

Информатика

УДК 004.932

EDN: KIJYJR

Эффективность расчёта и трудности при расчете расстояния до обозреваемого с использованием системы с одной камерой без калибровки

А.В. Воружев, К.С. Голубич, Д.С. Сыч, Е.В. Рафалова

Работа исследует эффективность и сложности расчёта расстояния до объектов с использованием системы на основе одной камеры без калибровки. Рассмотрены преимущества таких систем. Предложен алгоритм расчёта расстояния на основе параметров камеры и характеристик объекта, реализованный на Python с использованием метода каскадных признаков Хаара. Тестирование показало погрешность в пределах 3–8 %, что делает систему применимой в областях, где высокая точность не критична, таких как безопасность, видеонаблюдение, медицина и автомобильные технологии.

Ключевые слова: компьютерное зрение, однокамерная система, расчёт расстояния, каскадные признаки Хаара.

The study examines the effectiveness and challenges of calculating the distance to objects using a single-camera system without calibration. The advantages of such systems, including their cost-effectiveness and simplicity, are discussed, and an algorithm for distance calculation based on camera parameters and object characteristics is proposed, implemented in Python using the Haar cascade method. Testing revealed an error margin of 3–8 %, making the system applicable in areas where high precision is not critical, such as security, video surveillance, medicine, and automotive technologies.

Keywords: computer vision, single-camera system, distance calculation, Haar cascade features.

Введение. В последние десятилетия технологии компьютерного зрения шагнули далеко вперёд, современные системы распознавания способны точно распознавать объекты и расстояние до них с использованием различного рода датчиков, сенсоров и при использовании как минимум пары камер. При этом системы, основанные на использовании одной камеры, стали популярными благодаря их относительной дешевизне, компактности и широким возможностям применения. Однако отсутствие калибровки камеры и ограниченность данных, получаемых из одного источника, создают существенные проблемы в точности и надёжности таких систем.

Основной проблемой при расчёте расстояния с использованием системы с одной камерой является многозначность изображения, возникающая из-за плоской природы двумерных изображений [1]. Камера, получая изображение, не в состоянии напрямую измерить расстояние до объекта, поскольку на изображении отсутствует информация о глубине.

Одним из преимуществ использования системы с одной камерой является её относительная простота и дешевизна. Камера, в отличие от более сложных систем, таких как лидары или стерео-камеры, требует гораздо меньше ресурсов, как в плане оборудования, так и в плане вычислительных мощностей. Система с одной камерой без калибровки не требует точных настроек, что может существенно уменьшить сложность разработки и настройки системы. В приложениях, где требуется компактность и малый вес, такие решения являются предпочтительными.

Однако в сравнении с другими методами, такими как стерео-визия или использование лидаров, системы с одной камерой имеют явные ограничения. Стереовизия использует пару камер для создания глубинных карт, что позволяет более точно определять расстояние за счёт анализа различий в изображениях с разных точек обзора. Этот метод значительно повышает точность расчёта расстояния, но требует более сложной настройки и синхронизации камер, а также дополнительной вычислительной мощности для обработки данных.

Системы, использующие лидары, обеспечивают более точные измерения расстояний за счёт активного излучения лазера и измерения времени его возврата. Однако такие системы более дороги и сложны в реализации, а также требуют более сложных алгоритмов для обра-

ботки данных. В то время как система с одной камерой не может достичь той же точности, что и лидар или стерео-система, её простота и дешевизна делают её подходящей для менее критичных приложений [2].

Пример расчета. В связи с причинами, указанными выше, необходимо обозначить параметры, которые имеют вид констант или зависят от характеристик каждой камеры или от применяемого алгоритма распознавания лица на кадре [3]. Данные параметры: α – угол обзора камеры, l – константа равная средней ширине лица человека, k – коэффициент размера лица, необходим для настройки параметра размера лица, при сглаживании неточности нахождения лица на фото каскадными признаками Хаара, H – ширина разрешения камеры.

Далее представлены рассчитываемые параметры: D – угловой диаметр, h_{Π} – ширина лица в пикселях, L – искомое расстояние.

Изначальная формула расчета расстояния имеет вид:

$$L = \frac{D}{2 * tg(\frac{\alpha}{2})} \quad (1)$$

Однако необходимо учесть, что α – базовая характеристика камеры и необходимо рассчитать D :

$$D = \frac{l * H}{h_{\Pi} * k} \quad (2)$$

Итоговая формула имеет вид:

$$L = \frac{l * H}{2 * tg(\frac{\alpha}{2}) * h_{\Pi} * k} \quad (3)$$

Ниже на картинке 1 представлена визуализация основных параметров.

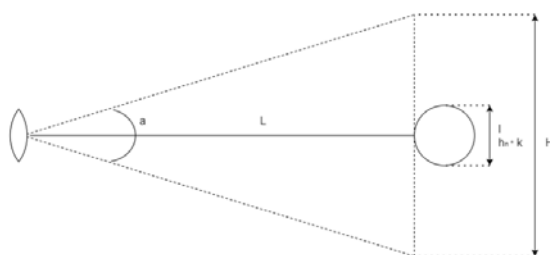


Рисунок 1 – Визуализация основных параметров

Алгоритм работы и реализация. Для реализации данного алгоритма был использован язык Python, с помощью которого был написан программный комплекс, который подразумевает использование вышеописанного алгоритма для расчета расстояния до пользователя. Основные этапы работы программного комплекса включают в себя: сбор данных об используемом оборудовании, подключение к камере и инициализация оконного интерфейса, получение кадра с камеры, распознавание лица и расчет основных параметров лица, расчет расстояния до пользователя и вывод данных на экран. Этапы и пример работы представлены на рисунке 2 и рисунке 3.

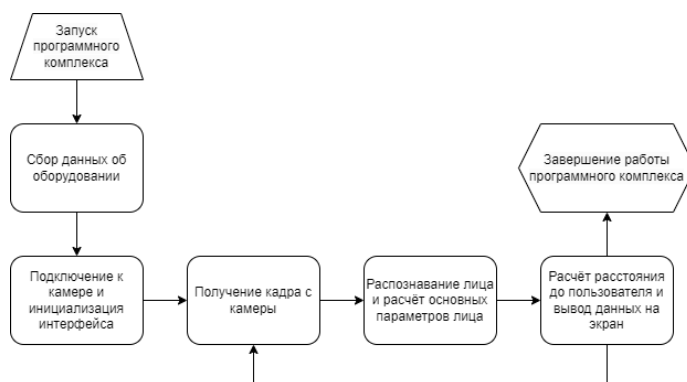


Рисунок 2 – Этапы работы программного комплекса

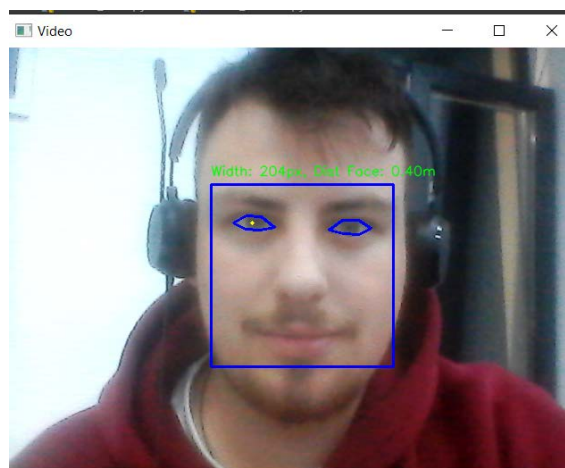


Рисунок 3 – Пример работы программного комплекса

Тестирование точности работы алгоритма. Составим таблицу для расчета точности расстояния до обозреваемого, а также график точности для измерения погрешности алгоритма от расстояния до пользователя (рисунок 4).

Таблица 1 – Расчет точности измерений

Реальное расстояние, м	Измеренное расстояние, м	Погрешность, %
0,25	0,24	4.00
0,50	0,52	3.90
0,75	0,71	5.43
1,00	0,97	3
1,25	1.15	8.00

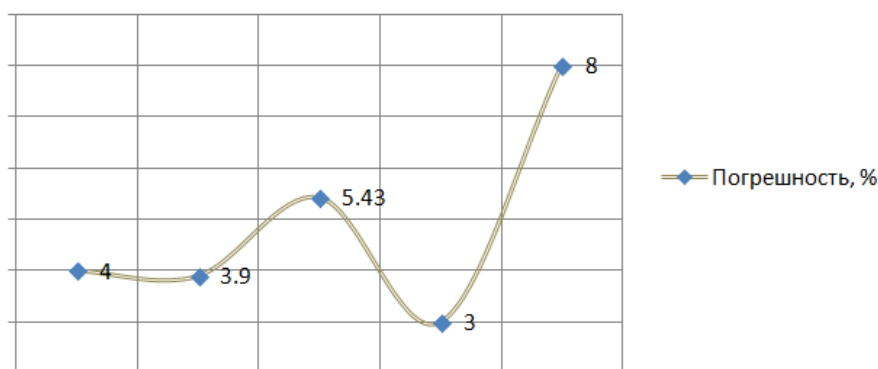


Рисунок 4 – График точности системы от расстояния

Сферы применения. Данный алгоритм может использоваться в системах безопасности и видеонаблюдения для определения расстояния до подозрительных объектов или лиц, а также для идентификации людей на расстоянии. В сфере человеко-компьютерного взаимодействия такая система может улучшить пользовательский опыт в видеозвонках и онлайн-играх, адаптируя интерфейс в зависимости от расстояния до пользователя.

В медицине система может применяться для мониторинга состояния пациентов, например, для оценки дыхания или других физиологических параметров на расстоянии. В образовании и дистанционном обучении она может использоваться для оценки вовлеченности студентов, определяя, насколько близко они находятся к экрану.

В розничной торговле и маркетинге система может анализировать поведение покупателей в магазинах, определяя, как близко они подходят к продуктам, и предлагать персонализированные предложения и рекламу. В автомобильных технологиях система может использоваться в системах помощи водителю для определения расстояния до пассажиров или объектов в автомобиле.

Работа программы. Так же стоит поговорить немного про работу программного комплекса. Основная информация выводится на оконный интерфейс программы, однако при работе программы все данные выводятся в файл логирования, пример которого представлен на рисунке 5.

```
INFO: __main__:Start program!!!
Time:2025-01-08 14:04:56.992368;
Frame width: 640.0;
Camera angle: 60;
Mean face width: 0.16;
Cascade coef: 1.1;

INFO: __main__:Timestamp: 2025-01-08 14:04:58.172218; Face width (px): 132; Distantion: 0.6107506979857199;
INFO: __main__:Timestamp: 2025-01-08 14:04:58.292221; Face width (px): 121; Distantion: 0.6662734887116943;
INFO: __main__:Timestamp: 2025-01-08 14:04:58.365220; Face width (px): 121; Distantion: 0.6662734887116943;
INFO: __main__:Timestamp: 2025-01-08 14:04:58.419219; Face width (px): 121; Distantion: 0.6662734887116943;
INFO: __main__:Timestamp: 2025-01-08 14:04:58.476214; Face width (px): 124; Distantion: 0.6501539688235083;
INFO: __main__:Timestamp: 2025-01-08 14:04:58.552223; Face width (px): 122; Distantion: 0.6608122306075002;
INFO: __main__:Timestamp: 2025-01-08 14:04:58.612218; Face width (px): 122; Distantion: 0.6608122306075002;
INFO: __main__:Timestamp: 2025-01-08 14:04:58.666216; Face width (px): 123; Distantion: 0.6554397734480896;
INFO: __main__:Timestamp: 2025-01-08 14:04:58.721217; Face width (px): 120; Distantion: 0.6718257677842918;
INFO: __main__:Timestamp: 2025-01-08 14:04:58.776213; Face width (px): 124; Distantion: 0.6501539688235083;
INFO: __main__:Timestamp: 2025-01-08 14:04:58.830220; Face width (px): 124; Distantion: 0.6501539688235083;
INFO: __main__:Timestamp: 2025-01-08 14:04:58.886219; Face width (px): 122; Distantion: 0.6608122306075002;
INFO: __main__:Timestamp: 2025-01-08 14:04:58.969218; Face width (px): 123; Distantion: 0.6554397734480896;
INFO: __main__:Timestamp: 2025-01-08 14:04:59.021216; Face width (px): 122; Distantion: 0.6608122306075002;
INFO: __main__:Timestamp: 2025-01-08 14:04:59.078216; Face width (px): 123; Distantion: 0.6554397734480896;
INFO: __main__:Timestamp: 2025-01-08 14:04:59.130221; Face width (px): 124; Distantion: 0.6501539688235083;
INFO: __main__:Timestamp: 2025-01-08 14:04:59.182217; Face width (px): 125; Distantion: 0.6449527370729202;
INFO: __main__:Timestamp: 2025-01-08 14:04:59.236214; Face width (px): 125; Distantion: 0.6449527370729202;
INFO: __main__:Timestamp: 2025-01-08 14:04:59.311218; Face width (px): 125; Distantion: 0.6449527370729202;
INFO: __main__:Timestamp: 2025-01-08 14:05:00.167221; Face width (px): 126; Distantion: 0.6398340645564684;
INFO: __main__:Stop program!!! Time:2025-01-08 14:01:00.525725;
```

Рисунок 5 – Файл логирования

Помимо выше перечисленного так же стоит внимания метод реализации данного алгоритма вычисления на языке программирования Python. Пример псевдокода:

```
# Читаем кадр из веб-камеры и отзеркаливаем, спроверкой на ошибку подключения
ret, frame = cap.read()
frame = flip(frame, 1)
if not ret:
    break
# Преобразуем кадр в градации серого и обнаруживаем лица
gray = cvtColor(frame, COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(gray)
# Если есть хотя бы одно лицо, то ищем расстояние до него
if len(faces) >= 1:
    # Получение расположения лица на кадре
    x, y, w, h = faces[0]
    # Рисуем рамку вокруг лица
    rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
    dist_to_face = (IMAGE_WIDTH * FACE_WIDTH) / ((2 * tan(CAMERA_ANGLE / 2 * pi
/ 180)) * (w * CASCADE_COEF))
    # Добавляем текст с шириной лица и дистанцией
    putText(frame, f'Width: {w}px, Dist to Face: {dist_to_face:.2f}m', (x, y -
10), FONT_HERSHEY_SIMPLEX, 0.5,
(0, 255, 0), 1, LINE_AA)
    # Вывод в лог
    logger.info(f"Timestamp: {datetime.now()}; Face width (px): {w}; Distantion:
{dist_to_face};")
```

Закключение. Вследствие особенностей работы с плоским изображением и необходимостью применять некоторые параметры константными данная система имеет значимую погрешность, что определяет сферы её использования вне пределов точных исследований. Однако простота использования, настройки, разработки, а так же минимальный необходимый перечень используемых технических средств позволяют использовать данную систему в качестве простого и дешёвого аналога дорогих и более масштабных систем.

Литература

1. Рудой, М. Т. Распознавание состояния оператора системы средствами дополненного интеллекта / М. Т. Рудой // Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях : материалы XXVI Республиканской научной конференции студентов и аспирантов, Гомель, 20–22 марта 2023 г. : в 2 ч. / М-во образования РБ, ГГУ им. Ф. Скорины ; редкол.: С. П. Жогаль [и др.]. – Гомель : ГГУ им. Ф. Скорины, 2023. – Ч. 2. – С. 170.
2. Sych, D. S. Forms of representation of 3D data for machine learning / D. S. Sych, N. A. Aksionova // Актуальные вопросы физики и техники : сб. матер. XII Республиканской научной конференции студентов, магистрантов и аспирантов, Гомель, 20 апреля 2023 г. / М-во образования Республики Беларусь, ГГУ им. Ф. Скорины ; редкол.: Д. Л. Коваленко (гл. ред.) [и др.]. – Гомель : ГГУ им. Ф. Скорины, 2023. – С. 523–524.
3. Demidenko, O. M. Identification of students' faces in a video stream using the Viola-Jones method / O. M. Demidenko, N. A. Aksionova, A. V. Varuyeu // 2022 International Conference on Information, Control, and Communication Technologies (ICCT). – Astrakhan, 2022. – P. 1–5.
4. Demidenko, O. M. Implementation of the identification and recognition system cognitive behavior of the observed / O. M. Demidenko, N. A. Aksionova, A. V. Varuyeu // Известия Саратовского университета. Новая серия. Сер., Математика. Механика. Информатика. – 2024. – № 24. – С. 275–296.

Гомельский государственный
университет имени Франциска Скорины

Поступила в редакцию 31.03.2025