

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ  
УНИВЕРСИТЕТ имени академика С.П.КОРОЛЕВА  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

**Современные методы и информационные технологии тематической  
обработки данных ДЗЗ**

*Электронные методические указания  
к лабораторной работе № 4*

**САМАРА**

**2010**

**Составитель:** ассистент Копенков Василий Николаевич

В лабораторной работе №4 по дисциплине «Методы обработки данных дистанционного зондирования» рассматриваются инструментарий, программное обеспечение и технологии тематической обработки данных ДЗЗ: выделение контуров, связывание границ, сегментации областей, классификация областей, выделение изменений и анализ областей на космических снимках, полученных с различных космических аппаратов

Методические указания предназначены для магистров по направлению 010400.68 «Прикладная математика и информатика», обучающихся по программе «Математические и компьютерные методы обработки изображений и геоинформатики».

**Цель работы** – изучение методов тематической обработки и анализа космических снимков; получение навыков работы со специальным программным обеспечением; применение навыков обработки изображений к исследованию и анализу космических снимков.

## **1. ОСНОВНЫЕ СВЕДЕНИЯ О МЕТОДАХ ТЕМАТИЧЕСКОЙ ОБРАБОТКИ КОСМИЧЕСКИХ ИЗОБРАЖЕНИЙ ЗЕМНОЙ ПОВЕРХНОСТИ**

Тематическое дешифрирование - отрасль научного знания и метод исследования закономерностей строения геосферы (геологическое строение, поверхностные и грунтовые воды, почвенный покров, растительность, ландшафты, антропогенные объекты), решающая задачи информационного обеспечения исследований естественно-исторического и природно-ресурсного блока путем анализа данных дистанционного зондирования. Кроме того, тематическое дешифрирование является одним из важнейших этапов процесса картографирования, что позволяет относить его к области технологий. Основным методическим принципом тематического дешифрирования является комплексный подход, выражающийся в совместном изучении структуры искомого объекта, отраженной в виде признаков дешифрирования и системы факторов, оказывающих влияние на его строение, эволюцию и функционирование. Процедура дешифрирования делится на последовательные логические этапы, основными из которых являются распознавание, интерпретация и принятие решений. На этапе распознавания путем анализа объектно-ориентированных свойств материала дистанционного зондирования - признаков дешифрирования - решается задача установления изображенных предметов, явлений или их свойств. Этот наиболее легко формализуемый этап сейчас получил широкое распространение в программах обработки растровых изображений.

О возможности составления картографических документов методами дешифрирования материалов авиационных и космических съемок сейчас широко известно. К сожалению, менее известно, что топографическое и объектное дешифрирование, методами, которых пользуются в большинстве случаев, не дают возможности полноценного описания, классификации, построения иерархической структуры объектов земной поверхности. Приемы тематического дешифрирования, позволяющие достичь более подробного уровня описания с учетом иерархической структуры и взаимосвязей компонентов природной среды, более узкоспециализированы и предполагают вовлечение в обработку эмпирических

априорных данных. Это отражается на универсальности методического обеспечения работ подобного типа. Как показывает опыт, время от создания методики до ее реального применения на практике занимает несколько лет. Основными тормозящими факторами в этом случае являются индивидуальные особенности дешифровочных признаков для конкретного региона, с одной стороны, и узкая специализация интерпретаторов - с другой. В итоге при создании конкретной методики приходится проводить жесткий отбор признаков. Из-за этого возникают сильно формализованные схемы дешифрирования, зачастую раскрывающие лишь часть свойств и особенностей строения тематических объектов.

Отнесение объекта к какой-либо классификационной группе происходит по набору нежестких правил, вытекающих из свойств и характеристик обрабатываемого дистанционного изображения. Интерпретация использует логические категории, основанные на коррелятивных связях между компонентами ландшафта. Принятие решений в тематическом дешифрировании связывается в основном с процедурой выделения текущего объекта графически.

При этом для выделения объектов (разделения космического снимка на области-однородности), сегментации изображений и последующей классификации объектов в большинстве случаев необходимо последовательно решить ряд задач, таких как:

- выделение контуров на космическом снимке;
- локализация и определение границ объектов;
- сегментация изображения на области;
- классификация и распознавание областей;
- выявление изменений областей на космическом снимке.

После решения всех этих задач возможно точное дешифрирование и анализ космических снимков.

### **1.1. Выделение контуров на космических снимках, локализация и определение границ объектов.**

Методы выделения контуров (градиентный метод, методы Собелла и Лапласа, статистический метод, использование вейвлет-преобразования и т.д.), локализация и определение границ объектов были рассмотрены в курсе лекций «Математические

методы обработки изображений», кроме того, в ходе чтения данного курса студентами были выполнены различные виды лабораторных работ по данной тематике.

## 1.2. Сегментация космических снимков.

### 1.2.1. Сегментация, основанная на контурах

Пусть имеется изображение, содержащее некоторую совокупность объектов на однородном фоне. Изображение бинарное, то есть значения отсчетов, соответствующих объектам, равны единице, а отсчеты фона имеют нулевые значения.

При работе с полутоновым/цветным изображением, для которого есть маска контуров, можно воспользоваться представлением в виде множества бинарных изображений, на каждом из которых выделены объекты в виде контуров областей близких друг к другу по яркости.

Требуется выделить объекты, уточнить их положение и размеры, а так же возможно определить какие-либо характеристики, например, определить общее число объектов, например, частиц, их площадь.

Будем считать, что площадь - это число отсчетов, принадлежащих объекту, то есть каждый отсчет представляет собой квадрат единичной площади (рисунок 4). Объект определяется по критерию четырехсвязности.

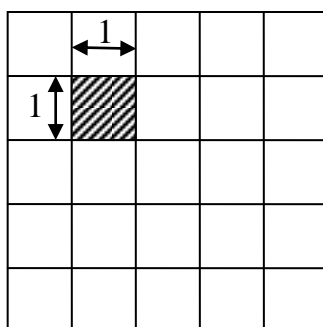


Рисунок 1 - Положение и размер отсчета

Выделение объектов происходит при последовательном обходе пикселей маски движения. Если в маске движения обнаружен ноль, запускается процедура поиска пикселей, связанных с данным, которым также соответствуют нули в маске движения. Найденные пиксели помечаются, чтобы избежать повторного их обнаружения.

Пусть прямоугольная матрица отсчетов обрабатывается в порядке построчной

развертки, то есть слева направо в строке и сверху вниз по строкам. Рассмотрим произвольный отсчет.

Если  $f_{n_1, n_2} = 0$ , то есть отсчет принадлежит фону, то осуществляется переход к следующему отсчету.

Если  $f_{n_1, n_2} = 1$ , то выполняется анализ принадлежности текущего отсчета к какому-либо объекту. Для этого дополнительно рассматриваются два соседних уже обработанных отсчета:  $f_{n_1-1, n_2} = 1$  и  $f_{n_1, n_2-1} = 1$ .

Если  $f_{n_1-1, n_2} = f_{n_1, n_2-1} = 0$ , то текущий отсчет  $f_{n_1, n_2}$  представляет собой начальную точку новой области (новый объект).

Если  $f_{n_1, n_2-1} = 1$  и  $f_{n_1-1, n_2} = 0$ , то отсчет присоединяется к области, к которой принадлежит соседний по горизонтали единичный отсчет  $f_{n_1, n_2-1}$ , пересчитывается характеристика этой области (прибавляется единица к площади).

Если  $f_{n_1, n_2-1} = 1$  и  $f_{n_1-1, n_2} = 0$ , то такое же действие выполняется по отношению к области, к которой принадлежит отсчет  $f_{n_1-1, n_2}$ .

Если  $f_{n_1-1, n_2} = f_{n_1, n_2-1} = 1$ , то анализируются области принадлежности этих отсчетов.

В случае если оба соседних отсчета принадлежат одной области, то выполняется присоединение к ней, как в двух предыдущих случаях.

В случае если эти отсчеты принадлежат разным областям, то эти области, а также текущий отсчет, объединяются в одну область, характеристики областей пересчитываются в общую характеристику (площади суммируются и прибавляется единица, чтобы учесть и текущий отсчет).

Обработка по приведенной схеме выполняется для всех отсчетов.

### 1.2.2. Сегментация, основанная на областях

Другим способом сегментации плоского изображения является сегментация в условиях отсутствия информации о контурах на изображении. При этом сегментация представляет собой операцию разбиения конечного множества плоскости  $R$ , на которой определена функция исходного изображения  $f(x, y)$ , на  $n$  непустых связанных подмножеств  $R_i$ ,  $i = 1, n$ . Разбиение производится в соответствии с

некоторым логическим предикатом  $P$ , определяемом на множестве всех точек из  $R = R_1, R_2, \dots, R_k$  и принимающим истинные значения, когда все пары точек из каждого подмножества  $R_i$  удовлетворяет некоторому критерию однородности. Например, это может быть критерий однородности, основанный на оценке максимальной разности яркости отдельного пикселя и среднего значения яркости, вычисленного по соответствующей области:

$$P(R_i) = True, \quad i = 1, \dots, n,$$

$$P(R_i \cup R_j) = False, \quad i \neq j.$$

### Наращивание областей

Методы этой группы основаны на использовании локальных признаков изображения. Идея метода наращивания областей состоит в выборе стартовых точек и последующем анализе соседних с ними точек в соответствии с некоторым критерием однородности (например, яркость в некотором диапазоне). В последующем проанализированные точки зачисляются в ту или иную группу (количество стартовых точек должно быть равно количеству однородных областей на изображении).

Если изображения монохромные, анализ областей проводится с использованием дескрипторов, основанных на значениях яркости и пространственных характеристиках (таких как текстура или статистические моменты).

Однако использование при выращивании областей одних лишь дескрипторов может привести к ошибочным результатам, если это делается в отрыве от информации о смежности или связности областей. Например, представим себе случайную схему расположения пикселей, принимающих только три отличающихся значения яркости. Если строить «области», группируя вместе пиксели с одинаковой яркостью и не обращая внимания на их связность, то это приведет к сегментации, лишенной смысла в контексте изображения.

### Разбиение и слияние областей

Описанная только что процедура выращивает области из множества начальных точек, играющих роль центров кристаллизации. Альтернативный подход состоит в том, чтобы провести первичное разбиение изображения на множество произвольных непересекающихся областей и в дальнейшем осуществлять слияние и/или разделение

этих областей, стремясь выполнять условия, сформулированные выше.

Пусть вся область изображения обозначена  $R$ , и выбран предикат  $P$ . Один из подходов к сегментации  $R$  состоит в том, чтобы последовательно разбивать ее на все более и более мелкие области  $R_j$ , пока для каждой из них не будет выполняться условие  $P(R_j) = True$ . Работа начинается со всей области изображения. Если  $P(R) = False$ , изображение делится на четверти вертикальной и горизонтальной прямыми, проходящими через середину. Если для какой-то четверти предикат  $P$  принимает значение  $False$ , она аналогичным способом делится на более мелкие четверти, и так далее, см. рисунок 2.

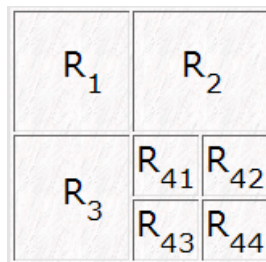


Рисунок 2 - Разбиение изображения на области

Если использовать только операцию разделения, то в окончательном разбиении изображения могут присутствовать соседние области с одинаковыми свойствами. Этот недостаток можно устранить, применяя наряду с разделением также операцию слияния. Для соблюдения ограничений требуется, чтобы слиянию подвергались только соседние области, пиксели которых в совокупности удовлетворяют предикату  $P$ . Иначе говоря, две соседних области  $R_i$  и  $R_j$  сливаются только в том случае, если  $P(R_i \cup R_j) = True$ .

Все вышеизложенное можно суммировать в виде процедуры, на каждом шаге которой выполняются следующие действия.

Любая область  $R_i$ , для которой  $P(R_i) = False$ , делится на четыре непересекающиеся четверти.

Любые две соседние области  $R_i$  и  $R_j$ , для которых  $P(R_i \cup R_j) = True$ , объединяются в одну.

Если невозможно выполнить ни одной операции слияния или разделения, то процедура заканчивается.



### 1.2.3. Сегментация с помощью метода водораздела

#### Описание метода водораздела

При обработке методом водораздела считается, что изображение представляет собой топологическую поверхность, которую затопляет водой. Каждый из «бассейнов водосбора» - отдельный сегмент изображения.

Рассмотрим одноканальное изображение. Как было сказано выше, оно представлено матрицей значений функции яркости  $f(x)$ . Введем множество  $X_i = \{x : f(x) \geq i\}$  - точки, в которых функция яркости принимает значения большие или равные  $i$ , и множество  $Z_i = \{x : f(x) \leq i\}$ .

Морфологический градиент изображения определим как  $g(f) = f \oplus B - f * B$ , где  $f \oplus B$  - дилатация, а  $f * B$  - эрозия функции яркости  $f$ , где  $B$  - некая маска. Если функция яркости  $f$  является непрерывно дифференцируемой, то этот морфологический градиент равен модулю градиента функции яркости  $f$  в точке. В иных случаях морфологический градиент можно аппроксимировать разницей между максимальным и минимальным значением функции яркости в некоторой окрестности точки.

Функция расстояния. Пусть имеется некоторое множество  $Y$  на  $R^2$ . Для каждого элемента определим расстояние от нее до дополнения множества  $Y$ :

$$\forall y \in Y, d(y) = \text{dist}(y, Y^c),$$

где  $Y^c$  - дополнение множества  $Y$ .

Можно показать, что

$$X_i(d) = \{y : d(y) \geq i\} = Y * B_i,$$

где  $B_i$  - круг с радиусом  $i$ .

Пусть имеется множество  $X$  на  $R^2$  и две точки этого множества  $x$  и  $y$ . Геодезическое расстояние  $d_x(x, y)$  между  $x$  и  $y$  определяется как длина кратчайшего пути, включенного в  $X$  и соединяющего  $x$  и  $y$ .

Пусть  $Y$  - некоторое множество, включенное в  $X$ , тогда обозначим через  $R_x(Y)$  множество всех точек из  $X$ , имеющих конечное геодезическое расстояние до  $Y$ :

$$R_X Y = \{x \in X : \exists y \in Y, d_X(x, y) < \infty\}.$$

Пусть  $Y$  состоит из нескольких областей  $Y_i$ , тогда зоной геодезического влияния  $Z_X Y_i$  области  $Y_i$  называются множество точек из  $X$ , геодезическое расстояние от которых до  $Y_i$  конечно и меньше, чем геодезическое расстояние до любой другой области  $Y_j$ :

$$Z_X Y_i = \{x \in X : d_X(x, Y_i) < \infty \vee \forall j \neq i, d_X(x, Y_i) < d_X(x, Y_j)\}.$$

Тогда зоной геодезического влияния всей области  $Y$  будет:

$$IZ_X Y = \bigcup_i Z_X Y_i.$$

А геодезическим скелетом зон влияния  $Y$  на  $X$ :

$$SKIZ_X Y = X / IZ_X Y.$$

Введем понятие локального минимума на изображении. Рассмотрим множество  $x, f(x)$  - его можно представить как топологическую поверхность  $S$ . Чем больше значение функции яркости  $f(x)$ , тем выше находится точка на данной поверхности.

Рассмотрим две точки  $s_1$  и  $s_2$  поверхности  $S$ . Путь из точки  $s_1(x_1, f(x_1))$  в точку  $s_2(x_2, f(x_2))$  - это любая последовательность  $s_i$  точек  $S$  такая, что  $s_i$  является соседней к  $s_{i+1}$ . Невозрастающий путь - это такой путь, в котором:

$$\forall s_i(x_i, f(x_i)), s_j(x_j, f(x_j)) \quad i \geq j \Leftrightarrow f(x_i) \leq f(x_j).$$

Точка  $s \in S$  принадлежит к точкам минимума, если не существует возрастающего пути, следующего из  $s$ . Минимум можно представить как точку водосбора на топографической поверхности.

Схожим образом вводится определение максимума. Все точки рассматриваются как принадлежащие к одному из трех видов:

- точки локального минимума;
- точки, с которых вода скатывается в один локальный минимум;
- точки, с которых вода с равной вероятностью скатывается более чем в один такой минимум.

Применительно к конкретному локальному минимуму, набор точек,

удовлетворяющих условию второго типа, называется бассейном (или водосбором) этого минимума. Множества точек, удовлетворяющих условию третьего типа, образуют линии гребней на топографической поверхности и называются линиями водораздела.

Иными словами, рассмотрим область  $Z_i f$  функции  $f$  на уровне  $i$  и предположим, что наводнение достигло этой высоты. Теперь рассмотрим область  $Z_{i+1} f$ . Можно заметить, что наводнение в  $Z_{i+1} f$  происходит в областях, связанных с ранее затопленными, но также есть ряд точек, которые являются минимумами на уровне  $i+1$ . Эти точки минимума также должны быть затоплены на уровне  $i+1$ .

Обозначая через  $W_i f$  области  $f$ , затопляемые на уровне  $i$ , а через  $M_i f$  - минимумы на уровне  $i$ , получаем

$$W_{i+1} f = \left[ IZ_{Z_{i+1} f} X_i f \right] \cup M_{i+1} f ,$$

а точки минимума на уровне  $i+1$  могут быть получены из соотношения

$$M_{i+1} f = Z_{i+1} f / R_{Z_{i+1} f} Z_i f .$$

Для запуска итеративного алгоритма принимается  $W_{-1} f = \emptyset$ . По окончании алгоритма линии водораздела находятся как:

$$DL f = W_N^c f ,$$

где  $N = \max f$ .

Для сегментации изображения по данному методу сначала находится его морфологический градиент и затем к нему применяется описанный выше метод водораздела.

#### Иерархический метод водораздела

Прямое применение метода водораздела зачастую приводит к неудовлетворительным результатам – однородные регионы оказываются разделены на множество сегментов (см. рисунок 3).

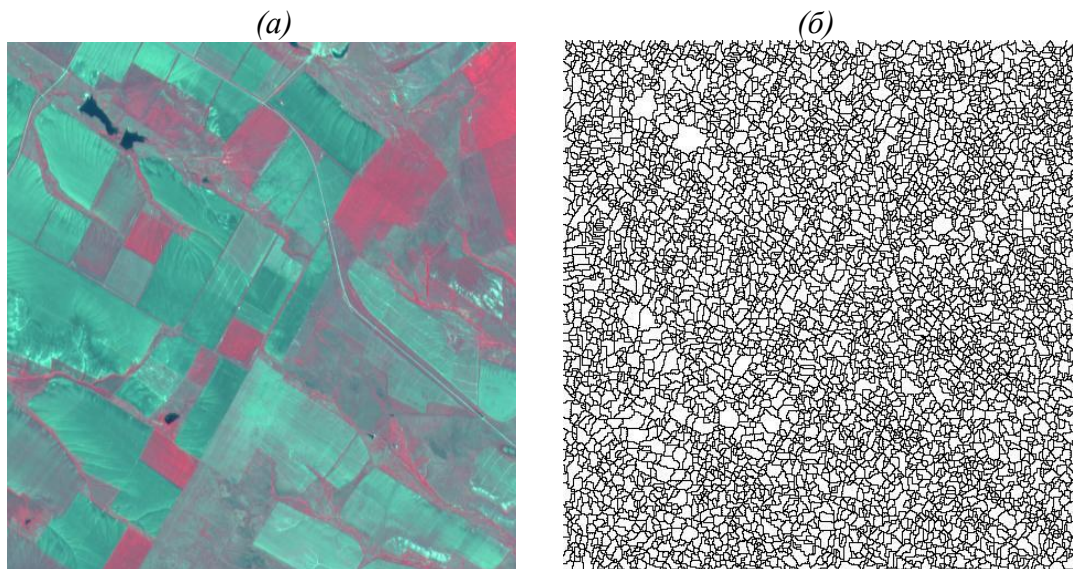


Рисунок 3 - (а) Исходное изображение; (б) Сегментация по методу водораздела

Из-за шумов на изображении образуется множество «бассейнов водосбора», которые и определяют подобный результат обработки. Данную проблему можно обойти использованием того же самого алгоритма повторно.

После первой итерации сегментации создается мозаичное изображение, в котором каждому сегменту присваивается значение функции яркости, которое соответствует минимальному значению морфологического градиента в данном сегменте.

После этого строится взвешенный граф границ между сегментами. (см. рисунок 4). Две границы считаются соседями, если они окружают один сегмент и вес границы определяется как разница между значениями функции яркости сегментов, которые он разделяет.

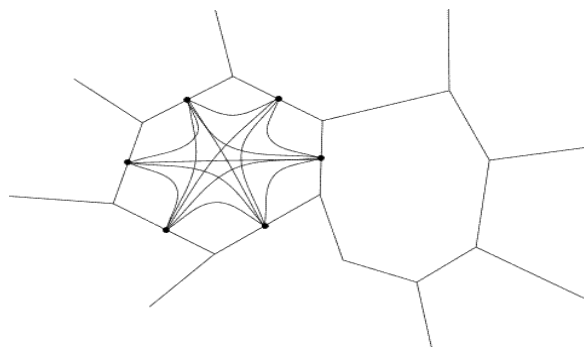


Рисунок 4 — Граф границ сегментов

Далее на основе полученного графа строится геодезическая поверхность и на ней применяется метод водораздела. Все лишние границы будут отброшены и останутся наиболее контрастные границы изображения. После проведения второй

итерации формируется изображение как на рисунке 5.

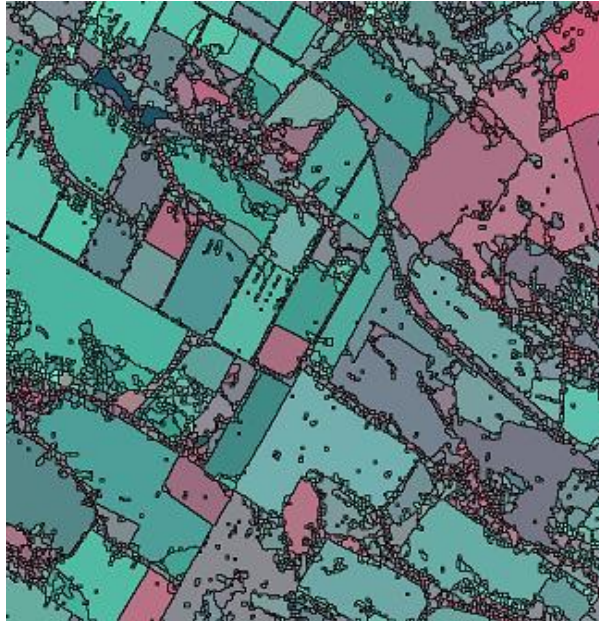


Рисунок 5 – Сегментация по методу водораздела.

#### 1.2.4. Алгоритм сдвига среднего

Алгоритм сдвига среднего основан на вычислении оценки плотности распределения точек в пространстве.

Пусть  $x_i$   $i=1..n$  – случайное множество из  $n$  точек в  $d$ -мерном Евклидовом пространстве. Тогда многомерная оценка плотности Розенблатта-Парзена с ядром  $K(x)$  и окном радиусом  $h$  определяется как:

$$\nabla f(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right).$$

Оптимальным ядром, при использовании которого возникает наименьшая среднеквадратичная ошибка, является ядро Епанечникова:

$$K_E(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2) (1-x^T x) & \text{если } x^T x < 1, \\ 0, & \text{иначе,} \end{cases}$$

где  $c_d$  - объем единичной  $d$ -мерной сферы.

Использование дифференцируемого ядра позволяет найти оценку градиента плотности как градиент оценки ядра плотности:

$$\widehat{\nabla} f(x) \equiv \nabla \widehat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n \nabla K\left(\frac{x-x_i}{h}\right).$$

При использовании ядра Епанечникова, оценка градиента плотности принимает вид:

$$\widehat{\nabla} f(x) = \frac{1}{nh^d c_d} \frac{d+2}{h^2} \sum_{x_i \in S_h(x)} x_i - x = \frac{n_x}{nh^d c_d} \frac{d+2}{h^2} \left( \frac{1}{n_x} \sum_{x_i \in S_h(x)} x_i - x \right),$$

где  $S_h(x)$  - гиперсфера радиуса  $h$  с объемом  $h^d c_d$  с центром в  $x$ , в которой содержится  $n_x$  точек.

Последний множитель, имеющий вид

$$M_h(x) \equiv \frac{1}{n_x} \sum_{x_i \in S_h(x)} x_i - x = \frac{1}{n_x} \sum_{x_i \in S_h(x)} x_i - x,$$

называется выборочным сдвигом среднего, а число  $\frac{n_x}{nh^d c_d}$  является оценкой ядра

плотности  $\widehat{f}(x)$ , вычисленной в гиперсфере  $S_h(x)$ . Таким образом, получаем:

$$\widehat{\nabla} f(x) = \widehat{f}(x) \frac{d+2}{h^2} M_h(x).$$

что приводит к следующему выражению:

$$M_h(x) \equiv \frac{h^2}{d+2} \frac{\widehat{\nabla} f(x)}{\widehat{f}(x)}$$

Откуда следует, что оценку нормализованного градиента можно получить вычислением выборочным сдвигом среднего в равномерном ядре, центрированном на  $x$ . Вектор выборочного сдвига среднего сонаправлен с вектором градиентом оценки плотности в точке  $x$ , в случае если оценка вычислена с применением ядра Епанечникова.

Так как вектор выборочного сдвига среднего направлен в сторону максимального увеличения плотности, он может определять путь к местному максимуму плотности. Схематичная иллюстрация работы алгоритма приведена на рисунке 6

Процедура сдвига среднего, работающая по следующему алгоритму, всегда

сходится:

- а) вычисление вектора сдвига среднего  $M_h x$ ,
- б) сдвиг окна  $S_h x$  на  $M_h x$ .

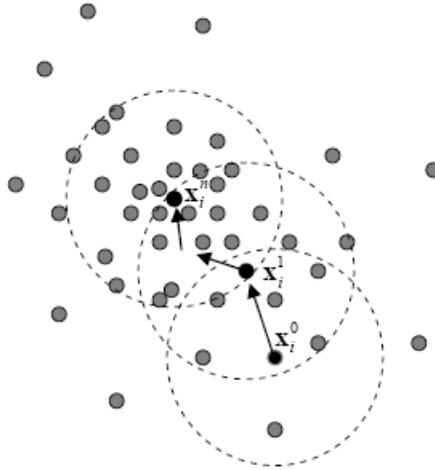


Рисунок 6 - Процедура сегментации на основе метода сдвига среднего

Изображения, подвергаемые обработке, представлены как двумерная матрица  $r$ -мерных векторов. Назовем двумерную область самого изображения областью пространства (spatial domain), а цветовую информацию – областью значений (range domain). После нормализации с использованием соответствующих среднеквадратичных отклонений  $\sigma_S$  и  $\sigma_R$  можно объединить эти две области в пространство размерностью  $d = r + 2$ . После проведения процедуры сдвига среднего для точки, она становится ассоциирована с некоей точкой сходимости.

Пусть  $x_j$   $j=1..n$  - точки изображения в области пространства – значений,  $z_j$   $j=1..n$  - точки сходимости, соответствующие каждой из точек изображения, а  $L_j$   $j=1..n$  - множество маркеров (скаляры). Предполагается, что изначально данные были нормализованы с использованием соответствующих среднеквадратичных отклонений  $\sigma_S$  и  $\sigma_R$ .

1) Для каждого  $j = \overline{1..n}$  запускается процедура сдвига среднего для  $x_j$  и записывается в  $z_j$ .

2) Находятся кластеры  $C_p$   $p=1..m$  точек сходимости. Для этого объединяются

все  $z_j$ , расстояние между которыми в области пространства – значений меньше  $\Delta$ .

3) Для каждого  $j = \overline{1..n}$  назначается  $L_j = p : z_j \in C_p$ .

### 1.3. Классификация областей на изображениях.

После проведения процедуры сегментации, когда изображение представляет собой множество непересекающихся областей необходимо произвести классификацию выделенных областей. Задачи синтеза и обучения алгоритмов классификации отсчетов выходного изображения по заданному вектору локальных признаков, соответствующих области, лежат в русле традиционной проблематики теории распознавания образов и, в принципе, могут иметь достаточно стандартные решения. Общая схема построения классификатора выглядит следующим образом. Вводятся предположения о статистических, структурных и др. свойствах признаков. Для каждого класса объектов по имеющимся данным строится дискриминантная функция, показывающая долю уверенности в принадлежности данной точки признакового пространства выбранному классу. Решение об отнесении конкретного вектора признаков к одному из классов принимается на основе сравнения значений дискриминантных функций по всем классам. Суммарный риск высчитывается как доля ошибочных ответов построенного решающего правила на обучающей выборке.

Однако предлагаемый подход к обработке информации требует выполнения классификации в весьма специфичных условиях. При синтезе классификатора не удастся использовать важные упрощающие предположения о линейной разделимости классов, гауссовских распределениях признаков и т.п., можно рассчитывать лишь на компактность классов в признаковом пространстве. При этом оказывается целесообразным аппроксимировать не дискриминантные функции для каждого класса, как это обычно делается в классификаторах, а непосредственно функцию решения.

В качестве таких классификаторов могут быть использованы различные стратегии, начиная от классификаторов по минимуму расстояния (Махаланобиса, Евклида и т.д.) и заканчивая иерархическими древовидными классификаторами и классификаторами основанными на нейронной сети.



### 1.3.1. Классификатор, основанный на расчете расстояний

Пусть все множество объектов/областей  $\Omega$  на изображении подразделяется на классы, соответствующие конкретным классам:

$$P_{\Omega} = \Omega_0, \Omega_1, \dots, \Omega_{L-1}, \quad \bigcup_{l=0}^{L-1} \Omega_l = \Omega.$$

Пусть, далее, каждая из областей  $X$  из всего множества областей  $\Omega$  представляется набором числовых характеристик, называемого *признаками*, которые формируют в совокупности *вектор признаков*  $\mathbf{Y} X = Y_0, Y_1, \dots, Y_{K-1}^T$ . Для построения правил, принимающих решение о близости некоторой конкретной области изображения определенному классу, производят разбиение *метрического пространства признаков*  $D = \mathbf{Y} : \mathbf{Y} \in D$  на множество непересекающихся областей:

$$P_D = D_0, D_1, \dots, D_{L-1}, \quad \bigcup_{l=0}^{L-1} D_l = D, \quad D_l \cap D_j = \emptyset, \quad l \neq j.$$

*Классификатором* называется решающее правило, предназначенное для указания, какой области  $D_l$  признакового пространства принадлежит предъявленный вектор признаков  $\mathbf{Y}$ .

Более удобным является задание классификатора не в виде областей признакового пространства, а в виде набора *дискриминантных* или *решающих функций*  $d_l \mathbf{Y} X$ ,  $l = \overline{0, L-1}$ . При этом процесс принятия решения осуществляется по следующему правилу: объект  $X$  считается принадлежащим классу  $\Omega_l$ , если дискриминантная функция этого класса  $d_l \mathbf{Y} X$  для вектора признаков  $\mathbf{Y} X$  объекта является максимальной, т.е.:

$$\forall j \neq l: d_l \mathbf{Y} X \geq d_j \mathbf{Y} X \Rightarrow \mathbf{Y} X \in D_l.$$

В случае, когда известной является дискриминантная функция только одного (текущего) класса, в этом случае для принятия решения используется сравнение дискриминантной функции с пороговым значением:

$$d_l \mathbf{Y} X \geq \text{Threshold} \Rightarrow \mathbf{Y} X \in D_l.$$

Качество классификатора в общем случае задается набором вероятностей:

$p_{lj}$   $l, j = \overline{0, L-1}$ . Каждая величины  $p_{lj}$  есть вероятность того, что классификатор принимает решение об отнесении вектора признаков  $\mathbf{Y}$   $X$  некоторого объекта (области изображения) к области  $D_j$ , в то время как сам объект  $X$  (область изображения) принадлежит классу  $\Omega_l$ :

$$p_{lj} = P \mathbf{Y} \in D_j / \Omega_l .$$

При  $l \neq j$  вероятности  $p_{lj}$  характеризуют ошибки распознавания и называются *вероятностями неверной или ошибочной классификации*, а при  $l = j$  вероятности  $p_{ll}$  задают вероятности *верной (правильной) классификации* представителей соответствующего класса. Уменьшение вероятностей ошибочной классификации – это основная задача, которая возникает при построении классификатора.

### 1.3.2. Классификатор по расстоянию Махаланобиса

Данный классификатор строится в предположении нормального закона распределения вектора признаков  $\mathbf{Y}$  в каждом из классов:

$$f \mathbf{Y} / \Omega_l = \frac{1}{2\pi^{K/2} \sqrt{|\mathbf{B}_l|}} \exp\left(-\frac{1}{2} \mathbf{Y} - \mathbf{M}_l^T \mathbf{B}_l^{-1} \mathbf{Y} - \mathbf{M}_l\right), \quad l = \overline{0, L-1},$$

где  $\mathbf{B}_l = M \mathbf{Y} - \mathbf{M}_l \mathbf{Y} - \mathbf{M}_l^T / \Omega_l$ ,  $\mathbf{M}_l = M \mathbf{Y} / \Omega_l$  – корреляционная матрица и математическое ожидание (среднее) вектора признаков из класса  $\Omega_l$ ,  $K$  – размерность вектора признаков  $\mathbf{Y}$ .

Следует отметить, что допущение о нормальности вектора признаков может быть принято, если признаки представляют собой некоторые интегральные характеристики исходной области изображения, например, коэффициенты спектрального преобразования.

#### Использование классификатора

В ситуации нормального распределения вектора признаков в каждом из классов известен явный вид байесовского (оптимального) классификатора. Дискриминантная функция классификатора представима в виде:

$$d_l(\mathbf{Y}) = \ln P \Omega_l - \ln \sqrt{|\mathbf{B}_l|} - \frac{1}{2} \mathbf{Y} - \mathbf{M}_l^T \mathbf{B}_l^{-1} \mathbf{Y} - \mathbf{M}_l, \quad l = \overline{0, L-1},$$

В предположении равных ковариационных матриц выражение может быть переписано в следующем виде:

$$d_l(\mathbf{Y}) = \ln P \Omega_l - \frac{1}{2} (\mathbf{Y} - \mathbf{M}_l)^T \mathbf{B}^{-1} (\mathbf{Y} - \mathbf{M}_l), \quad l = \overline{0, L-1}.$$

### Настройка классификатора

Для настройки классификатора требуется определить параметры распределения в каждом из классов и значение порога. Оценка параметров распределения в каждом из классов производится по следующим формулам, в которых вектора признаков берутся из соответствующего класса:

$$\mathbf{M} = \frac{1}{N} \sum_{n=1}^N \mathbf{Y}_n, \quad \mathbf{B} = \frac{1}{N} \sum_{n=1}^N (\mathbf{Y}_n - \mathbf{M})(\mathbf{Y}_n - \mathbf{M})^T = \frac{1}{N} \sum_{n=1}^N \mathbf{Y}_n \cdot \mathbf{Y}_n^T - \mathbf{M} \cdot \mathbf{M}^T$$

### **1.3.3. Метод иерархической классификации**

Изображение часто представляет собой набор объектов с различными характеристиками на неоднородном коррелированном фоне. Поэтому целесообразно для каждого объекта и фона строить независимые оценки выходных отсчетов, т.е. перейти к кусочно-линейному или кусочно-постоянному древовидному иерархическому классификатору. Синтез такого классификатора заключается в следующем. Каждый последовательный шаг разбиения включает следующие этапы:

- оценка перспективности деления вершин на основе ошибки аппроксимации в них;
  - нахождение оптимального порога разбиения признака;
  - определение признака, деление по которому, дает максимальное уменьшение ошибки;
  - деление (образование новых терминальных вершин);
  - проход по обучающей выборке и накопление статистик для новых вершин по элементам выборки, попавшим в них;
- построение функции аппроксимации и вычисление ошибки в новых вершинах.

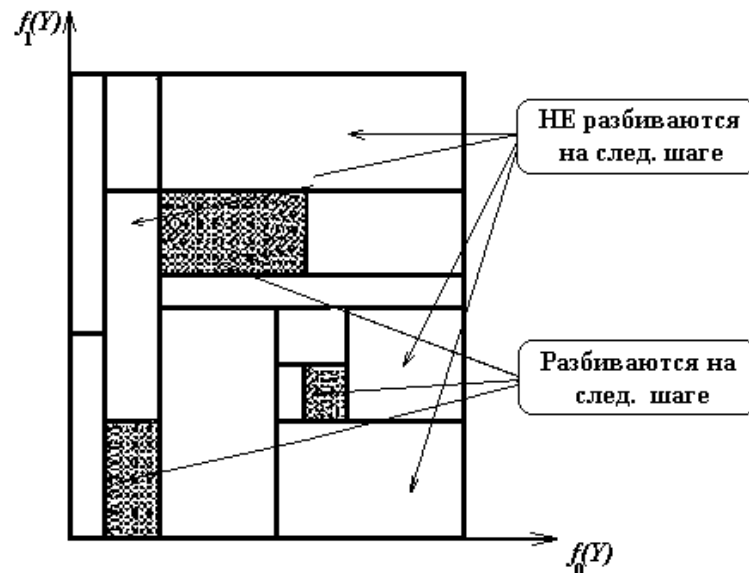


Рисунок 7 - Схема разбиения признакового пространства

Как показали эксперименты, такой классификатор при небольшой размерности признакового пространства достаточно быстро обучается и далее демонстрирует хорошие результаты распознавания при весьма незначительных вычислительных затратах.

Заметим, что качество построения классификатора в большой степени зависит от способа разбиения признакового пространства на подобласти и от подбора достаточно информативных признаков.

#### 1.3.4. Древоподобный классификатор с кусочно-постоянной аппроксимацией

Частным случаем линейной аппроксимации является аппроксимация постоянным значением. Пусть  $x_i$  – входной отсчет (идеальное изображение),  $y_i$  – выходной отсчет (искаженное изображение). Тогда, с учетом аппроксимации постоянным значением  $a$ , формула погрешности примет вид:

$$\varepsilon^2 = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) - a^2.$$

Минимальное значение среднеквадратичной ошибки  $\varepsilon^2$  достигается при:

$$a_{opt} = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2).$$

При использовании иерархического классификатора находится аппроксимация

$a_{onm}$   $a_{онт}$  в каждой из терминальных вершин (листьев) полученного дерева. При этом в  $i$ -ой терминальной вершине древовидного классификатора:

$$a_{i_{onm}} = \frac{1}{I} \sum_{j=0}^{I-1} x_j(n_1, n_2),$$

$$\varepsilon_i^2 = \frac{1}{I} \sum_{j=0}^{I-1} x_j(n_1, n_2) - a_{i_{onm}}^2,$$

где  $I$  – количество точек, попавших в  $i$ -ую терминальную вершину при разбиении.

Суммарная кусочно-постоянная погрешность аппроксимации обучающей выборки равна:

$$\varepsilon^2 = \frac{1}{N} \sum_{j=1}^M \varepsilon_j^2 * I_j = \frac{1}{N} \sum_{j=1}^M \sum_{l=0}^{I_j-1} x_l(n_1, n_2) - a_{j_{onm}}^2 = \frac{1}{N} \sum_{j=1}^M \left[ \sum_{l=0}^{I_j-1} x_l(n_1, n_2)^2 - (a_{j_{onm}})^2 \right],$$

где  $N=N_1N_2$  – количество точек изображения,  $M$  – количество терминальных вершин древовидного классификатора,  $I_j$  – количество точек изображения, попавших в  $j$ -ую терминальную вершину.

### 1.3.5. Классификация с использованием однослойной нейронной сети с обратным распространением

Функционирование нейронной сети (НС) имитирует функционирование человеческой нервной системы.

Рассмотрим нейронную сеть с  $K$  входами, одним скрытым слоем ( $M$  нейронов в скрытом слое) и одним выходом.

Отметим, что вершины-круги слева служат лишь для распределения входных сигналов. Они не выполняют каких-либо вычислений, и поэтому не считаются слоем. Каждый элемент из множества входов  $X$  отдельным весом соединен с каждым искусственным нейроном. А каждый нейрон выдает взвешенную сумму входов в сеть.

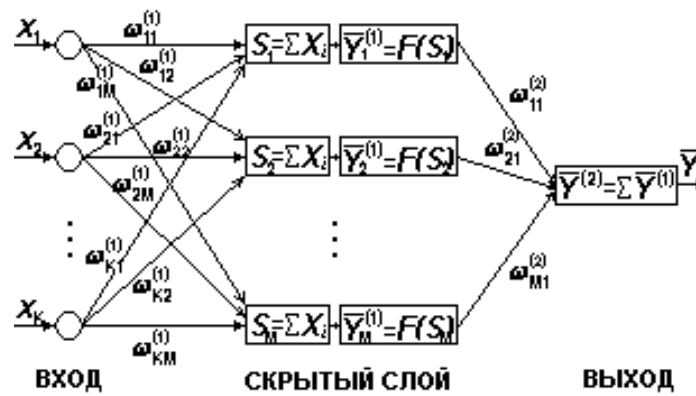


Рисунок 8 - Схема нейронной сети

Работа сети разделяется на *обучение* и *применение*. Под обучением понимается процесс адаптации сети к предъявляемым эталонным образцам путем модификации весовых коэффициентов связей между нейронами.

Для обучения использовался известный алгоритм обратного распространения ошибки (back propagation). При таком обучении распространение сигналов ошибки идет от выходов нейронной сети к ее входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы. Обучение заключается в подстройке весов – коэффициентов синоптических связей  $w_{ij}^{(k)}$ , где  $k$  - индекс номера слоя,  $i$  и  $j$  – индексы нейронов слоя  $k$  и  $(k+1)$ .

В случае двухслойной нейронной сети алгоритм обратного распространения выглядит следующим образом:

$$w_{ij}^{(k)} = w_{ij}^{(k)} - \eta \frac{\partial E^{(k)}}{\partial w_{ij}^{(k)}}, k = 1, 2,$$

$$\frac{\partial E^{(2)}}{\partial w_{ij}^{(2)}} = [\bar{Y}^{(2)}(n) - Y(n)] \bar{Y}_j^{(1)}(n),$$

$$\frac{\partial E^{(1)}}{\partial w_{ij}^{(1)}} = [\bar{Y}^{(2)}(n) - Y(n)] w_j^{(2)}(n) X_i(n) F' \bar{Y}_j^{(1)}(n),$$

где  $\eta$  – коэффициент скорости обучения,  $0 < \eta < 1$ .

В качестве активационной функции можно использовать следующие функции:

- функция Ферми:

$$y = f(s) = \frac{1}{1 + e^{-s}};$$

- гиперболический тангенс:

$$y = f(s) = \text{th } s ;$$

- рациональная функция:

$$y = f(s) = \frac{1}{1 + |s|} ;$$

- линейная функция:

$$y = f(s) = s$$

Критерий остановки итерационного процесса настройки весов – либо уменьшение ошибки до некоторого, наперед заданного, порога, либо отсутствие изменения ошибки в течение длительного времени.

### **1.3.6. Классификация с использованием нейронной сети Кохонена**

Самоорганизующиеся отображения (SOM), предложенные Кохоненом, являются однослойными нейронными сетями со слоем конкуренции, в которых входной вектор подается одновременно на входы всех нейронов. Единственный победивший нейрон, наиболее “соответствующий” входному вектору, определяется на основе одной из мер близости входного вектора и вектора весов каждого нейрона.

С другой стороны, мы интерпретируем вектора весов нейронов как центры классов, определяемых как дистанционное разбиение с использованием той же меры близости. Таким образом, устанавливается взаимно однозначное соответствие между нейронами нейронной сети и представляемыми ими классами в исходном пространстве признаков.

Обучение нейронной сети состоит в изменении вектора весов победившего нейрона (его сдвиге в пространстве признаков в направлении входного вектора). В отличие от обычных методов классификации, когда изменение отдельных классов производится независимо, в нейронной сети SOM все нейроны (а следовательно, и классы) упорядочены в некоторую топологическую решетку, задающую топологию нейронной сети. Эта топология не имеет отношения к положению векторов весов нейронов в исходном пространстве признаков и служит лишь для управления процессом обучения нейронной сети для ее самоорганизации. Использование топологической решетки состоит в определении локальной окрестности победившего

нейрона, в которую входят все расположенные рядом с ним в топологической решетке нейроны сети, независимо от близости их весов к входному вектору.

Главное отличие нейронной сети SOM от обычных систем классификации состоит в том, что кроме победившего нейрона обучаются и все нейроны в его локальной окрестности, что приводит к самоорганизации нейронной сети и выявлению внутренней структуры многослойного изображения. Именно выявление этой структуры и является результатом обучения нейронной сети.

### Калибровка нейронной сети

Обученная нейронная сеть выявляет в результате самоорганизации структуру исходного изображения, которая может использоваться для сегментации многослойных изображений. Однако наибольший интерес представляет соотнесение выявленной структуры и принадлежности пикселей к определенным тематическим классам.

Для такого соотнесения используется процесс тематической калибровки обученной нейронной сети, состоящий в предъявлении нейронной сети векторов, соответствующих пикселям с известной тематической интерпретацией. Нейронная сеть производит классификацию каждого такого вектора, и запоминает, какие тематические классы и в какой пропорции представлены в каждом классе нейронной сети. Фактически, при правильно построенной калибровочной выборке достаточного объема эти пропорции приближаются к вероятностям нахождения данного тематического объекта среди пикселей, отнесенных в данный класс нейронной сети, следовательно, вектор таких вероятностей для каждого класса нейронной сети может быть использован для дальнейшего построения процедур классификации как на основе теории вероятностей, так и с использованием нечеткой логики.

### Визуализация результатов самоорганизации сети

Для наглядного представления выявленной в результате обучения структуры многослойного изображения широко используется отображение Сэммона, при котором классы в исходном  $r$ -мерном пространстве изображения отображаются на плоскость так, что расстояния между образами классов равны расстояниям между самими классами. После нормировки полученных в результате отображения координат Сэммона, мы получаем инвариантное представление структуры исходного изображения (в смысле использованной меры близости). Нетрудно видеть, что такое

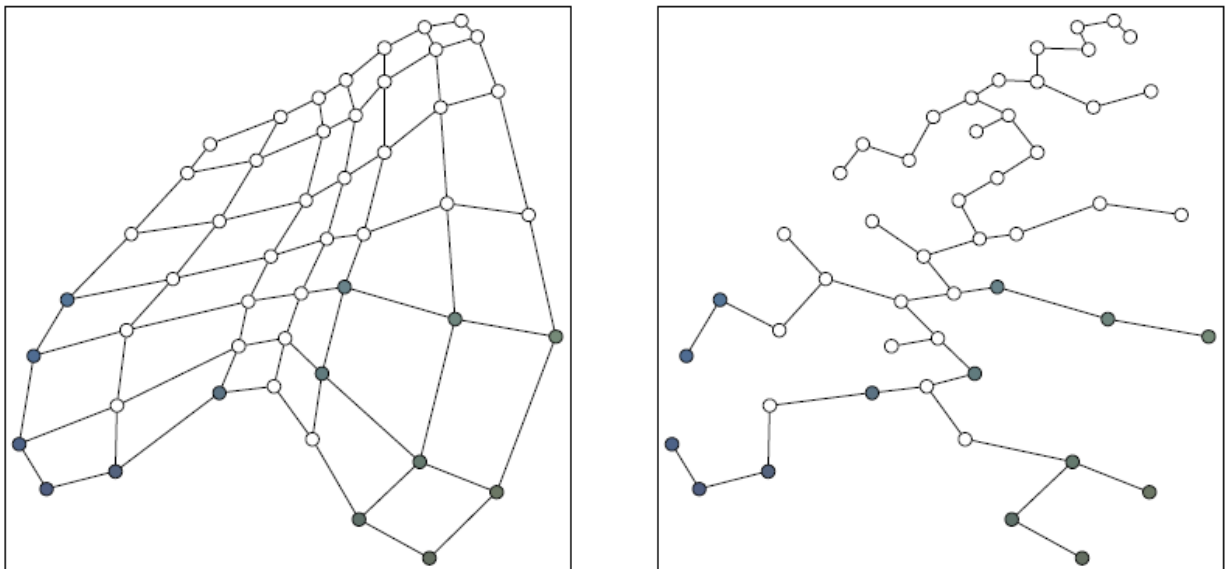


представление инвариантно относительно линейных преобразований значений отдельных плоскостей исходного изображения, если это преобразование не выводит значения за допустимый диапазон значений.

Поэтому, если мы используем координаты Сэммона для создания двух новых плоскостей изображения, то эти плоскости сохраняют всю информацию о структуре исходного изображения. Таким образом, обученная и тематически калиброванная нейронная сеть, координаты классов которой преобразованы в пространство Сэммона, может применяться для тематической интерпретации новых изображений, полученных при других атмосферных условиях (при условии, что это изображение тоже переведено в пространство Сэммона).

Для оценки качества построенной классификации и соответствия структуры нейронной сети, преобразованной в пространство Сэммона, структуре исходной нейронной сети, используется построение минимального остового дерева (как с использованием всей матрицы расстояний между классами, так и только расстояний в решетке нейронов), которое выводится поверх построенного отображения Сэммона.

Отсутствие самопересечений построенного дерева говорит об успешной самоорганизации нейронной сети и о возможности использования координат Сэммона для дальнейшей обработки.



*Рисунок 9 - Отображение Сэммона для обученной нейронной сети и изображение минимального остового дерева*

### 1.3.7. Алгоритм ISODATA

В качестве алгоритма автоматической классификации данных дистанционного зондирования наиболее часто используется алгоритм ISODATA (*Iterative Self-Organizing Data Analysis Techniques*). Данный алгоритм позволяет автоматически настраивать число кластеров во время итеративного процесса путем слияния схожих по заданной мере кластеров и расщепления кластеров с большим внутренним среднеквадратичным отклонением. Данный алгоритм настраивается следующими параметрами:

$K$  - желаемое число кластеров;

$I$  - максимально допустимое число итераций;

$P$  - максимальное число пар объединяемых кластеров;

$\Theta_N$  - пороговое значение для минимального числа элементов в кластере (используется для удаления малых кластеров);

$\Theta_S$  - пороговое значение для среднеквадратичного отклонения (используется для операции расщепления);

$\Theta_C$  - пороговое значение для межклассового расстояния (используется для операции слияния).

Сам алгоритм состоит из следующих шагов:

*Шаг 1.* Произвольный выбор  $k$  (не обязательно  $K$ ) начальных центров кластеров  $M_1, M_2, \dots, M_k$  из исходной выборки  $X_i, i = 1, 2, \dots, N$ .

*Шаг 2.* Определение каждого из  $N$  элементов выборки в ближайший кластер:

$$X \in \omega_j, \text{ если } D_L(X, M_j) = \max_i D_L(X, M_i), \quad i = 1, \dots, k$$

*Шаг 3.* Удаление кластеров с числом элементов меньше  $\Theta_N$ . Если для некоторого  $j$  выполнено  $N_j < \Theta_N$ , то кластер  $\omega_j$  удаляется,  $k$  уменьшается на единицу.

*Шаг 4.* Обновляются центры кластеров:

$$M_j = \frac{1}{N_j} \sum_{X \in \omega_j} X \quad (j = 1, \dots, k).$$

*Шаг 5.* Рассчитываются средние расстояния  $D_j$  от каждого элемента кластера  $\omega_j$  до соответствующего центра кластера:

$$D_j = \frac{1}{N_j} \sum_{X \in \omega_j} D_L(X, M_j) \quad (j=1, \dots, k).$$

*Шаг 6.* Вычисляется общее среднее расстояние от элементов до центра соответствующего кластера:

$$D = \frac{1}{N} \sum_{j=1}^k N_j D_j.$$

*Шаг 7.* Если  $k \leq K/2$  (кластеров мало), то переходим к шагу 8. Если  $k > 2K$  (кластеров много), то переходим к шагу 11, иначе – к шагу 12.

*Шаг 8.* Первый шаг расщепления. Находится вектор стандартного отклонения

$$\Sigma_j = \left[ \sigma_1^{(j)}, \dots, \sigma_n^{(j)} \right]^T$$

для каждого кластера:

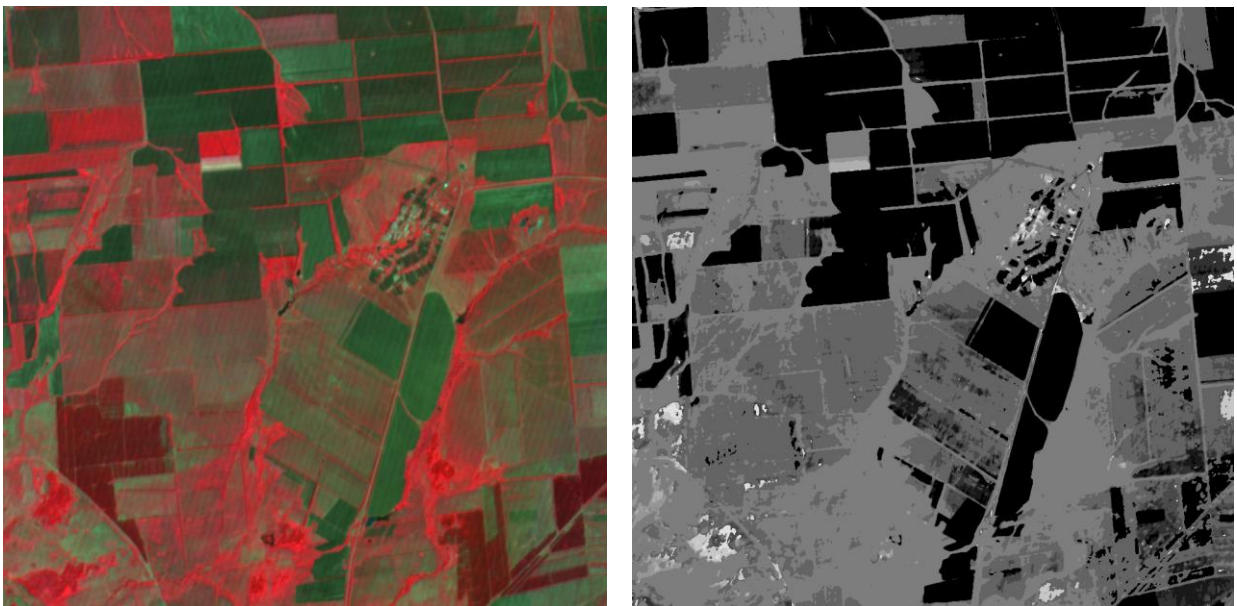
$$\sigma_i^{(j)} = \sqrt{\frac{1}{N_j} \sum_{X \in \omega_j} (x_i - m_i^{(j)})^2}, \quad \begin{cases} i = 1, \dots, n \\ j = 1, \dots, k \end{cases}.$$

*Шаг 9.* Находится максимальное из  $\Sigma_j$ .

*Шаг 10.* Расщепление кластера.

*Шаг 11.* Слияние кластеров по принципу Min расстояния между их центрами.

*Шаг 12.* Остановка в случае превышения Max числа итераций.



*Рисунок 10 - Пример работы алгоритма автоматической классификации*

#### **1.4. Выявление изменений на изображениях.**

В результате классификации областей и принятых решений о соответствии областей классам зачастую возникает задача, связанная с наблюдением конкретных объектов/областей на последовательности изображений со спутников, то есть возникает необходимость анализа динамики этих областей. Особенно часто такие задачи встают при исследовании различных объектов человеческой деятельности и анализе их влияния на окружающую среду. В этом случае возникает необходимость создания и использования специальных алгоритмов обработки изображений, позволяющих выделять зоны изменчивости по сериям изображений.

Основные трудности поиска различий (зон изменчивости) на спутниковых снимках одного и того же района, сделанных в различные моменты времени, связаны, в первую очередь, с разными условиями съемки, из-за чего различные снимки могут иметь:

- разное пространственное разрешение (из-за того, что они были сделаны под разными углами сканирования);
- различную яркость (из-за съемки при разных условиях освещенности);
- несколько отличающиеся калибровки (из-за съемки с различных космических аппаратов).

Конечно, эти различия обычно достаточно малы, и не влияют на исследование динамики крупных объектов. Однако на выделение изменений на пределе пространственного разрешения или на пределе чувствительности приборов влияют даже незначительные изменения условий съемки. Поэтому в таких случаях не всегда удается найти области изменений, пользуясь простым анализом разностей исследуемых изображений, так как в этом случае очень трудно отделить изменения, связанные с изменением условий съемки, от изменений реальных объектов. Поэтому при поиске зон изменений требуется использовать специальные алгоритмы. Эти алгоритмы должны, в первую очередь, обладать устойчивостью к изменениям условий съемки и калибровки.

##### **1.4.1. Разностный метод выделения изменений**

Первый метод является одним из широко распространенных подходов по выявлению различий между снимками. Он основывается на вычислении разностного

изображения между двумя изображениями и осуществлении пороговой обработки. Маска изменений при этом вычисляется следующим образом:

$$B(x) = \begin{cases} 1, & |D(x)| > \tau, \\ 0, & \text{иначе.} \end{cases}$$

Далее изображения, между которыми производится поиск изменений будем обозначать как  $I_1$  и  $I_2$ . Разностное изображение  $D(x)$  при этом может вычисляться одним из следующих методов:

$$D_1(x) = I_1 - I_2, \quad D_2(x) = \frac{I_1}{I_2}.$$

На основе проведенных исследований было получено, что использование  $D_2(x)$  приводит к лучшим результатам для выделения изменений на сельскохозяйственных объектах. Следует отметить, что данный метод не обеспечивает инвариантности к линейным изменениям между снимками, поэтому его целесообразно применять для выделения зон изменений на участках полей, сельскохозяйственных угодьях и т.д. с целью получения данных по динамике роста культур. Порог  $\tau$  в данном случае может быть выбран эмпирически.

Получаемая маска изменений при этом имеет вид, показанный на рисунке 11.

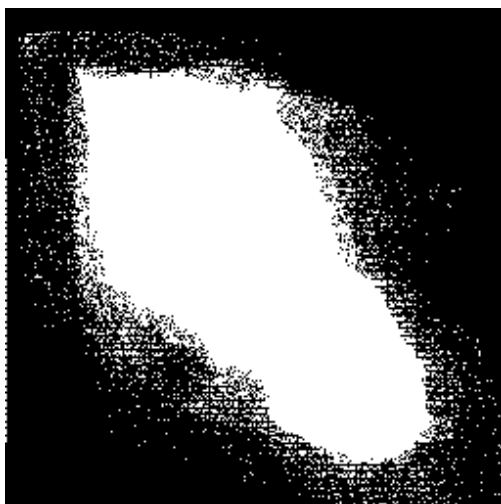


*Рисунок 11 - Результат работы разностного метода выделения изменений*

Для многокомпонентных изображений возможно применение метода на основе векторов изменений. При этом в качестве тестирующей статистики выступает модуль разности между векторами признаков, полученных по значениям яркости изображений.

### 1.4.2. Метод главных компонент

В данном случае использовался метод анализа главных компонент (РСА), называемый также методом анализа временных гистограмм. Основная идея алгоритма состоит в анализе временных двумерных гистограмм. Под двумерными временными гистограммами мы будем понимать гистограммы, построенные по изображениям одного и того же участка поверхности, полученным в разные моменты времени. Пример расчета гистограммы приведен на рисунке 12.



*Рисунок 12 - Пример двумерной временной гистограммы*

По осям гистограмм отложены яркости анализируемых изображений. Если два снимка совершенно не отличаются друг от друга, то гистограмма полностью сосредоточена вдоль диагонали. В реальных изображениях одних и тех же объектов, полученных в разных условиях, точки гистограммы уже не лежат точно на диагонали. Так, например, если снимки были сделаны в одинаковых условиях освещения и одинаково откалиброваны, но имеют несколько различное пространственное разрешение, значащие точки гистограммы группируются вдоль диагонали, но уже расположены не только на ней. В ситуации, когда снимки имеют различную калибровку, значащие точки гистограммы группируются вдоль прямой, не проходящей по диагонали. Однако во всех случаях, когда на снимках присутствуют только одни и те же неизменные объекты, точки на гистограммах будут расположены вдоль одной ("основной") прямой.

Если какие-то объекты на снимках изменяются и появляются новые, на гистограммах появляются области, расположенные не на прямой, определяемой главным направлением.

Исходя из приведенных примеров, легко понять, как должен быть построен алгоритм выделения зон изменений на снимках. Его основной целью является нахождение на двумерной гистограмме "основной" прямой, связанной с неизменными объектами, и построение двумерной таблицы преобразований, которая позволяла бы удалить с изображения все точки, расположенные в окрестности этой прямой, и оставить только изменяющиеся точки.

Пример работы данного метода приведен на рисунке 13.

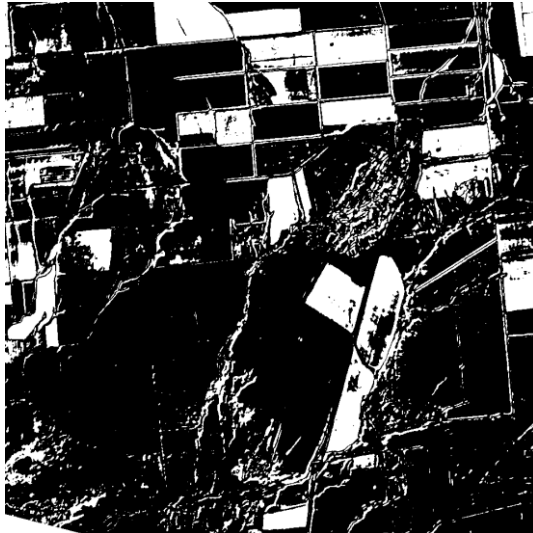


Рисунок 13 - Результат метода анализа главных компонент

Следует отметить, что ввиду того, что данный метод использует данные о положении главной оси временной гистограммы, он является инвариантным к линейным изменениям между изображениями и может быть использован для выделения яркостных вариаций, представляющих собой зоны пожаров и зоны сильного изменения роста сельскохозяйственных культур.

### 1.4.3. Методы на основе пространственных моделей

Одним из подходов к выделению изменений является подход на основе локальной аппроксимации значений яркости на изображениях некоторыми полиномиальными функциями:

$$\hat{I}_k(x, y) = \sum_{i=0}^p \sum_{j=0}^{p-i} \beta_{ij}^k x^i y^j$$

Изменения выделяются путем проверки качества аппроксимации двух изображений полиномами с одинаковыми коэффициентами. Коэффициенты полиномиальной модели получаются путем использования метода наименьших

квадратов. В качестве критерия изображений может использоваться следующая статистика:

$$F(x) = \frac{\sigma_0^{2N}}{\sigma_1^N \sigma_2^N},$$

где  $\sigma_1^2$  - дисперсия отклонений значений яркости первого изображения от полиномиальной модели в точке,

$\sigma_2^2$  - дисперсия отклонений значений яркости второго изображения от полиномиальной модели в точке,

$\sigma_0^2$  - дисперсия отклонений значений яркости смеси двух распределений от полиномиальной модели.

Также распространенным является метод с использованием следующей статистики для выявления изменений:

$$T(x) = \sum_{y \in \Omega_x} \left( \frac{\partial \hat{I}_1}{\partial x}(y) - \frac{\partial \hat{I}_2}{\partial x}(y) + \frac{\partial \hat{I}_1}{\partial y}(y) - \frac{\partial \hat{I}_2}{\partial y}(y) \right),$$

где  $\hat{I}_j(x)$  - значения аппроксимирующей квадратичной функции в данной точке.



## 2. ПРОГРАММНЫЕ СРЕДСТВА ТЕМАТИЧЕСКОЙ ОБРАБОТКИ КОСМИЧЕСКИХ СНИМКОВ

### 2.1. Программные средства сегментации.

#### ISODATA

Алгоритм выполняет кластеризацию исходного изображения по алгоритму ИСОМАД. На выходе получаем два изображения: 1 – полутоновое изображение, где яркость каждого пикселя соответствует номеру кластера, которому он принадлежит; 2 – цветное изображение, значение пикселя равно мат. ожиданию пикселей кластера, к которому относится текущий пиксель.

Возможны два варианта работы программы:

– в качестве исходных центров кластеров берутся первые  $N_c$  пикселей исходного изображения;

– центры кластеров определяются по индексному изображению, если значение пикселя на этом изображении равно  $i \geq 1$ , значит, соответствующий пиксель исходного изображения перед началом кластеризации будет отнесен к  $i$ -ому кластеру, а центром будет мат. ожидание пикселей попавших в этот кластер.

Алгоритм был реализован в виде программного модуля *iso.dll*.

Параметры вызова модуля:

Имя	Описание	Тип	Значение
<i>ImageIn</i>	Входное изображение	PT_IMAGENAME	ImgIn
<i>ImageOut</i>	Выходное изображение	PT_IMAGENAME	Imgout
<i>K</i>	желаемое число кластеров	PT_INTEGER	20
<i>I</i>	максимальное число итераций алгоритма	PT_INTEGER	1000
<i>P</i>	максимальное число пар объединяемых кластеров	PT_INTEGER	3
<i>thetaN</i>	минимальное число элементов в кластере	PT_INTEGER	10
<i>Thetas</i>	максимальная дисперсия в кластере	PT_FLOAT	1.2
<i>thetaC</i>	минимальное расстояние между кластерами	PT_FLOAT	1.1

Необязательные параметры вызова модуля:

Имя	Описание	Тип	Значение
<i>IndexImg</i>	индексное изображение, задающее	PT_IMAGENAME	ImgCl

	первоначальное разбиение исходного изображения на кластеры		
$N_c$	первоначальное число кластеров (используется, если индексное изображение не задано)	PT_INTEGER	20

Пример вызова программы:

- исходные кластеры – первые  $N_c$  пикселей исходного изображения

*RL ISO IsoData test.tif out\_test.tif 10 10 1 1 1 4  $N_c=4$*

- исходные кластеры определяются согласно индексному изображению

*RL ISO IsoData test.tif out\_test.tif 10 10 1 1 1 4 IndexImg="test\_Ind.bmp"*

## 2.2. Программные средства классификации.

### Иерархический классификатор.

Иерархический классификатор реализован в виде программного модуля *Klasifik.exe*.

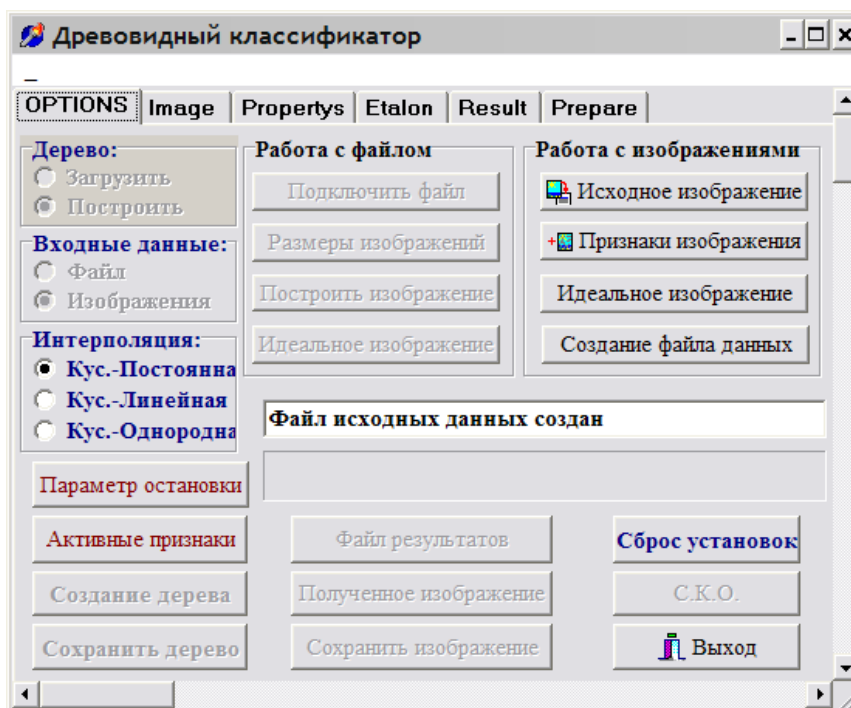


Рисунок 14 - Иерархический классификатор

Входные данные для создания древовидного классификатора могут быть представлены в двух видах – в виде текстового файла, каждая строка которого содержит одну точку в признаковом пространстве, либо в виде набора изображений.

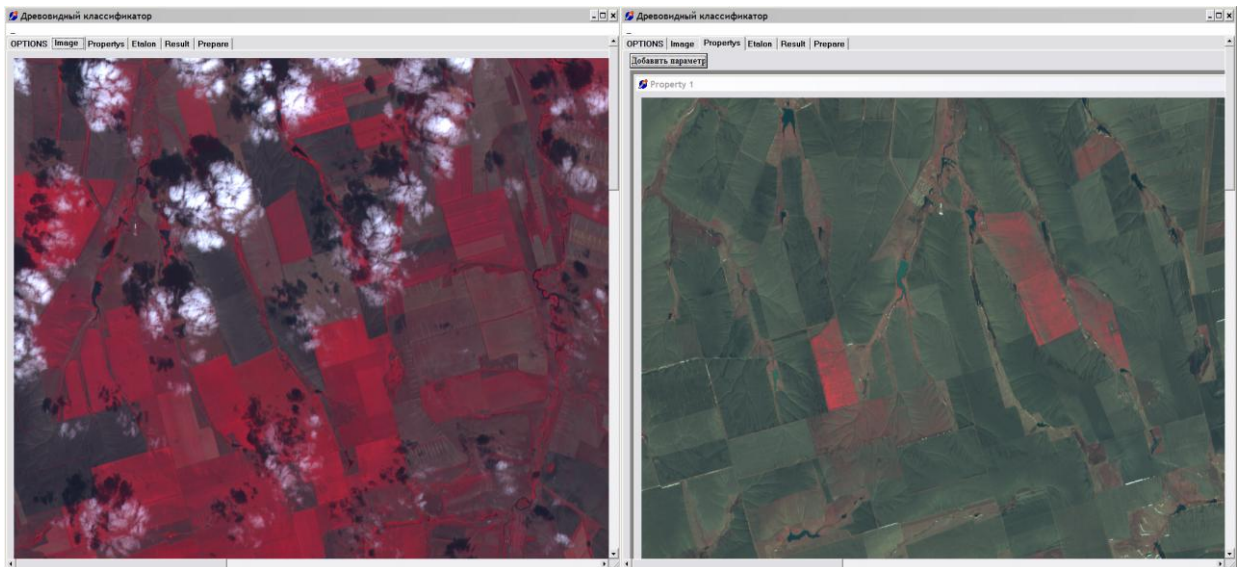


Рисунок 14 - Исходное изображение и изображения-признаки для классификации

Можно выбрать формат входных данных, вид аппроксимирующей функции в терминальных вершинах. После подключения входных данных необходимо выбрать *активные признаки* и *параметр остановки процесса разбиения*, а также количество разбиваемых на каждом шаге терминальных узлов. После этого возможно построение древовидного классификатора, его сохранение, использование, визуализация (на закладке Prepare).

После построения древовидного классификатора в файл *sko.dat* сохраняется информация об активных признаках, способе аппроксимации в узлах и среднеквадратическом отклонении от эталона (изображения определяющего классы). Кроме того, возможно сохранение (и последующая загрузка) древовидного классификатора в текстовом файле (с расширением *.tre*). При сохранении в файл сначала записывается вид аппроксимации в терминальных вершинах древовидного классификатора, потом общее количество признаков, количество признаков (активных), по которым построен древовидный классификатор, после чего перечислены номера активных признаков и максимальное и минимальное значения каждого из них. После чего сохраняется сам древовидный классификатор.

Затем возможно использование классификатора на других изображениях.

### ISODATA

Алгоритм ISODATA, кроме кластеризации изображения предназначен так же для классификации, реализован в виде программного модуля *iso.dll*, содержащий функции:

- IsoData – классификация RGB-изображения;
- IsoFloat – классификация float-изображения.

Параметры вызова модуля:

Имя	Описание	Тип	Пример
<i>ImageIn</i>	Входное изображение	PT_IMAGENAME	ImgIn
<i>ImageOut</i>	Выходное изображение	PT_IMAGENAME	Imgout
<i>K</i>	желаемое число классов	PT_INTEGER	20
<i>I</i>	максимальное число итераций алгоритма	PT_INTEGER	1000
<i>P</i>	максимальное число пар объединяемых классов	PT_INTEGER	3
<i>thetaN</i>	минимальное число элементов в классе	PT_INTEGER	10
<i>thetaS</i>	максимальная дисперсия в классе	PT_FLOAT	1.2
<i>thetaC</i>	минимальное расстояние между классами	PT_FLOAT	1.1

### 2.3. Программные средства выявления изменений.

#### На основе разностей

Реализован программный модуль *differ.dll*, содержащий функции:

- CVA – разностный метод на основе векторов;
- RedDiff – метод на основе вычисления разности яркостей;
- Ratio – метод на основе вычисления отношения яркостей.

Параметры вызова модуля:

Имя	Описание	Тип	Пример значения
<i>Image1</i>	Входное изображение 1	PT_IMAGENAME	ImgIn1
<i>Image2</i>	Входное изображение 2	PT_IMAGENAME	ImgIn2
<i>ImageOut</i>	Выходное изображение	PT_IMAGENAME	Imgout

#### На основе гистограммы

Реализован программный модуль *pca.dll*, вычисляющий разностное поле на основе гистограммы распределения

Параметры вызова модуля:

Имя	Описание	Тип	Пример значения
<i>Image1</i>	Входное изображение 1	PT_IMAGENAME	ImgIn1
<i>Image2</i>	Входное изображение 2	PT_IMAGENAME	ImgIn2
<i>Hist</i>	Гистограмма распределения яркости	PT_IMAGENAME	ImgHist
<i>Field</i>	Выходное изображение	PT_IMAGENAME	Imgout

#### На основе пространственной модели

Реализован программный модуль *sp.dll*, вычисляющий разностное поле на основе пространственной модели. Реализованные методы:

- Hsu – пространственной аппроксимации полиномами второго порядка;
- Skifstad – метод на основе производных.

Параметры вызова модуля:

Имя	Описание	Тип	Пример
<i>Image1</i>	Входное изображение 1	PT_IMAGENAME	ImgIn1
<i>Coeffs1</i>	Коэффициенты полиномиального разложения 1	PT_ARRAY	
<i>Image2</i>	Входное изображение 2	PT_IMAGENAME	ImgIn2
<i>Coeffs2</i>	Коэффициенты полиномиального разложения 2	PT_ARRAY	
<i>Size</i>	Размер окна расчета	PT_integer	ImgHist
<i>ImgOut</i>	Выходное изображение	PT_IMAGENAME	Imgout

Кроме того, для всех модулей расчета изменений имеется ряд необязательных параметров, повышающих эффективность работы модулей:

Дополнительные параметры вызова модулей:

Имя	Описание
<i>FieldsImage</i>	Позволяет задать изображение с растровой маской векторных объектов, для которых необходимо произвести анализ изменений.

Имя	Описание
<i>UseVectImg</i>	Флаг использования векторных объектов из маски <i>FieldsImage</i> для анализа изменений
<i>ChannelNum</i>	Определяет канал, по которому осуществляется анализ
<i>BlueThreshold</i>	Визуальное отображение результирующего дифференциального поля
<i>К</i>	Коэффициенты визуального отображение результирующего дифференциального поля

## 2.4. Стороннее программное обеспечение – ScanEx ImageProcessor

### Сегментация/классификация :

- **Исодата** – сегментация, неуправляемая классификация.

Порядок выполнения процедуры.

1. Откройте изображение на требуемую территорию с автоматическим определением проекции и разрешения (например Landsat\_7).
2. Загрузите в программу векторный слой mask.mif. Он является маской, ограничивающей область классификации.
3. Для проведения классификации откройте диалог **ISODATA Classification** (рис. 15) командой меню **Classification->Unsupervised->IsoData**.

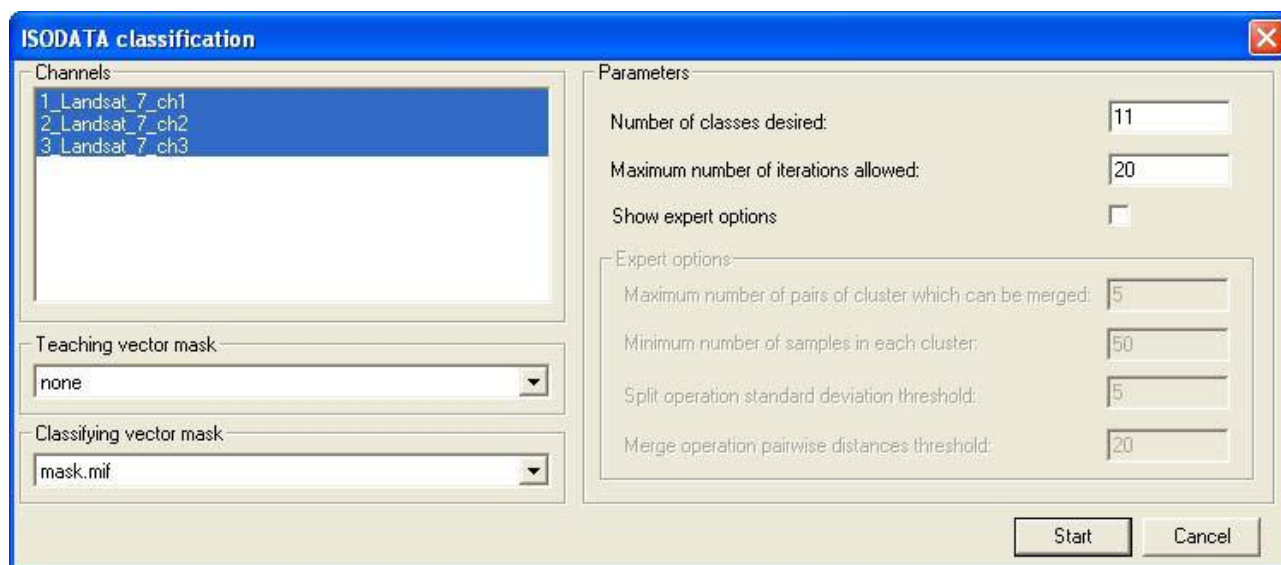


Рисунок 15 – Диалог ISODATA

4. В диалоге **ISODATA Classification** в списке каналов **Select Channels** укажите

спектральные каналы, которые необходимо обработать. В данном случае это будут все каналы снимка.

5. В группе **Parameters** настройте параметры обработки:
  - a. Определите примерное количество классов объектов, например, 11 и задайте их количество в поле **Number of clusters desired**.
  - b. Остальные параметры оставьте заданными по умолчанию.

В списке **Teaching vector mask** указывается векторный слой, по которому проводится обучение. Это важно в случаях, когда задается небольшое число классов и при этом необходимо «не потерять» отличающиеся по яркости объекты. В данном случае векторный слой отсутствует, поэтому оставьте в списке значение «**none**».

В списке **Classifying Vector mask** задайте границу области, которая будет использоваться при классификации. В качестве границы может быть использована векторная маска или определенная прямоугольная область.

Укажите векторный слой **mask.mif**.

После задания параметров нажмите кнопку **Start**.

6. Настройте палитру полученного изображения и оцените точность сегментации/классификации. Для этого откройте в двух рабочих окнах трехканальный снимок и классифицированное изображение и выполните команду **Window->Tile Vertical**.
7. Сохраните полученное изображение с названием **isodata.tif**

#### Классификация:



- **Neural Network** – классификация на основе нейронных сетей прямого распространения.

Для выполнения классификации **Neural Network** по алгоритму нейронных сетей прямого распространения требуется первоначально произвести обучение нейронной сети..

Порядок выполнения процедуры.

1. Откройте изображение на требуемую территорию с автоматическим определением проекции и разрешения (например Landsat\_7).
2. Откройте диалог **Vecor Control Layers** и создайте новый векторный слой, который должен содержать поле с переменными типа **integer**. В качестве

примера создайте слой **classification.mif**, содержащий поле **Code** с переменными типа **integer**. Данные этого столбца будут использоваться при обучении нейронной сети. Сохраните новый слой и загрузите его в программу. Обратите внимание на то, что слой должен быть редактируемым. Также загрузите в программу векторный слой, ограничивающий область классификации. В данном случае это слой **mask.mif**.

3. Определите участки, которые будут являться эталонными при обучении нейронной сети. Можно выделить 11 типичных видов территории:
  - a. водная поверхность (1)
  - b. 3 типа растительности (2, 3, 4)
  - c. 3 типа полей (5, 6, 7)
  - d. 2 типа застройки (8 и 9)
  - e. 2 типа открытых грунтов (10 и 11)
4. Выполните команду **VectorVector->Editor** или нажмите кнопку  панели инструментов. Убедитесь в возможности редактирования созданного слоя и оконтурьте выбранные участки инструментом  – **Add Region**. Каждому объекту задайте порядковый номер в поле **Code**.

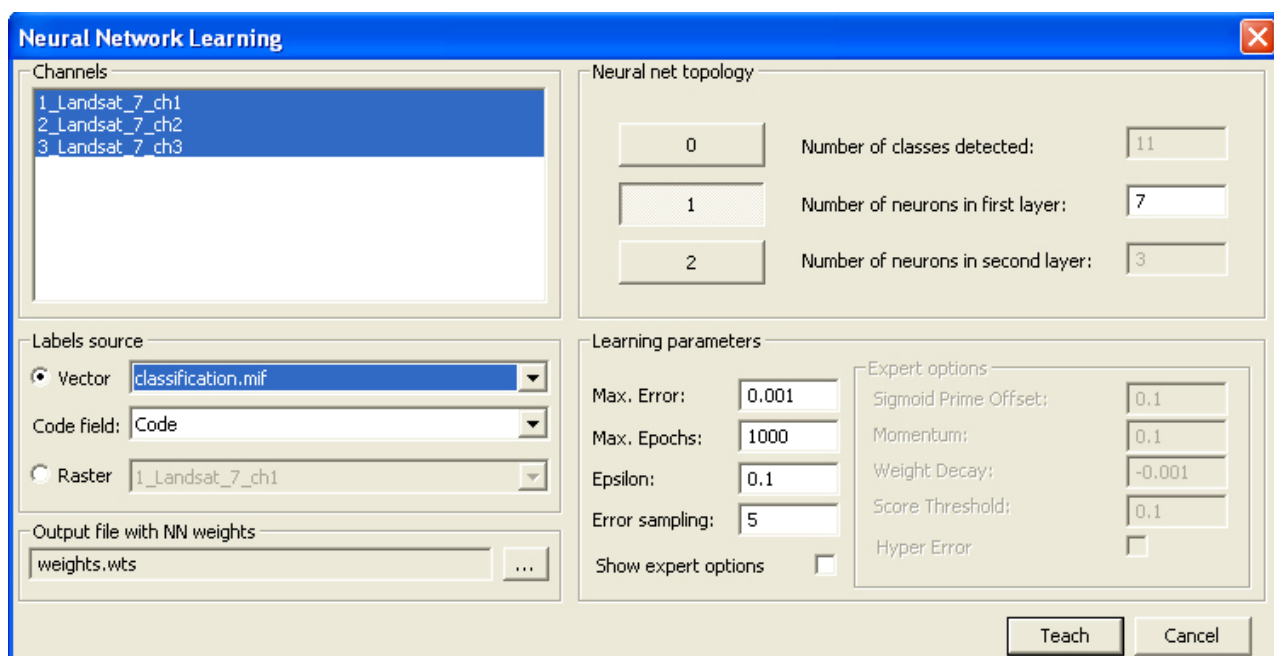


Рисунок 16 – Диалог *Neural Network Learning*

В поле **Select Input Channels** выберите используемые для классификации каналы снимка.



Остальные параметры оставьте заданными по умолчанию.

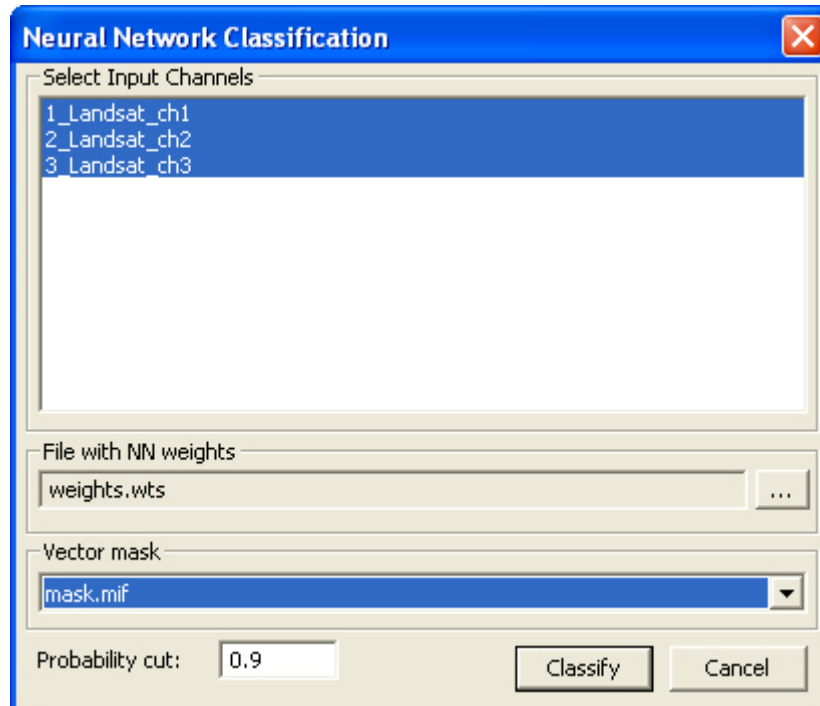


Рисунок 17 – Диалог *Neural Input Classification*

В группе **Select Input Channels** укажите используемые для классификации каналы. Обратите внимание на то, что они должны соответствовать каналам, по которым проходило обучение нейронной сети.

В поле **Vector mask** укажите векторный слой **mask.mif**.

В поле **Probability cut** указывается значение вероятности, которое определяет надежность классификации.

Остальные параметры оставьте заданными по умолчанию.

Для начала классификации нажмите кнопку **Classify**.

5. Настройте палитру полученного изображения.
6. Проведите автоматическую векторизацию полученных классов объектов. В диалоге **Vector Layers Control** настройте параметры отображения векторных объектов. Сохраните векторный файл.
7. Сохраните полученное изображение.

Анализ изменений:

- **Substraction** – вычитание яркостей одного снимка из яркостей другого;
- **Division** – деление яркостей одного снимка на яркости другого, если предполагается мультипликативное влияние искомым изменений на яркости;
- **PCA** – альтернативный способ, основанный на использовании метода

главных компонент изображений, или выявление направления максимальной изменчивости (в данном методе яркости одной точки на двух разновременных снимках рассматриваются как одна точка в многомерном (в данной случае 2-D) пространстве; кроме того, в методе реализована возможность оценки степени локального изменения пространственной разнесенности).

Условия проведения процедуры:

- проведение радиометрической нормализации изображений;
- снимки должны быть одного сезона на исследуемую территорию;
- снимки должны быть одного пространственного разрешения.

Порядок выполнения процедуры.

1. Откройте два изображения одной и той же съемочной системы на требуемую территорию.

2. Перед проведением *Change Detection* должна быть произведена геометрическая коррекция снимков. Так как снимки одного разрешения и незначительно разнесены друг от друга, то можно воспользоваться автоматическим режимом совмещения *Co-Registration (Transformation->Co-Registration)* с оценкой ошибки коррекции через опцию *List of GCPs*.

3. Выполните команду **Edit->Change Detection** главного меню программы, в результате будет открыт диалог **Change Detection**. Диалог состоит из двух закладок (см. рисунок 18).

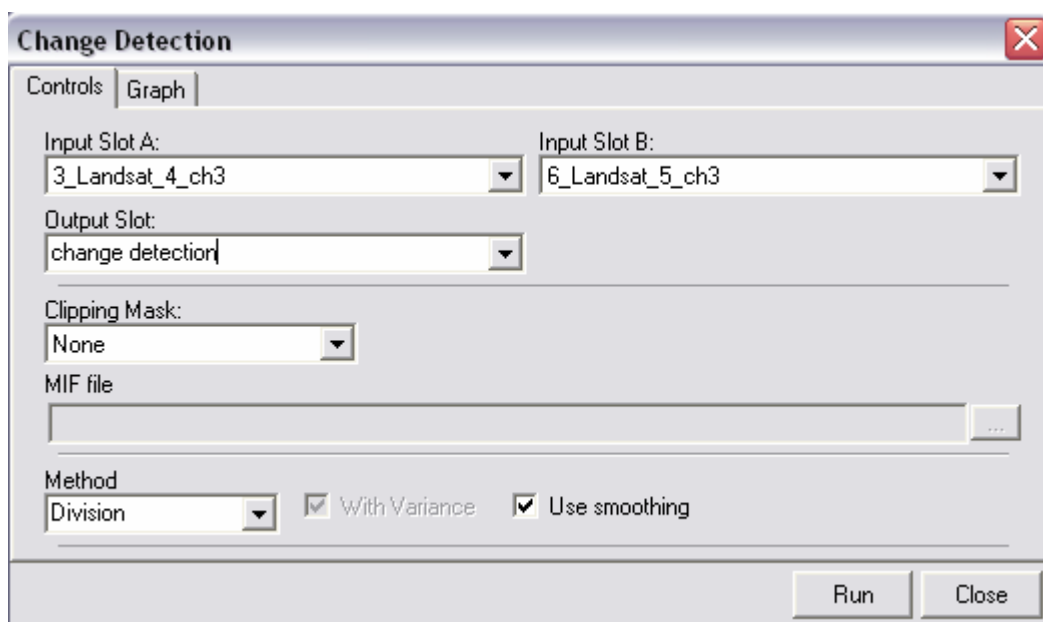


Рисунок 18 - Диалог *Change Detection*

В закладке *Controls* задайте:

- в списке **Input Slot A** – нужный канал изображения **1**, в списке **Input Slot B** – соответствующий канал снимка **2**, а в поле **Output Slot** - имя результирующего канала;

- в списке **Method** метод **Division**, так как изменения, зафиксированные на снимках однозначны, и включите опцию **Use smoothing** для сглаживания результата гауссовским фильтром;

- в списке **Clipping Mask** значение **None**.

Вторая закладка **Graph** диалога отображает распределение исходных точек и оси 1-ой и 2-ой главных компонент.

После задания параметров нажмите на кнопку **Run** для запуска операции, после чего закройте диалог кнопкой **Close**.

Правильное сохранение результатов операции **Change Detection** с черно-белой палитрой и с исходными значениями обеспечивает выбор в диалоге **Save Data** в списке **Output Pixel Format** формата **float32** (вещественный).

### 3. ЛИТЕРАТУРА

1. Вудс, Р. Цифровая обработка изображений [Текст]/ Р. Вудс, Р.Гонсалес — М.: Техносфера, 2005. — 1072 с.
2. Сойфер, В.А. Теоретические основы цифровой обработки изображений [Текст] /В.А. Сойфер, В.В. Сергеев, С.Б. Попов, В.В. Мясников. — Самарский государственный аэрокосмический университет, Самара, 2000 г. — 256 с.
3. ИТЦ «СканЭкс» ImageProcessor – программа обработки изображений. Руководство пользователя, Москва 2009 г. 287 с. [www.scanex.ru](http://www.scanex.ru)
4. ИТЦ «СканЭкс» ImageProcessor – “Упражнения по анализу и обработке изображений”, Москва 2009 г. 101 с.
5. Niemeier I., Canty M., Klaus D. Unsupervised change detection techniques using multispectral satellite images //Proc. IEEE Int. Geoscience and Remote Sensing Symp. – 1999. - pp. 327–329.
6. C. Yang, R. Duraiswami, D. DeMenthon, L. Davis. Mean-Shift Analysis Using Quasi-Newton Methods, In IEEE International Conference on Image Processing, pp. 447–450, vol.3, 2003
7. T. Kanungo, D.M. Mount, N. Netanyahu, C. Piatko, R. Silverman and A. Y. Wu, efficient k-means clustering algorithm: Analysis and implementation.” IEEE Trans. Pattern Analysis and Machine Intelligence, 24 (2002), 881-892

### 4. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

#### 4.1. Исходные данные

- Космические снимки с различных космических аппаратов с различным разрешением (Terra, Spot, Landsat) – исходные данные предоставляется преподавателем;
- Программные модули сегментации (Исодата), классификации (иерархический классификатор), выявления изменений (разностный, по гистограмме, по пространственной модели).
- Библиотеки ResLook.
- Параметры настройки программного обеспечения – ScanEx Image Processor .

## 4.2. Общий план выполнения работы

1. Ознакомьтесь с руководством пользователя программного обеспечения модулей тематической обработки данных ДЗЗ.
2. Поместите все примеры в отдельно созданную директорию.
3. Предварительно прочитайте задание, затем выполните действия, следуя инструкциям из раздела 6 (Варианты).
4. Составить отчет о выполненной работе.
5. Сдать отчет преподавателю, получить зачет по работе.

## 4.3. Содержание отчета

Отчет по работе должен содержать:

- Описание реализованного метода тематической обработки.
- Результаты применения метода.
- Результаты сравнения работы метода с аналогичными.
- Листинг программы.

## 5. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Описание алгоритма сегментации, основанной на контурах
2. Описание алгоритма сегментации, основанной на областях.
3. Описание алгоритма сегментации с помощью метода водораздела.
4. Описание алгоритма сегментации с помощью иерархического метода водораздела.
5. Описание алгоритма сегментации с помощью алгоритма сдвига среднего.
6. Классификатор, основанный на расчете расстояний.
7. Классификатор по расстоянию Махаланобиса.
8. Древоподобный классификатор с кусочно-постоянной аппроксимацией.
9. Метод иерархической классификации.
10. Классификация с использованием однослойной нейронной сети.
11. Метод обратного распространения ошибки.
12. Нейронная сеть Кохонена.
13. Алгоритм ISODATA.
14. Разностный метод выделения изменений.
15. Метод выявления изменений по гистограмме.
16. Метод выявления изменений на основе пространственных моделей.

## 6. ВАРИАНТЫ ЗАДАНИЙ

### 6.1. Варианты заданий на лабораторную работу.

Вар-т	Задание на выполнение.	Задание на сравнение.
1.	<ul style="list-style-type: none"> <li>➤ Реализовать алгоритм сегментации по контурам.</li> <li>➤ Выполнить сегментацию реализованным алгоритмом и алгоритмом ИСОДАТА</li> <li>➤ Выполнить классификацию несегментированного и сегментированного изображений.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Сегментация разными алгоритмами.</li> <li>➤ Классификация с сегментацией и без сегментации</li> </ul>
2.	<ul style="list-style-type: none"> <li>➤ Реализовать алгоритм сегментации по областям.</li> <li>➤ Выполнить сегментацию реализованным алгоритмом и алгоритмом ИСОДАТА</li> <li>➤ Выполнить классификацию несегментированного и сегментированного изображений.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Сегментация разными алгоритмами.</li> <li>➤ Классификация с сегментацией и без сегментации</li> </ul>
3.	<ul style="list-style-type: none"> <li>➤ Реализовать алгоритм сегментации методом водораздела.</li> <li>➤ Выполнить сегментацию реализованным алгоритмом и алгоритмом ИСОДАТА</li> <li>➤ Выполнить классификацию несегментированного и сегментированного изображений.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Сегментация разными алгоритмами.</li> <li>➤ Классификация с сегментацией и без сегментации</li> </ul>
4.	<ul style="list-style-type: none"> <li>➤ Реализовать алгоритм сегментации методом сдвига среднего.</li> <li>➤ Выполнить сегментацию реализованным алгоритмом и алгоритмом ИСОДАТА</li> <li>➤ Выполнить классификацию несегментированного и сегментированного изображений.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Сегментация разными алгоритмами.</li> <li>➤ Классификация с сегментацией и без сегментации</li> </ul>
5.	<ul style="list-style-type: none"> <li>➤ Реализовать алгоритм классификации по минимуму расстояний.</li> <li>➤ Выполнить сегментацию алгоритмом ИСОДАТА</li> <li>➤ Выполнить классификацию несегментированного и сегментированного изображений различными алгоритмами</li> </ul>	<ul style="list-style-type: none"> <li>➤ Классификация разными алгоритмами.</li> <li>➤ Классификация с сегментацией и без сегментации</li> </ul>

	классификации.	
6.	<ul style="list-style-type: none"> <li>➤ Реализовать алгоритм классификации нейросетевым методом.</li> <li>➤ Выполнить сегментацию алгоритмом ИСОДАТА</li> <li>➤ Выполнить классификацию несегментированного и сегментированного изображений различными алгоритмами классификации.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Классификация разными алгоритмами.</li> <li>➤ Классификация с сегментацией и без сегментации</li> </ul>
7.	<ul style="list-style-type: none"> <li>➤ Реализовать алгоритм выявления изменений методом разности.</li> <li>➤ Выполнить выявление изменений на изображениях среднего и низкого разрешения различными алгоритмами (реализованным, главных компонент, с использованием пространственной модели).</li> </ul>	<ul style="list-style-type: none"> <li>➤ Выявление изменений на изображениях разного времени года (осень, весна, либо с разницей в 1-2 месяца), на одну и ту же дату, но за разные годы.</li> </ul>
8.	<ul style="list-style-type: none"> <li>➤ Реализовать алгоритм выявления изменений методом главных компонент.</li> <li>➤ Выполнить выявление изменений на изображениях среднего и низкого разрешения различными алгоритмами (реализованным, разности, с использованием пространственной модели).</li> </ul>	<ul style="list-style-type: none"> <li>➤ Выявление изменений на изображениях разного времени года (осень, весна, либо с разницей в 1-2 месяца), на одну и ту же дату, но за разные годы.</li> </ul>
9.	<ul style="list-style-type: none"> <li>➤ Реализовать алгоритм выявления изменений методом с использованием пространственной модели.</li> <li>➤ Выполнить выявление изменений на изображениях среднего и низкого разрешения различными алгоритмами (реализованным, разности, главных компонент).</li> </ul>	<ul style="list-style-type: none"> <li>➤ Выявление изменений на изображениях разного времени года (осень, весна, либо с разницей в 1-2 месяца), на одну и ту же дату, но за разные годы.</li> </ul>

## 6.2. Примеры автоматизированной сегментации КС.



*Исходное изображение*

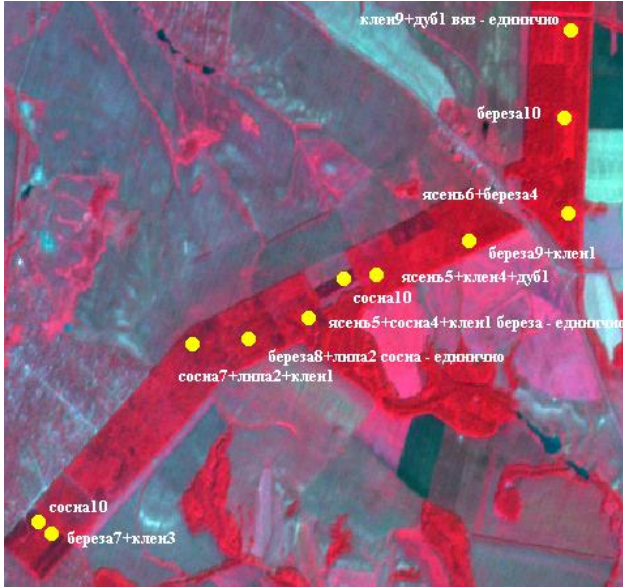


*Результаты применения алгоритма*

*Рисунок 19 - Использование алгоритма ISODATA для сегментации (при различных параметрах алгоритма)*



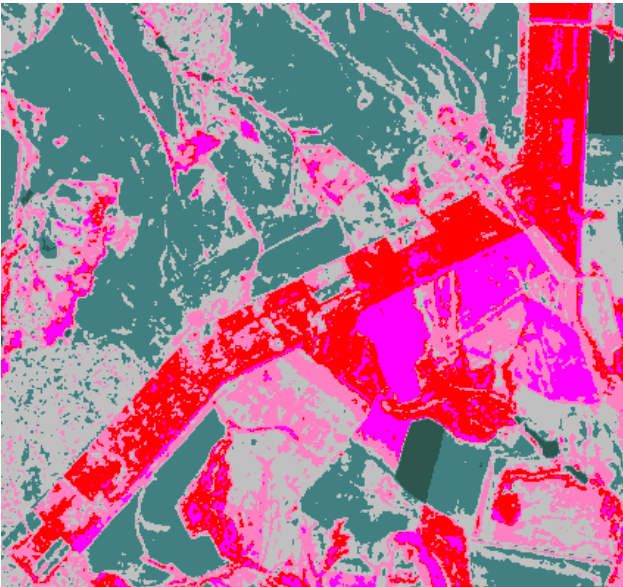
### 6.3. Примеры автоматизированной классификации объектов на КС



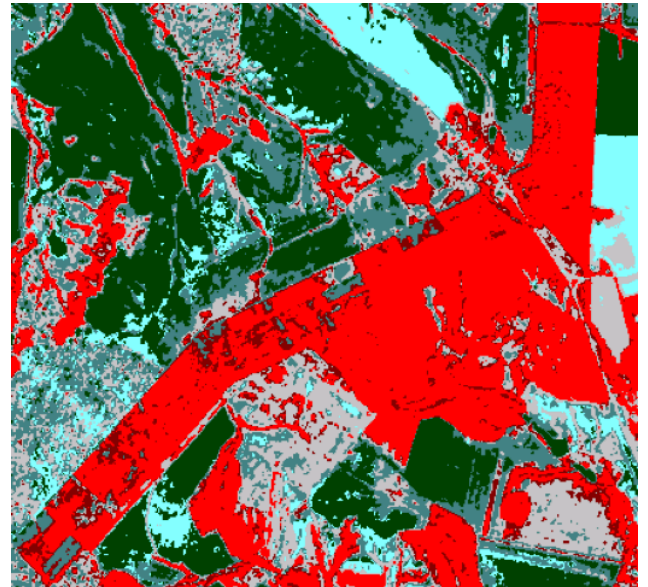
*Исходное изображение с нанесенными точками наземных наблюдений*



*Изображение в натуральных цветах*



*Результаты автоматической классификации с использованием алгоритма ИЗОДАТА*



*Результаты классификации с использованием нейронной сети Кохонена*

*Рисунок 20 - Классификация изображений*

## СОДЕРЖАНИЕ

1. Основные сведения о методах тематической обработки космических изображений земной поверхности .....	3
1.1. Выделение контуров на космических снимках, локализация и определение границ объектов. ....	4
1.2. Сегментация космических снимков. ....	5
1.2.1. Сегментация, основанная на контурах .....	5
1.2.2. Сегментация, основанная на областях .....	6
1.2.3. Сегментация с помощью метода водораздела .....	9
1.2.4. Алгоритм сдвига среднего .....	13
1.3. Классификация областей на изображениях. ....	16
1.3.1. Классификатор, основанный на расчете расстояний .....	17
1.3.2. Классификатор по расстоянию Махаланобиса .....	18
1.3.3. Метод иерархической классификации .....	19
1.3.4. Древоподобный классификатор с кусочно-постоянной аппроксимацией ..	20
1.3.5. Классификация с использованием однослойной нейронной сети с обратным распространением .....	21
1.3.6. Классификация с использованием нейронной сети Кохонена .....	23
1.3.7. Алгоритм ISODATA .....	26
1.4. Выявление изменений на изображениях. ....	28
1.4.1. Разностный метод выделения изменений .....	28
1.4.2. Метод главных компонент .....	30
1.4.3. Методы на основе пространственных моделей .....	31
2. Программные средства тематической обработки космических снимков .....	33
2.1. Программные средства сегментации .....	33
2.2. Программные средства классификации .....	34
2.3. Программные средства выявления изменений. ....	36
2.4. Стороннее программное обеспечение – ScanEx ImageProcessor .....	38
3. Литература .....	44
4. Порядок Выполнения лабораторной работы .....	44
4.1. Исходные данные .....	44
4.2. Общий план выполнения работы .....	45

	51
4.3. Содержание отчета.....	45
5. Контрольные вопросы.....	45
6. Варианты заданий.....	46
6.1. Варианты заданий на лабораторную работу.....	46
6.2. Примеры автоматизированной сегментации КС.....	48
6.3. Примеры автоматизированной классификации объектов на КС.....	49
Содержание.....	50