算法[J]. 计算机辅助设计与图形学学报, 2013, 25(12): 1801-1809.)
- [11] YU B, CHEN Y, WANG S, et al. Static gesture recognition algorithm based on characteristics of finger angle [J]. Journal of Xihua University: Natural Science Edition, 2014, 33(1): 69-72. (俞博, 陈永强, 王双一, 等. 基于手指角度特征的静态手势识别算法[J]. 西华大学学报: 自然科学版, 2014, 33(1): 69-72.)
- [12] WANG S, XU W. Fingertips recognition based on Kinect depth and skeleton data [EB/OL]. [2014-06-04]. <http://www.cnki.net/kcms/detail/10.3778/j.issn.1002-8331.1401-0135.html>. (王松林, 徐文胜. 基于深度信息与骨骼信息的手指指尖识别方法[EB/OL]. [2014-06-04]. <http://www.cnki.net/kcms/detail/10.3778/j.issn.1002-8331.1401-0135.html>.)
- [13] GANG Z. Static hand gesture recognition and its application based on Hu moments & support vector machine [D]. Qingdao: Qingdao University of Science and Technology, 2008. (甘志杰. 基于 Hu 矩和支持向量机的静态手势识别及应用[D]. 青岛: 青岛科技大学, 2008.)
- [14] YUAN X, WANG Q, BAI X, et al. A novel feature extracting method for dynamic gesture recognition based on support vector machine [C]// Proceedings of the 2014 IEEE International Conference on Information and Automation. Piscataway: IEEE, 2014: 437-441.
- [15] QIANG W, DAN X, YEN C, et al. Dynamic gesture recognition using 3D trajectory [C]// ICIST 2014: Proceedings of the 2014 4th IEEE International Conference on Information Science and Technology. Piscataway: IEEE, 2014: 598-601.
- [16] MARIN G, DOMINIO F, ZANUTTIGH P. Hand gesture recognition with leap motion and Kinect devices [C]// ICIP 2014: Proceedings of the 2014 IEEE International Conference on Image Processing. Piscataway: IEEE, 2014: 1565-1569.
- [17] ZHAO J, ZHANG D. Number gesture recognition method based on OpenCV [J]. Journal of Computer Applications, 2013, 33(S2): 193-196. (赵健, 张冬泉. 基于 OpenCV 的数字手势识别算法[J]. 计算机应用, 2013, 33(S2): 193-196.)
- [18] WANG Y, ZHANG Q. Gesture recognition based on Kinect depth information [J]. Journal of Beijing Information Science and Technology University: Natural Science Edition, 2013, 28(1): 22-26. (王艳, 张奇志. 基于 Kinect 深度信息的手势识别[J]. 北京信息科技大学学报: 自然科学版, 2013, 28(1): 22-26.)
- [19] CAO C, LI R, ZHAO L. Hand posture recognition method based on depth image technology [J]. Computer Engineering, 2012, 38(8): 16-21. (曹维清, 李瑞峰, 赵立军. 基于深度图像技术的手势识别方法[J]. 计算机工程, 2012, 38(8): 16-21.)