

## Поиск текстовых блоков на изображении

*Столяров Евгений Александрович*

КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ ИМЕНИ В.И. ВЕРНАДСКОГО  
ТАВРИЧЕСКАЯ АКАДЕМИЯ  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ИНФОРМАТИКИ (ГРУППА 601-И)  
e-mail: stolyarov.evgeniy@gmail.com

### ВВЕДЕНИЕ

Поиск текстовых блоков на изображении подразумевает нахождение областей, как правило прямоугольных, которые содержат текстовую информацию. Данная задача актуальна, так как является одним из этапов оптического распознавания символов [1], которое активно развивается в последние десятилетия. Поэтому исследование и разработка алгоритма быстрого и точного определения областей может найти широкое применение.

Чаще всего основными шагами, с точностью до порядка, распознавания являются:

- (1) Предобработка изображения;
- (2) Поиск текстовых блоков;
- (3) Последовательное выделение абзацев, строк, слов, символов;
- (4) Выделения признаков классифицируемого символа;
- (5) Распознавания символов;
- (6) Постобработка результата.

Только после окончания последнего этапа можно будет говорить, нашёл ли алгоритм какую-либо текстовую информацию и, соответственно, саму текстовую область.

### 1. АЛГОРИТМ ВЫДЕЛЕНИЯ ГРАНИЦ

Одним из самых популярных алгоритмов выделения границ является детектор границ Кэнни [2], разработанный Джоном Кэнни в 1986 году. Хотя его работа была проведена на заре компьютерного зрения, детектор границ Кэнни до сих пор является одним из лучших детекторов. Кроме особых частных случаев трудно найти детектор, который бы работал существенно лучше, чем детектор Кэнни.

Шаги алгоритма:

- (1) Сглаживание. Размытие изображения для удаления шума;
- (2) Поиск градиентов. Границы отмечаются там, где градиент изображения приобретает максимальное значение;

- (3) Подавление не-максимумов. Только локальные максимумы отмечаются как границы;
- (4) Двойная пороговая фильтрация. Потенциальные границы определяются порогами;
- (5) Трассировка области неоднозначности. Итоговые границы определяются путём подавления всех краёв, несвязанных с определенными (сильными) границами.

## 2. АЛГОРИТМ MAXIMALLY STABLE EXTREMAL REGION (MSER)

На вход алгоритму MSER поступает полутоновое изображение. Далее, по данному изображению создаётся последовательность изображений, которая получается следующим образом. Последовательно для каждого значения от 0 до 255 изображение бинаризуется по порогу. Белые области в последовательности называются областями экстремума. Проанализировав данную последовательность, можно обнаружить сколько уровней та или иная область экстремума не изменялась. Таким образом, выбрав значение порога  $T$ , можно выделить регионы, которые не изменялись в последовательности по крайней мере  $T$  раз. Такие регионы будут называться максимально устойчивыми областями экстремума. Достоинства алгоритма MSER [3] в том, что он хорошо подходит для поиска кандидатов в символы. Минусы алгоритма — он находит достаточно много регионов, не содержащих символы, и достаточно чувствителен к размытым изображениям, в которых алгоритм MSER может рассматривать несколько символов как один регион.

## 3. ПОИСК ОБЛАСТЕЙ-КАНДИДАТОВ

Для предобработки изображения используется библиотека OpenCV [4]. Библиотека также содержит реализацию вышеописанных алгоритмов. В данной работе выбор сделан на композицию различных подходов. Для определения областей, в которых находится текст, выполняются следующие шаги:

- (1) Бинаризация [5];
- (2) Поворот изображения;
- (3) Применение морфологических операций [6];
- (4) Поиск контуров и их анализ.

Для определения порога бинаризации используется метод Оцу [7], в основе которого лежит проверка гистограммы яркости на наличие двух пиков.

Для поворота изображения используется преобразование Хафа [8] — метод для поиска линий, кругов и других простых форм на изображении. Преобразование Хафа основывается на представлении искомого объекта в виде параметрического уравнения. Параметры этого уравнения представляют фазовое пространство (пространство Хафа). Затем, берётся двоичное изображение, перебираются все точки границ и делается предположение, что точка принадлежит линии искомого объекта — для каждой точки изображения рассчитывается нужное уравнение и получаются необходимые параметры, которые сохраняются в пространстве Хафа.

Финальным шагом является обход пространства Хафа и выбор максимальных значений, за которые «проголосовало» больше всего пикселей картинки, что и даёт нам параметры для уравнений искомого объекта.

Морфологические операции — размывание и растягивание, используют фильтр (ядро), размеры которого рассчитываются из эвристических соображений.

Поиск контуров в OpenCV реализован на основе [9]. После этого шага найденных областей может быть слишком много, что усложнит работу на последующих шагах распознавания текста. Поэтому на основе эвристик контуры фильтруются и остаются только наиболее информативные.

После проделанных шагов полученная информация используется для дальнейшего распознавания, а именно: выделения слов, символов и их классификации.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] OCR – [https://en.wikipedia.org/wiki/Optical\\_character\\_recognition](https://en.wikipedia.org/wiki/Optical_character_recognition)
- [2] John Canny *A Computational Approach to Edge Detection.* – [https://perso.limsi.fr/vezien/PAPIERS\\_ACS/canny1986.pdf](https://perso.limsi.fr/vezien/PAPIERS_ACS/canny1986.pdf)
- [3] MSER – [https://en.wikipedia.org/wiki/Maximally\\_stable\\_extremal\\_regions](https://en.wikipedia.org/wiki/Maximally_stable_extremal_regions)
- [4] OpenCV – <http://opencv.org/documentation.html>
- [5] Бинаризация – <https://habrahabr.ru/post/128768/>
- [6] Морфологические операции – [http://docs.opencv.org/2.4/doc/tutorials/imgproc/opening\\_closing\\_hats/opening\\_closing\\_hats.html](http://docs.opencv.org/2.4/doc/tutorials/imgproc/opening_closing_hats/opening_closing_hats.html)
- [7] Метод Оцу – <https://en.wikipedia.org/wiki/Otsu>
- [8] Преобразование Хафа – <http://www.ict.edu.ru/ft/002407/num1degt.pdf>
- [9] Suzuki, S. and Abe, K. *Topological Structural Analysis of Digitized Binary Images by Border Following.* – 1985 – 32-46