

Использование графа пересечений в задаче обнаружения документа на изображении, полученном со смартфона¹

Аннотация. Статья посвящена анализу задачи обнаружения документа на изображении и оценке качества работы мобильных приложений. Основываясь на данном анализе, в статье предлагается новый алгоритм получения изображения документа, состоящий из обнаружения отрезков границ документа и построения графа пересечений, удовлетворяющего проективной модели прямоугольника. Оценка качества работы алгоритма показала, что алгоритм способен к обнаружению 95% документов и превосходит все рассмотренные алгоритмы, реализованные в мобильных приложениях.

Ключевые слова: обнаружение документа, выделение отрезков, проективное преобразование, мобильные камеры.

Введение

В современном мире наблюдается серьезная тенденция к использованию мобильных устройств для обнаружения, распознавания и обмена документами [1-3]. Рассматривать подобные устройства в качестве полноценной платформы для решения прикладных задач позволил существенный рост возможностей современных смартфонов, как в плане вычислительных мощностей, так и в плане разрешения и качества фото- и видеосъемки. И с точки зрения пользователя, для того, чтобы получить изображение документа, намного быстрее и удобнее использовать смартфон, нежели стационарный сканер.

В научном сообществе задача обнаружения документа на изображениях, полученных с камеры, только начинает привлекать должное внимание. На данный момент нет стандартных наборов данных для оценки качества работы алгоритмов, и только недавно состоялся первый в мире конкурс по разработке подобных

алгоритмов [4]. С другой стороны, уже существует огромное количество мобильных приложений, реализующих подобные алгоритмы с той или иной точностью работы. Основной целью при обнаружении документа является получение изображения документа, и, возможно, улучшение его визуальных характеристик (Рис. 1). При подобной постановке задачи сразу заметно различие с задачами распознавания документа или обнаружения текста [5-7], которые нацелены на получение именно текстовой информации из изображения документа. Задача обнаружения документа не требует присутствия текста и оперирует только различиями между документом и фоном. В дополнение к этому, обнаружение документа можно рассматривать как первый шаг методов обнаружения текста или распознавания документа.

Статья построена следующим образом. В части 1 описана методика оценки качества алгоритмов обнаружения документов, приведены оценка качества работы ряда мобильных приложений и анализ ее результатов. В части 2 представлен новый алгоритм обнаружения документа,

¹ Работа выполнена при поддержке РФФИ (гранты 15-07-06520 А и 16-07-1051 А).



Рис. 1. Процесс обнаружения документа на изображении, полученном со смартфона и процесс его улучшения

основанный на проведенном анализе. В части 3 представлены экспериментальные результаты работы алгоритма на объемном наборе изображений. В части 4 приведено обсуждение результатов, ошибок алгоритма и планы дальнейшей работы.

1. Оценка качества работы алгоритмов, реализованных в мобильных приложениях

Для оценки качества работы мобильных приложений использовался небольшой набор изображений из 25 фотографий, полученных со смартфона². Были выбраны 12 наиболее популярных приложений из Apple App Store, реализующие функции захвата документа, и вручную замерено качество работы каждого приложения.

Будем считать, что документ на изображении найден, если найдены все четыре его угла. В свою очередь, угол документа будем считать найденным, если его найденное положение угла документа p_i^{found} отдалено от положения, размеченного человеком p_i^{ideal} , на расстояние, меньшее заданного порога T :

$$N_{corners} = \sum_{i=1}^4 \delta(\|p_i^{found} - p_i^{ideal}\|, T) \quad (1)$$

$$\delta(x, y) = \begin{cases} 1, & \text{если } x \leq y \\ 0, & \text{иначе} \end{cases} \quad (2)$$

В наших экспериментах мы принимаем пороговое расстояние $T=40$ пикселей из расчета, что оценка производится на изображении раз-

мера 3264x2448 (полученное с восьмимегapixelной камеры). Для подсчета статистики по набору изображений мы будем использовать количество найденных документов и найденных углов. Для мобильных приложений с взаимодействием с пользователем первая величина показывает «насколько часто нужно вручную исправлять положение документа», в то время как вторая показывает «насколько сложно доведение обнаруженного положения документа до требуемого». Также, совместное использование двух величин позволяет разделить случаи, когда алгоритм не способен к обнаружению документа, от случаев, когда он слегка ошибается.

Результаты измерения приведены на Рис. 2. При этом лучший показанный результат, который показывает обнаружение 16 из 25 документов (64%), позволяет сделать вывод, что качество обнаружения документа рассмотренными мобильными приложениями, намного меньше качества, приемлемого при промышленном использовании.

Анализ ошибок приложений позволил составить список основных сложностей, которые необходимо учитывать при разработке алгоритма обнаружения документа:

- границы документа могут не быть прямыми, они могут иметь изломы и разрывы;
- границы могут иметь слабый контраст;
- документ может быть частично неотличим от фона, на котором он расположен;
- условия освещения не контролируются, освещение может быть недостаточным и неравномерным;

² Набор изображений доступен онлайн: <https://bitbucket.org/zhukovsky/camera-based-documents-dataset>

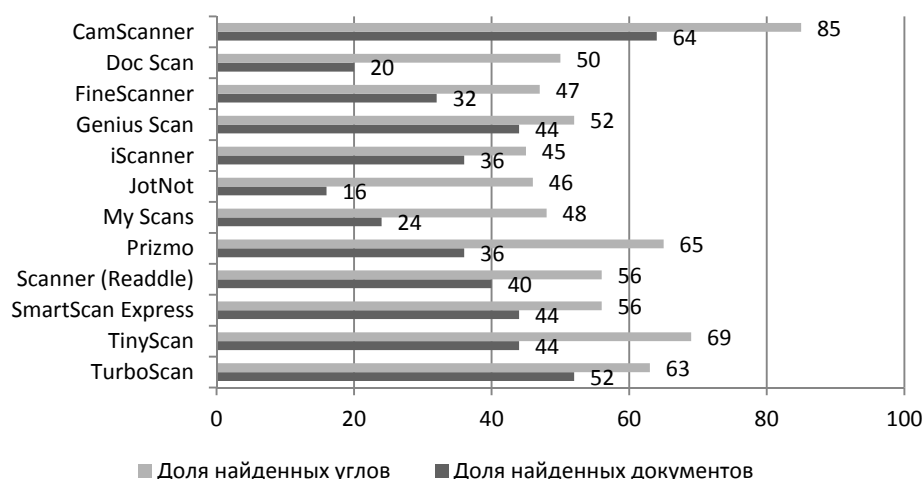


Рис. 2. Оценка качества работы алгоритмов, реализованных в мобильных приложениях

- документ может иметь сложную структуру (например, иметь рамку вдоль границ);
- фон также может иметь сложную структуру (например, съемка на шахматной доске);
- документ может частично выходить за рамки кадра.

2. Метод обнаружения документа на основе графа пересечений

Предлагаемый метод проектировался для работы с описанными сложностями и без использования дополнительной информации о положении документа (в отличие, например, от [8], где граница документа должна находиться в фиксированном месте изображения). Алгоритм основывается на обнаружении границ документа. Вначале производится первичная обработка изображения, состоящая из масштабирования (к размеру 400x300) и морфологического закрытия (для удаления мелких элементов, таких как символы). Далее, на изображении производится поиск отрезков двумя разными алгоритмами – для увеличения общей надежности алгоритма. После этого по полученным отрезкам строится четырехдольный ориентированный граф пересечений возможных сторон, по которому вычисляются возможные положения документа. Каждому возможному положению назначается свой вес, относительно которого они упорядочиваются. Для того, чтобы получить финальное решение о положении документа, проверяется соответствие проективной модели прямоугольного документа, и выбирается положение, удовлетворяющее этой

модели и имеющее максимальный вес. В конце работы границы документа уточняются на изображении в оригинальном масштабе.

Далее будут более подробно описаны все части алгоритма и процесс настройки параметров алгоритма.

2.1. Обнаружение отрезков

После первичной обработки изображения производится поиск отрезков прямых, которые могут являться сторонами документа. На бинарном изображении краев (полученном аналогично с [8]) выделяются восьмисвязные контура. Отрезком будем считать множество последовательных точек контура, в окрестности каждой из которых выполняется гипотеза о прямолинейности контура. Для проверки гипотезы будем для каждой точки (x_k, y_k) контура $\{(x_i, y_i)\}_{i=0}^N$ оценивать ковариационную матрицу точек в окрестности фиксированного размера l :

$$X_k = \{x_i\}, Y_k = \{y_i\}, i = \dots \min(N, k + l); \quad (3)$$

$$\text{cov}(X_k, Y_k) = M[X_k Y_k] - M X_k * M Y_k; \quad (4)$$

$$\Sigma(X_k, Y_k) = \begin{pmatrix} D X_k & \text{cov}(X_k, Y_k) \\ \text{cov}(X_k, Y_k) & D Y_k \end{pmatrix}. \quad (5)$$

Собственные векторы и значения этой матрицы позволяют оценить размеры и форму облака распределения участка контура, аппроксимировав его эллипсом [9]. Окончательно будем принимать гипотезу о прямолинейности контура в данной окрестности, если отношения



Рис.3. Процесс обработки изображения

Слева направо: масштабированное изображение; отфильтрованное изображение; полутоновое изображение границ; бинарное изображение контуров

длин осей данного эллипса превышают некоторое пороговое значение.

Так как использование бинарного изображения может привести к потере некоторых частей истинной границы (особенно в слабоконтрастных областях) для обнаружения отрезков дополнительно используется преобразование Хафа полутонового изображения границ [10-12]. В качестве полутонового изображения границ используется изображение, соответствующее поканальному максимуму модуля градиента. Отрезки, полученные в результате преобразования Хафа и контурного анализа, объединяются в один набор и, что важно отметить, параметры алгоритмов настраиваются для обеспечения наилучшей совместной производительности, а не для каждого по отдельности. Примеры всех используемых представлений изображения приведены на Рис. 3.

Далее, каждому полученному отрезку ставится в соответствие вес, равный его длине и лежащие на одной прямой отрезки объединяются. Два отрезка мы будем считать лежащими на одной прямой, если максимальное из расстояний от конца одного отрезка до прямой, содержащей другой отрезок, менее заданного порогового значения. В качестве концов отрезка, возникающего после объединения, выбираются наиболее удаленные друг от друга концы объединяемых отрезков. Вес объединенного отрезка выбирается как его длина, кроме случая, когда сумма длин объединяемых отрезков меньше, чем длина объединенного отрезка (т.е. имеется разрыв между ближними концами), тогда вес выбирается как сумма весов объединяемых отрезков. В конце работы выбирается

фиксированное число отрезков, наибольших по весу, которые и используются для построения графа пересечений.

2.2. Построение графа пересечений

Для того чтобы построить четырехугольник по найденным отрезкам, используется граф возможных сторон. Вершины данного графа соответствуют найденным отрезкам, а ребра - точкам пересечения прямых, которые содержат эти отрезки.

Для построения графа все отрезки делятся на преимущественно вертикальные и преимущественно горизонтальные, в зависимости от углов, которые они составляют с осями координат изображения. Если отрезок может быть отнесен к обоим классам (диагональный), то вместо него добавляются два отрезка: горизонтальный и вертикальный (например, на Рис.4 отрезку D слева соответствуют две вершины DH и DV по центру). Также каждой вершине присваивается вес w_i^{vertex} , равный весу соответствующего отрезка.

Ребра графа строятся как пересечения всех горизонтальных отрезков со всеми вертикальными. Если принять, что граф строится на N отрезках, то всего граф содержит $O(N^2)$ ребер. Каждое ребро классифицируется по типу угла документа, которым может быть соответствующая ему точка пересечения (левый верхний LT, правый верхний TR, правый нижний RB, левый нижний BL). Классификация происходит относительно положения точки пересечения относительно отрезков, которые ее породили. Когда точка пересечения является точкой пере-

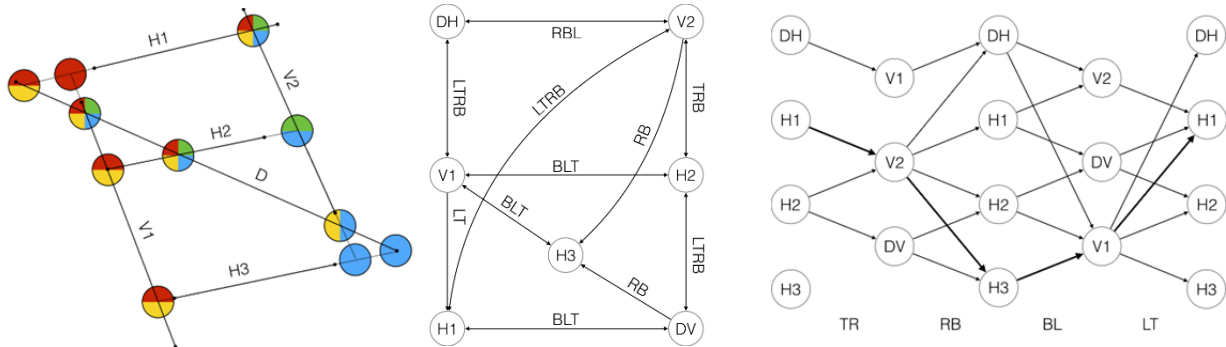


Рис.4. Пример использования графа пересечений по шести отрезкам

Слева: построение точек пересечения обнаруженных отрезков и назначение им типов

По центру: граф пересечений, каждое ребро соответствует одной точке пересечения

Справа: развернутый граф пересечений, каждое ребро соответствует возможной стороне четырехугольника

сечения самих отрезков, а не только прямых, на которых они лежат, она может соответствовать произвольному углу (LTRB), когда она лежит, например, левее горизонтального отрезка и на вертикальном отрезке, она может быть левой верхней или левой нижней (BLT). Каждому ребру присваивается вес w_i^{edge} , равный минимальному расстоянию между концами отрезков, которые образовали соответствующую точку пересечения.

Построенный таким образом граф можно раскрыть в четырехдольный направленный граф, так чтобы каждое ребро имело определенный и единственный тип (Рис. 4, справа). В таком графе всякий цикл, состоящий из четырех ребер и не проходящий через одноименные вершины, будет соответствовать возможному четырехугольнику на изображении. Обнаружение циклов можно организовать, например, при помощи радиальных обходов с максимальной глубиной обхода равной четырем, которые начинаются в одной из долей графа и могут многократно заходить в одну вершину. Сложность такого обхода составит $O(N^4)$, что накладывает дополнительные ограничения на количество исходных отрезков, описанные в конце пункта 2.1.

Каждому четырехугольнику назначается вес, равный разности сумм весов всех ребер и сумм весов всех вершин:

$$w^{quad} = A * \sum_{i=1}^4 w_i^{vertex} + B * \sum_{i=1}^4 w_i^{edge}, \quad (6)$$

где А и В – заданные параметры.

В конце работы выбирается необходимое количество циклов в графе, наибольших по ве-

су, и по каждому из них строится четырехугольник возможного документа.

2.3. Проективная модель прямоугольника

Однако при построении четырехугольника при помощи графа пересечений не принимается во внимание то, что он является проекцией прямоугольного документа на плоскость изображения (причем, возможно, даже с известным соотношением сторон). Предполагая же, что документ изначально имеет прямоугольную форму, становится возможным определить, может ли предъявленный четырехугольник быть его проекцией. Для этого рассмотрим центрально-симметричную модель построения изображения [13] – четырехгранный угол с вершиной О в центре линзы камеры, и сечением А'В'С'Д', образованным плоскостью изображения и пересекающим направляющие в углах найденного четырехугольника (Рис.5). Известно, что любой четырехгранный угол имеет только одно (с точностью до параллельного переноса) сечение ABCD, имеющее вид параллелограмма [14]. Точки такого сечения (при $OD=OD'$) можно вычислить следующим образом:

$$\overrightarrow{OA} = \overrightarrow{OA'} \cdot (\vec{b} \wedge \vec{c} + \vec{b} \wedge \vec{d} + \vec{c} \wedge \vec{d}) / k; \quad (7)$$

$$\overrightarrow{OB} = \overrightarrow{OB'} \cdot (\vec{a} \wedge \vec{c} - \vec{a} \wedge \vec{d} + \vec{c} \wedge \vec{d}) / k; \quad (8)$$

$$\overrightarrow{OC} = \overrightarrow{OC'} \cdot (\vec{a} \wedge \vec{b} - \vec{a} \wedge \vec{d} + \vec{b} \wedge \vec{d}) / k; \quad (9)$$

$$k = \vec{a} \wedge \vec{b} - \vec{a} \wedge \vec{c} + \vec{b} \wedge \vec{c}, \quad (10)$$

где $\wedge$ – операция косога произведения векторов 15; $\vec{a}, \vec{b}, \vec{c}, \vec{d}$ – двумерные векторы

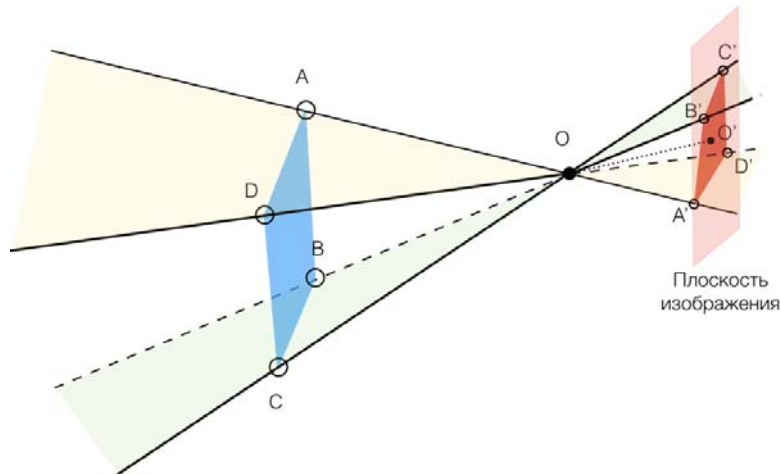


Рис. 5. Центральнo-симметричная модель построения изображения

$\overrightarrow{O'A'}, \overrightarrow{O'B'}, \overrightarrow{O'C'}, \overrightarrow{O'D'}$ в плоскости изображения; O' – проекция O на плоскость изображения. Ограничение интервалов углов и отношений сторон полученных параллелограммов позволяет изъять из рассмотрения четырехугольники, не удовлетворяющие данному условию.

2.4. Уточнение положения границ на изображении в оригинальном масштабе

Так как поиск отрезков производился на изображениях уменьшенного размера для ускорения алгоритмов обработки изображений, неизбежно возникают неточности в определении точного положения границ документа на изначальном изображении. Поэтому на окончательном этапе алгоритма мы уточняем точное положение границ. Для этого узкая окрестность каждой стороны уже найденного четырехугольника интерпретируется как регион, содержащий настоящую границу. Каждый такой регион вырезается из исходного изображения и в нем производится поиск прямых при помощи преобразования Хафа на изображении, полученном методом, описанным в части 2.1. В качестве уточненной границы выбирается прямая с максимальным аккумулятором в пространстве Хафа.

2.5. Настройка параметров

Каждый из представленных ранее базовых алгоритмов обладает рядом параметров, существенно влияющих на качество работы системы целиком. Общее количество параметров алгоритмов достигает 50. Правильная настройка данных параметров на большом наборе данных

является столь же ценным этапом разработки алгоритма, как и выбор базовых составляющих. При этом ручной подбор 50 параметров на большом наборе данных представляет собой довольно продолжительный и щепетильный процесс, который, к тому же не обязательно приведет к оптимальному решению.

Для оптимизации параметров был использован алгоритм MADS [16], реализованный в пакете NOMAD [17]. Данный алгоритм позволяет решать задачу нелинейной оптимизации произвольной функции (blackbox optimization). При этом принимается во внимание, что вычисление оптимизируемой функции в каждой точке является затратным процессом, что невозможно использовать такие свойства, как, например, производная и что сама функция может быть не определена на достаточно большом количестве точек и быть сильно зашумлена.

Как было описано в части 2.2, для каждого изображения алгоритм строит все возможные четырехугольники, из которых потом выбирает только заданное количество четырехугольников, удовлетворяющих проективной модели. Для того чтобы оценить количество обнаруженных углов по K четырехугольникам, оценим количество найденных углов для каждого из четырехугольников в отдельности по формуле 1 и выберем максимальный результат:

$$N_{corners} = \max_{k=1..K} \sum_{i=1}^4 \delta(\|p_{i,k}^{found} - p_i^{ideal}\|, T). \quad (11)$$

Данная оценка показывает, есть ли правильное решение среди K первых альтернатив или же, если оценка производится среди всех

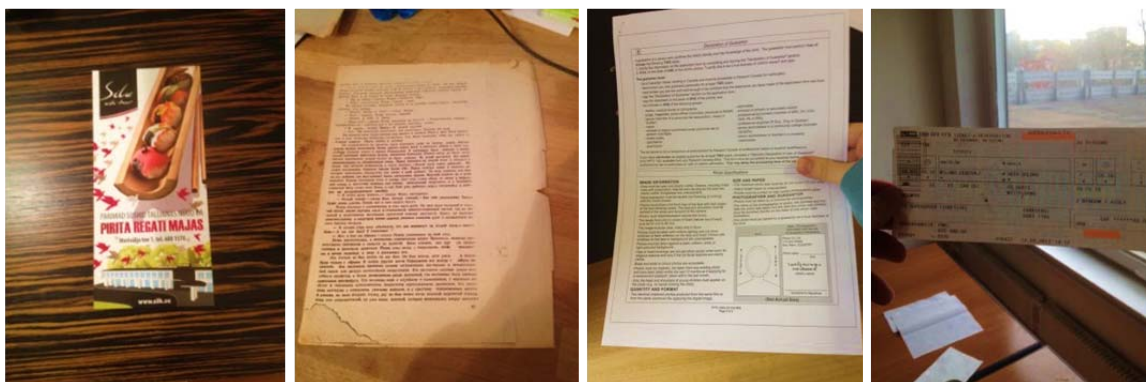


Рис. 6. Примеры изображений из используемого набора

найденных четырехугольников, способен ли алгоритм обнаружить правильное положение документа на данном изображении.

Для того, чтобы сократить сложность оптимизации, все параметры разбиваются на два набора: параметры обработки изображения и выделения отрезков (например, размер окна морфологического закрытия для предобработки изображения или минимальная длина рассматриваемого отрезка) и параметры получения четырехугольника по набору отрезков (например, A и B из формулы (6) или возможное отклонение от прямоугольника при проективной фильтрации), а процесс оптимизации – на два этапа оптимизации параметров соответствующего набора.

Целью первого этапа оптимизации является получение оптимальных параметров обработки изображения и выделения отрезков, чтобы в итоге работы первой части алгоритма получить набор отрезков, по которому можно построить наибольшее количество правильных четырехугольников на всем наборе изображений. Для этого в качестве оптимизируемого функционала используется оценка количества найденных углов, полученная по всем четырехугольникам (по формуле 11). При этом из алгоритма исключаются проективная фильтрация и уточнение границ. Таким образом, только параметры обработки изображения и выделения отрезков влияют на оптимизируемый функционал, по ним и производится оптимизация.

Получив после первого этапа оптимизации настроенные параметры первой части алгоритма, они фиксируются, и ведется оптимизация по оставшимся параметрам так, чтобы алгоритм наиболее часто в качестве лучшего четырех-

угольника по изображению выдавал бы правильный вариант. Здесь в качестве оптимизируемого функционала используется уже оценка количества найденных углов, полученная только по лучшему четырехугольнику (формула 1), и ничего из алгоритма не исключается.

3. Эксперимент

Для оценки качества работы системы использовался небольшой набор данных, который описывался в части 1 и также расширенный набор из 2757 изображений, полученных с разных смартфонов в разных условиях съемки. Каждое изображение содержит только один документ на произвольном фоне. Часть документа может выходить за рамки кадра, границы документа в большинстве случаев не являются прямыми, и иногда порваны или загнуты. Углы документа или стороны могут быть перекрыты рукой снимающего или чем-либо другим. Фон, отношение сторон документа, его содержание полностью произвольны и достаточно разнообразны. Примеры изображений из набора показаны на Рис. 6.

Методика оценки качества работы полностью совпадает с методикой, описанной в части 1. Кроме этого, в дополнение к оценке лучшего четырехугольника, приведено качество работы по нескольким лучшим альтернативам и по всем возможным к построению четырехугольникам. Получение нескольких возможных положений документа может быть полезно, например, в случаях, когда есть возможность обработать каждый из них и выбрать лучший результат. Все результаты оценки качества сведены в Табл. 1.

Табл. 1. Оценка качества алгоритма

Показатели	Кол-во найденных документов	Доля найденных документов	Кол-во найденных углов	Доля найденных углов
1 лучшее положение	2612	94.81%	10743	97.49%
2 лучших положений	2646	96.04%	10826	98.24%
4 лучших положений	2674	97.06%	10888	98.8%
8 лучших положений	2684	97.42%	10915	99.05%
16 лучших положений	2699	97.97%	10947	99.34%
Все возможные положения	2723	98.84%	10968	99.53%
Максимальное значение	2757	100%	11028	100%

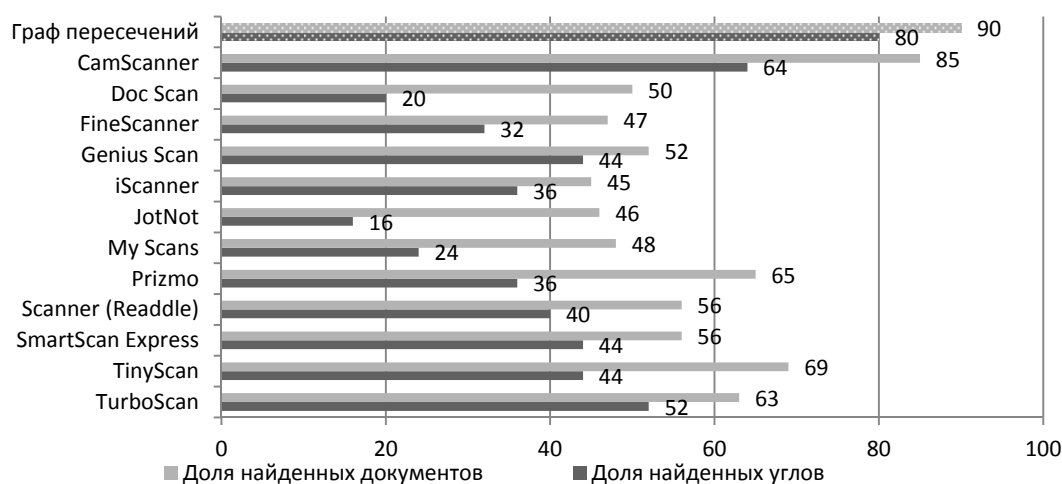


Рис. 7. Сравнение с алгоритмами, реализованными в мобильных приложениях

Для сравнения качества работы предложенного алгоритма с алгоритмами, реализованными в мобильных приложениях, также была проведена оценка работы по методике, используемой для построения Рис. 2. Результаты данного сравнения представлены на Рис. 7. Как видно, предложенный алгоритм превосходит по качеству работы все алгоритмы, реализованные в мобильных приложениях.

4. Обсуждение и планы

Представленный алгоритм показал хорошую надежность при работе с документами, стороны которых не являются строго прямыми. Однако подобная гибкость к выбору сторон иногда приводит к появлению неожиданных с человеческой точки зрения ошибок, что было выявлено при анализе ошибок алгоритма. Также в случаях, когда одна из сторон слабо выражена,

алгоритм склонен выбирать в качестве стороны более ярко выраженный отрезок, стороне не соответствующий. Данные недостатки можно решить, например, методами анализа распределения цвета вдоль сторон, внутри и вне документа. Наиболее сложными случаями для алгоритма также являются ситуации, когда граница документа практически отсутствует на изображении (например, в случае, когда один лист А4 лежит на нескольких других). Человек все же способен восстановить истинное положение документа, опираясь на заметные его углы, а не на стороны.

Дополнение представленной схемы детектором углов вкупе с проективной фильтрацией полученных четырехугольников может решить данную проблему. Кроме этого, представленный алгоритм заточен на работу на отдельном изображении. Однако при реальном применении может стоять задача обнаружения доку-

мента в видеопотоке, полученном с камеры мобильного устройства в режиме реального времени, например, для выбора оптимального момента захвата изображения. Использование подобного рода информации может быть полезно для выбора альтернатив из списка найденных четырехугольников (например, при помощи фильтра Калмана [18] или его модификации [19]) и предсказания положения документа, в случае, когда правильной альтернативы нет.

Заключение

В работе рассмотрена задача обнаружения плоского документа на изображении и представлен алгоритм ее решения. Была проведена оценка качества работы мобильных приложений, реализующих функции обнаружения документа, которая показала, что качество их работы недостаточно высоко. Совместное использование отрезков, обнаруженных при помощи контурного анализа и преобразования Хафа, их надлежащая обработка, использование графа пересечений для построения четырехугольника, геометрические ограничения исходя из проективных свойств, уточнение границ, и надлежащая настройка параметров позволили справиться со сложностями задачи обнаружения документа. Оценка качества работы алгоритма на наборе из 2757 фотографий показала, что предложенный алгоритм показывает хорошее качество работы как при применении в интерактивных мобильных приложениях (~95% обнаруженных документов по одному лучшему четырехугольнику), так и как часть алгоритма обработки или распознавания документа (~98% обнаруженных документов по 16 альтернативам). Также, алгоритм превосходит по качеству работы все рассмотренные алгоритмы, реализованные в мобильных приложениях.

Литература

1. Арлазаров В.В., Жуковский А.Е., Кривцов В.Е., Николаев Д.П., Полевой Д.В. Анализ особенностей использования стационарных и мобильных малоразмерных цифровых видео камер для распознавания документов, Информационные технологии и вычислительные системы, том 3, 2014.С. 71-81.
2. Liang J., Doermann D., Li H. Camera-based analysis of text and documents: a survey, *Int. J. of Document Analysis and Recognition*, Vol. 7, Issue 2, 2005. p. 84-104.
3. Doermann D., Liang J., Li H. Progress in Camera-Based Document Image Analysis, *IEEE Proc. 7th Int. Conf. on Document Analysis and Recognition*, Vol.1, 2003. p.606-616.
4. Burie JC., Chazalon J., Coustaty M., Eskenazi S., Luqman M.M., Mehri M., Nayef N., Ogier JM., Prum S., Rusinol M. ICDAR2015 Competition on Smartphone Document Capture and OCR (SmartDoc), *13th Int. Conf. on Document Analysis and Recognition*, 2015.
5. Chen F., Carter S., Denoue L., Kumar J. SmartDCap: Semi-Automatic Capture of Higher Quality Document Images from a Smartphone, *ACM Proc. of the 2013 Int. Conf. on Intelligent User Interfaces*, 2013. p.287-296.
6. Takeda K., Kise K., Iwamura M. Real-Time Document Image Retrieval on a Smartphone, *IEEE Proc. 10th Int. Workshop on Document Analysis Systems*, 2012. p. 225-229.
7. Nakai T., Kise K., Iwamura M. Camera-Based Document Image Retrieval as Voting for Partial Signatures of Projective Invariants, *IEEE Proc. 8th Int. Conf. on Document Analysis and Recognition*, Vol. 1, 2005. p.379-383.
8. Skoryukina N., Nikolaev D., Sheshkus A., Polevoy D. Real Time Rectangular Document Detection on Mobile Devices, *SPIE Proc. 9445, 7th Int. Conf. on Machine Vision* 2014.
9. Ширяев А. Н. Вероятность, 3-е изд., М.: МЦНМО. 2004.
10. Duda H. Use of the hough transformation to detect lines and curves in pictures, *Mag. Comm. ACM* 15.1972.
11. Nikolaev D., Karpenko S., Nikolaev I., Nikolayev P. Hough transform: underestimated tool in the computer vision field, *Proc. of the 22th European Conf. on Modelling and Simulation*, vol. 238.2008.
12. Ershov E., Terekhin A., Nikolaev D., Postnikov V., Karpenko S. Fast Hough transform analysis: pattern deviation from line segment, *8th Int. Conf. on Machine Vision*.2015.
13. Hartley R., Zisserman A. Multiple view geometry in computer vision, Cambridge University Press, New York.2003.
14. Шахно К.У. Сборник задач по элементарной математике повышенной трудности, М.: Рипол Классик.2013.
15. Прасолов В. В. Задачи по планиметрии, 4-е изд., М.: МЦНМО.2001.
16. Audet C., Dennis J. E. Mesh Adaptive Direct Search Algorithms for Constrained Optimization, *SIAM J. Optim.* 17, 2006. p.188-217.
17. Digabel S. Le. Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm, *ACM Trans. Math. Softw.* 37, 4.2011.
18. Kalman R. E. A New Approach to Linear Filtering and Prediction Problems, *J. of Basic Engineering* 82, 35.1960.
19. Karpenko S., Konovalenko I., Miller A., Miller B., Nikolaev D. UAV Control on the Basis of 3D Landmark Bearing-Only Observations, *Sensors*, vol. 15, 12.2015.

Арлазаров Владимир Викторович. Заведующий лабораторией ИСА ФИЦ ИУ РАН. Преподаватель кафедры когнитивных технологий МФТИ. Окончил МИСиС в 1999 году. Кандидат технических наук. Автор 16 печатных работ. Область научных интересов: распознавание образов, обработка изображений, системы массового обслуживания.
E-mail: vva777@gmail.com

Жуковский Александр Евгеньевич. Аспирант МФТИ. Преподаватель кафедры алгоритмов и технологий программирования МФТИ. Окончил МФТИ в 2013 году. Автор четырех печатных работ. Область научных интересов: компьютерное зрение, распознавание образов, машинное обучение. E-mail: zhukovsky@phystech.edu

Кривцов Валерий Евгеньевич. Декан МФТИ. Окончил МФТИ в 1971 году. Кандидат физико-математических наук. Автор более 95 печатных работ, в том числе трех монографий. Область научных интересов: информационные технологии и компьютерные науки, математическое программирование, математические модели и методы в экономике.
E-mail: krivtsov@phystech.edu

Постников Василий Валерьевич. Заведующий лабораторией ИСА ФИЦ ИУ РАН. Преподаватель кафедры когнитивных технологий МФТИ. Окончил МФТИ в 1990 году. Кандидат физико-математических наук. Автор более 40 печатных работ. Область научных интересов: компьютерное зрение. E-mail: vassili.postnikov@gmail.com