

где  $\tau_{\text{н}}$ ,  $\tau_{\text{п}}$  – постоянные времени заряда и разряда конденсатора ФНЧ;  $t_{\text{н}}$ ,  $t_{\text{п}}$  – длительности импульса и паузы,  $t_{\text{н}} = T - t_{\text{п}}$ ;  $\sigma$  – приведенная погрешность осреднения,  $\sigma = \frac{\Delta U}{E}$ , где  $\Delta U$  – абсолютная погрешность осреднения.

Следует отметить, что  $\sigma$  принимает максимальное значение при условии  $\tau_{\text{н}}, \tau_{\text{п}} > T$ ,  $t_{\text{н}} > \frac{T}{2}$  и составляет  $\sigma = 0,5\delta$ , где  $\delta$  – относительный разброс постоянных времени  $\tau_{\text{н}}$  и  $\tau_{\text{п}}$ .

Относительный разброс постоянных времени  $\tau_{\text{н}}$  и  $\tau_{\text{п}}$   $\delta$  может быть уменьшен до приемлемых уровней путем увеличения значений или  $R$  и/или  $C$  ФНЧ, однако это ведет к снижению быстродействия ФНЧ, т. е. к уменьшению быстродействия операции осреднения, что не всегда является допустимым.

Полученные количественные соотношения позволяют находить компромисс между погрешностью и быстродействием операции осреднения при заданных неравенствах постоянных времени  $\tau_{\text{н}}$  и  $\tau_{\text{п}}$ , т. е. при заданных разбросах в сопротивлении открытого состояния ключей верхнего и нижнего уровней источника ШИМ-последовательностей напряжений.

#### Литература

1. Карпов, В. А. Анализ инструментальной погрешности двухосевого электролитического инклинометра. / В. А. Карпов, О. М. Ростокина, А. В. Карпов // Вестн. Гомел. гос. ун-та им. П. О. Сухого. – 2017. – № 1. – С. 77–80.
2. Анализ методической погрешности двухосевого электролитического инклинометра / В. А. Карпов [и др.] // Метрология и приборостроение. – 2018. – № 2. – С. 35–38.

УДК 621.38

### КОНТРОЛЛЕР МЫШИ С УПРАВЛЕНИЕМ ЖЕСТАМИ

Д. А. Литвинов, А. В. Ковалев, А. В. Лашкевич

*Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», Республика Беларусь*

*Организация бесконтактного интерфейса человека с компьютером с помощью жестов.*

**Ключевые слова:** человеко-компьютерное взаимодействие, управление жестами, контроллер мыши.

### MOUSE CONTROLLER WITH MOTION CONTROL

D. A. Litvinov, A. V. Kovalev, A. V. Lashkevich

*Sukhoi State Technical University of Gomel, the Republic of Belarus*

*Organization of contactless human-computer interface using motion.*

**Keywords:** human-computer interaction, motion control, mouse controller.

Будущее информационно-коммуникационных технологий, как правило, связывают с ростом производительности вычислительных систем, совершенствованием каналов связи, а также с расширением функциональности приложений и операционных систем. Однако углубленный анализ эффектов технологического развития дает основание полагать, что ключевую роль в формирующейся экосистеме тотального

пользования компьютерами будут играть в первую очередь новые средства взаимодействия человека и компьютера [1].

Человеко-компьютерное взаимодействие (*human-computer interaction, HCI*) – это область компьютерных наук, дисциплина, занимающаяся проектированием и оценкой интерактивных вычислительных систем для использования человеком, а также изучением происходящих процессов. Основной задачей человеко-компьютерного взаимодействия является улучшение взаимодействия между человеком и компьютером, делая компьютеры более удобными и восприимчивыми к потребностям пользователей. В качестве основных факторов, влияющих на разработку человеко-компьютерного взаимодействия, выступают специфика деятельности пользователя и организация «дружественного» интерфейса.

Исследования в области человеко-компьютерного взаимодействия опираются на все более полную и сложную модель человека (оператор – пользователь – личность с уникальным опытом), а поэтому методы юзабилити-исследований будут совершенствоваться и представлять актуальный предмет исследований [2].

Человеко-компьютерное взаимодействие расширилось от настольных офисных приложений до включения в себя игр, обучения, образования, торговли, здравоохранения и медицинских приложений и прочее. Все системы для организации *HCI* можно разделить на контактные и бесконтактные. Во первых, управление компьютером осуществляется непосредственным указанием своих действий, с использованием дополнительных устройств – мышь, тачпад, тачскрин и др. Во-вторых, в качестве *HCI* используется бесконтактное управление с помощью голоса, жестов рук или других частей тела человека.

Распознавание жестов тесно связано с понятием отслеживания. Обычно отслеживание рассматривается как процесс поиска временных соответствий между кадрами. Применительно к отслеживанию жестов, алгоритм распознавания связывает идентифицированный жест в предыдущих кадрах с текущим фреймом (рис. 1). Что касается статических жестов, которые могут быть представлены одним кадром, их отслеживание не требуется.

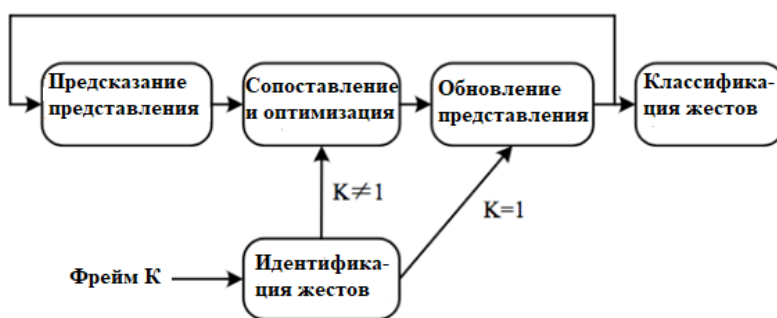


Рис. 1. Функциональная схема алгоритма отслеживания жестов

Классификация подходов к отслеживанию жестов:

- единичная гипотеза – быстрый и простой алгоритм. Подходит для отслеживания одного жеста в контролируемой среде;
- множественные гипотезы – возможность отслеживания нескольких целей одновременно. Подходит для отслеживания нескольких жестов в контролируемой среде;

– расширенная модель трекинга – целевая история сохраняется и доступна для целевой оценки. Подходит для долговременного отслеживания жестов;

– отслеживание путем обнаружения – алгоритм обучения повышает надежность и снижает уровень шума. Этот комбинированный подход имеет предпочтительную производительность в тестах. Подходит для отслеживания жестов в сложной среде.

Классификация жестов – это последний и самый важный шаг в распознавании жестов. Большинство человеческих жестов – это динамические жесты. Один динамический жест всегда состоит из нескольких кадров. Чтобы классифицировать динамические жесты, классификация жестов должна выполняться после или вместе с их отслеживанием.

В статье представлена разработка контроллера мыши с управлением жестами. Для организации взаимодействия человека с компьютером в устройстве предлагается использовать камеру для отслеживания жестов рук и действий пальцами. Основной задачей предлагаемой системы является выполнение функций указателя компьютерной мыши и функции тачпада.

Электронный модуль выполнен на основе одноплатного миникомпьютера *Raspberry Pi Zero W* (размер 67 x 30 мм). Модуль построен на основе *ARM* процессор *Broadcom BCM2835* с ядром *ARM1176JZ* и тактовой частотой 1 ГГц. *BCM2835* включает в себя графический сопроцессор *Broadcom VideoCore IV* с поддержкой *Open GL*. Миникомпьютер имеет 512 МВ оперативной памяти, и поддерживает следующие интерфейсы: *Wi-Fi 802.11 b/g/n*, *Bluetooth 4.1* и *Bluetooth (BLE)*, а также *CSI*-интерфейс для подключения камеры.

Для слежения за жестами в устройстве используется цветная цифровая камера *Omnivision 5647*, с *CSI*-интерфейсом. Она позволяет захватывать изображения с разрешением 5 Мп и записывать видео *HD*-качества 1080 p со скоростью 30 кадров в секунду.

Алгоритм работы программы основан на слежении за ключевыми ориентирами ладони. Для решения данной задачи используется метод создания ориентиров, называемых *landmarks* (рис. 2), с их последующим анализом. Система на кадрах изображений, поступающих от видеокamеры, распознает ладонь человека, фиксируют ключевые точки, их координаты и отслеживает перемещение ориентиров, интерпретируя жесты.

Для его программной реализации проекта используется библиотека алгоритмов обработки изображений и компьютерного зрения – *OpenCV* и фреймворк *MediaPipe*. *OpenCV* – это библиотека программного обеспечения для компьютерного зрения и машинного обучения с открытым исходным кодом. *OpenCV* был создан, чтобы обеспечить общую инфраструктуру для приложений компьютерного зрения и ускорить использование машинного восприятия в коммерческих продуктах.

*MediaPipe* – это кроссплатформенная платформа машинного обучения с открытым исходным кодом, используемая для построения систем машинного обучения. Одним из основных применений фреймворка является обнаружение лица и рук и извлечение ключевых точек для передачи в модель компьютерного зрения.

Проект реализован на языке программирования *Python*, с использованием библиотеки *Pynput*, которая реализует *API* для управления мышью.

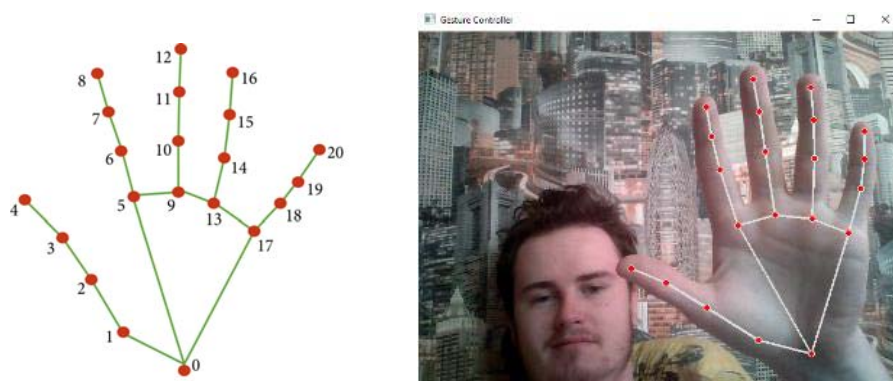


Рис. 2. Пример отображения ориентиров ладони

На рис. 3 представлен результат работы контроллера мыши. На картинках отображены жесты с нанесенными ориентирами для действий, интерпретируемых как щелчок левой (картинка слева) и правой кнопками мыши.

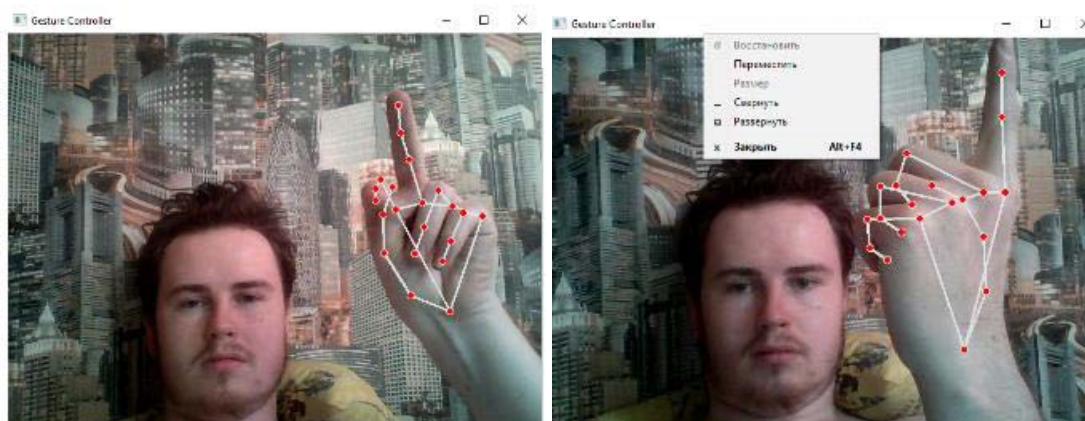


Рис. 3. Жесты, интерпретируемые как щелчок левой (слева) и правой кнопками мыши

Контроллер позволяет управлять курсором мыши с помощью жестов и выполняются следующие команды: перемещение курсора по экрану компьютера, щелчок правой и левой кнопкой мыши, скроллинг вниз–вверх, захват и перемещение объектов. При необходимости устройство может быть дополнено другими жестами.

Разработанные алгоритмы управления на основе жестов могут быть использованы как в отдельном устройстве, подключаемом к компьютеру по *Wi-Fi* или *Bluetooth* интерфейсу, так и на персональном компьютере с камерой или ноутбуке.

#### Литература

1. Ахметов, К. Взаимодействие человека и компьютера: тенденции, исследования, будущее / К. Ахметов // Форсайт. – 2013. – Т. 7, № 2. – С. 58–68.
2. Компаниец, В. С. Проектирование и юзабилити-исследование пользовательских интерфейсов : учеб. пособие / В. С. Компаниец, А. Е. Лызь. – Ростов н/Д. : ЮФУ. – 2020. – 107 с.
3. Кэлер, А. Изучаем OpenCV 3 / А. Кэлер, Г. Брэдски. – М. : ДМК-Пресс, 2017. – 826 с.