

Выделение графических примитивов для анализа структуры документа на примере локализации печатей

Сараев А.А.
НИТУ МИСИС
antonysaraev@gmail.com

Николаев Д.П.
ИППИ РАН
dimonstr@iitp.ru

Аннотация

В данной работе исследуются различные подходы к задаче локализации печатей и штампов круглой формы на изображении: обобщенное преобразование Хафа, алгоритм Виолы-Джонса, аппроксимация компонент связности краёв изображения фигурами искомой формы. Получены опытные данные по упомянутым методам, свидетельствующие об их пригодности для решения поставленной проблемы.

1. Введение

В современном мире широкое распространение получили системы распознавания и электронной обработки документов. Важной их частью являются алгоритмы выделения графических примитивов – элементов, неделимых с точки зрения прикладной программы. Зачастую даже знание о наличии на изображении определенного вида примитива может сыграть важную роль в задаче понимания документа, а точная локализация графических элементов может существенно улучшить качество распознавания, как самой структуры документов, так и его полей заполнения. Примерами таких примитивов являются линии разграфки, решетки знакомест, разнообразные штампы и печати. Именно последние были предметом анализа в данной работе.

Печати и штампы зачастую имеют схожий вид, но их значение на документе несколько отличается. Штамп применяется для быстрого нанесения на документ часто повторяющейся информации, например, названия фирмы, печати же, как правило, имеют юридическую силу.

Рассматривается широко распространенный подкласс печатей и штампов, а именно, круглые печати и штампы. Данный момент обуславливает применимость огромного количества материала, накопленного для задачи поиска окружностей на изображении.

2. Неформальная постановка задачи

Объектом интереса в данной работе являются круглые печати и штампы, которые можно рассматривать как сложные объекты, состоящие из смешанных текстовых и графических компонентов, заключенных в окружность. Подобное определение содержится, например, в [1].

Входным параметром системы является полутоновое чёрно-белое изображение (в случае цветного изображения выполняется его преобразование).

Результатом работы системы является список областей изображения, в которых обнаружен искомый объект.

3. Формальная постановка задачи

В качестве входного аргумента выступает «серое» изображение, представляющее собой двумерную матрицу известных размеров I , в узлах которой находятся значения яркости соответствующих пикселей с координатами x, y .

Под печатью мы понимаем графический объект, граница которого удовлетворяет уравнению окружности:

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \quad (1)$$

и, что эквивалентно:

$$a_{11}x^2 + a_{22}y^2 + 2a_{12}xy + 2a_{13}x + 2a_{23}y + a_{33} = 0 \quad (2)$$

где $a_{11} = 1, a_{22} = 1, a_{12} = 0$, причем, если обозначить круг, задаваемый этой окружностью, как C , то содержимое внутри этой границы отвечает условию:

$$\frac{\sum_{p(x,y) \in C} I(x,y)}{S_C} \ll \frac{\sum_{p(x,y) \in I} I(x,y)}{S_I} \quad (3)$$

$p(x,y)$ – точка с координатами (x, y) , S_C – площадь круга C , S_I – площадь изображения I .

Выходом системы являются координаты квадрата, описывающего искомую печать, с осями, параллельными границам изображения.

Печать считается найденной, если:

$$0,6 * S_T \leq S_{T \cap F} \leq 1,5 * S_T \quad (4)$$

где T – истинный квадрат охватывающий печать, F – найденный квадрат, а S_T и $S_{T \cap F}$ площади истинного квадрата и пересечения истинного и найденного.

Главными критериями оценки исследуемых методов являются точность:

$$\text{Precision} = \frac{|D_{rel} \cap D_{retr}|}{|D_{retr}|} \quad (5)$$

отношение числа печатей, обнаруженных на изображении, к общему числу найденных объектов, и полнота:

$$\text{Recall} = \frac{|D_{rel} \cap D_{retr}|}{|D_{rel}|} \quad (6)$$

отношение числа печатей, найденных системой, к общему числу печатей на изображении.

4. Описание алгоритмов и методов

Для определения методов, подходящих рассматриваемой задаче, естественно отталкиваться от следующих двух точек зрения. Первая, наиболее очевидная, – печать определяется окружностью. Следовательно, мы должны найти окружности на изображении, а после проверить их содержимое на предмет наличия графической информации. Именно таким способом построены два из трех рассматриваемых метода. В основе первого лежит обобщенное преобразование Хафа, второй построен на аппроксимации границ объектов изображения окружностью. Важным аспектом двух последних алгоритмов является выделение границ. В данной работе использовался детектор Канни.

4.1. Обобщенное преобразование Хафа

Для каждой точки на изображении существует конечное множество окружностей, которым она может принадлежать. Тогда, т.к. окружность задается тремя параметрами (координатами центра и радиусом), введя трехмерный массив (аккумулятор), можно провести процедуру голосования, где каждая точка голосует за все окружности, которым она может принадлежать, победителей которой можно объявить истинными окружностями [2]. Данная схема описывает первоначальную схему преобразования Хафа (с той оговоркой, что оно использовалось для поиска прямых [3], а не окружностей). Очевидно, что для изображений с размерностью порядка 2000x2000 такую процедуру без каких-либо ограничений провести невозможно – ни по памяти, ни по времени. Здесь следует заметить, что за те 40 лет, которые известно преобразование Хафа, оно стало целым семейством методов, специальные модификации которого имеются для очень многих фигур, и окружности не исключение [4][5]. В данной работе исследовалось преобразование Хафа, использующее в качестве дополнительной информации вектор градиента – мы увеличиваем значения аккумулятора в ячейках, лежащих вдоль его направления [2]. При этом также использовался

аккумулятор размерности три. С целью экономии памяти, он был сжат в 5 раз для каждой размерности (соответственно, одна ячейка аккумулятора описывает в пространстве Хафа куб с ребром 5). Победителями голосования считались ячейки не с наибольшим количеством голосов, т.к. это дискриминирует окружности с небольшим радиусом, а с наибольшим значением отношения $\text{count}(x,y,r) / 2\pi r$. При этом окружности, не содержащие внутри себя графической информации, отбраковывались.

4.2. Аппроксимация треков

На бинарном изображении, являющемся результатом работы детектора Канни, можно выделить группы пикселей, транзитивно соседствующих друг с другом по одному из 8-ми направлений. Такие множества будем называть треками. Т.к. на исходной картинке были печати, то естественно утверждать, что если некоторый трек похож на дугу окружности, вполне возможно, что он как-то связан с печатью. Вопрос лишь в том, как обнаружить это сходство. Задачу усугубляет то, что границы фигур могут пересекаться, в результате чего разные части одного трека могут принадлежать разным фигурам.

Последнее обстоятельство можно преодолеть, выделив на треках углы – т.к. мы ищем окружности (а на них углов нет), то можно смело объявить участки между углами отдельными треками. Углы можно обнаружить, двигаясь вдоль трека, измеряя при этом локальную кривизну в каждой точке. Если значение кривизны выше заданного порога, то данная точка является вершиной угла.

Теперь можно проанализировать данные точки на предмет соответствия окружности. Учитывая, что точки должны удовлетворять уравнению (2), для трека, состоящего из n точек, получаем систему уравнений:

$$\begin{bmatrix} 2x_1 & 2y_1 & 1 \\ \dots & \dots & \dots \\ 2x_n & 2y_n & 1 \end{bmatrix} \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} = \begin{bmatrix} -x_1^2 - y_1^2 \\ \dots \\ -x_n^2 - y_n^2 \end{bmatrix} \quad (7)$$

Решая ее, мы получаем параметры окружности с наименьшей среднеквадратической ошибкой аппроксимирующей данный трек. Очевидно, что если ошибка мала, то в зависимости от количества точек на треке это либо окружность, либо овал. Данный подход используется в частности, в [6].

После этого выполняется поиск окружностей – объекты с близкими параметрами объединяются, коэффициенты пересчитываются с учетом увеличения количества точек в уравнении (7). Во-первых, бракуются слишком маленькие окружности. Объекты, суммарная длина треков которых составляет более 50% соответствующей окружности, помечаются как истинные окружности. В противном случае анализируются точки, где должна находиться окружность. Алгоритм должен находить окружности, точки которых сильно разорваны, поэтому вводится следующая характеристика:

$$q = \frac{\sum l_i^2 - \sum d_i^2}{n} \quad (8)$$

где l_i – количество точек на непрерывном (в смысле 8-ми связности) отрезке дуги, i – номер дуги. Соответственно d_i – количество точек на неразрывном отрезке дуги инвертированной окружности. Если данная характеристика незначительно меньше нуля, мы можем предположить, что окружность действительно существует. Т.к. печать, как правило, дает целое множество центрированных окружностей, необходимо заменить все окружности с общим центром одной, наибольшего радиуса. В виду (3) бракуются все объекты, внутри которых количество точек границ меньше заданного порога.

4.3. Алгоритм Виолы-Джонса

Искать печати можно не только посредством анализа контуров. В настоящее время широкое распространение получили методы машинного обучения, самостоятельно определяющие свои параметры на основе заданной обучающей выборки, в нашем случае – размеченного множества изображений. Одним из таких методов является алгоритм Виолы-Джонса, появившийся в 2001 году [7]. Он быстро получил широкое признание в качестве детектора человеческих лиц, а позже был адаптирован и для поиска улыбок и других объектов.

В классической реализации алгоритма используются хааровские прямоугольные признаки.

Хааровы признаки дают высокий отклик на перепады интенсивности. На человеческом лице примером такого скачка является область глаз. Алгоритм в ходе обучения строит классификатор, состоящий из некоторого множества таких признаков, где каждому элементу соответствует некоторый коэффициент, и использует его для детектирования. Следует отметить, что использование хааровских признаков отнюдь не является обязательным, важнее, чтобы эти признаки можно было быстро вычислять, и они соответствовали специфике задачи.

В случае хааровских признаков вычислительная скорость достигается за счет использования интегрированного представления изображения. Для некоторой точки (x, y) , значение интегрированного изображения в этой точке рассчитывается как сумма всех пикселей, находящихся выше и левее данной точки:

$$I(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} i(x', y') \quad (9)$$

Данное представление позволяет быстро рассчитать суммарную интенсивность произвольного прямоугольника на изображении.

Сканирующее окно 24×24 содержит 45 396 возможных хааровских признаков. Вычислять их все слишком накладно. Для составления оптимального классификатора используется AdaBoost – алгоритм машинного обучения, строящий на основе T слабых классификаторов h_t , $t \in \{1, \dots, T\}$ один сильный

классификатор. Каждому из этих классификаторов присваивается вес α_t .

После обучения полученный сильный классификатор имеет вид:

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right) \quad (10)$$

На следующем этапе полученный классификатор организуется в так называемый каскад. То есть первая группа классификаторов обладает самыми высокими показателями детекции искомого объекта и имеет небольшой размер. Она сразу же может отсеять большое количество изображений, на которых нет объекта интереса. Изображения, успешно прошедшие первый каскад, поступают на второй каскад, где процедура повторяется. Соответственно мы получим выигрыш в скорости, потому что все значения слабых классификаторов h_t не будут рассчитываться для изображений, на которых нет объекта интереса [8].

5. Результаты

В ходе проделанной работы были реализованы обобщенное преобразование Хафа и алгоритм аппроксимации треков. За основу при исследовании метода Виолы-Джонса была принята реализация OpenCV. Реализованы программы подсчета статистики, размечена база изображений. Методы были протестированы на двух стендах изображений с помощью процессора Intel Core i5-2450M. В таблицах ниже приведены основные показатели по этим множествам и их объединению. В последних 3-х колонках результаты для объектов, радиус которых более 200. В данном случае это все печати, за исключением штампа с надписью печать (см. Рис.1 А, В). Двойное срабатывание – событие, при котором метод опознает одну печать как несколько разных объектов.

Анкета застрахованного лица Форма АДВ-1

Заполняется застрахованным лицом печатными буквами.
ВНИМАНИЕ! Прошью заполнить ширинкой или перьевой ручкой СИМВОЛЫ или
ЧЕРНЫМИ ЧЕРНИЛАМИ ЗАГЛУБЛЕННЫМИ БУКВАМИ по следующему образцу:
А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф Х
Ц Ч Ш Щ Ъ Ы Ь Э Ю Я 1 2 3 4 5 6 7 8 9 0

Фамилия: [печатные буквы]
Имя: [печатные буквы]
Отчество: [печатные буквы]

Пол: ☒ М ☐ Ж ☐ не определен
Дата рождения: 19 ЯНВАРЯ 1933
Место рождения: МОСКВА
район: [печатные буквы]
область (край, ...): [печатные буквы]
страна: [печатные буквы]

Адрес по прописке: [печатные буквы]

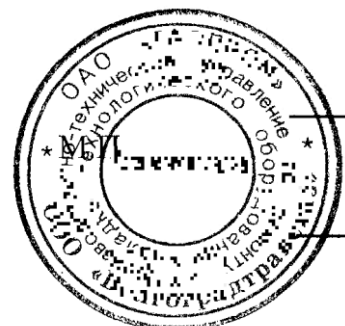
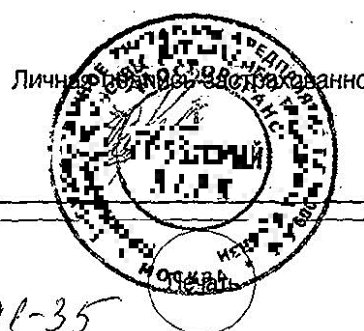
Адрес места жительства фактический (заполнить при отъезде от адреса по прописке):
Индент: [печатные буквы] Адрес: [печатные буквы]

Телефоны: [печатные буквы]

Документ: ПАСПОРТ
вид документа: [печатные буквы]
серия и номер: [печатные буквы]
дата выдачи: [печатные буквы]
кем выдан: [печатные буквы]

Дата заполнения: 10 АПРЕЛЯ 1997
Личная подпись застрахованного лица: [подпись]

Заполняется работодателем (для анкеты, заверенной работодателем):
Подпись руководителя: [подпись]



A

B

C

Рис. 1. (A) – пример изображения из стенда 1. (B) – искомые объекты из стенда 1. (C) – искомый объект из стенда 2.

Таблица 1. Результаты работы обобщенного преобразования Хафа

	Все изображения	Стенд 1	Стенд 2	Все изображения (только печати)	Стенд 1 (только печати)	Стенд 2 (только печати)
Количество изображений	1247	585	662	1247	585	662
Общее количество печатей	1821	1159	662	1236	574	662
Найденно печатей	1394	935	459	900	441	587
False positive	1811	1150	661	147	31	116
False negative	427	224	203	336	133	203
Повторное детектирование	38	26	12	36	24	12
Общее количество задетектированных объектов	3243	2111	1132	1083	496	587
Точность	0,43	0,44	0,41	0,83	0,89	0,78
Полнота	0,77	0,81	0,70	0,73	0,77	0,70
Общее время работы	8584 с.					
Среднее время работы для одного изображения	6,88 с.					

Таблица 2. Результаты работы алгоритма аппроксимации треков

	Все изображения	Стенд 1	Стенд 2	Все изображения (только печати)	Стенд 1 (только печати)	Стенд 2 (только печати)
Количество изображений	1247	585	662	1247	585	662
Общее количество печатей	1821	1159	662	1236	574	662
Найденно печатей	1638	1033	605	1159	554	605
False positive	671	440	231	93	34	59
False negative	183	126	57	77	20	57
Повторное детектирование	26	18	8	26	18	8
Общее количество задетектированных объектов	2335	1491	844	1278	606	672
Точность	0,70	0,69	0,72	0,91	0,91	0,90
Полнота	0,90	0,89	0,91	0,94	0,97	0,91
Общее время работы	1447 с.					
Среднее время работы для одного изображения	1,16 с.					

Таблица 3. Результаты работы алгоритма Виолы-Джонса

	Все изображения	Стенд 1	Стенд 2	Все изображения (только печати)	Стенд 1 (только печати)	Стенд 2 (только печати)
Количество изображений	1247	585	662	1247	585	662
Общее количество печатей	1821	1159	662	1236	574	662
Найденно печатей	1022	487	535	830	518	535
False positive	857	637	220	373	222	220
False negative	799	672	127	406	279	127
Повторное детектирование	1	1	0	1	1	0
Общее количество задетектированных объектов	1880	1124	755	1204	518	755
Точность	0,54	0,43	0,71	0,69	0,57	0,71
Полнота	0,56	0,42	0,81	0,67	0,51	0,81
Общее время работы	319 с.					
Среднее время работы для одного изображения	0.256 с.					



Рис. 2. Результат работы программы (аппроксимация треков)

6. Обсуждение

Оценку полученных результатов логично было бы проводить на основе задачи распознавания документа. Данная операция у программы ABBYY FineReader 11 занимает для одного изображения до 25 секунд. Учитывая, что локализация печатей является не единственным и не самым сложным этапом (в сравнении, например, с распознаванием текста), то время анализа печатей не должно превышать на имеющемся оборудовании двух секунд – данному критерию соответствуют алгоритмы аппроксимации треков и Виолы-Джонса.

Наилучшие результаты были показаны методом на основе аппроксимации треков. Особенно это касается работы с маленькими объектами. Немаловажной особенностью данного метода является строгое математическое обоснование, позволяющее по каждому задетектированному объекту получить достоверную оценку в виде среднеквадратической ошибки.

Плохие показатели были продемонстрированы обобщенным преобразованием Хафа. Особенно плохо в плане точности он обрабатывал небольшие печати, что связано с тем, что область текста дает отклик, сравнимый с маленьким штампом. Огромной проблемой преобразования Хафа является время обработки им изображения. Это наталкивает на мысль, что имеет смысл сжимать картинку. Тем более что с уменьшением количества пикселей на изображении резко возрастает актуальность и других методов данного семейства.

Слабый результат продемонстрировал и алгоритм Виолы-Джонса. Правда, есть основания полагать, что его результаты можно резко улучшить за счет грамотного обучения. Ведь объекты в рассматриваемых стендах обладают ясными отличительными признаками: большие печати есть не что иное, как круглая клякса, рядом с которой обычно, по большому счету, ничего нет. Маленькие же штампы могут быть опознаны по слову «Печать» в круглой рамке.

Все методы показывают лучшие результаты на больших печатях. В случае Виолы-Джонса все, скорее всего, упирается в неправильно проводимое обучение. Для остальных же методов данный результат обусловлен тем, что мелкие детали дают отклик, похожий на анализируемый рассматриваемыми методами. Например, рассматривая аппроксимацию, мы можем заметить, что короткие прямые могут быть распознаны как дуги с большим радиусом кривизны. Ситуация усугубляется тем, что маленький штамп находится, как правило, рядом с большой печатью, что искажает его границы.

Важным вопросом является соотношение множеств распознанных печатей для разных методов, т.к. это открывает возможности к комбинации методов. На данный момент предпочтительным является совместное использование Виолы-Джонса вместе с аппроксимацией треков, т.к. обобщенное преобразование Хафа будет сильно замедлять любой из них.

Ссылки

- [1] Partha Pratim Roy, Umapada Pal, Josep Lladós, *Document Seal Detection using GHT and Character Proximity Graphs*, Computer Vision Centre, Universitat Autònoma de Barcelona, Spain, 2010.
- [2] Sumit Tandon, *Hough Transform*, presentation by de Department of Electrical Engineering University of Texas at Arlington.
- [3] Gonzalez, Rafael C.; Richard E. Woods. *Digital Image Processing (2nd ed.)*.
- [4] Ali Ajdari Rad, Karim Faez, Navid Qaragozlou, *Fast Circle Detection Using Gradient Pair Vectors*, Tehran, Iran.
- [5] Shin-Hsuan Chiu, Jiun-Jian Liaw, Kuo-Hung Lin, *A Fast Randomized Hough Transform For Circle/Circular Arc Recognition*, Taiwan.
- [6] A. Levashov, D. Yurin, *Accurate and Reliable Framework for Fast Parametric Curves Detection*, Lobachevsky State University of Nizhny Novgorod, 2011.
- [7] Viola, Jones, *Robust Real-time Object Detection*, IJCV 2001.
- [8] Gary Bradski, Adrian Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly, 2008.