



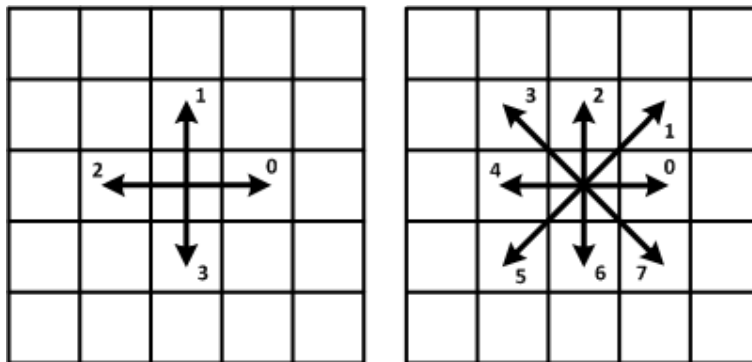
**ІТМО**

**Сегментация**  
**Техническое зрение**

# Сегментация

# Выделение связанных областей

- **Соседство:**
  - соседство «по кресту» и 4-связность;
  - соседство «по квадрату» и 8-связность.
- Соседи **по горизонтали и вертикали** находятся на расстоянии 1 от центрального пикселя окрестности.
- Соседи **по диагонали** находятся на расстоянии  $\sqrt{2}$  от центрального пикселя окрестности.



- **Связная область** изображения это область (множество точек):
  - все точки которой **имеют одинаковое значение**;
  - **между любыми** двумя точками из данной области существует **непрерывный путь**, состоящий из точек, также принадлежащих данной области и являющихся **при этом «соседями»**.
- **Алгоритмы** выделения связных областей бинарных изображений:
  1. Метод «лесного пожара».
  2. Двух проходной алгоритм.

- **Сегментация** – это разделение изображения на непересекающиеся области, каждая из которых представлена цветом или текстурой одного типа.
- **Цель сегментации** в «широком смысле»: разбиение изображения на семантические области, которые имеют строгую корреляцию с объектами или областями наблюдаемой трехмерной сцены.



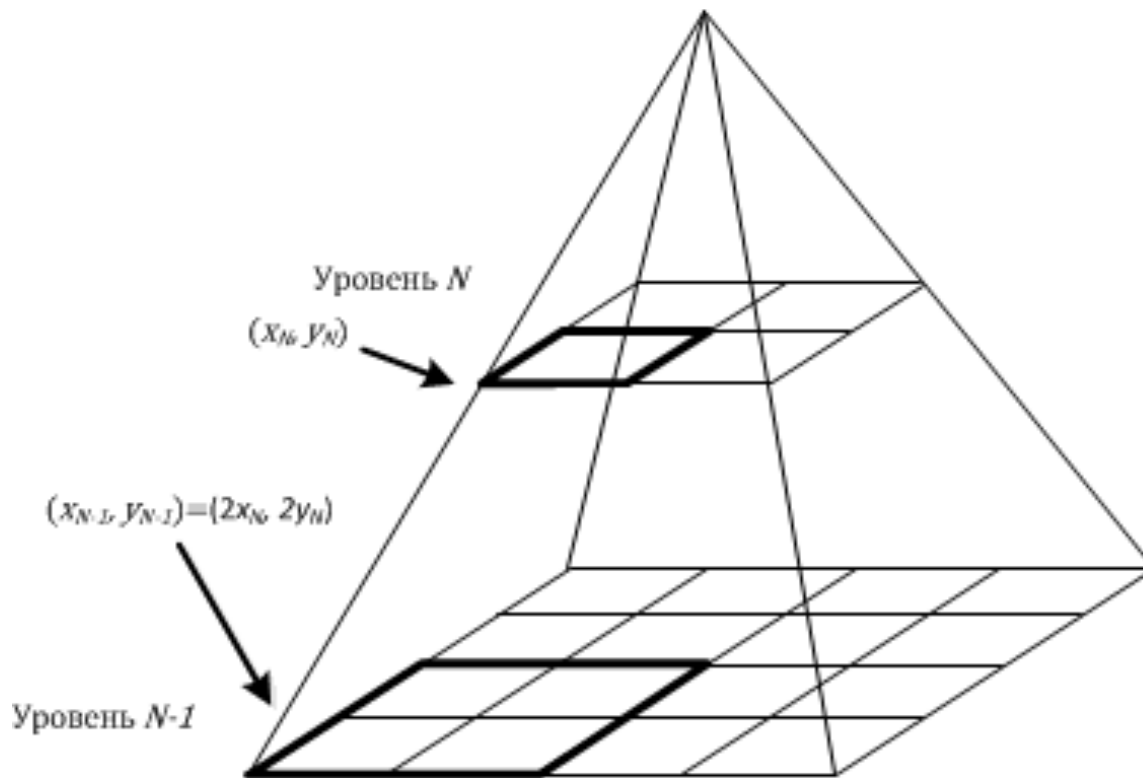
- $R$  – вся область кадра.
- Сегментация – это процесс разбиения  $R$  на такую совокупность связных областей  $\{R_i\}, i = 1, \dots, n$ , что для них выполняются следующие основные условия:
  - $R = \bigcup R_{i=1, \dots, n}$  – области разбиения **целиком покрывают кадр**;
  - $R_i \cap R_j = \emptyset, \forall i \neq j$ ; – области разбиения **попарно не пересекаются**;
  - $\text{Pred}(R_i) = \text{TRUE}, i = 1, \dots, n$ , где  $\text{Pred}(R)$  – **булевский предикат однородности** области;
  - $\text{Pred}(R_i \cup R_j) = \text{FALSE}, \forall i \neq j$  – **попарное объединение любых двух областей данного разбиения не удовлетворяет тому же условию однородности.**

- Осуществить пресегментацию изображения на «стартовые» области неитеративным (однократным) методом.
- Определить критерий слияния двух соседних областей.
- Итеративно находить и объединять все пары соседних областей, удовлетворяющие критерию слияния.
- Если ни одной пары кандидатов на объединение не найдено – остановиться и выйти из алгоритма.

- Разбиение начинают с представления всего изображения как простой области, которая не всегда соответствует условию однородности.
- В процессе сегментации текущие области изображения последовательно расщепляются в соответствии с заданными условиями однородности.
- Методы слияния и разбиения областей далеко не всегда приводят к одним и тем же результатам сегментации, даже если в них используется один и тот же критерий однородности.



# Слияние / разделение областей



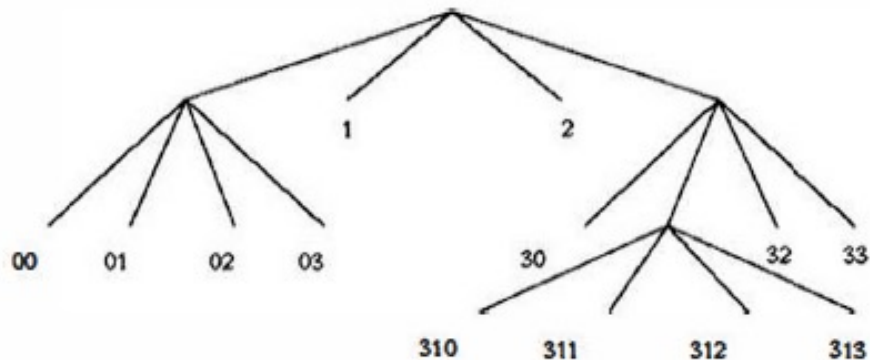
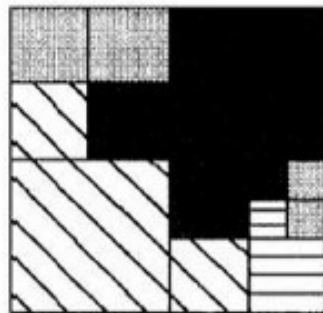
# Слияние / разделение областей

- Процессы слияния и разбиения областей идут поочередно на каждой итерации.
- Если какая-либо область на каком-либо пирамидальном уровне неоднородна, она разделяется на четыре подобласти.
- Напротив, если на каком-либо уровне пирамиды наблюдаются четыре соседние области с приблизительно одинаковой величиной однородности, они сливаются в простую область на более высоком уровне пирамиды.



1. Провести начальную сегментацию областей, определить критерий однородности и пирамиду структуры данных.
2.
  - Если какая-либо область  $R$  в пирамиде структуры данных неоднородна ( $Pred(R) = FALSE$ ), разделяем ее на четыре дочерние области.
  - Если любые четыре области, имеющие одинаковых родителей, могут быть слиты в простую однородную область, то осуществляется слияние областей.
  - Если нет больше областей, которые могли бы быть разделены или слиты на данном шаге, переходим к шагу 3.
3. Если имеются какие-либо две соседние области  $R_i, R_j$ , которые могут быть слиты в однородную область, объединяем их.
4. Производим слияние малых областей с самой большой подобной соседней областью.

# Слияние / разделение областей



# Пороговая сегментация

- По уровням яркости

$$\begin{cases} g(i, j) = 1, \text{ для } f(i, j) \geq T, \\ g(i, j) = 0, \text{ для } f(i, j) < T, \end{cases}$$

где  $g(i, j)$  – элемент результирующего бинарного изображения,

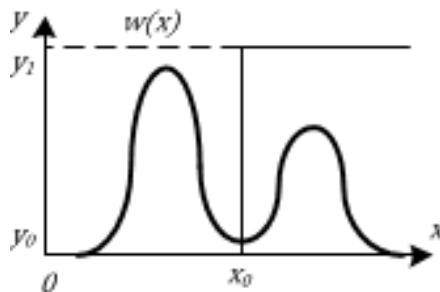
$f(i, j)$  – элемент исходного изображения,

$T$  – порог яркости.

- **Основной вопрос** – определение порога сегментации.

# Определение порога сегментации

- $w(x)$  – гистограмма изображения, где  $0 \leq x \leq 255$ .



Определение порога  
по гистограмме

# Диапазонная пороговая сегментация

- Модель сегментации:

$$\begin{cases} g(i, j) = 1, \text{ для } f(i, j) \in D, \\ g(i, j) = 0, \text{ в противном случае,} \end{cases}$$

где  $D$  – диапазон значений.

# Мультипороговая сегментация

- Модель сегментации:

$$\left\{ \begin{array}{l} g(i, j) = 1, \text{ для } f(i, j) \in D_1, \\ g(i, j) = 2, \text{ для } f(i, j) \in D_2, \\ \dots \\ g(i, j) = 0, \text{ в противном случае,} \end{array} \right.$$

где  $D_i$  – диапазоны значений.



1. **Задать число классов  $k$** , на которое нужно разбить изображение. Все пиксели рассматриваются как набор векторов  $x_i, i = 1, \dots, p$ .
2. **Определить  $k$ -векторов  $m_j, j = 1, \dots, k$** , которые объявляются **начальными центрами кластеров**. Выбрать значения  $m_j, j = 1, \dots, k$  (например, случайным образом).
3. **Обновить значения средних векторов  $m_j, j = 1, \dots, k$** , (центров кластеров). Для этого:
  - **вычислить расстояние** от каждого  $x_i, i = 1, \dots, p$  до каждого  $m_j, j = 1, \dots, k$ ;
  - **отнести** каждый  $x_i$  к кластеру  $j^*$ , расстояние до центра которого  $m_{j^*}$  минимально;
  - **пересчитать средние значения  $m_j$**  по всем кластерам.
4. **Повторять шаги 2, 3** пока центры кластеров не перестанут изменяться.

# Сегментация по принципу Вебера

- Функция Вебера  $W(I)$ :

$$W(I) = \begin{cases} 20 - \frac{12I}{88}, & 0 \leq I \leq 88, \\ 0,002(I - 88)^2, & 88 < I \leq 138, \\ \frac{7(I - 138)}{255 - 138} + 13, & 138 < I \leq 255, \end{cases}$$

где  $I$  – значение яркости.

- **Принцип Вебера:** человек не различает уровней серого между  $[I(n), I(n) + W(I(n))]$ .

1. Установить номер первого класса  $n = 1$  и начальный уровень серого  $I(n) = 0$ .
  2. Вычислить значение  $W(I(n))$ , соответствующее яркости  $I(n)$  по формуле Вебера.
  3. В исходном изображении  $I$  установить значения яркости  $I(n)$  для всех пикселей, яркость которых находится в диапазоне  $[I(n), I(n) + W(I(n))]$ .
  4. Найти пиксели, значение яркости которых выше  $G = I(n) + W(I(n)) + 1$ . Если такие пиксели есть, увеличить номер класса  $n = n + 1$ ,  $I(n) = G$ , перейти к шагу 2. Если таких нет, закончить работу.
- Изображение сегментировано на  $n$  классов, каждый класс показан яркостью  $W(I(n))$ . Данный способ сегментации удобно реализовать, построив таблицу LUT.

1. Обход изображения от **левого верхнего** пикселя, являющегося классом  $C_1$ .
  - Для пикселей **первой строки** вычислить отклонение от класса **левого пикселя** и **сравнить** с заданным порогом. Если меньше порога, добавить **пиксель к классу соседа**, иначе – завести новый класс  $C_{1+i}$ .

2. Первый пиксель **каждой следующей строки** сравнить с классами **двух соседей: левого и верхнего**.
- Если **отклонение** от **обоих** сравниваемых классов **больше порога**, то **завести новый класс**.
  - Если **отклонение** больше только для **одного класса**, то добавить пиксель к тому классу, отклонение от **которого меньше порога**.
  - В случае если **отклонение допустимо** для обоих классов, возможны два варианта:
    1.  $L(g(C_i) - g(C_j)) < \delta$  – объединить эти два класса (если это не один и тот же класс) и добавить к объединенному классу текущий пиксель;
    2.  $L(g(C_i) - g(C_j)) > \delta$  – добавить пиксель к тому из двух классов, от которого отклонение минимально.
  - В качестве **меры  $L$**  можно использовать любую функцию расстояния, например, разность в RGB-пространстве.

- **Преимущества:** цвет кожи не зависит от ориентации лица, анализ цвета пикселей вычислительно эффективен.
- **Задача:** нахождение критерия оценки близости цвета каждого пикселя к оттенку кожи.
- **Разработка модели цвета кожи:**
  1. **Накопление тренировочных данных,** используя изображения, на которых указывают области «кожа» и «не кожа». По этим данным накапливается статистика оттенков кожи.
  2. **Обработка полученной статистики** и выбор параметров модели цвета кожи для последующего использования; выбор критериев оценки принадлежности пикселей к области «кожа»;
  3. **Обработка изображений** с использованием полученных критериев.

- Пороговые критерии, т.е. контекстно-независимая сегментация: цвет пикселя (R,G,B) причисляется к области «кожа», если выполнены следующие условия:

либо

$$\begin{aligned} R > 95, G > 40, B < 20, R > G, R > B, \\ \max\{R, G, B\} - \min\{R, G, B\} > 15, \\ |R - G| > 15, \end{aligned}$$

либо

$$\begin{aligned} R > 220, G > 210, B > 170 \\ |R - G| \leq 15, G > B, R > B \end{aligned}$$

либо

$$\begin{aligned} r &= \frac{R}{R + G + B}, g = \frac{G}{R + G + B}, b = \frac{B}{R + G + B}, \\ \frac{r}{g} &> 1.185, \quad \frac{rb}{(r + g + b)^2} > 0.107, \quad \frac{rg}{(r + g + b)^2} > 0.112 \end{aligned}$$

# Сегментация по цвету кожи

- После сегментации по цвету кожи необходимо выполнить медианную фильтрацию.





- **Подходы к текстурной сегментации:**

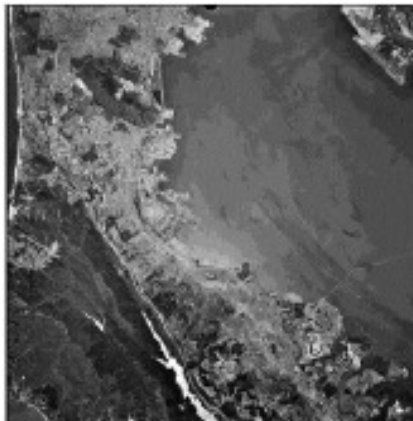
1. **Статистический** – позволяют охарактеризовать текстуру области как гладкую, грубую и зернистую.
2. **Структурный** – определяют и описывают взаимное расположение простейших повторяющихся элементов изображения, например, отрезков параллельных линий, проходящих с постоянным шагом, клеток на шахматной доске.
3. **Спектральный.**



Текстуры

# Текстурная сегментация

- **Статистический подход** на примере изображения, содержащего **два типа областей**, представленных разными текстурами.
- Результат: **разбиение** изображения на **водную поверхность и сушу**.
- Это невозможно сделать методами бинаризации, только **путем анализа параметров текстуры** в окрестности каждого пикселя.



Изображение с областями текстуры разных типов,  
соответствующих суше и воде

# Алгоритм текстурной сегментации

1. Считать изображение.
2. Определить параметры текстуры.
  - Предположим, что **яркость** в пикселях изображения – это **случайная величина**  $z$ ,
  - Ей соответствует **вероятность распределения**  $p(z_i)$ , взятая из гистограммы ( $L$  – число уровней яркости).
3. **Создать маску** для выделения более **крупной текстуры**.
4. Выполнить сегментацию.

# Определение параметров текстуры

- **Центральный момент** порядка  $n$  случайной величины  $z_i$  равен:

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i),$$

где  $m$  – среднее значение  $z$  (средняя яркость изображения),

$$m = \sum_{i=0}^{L-1} z_i p(z_i),$$

$\mu_0 = 1$  и  $\mu_1 = 0$ .

- Для описания текстуры важен **второй момент**, т.е. дисперсия:  $\sigma^2(z) = \mu_2(z)$ . Она является **мерой яркостного контраста**, что можно использовать для вычисления признаков **относительной гладкости**:

$$R = 1 - \frac{1}{1 + \sigma^2(z)}.$$

- $R$  **равно нулю** для областей **постоянной яркости** (где дисперсия нулевая),
- $R$  **приближается к единице** для больших значений  $\sigma^2(z)$ .

# Определение параметров текстуры

- В **полутоновых** изображениях целесообразно **нормировать** дисперсию до интервала  $[0,1]$ . Для этого необходимо разделить  $\sigma^2(z)$  на  $(L - 1)^2$ .
- Стандартное **отклонение** используется в качестве характеристики текстуры:

$$s = \sigma(z).$$

- **Третий момент** является характеристикой **симметрии** гистограммы:

$$\mu_3(z) = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i).$$

- Для оценки **разброса по яркости** соседних пикселей используется **функция энтропии**:

$$e = - \sum_{i=0}^{L-1} p(z) \log_2 p(z_i),$$

где  $p(z_i)$  – вероятность текущей яркости в окрестности точки;

$L$  – количество уровней яркости;

$e$  – значение энтропии в текущей точке.

# Определение параметров текстуры

- Для описания текстуры также используется **мера однородности**, оценивающая **равномерность гистограммы**:

$$U = \sum_{i=0}^{L-1} p^2(z_i).$$

- В таблице приведены значения описанных характеристик, подобранные для гладкой, грубой и периодической текстур.

Текстура	Среднее	Стандартное отклонение	R (нормированное)	Третий момент	Однородность	Энтропия
Гладкая	82.64	11.79	0.002	-0.105	0.026	5.434
Грубая	143.56	74.63	0.0079	-0.151	0.005	7.783
Периодическая	99.72	33.73	0.017	0.750	0.013	6.674

# Пример текстурной сегментации

- Пусть на изображении имеются текстуры двух типов: **крупная и мелкая** (зернистая). Мелкая соответствует **водной зоне**.
- Для отделения одной области от другой создать маску, которая удаляет маленькие объекты. Для этого использовать функцию **определения связного множества** пикселей на бинарном изображении и **вычислить площади полученных объектов**. Использовать тип связности восемь.
- Если цвет соседей совпадает, то они принадлежат одному объекту, в противном случае – к различным.
- Все объекты с **площадью меньше заданной** величины  $S$  удалить.

# Пример текстурной сегментации



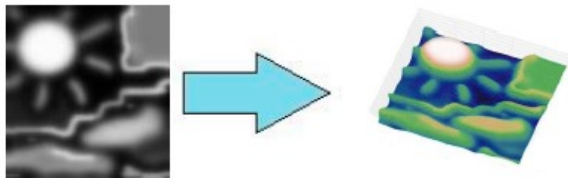
Слева результат текстурной фильтрации на базе вычисления энтропии в окне 9x9.

Области двух текстур показаны темными (вода) и светлыми (суша) оттенками;  
по центру – маска водной поверхности после удаления объектов малой площади;  
справа – результат сегментации суши.



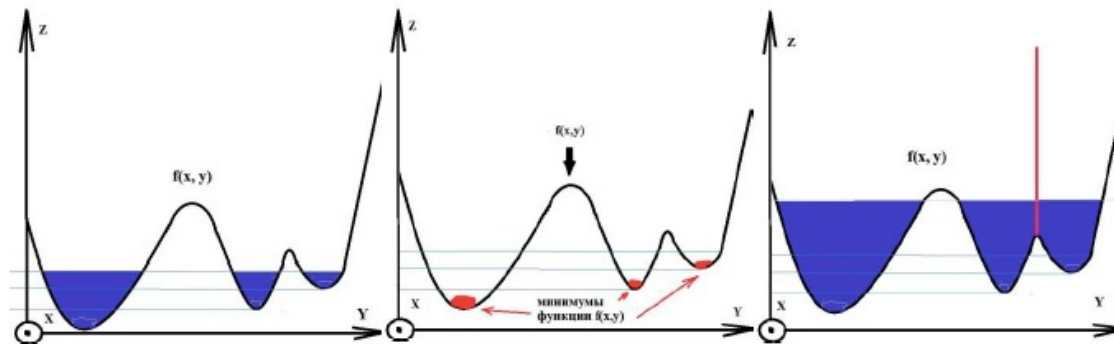
# Метод морфологического водораздела

- **Полутоновое изображение** – цифровая модель местности, где значения яркости – это **высоты относительно некоторого уровня**, т.е. изображение – это матрица высот.



# Метод морфологического водораздела

- Если на такую местность **льет дождь**, образуется **множество бассейнов**. Вода **заполняет маленькие бассейны**, затем из переполненных бассейнов **выливается** и **бассейны объединяются** в более крупные **согласно высотам** уровня воды.
- Места объединения бассейнов отмечаются как **линии водораздела**. В итоге вся местность может быть затоплена.
- Результат сегментации зависит **от момента прекращения поступления воды**. Если процесс остановить рано, изображение будет сегментировано на мелкие области, если поздно – на очень крупные.



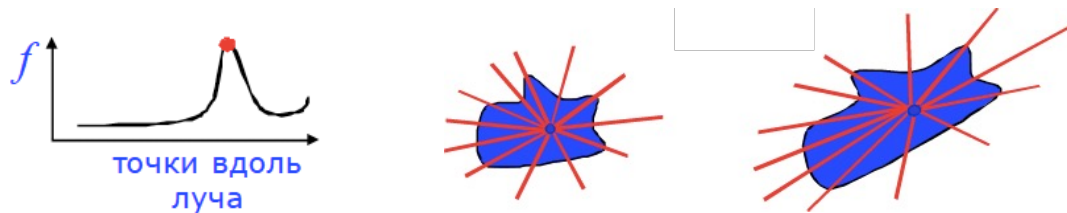
# Метод морфологического водораздела

- Все пиксели делятся на три типа:
  1. **локальные минимумы;**
  2. **находящиеся на склоне, т.е. те, с которых вода скатывается в один и тот же локальный минимум;**
  3. **локальные максимумы, т.е. те, с которых вода скатывается более чем в один минимум.**
- При сегментации с помощью данного метода нужно **определить водосборные бассейны и линии водораздела** на изображении путем обработки **локальных областей** в зависимости от их **яркостных характеристик**.

- Алгоритм реализации метода:
  1. **вычисляется функция сегментации** (это касается изображений, где объекты размещены в темных областях и являются трудно различимыми);
  2. **вычисляются маркеры переднего плана** изображения на основании анализа связности пикселей каждого объекта;
  3. **вычисляются фоновые маркеры**, представляющие собой пиксели, которые не являются частями объектов;
  4. **модифицируется функция сегментации** на основании значений расположения маркеров фона и маркеров переднего плана.
  5. **выделение на фоне** изображения **однородных по яркости объектов** (в виде пятен).
- Области, характеризующиеся **малыми вариациями яркости**, имеют **малые значения градиента**. Поэтому на практике метод сегментации по водоразделу обычно применяется **не к самому изображению, а к его градиентному представлению**.

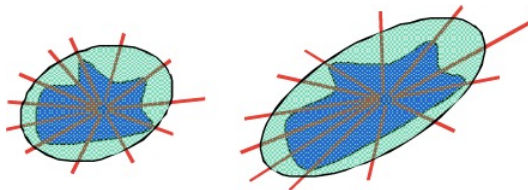
- IBR – Метод сегментации областей на основе экстремумов интенсивности (Intensity-Extrema Based Regions).
- Необходимо идти от точек локального экстремума яркости  $I_0$  по лучам, вычисляя некоторую величину  $f$ .
- Как только будет найден пик величины  $f$ , необходимо остановиться. Эта точка и будет являться границей области.

$$f(t) = \frac{|I(t) - I_0|}{\frac{1}{t} \int_0^t |I(t) - I_0| dt}.$$

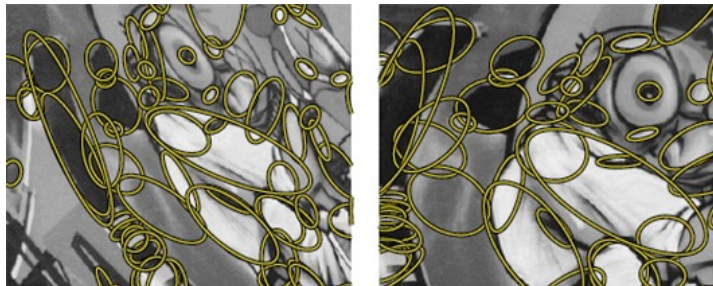


Пример работы детектора IBR

- Области на паре подобных изображений могут различаться, поэтому опишем вокруг них эллипсы.
- Если эллипсы превратить в окружности, то получим полное сходство с точностью до поворота.



Описанные эллипсы вокруг объектов



Пример работы детектора IBR

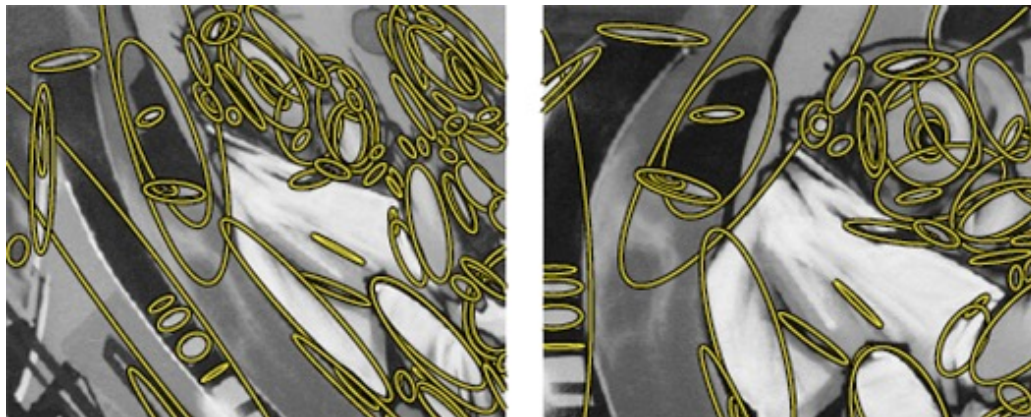
# Детектор MSER

- MSER – Maximally Stable Extremal Regions (максимально устойчивые экстремальные области).
- Решает проблему инвариантности особых точек при масштабировании.

1. **Сортировка множества всех пикселей** изображения в порядке возрастания/убывания интенсивности.
2. **Построение пирамиды связных компонент.** Для каждого пикселя отсортированного множества выполним последовательность действий:
  - обновление списка точек, входящих в состав компоненты;
  - обновление областей следующих компонент, в результате чего пиксели предыдущего уровня будут подмножеством пикселей следующего уровня.
3. Для всех компонент **поиск локальных минимумов** (находим пиксели, которые присутствуют в данной компоненте, но не входят в состав предыдущих). Набор локальных минимумов уровня соответствует экстремальному региону на изображении.



# Детектор MSER

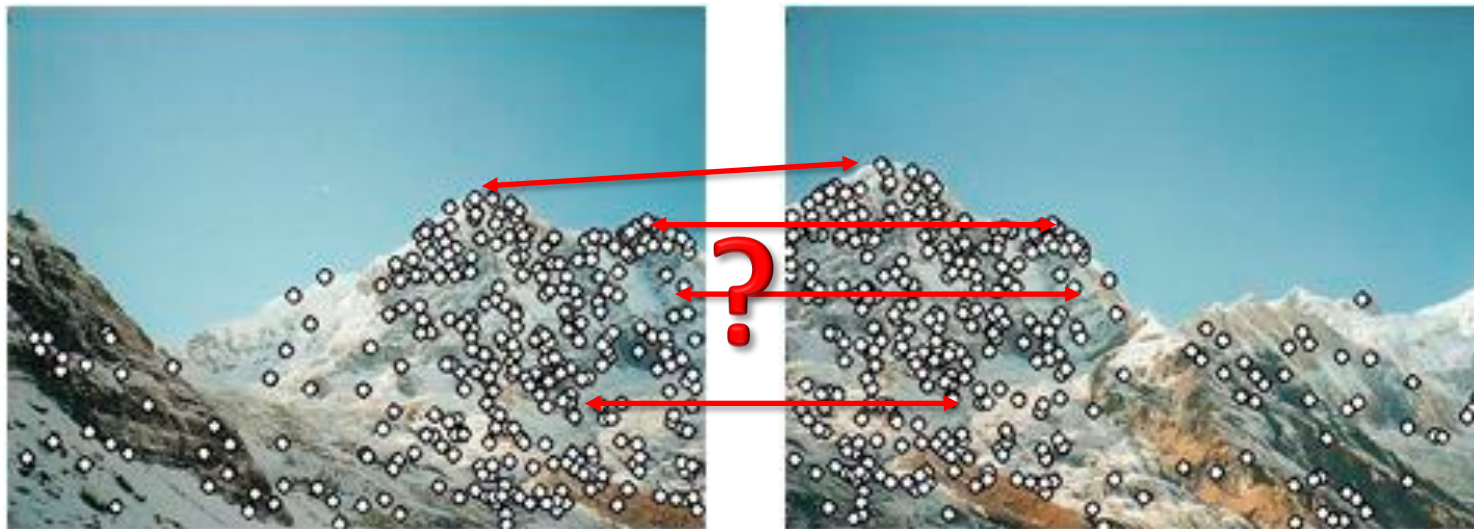


Пример работы детектора MSER

- Помимо рассмотренных, существуют многие другие детекторы: Shi-Tomasi, SUSAN, Trajkovic, CSS, CPDA, Harris-Affine, EBR, Hessian, Hessian-Laplace/Affine, Salient Regions, Superpixels, SURF, FAST-9, FAST-12, FAST-ER, ORB и др.

# Проблема сопоставления

- Как сопоставить характеристические точки на разных изображениях?
- Мы должны описать признаки, чтобы иметь возможность сравнивать их.



# Вопросы?

**ITMO** *re than a*  
**UNIVERSITY**

[s.shavetov@itmo.ru](mailto:s.shavetov@itmo.ru)