

ГИБРИДНЫЙ АЛГОРИТМ РАСПОЗНАВАНИЯ СТРОЕНИЙ НА СПУТНИКОВЫХ СНИМКАХ НА ОСНОВЕ МЕТОДА ЖУКА И АЛГОРИТМА ИСКЛЮЧЕНИЯ ОБЛАСТЕЙ

© 2024 И.В. Баранова, С.В. Гилин

Сибирский федеральный университет

(660041 Красноярск, пр. Свободный, д. 79)

E-mail: ibaranova@sfu-kras.ru, gilin.stepan@mail.ru

Поступила в редакцию: 10.01.2024

В статье предлагается новый метод распознавания строений на спутниковых снимках. Представленный метод является гибридным, он основан на алгоритме исключения областей и методе жука. Алгоритм исключения областей представляет собой хорошо известный и эффективный способ сегментации изображения на регионы схожих пикселей по различным признакам: цвет, текстура, яркость, форма и др. Метод жука — классический метод контурного анализа, выполняющий последовательное вычерчивание границы между объектом и фоном. В рамках работы предлагаемого алгоритма сначала метод исключения областей выделяет потенциальные области, в которых могут находиться строения и устраняет нежелательные элементы на изображении (растительность, водные поверхности и дороги), которые могут быть ложно распознаны как строения. Далее модифицированный метод жука определяет местоположение и контуры строений. На финальном этапе среди обнаруженных объектов выявляются искусственно созданные объекты, у которых имеется объем. Для реализации проверки объектов на искусственное происхождение и объемность разработаны собственные методы. Представленный алгоритм распознавания показывает хорошую точность распознавания и не требует обучающей выборки. В статье описывается программная реализация предлагаемого метода. Демонстрируются результаты вычислительных экспериментов по оцениванию эффективности метода и сравнению с тремя известными алгоритмами распознавания.

Ключевые слова: распознавание строений, спутниковые снимки, гибридный метод, метод жука, метод исключения областей, текстурные характеристики, границы зданий, потенциальные области, классификация.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Баранова И.В., Гилин С.В. Гибридный алгоритм распознавания строений на спутниковых снимках на основе метода жука и алгоритма исключения областей // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2024. Т. 13, № 2. С. 56–76. DOI: 10.14529/cmse240204.

Введение

В настоящее время задача обнаружения на спутниковых снимках зданий и различных строений является одной из актуальных и практически важных задач в области распознавания образов, компьютерного зрения, искусственного интеллекта и анализа данных. Поставленная задача распознавания заключается в обнаружении объектов заданных классов на изображении. Распознавание указанных объектов на аэрокосмических снимках может применяться для решения широкого круга практических задач в различных областях деятельности: ориентации на местности, составлении карт (топологического дешифрирования снимков), обследовании и строительном контроле зданий и сооружений, нахождении оптимального маршрута движения транспорта, мониторинга инженерных сетей в построенных комплексах, определении законности строений и других. Так, например, в работе [1] нами

решается практический пример подобной задачи — задача распознавания строений, находящихся в водоохраных зонах.

Решение задачи распознавания объектов на изображениях выполняется в несколько этапов: предварительная обработка исходных изображений; сегментация изображения (группировка областей с одинаковыми визуальными характеристиками); обнаружение объектов и извлечение векторов признаков для обнаруженных объектов; классификация объектов на основании выделенных признаков. Этапы обнаружения и классификации объектов чаще всего формулируются как структурные и оптимизационные задачи на многослойных объектах, что, несомненно, увеличивает сложность решаемой задачи распознавания. Поэтому разработка новых и усовершенствование существующих автоматических алгоритмов распознавания аэрокосмических снимков высокого разрешения, способных находить решение задачи за реальное время, является одним из актуальных и востребованных направлений в широком спектре предметных областей, связанных с анализом визуальных данных.

Целью данной работы является разработка нового алгоритма распознавания строений на спутниковых снимках. Для достижения цели решаются следующие задачи:

1. разработка и обоснование собственного алгоритма распознавания изображений как синтеза метода жука и алгоритма исключения областей;
2. создание программного обеспечения на языке Python, реализующего работу предложенного метода распознавания;
3. проведение вычислительных экспериментов по оцениванию характеристик разработанного алгоритма, а также его сравнению с тремя наиболее распространенными методами распознавания по скорости работы и точности распознавания.

Статья организована следующим образом. Она состоит из введения, пяти разделов и заключения. Раздел 1 содержит обзор работ по теме исследования. В разделе 2 описана общая постановка задачи. В разделе 3 рассмотрена предобработка изображений и существующие методы решения задачи. В разделе 4 описан собственный алгоритм распознавания строений на спутниковых снимках. В разделе 5 описаны результаты работы предложенного алгоритма, описаны вычислительные эксперименты и результаты сравнения предложенного алгоритма с некоторыми из существующих алгоритмов. Заключение содержит краткое изложение полученных результатов и выводы об их применении.

1. Обзор работ по теме исследования

На данный момент существует ряд методов, успешно решающих задачу распознавания объектов на спутниковых снимках. Одним из широко используемых подходов к автоматическому распознаванию снимков является группа методов, основанных на разрезах в графах. Ярким представителем этой группы является алгоритм сегментации спутниковых изображений на основе суперпикселей и разрезов на графах, представленный в работе [2]. Несомненным преимуществом алгоритмов графового подхода является сокращение времени на классификацию и детекцию объектов, но из-за представления каждого пикселя изображения в виде вершины графа увеличивается вычислительная сложность задачи. Еще одним известным подходом к решению поставленной задачи служат методы исключения областей [3, 12], основывающиеся на сегментации изображения с разделением на регионы, объединяющие схожие пиксели по различным признакам, таким как цвет, текстура или форма. Также в этих методах предполагается, что объекты и фон имеют разные интен-

сивности пикселей или цветовые характеристики, что позволяет легко выделить объекты, установив пороговое значение для пикселей или цветов.

В последние годы одним из самых популярных подходов к распознаванию спутниковых снимков является нейросетевой подход. Нейронные сети различных архитектур были успешно применены для обработки многопиксельных спутниковых изображений высокого разрешения и позволили найти эффективное решение задачи сегментации объектов. В том числе ряд современных нейросетевых архитектур использовался и непосредственно для решения задач обнаружения зданий на снимках — например, архитектуры UNet, SegNet и Mask R-CNN. Так, в статье [4] применяется архитектура SegNet. Довольно часто в рамках данного подхода использование нейронных сетей сочетается с применением методов машинного обучения, преимущественно — методов глубокого обучения. Например, в работе [5] используется ассиметричная CNN. На данный момент одной из лучших нейросетей обнаружения строений является архитектура UANet + PVT-V2-B2 [6], поэтому в данной работе будет выполняться сравнение предлагаемого нами алгоритма распознавания именно с указанной нейросетью. В целом, можно отметить, что нейросетевые методы успешно справляются с распознаванием зданий и строений на аэроснимках, но для высокой точности распознавания требуется иметь достаточно объемную обучающую выборку, что в некоторых практических задачах является существенной проблемой.

Еще одним активно используемым подходом к решению задачи распознавания объектов на спутниковых снимках являются методы с выделением контуров объектов: алгоритмы активных контуров (например, метод затравочных точек или метод на основе энергии уровней) или алгоритмы водораздела. Модели, основанные на активных контурах, показывают высокую точность распознавания вне зависимости от качества изображения [7]. Данные методы не требуют обучающей выборки для своей работы. Ряд авторов предлагает методы, комбинирующие в себе часть из перечисленных подходов. К примеру, в работе [8] применялась структура из замкнутых контуров с дальнейшей постобработкой. Как правило, подобное комбинирование позволяет избавиться от недостатков, присущих определенному подходу, и получить более точные результаты распознавания. Например, часто используется пороговая сегментация для грубой предварительной сегментации, затем для уточнения результатов применяются более сложные методы, такие как нейросети или графовые алгоритмы, как представлено в работе [9].

В данной работе предлагается собственный алгоритм распознавания строений на спутниковых снимках, основанный на алгоритме исключения областей, с использованием собственной модификации метода обнаружения контуров — метода жука. Предлагаемый алгоритм распознавания будет базироваться на версии алгоритма исключения областей, предложенного в работе [3].

2. Постановка задачи

Как уже было сказано выше, в данной работе решается задача распознавания строений на изображении. Приведем необходимые определения, позволяющие уточнить условия поставленной задачи.

2.1. Необходимые уточнения для формулировки задачи

В Федеральном законе РФ от 30 декабря 2009 года N 384-ФЗ «Технический регламент о безопасности зданий и сооружений» дается следующее определение: сооружение — на-

земная или подземная система, выполненная методом строительства и предназначенная для проживания и/или хозяйственной деятельности людей. Для сооружений также можно использовать термины «искусственная структура» или «искусственный объект». В том же законе под строением понимается любое объемное наземное строительное сооружение, технологически имеющее неразрывную связь с грунтом, а под зданием понимается неподвижное наземное строение, имеющее внутренний полезный объем, а также имеющее сети и системы инженерно-технического обеспечения. Таким образом, в рамках задачи, которая решается в данной работе, будет выполняться распознавание наземных строений с внутренним полезным объемом. К числу таких строений будем относить здания (жилые и нежилые), сарай, гаражи, бани и др. Суть решаемой задачи заключается в следующем: алгоритму на вход поступает спутниковый снимок, после чего он должен обнаружить на данном изображении все строения, выделив их контуры. После детекции некоторого объекта на изображении выделяются его признаки (цвет пикселей, текстура, углы объекта, форма контура), на основании которых затем выполняется классификация обнаруженного объекта — т.е. отнесение его к категории искусственно созданных объектов. Данная классификация возможна благодаря свойству, которым обладают все искусственные объекты: они либо имеют малую постоянную кривизну (объекты круглой формы), либо несколько точек высокой кривизны (объекты прямоугольной формы) или вообще не имеют кривизны. Также стоит отметить, что мы выполняем распознавание строений, которые в отличие от линейных и плоскостных сооружений обязательно имеют внутренний объем, что позволяет их явным образом отличать от большинства инженерно-технических сооружений. Примерами линейных сооружений являются линии электропередач, мосты, телевизионные башни, мачты и тому подобное. К плоскостным, например, относятся оборудованные стоянки автомашин. Признаком того, что строение имеет объем, является наличие у строения на спутниковом снимке тени, которая в зависимости от времени суток меняет свое расположение и форму. Таким образом, для отнесения обнаруженного искусственного объекта к строениям необходимо наличие у него тени.

2.2. Математическая постановка задачи распознавания образов

Пусть $X = \{x_1, x_2, \dots, x_n\}$ — множество объектов распознавания, $x : x \in X$ — объект распознавания. Каждый объект $x \in X$ описывается вектором признаков, т.е. задано F — пространство признаков, функция $f(x) : X \rightarrow F$ ставит в соответствие каждому объекту $x \in X$ точку $f(x)$. Таким образом, вектор $f(x)$ представляет собой образ объекта $x \in X$.

Пусть в пространстве признаков F определено множество классов $C = \{C_1, \dots, C_m\}$, $C_i \in F$, $i = 1, \dots, m$, $C_i \cap C_j = \emptyset$ при $i \neq j$. Функция классификации $g(x) : X \rightarrow C$ — неизвестная индикаторная функция, разбивающая пространство образов F на m непересекающихся классов C_i . Решающее правило $\hat{g}(f) : F \rightarrow C$ — функция, которая является оценкой для $g(x)$ на основании образа объекта, то есть $\hat{g}(f) = \hat{g}(f(x))$. Пусть задано множество $\hat{X} = \{x_j\}$, $j = 1, \dots, k$, $k < n$, $\hat{X} \subset X$, для которого задана информация о функциях $f(x)$ и $g(x)$, т.е. $f_j = f(x_j)$ и $g_j = f(g_j)$, для $x_j \in \hat{X}$, $j = 1, \dots, k$. Множество (g_j, f_j) называется множеством прецедентов.

Требуется построить такое решающее правило (*алгоритм классификации*) $\hat{g}(f)$, чтобы $Q(\hat{g}, C) = \min_f Q(g, C)$, где $Q(g, C)$ — выбранный критерий качества классификации. В данной работе в качестве критерия используются четыре метрики (точность, полнота, усредненный индекс Жаккара и мера F1), представляющие собой функции от количества

ошибок первого и второго рода [10]. Ошибка первого рода — ситуация, когда происходит ложное срабатывание алгоритма классификации на объекте, т.е. обнаруженный объект класса фактически отсутствует на изображении. В статистическом контексте это можно интерпретировать как отклонение верной гипотезы об отсутствии объекта искомого класса на изображении. Ошибка второго рода заключается в упущении объекта, который действительно присутствует на изображении, т.е. алгоритм не обнаруживает его. Определения вышеуказанных метрик приводятся в разделе 5.

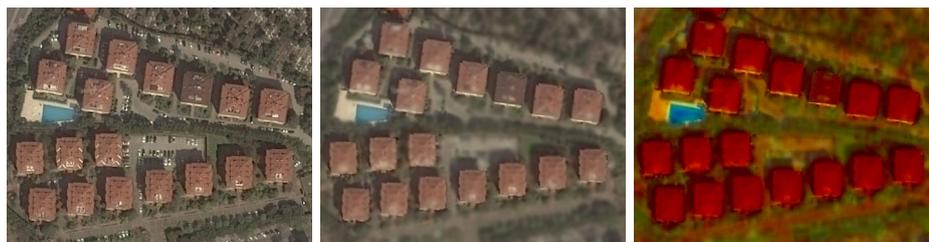
2.3. Особенности распознавания объектов на изображениях

Итак, в качестве объектов распознавания $x \in X$ выступают строения. Как уже было сказано выше, распознавание строений выполняется на спутниковых снимках. В рамках данной работы будут использоваться растровые изображения. Пусть имеется растровое изображение D размером $w \times h$, представленное двумерной матрицей пикселей, принадлежащих цветовому пространству RGB . Т.е. изображение задается в следующем виде: $D = \{p_{i,j}, i = 1, \dots, h, j = 1, \dots, w, \}$, где $p_{i,j}$ — пиксель, (i, j) — координаты пикселя на изображении, w, h — ширина и высота изображения, $p_{i,j} = \{R(p_{i,j}) \in [0, 255], G(p_{i,j}) \in [0, 255], B(p_{i,j}) \in [0, 255]\}$. Изображение содержит множество объектов распознавания $X = \{x_1, x_2, \dots, x_n\}$: $X \subseteq D$. Формально каждый объект распознавания $x_k \in X$ можно представить в виде множества пикселей: $x_k = \{p_{ij}, i = 1, \dots, h_k, j = 1, \dots, w_k\}$, где w_k, h_k — ширина и высота объекта.

2.4. Методы предварительной обработки и фильтрации изображения

Перед выполнением распознавания выполняется предварительная обработка изображения для устранения шумов и дефектов изображения. Шумы на изображении появляются в самых разнообразных формах, поэтому существует множество методов шумоподавления. В данной работе для устранения шума изображения используется метод скалирования изображения, который опирается на значения пикселей в исходном изображении и вычисляет новые значения для пикселей в увеличенном и уменьшенном изображении на основе окружающих пикселей.

Для выявления объектов с однотипным внешним видом используется сегментация изображения, т.е. группировка областей с одинаковыми визуальными характеристиками. В работе [11] показано, что одного набора параметров недостаточно для корректного сегментирования каждого здания на снимке. Если здания имеют сложную форму, иногда бывает невозможно сегментировать даже одно здание по одному набору параметров. Для упрощения определения признаков выполним фильтрацию изображения. Для фильтрации изображения выбран метод сдвига среднего значения. Сдвиг среднего значения — непараметрический метод анализа пространства признаков для определения максимума плотности вероятности [12]. Он используется для уменьшения шума, сглаживания текстур, улучшения контрастности. Данные свойства способствуют более эффективному распознаванию объектов на изображениях. Для программной реализации данного метода в работе используется функция `rugMeanShiftFiltering()` открытой библиотеки компьютерного зрения и обработки изображений `OpenCV`. Результаты применения предварительной обработки и выбранного метода фильтрации к исходному изображению показаны на рис. 1. Слева на рисунке представлено исходное спутниковое изображение, по центру — предобработанное и сглаженное изображение.



а) Исходное изображение б) Сглаженное изображение в) Преобразование к схеме HSV

Рис. 1. Предварительная обработка, сглаживание и фильтрация изображения

В качестве следующего шага фильтрации изображения выполним преобразование цветовой компонент изображения к цветовой модели HSV. В схеме HSV цвет представляется точкой в трехмерном пространстве (H, S, V), где H (Hue) — это цветовой тон, S (Saturation) — насыщенность, а V (Value) — яркость цвета. Указанная цветовая модель благодаря своей специфике позволяет более корректно определить порог необходимого значения для сегментации изображения, которое будет выполняться на следующем этапе задачи распознавания с помощью предлагаемого нами алгоритма распознавания. На рис. 1 справа продемонстрирован результат преобразования изображения к схеме HSV, причем выполняется преобразование изображения, уже прошедшего предобработку и сглаживание сдвигом среднего значения (т.е. выполняется преобразование изображения, приведенного на рис. 1б).

3. Модифицированный алгоритм синтеза метода «жука» и метода исключения областей

В работе предлагается новый метод автоматического обнаружения строений на спутниковых снимках, который является модифицированным гибридом алгоритма исключения областей и метода жука. Как уже было сказано ранее, в рамках поставленной задачи нам необходимо найти на изображении строения, которые относятся к искусственным объектам. Кроме того, алгоритм должен учесть требование из определения строения (из раздела 2.1), т.е. то, что строение обязательно имеет внутренний объем, что позволяет явным образом отличать его от большинства инженерно-технических сооружений. Исходя из всего вышесказанного, последовательность этапов работы предложенного алгоритма выглядит следующим образом:

1. Сегментация изображения, т.е. группировка областей с одинаковыми визуальными характеристиками и выявление потенциальных областей, в которых могут находиться искомые объекты (строения);
2. Обнаружение объектов и выделение их контуров;
3. Проверка контуров найденных объектов на кривизну для выявления искусственно созданных объектов;
4. Проверка найденных искусственно созданных объектов на наличие у них объема.

На первом этапе работы разработанного алгоритма распознавания для выполнения сегментации изображения используется метод исключения областей. На втором этапе для обнаружения объектов и выделения их контуров используется предложенный нами модифицированный алгоритм жука. Для выявления искусственно созданных объектов нами разработан алгоритм проверки контуров объектов на кривизну, а для выявления среди обна-

руженных искусственных объектов строений (в т.ч. зданий) — алгоритм проверки объекта на то, что у него есть объем.

3.1. Метод исключения областей

Метод исключения областей представляет собой хорошо известный и эффективный подход к распознаванию объектов на изображении. Он позволяет выявить потенциальные области, где могут находиться искомые объекты, основываясь на изменениях интенсивности пикселей и текстурных переходах, связанных с контурами объектов. Этот алгоритм будет применяться для удаления нежелательных элементов на изображении, таких как деревья, водные поверхности и дороги, которые могут быть ложно распознаны как здания. В нашей работе предлагается модификация метода исключения областей, базирующаяся на методе, приведенном в работе [3]. Оригинальный метод выполняет бинаризацию исходного полутонового изображения путем сравнения яркости каждого пикселя с заданным пороговым значением. Пиксели со значением яркости ниже порога окрашиваются черным цветом, в противном случае — белым. Отличия предлагаемой нами модификации метода заключаются в следующем.

1. На вход алгоритму поступает цветное изображение в схеме HSV.
2. Для каждого цветового тона (красного, зеленого и синего) устанавливается интервал значений яркости и насыщенности, при помощи которого можно выделить участки с необходимыми спектральными характеристиками.
3. Для каждого цвета на основе найденных интервалов задаются пороговые значения для яркости и насыщенности.
4. Для каждого пикселя изображения определяется его цветовой тон (путем проверки попадания в соответствующие интервалы значений), затем выполняется сравнение его яркости с пороговым значением яркости данного цвета и его насыщенности с пороговым значением насыщенности данного цвета. Если значение яркости ниже соответствующего порога, а насыщенности — ниже, то пиксель окрашивается белым цветом, в противном случае — черным. Это условие объясняется тем, что искусственные объекты имеют большую цветовую насыщенность, чем естественные объекты.

Введем обозначения, необходимые для описания алгоритма. Для корректной работы алгоритма необходимо задать интервалы значений каждого цветового тона: HR_{\min} — начало диапазона красного цвета, HR_{\max} — конец диапазона красного цвета, HG_{\min} и HG_{\max} — начало и конец диапазона зеленого цвета, HB_{\min} и HB_{\max} — начало и конец диапазона синего цвета. Указанные значения задаются экспертно на основе знаний о характеристиках изображений: $HR_{\min} = 300$, $HR_{\max} = 360$, $HG_{\min} = 90$, $HG_{\max} = 164$, $HB_{\min} = 165$, $HB_{\max} = 255$. В переменных SR_{\min} и SR_{\max} хранятся начало и конец диапазона насыщенности красного цвета, в SG_{\min} и SG_{\max} — начало и конец диапазона насыщенности зеленого цвета, в SB_{\min} и SB_{\max} — начало и конец диапазона насыщенности синего цвета. Переменные VR_{\min} , VR_{\max} , VG_{\min} , VG_{\max} , VB_{\min} , VB_{\max} задают границы диапазонов яркости для каждого из перечисленных цветов. Также для работы алгоритма используются пороговые переменные: TSR — пороговое значение насыщенности красных оттенков, TVR — пороговое значение яркости красных оттенков. Схожим образом определяются переменные TSG и TVG , TSB и TVB . Обозначим получение порогового значения насыщенности как $TS(f) = 3/4 * (f_{\max} - f_{\min})$, а получение порогового значения яркости как $TV(f) = 1/3 * (f_{\max} - f_{\min})$. Изображение в схеме HSV задается следующим образом: $D_{HSV} = \{d_{i,j}, i = 1, \dots, h, j = 1, \dots, w\}$, где $d_{i,j}$ — пик-

сель, (i, j) — координаты пикселя на изображении, w, h — ширина и высота изображения, $d_{i,j} = \{H(p_{i,j}) \in [0, 360], S(p_{i,j}) \in [0, 100], V(p_{i,j}) \in [0, 100]\}$. $D_{\text{Binary}} = \{p_{ij}, i = 1, \dots, h, j = 1, \dots, w\}$, $p_{ij} = 0$ или 1 — получаемое бинарное изображение.

Опишем этапы работы предлагаемой модификации метода исключения областей Exclusion:

Вход алгоритма: D_{HSV} — цветное изображение в схеме HSV.

Выход алгоритма: D_{Binary} — бинарное изображение.

Шаг 1. $i := 1; j := 1; SR_{\min} := 100; SG_{\min} := 100; SB_{\min} := 100; SR_{\max} := 0; SG_{\min} := 0; SB_{\min} := 0; D_{\text{Binary}} := \{0\}$.

Шаг 2. Если $H(d_{ij}) \in (HR_{\min}, HR_{\max})$ и $S(d_{ij}) < SR_{\min}$, то $SR_{\min} := S(d_{ij})$.

Шаг 3. Если $H(d_{ij}) \in (HR_{\min}, HR_{\max})$ и $S(d_{ij}) > SR_{\max}$, то $SR_{\max} := S(d_{ij})$; перейти на шаг 14.

Шаг 4. Если $H(d_{ij}) \in (HR_{\min}, HR_{\max})$ и $V(d_{ij}) < VR_{\min}$, то $VR_{\min} := V(d_{ij})$.

Шаг 5. Если $H(d_{ij}) \in (HR_{\min}, HR_{\max})$ и $V(d_{ij}) > VR_{\max}$, то $VR_{\max} := V(d_{ij})$; перейти на шаг 14.

Шаг 6. Если $H(d_{ij}) \in (HG_{\min}, HG_{\max})$ и $H(d_{ij}) < SG_{\min}$, то $SG_{\min} := H(d_{ij})$.

Шаг 7. Если $H(d_{ij}) \in (HG_{\min}, HG_{\max})$ и $H(d_{ij}) > SG_{\max}$, то $SG_{\max} := H(d_{ij})$; перейти на шаг 14.

Шаг 8. Если $H(d_{ij}) \in (HG_{\min}, HG_{\max})$ и $V(d_{ij}) < VG_{\min}$, то $VG_{\min} := V(d_{ij})$.

Шаг 9. Если $H(d_{ij}) \in (HG_{\min}, HG_{\max})$ и $V(d_{ij}) > VG_{\max}$, то $VG_{\max} := V(d_{ij})$; перейти на шаг 14.

Шаг 10. Если $H(d_{ij}) \in (HB_{\min}, HB_{\max})$ и $H(d_{ij}) < SB_{\min}$, то $SB_{\min} := H(d_{ij})$.

Шаг 11. Если $H(d_{ij}) \in (HB_{\min}, HB_{\max})$ и $H(d_{ij}) > SB_{\max}$, то $SB_{\max} := H(d_{ij})$; перейти на шаг 14.

Шаг 12. Если $H(d_{ij}) \in (HB_{\min}, HB_{\max})$ и $V(d_{ij}) < VB_{\min}$, то $VB_{\min} := V(d_{ij})$.

Шаг 13. Если $H(d_{ij}) \in (HB_{\min}, HB_{\max})$ и $V(d_{ij}) > VB_{\max}$, то $VB_{\max} := V(d_{ij})$; перейти на шаг 14.

Шаг 14. $j := j + 1$.

Шаг 15. Если $j \leq w$, то перейти на шаг 2.

Шаг 16. $i := i + 1; j := 1$.

Шаг 17. Если $i \leq h$, то перейти на шаг 2.

Шаг 18. $TSR := TS(SR); TVR := TV(VR); TSG := TS(SG); TVG := TV(VG); TSB := TS(SB); TVB := TV(VB); i := 1; j := 1$.

Шаг 19. Если $H(d_{ij}) \in (HR_{\min}, HR_{\max})$ и $S(d_{ij}) < TSR$ и $V(d_{ij}) > TVR$, то $p_{ij} := 0$; перейти на шаг 26.

Шаг 20. Если $H(d_{ij}) \in (HR_{\min}, HR_{\max})$ и $S(d_{ij}) > TSR$ и $V(d_{ij}) < TVR$, то $p_{ij} := 1$; перейти на шаг 26.

Шаг 21. Если $H(d_{ij}) \in (HG_{\min}, HG_{\max})$ и $S(d_{ij}) < TSG$ и $V(d_{ij}) > TVG$, то $p_{ij} := 0$; перейти на шаг 26.

Шаг 22. Если $H(d_{ij}) \in (HG_{\min}, HG_{\max})$ и $S(d_{ij}) > TSG$ и $V(d_{ij}) < TVG$, то $p_{ij} := 1$; перейти на шаг 26.

Шаг 23. Если $H(d_{ij}) \in (HB_{\min}, HB_{\max})$ и $S(d_{ij}) < TSB$ и $V(d_{ij}) > TVB$, то $p_{ij} := 0$; перейти на шаг 26.

Шаг 24. Если $H(d_{ij}) \in (HB_{\min}, HB_{\max})$ и $S(d_{ij}) > TSB$ и $V(d_{ij}) < TVB$, то $p_{ij} := 1$; перейти на шаг 26.

Шаг 25. $p_{ij} := 1$.

Шаг 26. $j := j + 1$.

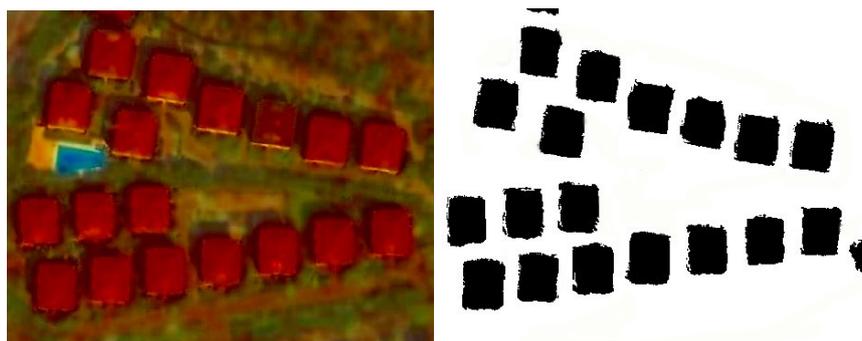
Шаг 27. Если $j \leq w$, то перейти на шаг 19.

Шаг 28. $i := i + 1$; $j := 1$.

Шаг 29. Если $i \leq h$, то перейти на шаг 19.

Шаг 30. Стоп.

На рис. 2 продемонстрирован результат сегментации изображения с помощью модифицированного метода исключения областей. На вход подается изображение в схеме HSV (рис. 2а). В результате работы алгоритма получаем бинарное изображение, представленное на рис. 2б.



а) Изображение в схеме HSV б) Бинаризованное изображение

Рис. 2. Сегментирование изображения методом исключения областей

3.2. Модифицированный метод жука

На втором этапе работы предложенного нами метода распознавания строений выполняется обнаружение объектов и выделение их контуров. Как уже было сказано ранее, для решения данной задачи используется разработанный нами модифицированный алгоритм жука. Метод жука — классический метод контурного анализа, предназначенный для изучения текстурных характеристик и особенностей изображений [13]. Его основная идея заключается в последовательном вычерчивании границы между объектом и фоном в бинаризованном изображении. Области изображения с черным цветом соответствуют объектам поиска, а с белым цветом — фону. Прослеживающая точка в виде «жука» движется по изображению до тех пор, пока не дойдет до темной области (объекта). Тогда «жук» поворачивается налево и движется вперед, пока не достигнет границ объекта, после этого поворачивается направо и повторяет процесс, пока не достигнет окрестности начальной точки. Координаты точек перехода с черного на белое и с белого на черное и описывают границу объекта [14]. Жук может перемещаться по четырем направлениям: влево, вверх, вправо, вниз.

В нашей работе предлагается модифицированная версия алгоритма жука, основанная на «маршруте обхода контура», т.е. списке координат пикселей изображения, посещенных «жуком» при обходе контура. Суть модификации метода состоит в следующем.

1. Для перемещения жука добавлены четыре диагональные направления, т.е. перемещения возможны по восьми направлениям: влево, влево-вверх, вверх, вверх-вправо, вправо, вниз-вправо, вниз, вниз-влево.
2. Добавлен «маршрут обхода контура», представляющий собой список координат точек, посещенных алгоритмом при поиске контура. В процессе обработки изображения вы-

полняется запись координат непроверенного пикселя в «маршрут», чтобы после он стал новой точкой отсчета движения жука. Окрестность нового пикселя верифицируется на наличие пикселей черного цвета — т.е. потенциальных точек для перемещения в них жука. Если пиксель не проходит проверку (т.е. из него никуда нельзя перейти), то он удаляется из «маршрута» жука и проверяется следующая точка (если она есть).

Добавленные диагональные направления позволяют алгоритму лучше справляться с объектами сложной формы, содержащие диагональные линии или изломы (классический алгоритм жука в случае, когда контур объекта содержал последовательность диагональных пикселей, упирался в первую диагональную точку и заканчивал свою работу). Также добавление новых направлений снизило вероятность получения «дыр» (областей белых пикселей) внутри объекта и многократного прохождения по одним и тем же пикселям, что повысило общую эффективность и скорость работы алгоритма. Использование «маршрута обхода контура» позволило алгоритму корректно возвращаться к ранее сохраненным точкам с несколькими вариантами пути и продолжать обход контуров в случае тупиков. Таким образом, алгоритм избегает заикливания и пропусков участков контура, обеспечивая более точное и полное распознавание объектов.

Введем несколько необходимых определений. Определим множество контуров обнаруженных объектов E следующим образом: $E = \{E_k\}$, где E_k — контур k -го объекта, $E_k = \{(i_t, j_t), t = 1, \dots, m\}$, $i_t \in \{1, \dots, h\}$, $j_t \in \{1, \dots, w\}$. Другими словами, E_k — множество точек (пикселей) контура выделенного объекта, где каждая точка задается координатами (i_t, j_t) , m — количество точек контура. $SP = \{(i_t, j_t)\}, t = 1, \dots, M$ — «маршрут обхода контура», представляющий собой последовательность координат точек, посещенных алгоритмом при поиске контура. $P = \{p_{ij}, i = 1, \dots, h, j = 1, \dots, w\}$ — копия изображения, используемая для работы алгоритма. Здесь h — высота изображения, w — ширина.

В переменной Dir хранится значение направления, в котором выполняется перемещение по точкам изображения в конкретный момент работы алгоритма. Приведем перечень допустимых значений, хранимых в данной переменной. Если значение Dir равно 1, то выполняется перемещение влево (из пикселя с координатами (i, j) в пиксель с координатами $(i, j - 1)$). Если $Dir = 2$, то выполняется перемещение влево-вверх (в пиксель с координатами $(i - 1, j - 1)$). Значение $Dir = 3$ соответствует перемещению вверх (в $(i - 1, j)$). $Dir = 4$ означает перемещение вверх-вправо (в $(i - 1, j + 1)$), $Dir = 5$ — перемещение вправо (в $(i, j + 1)$), $Dir = 6$ — перемещение вниз-вправо, $Dir = 7$ — перемещение вниз, $Dir = 8$ — перемещение вниз-влево. Переменная $Neighbors$ предназначена для хранения количества значимых пикселей (черного цвета), находящихся в окрестности текущей точки (с координатами (i, j)). В качестве окрестности точки (i, j) будут рассматриваться точки (t, s) , где $t = i - 1, \dots, i + 1, s = j - 1, \dots, j + 1, t \neq i, s \neq j$.

Опишем этапы работы предлагаемого алгоритма выделения контуров Contours:

Вход алгоритма: D_{binary} — бинарное изображение.

Выход алгоритма: E — множество контуров объектов.

Шаг 1. $E := \emptyset; P := D_{binary}; k := 1; i := 1; j := 1$.

Шаг 2. $E_k := \emptyset; SP := \emptyset; Dir := 1$.

Шаг 3. Если $(i, j) \notin E$ и $p_{ij} = 1$, то перейти на шаг 9.

Шаг 4. $j := j + 1$.

Шаг 5. Если $j \leq w$, то перейти на шаг 3.

Шаг 6. $i := i + 1; j := 1$.

Шаг 7. Если $i \leq h$, то перейти на шаг 3.

Шаг 8. Перейти на шаг 34.

Шаг 9. $SP := SP \cup \{(i, j)\}$.

Шаг 10. $Neighbors := \left(\sum_{t=i-1}^{i+1} \sum_{s=j-1}^{j+1} p_{ij} \right) - 1$.

Шаг 11. Если $Neighbors = 0$, то $SP := SP \setminus \{(i, j)\}$; $p_{ij} := 0$; перейти на шаг 4.

Шаг 12. Если $Neighbors < 8$, то $E_k := E_k \cup \{(i, j)\}$.

Шаг 13. Если $Dir = 5$ или $Dir = 6$ или $Dir = 7$ или $Dir = 8$, то перейти на шаг 22.

Шаг 14. Если $j > 1$ и $p_{ij-1} = 1$ и $(i, j-1) \notin SP$, то $j := j-1$; $Dir := 1$; перейти на шаг 9.

Шаг 15. Если $i > 1$ и $j > 1$ и $p_{i-1j-1} = 1$ и $(i-1, j-1) \notin SP$, то $i := i-1$; $j := j-1$; $Dir := 2$; перейти на шаг 9.

Шаг 16. Если $i > 1$ и $p_{i-1j} = 1$ и $(i-1, j) \notin SP$, то $i := i-1$; $Dir := 3$; перейти на шаг 9.

Шаг 17. Если $i > 1$ и $j < w$ и $p_{i-1j+1} = 1$ и $(i-1, j+1) \notin SP$, то $i := i-1$; $j := j+1$; $Dir := 4$; перейти на шаг 9.

Шаг 18. Если $j < w$ и $p_{ij+1} = 1$ и $(i, j+1) \notin SP$, то $j := j+1$; $Dir := 5$; перейти на шаг 9.

Шаг 19. Если $i < h$ и $j < w$ и $p_{i+1j+1} = 1$ и $(i+1, j+1) \notin SP$, то $i := i+1$; $j := j+1$; $Dir := 6$; перейти на шаг 9.

Шаг 20. Если $i < h$ и $p_{i+1j} = 1$ и $(i+1, j) \notin SP$, то $i := i+1$; $Dir := 7$; перейти на шаг 9.

Шаг 21. Если $i < h$ и $j > 1$ и $p_{i+1j-1} = 1$ и $(i+1, j-1) \notin SP$, то $i := i+1$; $j := j-1$; $Dir := 8$; перейти на шаг 9.

Шаг 22. Если $j < w$ и $p_{ij+1} = 1$ и $(i, j+1) \notin SP$, то $j := j+1$; $Dir := 5$; перейти на шаг 9.

Шаг 23. Если $i < h$ и $j < w$ и $p_{i+1j+1} = 1$ и $(i+1, j+1) \notin SP$, то $i := i+1$; $j := j+1$; $Dir := 6$; перейти на шаг 9.

Шаг 24. Если $i < h$ и $p_{i+1j} = 1$ и $(i+1, j) \notin SP$, то $i := i+1$; $Dir := 7$; перейти на шаг 9.

Шаг 25. Если $i < h$ и $j > 1$ и $p_{i+1j-1} = 1$ и $(i+1, j-1) \notin SP$, то $i := i+1$; $j := j-1$; $Dir := 8$; перейти на шаг 9.

Шаг 26. Если $j > 1$ и $p_{ij-1} = 1$ и $(i, j-1) \notin SP$, то $j := j-1$; $Dir := 1$; перейти на шаг 9.

Шаг 27. Если $i > 1$ и $j > 1$ и $p_{i-1j-1} = 1$ и $(i-1, j-1) \notin SP$, то $i := i-1$; $j := j-1$; $Dir := 2$; перейти на шаг 9.

Шаг 28. Если $i > 1$ и $p_{i-1j} = 1$ и $(i-1, j) \notin SP$, то $i := i-1$; $Dir := 3$; перейти на шаг 9.

Шаг 29. Если $i > 1$ и $j < w$ и $p_{i-1j+1} = 1$ и $(i-1, j+1) \notin SP$, то $i := i-1$; $j := j+1$; $Dir := 4$; перейти на шаг 9.

Шаг 30. $t := 1$;

Шаг 31. Если $t > |SP|$, то перейти на шаг 33.

Шаг 32. Для $(i_t, j_t) \in SP$ $p_{i_t j_t} := 0$; $t := t + 1$; перейти на шаг 31.

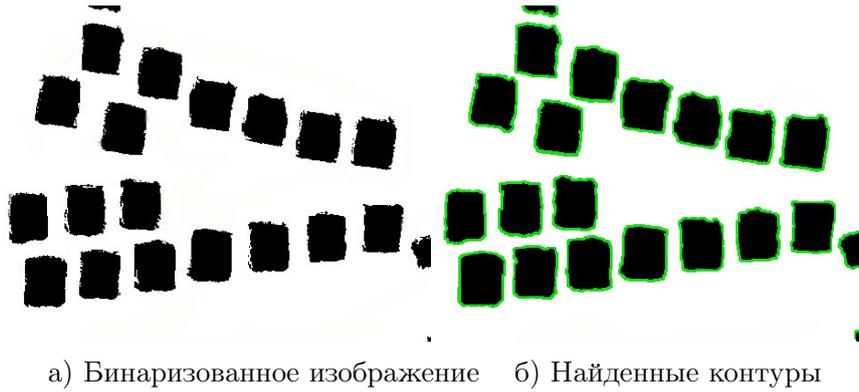
Шаг 33. $i := 1$; $j := 1$; $k := k + 1$; перейти на шаг 2.

Шаг 34. Стоп.

Результаты работы предлагаемого модифицированного метода жука по нахождению контуров объектов демонстрируются на рис. 3.

3.3. Алгоритм проверки искусственного происхождения объекта

После обнаружения объектов, претендующих на отнесение к зданиям и строениям, необходимо проверить, являются ли они искусственными объектами. Как уже было сказано ранее, искусственно созданные объекты имеют особенности формы: малую постоянную кривизну (объекты круглой формы), либо несколько точек высокой кривизны (объекты



а) Бинаризованное изображение б) Найденные контуры

Рис. 3. Поиск контуров модифицируемым методом жука

прямоугольной формы) или вообще не имеют кривизны. Нами был предложен алгоритм проверки объекта на искусственное происхождение, проверяющий выполнение указанных свойств для каждого объекта, обнаруженного ранее методом жука. Приведем описание данного алгоритма и необходимые определения.

Пусть дано E_k — множество точек (пикселей) контура объекта, где каждая точка задается его координатами (i_t, j_t) , а m — количество точек контура. Если поверхность в евклидовом пространстве задана параметрически кусочно C^{-1} -гладкой функцией $r(u, v)$, где параметры u, v изменяются в области E_k на плоскости (u, v) , то площадь $Area(E_k)$ находится по следующей формуле: $Area(E_k) = \int_{E_k} |r_u \times r_v| du dv$, где \times обозначает векторное произведение, а r_u и r_v — частные производные по u и v .

Средняя кривизна области, ограниченной контуром E_k , вычисляется следующим образом: $CU(E_k) = \frac{\oint_{E_k} r ds}{|E_k|}$, где r — радиус локальной кривизны области. Для нахождения радиуса кривизны воспользуемся подходом, используемым в обобщенном преобразовании Хафа, предназначенным для поиска объектов, принадлежащих определенному классу фигур. В данном случае нас будут интересовать фигуры прямоугольной и округлой формы. Главная идея преобразования Хафа для нахождения прямой линии: представление прямой с помощью параметров r и θ , где параметр r — длина радиус-вектора ближайшей к началу координат точки на прямой (нормали к прямой, проведенной из начала координат), а θ — угол между этим вектором и осью абсцисс. Тогда уравнение прямой можно записать $y = (-\frac{\cos \theta}{\sin \theta}) x + (\frac{r}{\sin \theta})$ или после преобразования $r = x \cdot \cos \theta + y \cdot \sin \theta$. Пусть прямая задается уравнением $y = kx + b$. Тогда $k = -\frac{\cos \theta}{\sin \theta}$, а $r = \frac{r}{\sin \theta}$. Следовательно, $r = b \sin \theta$, где $\theta = \arctan(-1/k)$. Мы можем найти значения k и b по двум соседним точкам $(x_1, y_1), (x_2, y_2) \in E_k$: $k = (y_2 - y_1)/(x_2 - x_1)$, $b = y_1 - x_1 \cdot ((y_2 - y_1)/(x_2 - x_1))$. Тогда средняя кривизна области, ограниченной контуром E_k , считая ее прямоугольной фигурой (ограниченной прямыми линиями), вычисляется так:

$$CU_{\text{square}}(E_k) = \frac{\sum_{(x_1, y_1) \in E_k} r_{x_1 y_1}}{|E_k|},$$

где $r_{x_1 y_1} = b \sin(\arctan(-1/k))$, где k и b находятся по формулам, указанным выше, а (x_2, y_2) — точка, следующая за (x_1, y_1) во множестве E_k .

Для определения кривизны округлой фигуры воспользуемся идеей, предложенной в преобразовании Хафа для нахождения окружностей. Пространство всех возможных окружностей, проходящих через точку, состоит из трех измерений: координат центра (x, y) и

радиуса возможной окружности R . Уравнение множества всех центров окружностей с радиусом R , проходящих через точку (x_0, y_0) : $(x - x_0)^2 + (y - y_0)^2 = R^2$. Если перебрать все точки $(x, y) \in E_k$, для каждой точки найти радиус окружности R , который можно через нее провести, а затем усреднить полученные значения, то будет найдена средняя кривизна области при предположении, что она является округлой фигурой:

$$CU_{\text{circle}}(E_k) = \frac{\sum_{(x,y) \in E_k} R}{|E_k|},$$

где $R = \sqrt{(x - x_0)^2 + (y - y_0)^2}$, а (x_0, y_0) — точка, следующая за (x, y) во множестве E_k .

Нами были вычислены значения функции $CU_{\text{circle}}(E_k)$ для окружности и значения $CU_{\text{square}}(E_k)$ для эталонных фигур прямоугольной формы (прямоугольной, Г-образной, Н-образной и т.д.) Введем переменную α для хранения порогового значения при определении округлости контура и β — пороговое значение при определении прямоугольной формы контура, $\alpha = 19/20$, $\beta = 9/20$. Также введем переменную $curv1$ для хранения вычисленного значения средней кривизны области, ограниченной контуром E_k , для прямоугольной формы, переменную $curv2$ — для хранения значения кривизны округлой формы, ϵ — допустимая погрешность при сравнении контуров с эталонами.

Опишем этапы работы предлагаемого алгоритма проверки объекта на искусственное происхождение Artificials:

Вход алгоритма: E — множество контуров объектов.

Выход алгоритма: EA — множество контуров искусственных объектов.

Шаг 1. $k := 1; R := 0; \alpha := 19/20; \beta := 9/20; EA := E; \epsilon := 1/10$.

Шаг 2. Для $E_k \in E$ $curv1 := CU_{\text{circle}}(E_k); curv2 := CU_{\text{square}}(E_k)$.

Шаг 3. Если $curv1 > \alpha - \epsilon$ и $curv1 < \alpha + \epsilon$, то перейти на шаг 7.

Шаг 4. Если $curv2 > \beta - \epsilon$ или $curv1 < \beta + \epsilon$, то перейти на шаг 7.

Шаг 5. Если $\frac{Area(E \cap E_k)}{Area(E_k)} > 1/20$, то перейти на шаг 7.

Шаг 6. $EA := EA \setminus \overline{E_k}$.

Шаг 7. $k := k + 1$.

Шаг 8. Если $k \leq |E_k|$, то перейти на шаг 2.

Шаг 9. Стоп.

Шаг 5 алгоритма реализует дополнительную проверку для строений, ограниченных прямыми линиями, но не являющихся прямоугольными, чтобы обойти исключение из списка найденных объектов зданий с Г- или Н-образными частями.

3.4. Алгоритм проверки объекта на объемность

Предыдущий алгоритм позволил обнаружить на изображении объекты искусственного происхождения. В рамках нашей задачи требуется находить строения. Очевидно, что существует ряд искусственных плоскостных сооружений, которые при отображении на двумерном рисунке будут похожи на строения (например, парковки автомашин, дороги, линии электропередач и другие). Единственным отличием таких объектов от строений является то, что у них нет объема. Для устранения данной проблемы, нами был разработан алгоритм проверки объекта на то, что у него есть объем. Как уже было сказано выше, признаком того, что объект имеет объем, является наличие у него на спутниковом снимке тени. Помимо указанного способа отличения объемных объектов от плоскостных в алгоритме предполагается использовать функцию выпуклости области C_0 , которую можно задать следующим

образом: $C_0(E_k) = \frac{E_k}{Area(E_k)}$, где E_k — множество точек контура объектов, $Area(E_k)$ — площадь фигуры, ограниченной контуром E_k .

Пусть D — исходное изображение, $E_k = \{(i_t, j_t), t = 1, \dots, m\}$, $i_t \in 1, \dots, h$, $j_t \in 1, \dots, w$, RGB — множество оттенков, для которых $R = \{0, \dots, 55\}$, $G = \{0, \dots, 55\}$, $B = \{0, \dots, 55\}$. Для выявления цветовых значений в пикселе воспользуемся функцией $RGB(p_{ij})$, которая возвращает $R(p_{ij})$, $G(p_{ij})$, $B(p_{ij})$. \bar{D} — промежуточное изображение для поиска области тени. Определим функцию $invert(D)$, которая для каждого пикселя изображения D заменяет пиксели в формате RGB на значения $255 - p_{ij}$ в случае, если $p_{ij} \in E_k$. Функция $\max(D_1, D_2)$ вычисляет максимальное значение тона пикселя следующим образом: $\max(RGB(p_{ij}), RGB(\bar{p}_{ij}))$, $p_{ij} \in D$, $\bar{p}_{ij} \in \bar{D}$. Введем переменную $convexity$ для хранения вычисленного значения функции $C_0(E_k)$.

Опишем этапы работы предлагаемого алгоритма проверки объекта на объемность VolumeObjects:

Вход алгоритма: EA — множество контуров искусственных объектов.

Выход алгоритма: EV — множество контуров объемных объектов.

Шаг 1. $k := 1$; $EV := \emptyset$; $D_{invert} := invert(D)$; $\bar{D} := \max(D, D_{invert})$.

Шаг 2. Для $E_k \in EA$ $convexity := C_0(E_k)$.

Шаг 3. Если $convexity \geq 0$, то перейти на шаг 8.

Шаг 4. $EV := EV \cup E_k$.

Шаг 5. $k := k + 1$.

Шаг 6. Если $k > |E_k|$, то перейти на шаг 19.

Шаг 7. Перейти на шаг 2.

Шаг 8. $t := 1$.

Шаг 9. Если $t > |E_k|$, то перейти на шаг 5.

Шаг 10. $Neighbors := 0$.

Шаг 11. $i := i_t - 1$; $j := j_t - 1$.

Шаг 12. Для $p_{ij} \in \bar{D}$ $r := R(p_{ij})$, $g := G(p_{ij})$, $b := B(p_{ij})$.

Шаг 13. Если $r \in (0, 55) \cup g \in (0, 55) \cup b \in (0, 55)$, то $Neighbors := Neighbors + 1$.

Шаг 14. $j := j + 1$.

Шаг 15. Если $j \leq j_t + 1$, то перейти на шаг 12.

Шаг 16. $i := i_t + 1$; $j := j_t - 1$.

Шаг 17. Если $i \leq i_t + 1$, то перейти на шаг 12.

Шаг 18. Если $Neighbors \geq \frac{|E_k|}{4}$, то перейти на шаг 4.

Шаг 19. Стоп.

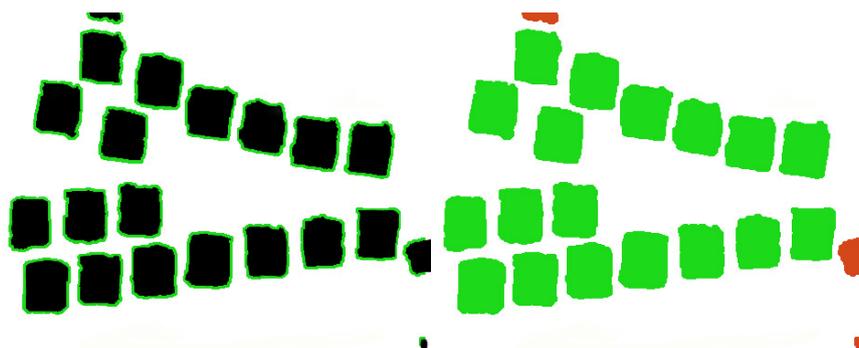
На рис. 4 демонстрируются результаты работы предлагаемого алгоритма проверки объекта на объемность. На рис. 4а приводится изображение с контурами искусственных объектов, подаваемое на вход алгоритму. В результате работы алгоритма получаем изображение с контурами объемных объектов, представленное на рис. 4б.

3.5. Последовательность этапов предложенного гибридного алгоритма распознавания строений

Таким образом, можно записать последовательность этапов работы предложенного гибридного метода распознавания строений следующим образом:

Вход алгоритма: D_{HSV} — цветное изображение в схеме HSV.

Выход алгоритма: EV — множество контуров искусственных объемных объектов.



а) Изображение с контурами искусственных объектов б) Контурные объемных строений

Рис. 4. Выявление искусственных объемных объектов

Шаг 1. $D_{\text{Binary}} = \text{Exclusion}(D_{\text{HSV}})$.

Шаг 2. $E = \text{Contours}(D_{\text{Binary}})$.

Шаг 3. $EA = \text{Artificials}(E)$.

Шаг 4. $EV = \text{VolumeObjects}(EA)$.

Шаг 5. Стоп.

В результате работы алгоритма на изображении будет найдено множество контуров искусственных объемных объектов EV , которое как раз и представляет собой множество контуров строений, а, следовательно, является решением поставленной задачи распознавания строений на спутниковых снимках.

4. Программная реализация

Для решения поставленной в работе задачи распознавания было разработано программное обеспечение «Распознавание строений» на языке Python с использованием библиотек OpenCV, Rasterio, Matplotlib, NumPy. Программа состоит из следующих модулей: предобработки изображения, фильтрации, распознавания строений и вывода результата. После загрузки изображения модуль предобработки преобразует изображение к определенному размеру (720×720 пикселей), что позволяет установить на изображении метрическую систему для дальнейшего определения площади объектов. В модуле фильтрации реализованы ряд фильтров и преобразований изображений для улучшения качества изображения, устранения шумов и повышения контрастности: метод сдвига среднего значения, метод скалирования изображения, преобразование изображения в HSV-представление. После этого выполняется модуль распознавания, который реализует предлагаемый в работе гибридный алгоритм распознавания строений. Вид созданного программного обеспечения представлен на рис. 5.

Описанный алгоритм определяет на изображении строения (выделяются на рисунке зеленым цветом). Кроме того, приложение отображает ложно распознанные объекты (искусственные объекты, не прошедшие проверку на объем или слишком малые по площади). Они изображаются на рисунке красным цветом.

5. Результаты проведенных вычислительных экспериментов

В работе были проведены вычислительные эксперименты по оцениванию эффективности предложенного метода, а также его сравнению с тремя известными алгоритмами распознавания. В качестве алгоритмов для сравнения были выбраны классический метод

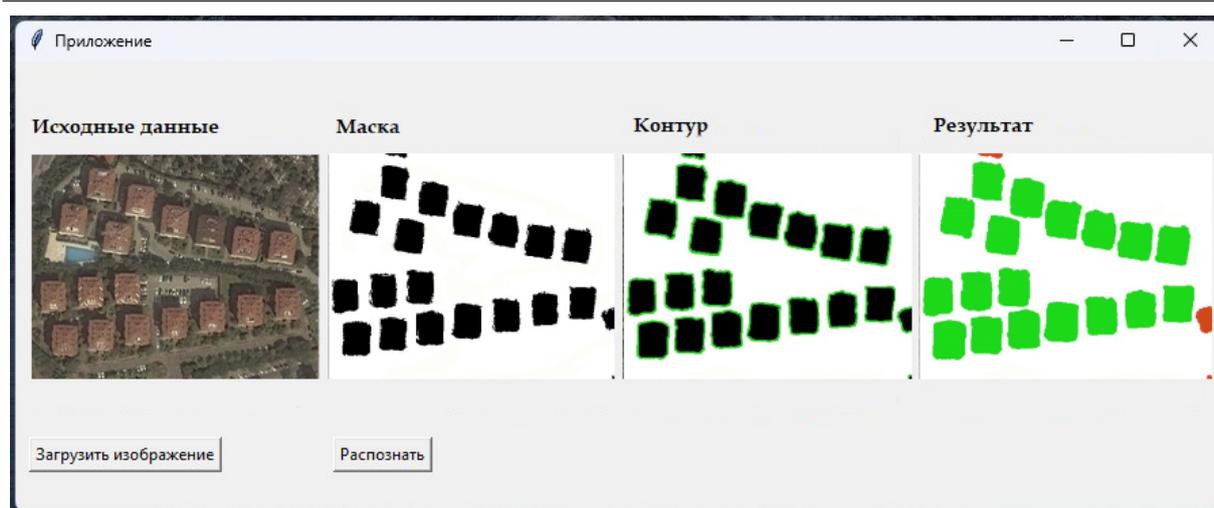


Рис. 5. Вид разработанного приложения

исключения областей [12], метод активных контуров [7] и нейронная сеть с архитектурой UANet + PVT-V2-B2, которая на данный момент является одной из лучших нейросетей обнаружения строений [6]. Сравнение алгоритмов выполнялось на основе известного датасета Inria, представляющего собой набор размеченных спутниковых снимков зданий в формате RGB. Указанный датасет является одним из известных открытых наборов изображений, на которых производят сравнение алгоритмов семантической сегментации строений. Набор данных Inria building содержит 1800 изображений, собранных из пяти городов (Остин, Чикаго, Китсап, Тироль и Ви-Энна), по 360 изображений на каждый город.

Под эффективностью алгоритма будет пониматься качество распознавания и время работы алгоритма. Как уже было сказано в разделе 2.2, в работе в качестве критерия качества распознавания используются четыре метрики: точность, полнота, усредненный индекс Жаккара и мера F1, представляющие собой функции от количества ошибок первого и второго рода [10]. Понятия ошибок первого и второго рода были приведены выше, в разделе 2.2. Под точностью (Precision) понимается доля объектов, действительно принадлежащих данному классу, относительно всех объектов, которые алгоритм отнес к этому классу. Точность вычисляется по формуле $Precision = \frac{TP}{TP+FP}$, где TP — истинно положительные значения, FP — ложно положительные значения, т.е. значения ошибки первого рода. Полнота (Recall) показывает, какую долю объектов класса из всех объектов, действительно относящихся к этому классу, нашел алгоритм: $Recall = \frac{TP}{TP+FN}$, где FN — ложно отрицательные значения (ошибки второго рода). Индекс Жаккара или метрика IoU (Intersection over Union) — метрика степени пересечения между двумя ограничивающими рамками: $IoU = \frac{TP}{TP+FP+FN}$. Мера F1 (F1-Score) — мера, сочетающая точность и полноту: $F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$.

Для обучения нейросети дополнительно использовалась обучающая выборка из 30000 изображений. Вычислительные эксперименты проводились на компьютере с AMD Ryzen 5 2600 Six-Core Processor (8 MB Cache, 3.40 ГГц) и ОЗУ объемом 16 Гбайт. В таблице в качестве времени обучения и распознавания приводится среднее время для обработки одного изображения.

Из табл. 1 видно, что нейронная сеть показала наилучшую скорость работы и при этом имеет хорошую точность распознавания. Однако стоит напомнить, что нейронную сеть необходимо предварительно обучить на достаточно объемной выборке. Процесс обучения

Таблица 1. Сравнение работы алгоритмов

Метод	Время, с		Precision	IoU	Recall	F1
	Обуч.	Расп.				
Метод исключения областей [12]	0.1	27	83.35	71.30	82.41	82.87
Метод активных контуров [7]	–	90	92.10	79.93	90.85	91.47
UUNET [6]	0.12	5	86.39	74.40	84.28	85.32
Гибридный алгоритм	–	15	87.28	76.44	85.37	86.31

трудоемок и занимает значительное время. Предложенный нами гибридный алгоритм не требует предварительного обучения и лишь незначительно проигрывает по скорости уже обученной нейросети, что, несомненно, показывает его преимущество. Наилучшую точность распознавания продемонстрировал метод активных контуров, однако он имеет наименьшую скорость работы и чувствительность к качеству изображения. Предложенный же нами метод показал второй результат по точности распознавания при явном выигрыше в скорости работы. Такая же ситуация наблюдается и по остальным метрикам качества распознавания: метод активных контуров занимает первое место, а разработанный нами гибридный метод является вторым.

Таким образом, можно утверждать, что предложенный алгоритм распознавания имеет хорошее быстроедействие и высокую эффективность в распознавании зданий на спутниковых снимках. Полученные показатели эффективности работы метода позволяют успешно решать с его помощью задачи мониторинга городской застройки, обнаружения изменений в инфраструктуре, обнаружения незаконно возведенных строений, анализа сельскохозяйственных угодий, исследования лесных ресурсов. Представленные задачи не требуют распознавания в реальном времени и чаще всего решаются в режиме оффлайн (в отличие от ряда задач, имеющих существенные требования к скорости распознавания — например, при распознавании дорожных знаков и препятствий на дороге), поэтому скорость распознавания, продемонстрированная нашим алгоритмом, является достаточной.

Выявленное в результате экспериментов преимущество метода активных контуров в качестве распознавания стало интересным результатом, который требует дальнейшего исследования. К числу известных недостатков данного алгоритма относятся чувствительность к качеству и размеру изображений, поэтому мы планируем провести еще серию экспериментов по сравнению всех методов при первоначальном преобразовании входных изображений больших размеров.

Заключение

В данной статье предложен новый метод распознавания строений на спутниковых снимках, являющийся модифицированным гибридом алгоритма исключения областей и метода жука. Распознавание строений выполняется на цветных спутниковых снимках формата RGB. Перед работой алгоритма распознавания происходит предобработка и фильтрация снимков с помощью метода скалирования изображения, сдвига среднего значения и перевода в цветовую схему HSV. Для решения поставленной задачи предложенный гибридный метод распознавания строений выполняет следующие действия: сегментацию изображения на регионы схожих пикселей; обнаружение объектов и выделение их контуров; выявление среди потенциальных объектов искусственно созданных объектов; обнаружение среди них

объемных объектов. Для сегментации изображения разработан модифицированный метод исключения областей, работающий с цветным изображением в схеме HSV и выполняющий пороговую сегментацию отдельно для каждого цветового тона (красного, зеленого и синего) по яркости и насыщенности. Для обнаружения объектов и выделения контуров в работе предложен модифицированный алгоритм жука, основанный на «маршруте обхода контура». Для перемещения жука добавлены четыре диагональных направления, которые позволяют алгоритму лучше справляться с объектами сложной формы и избегать заикливания и пропусков участков контура. Использование «маршрута обхода контура» позволило существенно ускорить работу алгоритма. Для выявления искусственно созданных объектов разработан собственный алгоритм проверки контуров объектов на кривизну, а для выявления среди искусственных объектов строений — алгоритм проверки объекта на объемность. Разработано программное обеспечение, реализующее работу предложенного алгоритма распознавания и вышеперечисленных методов предобработки и фильтрации. Проведены вычислительные эксперименты по оцениванию эффективности предложенного метода, а также его сравнению с тремя известными алгоритмами распознавания: классическим методом исключения областей, методом активных контуров и нейронной сетью с архитектурой UANet. Сравнение алгоритмов выполнялось на основе известного датасета Ingia. Предложенный гибридный алгоритм показал хорошую скорость и точность распознавания. Одним из самых важных преимуществ разработанного алгоритма является то, что он не требует наличия обучающей выборки и предварительного обучения. Также необходимо отметить автоматическую настройку параметров гибридного алгоритма. В рамках дальнейших исследований мы планируем провести вычислительные эксперименты:

- 1) по исследованию чувствительности предложенного метода к качеству и разрешению исходных спутниковых изображений;
- 2) по исследованию чувствительности к качеству и разрешению исходных спутниковых изображений трех перечисленных ранее алгоритмов распознавания;
- 3) по сравнению алгоритмов на основе других открытых наборов размеченных спутниковых снимков.

Литература

1. Гилин С.В. Задача автоматического распознавания зданий в водоохраных зонах на спутниковых снимках // Информационные технологии и математическое моделирование (ИТММ-2021): Материалы XX Международной конференции имени А.Ф. Терпугова, Томск, 1–5 декабря, 2021. Томск: Национальный исследовательский Томский государственный университет, 2022. С. 6–12.
2. Захаров А.А., Тужилкин А.Ю. Сегментация спутниковых изображений на основе суперпикселей и разрезов на графах // Программные системы и вычислительные методы. 2018. № 1. С. 7–17. DOI: 10.7256/2454-0714.2018.1.25629.
3. Фукунага К. Введение в статистическую теорию распознавания образов. Москва: Наука, 2009. 368 с.
4. Горбачев В.А., Криворотов И.А., Маркелов А.О., Котлярова Е.В. Семантическая сегментация спутниковых снимков аэропортов с помощью сверточных нейронных сетей // Компьютерная оптика. 2020. Т. 44, № 4. С. 636–645. DOI: 10.18287/2412-6179-СО-636.
5. Shahoud A., Shashev D., Shidlovskiy S. Detection of Good Matching Areas Using

- Convolutional Neural Networks in Scene Matching-Based Navigation Systems // Ежегодная международная конференция по компьютерной графике и машинному зрению ГрафиКон (GraphiCon-2021): Труды Международной конференции по компьютерной графике и машинному зрению «Графикон», Томск, 27–30 сентября, 2021. Москва: Институт прикладной математики имени М.В. Келдыша РАН, 2021. С. 443–452.
6. He W., Li J., Cao W., Zhang L., Zhang H. Building Extraction from Remote Sensing Images via an Uncertainty-Aware Network. URL: <https://arxiv.org/pdf/2307.12309> (дата обращения: 15.02.2024).
 7. Сорокин Д.В., Крылов А.С. Метод активных контуров для сегментации изображений. Москва: ООО «МАКС Пресс», 2022. 116 с.
 8. Чернов А.В., Чупшев Н.В. Автоматическое распознавание контуров зданий на картографических изображениях // Компьютерная оптика. 2007. Т. 31, № 4. С. 636–645.
 9. Панченко Д.С., Путятин Е.П. Сравнительный анализ методов сегментации изображений // Радиоэлектроника и информатика. 1999. Т. 4. С. 109–114.
 10. Воронцов К.В. Математические методы обучения по прецедентам. URL: <http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf> (дата обращения: 10.09.2023).
 11. Xu Yu., Wang K., Liu S., Yang S., Yan B. Atmospheric correction of hyperspectral data using MODTRAN model // Remote Sensing of the Environment: 16th National Symposium on Remote Sensing of China. 2008. Vol. 7123. P. 1–7. DOI: 10.1117/12.815552.
 12. Голуб Ю.И., Старовойтов В.В., Коноплин Е.Е. Сегментация примерно однородных по яркости областей // Искусственный интеллект. 2008. Т. 3. С. 332–338.
 13. Фурман Я.А., Кревецкий А.В., Передреев А.К. Введение в контурный анализ и его приложения к обработке изображений и сигналов. Москва: ФИЗМАТЛИЗ, 2002. 592 с.
 14. Сакович И.О., Белов Ю.С. Обзор основных методов контурного анализа для выделения контуров движущихся объектов // Инженерный журнал: наука и инновации. 2014. Т. 36, № 12. С. 1–11.

Баранова Ирина Владимировна, к.ф.-м.н., доцент, базовая кафедра вычислительных и информационных технологий, Институт математики и фундаментальной информатики Сибирского федерального университета (Красноярск, Российская Федерация)

Гилян Степан Валентинович, аспирант, базовая кафедра вычислительных и информационных технологий, Институт математики и фундаментальной информатики Сибирского федерального университета (Красноярск, Российская Федерация)

BUILDING RECOGNITION HYBRID ALGORITHM FOR SATELLITE IMAGES BASED ON THE BEETLE METHOD AND THE AREA EXCLUSION ALGORITHM

© 2024 I.V. Baranova, S.V. Gilin

Siberian Federal University

(660041 Krasnoyarsk, pr. Svobodny, 79)

E-mail: ibaranova@sfu-kras.ru, gilin.stepan@mail.ru

Received: 10.01.2024

The article proposes a new method for recognizing buildings on satellite images. The proposed method is a hybrid, it is based on the region exclusion algorithm and the beetle method. The region exclusion algorithm is a well-known and effective approach to object detection on the image. Its main idea is to segment an image into regions of similar pixels based on various characteristics: color, texture, brightness, shape, etc. The beetle method is a classic contour analysis method that sequentially draws the boundary between an object and its background. As part of the proposed method, the beetle method first identifies potential areas where buildings may be located. The region exclusion method then eliminates unwanted elements in the image (vegetation, water surfaces and roads) that could be falsely identified as buildings, and accurately determines the location and outline of buildings. The offered algorithm shows good recognition accuracy regardless of image quality and does not require a training sample. The article also describes the software implementation of the proposed method and discusses the results of computational experiments to assess the quality of the method and compare it with three well-known recognition algorithms.

Keywords: building recognition, satellite imagery, hybrid method, beetle method, area exclusion method, textural characteristics, building boundaries, potential areas, classification.

FOR CITATION

Baranova I.V., Gilin S.V. Building Recognition Hybrid Algorithm for Satellite Images Based on the Beetle Method and the Area Exclusion Algorithm. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2024. Vol. 13, no. 2. P. 56–76. (in Russian) DOI: 10.14529/cmse240204.

References

1. Gilin S.V. The task of automatic recognition of buildings in water protection zones on satellite images. *Information Technologies and Mathematical Modeling (ITMM-2021): Proceedings of the XX International Conference named after A.F. Terpugov, Tomsk, Russia, December 1–5, 2021*. Tomsk: National Research Tomsk State University, 2022. P. 6–12. (in Russian)
2. Zaharov A.A., Tujilkin A.Y. Segmentation of satellite images based on super pixels and graph sections. *Software systems and computational methods*. 2018. No. 1. P. 7–17. (in Russian) DOI: 10.7256/2454-0714.2018.1.25629.
3. Fukunaga K. *An introduction to the statistical theory of pattern recognition*. Moscow: Science, 2009. 368 p. (in Russian)
4. Gorbachev V.A., Krivorotov I.A., Markelov A.O., Kotlyarova E.V. Semantic segmentation of satellite images of airports using convolutional neural networks. *Computer optics*. 2020. Vol. 44, no. 4. P. 636–645. (in Russian) DOI: 10.18287/2412-6179-CO-636.
5. Shahoud A., Shashev D., Shidlovskiy S. Detection of Good Matching Areas Using Convolutional Neural Networks in Scene Matching-Based Navigation Systems. *Annual*

- International Conference on Computer Graphics and Machine Vision GraphiCon (GraphiCon-2021): Proceedings of the International Conference on Computer Graphics and Machine Vision “Graphicon”, Tomsk, Russia, September 27–30, 2021. Moscow, M.V. Keldysh Institute of Applied Mathematics of the Russian Academy of Sciences, 2021. P. 443–452. (in Russian)
6. He W., Li J., Cao W., Zhang L., Zhang H. Building Extraction from Remote Sensing Images via an Uncertainty-Aware Network. URL: <https://arxiv.org/pdf/2307.12309> (accessed: 15.02.2024).
 7. Sorokin D.V., Krylov A.S. The active contour method for image segmentation. Moscow: LTD “MAX Press”, 2022. 116 p. (in Russian)
 8. Chernov A.V., Chupshev N.V. Automatic detection of building contours in cartographic images. Computer optics. 2007. Vol. 31, no. 4. P. 636–645. (in Russian)
 9. Panchenko D.S., Putyanin E.P. Comparative analysis of image segmentation methods. Radio electronics and computer science. 1999. Vol. 4. P. 109–114. (in Russian)
 10. Vorontsov K.V. Mathematical methods of teaching by use cases. URL: <http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf> (accessed: 10.09.2023). (in Russian)
 11. Xu Yu., Wang K., Liu S., Yang S., Yan B. Atmospheric correction of hyperspectral data using MODTRAN model. Remote Sensing of the Environment: 16th National Symposium on Remote Sensing of China. 2008. Vol. 7123. P. 1–7. DOI: 10.1117/12.815552.
 12. Golub Y.I., Starovoytov V.V., Konoplin E.E. Segmentation of areas with approximately uniform brightness. Artificial intelligence. 2008. Vol. 3. P. 332–338. (in Russian)
 13. Furman Y.A., Kreveckiy A.V., Peredreev A.K. An introduction to contour analysis and its applications to image and signal processing. Moscow: PHYSICAL ANALYSIS, 2002. 592 p. (in Russian)
 14. Sakovich I.O., Belov Y.S. Overview of the main methods of contour analysis for highlighting the contours of moving objects. Engineering Journal: Science and Innovation. 2014. Vol. 36, no. 12. P. 1–11. (in Russian)