УДК 004

МЕТОДИКА ПОСТРОЕНИЯ ЧЕЛОВЕКО-МАШИННОГО ИНТЕРФЕЙСА ДЛЯ УПРАВЛЕНИЯ КОМПЬЮТЕРОМ НА ОСНОВЕ БИБЛИОТЕКИ MEDIAPIPE

Б. Я. Колегаев
Донской государственный технический университет (г. Ростов-на-Дону, Российская Федерация)

Аннотация. На смену кнопочным телефонам уже давно пришли сенсорные смартфоны. Доказано, что управление собственными руками является наиболее интуитивно понятным способом взаимодействия с устройством. Однако технология сенсорного управления достаточно дорога, чтобы использовать ее на персональных компьютерах и не так удобна из-за большого размера экрана монитора. Флагманские модели современных смартфонов уже начали использовать технологию бесконтактного управления устройством, это новый виток развития пользовательских интерфейсов. Автором статьи сделана попытка реализовать функции бесконтактного управления компьютерной мышью, уровнем звука, а также голосовым интерфейсом для поиска информации в Интернете при управлении компьютером с операционной системой Windows 10. Целью работы является обеспечение удобства пользователя и скорости взаимодействия его с персональным компьютером посредством реализации бесконтактного управления.

Ключевые слова: человеко-машинный интерфейс, Python, распознавание жестов, mediapipe.

METHOD FOR DESIGNING A HUMAN-MACHINE INTERFACE FOR CONTROLLING COMPUTER BASED ON MEDIAPIPE LIBRARY

Boris Ya. Kolegaev
Don State Technical University, (Rostov-on-Don, Russian Federation)

Abstract. Touch-sensitive smartphones have long replaced push-button phones. It is proved that hand-control is the most intuitive way to interact with the device. However, touch-sensitive control technology is expensive enough to use on personal computers and is not so convenient due to the large size of the monitor screen. Flagship models of modern smartphones have already started using contactless device control technology. This is a new stage in the development of user interfaces. The author of the article attempts to implement the functions of contactless control of a computer mouse, sound level, as well as a voice interface for searching information on the Internet when controlling a computer with the Windows 10 operating system. The work objective is to ensure user convenience and the speed of its interaction with a personal computer through the implementation of contactless control.

Keywords: human-machine interface, Python, gesture recognition, mediapipe.

Введение. В работе использована методика построения человеко-машинного интерфейса для управления компьютером. Команды на компьютер подаются с помощью жестов человеческой руки. Распознавание выполнено на основе библиотеки mediapipe. Для достижения заявленных целей по обеспечению взаимодействия пользователя с ПК необходимо решить следующие задачи:

1) реализация бесконтактного управления компьютерной мышью (движение, нажатие кнопок мыши и скроллинг);

2) реализация бесконтактного управления уровнем звука на компьютере;

3) обеспечение возможности переключения режимов управления мышью и звуком посредством голосового интерфейса;

4) реализация возможности использования голосового интерфейса для поиска информации в Интернете.

В качестве языка программирования был выбран Python из-за большого количества представленных на нем библиотек. В проекте использовалась объектно-ориентированная парадигма для удобства расширения возможностей программы в будущем.

Основная часть. Задача обнаружения ладони. Одним из подходов к детекции человеческой руки является использование сегментации по цвету и поиск контура руки.

Плюс данного подхода заключается в его простоте. Стандартными методами библиотеки OpenCV можно выделить на изображении те участки, цвет пикселей которых будет находиться в заданном диапазоне. Так, например, можно выделить на изображении области, в которых присутствует цвет кожи человека, затем найти замкнутые контуры и получить в результате программу для обнаружения руки.

Минусом такого подхода является сложность дальнейшей обработки изображения, ведь для движения мышью и клика придется использовать жесты, а это потребует определения положения пальцев. Помимо прочего, достаточно трудно подобрать цветовой диапазон, в котором рука будет успешно сегментироваться по цвету при различном освещении. Пример такого подхода для обнаружения руки представлен в работе [1].

Другим подходом является использование машинного обучения для решения данной задачи. Анализ существующих методов решения проблемы распознавания человеческой руки привел к выбору библиотеки mediapipe [2]. Данная библиотека разработана компанией Google и содержит заранее предобученные и оптимизированные для работы на процессоре нейронные сети, которые позволяют обнаружить на изображении человеческое лицо, тело, руки и в том числе ладони [3]. Mediapipe определяет на ладони 21 отдельную точку и их координаты. Пример этих точек представлен на рис. 1.

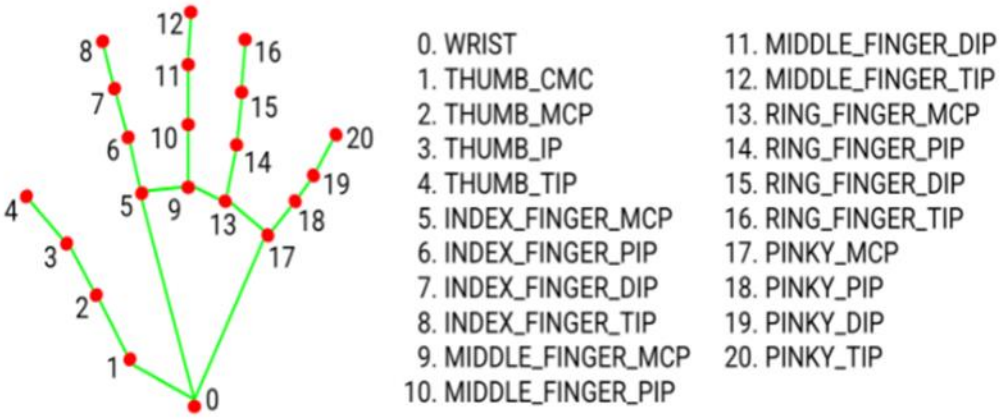

Рис. 1. Точки, обнаруживаемые с помощью mediapipe [4]

Информацию о координатах каждой точки можно использовать для обнаружения конкретных жестов.

Движение и клик. Для управления экранной мышью был выбран жест с поднятым указательным пальцем. Как видно на рис. 2, указательный палец поднят, если координата точки Y 8 больше координаты точки Y 7. По схожему принципу легко задать правило и для опущенных пальцев.

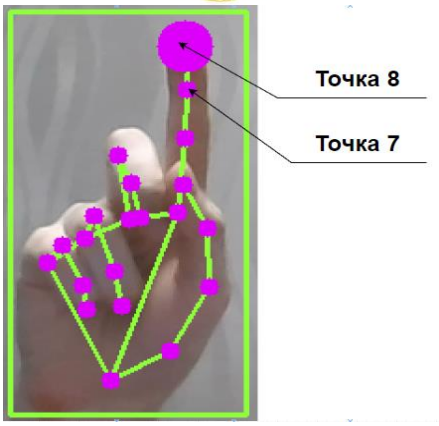

Рис. 2. Пример жеста для управления экранной мышью (составлено автором)

Для клика левой кнопкой мыши был выбран жест с одновременно поднятыми указательным и средним пальцами, а для клика правой кнопкой — поднятым мизинцем (рис. 3).

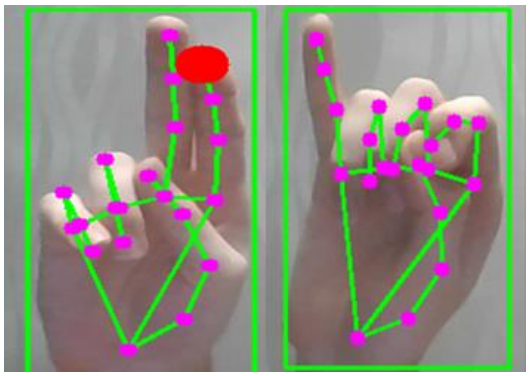

Рис. 3. Пример жеста для левого клика (слева) и для правого (справа) (составлено автором)

Следующим этапом после определения жестов была привязка координат точки 8 (рис. 2) к координатам самой экранной мыши. Для этой цели была использована библиотека autopy, которая позволяет автоматизировать движение мыши путем задания конкретных координат для перемещения [5].

Таким образом, остается лишь передать найденные ранее координаты в метод autopy.mouse.move(X, Y). Для клика используется метод autopy.mouse.click().

Оптимизация работы алгоритма. После тестирования проекта были выявлены две проблемы:

1. При выходе курсора за пределы области экрана скрипт завершал работу с ошибкой.
2. Курсор следовал за пальцем руки недостаточно плавно.

Для устранения первой проблемы было решено считывать положение пальцев лишь в определенной области экрана, конвертируя координаты относительно всего экрана.

Для более плавного движения курсора был применен расчет координат по формуле 1:

$$X_q = X_{\text{пред}} + \left(X_{\text{тек}} - X_{\text{пред}}\right) / k\,, \quad (1)$$

где Xq – полученная координата;

$X_{\text{пред}}$ – предыдущая координата;

$X_{\text{тек}}$– текущая координата;

k – коэффициент сглаживания.

Данный подход применяется также и для координаты Y. Коэффициент сглаживания подбирался экспериментально, при k = 7 движение курсора стало наиболее плавным.

Скроллинг. Одним из путей осуществления бесконтактного скроллинга является использование метода scroll() библиотеки PyAutoGUI [6]. Однако для реализации такого подхода пришлось бы отвести под скроллинг целых два отдельных жеста. Вместо этого предлагается использовать бесконтактный клик по колесику мыши, в результате которого мышь переходит в режим скроллинга. Для этого режима используется три поднятых пальца (рис. 4). Данный режим очень удобен для прокрутки на веб-страницах или в текстовых редакторах.

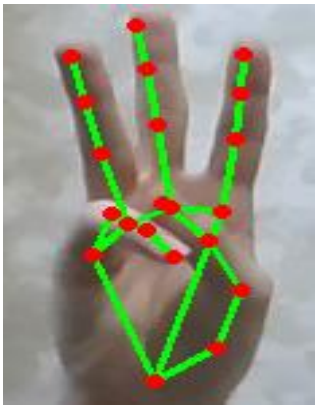

Рис. 4. Пример жеста прокрутки страницы (составлено автором)

Голосовой интерфейс. Голосовой интерфейс предназначен для перехода программы в режим поиска информации в сети Интернет либо в режим управления уровнем звука. Для активации голосового интерфейса используется жест с поднятыми указательным, большим пальцами и мизинцем (рис. 5). При распознавании голоса используется библиотека speech recognition [7].

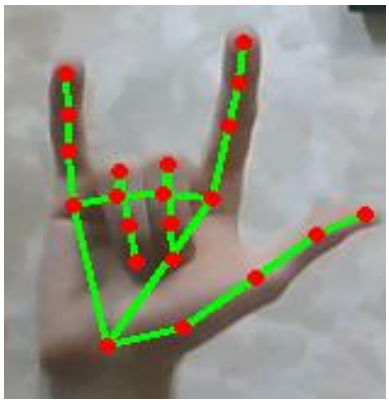

Рис. 5. Пример жеста для перехода в голосовой интерфейс (составлено автором)

Для удобства пользования голосовым интерфейсом предусмотрен синтез речи на основе библиотеки pyttsx3 [8]. Программа произносит одну из заготовленных фраз в тот момент, когда микрофон компьютера становится активным, а также когда будет закончено прослушивание голосовой команды. Если в голосовой команде присутствует слово «найди», то происходит поиск информации в Интернете, если присутствует слово «звук», то программа на пять секунд переходит в режим бесконтактной регулировки звука. По истечении пяти секунд программа возвращается в режим бесконтактной мыши.

Управление уровнем звука. В режиме управления громкостью динамиков используется информация о расстоянии между указательным и большим пальцем. Чем оно больше, тем больше громкость динамиков. Для расчета расстояния между пальцами используется метод hypot() из библиотеки math. Получив информацию о расстоянии, необходимо интерпретировать ее для

задания уровня громкости в операционной системе, для этого используется библиотека pycaw [9]. Пример работы программы в режиме управления звуком представлен на рис. 6.

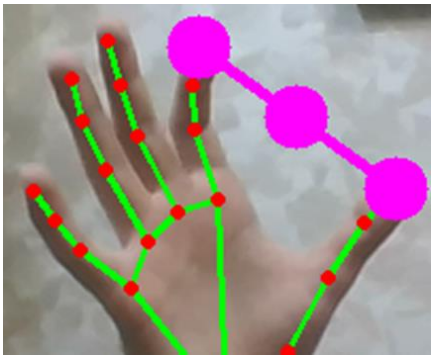

Рис. 6. Пример жеста для управления звуком (составлено автором)

Структура проекта. Основные файлы проекта (рис. 7):

- main – содержит основной цикл программы;
- hand_tracking_service – класс, содержащий основные методы для работы с mediapipe;
- speech_reconition_service – класс, содержащий методы для распознавания речи;
- synthesis_service – класс, содержащий методы для синтеза речи.

В папке resources содержатся два файла:

- configuration – с настройками подключения к голосовому движку windows для синтеза речи;
- my_enum – с основными ответными фразами для синтеза речи.

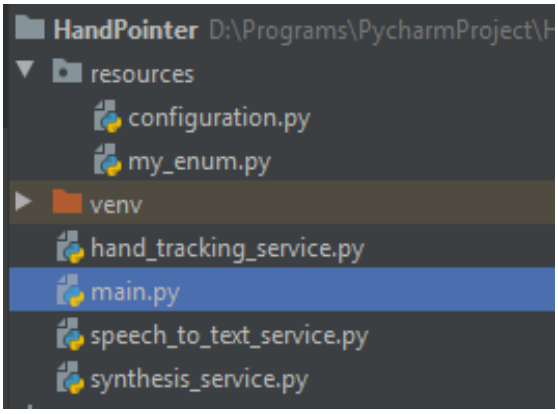

Рис. 7. Структура проекта (составлено автором)

Заключение. Разработанная методика построения человеко-машинного интерфейса для управления компьютером позволяет создать интерфейс, заменяющий функции компьютерной мыши. При этом сформулированные задачи предлагается решать с помощью библиотеки mediapipe. Интерфейс содержит функцию управления уровнем звука, а также голосовой интерфейс для поиска информации в сети Интернет и перехода в режим настройки звука. Следует заметить, что данный подход пригоден только при условии хорошей освещенности комнаты, что весьма ограничивает возможности проекта. Тем не менее, библиотека mediapipe позволяет с высокой точностью обнаруживать жесты руки человека, что открывает большие возможности для реализации различных интерактивных проектов на ее основе. Среди вариантов дальнейшего развития проекта можно выделить следующие функции для бесконтактного управления: перенос файлов между папками, открытие определенных приложений, выключение компьютера, поиск информации на жестком диске.

Библиографический список

1. Hand gesture recognition on python and opencv / A.P. Ismail, Farah Aziz, Nazirah Kasim, Kamarulazhar Daud // IOP Conference Series: Materials Science and Engineering. —2021— 1045. — 012043.10.1088/1757-899X/1045/1/012043.

2. Live ML anywhere / Mediapipe : [сайт]. — URL: https://mediapipe.dev/ (дата обращения: 20.12.2022).

3. Creating a Hand Tracking Module using Python, OpenCv, and MediaPipe / section : [сайт]. — URL: https://www.section.io/engineering-education/creating-a-hand-tracking-module/ (дата обращения: 20.12.2022).

4. MediaPipe Hands / google.github.io : [сайт]. — URL: https://google.github.io/mediapipe/ (дата обращения: 20.12.2022).

5.AutoPy Introduction and Tutorial. / Autopsy : [сайт]. — URL: https://pypi.org/project/autopy/ (дата обращения: 20.12.2022).

6. Mousecontrolfunctions / PyAutoGUI : [сайт]. — URL: https://pyautogui.readthedocs.io/en/latest/mouse.html (дата обращения: 20.12.2022).

7. SpeechRecognition / pypi.org : [сайт]. — URL: https://pypi.org/project/SpeechRecognition/ (дата обращения: 20.12.2022).

8. pyttsx3 / pypi.org : [сайт]. — URL:https://pypi.org/project/pyttsx3/ (дата обращения: 20.12.2022).

9. PyCaw / pypi.org : [сайт]. — URL: https://pypi.org/project/pycaw/ (дата обращения: 20.12.2022).

Об авторе:

Колегаев Борис Ярославич, аспирант кафедры «Вычислительные системы и информационная безопасность» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), kolegaevboris@gmail.com

About the Author:

Kolegaev, Boris Ya., postgraduate student of the Computing Systems and Information Security Department, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), kolegaevboris@gmail.com