

ОБНАРУЖЕНИЕ И ЛОКАЛИЗАЦИЯ СТРОЕНИЙ ЗАДАННОЙ ФОРМЫ НА АЭРОСНИМКАХ В ИНФРАКРАСНОМ ДИАПАЗОНЕ

© 2017 А.В. Дунаева^{1,2}, Ф.А. Корнилов¹

¹ ФГБУН Институт математики и механики им. Н.Н. Красовского УрО РАН

(620990 Екатеринбург, ул. Софьи Ковалевской, д. 16),

² ФГБОУ ВО Уральский федеральный университет

им. первого Президента России Б.Н. Ельцина

(620002 Екатеринбург, ул. Мира, д. 19)

E-mail: maryanova.av@yandex.ru, kornilovfa@imm.uran.ru

Поступила в редакцию: 24.07.2017

В работе рассматривается задача обнаружения и локализации строений заданной формы на аэроизображениях земной поверхности в инфракрасном диапазоне с использованием аппарата контурного анализа. Приводится описание модификации алгоритма обобщенного преобразования Хафа для обнаружения контуров, заданных небольшим количеством параметров. Идея предлагаемого метода заключается в построении двумерных аккумуляторных массивов для каждого набора параметров фигуры в зависимости от ее положения, и их последующего объединения в результирующий аккумуляторный массив. Заполнение массивов осуществляется на основе значений модулей градиентов яркостей исходного изображения с учетом близости рассматриваемого контура к заданной форме. Близость формы определяется путем морфологического анализа контуров, найденных с помощью алгоритма Канни. Фильтрация обнаруженных объектов на основе плотности границ в их внутренней области, а также соотношения средних яркостей внутри и снаружи контура обеспечивает высокую чувствительность к заданным типам объектов и уменьшает количество ложных срабатываний алгоритма. Качество работы метода проверено на задаче локализации малоэтажных построек прямоугольной формы. Полученные результаты позволяют судить о применимости предложенного подхода для решения практических задач распознавания объектов местности.

Ключевые слова: обработка изображений, обнаружение объектов, контурный анализ, математическая морфология.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Дунаева А.В., Корнилов Ф.А. Обнаружение и локализация строений заданной формы на аэроизображениях в инфракрасном диапазоне // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 3. С. 84–100. DOI: 10.14529/cmse170306.

Введение

Визуальная информация является важнейшим источником информации об окружающем мире. Когда развитие вычислительной техники позволило создавать достаточно сложные системы, значительные усилия были направлены на решение задач обработки и анализа изображений. Последовавшее за этим усложнение как аппаратных, так и программных средств привело к тому, что объем входных данных стал чрезмерным для обработки и хранения, и встал вопрос о его уменьшении путем автоматизации процесса выделения наиболее значимых в контексте конкретной ситуации признаков, деталей, объектов и т.д. Некоторые из этих признаков носили физический характер (яркость, текстура) и могли быть верифицированы визуально. Другие представляли собой различные математические конструкции, операции обработки входного сигнала,

позволяющие выделить данные, недоступные для непосредственного наблюдения (спектры сигналов и др.) [1, 2]. Одним из способов уменьшения размерности стал контурный анализ.

Контур — это совокупность точек границы объекта, отделяющая его от фона. Будучи не связанным напрямую с яркостями пикселей изображения, контур является признаком, инвариантным как к изменениям яркости и контраста, так и к некоторым геометрическим искажениям: поворотам, сдвигам и изменениям масштаба [3, 4]. При этом предполагается, что контур содержит достаточно информации, необходимой для описания формы объекта. Ключевым фактором для эффективного применения данного предположения в различных задачах является отсутствие необходимости рассматривать внутренние точки объекта, что сразу переводит задачу из разряда двухмерных к значительно более простой одномерной постановке.

Контурный анализ является совокупностью методов выделения, описания, преобразования и распознавания контуров объектов на изображениях [5]. Под контуром с точки зрения поля значений яркости изображения понимается пространственно-протяженный разрыв (резкое изменение) значений яркости, который кодируется последовательностью, состоящей из комплексных чисел. Для этого выполняется обход контура (например, по часовой стрелке), начиная с некоторой зафиксированной начальной точки, и каждый вектор смещения относительно предыдущей точки записывается соответствующим комплексным числом. Такое кодирование, очевидно, инвариантно к сдвигам. Более того, имея два подобных комплекснозначных вектора, соответствующих двум контурам, можно вычислить для них нормированное скалярное произведение, которое будет тем ближе к единице, чем более похожи по форме исходные контура. Данное утверждение будет справедливо независимо от угла поворота контуров относительно друг друга. Однако, это свойство не сохраняется при рассмотрении действительных векторов координат пикселей контуров, что обуславливает необходимость перехода к комплекснозначному представлению.

Использование нормированного скалярного произведения позволяет эффективно сравнивать между собой контура (например, входной с контуром из базы данных). Однако его применение будет затруднено из-за проблематичности выделения контура на изображении. Во-первых, контур может быть виден лишь частично по причине перекрытия объектами сцены друг друга. Во-вторых, реальные изображения, с одной стороны, неизбежно содержат шум, а с другой — объекты на них зачастую незначительно отличаются по цвету и текстуре от фона, из-за чего становится крайне сложно выделить контура и произвести дальнейший анализ их формы.

В данной работе представлен подход, позволяющий выделить на изображении контура заданной формы, соответствующие определенным типам объектов. В качестве исходной задачи было выбрано дешифрирование видовой аэрокосмической съемки в инфракрасном диапазоне; в частности, рассматривается задача обнаружения строений прямоугольной формы. Изображения земной поверхности характеризуются изменчивостью отображения объектов сцены, которая, к тому же, отличается высокой структурной сложностью. В то же время, изображения инфракрасного диапазона обладают как слабой текстурированностью — что, в целом, является плюсом, поскольку значительно уменьшает объем не имеющей ценности для рассматриваемой задачи информации, так и слабой контрастностью — это, в свою очередь, затрудняет выделение перепадов яркости.

Статья организована следующим образом. В разделе 1 описаны стандартные подходы и методы выделения и анализа контуров изображений: предварительная обработка, обнаружение градиентов яркостей, прореживание контуров и определение их формы. В разделе 2 предложен алгоритм обнаружения строений заданной формы, основанный на модификации преобразования Хафа [6]. Раздел 3 посвящен деталям реализации алгоритма и результатам вычислительного эксперимента на реальных данных. В заключении подводятся итоги и намечаются направления дальнейшей работы.

1. Методы выделения и анализа границ объектов

Выделение перепадов яркости находит широкое применение в задачах обработки изображений. Помимо указанного во введении сравнения формы объектов для целей распознавания и классификации, это может быть сегментация — выделение границ между однородными по структуре областями, обнаружение изменений в составе объектов сцены и т.д. [7]. Благодаря такому многообразию прикладных задач было разработано большое количество различных подходов для выделения контуров объектов. Здесь мы в общих чертах представим лишь некоторые базовые из них.

Алгоритм контурного анализа состоит из следующих шагов:

1. Сглаживание входного изображения.
2. Поиск, прореживание и бинаризация модулей градиентов яркости.
3. Фильтрация и анализ найденных контуров.

Для решения задач контурного анализа необходимо предварительно сгладить исходное изображение. Рассматривая границу как перепад яркости между соседними пикселями, стоит учитывать, что, во-первых, по причине сложности сцены подобных границ будет множество, а во-вторых шум камеры, оптические и геометрические искажения приведут к тому, что практически никакие два соседних пикселя не будут иметь одинаковую яркость. Сглаживание изображения помогает заметно уменьшить количество таких перепадов. Однако, применение операции сглаживания неизбежно размывает и истинные границы. Для борьбы с этим недостатком разработан ряд методов, подавляющих шум с сохранением точности границ. Одним из классических является билатеральный фильтр [8], но в данной работе был использован более качественный *управляемый фильтр* (guided filter) [9]. Данный метод отличается простотой и гибкостью реализации, а также высоким качеством работы.

Сгладив исходное изображение, можно перейти к поиску перепадов яркости. Особенности формирования изображения на матрице камеры, а также шум, приводят к тому, что граница между объектом и фоном, как правило, представляет собой не «ступеньку», а плавное изменение яркости. Для обнаружения таких градиентов можно воспользоваться оператором Собела [10], который представляет собой дискретный дифференциальный оператор, основанный на свертке фрагментов изображения со специальными фильтрами, выделяющими перепады яркости в вертикальном и горизонтальном направлении. Для размера окна 3×3 пикселя маски фильтра выглядят следующим образом:

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * Image, \quad Sobel_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * Image,$$

где $Image$ — исходное полутоновое 8-битное изображение, символом $*$ обозначен оператор свертки. На основе $Sobel_x$ и $Sobel_y$ можно вычислить модуль вектора градиента и его направление:

$$Sobel = \sqrt{Sobel_x^2 + Sobel_y^2}, \quad \Theta = \arctg \frac{Sobel_y}{Sobel_x}. \quad (1)$$

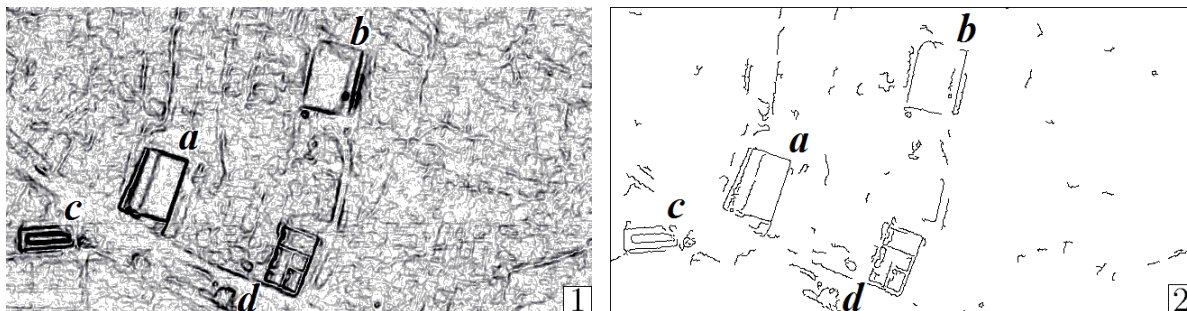


Рис. 1. Примеры работы оператора Собела (1) и алгоритма Канни (2). Для оператора Собела чем темнее точка, тем больше в ней значение модуля градиента. Найденные с помощью алгоритма Канни контура показаны черным цветом

Однако оператор Собела выдает не столько контура, сколько области градиентов: иными словами, широкие полосы (толщиной в несколько пикселей) вокруг истинной границы (рис. 1(1)). Для того, чтобы получить границу как линию толщиной в один пиксель, требуется применить алгоритм Канни [11], состоящий из следующих шагов.

1. *Сглаживание*: как правило, размытие фильтром Гаусса с маской 5×5 .
2. *Поиск градиентов*: с помощью оператора Собела, описанного выше.
3. *Подавление немаксимумов*: рассматривая известное направление градиента Θ (1), оставляем вдоль него лишь точку, имеющую максимальное значение градиента $Sobel$ (1). Благодаря этому шагу широкая полоса градиента, найденная с помощью оператора Собела, сжимается до линии, толщиной в один пиксель.
4. *Применение двух порогов*: задаются два пороговых значения, и точки, в которых значение градиента выше большего из них, считаются надежными границами, а те, что ниже меньшего отбрасываются. Точки, значение градиента в которых оказалось между двумя порогами, отбрасываются, если они не связаны (между ними нет пути, проходящего по точкам с ненулевым градиентом) с надежными границами.

Использование алгоритма Канни позволяет найти все существенные границы на изображении (рис. 1(2)). Однако, в конкретных приложениях, как правило, требуется обнаружить не все границы, а только принадлежащие определенным объектам, причем известно, что эти объекты характеризуются заранее известной формой. Преобразование Хафа [6] является одним из методов поиска кривых на бинарных изображениях. Параметры искомой кривой образуют пространство, в котором осуществляется поиск локальных максимумов. Заполнение пространства параметров (или аккумуляторного пространства) происходит с помощью процедуры голосования.

Первоначальный вариант преобразования Хафа разработан для поиска прямых линий. Если прямая задана уравнением $x \cos \phi + y \sin \phi = \rho$ с параметрами ρ и ϕ , то пространство параметров будет двумерным. На рис. 2 представлен пример, состоящий из четырех точек границ, выделенных на исходном изображении. Каждая точка голосует за набор прямых, которые потенциально могут пройти через нее с заданным шагом по углу (например,

четыре, если выбран шаг по углу, равный 45 градусов). Накапливая таким образом значения, получаем массив (аккумулятор), локальные максимумы которого соответствуют искомым прямым. На рис. 2 этот максимум будет соответствовать прямой, проходящей через все четыре точки. Аналогичным образом можно искать любые аналитически заданные кривые: например, окружности, для которых пространство параметров будет трехмерным.

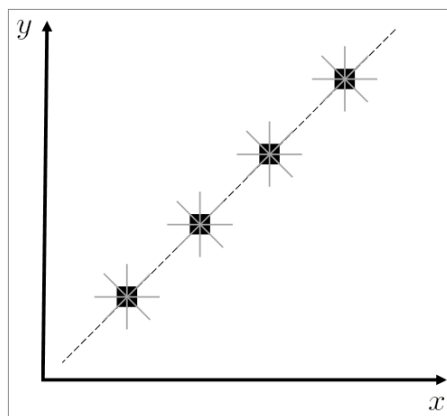


Рис. 2. Поиск прямых линий с помощью преобразования Хафа. Перебирая для каждого пикселя исходного изображения возможные прямые с шагом по углу 45 градусов, получим, что каждый пиксель голосует за четыре возможных прямых. Прямая, проходящая через все четыре точки, соберет четыре голоса, в то время как все остальные прямые соберут только по одному

Для случая, когда мы ищем не прямую, а некоторую кривую, либо описывающуюся слишком большим количеством параметров, либо чье аналитическое выражение неизвестно, были разработаны различные модификации: обобщенное [3] и рекуррентное [12] преобразования Хафа. Обобщенное преобразование Хафа использует принцип сравнения с шаблоном (эталонным образцом). Для этого осуществляется поиск положения заданной шаблоном искомой кривой на изображении: выбираются центр фигуры (например, центр тяжести) и множество точек на границе, после чего строится *просмотровая таблица*, в которой для каждой граничной точки записывается расстояние от выбранного центра и угол между вектором из центра до этой точки и вектором нормали к кривой. Таким образом осуществляется запоминание формы фигуры. При таком представлении каждая точка исходного изображения отождествляется с точками просмотрной таблицы и голосует за набор возможных центров фигуры.

Можно заключить, что преобразование Хафа по своей сути является алгоритмом перебора всех возможных положений искомой кривой для всех точек контуров исходного изображения и выбора наиболее часто встречающегося набора (или наборов) параметров этой кривой. Очевидно, что, во-первых, такой перебор для достаточно сложной по форме кривой будет чрезвычайно затратным по времени и ресурсам, так как будет требовать построения пространства параметров значительной размерности. Во-вторых, сами контура входного изображения отличаются сложностью и разнообразием: с одной стороны, их очень много (например, при поиске крыши дома придется неизбежно столкнуться с контурами лифтовых шахт, вентиляционных окон и других мелких объектов, загромождающих крышу и сбивающих алгоритм). С другой стороны, сам периметр крыши из-за геометрических и яркостных искажений скорее всего не будет иметь строго прямоугольную форму. Для

преодоления указанных ограничений в следующем пункте будет представлен алгоритм, основанный на преобразовании Хафа и решающий задачу обнаружения строений заданной формы.

Помимо указанных выше, существуют другие методы выделения и анализа контуров. Следует упомянуть детекторы угловых (характерных) точек: Харриса [13], FAST [14] и других. Однако, их применение к инфракрасным изображениям затруднено вследствие значительного размытия границ объектов. Также ряд работ посвящен поиску строений на последовательностях аэроизображений в видимом диапазоне, например [15].

2. Алгоритм локализации строений заданной формы

Изложенная в данной работе идея применима к поиску строений любой формы, лишь необходимо, чтобы эта форма была параметризуема некоторым небольшим количеством параметров. При этом не требуется наличия аналитического выражения: в общем случае искомый контур можно задать либо последовательностью точек с привязкой к некоторому общему центру (например, центру масс), аналогично просмотрным таблицам в обобщенном преобразовании Хафа, либо как комплекснозначный вектор смещений, как было упомянуто во введении.

Введем основные обозначения. Через $Image(x, y)$ обозначим исходное полутоновое изображение; $Sobel(x, y)$ — изображение модулей градиентов яркости $Image$ (1); $Canny(x, y) \in \{0, 1\}$ — изображение, содержащее контура $Image$, найденные с помощью алгоритма Канни, где значение 1 соответствует точкам контуров.

Работу алгоритма будем рассматривать на примере задачи локализации строений прямоугольной формы. Обозначим через $P = P_{w,h,\alpha}(x, y)$ — прямоугольник с центром в точке (x, y) , размером $w \times h$ пикселей, повернутый на угол α . Везде далее будем полагать, что $w \geq h$:

$$\begin{cases} w \in [l_{min}, l_{max}], \\ h \in [l_{min}, w], \\ \alpha \in [0, \pi], \end{cases} \quad (2)$$

с заданным шагом по каждому параметру. Параметры l_{min} и l_{max} определяются исходя из требований конкретной задачи, и при наличии априорной информации о размере искомых объектов можно значительно сократить перебор, тем самым уменьшив как время работы, так и количество ложных срабатываний алгоритма. Каждый прямоугольник разбивается на два множества: $P = \partial P \cup P^0$, где ∂P — граница прямоугольника, P^0 — внутренняя область, $\partial P \cap P^0 = \emptyset$, причем площадь $S(P) = S(\partial P) + S(P^0)$. Обозначение $S(\partial P)$ соответствует длине периметра P : поскольку мы работаем с изображениями, состоящими из дискретного набора пикселей, то длиной периметра будет считаться количество пикселей граничного слоя рассматриваемого объекта.

Поиск прямоугольных строений с использованием преобразования Хафа на основе найденных алгоритмом Канни контуров (аналогично прямым) приведет к неустойчивому обнаружению даже строго прямоугольных строений из-за выбора шага при переборе возможных размеров прямоугольников (2): например, если перебирать нечетные длины сторон, в то время как контур образует прямоугольник со сторонами четной длины. Поэтому в предлагаемом алгоритме будут использоваться не контура, а модули градиентов (1): во-первых, в них нет разрывов, а во-вторых, за счет того, что они представляют собой достаточно широкую линию, то через них всегда можно провести прямой отрезок,

являющийся стороной некоторого прямоугольника. Более того, каждая точка будет голосовать за выбранный прямоугольник значением модуля градиента в ней, что позволит выбирать только те контура, которые соответствуют наибольшему суммарному перепаду яркости вдоль периметра прямоугольника.

Общая идея применения преобразования Хафа для поиска прямоугольных контуров изложена в [16]: предлагается на основе значений градиентов яркости построить для каждого набора параметров (w, h, α) двумерный аккумуляторный массив, чей размер совпадает с размером исходного изображения, соответствующий точкам — центрам прямоугольников с выбранными параметрами. Далее строится результирующий аккумуляторный массив путем выбора в каждой точке максимального из значений построенных на предыдущем шаге аккумуляторов. Иными словами, мы рассматриваем каждую точку как потенциальный центр прямоугольника, перебираем все возможные прямоугольники, центром которых могла бы быть эта точка, и выбираем тот, который проходит через наибольшее (относительно длины периметра) количество граничных точек. Затем проводится пороговая обработка полученного результирующего аккумуляторного массива, выбираются локальные максимумы, и соответствующие им прямоугольники передаются на выход алгоритма. Необходимость выбора локальных максимумов обусловлена тем, что реальные снимки содержат шум и иные искажения, и даже один отдельно стоящий дом порождает не один контур, а целую группу, как правило, около десятка, сгруппированных вокруг его истинного центра.

Такой подход обладает очевидным недостатком: на реальных изображениях присутствует большое количество контуров, многие из которых могут образовывать прямоугольную структуру и приводить к появлению ложных срабатываний алгоритма: например, участки дорог, пространства между двумя близкорасположенными домами, лесные массивы и т. д. Поэтому требуется дополнить изложенную выше идею некоторыми условиями C_1, C_2, C_3 заполнения аккумуляторного массива, отсекающими контура, не соответствующие строениям, но тем не менее имеющими форму, близкую к прямоугольной.

Условие C_1 : форма. Данное условие учитывает близость формы рассматриваемых контуров к прямоугольнику. Используемые для построения аккумуляторного массива градиенты яркости позволяют не пропускать прямоугольники даже с размытыми границами. Однако, их использование порождает ложные тревоги, связанные с непрямоугольными объектами, которые, благодаря широкой полосе градиента, будут давать значительный вклад в аккумулятор. Поэтому необходим более строгий учет формы обнаруженного преобразованием Хафа прямоугольника $P_{w,h,\alpha}$, для чего будут использоваться контура, найденные алгоритмом Канни. Чтобы избежать пропуска отдельных точек контуров, которые относятся к рассматриваемому прямоугольнику, но из-за шума оказываются смещены относительно его периметра, будем рассматривать не только точки t периметра прямоугольника, но и их окрестность $O(t)$. Если в $O(t)$ есть хоть одна точка контура, то считаем, что в точке t периметра есть контур. Для этого определим следующее одномерное изображение на точках периметра прямоугольника $\partial P_{w,h,\alpha}$:

$$F(t) = \max_{v \in O(t)} Canny(v), \quad (3)$$

причем $F(t) \in \{0, 1\}$. Размер окрестности выбирается в зависимости от степени зашумленности изображения; в данной работе была выбрана четырехсвязная окрестность.

Далее к изображению F последовательно применяются операции *эрозии* (\ominus):

$$F'(t) = (F \ominus Q_q)(t) = \min_{v \in Q_q(t)} F(v) \quad (4)$$

и *дилатации* (\oplus):

$$F''(t) = (F' \oplus Q_q)(t) = \max_{v \in Q_q(t)} F'(v), \quad (5)$$

где Q_q — одномерный структурирующий элемент [3] размера $q = \frac{h}{4}$, по сути задающий на периметре одномерную окрестность точки t радиуса q . В теории морфологического анализа последовательность операций (4)–(5) называется *открытием*. Применение операции открытия к $F(t)$ приведет к удалению последовательностей точек контура длины менее, чем четверть длины меньшей из сторон прямоугольника. Такой подход позволяет учитывать лишь те контура, которые «сонаправлены» сторонам рассматриваемого прямоугольника, при этом отбрасывая слишком короткие отрезки контуров (которые, скорее всего, перпендикулярны сторонам прямоугольника).

После этого вычисляется сумма значений функции $F''(\cdot)$ и выполняется проверка, что эта сумма не меньше, чем некоторый порог, в качестве которого в данной работе была выбрана половина длины периметра:

$$\sum_{t \in \partial P_{w,h,\alpha}(x,y)} F''(t) \geq \frac{S(\partial P_{w,h,\alpha})}{2}. \quad (6)$$

Условие C_2 : плотность границ. При работе алгоритма, основанного на контурном анализе, значительную часть ложных тревог порождают сложнотекстурированные области, содержащие большое количество границ всевозможных направлений (характерным примером могут служить лесные массивы). В таких областях до трети точек могут быть отмечены алгоритмом Канни как содержащие контур, и, следовательно, в них можно найти большое количество ложных прямоугольников. На рис. 1 представлено два примера искомых строений с однородной внутренней областью (a и b) и контуры, ограничивающие область со сложной структурой (c и d).

Если рассматривать исключительно точки периметра прямоугольника, обе области содержат несколько искомых контуров, но можно отметить одно различие между ними: поскольку снимки инфракрасного диапазона отличаются нечувствительностью к мелким объектам, то крыша дома представляет собой достаточно однородную область, внутри которой мало контуров. Это наблюдение может служить еще одним признаком: во внутренней области рассматриваемого прямоугольника плотность границ должна быть незначительна (мы рассматривали значение в 5% от площади внутренней области). Данное условие можно записать следующим образом:

$$\sum_{t \in P_{w,h,\alpha}^0(x,y)} Canny(t) \leq \frac{S(P_{w,h,\alpha}^0)}{20}. \quad (7)$$

Условие C_3 : средняя яркость. Особенностью спутниковых снимков является большое количество контуров в областях растительности, что может породить значительное количество ложных тревог. В то же время инфракрасные изображения характеризуются тем, что яркость изображений строений на них в среднем ниже, чем яркость окружающей растительности (на этом наблюдении основаны вегетационные индексы, в частности,

широкоизвестный NDVI [17]). Подобная информация никак не связана непосредственно с контурами, а потому хорошо их дополняет, позволяя отбрасывать однородные прямоугольные участки типа полей, газонов и т. д. Предлагаемое условие имеет следующий вид:

$$\frac{\sum_{t \in P_{w,h,\alpha}^0(x,y)} Image(t)}{S(P_{w,h,\alpha}^0)} < \frac{\sum_{t \in P_{w+r+d,h+r+d,\alpha}(x,y)} Image(t) - \sum_{t \in P_{w+r,h+r,\alpha}(x,y)} Image(t)}{S(P_{w+r+d,h+r+d,\alpha}) - S(P_{w+r,h+r,\alpha})}. \quad (8)$$

Здесь средняя яркость внутренней области прямоугольника (P^0) сравнивается со средней яркостью прямоугольного «кольца» шириной d , построенного вокруг него на расстоянии r . Это расстояние представляет собой отступ от края прямоугольника: границы на реальных изображениях крайне редко имеют вид ступеньки, зато гораздо чаще они оказываются размыты на несколько пикселей (именно это размытие находит оператор Собела как градиент яркости), вследствие чего ближайšie к границе прямоугольника точки будут иметь практически такую же яркость, что и сам объект, а фон будет начинаться на несколько пикселей далее. Величина отступа зависит от типа и качества изображения, а ширина кольца d выбиралась так, чтобы его площадь была примерно равна площади внутренней области прямоугольника.

Таким образом, с учетом предложенных условий решающее правило алгоритма, определяющее наличие в некоторой произвольной точке (x, y) прямоугольника с параметрами (w, h, α) , выглядит следующим образом:

$$P(x, y) = \begin{cases} P_{w,h,\alpha}(x, y), & \text{если } Acc(x, y) = \max_{(x',y') \in O_w(x,y)} Acc(x', y') \\ & \text{и выполнены } C_2, C_3, \\ \emptyset & \text{иначе,} \end{cases} \quad (9)$$

где

$$Acc(x, y) = \max_{w,h,\alpha} \left(\left(\frac{1}{w+h} \sum_{t \in \partial P_{w,h,\alpha}(x,y)} Sobel(t) \right) | C_1 \right), \quad (10)$$

$$P_{w,h,\alpha}(x, y) = \arg \max_{w,h,\alpha} \left(\left(\frac{1}{w+h} \sum_{t \in \partial P_{w,h,\alpha}(x,y)} Sobel(t) \right) | C_1 \right), \quad (11)$$

$O_w(x, y)$ — окрестность точки (x, y) радиуса w , т. е. совпадающего с максимальной по длине стороне прямоугольника, что позволяет отсеять случаи пересечения расположенных рядом прямоугольников; C_1, C_2, C_3 — специальные условия, отражающие характерные свойства искомых объектов.

Завершим данный пункт описанием общей схемы предлагаемого алгоритма, состоящего из следующих шагов.

1. Сглаживание входного изображения (*guided filter*), вычисление градиентов яркости оператором Собела и поиск контуров алгоритмом Канни.

2. Для каждого набора параметров (w, h, α) из выбранного диапазона значений (2) с помощью процедуры голосования точек градиентов яркости изображения (10) и условием совпадения формы C_1 (6) строится аккумуляторный массив (размером, совпадающим с размером входного изображения, и соответствующий центрам прямоугольников). Затем

полученные аккумуляторные массивы объединяются операцией поточечного максимума с сохранением набора параметров (11), на котором этот максимум достигается.

3. В результирующем аккумуляторном массиве отбрасываются точки, не удовлетворяющие условиям плотности границ (7) и средней яркости (8). Среди оставшихся ищутся локальные максимумы (9) по окрестности размером с большую сторону прямоугольника w , которые передаются на выход алгоритма.

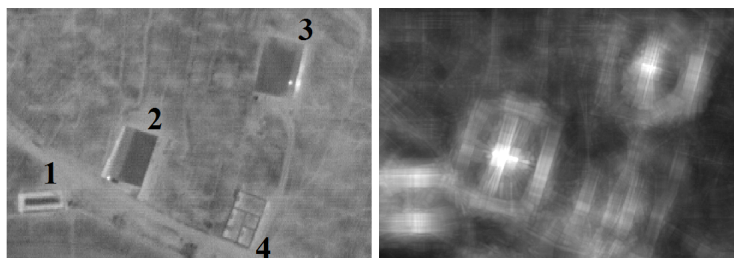


Рис. 3. Пример заполнения аккумуляторного массива при поиске прямоугольных контуров с помощью преобразования Хафа. Яркость соответствует значению аккумулятора: чем она выше, тем больше точек контуров исходного изображения проголосовали за прямоугольник с центром в этой точке. Представлены примеры строений (2 и 3), и типичные случаи прямоугольных объектов, искомыми строениями не являющихся (1 и 4)

На рис. 3 представлен пример построенного аккумуляторного массива, полученного в ходе поиска прямоугольных строений на изображении загородной местности. Можно видеть, что два прямоугольных строения породили в аккумуляторном массиве два выраженных локальных максимума (две яркие точки), соответствующих центрам зданий 2 и 3. В то же время объект 1 в левой нижней части изображения, чья ширина оказалась меньше рассматриваемых параметров фигур, породил лишь две размытые полосы, не локализуемые в конкретный объект. Помимо этого, учет однородности яркости внутри рассматриваемого прямоугольника позволил отбросить прямоугольный объект 4, не являющийся строением.

В следующем разделе будет приведено обсуждение реализации данного алгоритма, а также результаты численных экспериментов на реальных данных.

3. Реализация алгоритма и результаты численных экспериментов

Предложенный в предыдущем пункте алгоритм выполняет перебор контуров заданной формы всевозможных размеров и ориентаций, которые могли бы быть на входном изображении. Разумеется, если у нас нет априорной информации о размерах искомым объектов (при поиске строений на аэрокосмических изображениях это обычная ситуация), то такой перебор может потребовать значительного количества времени. Поэтому, если говорить о практическом применении данного метода, то остро встает вопрос об оптимизации вычислений и ускорении работы алгоритма. Последняя задача в настоящее время решается с помощью использования высокопроизводительных многоядерных графических процессоров. В связи с этим было принято решение оптимизировать алгоритм для работы на видеокартах; в качестве архитектуры параллельных вычислений была выбрана технология CUDA [18].

Сформулируем следующие идеи по оптимизации вычислений алгоритма.

1. Поскольку *guided filter* и оператор Собела представляют собой оконные методы (свертку), то их реализация на видеокарте не представляет затруднений. Быстрая реализация алгоритма Канни была взята из библиотеки OpenCV [19].

2. Рекуррентное вычисление аккумуляторного массива для каждого набора параметров (w, h, α) позволяет заметно сократить объем требуемой памяти, если хранить только результирующий аккумуляторный массив, по мере необходимости обновляя его на каждой итерации. Также за счет того, что этот массив получается двумерным, в нем быстрее осуществляется поиск локальных максимумов.

3. Прореживание точек: снимки земной поверхности неизбежно содержат некоторое размытие изображенных объектов и шум, из-за чего локализации контура со сдвигом на пиксель по горизонтали или вертикали будут практически идентичны. В связи с этим нет необходимости рассматривать каждый пиксель изображения как потенциальный центр фигуры, так как значения аккумуляторного массива в двух соседних точках будут мало отличаться друг от друга: они оба будут содержать одинаковые параметры фигуры и вполне корректно локализовать искомый объект. Поэтому возможно учитывать каждый второй или третий пиксель, и тем самым заметно сократить размер аккумуляторного массива.

4. Идея преобразования Хафа — «точка голосует за центр фигуры» — может замедлить выполнение алгоритма на видеокарте, так как за этот центр может одновременно голосовать множество точек. Поэтому предлагается иной подход — «центр контура опрашивает все точки, которые могли бы за него проголосовать». Раз форма известна, то можно проверить все пиксели, которые могли бы проголосовать за фигуру с центром в выбранной точке, тем самым избегая возможной записи в одну и ту же ячейку памяти из разных потоков вычислений.

5. Предварительный расчет векторов сдвигов: при фиксированных параметрах фигуры каждый центр опрашивает точки с одним и тем же смещением относительно себя, поэтому можно предварительно рассчитать вектора сдвигов всех точек периметра фигуры, и далее использовать их для вычисления значения аккумулятора в каждой точке. Расчет поворота требует вычисления синусов и косинусов, и делать его для каждой точки в отдельности крайне затратно. Предварительный расчет векторов сдвигов позволяет выполнить эту операцию единожды, и затем вычислять координаты точек повернутой фигуры с помощью операций сложений и вычитаний.

6. Заполнение аккумуляторного массива (10) и проверка условия близости формы C_1 (6) производятся по периметру фигуры, т. е. для прямоугольников, например, это порядка $2(w + h)$ точек. В то же время условия C_2 (7) и C_3 (8) рассчитываются по внутренней области фигуры, что требует перебора порядка $w \times h$ точек, а, следовательно, — значительно большего времени работы. Именно поэтому в (10) и (11) в качестве условий фигурирует лишь C_1 , а два других условия вынесены за пределы заполнения аккумулятора (т. е. за пределы параллельной части) и проверяются только для точек предполагаемых центров прямоугольников (9). Добавление условий C_2 и C_3 в параллельную часть сильно замедлит работу алгоритма, но не приведет к заметному повышению качества работы.

Построенный на основе этих предложений алгоритм с перебором прямоугольников с размерами от $l_{min} = 31$ до $l_{max} = 91$ с шагом 2 (2), продемонстрировал время работы порядка двадцати двух секунд для изображения 3000×3000 пикселей на системе, оснащенной процессором Intel i7-4770 и видеокартой Nvidia GeForce GTX 780 Ti.

Для проверки качества работы предложенного алгоритма использовалась база из 100 инфракрасных аэроснимков загородной местности размером 3000×3000 пикселей, для которых была сделана ручная разметка строений прямоугольной формы. В случае, если алгоритм находил прямоугольный контур в небольшой окрестности от размеченного вручную и их размеры не сильно отличались, то такой контур считался верным обнаружением. Полученные результаты приведены в таблице.

Таблица

Результаты численного эксперимента по локализации строений прямоугольной формы

Характеристика	Количество	Процент
Строений найдено	2 213	78,45%
Строений пропущено	608	21,55%
Ложных тревог	410	15,63%

На рис. 4 приведены примеры работы алгоритма. В случае разреженного расположения объектов и относительно однородного фона алгоритм достаточно корректно обнаруживает искомые объекты, не допуская ошибок. Для более сложных сцен, характеризующихся плотной и хаотичной застройкой, алгоритм может пропускать некоторые объекты: либо по причине сложной формы (примеры 2 и 6), либо из-за того, что строения расположены вплотную друг к другу (примеры 3 и 4). Ложные тревоги связаны с объектами, подобно прямоугольным строениям также имеющим две параллельные и достаточно протяженные границы: дороги, мосты, участки берега и т. д. (примеры 1, 5 и 7). В то же время стоит отметить, что большую часть объектов алгоритм обнаружил корректно.



Рис. 4. Примеры работы алгоритма на аэроснимках плотной загородной застройки. Белым цветом выделены найденные строения. Цифрами 1, 5 и 7 отмечены ложные тревоги, цифрами 2, 3, 4 и 6 — пропуски

Заключение

Предложенный в работе метод локализации строений заданной формы на аэроснимках в инфракрасном диапазоне применим к задаче поиска объектов различной формы, в общем случае, произвольных фигур, не обязательно выпуклых. При этом представляется, что чем сложнее искомая фигура, тем точнее будет работать алгоритм, но в то же время и значительно дольше. Однако стоит учитывать, что наиболее эффективную работу алгоритм сможет продемонстрировать для определенного класса изображений с особыми

спектральными и текстурными характеристиками: однородность фона, разреженное положение искомым объектов, разрешение порядка 1–3 метров на пиксель.

Помимо этого стоит учесть, что анализ контуров представляет собой достаточно неустойчивую к шуму и сложную в настройке операцию. Применение такого анализа в практических приложениях без использования дополнительных методов может быть затруднительно. Повышения универсальности можно добиться путем добавления элементов сегментации и анализа фона. Значительная часть ложных тревог связана с протяженными объектами типа дорог, чья ширина примерно соответствует ширине обычного дома. Предварительное разбиение снимка на однородные области поможет исключить такие ситуации. В свою очередь, уменьшение уровня ложных тревог позволит понизить пороги основного алгоритма и добиться более высокого уровня распознавания.

Также стоит упомянуть один из наиболее популярных в настоящее время подходов в обработке изображений, связанный с обучением многослойных сверточных нейронных сетей. Применение последних в основном связано с поиском объектов, обладающих достаточно сложной структурой: машин, животных и т.д. Строения на снимках в инфракрасном диапазоне, как было показано выше, подобной сложной структурой не обладают. Однако и для них можно подобрать некоторые характерные признаки. Причем некоторые из этих признаков сами по себе выражаются через операцию свертки: оператор Собела, плотность границ, средняя яркость области. Следовательно, приведенный в данной работе алгоритм может помочь при определении структуры нейронной сети, которая решала бы рассмотренную задачу.

Работа выполнена при частичной поддержке комплексной программы ФНИ УрО РАН (проект 15-16-1-14).

Литература

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. Пер. с англ. М.: Техносфера, 2005. 1072 с.
2. Сирота А.А., Соломатин А.И. Статистические алгоритмы обнаружения границ объектов на изображениях // Вестник ВГУ. Серия: Системный анализ и информационные технологии. 2008. № 1. С. 58–64.
3. Визильтер Ю.В., Желтов С.Ю., Бондаренко А.В., Ососков М.В., Моржин А.В. Обработка и анализ изображений в задачах машинного зрения: Курс лекций и практических занятий. М.: Физматкнига, 2010. 672 с.
4. Леухин А.Н. Многомерный гиперкомплексный контурный анализ и его приложения к обработке изображений и сигналов. Диссертация на соискание ученой степени доктора физико-математических наук. Йошкар-Ола: Редакционно-издательский центр МарГТУ, 2004. 389 с.
5. Фурман Я.А. Введение в контурный анализ. 2-е изд., испр. М.: ФИЗМАТЛИТ, 2003. 592 с.
6. Hough P.V.C. Methods, Means for Recognizing Complex Patterns / U.S., Patent 3069654, 1962.
7. Heikkila M., Pietikainen M. A Texture-Based Method for Modeling the Background and Detecting Moving Objects // IEEE Transactions on Pattern Analysis and Machine

- Intelligence. 2006. Vol. 28, No. 4. pp. 657–662. DOI: 10.1109/TPAMI.2006.68.
8. Tomasi C., Manduchi R. Bilateral Filtering for Gray and Color Images // IEEE Proceedings of the 6-th International Conference on Computer Vision, Bombay, India. 7-th Jan. 1998. pp. 839–846. DOI: 10.1109/ICCV.1998.710815.
 9. K. He, J. Sun, X. Tang. Guided Image Filtering // IEEE Transactions on Software Engineering. 2013. Vol. 35(6). pp. 1397–1409. DOI: 10.1109/TPAMI.2012.213.
 10. Duda R., Hart P. Pattern Classification and Scene Analysis. John Wiley and Sons, 1973. pp. 271–272. DOI: 10.2307/1573081.
 11. Canny J. A Computational Approach to Edge Detection // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1986. Vol. PAMI-8, No. 6. pp. 679–698. DOI: 10.1109/TPAMI.1986.4767851.
 12. L. Xu, E. Oja Randomized Hough Transform (RHT): Basic Mechanisms, Algorithms, and Computational Complexities // CVGIP: Image Understanding. 1993. Vol. 57, No. 2. pp. 131–154. DOI: 10.1006/ciun.1993.10091.
 13. Harris C., Stephens M. A Combined Corner and Edge Detector // Proceedings of the 4-th Alvey Vision Conference, University of Manchester. 31-st Aug. — 2-nd Sept. 1988. pp. 147–151. DOI: 10.5244/C.2.23.
 14. Rosten E., Porter R., Drummond T. Faster and Better: A Machine Learning Approach to Corner Detection // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010. Vol. 32, No. 1. pp. 105–119. DOI: 10.1109/CIT.2010.109.
 15. Noronha S., Nevatia R. Detection and Modeling of Buildings from Multiple Aerial Images // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2001. Vol. 23, No. 5. pp. 501–518. DOI: 10.1109/34.922708.
 16. Корнилов Ф.А. Поиск прямоугольных контуров на изображениях с помощью преобразования Хафа // Современные проблемы математики и ее приложений: Труды 45-й международной молодежной школы-конференции (Екатеринбург, 2–8 февраля 2014 г.). Екатеринбург: ИММ УрО РАН, 2014. С. 195–198.
 17. Вегетационные индексы: URL: <https://earthobservatory.nasa.gov/Features/MeasuringVegetation/> (дата обращения: 18.05.2017)
 18. Боресков А.В. и др. Параллельные вычисления на GPU. Архитектурная и программная модель CUDA. М.: Изд-во Моск. ун-та. Серия: Суперкомпьютерное образование, 2012. 336 с.
 19. Библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом: URL: <http://www.opencv.org/> (дата обращения: 18.05.2017)

Дунаева Александра Валерьевна, математик, отдел прикладных проблем управления, ФГБУН Институт математики и механики им. Н.Н. Красовского УрО РАН (Екатеринбург, Российская Федерация); ассистент, кафедра высокопроизводительных компьютерных технологий, Институт естественных наук и математики, ФГБОУ ВО Уральский федеральный университет им. первого Президента России Б.Н. Ельцина (Екатеринбург, Российская Федерация)

Корнилов Фёдор Андреевич, к.ф.-м.н., н.с., отдел прикладных проблем управления, ФГБУН Институт математики и механики им. Н.Н. Красовского УрО РАН (Екатеринбург, Российская Федерация)

DOI: 10.14529/cmse170306

SPECIFIC SHAPE BUILDING DETECTION FROM AERIAL IMAGERY IN INFRARED RANGE

© 2017 A.V. Dunaeva^{1,2}, F.A. Kornilov¹

¹*N.N. Krasovskii Institute of Mathematics and Mechanics of the Ural Branch of the Russian Academy of Sciences (S.Kovalevskaya Str. 16, Yekaterinburg, 620990 Russia),*

²*Ural Federal University named after the first President of Russia B.N.Yeltsin (Mira Str. 19, Yekaterinburg, 620002 Russia)*

E-mail: maryanova.av@yandex.ru, kornilovfa@imm.uran.ru

Received: 24.07.2017

This paper describes an approach to detection of specific shape buildings from the aerial imagery in the infrared range. The proposed algorithm uses contour analysis and is based on a modification of the generalized Hough transform that allows to detect curves defined by a small number of parameters. The main idea is to build a two-dimensional accumulator array for each possible parameter set of the specified curve, and to combine obtained arrays into the resultant accumulator array whose local maxima correspond to the positions of the sought objects. The gradient magnitudes of the original image are used to fill the arrays. The closeness of the found contours to the predefined curve is determined by the morphological analysis of the values calculated by the Canny edge detector. Filtering detected objects relies on the density of boundaries in their internal area with the ratio of the average intensity inside and outside the contour that provides high sensitivity to the specified types of objects and reduces the number of false alarms of the algorithm. The proposed approach was tested on the problem of localization of rectangular buildings and showed the appropriate quality for practical use.

Keywords: image processing, object detection, contour analysis, mathematical morphology.

FOR CITATION

Dunaeva A.V., Kornilov F.A. Specific Shape Building Detection from Aerial Imagery in Infrared Range. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2017. vol. 6, no. 3. pp. 84–100. (in Russian) DOI: 10.14529/cmse170306.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Gonzalez R.C., Woods R.E. Digital Image Processing. 2-nd Edition, 2001. 1072 p.
2. Sirota A.A., Solomatin A.I. Statistical and Neural Network Algorithms of Allocation of Objects Border in Images. Proceedings of Voronezh State University. Series: Systems Analysis and Information Technologies. 2008. no. 1. pp. 58–64. (in Russian)
3. Vizilter Y.V., Zheltov S.Y., Bondarenko A.V., Ososkov M.V., Morzhin A.V. Image Processing and Analysis in Computer Vision: A Course of Lectures and Practical Exercises. Moscow, Publishing Fizmatkniga, 2010. 672 p. (in Russian)

4. Leukhin A.N. Multidimensional Hypercomplex Contour Analysis and Its Applications to Image and Signal Processing. Thesis for the Degree of Doctor of of Physico-Mathematical Sciences. Yoshkar-Ola, Publishing Mari State Technical University, 2004. 389 p. (in Russian)
5. Furman Y.A. Introduction to Contour Analysis. 2-nd Edition. Moscow, Publishing FIZMATLIT, 2003. 592 p. (in Russian)
6. Hough P.V.C. Methods, Means for Recognizing Complex Patterns. U.S., Patent 3069654, 1962.
7. Heikkila M., Pietikainen M. A Texture-Based Method for Modeling the Background and Detecting Moving Objects. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2006. vol. 28, no. 4. pp. 657–662. DOI: 10.1109/TPAMI.2006.68.
8. Tomasi C., Manduchi R. Bilateral Filtering for Gray and Color Images. IEEE Proceedings of the 6-th International Conference on Computer Vision, Bombay, India. 7-th Jan. 1998. pp. 839–846. DOI: 10.1109/ICCV.1998.710815.
9. K. He, J. Sun, X. Tang. Guided Image Filtering. IEEE Transactions on Software Engineering. 2013. vol. 35(6). pp. 1397–1409. DOI: 10.1109/TPAMI.2012.213.
10. Duda R., Hart P. Pattern Classification and Scene Analysis. John Wiley and Sons, 1973. pp. 271–272. DOI: 10.2307/1573081.
11. Canny J. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1986. vol. PAMI-8, no. 6. pp. 679–698. DOI: 10.1109/TPAMI.1986.4767851.
12. L. Xu, E. Oja Randomized Hough Transform (RHT): Basic Mechanisms, Algorithms, and Computational Complexities. CVGIP: Image Understanding. 1993. vol. 57, no. 2. pp. 131–154. DOI: 10.1006/ciun.1993.10091.
13. Harris C., Stephens M. A Combined Corner and Edge Detector. Proceedings of the 4-th Alvey Vision Conference, University of Manchester. 31-st Aug. — 2-nd Sept. 1988. pp. 147–151. DOI: 10.5244/C.2.23.
14. Rosten E., Porter R., Drummond T. Faster and Better: A Machine Learning Approach to Corner Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010. vol. 32, no. 1. pp. 105–119. DOI: 10.1109/CIT.2010.109.
15. Noronha S., Nevatia R. Detection and Modeling of Buildings from Multiple Aerial Images. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2001. vol. 23, no. 5. pp. 501–518. DOI: 10.1109/34.922708.
16. Kornilov F.A. Using the Hough Transform to Detect Rectangular Contours on Images. *Sovremennye problemy matematiki i ee prilozhenii: Trudy 45-i mezhdunarodnoi molodezhnoi shkoly-konferentsii (Yekaterinburg, 2 –8 fevralya 2014)* [Topical problems of mathematics and its applications: Proceedings of International (45-th National) Youth School-Conference (Yekaterinburg, Russia, February, 2–8, 2014)]. Yekaterinburg, Publishing IMM UB RAS, 2014. pp. 195–198. (in Russian)
17. Vegetation indices: URL: <http://go-url.ru/ix9o> (accessed: 18.05.2017)
18. Boreskov A.V. et. al. Parallel Computing on GPU. Architectural and Software Model CUDA. Moscow, Publishing Lomonosov Moscow State University. Series: Supercomputer education, 2012. 336 p. (in Russian)

19. The library of computer vision, image processing and computational mathematics algorithms with opened source code: URL: <http://www.opencv.org/> (accessed: 18.05.2017)