文章编号:1003－6199(2024)02－0177－05 DOI:10.16339/j.cnki.jsjsyzdh.202402029

基于 OpenCV 与 MediaPipe 的面部动作控制鼠标操作技术

王骏祥†,马健为,宋笛秋,朱思哲

(南开大学 软件学院,天津 300457)

摘 要:为丰富人机交互模式,基于 OpenCV 和 Mediapipe 在面部识别中的应用,在个人电脑上实现通过面部动作进行常用的鼠标控制。具体的方法是通过捕捉摄像头传入视频帧中用户脸部的关键点,并根据关键点的位置变化来执行对应鼠标操作。实际测试结果表明:该程序仅通过简单的面部动作就可以较好地实现诸如移动光标、单击鼠标左右键以及拖动滚动条等常用鼠标操作,且识别准确率高,是一种针对特殊用户和特定环境条件下替代鼠标操作的低成本实用技术。

关键词:MediaPipe; OpenCV; 面部识别; 鼠标操作

中图分类号:TP317.1 **文献标识码**:A

Research on Facial Action Control Mouse Operation Based on OpenCV and MediaPipe

WANG Junxiang†, MA Jianwei, SONG Diqiu, ZHU Sizhe

(College of Software, Nankai University, Tianjin 300457, China)

Abstract: In order to enrich the human-computer interaction methods, this work realizes common mouse control through facial movements on PC based on the application of OpenCV and MediaPipe in facial recognition. The specific method is to capture the key points of user's face in the video frame transmitted by the camera, and perform corresponding operations according to the position changes of the key points. The actual test results show that this program can effectively achieve common mouse operations such as moving the cursor, clicking the left and right mouse buttons, and dragging the scroll bar through simple facial movements, and has high recognition accuracy. It is a low-cost practical technology that can replace mouse operations for special users and specific environmental conditions.

Key words: MedaiPipe; OpenCV; face recognition; mouse-control

人机交互是研究人与计算机及其相互作用的技术,包括交互的方式、方法、设备和界面[1],其目的是实现人与计算机的双向沟通。鼠标光标操作是当前最主要的人机交互方式之一,但该技术存在着残障或手部劳损人士使用不便和使用受场合受限等缺陷[2]。因此有必要开发替代鼠标操作的技术。

当前国内外研究人员在开发可替代鼠标操作的计算机操作方面做了大量的研究。现有的新型人机交互技术大致包含手势控制、语音控制和面部动作控制三类。手势控制方面有美国 Ultraleap 公司推出的 Leap Motion 手势控制仪,国内陈敬宇等[3]也研究了基于手势识别的人机交互系统。该技术采用人手势变化实现鼠标操作功能,可以避免

收稿日期:2024－02－04

作者简介:王骏祥(2003—),男,安徽郎溪人,本科生,研究方向:机器学习。

† **通信联系人**,E-mail: 2018824763@qq.com

长期操作鼠标而造成的关节劳损，但手势识别技术常常需要依赖于较为昂贵的硬件，且不适合于手部残障的用户，另外，频繁进行幅度较大手部动作同样会使用户感到疲劳。研究语音控制技术的有美国微软公司，该公司在 Windows 11 操作系统中提供语音访问功能，国内朱全胜等人[4]也研究了语音交互识别系统，该技术仅仅依赖麦克风进行语音采集，就能实现鼠标的操作功能，对硬件要求低，但语音控制无法在嘈杂的环境下使用，且还会影响周围的人。面部动作控制技术主要通过和计算机连接的摄像头捕捉操作者的面部表情或动作，并将其转化为对应的鼠标操作，称为相机鼠标技术。当下的相机鼠标技术有多种实现方式，Vasanthan 等人[5]使用了微传感器和粘贴在面部的发光贴纸，通过监控贴纸标记的轮廓来实现鼠标动作控制。Mohamed 等人[6]则将摄像头传入的影像转化为灰度图后使用支持向量机(SVM)技术来识别面部位置以及实际坐标来实现鼠标动作控制。Gyawal 等[7]使用了特殊的眼部监测算法，通过在图片中定位眼部位置来控制鼠标动作。微软公司在 Windows 11 中提供视线控制以及美国 Oculus 公司推出的虚拟现实装置 Oculus Quest2 都是基于眼动控制鼠标动作的技术。相对于采用语音和手势控制，采用面部动作实现鼠标控制技术可以很好地克服语言控制干扰他人以及手部控制易产生疲劳的缺陷。然而，上述通过面部动作控制鼠标的技术也或多或少存在一些缺陷，如在脸部粘贴发光贴纸的做法操作麻烦也不符合大多数用户的习惯；Mohamed 和 Gyawal 等人的相机鼠标技术识别精度较低，需反复操作多次才能完成预期操作，并且，在识别精度以及操作逻辑的限制下，双击或更加复杂的操作难以实现；微软公司的视线控制和 Oculus 公司虚拟现实装置都需要配置昂贵硬件的系统。

为了克服上述技术中的不足，从机器视觉的角度出发，对发展前景好的相机鼠标技术进行优化，通过面部动作来实现常用的鼠标操作，为特殊需求的人群和不便于进行鼠标操作的场合提供一种简便易行的人机交互方式。本文采用基于 OpenCV 库以及由美国 Google 公司推出的 MediaPipe 库来实现这一技术的开发。本技术仅需使用与计算机连接的摄像头作为额外硬件或利用显示器屏幕上自带的摄像头，通过捕捉摄像头传入视频帧中用户脸部的关键点的位置变化来执行诸如光标移动，鼠标左、右键单击和滚轮滚动等基本鼠标操作[8]。

1 面部动作实现鼠标操作的解决方案

相机鼠标技术的主要难点在于如何精确识别用户的有意识面部动作，并将不同的动作以合理的方式映射到鼠标动作的对应操作上。过去的技术因为面部识别的精度有限，难以识别用户的微小动作，只能通过识别头部转动等较大幅度动作对电脑进行操作。为解决这类的问题，基于 OpenCV 和 Mediapipe 在面部识别中的应用，在个人电脑上实现通过面部动作进行常用的鼠标控制。

1.1 OpenCV

OpenCV 是一个开源的计算机视觉和机器学习软件库，为计算机视觉应用提供通用的架构，并加速商业产品中的机器感知而构建。OpenCV 基于 Apache 2.0 许可(开源)发行，包含了超过 2500 种优化过的计算机视觉以及机器学习算法。OpenCV 为 C++、Python、JAVA、MATLAB 语言提供接口，并且支持 Windows、Linux、Android、以及 Mac OS 操作系统[9]。本应用主要使用了其中的摄像头调用，图像放大处理等计算机视觉领域的算法。

1.2 MediaPipe

MediaPipe 是一个基于 Apache 2.0 许可(开源)发行的，为流媒体及直播提供的跨平台机器学习解决方案，支持面部检测、手势检测、物体追踪、人体骨骼检测等多种功能。MediaPipe 的核心框架由 C++实现，并且为 Python、Java 等语言提供支持[10]。本技术中主要使用了 MediaPipe 的 face_mesh 模块。该模块可以识别图像中是否有人脸，若包含则将人脸部分进行统一大小的裁剪，然后将裁剪后的图像传入 Attention mesh 模型中提取特征，定位、标记出 468 个面部关键点(图 1)，并能时刻向电脑中返回这些关键点在图像中的位置[11][12]。

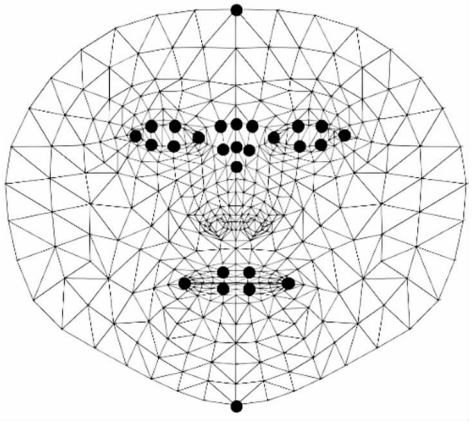

图 1 468 个面部关键点以及对应标号

MediaPipe 提供的算法可以在较低性能消耗的前提下快速识别出人脸，并精确返回面部关键点坐标。这使得通过识别出更小幅度的动作（例如眨眼）成为可能。为了平衡识别的准确率以及操作的便利性，本程序选用图 1 中加粗显示的 27 个特征点，设定一系列更加精确的特定动作进行操作，并引入一个简单的判断算法，在适应用户使用习惯的同时过滤用户的无意识动作，从而提升操作的精确度。

2　程序开发环境和工作过程

2.1　开发环境

采用的编程语言是 Python 3.10，集成开发环境是 PyCharm，依赖库有：MediaPipe 0.10.9，PyQt6 6.6.1，PyAutoGUI 0.9.54，NumPy 1.26.3，OpenCV-contrib-Python 4.9.0.80 和 OpenCV-Python 4.9.0.80。其中核心依赖库为 OpenCV 和 MediaPipe。

2.2　程序工作过程

程序的主要功能为持续调取摄像头传入的图像信息，识别并判断用户脸部是否进行了特定动作，以决定是否执行对应的鼠标光标操作。程序工作流程如图 2 所示。

上述面部特定动作包含用户头部的转动、同时眨双眼、张嘴和左右倾斜头部。用户可以通过这四个动作分别执行移动光标、单击鼠标左键、单击鼠标右键和滑动滚轮四种操作。下文中将给出该程序的核心思路。

2.3　程序的实现

2.3.1　面部关键点的捕捉和动作识别

（1）获取面部关键点坐标

MediaPipe 的 face_mesh 方法可以将视频帧中的人脸识别为关键点，并返回所有关键点在视频帧中的相对位置。为了实现对特定动作的观测，程序需要读取包含眉心 7 个，双眼眼周共 12 个，嘴部 6 个，以及头顶、下巴下端 2 个共 27 个识别点的坐标，并观测坐标的变化。

（2）获取视点位置作为视线追踪的一种低成本近似，程序读取两眼之间的眉心位置的 7 个识别点的横、纵坐标平均值作为眉心坐标，并将这个坐标在屏幕上的位置、在不同时间位置的差异，作为光标移动的依据。

（3）判断是否眨眼

为了识别用户是否进行眨眼动作，程序按照图 3 的方式（以右眼为例）读取左、右眼的面部关键点。

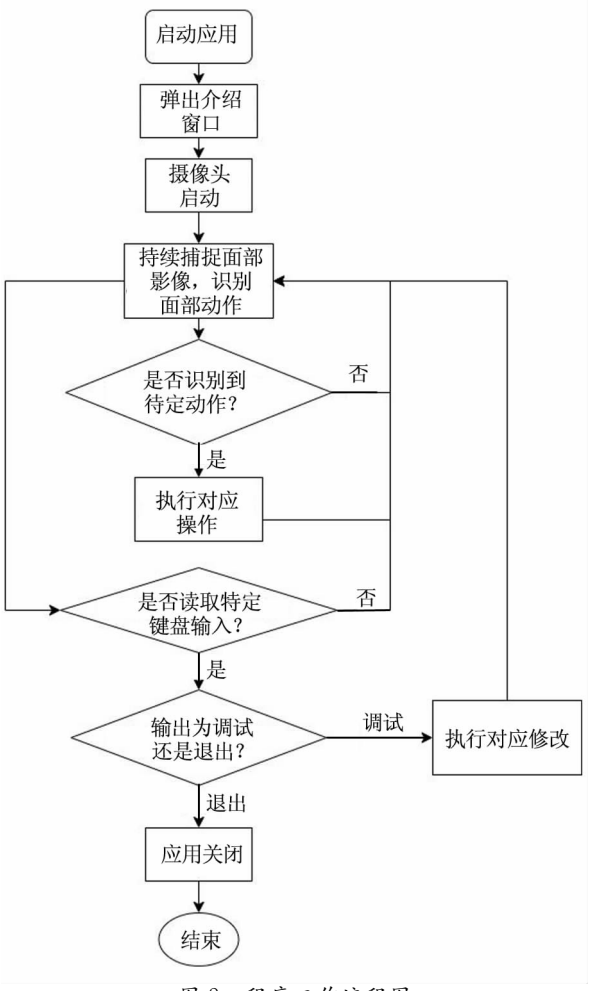

图 2　程序工作流程图

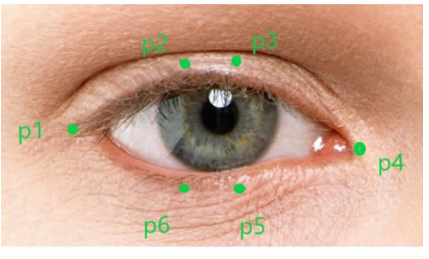

图 3　以右眼为例的眼部关键点示意图

程序将按照公式（1），计算图 3 中 p2、p6 和 p3、p5 两组对应点的欧式距离以及 p1、p4 欧氏距离的比值的平均值（e_ratio）作为判定用户眼睛状态的依据。当 e_ratio 值减小时，用户的上下眼睑靠近，可以认为用户正在闭上眼睛；反之，可以认为用户正在睁大眼睛。程序将按照相同的方法计算左眼的 e_ratio_l 值，并以这两个数值的平均值作为 e_ratio 的值。

$$e\_ratio\_r(l) = \frac{\| p_2 - p_6 \| + \| p_3 - p_5 \|}{2 \| p_1 - p_4 \|} \tag{1}$$

为了解决固定阈值无法应对不同用户眼睛大小不同的问题，该程序维护一个大小为 5 的数组e_ratiolist，数组将存放最近 5 帧中的 e_ratio 值。考虑到用户无意识眨眼一般发生在程序运行的1～2帧以内，程序将 e_ratiolist 的平均值作为用户不眨眼时的参考值。每当摄像头传入新的一帧时，数组将从数组最前端移除最旧的 e_ratio 值，并将最新值添加在数组尾端。若在最新一帧中的 e_ratio 值小于 e_ratiolist 元素平均值与特定阈值 eye_thresh 的和时(阈值可调节，用于调节识别的灵敏度)，程序认为用户正在进行一个眨眼的动作，并开始累积 e_ratio 值小于 e_ratiolist 均值＋eye_thresh 的连续帧数。当这个数字大于另一特定阈值 closed_eyes_frame(长于自然眨眼所用时长)，并且最新一帧的 e_ratio 值大于 e_ratiolist 平均值和 eye_thresh 和时，程序识别出用户完成一次有意识的眨双眼动作。

(4)判断是否张嘴

判断是否张嘴的方法和判定是否眨眼非常相似。按照在眼沿选取关键点的方式取上唇两个，下唇两个，以及左、右嘴角共 6 个关键点，并按照公式 1 的计算方式计算出嘴部比例 m_ratio，同样维护一个大小为 5 的数组 m_ratiolist 记录用户自己的常态嘴部大小。与判断眨眼相反的是，程序识别张嘴的动作，所以程序会在当前 m_ratio 值大于 m_ratiolist 与可调阈值 mouth_thresh 的和时，程序开始累积这个保持这个状态的连续帧数。当帧数大于特定阈值 opened_mouth_frame 时，程序识别到有意识的张嘴动作。

(5)判断头部倾斜

程序计算头顶、下巴下端两个识别点的横轴坐标差值 inc。在头部向不同方向倾斜的时候，inc 的正负、大小都将相应发生变化。这个差值将作为滚轮操作的依据。

2.3.2 鼠标控制的实现

(1)光标的移动

程序将维护一个大小为 2 的数组，用于存储当前帧和上一帧中两个眉心坐标。当两个坐标的欧式距离大于某个阈值时，鼠标光标将按照一定的比例按照横、纵坐标的变化值在电脑屏幕上运动。

(2)鼠标的左、右键操作

当程序成功识别到一次有意识的眨双眼动作时，程序将执行鼠标左键单击一次的操作。当成功识别到一次有意识的张嘴动作时，程序将执行鼠标右键单击一次的操作。

(3)鼠标滚轮的移动

当头顶和下巴下端两个识别点的横轴坐标差值(inc)大于一个特定的正值时，程序识别出用户在向左歪头，并执行向上滚动滚轮的操作。而 inc 值小于一个特定的负值时(绝对值和正值相同)程序识别出用户在向右歪头，并执行向下滚动滚轮的操作。

2.3.3 全局参数以及调整

本程序包含了多个全局变量。这些全局变量控制着程序的具体运作，用户也可以根据自己的需要以及喜好对其进行修改。下面将对程序使用到的全局变量以及程序校准方法进行介绍：sensitivity：在 mouse_move()函数中，为了防止光标的抖动，程序只在检测到视点移动欧氏距离超过 sensitivity 值时，程序才会进行光标位置的移动。用户可以通过键盘输入“w”和“e”键来控制 sensitivity 的值。考虑到键盘使用困难的可能，这一操作可以通过系统自带的屏幕键盘实现。

eye_thresh：是用于判定眼睛是否闭上的阈值。只有当用户当前的 e_ratio 值小于 e_ratiolist 列表中值的平均值＋eye_thresh 值时，程序认定用户开始闭眼，才会开始累积闭眼判定帧数。eye_thresh 值越大，判定闭眼的标准越高。eye_thresh 的值可以通过键盘输入“u”和“d”来修改。

closed_eyes_frame：是判定闭眼帧数的阈值。当 e_ratio 大于 e_ratiolist 均值和 eye_thresh 的和时，程序将对 e_ratio 连续小于 e_ratiolist 均值和 eye_thresh 的帧数进行判定。若超过了 closed_eyes_frame，则执行左键单击操作。closed_eyes_frame 可以通过键盘输入“j”和“k”来增大或减小。

mouth_thresh：原理和 eye_thresh 相近，用户可以通过键盘输入“v”和“p”来增大或减小 mouth_thresh 的值。

opened_mouth_frame：原理和 closed_eyes_frame 相近，用户可以通过输入“m”和“n”来增大或减小 opened_mouth_frame 的值。

2.4 实现效果

运行程序时，光标位置被初始化在屏幕正中心。通过转动头部、张嘴、眨双眼、以及歪头的动作，我们依次完成选定文件夹、打开右键菜单、打开文件夹、向下滚动页面的操作。过程如图 4 所示。

经过实际反复测试，光标移动，页面滚动响应快速。程序识别用户张嘴动作的正确率为 100%。程序在用户不佩戴眼镜时识别用户眨双眼动作的正确率为 88.89%，在用户佩戴眼镜时识别正确率为 82.22%，准确率较高。与传统的光标操控方式，如鼠标、触摸板、键盘控制方法相比，本文提供的面部动作控制方法完全摆脱了对双手控制的依赖，校准和调整也可以通过操作系统内置的屏幕键盘完成；

相比于语音识别控制光标，本文提供的方法对输入的响应更快，操作更加方便、迅速；而相比手势控制光标方法，本文提供的方法无需手部的任何动作，在操作更加轻松的同时更加符合用户习惯。

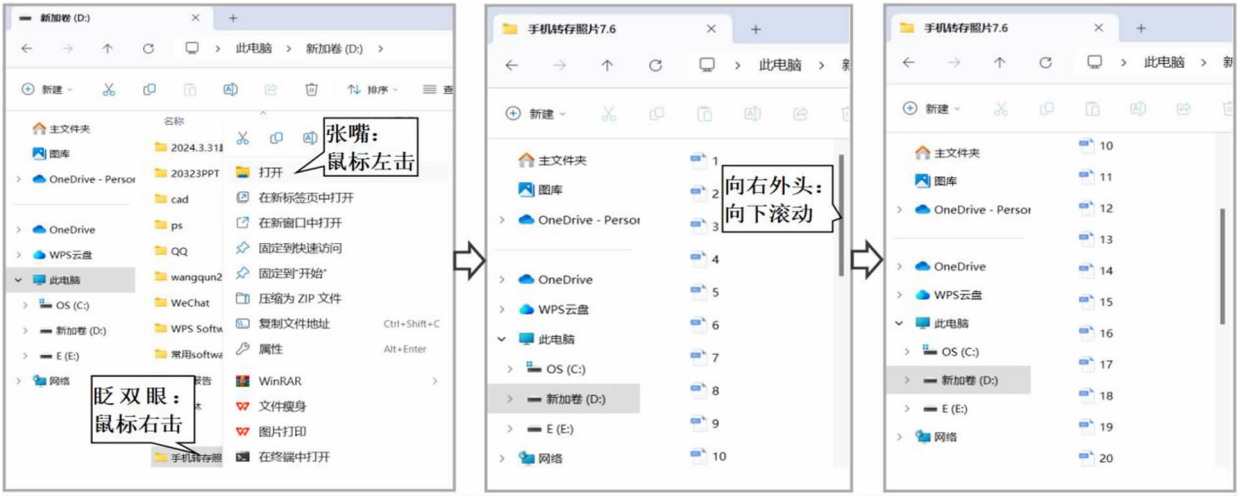

图 4　程序运行效果

从人机交互的发展进程来看，当下正处于可以通过语音、手势等多样方式进行交互的智能人机交互阶段[14]，本文实现了面部动作对光标控制，通过不同的面部动作控制光标，识别速度快，准确率较高，性能要求低，在光标操作中解放了用户的双手，适应了更广泛的使用人群，是智能人机的交互方式的一个有益补充。这些智能人机交互的操作方式越发多样，操作效率不断提高，操作本身的自然性，也就是操作是否符合直觉，也越来越被强调。这样的趋势与深度学习、计算机视觉技术的发展、使用门槛的降低息息相关。今后，人机交互会继续向着多通道的方向发展，可以读取、识别用户包含言语、表情、手势、视线、姿态等多种信息[15]，使得操作效率、便利性以及无障碍程度得到进一步的发展。

3　结　论

使用 Python 编程调用 OpenCV 和 MediaPipe 等库，并通过识别摄像头输入的用户面部动作实现了光标移动，鼠标左键、右键单击，滚轮滑动的操作。该程序具有识别准确率高、响应快速、操作简单、成本低等优点。经进一步完善后，有望成为一种有发展前途的替代鼠标光标操作的实用技术。

参考文献

[1] 董士海. 人机交互的进展及面临的挑战[J]. 计算机辅助设计与图形学学报，2004(1)：1－13.

[2] 李龑，胡立夫，王嘉浩. 一种基于 OpenCV 的残障人士计算机交互系统设计[J]. 中国科技信息，2021(17)：62－64.

[3] 陈敬宇，徐金，罗容，等. 基于手势识别的 3D 人机交互系统[J]. 现代信息科技，2023，7(22)：88－91.

[4] 朱全胜，刘娆，李卫东. 语音识别技术应用于 EMS 人机交互初探[J]. 电力系统自动化，2008(13)：45－48＋100.

[5] VASANTHAN M，MURUGAPPAN M，NAGARAJAN R，et al. Facial expression based computer cursor control system for assisting physically disabled person[C]. 2012 IEEE International Conference on Communication，Networks and Satellite (ComNetSat)，2012：172－176.

[6] MOHAMED A W，KOGGALAGE R. Control of mouse movements using human facial expressions [C]. 2007 3rd International Conference on Information and Automation for Sustainability，2007：13－18.

[7] GYAWAL P，ABEER ALSADOON，PRASAD P W C，et al. A novel robust camera mouse for disabled people (RCMDP)[C]. 2016 7th International Conference on Information and Communication Systems (ICICS)，2016：217－220.

[8] 谢鹏. 基于 OpenCV＋MediaPipe 实现运动姿态 AI 检测在体育训练中的应用[J]. 无线互联科技，2023，20(18)：100－104.

[9] OpenCV team. About[EB/OL]. htpp://opencv.org/about，2024.

[10] Google team. MediaPipe Solutions guide. [EB/OL]. http://developers. google. com/mediapipe/solutions/guide，2023－11－09.

[11] Google team. Face landmark detection guide[EB/OL]. http://developers. google. com/mediapipe/solutions/vision/face_landmarker＃get_started，2024－1－23.

[12] 王福友. 基于面部特征的驾驶疲劳检测应用研究[D]. 呼和浩特：内蒙古工业大学，2023.

[13] ISCHMELZEISEN L S. Understanding-mediapipe-facemesh-output[EB/OL]. https://github. com/lschmelzeisen/understanding-mediapipe-facemesh-output，2022－8－22.

[14] 黄进，韩冬奇，陈毅能，等. 混合现实中的人机交互综述[J]. 计算机辅助设计与图形学学报，2016(6). 869－880.

[15] 贾计东，张明路. 人机安全交互技术研究进展及发展趋势[J]. 机械工程学报，2020(3). 16－30.