

# Разработка алгоритма отслеживания лица человека на основе применения оптического потока

А.А. Друки<sup>1</sup>, В.Г. Спицын<sup>2</sup>, Ю.А. Болотова<sup>3</sup>, Д. Олива<sup>4</sup>, А.М. Гельгинберг<sup>5</sup>

Национальный исследовательский Томский политехнический университет, Россия  
Университет Гвадалахары, Мексика

<sup>1</sup> ORCID: 0000-0002-8221-6029, [druki@tpu.ru](mailto:druki@tpu.ru)

<sup>2</sup> ORCID: 0000-0001-5978-1321, [spvg@tpu.ru](mailto:spvg@tpu.ru)

<sup>3</sup> ORCID: 0000-0002-8161-3575, [julya.bolotova@gmail.com](mailto:julya.bolotova@gmail.com)

<sup>4</sup> ORCID: 0000-0001-8781-7993, [inform@safety.ru](mailto:inform@safety.ru)

<sup>5</sup> ORCID: 0000-0003-4800-4185, [gelginberg@gmail.com](mailto:gelginberg@gmail.com)

## Аннотация

Статья посвящена разработке алгоритмов и программных средств, обеспечивающих повышение эффективности обнаружения и отслеживания человеческих лиц на видеопоследовательностях. Проведен обзор существующих методов обнаружения и отслеживания объектов на видеопоследовательностях, указаны их достоинства и недостатки. Разработан алгоритм отслеживания лица человека на основе применения оптического потока, использующий в качестве локализации объектов детектор Виолы-Джонса. Предобработка кадра осуществляется с помощью фильтра Гаусса, вычисление потока – пирамидальным методом Лукаса-Канаде. В статье представлены результаты программной реализации алгоритма с описанием основных функций и классов. Проведено тестирование разработанного алгоритма и сравнение результатов работы с аналогами.

**Ключевые слова:** оптический поток, отслеживание объектов, детектирование лиц, компьютерное зрение, классификация образов.

## 1. Введение

Методы компьютерного зрения призваны решать задачи, связанные со сбором зрительной информации и её последующим анализом (детектированием и распознаванием различных классов объектов, предварительной обработкой, выделением деталей, позиционированием их в трехмерном пространстве и т.д.). Результатом обычно является видоизмененное изображение или список значений различных параметров изображения, таких как размер объекта, цвет, координаты, скорость, ориентация по отношению к камере и т.п. Данная информация находит широкий спектр применений во многих областях: системах обеспечения связи и безопасности,

робототехнике, анализе автомобильного потока на автомагистралях, системах дополненной реальности, научных исследованиях, интерактивных и игровых приложениях. Поскольку статические изображения содержат недостаточно информации, чаще всего анализируется видеопоследовательность. Это влечет за собой высокие требования к производительности используемой вычислительной системы и вычислительной эффективности используемых алгоритмов [1].

Задача отслеживания объектов, трекинг (англ. tracking) предполагает определение местоположения движущегося объекта (или нескольких объектов) во времени на видеопоследовательности [2]. Многие современные системы используют различные подходы к задаче

отслеживания объекта на основе математического анализа, математической статистики, теории принятия решений, обработки изображения, методов искусственного интеллекта. Однако в большинстве своем у них обнаруживается ряд проблем, связанных с геометрическими факторами (класс объекта, поза, артикуляция, количество и степень перекрытий) и фотометрическими факторами (освещение, яркость, контрастность), которые оказывают существенное влияние на прогнозируемость движения [3].

Основными требованиями к разрабатываемому алгоритму является обеспечение средней точности обнаружения кадров с лицами на видеопоследовательностях не менее 92% при средней длине трека 40 кадров с момента фиксации лица до его потери.

При отслеживании объектов можно выделить следующие основные подзадачи [4]:

1. Нахождение признаков объектов (англ. feature extraction);
2. Выбор представления цели (англ. target representation);
3. Сопровождение (англ. tracking);
4. Управление целями (англ. target management);
5. Пост-обработка (англ. postprocessing).

В свою очередь к каждой подзадаче имеется несколько методов решения, выбор которых обусловлен применением трекера для конкретных объектов. При выделении признаков могут рассматриваться цвет в различных цветовых пространствах, градиент [3], границы [5], особые точки [5], векторное поле или оптический поток [6], единообразные области [7, 8] и т.д.

Среди множества алгоритмов отслеживания объектов на видеопоследовательностях можно выделить несколько основных.

1. Вычитание фона [9, 10]. Данный метод заключается в вычитании фона изображения и выделении переднего плана. Качество работы данных методов зависит от скорости движения объектов. Как пра-

вило, медленно передвигающиеся объекты обнаруживаются плохо, это связано с главной особенностью методов – нерекурсивным обновлением фона по предыдущим кадрам. Более того, не достигается качественный результат при изменении света либо наличии динамического фона (листья деревьев, струящейся воды и т.п.).

2. Гистограммный анализ [11]. Для обновления фона используется информация об интенсивности пикселей текущего кадра. Считается, что пиксели принадлежат фону, если величина интенсивности области изображения меньше фиксированного порогового значения яркости/цвета, иначе пиксель принадлежит объекту. Основным недостатком гистограммного метода состоит в том, что необходимо использовать дополнительную память и последующее количество операций обращения к ней возрастает.
3. Перебор возможных вариантов смещения [12]. Выбирается метрика для оценки степени сходства фрагментов изображения и задача определения движения сводится к минимизации функции ошибки по всем возможным направлениям смещения. На практике метод полного перебора работает достаточно медленно, поэтому имеет смысл использовать его только в совокупности с другими методами.
4. Байесовский подход к отслеживанию [13]. Данный подход относится к вероятностным методам визуального отслеживания и заключается в построении функции плотности распределения вероятности вектора состояния отслеживаемого объекта. При данном подходе проблема отслеживания заключается в рекурсивном вычислении некоторой степени доверия к состоянию объекта на кадре с учетом всех имеющихся на текущий момент наблюдений.

5. Фильтр частиц [5, 14]. Функция плотности распределения вектора состояний может быть аппроксимирована набором частиц, каждая из которых характеризуется собственным весом (под частицей подразумевается некоторое возможное состояние объекта). Базовым алгоритмом фильтра частиц является алгоритм воспроизведения условной плотности (англ. conditional density propagation), на его основе строятся различные модификации под необходимое отслеживание. Недостатком использования фильтров частиц является необходимость эффективно генерировать выборку из распределения вероятности, поэтому рекомендуется брать простые распределения, допускающие эффективную генерацию выборки.
6. Детекторы [15, 16, 17]. Детекторы, определяющие положение цели на конкретном кадре, так же могут быть использованы для анализа движения объекта. Наиболее популярным и хорошо зарекомендовавшим себя алгоритмом является алгоритм Виолы-Джонса [18]. В данном алгоритме используются признаки Хаара, значение которых вычисляется как разность сумм интенсивностей пикселей в их черных и белых областях.
7. Оптический поток [19]. Оптический поток (англ. optical flow) – это векторное поле скоростей пикселей, которое переводит одно изображение в другое, осуществляя сдвиг каждой точки между этими изображениями. В результате этого определяется видимое движение объектов, поверхностей или краев сцены, получаемое в результате перемещения наблюдателя относительно сцены.

В основе понятия оптического потока лежат предположения о том, что при изменении положения конкретного пикселя от кадра к кадру его яркость и интенсивность не меняются, а ближайшие точки, принадлежащие одному

объекту, в плоскости изображения двигаются с одинаковой скоростью. С каждым пикселем изображения можно связать некоторый вектор скорости, который определяет, какое расстояние "прошел" пиксель в течение временного промежутка между предыдущим и текущим кадрами. С помощью этого уравнения можно найти и построить вектор скорости пикселя.

Уравнение оптического потока содержит две неизвестные и не может быть однозначно решено. Алгоритм Лукаса-Канаде обходит эту неоднозначность за счет использования информации о соседних пикселях в каждой точке. Метод основан на предположении, что в локальной окрестности каждого пикселя значение оптического потока одинаково, таким образом можно записать основное уравнение оптического потока для всех пикселей окрестности.

Плюсы этого метода прежде всего в его локальности, можно искать поток совершенно в любой точке не придавая значение тому, что происходит за ее пределами. Кроме того, метод Лукаса-Канаде нечувствителен к шуму, это связано с выполнением усреднения.

Следует отметить, что на сегодняшний день существует ряд фреймворков, библиотек компьютерного зрения и коммерческих систем, которые обеспечивают хорошие результаты классификации образов на изображениях и видеопоследовательностях. Большинство из них являются закрытыми, что затрудняет возможность их исследования с научной точки зрения. Поэтому возникает потребность в разработке новых алгоритмов классификации образов и их программных реализаций, предназначенных для повышения качества классификации образов. Несмотря на то, что многие новые разрабатываемые алгоритмы могут уступать по скорости и точности существующим коммерческим аналогам, при этом они могут являться основой для дальнейшего развития данной научной области.

В данной работе предлагается оригинальный комплексный алгоритм трекинга лиц. На первом этапе произво-

дится предобработка кадров с применением фильтра Гаусса. Затем проводится локализация объектов на основе применения детектора Виолы-Джонса. На заключительном этапе осуществляется отслеживание лиц на основе вычисления оптического потока методом Лукаса-Канаде.

## 2. Разработка алгоритма отслеживания лица

На основе проведенного анализа существующих методов и алгоритмов, для решения задачи обнаружения лица на изображении был выбран алгоритм Виолы-Джонса. Обнаружение выполняется путем применения классификатора на нескольких локальных участках изображения. Однако если лицо искажается, каскадный классификатор может не сработать и обнаружение внезапно прервется.

Для устранения этого недостатка было решено использовать преимущества оптического потока, который описывает принадлежность пикселей локальной области найденного на кадре лица и вычисляет значения вектора скоростей между двумя последовательно идущими кадрами. В качестве алгоритма вычисления оптического потока выбран метод Лукаса-Канаде за его надежность и гибкость к различным модификациям.

Для ускорения вычисления потока были применены следующие модификации: активация детектора каждые  $N$  кадров; вычисления потока для набора заданных точек; пирамидальная структура и применение фильтрации методом Гаусса на первом уровне пирамиды изображений.

Алгоритм работы трекинга лиц можно представить в виде следующей последовательности шагов:

1. Входной видеопоток. Последовательность описывается как последовательность входных кадров:  $I_1, I_2, \dots, I_n$ .
2. Поиск лица. Детектор Виолы-Джонса фиксирует найденное лицо и область его расположения на кадре.

3. Расстановка характерных точек. В пределах обнаруженной детектором области фиксируются характерные точки. В данном алгоритме в качестве характерной точки выбран центр области, выделенной прямоугольником.
4. Наложение фильтра на кадр. Для устранения шума применяется фильтр Гаусса со значением стандартного отклонения  $\sigma = 1$ .
5. Построение пирамидальной структуры для двух кадров. Два последовательно идущих кадра масштабируются, формируя 4 уровня разрешений.
6. Расчет вектора скорости. Методом Лукаса-Канаде вычисляются вектор скорости на уровнях соседних кадров – от меньшего к большему.
7. Визуализация смещения. Характерная точка на следующем кадре смещается в окрестностях закрепленной области.
8. Обновление информации каждые  $N$  кадров. После заданного количества кадров выполняется повторная активация детектора. Это необходимо для того, чтобы обновить положение характерных точек для ранее присутствовавших в кадре лиц, а также для тех, которые только появились в поле зрения наблюдателя. Значение  $N$  выбрано равным 10 и подобрано экспериментальным путем. Активация детектора через каждые 10 кадров позволяет сэкономить вычислительные ресурсы, сохраняя приемлемое качество детектирования.

Программная реализация алгоритмов выполнялась на языке программирования C++ в среде разработки Microsoft Visual Studio с использованием библиотеки компьютерного зрения OpenCV [20]. Для применения алгоритма Виолы-Джонса в библиотеке OpenCV использовался класс CascadeClassifiers и наборы признаков Хаара `haarcascade_frontalface.xml` и `haarcascade_frontalface_alt.xml`.

Для осуществления поиска лица на вход метода, реализующего классификатор `detectMultiScale()`, подаются следующие данные: вектор для искомого

областей местоположения лиц; коэффициент увеличения сканирующего окна (равен 1,1, подбирался экспериментально); количество соседних окон, в которых также определяется лицо; минимальный и максимальный размер сканирующего окна.

На основе описанных выше методов был создан класс FaceDetector.h с методом `int FaceDetector::detect()`, где реализуется работа классификатора, и методом для отображения прямоугольной области `bool FaceDetector::getFace()`.

Отсчет  $N$ -ого кадра реализован в классе, отвечающем за обработку кадров в текущий момент, `FrameProcessingRunnable.h`. Цикл отсчитывает заданное число  $N$ , после чего включает алгоритмы детектора для обновления/получения информации на 1 кадр и снова передает управление оптическому потоку.

## 2.1. Выбор характерной точки

В качестве характерной точки было решено использовать центр области расположения лица. Поскольку оптический поток рассматривает вектор смещения в окрестности некоторой области, то использовать информацию об одной точке  $P_c$  в предположительном центре лица вполне достаточно (рис. 1), чтобы максимально качественно зафиксировать его на протяжении  $N$  кадров.

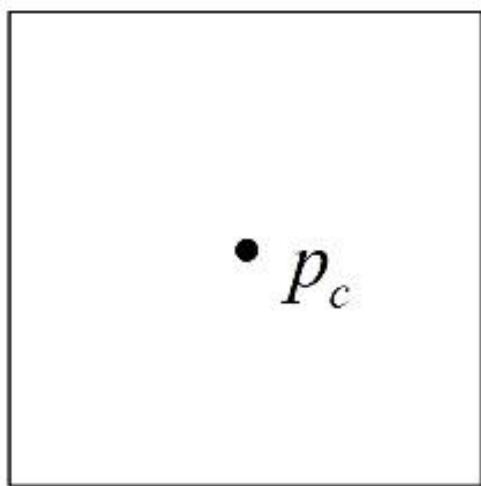


Рис. 1. Центр обнаруженной области

Координаты центра области находятся по формуле:

$$P_c = \left( \frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right),$$

где  $x_1, y_1$  и  $x_2, y_2$  — координаты найденного вектора скорости.

За расстановку характерной точки отвечает метод `int OpticalFlow::putSpecialPoint()` класса `OpticalFlow`.

## 2.2. Наложение фильтра на кадр

Слабым местом алгоритма вычисления оптического потока является то, что на некоторых фрагментах изображения интенсивность может неравномерно меняться без явных закономерностей. Применение пространственного фильтра Гаусса, который является фильтром низких частот, позволяет ослабить высокие частоты и пропустить низкие. Результирующее изображение будет содержать меньше резких деталей вследствие чего будет немного размытым (сглаженным). Такой результат облегчает нахождение потока по заданным характерным точкам, поскольку интенсивность в определенной окрестности нормализуется.

Для построения фильтра Гаусса задается матрица свертки размером  $3 \times 3$ , что реализовано в классе `GaussianFiltering.h`. Программный код приведен в листинге 1.

Листинг 1

```
int k = 3;
float div = 1.0f / (2.0f * (float) Math.PI *
sigma * sigma);
float[,] A = new float[k,k];
for (int i = 0; i < k; i++)
{
    for (int j = 0; j < k; j++)
    {
        A[i, j] = div * (float) Math.Exp(-
(Math.Pow(k-i-1, 2) + Math.Pow(k-
j-1, 2))) / (2*sigma*sigma));
    }
}
```

Параметр стандартного отклонения был выбран равным 1, т.к. при высоких значениях он дает результат с сильным размытием.

### 2.3. Нахождение вектора скорости

На данном этапе применяется построение пирамиды изображений разного масштаба. Размеры двух соседних кадров изменяются в 2 раза по каждой оси до получения 4 уровней (рис. 2).

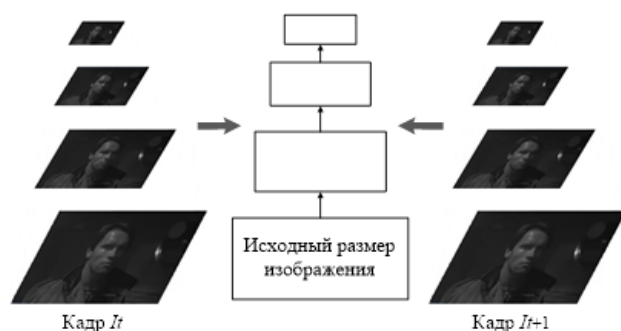


Рис. 2. Построение пирамиды по двум кадрам

Данный способ позволяет обнаружить движение в больших пространственных масштабах, начиная анализ от меньшего изображения к большему. Сдвиг на маленьком изображении будет соответствовать сдвигу на большом изображении.

Функция `cvCalcOpticalFlowPyrLk()` из библиотеки OpenCV используется для реализации пирамидального вычисления оптического потока методом Лукаса-Канаде и имеет следующий набор параметров:

1. Массив `featuresA` – координаты исходных характерных точек.
2. Массив `featuresB` – координаты новых вычисленных точек из массива `featuresA`.
3. Параметр `count` – количество точек в массиве `featuresA`.
4. Параметр `winSize` – размер локальной области, в которой вычисляется значение перемещения.
5. Параметр `level` – глубина пирамиды изображений, если установлен в 0, то пирамида не используется.

6. Параметр `track_error` – разность чисел между координатами конкретной отслеживаемой точки на текущем и следующем кадре.

7. Аргумент `criteria` – условия для окончания вычислений.

Для генерации необходимого условия остановки используется следующая структура:

```
cvTermCriteria(
    int type // CV_TERMCRIT_ITER,
    CV_TERMCRIT_EPS
    ,int max_iter
    ,double epsilon
);
```

Первый аргумент `CV_TERMCRIT_ITER` или `CV_TERMCRIT_EPS` сообщает алгоритму момент прекращения – либо после некоторого количества итераций либо, когда метрика сходимости достигнет некоторого малого значения, а следующие два параметра (`max_iter`, `epsilon`) задают непосредственно критерии останова.

### 2.4. Графический интерфейс пользователя

Графический интерфейс пользователя был разработан с применением средств фреймворка Qt [21].

Разработанная программная система состоит из нескольких модулей, осуществляющих основные функции и алгоритмы: модуль настройки, сохранения и загрузки конфигураций (рис. 3); модуль запуска и отображения результатов отслеживания. В соответствии с шаблонами Qt GUI для интерфейса сгенерированы следующие файлы:

1. `main.cpp` – файл основной программы;
2. `qtfrsmainwindow.h` – заголовочный файл класса основного окна;
3. `qtfrsmainwindow.cpp` – файл реализации класса окна;
4. `qtfrsmainwindow.ui` – файл описания класса образа окна;
5. `qtfrsconfigurationmanager.h` – заголовочный файл класса менеджера конфигураций;

6. qtfrsconfigurationmanager.cpp – файл реализации класса менеджера конфигураций;

7. qtfrsconfigurationmanager.ui – файл описания класса образа менеджера конфигураций.



Рис. 3. Интерфейс основного окна программы

Основное окно содержит главный экран для отображения видео (Dashboard); вкладку для отображения лиц, найденных детектором (Face detection); вкладку для отображения вспомогательной информации, которая активируются при необходимости, и панель управления (Control Panel) с функциями управления видеопотоком (Start, Pause/Continue, Stop); выпадающим списком выбора конфигурации (Configuration) и менеджером настройки конфигураций (Configuration manager).

### 3. Тестирование и анализ результатов

Проведение тестирования, подтверждающего эффективность разработанного алгоритма, проводилось в два этапа. На первом этапе проверялась эффективность вычисления оптического потока с применением низкочастотного фильтра Гаусса на базе синтетических данных Baker et al [22]. На втором этапе

проводилось отслеживание лица человека с использованием видеофайлов эталонной базы Boston Head Tracking [23].

База данных для тестирования оптического потока Baker et al состоит из набора изображений ground-truth (GT), оценивающих параметры вектора скорости. Для хранения файлов GT используется формат .flo, где хранятся координаты характерных точек изображения, а также значение смещения горизонтальной ( $u$ ) и вертикальной ( $v$ ) составляющей вектора скорости оптического потока. Начальные значения точек и GT были получены путем съемки сцены с очень маленьким шагом, в результате чего изменения на фотографиях практически не видны человеческому взгляду. Шаг сдвига камеры отсчитывался с помощью компьютера таким образом, что никакая точка сцены не перемещалась более, чем на 2 пикселя между двумя кадрами. Характерные точки и величина ground-truth были получены

путем анализа изображений с высоким разрешением (3504x2336) при УФ освещении в исходной последовательности. В базе разрешение изображений преобразовано до 584x388.

Для тестирования оптического потока были выполнены следующие шаги:

1. Считывание двух фреймов входных изображений из базы.
2. Считывание ground-truth. Для этого в программу был добавлен модуль обработки файлов расширения .flo.
3. Запуск метода оптического потока без модификаций, а после с предобработкой кадра фильтром Гаусса. Параметры запуска в менеджере конфигураций: Number of frame = empty; optical flow = on; Filter = off и

Number of frame = empty; optical flow = on; Filter = on соответственно.

4. Вычисление показателя эффективности. Наиболее часто используемый показатель эффективности оптического потока – ошибка в конечной точке потока (endpoint error) – определяется по формуле:

$$E = \sqrt{(u - u_{gt})^2 + (v - v_{gt})^2},$$

где  $u, v$  – координаты найденного векторы скорости,  $u_{gt}, v_{gt}$  – координаты вектора *ground-truth*.

Диаграмма средних значений  $E$  для каждого набора изображений представлена на рисунке 4.

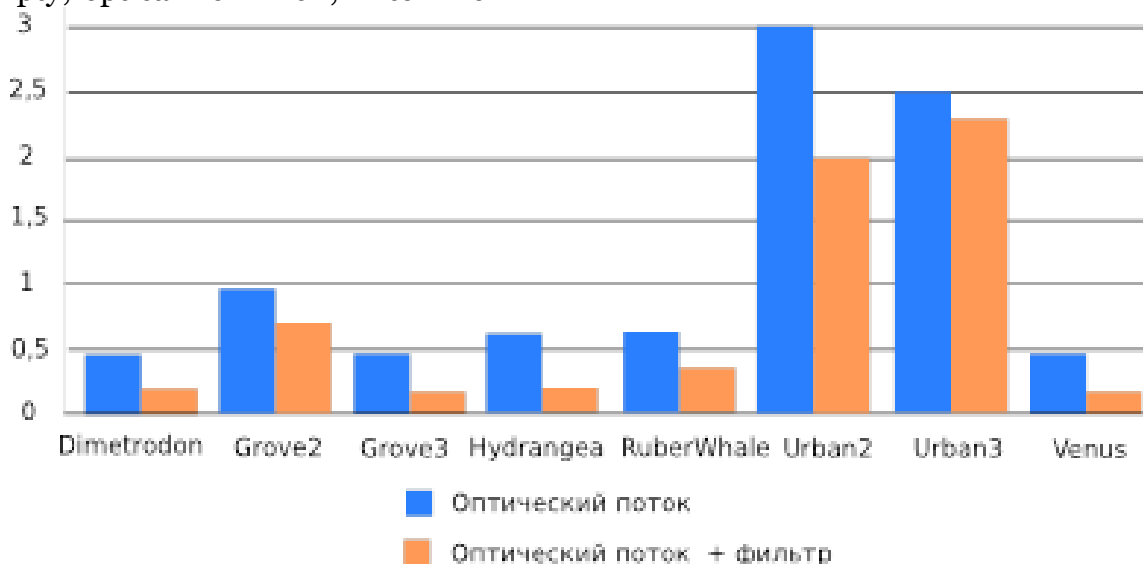


Рис. 4. Диаграмма средних ошибок в конечных точках

Результаты, отраженные на диаграмме, показывают, что средние значение  $E$  получились меньше, когда изображение предварительно обрабатывали фильтром. Это связано с тем, что интенсивность кадра более равномерна, вследствие чего, отслеживание объекта будет происходить достовернее.

Также было выполнено тестирование на базе данных Boston Head Tracking (рис. 5).



Рис. 5. Кадры из базы Boston Head Tracking



База предназначена для тестирования методов отслеживания головы человека при наличии больших поворотов, изменений выражения лица и освещенности.

Тестирование было проведено на 50 видеороликах с разрешением 320x240 при частоте 30 кадров в секунду. На каждом видеоролике представлен один человек, который двигает головой в разные стороны на протяжении 6 секунд. Общее количество кадров для каждого видео 199. Метод, с которым сравнивается предложенный в данной работе алгоритм – это трекер на основе детектора лица Виолы-Джонса. Условия для сравнения методов следующие: трекер на основе детектора работает на каждом кадре, а при подключении оптического потока активируется на каждом десятом кадре (данная величина была подобрана экспериментальным путем).

В процессе тестирования была выполнена поочередная обработка видеоролика двумя методами: сначала без использования оптического потока, а затем с его использованием. Для каждого способа в менеджере конфигураций задаются следующие настройки: Number of frame = 1; optical flow = off; Filter = off и Number of frame = 10; optical flow = on; Filter = on соответственно.

Работа двух сравниваемых методов показана на рисунке 6.



Рис. 6. Процесс работы методов: а) детектор; б) детектор плюс оптический поток. На рисунке 5а видно, что лицевой трекер, основанный только на детекторе, теряет объект при наклоне головы, в то время как при использовании связки детектор плюс оптический поток (рисунк 5б) фокусировка остается, что позволяет не потерять зафиксированный объект.

Качество отслеживания вычисляется путем суммирования кадровых обнаружений лиц в видео и делением полученной суммы на общее количество кадров. Найденное значение выражается в процентах:

$$Vd = \frac{\sum_{i=1}^n d_i}{I} * 100\%$$

где  $Vd$  – качество отслеживания;  $d_i$  – кадровые обнаружения лиц;  $I$  – общее количество кадров.

На рисунках 7-12 приведены диаграммы с полученным значением  $Vd$  для каждого метода. Пять диаграмм соответствуют набору из 10 видео для 5 разных людей. На горизонтальной оси указаны названия видеофайлов.



Рис. 7. Процент обнаруженных кадров с лицами для первого набора видеофайлов

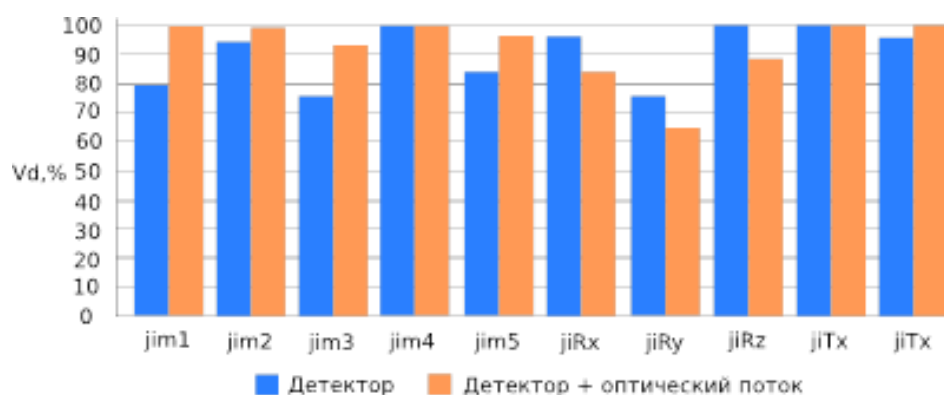


Рис. 8. Процент обнаруженных кадров с лицами для второго набора видеофайлов

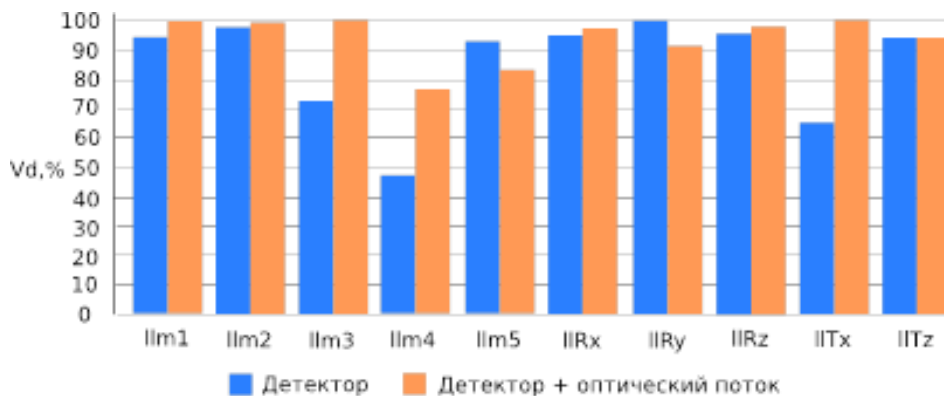


Рис. 9. Процент обнаруженных кадров с лицами для третьего набора видеофайлов

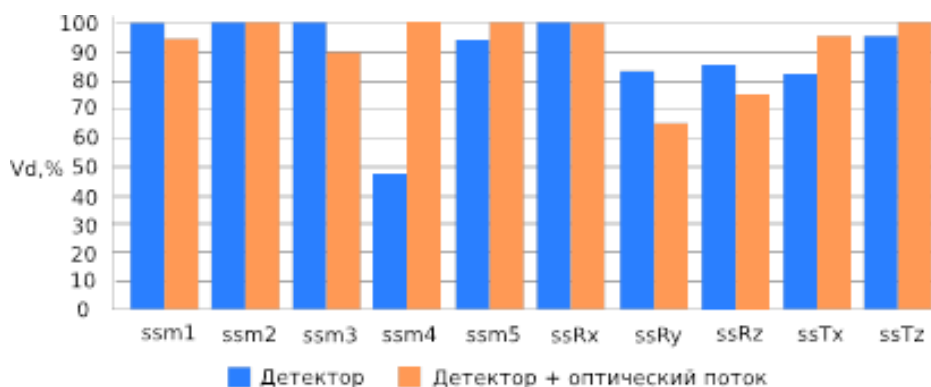


Рис. 10. Процент обнаруженных кадров с лицами для четвертого набора видеофайлов

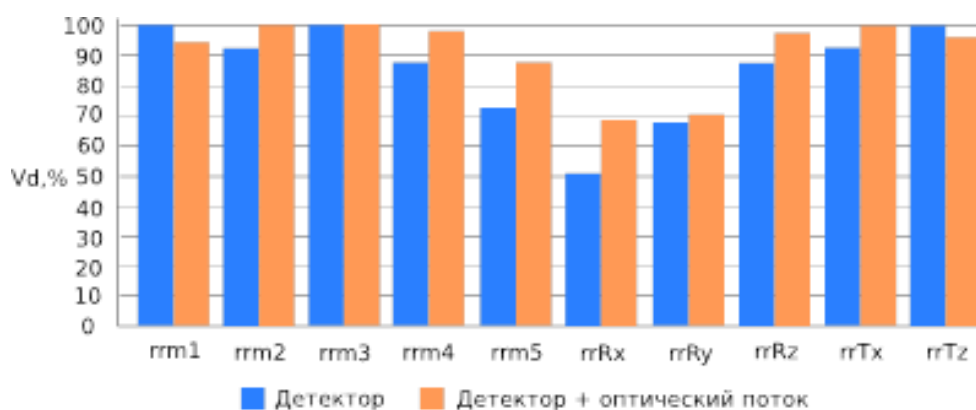


Рис. 11. Процент обнаруженных кадров с лицами для пятого набора видеофайлов

Средняя длина трека вычисляется путем суммирования всех треков (последовательности кадров от начала отслеживания и до прерывания) для видео и деления на их общее количество:

$$\overline{Trk} = \frac{\sum_{i=1}^n trk_i}{n},$$

где  $\overline{Trk}$  – средняя длина трека;  $trk_i$  –  $i$ -ый трек;  $n$  – общее количество треков.

На рисунке 12 приведена диаграмма средней длины трека для каждого набора данных.

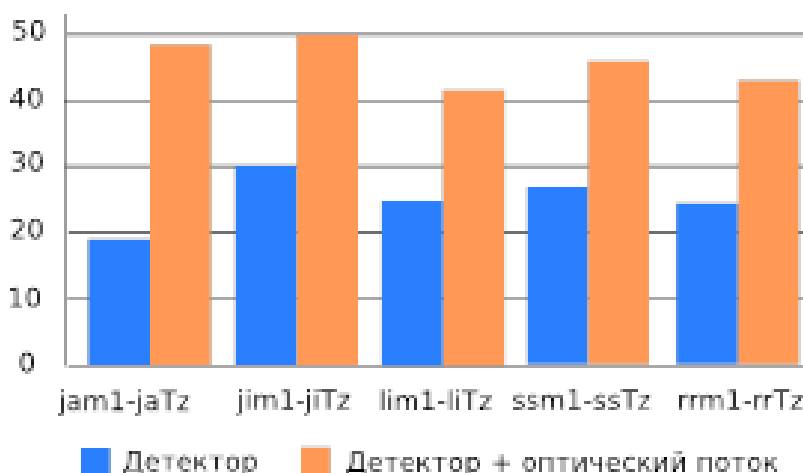


Рис. 12. Средняя длина трека

В результате средний показатель обнаружения для всех видео составляет 88,04% при средней длине трека 26 кадров для отслеживания на основе детектора Виолы-Джонса и 93,36% для разработанного алгоритма отслеживания лиц на основе оптического потока при средней длине трека 45 кадров.

Тестирование проводилось на компьютере под управлением операционной системы Windows 10, x64 обладающего следующими характеристиками: Intel Core i-5-2410M CPU @ 2.30 ГГц, 4 Гб ОЗУ, Intel(R) HD Graphics 3000.

#### 4. Заключение

В рамках данной работы были получены следующие результаты:

1. Разработан алгоритм отслеживания лица человека на основе применения оптического потока, использующий в качестве локализации объектов детектор Виолы-Джонса. Предобработка кадра осуществляется с помощью фильтра Гаусса, вычисление потока – пирамидальным методом Лукаса-Канаде.
2. Проведенное тестирование на базе оценки оптического потока показало, что средний показатель ошибки в конечной точке вектора скорости при фильтрации составляет 0,8 пикселя, в то время как без фильтрации средний показатель составляет 1,2 пикселя.
3. Тестирование на базе видеофайлов дало средний показатель обнаружения кадров с лицами 88,04 % при отслеживании на основе детектора Виолы-Джонса и 93,36% для разработанного алгоритма на основе ОП. Средняя длина трека с момента фиксации лица и до момента его потери составляет для детектора Виолы-Джонса 25 кадров, для оптического потока более 45. Таким образом, требования к разрабатываемому алгоритму, сформулированные во Введении, успешно выполнены.

Работа выполнена при поддержке Российского фонда фундаментальных исследований (РФФИ) в рамках научного проекта № 18-08-00977 А и в рамках Программы повышения конкурентоспособности Томского политехнического университета.

#### Литература

1. Szeliski, R. Computer vision. Algorithms and applications / R. Szeliski // Springer US. – 2010. – P. 957.
2. Maggio, E. Video tracking theory and practice / E. Maggio, A. Cavallaro. – Wiley, 2011. – P. 292.
3. Wu, Y. Tracking appearances with occlusions // Y. Wu, T. Yu, G. Hua // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2003. – P. 289-295.
4. Harris, C. A combined corner and edge detector / C. Harris, M. Stephens // Proceedings of 4th Alvey Vision Conference. – 1988. – P. 147-151.
5. Mitiche, A. Computer vision analysis of image motion by variational methods / A. Mitiche, J. K. Aggarwal // Springer US. – 2013. – P. 207.
6. Fleet, D. Mathematical models for computer vision / D. Fleet, Y. Weiss // Springer US. – 2006. – P. 606.
7. Alatan, A. Image sequence analysis for emerging interactive multimedia services / A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, T. Sikora // IEEE Transactions on Circuits and Systems for Video Technology. – 1998. – P. 525-538.
8. Cavallaro, T. Interaction between high-level and low-level image analysis for semantic video object extraction / T. Cavallaro // Journal on Advances in Signal Processing. – 2004. – Vol. 2008. – № 1. – P. 786-797.
9. Freeman, W. Orientation histograms for hand gesture recognition / W. Freeman, M. Roth // Proceedings of the Workshop on Automatic Face and Gesture Recognition. – 1995. – P. 296-301.
10. Гаганов, В. Сегментация движущихся объектов в видеопотоке / В. Гаганов, А. Конушин // Компьютерная гра-

- фика и мультимедиа. – 2004. - № 2 (3). – Режим доступа: <http://cgm.computergraphics.ru/content/view/67> (дата обращения: 01.02.2017).
11. Heikkila, M. A texture-based method for detecting moving objects / M. Heikkila, M. Pietikainen, J. Heikkila // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2006. – P. 657-662.
  12. Tomasi, C. Good features to track / C. Tomasi, J. Shi // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 1994. – P. 593-600.
  13. Arulampalam, M. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking / M. Arulampalam, S. Maskell, N. Gordon, T. Clapp // IEEE Transactions on Signal Processing. -2002. - № 2 (50). – P. 174-178.
  14. Del Bimbo, A. Particle filter-based visual tracking with a first order dynamic model and uncertainty adaptation / A. Del Bimbo, F. Dini // Computer Vision and Image Understanding. – 2011. - № 6 (115). – P. 771-786.
  15. Comaniciu, D. Kernel-based object tracking / D. Comaniciu, V. Ramesh, P. Meer // IEEE Transaction on Pattern Analysis and Machine Intelligence. – 2003. – P. 564-577.
  16. Rahimi, L. Reducing drift in differential tracking / L. Rahimi, P. Morency, T. Darrell // Computer Vision and Image Understanding. – 2008. – № 2 (109). – P. 97-111.
  17. Kakovkin, P.A. Application of artificial neural networks for solving the problem of two-dimensional geofields reconstruction / P.A. Kakovkin, A.A. Druki // Bulletin of the Tomsk Polytechnic University, Geo Assets Engineering. – 2016. - № 327 (2). – P. 35-43.
  18. Viola, P. Rapid object detection using a boosted cascade of simple features / P. Viola, M. Jones // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2001. – P. 511-518.
  19. Shapiro, L. Computer Vision / L. Shapiro, G. Stockman // Prentice Hall. – 2001. – P. 617.
  20. Библиотека обработки изображений OpenCV [Электронный ресурс]. – URL: <https://opencv.org> (дата обращения: 17.11.2017).
  21. Библиотека для разработки пользовательского интерфейса QT. [Электронный ресурс]. – URL: <https://www.qt.io> (дата обращения: 17.11.2017).
  22. База изображений Baker et al [Электронный ресурс]. – URL: <http://vision.middlebury.edu/flow/> (дата обращения: 17.11.2017).
  23. База видео файлов Boston Head Tracking [Электронный ресурс]. – URL: <https://www.cs.bu.edu/groups/ivc/HeadTracking/Home.html> (дата обращения: 17.11.2017).

## Development of the human face tracking algorithm based on the optical flow application

A.A. Druki<sup>1</sup>, V.G. Spitsyn<sup>2</sup>, Yu.A. Bolotova<sup>3</sup>, D. Oliva<sup>4</sup>, A.V. Gelginberg<sup>5</sup>

National Research Tomsk Polytechnic University, Russia  
University of Guadalajara, Mexico

<sup>1</sup> ORCID: 0000-0002-8221-6029, [druki@tpu.ru](mailto:druki@tpu.ru)

<sup>2</sup> ORCID: 0000-0001-5978-1321, [spvg@tpu.ru](mailto:spvg@tpu.ru)

<sup>3</sup> ORCID: 0000-0002-8161-3575, [julya.bolotova@gmail.com](mailto:julya.bolotova@gmail.com)

<sup>4</sup> ORCID: 0000-0001-8781-7993, [inform@safety.ru](mailto:inform@safety.ru)

<sup>5</sup> ORCID: 0000-0003-4800-4185, [gelginberg@gmail.com](mailto:gelginberg@gmail.com)

### Abstract

The article is devoted to the development of algorithms and software to increase the efficiency of detection and tracking of human faces on video sequences. The review of existing methods of objects detection and tracking on video sequences indicates their advantages and disadvantages. A human face tracking algorithm based on the application of an optical flow is developed, which uses the Viola-Jones detector to objects localization. The preprocessing of the frame is carried out using the Gaussian filter, flow calculation by the pyramidal Lucas-Kanade method. The article presents the results of the algorithm software implementation with description of the main functions and classes. The developed algorithm was tested and comparison of work results with analogues was carried out.

**Keywords:** optical flow, object tracking, face detection, computer vision, image classification.

### References

1. Szeliski, R. Computer vision. Algorithms and applications / R. Szeliski // Springer US. – 2010. – P. 957.
2. Maggio, E. Video tracking theory and practice / E. Maggio, A. Cavallaro. – Wiley, 2011. – P. 292.
3. Wu, Y. Tracking appearances with occlusions // Y. Wu, T. Yu, G. Hua // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2003. – P. 289-295.
4. Harris, C. A combined corner and edge detector / C. Harris, M. Stephens // Proceedings of 4th Alvey Vision Conference. – 1988. – P. 147-151.
5. Mitiche, A. Computer vision analysis of image motion by variational methods / A. Mitiche, J. K. Aggarwal // Springer US. – 2013. – P. 207.
6. Fleet, D. Mathematical models for computer vision / D. Fleet, Y. Weiss // Springer US. – 2006. – P. 606.
7. Alatan, A. Image sequence analysis for emerging interactive multimedia services / A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, T. Sikora // IEEE Transactions on Circuits and Systems for Video Technology. – 1998. – P. 525-538.
8. Cavallaro, T. Interaction between high-level and low-level image analysis for semantic video object extraction / T. Cavallaro // Journal on Advances in Signal Processing. – 2004. – Vol. 2008. – No 1. – P. 786-797.

9. Freeman, W. Orientation histograms for hand gesture recognition / W. Freeman, M. Roth // Proceedings of the Workshop on Automatic Face and Gesture Recognition. – 1995. – P. 296-301.
10. Gaganov, V. Segmentatsiya dvizhushhikh ob'ektov v videopotoke / V. Gaganov, A. Konushin // Komp'yuternaya grafika i mul'timedia. – 2004. - № 2 (3). – <http://cgm.computergraphics.ru/content/view/67> (date of access: 01.02.2017). [In Russian]
11. Heikkila, M. A texture-based method for detecting moving objects / M. Heikkila, M. Pietikainen, J. Heikkila // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2006. – P. 657-662.
12. Tomasi, C. Good features to track / C. Tomasi, J. Shi // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 1994. – P. 593-600.
13. Arulampalam, M. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking / M. Arulampalam, S. Maskell, N. Gordon, T. Clapp // IEEE Transactions on Signal Processing. -2002. - № 2 (50). – P. 174-178.
14. Del Bimbo, A. Particle filter-based visual tracking with a first order dynamic model and uncertainty adaptation / A. Del Bimbo, F. Dini // Computer Vision and Image Understanding. – 2011. - № 6 (115). – P. 771-786.
15. Comaniciu, D. Kernel-based object tracking / D. Comaniciu, V. Ramesh, P. Meer // IEEE Transaction on Pattern Analysis and Machine Intelligence. – 2003. – P. 564-577.
16. Rahimi, L. Reducing drift in differential tracking / L. Rahimi, P. Morency, T. Darrell // Computer Vision and Image Understanding. – 2008. – № 2 (109). – P. 97-111.
17. Kakovkin, P.A. Application of artificial neural networks for solving the problem of two-dimensional geofields reconstruction / P.A. Kakovkin, A.A. Druki // Bulletin of the Tomsk Polytechnic University, Geo Assets Engineering. – 2016. - № 327 (2). – P. 35-43.
18. Viola, P. Rapid object detection using a boosted cascade of simple features / P. Viola, M. Jones // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2001. – P. 511-518.
19. Shapiro, L. Computer Vision / L. Shapiro, G. Stockman // Prentice Hall. – 2001. – P. 617.
20. The image processing library OpenCV. Source: <https://opencv.org>.
21. The GUI development library QT. Source: <https://www.qt.io>.
22. Database of images Baker et al. Source: <http://vision.middlebury.edu/flow>.
23. Database of videofiles Boston Head Tracking. Source: <https://www.cs.bu.edu/groups/ivc/HeadTracking/Home.html>.