

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
Кафедра компьютерных технологий и систем

**Авдюшкина Людмила Сергеевна**

**Выпускная квалификационная работа бакалавра**

**Применение текстурной сегментации для  
определения типа местности на  
аэрофотоснимках**

Направление 010300

Фундаментальная информатика и информационные технологии

Научный руководитель,  
доктор физ.-мат. наук,  
профессор  
Веремей Е.И.

Санкт-Петербург  
2016

# Содержание

<b>Введение</b> . . . . .	<b>3</b>
<b>Постановка задачи</b> . . . . .	<b>5</b>
<b>Обзор литературы</b> . . . . .	<b>9</b>
<b>Глава 1. Сегментация изображения</b> . . . . .	<b>13</b>
1.1. Алгоритм наращивания областей блоками . . . . .	13
1.1.1. Форма блоков и связность . . . . .	13
1.1.2. Алгоритм наращивания блоками единственной области	16
1.2. Автоматический выбор начальных областей . . . . .	18
1.2.1. Подходы к выбору начальных областей . . . . .	18
1.2.2. Алгоритм автоматической сегментации . . . . .	21
1.3. Объединение областей малой площади . . . . .	23
1.3.1. Алгоритм объединения областей . . . . .	25
<b>Глава 2. Текстульная классификация</b> . . . . .	<b>27</b>
2.1. Библиотека текстур . . . . .	27
2.2. LBP-оператор . . . . .	30
2.3. Дескрипторы текстур . . . . .	31
2.4. Двухшаговый алгоритм классификации . . . . .	33
<b>Выводы</b> . . . . .	<b>36</b>
<b>Заключение</b> . . . . .	<b>37</b>
<b>Список литературы</b> . . . . .	<b>38</b>

## Введение

Задача выделения объектов на фото - одна из первых задач анализа цифровых снимков. В работе большинства алгоритмов выделения объектов используется обнаружение их контуров. Легко заметить, что контуры являются границами областей, для внутренних элементов которых выполняется условие однородности по какому-либо признаку. В случае растровых изображений можно сказать, что каждая область - это набор точек (пикселей), имеющих одинаковые свойства (например яркость), по которым они и объединены. Таким образом, появилась возможность выделять отдельный объект по его текстуре.

В текстурах отражены индивидуальные характеристики поверхности, такие как способность отражать или рассеивать свет, однородность. Эту особенность текстур активно используют в 3D-моделировании, где сначала создают «пластилиновый» макет, все части которого одинаковы, а затем накладывают на отдельные части соответствующие изображения текстуры. Однако это только пример использования текстур для задания свойств объекта. Более интересна задача текстурного анализа изображения.

Две наиболее крупные области применения текстурного анализа - это медицина и картография. Медики с помощью него способны выявить опухоли или патологии на медицинских снимках, а так же определить объемы тканей и органов. Чаще всего он применяется для выявления злокачественных новообразований на рентгеновских снимках. Однако на сегодняшний день такой подход не достаточно точен для того, чтобы полностью на него положиться и убрать человеческий фактор при анализе рентгеновских снимков. Но он получил широкое распространение в качестве вспомогательного механизма диагностики.

Стартовой площадкой для развития первых методов анализа аэрофотоснимков стала военная сфера. Именно в ней впервые начали массово использовать алгоритмы распознавания изображений. Примером этого может служить обнаружение вражеских позиций. Позднее задача обнаружения отдельных объектов переросла в задачу комплексного анализа фотоснимка, после чего, как это часто бывает, она распространилась и на гражданскую сферу. На сегодняшний день анализ спутниковых снимков и аэрофотоснимков широко используется в картографии. К сожалению, при всём уровне развития технологий, полностью автономного метода, гарантирующего желаемую точность преобразования фото в карту разработано не было, но существующие алгоритмы являются неотъемлемыми инструментами, существенно экономящими время любого современного картографа.

В картографии текстурный анализ является одним из средств автоматического создания карты из аэрофотоснимка. Текстурный анализ снимка - далеко не тривиальная задача. Первым шагом обработки обычно является сегментация фото на области разного типа. Однако требуется не просто разбить изображение на области, но и выявить аномалии - места, которые явным образом выделяются из однородного фона. Кроме того, в результате области со схожими свойствами должны быть и отмечены схожим образом. Для этого посредством анализа происходит выделение характерных свойств каждой области. Этот аспект задачи и рассматривается текстурной классификацией.

Основной сложностью задачи текстурного анализа аэрофотоснимков является подбор методов сегментации и текстурной идентификации, учитывающих особенности данной предметной области и подбор дескрипторов, максимально точно описывающих именно текстуры природного происхождения.

## Постановка задачи

Определим основные термины, необходимые для постановки задачи.

**Определение 1.** Будем называть *текстурой* изображение, состоящее из произвольных повторяющихся элементов и отражающее какие-либо визуальные свойства физического объекта. По типу и размеру примитивов текстуры можно разделить на *регулярные*, когда структурные элементы текстуры распределены равномерно, *нерегулярные*, когда элементы расположены неравномерно и *фрактальные*, когда структурные элементы образуют фрактал. На рис. 1 приведены примеры различных типов текстур.

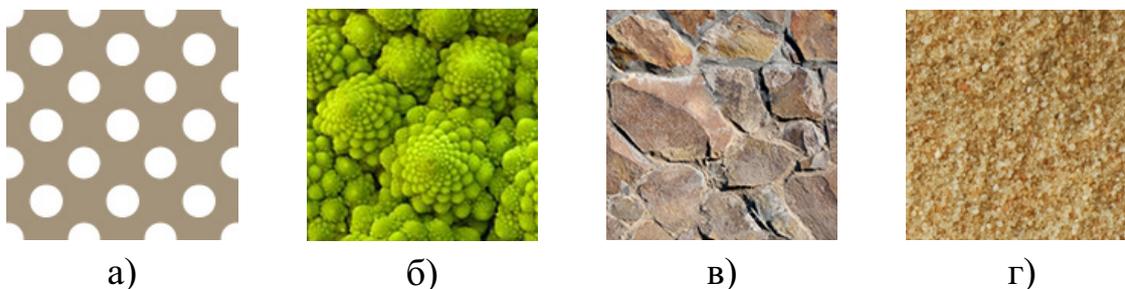


Рис. 1: Примеры различных текстур: а) регулярная, б) нерегулярная фрактальная, в) нерегулярная, г) имеет признаки как регулярной, так и нерегулярной текстур.

**Определение 2.** *Дескриптором текстуры* называется функция, описывающая текстуру по основным характеристикам. Можно считать его альтернативным представлением текстур для которого, к тому же, можно ввести меру схожести.

Примеры дескрипторов:

- Словесное описание (название)
- Гистограмма
- Многомерный вектор

**Определение 3.** *Сегментацией изображения* называется процесс его разбиения на несколько связанных замкнутых непересекающихся областей,

при этом связи между элементами внутри одной области сильнее связей между элементами разных областей.

**Определение 4.** *Классификацией текстур* называется определение принадлежности текстуры к одному из заданных заранее типов (классов) текстур.

Целью данной работы является разработка метода анализа аэрофотоснимков, который сегментирует фото на текстурные области и определяет тип каждой области. Перечень возможных областей задаётся заранее в виде базы текстур. Для обозначения каждого класса на результирующем изображении используется индивидуальный цвет или условные обозначения. В итоге необходимо получить план местности с обозначениями типов местности, который может служить одним из инструментов автоматического создания карты из фотоснимка.

На сегодняшний день существует множество алгоритмов и даже программных пакетов для автоматической генерации карты или плана местности по фото, результатом работы которых являются карта или план местности в виде растрового или векторного изображения с метаданными и условными символами. В контексте данной работы отличие плана местности от карты состоит в масштабе (у карты он значительно меньше, чем у плана местности) и в точности (линии дорог, границы каких-либо областей наносятся приблизительно, схематично). Однако большая часть метаданных задаётся пользователем вручную. Разрабатываемый в данной работе метод частично автоматизирует процесс добавления метаданных. В данной работе метаданными является тип местности каждой области на снимке.

Аэрофотоснимком в контексте данной работы будем считать любой фотоснимок, сделанный с воздуха. Таким образом, между аэрофотоснимком и снимком со спутника отличий не делается. Возможность использования

аэрофотоснимков наравне со спутниковыми обуславливается тем, что часто для решения поставленных задач не требуется настолько большой объём данных. Для создания карт города, района или отдельной местности не имеет смысла использование огромных снимков со спутника. Более эффективно будет использование локальных аэрофотоснимков. Очевидным плюсом таких исходных данных является тот факт, что такие снимки можно самостоятельно получить с помощью любого летающего объекта, оснащённого камерой. Однако, для корректного анализа изображений к ним предъявляются требования, чтобы на них не было облаков и летящих объектов, так как они являются инородными телами, не принадлежащими ландшафту. Стоит заметить, что к качеству снимков жёстких требований не предъявляется, так как используемые алгоритмы устойчивы к шуму.

На рис. 2 приведены примеры аэрофотоснимков.



а)



б)

Рис. 2: Примеры аэрофотоснимков: а) обработанный спутниковый снимок, б) снимок с летательного аппарата, сделанный под углом

Алгоритм обработки аэрофотоснимков:

1. Преобработка. Данный этап включает в себя коррекцию освещения для слишком тёмных или слишком освещённых фотоснимков, повышение контрастности. Этот этап не является обязательным, но в некоторых случаях значительно повышает точность дальнейшей обработки.

2. Сегментация на текстурные блоки. Всё изображение разбивается на отдельные области.
3. После проведения сегментации могут возникнуть области, площадь которых недостаточна для проведения классификации. Поэтому для обработки необходимо объединить каждую область с примыкающей к ней наиболее похожей областью. В итоге получаем набор регионов, размер каждого из которых, достаточен для текстурной классификации. На этом этапе предстоит определить критерий «примыкания» областей и подобрать минимальную классифицируемую площадь.
4. Классификация текстур каждой области. Для всех полученных на третьем шаге регионов вычисляется дескриптор, по которому с помощью меры схожести вычисляется степень принадлежности текстуры рассматриваемой области каждому классу текстур. База текстур с уже вычисленными для каждого класса дескрипторами подбирается заранее и не зависит от используемого алгоритма классификации и сегментации, но зависит от предметной области. На этом этапе необходимо выбрать тип дескриптора (напрямую связан с методом классификации), меру схожести (ввести метрику на множестве дескрипторов) и подобрать базу текстур наиболее точно отражающую особенности предметной области.
5. Объединение примыкающих областей одного класса. Это позволит сократить объем хранимой информации и ускорить любую дальнейшую работу с результатами обработки фотоснимка за счёт сокращения числа областей.

## Обзор литературы

Проблема распознавания текстур на спутниковых снимках ставилась в [1]. Для решения использовался анализ энергетического спектра изображения. Строились гистограммы относительных частот спектра изображения и типов текстур, после чего производилась их бинаризация. На основе бинарной модели производится определение границ, после чего проводится сравнение спектров каждой области со спектрами текстур каждого класса и определяется наиболее вероятный класс каждой области методом ближайшего соседа.

Достоинствами такого подхода являются инвариантность относительно поворота и масштабируемость, что важно для идентификации текстур на снимках. А так же устойчивость к шуму, что важно при анализе фотоснимков плохого качества. Кроме того, метод не требует сложных вычислений и не является ресурсоёмким, если использовать стандартные алгоритмы бинаризации и отслеживания границ. При более сложных и точных методах бинаризации и отслеживания границ сложность всего метода зависит от них.

Главным недостатком является работа с изображениями и текстурами в градациях серого из-за чего теряется информация, содержащаяся в цвете, что при анализе текстур природного происхождения иногда играет важную роль. Так же точность метода напрямую зависит от точности бинаризации изображения.

Для устранения недостатков, изложенных выше, для классификации текстур следует выбрать алгоритм, работающий с цветным представлением образцов текстур. Таким образом, весь процесс текстурного анализа изображений можно разделить на два больших этапа: текстурную сегментацию и

текстурную классификацию. Так как действия на одном из этих этапов не затрагивают второй, то и обзор существующих к ним подходов имеет смысл производить для каждого из них отдельно.

Так как часто текстура участка изображения может мешать точному выявлению его границ, следует выбрать те методы, которые учитывают эту особенность. В [2] и [3] описаны методы с использованием фильтров Габор. В них для описания изображения используется набор фильтров Габор. После чего проводится кластеризация по выявленным признакам текстур.

Достоинствами использования фильтров Габор является возможность работы с цветными изображениями в пространстве RGB и HSL, небольшое количество вычислений, устойчивость к градиентам и высокая точность.

Минусом же является необходимость иметь банк фильтров Габор. На данный момент не существует метода, позволяющего заранее сказать минимальное число фильтров, необходимое для точного описания изображения. Следовательно, для обеспечения наибольшей точности банк текстур следует подбирать самостоятельно экспериментальным или экспериментально-аналитическим образом. Существует методика адаптивного выбора фильтров [4], однако её применение возможно не везде. Кроме того, обратной стороной устойчивости к градиентам является невозможность распознать границу между похожими, но не одинаковыми текстурами, с чем не возникает проблем у других методов. Так же метод требует достаточно тонкой ручной настройки отдельно для каждой предметной области.

Проблема подбора банка фильтров не встаёт при использовании сегментации методом наращивания областей. Классический алгоритм сегментации [5] представляет собой итерационный процесс, при котором к одной начальной области, которая представлена единственным пикселем, на каждом шаге присоединяется только один соседний пиксель, для которого мера

отличия от пикселя, принадлежащего области и непосредственно касающегося рассматриваемого пикселя, минимальна. Остановка происходит когда минимальная мера отличия больше некоторого порогового значения. Отличие данного подхода от текстурной сегментации состоит в обнаружении однородной области, а не границ областей.

В чистом виде метод наращивания областей мало применим, поэтому в [6] рассматривается его модификация. Наращивание происходит из нескольких областей, которые выбираются автоматически.

Недостатками методов, описанных выше, является присоединение только одного пикселя, что исключает возможность применения такой сегментации для текстурированных областей. Так же методы работают с изображениями в градациях серого и не учитывают информацию, содержащуюся в цвете. Однако эта проблема решена в других работах. Например, в [7] и [8] описываются методы работы алгоритма с цветными изображениями. В [7] рассматривается сегментация в цветовом пространстве RGB, а в [8] — в пространстве YCbCr.

Для обеспечения возможности применения сегментации наращиванием областей для текстурированных участков необходимо присоединять на каждой итерации не один пиксель, а блок соседних пикселей. Для описания текстур каждого блока возможно использование разнообразных дескрипторов. Например, в [9] для описания признаков текстур в градациях серого были предложены матрицы смежности.

Выбор дескриптора является основополагающим шагом текстурной классификации. Все методы можно разделить на использующие банки фильтров (например, [10], [11], [12], [13], [14]) для вычисления дескрипторов и не использующие их.

Достоинствами методов, использующих банки фильтров, является их

универсальность. Для каждой предметной области всегда можно подобрать набор фильтров, позволяющих точно описать текстуры. Кроме того из всех фильтров можно выделить отвечающие за инвариантность относительно поворота, масштаба, перспективного искажения, устойчивость к шуму. Таким образом для обеспечения инвариантности достаточно добавить определённые фильтры и нет необходимости изменения самого метода.

Основным недостатком всех методов, названных выше, является необходимость подбора банков фильтров. Для эффективной работы требуется подобрать набор фильтров, обеспечивающий определение основных текстурных признаков и при этом имеющий минимальный возможный объём. Для многих банков фильтров, например, фильтров Габора, автоматический выбор такого набора на данное время не разработан.

Использование локальных бинарных шаблонов (LBP) для вычисления дескрипторов текстур решает проблему предварительного выбора фильтров. В [15] приведено определение локального бинарного шаблона, который описывает каждую точку при помощи соседних с ней точек. С помощью данного метода описываются изображения в градациях серого. Полученное описание точек неинвариантно относительно масштаба, поворота, искажения и неустойчиво к шуму.

Модификации LBP обеспечивающие инвариантность относительно масштаба рассматривается в [16], [17], [18], [19], относительно поворота в [16], устойчивость к шуму [16]. Работа с цветными изображениями рассматривается в [20], [19].

# Глава 1. Сегментация изображения

## 1.1. Алгоритм наращивания областей блоками

### 1.1.1. Форма блоков и связность

**Определение 5.** *Замощением* области  $S$  называется такое разбиение на множество непересекающихся многоугольников, объединение которых даёт саму область  $S$ . Альтернативное название замощения — мозаика. Многоугольники, на которые разбивается область, называются полигонами.

По типу полигонов, замощения можно разделить на:

- **Регулярные** - мозаики, составленные из полигонов одного типа и размера. Обобщением можно считать полигоны одинаковой площади без уточнения касательно типа.
- **Нерегулярные** - мозаики, составленные из нескольких различных типов полигонов или полигонов одного типа, но разных размеров.

На рис. 3 представлены примеры различных видов замощений.

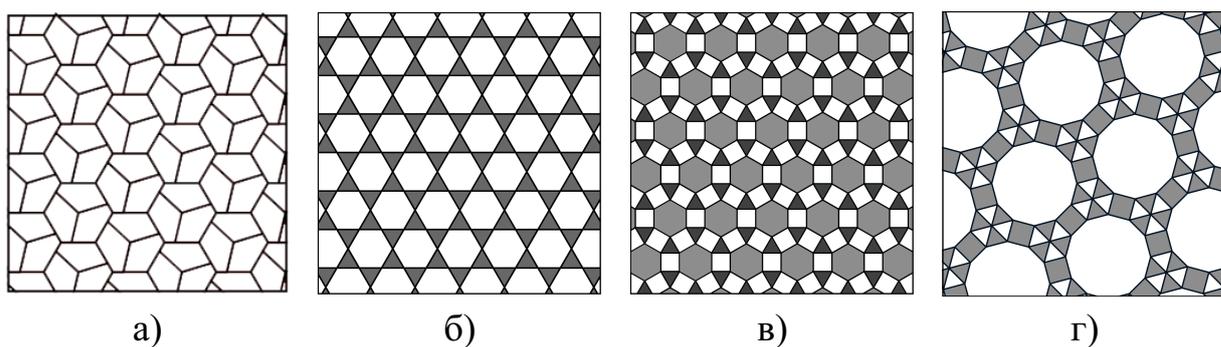


Рис. 3: Примеры замощений: а) регулярное, б) нерегулярное квазиправильное, в) нерегулярное полуправильное г) нерегулярное

Регулярные замощения по типу полигонов, на которые разбивается плоскость, можно разделить на правильные и неправильные. При неправильном регулярном замощении все полигоны имеют одинаковую форму, но не яв-

ляются правильными. Например, на рис. 4 приведены все открытые на сегодняшний день варианты регулярных неправильных замощений, полигоны которых являются выпуклыми пятиугольниками.

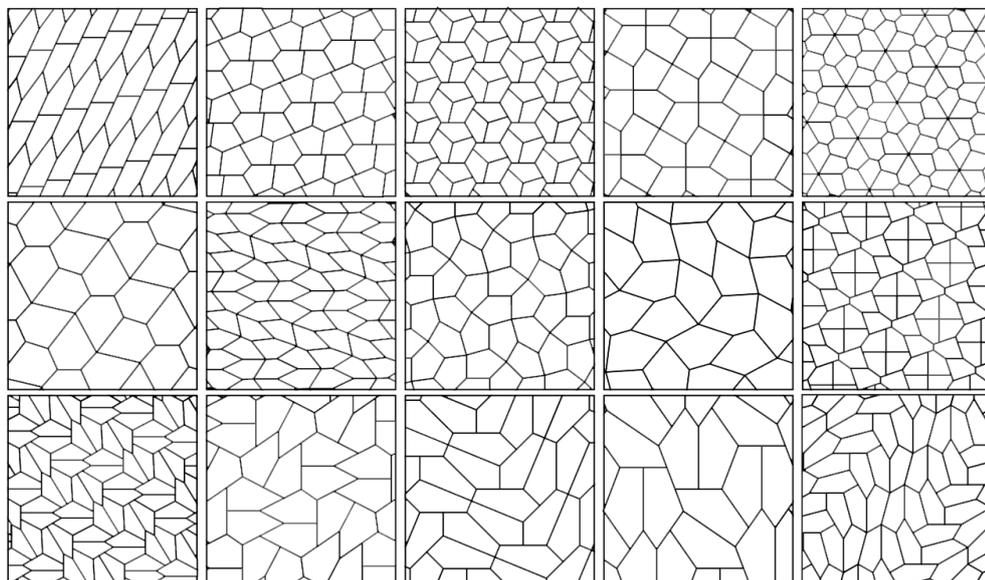


Рис. 4: 15 известных на данный момент вариантов замощения неправильными пятиугольниками

Правильные замощения составляют правильные многоугольники. В [21] доказано, что существует только три правильных замощения: треугольное, квадратное и гексагональное. Данные мозаики представлены на рис. 5.

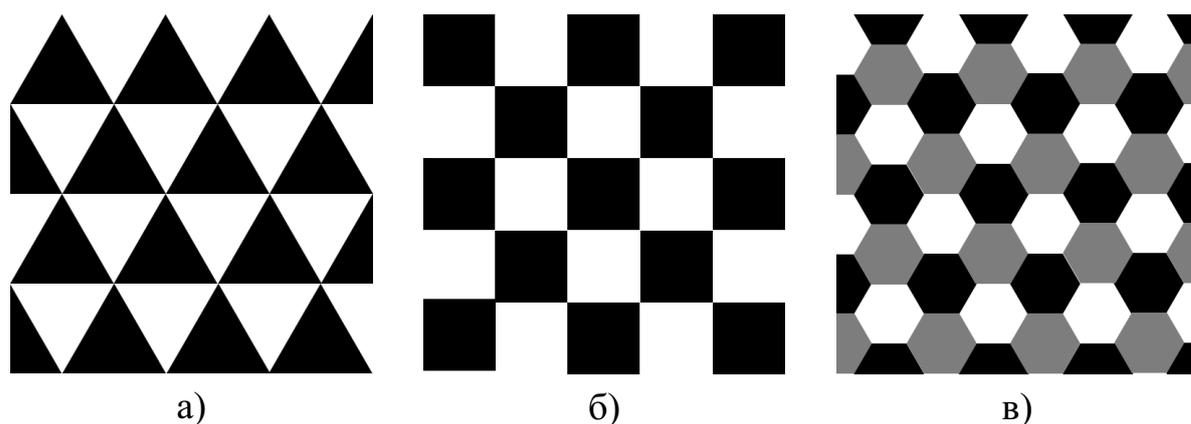


Рис. 5: Виды правильных замощений: а) треугольное, б) квадратное, в) гексагональное

Для наращивания областей блоками можно использовать любой из представленных выше вариантов замощения, как регулярных, так и нерегулярных. Примером нерегулярной сетки может служить любая сетка, получен-

ная в результате триангуляции. Однако на практике использование сложных блоков нецелесообразно, так как для определения принадлежности пикселей конкретному блоку необходимо проводить дополнительные расчёты, учитывающие форму блока. По этой причине в компьютерной графике используют правильные замощения.

**Определение 6.** Существует два подхода к определению смежности:

- *Смежными* называются полигоны имеющие общую сторону.
- *Смежными* называются полигоны имеющие общую сторону или общую вершину.

**Определение 7.** Полигон смежный с  $n$  полигонами называется  *$n$ -связным*.

На рис. 6 для полигонов правильных замещений продемонстрированы типы связности согласно двум вышеизложенным определениям. Области, смежные с выделенной, по общей стороне выделены синим, по общей вершине - голубым.

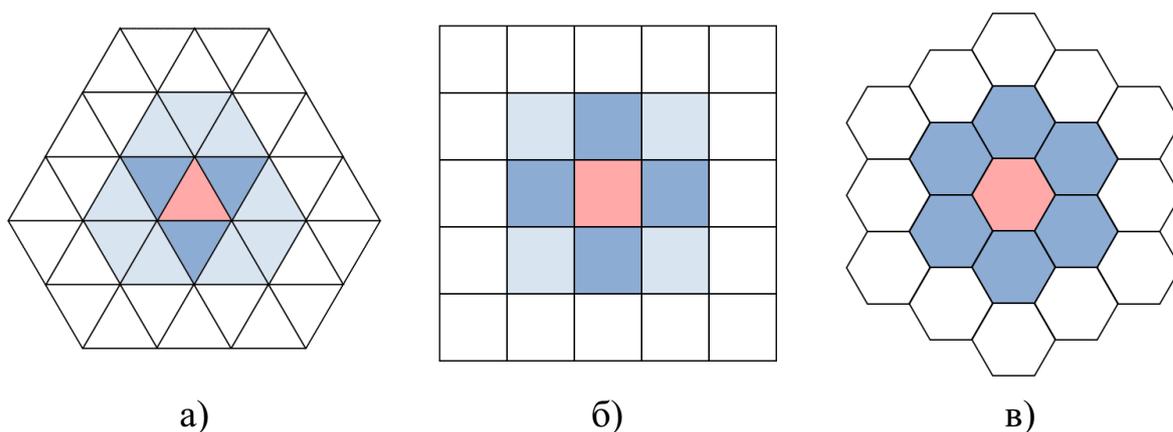


Рис. 6: Смежность при правильных замощениях:

а) 3-связная (синие смежные блоки) и 12-связная (синие и голубые блоки) треугольная область, б) 4-связная (синие смежные блоки) и 8-связная (синие и голубые блоки) квадратная область, в) 6-связная гексагональная область.

В данной работе в качестве блоков при наращивании областей блоками будут использоваться квадратные 4-связные полигоны. Выбор формы

блоков обуславливается тем, что она наиболее естественно отражает расположение пикселей растрового изображения. Вычисление принадлежности пикселей каждому полигону в данном случае не требует вычислительных затрат. 4-связность обеспечивает отсутствие «одиночных» блоков, касающихся наращиваемой области только углами. Таким образом исключаются участки областей, идентификация которых невозможна из-за их недостаточной площади.

### 1.1.2. Алгоритм наращивания блоками единственной области

Идея наращивания области блоками заключается в распространении алгоритма, работающего с пикселями, на изображения, представленные блоками. При таком представлении блок является наименьшей структурной единицей. В этом случае ключевым моментом алгоритма является выбор функции перехода от группы пикселей к блоку, то есть задание дескриптора.

Алгоритм можно разбить на следующие этапы:

1. Перевод растрового представления изображения в блочное, путём вычисления дескриптора каждой области:

$I = \{x_1, \dots, x_N\} \rightarrow Y = \{y_1, \dots, y_M\}$ , где  $N$  - количество точек изображения (пикселей),  $M$  - количество блоков в блочном представлении,  $M < N$ . После чего с новым представлением можно будет работать как с растровым изображением. Для начальной точки определяется соответствующий блок, который является стартовым для алгоритма, работающего с блоками, то есть  $x^0 \rightarrow y^0$ . Задаётся пороговое значение функции схожести  $\mu$ .

2. Стартовый блок принимается в качестве начальной области.  $S^{(0)} = y^0$ .

Счётчику присоединённых блоков присваивается значение 1.  $m = 1$ .

3. Для начальной области составляется список смежных с ней ещё не присоединённых блоков–кандидатов на присоединение.  $S_i = \{y_k, \dots, y_l\}$  — список кандидатов на присоединение на  $i$ -м шаге.
4. Для каждого блока-кандидата вычисляются значения функции схожести с блоками, принадлежащими наращиваемой области, и непосредственно смежными с ним.  $D_{ij} = \max \left\{ d(s_{ij}, s_k^{(i)}) \mid s_k^{(i)} \in S^{(i)} \right\}$  — значение функции схожести блока–кандидата  $s_{ij} \in S_i$  с наращиваемой областью на  $i$ -м шаге.
5. Определяются блоки–кандидаты, значение функции схожести которых больше порогового значения  $\mu$ , которое задаётся заранее.  
 $\tilde{S} = \{s_{ij} \mid s_j \in S_i, D_{ij} > \mu\}$ . Найденные блоки присоединяются наращиваемой области.  $S^{(i+1)} = S^{(i)} \cup \tilde{S}$ .  
Счётчик присоединённых областей увеличивается на мощность множества присоединяемых блоков  $m = m + |\tilde{S}|$ .
6. Если  $m = M$  или  $|\tilde{S}| = 0$ , то дальнейшее присоединение блоков невозможно. Переход к шагу 7. Иначе — переход к шагу 3.
7. Создание маски выделенной области.

$$I(y_j) = \begin{cases} 1, & y_j \in S \\ 0, & y_j \notin S \end{cases}$$

Переход от блоков к пикселям  $I(y_j) \rightarrow I(x_i)$ . Конец.

На рис. 7–8 приведены примеры применения немодифицированного метода для цветных изображений (рис. 7б), немодифицированного метода для

изображений в градациях серого (рис. 8а) и метода наращивания блоками (рис. 8б) к исходному фото (рис. 7а).

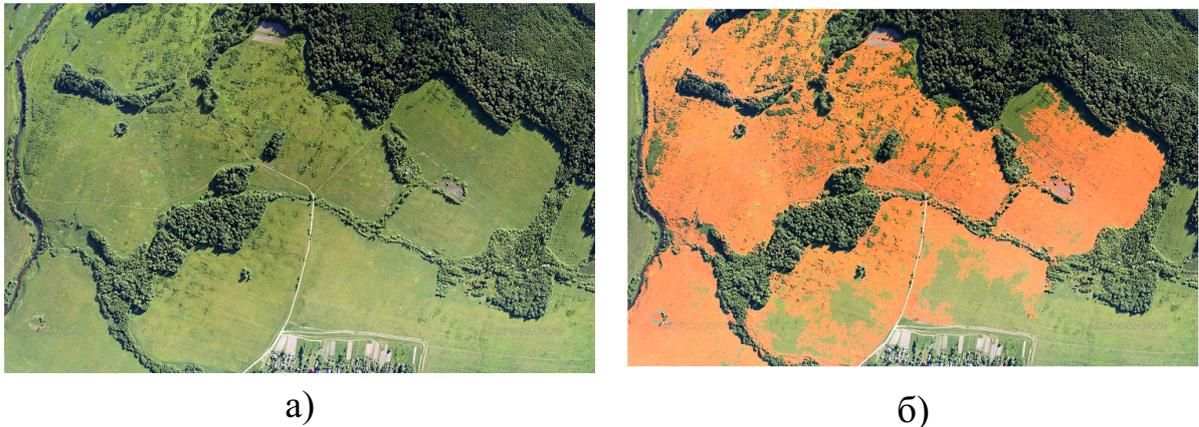


Рис. 7: Исходное фото (а) и сегментированное немодифицированным методом (б)

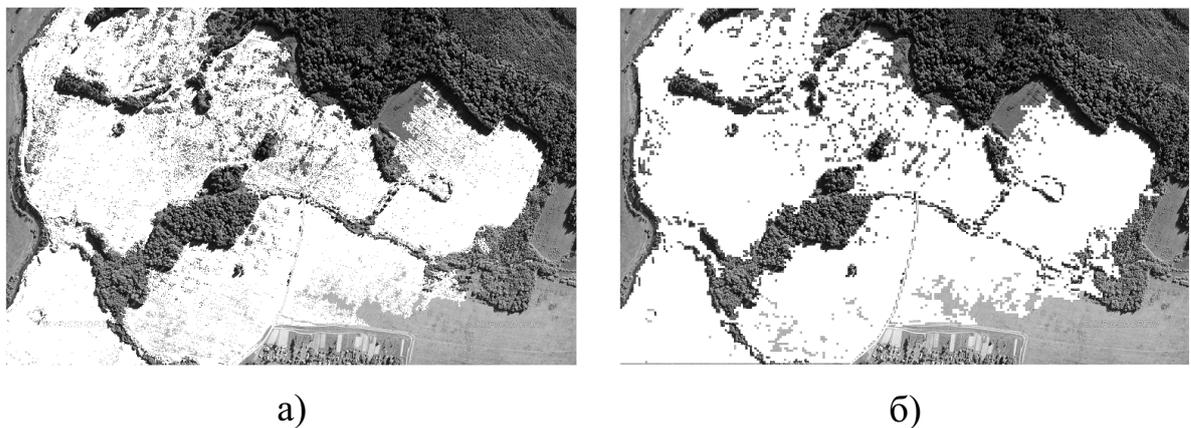


Рис. 8: Сегментация немодифицированным методом наращивания областей (а) и методом наращивания областей блоками (б)

## 1.2. Автоматический выбор начальных областей

### 1.2.1. Подходы к выбору начальных областей

Особенностью всех методов сегментации на базе наращивания областей является необходимость задания начальных точек. Количество областей в результате напрямую зависит от количества начальных точек. Однако при автоматической сегментации изображений нельзя заранее сказать на сколько областей они будут разделены. Из этого вытекает вопрос определения

количества и координат начальных точек.

В зависимости от момента определения начальных точек все методы автоматического их определения можно разделить на статические и динамические.

- **Статический подход** характеризуется определением количества и расположения начальных точек до запуска алгоритма сегментации. В этом случае на вход функции, выполняющей сегментацию методом на основе наращивания областей, подаётся уже заранее определённый набор координат стартовых точек.

Статические подходы можно разделить на два направления: задание точек в диалоговом режиме и вычисление точек программными методами.

1. В **диалоговом режиме** пользователь для каждого изображения сам задаёт количество начальных точек и их координаты. Для опытного специалиста не составит труда определить количество областей на фото, а следовательно и стартовых точек. Но само наличие специалиста сводит на нет попытку автоматизировать процесс.

2. **Программный подход** представляет собой проведение предварительной обработки изображения для определения количества начальных точек. Для преобработки используется алгоритм обнаружения границ [6]. Так как вычислительные затраты на преобработку не должны приближаться к затратам основной части алгоритма сегментации, то метод обнаружения границ должен быть максимально простым и быстрым, что может отрицательно сказаться на точности.

Так как точность на этапе преобработки не важна, то могут быть обнаружены не все области или, наоборот, структурные единицы текстуры могут быть приняты за отдельные области. Поэтому данный

подход часто применяется для предварительной оценки количества и расположения начальных областей для с целью дальнейшего их динамического уточнения.

- При *динамическом подходе* начальные точки определяются непосредственно в процессе выполнения алгоритма сегментации. Количество областей, на которые будет разделено изображение, заранее не известно. Алгоритм получает на вход только анализируемое изображение. На каждой итерации генерируется одна начальная точка и для неё проводится наращивание соответствующей области. Так как расположение каждой следующей начальной точки зависит от уже выявленных структурных областей, результат сегментации может зависеть от последовательности выбора этих точек.

Данный подход наиболее удобен, так как не требует преобработки изображения, позволяет полностью разбить фото на области и обеспечивает максимальную степень автоматизации сегментирования.

**Замечание 1.** При динамическом определении начальных точек существует возможность реализации интерактивного режима. В этом случае на каждой итерации пользователь выбирает из доступной области одну начальную точку, для которой производится наращивание, после чего обработанные точки или блоки из доступной области убираются, а пользователь снова выбирает следующую точку из уже сокращённой доступной области. Однако такой подход не является автоматическим и относится к динамическим только формально. На практике его использование при большом числе возможных областей неудобно.

### 1.2.2. Алгоритм автоматической сегментации

1. Перевод растрового представления изображения в блочное, путём вычисления дескриптора каждой области:

$I = \{x_1, \dots, x_N\} \rightarrow Y = \{y_1, \dots, y_M\}$ , где  $N$  - количество точек изображения (пикселей),  $M$  - количество блоков в блочном представлении,  $M < N$ . После чего с новым представлением можно будет работать как с растровым изображением. Задаётся пороговое значение функции схожести  $\mu$ . Счётчику выявленных областей, который будет использоваться для присваивания меток обработанным блокам, присваивается стартовое значение  $K = 0$ .  $Q = \emptyset$  - множество присоединённых блоков (блоков с метками).

2. Выбирается стартовый блок  $y^0 \notin Q$ , который принимается в качестве начальной области. Счётчик областей увеличивается на единицу

$K = K + 1$ . Начальному блоку присваивается текущая метка  $K$ .

$Q = Q \cup y^0$ .  $S^{(K)} = y_0$  - начальная область с меткой  $K$ .

3. Для начальной области составляется список смежных с ней блоков, ещё не имеющих меток, - кандидатов на присоединение.

$S_i = \{y_l \mid y_l \notin Q, y_l \text{ смежен } S^{(K)}\}$  - список кандидатов на присоединение на  $i$ -м шаге.

4. Для каждого блока-кандидата вычисляются значения функции схожести с блоками, принадлежащими наращиваемой области, и непосредственно смежными с ним.  $D_{ij} = \max \{d(s_{ij}, s_k^{(i)}) \mid s_k^{(i)} \in S_i^{(K)}\}$  - значение функции схожести блока-кандидата  $s_{ij} \in S_i$  с наращиваемой областью на  $i$ -м шаге.

5. Определяются блоки-кандидаты, значение функции схожести которых

больше порогового значения  $\mu$ , которое задаётся заранее.

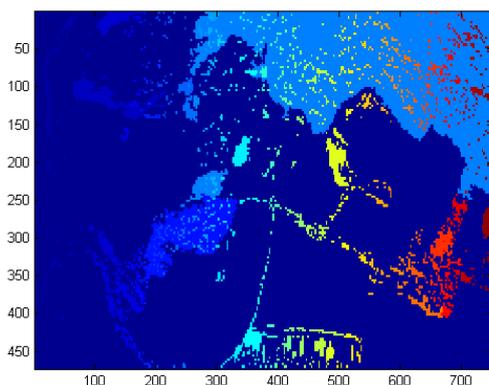
$\tilde{S} = \{s_{ij} \mid s_j \in S_i, D_{ij} > \mu\}$ . Найденные блоки присоединяются к наращиваемой области.  $S_{i+1}^{(K)} = S_i^{(K)} \cup \tilde{S}$ .

Присоединённым блокам присваиваются метки текущей области.

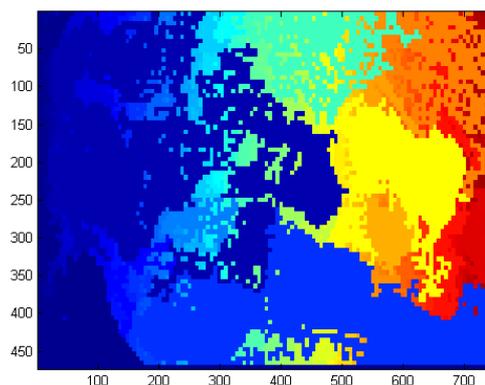
$$Q = Q \cup \tilde{S}.$$

6. Если  $|Q| = M$ , то не осталось необработанных блоков, всё изображение сегментировано. Переход к шагу 8. Иначе переход к шагу 7.
7. Если  $|\tilde{S}| = 0$ , то дальнейшее присоединение блоков к данной области невозможно. Переход к шагу 2. Иначе — переход к шагу 3.
8. Переход обратно от блоков к пикселям  $y_j \rightarrow x_i$ .  $x_i = x_i^{(K)}$  для  $y_j \in S^{(K)}$ .  
Конец.

В результате получаем изображение, каждой точке которого присвоена метка, однозначно указывающая на область, которой та принадлежит. Одинаковые метки говорят о принадлежности точек одной области, разные — разным областям.



а) блок  $3 \times 3$   
1492 сегмента



б) блок  $6 \times 6$   
998 сегментов

Рис. 9: Сегментация с автоматическим выбором начальных областей

На рис. 9 представлены результаты автоматической сегментации для блоков разных размеров. Сегментация и визуализация выполнены в пакете MATLAB. Каждому сегменту соответствует свой оттенок цвета. На шаге 2 алгоритма начальный блок каждой области выбирался произвольным образом из множества непомеченных блоков.

### 1.3. Объединение областей малой площади

Главным недостатком метода сегментации наращиванием областей с автоматическим выбором начальных точек является появление большого количества областей малой площади, то есть одиночных областей, состоящих из одного или нескольких блоков. Причинами их появления могут являться плохое качество исходного изображения, шум, особенности текстуры, прочие особенности изображения.

Так как при сегментации площадь присоединяемых блоков выбиралась таким образом, чтобы одновременно обеспечить достоверное представление текстурных признаков каждого блока и при этом гарантировать в достаточной степени высокую точность, однако размер таких блоков недостаточен для максимального выделения текстурных признаков - главной задачи классификации. Таким образом области, представленные одиночными блоками или малой группой блоков, общая площадь которых меньше минимальной классифицируемой области не могут быть обработаны. Для устранения этого производится объединение областей.

**Определение 8.** *Связной областью* называется область, любые две точки которой могут быть соединены кривой, полностью принадлежащей данной области.

В зависимости от требований связности существует две точки зрения на

цель и процесс объединения областей.

Если ставится требование обязательной связности каждой области, при объединении рассматриваются только смежные сегменты. В этом случае целью является исключение смежных сегментов одного типа путём их объединения. В данном контексте тип сегмента строго не определяется. Для каждой конкретной задачи понятие типа сегмента определяется индивидуально путём выбора соответствующих дескрипторов и определения функции схожести типов.

Если допускаются несмежные области, то для объединения широко применяется кластеризация. Как и в предыдущем случае, при кластерном анализе необходимо введение функции схожести, которая выполняет роль функции расстояния в пространстве текстурных признаков. Так как анализируется произвольное изображение и невозможно заранее сказать, сколько на нём представлено типов текстур, следует отметить, что для кластеризации не подходят методы, требующие предварительного указания числа кластеров, такие как K-means.

Достоинством объединения областей посредством кластерного анализа является существенное упрощение дальнейшей классификации, так как в идеальном случае изображение будет разделено на текстурные области, тип текстуры каждой из которых встречается ровно один раз. Это позволит после классификации текстур пропустить шаг объединения. На практике же из-за схожести текстур некоторых типов возможна их потеря из-за объединения нескольких кластеров в один.

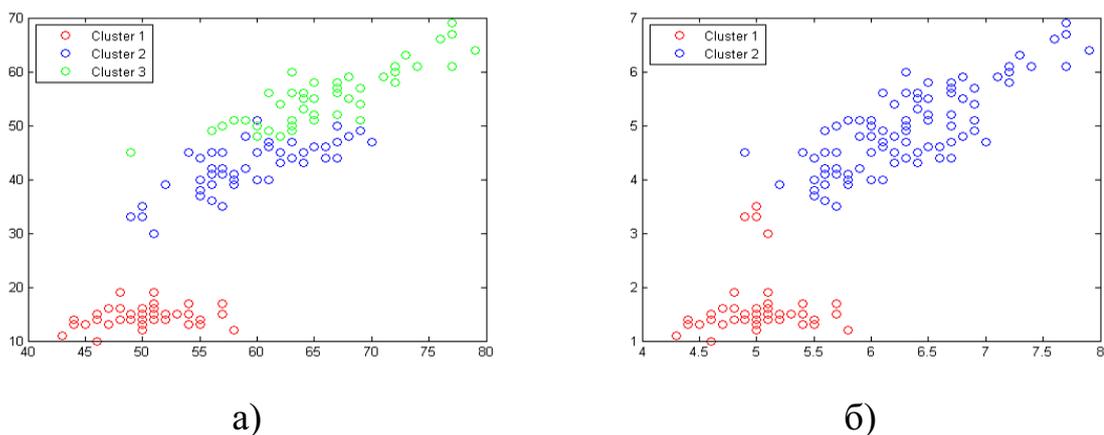


Рис. 10: Неточности кластеризации

На рис. 10 показана возможная ситуация с объединениями кластеров<sup>1</sup>.

Альтернативой является использование алгоритмов кластеризации с малым радиусом влияния для объединения только тех областей, меры схожести которых принимают значения близкие к 1 (в том случае, если мера схожести изменяется на отрезке  $[0;1]$ ). Это поможет сократить число сегментов, но не приведёт к повышению процента возможных неправильно классифицированных из-за неверного сегментирования участков.

### 1.3.1. Алгоритм объединения областей

Для решения поставленной задачи на этапе сегментации достаточно обеспечить отсутствие неклассифицируемых участков. Для этого производится простая проверка:

1. Пусть изображение разделено на  $N$  сегментов  $P = \{P_1, \dots, P_N\}$ . Для каждой области вычисляется её площадь  $S = \{S_1, \dots, S_N\}$ . Задаётся минимальная классифицируемая площадь  $\mu$ .

### 2. Составляется список областей недостаточной площади

<sup>1</sup>Для примера был взяты частичные данные из набора данных «ирисы Фишера», а именно длина чашелистика, длина лепестка и вид ириса (для сравнения с полученным результатом). Графики сгенерированы пакетом MATLAB. На примере видно, что при радиусе влияния 0,7 субтрактивный алгоритм нечёткой кластеризации обнаруживает только два кластера (рис. 10б), тогда как на самом деле их три (рис. 10а).

$$P^* = \{P_i \mid S_i < \mu, \quad i = \overline{1, N}\}.$$

3. Если  $P^* \neq \emptyset$  переход к шагу 4, иначе — конец.
4. Для  $P_j \in P^*$  среди смежных с ним сегментов выбирается наиболее похожий  $\tilde{P} = \max \{f(P_j, P_k)\}$ , где  $P_k \in P$ ,  $f(P_j, P_k)$  — функция схожести текстур областей  $P_j$  и  $P_k$ .
5. Если  $P_j \in P^*$  и  $P_k \in P^*$  и  $S_j + S_k < \mu$ , то  $P_i = P_j \cup P_k$ . Изменяем список неклассифицируемых участков:  $P^* = P^*/(P_j \cup P_k) \cup P_i$ . Изменяем список соответствующих площадей сегментов  $S = S/(P_j \cup P_k) \cup S_i$ , где  $S_i = S_j + S_k$ . Переход к шагу 3.
6. Если  $P_j \in P^*$  и  $P_k \in P$ , то изменяем список неклассифицируемых участков:  $P^* = P^*/P_j$ . Изменяем список соответствующих площадей сегментов  $S = S/P_j$ ,  $S_i = S_j + S_k$ . Переход к шагу 3.

На рис. 11 продемонстрировано сегментированное изображение до (рис. 11а) и после (рис. 11б) объединения блоков. Объединение блоков проводилось по алгоритму, объединяющему только неклассифицируемые блоки, описанному выше.

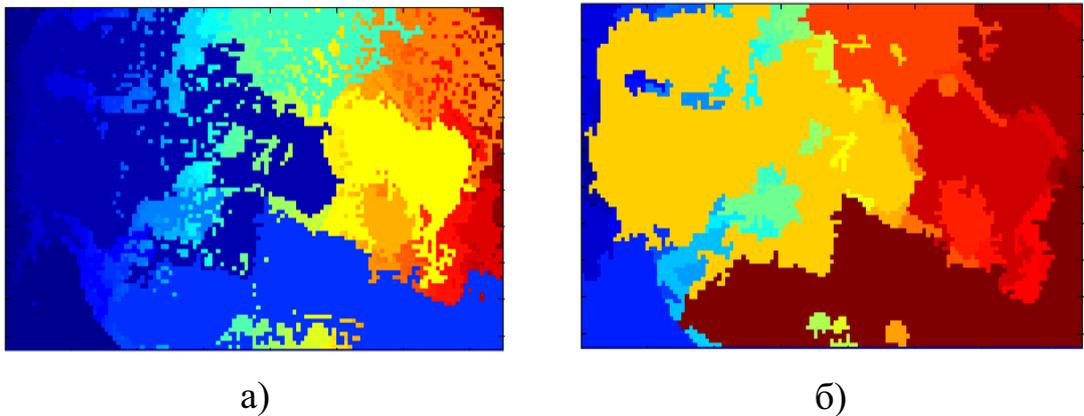


Рис. 11: Сегментированные изображения до (а) и после (б) объединения областей

## Глава 2. Тексту́рная классификация

### 2.1. Библиотека текстур

Библиотека текстур необходима для обучения нейронной сети, которая широко применяется при решении задач классификации образов. От выбора базы текстур зависит точность работы алгоритмов классификации. При недостаточном объёме библиотеки высока вероятность обнаружения областей, для которых не выполняется необходимое условие классификации (см. утв. 2). Под недостаточным объёмом библиотеки в этом случае понимают как объём всех образцов, так и число классов.

**Утверждение 1.** Необходимым условием принадлежности текстуры данному классу является требование, чтобы значение функции схожести текстуры с этим классом было выше некоторого порога уверенности.

$f(S, T) > \eta$ , где  $f(S, T)$  – мера принадлежности текстуры  $S$  классу  $T$ ,  $\eta$  – порог уверенности.

**Утверждение 2.** Необходимым условием классификации текстуры является выполнение необходимого условия принадлежности текстуры (утв. 1) хотя бы для одного класса.  $\max_i \{f(S, T_i)\} > \eta \quad i = \overline{1, N}$ , где  $f(S, T_i)$  – мера принадлежности текстуры  $S$  классу  $T_i$ ,  $N$  – число возможных классов,  $\eta$  – порог уверенности.

Порог уверенности  $\eta$  задаётся пользователем и позволяет регулировать точность классификации. По умолчанию  $\eta = 0$ , то есть необходимое условие классификации выполняется всегда. В этом случае каждая текстура принадлежит классу, мера принадлежности которому больше остальных.

Стремление к отсутствию неклассифицируемых текстур может привести к излишнему увеличению числа классов библиотеки. Невозможно создать универсальную базу текстур без вреда точности и трудоёмкости класси-

кации. В первую очередь вредны текстуры, вероятность появления которых на анализируемом изображении крайне мала или вообще равна нулю. Например, бессмысленно добавлять в библиотеку класс «мех», когда она будет использоваться для анализа аэрофотоснимков. Из этого следует, что важным условием эффективной классификации является правильный выбор библиотеки.

Перед непосредственным сбором образцов текстур для решения конкретной задачи следует определить следующие моменты:

1. Определить природу предметной области. Это могут быть рентгеновские снимки, снимки с помощью микроскопа, фото различных материалов [22],[23], фото местности.
2. На основе предметной области определить предполагаемые классы текстур. По возможности выделение классов производить от общих, дробя их на более частные в процессе анализа предметной области. Это позволит избежать пересечения классов и при необходимости даст возможность варьировать уровень детализации без лишних усилий.
3. Определить приблизительный диапазон изменения масштаба текстур. Например, на каком увеличении делаются снимки электронным микроскопом или с какой высоты делаются фотографии с воздуха. На основе этого производить отбор текстур для каждого класса. Пусть предметной областью являются аэрофотоснимки, а среди классов текстур выбран класс «трава», то фотографии газона, сделанные под углом  $90^\circ$  относительно поверхности, формально подходят, но, так как аэрофотоснимки делаются с высоты не менее 500 м, на практике не применимы, а следовательно не должны быть добавлены в качестве образцов в данный класс.

Для поставленной задачи классификации текстур на аэрофотоснимке предметом исследования являются фото земной поверхности сделанные с воздуха. Дабы по возможности сократить количество классов на этапе проверки работы алгоритмов было принято решение ограничить анализируемые аэрофотоснимки снимками средней полосы. При данных условиях было выделено 8 основных классов текстур: «вода» (реки, озёра, моря), «почва», «болото», «лес», «степь», «горы», «асфальт» (большие заасфальтированные участки как, например, аэродромы), «строения» (города, посёлки). Так как при постановке задачи было определено, что аэрофотоснимками считаются любые снимки, сделанные с летательного аппарата, то текстуры на образцах должны быть сфотографированы не менее чем с высоты 500 метров над уровнем земли.

В качестве образцов текстур использовались элементы различных баз данных ([22], [23]), а так же вручную выделялись области на изображениях, найденных в сети Интернет. Принимались цветные изображения и изображения в градациях серого. Цветные изображения переводились в цветовую модель RGB для единообразия представления и упрощения дальнейшей обработки библиотеки. Низкий уровень специализации классов позволил на данном этапе обойтись без обращения к специализированным экологическим сайтам. В конечном счёте каждый класс составленной библиотеки текстур насчитывал по 100 образцов размера  $200 \times 200$  пикселей с разной долей цветных и монохромных (см. табл. 1).

	Вода	Почва	Болото	Степь	Горы	Лес	Асфальт	Строения
Цветные	72	67	51	41	46	62	30	44
Монохромные	28	33	49	59	54	38	70	56
Всего	100	100	100	100	100	100	100	100

Таблица 1: Число цветных и монохромных образцов каждого класса библиотеки текстур

## 2.2. LBP-оператор

**Определение 9.** *Локальный бинарный шаблон* (Local Binary Pattern - LBP) - оператор, описывающий точку изображения с помощью соседних пикселей [15].

Пусть  $T = t(p_c, p_0, \dots, p_{n-1})$  - фрагмент текстуры,  $p_i$  - точки, составляющие его. Тогда локальный бинарный шаблон определяется следующим образом

$$\text{LBP}_{r,n} = \sum_{i=0}^{n-1} f(p_i - p_c) 2^i$$

где  $r$  - радиус окружности, на которой берутся точки для описания центральной,  $n$  - количество точек для описания центральной.

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

### Инвариантность относительно поворота

Классический LBP-оператор не обладает инвариантностью относительно поворота, что приводит к необходимости добавления в банк текстур одинаковых образцов, повернутых на различные углы. С целью сокращения размера банка текстур и повышения точности описания вводится модифицированный, инвариантный относительно поворота, оператор  $\text{LBP}^{ri}$ .

$$\text{LBP}_{r,n}^{ri} = \min \{ \text{ROR}(\text{LBP}_{r,n}, i) \}, \quad i = \overline{0, n-1},$$

где  $\text{ROR}(X, i)$  - функция поворота блока  $X$  на  $i$  пикселей по часовой стрелке.

## Устойчивость к шуму

LBP-оператор в классическом представлении крайне чувствителен к шуму, который неизбежен на аэрофотоснимках. Следовательно небольшой, не влияющий на текстуру, шум при использовании LBP, особенно с малым радиусом, может привести к большой доле ошибочных классификаций.

Для обеспечения устойчивости к шуму используется усреднение. Если пиксель описывается с помощью  $n$  соседних точек, а усреднение производится по  $q$  точкам, то для описания каждого пикселя необходимо брать  $qn$  точек.

$$\bar{p}_{r,q} = \frac{1}{q} \sum_{i=0}^{q-1} p_{r,kq+i}, \quad k = \overline{0, n-1}. \quad (1)$$

## Инвариантность относительно масштаба

Так как не всегда существует возможность подбора банка текстур таким образом, чтобы масштаб текстур на образцах совпадал с масштабом любых анализируемых текстур, встаёт необходимость обеспечить определённый диапазон допустимого различия масштабов изображения и образцов из банка данных. Для LBP-операторов это возможно осуществить путём изменения их радиуса. При этом вычисляются сразу несколько дескрипторов с разными радиусами. Число различных радиусов можно варьировать, тем самым контролируя сложность и эффективность классификации.

### 2.3. Дескрипторы текстур

Для описания текстурных признаков областей и блоков используются функции-дескрипторы, получающие на вход набор пикселей каждого блока, а на выходе определяющие вектор признаков каждого блока.

Определим дескриптор  $LBP_{r,n}^{ri}$ , описывающий центральный пиксель через соседние. Пусть  $r$  – радиус LBP-оператора,  $q$  – число пикселей для описания одной соседней точки,  $n$  – число соседних пикселей для описания центрального. Будем считать, что  $n = 8r$ , так как такой выбор соседних пикселей наиболее естественен для растровых изображений.  $p_c$  – центральный пиксель LBP-оператора,  $\bar{p}_{r,n}$  определён в (1).

$$LBP_{r,n} = \sum_n^{i=0} f(\bar{p}_{r,i} - p_c)2^i,$$

$$LBP_{r,n}^{ri} = \min \{ \text{ROR}(LBP_{r,n}, i) \}, \quad i = \overline{0, n-1}.$$

Определим дескриптор  $LBP_{r,n}^{ri}$ , описывающий степень различия между центральным пикселем и соседними.

$$z_{r,k} = \frac{1}{n} \sum_{i=0}^{n-1} |\bar{p}_{r,k+i} - p_c|, \quad k = \overline{0, n-1},$$

$$LBP_{r,n} = \sum_n^{i=0} f(z_{r,i} - \frac{1}{n} \sum_{j=0}^{n-1} z_{r,j})2^i,$$

$$LBP_{r,n}^{ri} = \min \{ \text{ROR}(LBP_{r,n}, i) \}, \quad i = \overline{0, n-1}.$$

$$LBP_{r,n} = LBP_{r,n}^{ri} \cup LBP_{r,n}^{ri}$$

где под операцией объединения имеется ввиду объединение гистограмм  $LBP_{r,n}^{ri}$  и  $LBP_{r,n}^{ri}$ .

Итоговым дескриптором, устойчивым к изменению масштаба, является дескриптор LBP:

$$LBP = \bigcup_{k=1}^m LBP_{k,8k},$$

где  $m$  – максимальный рассматриваемый радиус LBP-оператора.

Для цветных изображений вычисляются LBP\_R, LBP\_G, и LBP\_B, по значениям насыщенности соответствующих каналов. После чего полученные гистограммы частот объединяются в общий дескриптор LBP\_RGB.

**Замечание 2.** По умолчанию входное изображение представлено в цветовом пространстве RGB. Переход от других цветовых моделей, например, YCbCr, HSV, CMYK, к RGB и обратно осуществляется по выведенным для каждой модели стандартным формулам.

## 2.4. Двухшаговый алгоритм классификации

При переводе цветного изображения в градации серого неизбежна потеря информации. Это приводит к тому, что разным цветам могут соответствовать одинаковые значения уровня серого. Так как в природе цвет играет основополагающую роль, и окраска растений, песка, почвы и т.д. никогда не бывает случайной, а определяется группой экологических и климатических факторов, то при классификации в градациях серого результаты могут быть недостаточно точными. Однако часто исследователи сталкиваются с невозможностью сбора библиотеки текстур полностью из цветных образцов. В этом случае приходится работать со всей библиотекой в градациях серого, так как переход к цветовому пространству для серого изображения неоднозначен.

Двухшаговый подход к классификации текстур заключается в работе на разных этапах с разными представлениями изображений и банка текстур [24].

**Этап 0.** Для работы с разными представлениями изображений необходимо провести преобразование библиотеки текстур. Для этого она делится на две части: цветные образцы и образцы в градациях серого. После чего формируется две новых базы текстур (см. рис. 12): в градациях серого, которая формируется путём объединения части начальной библиотеки в уровнях серого и цветной части, переведённой в уровни серого, и полностью цветная, представляющая собой цветную часть начальной библиотеки.

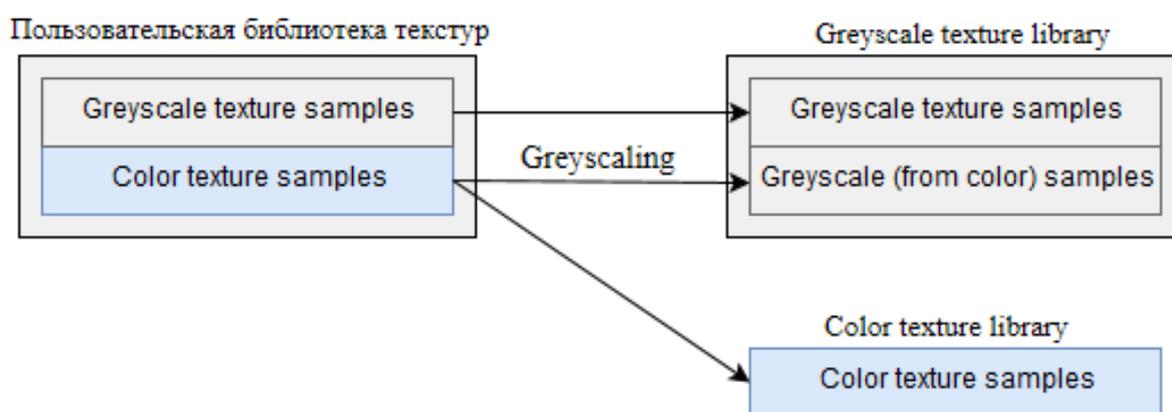


Рис. 12: Схема формирования библиотек текстур для каждого этапа классификации

**Этап 1.** На данном этапе анализируются структурные особенности текстуры. Осуществляется перевод копии анализируемой текстуры в представление в градациях серого (если оно изначально цветное) и последующая классификация с использованием соответствующей библиотеки текстур. В результате для каждого класса получаем степень принадлежности текстуры к нему. На следующий этап передаются только те классы, степень принадлежности к которым выше порогового значения. Если такой класс один, то второй этап игнорируется. Если таких классов нет совсем, то выбирается наиболее вероятный класс и второй этап так же игнорируется. Возможен альтернативный подход к отбору классов, согласно которому они сортируются по степени вероятности и

на второй этап передаются  $n$  самых вероятных.

**Этап 2.** На данном этапе для наиболее схожих по фактуре с исходным изображением классов текстур проводится цветовой анализ, позволяющий уточнить результаты. Вычисляются вероятности принадлежности текстуры классам, представленным цветными образцами. После чего вычисляется итоговые вероятности принадлежности (вероятность совместного появления событий) путём умножения вероятностей полученных на первом и на втором этапах. В итоге получаем классом текстуры является класс с наибольшей вероятностью принадлежности.

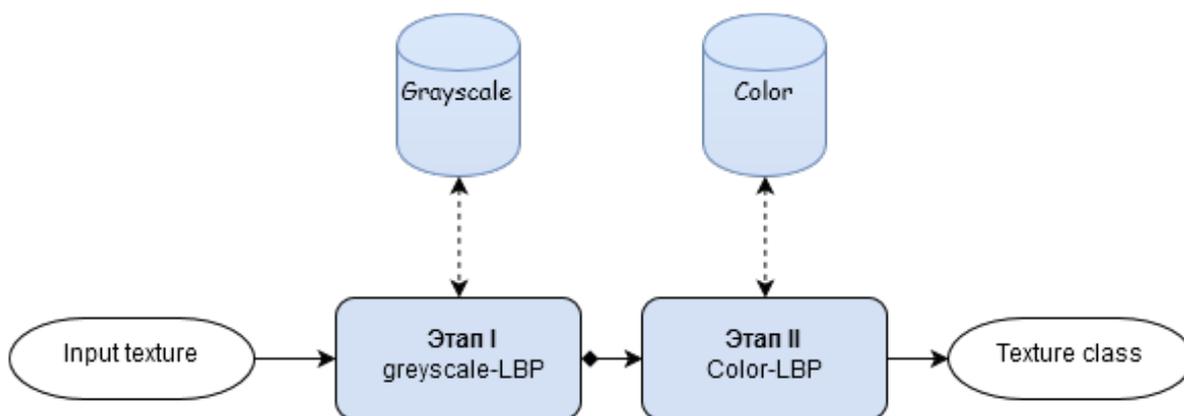


Рис. 13

В табл. 2 приведены сравнительные результаты работы двухшагового алгоритма и одношаговых алгоритмов для библиотек текстур в градациях серого и в цветовом пространстве RGB. Тестирование было проведено на 50 произвольно взятых образцах текстур.

	Color Texture Base	Greyscale Texture Base	Two-step Texture Base
Точность	88%	96%	92%

Таблица 2: Сравнительная таблица точности результатов работы методов классификации с различными библиотеками текстур

Результаты окончательной обработки аэрофотоснимков представлены на рис. 14.

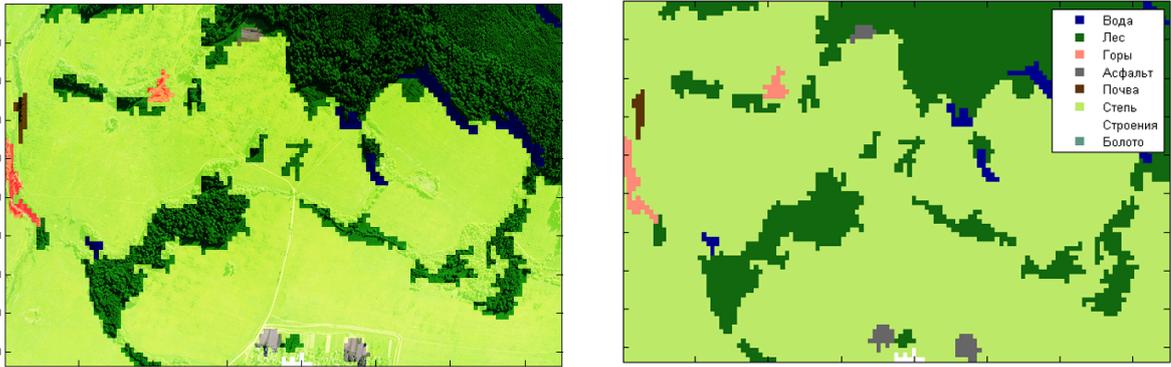


Рис. 14: Результаты текстурного анализа аэрофотоснимков по алгоритму, описанному в главе 1 и главе 2

## Выводы

В ходе решения поставленной задачи был разработан алгоритм, на основе которого написана программа, позволяющая анализировать аэрофотоснимки на предмет определения представленных на них типов местности. Было проведено тестирование её работы на нескольких снимках. В результате проведённых экспериментов было установлено, что достигается желаемая точность классификации.

Однако, так как в качестве классов был выбран сокращённый набор типов местности, результаты не отображают всего многообразия существующих типов местности. Это может быть решено путём расширения числа классов. Но более точное формирование новых классов и сбор образцов соответствующих текстур невозможны без помощи специалистов-экологов и картографов.

Недостатком работы программы является условность выявленных областей. В результате получаем только приблизительный план местности, который, впрочем, может быть наложен на уже существующую топографическую карту. Неточность при определении областей может быть сокращена за счёт модификации алгоритма сегментации и изменения типов или набора

дескрипторов текстур при сегментации.

Двухшаговый подход на этапе классификации позволил увеличить эффективность работы со смешанной базой текстур, состоящей из цветных образцов и образцов в градациях серого и унифицировать работу алгоритма для пользовательских библиотек текстур.

## **Заключение**

В данной работе был рассмотрен подход к картографическому анализу аэрофотоснимков. Были выявлены особенности текстур природного происхождения. Рассмотрены подходы к сегментации с учётом данных особенностей и проведена модификация метода сегментации наращивания областей. После чего были рассмотрены подходы к объединению полученных сегментов. На этапе классификации были рассмотрены модификации LBP-операторов и на их основе были сконструированы дескрипторы текстур. Был разработан двухшаговый алгоритм классификации текстур, позволяющий работать с неоднородной библиотекой текстур (содержащей и цветные образцы и образцы в градациях серого).

Плюсом разработанного алгоритма является информационная ёмкость полученных с помощью него результатов обработки аэрофотоснимка. В итоге мы получаем не просто сегментированное изображение, но и имеем информацию о каждой отдельной области. Это позволяет выделить характерные особенности каждой области, на основе чего в дальнейшем возможно построение 3D-модели местности.

## Список литературы

1. Антощук С. Г., Сербина Н. А. Система распознавания текстурных изображений при экологическом мониторинге // Искусственный интеллект. 2002. Т. 4. С. 406–413.
2. Jain A. K., Farrokhnia F. Unsupervised texture segmentation using Gabor filters // Pattern Recognition. 1991. Vol. 24, No 12. P. 1167–1186.
3. Sandler R., Lindenbaum M. Gabor filter analysis for texture segmentation // Computer vision and pattern recognition workshop. 2006. P. 178.
4. Hou Z., Parker J. M. Texture defect detection using support vector machines with adaptive Gabor wavelet features // Application of Computer Vision. 2005. Vol. 1. P. 275–280.
5. Adams R., Bischof L. Seeded region growing // Pattern Analysis and Machine Intelligence. 1994. Vol. 16, No 6. P. 641–647.
6. Ugarriza L. G. Automatic image segmentation by dynamic region growth and multiresolution merging // Image Processing, 2009. Vol. 18, No 10. P. 2275–2288.
7. Tremeau A., Borel N. A region growing and merging algorithm to color segmentation // Pattern recognition. 1997. Vol. 30, No 7. P. 1191–1203.
8. Shih F. Y., Cheng S. Automatic seeded region growing for color image segmentation // Image and vision computing. 2005. Vol. 23, No 10. P. 877–886.
9. Haralick R. M., Shanmugam K., Dinstein I. H. Textural features for image classification // Systems, Man and Cybernetics. 1973. No 6. P. 610–621.

10. Varma M. Statistical approaches to texture classification. // DPhil Thesis, University of Oxford. 2004.
11. Konishi S., Yuille A. L. Statistical cues for domain specific image segmentation with performance analysis // Computer Vision and Pattern Recognition. 2000. Vol. 1. P. 125–132.
12. Cohen F. S., Fan Z., Patel M. A. Classification of rotated and scaled textured images using Gaussian Markov random field models // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1991. No 2. P. 192–202.
13. Randen T., Husoy J. H. Filtering for texture classification: a comparative study // Pattern Analysis and Machine Intelligence. 1999. Vol. 21, Np 4. P. 291–310.
14. Grigorescu S. E., Petkov N., Kruizinga P. Comparison of texture features based on Gabor filters // Image Processing. 2002. Vol. 11, No 10. P. 1160–1167.
15. Ojala T., Pietikainen M., Maenpaa T. Gray scale and rotation invariant texture classification with local binary patterns // Computer Vision-ECCV. 2000. P. 404–420.
16. Liu L. BRINT: binary rotation invariant and noise tolerant texture classification // Image Processing. 2014. Vol. 23. No 7. P. 3071–3084.
17. Hadid A., Ylioinas J., Bengherabi M. Gender and texture classification: A comparative analysis using 13 variants of local binary patterns // Pattern Recognition Letters. 2015. Vol. 68. P. 231–238.
18. Wu X., Sun J. Joint-scale LBP: a new feature descriptor for texture classification // The Visual Computer. 2015. P. 1–13.

19. Zhu C., Bichot C. E., Chen L. Multi-scale color local binary patterns for visual object classes recognition // Pattern Recognition (ICPR). 2010. P. 3065–3068.
20. Banerji S., Verma A., Liu C. Novel color LBP descriptors for scene and image texture classification // 15th International Conference on Image Processing, Computer Vision, and Pattern Recognition. 2011. P. 537–543.
21. Кокстер Г. Введение в геометрию. М.: Наука, 1966.
22. База текстур CURET [Электронный ресурс]: URL:<http://www.cs.columbia.edu/CAVE/software/curet/> (дата обращения: 15.05.16).
23. The Heriot-Watt University TextureLab database [Электронный ресурс]: URL:<http://www.macs.hw.ac.uk/texturelab/resources/databases/> (дата обращения: 15.05.16).
24. Авдюшкина Л. С. Двухшаговая классификация природных текстур // Процессы управления и устойчивость. 2016. (в печати)