

БЕСКОНТАКТНЫЙ ИНТЕРФЕЙС ВЗАИМОДЕЙСТВИЯ С ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКОЙ

Белоброцкий Д. В.

*Белорусский государственный университет, Минск, Беларусь,
e-mail: denis.belobrotski@gmail.com*

Для большинства из нас зрение является основным источником данных об окружающей среде. Поэтому информация о том, на чем сфокусирован взгляд, может быть полезна. Такая информация может быть использована для работы интерфейса взаимодействия с вычислительными устройствами, такими как компьютеры, смартфоны, планшеты и др.

История отслеживания глаз (айтрекинга) и взгляда восходит к концу XIX века. Развитие фотографии и видеозаписи позволило открыть более надежные и неинвазивные методы наблюдения за движением глаз. Такие исследования стали популярными, особенно в психологии и медицинских исследованиях, таких как диагностика различных болезней. В настоящее время техника айтрекинга развивается в двух направлениях: электроокулография и анализ цифровых изображений [1 – 3].

К сожалению, коммерчески доступные приложения для трекинга требуют наличия специального оборудования, нередко дорогостоящего, а программное обеспечение закрыто для модернизации и адаптации под собственные цели, так как является собственностью коммерческих организаций. Таким образом, для самостоятельной разработки интерфейса взаимодействия взглядом необходима библиотека, с помощью которой можно разрабатывать собственное ПО, нуждающееся в такой библиотеке. Разработка такой библиотеки и алгоритмов для неё и является целью данной работы.

Целевой платформой для библиотеки является ОС Android, так как количество мобильных устройств на ней превышает 70% [4] и эти мобильные устройства обладают необходимым набором аппаратных средств (экран, камера) для реализации бесконтактных интерфейсов, использующих обработку изображения.

Основной алгоритм библиотеки, который позволяет определить положение центров глаз и центров зрачков, состоит из следующих шагов [5 – 8]:

1. Выделение глаз на изображении:
 - a. Преобразование цветного изображения в монохромное (grayscale);
 - b. Выравнивание гистограммы изображения;
 - c. Выделение лиц и соответствующих им глаз с помощью метода Виолы-Джонса и признаков распознавания образов Хаара;
2. Предварительная обработка области каждого глаза:
 - a. Выделение ранее полученных областей глаз на цветном изображении;
 - b. Отсечение области брови (25-40% сверху);
 - c. Переход от цветовой модели RGB к HSV;
3. Уточнение центра глаза по содержимому канала Saturation:
 - a. Применение операции inverted binary thresholding;
 - b. Применение операции erosion;
 - c. Применение операции dilation;
 - d. Вычисление центра масс на изображении (белый пиксель – самый «тяжелый», черный пиксель – самый «лёгкий»);

4. Вычисление центра зрачка по содержимому канала Value:
 - a. Выравнивание гистограммы изображения;
 - b. Применение операции inverted binary thresholding;
 - c. Применение операции erosion;
 - d. Применение операции dilation;
 - e. Вычисление центра масс на изображении;
5. Постобработка глаз:
 - a. Обратный переход текстурных координат для их корректировки после отсекания области брови.

Реализация описанного алгоритма с помощью OpenCV 4.4.0 на C++ для ОС Windows при тестировании на нескольких фотографиях показала результаты, представленные на рисунке 1.

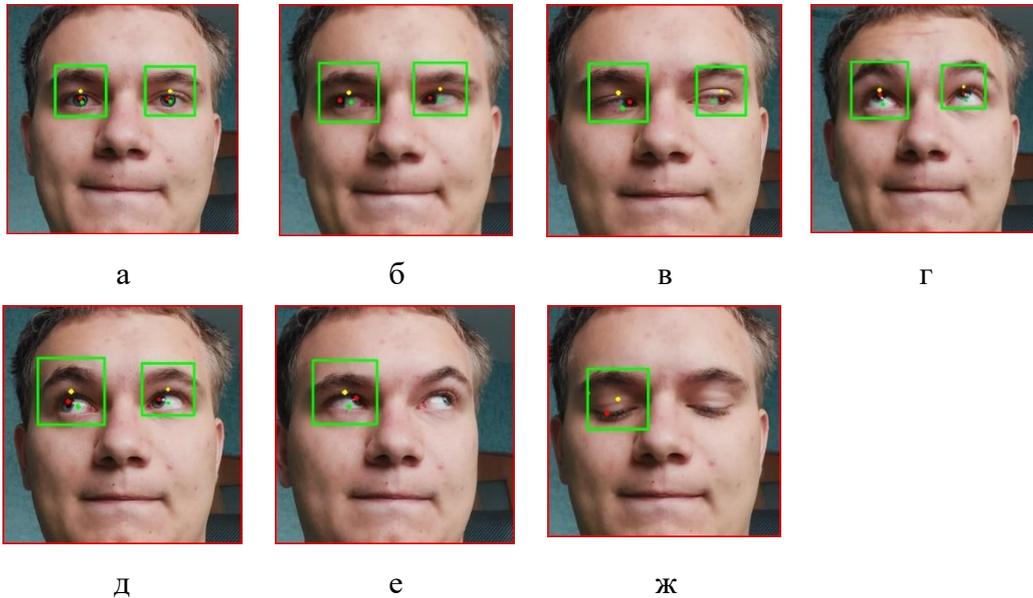


Рис. 1. Результаты работы алгоритма. Желтый маркер – центр области с глазом, зеленый маркер – центр склеры, красный маркер – центр зрачка

Для реализации описанных выше алгоритмов на ОС Android используется OpenCV Android SDK v4.4.0. Реализация для ОС Android показала аналогичные результаты при использовании библиотеки в тестовом приложении. В процессе реализации алгоритмов также были написаны различные утилиты, которые облегчили разработку самой библиотеки. Также, использование этих утилит может упростить использование библиотеки.

Для взаимодействия пользователя с устройством с помощью глаз реализован алгоритм, позволяющий вычислить направление взгляда. Этот алгоритм использует тот факт, что скалярное произведение сонаправленных векторов равно произведению их длин. Так, вычислив разницу между радиус-вектором центра зрачка и радиус-вектором центра глаза и перемножив полученный вектор скалярно с каждым из заданных векторов направлений, можно получить с каким из этих векторов вероятнее всего сонаправлен взгляд.

Так как одним из возможных сценариев использования библиотеки является обработка изображения в реальном времени, то потребовались оптимизации, такие как:

1. Уменьшение разрешения обрабатываемого изображения;
2. Сокращение количества выделений памяти, переиспользование уже выделенной памяти;
3. Ручная очистка выделяемой под изображения памяти;
4. Обработка только одного лица на изображении.

Система реального времени должна реагировать на изменения во внешней системе в рамках требуемых временных ограничений. Минимальная частота обновления экрана на мобильных устройствах, широко представленных на рынке, является 60 Гц, чему соответствует отображение 60 кадров в секунду (fps). То есть время показа одного кадра составляет около 16 – 17 мс. 24 кадра в секунду является принятой нормой кинематографического изображения (41 – 41 мс на показ кадра).

После всех описанных выше оптимизаций время работы алгоритма на изображениях, представленных на рисунке 1, соответствует временным ограничениям для системы реального времени, что можно увидеть в таблице 1.

Табл. 1. Время обработки изображений

Изображения	а	б	в	г	д	е	ж
Время обработки	18 мс	16 мс	15 мс	15 мс	17 мс	13 мс	13 мс

Полученный в ходе работы алгоритм позволяет выделить на изображении (с лицом) направление взгляда, однако, исходя из полученных результатов, наиболее успешно алгоритм выделяет направления влево-центр-вправо, что нельзя сказать про направления вниз-центр-вверх. Проблема со взглядом вверх связана со смещением области глаза вверх, что смещает центр глаза вверх вместе с положением зрачка, а это в свою очередь влечет неотличимость взгляда вверх и взгляда в центр. Также, проблемой является распознавание взгляда вниз, так как при взгляде вниз глаз почти полностью не отличим от закрытого глаза, по крайней мере для разработанного алгоритма.

Литература

1. Ярбус, А.Л. Роль движений глаз в процессе зрения / А.Л. Ярбус. – АН СССР. – М.: Наука, 1965.
2. Демидов В.Е. Как мы видим то, что видим / В.Е. Демидов. – Наука и прогресс. – М.: Знание, 1979.
3. Koziol P., Eye Pupil Location Using Webcam / M. Ciesla, P. Koziol. – ResearchGate. – Krakow, 2012.
4. Mobile Operating System Market Share Worldwide [Electronic resource] / StatCounter GlobalStats. – StatCounter, 2020. – Mode of access: <https://gs.statcounter.com/os-market-share/mobile/worldwide>. – Date of access: 23.09.2020.
5. Types of Morphological Operations [Electronic resource] / Matlab Help. – The MathWorks, Inc. – Mode of access: <https://www.mathworks.com/help/images/morphological-dilation-and-erosion.html>. – Date of access: 13.11.2020.
6. Trott, M. Profiling the Eyes: faithful or ROTen? Or Both? [Electronic resource] / M. Trott. – Wolfram Research Blog, 2016. – Mode of access: <https://blog.wolfram.com/2016/03/02/profiling-the-eyes-phiathful-or-roten-or-both/>. – Date of access: 24.02.21.
7. Center of Mass [Electronic resource] / HyperPhysics. – Department of Physics and Astronomy, Georgia State University, 2016. – Mode of access: <http://hyperphysics.phy-astr.gsu.edu/hbase/cm.html>. – Date of access: 15.11.2020.
8. OpenCV documentation [Electronic resource] / Open Source Computer Vision Library. – OpenCV team, 2020. – Mode of access: <https://docs.opencv.org/>. – Date of access: 12.03.2020.