

基于曲率的指尖检测方法

李博男 林凡

(厦门大学软件学院, 厦门, 361005)

摘要: 指尖检测是基于视觉的徒手人机交互系统的关键环节, 由于背景的复杂性和系统的实时性要求, 导致指尖的精确定位在处理速度和准确性方面存在很大问题。本文针对这一问题, 提出了一种简单高效的基于曲率的指尖检测方法。该方法首先将输入视频流基于肤色空间进行二值化, 并将二值化后的视频序列作为输入数据; 然后采用边缘检测算法提取出肤色区域的边缘(肤色区域的轮廓), 在肤色区域的轮廓上根据曲率来对类指尖的点进行检测, 并且根据类指尖点的位置关系来判定一个肤色区域是不是手; 最后通过过滤算法过滤掉误判手臂点。实验结果表明, 该方法在不同的应用背景下都具有很好的检测效果, 对光照的鲁棒性也较高, 并且能够达到实时检测的效果。

关键词: 指尖检测; 计算机视觉; 曲率; 人机交互

中图分类号: TP391 **文献标识码:** A **文章编号:** 1005-2615(2012)04-0587-05

Fingertip Detection Method Based on Curvature

Li Bonan, Lin fan

(School of software, Xiamen University, Xiamen, 361005, China)

Abstract: The fingertip is an important feature whose detection is the key connection for many vision-based barehanded human computer interaction (HCI) systems. A novel method for fingertip detection is presented. The input data is the binary image segmented from the input video stream by the skin color space. Then, the edge detection algorithm is used to extract the edge of skin region. On the contour of skin region, fingertip-like points are detected using curvature information. The positions of these fingertip-like points are used to determine whether a skin region is a hand or not. Finally, it filters out some incorrect fingertip points. Experimental results show that the proposed algorithm performs well in different backgrounds and is robust to the influence of illumination in a real-time system.

Key words: fingertip detection; computer vision; curvature; human computer interaction

近年来, 越来越多的研究开始关注基于手势识别的人机交互方案。然而, 手势识别是一个复杂的任务, 它包含了运动建模、运动分析、模式识别和机器学习等方面, 甚至涉及到心理语言学。人机交互领域中对手势识别的研究和开发还可以扩展到诸如视频监控、机器人控制、远程会议等其他相关领域^[1]。

在手势识别技术中, 数据的采集是其中比较关键的部分, 目前所用的数据采集方法主要分为两

类。一类是基于一些特定设备的数据采集方法, 该方法采用专门制作的数据手套^[2]或者让使用者戴上特定颜色的指套或手套^[3]等; 另一类则是裸手的手势识别, 即使用者无需佩戴任何其他设备, 仅靠对裸手进行数据采集(主要是基于计算机视觉的数据采集)即可完成数据输入工作。

完成数据采集后需要对手进行建模, 形成手部特征。关于手部特征的选取和建模, 相关研究人员做了大量工作^[4-6]。目前常见的手部特征主要有肤

色特征、纹理特征以及轮廓特征等。其中,对指尖位置的检测和提取是一种常用的、技术常见的解决方法。如文献[7,8]在手势分割的基础上采用模板匹配的指尖检测方法实现手势识别。文献[9]通过计算掌心与轮廓边缘的距离检测单个指尖的位置实现与虚拟的交互。当然,指尖检测也还有一些比较特殊的方法,比如采用多个摄像机检测深度获取单个指尖位置^[10]等。但由于背景的复杂性和系统的实时性要求,往往很难做到既快又好,导致指尖的精确定位仍然是一个具有挑战性的课题。

本文针对这一问题,提出一种简单高效的基于曲率的指尖检测方法。该方法首先将输入视频流基于肤色空间进行二值化,并将二值化后的视频序列作为输入数据;然后采用边缘检测算法提取肤色区域的边缘(肤色区域的轮廓),在肤色区域的轮廓上根据曲率来对类指尖的点进行检测,并且根据类指尖点的位置关系来判定一个肤色区域是不是手;最后通过过滤算法过滤掉误判手臂点。

1 输入数据和预处理

本文方法的核心思想是根据手边缘轮廓上的点的曲率来进行指尖点判定,因此,算法的第一步是从输入的二值化图像序列中提取边缘信息。

对于输入图像序列的每一帧,都获取了图像中的边缘轮廓信息。但是在帧与帧之间,这些信息相互之间是独立的,也就是说程序并不知道诸如“这一帧中的轮廓 Contour2 是由上一帧的轮廓 Contour1 发生了一定位移、或发生了一些形变、或静止不动而产生的,Contour1 和 Contour2 其实本质上是同一个物体(下文中将这样的物体定义为 Object)”这样的信息。为了在帧序列之间建立这样的信息传递,本文加入了肤色区域跟踪算法。

在算法中,一个肤色区域的轮廓(或简称为一个肤色区域)被称为一个 Object。在全局变量中,保存一个 vector< Object> objects,用于记录场景中出现的所有 Object。给每一个 Object 增加了一个 score 参量,用来评价这个 Object 在图像中出现的频率。当对一帧图像进行边缘检测时,对于某个肤色区域 region1 和全局 objects 中的某个肤色区域 region2,若它们的外接矩形(各边平行于坐标轴的外接矩形) rect1 和 rect2 的面积比在[0.5, 2.0]之间,并且 rect1 和 rect2 相交出的矩形的面积 intersectArea 达到了两个父矩形中面积较小者的面积的2倍或以上,则可判定为 region1 与 region2

是同一个 Object 的不同形态”,将对应的 region2 的 score 值增加 10.0。当一个 Object 的 score 足够大时,就可以认为这个 Object 在每一帧中都出现了。

2 类指尖点提取

2.1 基于曲率的类指尖点检测

基于曲率的类指尖点检测是本文的核心部分,它能够从肤色区域的轮廓上提取出类指尖点信息。之所以称之为“类指尖点”而不直接称之为“指尖点”,是因为此时尚无法判定对应的肤色区域是不是手。事实上,本文中关于一个肤色区域是不是手区域的判定,是要基于该肤色区域中的类指尖点的位置信息的。关于手检测的部分详见 3.1 节。

首先对第 1 节中提取出肤色区域的边缘轮廓点序列进行插值,保证相邻点之间的距离大致相等。此时,可以考虑计算边缘上点的曲率,通过设定边缘点曲率的阈值来筛选出类指尖点的集合。同时,通过边缘轮廓的曲线可以看出,由于光照的影响,轮廓并不是一条足够光滑的曲线,在小范围内轮廓可能出现凹凸不平的情况。

因此,考虑采用如下方法定义轮廓上的点 p 的曲率:给出一个常数 L ,取轮廓上 p 点的前面第 L 个点 p_1 ,及后继第 L 个点 p_2 ,则使用向量 pp_1 与向量 pp_2 的夹角 p_1-p-p_2 的余弦值作为点 p 的曲率。计算时采用向量夹角公式,即

$$\cos\alpha = \frac{pp_1 \cdot pp_2}{|pp_1| |pp_2|} \quad (1)$$

实际实现中,选取的 L 值并不是固定的常数,而是根据肤色区域边缘轮廓在图像中占空比选定的一个区间 $[L_1, L_2]$ 。 p 点的曲率被定义为在这个区间范围内的所有 L 中向量 pp_1 与向量 pp_2 的夹角 p_1-p-p_2 的余弦值最大的数。显然,每个边缘点的曲率在区间 $[-1, 1]$ 内。

求出肤色区域边缘轮廓上每个点的曲率后,通过设定一个阈值 T ,将曲率大于等于 T 的所有点筛选出来,构成一个集合。然后通过 2.3 节的类指尖点聚类方法,就可以在集合中选取若干个点作为类指尖点。

从图 1 中可以看出,曲率算法对于指尖的检测效果很好,所有指尖均被检测到。但是同时可以发现,该算法出现了一些误判点,主要是那些位于两个手指之间的凹槽的轮廓点,以及手掌末端与手臂连接处的点。手指间凹槽误判点的去除将在 2.2 节中给出,而手掌末端与手臂连接处的误判点去除将在 3.2 节中给出。

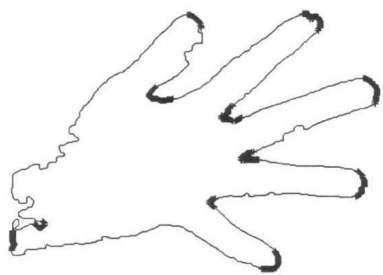


图1 基于曲率的类指尖点检测效果图

2.2 凹槽点过滤

从2.1节算法的运行效果图中可以看出,除了指尖附近的轮廓点拥有较大曲率外,那些位于两个手指之间的凹槽的轮廓点(简称为凹槽点)也有较大的曲率。因此,必须设计算法将这些凹槽点去除掉。

分别考虑图2中位于指尖的点 p 和凹槽的点 lp :

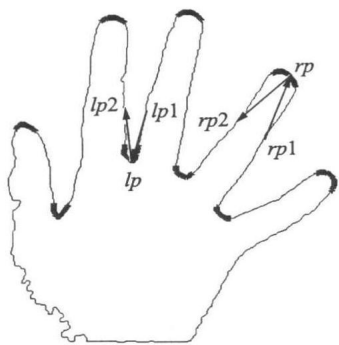


图2 凹槽点和指尖点的区别

在保证边缘轮廓点总是按逆时针序(或总是按顺时针序)排列时,分别对于点 lp 和 p ,考虑向量 $p1p$ 与向量 $pp2$ 的叉积(向量积)的符号,即 $p1p \times pp2$ 的符号,计算发现,当 p 为指尖点时叉积的符号,与当 p 为凹槽点时叉积的符号不同。

据此,就可以根据 $p1p \times pp2$ 的符号来判断某个点 p 是属于指尖点集合还是凹槽点集合,并过滤掉凹槽点。

2.3 类指尖点聚类

在2.1节中,对于肤色区域的边缘轮廓上的每一个边缘点 p 都求出了对应的曲率。然后,设定一个阈值 T ,当某个点 p 的曲率小于 T 时则将其曲率置为 -1 ,否则保持不变,筛选之后效果如图1所示。

中地、连续地分布在手指指尖附近。如果考虑环状轮廓点序列所对应的环状曲率序列,那么序列中大部分的值都是 -1 ,少部分非负的值均集中地、连续地分布在若干个一维区间内。因此,只要设计一个聚类算法,将每个一维区间找出,并从每个一维区间中选出一个代表点作为聚类的结果即可。

在本文的算法设计中,每个一维区间允许出现极少量的曲率为 -1 的点(这样的点不允许出现在区间端点处)。对于每个一维区间,选取其中的中位点(当一维区间的下标范围是 $[1,]$ 时,称下标为 $(1+)/2$ 的点为区间的中位点)作为其代表点,即候选的类指尖点。聚类效果如图3所示。

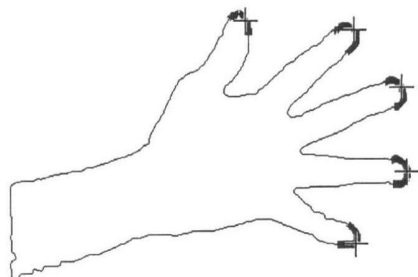


图3 类指尖点聚类效果

可见,该算法对于每一个区间选出了中位点作为类指尖点的代表,可以反映指尖的位置,能够达到较好的效果。

3 误判信息纠正

3.1 手检测

上文中考虑的所有算法的出发点都是与手或者指尖相关的。但实际上,分割出的肤色区域有可能是手,也有可能是脸或其他人体部位。因此,有必要设计一个算法来判断一个肤色区域是不是手。

在本文中,如果一个肤色区域已经被判定为“是手”,那么在第1节肤色区域跟踪的算法下,在之后视频流的每一帧中凡是被判定为与该肤色区域属于同一个Object的肤色区域就一定是手,即假定不可能出现一个肤色区域曾经是手后来变成了非手的情况。在这样的条件下,设计如下的手判定算法:

(1) 若一个肤色区域从进入视频开始,经过 T 帧还未判定为“是手”,则认定该肤色区域永远不可能是手。在之后的肤色区域跟踪过程中直接跳过对该区域的手判定过程。

(2) 对于在 T 帧内又尚未判定为“是手”的肤色区域,采用类似第1节肤色区域跟踪的算法:给

每一个肤色区域(即每个Object, Object的定义请参见第1节)增加一个hand-score参量,初始时hand-score为0。每当某个肤色区域被判定为“类手区域”时(判定方法见下文),若该区域的hand-score不足10.0则设为10.0,否则将该区域的hand-score增加10.0。当一个肤色区域的hand-score值超过50.0时,即可认为该区域已经足够多次满足“类手区域”的要求,可以判定为“是手”。反之,若该区域的“类手区域”判定失败,则将其hand-score减去5.0。

(3) 判定一个区域是否是“类手区域”时,采用如下算法。

① 若该区域中的类指尖点少于5个或大于7个,则该区域判定为“非类手区域”。

② 遍历该区域的所有类指尖点 p_1 。每次遍历时,取出 p_1 之后的4个点 p_2, p_3, p_4, p_5 (算法的目的是将这5个点判定为5根手指的指尖点),并求出相邻点之间的距离(Euclidean距离)。将4个距离由小到大排序得到序列 $\{d_1, d_2, d_3, d_4\}$ 。

③ 若 p_1, p_2, p_3, p_4, p_5 分别为一个手掌摊开、5指张开的手的5个指尖,一般地,大致有 $d_1 > 0.4 * d_2, 1.2 * d_2 > d_3, 3 * d_3 > d_4$ 。若满足以上条件,则判定该区域是“类手区域”。

④ 若遍历完所有的类指尖点 p_1 均不能判定该区域为“类手区域”,则该区域判定为“非类手区域”。

⑤ 当一个区域被判定为“类手区域”后,修改该区域的类指尖点列表,只保留 p_1, p_2, p_3, p_4, p_5 ,其余类指尖点均删除。这样做的目的是为了在3.2节中判定初始手向向量时,不被除指尖点外的其余误判点干扰,影响算法正确性。

以上就是判断一个肤色区域是不是手的全过程。显然,若能够判断一个肤色区域不是手,则没有必要对其进行轮廓点插值、类指尖点检测等一系列操作。在提高算法准确性的同时也提高了运行效率。

3.2 手臂点过滤算法设计

在2.1节中,曾经提到过手掌末端与手臂连接处的边缘点可能被误判为类指尖点(称为“误判手臂点”,简称“手臂点”)的问题。因此,需要设计一个算法来去除这样的“手臂点”,保证整体算法的正确性。

为了去除手臂点的干扰,需要求出一个手向向量 V ,用于反映当前手的大致朝向。计算手向向量 V 主要有两种类型,(1)适用于当手第一次出现时的

图像,(2)适用于之后的图像,即已知上一帧中手的手向向量,要求这一帧的手向向量。

对于第1种情况,即当一只手刚进入图像时,取手的最小包围矩形minRect(各边不一定平行于坐标轴的最小外接矩形)的4个顶点和各边中点共8个点,分别计算这8个点到该手内所有类指尖点的距离和(此处“距离”的定义采用Euclidean距离),选取其中距离和最大的点作为初始手向向量的起点。确定了向量起点后,再选择“对面”的点作为向量的终点。这里“对面”的定义对于minRect的顶点而言,则是对角顶点;对于minRect的某条边的中点而言,则是对边中点。

对于第2种情况,此时可用“上一帧时该手的手向向量”预测“这一帧时该手的手向向量”,同时再结合该帧中的数据即可。

手向向量 V 的计算效果如图4,5所示:

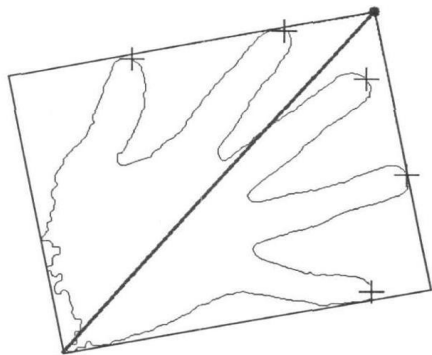


图4 长袖穿着时的手向向量

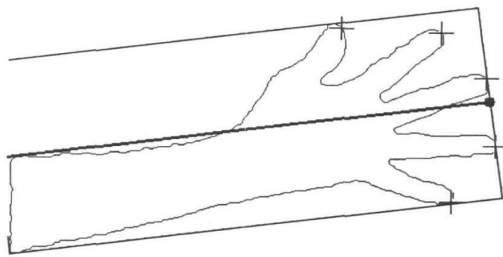


图5 短袖穿着时的手向向量

图中深色直线表示手向向量 V ,直线上以圆点结尾的一端为手向向量 V 的终点。实际运行中能够不断跟踪出正确的手向向量 V 。

计算出手向向量 V 后,就能够回答“哪边是手臂,哪边是手指”这个问题了。下面需要借助物理学中“场”和“势”的概念:假设在该平面中存在一个平行场 F ,手向向量 V 为场强方向,定义 V 的起点的势

为0, V 的终点的势为1, 并假设势沿场强方向均匀增加。

这样, 就可以根据场 F 来计算平面中任意一个点的势。那么对于包括手臂点在内的所有类指尖点, 分别计算该类指尖点的势, 并将势太低的类指尖点判定为手臂点, 去掉即可。

经过了以上算法后, 剩余的类指尖点即可判定为指尖点。

4 结束语

本文的核心是基于曲率的类指尖点检测。为实现基于曲率的类指尖点检测, 首先需要进行各类预处理, 例如边缘检测、跟踪、边缘点插值等, 为类指尖点检测提供基本条件; 然后去除误判点, 例如去除“凹槽点”、“手臂点”等, 将“类指尖点”确认为“指尖点”。

该算法的大部分操作是基于“类指尖点”的, 而“类指尖点”的数量在每一帧中很少(一般不超过10个), 因此算法整体效率较高, 能够满足系统的实时性需求。

但是该算法也存在一定的缺陷, 由于算法从根源上是基于边缘检测的, 因此, 要求输入的视频流中手部区域必须与其他肤色区域分离。否则, 若手部区域的背景也是肤色区域(例如脸)或者与肤色颜色非常相近的区域, 算法将无法检测到手部边缘, 因此无法进行下一步“类指尖点检测”, 整个算法就无法继续下去。

存在这一缺陷的根本原因在于单视频流中只能获取2D信息。针对这一问题, 可以考虑采用多个摄像头从多角度对手进行拍摄并建立3D模型, 或者加入深度传感器, 获取深度信息并建立3D模型。

参考文献:

[1] Wu Ying, Huang T S. Vision-based gesture recognition: a review [J]. Lecture Notes in Computer Science, 1999, 1739/1999: 103-115.

[2] Weissmann J, Salomon R. Gesture recognition for virtual reality applications using data gloves and neural networks[J]. International Joint Conference on Neural Networks (IJCNN'99), 1999(3): 2043-

2046.

[3] Iwai Y, Watanabe K, Yagi Y, et al. Gesture recognition using colored glove[J]. Proceedings of the 13th International Conference on Pattern Recognition, 1996(1): 662-666.

[4] Bretzner L, Laptev I, Lindeberg T. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering[C]. Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FG'02). [S.l.]: IEEE, 2002: 423-428.

[5] Cui Y, Swets D, Weng J. Learning-based hand sign recognition using SHOSLIF-M [C]. International Workshop on Automatic Face and Gesture Recognition. Zurich:[s. n.], 1995: 201-206.

[6] Wren C R, Pentland A P. Dynamic models of human motion [C]. Third IEEE International Conference on Automatic Face and Gesture Recognition. [S. l.]: IEEE, 1998: 22-27.

[7] 杨端端, 金连文, 尹俊勋. 手指书写汉字识别系统中的指尖检测方法 [J]. 华南理工大学学报, 2007, 35(1): 58-63.

Yang Duanduan, Jin Lianwen, Yin Junxun. Fingertip detection approach for finger writing chinese character recognition system [J]. Journal of South China University of Technology, 2007, 35(1): 58-63.

[8] Zeng Jianchao, Wang Yue, Turner R, et al. Color feature finger tracking for breast palpation quantification [C]. Proceedings of IEEE International Conference on Robotic and Albuquerque. Albuquerque, NM, USA: [s. n.], 1997: 2565-2570.

[9] Lee M, Green R, Billinghurst M. 3D natural hand interaction for AR applications [C]. 23rd International Conference Image and Vision Computing. Christchurch, New Zealand: [s. n.], 2008: 6-12.

[10] Ying Hongwei, Song jiatao, Ren Xiaobo, et al. Fingertip detection and tracking using 2D and 3D information[C]. 7th World Congress on Intelligent Control and Automation. Chongqing, China: [s. n.], 2008: 1149-1152.