

## ОЦЕНКА СЛОЖНОСТИ АЛГОРИТМА РЕКУРСИВНОГО ПОИСКА ОБЛАСТИ ИЗОБРАЖЕНИЯ

К.А. Кивва

k.kivva@gmail.com

И.В. Рудаков

irudakov@yandex.ru

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

---

### Аннотация

Приведены описание алгоритма рекурсивного поиска области изображения и оценка его вычислительной сложности, а также сравнение вычислительной сложности данного алгоритма с вычислительной сложностью алгоритма скользящего окна. Предложен критерий определения эффективности использования алгоритма рекурсивного поиска вместо алгоритма скользящего окна. Согласно предложенному критерию, обоснована эффективность использования рекурсивного алгоритма

### Ключевые слова

*Обработка изображения, компьютерное зрение, скользящее окно, локализация объекта*

Поступила в редакцию 09.06.2016  
© МГТУ им. Н.Э. Баумана, 2017

---

**Введение.** Компьютерное зрение — это быстро развивающаяся область искусственного интеллекта, находящая применение в различных сферах человеческой деятельности: от картографии [1] и контроля качества [2, 3] до создания беспилотных систем [4, 5] и дополненной реальности [6]. Одной из задач компьютерного зрения является задача локализация объекта на изображении. Несмотря на множество разработанных подходов к ее решению, большинство из них не универсальны и требуют значительных вычислительных затрат.

Одна из подзадач локализации объекта — разбиение изображения на области, потенциально содержащие искомым объект. Используемый во многих работах механизм локализации основан на алгоритме скользящего окна [7]. Данный алгоритм осуществляет полный перебор множества возможных областей изображения. Можно предположить, что рекурсивный подход к разбиению изображения на подобласти с исключением из рассмотрения части подокон на каждом шаге рекурсии позволит сократить число рассматриваемых областей и, соответственно, уменьшить вычислительную сложность локализации объекта. Данный алгоритм разбиения с точки зрения оценки вычислительной сложности фактически построен по принципу «разделяй и властвуй» [8], заключающемуся в разбиении исходной задачи на подзадачи меньшего размера с последующим объединением результатов.

Анализ сложности рекурсивных алгоритмов — одна из наиболее сложных и до конца не решенных проблем метрической теории алгоритмов [9]. Основными современными инструментами исследования сложности рекурсивных алгоритмов являются метод рекуррентных соотношений и теоретико-графовый метод исследования дерева рекурсии [9].

Работы, в которых бы проводилась оценка вычислительной сложности алгоритма скользящего окна или оценка вычислительной сложности алгоритма рекурсивного разбиения изображения на подобласти, авторам неизвестны. Однако хорошо известны методы оценки интеративных алгоритмов [8].

В настоящей статье оценивается сложность рекурсивного разбиения изображения на области по сравнению со сложностью алгоритма скользящего окна и предлагается критерий, позволяющий определить, какой из алгоритмов имеет меньшую вычислительную сложность в каждом конкретном случае.

Наиболее общий подход к решению задачи локализации объекта на изображении — скользящее окно — заключается в прохождении сканирующим окном (различного масштаба) по всей плоскости изображения и определении для каждого конкретного положения такого окна наличия в соответствующей области изображения искомого объекта [10]. Для проверки наличия объекта в рассматриваемой области используются решения, применяемые в задачах классификации изображений, такие как нейронные сети [11], гистограммы ориентированных градиентов [12, 13], детектор Виолы — Джонса [14] и др.

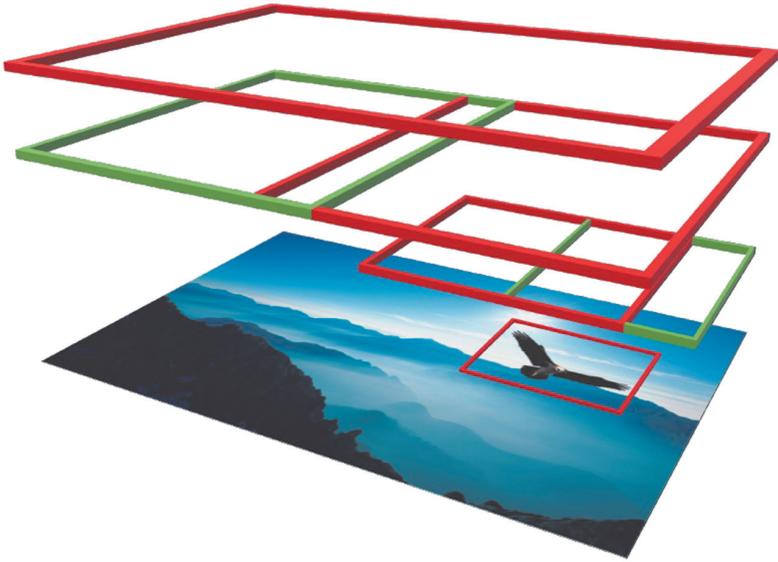
Локализация объекта с помощью скользящего окна требует значительных вычислительных затрат на полный перебор всех возможных позиций и масштабов такого окна, что затрудняет локализацию объектов в задачах, имеющих ограничение на время обработки изображения, таких как обработка кадров видеопотока при наличии ограниченных вычислительных мощностей (например, при использовании встраиваемых систем и мобильных платформ).

**Предлагаемый рекурсивный подход к локализации объекта на изображении.** За основу предлагаемого подхода взята идея скользящего окна: рассмотреть все такие прямоугольные области изображения различного размера, которые могут содержать искомые объекты. Чтобы уйти от полного перебора всех вариантов расположения и размеров окна, предлагается разбивать область изображения на подобласти уменьшающегося размера рекурсивно, на каждом шаге отбрасывая подобласти, не содержащие ни одного экземпляра искомого объекта. Вариант подобного подхода приведен на рис. 1.

Основной задачей, которую необходимо решить при таком подходе к перебору областей изображения, потенциально содержащих искомый объект, является задача определения самого факта наличия искомого объекта в произвольном месте каждой подобласти. Это позволит на каждом шаге разбиения отбрасывать области, заведомо не содержащие искомый объект.

Если на изображении искомый объект отсутствует вовсе, то в идеальном случае подобный подход позволит на первом же шаге разбиения, т. е. при рассмотрении самого изображения целиком, как начальной области, определить, что дальнейшего разбиения не требуется.

Остановить работу алгоритма разбиения (при наличии в рассматриваемой области искомого объекта) имеет смысл в двух случаях: если следующий шаг разбиения рассматриваемой области не выявил ни одной подобласти, содержащей искомый объект, либо если дальнейшее разбиение приведет к получению подобластей, размер которых недостаточен по условиям поиска объекта.



**Рис. 1.** Локализация объекта рекурсивным разбиением изображения на области

Таким образом, алгоритм найдет те подобласти изображения (если они есть), которые содержат искомый объект (или несколько объектов).

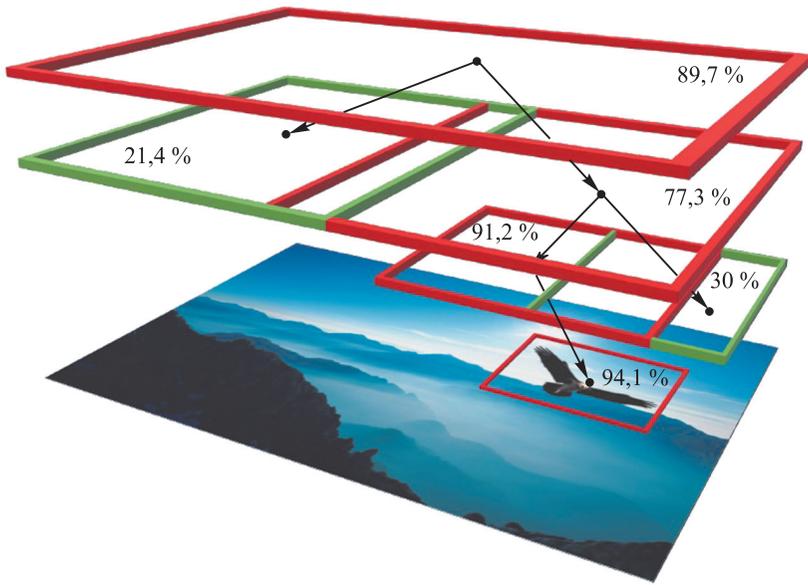
В случае, если определить факт наличия искомого объекта в произвольном месте некоей подобласти возможно лишь с некоторой вероятностью (способ определения вероятности зависит от реализации классификатора), то такое разбиение изображения может быть представлено в виде дерева принятия решений [15], как показано на рис. 2.

**Оценка вычислительной сложности алгоритма рекурсивного поиска области изображения, содержащей объект.** Примем исходные размеры окна, равными размерам самого изображения  $M \times N$  пикселей и, уменьшая на каждом из  $t$  шагов ширину и высоту окна в  $p$  и  $q$  раз, закончим рассмотрение (в худшем случае) окнами размерами  $1 \times 1$  пиксель. Следовательно, число шагов изменения размера окна  $t = \log_p M = \log_q N$ .

В худшем случае, во-первых, имеем шаги смещения подокна в окне по горизонтали и вертикали (обозначим их  $h_m$  и  $h_n$ ) равные 1 пиксель, а во-вторых, как нетрудно заметить, если на последнем шаге размеры окон будут равны  $1 \times 1$  пиксель, то на предпоследнем —  $p \times q$  пикселей, на шаге перед ним —  $p^2 \times q^2$  и т. д. Число положений окна  $(i+1)$ -го размера в окне размера  $i$ -го равно

$$c_i = (p^{t-i+1} - p^{t-i} + 1)(q^{t-i+1} - q^{t-i} + 1), \quad (1)$$

где  $i = \overline{1, t}$ , при  $i = 1$  имеем число подокон, рассматриваемых на первом шаге алгоритма. Полагаем, что  $i = 0$  соответствует одному окну размером со всю область изображения, которое не относится к данной формуле, поскольку такое окно не содержится ни в каком большем окне и, следовательно, говорить о количестве его положений в окне более крупного размера некорректно.



**Рис. 2.** Локализация объекта рекурсивным разбиением изображения с учетом вероятностей нахождения объекта в окне

Предположим, что в худшем случае на каждом шаге рекурсивного разбиения искомый объект обнаруживается алгоритмом не в одном окне, а в нескольких, и/или изображение содержит несколько искомых объектов. Тогда после каждого шага разбиения (кроме последнего) для дальнейшего рассмотрения выбираются несколько подокон. Объект будет считаться найденным, если его присутствие обнаружено в данном окне и выполняется одно из двух условий.

1. Окно, содержащее объект, имеет минимально допустимый размер. Для оценки сложности в худшем случае минимально допустимым размером окна выбран  $1 \times 1$  пиксель, что произойдет при  $i = t(1)$ . В реальных задачах этот размер в основном будет больше, поскольку один пиксель малоинформативен для анализа.

2. Размер данного окна больше минимально допустимого, но ни одно из его подокон не содержит искомый объект.

Таким образом, любая ветвь рекурсии остановится либо когда найдет объект в соответствии с приведенным критерием, либо когда ни одно подокно, предназначенное для рассмотрения на следующем этапе рекурсии, не будет содержать искомый объект.

Определим следующие величины [8]: вычислительные сложности случаев, в которых рекурсия останавливается (назовем такие случаи конечными); число подзадач, получаемых при разбиении задачи на каждом шаге; вычислительную сложность разбиения задачи на подзадачи; вычислительную сложность объединения результатов.

Вычислительная сложность конечных случаев полностью определяется сложностью анализа одного окна классификатором в целях определения наличия

в нем объекта. Поскольку цель настоящей статьи — сравнить сложности алгоритмов скользящего окна и рекурсивного разбиения изображения вне зависимости от используемых классификаторов, применяемых к каждому окну, то примем сложность конечных случаев равной  $O(1)$ .

Число подзадач, получаемых на  $i$ -м шаге рекурсии, соответствует числу окон, потенциально содержащих объект, и предназначенных, таким образом, к дальнейшему разбиению. Обозначим это число  $k_i$ .

Сложностью разбиения задачи на подзадачи можно считать сложность анализа всех подокон данного окна (для выбора потенциально содержащих искомым объект). Поскольку сложность анализа одного окна было решено принять равной  $O(1)$ , то сложность анализа всех окон, полученных на  $i$ -м шаге, считаем равной

$$D_i = O(c_i). \tag{2}$$

Сложность объединения результатов (параметров искомым окон) в худшем (с вычислительной точки зрения) случае равна  $O(r_i)$ , где  $r_i$  — число объектов, найденных на данном шаге.

Учитывая сказанное, определяем сложность алгоритма рекурсивного разбиения изображения на подобласти с отбрасыванием областей, заведомо не содержащих искомым объект, используя рекуррентное соотношение:

$$T(i) = \begin{cases} O(1), & \text{если } i = t; \\ k_i T(i+1) + O(c_i) + O(r_i), & \text{если } i < t. \end{cases} \tag{3}$$

Выбрав в качестве размера одиночной задачи на каждом шаге рекурсии площадь  $s$  текущего окна, исключив при этом величину  $O(r_i)$  из соотношения (3), как не влияющую на сложность алгоритма при увеличении размера анализируемого изображения, можно переписать это соотношение в виде

$$T(s) = \begin{cases} O(1), & \text{если } s = 1, i = t; \\ k_i T(s/d) + O(s), & \text{если } s > 1, i < t, \end{cases} \tag{4}$$

где

$$d = pq = s^{1/t}. \tag{5}$$

Коэффициенты  $k_i$  в выражении (4) на каждом шаге, вообще говоря, зависят, во-первых, от числа и размеров искомым объектов на изображении, а во-вторых, от выбора  $p, q$ , шагов  $h_n, h_m$  (см. формулу (1)). Дело в том, что при слишком большой площади перекрытия подокон друг другом возрастает вероятность попадания одного и того же объекта в несколько подокон одновременно. Также  $k_i$  зависят от способности классификатора успешно отличать окна, содержащие объект, от тех, которые его не содержат.

Разложив рекуррентное соотношение (4) в сумму, получим

$$T(s) = \prod_{i=1}^{t-1} k_i + \sum_{j=1}^{t-2} \left( \frac{s}{d^{t-j-1}} \prod_{i=1}^{t-j-1} k_i \right) + s. \tag{6}$$

С учетом формул (1), (5), после раскрытия произведения  $\prod_{i=1}^{t-j-1} k_i$  наиболее быстро растущим членом получившегося полинома оказывается:

$$\prod_{i=1}^{t-j-1} (pq)^{(t-i+1)(t-j-i)} = s^{\frac{(j-t)(t+1) - \frac{(j-t)(t-j+1)(j+2t+4)}{6}}{t}}.$$

При этом в произведении  $\prod_{i=1}^{t-1} k_i$  (6) наиболее быстро растущим членом полинома аналогично является

$$s^{1/3(t^2-1)}. \quad (7)$$

Тогда наиболее быстро растущим членом полинома (6) для  $t > 1$  будет или  $s^{1/3(t^2-1)}$ , или

$$\frac{s}{d^{t-j-1}} s^{\frac{(j-t)(t+1) - \frac{(j-t)(t-j+1)(j+2t+4)}{6}}{t}} = s^{\frac{1}{6t}(j^3+3j^2-j(3t^2+3t-8)+2t^3-2t+6)}. \quad (8)$$

В рамках рассматриваемой задачи  $j$  является целым числом. Однако для определения промежутков монотонности выражения (8) удобно рассмотреть его как функцию от вещественного аргумента  $j$ , после чего для определения конкретных значений выражения (8), которые могут быть получены при решении задач локализации объекта на изображении, брать только целые значения  $j$ .

Взяв частную производную от функции (8) по  $j$ , с учетом области допустимых значений  $j$  согласно условию задачи и формуле (6), нетрудно убедиться в том, что функция (8) от  $j$  в области допустимых значений  $j$  монотонно убывает.

В этом случае на рассматриваемом промежутке значений  $j$  функция (8) имеет наибольшее значение при наименее возможном  $j$ , т. е. при  $j = 1$ :

$$\left. \frac{1}{s^{6t}(j^3+3j^2-j(3t^2+3t-8)+2t^3-2t+6)} \right|_{j=1} = s^{\frac{1}{6}(2t^2-3t+\frac{18}{t}-5)}. \quad (9)$$

Видно, что значения выражения (7) больше, чем значения выражения (9), на всем промежутке допустимых значений  $t$ .

Подводя итог, можно заключить, что самым быстро растущим членом полинома (6) при  $t > 1$  является  $s^{1/3(t^2-1)}$ .

Таким образом, сложность  $T(s)$  рассматриваемого рекурсивного алгоритма разбиения изображения на подокна в худшем случае имеет порядок

$$O\left(S^{1/3(t^2-1)}\right), \quad (10)$$

т. е. является суперполиномиальной.

**Сравнение вычислительных сложностей рекурсивного алгоритма и алгоритма скользящего окна.** Нетрудно показать, что алгоритм скользящих окон имеет в худшем случае линейно-логарифмическую вычислительную сложность порядка  $O(S \log S)$  от площади изображения  $S$ .

Очевидно, что порядок вычислительной сложности (10) алгоритма рекурсивного поиска разбиением изображения на подобласти в худшем случае значительно выше порядка вычислительной сложности алгоритма скользящего окна.

Однако сложность рекурсивного алгоритма соответствует сложности, определенной в предыдущем разделе, в случае, если на каждом шаге рекурсии для дальнейшего рассмотрения выбирать все подокна. Кроме того, при оценке вычислительной сложности рекурсивного алгоритма в худшем случае было допущено, что окна значительно перекрывают друг друга, поскольку смещения окон было решено считать равными одному пикселю по горизонтали и одному пикселю по вертикали. Эти условия ведут к многократному повторному рассмотрению подокон, полученных на разных ветвях рекурсии.

Часто в задачах локализации объекта на изображении дело обстоит иначе. Сложность алгоритма скользящего окна можно уменьшить, лишь увеличивая смещения окна за один шаг и/или уменьшая число размеров рассматриваемых окон, что ведет к повышению вероятности пропуска окном искомого объекта. Алгоритм рекурсивного разбиения изображения помимо тех же возможностей уменьшения вычислительной сложности, применимых для скользящего окна, позволяет значительно сократить время вычисления на каждом шаге рекурсии (и всех последующих) за счет уменьшения числа выбираемых для дальнейшего рассмотрения подокон данного окна.

Использовать алгоритм предполагалось совместно с классификатором, способным давать ответ «Да» или «Нет» на вопрос, есть ли в данном подокне искомый объект, или оценить вероятность присутствия объекта в том или ином окне. Такой подход (при условии построения соответствующего классификатора) позволит на каждом шаге рекурсии отбросить большинство окон и значительно сократить вычислительные затраты для всех последующих шагов.

Очевидно, что рекурсивное разбиение изображения на подокна будет эффективно тогда, когда при прочих равных условиях (одинаковых  $p, q, t, s$ ) и рекурсивном обходе будет в сумме рассмотрено (за счет отбрасывания заведомо не содержащих объект окон) меньше окон, чем существует позиций окон всех  $t + 1$  размеров (включая исходное изображение) в алгоритме скользящего окна. Число всех положений скользящего окна для всех  $t + 1$  размеров в изображении  $M \times N$  может быть найдено по формуле

$$\sum_{i=0}^t (M - p^i + 1)(N - q^i + 1).$$

По-прежнему считаем, что сдвиг окон относительно друг друга равен одному пикселю по вертикали и/или по горизонтали. С другой стороны, число всех окон, рассматриваемых рекурсивно при произвольном выборе на  $i$ -м шаге рекурсии  $k_i$  подокон для рассмотрения на  $(i + 1)$ -м шаге (см. формулу (3)) равно

$$1 + (p^t - p^{t-1} + 1)(q^t - q^{t-1} + 1) + \sum_{i=1}^{t-1} k_i.$$

Таким образом, условие, при котором рекурсивный алгоритм будет эффективнее алгоритма скользящего окна, можно записать (учитывая, что  $M = p^t$ ,  $N = q^t$ ) в виде

$$\sum_{i=1}^{t-1} k_i < \sum_{i=0}^{t-2} (p^t - p^i + 1)(q^t - q^i + 1). \quad (11)$$

Хотя нахождение точного решения данного неравенства — задача крайне сложная из-за большого числа переменных, однако, во-первых, решения существуют (например, очевидно, к области решения относится  $k_i = 1$  для любого  $k_i$ , как было показано ранее), во-вторых, можно найти решения в некоторых частных случаях. Например, пусть в каждом окне на каждом шаге выбирается для дальнейшего рассмотрения одинаковое число подокон (т. е.  $k_1 = k$ ,  $k_2 = k^2$  и т. д., поскольку  $k_i$ , как указано, это суммарное число всех выбранных на  $i$ -м шаге подокон всех выбранных на предыдущем шаге окон). Тогда выражение (11), с учетом суммы геометрической прогрессии, принимает вид

$$\frac{k(1 - k^{t-1})}{(1 - k)} < \sum_{i=0}^{t-2} (p^t - p^i + 1)(q^t - q^i + 1). \quad (12)$$

Для иллюстрации данного условия приведен график (рис. 3), отражающий преимущество рекурсивного подхода перед скользящим окном для некоторых значений  $k$  при различных  $t$  в данном случае (т. е. при  $k_i = k^i$ ).

Видно, если в каждом рассматриваемом во время рекурсии окне выбирать одинаковое число  $k$  подокон, то существуют целые  $k$ , относящиеся к области решения неравенства (12), т. е. такие, при которых использование рекурсивного алгоритма позволит обработать меньше окон, чем в случае использования алгоритма скользящего окна. Как следует из приведенного графика, в некоторых случаях использование рекурсивного алгоритма вместо скользящего окна способно дать значительный прирост в скорости обработки изображения. Данное преимущество постепенно теряется при увеличении числа  $t$  шагов изменения размеров рассматриваемых окон. Это объясняется экспоненциальным ростом числа рассматриваемых рекурсивно окон при увеличении  $t$ , как следует из неравенства (12). В самом деле, каждое выбранное окно на начальных этапах рекурсии создает отдельную ветвь рекурсии, умножая число окон, выбираемых на последующих этапах.

Таким образом, очевидно, что выгоднее на более ранних этапах рекурсии исключать из рассмотрения как можно больше окон, не содержащих искомый объект.

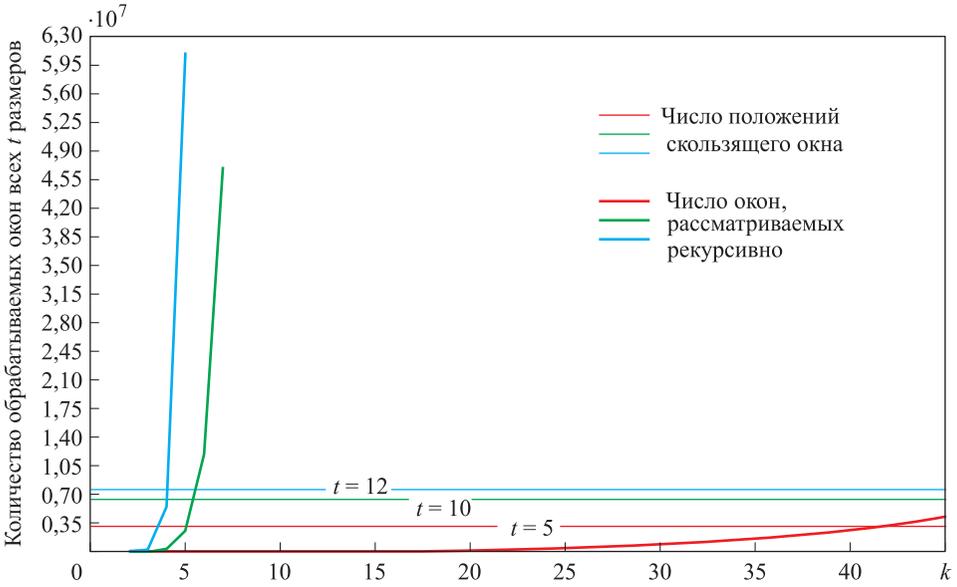


Рис. 3. Сравнение числа обрабатываемых окон для различных количеств  $t$  шагов изменения размеров окон для изображения размерами  $1024 \times 768$  пиксель

**Заключение.** Проведено сравнение двух алгоритмов разбиения изображения на пересекающиеся подобласти в целях поиска в нем некоторых объектов: алгоритма скользящего окна и предложенного в статье алгоритма рекурсивного разбиения изображения на подобласти. В ходе сравнения данных алгоритмов обнаружено, что в худших случаях алгоритм скользящего окна имеет меньший порядок сложности, чем алгоритм рекурсивного разбиения изображения. Однако в некоторых частных случаях использование алгоритма рекурсивного разбиения изображения может быть предпочтительнее, поскольку способно сократить число анализируемых окон по сравнению с алгоритмом скользящего окна за счет исключения из дальнейшего рассмотрения на каждом шаге рекурсии части окон, заведомо не содержащих искомым объект.

Также был выработан критерий (11), определяющий условия, при которых использование рекурсивного алгоритма предпочтительнее алгоритма скользящего окна.

В качестве развития настоящей работы предполагается сформировать классификатор, позволяющий отбрасывать на каждом шаге рекурсии часть окон, заведомо не содержащих объект, и разработать метод локализации объектов с использованием этого классификатора на основе алгоритма рекурсивного разбиения области изображения.

## ЛИТЕРАТУРА

1. Shet V. Street View and reCAPTCHA technology just got smarter // Google Security Blog: веб-сайт. URL: <https://security.googleblog.com/2014/04/street-view-and-recaptcha-technology.html> (дата обращения: 10.01.2017).

2. *Silveira J., Ferreira M.J., Santos C., Martins T.* Computer vision techniques applied to the quality control of ceramic plates // Proc. IEEE international Conference on Industrial Technology. Gippsland: ICIT, 2009.  
URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.973.1308&rep=rep1&type=pdf>  
(дата обращения: 15.01.2017).
3. *Kuzmanić I., Vujović I., Šoda J.* Damage detection in materials based on computer vision wavelet algorithm, advanced structured materials. Heidelberg, Germany: Springer Cham Heidelberg, 2014. P. 157–186.
4. *Obstacle avoidance system for UAVs using computer vision / Blin Richards, John Dayton, Miguel Enriquez, Mathew Gan, James Liu, Jordan Quintana* // Cal Poly Pomona Student RSCA Conference. Pomona: California State Polytechnic University, 2014.
5. *Guizzo E.* How Google's self-driving car works // IEEE Spectrum: веб-сайт.  
URL: <http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works> (дата обращения: 09.01.2017).
6. *Lepetit V.* On computer vision for augmented reality // International Symposium on Ubiquitous Virtual Reality. 2008. P. 13–16. DOI: 10.1109/ISUVR.2008.10  
URL: <http://ieeexplore.ieee.org/document/4568635>
7. *Forsyth D.A., Ponce J.* Computer vision: a modern approach. 2nd Edition. Upper Saddle River, New Jersey: Pearson Education, Inc., 2011. 793 с.
8. *Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К.* Алгоритмы: построение и анализ. М.: Вильямс, 2015. 1328 с.
9. *Быкова В.В.* Математические методы анализа рекурсивных алгоритмов // Журнал СФУ. Сер. Математика и физика. 2008. Т. 1. № 3. С. 236–246.  
URL: <http://elib.sfukras.ru/bitstream/handle/2311/772%20%20%20%20%20%20.pdf;jsessionid=B504A2DC25BD7FACCC6562B8810DFCF0?sequence=1>
10. *Murphy K., Torralba A., Eaton D., Freeman W.* Object detection and localization using local and global features // Toward category-level object recognition. Heidelberg, Germany: Springer Berlin Heidelberg, 2006. С. 382–400. DOI: 10.1007/11957959\_20  
URL: [http://link.springer.com/chapter/10.1007/11957959\\_20](http://link.springer.com/chapter/10.1007/11957959_20)
11. *Erhan D., Szegedy C., Toshev A., Anguelov D.* Scalable object detection using deep neural networks // IEEE Conference on Computer Vision and Pattern Recognition. 2014. P. 2155–2162. DOI: 10.1109/CVPR.2014.276 URL: <http://ieeexplore.ieee.org/document/6909673>
12. *Dalal N., Triggs B.* Histograms of oriented gradients for human detection // IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). 2005. Vol. 1. P. 886–893. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) DOI: 10.1109/CVPR.2005.177 URL: <http://ieeexplore.ieee.org/document/1467360>
13. *Corvee E.* Body parts detection for people tracking using trees of histogram of oriented gradient descriptors // 7th IEEE International Conference on Advanced Video and Signal Based Surveillance. 2010. P. 469–475. DOI: 10.1109/AVSS.2010.51  
URL: <http://ieeexplore.ieee.org/document/5597093>
14. *Paul V., Jones M.* Robust real-time face detection // International Journal of Computer Vision. 2004. Vol. 57. No. 2. P. 137–154. DOI: 10.1023/B:VISI.0000013087.49260.fb  
URL: <http://link.springer.com/article/10.1023/B%3AVISI.0000013087.49260.fb>
15. *Rokach L., Maimon O.* Classification trees, data mining and knowledge discovery handbook. New York: Springer US, 2010. P. 149–151.

**Кивва Кирилл Андреевич** — ассистент кафедры «Программное обеспечение ЭВМ и информационные технологии» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5).

**Рудаков Игорь Владимирович** — канд. техн. наук, заведующий кафедрой «Программное обеспечение ЭВМ и информационные технологии» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5).

**Просьба ссылаться на эту статью следующим образом:**

Кивва К.А., Рудаков И.В. Оценка сложности алгоритма рекурсивного поиска области изображения // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2017. № 2. С. 33–45. DOI: 10.18698/0236-3933-2017-2-33-45

## RECURSIVE IMAGE PART SEARCH ALGORITHM COMPLEXITY ESTIMATION

K.A. Kivva

k.kivva@gmail.com

I.V. Rudakov

irudakov@yandex.ru

**Bauman Moscow State Technical University, Moscow, Russian Federation**

---

### Abstract

Most common part for many approaches to object localization on image is a “sliding window” algorithm, which requires an exhaustive search in a wide range of window positions and size combinations. It can be assumed that a recursive approach of searching some image parts with a rejection of most areas at each step of the recursion could be more efficient in some cases. This article provides a recursive image part search algorithm description and an estimation of this algorithm’s complexity. It also contains a comparison of complexity between the recursive algorithm and the “sliding window” algorithm. According to the results, an efficiency identifying condition for using the recursive algorithm instead of the “sliding window” algorithm is provided. According to the proposed criterion we proved the effectiveness of using the recursive algorithm in some cases

---

### Keywords

*Image processing, computer vision, sliding window, object localization*

## REFERENCES

- [1] Shet V. Street View and reCAPTCHA technology just got smarter. Google Security Blog: website (in Russ.). Available at: <https://security.googleblog.com/2014/04/street-view-and-recaptcha-technology.html> (accessed 10.01.2017).
- [2] Silveira J., Ferreira M.J., Santos C., Martins T. Computer vision techniques applied to the quality control of ceramic plates. *Proc. IEEE Int. Conf. on Industrial Technology*. Gippoland, ICIT, 2009. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.973.1308&rep=rep1&type=pdf> (accessed 15.01.2017).

- [3] Kuzmanić I., Vujović I., Šoda J. Damage detection in materials based on computer vision wavelet algorithm, advanced structured materials. Heidelberg, Germany, Springer Cham Heidelberg, 2014, pp. 157–186.
- [4] Richards B., Dayton J., Enriquez M., Gan M., Liu J., Quintana J. Obstacle avoidance system for UAVs using computer vision. *Cal Poly Pomona Student RSCA Conference*. Pomona, California State Polytechnic University, 2014.
- [5] Guizzo E. How Google's self-driving car works. *IEEE Spectrum*: website. Available at: <http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works> (accessed 09.01.2017).
- [6] Lepetit V. On computer vision for augmented reality. *International Symposium on Ubiquitous Virtual Reality*. 2008, pp. 13–16. DOI: 10.1109/ISUVR.2008.10 Available at: <http://ieeexplore.ieee.org/document/4568635>
- [7] Forsyth D.A., Ponce J. Computer vision: a modern approach. 2nd Edition. Upper Saddle River, New Jersey, Pearson Education, Inc., 2011. 793 p.
- [8] Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to algorithms. Mit Press, 2001. 1180 p. (Russ. ed.: Алгоритмы: построение и анализ. Moscow, Vil'yams Publ., 2015. 1328 p.).
- [9] Bykova V.V. Mathematical methods for the analysis of recursive algorithms. *Zhurnal SFU. Ser. Matematika i fizika* [Journal of Siberian Federal University. Ser. Mathematics and Physics], 2008, vol. 1, no. 3, pp. 236–246 (in Russ.). Available at: <http://elib.sfukras.ru/bitstream/handle/2311/772/%20%20%20%20%20%20.pdf;jsessionid=B504A2DC25BD7FACCC6562B8810DFCF0?sequence=1>
- [10] Murphy K., Torralba A., Eaton D., Freeman W. Object detection and localization using local and global features. In: Toward category-level object recognition. Heidelberg, Germany, Springer Berlin Heidelberg, 2006, pp. 382–400. DOI: 10.1007/11957959\_20 Available at: [http://link.springer.com/chapter/10.1007/11957959\\_20](http://link.springer.com/chapter/10.1007/11957959_20)
- [11] Erhan D., Szegedy S., Toshev A., Anguelov D. Scalable object detection using deep neural networks. *IEEE Conf. on Computer Vision and Pattern Recognition*. 2014, pp. 2155–2162. DOI: 10.1109/CVPR.2014.276 Available at: <http://ieeexplore.ieee.org/document/6909673>
- [12] Dalal N., Triggs B. Histograms of oriented gradients for human detection. *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'05)*. 2005, vol. 1, pp. 886–893. DOI: 10.1109/CVPR.2005.177 Available at: <http://ieeexplore.ieee.org/document/1467360>
- [13] Corvee E. Body parts detection for people tracking using trees of histogram of oriented gradient descriptors. *7th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*. 2010, pp. 469–475. DOI: 10.1109/CVPR.2005.177 Available at: <http://ieeexplore.ieee.org/document/5597093>
- [14] Paul V., Jones M. Robust real-time face detection. *International Journal of Computer Vision*. 2004, vol. 57, no. 2, pp. 137–154. DOI: 10.1023/B:VISI.0000013087.49260.fb Available at: <http://link.springer.com/article/10.1023/B%3AVISI.0000013087.49260.fb>
- [15] Rokach L., Maimon O. Classification trees, data mining and knowledge discovery handbook. New York, Springer US, 2010, pp. 149–151.

**Kivva K.A.** — assistant of Computer Software and Information Technology Department, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, Moscow, 105005 Russian Federation).

**Rudakov I.V.** — Cand. Sc. (Eng.), Head of Computer Software and Information Technology Department, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, Moscow, 105005 Russian Federation).

**Please cite this article in English as:**

Kivva K.A., Rudakov I.V. Recursive Image Part Search Algorithm Complexity Estimation. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Priborostr.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2017, no. 2, pp. 33–45. DOI: 10.18698/0236-3933-2017-2-33-45



В Издательстве МГТУ им. Н.Э. Баумана  
вышло в свет учебное пособие  
под редакцией **А.В. Пролетарского**  
«**Технологии современных  
беспроводных сетей Wi-Fi**»

Изложены основные сведения о современных технологиях беспроводных сетей Wi-Fi и показано поэтапное проектирование беспроводных сетей – от планирования производительности и зоны действия до развертывания и тестирования сети. Подробно рассмотрен стандарт IEEE 802.11, включая управление доступом к среде, а также физический уровень 802.11. Описаны особенности радиочастотного спектра, принципы модуляции, приведены варианты спецификаций 802.11, технологии повышения производительности и механизмы защиты. Подробно рассмотрено подключение клиента к беспроводной сети в инфраструктурном режиме – сканирование, методы аутентификации и ассоциации, а также вопросы безопасности передачи данных в беспроводных сетях (WEP, TKIP, CCMP, WPA, WPA2, WPS). Приведены оценка беспроводной линии связи и пример расчета. Теоретические положения дополнены лабораторными работами по всем рассмотренным в книге темам.

**По вопросам приобретения обращайтесь:**  
105005, Москва, 2-я Бауманская ул., д. 5, стр. 1  
+7 (499) 263-60-45  
press@bmstu.ru  
www.baumanpress.ru