Экспериментальная психология
2019. Т. 12. № 4. С. 5—11
doi: 10.17759/exppsy.2019120401
ISSN: 2072-7593
ISSN: 2311-7036 (online)


Experimental Psychology (Russia)
2019, vol. 12, no. 4, pp. 5—11
doi: 10.17759/exppsy.2019120401
ISSN: 2072-7593
ISSN: 2311-7036 (online)


# СЛЕЖЕНИЕ ЗА ДВИЖУЩИМИСЯ ОБЪЕКТАМИ НА ВИДЕОЗАПИСИ В ПСИХОЛОГИЧЕСКИХ ИССЛЕДОВАНИЯХ

***ЖЕГАЛЛО А.В.****, *Институт психологии РАН (ИПРАН);*
*Московский государственный психолого-педагогический университет (МГППУ),*
*Москва, Россия,*
*e-mail: zhegs@mail.ru*

В статье рассматриваются возможности использования библиотек OpenCV и dlib для отслеживания положения заданных объектов на видеоизображении и отслеживания набора основных анатомических точек лица человека. Показывается, что данные задачи могут быть эффективно решены с помощью компактных программ на языке Python. Язык Python рекомендуется вниманию психологов-исследователей как средство анализа видеоизображений и конструирования сложных экспериментов.

***Ключевые слова:*** object tracking, face detection, Python, OpenCV, dlib.

В нашей предыдущей работе (Жегалло, 2018) мы обсуждали возможности использования айтрекинга при решении задачи сохранения и передачи когнитивного опыта. Было показано, что регистрация движений глаз эксперта при просмотре видеозаписей, которые демонстрируют процесс выполнения изучаемой деятельности, позволяют выделить ключевые элементы и динамические паттерны деятельности, исходно «невидимые» для субъекта. Этот инструмент оказался существенным дополнением к процедуре кооперативного дебрифинга, предназначенной для выявления наиболее значимых составляющих когнитивного опыта (Носуленко, Самойленко, 2016). Видеозаписи, получаемые в результате такого анализа, предназначены для использования в интерактивной мультимедийной системе передачи опыта. И здесь возникает задача привлечения внимания человека, просматривающего видеозапись, к определенным объектам на движущемся изображении. Такое определение положения подвижного объекта на видеоизображении представляет собой актуальную задачу не только в задачах передачи когнитивного опыта, но и при оценке состояния собеседника по видеоизображению (Барабанщиков, Жегалло, Хозе, Соломонова, 2018). С целью отработки методов анализа нами была выполнена оценка функциональных возможностей библиотек OpenCV и dlib. Библиотека OpenCV (Open Source Computer Vision Library) разрабатывается с 2000 г. и на сегодняшний день поддерживает широкий спектр функциональности: чтение и запись статических изображений; анализ и обработку статических видеоизображений; чтение и запись видео; анализ видео; поддержку стереоскопических изображений; поиск заданных объектов на изображении и др (OpenCV Online Documentation, 2019).

**Для цитаты:**
*Жегалло А.В.* Слежение за движущимися объектами на видеозаписи в психологических исследованиях // Экспериментальная психология. 2019. Т. 12. №. 4. С. 5—11. doi:10.17759/exppsy.2019120401

* *Жегалло А.В.* Кандидат психологических наук, старший научный сотрудник, Институт психологии РАН (ИПРАН); старший научный сотрудник, Институт экспериментальной психологии, Московский государственный психолого-педагогический университет (МГППУ). E-mail: zhegs@mail.ru

Для решения задачи слежения за движущимся объектом оптимально подходит модуль OpenCV Tracking API (OpenCV Tracking API, 2019).

«Трекеры» в OpenCV специально предназначены для слежения за движущимися объектами на видеоизображении и при анализе следующего кадра видео учитывают расположение отслеживаемого объекта на предыдущем кадре. За счет этого достигается высокая производительность и устойчивость работы по сравнению с «детекторами», ориентированными на независимый поиск объекта на изображении по заданному образцу. Tracking API включает несколько реализаций «трекеров», различающихся используемыми алгоритмами слежения, но имеющих единый интерфейс использования.

Работа с «трекером» включает следующие шаги: создание объекта — «трекера»; инициализация — указание прямоугольной области на начальном изображении, содержащей отслеживаемый объект; поиск отслеживаемого объекта на очередном видеокадре. Поскольку мы выполняем поиск объекта на видеозаписи, нам также будут необходимы стандартные функции OpenCV, обеспечивающие открытие видео и чтение очередного кадра. Мы использовали доработанный пример реализации данной задачи на языке Python (Object Tracking using OpenCV, 2019).

Работа примера проверялась в ОС Windows 7, python 3.7.2, opencv_python 4.0..0.21, opencv_contrib_python 4.1.2.30; в ОС Lubuntu 18.04, python 3.7.4, opencv_python 4.1.2.30, opencv_contrib_python 4.1.2.30. Тестирование примера проводилось на примерах видеозаписей производственного процесса, полученных в ходе изучения способов передачи когнитивного опыта (6 отрывков видео продолжительностью от 10 до 30 секунд). Наилучшие результаты были получены при использовании трекера CSRT (Lukezic et al, 2018). Ниже приводится модифицированный исходный код программы, представленный на (Object Tracking using OpenCV, 2019).

```
#coding:utf-8
import cv2
import sys

if __name__ == '__main__' :

    # Инициализация трекера.
    tracker_types = ['BOOSTING', 'MIL','KCF', 'TLD', 'MEDIANFLOW', 'GOTURN', 'MOSSE', 'CSRT']
    tracker_type = tracker_types[7]

    if tracker_type == 'BOOSTING':
        tracker = cv2.TrackerBoosting_create()
    if tracker_type == 'MIL':
        tracker = cv2.TrackerMIL_create()
    if tracker_type == 'KCF':
        tracker = cv2.TrackerKCF_create()
    if tracker_type == 'TLD':
        tracker = cv2.TrackerTLD_create()
    if tracker_type == 'MEDIANFLOW':
        tracker = cv2.TrackerMedianFlow_create()
    if tracker_type == 'GOTURN':
        tracker = cv2.TrackerGOTURN_create()
    if tracker_type == 'MOSSE':
        tracker = cv2.TrackerMOSSE_create()
    if tracker_type == "CSRT":
        tracker = cv2.TrackerCSRT_create()
    # Открываем видеофайл
```

```
argc = len(sys.argv)
if argc > 1:
    video = cv2.VideoCapture(sys.argv[1])
else:
    print ("Video not specified")
    sys.exit()
#=========================
# Выход если видеофайл не был отрыт
if not video.isOpened():
    print ("Could not open video")
    sys.exit()

# Читаем первый кадр
ok, frame = video.read()
if not ok:
    print ("Cannot read video file")
    sys.exit()
# Интерактивный выбор начальной области интереса
bbox = cv2.selectROI(frame, False)
# Инициализация трекера
ok = tracker.init(frame, bbox)

while True:
    # Читаем очередной кадр видео
    ok, frame = video.read()
    if not ok:
        break
    # Начальное значение таймера
    timer = cv2.getTickCount()
    # Поиск заданного объекта на текущем кадре видео
    ok, bbox = tracker.update(frame)
    # Вычисляем частоту кадров в секунду (FPS)
    fps = cv2.getTickFrequency() / (cv2.getTickCount() - timer);
    # Рисуем положение отслеживаемого объекта на текущем кадре
    if ok:
        # Если объект обнаружен - рисуем область интереса
        p1 = (int(bbox[0]), int(bbox[1]))
        p2 = (int(bbox[0] + bbox[2]), int(bbox[1] + bbox[3]))
        cv2.rectangle(frame, p1, p2, (255,0,0), 2, 1)
    else :
        # Иначе - сообщение об ошибке
        cv2.putText(frame, "Tracking failure detected", (100,80), cv2.FONT_HERSHEY_
SIMPLEX, 0.75,(0,0,255),2)
    # Выводим тип трекера на текущем кадре
    cv2.putText(frame, tracker_type + " Tracker", (100,20), cv2.FONT_HERSHEY_
SIMPLEX, 0.75, (50,170,50),2);
    # Выводим FPS на текущем кадре
    cv2.putText(frame, "FPS : " + str(int(fps)), (100,50), cv2.FONT_HERSHEY_
SIMPLEX, 0.75, (50,170,50), 2);
    # Выводим текущий кадр с нанесенной информацией
    cv2.imshow("Tracking", frame)
    # Выход по нажатию ESC
    k = cv2.waitKey(1) & 0xff
    if k == 27 : break
```

На всех протестированных примерах ключевые движущиеся объекты (обрабатывающий инструмент в руках мастеров) успешно отслеживались от начала до конца записи. Единственное исключение составила видеозапись, содержащая монтажный переход. Корректное отслеживание движущегося объекта в этом случае прекратилось в момент смены сцены. Решение этой проблемы возможно путем введения ограничений на тип анализируемой видеозаписи или переходом к частично интерактивной разметке, при которой оператор вручную останавливает проигрывание видео при нарушении корректности отслеживания и заново вручную задает отслеживаемый объект.

При анализе видеоизображения лица человека представляет интерес задача выделения набора ключевых анатомических точек. Данный алгоритм эффективно реализован в библиотеке машинного зрения dlib (Dlib C++ Library, 2019). Пример исходного кода, решающего данную задачу (Jose, 2018), с незначительными изменениями приведен далее.

```
#coding:utf-8
# импорт пакетов
from imutils import face_utils
import dlib
import cv2
import sys


# инициализация детектора dlib's (HOG-based)
# создание facial landmark predictor
p = "shape_predictor_68_face_landmarks.dat"
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(p)

# Открываем видеофайл
argc = len(sys.argv)
if argc > 1:
    video = cv2.VideoCapture(sys.argv[1])
else:
    print ("Video not specified")
    sys.exit()

# Выход если видеофайл не был отрыт
if not video.isOpened():
    print ("Could not open video")
    sys.exit()

while(True):
    # захват очередного кадра видео
    ret, frame = video.read()
    # преобразование в оттенки серого
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    # детектирование лиц
    rects = detector(gray, 0)
    # цикл по обнаруженным лицам
    for (i, rect) in enumerate(rects):
        # определение опорных точек лица для заданной области, содержащей лицо
        # преобразование координат (x, y) опорных точек
        # в массив NumPy
```

```
        shape = predictor(gray, rect)
        shape = face_utils.shape_to_np(shape)
        # цикл по координатам (x, y) опорных точек
        # и отображение их на текущем кадре видео
        for (x, y) in shape:
            cv2.circle(frame, (x, y), 2, (0, 255, 0), -1)
    # отображение кадра видео с нанесенной разметкой
    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# завершение работы
video.release()
cv2.destroyAllWindows()
```

Работа примера проверялась в ОС Lubuntu 18.04, python 3.7.4, opencv_python 4.1.2.30, opencv_contrib_python 4.1.2.30, dlib .19.18.0, imutils 0.5.4. Запустить пример в Windows 7 не удалось, так как установка dlib требовала наличия компилятора C++ 11.0. Для этого предлагалось установить Visual Studio 15.0 или выше. На момент написания статьи на сайте Microsoft была доступна версия 19.0, однако после ее установки искомый компилятор обнаружен не был. Таким образом, работа с dlib под ОС Windows требует дополнительного уточнения процедуры установки библиотеки.

Пример проверялся на видеозаписях лиц, полученных в ходе изучения общения, опосредованного видеокоммуникацией. Записи проводились на доработанные WEB-камеры Sqny Playstation Eye с вариофокальным объективом 2,8—12 мм. Разрешение видео — 640×480 точек, частота — 60 к/сек, продолжительность записей — до 15 мин. Во всех случаях наблюдалась уверенная устойчивая детекция набора опорных точек изображения.

Полученные результаты показывают возможность эффективного использования полученных за последние годы Open Source решений для выполнения актуальных задач экспериментальной психологии. При этом язык Python выступает в роли эффективного средства интеграции имеющихся решений. Для сравнения можно отметить, что, несмотря на то, что библиотеки OpenCV и dlib имеют хорошо документированные интерфейсы на языке C++, самостоятельная сборка под Linux примера, включающего одновременно использование обеих библиотек, оказалась достаточно сложной задачей, которую автору не удалось решить за разумное время.

За последнее время значительно выросла популярность Python в качестве инструментария для проведения психологических экспериментов (Dalmaijer, 2016). Возможности Python в части анализа данных имеют ограниченный характер (Маккини, 2015). Тем не менее, по совокупности имеющихся возможностей следовало бы рекомендовать психологам-исследователям язык Python как средство решения экспериментальных исследовательских задач, в частности, в части анализа видезаписей и конструирования сложных экспериментов (Товуу и др., 2017). При разработке учебных программ для студентов соответствующих специальностей следует планировать введение развернутого учебного курса по языку Python и его приложениям для психологических исследований.

---

Исходный код программных решений, обсуждаемых в статье, доступен через профиль ResearchGate: https://www.researchgate.net/profile/Alexander_Zhegallo

*Финансирование*

Исследование выполнено в рамках государственного задания Министерства науки и высшего образования Российской Федерации, проект 25.3471.2017/ПЧ «Выявление значимых составляющих когнитивного опыта специалиста в задачах их сохранения и передачи».

***Литература***

1. *Барабанщиков В.А., Жегалло А.В., Хозе Е.Г., Соломонова А.В.* Невербальные предикторы оценок достоверности/недостоверности сообщаемой информации // Экспериментальная психология. 2018. Т. 11. № 4. С. 94—106. doi:10.17759/exppsy.2018110408
2. *Жегалло А.В.* Технологии айтрекинга в задачах сохранения и передачи когнитивного опыта // Экспериментальная психология. 2018. Т. 11. № 4. С. 135—141. doi:10.17759/exppsy.2018110412
3. *Маккини У.* Python и анализ данных / Пер. с англ. А.А. Слинкин. М.: ДМК Пресс, 2015.
4. *Носуленко В.Н., Самойленко Е.С.* Полипозиционное наблюдение // Технологии сохранения и воспроизведения когнитивного опыта / Под ред. В.Н. Носуленко. М.: Институт психологии РАН, 2016. С. 261—278.
5. *Товуу Н.О., Монгуш А.М., Харитонов А.Н., Ананьева К.И., Басюл И.А.* Решение графической задачи в традиционной и техногенной коммуникативно-когнитивных средах // Эволюционная и сравнительная психология в России. Теория и практика исследований. М.: Когито-Центр, 2017. С. 312—321.
6. Dlib C++ Library, 2019 // URL: dlib.net (дата обращения: 25.11.2019).
7. *Jose I.* Facial mapping (landmarks) with Dlib + python, 2018 // URL: https://towardsdatascience.com/facial-mapping-landmarks-with-dlib-python-160abcf7d672 (дата обращения: 25.11.2019).
8. *Lukezic A., Voj'ir T., Cehovin Zajc L., Matas J., Kristan M.* Discriminative correlation filter tracker with channel and spatial reliability // International Journal of Computer Vision. 2018. https://doi.org/10.1007/s11263-017-1061-3
9. *Mallick S.* Object Tracking using OpenCV, 2017 // URL: https://www.learnopencv.com/object-tracking-using-opencv-cpp-python (дата обращения: 25.11.2019).
10. OpenCV Online Documentation, 2019 // OpenCV: OpenCV modules. URL: https://docs.opencv.org/master/index.html (дата обращения: 25.11.2019)
11. OpenCV Tracking API, 2019 // OpenCV: Tracking API. URL: https://docs.opencv.org/master/d9/df8/group__tracking.html (дата обращения: 25.11.2019).
12. *Dalmaijer E.S.* Python for Experimental Psychologists. Abingdon, United Kingdom: Routledge (part of Taylor and Francis Group), 2016

# TRACKING MOVING OBJECTS ON VIDEO IN PSYCHOLOGICAL STUDIES

***ZHEGALLO A.V.****, *Institute of Psychology of Russian Academy of Sciences; Moscow State University of Psychology and Education, Moscow, Russia, e-mail: zhegs@mail.ru*

The article discusses the possibilities of using OpenCV and dlib libraries to track the position of specified objects on the video image and to track a set of landmark points of the human face. It is shown that these

**For citation:**
Zhegallo A.V. Tracking moving objects on video in psychological studies. *Eksperimental'naya psikhologiya = Experimental psychology (Russia)*, 2019, vol. 12, no. 4, pp. 5—11. doi:10.17759/exppsy.2019120401

* *Zhegallo A.V.* Candidate of Psychological Sciences, Senior Researcher, Institute of Psychology of Russian Academy of Sciences; Senior Researcher, Institute of Experimental Psychology, Moscow State University of Psychology and Education. E-mail: zhegs@mail.ru

problems can be effectively solved with the help of compact programs in Python. The Python language is recommended to the attention of research psychologists as an instrument for analyzing video images and constructing complex experiments.

***Keywords:*** object tracking, face detection, Python, OpenCV, dlib.

*Funding*

The study was carried out within the framework of the state project of the Ministry of Education and Science of the Russian Federation, project 25.3471.2017/PC "Identification of significant components of the cognitive experience of a specialist to their capturing and transfer".

## *References*

1. *Barabanshchikov V.A., Zhegallo A.V., Hoze E.G., Solomonova A.V.,* Neverbal'nye prediktory ocenok dostovernosti/nedostovernosti soobshchaemoj informacii [Nonverbal predictors of assessments of the reliability/unreliability of reported information] Eksperimental'naya psihologiya [Experimental Psychology], 2018 Vol. 11. no 4. pp. 94—106. (In Russ., abstr. in Engl.). doi:10.17759/exppsy.2018110408
2. *Zhegallo A.V.* Tekhnologii ajtrekinga v zadachah sohraneniya i peredachi kognitivnogo opyta [Eyetracking technology in the task of preservation and transmission of the cognitive experience] Eksperimental'naya psihologiya [Experimental Psychology], 2018. Vol. 11. no. 4. pp. 135—141. (In Russ., abstr. in Engl.). doi:10.17759/exppsy.2018110412
3. *McKinney Wes.* Python i analiz dannyh [Python and data analysis] Trans. from Engl. Silkin A.A. Moscow, DMK Press, 2015 (In Russ.).
4. *Nosulenko V.N., Samoilenko E.S.* Polipozicionnoe nablyudenie [Polypositional observation] // Tekhnologii sohraneniya i vosproizvedeniya kognitivnogo opyta [Technologies of preservation and reproduction of cognitive experience] ed. by Nosulenko V.N. Moscow, Institut psikhologii RAN, 2016. pp. 261—278. (In Russ.).
5. *Tovuu N.O., H.O., Mongush A.M., Kharitonov A.N., Ananieva K.I. Basul I.A.* Reshenie graficheskoj zadachi v tradicionnoj i tekhnogennoj kommunikativno-kognitivnyh sredah [Solving of graphic task in traditional and technogenic communicative and cognitive environments] // Evolyucionnaya i sravnitel'naya psihologiya v Rossii. Teoriya i praktika issledovanij. [Evolutionary and comparative psychology in Russia. Theory and practice of research] Moscow, Cogito-Center, 2017. pp. 312—321. (In Russ.).
6. *Dlib C++ Library,* 2019 // URL: dlib.net (дата обращения: 25.11.2019).
7. *Jose I.* Facial mapping (landmarks) with Dlib + python, 2018 // URL: https://towardsdatascience.com/facial-mapping-landmarks-with-dlib-python-160abcf7d672 (date of issue: 25.11.2019).
8. *Lukezic A., Voj'ir T., Cehovin Zajc L., Matas J., Kristan M.* Discriminative correlation filter tracker with channel and spatial reliability // International Journal of Computer Vision, 2018. https://doi.org/10.1007/s11263-017-1061-3
9. *Mallick S.* Object Tracking using OpenCV, 2017 // URL: https://www.learnopencv.com/object-tracking-using-opencv-cpp-python (date of issue: 25.11.2019).
10. *OpenCV Online Documentation,* 2019 // OpenCV: OpenCV modules. URL: https://docs.opencv.org/master/index.html (date of issue: 25.11.2019)
11. *OpenCV Tracking API,* 2019 // OpenCV: Tracking API. URL: https://docs.opencv.org/master/d9/df8/group__tracking.html (date of issue: 25.11.2019).
12. Dalmaijer, E.S. Python for Experimental Psychologists. Abingdon, United Kingdom: Routledge (part of Taylor and Francis Group), 2016