

# Отчёт по практическому заданию по обработке и сегментации изображений

Михаил Солоткий

5 апреля 2018 г.

## Постановка задачи

Дано 7 изображений карточек игрового набора «Геометрика». На каждой карточке изображена геометрическая фигура, а также (необязательно) некоторые канцелярские принадлежности или их части. Карточки могут накладываться друг на друга. Фон, на котором расположены карточки, может быть монотонным или пёстрым. Примеры приведены ниже.



Необходимо разработать программу, использующую точечные и пространственные преобразования, которая решает следующие задачи:

- 1 Сегментация карточек: для исходного изображения создать и сохранить размеченную копию, где под разметкой понимается создание надписи на каждой карточке с её номером. Порядок нумерации не имеет значения.
- 2 Определение геометрической фигуры на карточке:
  - А) многоугольник или фигура с гладкой границей.
  - Б) для многоугольников определить количество вершин.
  - В) определить, является ли многоугольник выпуклым.

Необходимо разметить каждую карточку следующим образом: если на карточке изображена фигура с гладкой границей, не размечать её, если изображён многоугольник, создать надпись вида  $P_nC$  или  $P_n$ , где вместо  $n$  необходимо подставить количество вершин многоугольника, букву  $C$  надо добавлять в случае, если многоугольник является выпуклым. Ниже приведены примеры результатов работы программы.

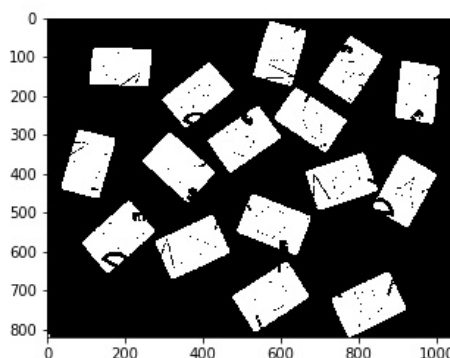
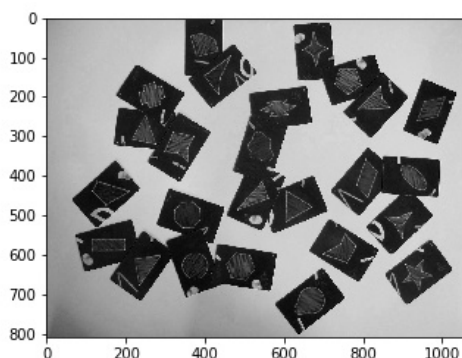


## Интерфейс программы

Решение обеих задач представлено в одном IPython notebook под названием `experiments.ipynb`. Ноутбук разбит на блоки, отвечающие этапам решения задач, все промежуточные изображения и гистограммы выводятся на экран. Для того, чтобы запустить обработку изображения необходимо указать путь к нему (переменная `filename` в блоке «Чтение изображения»). Если далее последовательно запустить все ячейки ноутбука, начнётся процесс решения обеих задач. Результаты будут сохранены в той же директории, что и исходное изображение. Результат решения первой задачи будет иметь пометку `LABELED` в названии, а результат решения второй задачи – пометку `POLYGONS`. В основном в реализации использовались морфологические операции. Для работы с ними использовалась библиотека `skimage`.

## Решение задачи сегментации

Решение задачи в случае монотонного фона отличается от решения задачи с пёстрым фоном. Эксперименты показали, что в случае монотонного фона достаточно работать с красным каналом исходного изображения, тогда как при пёстром фоне применение тех же преобразований приводило к плохому качеству. Связано это с цветом монотонного фона на выданных изображениях. Сами карточки синие, но фон светло-синий. Далее производилась бинаризация методом Оцу. Ниже приведён примеры красного канала в сером цвете и результат бинаризации.

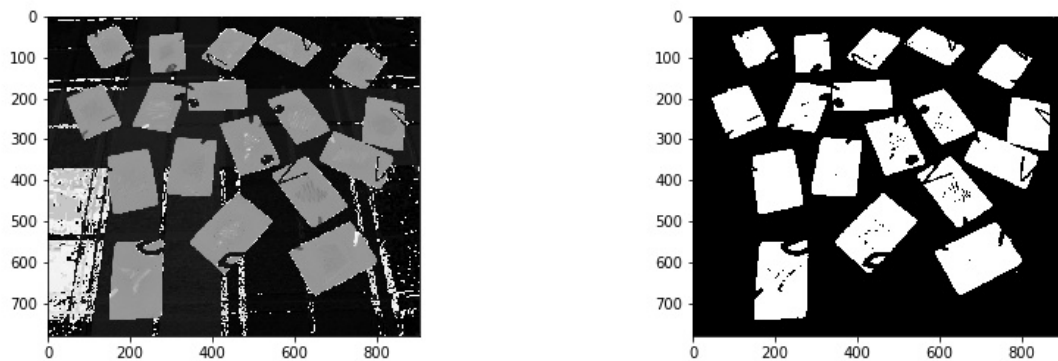


Для осуществления бинаризации изображений с пёстрым фоном происходил переход в систему `hsv` (тон, насыщенность, значение) и вместо красного канала обрабатывался канал

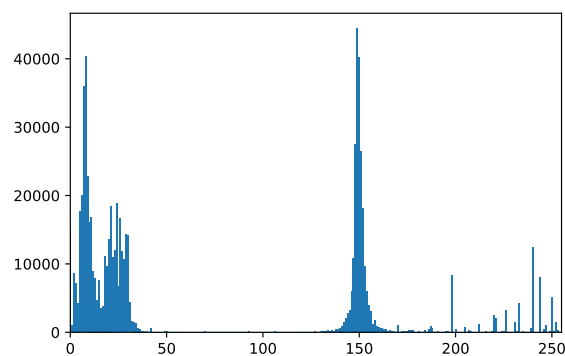
тона. Канал тона характеризует цвет пикселя в RGB системе, Значение канала тона есть значение координаты точки на шкале оттенков.



Видно, что синий цвет расположен примерно посередине на шкале оттенков далеко от красного и жёлтого цветов. Также экспериментальным путём было получено, что по каналу тона можно хорошо бинаризовать изображение. В случае изображений с монотонным фоном использование канала тона не приводит к хорошей бинаризации, так как всё изображение состоит из оттенков синего, которые находятся близко на шкале оттенков. Ниже приведены примеры канала hue в сером цвете и бинаризованного изображения.



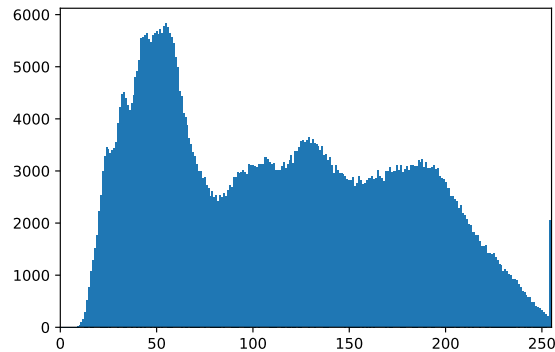
Можно заметить, что теперь нужно разбить множество значений пикселей не одним порогом, а двумя, и метод Оцу в стандартной форме неприменим. Попытки воспользоваться методом Оцу на части изображений приводили к плохой сегментации. Далее помогло наблюдение гистограммы.



Было похоже, что можно установить пороги вокруг моды распределения. Было выбрано отступить на 15 единиц влево и 15 вправо, так как на многих изображениях пик выглядел примерно так же, и такие пороги давали хорошую бинаризацию. Стоит сделать замечание, что задачу выделения карточек на данном этапе не обязательно решать точно. В случае если пороги окажутся слишком мягкими, будут выделены небольшие области,

не относящиеся к карточкам, но дальше будет видно, что это не проблема, так как будет применяться эрозия и удаление компонент с малой площадью. Случай, когда пороги слишком жёсткие маловероятен, так как можно ожидать от новых изображений такого же распределения цвета на карточках, как и на выданных примерах (они же взяты из одного игрового набора).

Далее нужно было придумать правило, по которому происходил выбор между RGB и HSV системами. Если изображение с пёстрым фоном, гистограмма красного канала выглядит примерно так:

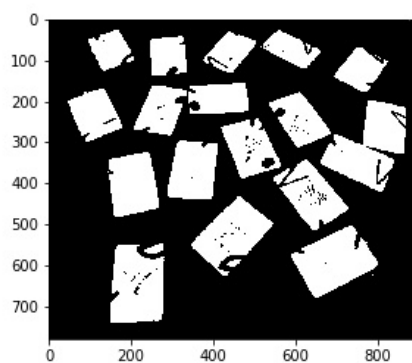
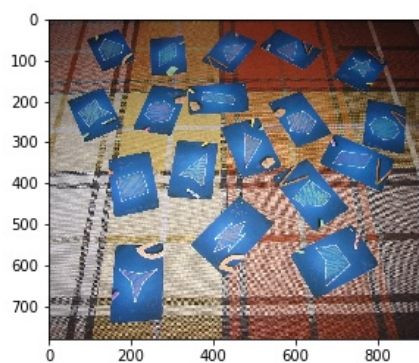


По гистограмме можно оценить, какое значение будет в ячейке, отвечающей порогу бинаризации, полученной методом Оцу. В случае, если по гистограмме можно хорошо бинаризовать изображение, отношение максимального значения к значению в точке порога велико, а если всё разделяется плохо, значение мало. Правило было введено следующее: если на гистограмме красного фона введённое отношение больше 315, то бинаризация происходит по красному каналу, если меньше либо равно, то по каналу тона в HSV системе. Значение 315 не обязательно точно подбирать, ведь на изображениях с пёстрым фоном это отношение было примерно равно 70, а на изображениях с монотонным фоном - около 500. Ниже в таблице приведены значения данного параметра, посчитанного на имеющихся изображениях. Значение 315 было взято, как среднее арифметическое между опорными значениями, по сути была производится классификация методом опорных векторов.

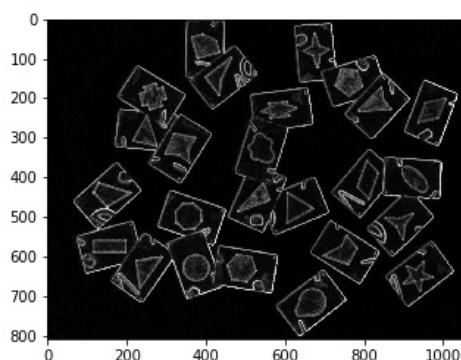
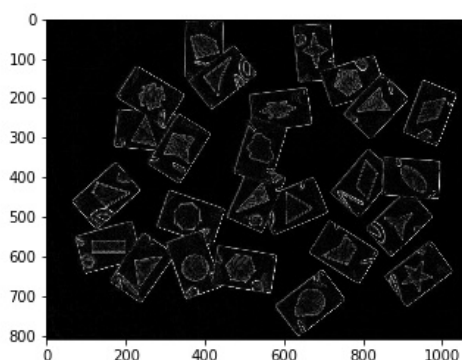
Номер изображения	1	2	3	4	5	6	7
Ratio	2425.24	676.66	637.71	553.96	1541.92	71.25	76.39
Фон	монот	монот	монот	монот	монот	пёстрый	пёстрый

Таблица 1: Отношение максимального значения гистограмме ко значению, полученному методом Оцу (Ratio).

После бинаризации полученные изображения обрабатывались одинаково вне зависимости от фона, который был на исходном изображении. В результате бинаризации получались изображения с шумом типа «соль и перец», а также с некоторыми дырками и вырезами, на месте которых на исходном изображении находились ластик или часть линейки. Пример приведён ниже.



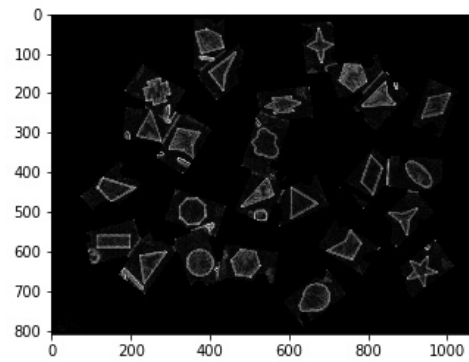
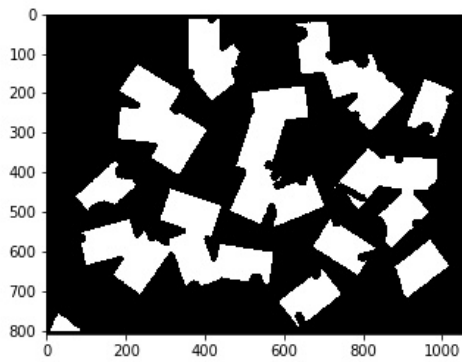
Была произведена заливка внутренних областей изображения. То есть, получена почти полная бинарная маска карточек. Далее производилась эрозия с диском диаметра 5 и удаление малых по площади связных компонент. Это было сделано для того, чтобы связанные компоненты, на месте которых в исходном изображении был ластик а также контуры границ карточек, исчезли. Отсутствие границ на бинарной маске было использовано в комбинации с фильтром собеля для выделения геометрических фигур. Дело в том, что сам по себе фильтр собеля выдаёт изображение на подобии представленного ниже. Справа ещё показан результат применения дилатации с размером ядра 1 к результату фильтра собеля. Исходное изображение, как можно заметить, с накладывающимися друг на друга карточками. Смысл дилатации был в том, чтоб связать пунктирные линии в одну непрерывную линию.



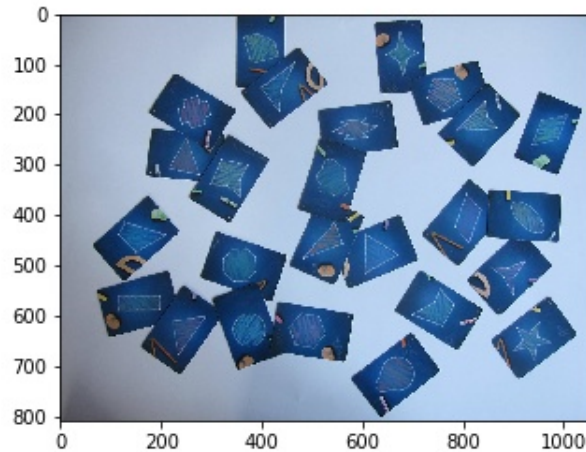
Все эксперименты по сегментации карточек по их бинарным картам не приводили к хорошему результату и тогда было принято решение выделять бинарную карту геометрических фигур. Геометрические фигуры не накладываются друг на друга, а также их число и местоположение совпадает с числом и местоположением карточек.

Урезанная бинарная маска карточек была применена к карте границ. Ниже показана сама маска и результат её применения на примере изображения IMG\_4.

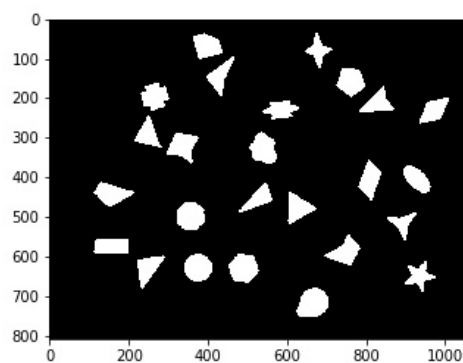
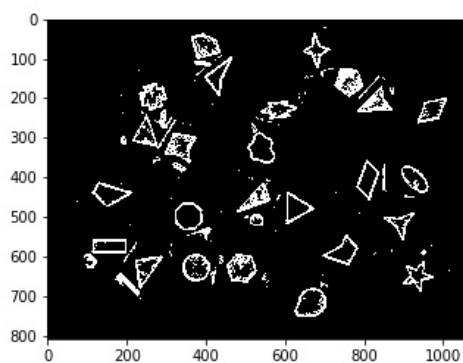




Можно заметить, что на маске в левом нижнем углу есть артефакт: карточки нет, но область выделена так, как будто там находится кусочек карточки. Это результат неравномерного освещения. Можно взглянуть на исходное изображение IMG\_4 и увидеть, что именно там находится область тёмного цвета. Однако после применения фильтра собеля эта область не отличается от остального фона. Именно так происходила борьба с неравномерной освещённостью.

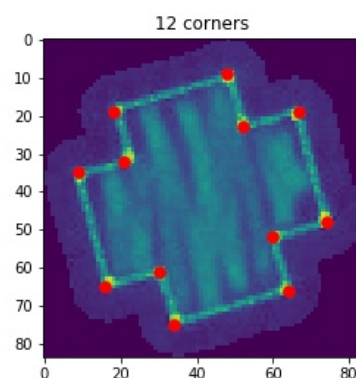
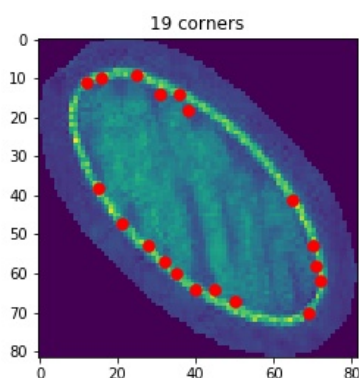


После применения маски к карте границ для полученного изображения снова методом Оцу искался порог бинаризации. Затем после серии операций дилатации, эрозии, заполнения внутренних областей и удаления малых по площади связных компонент удалось получить карту геометрических фигур, по которой уже можно было делать разметку. В цикле перебирались связные компоненты и на исходное изображение накладывались надписи с номерами карточек. Примеры полученных бинаризованного изображения, а также этого же изображения после применения указанных морфологических операций показан ниже. Как можно увидеть, на изображении слева осталась есть граница ластика, а на изображении справа её уже нет. Заметим, что ластик по площади меньше, чем остальные геометрические фигуры и за счёт подбора порога по площади (все связные компоненты площадью меньше, чем 900 пикселей убирались) удалось избавиться от ластика и оставить геометрические фигуры. Значение 900 было подобрано также экспериментальным путём: компромисс между 700 (лишние связные компоненты не исчезали) и примерно 1100 (минимальная площадь геометрической фигуры).



## Решение задачи определения геометрической фигуры

Так как бинарные карты геометрических фигур уже были выделены в предыдущей задаче, остаётся только для каждой связной компоненты ответить на вопрос, является ли она многоугольником, если да, то сколько у него углов и является ли он выпуклым. Для определения точек, которые являются углами использовался детектор Харриса, у которого был параметр: минимальное расстояние между соседними углами, который был выставлен равным 2. У детектора Харриса с небольшим значением данного параметра есть интересное свойство: в случае гладкой неровной границы он обозначает довольно много точек углами. То есть, можно по количеству выделенных углов на изображении можно судить о том, какой тип геометрической фигуры на нём изображён. В результате решающее правило было сформулировано следующим: если углов на изображении больше 12, считать, что там фигура с гладкой границей, в противном случае - многоугольник. Далее стояла задача определить, является ли изображённый многоугольник выпуклым. Для этого вычислялась площадь выпуклой оболочки и сравнивалась с площадью исходной фигуры. Если площадь выпуклой оболочки превосходила площадь фигуры минимум на 300 единиц, фигура считалась невыпуклым многоугольником. Значение 300 подбиралось экспериментально. Ниже приведён пример работы детектора Харриса на изображении эллипса, а также фигуры, у которой площадь выпуклой оболочки отличалась на минимальное количество единиц от площади исходной фигуры.



## Выводы

- Задача выделения объектов на изображении с использованием морфологических методов может быть хорошо решена только для узкого класса изображений при полном

знании специфики тестовой выборки. И даже при полном знании возможны погрешности.

- Для решения задачи выделения объектов с использованием методов морфологии, а также точечных преобразований приходится строить довольно сложную систему, которую необходимо к тому же долго тестировать.
- + С использованием априорных знаний о специфике выборки можно настроить сложную систему выделения и распознавания объектов или выделения хороших признаков на изображениях, которые в дальнейшем могут быть поданы на вход алгоритму машинного обучения.
- + Данные методы, основанные на морфологии, а также точечных преобразованиях при условии знания специфики тестовой выборки могут показывать хорошие результаты даже при малом числе размеченных примеров.