

# Программа, условия и описание конкурса по сопоставлению силуэтов автомобильных номеров методами компьютерного зрения

## Цель конкурса

Проводимый ЗАО «Элвис-Неотек» конкурс призван привлечь молодых специалистов к улучшению программного обеспечения, которое в автоматическом режиме контролирует транспортные потоки на изображении видеокамер.

В рамках продукта «Travio», выпускаемого компанией «Элвис-Неотек», реализовано отслеживание движения автомобиля с помощью детектора автомобильных номеров, см. рис. 1.



Рисунок 1. Пример работы алгоритма отслеживания автомобилей

На ближнем плане автомобильный номер отслеживается с высокой надёжностью с помощью механизма распознавания номеров.

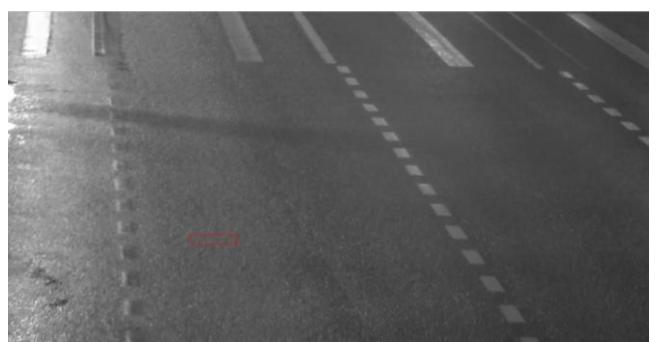


Рисунок 2. Примеры ошибок в отслеживании номера

На дальнем плане автомобильный номер отслеживается по косвенным признакам, и поэтому возможны случаи, когда выдавая предполагаемое положение номера, алгоритм отслеживания выдаёт гипотетическое расположение номера в том месте, где его не видно, например, из-за того, что он временно загорожен впереди стоящим автомобилем, или из-за ошибки сопровождения, см. рис. 2.

Задачей конкурса является разработка алгоритма, на вход которого будут поданы два изображения, одно из которых содержит ничем не загороженное изображение автомобильного номера на ближнем плане с координатами прямоугольника на изображении, ограничивающего изображение номера, и изображение автомобиля на дальнем плане с предполагаемыми координатами ограничивающего номер прямоугольника. В качестве ответа функция должна вернуть одно из двух: либо номер на дальнем плане виден полностью, либо он загорожен. Всем конкурсантам будет предоставлена обучающая тестовая выборка пар изображений, для которых отмечено, перегорожен ли номер автомобиля на заднем плане. Проверка качества представляемых конкурсантами реализаций алгоритма будет производиться с помощью «секретного» тестового набора.

## Условия конкурса

Конкурс проходит с 23.04.2015 по 07.05.2015.

Участвовать в конкурсе может любой желающий.

Призовой фонд конкурса составляет 20 тысяч рублей.

Если количество участников<sup>1</sup> конкурса от 1 до 3, то призовое место – одно, и вознаграждение за победу – 20 тысяч рублей.

Если количество участников конкурса от 4 до 6, то призовых места два, за первое место – вознаграждение 15 тысяч рублей, за второе – 5 тысяч рублей.

Если количество участников конкурса более 6, то призовых места три, за первое место – награда 14 тысяч рублей, за второе – 4 тысячи рублей, за третье – 2 тысячи рублей.

При равных результатах вверху конкурсной таблицы награды за поделенные места складываются и делятся поровну.

Конкурс состоит из этапа регистрации участников, этапа разработки алгоритма и этапа проверки алгоритма.

На этапе регистрации участников желающие соревноваться должны обратиться оргкомитет, и оставить свои персональные данные: имя, фамилия, электронная почта. При регистрации необходимо иметь с собой носитель информации со свободным объёмом несколько гигабайт. На этот носитель после регистрации будут скопированы тестовые данные в специальном формате, и утилита для чтения и разметки тестовых данных. См. приложение 1 с описанием утилиты разметки и формата тестовых данных.

---

<sup>1</sup> Под количеством участников понимается число участников, прошедших квалификацию (см. далее), а не регистрацию.

На этапе разработки алгоритмов каждый участник с помощью утилиты разметки формирует тот объём тестовых данных, который он считает необходимым, и реализует один из двух вариантов функций, сигнатура которых представлена в приложении 2а (вариант, не зависящий от сторонних библиотек) или в приложении 2б (вариант, подразумевающий использование библиотеки OpenCV). Основная функция должна быть определена в файле pattern\_matcher.h. Задача функции – с помощью изображения, на котором номер виден, определить, виден ли номер на изображении, для которого нет уверенности, что номер не загорожен и вообще присутствует (см. рис. 2). В качестве входных данных кроме изображения в функцию передаётся прямоугольник, ограничивающий номер на шаблонном изображении, и прямоугольник, определяющий примерное положение номера на изображении, которое надо проверить. Приложение, загружающее тестовые файлы изображений и передающее их в разрабатываемую функцию, должно быть реализовано в отдельных файлах, т.е. файлы реализации функции должны составлять отдельную независимую библиотеку. Кроме ответа, присутствует ли номер на проверяемом изображении, программа должна выдать уточнённое положение рамки с номером.

В файлах реализации функции допускается использовать OpenCV версии 2.4.8. Другие библиотеки могут поставляться в виде дополнительных исходников, при условии, что архив с исходными файлами реализации функции не будет превышать по объёму 5МБ. Код должен быть не зависящим от платформы (в основном коде и в коде зависимых библиотек допустимо использовать только стандартные функции C++03).

Чтение файлов в процессе работы функции не допускается. Исходные данные для работы функций должны быть встроены в код.

Представляемые для проверки исходные файлы должны находиться в одном каталоге.

На этапе проверки каждый участник может сделать несколько попыток представить свою реализацию кода. Попытка представления заключается в отправке zip-архива с исходными файлами реализации алгоритма на адрес [anatoly@elvees.com](mailto:anatoly@elvees.com) с почтового адреса, указанного участниками при регистрации. Проверка заключается в компиляции кода в составе программы, которая запустит тесты на «секретной» тестовой выборке. В результате теста возможны следующие результаты:

1. Попытка теста будет провалена, и участнику, сделавшему попытку, будет выслано письмо с уведомлением о неудаче по нижеперечисленным причинам.
  - 1.1. Программа не скомпилировалась<sup>2</sup> (в этом случае лог с ошибками будет выслан по почте участнику).
  - 1.2. Программа скомпилировалась, но в процессе работы возникло необработанное исключение.
  - 1.3. Была замечена утечка памяти (организаторы не гарантируют выявление утечек, но выборочные проверки будут).
  - 1.4. Время обработки одной из пары кадров превысило 200мс.
  - 1.5. Результаты программы будут отличаться в худшую сторону от «базового» результата более, чем на 10%. «Базовый» результат – это результат работы функции сравнения с помощью корреляции.
  - 1.6. Квалификацию уже прошли 15 человек.

---

<sup>2</sup> Рекомендуем вспомогательные функции скрывать в namespace, либо использовать сложные имена во избежание конфликтов сборки проверяющего код проекта.

- Компиляция и запуск пройдут успешно, и в результате для «секретного» набора пар тестовых изображений будет посчитано число ошибок: сколько раз функция неправильно определила, есть ли возле рамки номер, или его нет. Кроме того, будет оценено среднее время обработки пары кадров, и для некоторых изображений точность коррекции положения номера на проверяемой картинке. Точность позиционирования будет определена как среднее отношение площади разности множества точек прямоугольников, выданных программой, и прямоугольников, отмеченных вручную, к площади прямоугольников ручной разметки.

В случае успешной попытки участник будет признан прошедшим квалификацию, и результаты работы программы будут помещены в таблицу следующего вида:

Участник	Место	Число ошибок	Точность позиционирования	Время работы
	1			
	2			
	3			

Места участникам присуждаются с помощью сортировки сначала по числу ошибок, затем (при одинаковом числе ошибок) по точности позиционирования (сравнение будет производиться с учётом погрешности, которую положим равной 5%, т.е. показатели, отличающиеся не более, чем на 5%, считаются одинаковыми) и затем по времени работы (также в пределах погрешности, равной 10%). Подробнее о ранжировании см. приложение 3.

При определении места участника учитывается лучшая попытка. В тех случаях, когда положение участников в турнирной таблице меняется, уведомление об этом посыпается всем участникам конкурса на почту.

В случае интенсивного наплыва заявок на тестирование, заявки обрабатываются по очереди, из двух необработанных заявок участника тестируется код последней заявки.

## Программа конкурса

Этап регистрации проходит с 23.04.2015 по 27.04.2015.

Этап разработки проходит с 23.04.2015 по 07.05.2015.

Этап проверки проходит с 05.05.2015 по 07.05.2015, с 10:00 по 18:00 первые два дня и с 10:00 по 15:00 в последний день.

Награждение участников в случае, если будет хотя бы одна прошедшая квалификацию проверка, будет 12.05.2015.

## Приложение 1. Утилита разметки и формат тестовых данных

Утилита разметки – это программа FrameChecker.exe, которая будет выдана участникам конкурса вместе с набором тестовых данных. Она умеет открывать файлы CROSSROAD<длинный\_цифровой\_идентификатор>.zip, которые также представляются участником конкурса.

Запустив FrameChecker.exe, мы можем выбрать пункт меню «Загрузить» диалога, и выбрать один из файлов CROSSROAD<...>.zip. Появится несколько вкладок с изображениями рамок. Если рамка рядом с номером (смещена не далее двух высот букв от положения номера), надо нажать кнопку «Не заслонён». Если номер виден только частично, или вообще не видно даже транспортного средства с номером, надо нажать кнопку «Заслонён». При нажатии на кнопку

«Заслонён» программа сохранит изображение в папку Images\obscured, при нажатии на кнопку «Не заслонён» программа сохранит изображение в папку Images\clear, и текущая вкладка с изображением будет закрыта. Если повторить эту операцию для всех вкладок, то файл CROSSROAD<...>.zip будет перемещён из исходного каталога в каталог Checked. Далее можно снова выбирать меню «Загрузить», по умолчанию будет открыт последний каталог, из которого читали файлы CROSSROAD<...>.zip.

В результате обработки будут сформированы изображения png в каталогах Images\clear и Images\obscured, формат имени которых задаётся следующим образом: <index>\_<subindex>\_<type>\_(x1,y1,x2,y2)<пробел><гос. номер>.png. В данном формате <index> определяет номер группы картинок, которые относятся к одному и тому же автомобилю, зафиксированному системой. <subindex> позволяет различать разные картинки внутри одной группы (соответствует номеру кадра). <type> является символом R для изображений, для которых рамка номерного знака установлена не точно (такие изображения должны подвергаться проверке на видимость номера внутри изображения), и символом O для изображений, в которых номерной знак точно обведён рамкой (такие изображения должны быть шаблонными). Определённый системой «Travio» государственный номер приводится скорее в информационных целях, что, впрочем, не мешает использовать его при построении алгоритма.

## Приложение 2а. Сигнатура функции без OpenCV

```
#define USE_RAW_PATTERN_MATCHER
namespace raw_pattern_matcher {
    struct Rect {
        int x1, y1;//один из углов прямоугольника
        int x2, y2;//противоположный к (x1, y1) угол прямоугольника
    };
    ///! Searches image_to_search for pattern defined as rectangle
    ///! in a image_with_pattern which position and size are like
    ///! approximate_expected_location.
    ///! If pattern is found returns true and fills
    ///! approximate_expected_location by found location.
    ///! Otherwise returns false and doesn't modify
    ///! approximate_expected_location.
    ///! Rectangles coordinates are image-based and zero-indexed.
    int IsPatternOnImageRaw(const unsigned char * image_with_pattern,
                           int w, int h, int bpl, int bpp, bool is_bottom_base,
                           const Rect & pattern_location,
                           const unsigned char * image_to_search,
                           int w2, int h2, int bpl2, int bpp2, bool is_bottom_base2,
                           Rect & approximate_expected_location);
}
```

## Приложение 2б. Сигнатура функции с использованием OpenCV

```
#include "opencv2/core/core.hpp"
///! Searches image_to_search for pattern defined as rectangle
///! in a image_with_pattern which position and size are like
///! approximate_expected_location.
///! If pattern is found returns true and fills
///! approximate_expected_location by found location.
///! Otherwise returns false and doesn't modify
///! approximate_expected_location. Both images are top based
///! Rectangles coordinates are top-based too and zero-indexed.
```

```
bool IsPatternOnImage(
    const cv::Mat<uint8_t>& image_with_pattern,
    const cv::Rect& pattern_location,
    const cv::Mat<uint8_t>& image_to_search,
    cv::Rect* approximate_expected_location
);
```

### Приложение 3. Алгоритм ранжирования мест.

Алгоритмы, по одному от каждого участника, сортируются по количеству ошибок в порядке убывания. Алгоритмы делятся на группы. В первую группу попадает первый по порядку алгоритм, а также алгоритмы, отличающиеся по количеству ошибок не более, чем на 3% от первого. В каждую следующую группу попадает первый из оставшихся алгоритмов, а также алгоритмы, которые отличаются от первого в группе по количеству ошибок не более, чем на 3%.

В каждой группе, в которую попало более одного алгоритма, производится дополнительная сортировка и деление на подгруппы по точности позиционирования с допуском 5% по алгоритму, аналогичному описанному выше.

Выделенные подгруппы могут быть далее поделены на более мелкие подгруппы тем же алгоритмом по среднему времени работы с допуском 10%.

Алгоритмы, попавшие в одну группу, образованную после трёх описанных процедур, считаются эквивалентными.