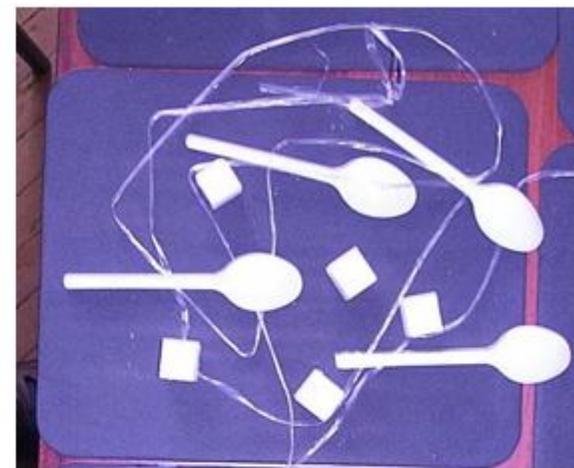
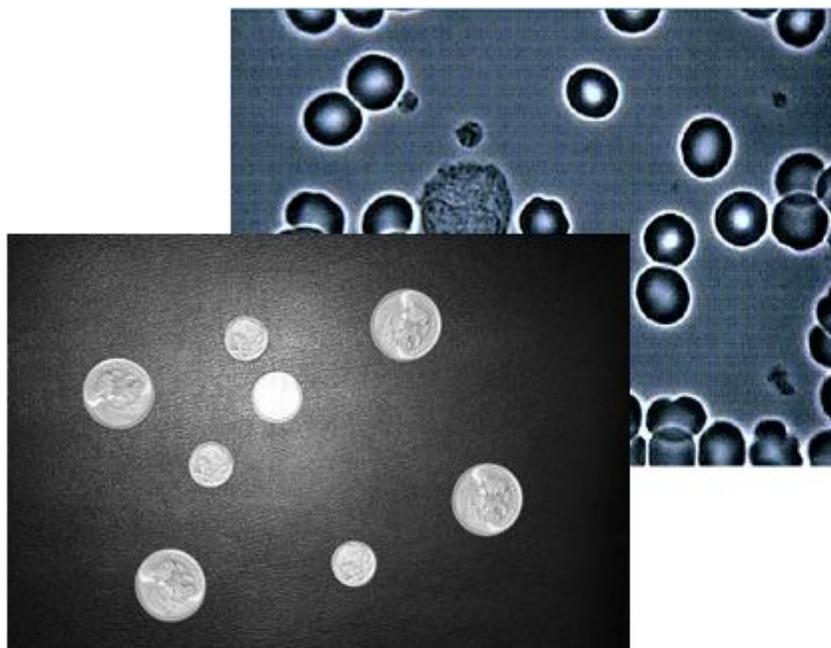
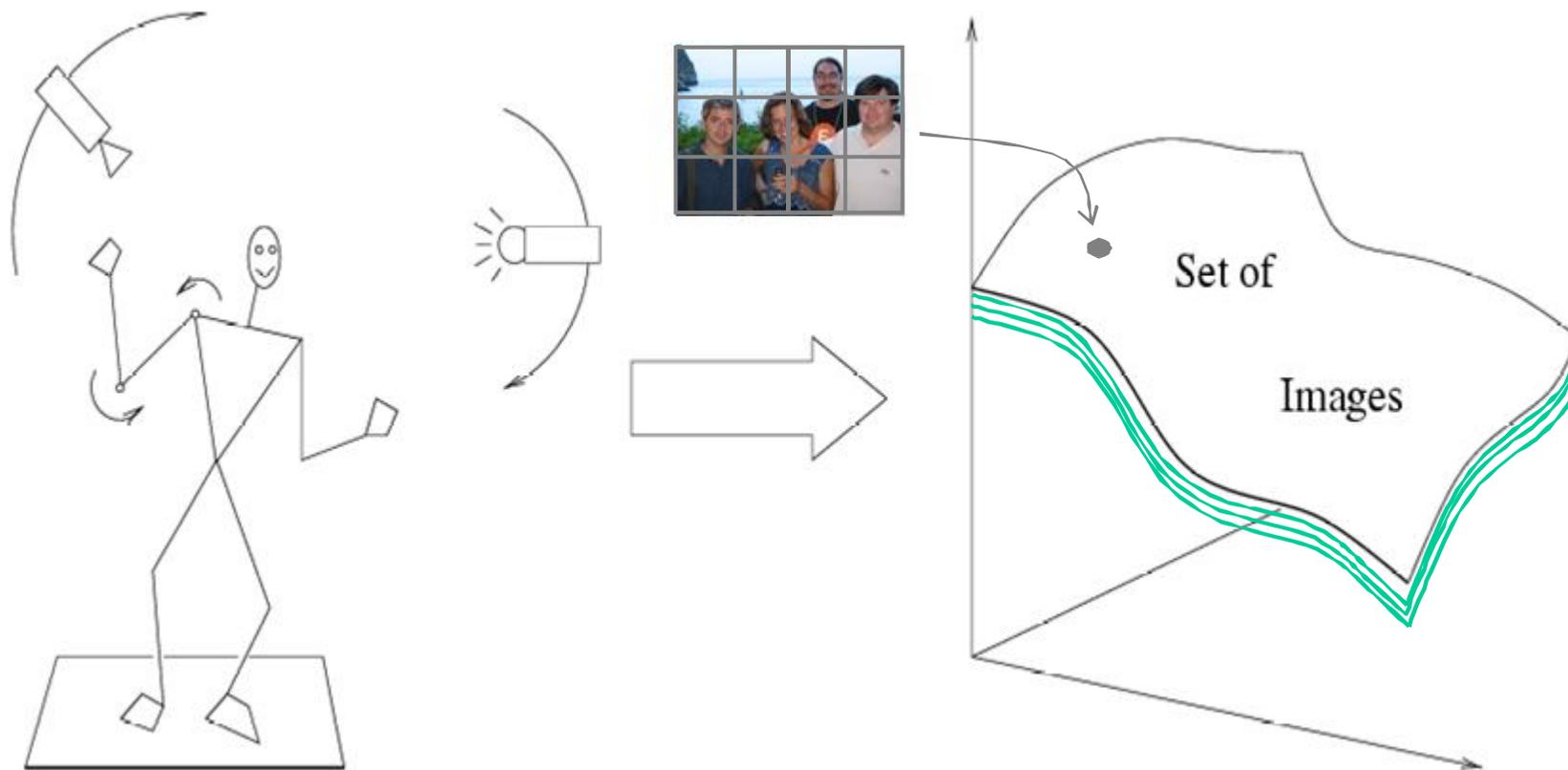


Простой анализ изображений





Изменчивость изображений



Внешние факторы:

Положение камеры
Освещение

Внутренние факторы:

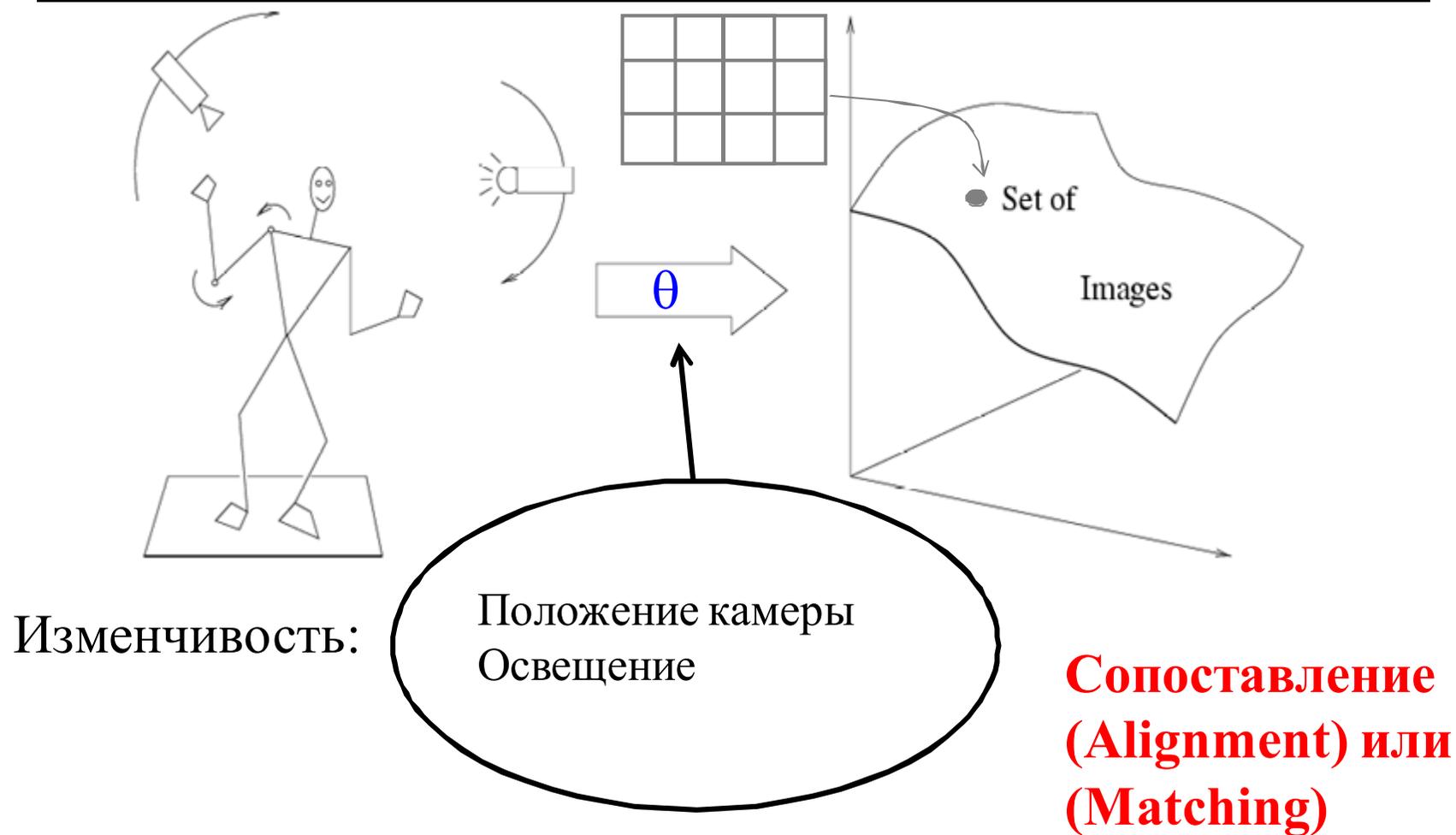
Внутриклассовая изменчивость

Внутриклассовая изменчивость





Сопоставление

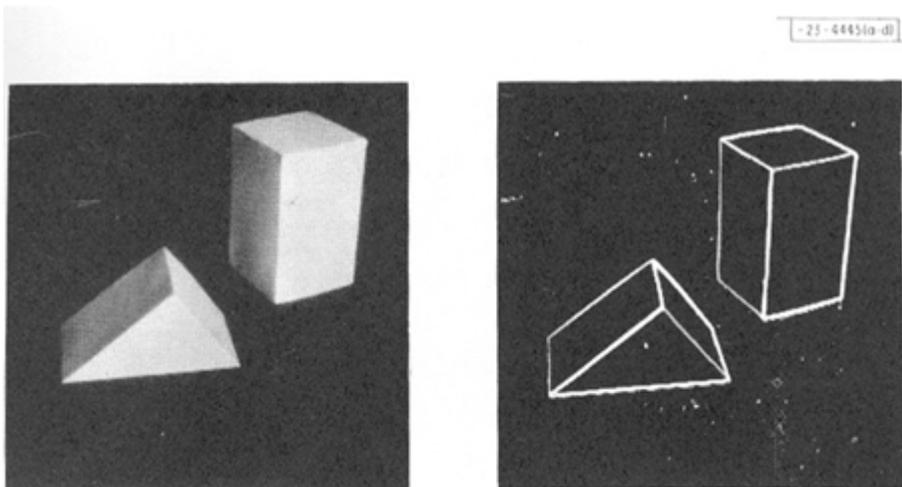


Фиксируем форму объекта

Roberts (1965); Lowe (1987); Faugeras & Hebert (1986); Grimson & Lozano-Perez (1986); Huttenlocher & Ullman (1987)

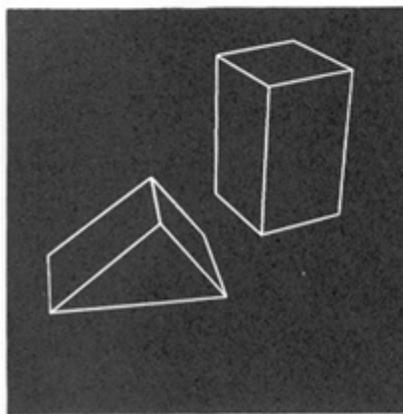


Сопоставление

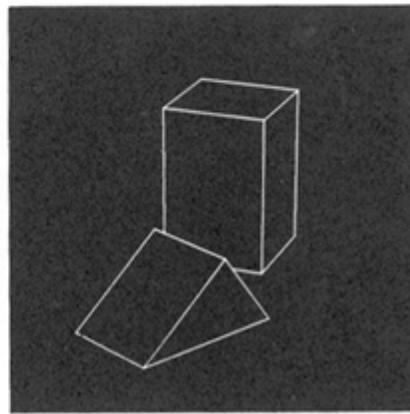


(a) Original picture.

(b) Differentiated picture.



(c) Line drawing.



(d) Rotated view.

L. G. Roberts, [*Machine Perception of Three Dimensional Solids*](#),
Ph.D. thesis, MIT Department of
Electrical Engineering, 1963.



Сопоставление шаблонов

- Фиксируем объект
- Опишем объект его изображением – шаблоном (pattern)
- Хотим найти объект в изображении
- Ограничим возможные преобразования (внешние факторы)
 - Сдвиг, размер, поворот
 - Освещение?
- Будем искать объект в изображении путём попиксельного сравнения шаблона и всех фрагментов изображения
- «Pattern matching»





Метрики

$$\sum_X \sum_Y |I_1(X, Y) - I_2(X, Y)| \quad (\text{SAD}) \text{ Sum of absolute differences}$$

$$\sum_X \sum_Y (I_1(X, Y) - I_2(X, Y))^2 \quad (\text{SSD}) \text{ Sum of squared differences}$$

$$\sum_X \sum_Y I_1(X, Y) I_2(X, Y) \quad (\text{CC}) \text{ Cross-correlation}$$

$$\frac{1}{N-1} \sum_{x,y} \frac{(I_1(x, y) - \bar{I}_1)(I_2(x, y) - \bar{I}_2)}{\sigma_1 \sigma_2} \quad (\text{NCC}) \text{ Normalized cross-correlation}$$

- SAD, SSD – минимизируются (0 – точное совпадение)
- CC, NCC – максимизируется (1 – точное совпадение)

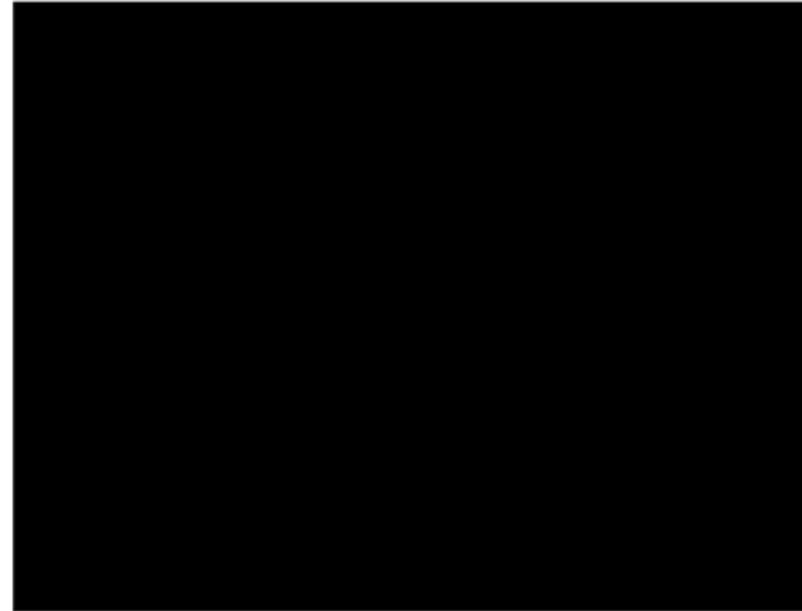


Пример: пульт ТВ



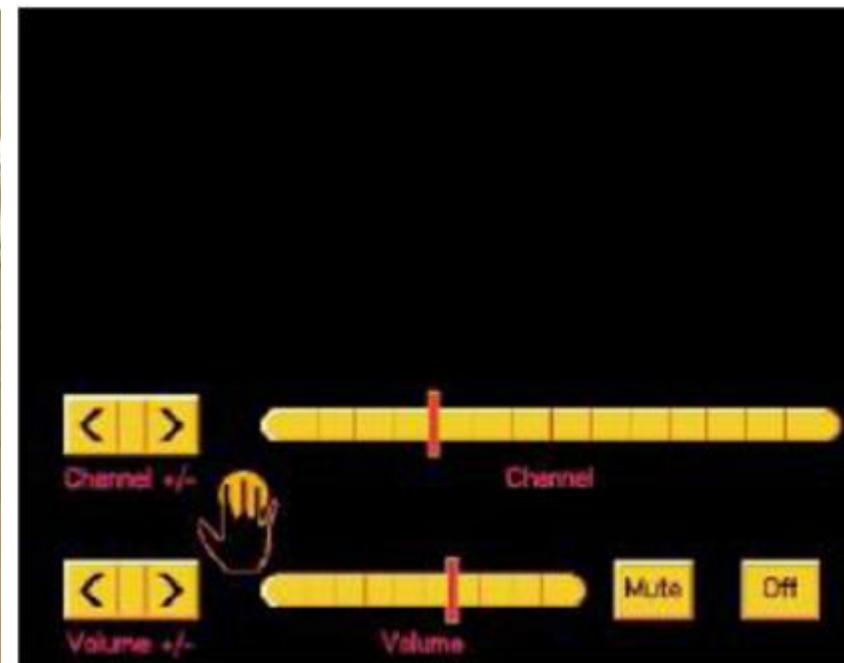
- Шаблон (слева), изображение (в центре), карта нормализованной корреляции (справа)
- Пик яркости (максимум корреляции) соответствует положению руки (искомого шаблона)

Пример: пульт ТВ

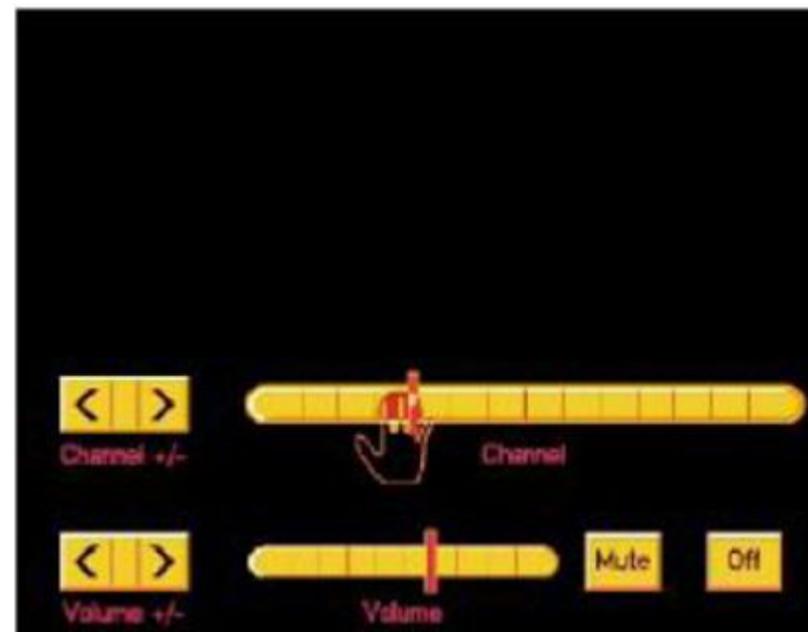


Credit: W. Freeman *et al*, "Computer Vision for Interactive Computer Graphics," *IEEE Computer Graphics and Applications*, 1998

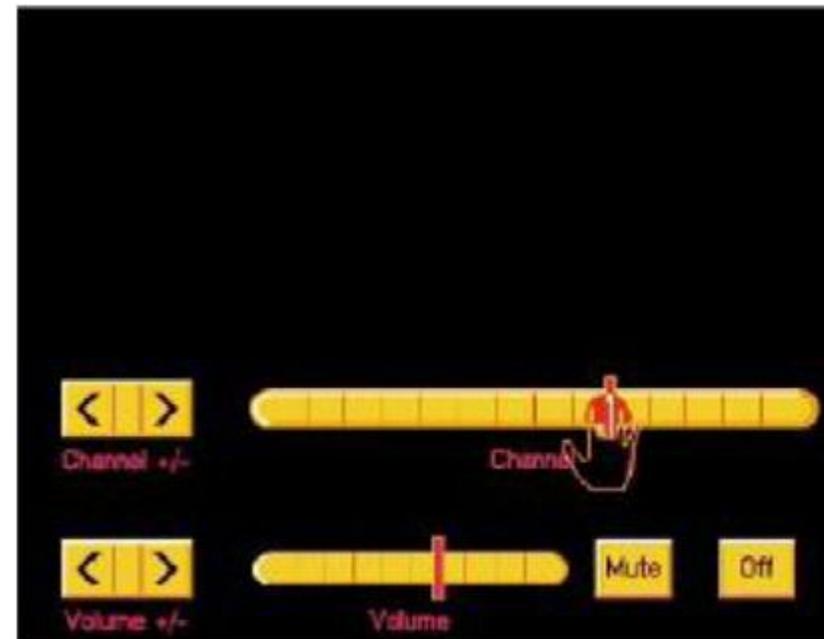
Пример: пульт ТВ



Пример: пульт ТВ



Пример: пульт ТВ





Ограничения и проблемы

- Ищем конкретный объект, а не класс / категорию объектов
 - Не «символ», а конкретную букву в конкретном шрифте
- Трудоёмкость
 - Полный перебор параметров
- Модель преобразования
 - В простом варианте неизвестно только положение, размер и ориентация фиксированы
 - Чтобы учесть поворот и ориентацию придётся перебрать все возможные параметры
- Шаблонов может быть много
 - OCR – распознавание символов
 - По шаблону на каждый символ

Как улучшить подход?



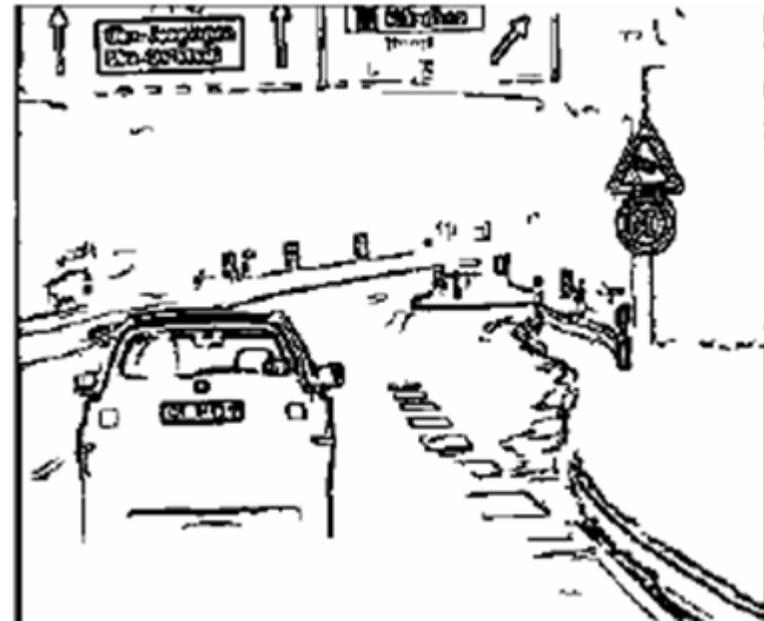
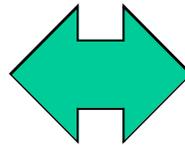
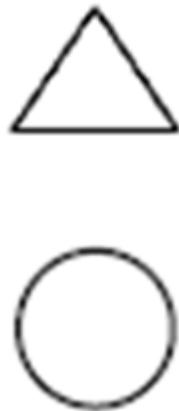


Выделение краев

- Интуитивно понятно, что основная информация в картинке содержится в контрастных «краях»
 - Компактное представление
 - Соответствует устройству мозга
- **Задача:** Выделить резкие изменения (разрывы) изображения
- **Идеал:** рисунок художника (но артист уже пользуются своими знаниями об объектах)



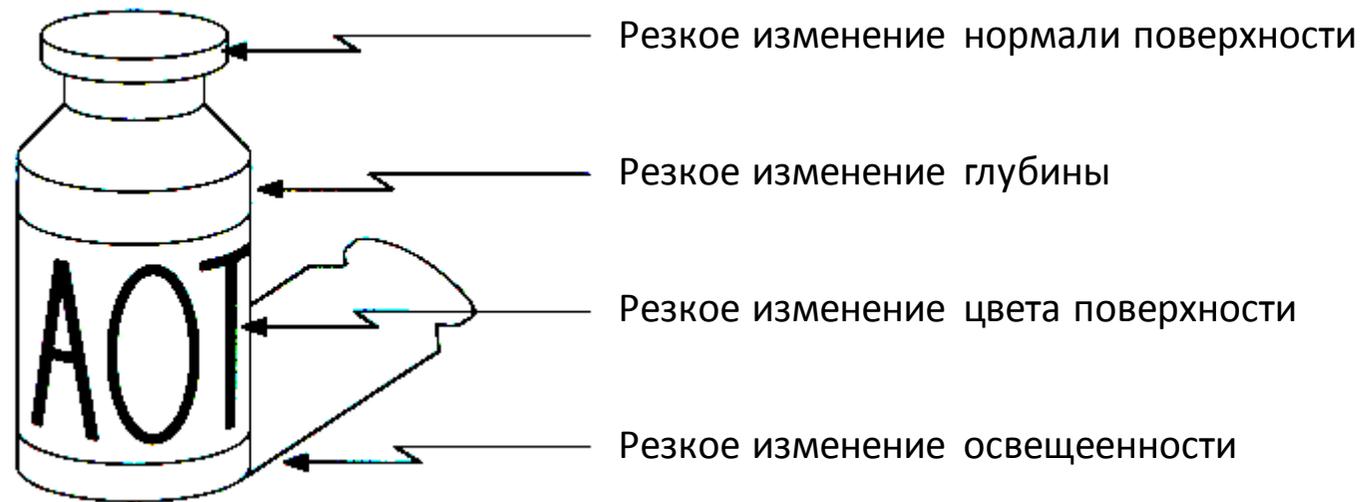
Края для сопоставления шаблонов



- Будем учитывать только часть информации, но самую важную, для распознавания шаблонов
- Даже улучшим обобщающую способность



Откуда берутся границы



Резкое изменение = «разрыв»

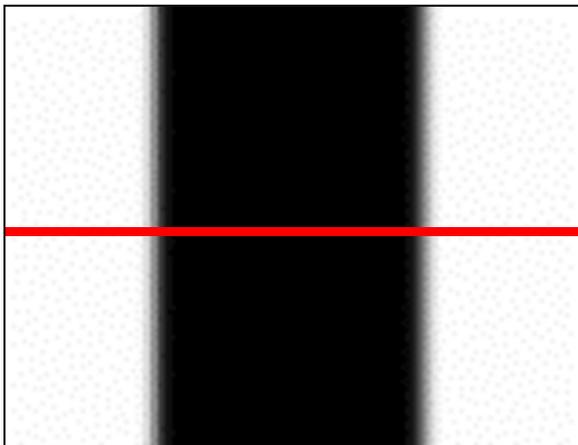
- Существует множество причин формирования границ на изображении



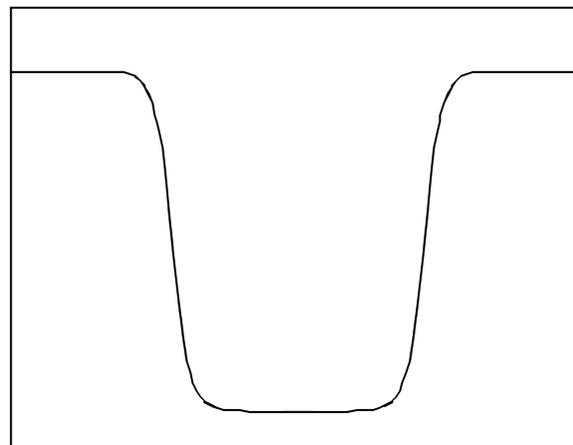
Описание «края»

- Край – это точка резкого изменения значений функции интенсивности изображения

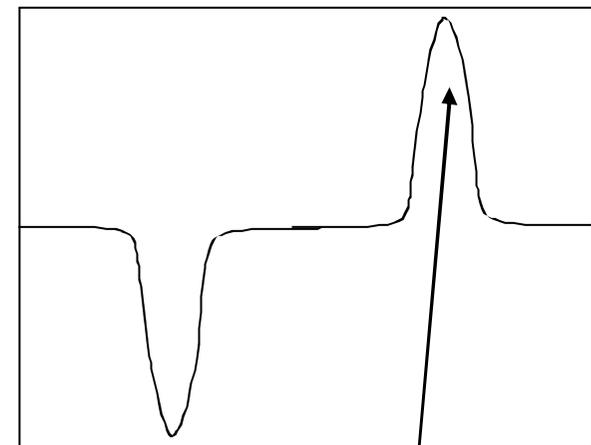
изображение



Функция интенсивности
(строка изображения)



1ая производная



Края соответствуют
экстремумам производной



Градиент изображения

- Градиент изображения: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$

- $\nabla f = \left[\frac{\partial f}{\partial x}, 0 \right]$ $\nabla f = \left[0, \frac{\partial f}{\partial y} \right]$ $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$

Градиент направлен в сторону наибольшего изменения интенсивности

Направления градиента задается как: $\theta = \tan^{-1} \left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$

- Как направление градиента соответствует направлению края?
- *Сила края* задается величиной (нормой) градиента:

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$



Дифференцирование и свёртка

• Для функции 2х переменных,
 $f(x,y)$:

$$\frac{\partial f}{\partial x} = \lim_{\varepsilon \rightarrow 0} \left(\frac{f(x + \varepsilon, y) - f(x, y)}{\varepsilon} \right)$$

• Линейная и инвариантная к
переносу, поэтому м.б.
Результатом свертки

• Разностная производная:

$$\frac{\partial f}{\partial x} \approx \frac{f(x_{n+1}, y) - f(x_n, y)}{\Delta x}$$

• Свёртка!

-1	1
----	---



Вычисление градиента

Семейство методов основано на приближенном вычислении градиента, анализе его направления и абсолютной величины. Свертка по функциям:

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Робертса

Превитт

Собея

Математический смысл – приближенное вычисление производных по направлению



Примеры карты силы краев

Примеры:



Робертса



Превитт

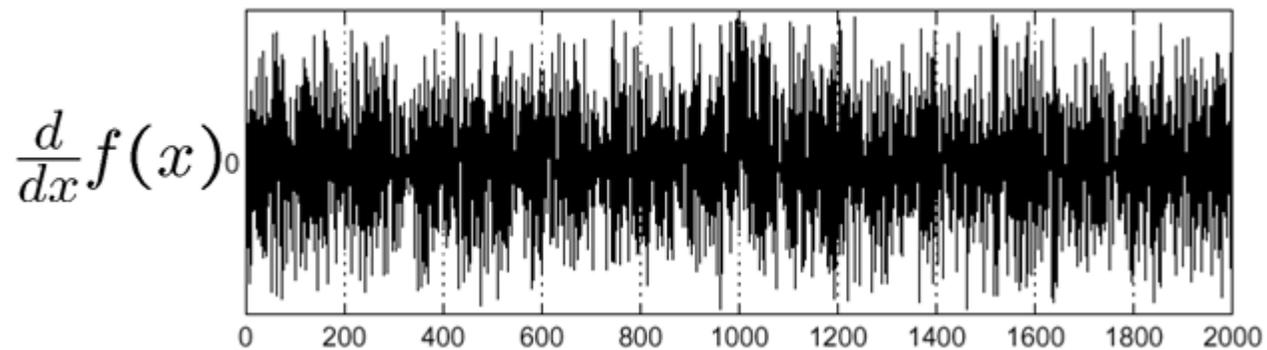
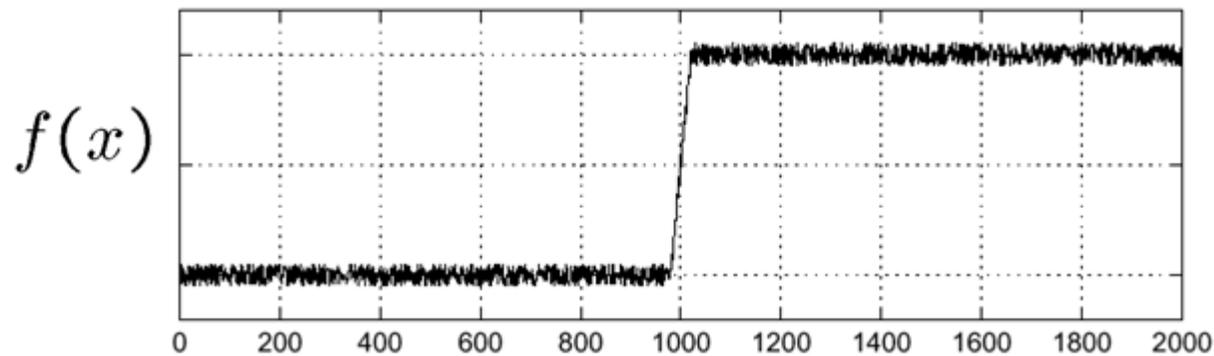


Собея



Влияние шума

- Рассмотрим строку или столбец изображения
 - Интенсивность от положения можно рассматривать как сигнал



Край исчез

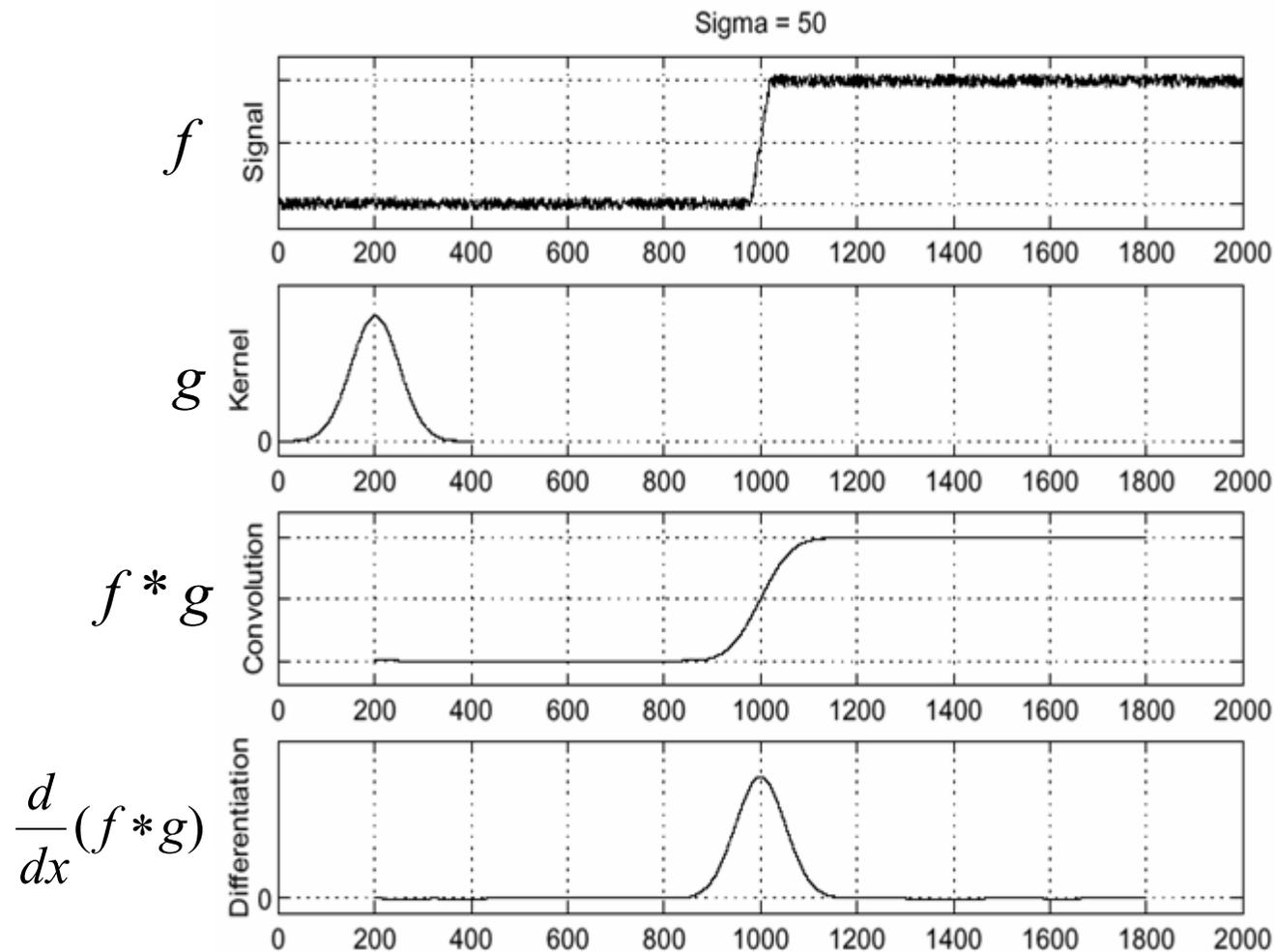


Влияние шума

- Разностные производные очень чувствительны к шуму
 - Зашумленные пиксели отличаются от соседей
 - Чем сильнее шум, тем выше отклик
- Сглаживание
 - Сглаживание делает все пиксели (зашумленные?) чуть более похожими на соседей



Предобработка (сглаживание)

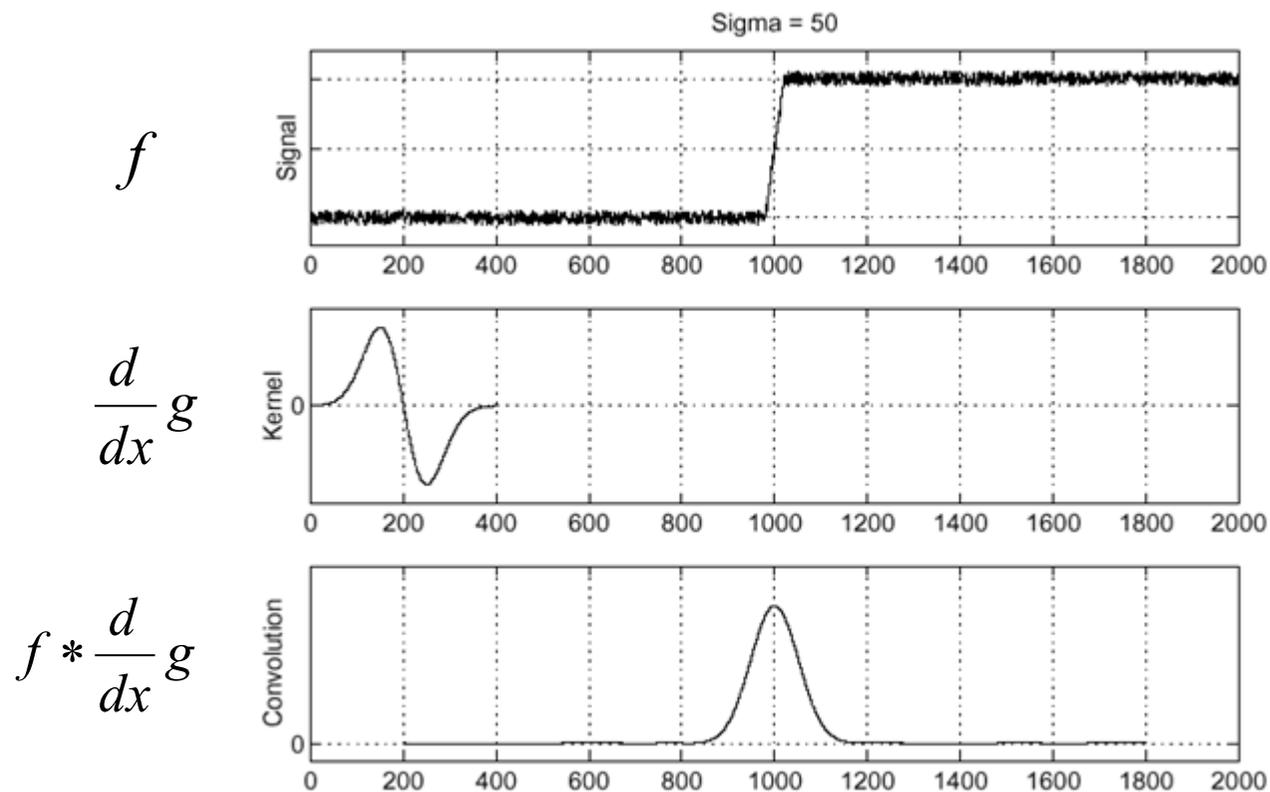


- Для поиска краев ищем пики в: $\frac{d}{dx}(f * g)$



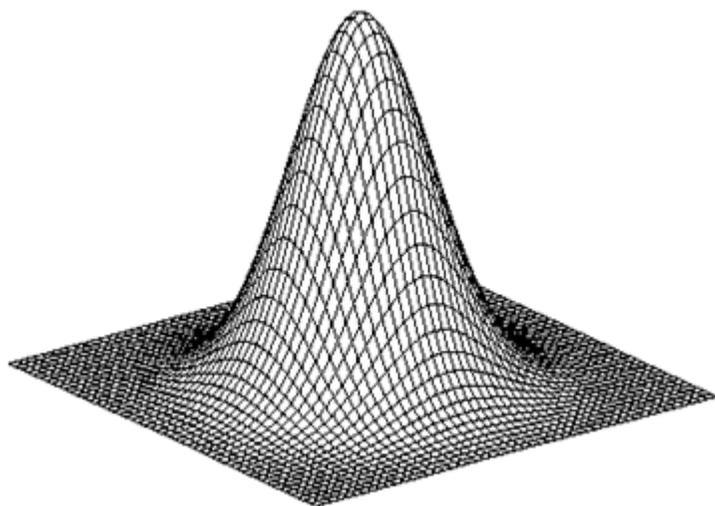
Свойства свертки

- Операции свертки и дифференцирования ассоциативны: $\frac{d}{dx}(f * g) = f * \frac{d}{dx}g$
- Это экономит 1 операцию:

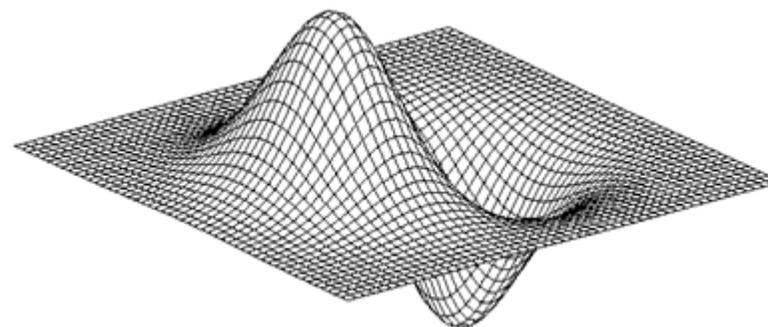




Производная фильтра гаусса

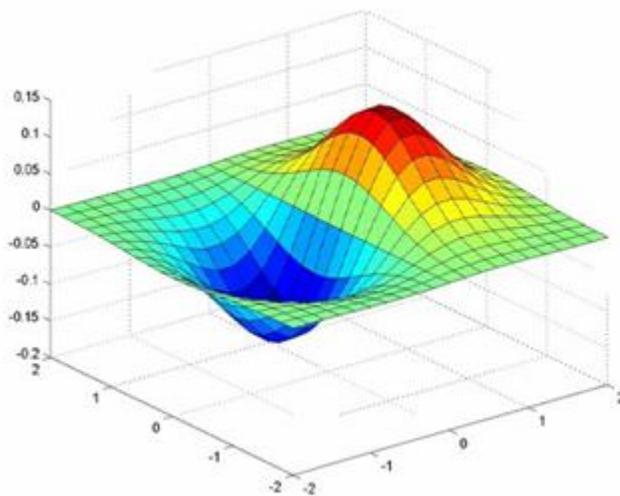


$$* [1 \ -1] =$$

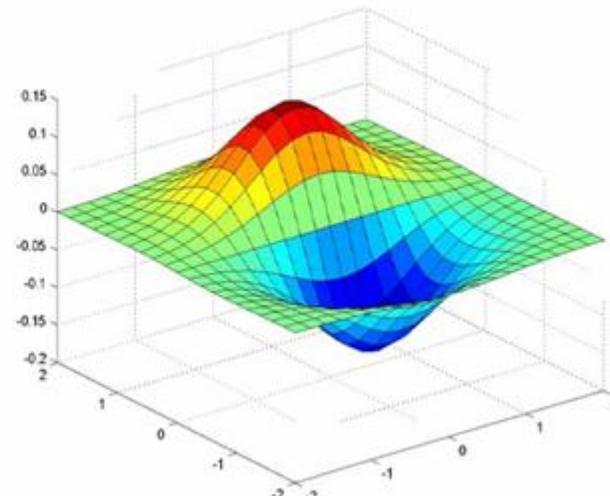
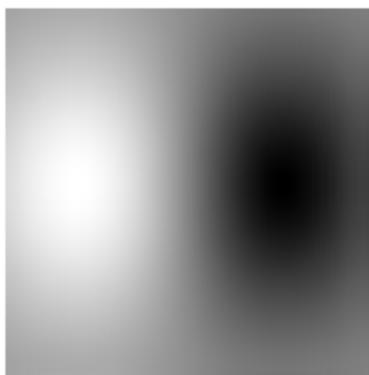




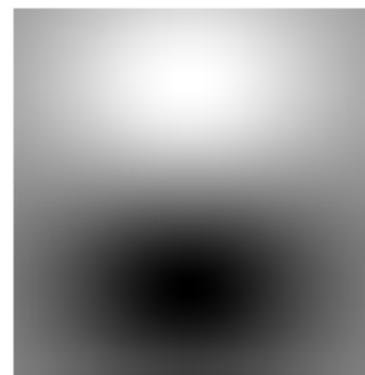
Производная фильтра гаусса



x-direction

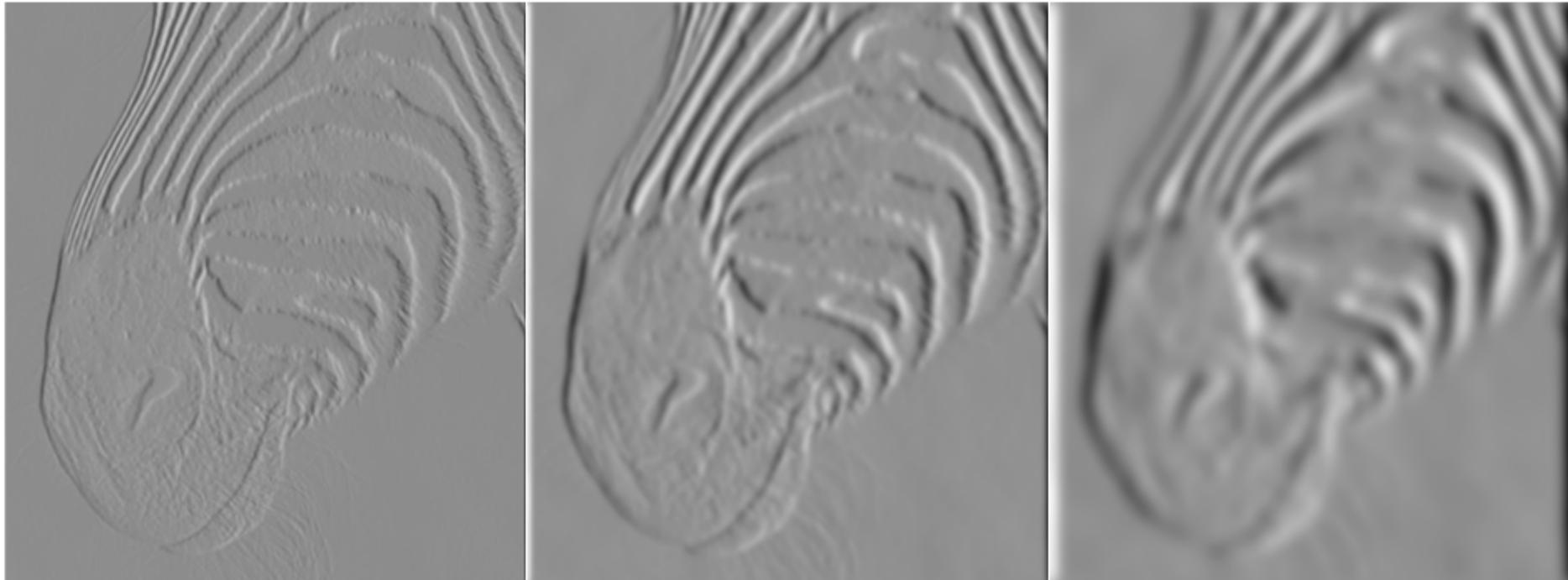


y-direction





Поиск баланса



1 pixel

3 pixels

7 pixels

- Сглаженные производные подавляют шум, но размывают края. Плюс края находятся на разных «масштабах»



Выделение краев

- Вычисление градиента – это еще не всё...



Исходное изображение



Карта силы краев

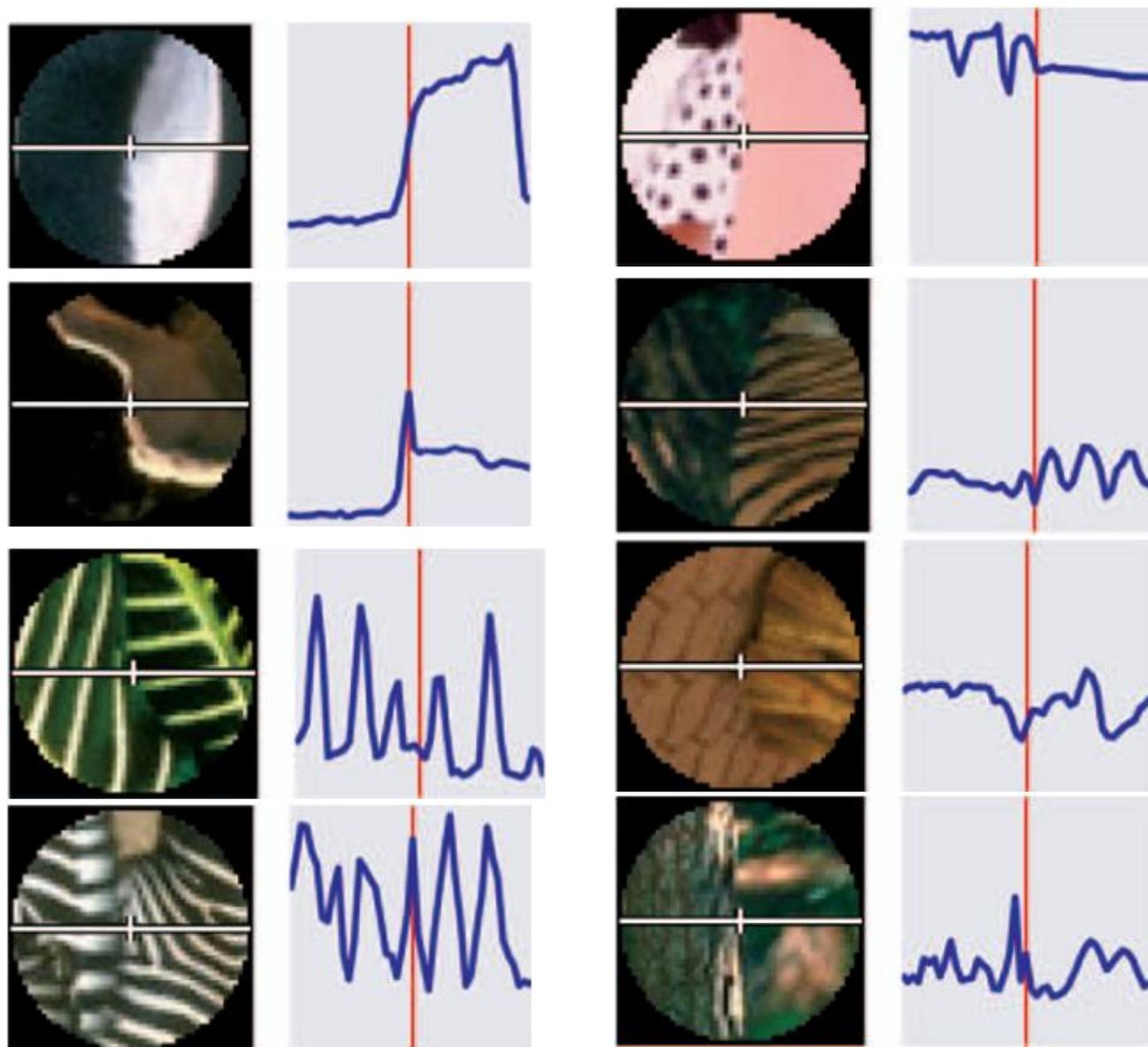


Обработанная карта

- После выделения края с помощью фильтром не хватает точности, края «толстые» и размытые
- Существуют алгоритмы для обработки карт градиентов, например, детектор Кэнни (Canny Edge Detector)



Ограничения градиента яркости



Source: Martin et al. 2003

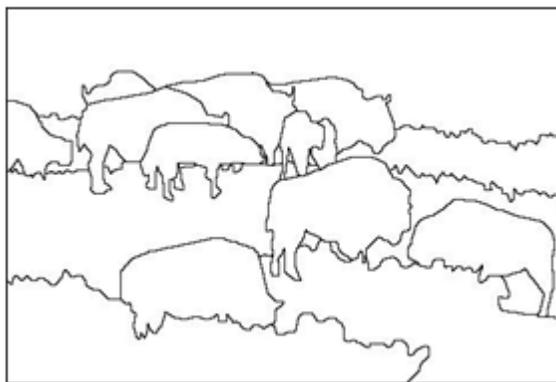


Поиск краев – это только начало...

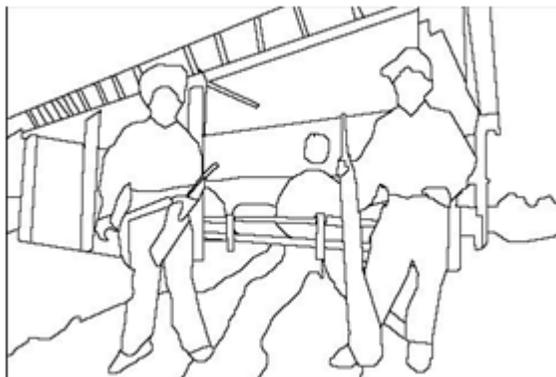
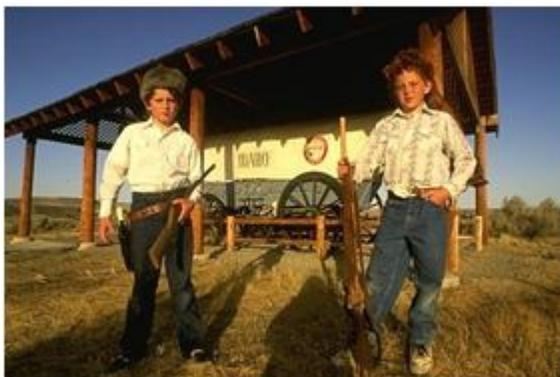
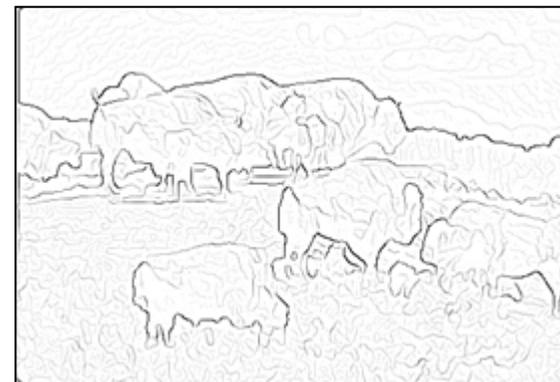
image



human segmentation



gradient magnitude

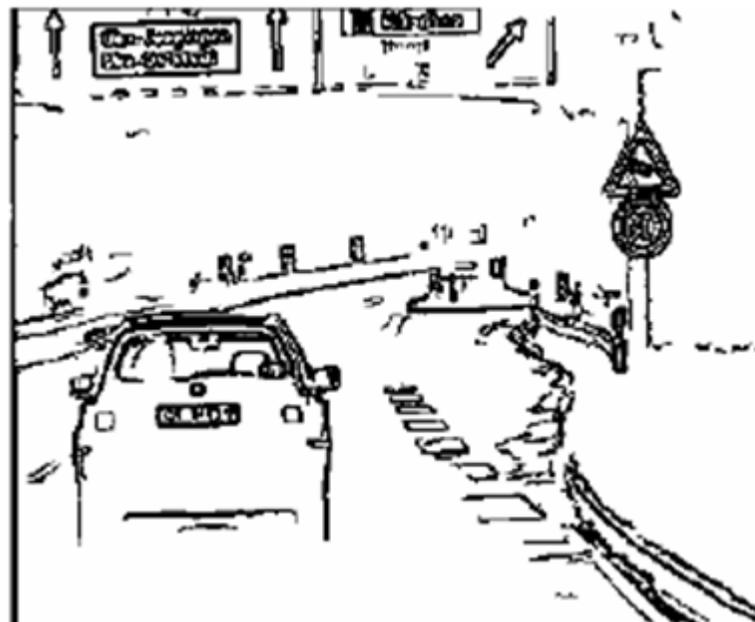
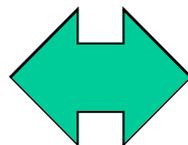


- Berkeley segmentation database:

<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>



Края для сопоставления шаблонов



- Получили карту краёв шаблона и изображения
- Как их сравнить друг с другом?
 - Просто попиксельно явно не оптимально



Метрики

• Chamfer Distance

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

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

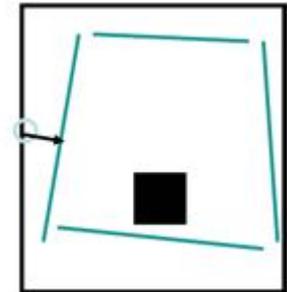
- Суммируем все найденные расстояния

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

• Hausdorff Distance

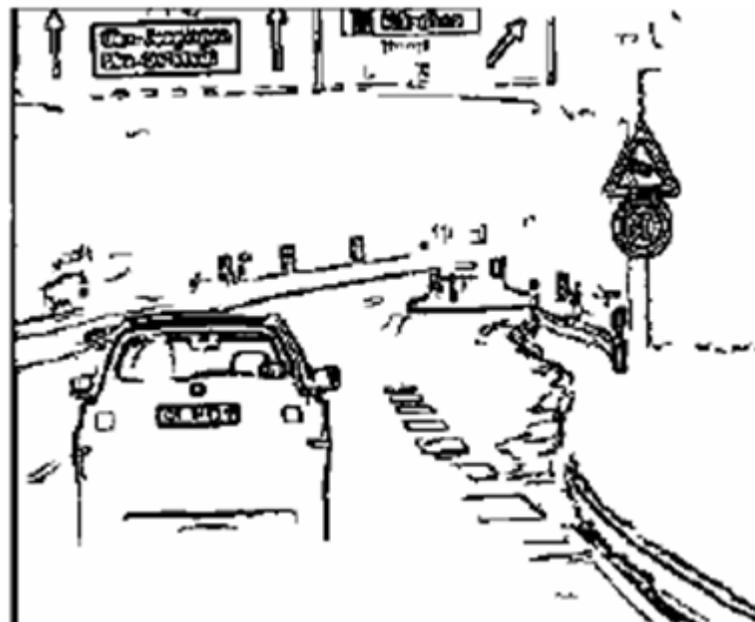
- Почти то же самое, но берём не сумму, а максимальное расстояния

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





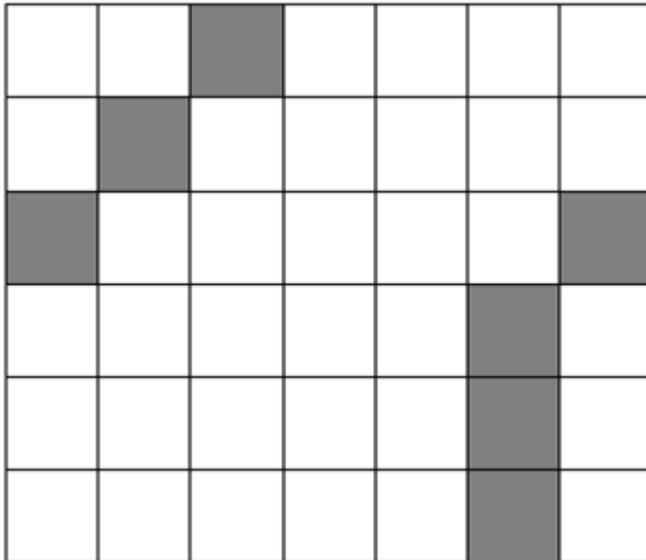
Поиск ближайших пикселей края



Вопрос: как найти ближайший пиксель края на изображении?



Distance Transform



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



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

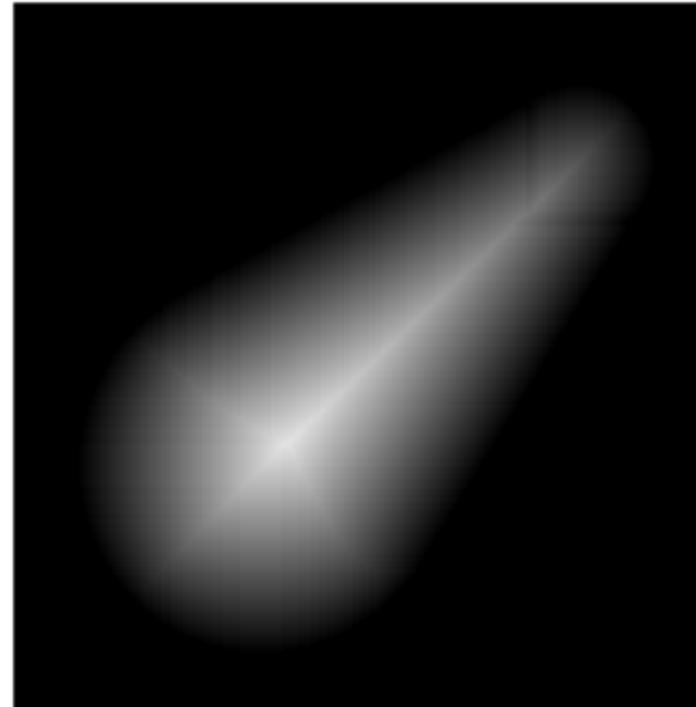
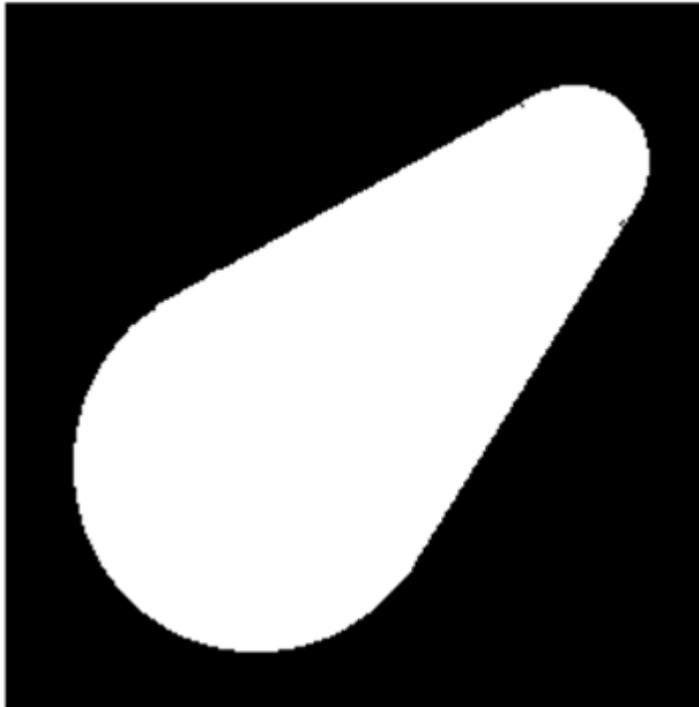
- Совмещаем шаблон и карту DT
- Вычисляем ошибку, суммируя все значения в пикселях краев



Вычисление DT

- Простейший алгоритм – N проходов
 - Первый проход помечает края 0
 - На втором помечаем все граничащие с 0 пиксели как 1
 - И т.д.
- Существует двухпроходный алгоритм

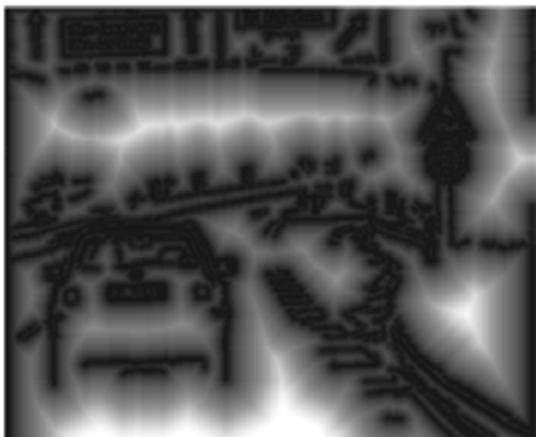
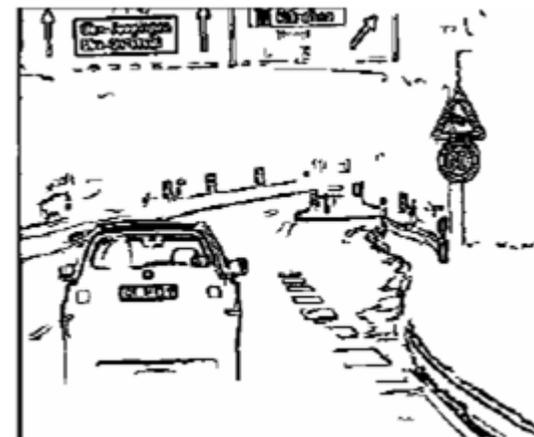
Пример DT



DT может использоваться для поиска «скелета»
– осей объекта



Пример поиска с помощью DT



Пример





Резюме сопоставления шаблонов

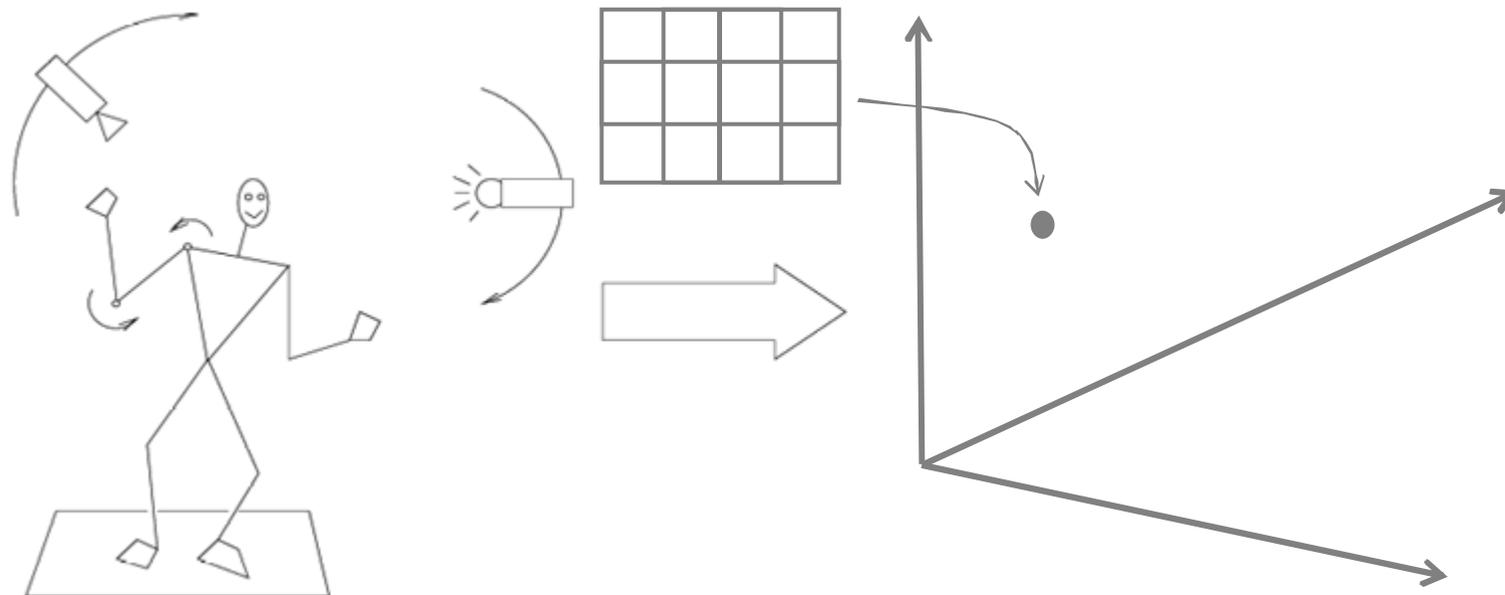
- Подходит в тех случаях, когда объекты фиксированы и модель преобразования не очень сложная
 - Цифры на знаках
 - Цифры на конвертах
 - Аэрофотосъёмка / Космическая съёмка
- Не очень быстрые методы
 - Требуются специальные процедуры для ускорения, пр. отбраковка ложных фрагментов по упрощённым критериям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



Номера



Инвариантность



~~Изменчивость~~

Инвариантность

к:

Положение камеры

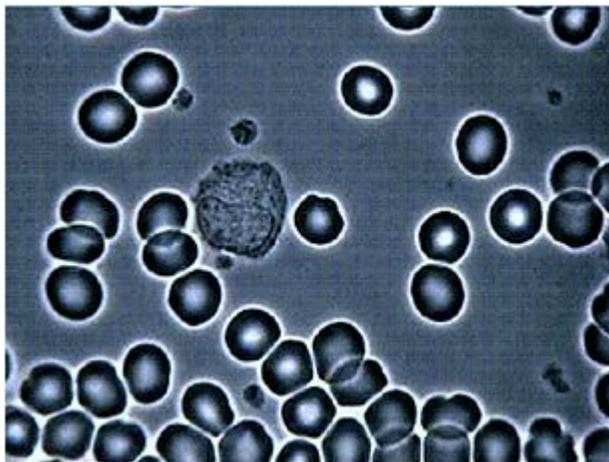
Освещение

Внутренние параметры

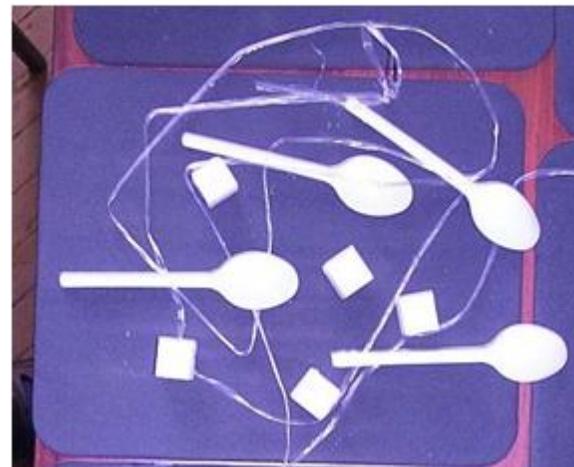
Duda & Hart (1972); Weiss (1987); Mundy et al. (1992-94);
Rothwell et al. (1992); Burns et al. (1993)



Примеры



Клетки крови



Ложки и сахар



Монеты и купюры



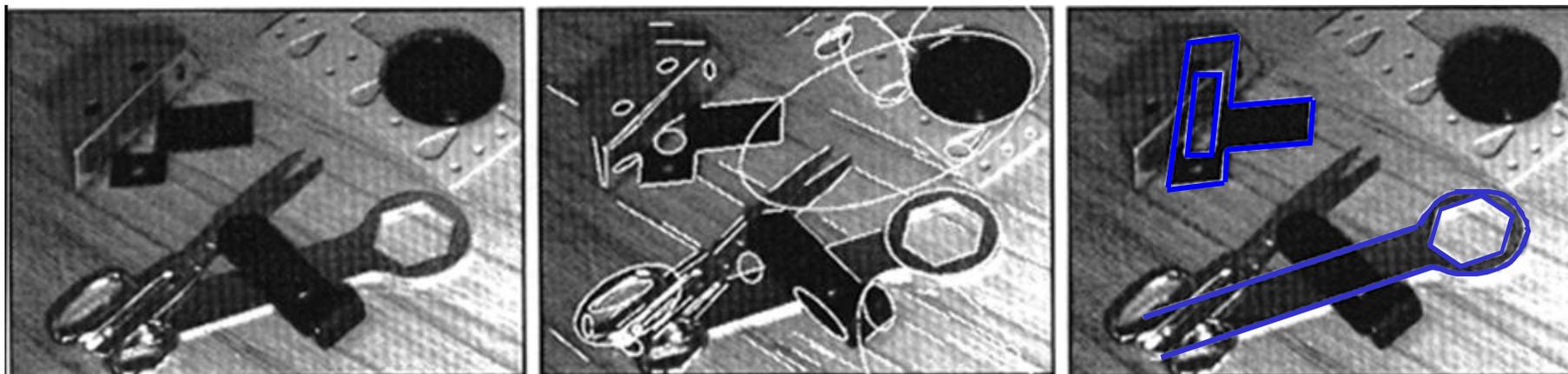
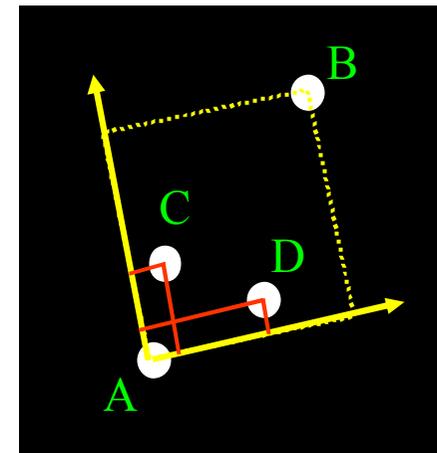
Номера

Контрастные объекты на фоне!



Более сложные примеры

Инвариантность к
перспективным искажениям –
проективные инварианты
(Rothwell et al., 1992)



В общем случае, для 3D объектов не существует проективных инвариантов (Burns et al., 1993)



Схема простого алгоритма

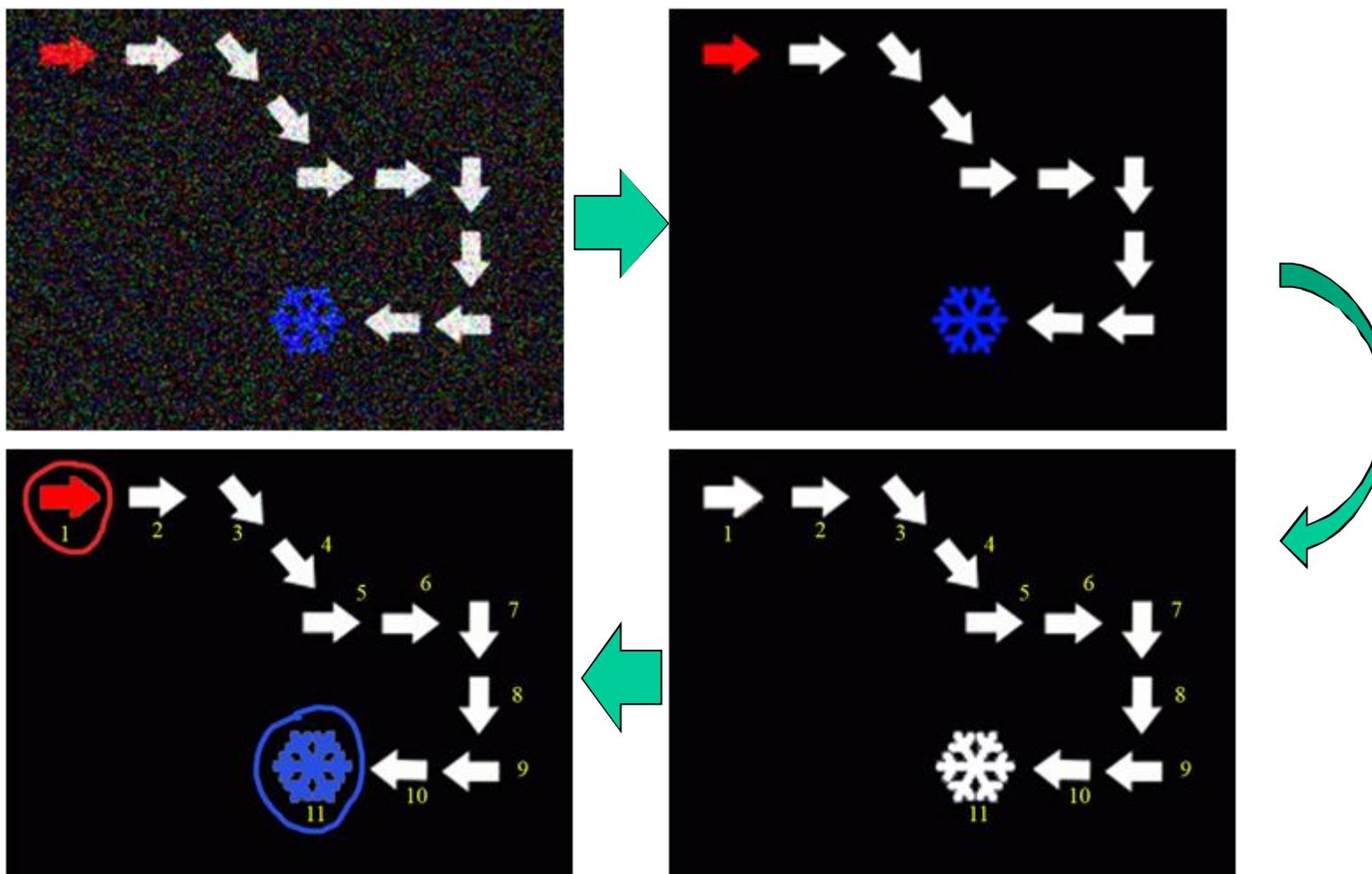




Схема простого алгоритма

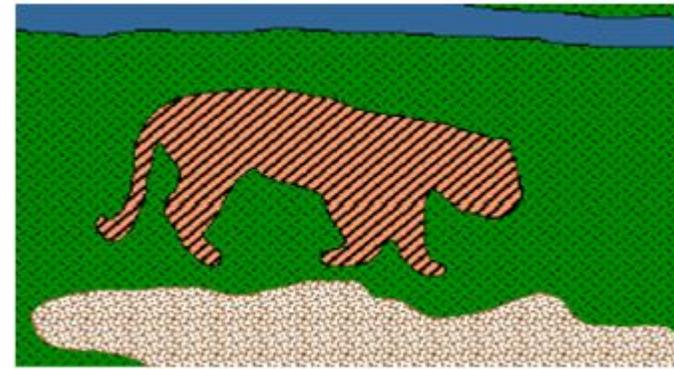
- Предобработка изображения для упрощения анализа (например – шумоподавление)
- Выделение на изображении областей-кандидатов в которых может находиться искомый объект
- Вычисление признаков (инвариантов) по выделенным фрагментам
- Проверка – является ли фрагмент изображения изображением нужного нам объекта по измеренным параметрам



Сегментация изображений

Разбиение изображения на неперекрывающиеся области, покрывающие все изображение и однородные по некоторому признаку:

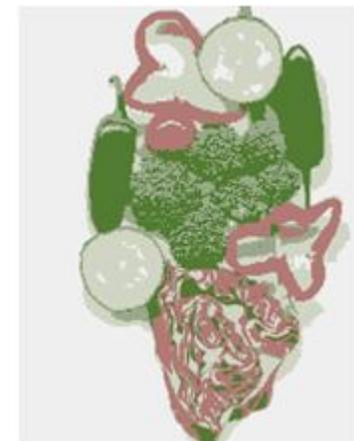
- Яркости
- Цвету
- Текстуре





Алгоритмы к рассмотрению

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





Алгоритмы к рассмотрению

- Бинаризация и выделения связанных компонент



- Последовательное сканирование

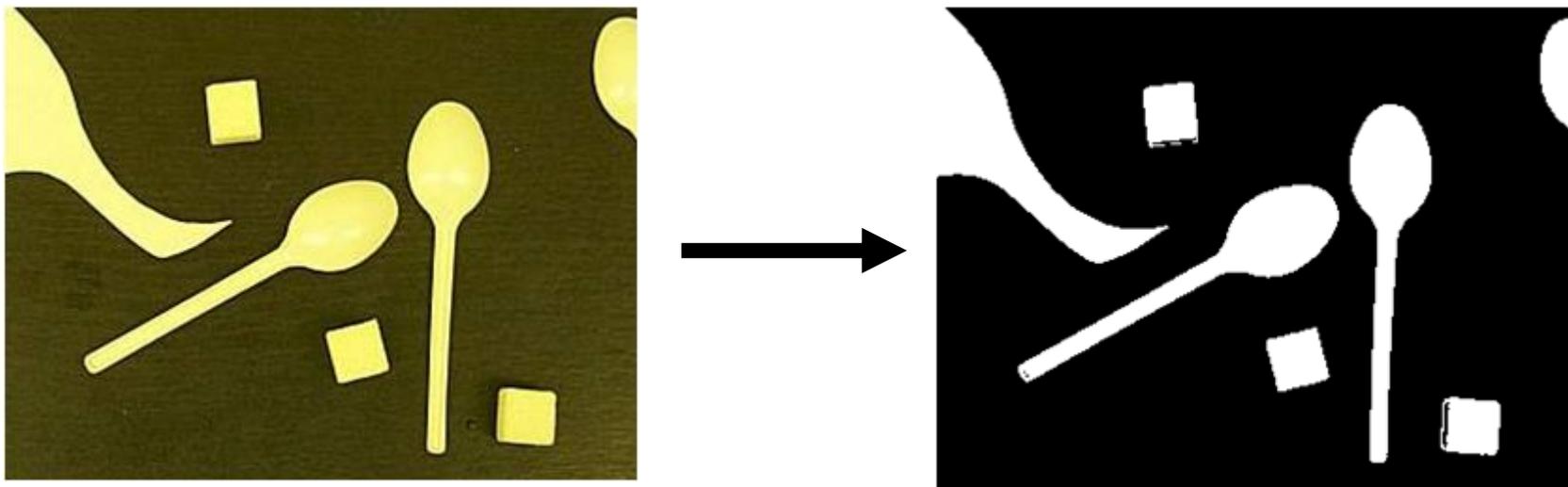


- Метод К-средних





Бинаризация изображений



- Пиксель бинарного изображения может принимать только значения 0 и 1
- Бинаризация – построение бинарного изображения по полутоновому / цветному
- Смысл?
 - Разделить изображение на фон и контрастные объекты
 - Объекты помечены 1, фон 0



Пороговая фильтрация

- Простейший вариант - пороговая фильтрация (thresholding)
 - Выделение областей, яркость которых выше/ниже некоторого порога, заданного «извне»

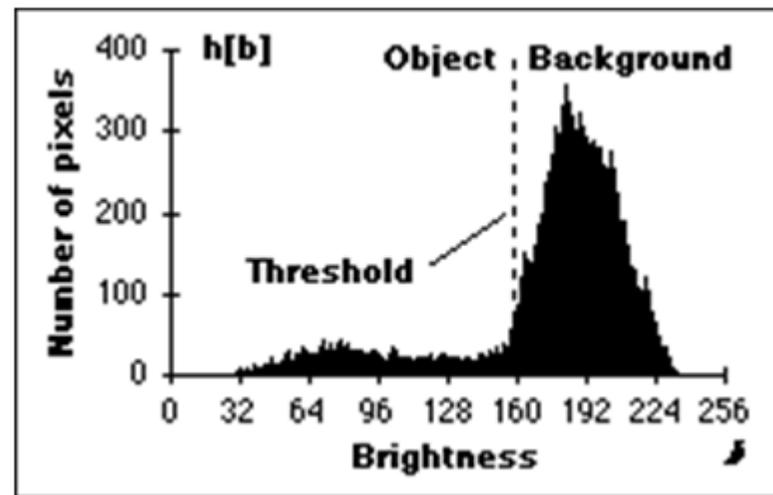
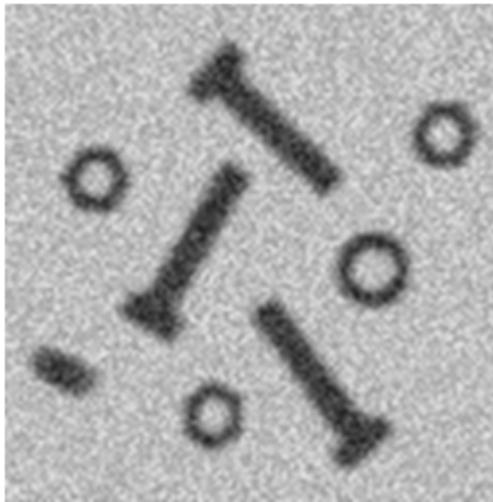




Пороговая фильтрация

Более интересный способ – определение порога автоматически, по характеристикам изображения

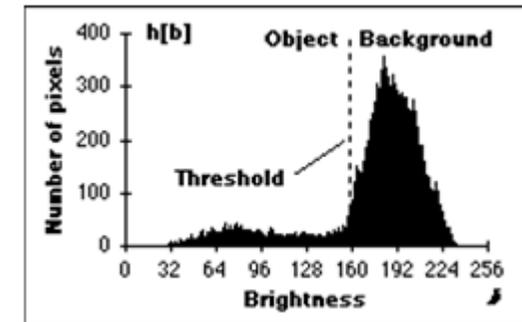
- Анализ гистограммы



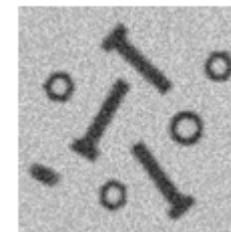


Анализ гистограммы

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



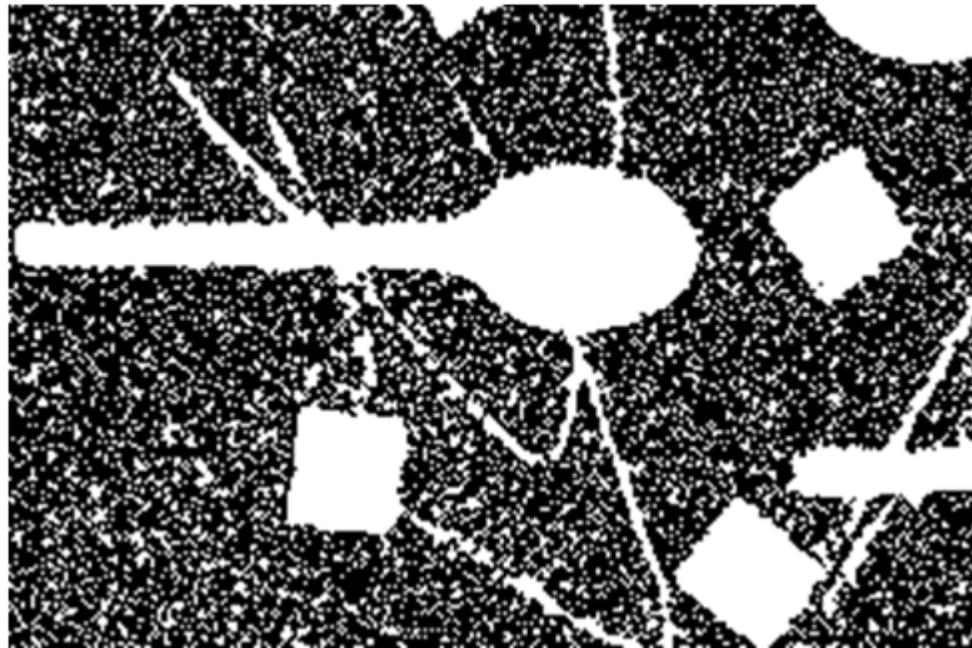
1. Сгладить гистограмму;
2. Найти ячейку гистограммы h_{\max} с максимальным значением;
3. На стороне гистограммы не относящейся к объекту (на примере – справа от пика фона) найти яркость h_p , количество пикселей с яркостью $\geq h_p$ равняется $p\%$ (например 5%) от пикселей яркости которых $\geq h_{\max}$;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





Шум в бинарных изображениях

Пример бинарного изображения с сильным шумом

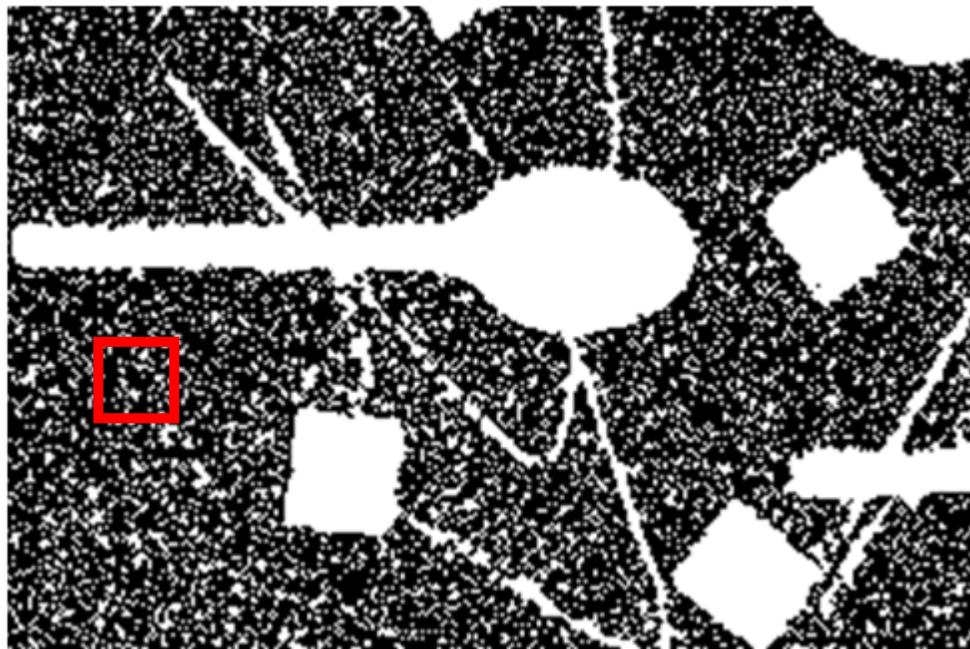


Часто возникает из-за невозможности полностью подавить шум в изображениях, недостаточной контрастности объектов и т.д.



Шум в бинарных изображениях

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





Подавление и устранение шума

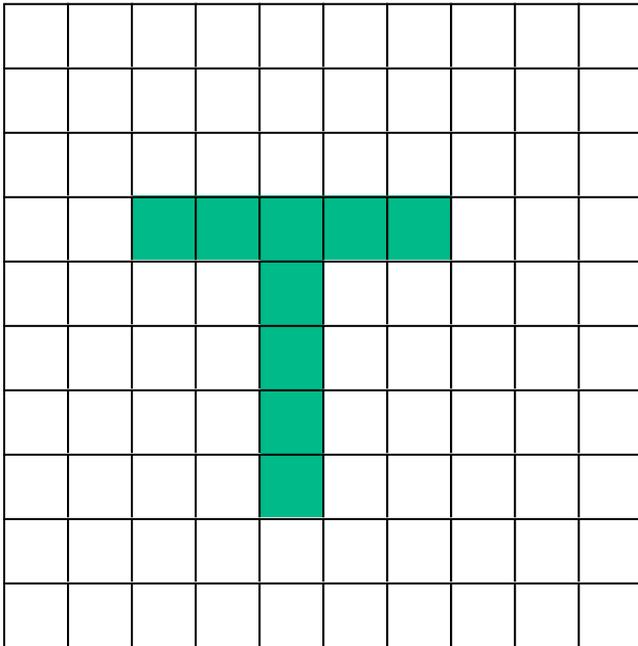
Широко известный способ - устранение шума с помощью операций математической морфологии:

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

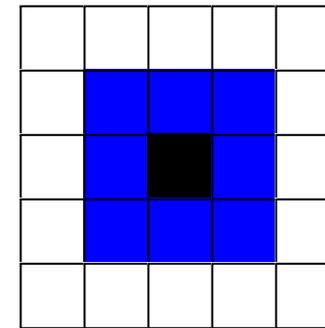


Математическая морфология

A



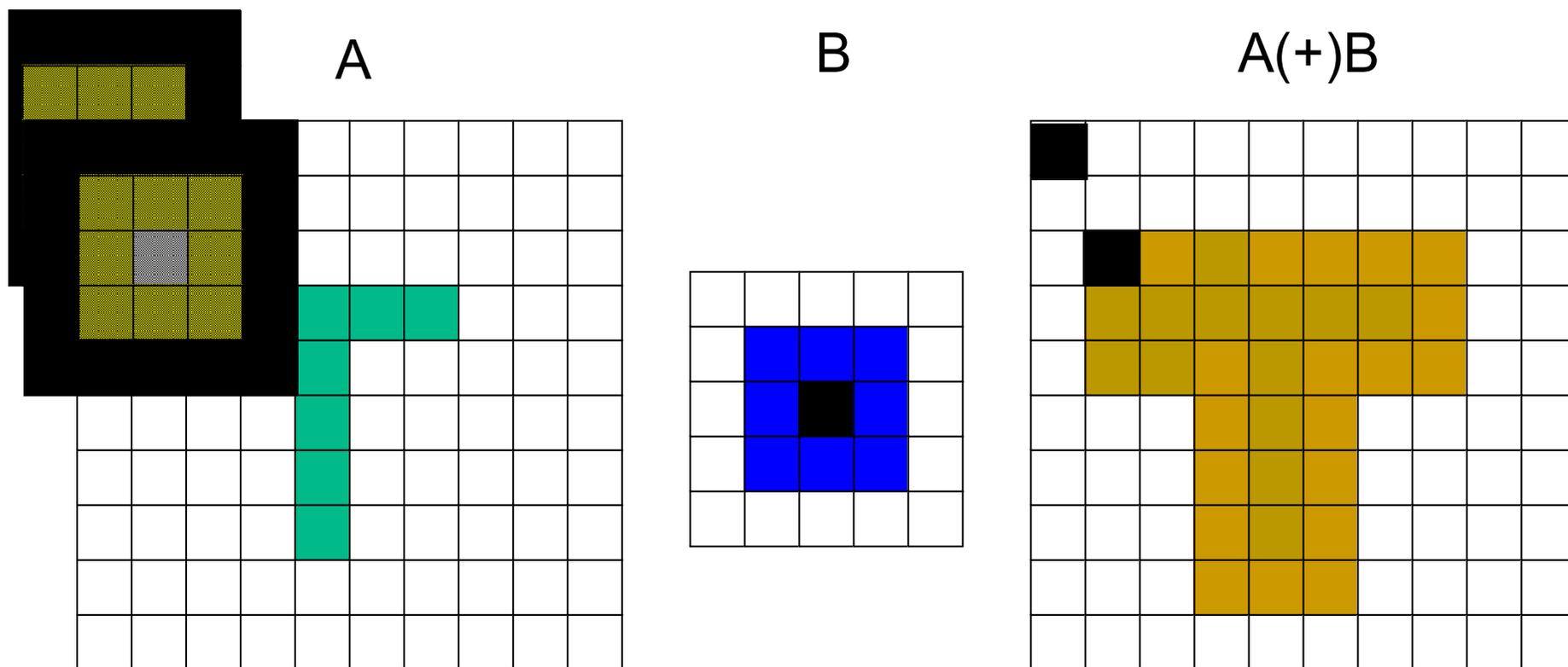
B



Множество A обычно является объектом обработки, а множество B (называемое структурным элементом) – инструментом.



Расширение в дискретном случае



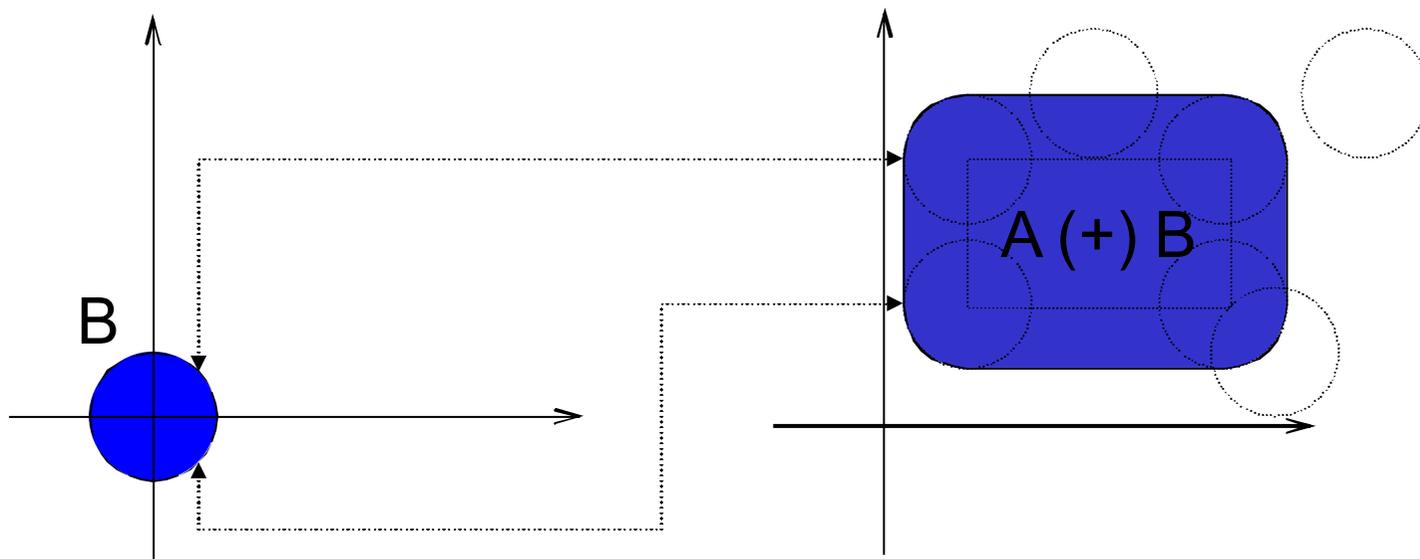
Операция «расширение» - аналог логического «или»



Расширение

Расширение (dilation)

$$A (+) B = \{t \in \mathbb{R}^2: t = a + b, a \in A, b \in B\}$$

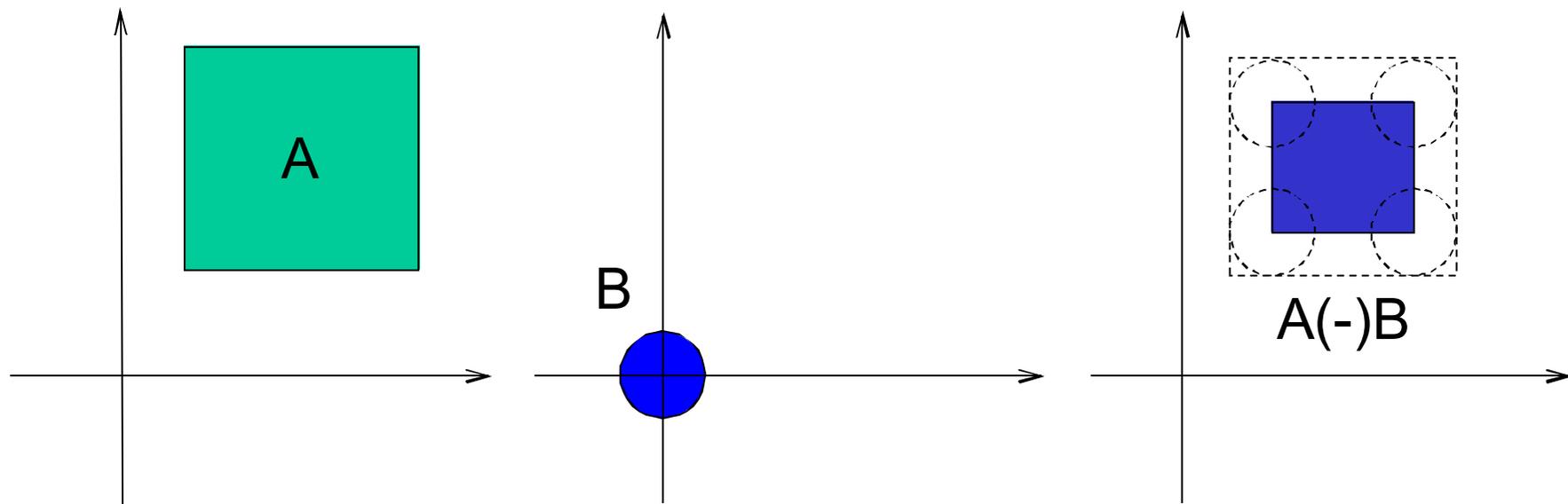




Сужение

Сужение (erosion)

$A (-) B = (A^C (+) B)^C$, где A^C – дополнение A





Результат операции сужения



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



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



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



Свойства

Коммутативный закон

- $A (+) B = B (+) A$
- $A (-) B < > B (-) A$

Ассоциативный закон

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



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)

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



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:



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



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



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



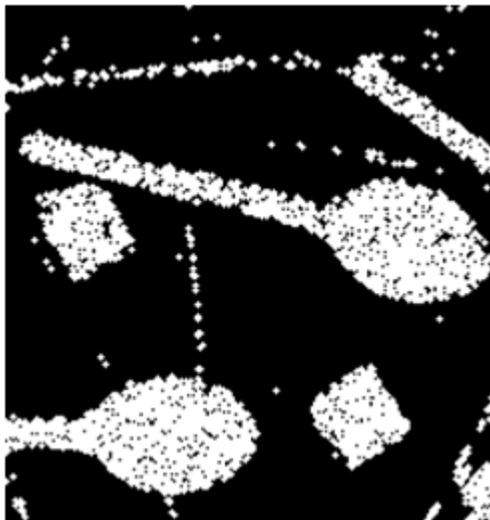
Не лучший пример для морфологии

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

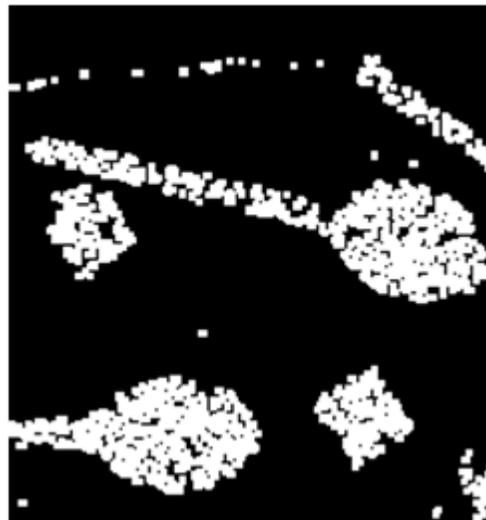




Применения операции открытия



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



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



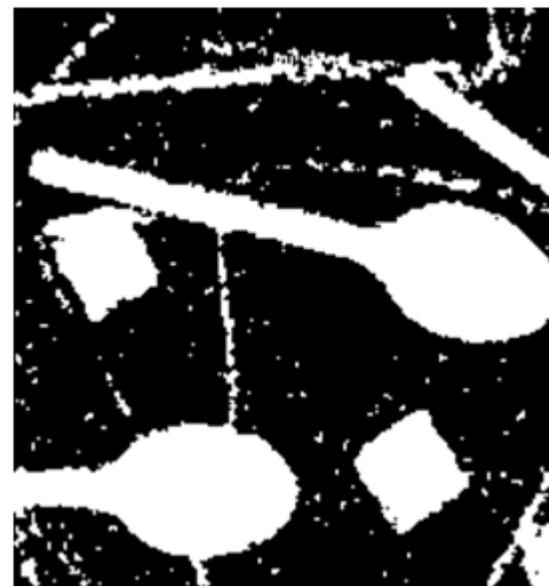
$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Часто помогает медианная фильтрация!



Медианный фильтр

Фильтр с окрестностью 3x3





Выделение связанных областей

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



Соседи пикселей:

	1	
2	*	3
	4	

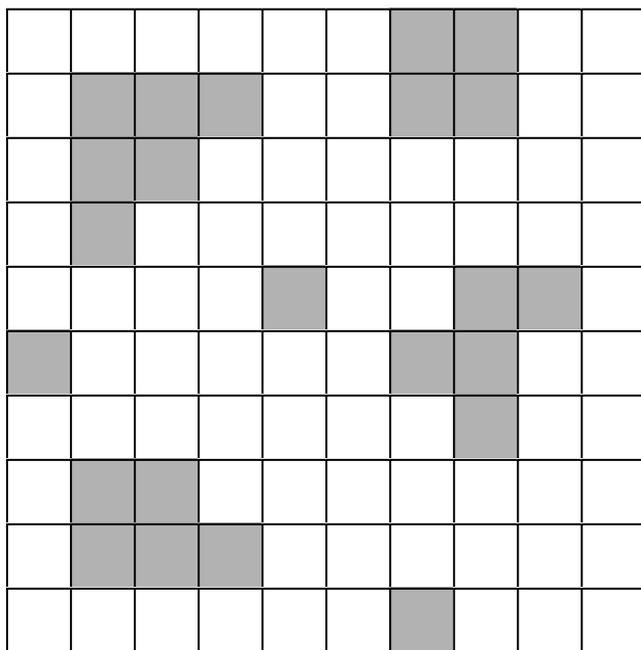
4-связность

1	2	3
4	*	5
6	7	8

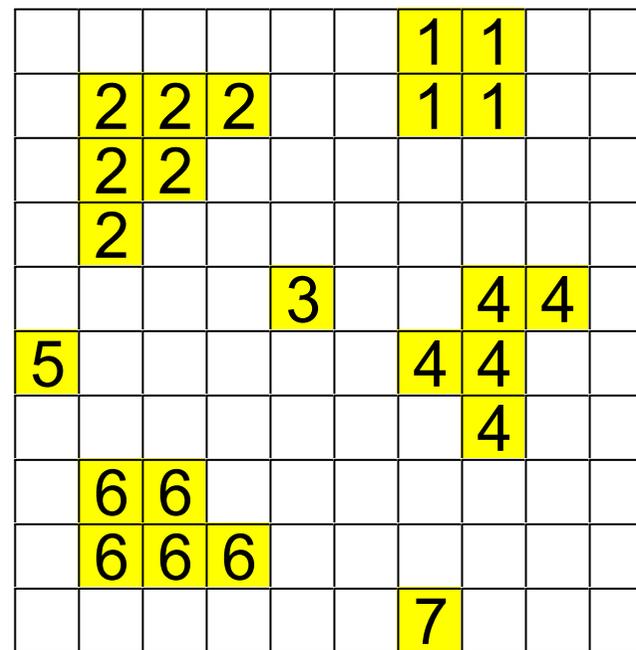
8-связность



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++);
        }
}
```



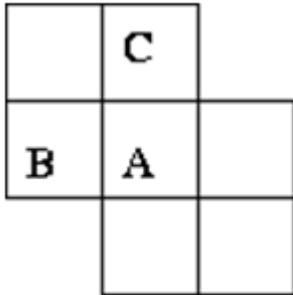
Рекурсивный алгоритм

```
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
```



Последовательное сканирование

Случай конфликта:

						1
					1	1
2				1	1	1
2	2		1	1	1	1
2	2	2	?			

Постобработка - переразметка с учетом эквивалентностей областей
(второй проход в алгоритме)



Выделенные связанные компоненты





Алгоритмы к рассмотрению

- Бинаризация и выделения связанных компонент



- Последовательное сканирование



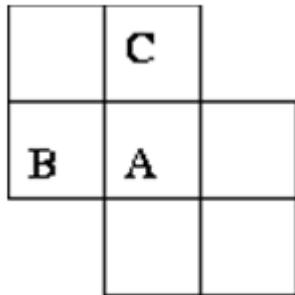
- Метод К-средних





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 А
2. if $I(A) - I_{avg}(CI(B)) < \delta$ xor $I(A) - I_{avg}(CI(C)) < \delta$ –
добавить А 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 В и С.
 2. $I_{avg}(CI(B)) - I_{avg}(CI(C)) > \delta$ –
добавляем пиксел А к тому классу, отклонение от которого минимально.

$I(A)$ – яркость пиксела А

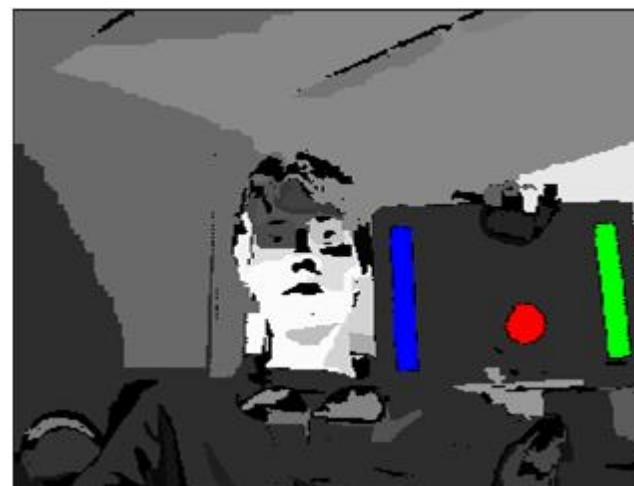
$CI(B)$ – область к которой принадлежит пиксел В

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

Выделение однородных областей методом последовательного сканирования



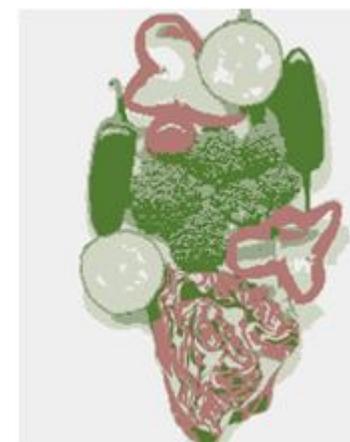
Пример :





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 классы (кластеры) на основе некоторой меры сходства объектов.

Что в случае сегментации изображения «объекты»?

- «Пиксели» изображения
- Каждый пиксель можно описать набором параметров («вектором»)
 - Яркость
 - Цвет
 - 3 параметра для RGB, HSV
 - Положение (x,y)



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;

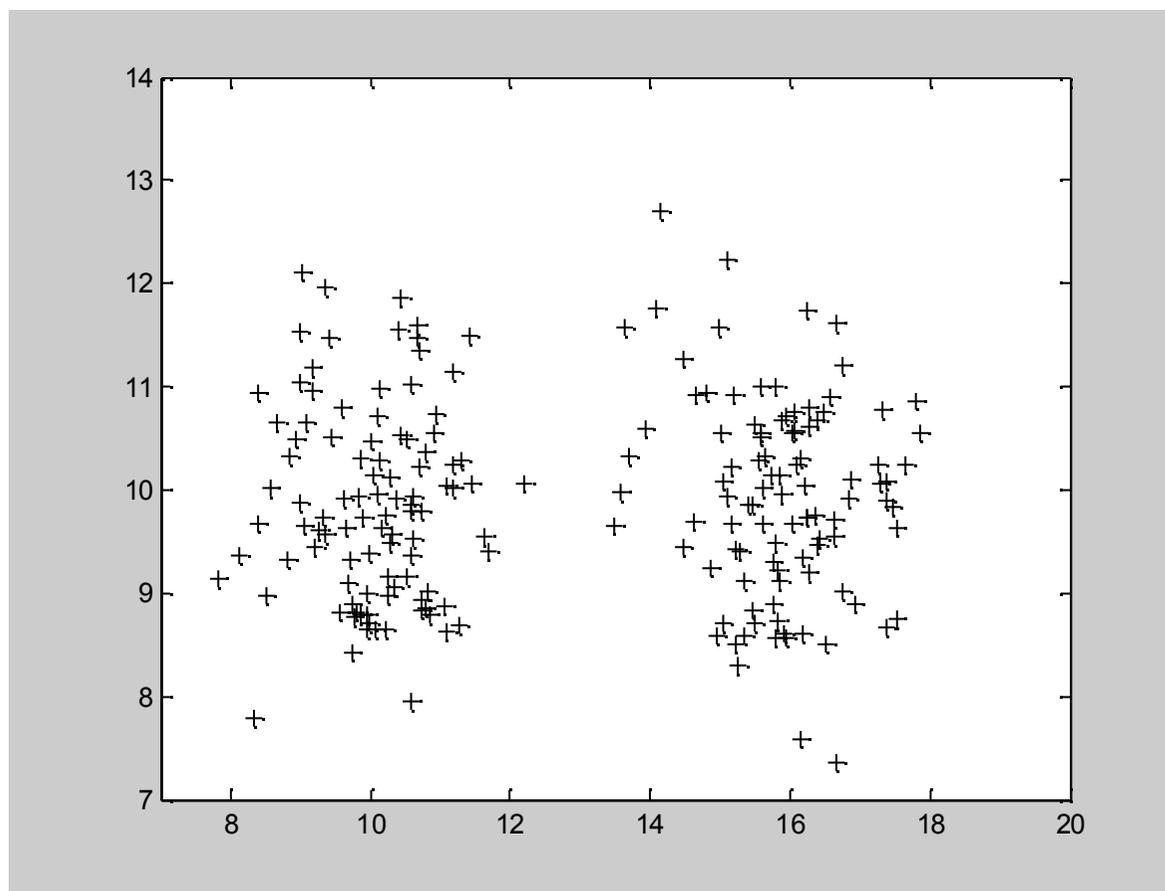


Алгоритм k-средних

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;



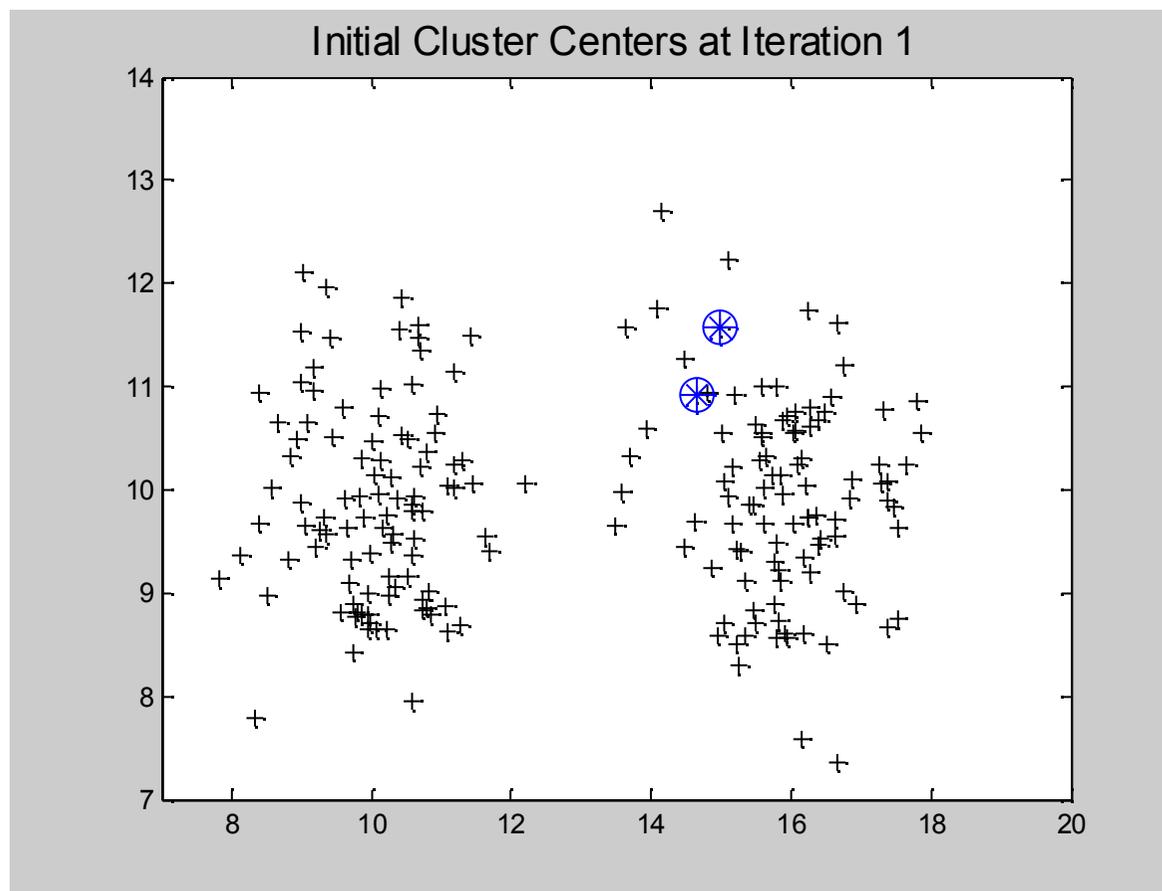
Пример кластеризации в 2D



Исходные данные



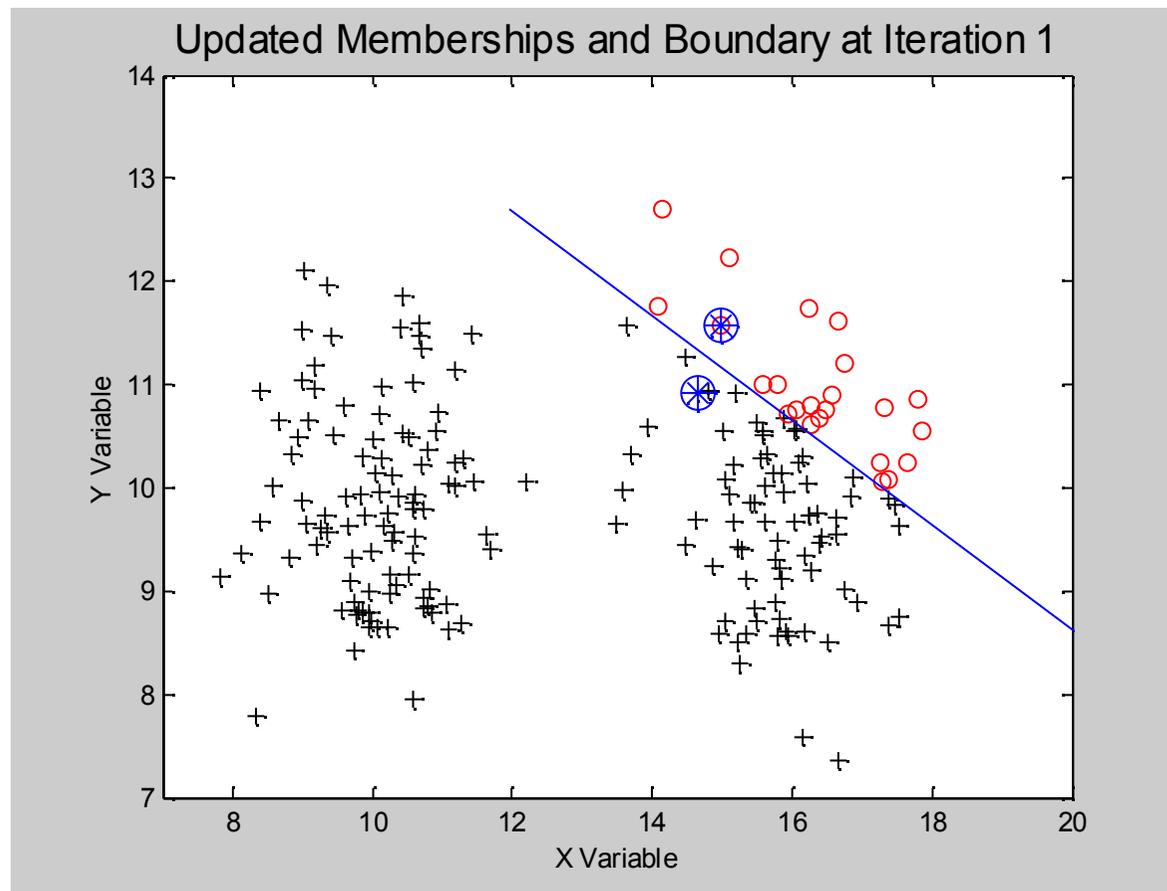
Пример кластеризации в 2D



Случайная инициализация центров кластеров (шаг 1)



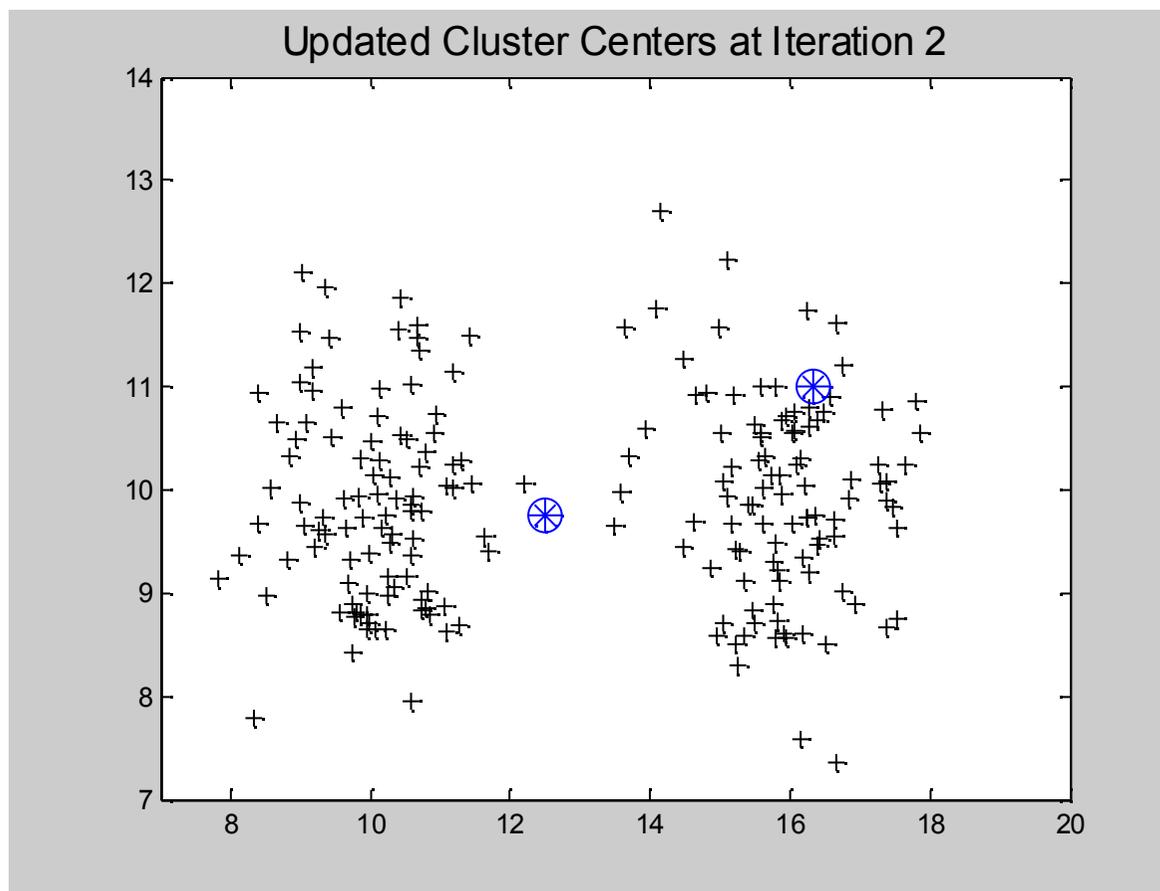
Пример кластеризации в 2D



Кластеры после первой итерации (шаг 2)



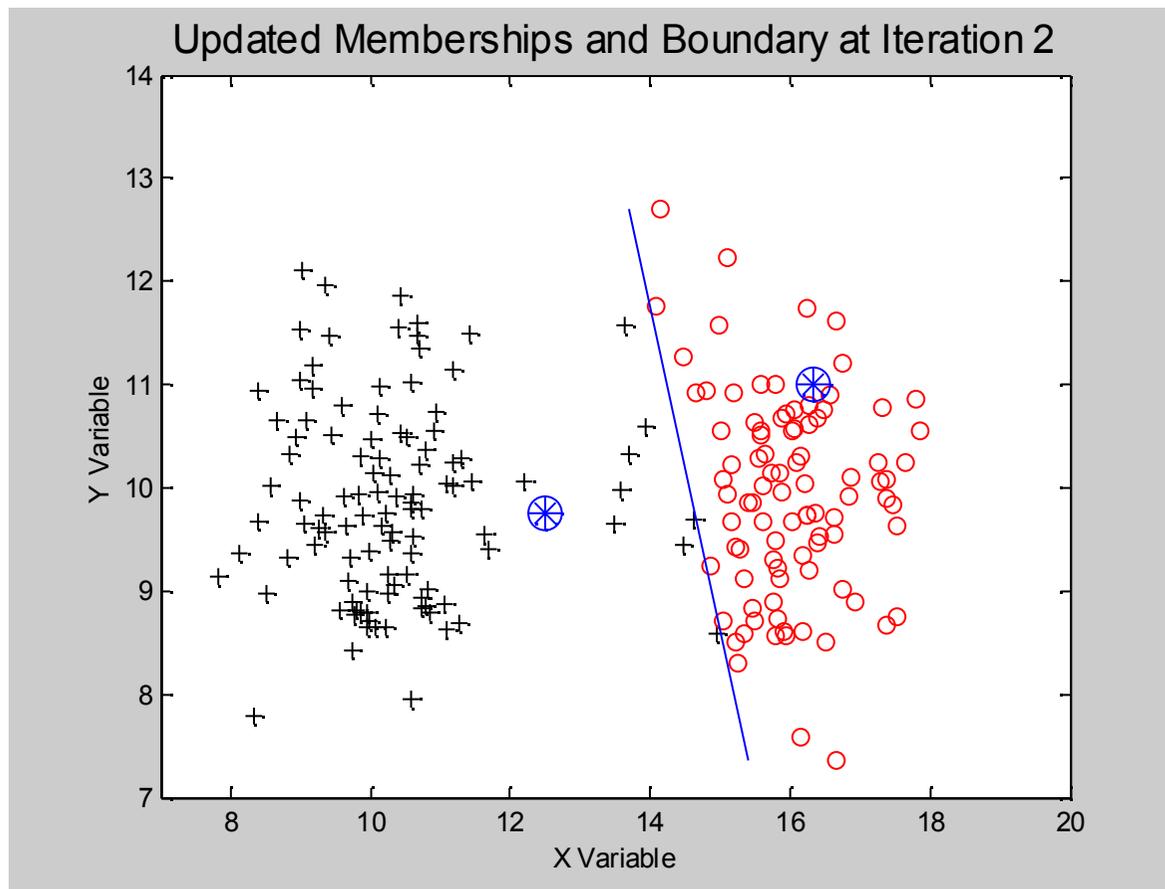
Пример кластеризации в 2D



Пересчет центров кластеров после первой итерации (шаг 3)



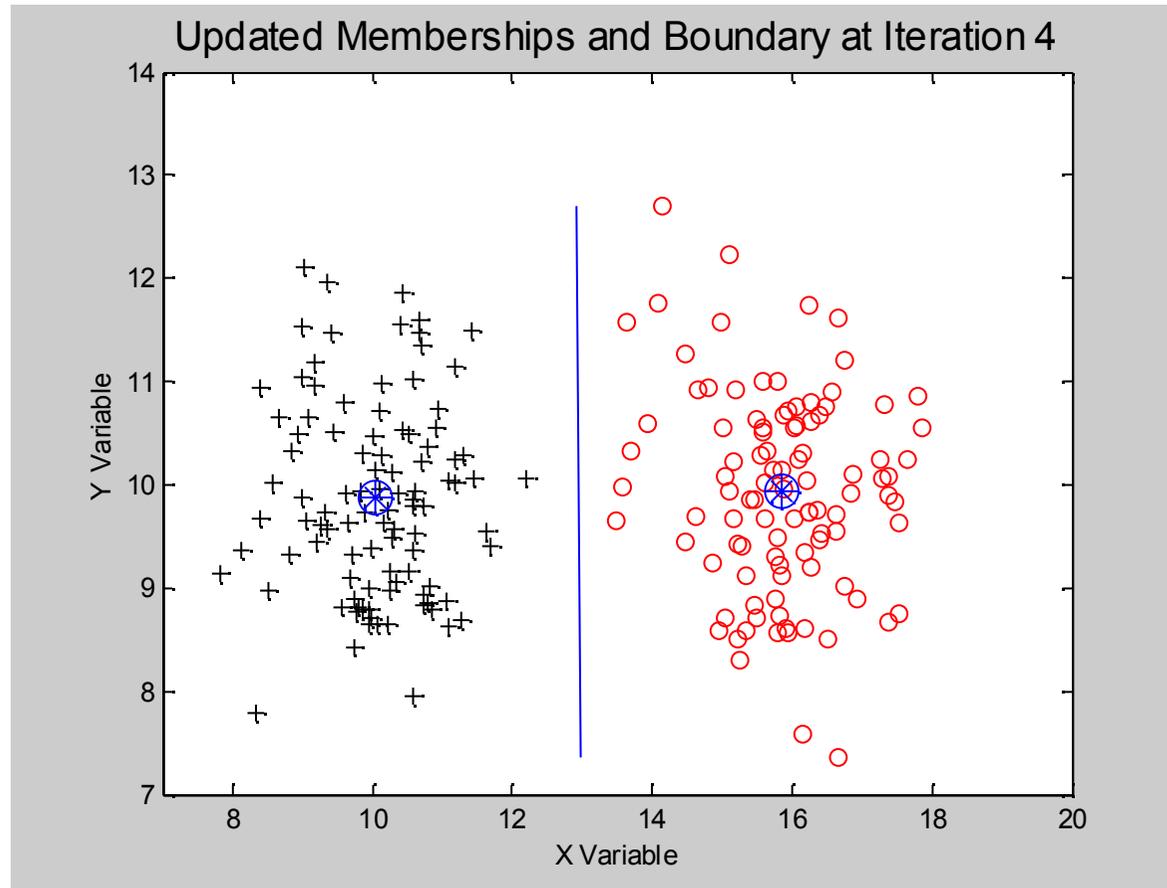
Пример кластеризации в 2D



Кластеры после второй итерации (шаг 2)



Пример кластеризации в 2D

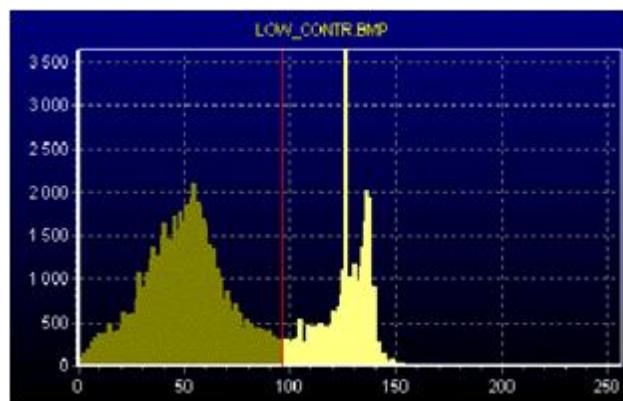


Стабильная конфигурация после четвертой итерации



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 $k=2$)



K-средних для бинаризации

Алгоритм:

1. Выбрать порог T равным середине диапазона яркостей;
2. Вычислить среднюю яркость всех пикселей с яркостью $< T$ m_1 , аналогично m_2 для пикселей с яркостью $> T$;
3. Пересчитать порог $T = (m_1 + m_2) / 2$;
4. Повторять шаги 2, 3 порог не перестанет изменяться;



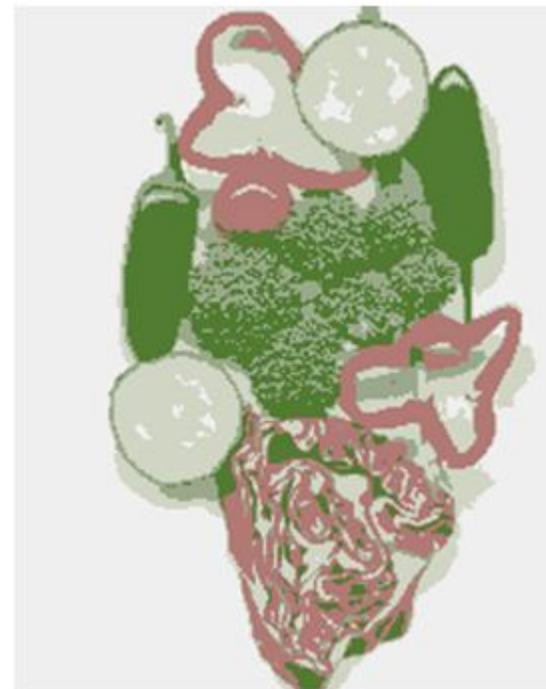
Алгоритм K-средних. Примеры.



Исходное изображение



Кластеры по яркости

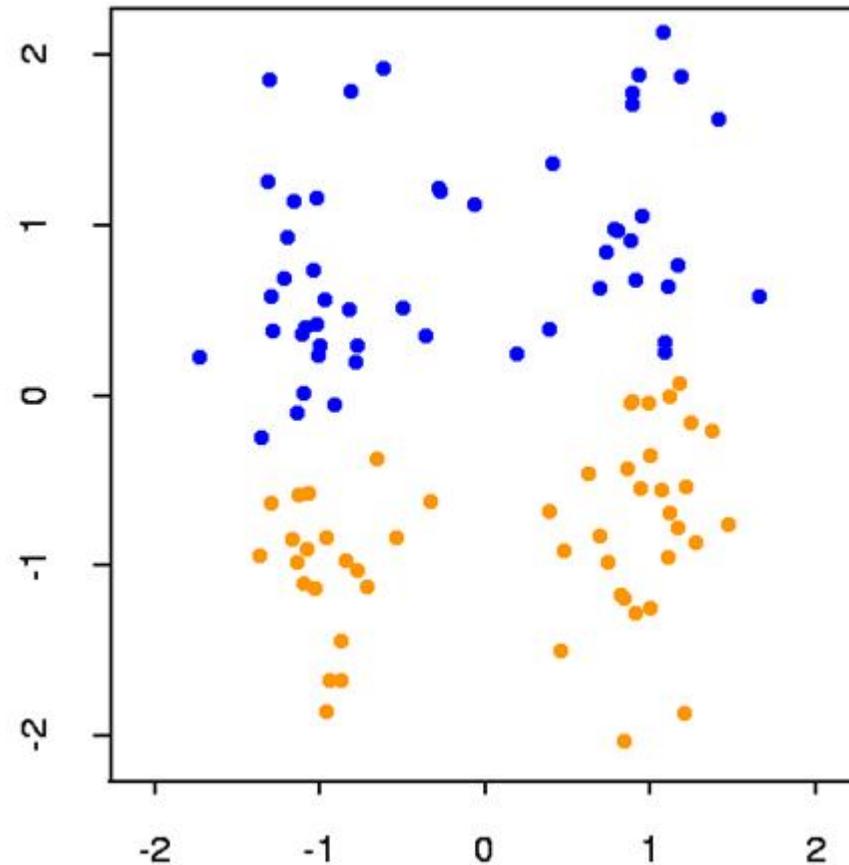


Кластеры по цвету



Алгоритм К-средних

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





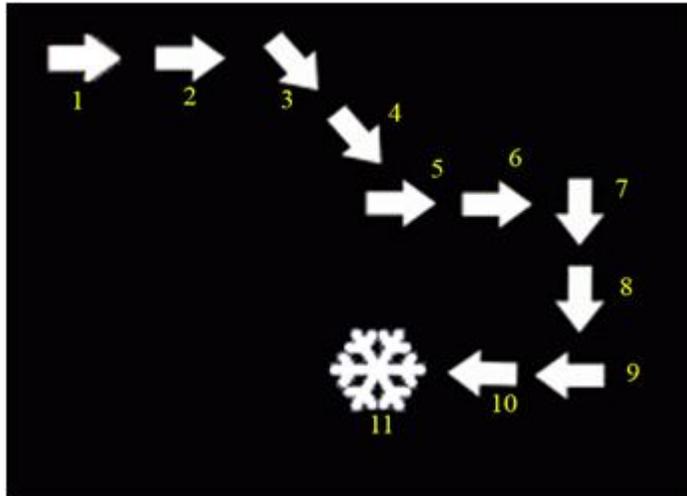
Современные алгоритмы



Задача сегментации изображений продолжает активно исследоваться. Некоторый из современных алгоритмов: MeanShift (2002), QuickShift (2008), TurboPixel (2010)



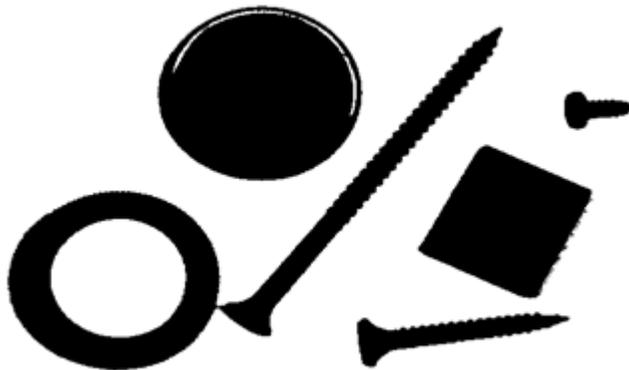
Анализ выделенных областей



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

- геометрические признаки
- фотометрические признаки

На основе этих характеристик можно классифицировать получаемые области

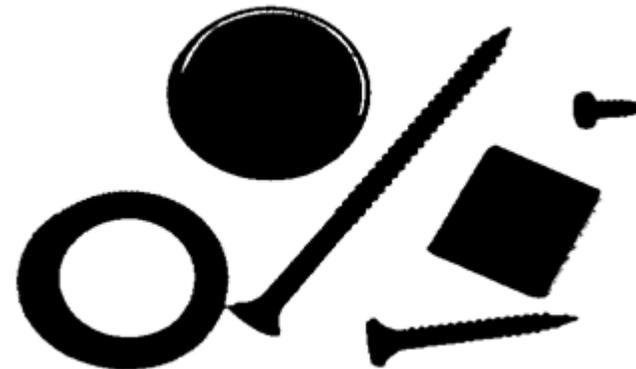




Геометрические признаки

Для каждой области можно подсчитать некий набор простейших числовых характеристик:

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

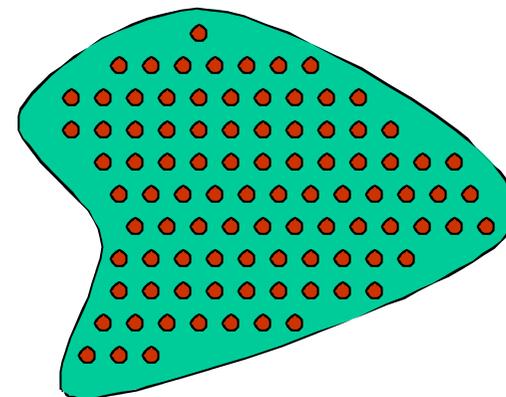




Площадь и центр масс

- Площадь – количество пикселей в области;

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



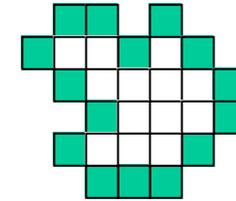
- Центр масс

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



Периметр и компактность

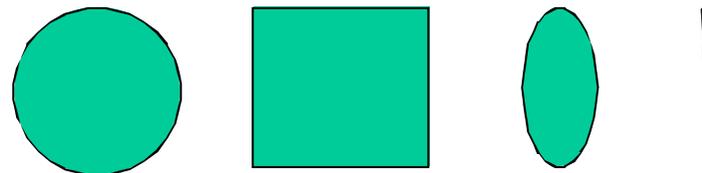
- Периметр – количество пикселей принадлежащих границе области;



- Компактность – отношение квадрата периметра к площади;

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

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





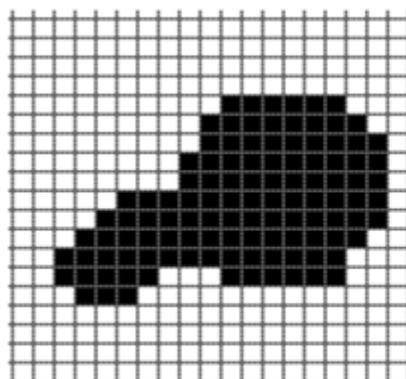
Подсчет периметра области

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*)

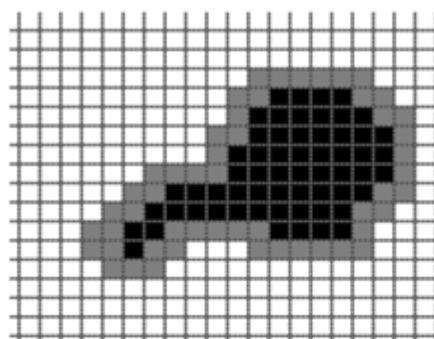
Периметр зависит также от того 4-х или 8-ми связность используется для определения соседей.



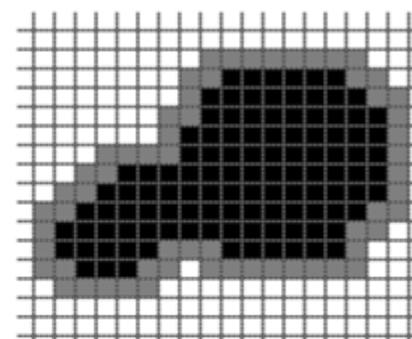
Пример периметров области



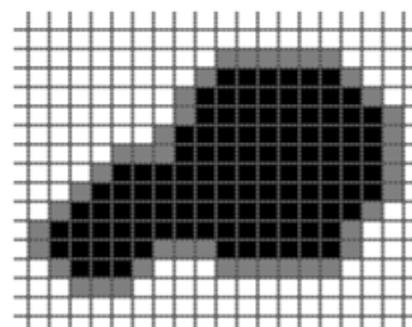
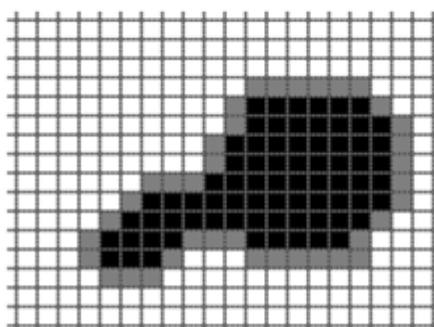
Область



Внутренняя граница



Внешняя граница





Операция оконтуривания объекта

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

Внутреннее оконтуривание

- $C_1 = A - (A (-) B)$

Внешнее оконтуривание

- $C_0 = (A (+) B) - A$

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

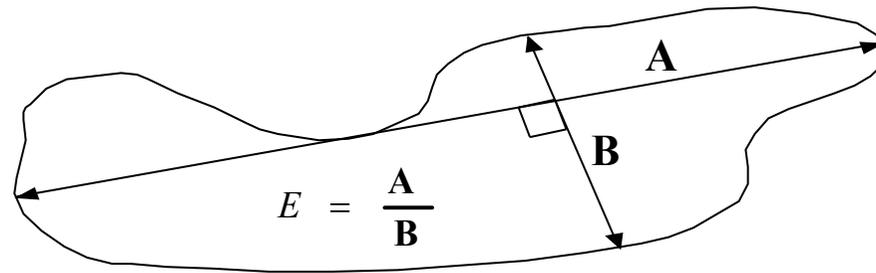
Для распознавания нас интересуют характеристики инвариантные по отношению к масштабированию, переносу, повороту:

- 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

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

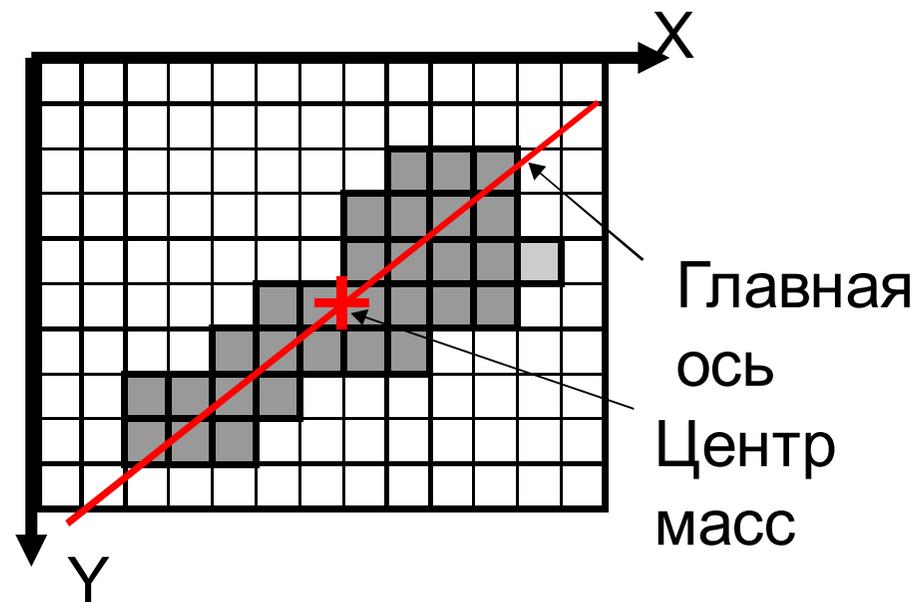




Ориентация главной оси инерции

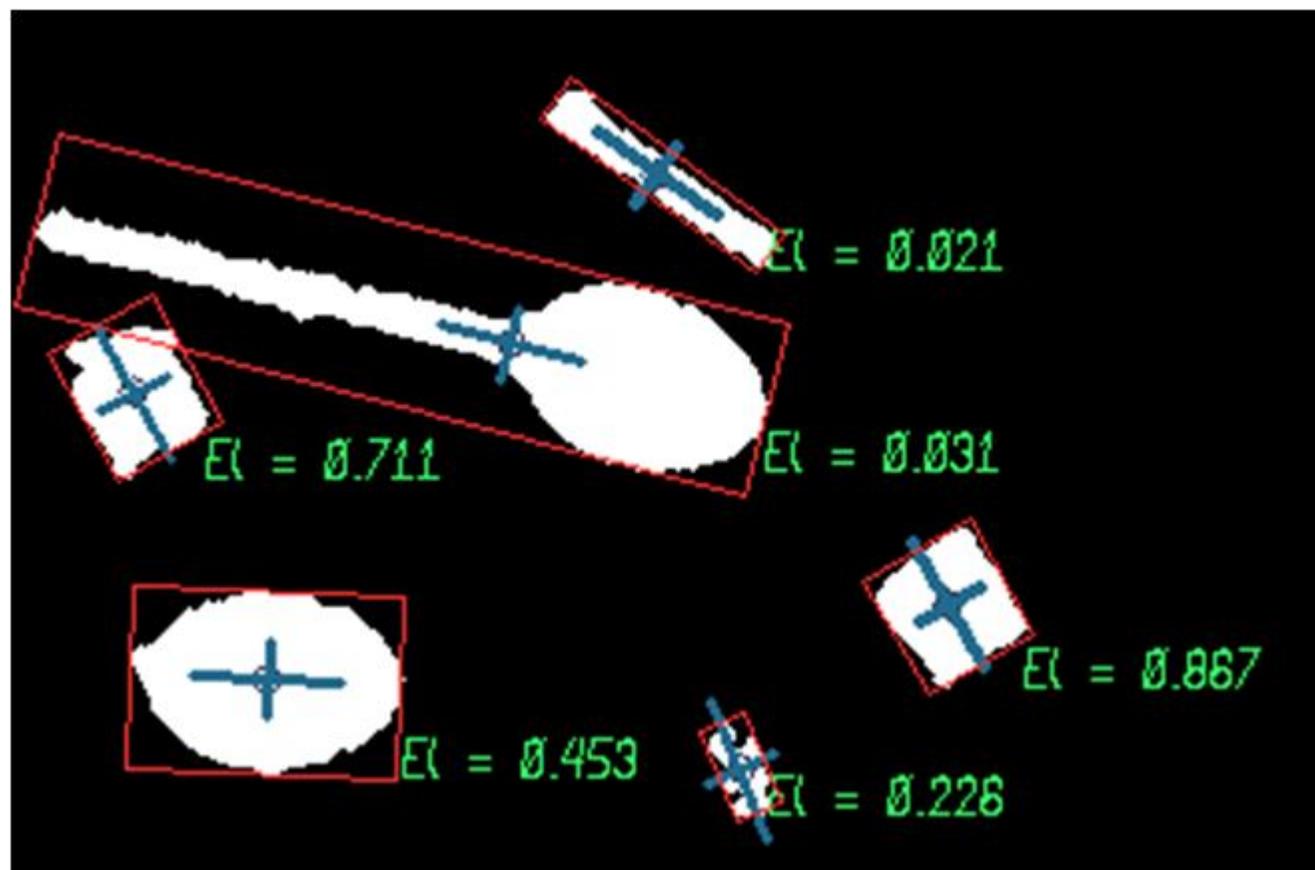
Не является инвариантной к повороту, но в ряде случаев предоставляет полезную информацию об ориентации объекта:

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





Пример



Вычисленные значения признаков



Другие признаки

Другие инвариантные характеристики области:

$$M_1 = m_{20} + m_{02}$$

$$M_2 = (m_{20} - m_{02})^2 + 4m_{11}^2$$

$$M_3 = (m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2$$

$$M_4 = (m_{30} + m_{12})^2 + (m_{21} + m_{03})^2$$

$$M_5 = (m_{30} - 3m_{12})(m_{30} + m_{12}) [(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2] \\ + (3m_{21} - m_{03})(m_{21} + m_{03}) [3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2]$$

$$M_6 = (m_{20} + m_{02}) [(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2] \\ + 4m_{11}(m_{30} + m_{12})(m_{03} + m_{21})$$

$$M_7 = (3m_{21} - m_{03})(m_{12} + m_{30}) [(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2] \\ - (m_{30} - 3m_{12})(m_{12} + m_{03}) [3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2]$$

(54)



Фотометрические признаки

Для каждой области можно подсчитать некий набор простейших числовых характеристик:

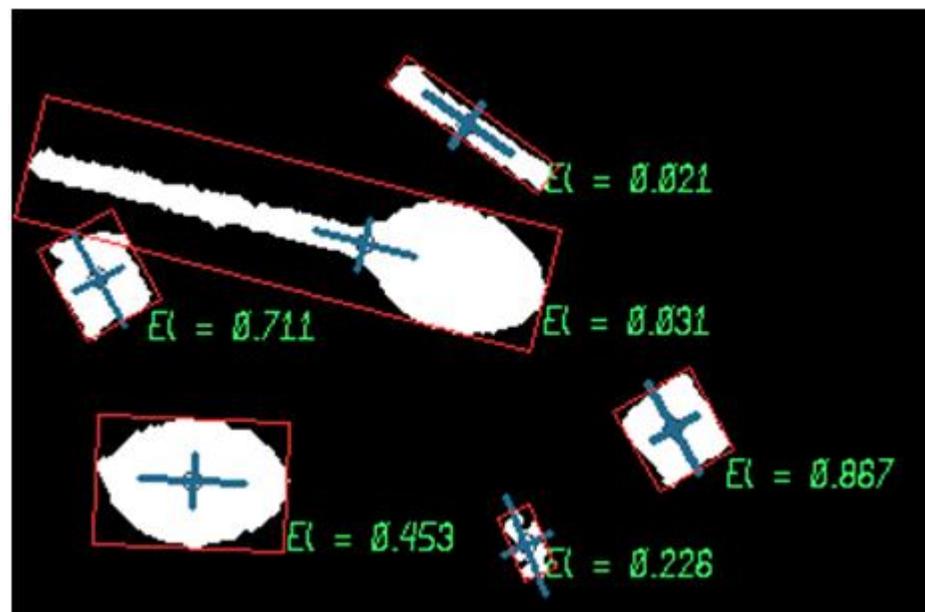
- Средняя яркость
- Средний цвет (если изображение цветное)
- Гистограмма распределения яркостей (или три гистограммы распределения R, G, B)
- Дисперсию (разброс) яркостей или цвета

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
 - На следующей лекции!
 - Второе задание!



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

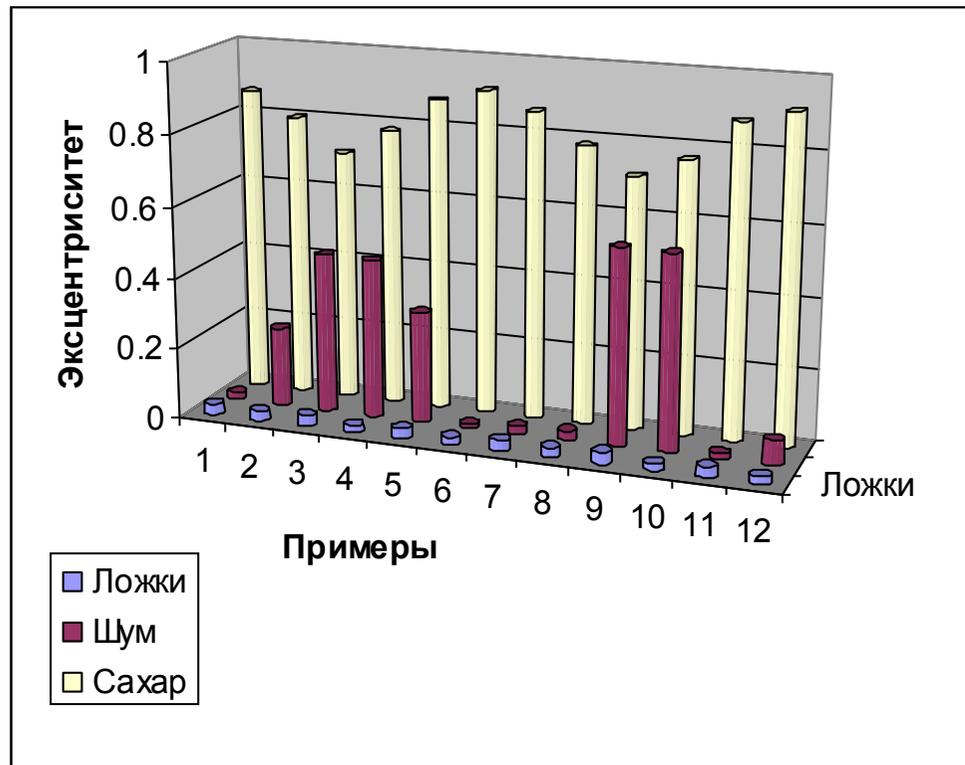
- Собрать тренировочную базу изображений
 - Где только ложки
 - Где только сахар
 - Где только шум

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



Графический анализ

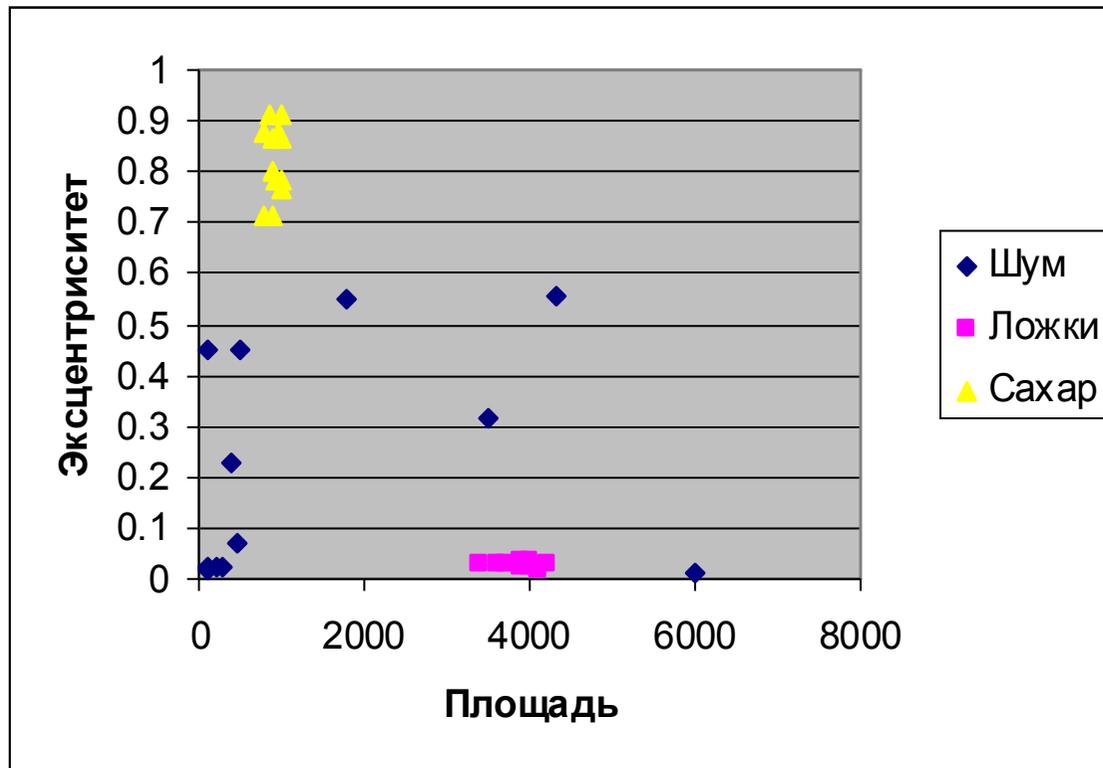
- Диаграмма распределения эксцентриситета (проблема – не получается отличить шум от ложек)





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.
- Рассмотрим на следующей лекции!



Резюме лекции

- Сопоставление шаблонов как метод поиска объектов
- Выделение краёв
- Сегментация изображения для выделения областей-кандидатов
- Вычисление цветовых и геометрических признаков изображения