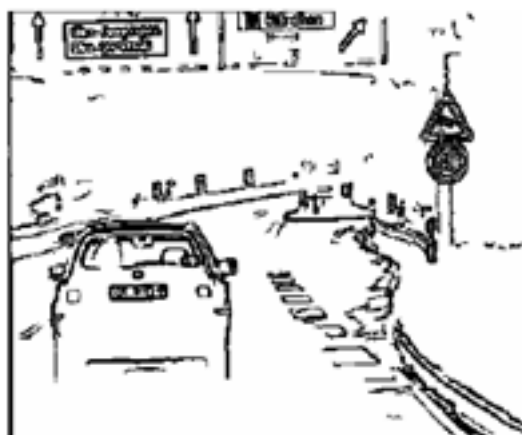
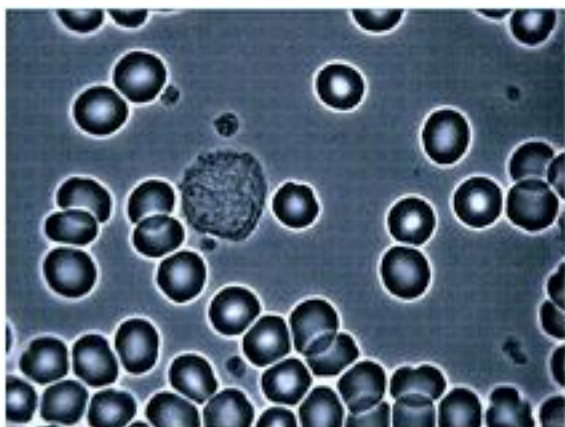




Введение в распознавание изображений, ч. 1



АНТОН КОНУШИН

Many slides adapted from Fei-Fei Li, Rob Fergus, Antonio Torralba, Jean Ponce and Svetlana Lazebnik



Задача распознавания изображений

- Понять, что запечатлено на изображении
- «To see means to know what is where by looking»
 - David Marr, ***Vision***, 1982
- «Тест Тьюринга» - компьютер должен ответить на любой вопрос об изображении, на который может ответить человек



Мы видим

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

Компьютер видит



Выделение объектов



Необходимо определить, есть ли на изображении объекты заданного типа и если да, то определить их положение



Изменчивость изображений



Внешние факторы:

- Положение камеры
- Внутренние свойства камеры
- Освещение

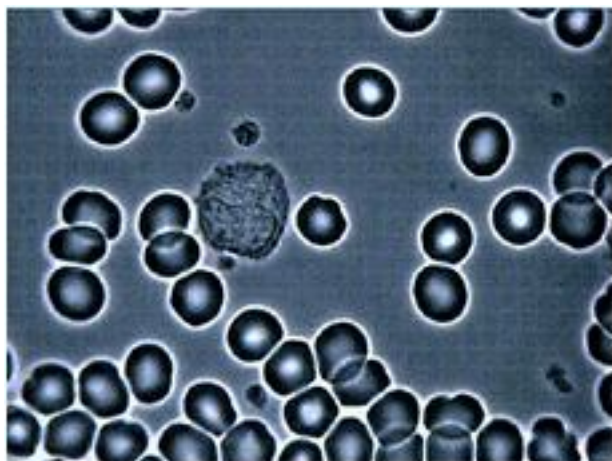
Внутренние факторы:

- Деформации объектов
- Внутриклассовая изменчивость

- Пока приходится задачу упрощать, вводя ограничения на ракурс съёмки, условия освещения, типы объектов
- Мы будем рассматривать простые случаи, когда все факторы варьируются незначительно



Примеры простых задач



Клетки крови



Ложки и сахар



Монеты и купюры



Номера



План

Лекция 3

- Сопоставление шаблонов
- Основы сегментации изображений
- Анализ сегментов

Лекция 4

- Введение в машинное обучение на примере метода опорных векторов
- Алгоритм поиска пешеходов [Dalal & Triggs 2005]



Сопоставление шаблонов

- Фиксируем изображение объекта (шаблон – pattern)
- Будем искать объект в кадре, «прикладывая» шаблон к изображению во всех возможных точках
- Попиксельно будем сравнивать шаблон и фрагмент нового кадра с помощью какой-нибудь метрики
- Например:

$$\frac{1}{n-1} \sum_{x,y} \frac{(f(x,y) - \bar{f})(t(x,y) - \bar{t})}{\sigma_f \sigma_t}$$

(NCC) Normalized cross correlation





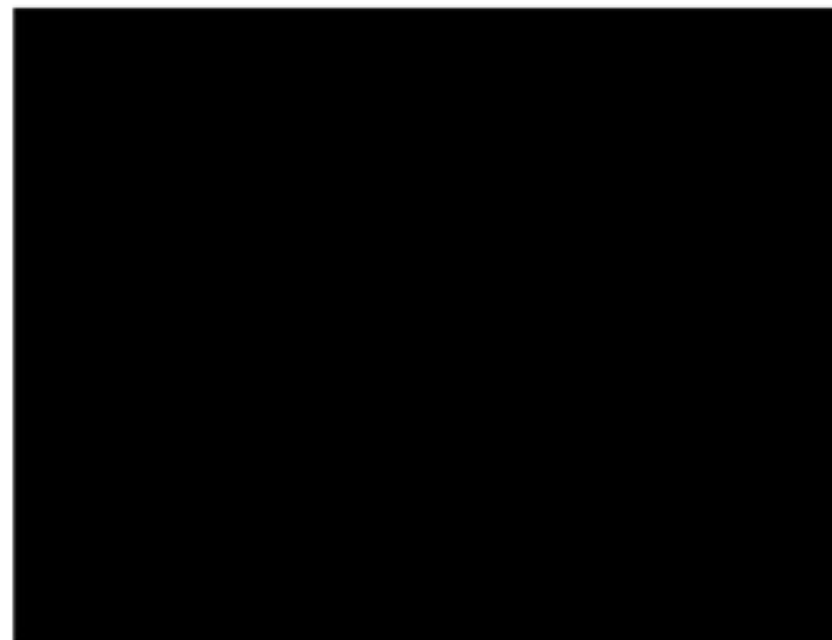
Пример: пульт ТВ



- Шаблон (слева), изображение (в центре), карта нормализованной корреляции (справа)
- Пик яркости (максимум корреляции) соответствует положению руки (искомого шаблона)



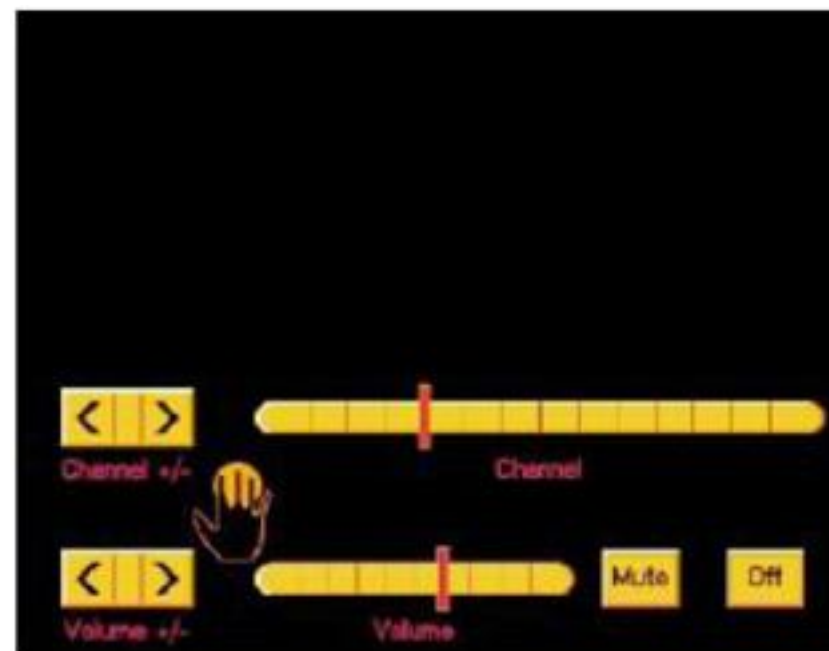
Пример: пульт ТВ



Credit: W. Freeman *et al*, "Computer Vision for Interactive Computer Graphics," *IEEE Computer Graphics and Applications*, 1998

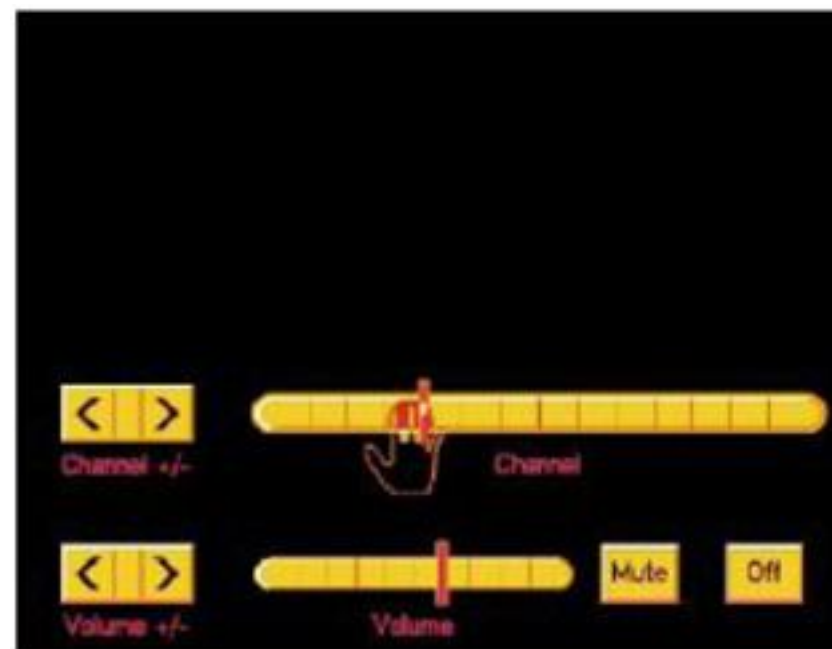


Пример: пульт ТВ



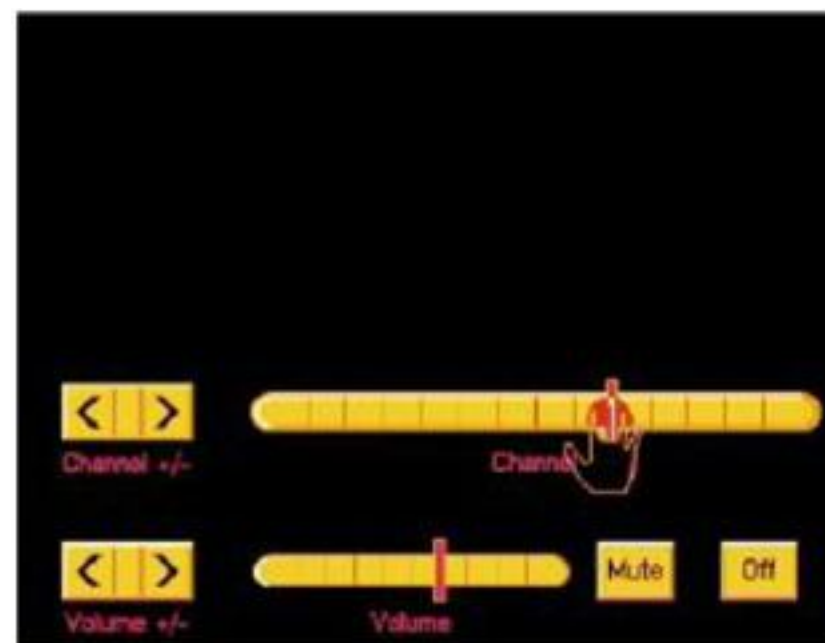


Пример: пульт ТВ





Пример: пульт ТВ

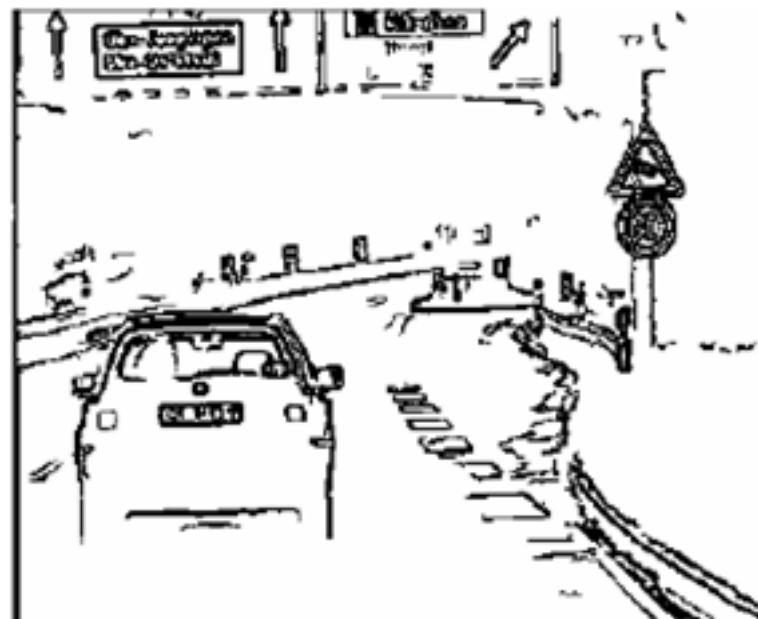
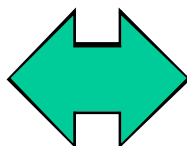




Как улучшить подход?



Края для сопоставления шаблонов



- Мы знаем, что в края – очень информативный признак, и они устойчивы к изменению освещения
- Попробуем использовать только края для поиска / отслеживания объекта
- Как эффективно сопоставлять карты краев?
 - Попиксельно явно нельзя!



Метрики

- Chamfer Distance

- Для каждого пикселя a края шаблона A вычисляем расстояние до ближайшего пикселя b края изображения B

$$r(a, B) = \min_{b \in B} \|a - b\|$$

- Суммируем все найденные расстояния

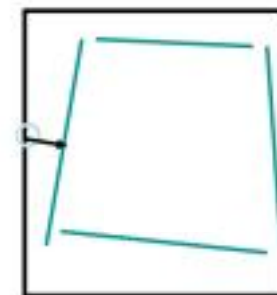
$$ChDist(A, B) = \sum_{a \in A} \min_{b \in B} \|a - b\|$$

- Hausdorff Distance

- Почти то же самое, но берём не сумму, а максимальное расстояния

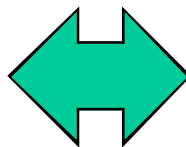
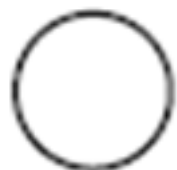
$$HausDist(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$

Какую метрику использовать заранее сказать нельзя, нужна экспериментальная проверка





Поиск ближайших пикселей края

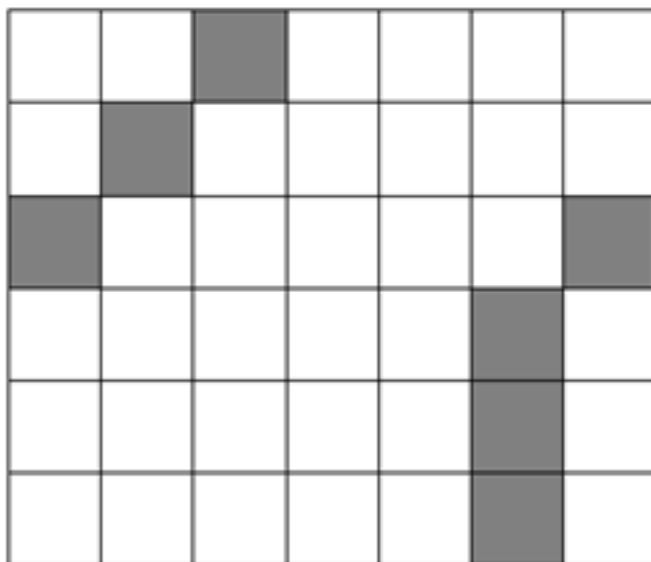


Вопрос: как найти ближайший пиксель края на изображении?



Distance Transform

«Дистантное преобразование»



2	1	0	1	2	3	2
1	0	1	2	3	2	1
0	1	2	3	2	1	0
1	2	3	2	1	0	1
2	3	3	2	1	0	1
3	4	3	2	1	0	1

Для каждого пикселя вычисляется расстояние до ближайшего пикселя края

- Многопроходный алгоритм (позметить соседей, потом их соседей и т.д.)
- Двухпроходный алгоритм



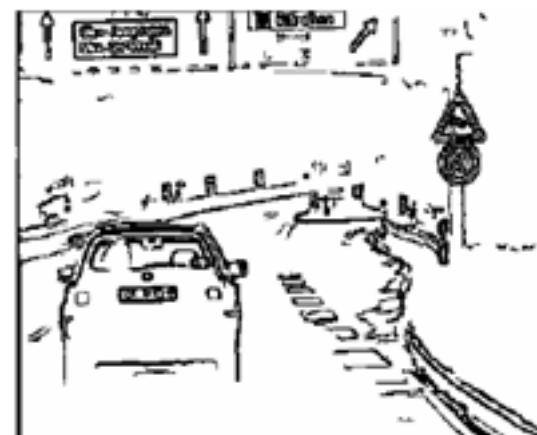
Применение DT

2	1	0	1	2	3	2
1	0	1	2	3	2	1
0	1	2	3	2	1	0
1	2	3	2	1	0	1
2	3	3	2	1	0	1
3	4	3	2	1	0	1

- Совмещаем шаблон и карту DT
- Вычисляем ошибку
 - Суммируя все значения в пикселях краев (Chamfer distance)
 - Находя максимум (Hausdorf distance)



Пример поиска с помощью DT



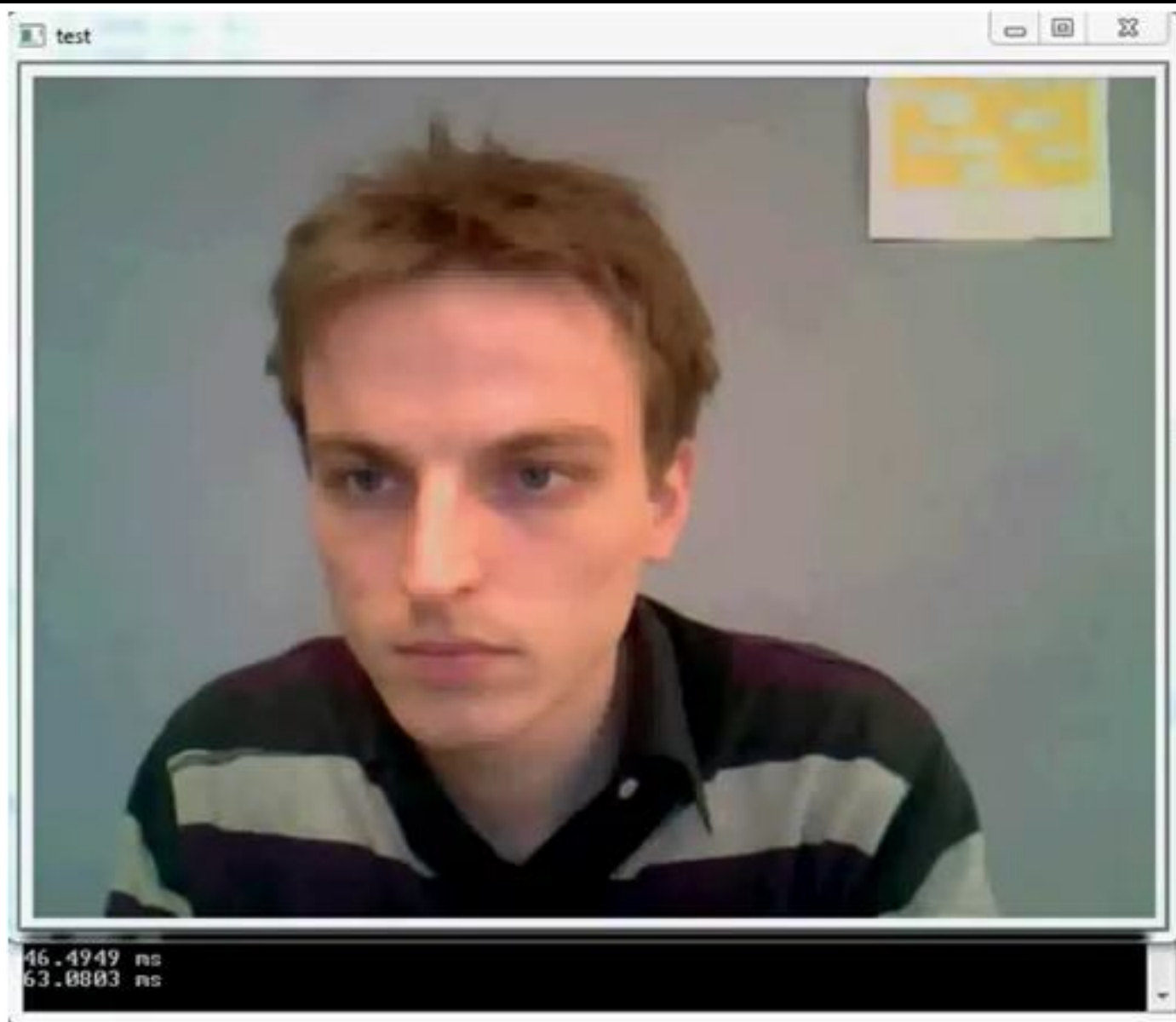


Пример задачи





Пример





Резюме сопоставления шаблонов

- Подходит в тех случаях, когда объекты фиксированы и модель преобразования не очень сложная
 - Цифры на знаках
 - Цифры на конвертах
 - Аэрофотосъёмка / Космическая съёмка
- Не очень быстрые методы
 - Требуются специальные процедуры для ускорения, пр. отбраковка ложных фрагментов по упрощённым критериям и т.д.



Номера



План

Лекция 3

- Сопоставление шаблонов
- Основы сегментации изображений
- Анализ сегментов

Лекция 4

- Введение в машинное обучение на примере метода опорных векторов
- Алгоритм поиска пешеходов [Dalal & Triggs 2005]



Из чего состоит изображение?



Slide by Alexey Efros



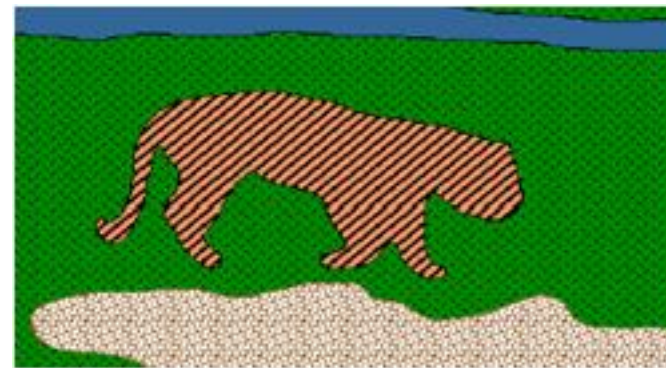
Из «кусков» - отдельных объектов





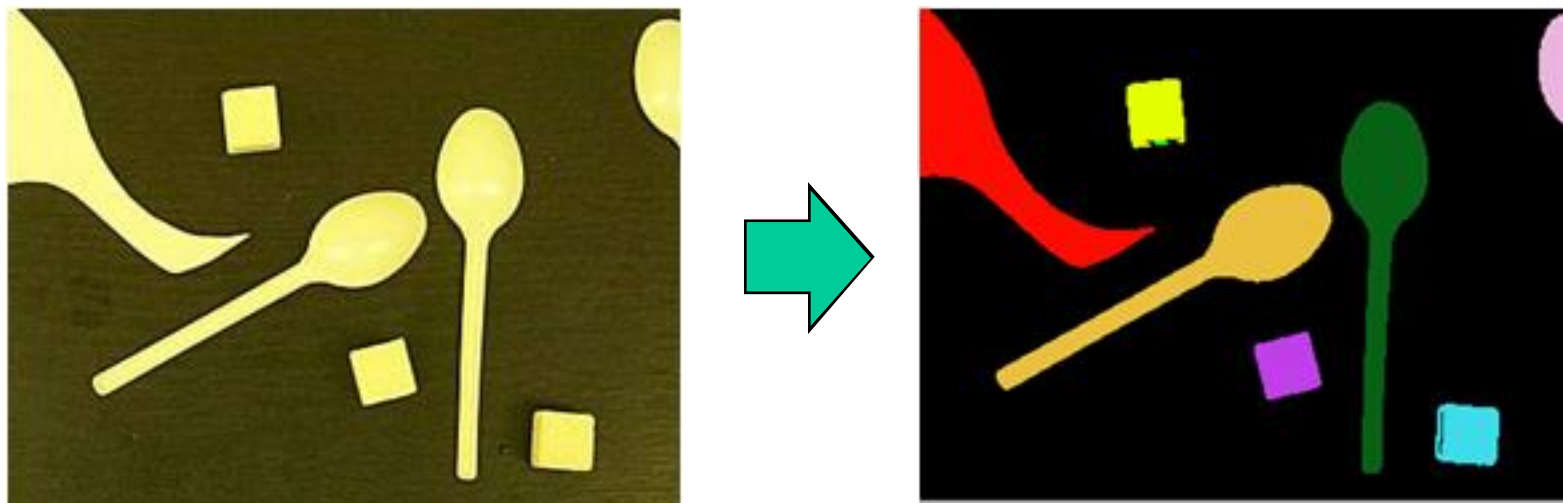
Сегментация

- Сегментация - это способ разделения сцены на «куски», с которыми проще работать
- Тесселяция - разбиение изображения на неперекрывающиеся области, покрывающие все изображение и однородные по некоторым признакам
- Можно и по другому сегментировать изображение
 - Пересекающиеся области
 - Иерархическое представление
- Есть ли сегментация в сопоставлении шаблонов?





Результат сегментации

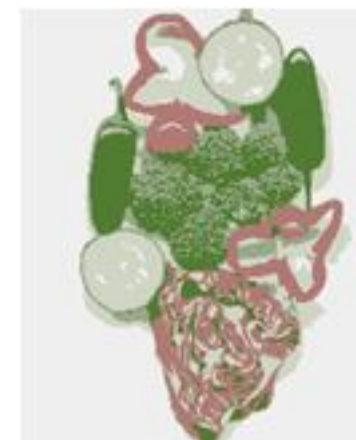


- Как мы будем записывать результат сегментации?
- Сделаем карту разметки – изображение, в каждом пикселе которого номер сегмента, которому принадлежит этот пиксель
- Визуализировать удобно каждый сегмент своим цветом



Алгоритмы к рассмотрению

- Бинаризация и выделения связанных компонент
- Последовательное сканирование
- Метод К-средних





Алгоритмы к рассмотрению

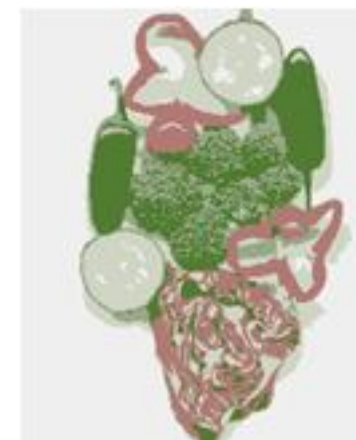
- Бинаризация и выделения связанных компонент



- Последовательное сканирование



- Метод К-средних





Простейшая сегментация

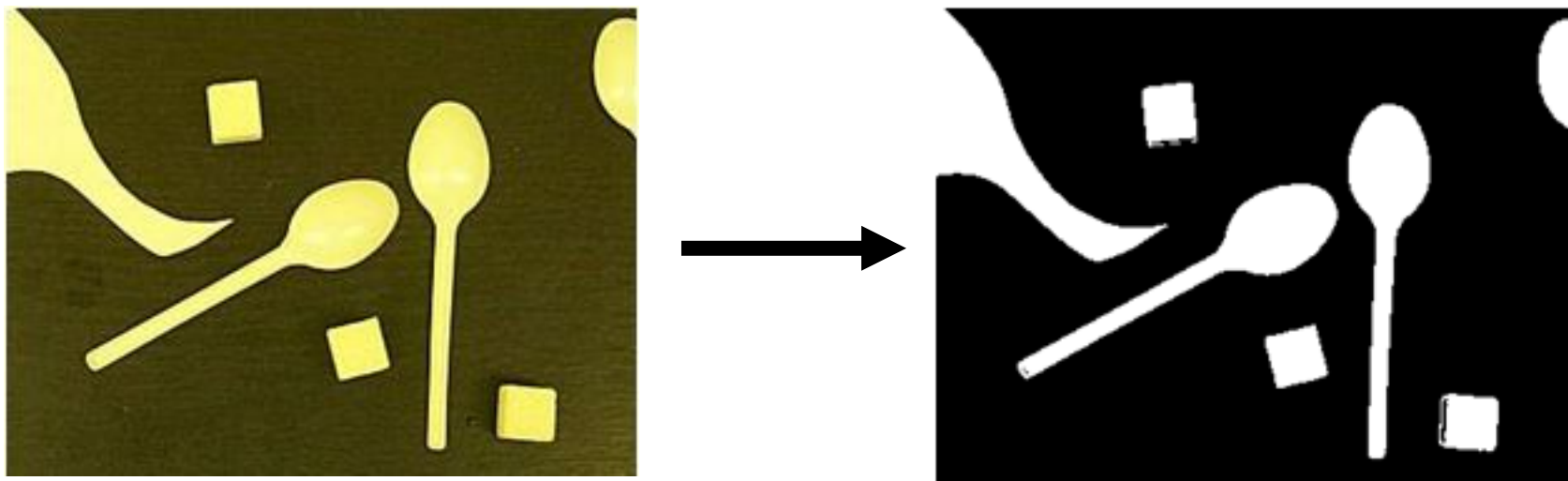
Чем отличаются объекты на этом изображении?



- Все объекты яркие, фон тёмный
- Для сегментации такого изображения нам достаточно:
 - пороговая бинаризация
 - обработки шума
 - выделения связанных компонент



Пороговая бинаризация



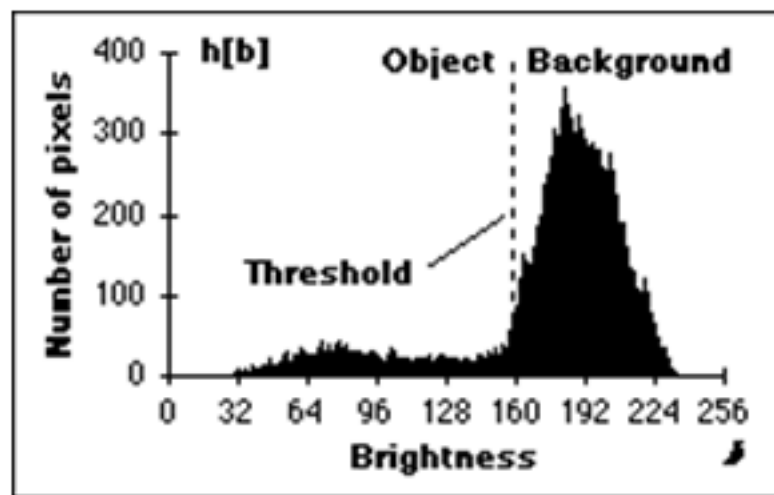
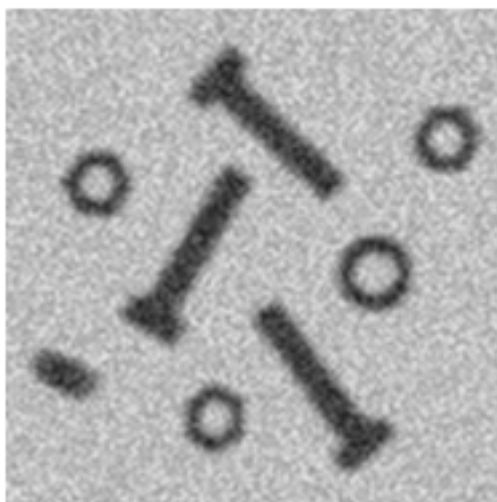
- Пороговая фильтрация (thresholding)
 - Пиксели, которых выше/ниже некоторого порога, заданного «извне», помечаются 1
 - Ниже порога помечаются 0
- Бинарное изображение – пиксели которого могут принимать только значения 0 и 1
- Бинаризация - построение бинарного изображения по полутоновому / цветному



Пороговая фильтрация

Более интересный способ – определение порога автоматически, по характеристикам изображения

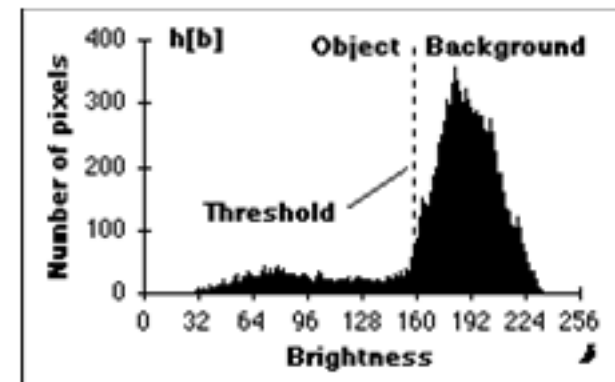
- Анализ гистограммы



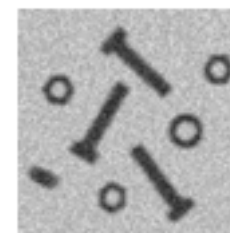


Анализ гистограммы

- Анализ симметричного пика гистограммы
- Применяется когда фон изображения дает отчетливый и доминирующий пик гистограммы, симметричный относительно своего центра.



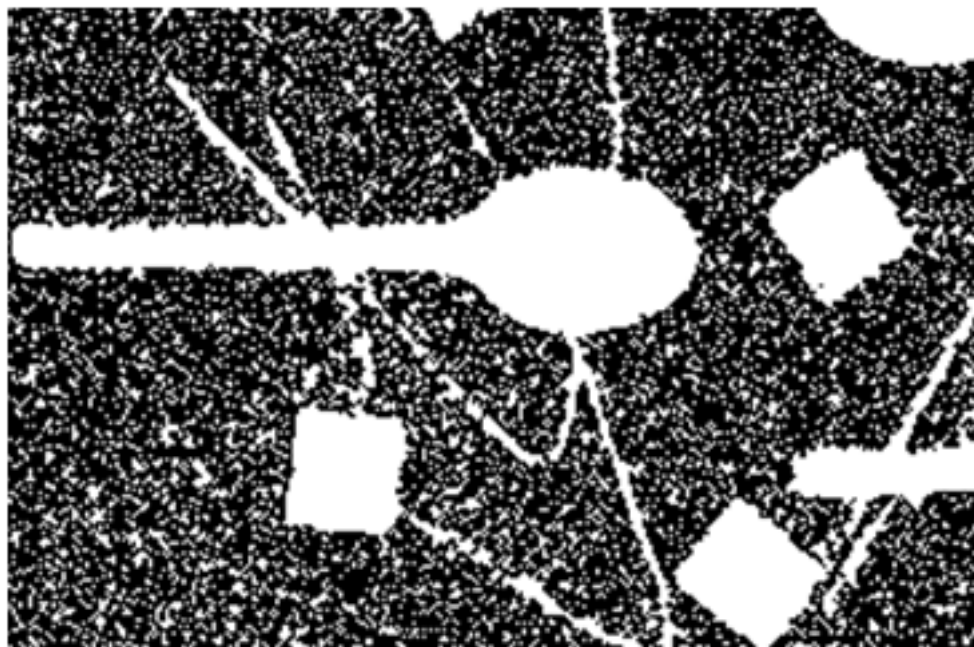
1. Сгладить гистограмму;
2. Найти ячейку гистограммы h_{\max} с максимальным значением;
3. На стороне гистограммы не относящейся к объекту (на примере – справа от пика фона) найти яркость h_p , количество пикселей с яркостью $\geq h_p$ равняется $p\%$ (например 5%) от пикселей яркости которых $\geq h_{\max}$;
4. Пересчитать порог $T = h_{\max} - (h_p - h_{\max})$;





Шум в бинарных изображениях

Пример бинарного изображения с сильным шумом

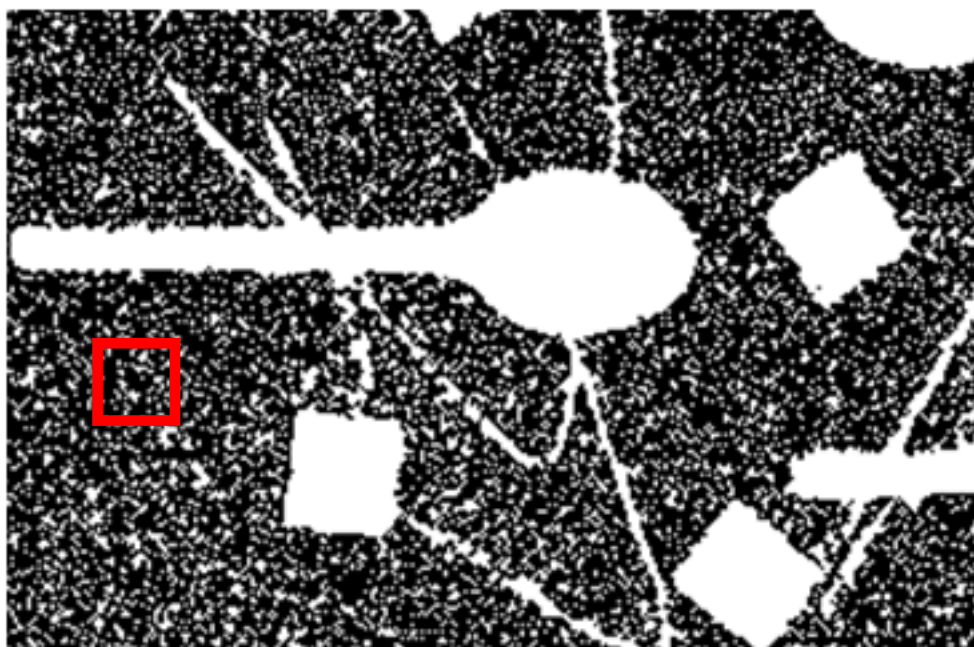


Часто возникает из-за невозможности полностью подавить шум в изображениях, недостаточной контрастности объектов и т.д.



Шум в бинарных изображениях

- ⑩ По одному пикселю невозможно определить – шум или объект?
- ⑩ Нужно рассматривать окрестность пикселя!





Подавление и устранение шума

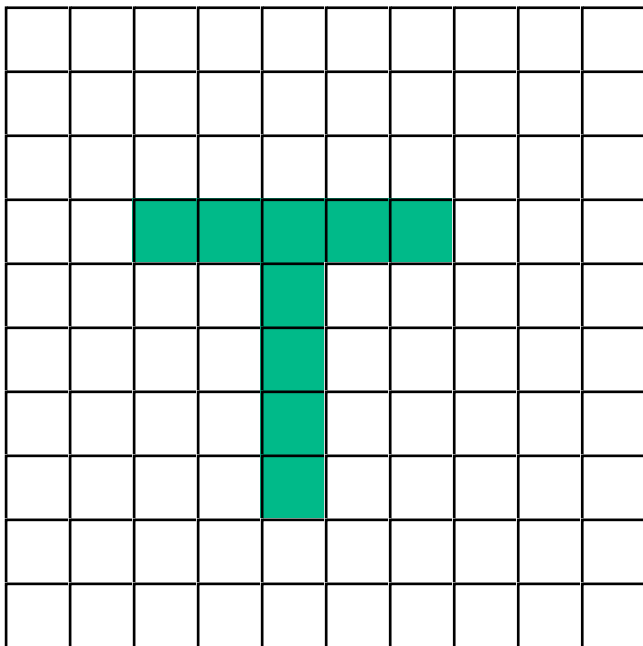
Широко известный способ - устранение шума с помощью операций математической морфологии:

- Сужение (erosion)
- Расширение (dilation)
- Закрытие (closing)
- Раскрытие (opening)

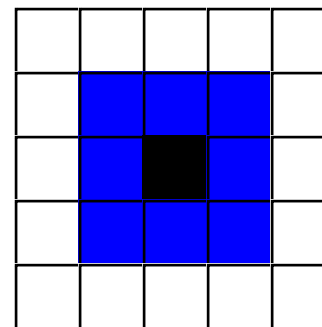


Математическая морфология

A



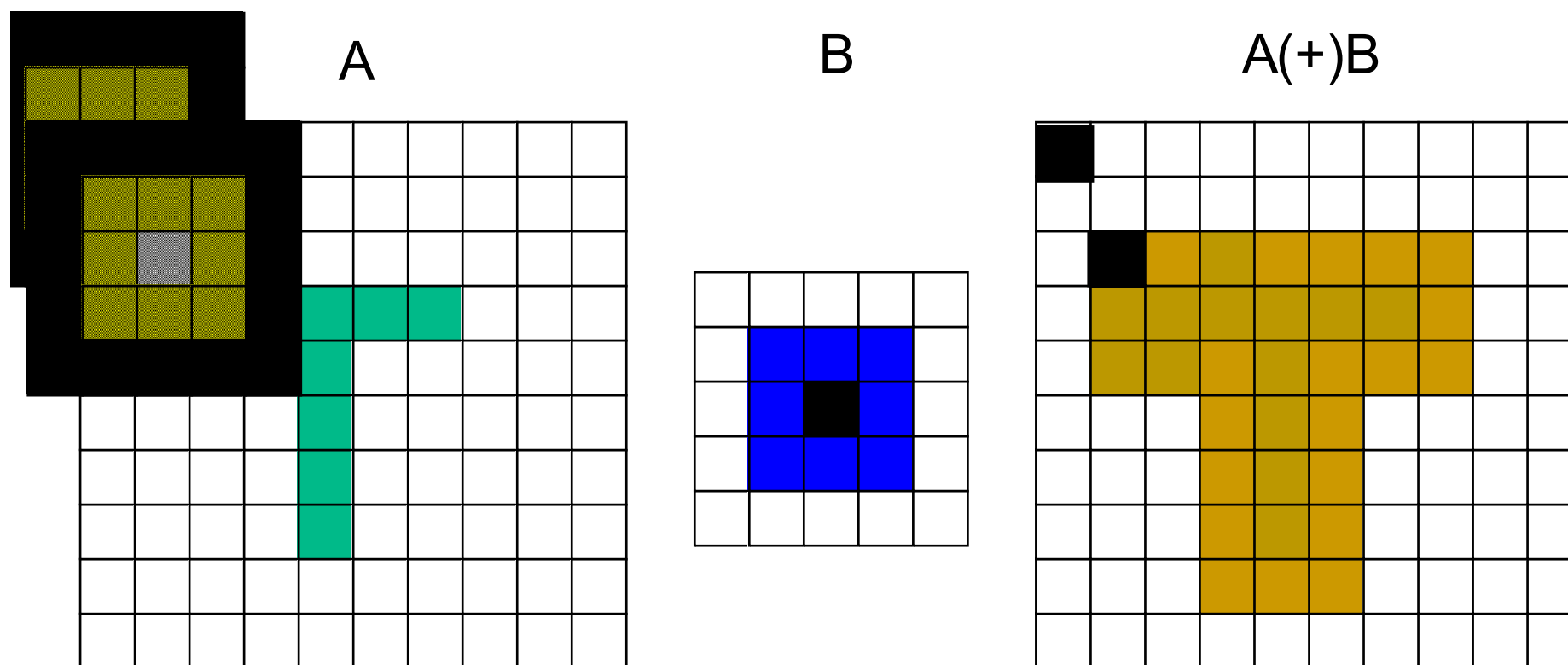
B



- Множество A обычно является объектом обработки
- Множество B (называемое структурным элементом) – инструмент обработки



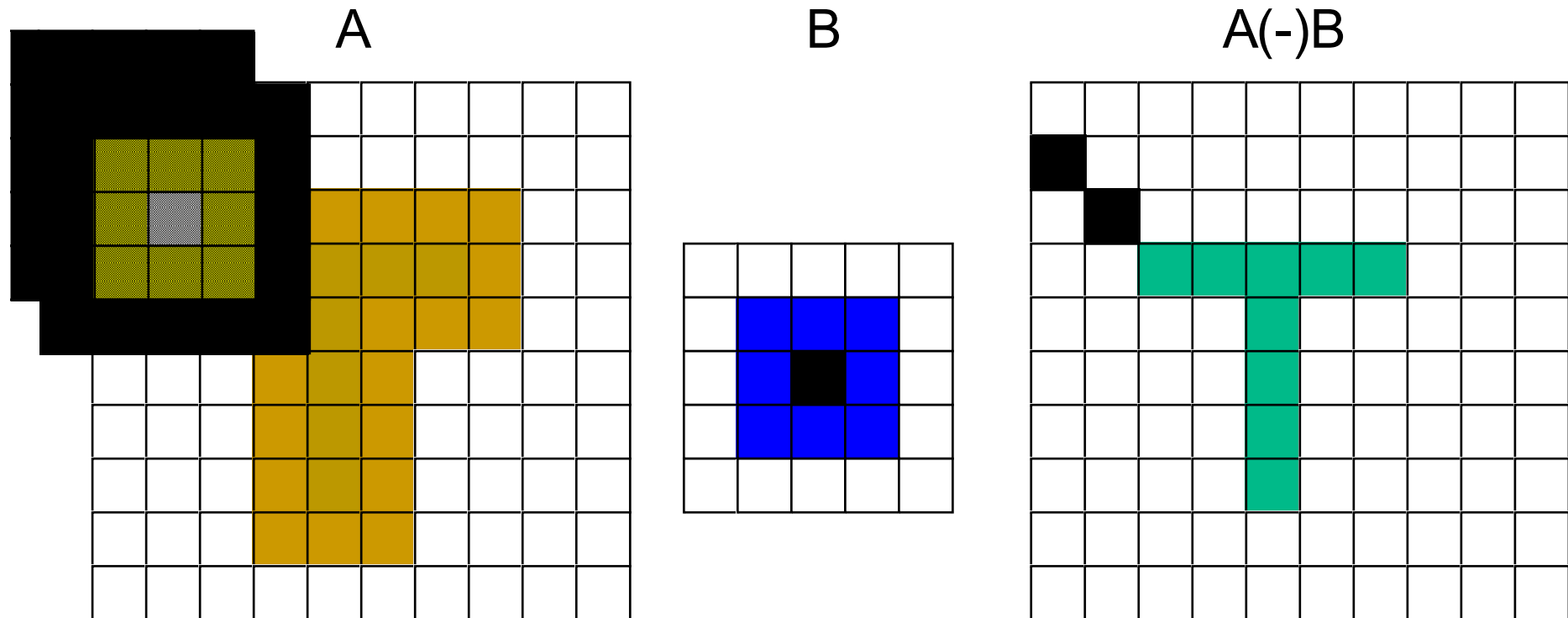
Расширение



Операция «расширение» - аналог логического «или»



Сужение



Операция «расширение» - аналог логического «и»



Результат операции сужения



$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & [1] & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & [1] & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & [1] & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$



Важное замечание

Результат морфологических операций во многом определяется применяемым структурным элементом. Выбирая различный структурный элемент можно решать разные задачи обработки изображений:

- Шумоподавление
- Выделение границ объекта
- Выделение скелета объекта
- Выделение сломанных зубьев на изображении шестерни



Операции раскрытия и закрытия

Морфологическое раскрытие (opening)

- **$\text{open}(A, B) = (A (-) B) (+) B$**

Морфологическое закрытие (closing)

- **$\text{close}(A, B) = (A (+) B) (-) B$**

Попробуйте догадаться, что эти операции делают?



Применение открытия

Применим операцию открытия к изображению с сильным шумом:



$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$



Сужение vs Открытие



Сужение



Открытие



Дефекты бинаризации

Пример бинарного изображения с дефектами
распознаваемых объектов





Применение закрытия

Применим операцию закрытия к изображению с дефектами объектов:



$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$



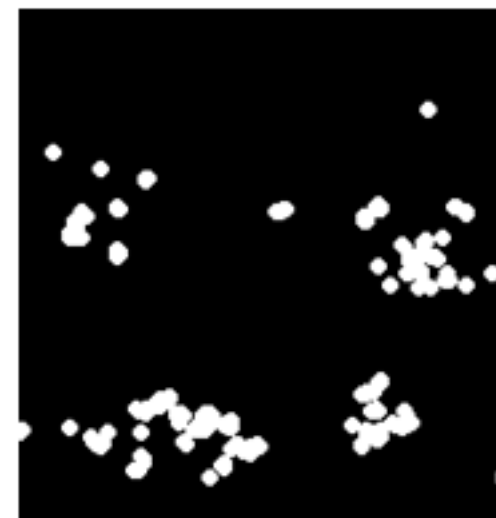
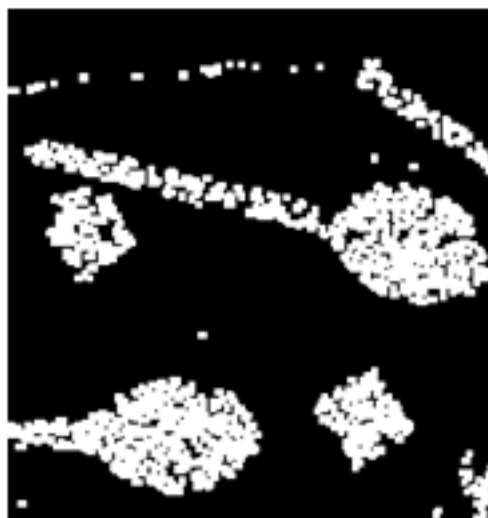
Не лучший пример для морфологии

Не во всех случаях математическая морфология так легко убирает дефекты, как хотелось бы...





Применения операции открытия



$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

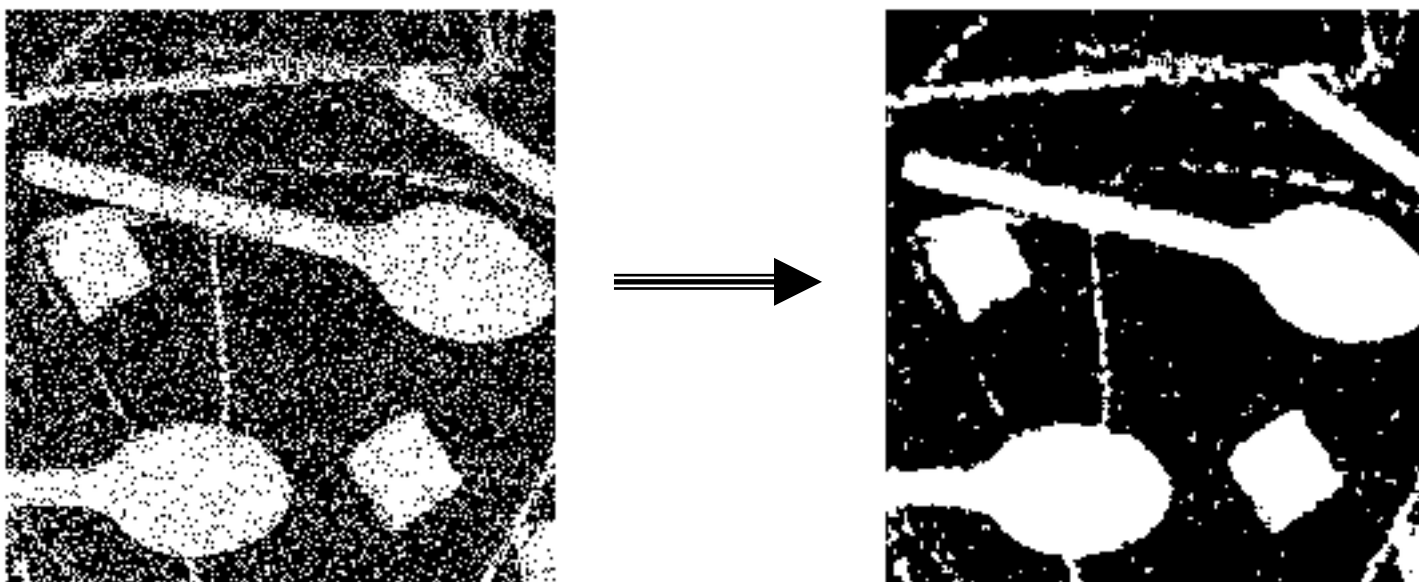
$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Часто помогает медианная фильтрация!



Медианный фильтр

Фильтр с окрестностью 3x3



Теперь можем с помощью морфологии убрать оставшиеся точки, тонкие линии и т.д.



Что дальше?



Получили бинарное
изображение



Нужна карта разметки



Выделение связных областей

- Определение связной области:
 - Множество пикселей, у каждого пикселя которого есть хотя бы один сосед, принадлежащий данному множеству.



Соседи пикселей:

	1	
2	*	3
	4	

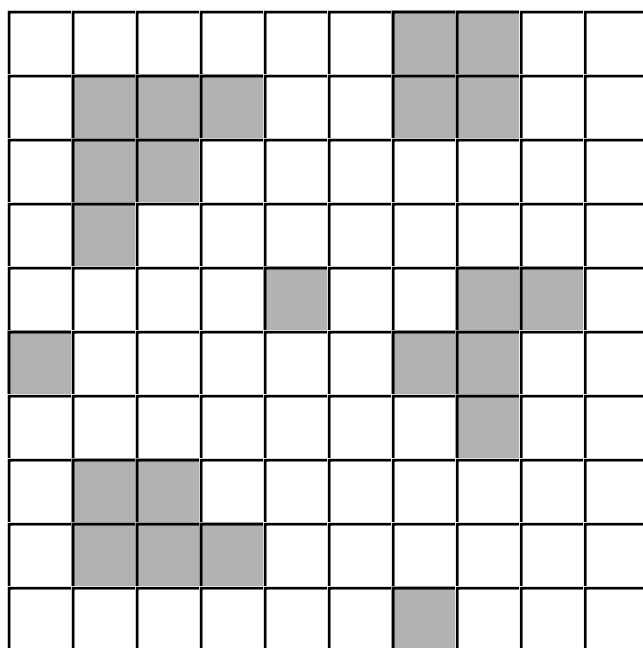
4-СВЯЗНОСТЬ

1	2	3
4	*	5
6	7	8

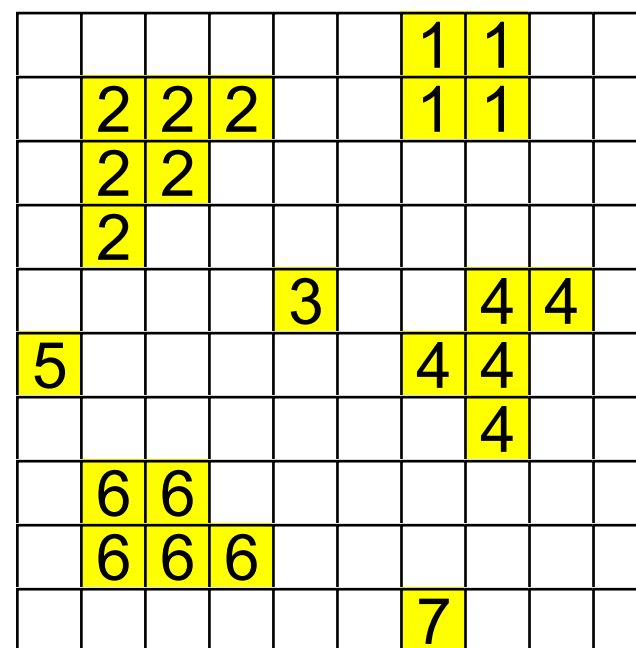
8-СВЯЗНОСТЬ



Разметка связных областей



Бинарное изображение



Размеченное изображение



Рекурсивный алгоритм

```
void Labeling(BIT* img[], int* labels[])
{
    // labels должна быть обнулена
    L = 1;
    for(y = 0; y < H; y++)
        for(x = 0; x < W; x++)
        {
            Fill(img, labels, x, y, L++);
        }
}
```



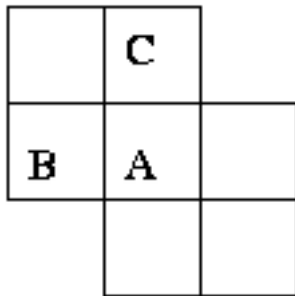
Рекурсивный алгоритм

```
void Fill(BIT* img[], int* labels[], int x, int y, int L)
{
    if( (labels[x][y] == 0) && (img[x][y] == 1) )
    {
        labels[x][y] = L;
        if( x > 0 )
            Fill(img, labels, x - 1, y, L);
        if( x < W - 1 )
            Fill(img, labels, x + 1, y, L);
        if( y > 0 )
            Fill(img, labels, x, y - 1, L);
        if( y < H - 1 )
            Fill(img, labels, x, y + 1, L);
    }
}
```



Последовательное сканирование

Последовательно, сканируем бинарное изображение сверху вниз, слева направо:



```
if A = 0
    do nothing

else if (not B labeled) and (not C labeled)
    increment label numbering and label A

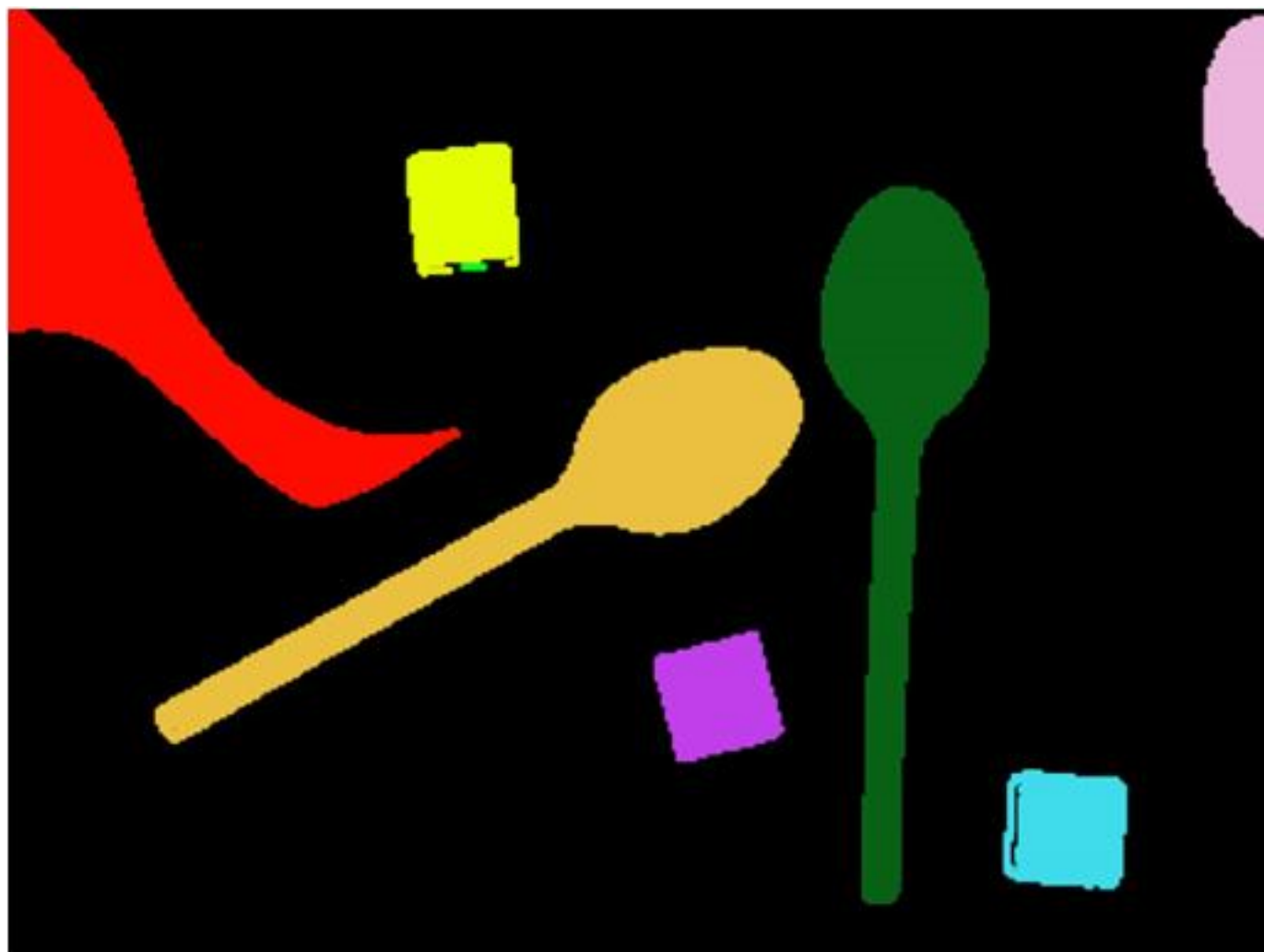
else if B xor C labeled
    copy label to A

else if B and C labeled
    if B label = C label
        copy label to A
    else
        copy either B label or C label to A
        record equivalence of labels
```

За сколько операций мы разметим изображение?



Выделенные связанные компоненты





Алгоритмы к рассмотрению

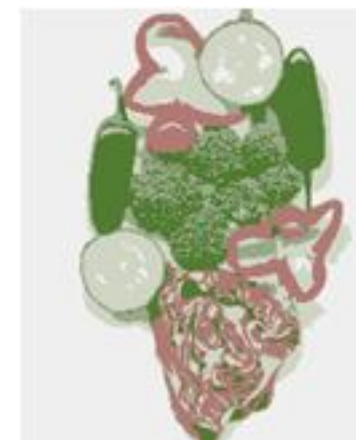
- Бинаризация и выделения связанных компонент



- Последовательное сканирование



- Метод К-средних





Сегментация

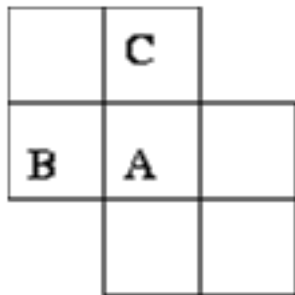


На основе
последовательного
сканирования можно
сделать и метод
сегментации изображений
на однородные области



Выделение однородных областей методом последовательного сканирования

Сканируем изображение сверху вниз, слева направо:



1. if $I(A) - I_{avg}(CI(B)) > \delta$ and $I(A) - I_{avg}(CI(C)) > \delta$ -
создаем новую область, присоединяем к ней пиксел A
2. if $I(A) - I_{avg}(CI(B)) < \delta$ xor $I(A) - I_{avg}(CI(C)) < \delta$ –
добавить A к одной из областей
3. if $I(A) - I_{avg}(CI(B)) < \delta$ and $I(A) - I_{avg}(CI(C)) < \delta$:
 1. $I_{avg}(CI(B)) - I_{avg}(CI(C)) < \delta$ –
сливаем области B и C.
 2. $I_{avg}(CI(B)) - I_{avg}(CI(C)) > \delta$ –
добавляем пиксел A к тому классу, отклонение от которого минимально.

$I(A)$ – яркость пиксела A

$CI(B)$ – область к которой принадлежит пиксел B

$I_{avg}(CI(B))$ – средняя яркость области к которой принадлежит B



Выделение однородных областей методом последовательного сканирования

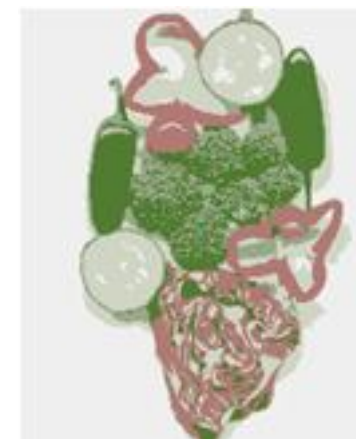
Пример:





Алгоритмы к рассмотрению

- Бинаризация и выделения связанных компонент
- Последовательное сканирование
- Метод К-средних





Кластеризация

Метод k-средних – метод **кластеризации** данных.

Целью *задачи кластеризации* является разбиение множества объектов на группы (кластеры) на основе некоторой меры сходства объектов.

Что в случае сегментации изображения «объекты»?

- «Пиксели» изображения
- Мы «группируем» пиксели по схожести
- По каким признакам можем оценить схожесть пикселей?



Алгоритм k-средних

Дано:

- Набор векторов x_i $i=1, \dots, p$;
- k – число кластеров, на которые нужно разбить набор x_i ;

Найти:

- k средних векторов m_j $j=1, \dots, k$ (центров кластеров);
- отнести каждый из векторов x_i к одному из k кластеров;

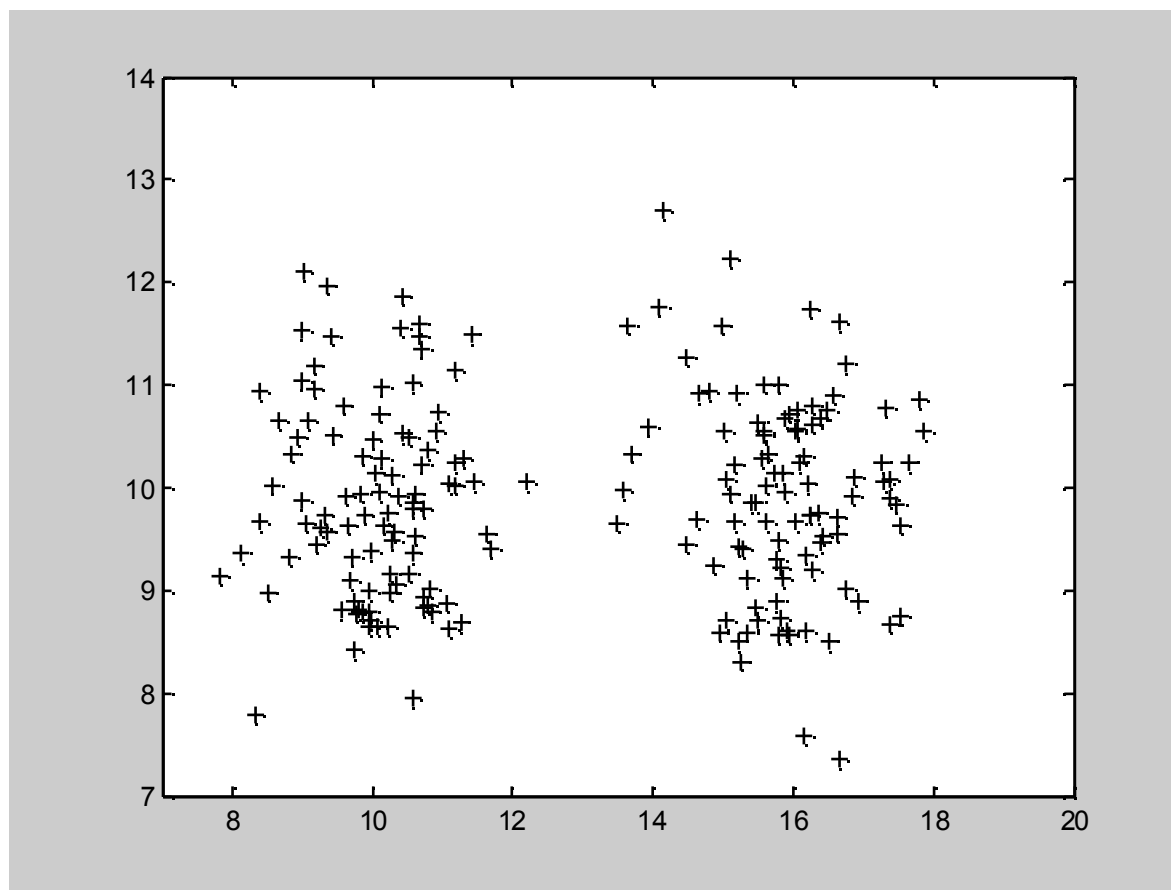


Алгоритм k-средних

1. Случайным образом выбрать k средних m_j $j=1, \dots, k$;
2. Для каждого x_i $i=1, \dots, p$ подсчитать расстояние до каждого из m_j $j=1, \dots, k$,
Отнести (приписать) x_i к кластеру j' ,
расстояние до центра которого $m_{j'}$ минимально;
3. Пересчитать средние m_j $j=1, \dots, k$ по всем кластерам;
4. Повторять шаги 2, 3 пока кластеры не перестанут изменяться;



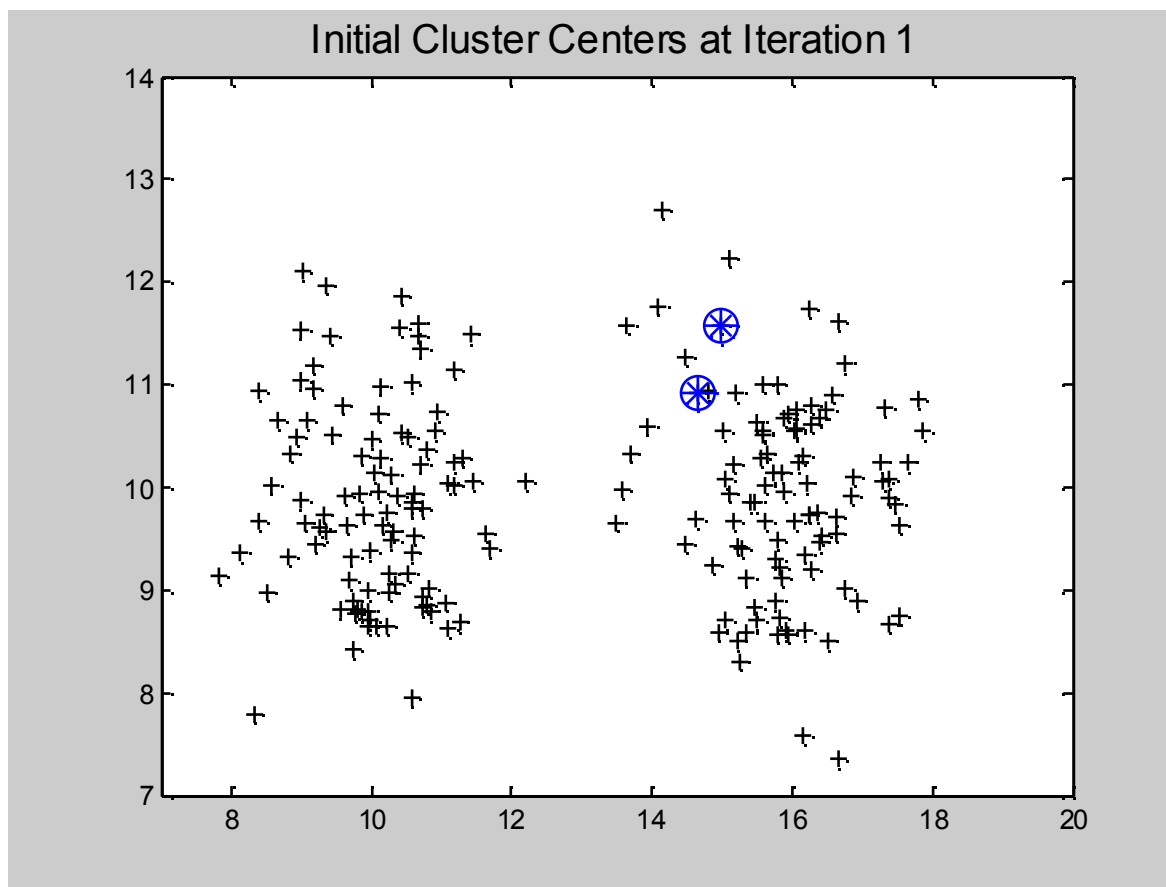
Пример кластеризации в 2D



Исходные данные



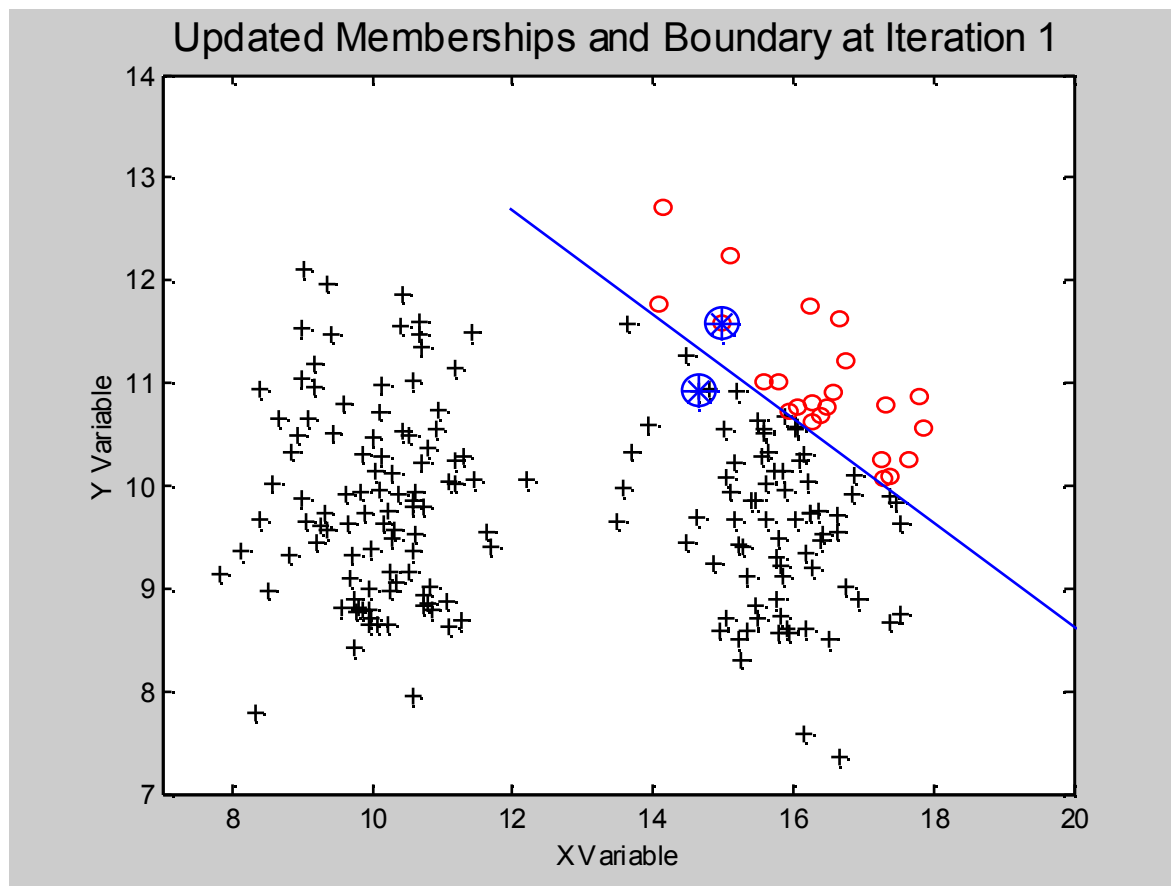
Пример кластеризации в 2D



Случайная инициализация центров кластеров (шаг 1)



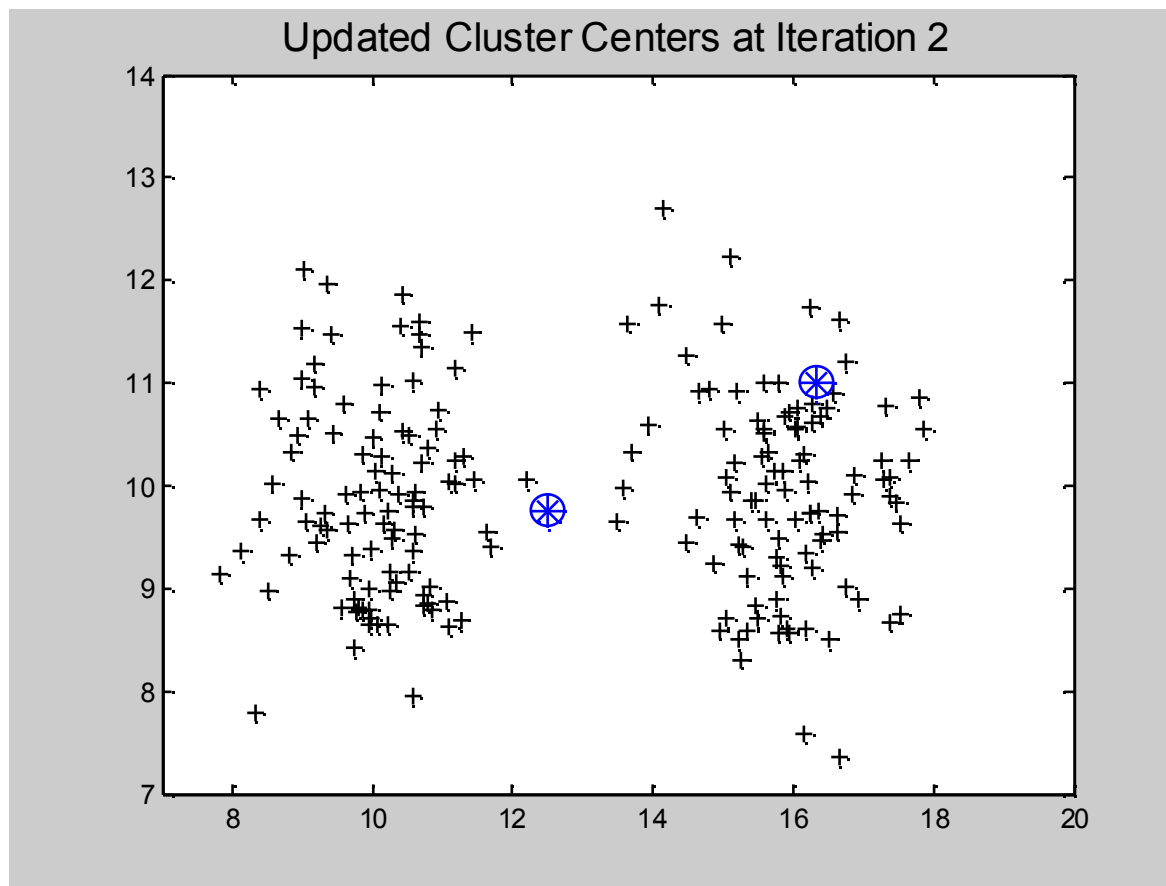
Пример кластеризации в 2D



Кластеры после первой итерации (шаг 2)



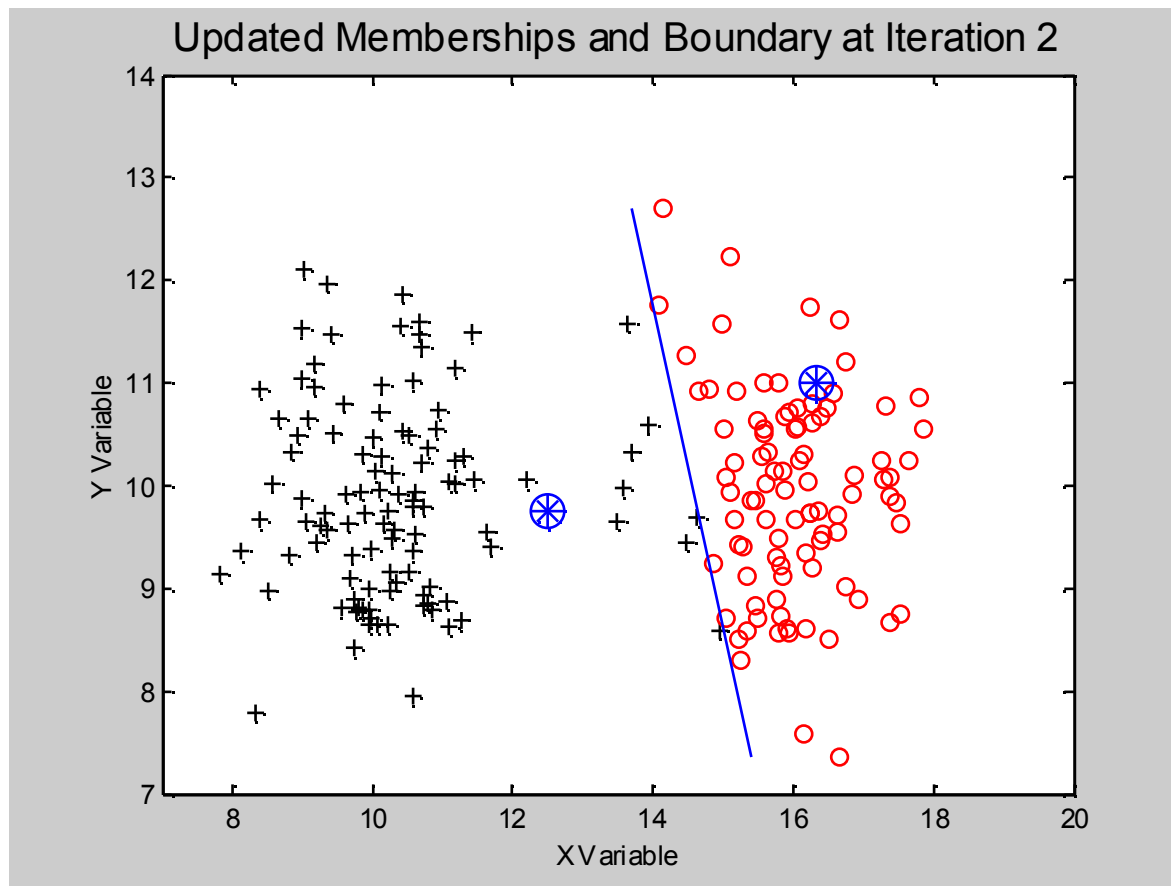
Пример кластеризации в 2D



Пересчет центров кластеров после первой итерации (шаг 3)



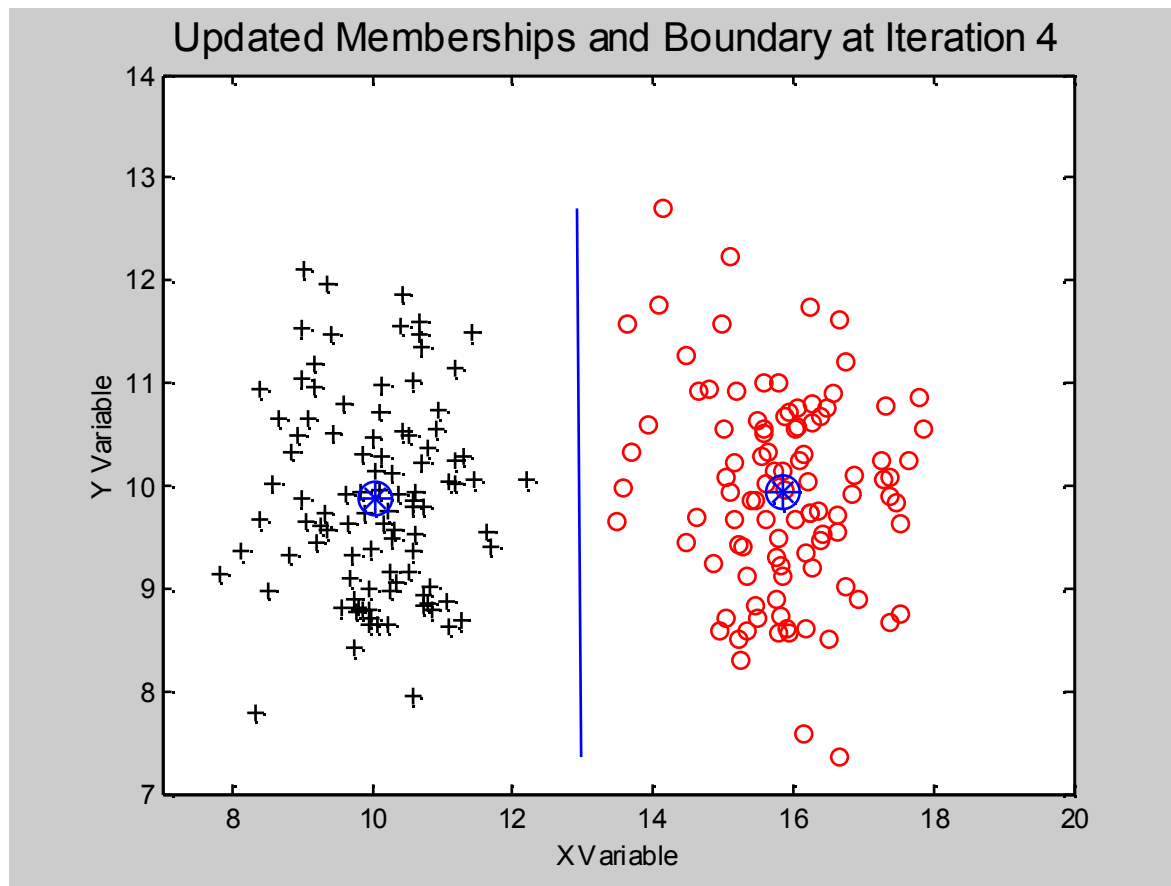
Пример кластеризации в 2D



Кластеры после второй итерации (шаг 2)



Пример кластеризации в 2D

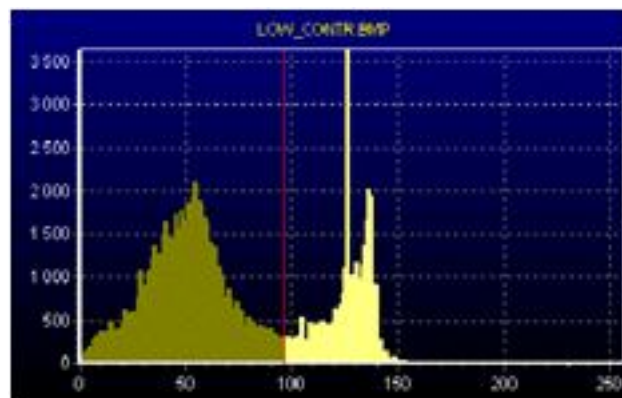


Стабильная конфигурация после четвертой итерации



Сегментации изображения по яркости

Рассматриваем *одномерное пространство яркостей пикселей* и производим в нем кластеризацию с помощью *k-средних*. Это дает автоматическое вычисление яркостных порогов.



(Для получения бинарного изображения $k=2$)



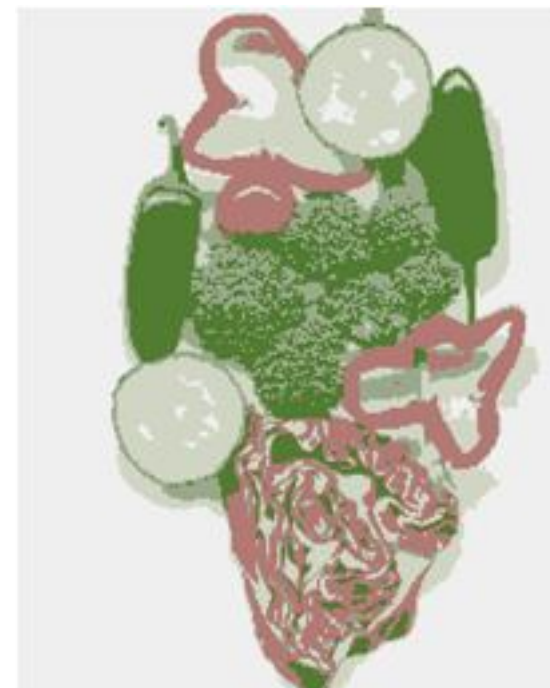
Алгоритм К-средних. Примеры.



Исходное изображение



Кластеры по яркости

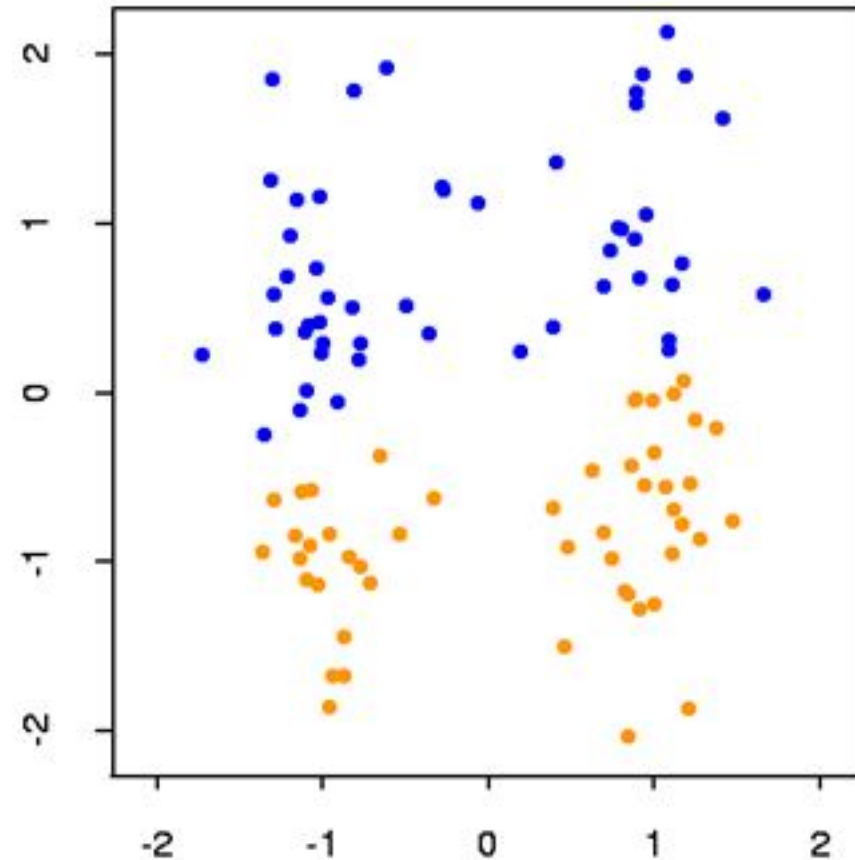


Кластеры по цвету



Алгоритм К-средних

- Однопараметрический
 - Требуется знание только о количестве кластеров
- Рандомизирован
 - Зависит от начального приближения
- Не учитывает строения самих кластеров



Есть целый ряд других, более совершенных методов кластеризации!



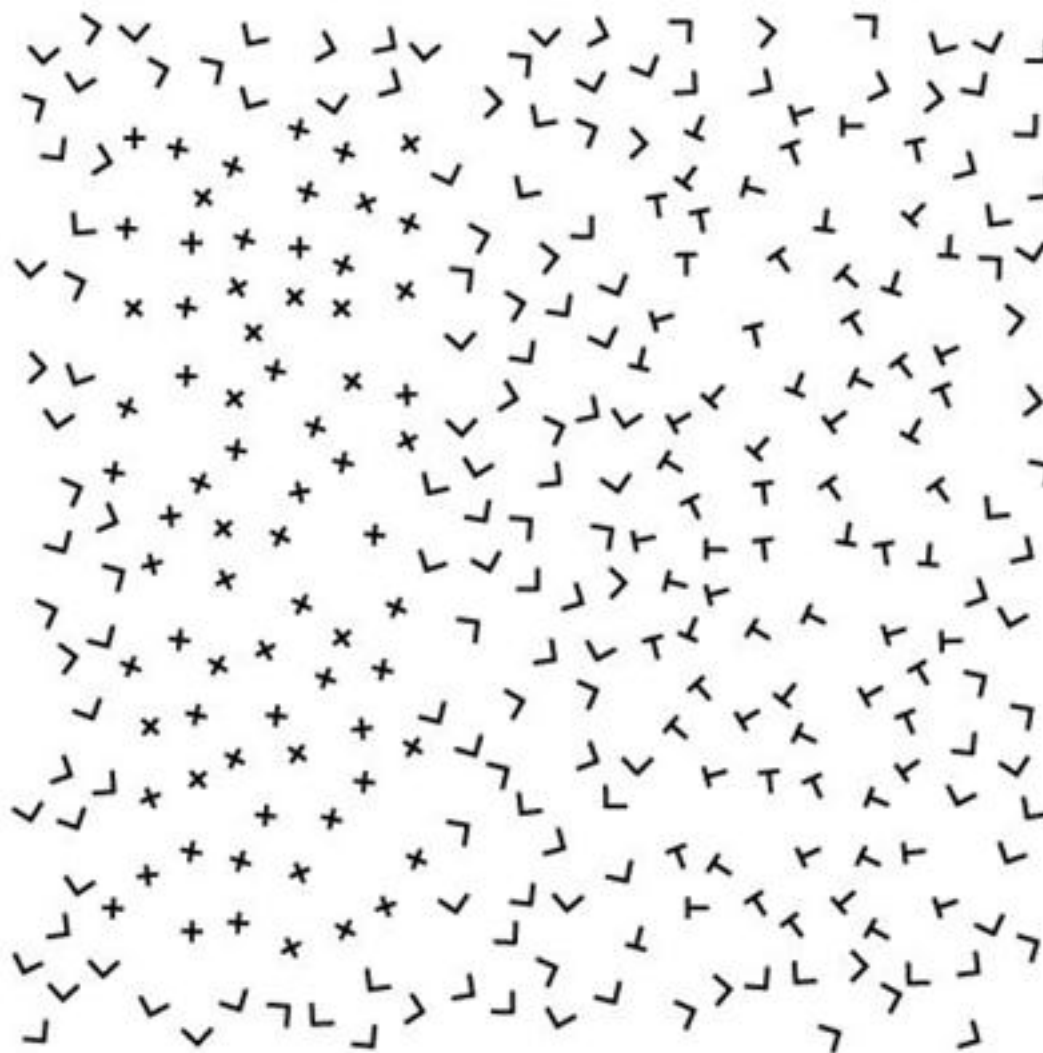
Признаки изображения

Какие признаки мы можем использовать для сравнения пикселей и регионов?

- Яркость
- Цвет
- ?



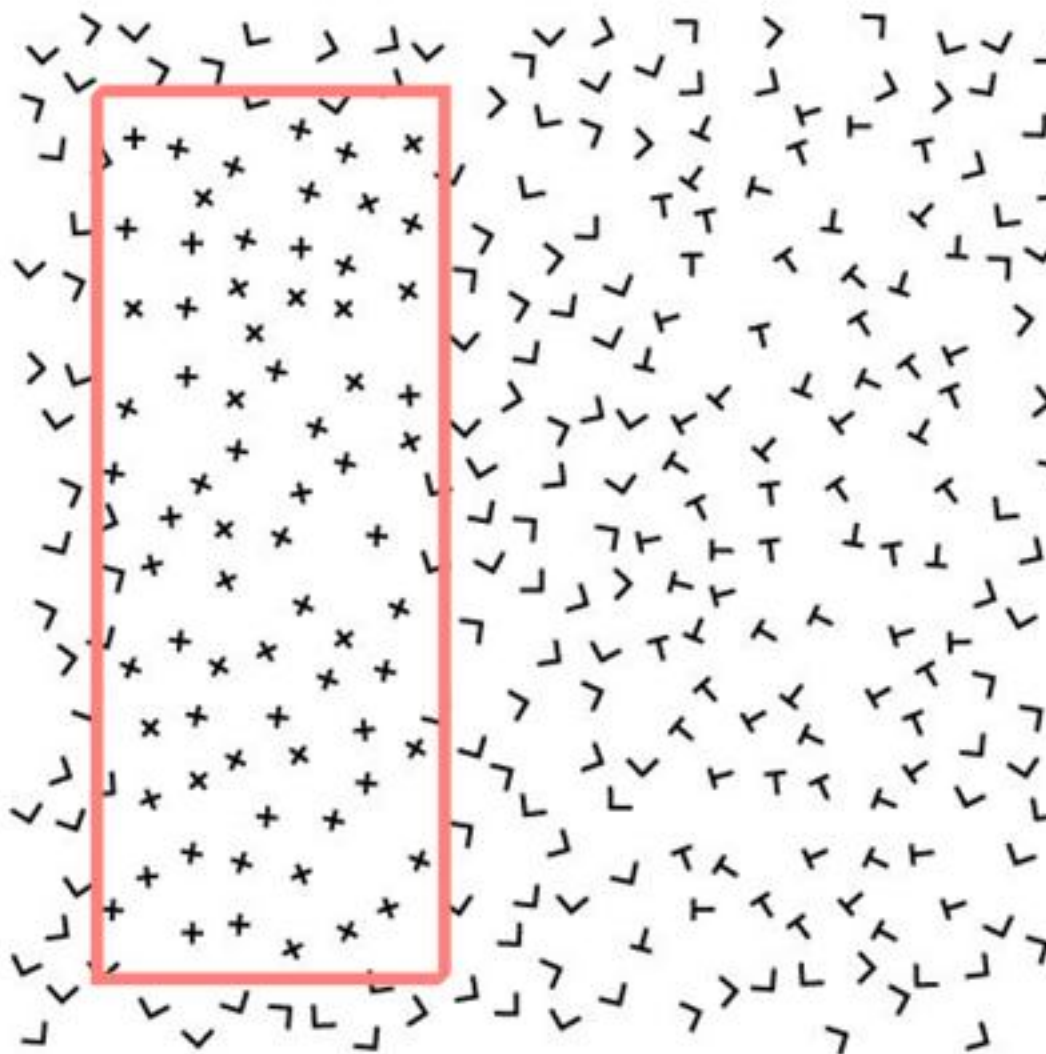
Пример



Видите отдельные области?



«Текстура»





«Текстура»

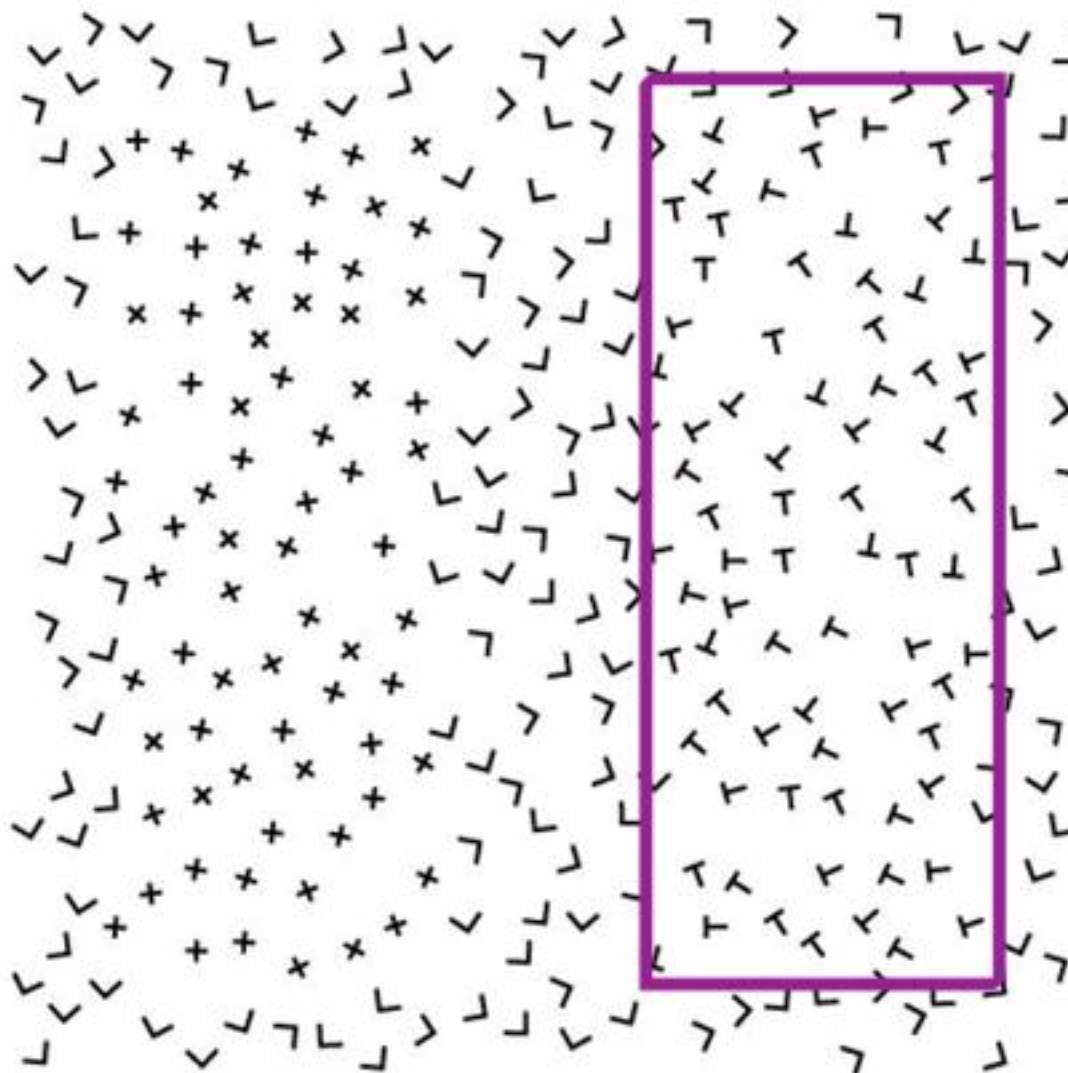
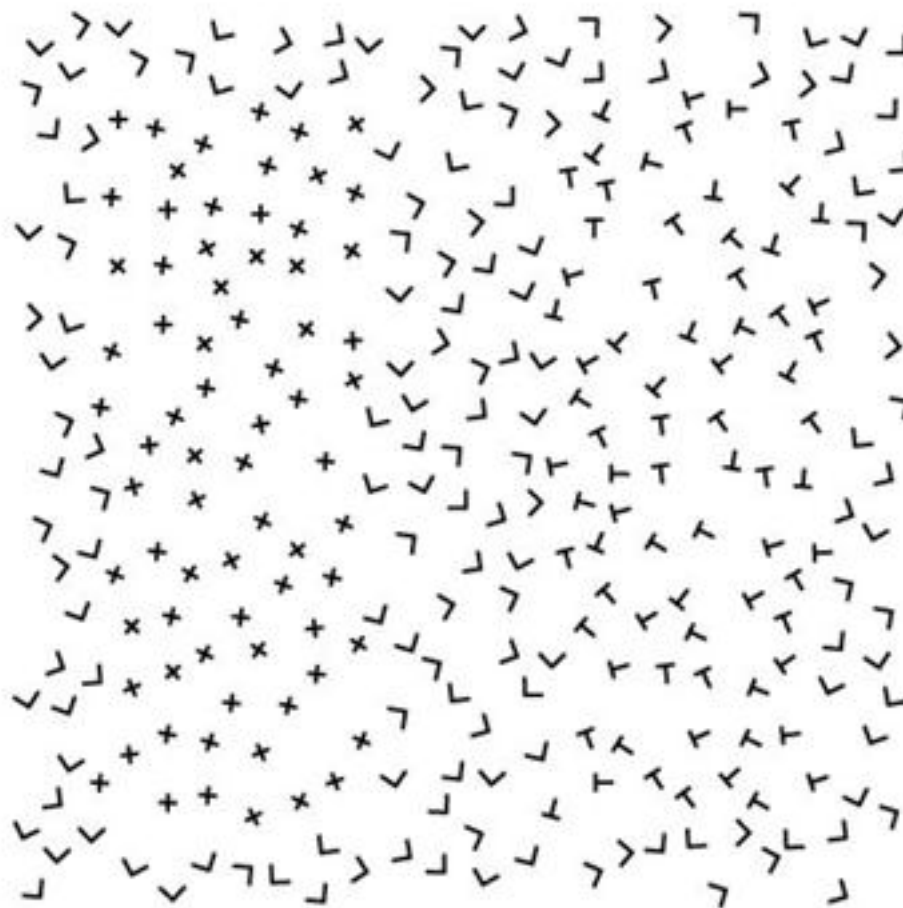


Image source: Landy & Graham (2004)



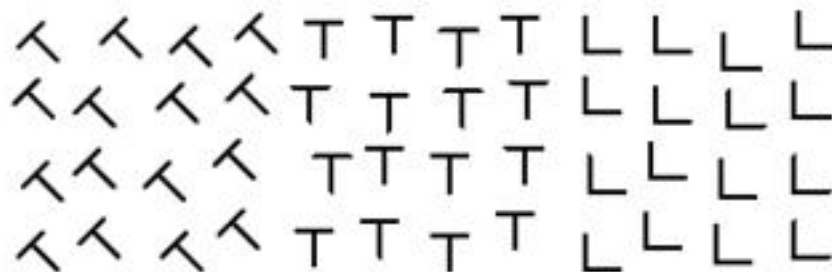
«Текстура»



Типичный пример текстурного шаблона для исследований психофизиологического восприятия изображений



Текстура



(a)

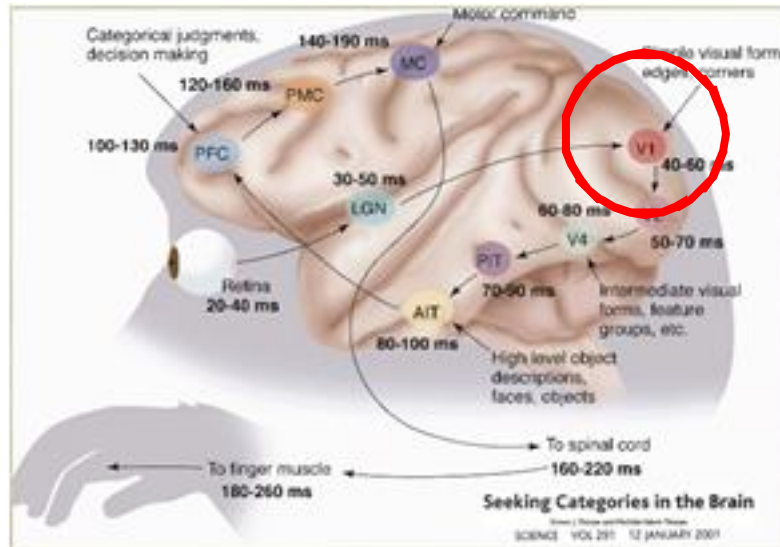


(b)

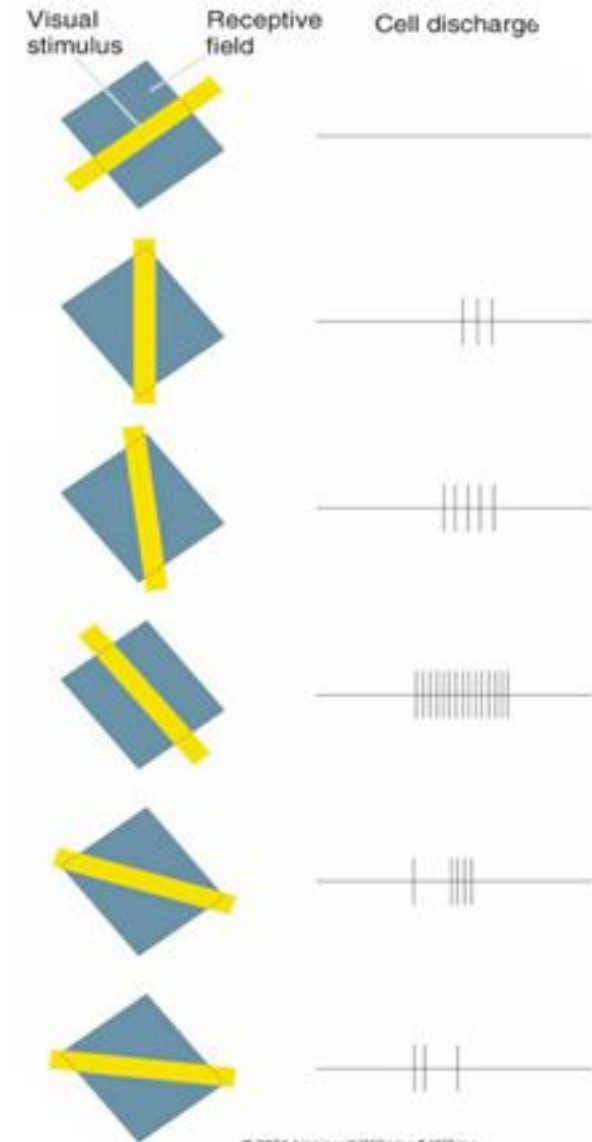
- Это типичные примеры текстурных шаблонов для исследований психофизиологического восприятия изображений
- Человек явно использует не только яркость и цвет, но и ориентацию краёв (градиентов изображения), их распределение, для анализа изображений



«Простые клетки» V1

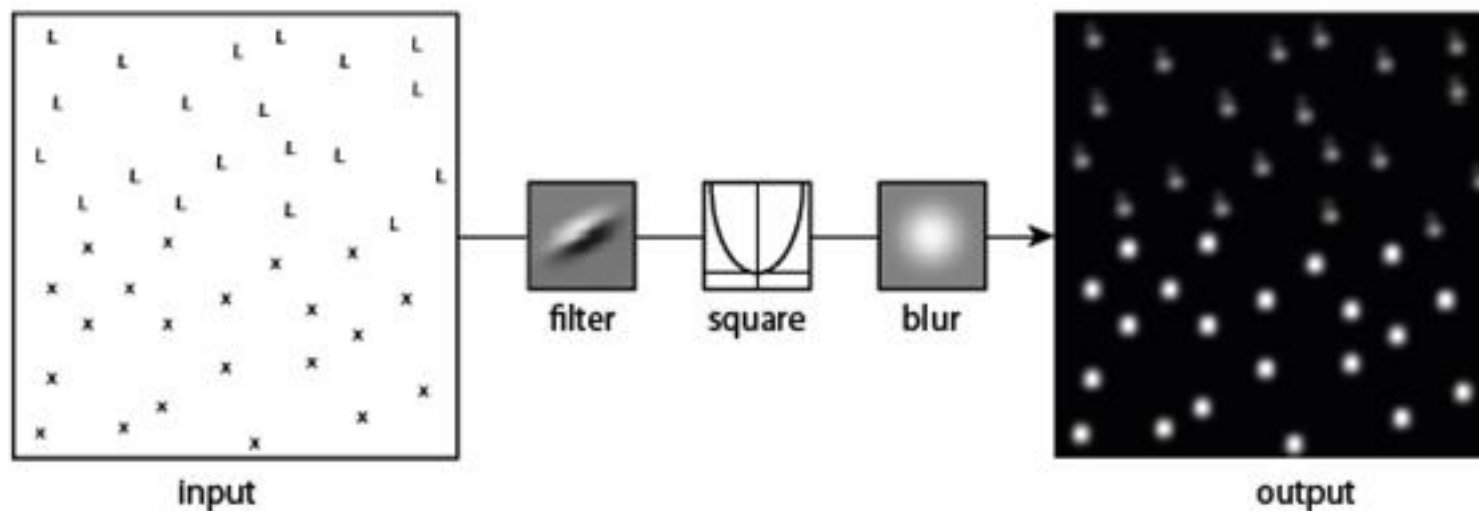


- В первичной визуальной коре головного мозга есть клетки, чувствительные к краям определенной ориентации
- Для каждой области есть набор таких клеток, чувствительные к краям разной ориентации





Анализ текстуры



- Выберем фильтр, чувствительный к краю определенной ориентации
- Результат фильтрации сгладим
- Будут «подсвечены» области, содержащие текстуру с краями заданной ориентации

Pietro Perona and Jitendra Malik «Detecting and Localizing edges composed of steps, peaks and roofs», ICCV 1990



Банки фильтров

- Возьмём теперь несколько фильтров разного масштаба и ориентации
- Такой набор называют «банк фильтров»
- Каждый пиксель изображения после обработки банком фильтров даёт вектор признаков
- Этот вектор признаков эффективно описывает локальную текстуру окрестности пикселя
- Активно используется в сегментации, распознавании изображений и т.д.





Фильтры Габора

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

$$x' = x \cos(\theta) + y \sin(\theta)$$

$$y' = -x \sin(\theta) + y \cos(\theta)$$

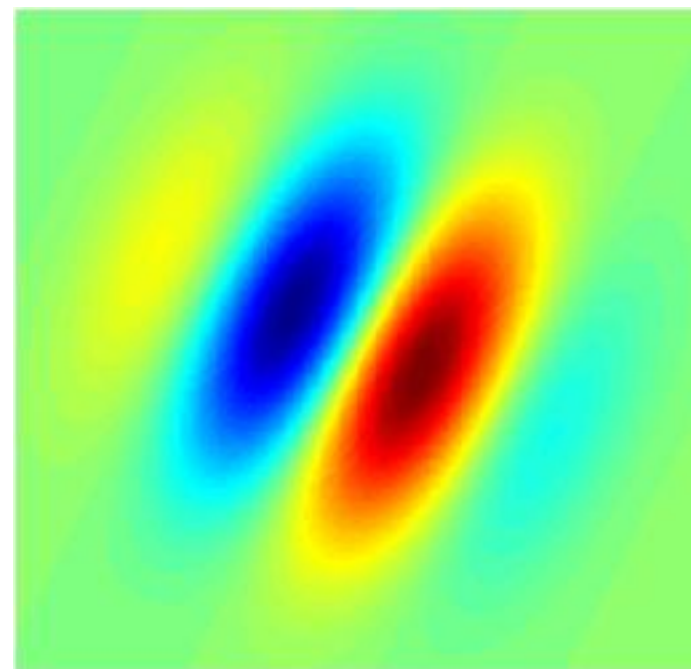
θ - ориентация

λ - длина волны

σ - сигма гауссиана

γ - соотношение размеров (aspect ratio), «эллиптичность фильтра»

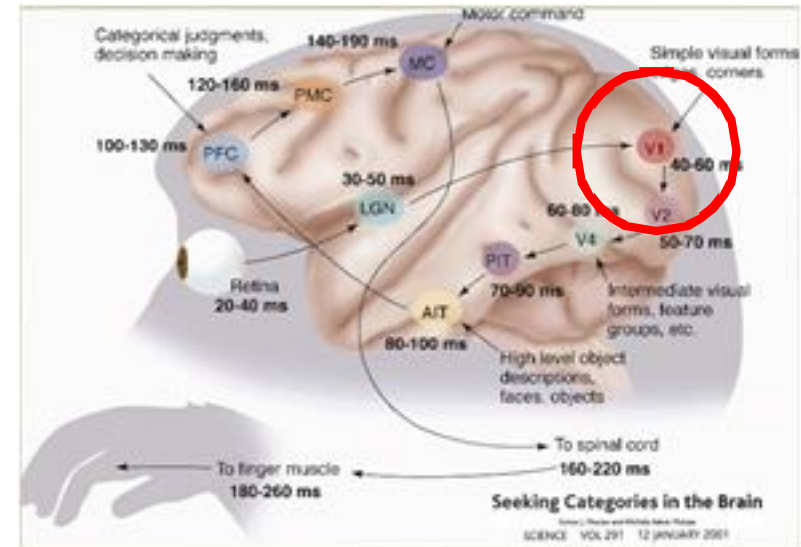
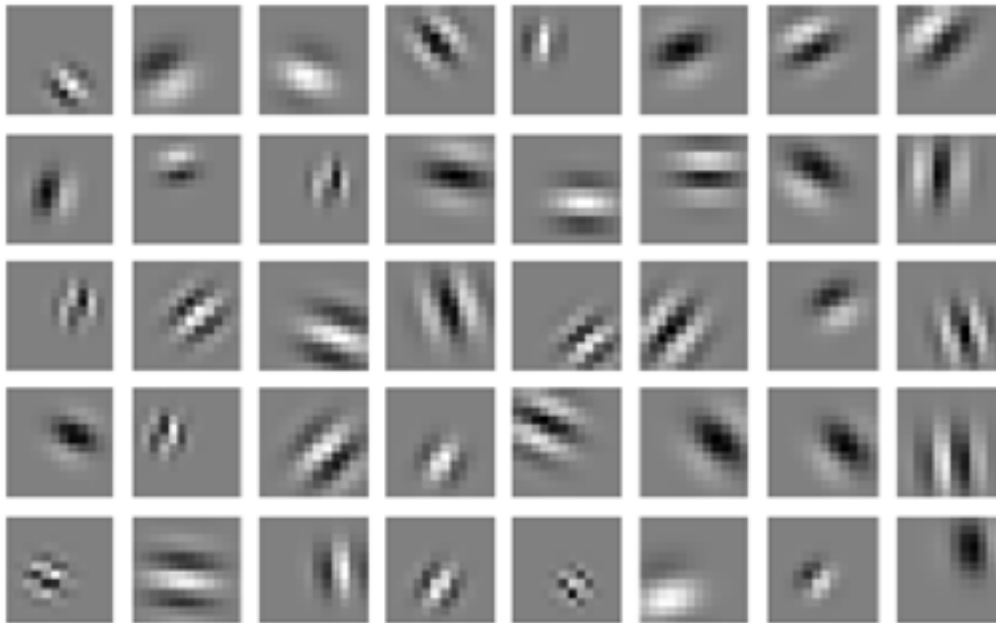
ψ - сдвиг фазы



- 2D фильтр Габора – ядро гауссиана, домноженное на синусоиду
- Предложены в 1947 Денисом Габором (нобелевским лауреатом), независимо переоткрыты в 1980 году



Связь со зрением человека



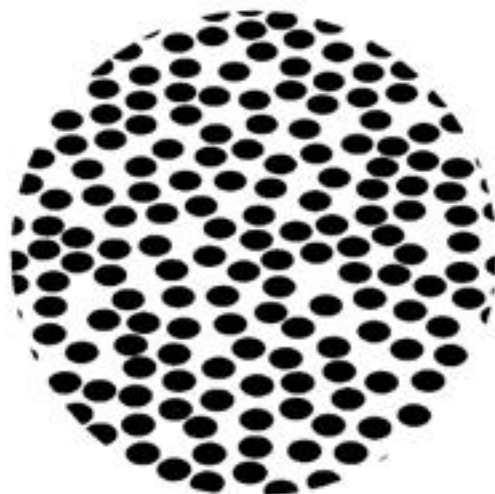
- Похожи на форму рецептивных полей простых клеток (simple cells) в визуальной коре мозга человека
- Стандартный банк фильтров для анализа текстуры, называют «биологически обусловленным»

J. G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles.," *Vision research*, vol. 20, no. 10, pp. 847–856, 1980.

J. G. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. Acoust. Speech Sigal Process.*, vol. 36, no. 7, pp. 1169–1179, 1988



Психологическое свойство текстуры





Форма из текстуры

- Человек интуитивно считает текстуру **изотропной**, т.е. с постоянными свойствами на поверхности объекта
- Shape from texture: Исходя из предположения об изотропности шаблона текстуры, можно определить наклон поверхности

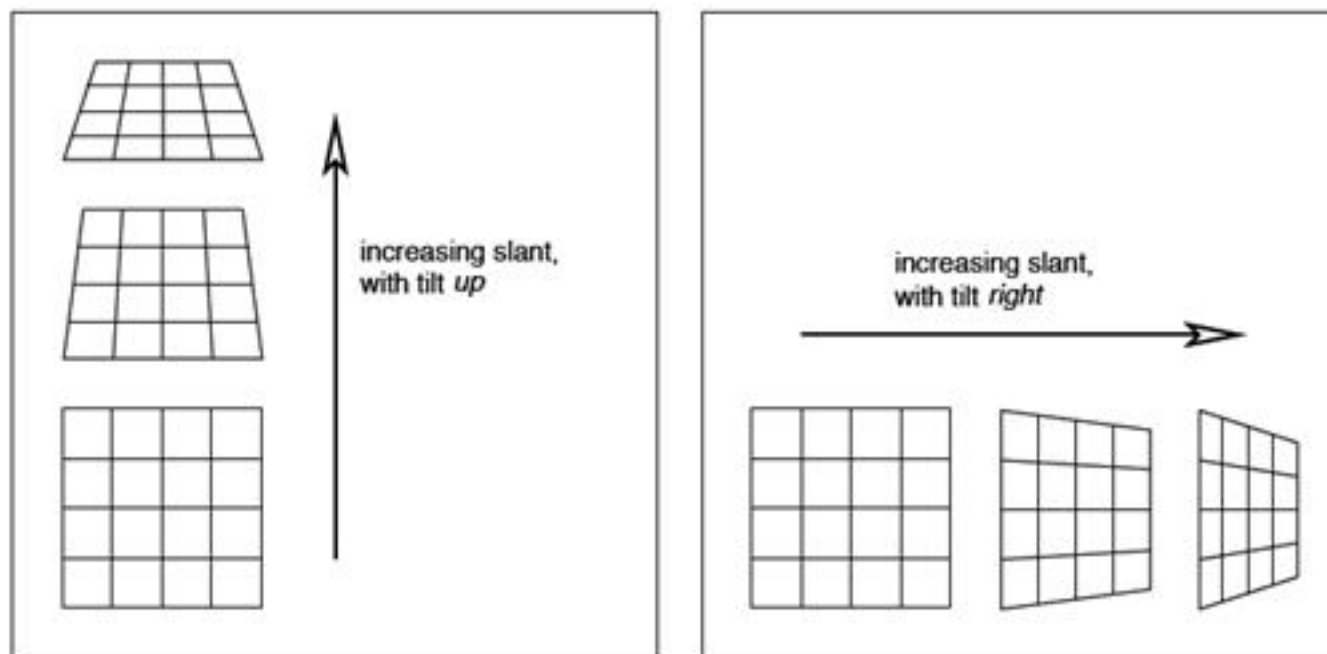
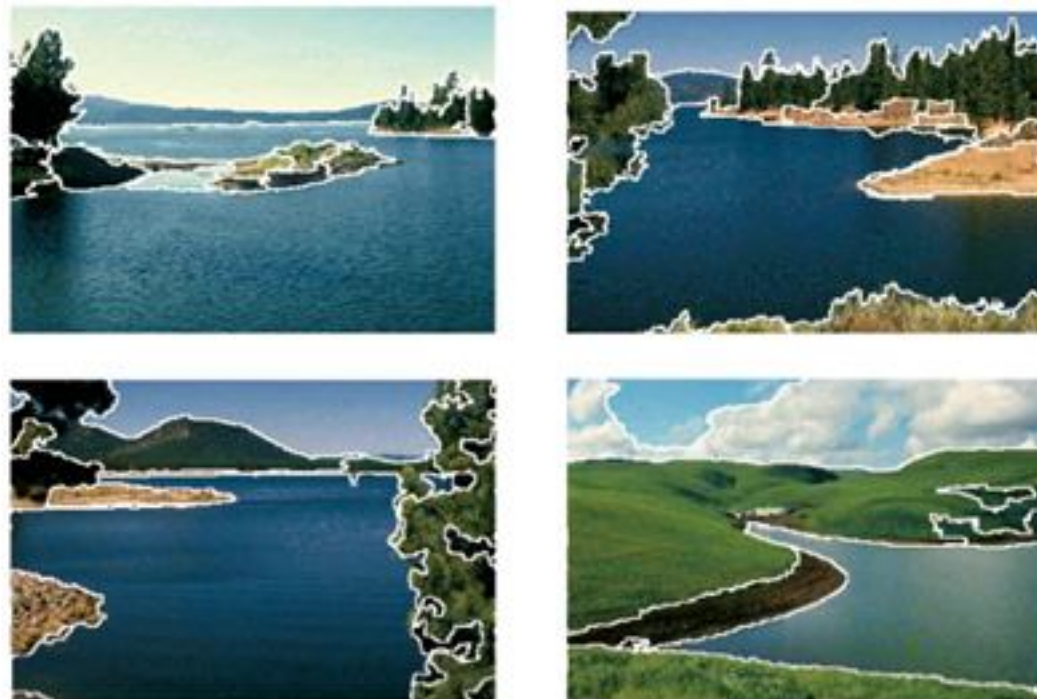


Figure 8.7. Surface orientation is often characterized in terms of *slant* and *tilt*.



Современные алгоритмы



- Задача сегментации изображений продолжает активно исследоваться.
- Иногда используется один признак, иногда множество
- Подробнее алгоритмы сегментации рассматриваются в с\к «Доп. главы компьютерного зрения»



План

Лекция 3

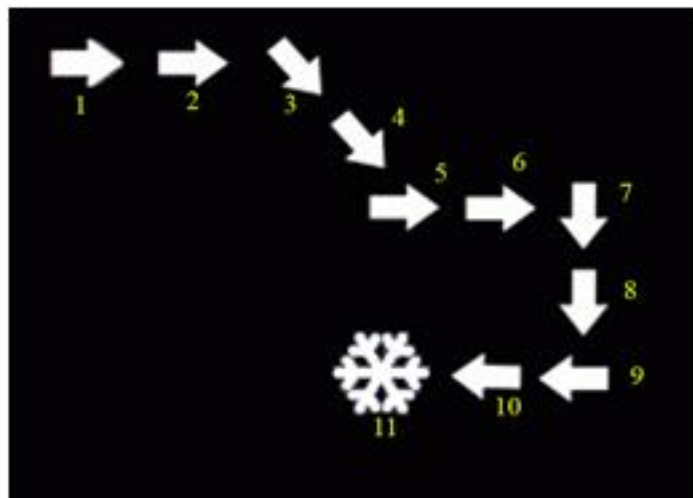
- Сопоставление шаблонов
- Основы сегментации изображений
- **Анализ сегментов**

Лекция 4

- Введение в машинное обучение на примере метода опорных векторов
- Алгоритм поиска пешеходов [Dalal & Triggs 2005]



Анализ выделенных областей



Для анализа требуется вычислить некоторые числовые характеристики (признаки) областей:

- геометрические признаки
- фотометрические признаки

На основе этих характеристик можно классифицировать получаемые области





Геометрические признаки

Для каждой области можно подсчитать некий набор простейших числовых характеристик:

- Площадь
- Центр масс
- Периметр
- Компактность
- Ориентацию главной оси инерц
- Удлиненность (эксцентриситет)

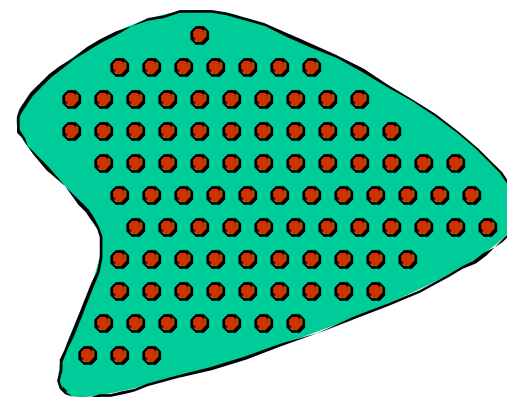




Площадь и центр масс

- Площадь – количество пикселей в области;

$$A = \sum_{x=0}^m \sum_{y=0}^n I(x, y)$$



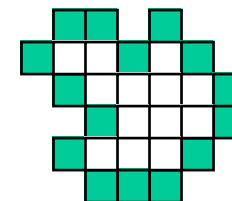
- Центр масс

$$\bar{x} = \frac{\sum_{x=0}^m \sum_{y=0}^n x I(x, y)}{A}; \bar{y} = \frac{\sum_{x=0}^m \sum_{y=0}^n y I(x, y)}{A}$$



Периметр и компактность

- Периметр – количество пикселей принадлежащих границе области;



- Компактность – отношение квадрата периметра к площади;

$$C = \frac{P^2}{A}$$

Наиболее компактная фигура – $C = 4\pi$
круг:





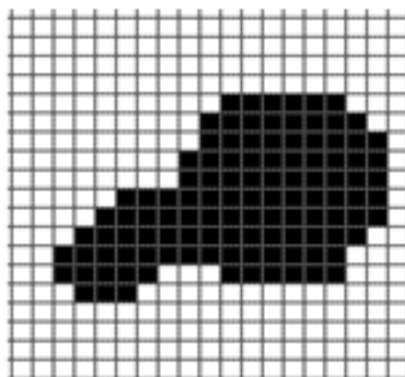
Подсчет периметра области

1. Пиксель лежит на границе области, если он сам принадлежит области и хотя бы один из его соседей области не принадлежит.
(внутренняя граница)
2. Пиксель лежит на границе области, если он сам не принадлежит области и хотя бы один из его соседей области принадлежит.
(внешняя граница)

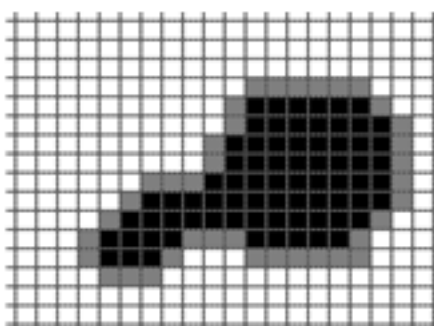
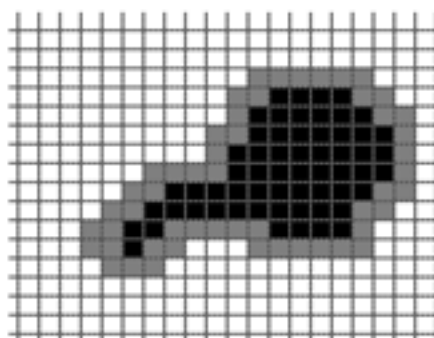
Периметр зависит также от того 4-х или 8-ми связность используется для определения соседей.



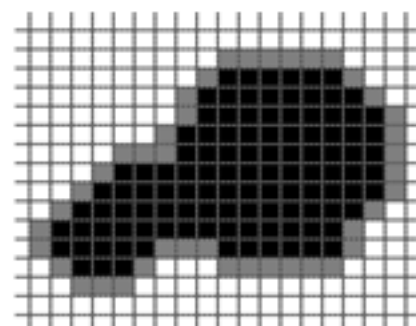
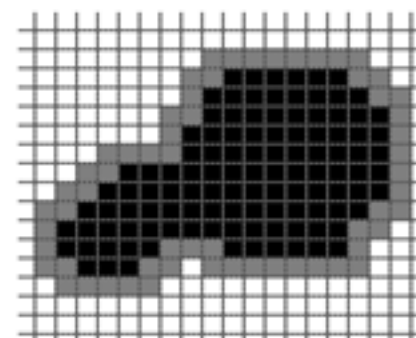
Пример периметров области



Область



Внутренняя граница



Внешняя граница



Операция оконтуривания объекта

При работе с бинарными изображениями контуры объекта можно получить с помощью операций математической морфологии

Внутреннее оконтуривание

- $C_I = A - (A (-) B)$

Внешнее оконтуривание

- $C_O = (A (+) B) - A$



Пример оконтуривания объекта





Статистические моменты области

Дискретный центральный момент m_{ij} области определяется следующим образом:

$$m_{ij} = \sum_{x,y \in S} (x - \bar{x})^i (y - \bar{y})^j I(x, y)$$

Центр масс области



Инвариантные характеристики

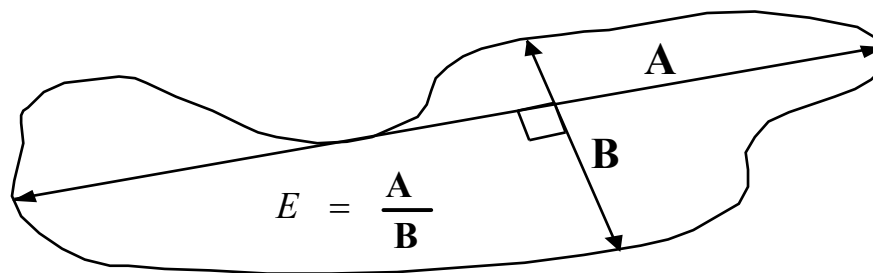
Для распознавания нас интересуют характеристики инвариантные по отношению к масштабированию, переносу, повороту:

- Удлиненность, нецентрированность (эксцентриситет)

$$elongation = \frac{m_{20} + m_{02} + \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}{m_{20} + m_{02} - \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}$$

- Компактность

$$C = \frac{P^2}{A}$$

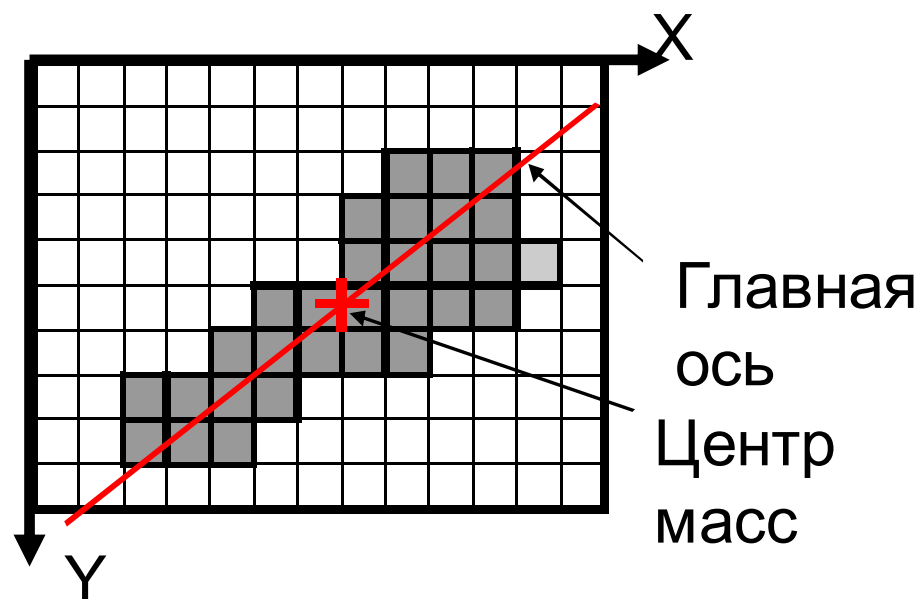




Ориентация главной оси инерции

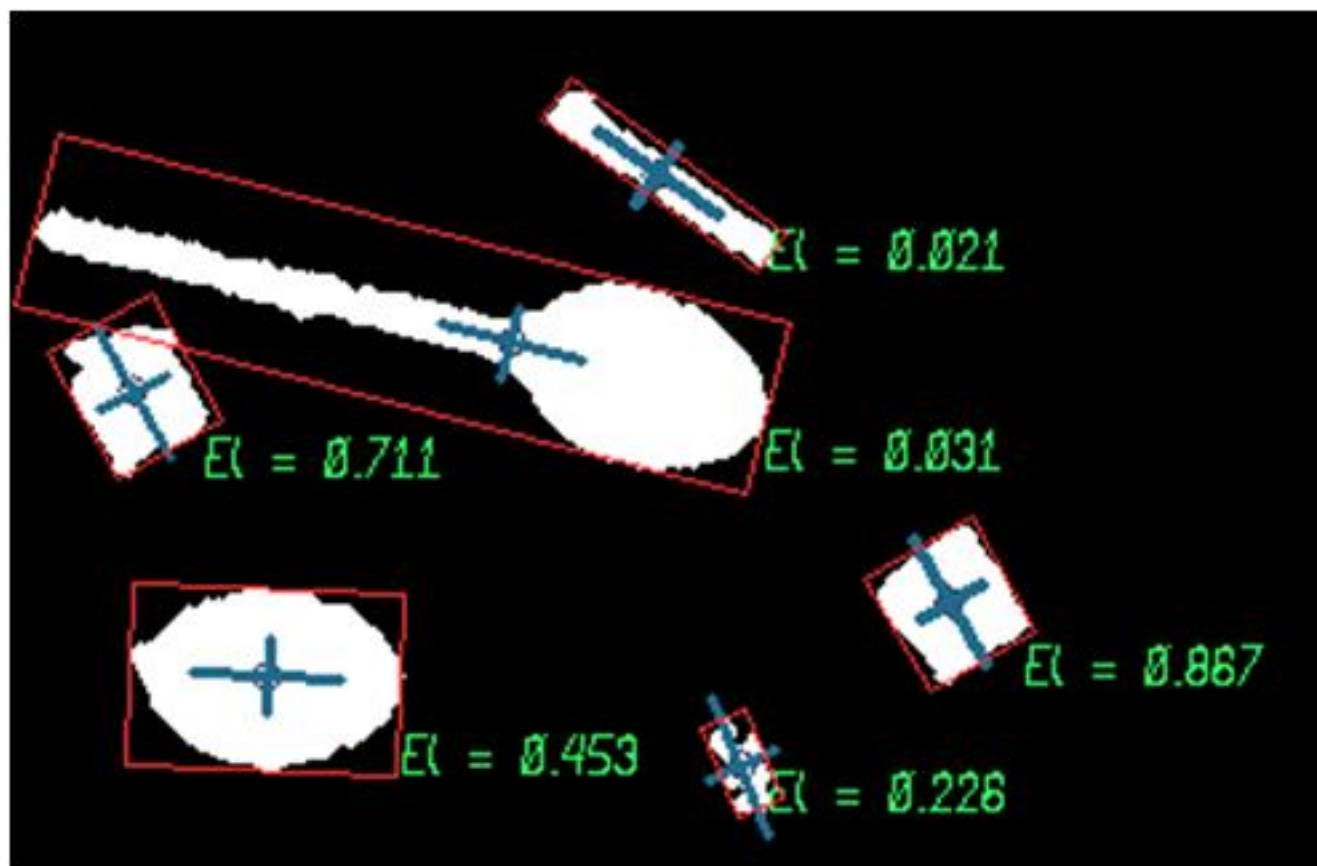
Не является инвариантной к повороту, но в ряде случаев предоставляет полезную информацию об ориентации объекта:

$$\theta = \frac{1}{2} \arctan \left(\frac{2 m_{11}}{m_{20} - m_{02}} \right)$$





Пример



Вычисленные значения признаков



Фотометрические признаки

Для каждой области можно подсчитать некий набор простейших числовых характеристик:

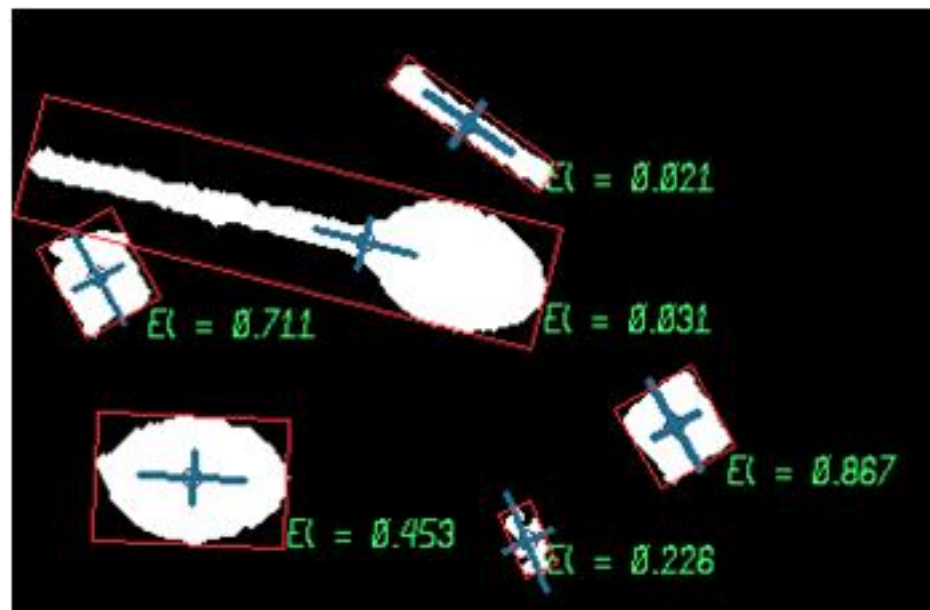
- Средняя яркость
- Средний цвет (если изображение цветное)
- Гистограмма распределения яркостей
(или три гистограммы распределения R, G, B)
- Дисперсию (разброс) яркостей или цвета
- Текстурные признаки

Разумеется, все это считается по исходному, а не бинарному изображению!



Как анализировать признаки

- Пример – ложки и сахар





Как анализировать признаки

- Как воспользоваться признаками для классификации?
 - Подобрать диапазоны значений для разных классов вручную, экспериментально
(может быть весьма трудоемко)
 - Подобрать диапазоны значений графически
(нужна база для тренировки, трудно, если признаков много)
 - Обучить классификатор с помощью машинного обучения



Ручной подбор

- Из общих соображений:
 - Ложки более вытянутые, чем сахарные кусочки
 - Ложки больше чем сахарные кусочки
 - Сахарные кусочки квадратные
 - Области появляющиеся из-за шума обычно небольшие и неквадратные
- Пытаемся сконструировать решающее правило, проверяем экспериментально
- Может быть весьма утомительно



Графический анализ

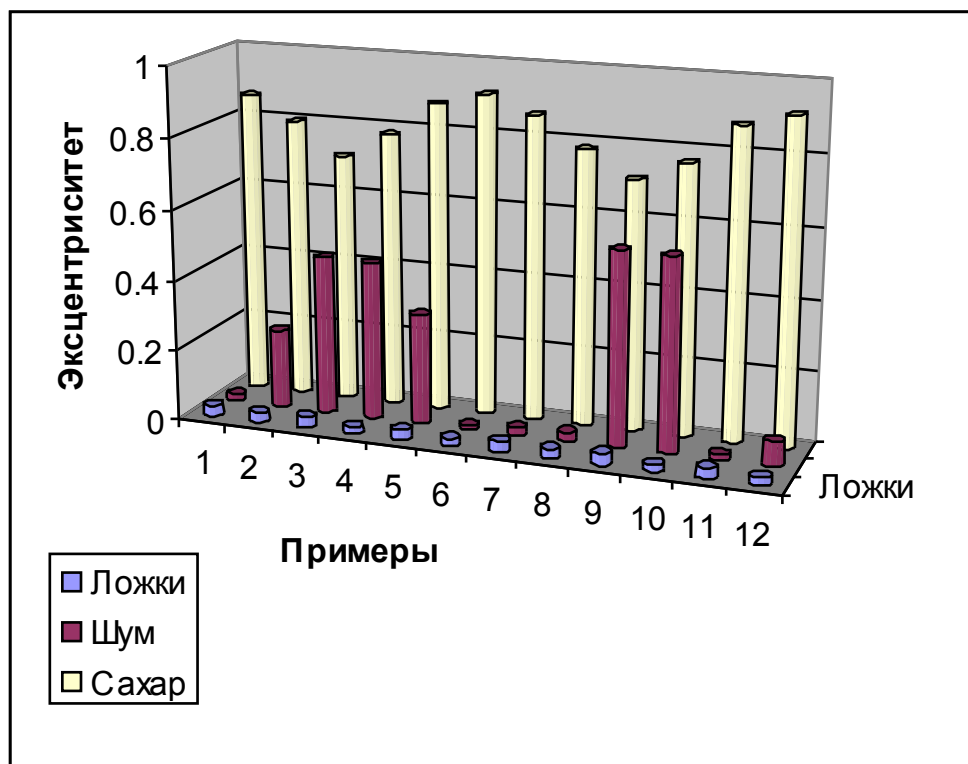
- Собрать тренировочную базу изображений
 - Где только ложки
 - Где только сахар
 - Где только шум

Как получить такие? Да просто закрасить все остальное.
- Брать признаки и строить графики



Графический анализ

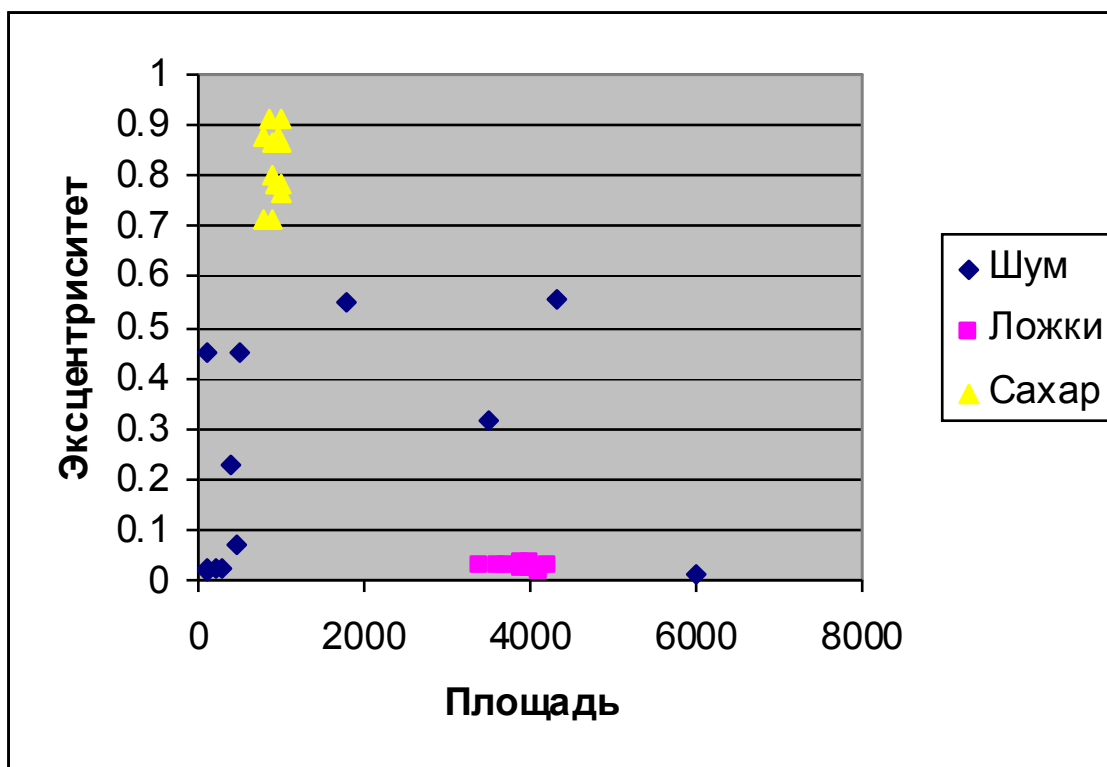
- Диаграмма распределения эксцентриситета (проблема – не получается отличить шум от ложек)





Графический анализ

- График распределения эксцентриситета и площади (гораздо лучше – можем подобрать значения порогов)



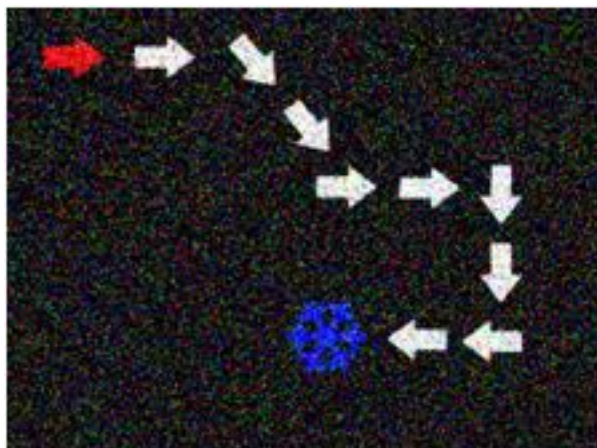


Машинное обучение

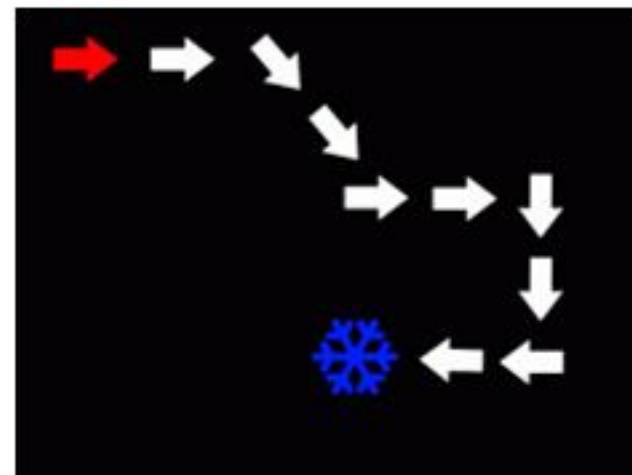
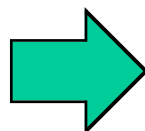
- Причина бурного развития компьютерного зрения в последние годы.
- Требуются большие коллекции примеров для обучения.
- На следующей лекции рассмотрим один из методов



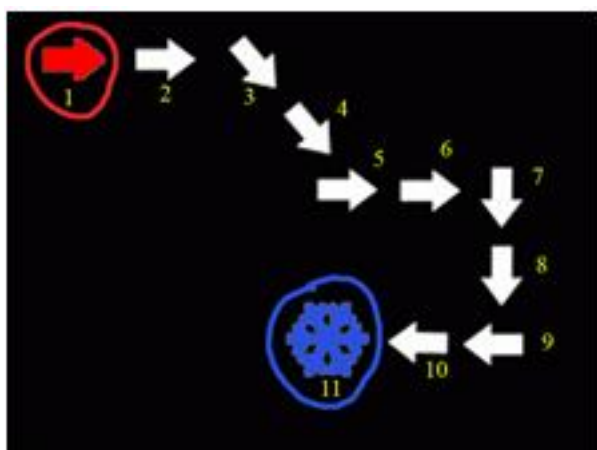
Схема простого алгоритма



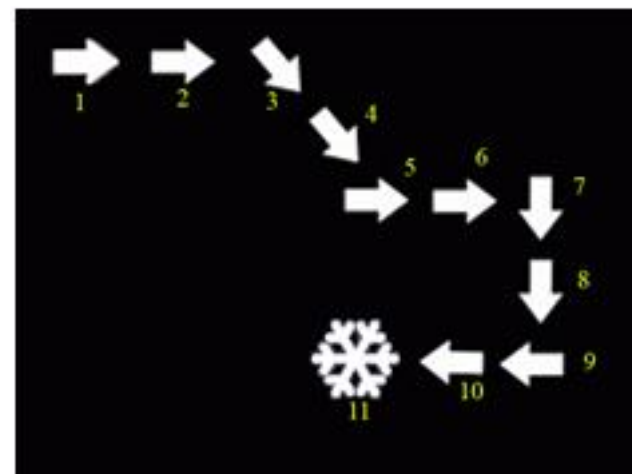
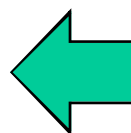
Предобработка
изображения



Сегментация
изображения



Вычисление
признаков
сегментов и
классификация





Резюме сегментации

- Сегментация изображения позволяет работать не со всем изображением в целом, а с отдельными областями
- Сегменты могут быть однородны по яркости, цвету, текстуре и по комбинации этих признаков
- В отдельных случаях мы можем решить задачу распознавания, анализируя геометрические и фотометрические признаки сегментов