

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 29.07.2022 17:52:13

Уникальный программный ключ:

c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

Компьютерное зрение. Введение

Лекция 1

Что это такое и почему это сложно?



Задача компьютерного зрения

- Понять, что запечатлено на изображении



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

Source: S. Narasimhan

Тест Тьюринга

- «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
- Что это в действительности обозначает?
 - Зрение — источник семантической информации о мире
 - Зрение — источник метрической информации о трехмерном мире

Семантическая информация



Классификация сцены



Выделение объектов



Выделение границ объектов

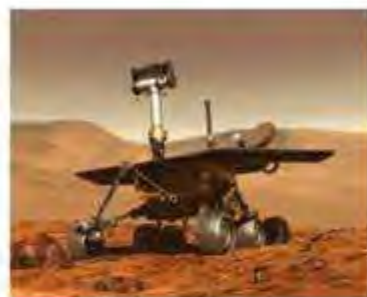


Признаки объектов

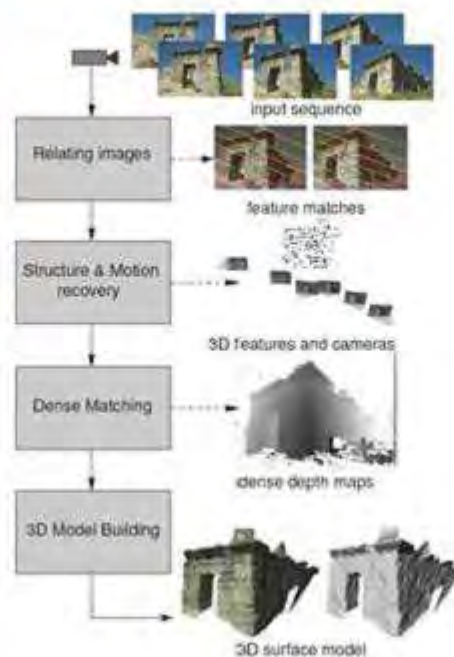


Измерения

Сtereo
реконструкция



Структура из
движения



Моделирование по
пользовательским
снимкам



Slide: Svetlana Lazebnik

Зрение... принятые названия

- Фотограмметрия (Photogrammetry)
 - Ист: измерение расстояний между объектами по 2D изображениям
- Компьютерное зрение (Computer vision)
 - Восстановление 3D структуры по 2D изображения.
 - Шире: принятие решений о физических объектах, основываясь на их изображениях
- Машинное зрение (Machine vision)
 - Решение промышленных, производственных задач (исторически)
- Обработка изображений (Image processing)
 - На входе и выходе изображение
- Анализ изображений (Image analysis)
 - Фокусируется на работе с 2D изображениями
- Распознавание образов (Pattern recognition)
 - Распознавание, обучение на абстрактных величинах, полученных в том числе и из изображений

Зачем?

- Наглядное применение многих математических методов анализа данных
- Много практических применений
- Много нерешенных задач
 - Стимул для развития методов анализа данных и высокопроизводительных вычислений
- Сложно
 - 25+\% мозга человека отвечает за зрение
 - «ИИ-полная» задача – решение задачи зрения на уровне человека равносильно решению задачи искусственного интеллекта

Почему зрение — это сложно?

Точка наблюдения



Michelangelo 1475-1564



slide credit: Fei-Fei, Fergus & Torralba



Освещение



image credit: J. Koenderink

Масштаб



Slide credit: Fei-Fei, Fergus & Torralba

Деформация



Ху, Вейхонг 1943

Slide credit: Fei-Fei, Fergus & Torralba

Перекрытие

Magritte, 1957



Маскировка



Движение

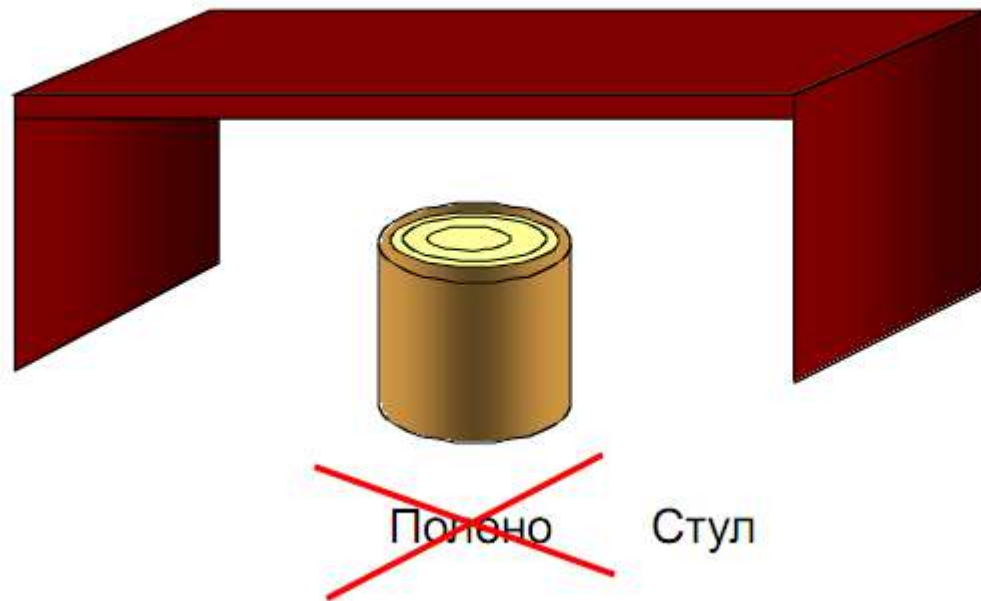


Внутриклассовая изменчивость



Slide credit: Fei-Fei, Fergus & Torralba

Контекст



Контекст и локальная неоднозначность



Slide credit: Fei-Fei, Fergus & Torralba

Сложности или возможности?

- Изображение запутывает, но дает много подсказок
- Наша задача – интерпретировать подсказки

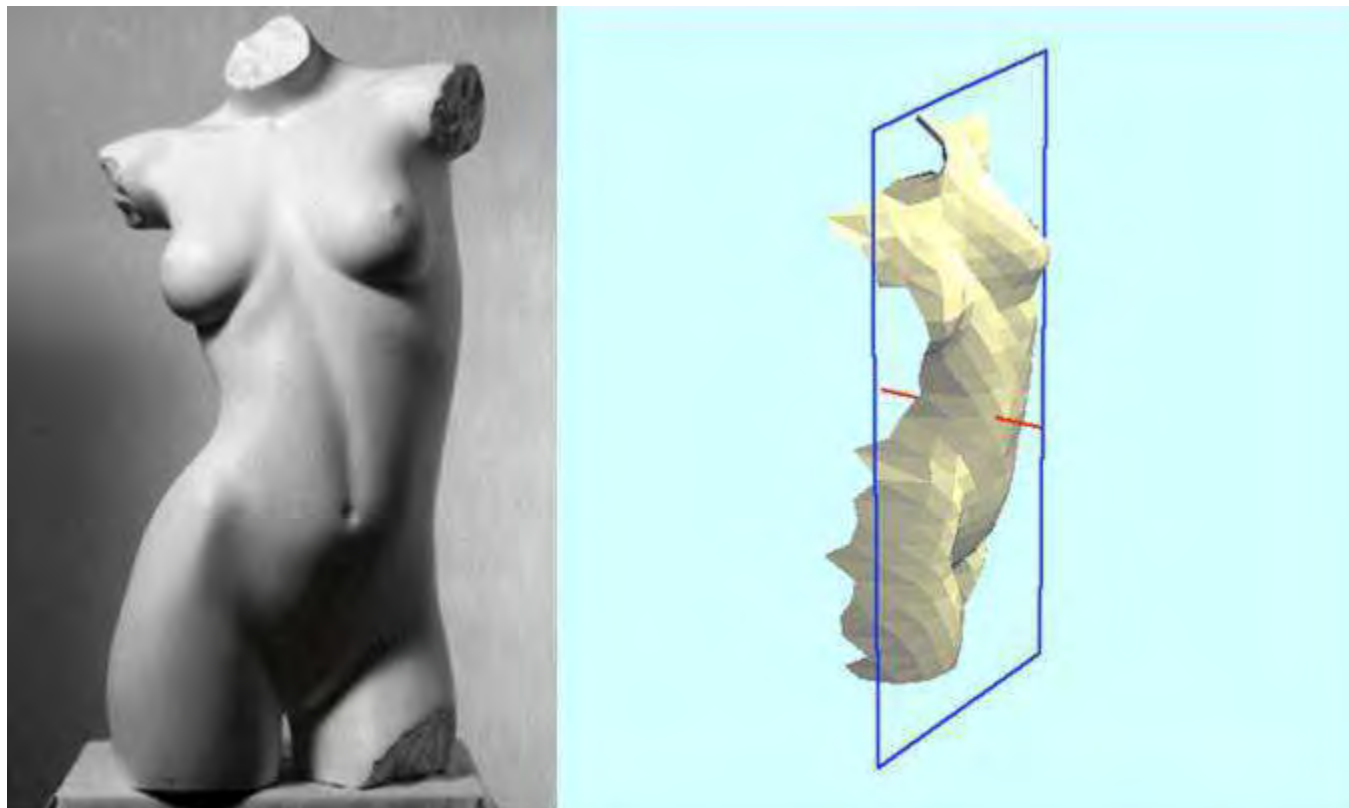


Image source: J. Koenderink

Цвет



Тени и освещение



Source: J. Koenderink

Отбрасываемые тени



Source: J. Koenderink

Текстура



Яркостные преобразования

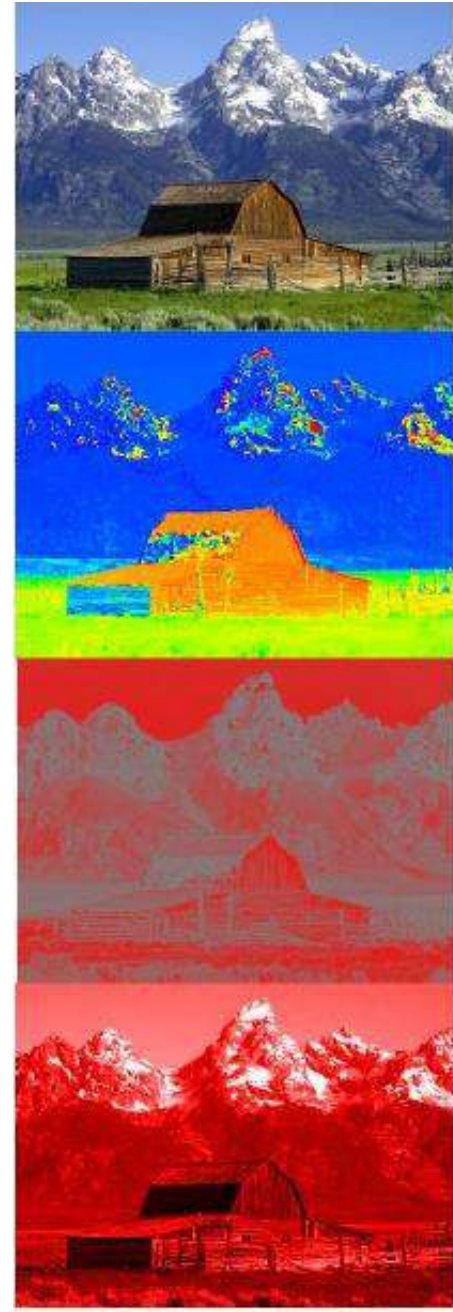
Лекция 2

Изображения

- Бинарные
- Полутоновые
- Многоканальные

Обработка цветных изображений

- По каналам RGB
- По HSL



Подходы к обработке цветных изображений

- Покомпонентная обработка
- Векторная обработка

$$\mathbf{c} = \begin{bmatrix} c_R \\ c_G \\ c_B \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Преобразование цветного к полутоновому

$$\text{Gray} = 0.3 * R + 0.59 * G + 0.11 * B;$$

$$\text{Gray} = (R + G + B) / 3,$$

Яркостные преобразования

$$I'(x, y) = f(I(x, y))$$

$$f(I) = aI + b$$

LUT (Look-Up-Table)

$$I'(x, y) = f(I(x, y))$$

$$\text{Im}[i, j] = \text{LUT}[\text{Im}[i, j]]$$

LUT: Osiris и Impulz, представленные компанией «VisionColor»



Задание LUT

- таблица отображения;
- математическая функция;
- адаптивное задание по гистограмме.

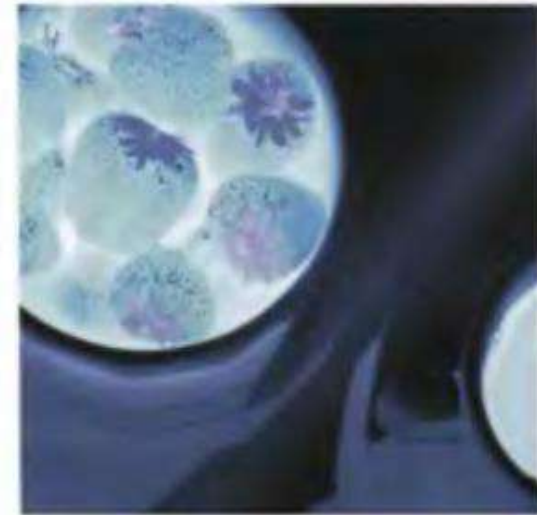
Основные яркостные преобразования



Цветовое дополнение

- Аналогично негативу для полутонного изображения
- Полезно для выявления деталей внутри темных областей

$$\text{neg} = \text{Max} - \text{pos}$$



Демонстрация гамма-коррекции изображения

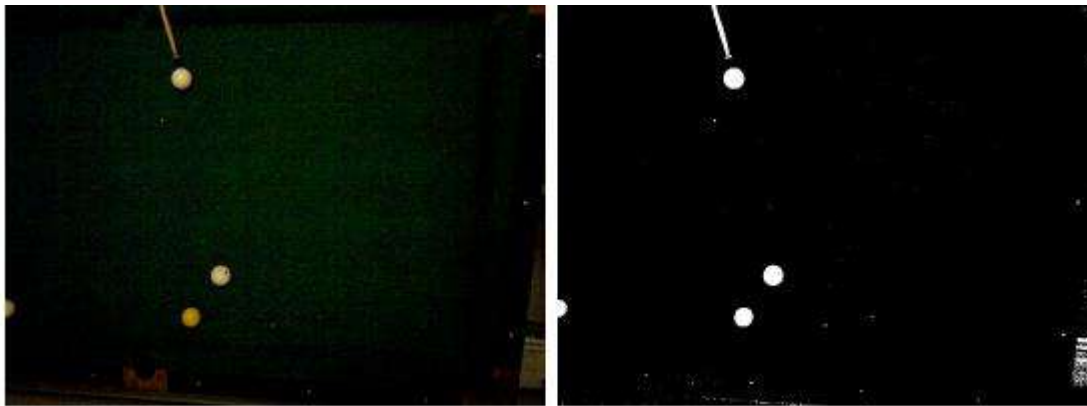
$$y = c \cdot x^\gamma$$



Типовые операции:

- бинаризация изображения по одному или двум порогам;
- сегментация изображения на несколько яркостных диапазонов;
- вырезание определенных диапазонов яркости;
- вырезание битовых плоскостей.

Пороговое преобразование



$f(v)=1$, если $v>t$

t – порог

$f(v)=0$, иначе

Пороговая бинаризация полутоновых изображений

- Верхняя пороговая бинаризация
- Нижняя пороговая бинаризация
- Пороговая бинаризация по диапазону

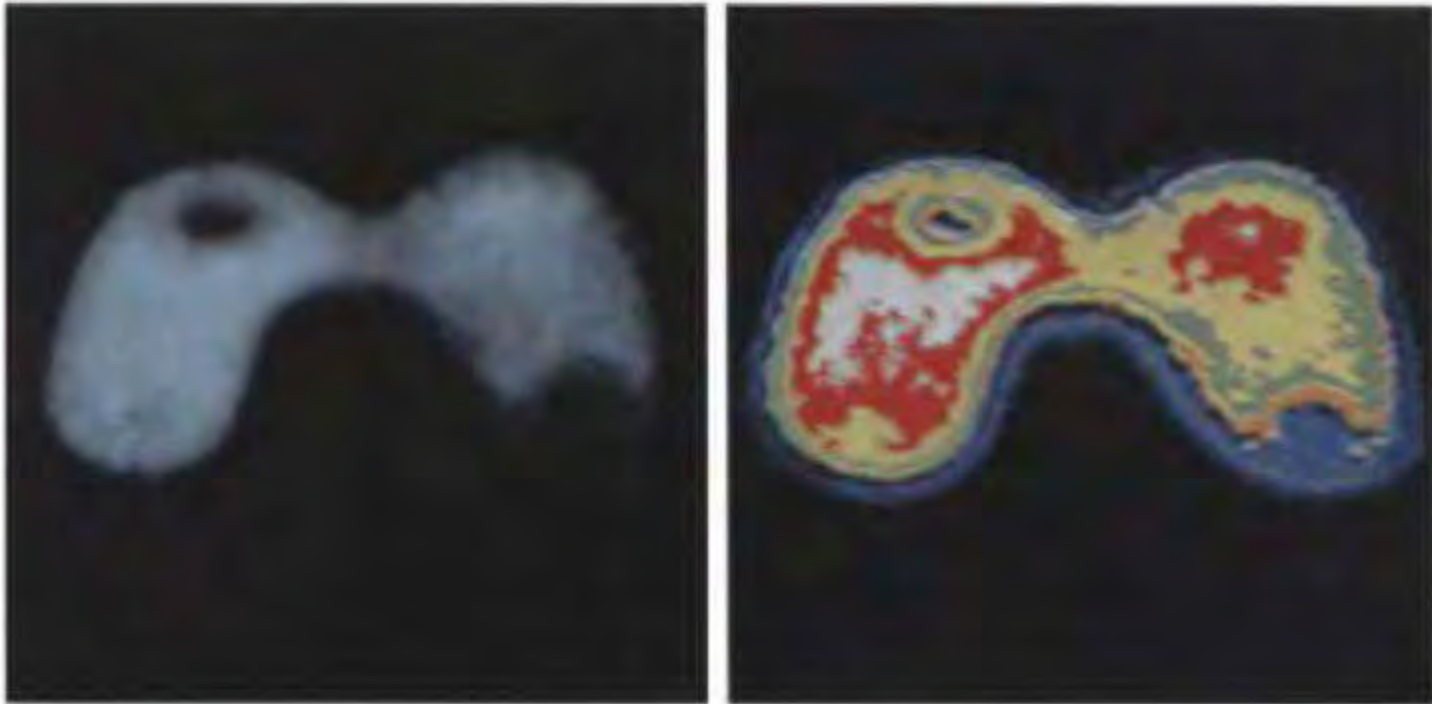
Вырезание цветового диапазона

$$s_i = \begin{cases} 0,5, & \text{если } \left[|r_j - a_j| > \frac{W}{2} \right] \text{ для любого } 1 \leq j \leq n; \\ r_i, & \text{в остальных случаях} \end{cases} \quad i = 1, 2, \dots, n$$

$$s_i = \begin{cases} 0,5, & \text{если } \sum_{j=1}^n (r_j - a_j)^2 > R_0^2; \\ r_i, & \text{в остальных случаях;} \end{cases} \quad i = 1, 2, \dots, n$$



Квантование по яркости



Цветовая коррекция изображений

- Изменение цветового баланса
 - Компенсация:
 - Неверного цветовосприятия камеры
 - Цветного освещения



Коррекция с опорным цветом

- Предположение
 - Пользователь указывает цвет вручную;
- Источник:
 - Априорные знания – «облака – белые»
 - Хорошая фотография этой же сцены
- Метод
 - Преобразовать по каждому из каналов цвета по формуле:

$$R * \frac{R_{dst}}{R_{src}}; \quad G * \frac{G_{dst}}{G_{src}}; \quad B * \frac{B_{dst}}{B_{src}};$$

Коррекция с опорным цветом

- Примеры:



«Серый мир»

- Предположение:
 - Сумма всех цветов на изображении естественной сцены дает серый цвет;
- Метод:
 - Посчитать средние яркости по всем каналам:

$$\bar{R} = \frac{1}{N} \sum R(x, y); \quad \bar{G} = \frac{1}{N} \sum G(x, y); \quad \bar{B} = \frac{1}{N} \sum B(x, y); \quad Avg = \frac{\bar{R} + \bar{G} + \bar{B}}{3};$$

- Масштабировать яркости пикселей по следующим коэффициентам:

$$R' = R \cdot \frac{Avg}{\bar{R}}; \quad G' = G \cdot \frac{Avg}{\bar{G}}; \quad B' = B \cdot \frac{Avg}{\bar{B}};$$

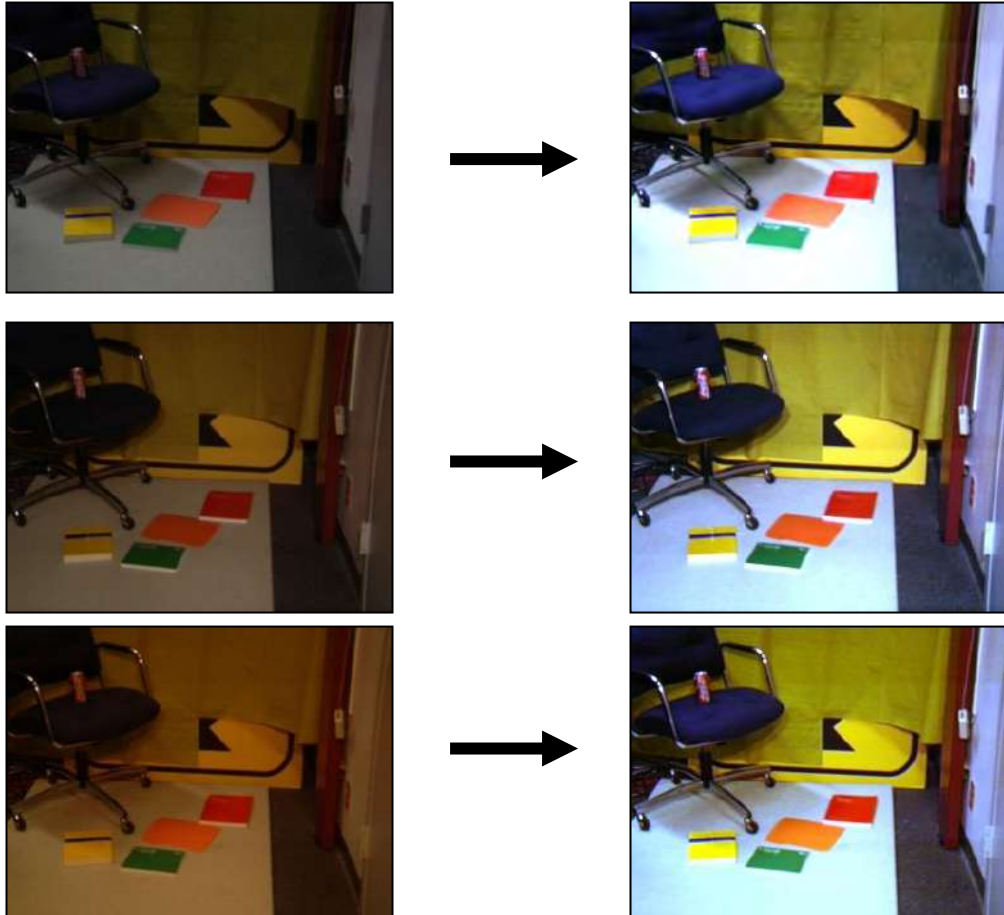
«Серый мир» - примеры



«Серый мир» - примеры

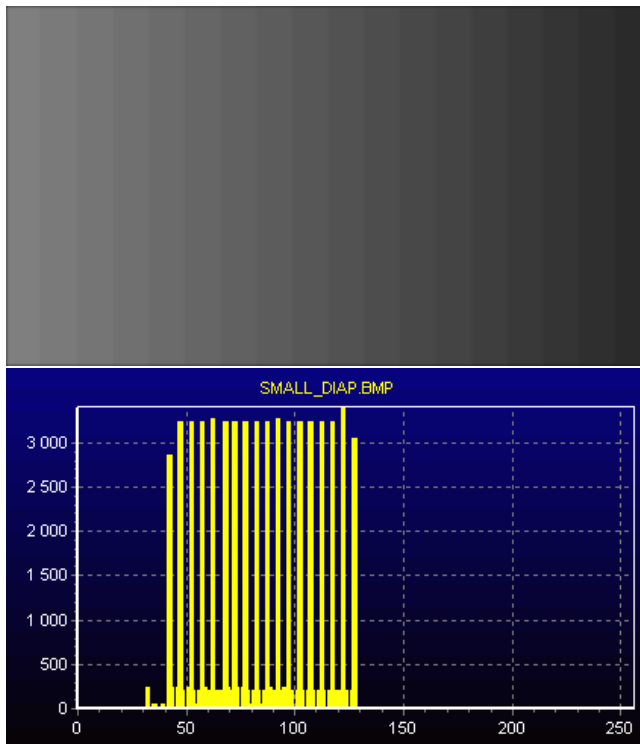


«Серый мир» - примеры



Линейная коррекция яркости

Компенсация узкого диапазона яркостей – линейное растяжение:



$$f^{-1}(y) = (y - y_{\min}) * \frac{(255 - 0)}{(y_{\max} - y_{\min})}$$

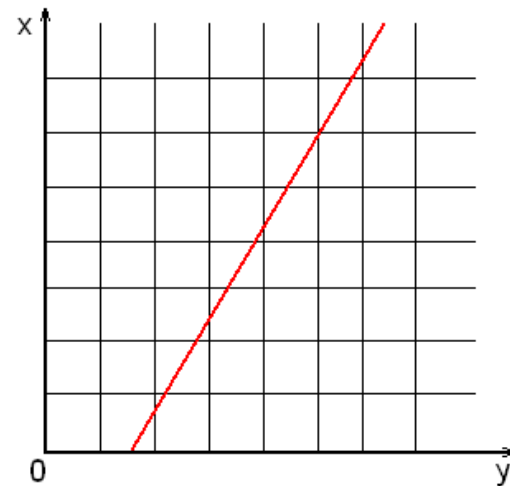
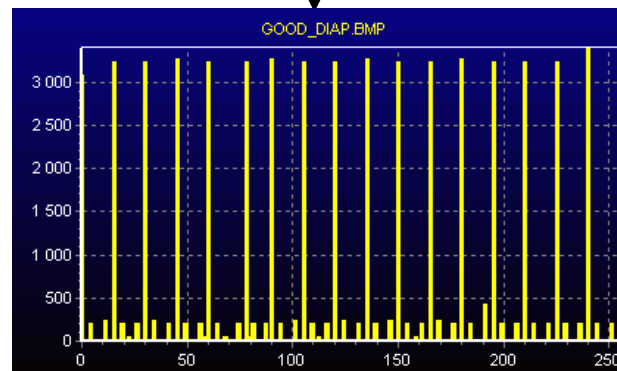
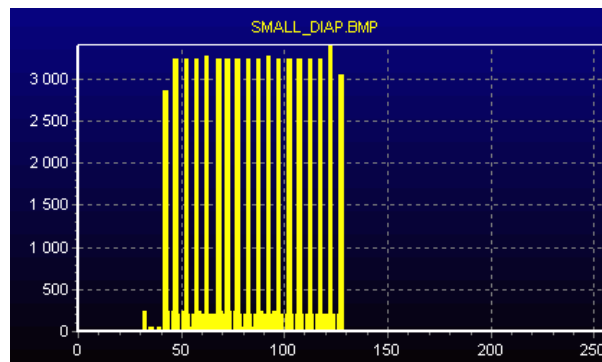
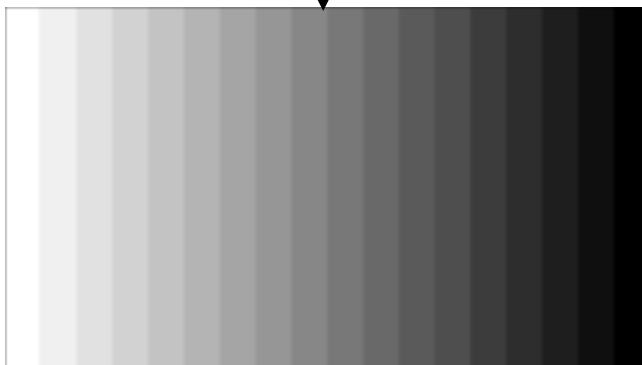


График функции $f^{-1}(y)$

Линейная коррекция яркости

Компенсация узкого диапазона яркостей – линейное растяжение:



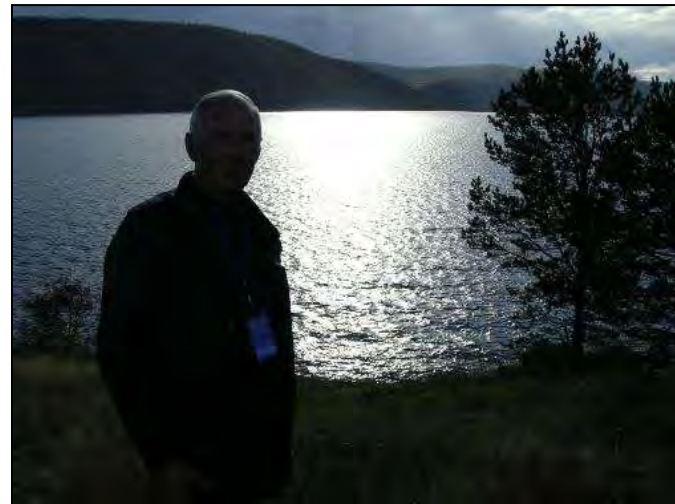
Линейная коррекция яркости

Линейное растяжение – «как AutoContrast в Photoshop»



Линейная коррекция

Линейная коррекция помогает не всегда!



Нелинейная коррекция

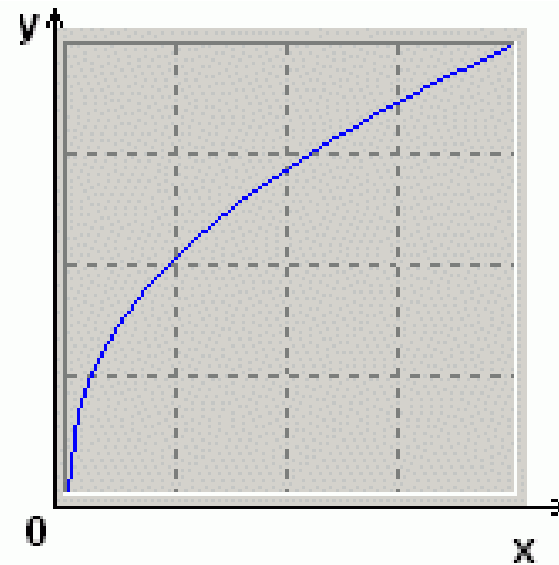


График функции $f^{-1}(y)$

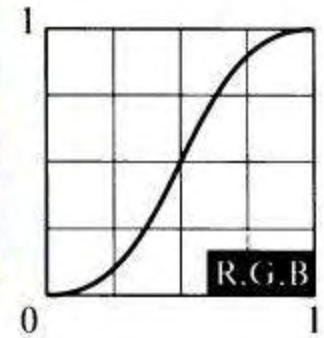
Яркостная коррекция



Малоконтрастное изображение



Результат коррекции



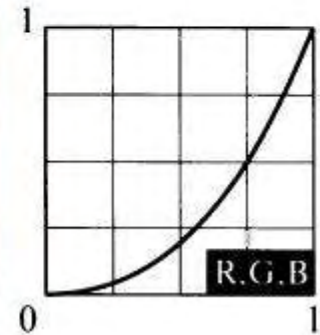
Яркостная коррекция



Светлое изображение



Результат коррекции



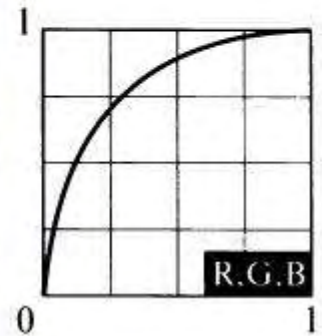
Яркостная коррекция



Темное изображение

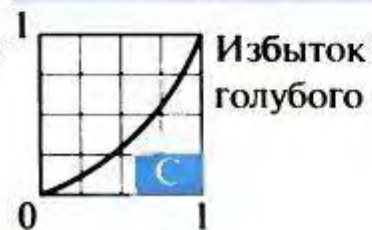
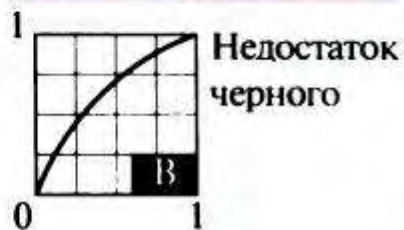
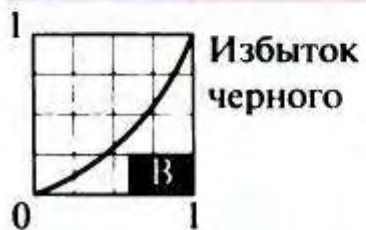


Результат коррекции





Цветовая коррекция



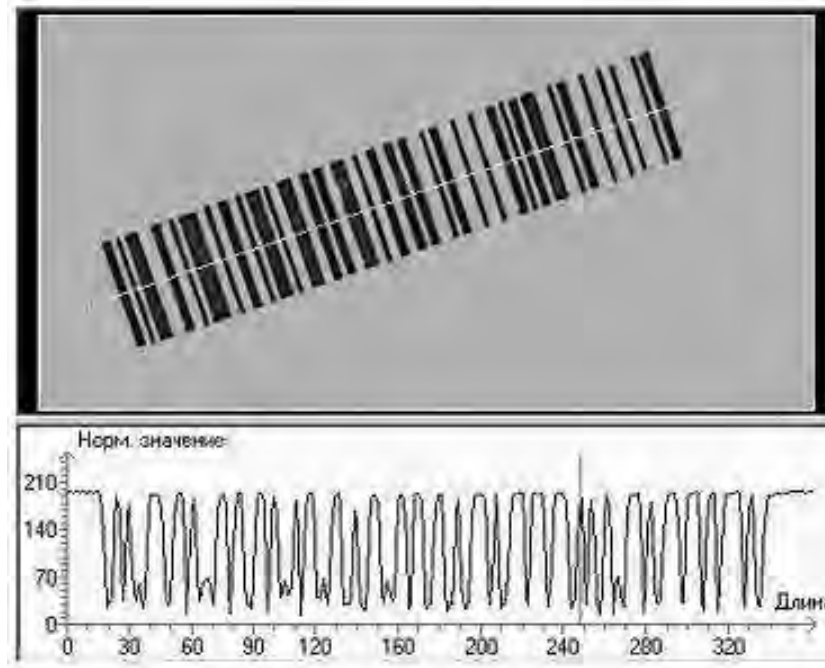
Раздельный анализ яркостной и геометрической составляющих

$$\langle x, y, I \rangle$$

Сведение к одномерным задачам

- гистограммы
- профили
- проекции

Профиль вдоль линии



Гистограмма изображения

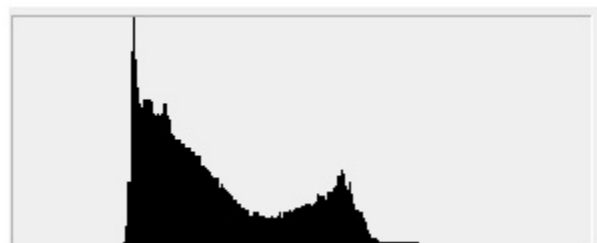


$$h(m) = |\{(r, c) \mid I(r, c) = m\}|$$

Яркостная нормализация изображения



Исходное



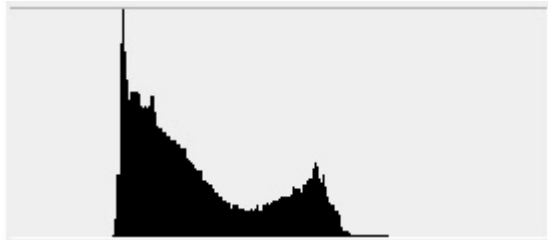
Нормализованное



Эквализация гистограммы



Исходное



С эквализованной гистограммой



Эквализация гистограммы

- Цель – максимальное соответствие равномерному закону распределения

Эквализация гистограммы

- Значение яркости 0:

$$\text{sum}(\text{hist}[i], i=0..i(0)) \leq q$$

- Значение яркости 1:

$$\begin{aligned} \text{sum}(\text{hist}[i], i=i(0)+1..i(1)) &\leq q && \text{или} \\ \text{sum}(\text{hist}[i], i=0..i(1)) &\leq 2q && \text{и т.д.} \end{aligned}$$

$$q = \frac{w * h}{N}, \quad N = 256$$

- Значение яркости n:

$$\begin{aligned} \text{sum}(\text{hist}[i], i=i(n-1)+1..i(n)) &\leq q && \text{или} \\ \text{sum}(\text{hist}[i], i=0..i(n)) &\leq nq \end{aligned}$$

Сравнение нормализации и эквализации



Нормализованное



С эквализованной гистограммой



```
const k = 255;
h: array [0 .. k] of double;

//построение гистограммы
for i := 0 to k do h[i] := 0;
for i := 0 to Height - 1 do
  for j := 0 to Width - 1 do
    h[round(k * Pixels[i, j])] := h[round(k * Pixels[i, j])] + 1;

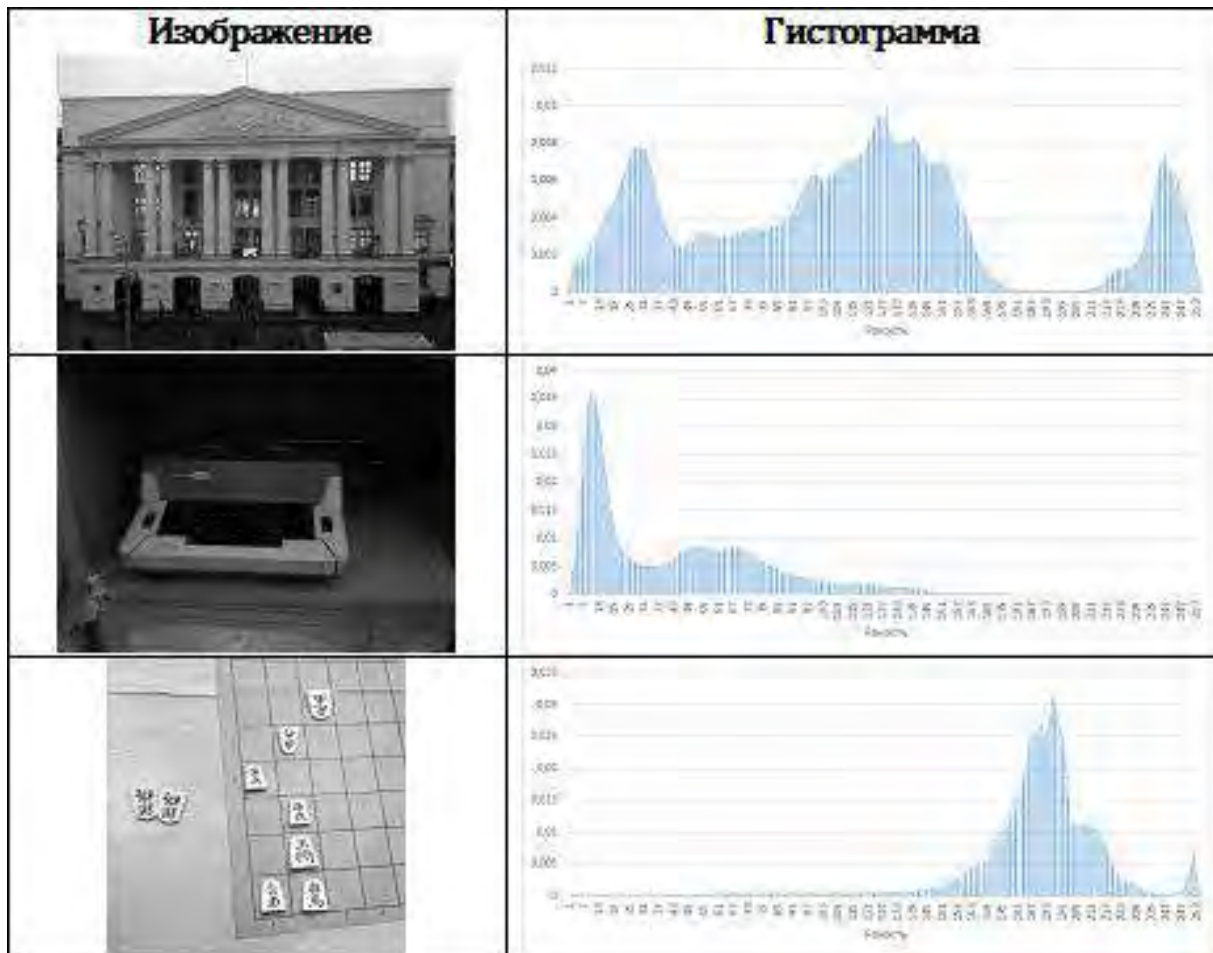
//нормирование гистограммы
for i := 0 to k do
  h[i] := h[i] / (Height * Width);

// построение гистограммы с накоплением
for i := 1 to k do
  h[i] := h[i - 1] + h[i];
```

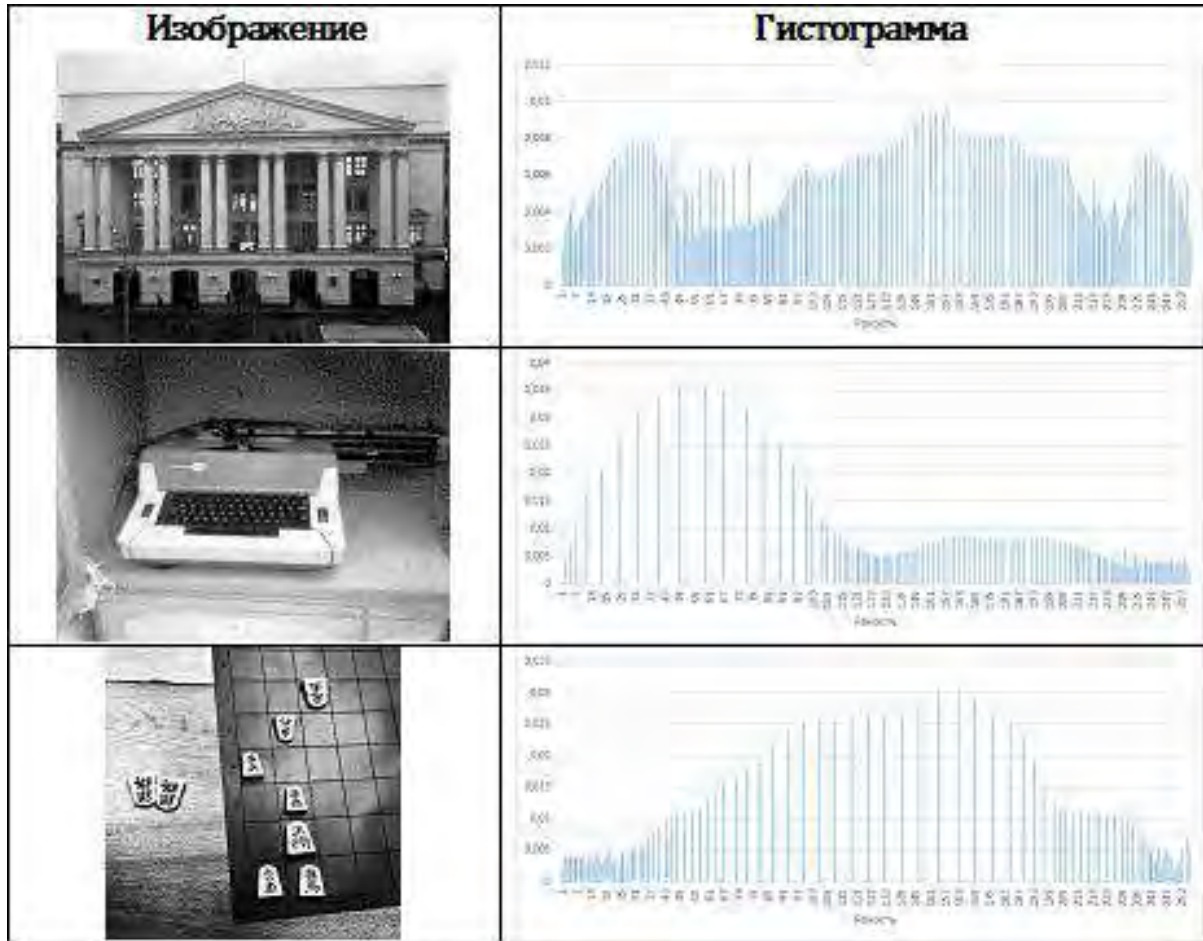


```
// равномерное распределение значений
for i := 0 to Height - 1 do
  for j := 0 to Width - 1 do
    Pixels[i, j] := h[round(k * Pixels[i, j])];
```

Изображения и их гистограммы



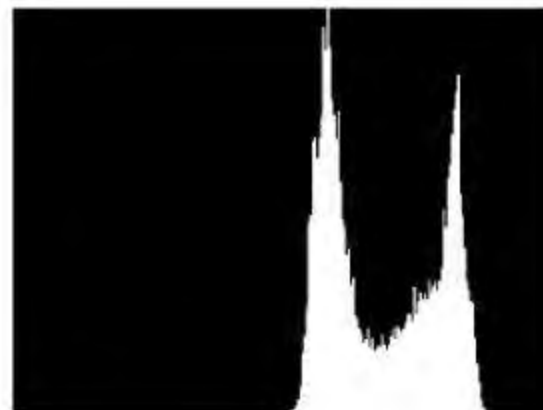
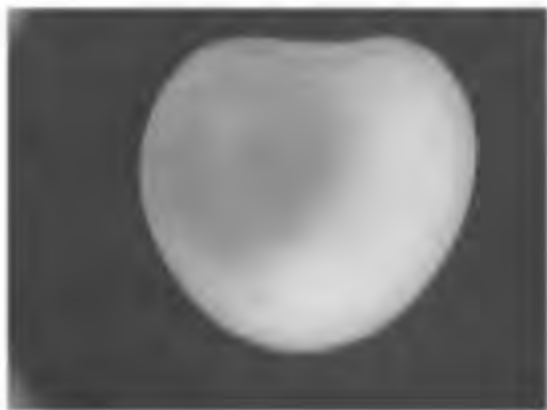
Изображения и их гистограммы после эквализации



Эквализация для различных цветковых моделей



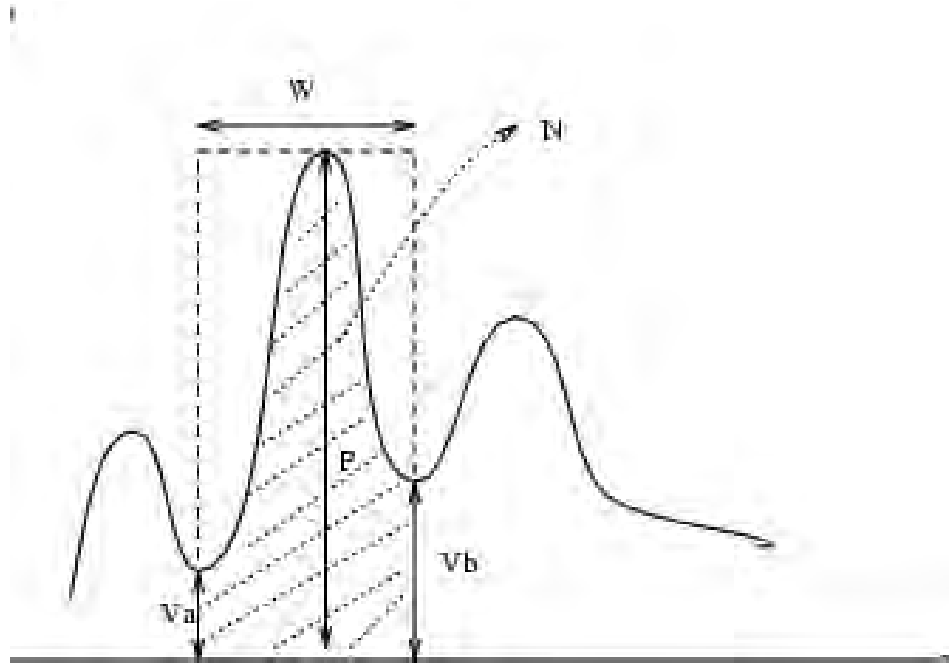
Бимодальная гистограмма



Поиск пиков в гистограмме

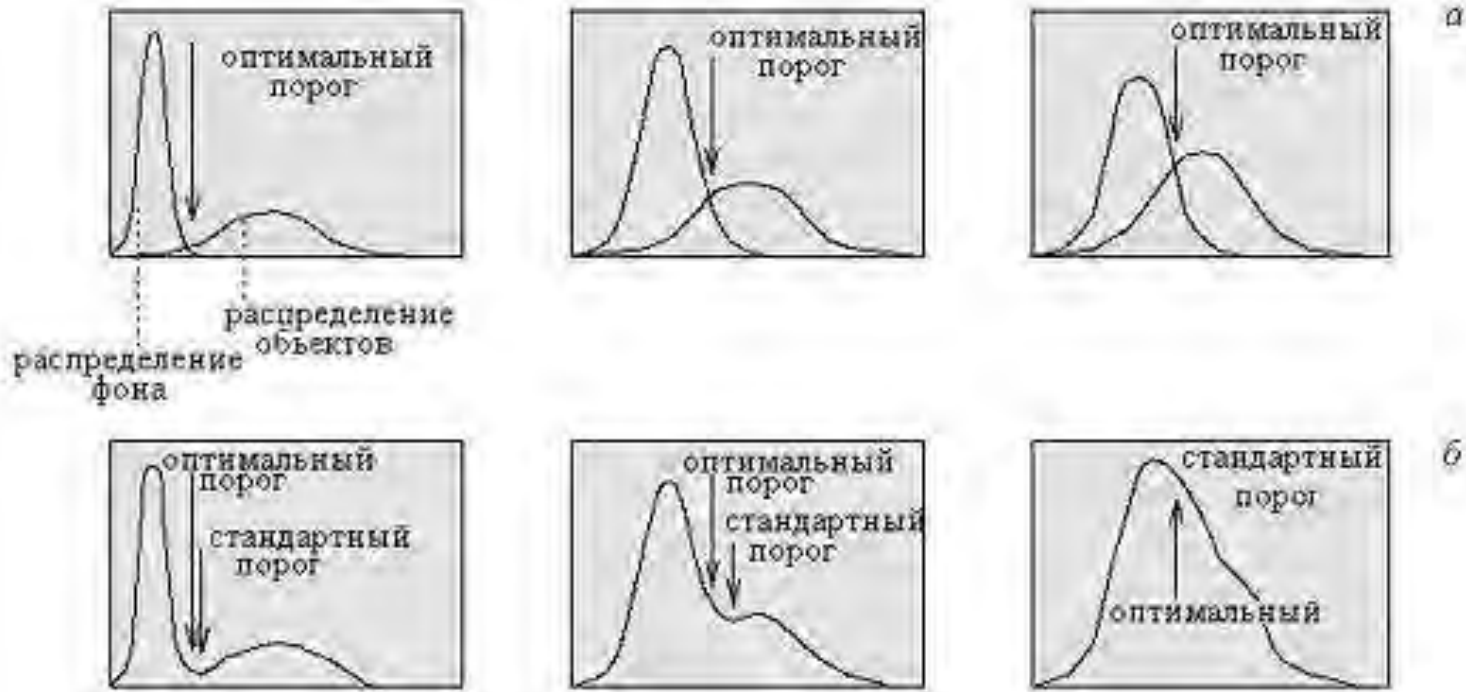
1. Найти соседние локальные максимумы в гистограмме g_i
2. Рассчитать меру «пиковости» для g_i
3. Отфильтровать пики со слишком маленькой «пиковостью».
4. Для оставшихся найти самые «низкие» точки между пиками – это и будут пороги.

Мера «пиковости»





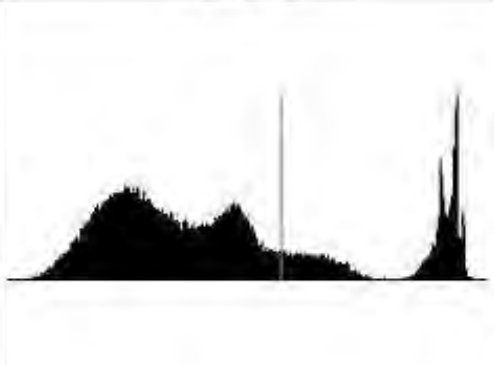


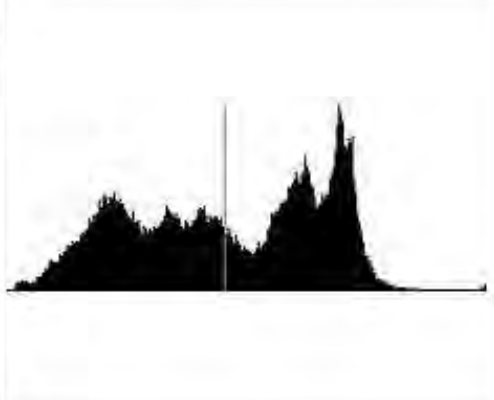


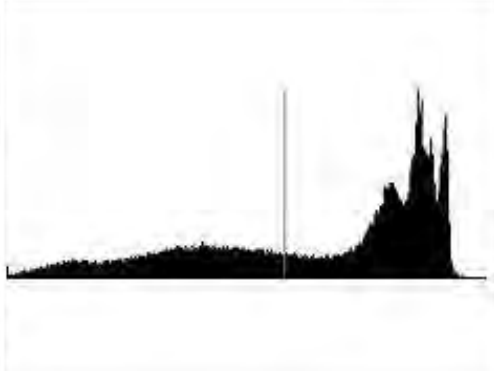
$$Peak = \left(1 - \frac{(V_a + V_b)}{2P} \right) \cdot \left(1 - \frac{N}{(W \cdot P)} \right)$$

Выбор порога бинаризации по гистограмме



а - функции распределения объекта и фона;

б - соответствующие гистограммы и оптимальный порог

Greyscale Image	Binary Image	Histogram
		
		
		

Метод Оцу (Otsu)

- Минимизация внутриклассовой дисперсии
- Максимизация межгрупповой дисперсии

$$sc(t) = 1 - \frac{disp(0, t) + disp(t + 1, \max t)}{disp(0, \max t)}$$

$$T = \arg \max_{t \in 0.. \max t} sc(t)$$



Метод Оцу (Otsu) 1

Минимизация внутригрупповой дисперсии

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

q_1 и q_2 — вероятности первого и второго классов соответственно.

$$\sigma_1^2(t) = \sum_{i=1}^t \frac{[i - \mu_1(t)]^2 P(i)}{q_1(t)} \quad q_1(t) = \sum_{i=1}^t P(i) \quad \mu_1(t) = \sum_{i=1}^t \frac{iP(i)}{q_1(t)}$$

$$\sigma_2^2(t) = \sum_{i=t+1}^T \frac{[i - \mu_2(t)]^2 P(i)}{q_2(t)} \quad q_2(t) = \sum_{i=t+1}^T P(i) \quad \mu_2(t) = \sum_{i=t+1}^T \frac{iP(i)}{q_2(t)}$$

Метод Оцу (Otsu) 2

Максимизация межгрупповой дисперсии

$$\sigma_B^2(t) = q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2$$

$$q_1(t) = \sum_{i=1}^t P(i)$$

$$q_2(t) = \sum_{i=t+1}^T P(i)$$

$$\mu_1(t) = \sum_{i=1}^t \frac{iP(i)}{q_1(t)}$$

$$\mu_2(t) = \sum_{i=t+1}^T \frac{iP(i)}{q_2(t)}$$

Метод Оцу (Otsu) - Быстрый

Рекуррентные соотношения

$$q_1(t+1) = q_1(t) + P(t+1) \quad \mu = q_1(t)\mu_1(t) + q_2(t)\mu_2(t)$$

$$q_1 = P(1)$$

$$\mu_1(t+1) = \frac{q_1(t)\mu_1(t) + (t+1)P(t+1)}{q_1(t+1)} \quad \mu_1(0) = 0$$

$$\mu_2(t+1) = \frac{\mu - q_1(t+1)\mu_1(t+1)}{1 - q_1(t+1)}$$

Быстрый алгоритм метода Оцу

- Вычисляем гистограмму (один проход через массив пикселей). Дальше нужна только гистограмма; проходов по всему изображению больше не требуется.
- Начиная с порога $t = 1$, проходим через всю гистограмму, на каждом шаге пересчитывая дисперсию $\sigma_b(t)$. Ищем максимум по $\sigma_b(t)$, запоминая $T = t$.
- Искомый порог равен T .

Пороговое значение определено методом Оцу



a) original image



b) pixels below 93



c) pixels above 93

Метод Оцу (Otsu)

- Достоинства метода Оцу:
 - простота реализации;
 - быстрое время выполнения (требуется $O(N)$ операций, где N — количество пикселей в изображении);
 - нет необходимости подбирать какие-либо коэффициенты, размеры окон для прохода по изображению и проч. (в MatLab это функция `graythresh()` без аргументов);
 - дает предсказуемые результаты.
- Недостатки метода Оцу:
 - не применим для сложных изображений с графикой, цветными надписями, различными градиентами;
 - пороговая бинаризация чувствительна к неравномерной яркости изображения. Решением такой проблемы может быть введение локальных порогов, вместо одного глобального.

Литература для метода Оцу

- 1. N. Otsu. A threshold selection method from gray-scale histogram. IEEE Transactions on System, Man, and Cybernetics. 9, 62–66, 1979.
- 2. PING-SUNG LIAO, TSE-SHENG CHEN AND PAU-CHOO CHUNG. Fast Algorithm for Multilevel Thresholding. JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 17, 713-727 (2001)

Пространственная обработка изображений

Лекция 5

Обработка изображений

- В пространственной области
- В частотной области

Пространственная обработка

- Яркостные преобразования
 - Оперируют отдельными пикселями
 - Управляют контрастом и «цветностью»
- Пространственная фильтрация
 - Оперирует окрестностью точки
 - Управляет резкостью

Процессы пространственной обработки

$$g(x, y) = T[f(x, y)]$$

- Одно изображение или набор изображений
- Окрестности разной формы и разного размера

Арифметические операции

$$g(x, y) = \text{Op}(f_1(x, y), f_2(x, y))$$

for all pixel positions x, y :

$$\text{out}[x, y] = \text{func}(\text{image1}[x, y], \text{image2}[x, y])$$

Addition / Blending

$$g(x, y) = f_1(x, y) + f_2(x, y)$$



+



=



$$g(x, y) = \alpha f_1(x, y) + \beta f_2(x, y)$$

Subtraction

$$g(x, y) = f_1(x, y) - f_2(x, y)$$



-



=

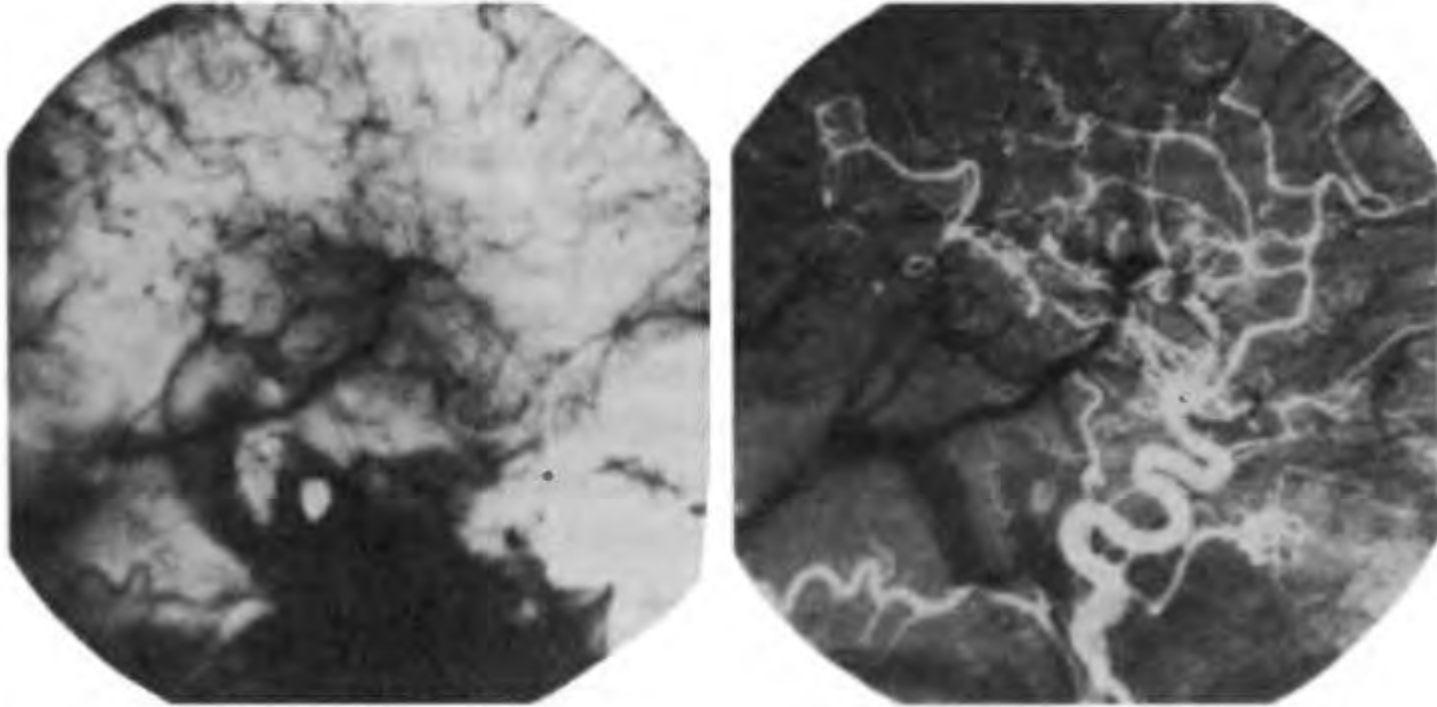


$$g(x, y) = |f_1(x, y) - f_2(x, y)|$$

Улучшение изображения путем вычитания

Enhancement by image subtraction

Digital Subtraction Angiography



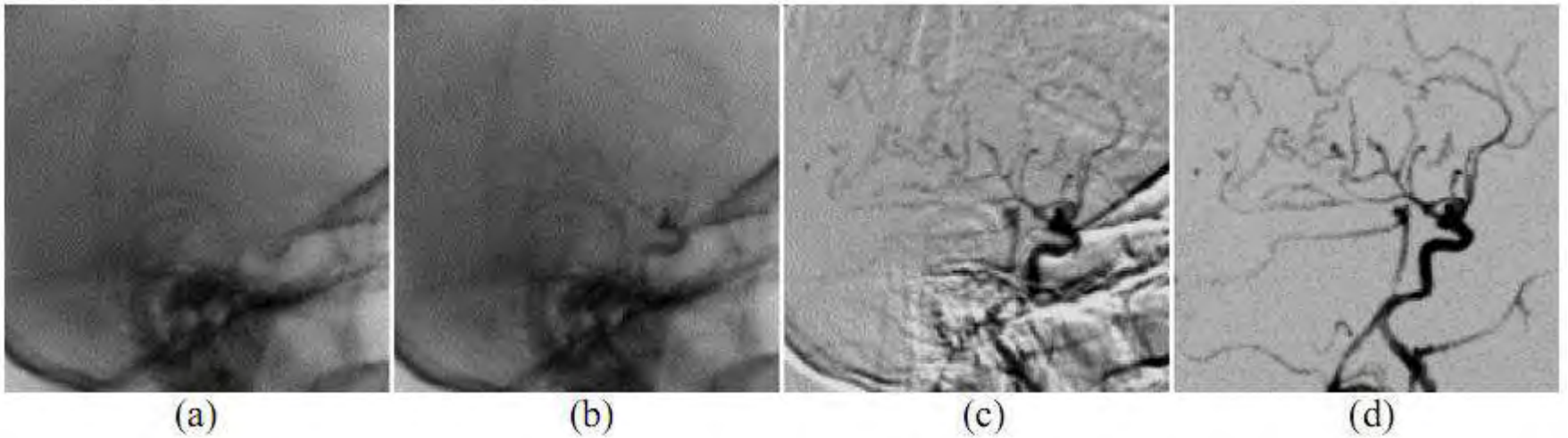
Диапазон от -255 до 255

Требуется масштабирование

Два основных варианта масштабирования

1. $(i+255)/2$ - быстрый, неточный
2. Более точный
 1. min разности
 2. $i - \min$
 3. max модифицированного изображения
 4. $i * 255 / \max$

Digital Subtraction Angiography



(a) original “mask”

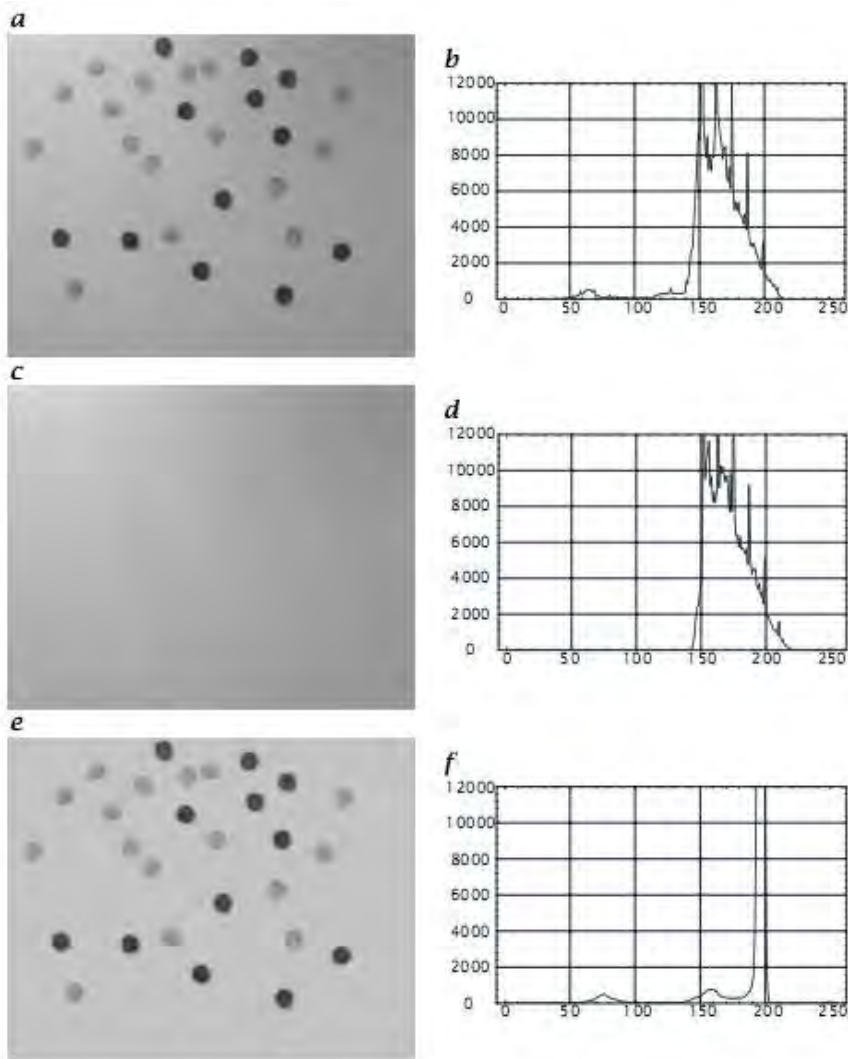
(b) with contrast

(c) Subtraction of (a) and (b)

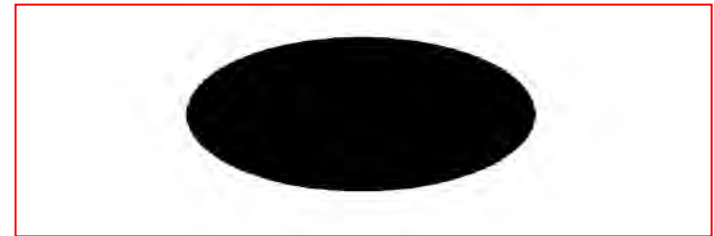
(d) Subtraction to compensate for motion.

Коррекция неоднородной освещенности делением изображения на опорное

$$G' = c \cdot G / R$$



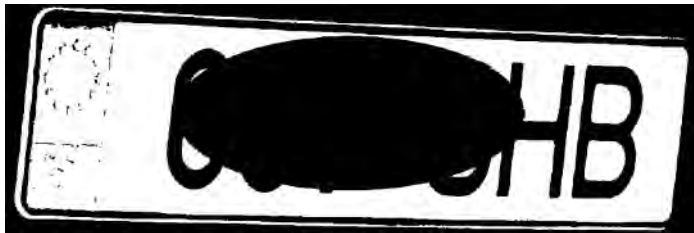
Логические операции



b



NOT(a)

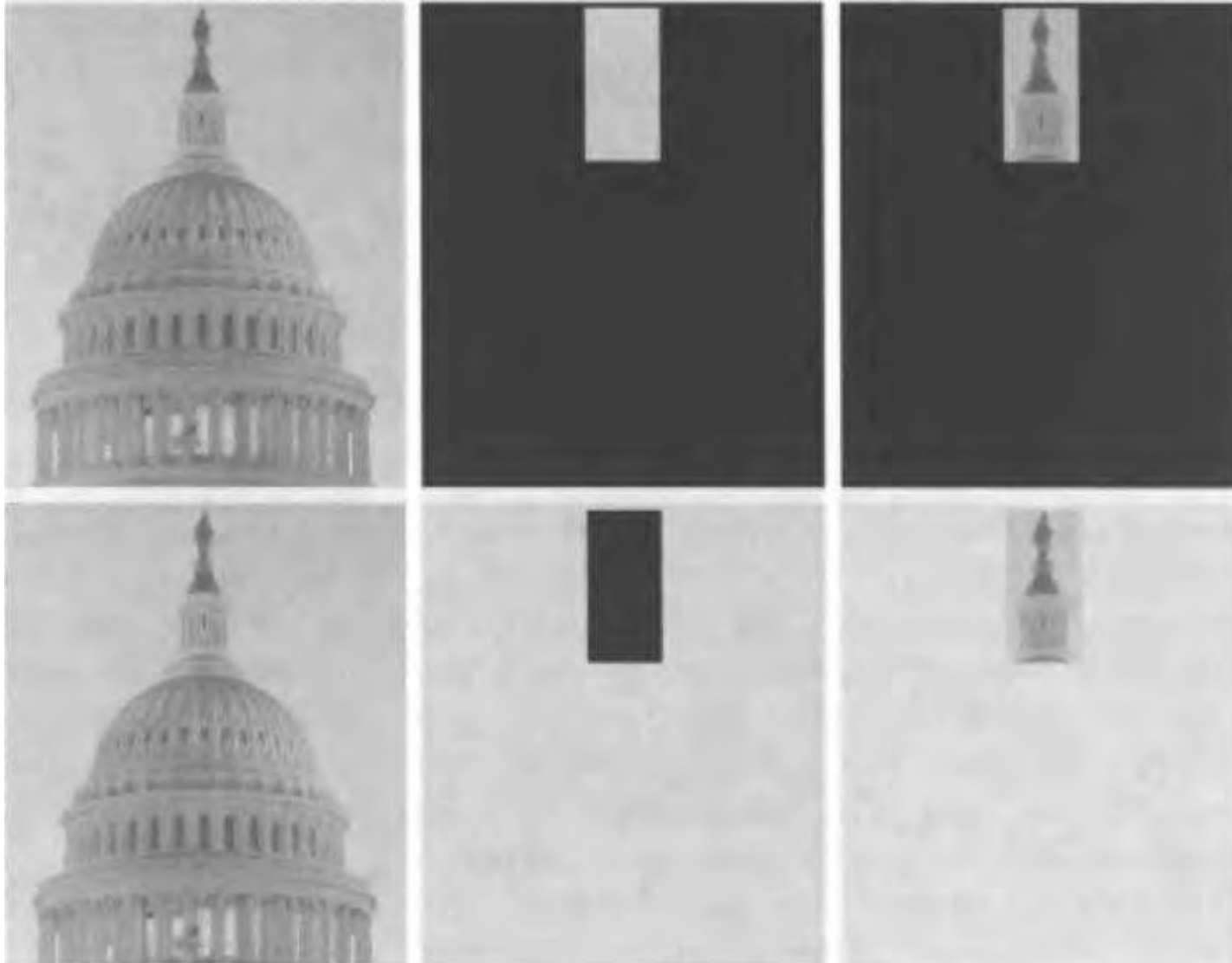


a . b



a + b

Маскирование (AND и OR)



Виды шума



Гауссов



Импульсный (соль.перец)

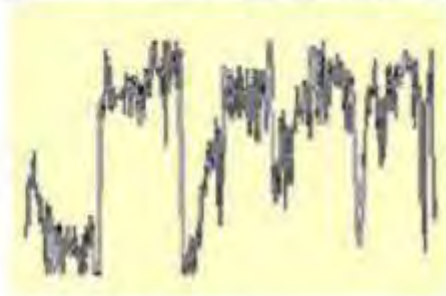
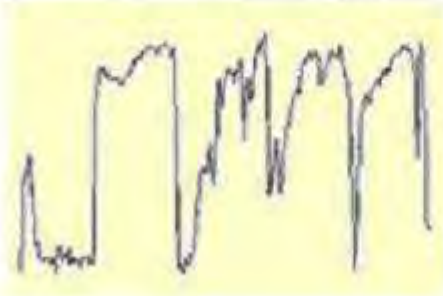


смешанный



Гауссов шум

Image
Noise



Шум
некоррелирован и
имеет нулевое
среднее

$$f(x, y) = \overbrace{\hat{f}(x, y)}^{\text{Ideal Image}} + \overbrace{\eta(x, y)}^{\text{Noise process}}$$

Gaussian i.i.d. ("white") noise:
 $\eta(x, y) \sim \mathcal{N}(\mu, \sigma)$

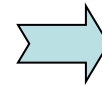
Source: M. Hebert

Усреднение кадров

10 зашумленных изображений



+ +



Усреднение по 10 изображениям

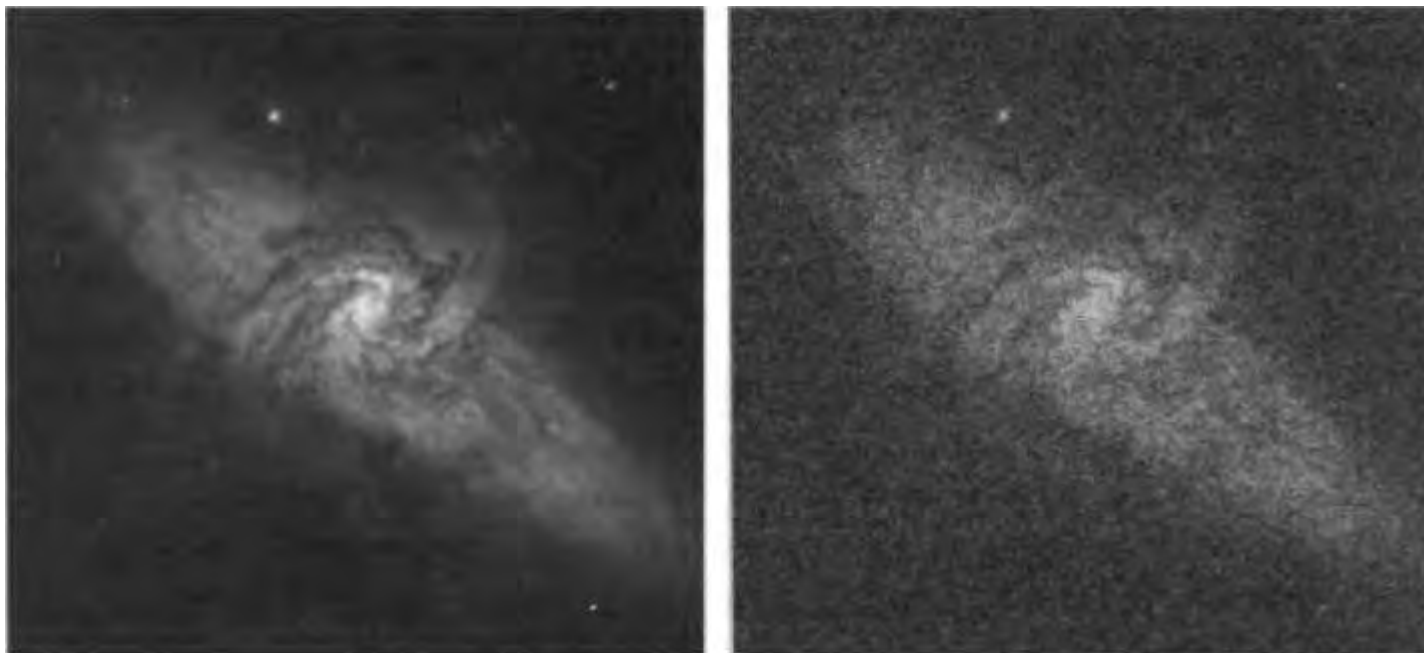


$$I(i, j) = g_r(i, j) + Err(i, j);$$

$$\bar{I}(i, j) = \frac{1}{N} \sum_{k=1}^N I_k(i, j);$$

$$E(\bar{I}(i, j)) = g_r(i, j);$$

Исходное и шум



Результаты усреднения

K=8



K=16



K=64



K=128

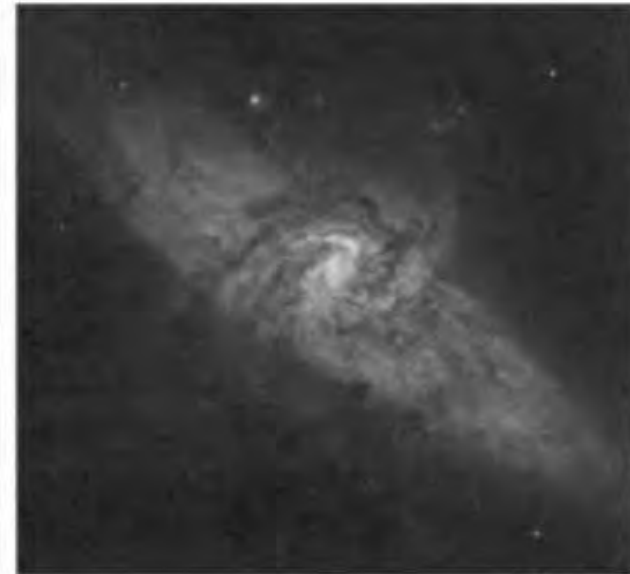
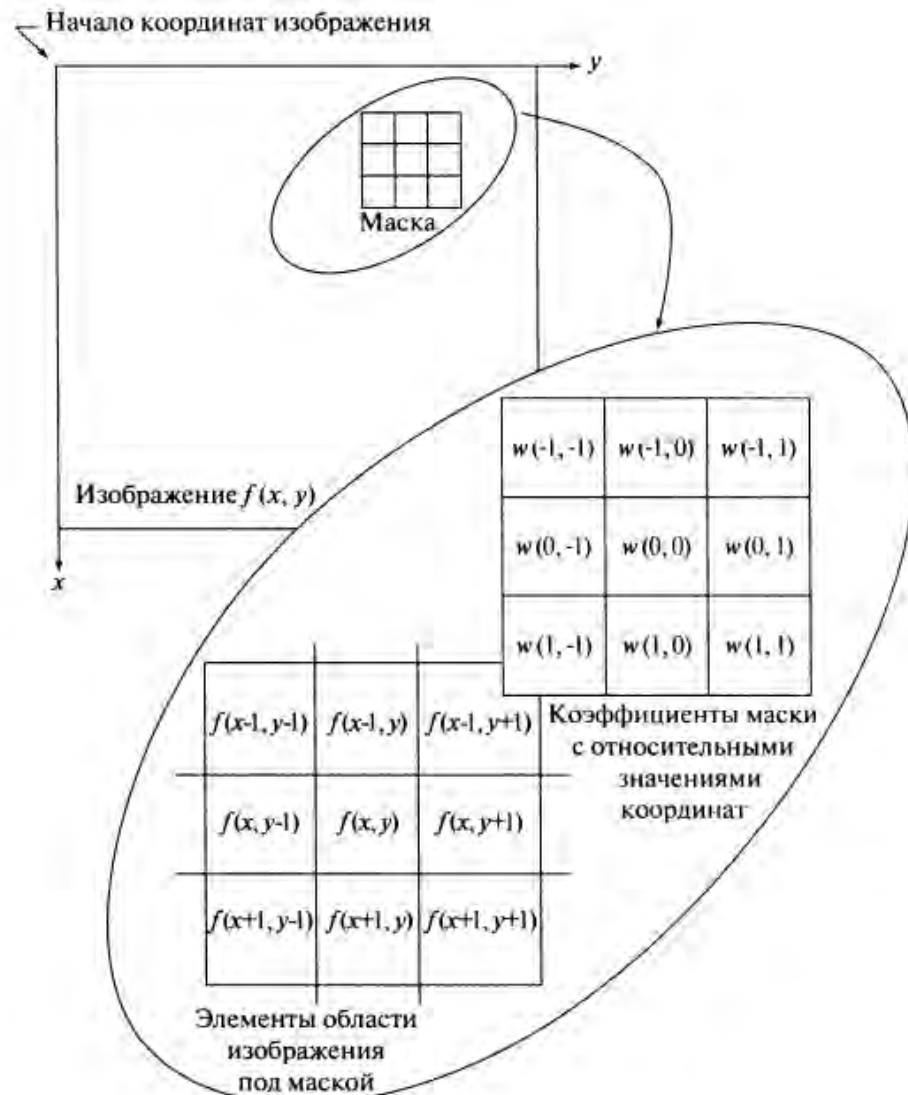


Схема пространственной фильтрации

Фильтр задается матрицей

- фильтр
- маска
- ядро
- шаблон
- окно
- оператор



- Размер окрестности
- Положение пикселя относительно окна
- Симметрия

Решение проблемы краевых пикселей

- Экстраполяция значений краевых пикселей
- Зеркальное отражение исходного изображения относительно каждого края, и центральное по углам
- Тесселяция исходного изображения /Использование цикличности/
- Игнорирование /Меньшее изображение/
- Использование части маски
- Добавление пограничных областей

Фильтрация

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)$$

корреляция

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x - s, y - t)$$

свертка

Свертка маски с изображением

Основные свойства

- Линейность:
 $\text{filter}(f_1 + f_2) = \text{filter}(f_1) + \text{filter}(f_2)$
- Инвариантность к сдвигу -- не зависит от сдвига пиксела:
 $\text{filter}(\text{shift}(f)) = \text{shift}(\text{filter}(f))$
- Любой линейный оператор, инвариантный к сдвигу, может быть записан в виде свертки

Свойства

- Коммутативность: $a * b = b * a$
 - Нет никакой разницы между изображением и ядром фильтра
- Ассоциативность: $a * (b * c) = (a * b) * c$
 - Последовательное применение фильтров: $((a * b_1) * b_2) * b_3$ эквивалентно применению такого фильтра: $a * (b_1 * b_2 * b_3)$
- Дистрибутивность по сложению:
 $a * (b + c) = (a * b) + (a * c)$
- Домножение на скаляр можно вынести за скобки:
 $ka * b = a * kb = k(a * b)$
- Единица: $e = [\dots, 0, 0, 1, 0, 0, \dots]$, $a * e = a$

Простейшие фильтры



Original

0	0	0
0	0	1
0	0	0

?

Source: D. Lowe

Простейшие фильтры



Original

0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel

Source: D. Lowe

Простейшие фильтры



Original

$$\frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

?

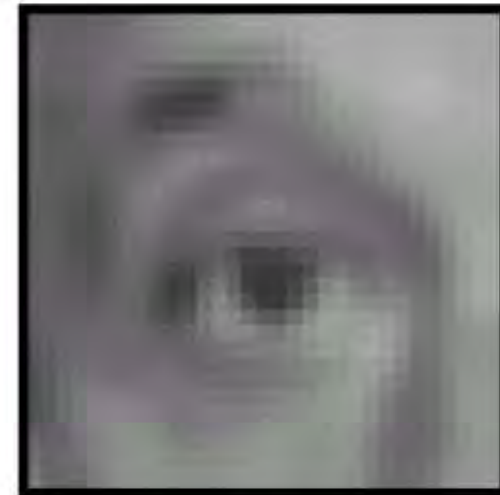
Source: D. Lowe

Простейшие фильтры



Original

$$\frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$



Blur (with a
box filter)

Source: D. Lowe

Сглаживающие пространственные фильтры

Применяются для:

расфокусировки изображения и подавления шума

Выделяют:

линейные и нелинейные

Линейные сглаживающие фильтры

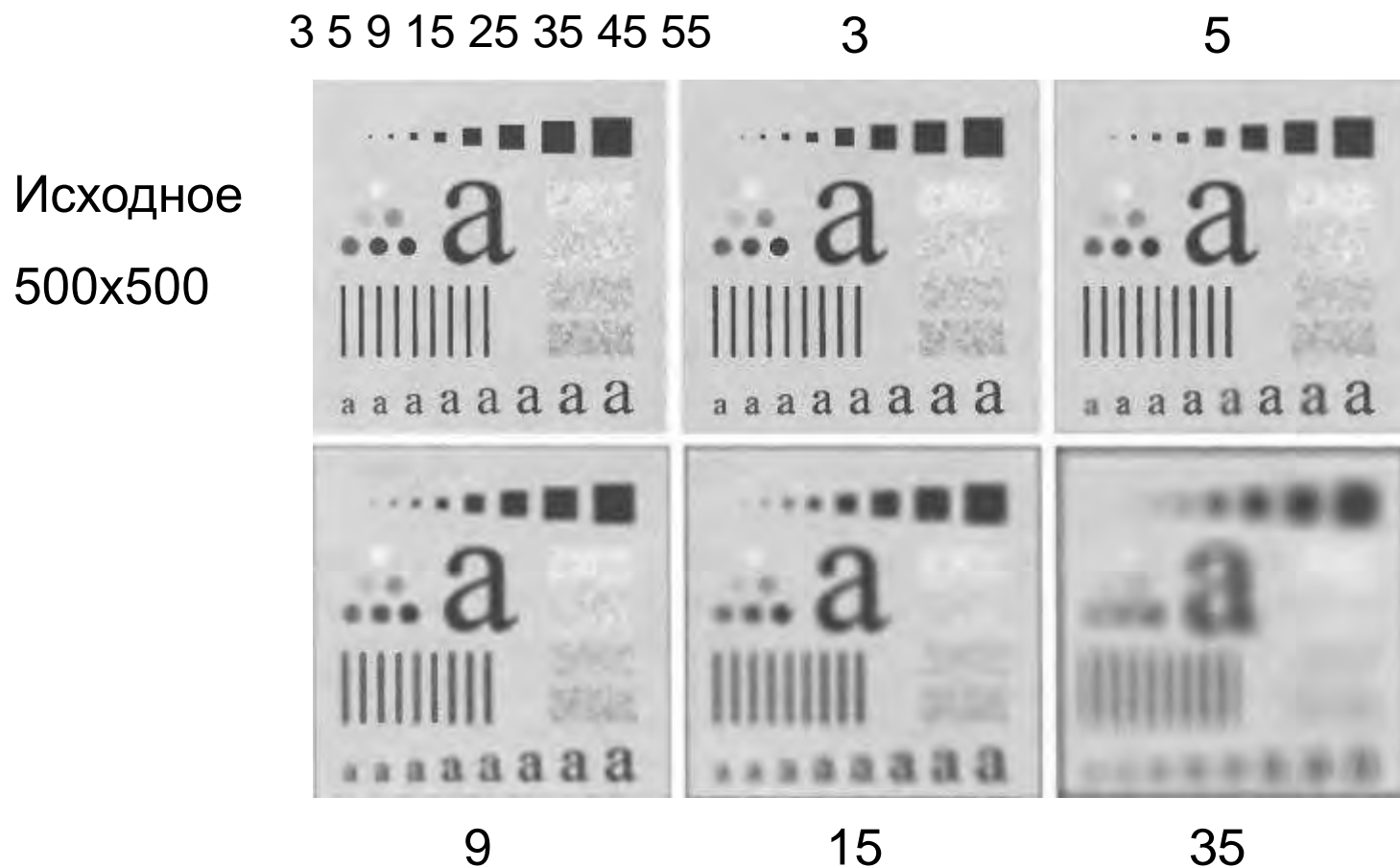
Низкочастотные

$$\frac{1}{9} \times$$

1	1	1
1	1	1
1	1	1

Однородный усредняющий фильтр

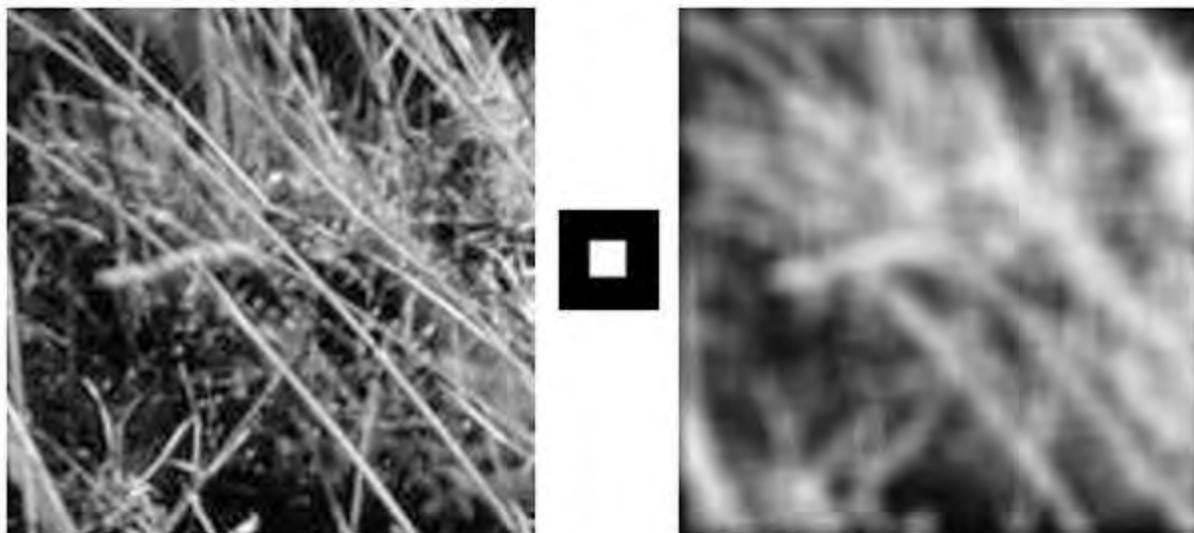
Результаты сглаживания усредняющим фильтром



Сглаживающая маска + пороговое обнаружение

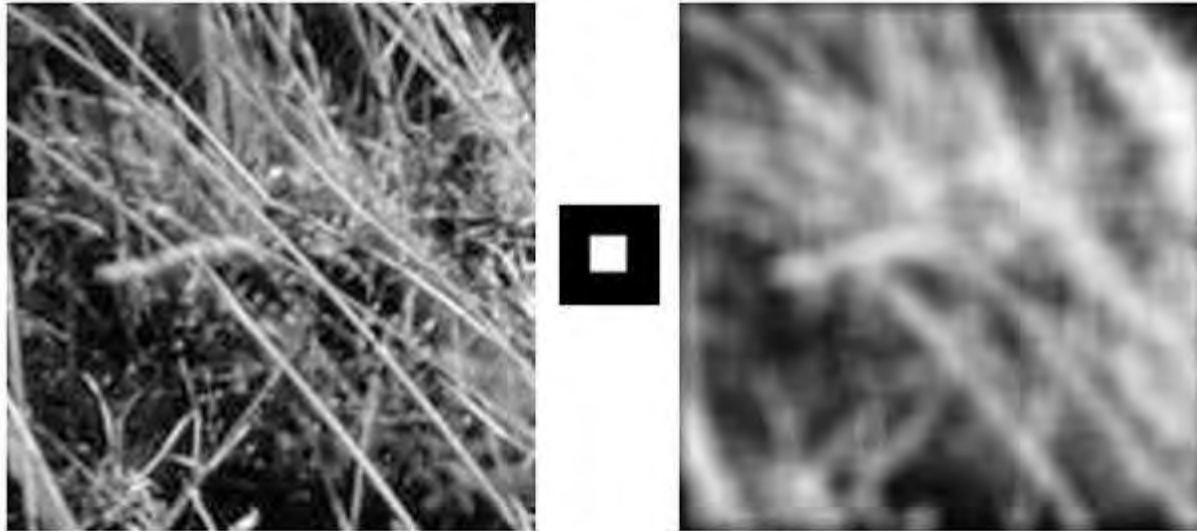


Сглаживание с box-фильтром



Что вы видите?

Сглаживание с box-фильтром



При сглаживании с box-фильтром на изображении могут образовываться паразитные линии

Сглаживание

- Точка света, наблюдаемая с расфокусированного объектива, выглядит как кружок света, а усреднение дает квадратик
- Другой способ: взвешиваем вклад пикселей по окрестности с учетом близости к центру:



fuzzy blob

Фильтр взвешенного среднего

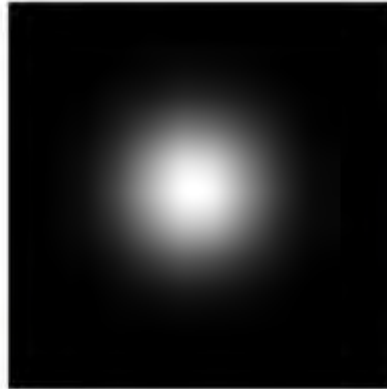
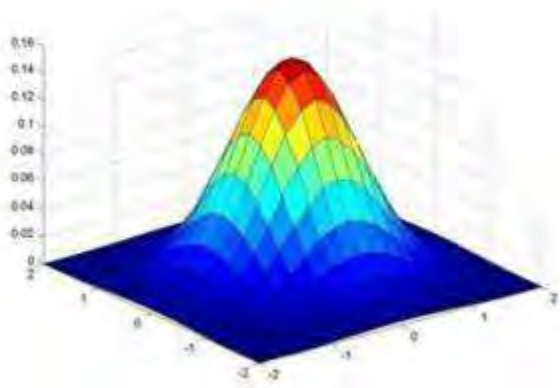
$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x+s, y+t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$$

 $\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

Гауссов фильтр

$$h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x+y)^2}{2\sigma^2}}$$



0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, $\sigma = 1$

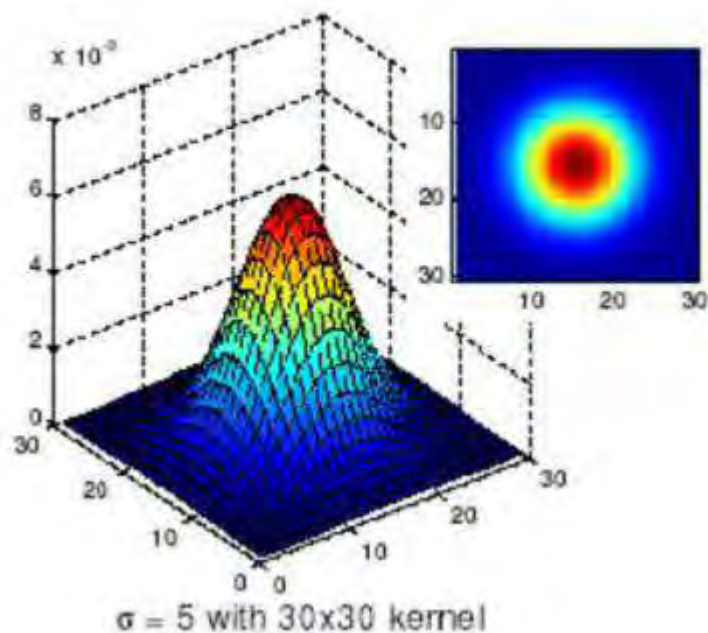
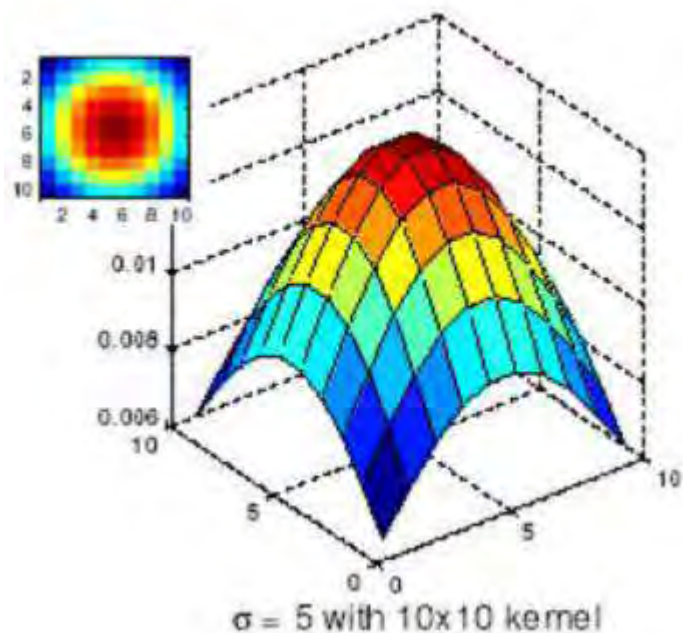
Source: C. Rasmussen

Свертка по функции:

$$I'(i, j) = \sum_{l=-n}^n \sum_{k=-m}^m I(i+l)(j+k) \cdot \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{d^2}{2\sigma^2}}$$

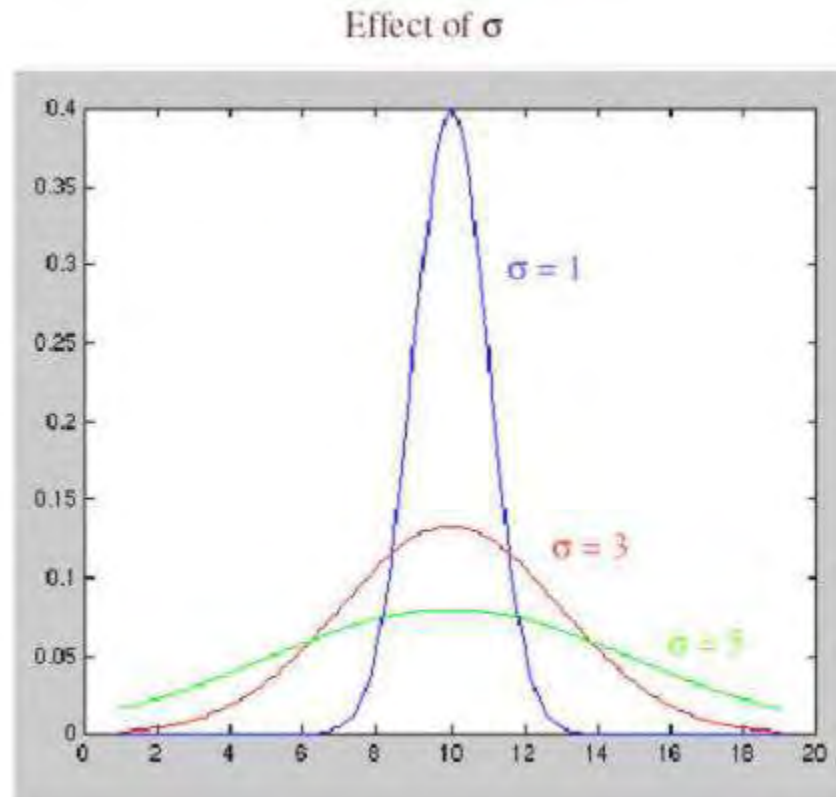
$$d = \sqrt{l^2 + k^2}$$

Выбор размера ядра



Source: K. Grauman

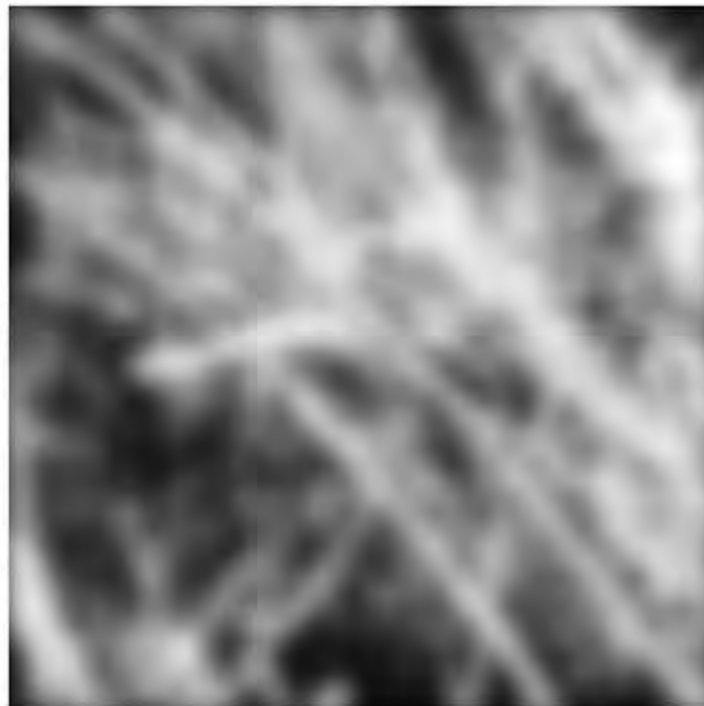
Выбор размера ядра



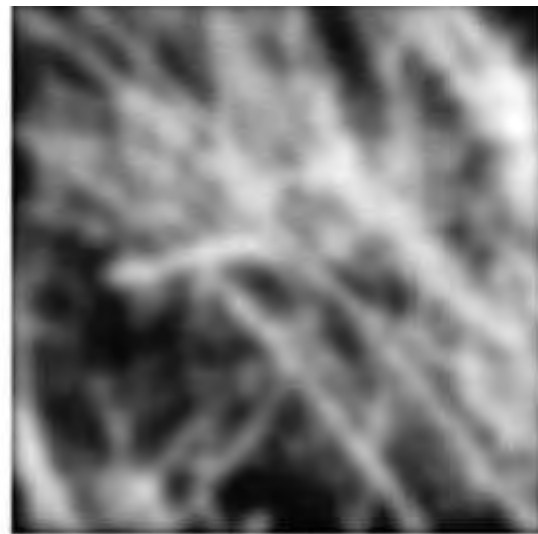
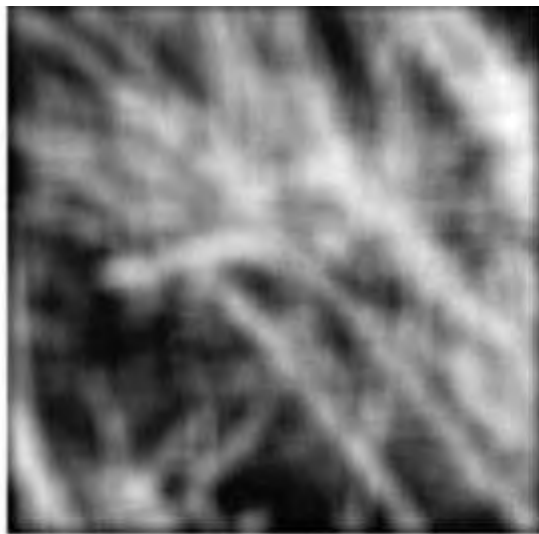
Полуразмер
фильтра = 3σ

Source: S. Lazebnik

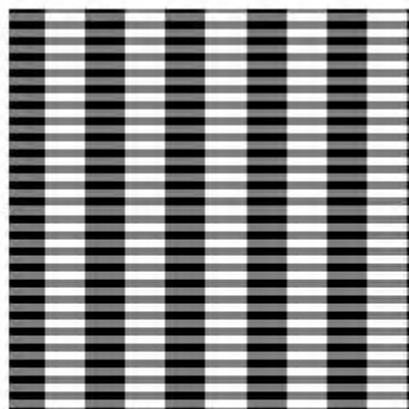
Сглаживание методом Гаусса



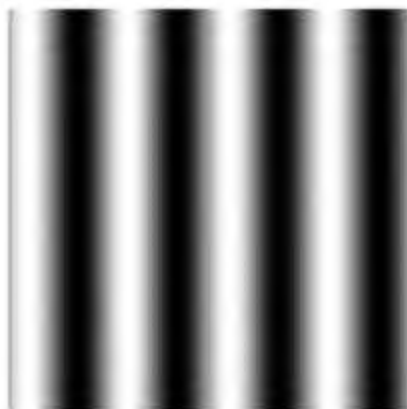
Сравнение



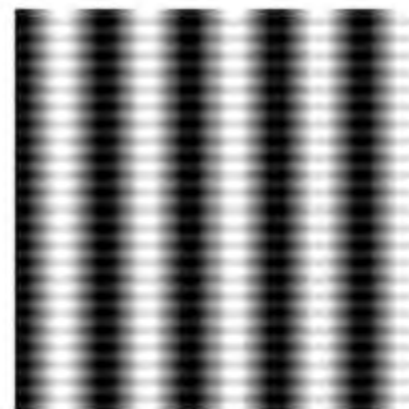
Результаты свертки по функции Гаусса и по константной функции (усреднения).



Исходное изображение



Фильтр Гаусса с
Sigma = 4



Усреднение по 49
пикселям (7x7)

Свойства фильтра Гаусса

- Сглаживание несколько раз фильтром с маленьким ядром дает результат сглаживания с большим ядром
- Свертка 2 раза фильтром радиуса σ дает тот же результат, что один раз фильтром радиуса $\sigma\sqrt{2}$

Сепарабельность

$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \right) \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{2\sigma^2}} \right)$$

Пример

2D свертка

$$\begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 2 & 3 & 3 \\ \hline 3 & 5 & 5 \\ \hline 4 & 4 & 6 \\ \hline \end{array} =$$

Фильтр раскладывается
в произведение двух 1D
фильтров:

$$\begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} = \begin{array}{|c|} \hline 1 \\ \hline 2 \\ \hline 1 \\ \hline \end{array} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline \end{array}$$

Свертка по строкам:

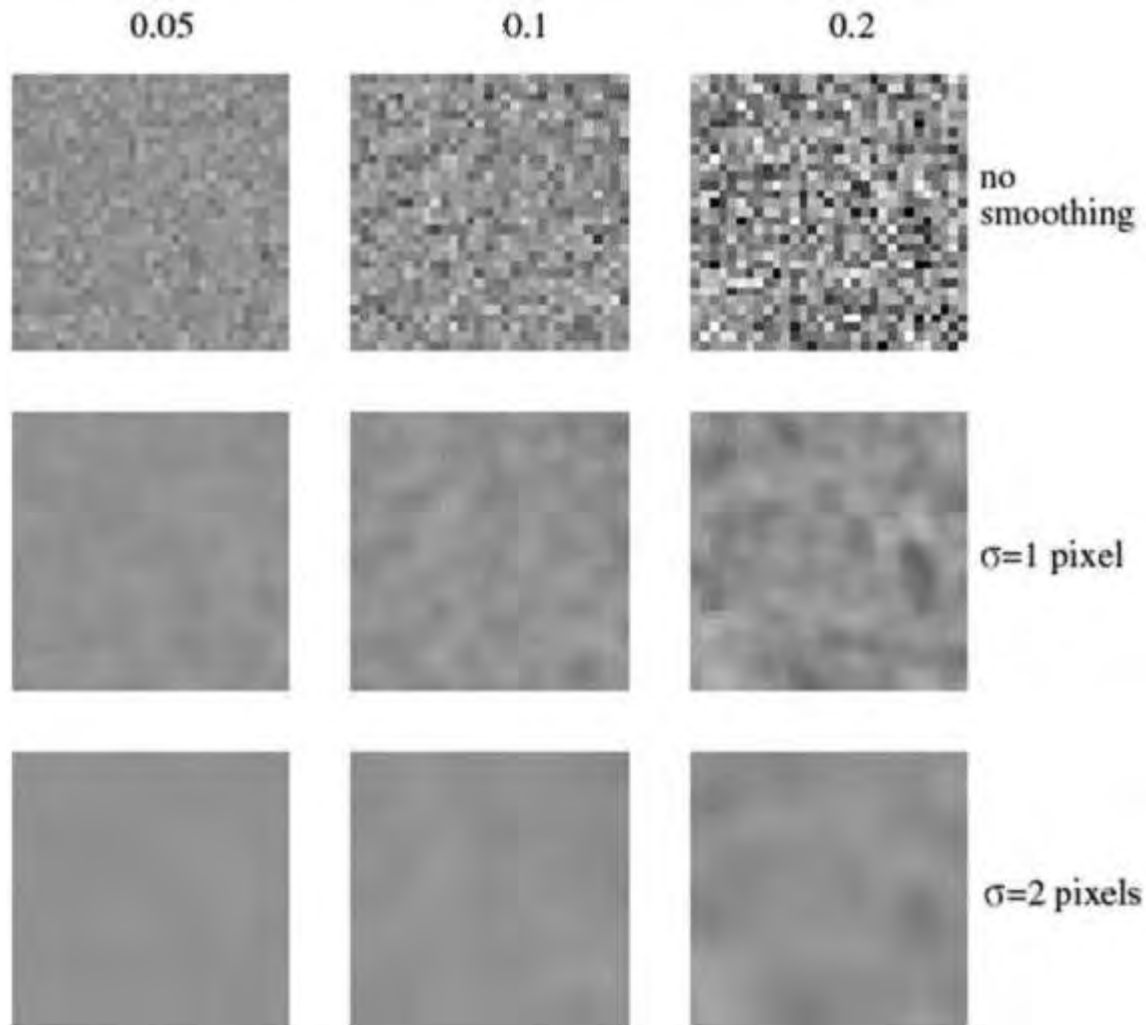
$$\begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 2 & 3 & 3 \\ \hline 3 & 5 & 5 \\ \hline 4 & 4 & 6 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline & 11 & \\ \hline & 18 & \\ \hline & 18 & \\ \hline \end{array}$$

Затем свертка по столбцу:

$$\begin{array}{|c|} \hline 1 \\ \hline 2 \\ \hline 1 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline & 11 & \\ \hline & 18 & \\ \hline & 18 & \\ \hline \end{array} =$$

Source: K. Grauman

Подавление гауссова шума



Подавление шума «соль и перец»

3x3



5x5



7x7



Чем результат плох?

Slide by S. Lazebnik

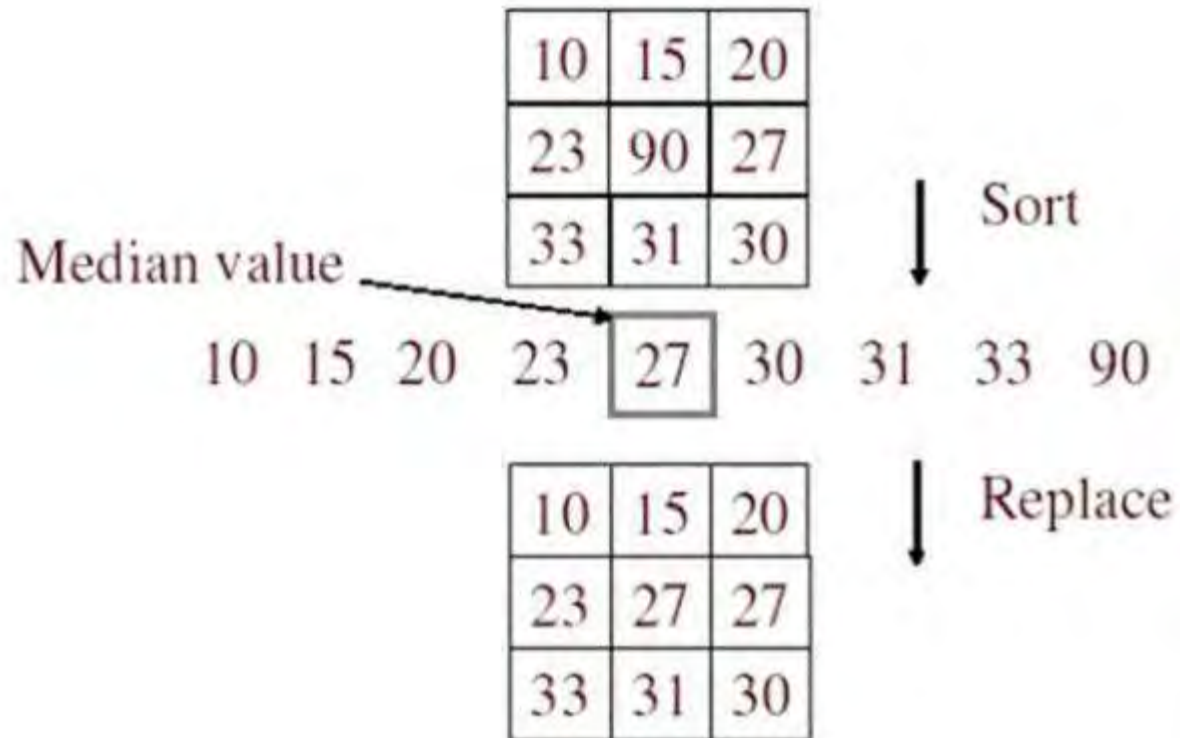
Нелинейное усреднение

Фильтры, основанные на порядковых статистиках:

- классический медианный
- максимума
- минимума

Эффективны при фильтрации импульсных шумов (соль и перец)
– значительно меньший эффект расфокусировки

Медианный фильтр



Source: K. Grauman

Пример медианного фильтра

$$x = [2 \ 80 \ 6 \ 3]$$

$$y[1] = \text{медиана}[2 \ 2 \ 80] = 2$$

$$y[2] = \text{медиана}[2 \ 80 \ 6] = \text{медиана}[2 \ 6 \ 80] = 6$$

$$y[3] = \text{медиана}[80 \ 6 \ 3] = \text{медиана}[3 \ 6 \ 80] = 6$$

$$y[4] = \text{медиана}[6 \ 3 \ 3] = \text{медиана}[3 \ 3 \ 6] = 3$$

и в итоге:

$$y = [2 \ 6 \ 6 \ 3] \text{ — выход медианного фильтра}$$

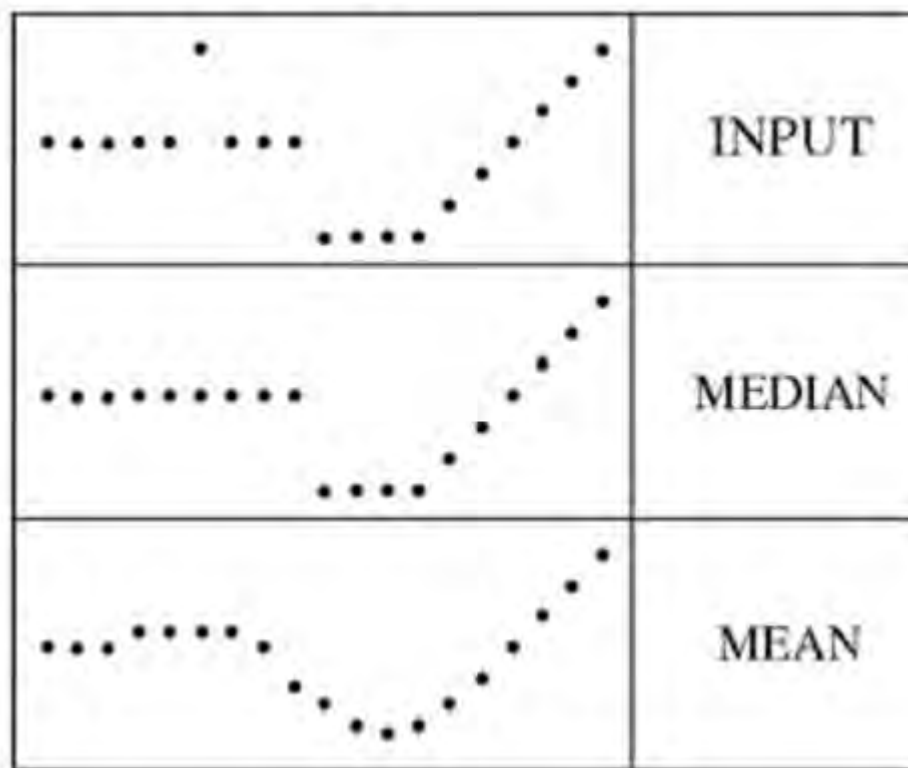
Нелинейность медианного фильтра

$$M([\dots 0 1 0 0 \dots] + [\dots 0 0 1 0 \dots]) = [\dots 0 1 1 0 \dots]$$

$$M[\dots 0 1 0 0 \dots] + M[\dots 0 0 1 0 \dots] = [\dots 0 0 0 0 \dots]$$

В чем преимущество медианного фильтра перед фильтром Гаусса?

filters have width 5 :



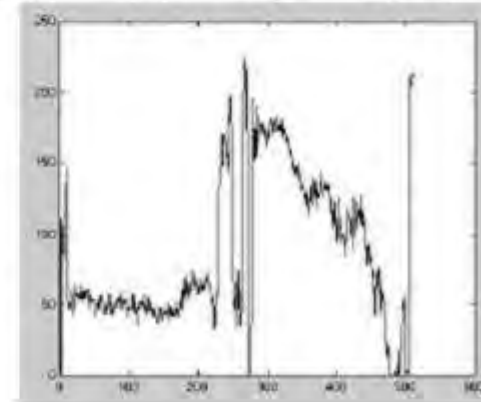
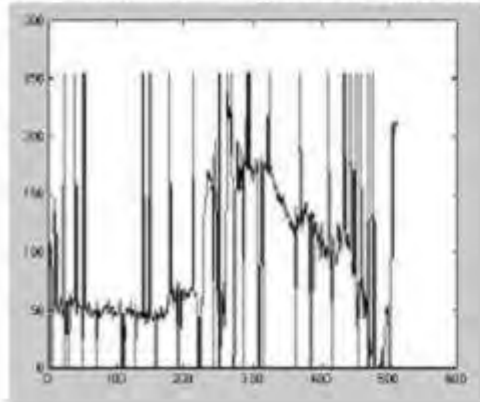
Source: K. Grauman

Медианный фильтр

Salt-and-pepper noise

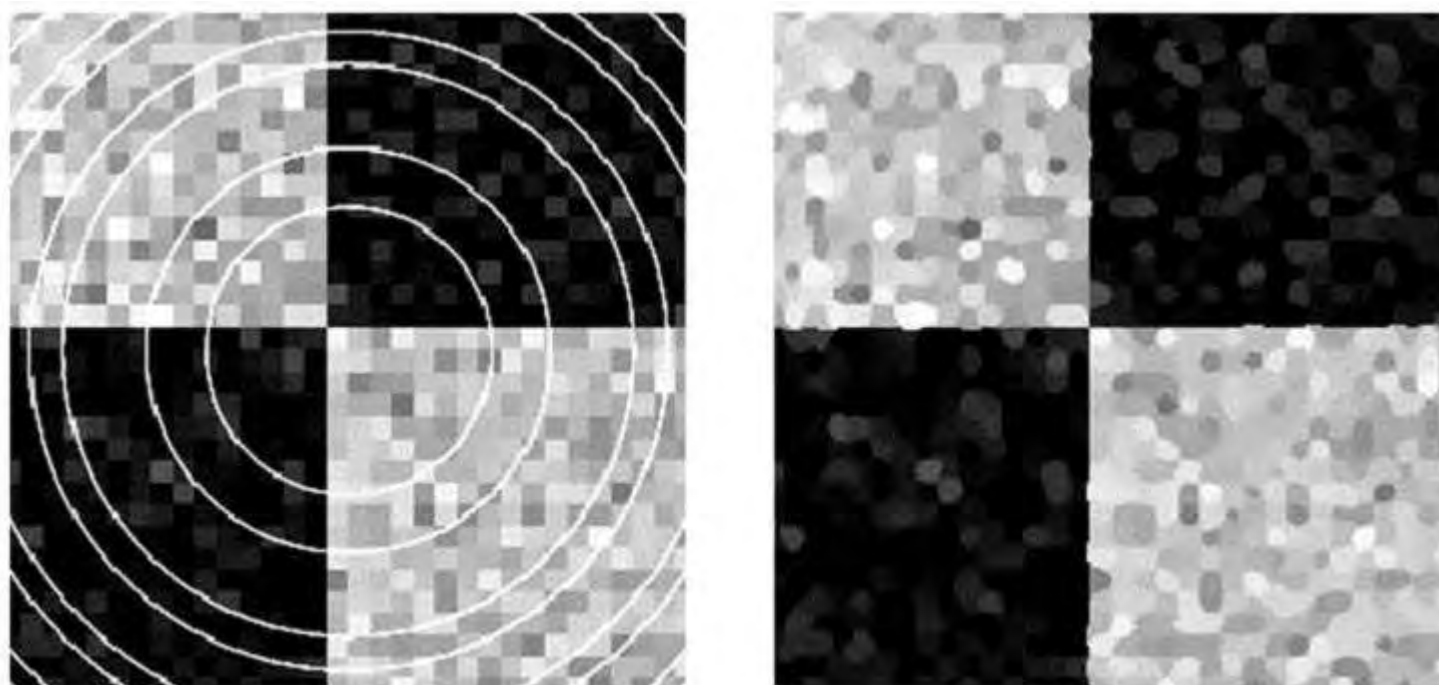


Median filtered



Source: M. Hebert

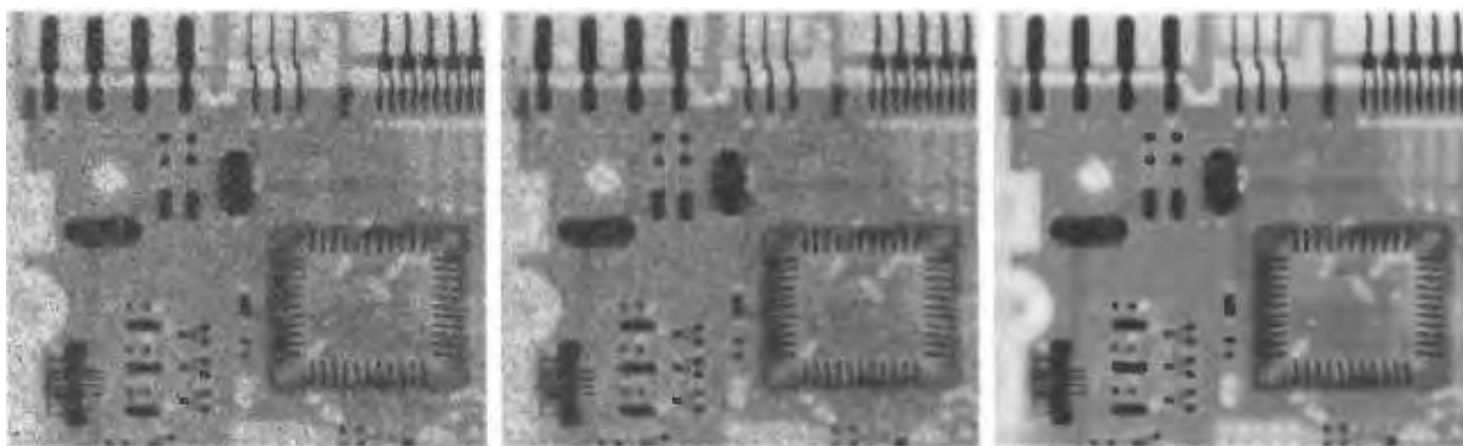
Медианный фильтр



Результат применения медианного фильтра с радиусом в 7 пикселей к изображению с шумом и артефактами в виде тонких светлых окружностей.

Рентгеновский снимок монтажной платы

Подавление шума фильтрами 3x3



усредняющим

медианным

Сравнение

3x3

5x5

7x7

Гауссов



Медианный



Source: S. Lazebnik

Медианный фильтр



original image



1px median filter



3px median filter



10px median filter

Быстрая реализация медианного фильтра

- Медианный фильтр считается дольше, чем операция свертки, поскольку требует частичной сортировки массива яркостей окрестных пикселей.
- Возможности ускорения:
 - Использовать алгоритмы быстрой сортировки
 - Конкретная реализация для каждого радиуса (3x3, 5x5)
 - Не использовать сортировку вообще – считать через гистограмму окрестности точки

Фильтры максимума и минимума

Фильтр «минимум» – также известный как фильтр эрозии, заменяет значение минимальным в окрестности.

Фильтр «максимум» – также известный как фильтр расширения, заменяет значение максимальным в окрестности.

Фильтр выборочной уборки шума

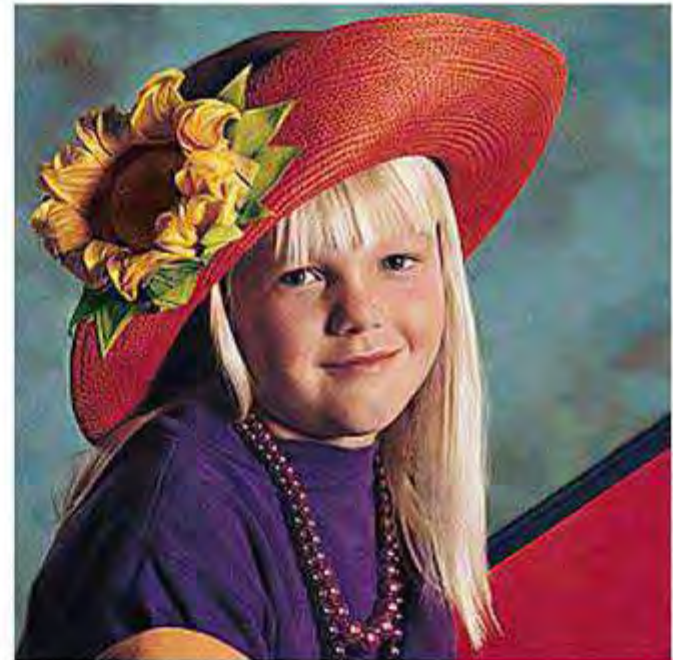
Фильтр выборочной уборки шума – проверяет, не является ли значение экстремумом в окрестности. Если да, то заменяет значение следующим самым большим (наименьшим) значением.

Фильтр «убрать шахматную доску»

Фильтр «убрать шахматную доску» – сглаживающий фильтр, специально предназначенный для удаления текстур, напоминающих шахматную доску из изображения при сохранении других особенностей. Это фильтр свёртки с ядром

$$w_{\text{dechecker}} = \begin{pmatrix} 0 & 1/144 & -1/72 & 1/144 & 0 \\ 1/144 & -1/18 & 1/9 & -1/18 & 1/144 \\ -1/72 & 1/9 & 7/9 & 1/9 & -1/72 \\ 1/144 & -1/18 & 1/9 & -1/18 & 1/144 \\ 0 & 1/144 & -1/72 & 1/144 & 0 \end{pmatrix}$$

«Продвинутые» фильтры



В чем отличие разных фильтров?

- Box filter (простое размытие) – помимо подавления шума портит резкие границы и размывает мелкие детали изображения
- Gaussian filter – меньше размывает мелкие детали, лучше убирает шум
- Median filter – резких границ не портит, убирает мелкие детали, изображение становится менее естественным
- «Продвинутые» фильтры – лучшее сохранение деталей, меньше размытие. Часто сложны в реализации и очень медленные.
- Что лучше? – зависит от конкретной задачи

Повышение резкости



Необходимо повысить
резкость (четкость)
изображения

Лирическое отступление

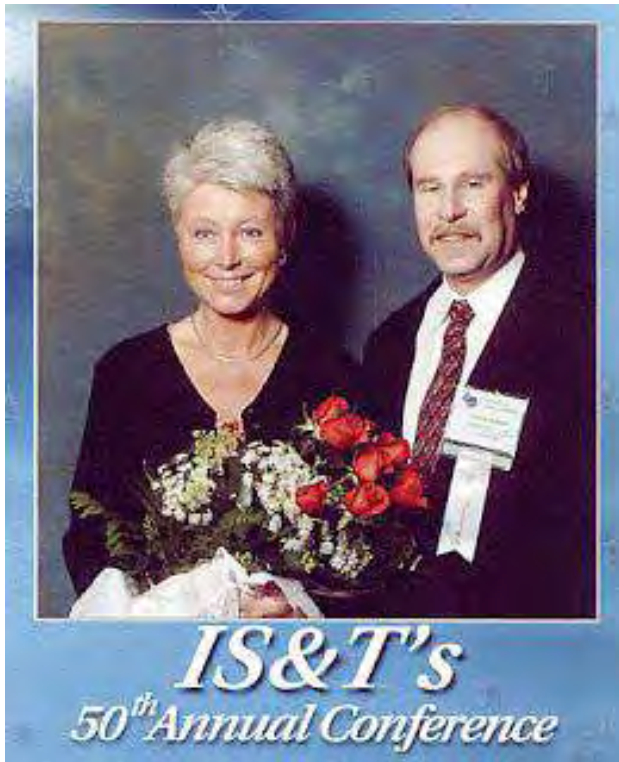


Lena (Lenna)

- Обрезанная (512x512) часть изображения с разворота Playboy, Nov 1972
- Самый популярный, но не первый случай использования Playboy в обработке изображений (первый в 1961)
- Пригласили на 50ую конференцию Society for Imaging Science and Technology (IS&T) in 1997

Lena Söderberg, born Sjööblom (born 31 March 1951)

is a Swedish, as **Lenna Sjööblom**



Повышение резкости

Что теряется при сглаживании?



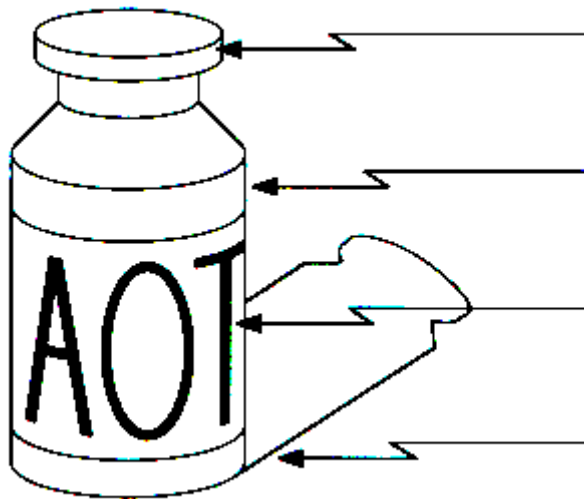
Добавим дополнительно высокие частоты:



Slide by S. Lazebnik

Откуда берутся границы

Существует множество причин формирования границ на изображении



Резкое изменение нормали поверхности

Резкое изменение глубины

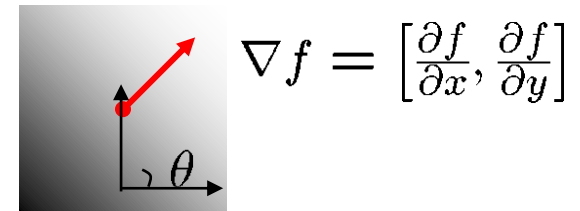
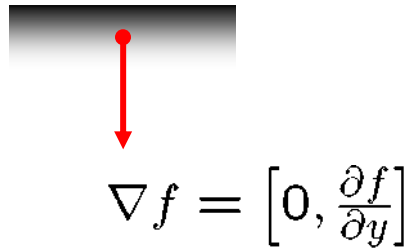
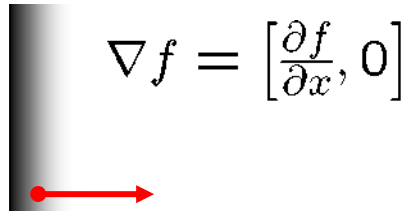
Резкое изменение цвета поверхности

Резкое изменение освещенности

Резкое изменение = «разрыв»

Градиент изображения

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



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:

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

Пример повышения резкости

Ядро
свертки

$$\frac{1}{10} \cdot \begin{vmatrix} -1 & -2 & -1 \\ -2 & 22 & -2 \\ -1 & -2 & -1 \end{vmatrix}$$



Пространственные фильтры повышения резкости

- достигается пространственным дифференцированием
- основаны на первой и второй производной

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

Упрощенная схема



Кросс-корреляция сигналов с маской первой производной $M=[-1, 0, 1]$

S_1	12	12	12	12	12	24	24	24	24	24
$S_1 * M$	0	0	0	0	12	12	0	0	0	0
S_2	24	24	24	24	24	12	12	12	12	12
$S_2 * M$	0	0	0	0	-12	-12	0	0	0	0
S_3	12	12	12	12	15	18	21	24	24	24
$S_3 * M$	0	0	0	3	6	6	6	3	0	0
S_4	12	12	12	12	24	12	12	12	12	12
$S_4 * M$	0	0	0	12	0	12	0	0	0	0

Кросс-корреляция сигналов с маской второй производной $M=[-1, 2, -1]$

S_1	12	12	12	12	12	24	24	24	24	24
$S_1 * M$	0	0	0	0	-12	12	0	0	0	0
S_2	24	24	24	24	24	12	12	12	12	12
$S_2 * M$	0	0	0	0	12	-12	0	0	0	0
S_3	12	12	12	12	15	18	21	24	24	24
$S_3 * M$	0	0	0	-3	0	0	0	3	0	0
S_4	12	12	12	12	24	12	12	12	12	12
$S_4 * M$	0	0	0	-12	24	-12	0	0	0	0

Первая и вторая производные

- **Первая** производная обычно формирует более **толстые** контуры.
- **Вторая** производная формирует **большой** по величине отклик на **мелкие детали** (на отдельных точках и на тонких линиях).
- Отклик на ступенчатое изменение яркости у первой производной, как правило, выше, чем у второй.
- Для участков постоянного изменения яркости вторая производная формирует двойной отклик.
- При одинаковых амплитудах изменения сигнала вторая производная формирует более сильный отклик для тонкой линии, чем для ступеньки, а для отдельной точки - более сильный, чем для линии.

Применение производных

- Первая производная – в задачах выделения контуров
- Вторая производная – в задачах улучшения изображений
(Эффект двойных контуров – помеха в задаче обнаружения контуров)
- Объединение методов

Лапласиан

Лапласиан – простейший изотропный оператор

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Изотропные фильтры – отклик не зависит от направления неоднородностей

Изотропные фильтры – инвариантность к повороту

Дискретный вид лапласиана

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y)$$

Лапласианы

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

a)

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Инвариантны относительно поворота на углы, кратные 90° и 45°

Повышение
резкости
изображения
Северного
полюса Луны
лапласианом



Восстановление фона с сохранением резкости

- Сложить исходное изображение и лапласиан

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y), & \text{если } w(0,0) < 0 \\ f(x, y) + \nabla^2 f(x, y), & \text{если } w(0,0) \geq 0 \end{cases}$$

$w(0,0)$ – центральное значение маски лапласиана

Упрощения – Составная маска лапласиан

$$g(x, y) =$$

$$f(x, y) - [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] + 4f(x, y) =$$
$$5f(x, y) - [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)]$$

0	-1	0
-1	5	-1
0	-1	0

-1	-1	-1
-1	9	-1
-1	-1	-1

Вольфрамовая нить накала

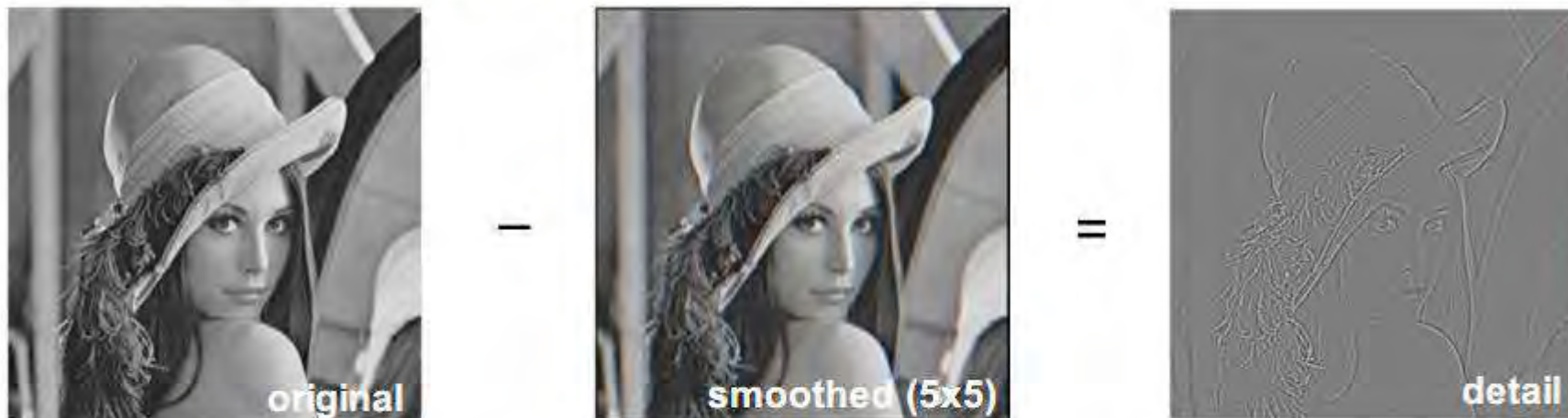
0	-1	0
-1	5	-1
0	-1	0

-1	-1	-1
-1	9	-1
-1	-1	-1



Результаты диагональной маски более резкие

Что теряется при сглаживании?



Добавим дополнительно высокие частоты:



Source: S. Lazebnik

Нерезкое маскирование

Для повышения резкости

Применяется в полиграфии

$$f_{mask}(x, y) = f(x, y) - \bar{f}(x, y)$$

Создание нерезкого изображения

Вычитание из изображения его расфокусированной копии
(результат - маска)

Прибавление маски к исходному изображению

Фильтрация с подъемом высоких частот

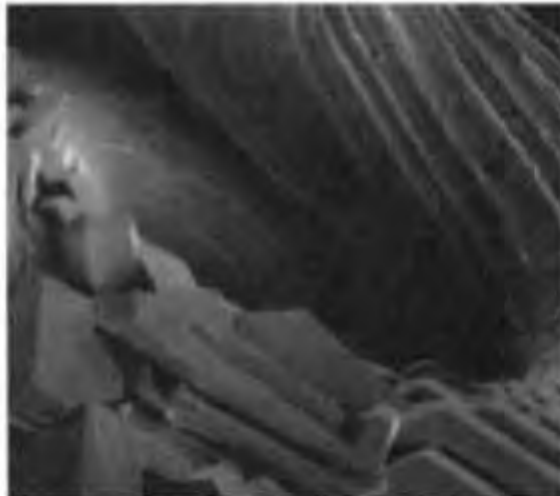
Некоторое обобщение нерезкого маскирования

$$f_{mask}(x, y) = f(x, y) - \bar{f}(x, y)$$

$$f_{hb}(x, y) = f(x, y) + Af_{mask}(x, y)$$

Типичное приложение фильтрации с подъемом высоких частот

Исходное изображение темнее чем требуется



а) исходное

б) маска

в) $A=1$

г) $A=1,7$



Фильтрация с подъемом высоких частот



a) $A=1$



б) $A=2$



в) $A=2.8$

Фильтры с подъемом высоких частот на базе лапласиана

0	-1	0
-1	$A+4$	-1
0	-1	0

-1	-1	-1
-1	$A+8$	-1
-1	-1	-1

Увеличение среднего уровня яркости

$$f_{\text{hb}}(x, y) = \begin{cases} Af(x, y) - \nabla^2 f(x, y), & \text{если } w(0, 0) < 0 \\ Af(x, y) + \nabla^2 f(x, y), & \text{если } w(0, 0) \geq 0 \end{cases}$$

Улучшение изображений с использованием первых производных: градиент

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

$$|\nabla f| = [G_x^2 + G_y^2]^{1/2} = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

$$|\nabla f| \approx |G_x| + |G_y|$$

Перекрестный градиентный оператор Робертса

$$G_x = (z_9 - z_5) \quad \text{и} \quad G_y = (z_8 - z_6)$$

$$\nabla f = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2 \right]^{1/2}$$

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

-1	0
0	1

0	-1
1	0

1965г.

Оператор Собела

$$\nabla f \approx \left| (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \right| + \left| (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \right|.$$

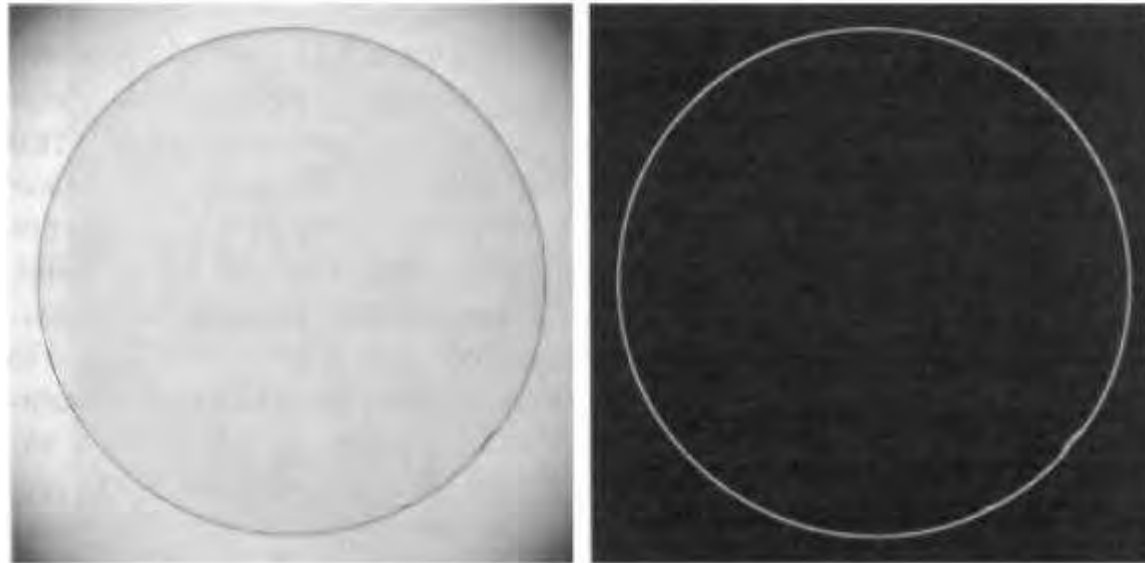
z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Пример. Применение градиента для улучшения контуров – оператор Собела

Для дальнейшего автоматического контроля



Определение оператора Превитт (Джуди Превитт)

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Примеры



Робертса



Превитт



Собея

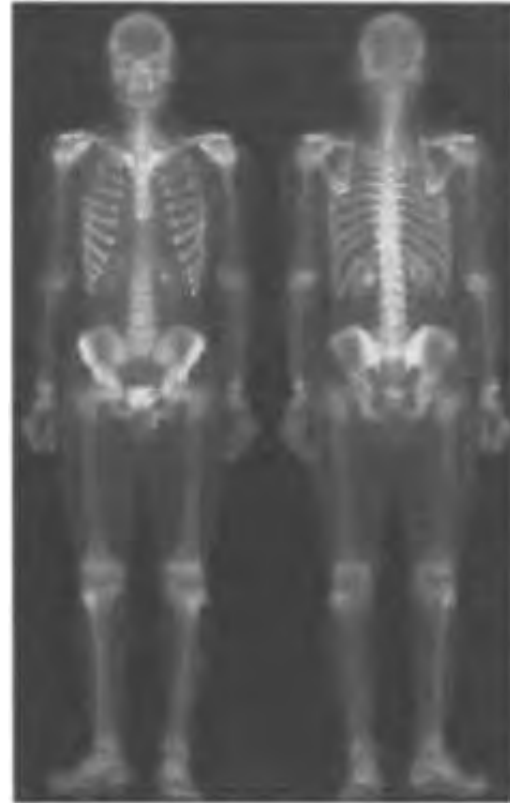
Свойства масок для вычисления производных

- Для получения высокого отклика в областях сигнала с высоким контрастом элементы масок производных имеют **разные знаки**
- **Сумма** элементов масок производных **равна 0**, поэтому в областях сигнала с постоянными значениями маски дают нулевой результат
- Точкам **высокого контраста** соответствуют **большие абсолютные значения** отклика масок **первых производных**
- Среди значений отклика масок **второй производной** точкам **высокого контраста** соответствуют точки **перехода через 0**

Свойства сглаживающих масок

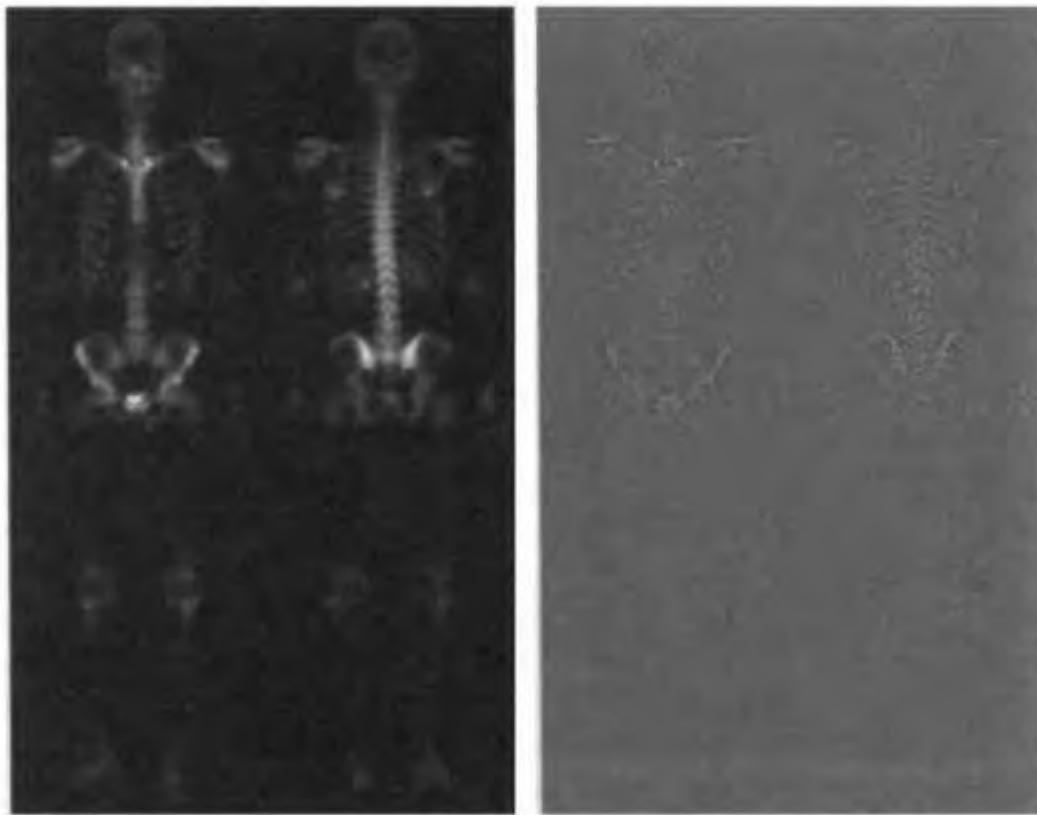
- Элементы сглаживающих масок имеют **положительные значения** и их **сумма равна 1**
- Отклик масок на **областях постоянных значений совпадает с входными значениями**
- **Степень сглаживания** и снижения шума **пропорциональна размеру** маски
- Ступенчатые края размываются **пропорционально** размеру маски

Было – будет



Снимок с помощью гамма-лучей

Комбинирование методов пространственного улучшения



1

2

Применение Лапласиана для выявления мелких деталей

Комбинирование методов пространственного улучшения

Повышение
резкости
сложением
двух
предыдущих
изображений



3

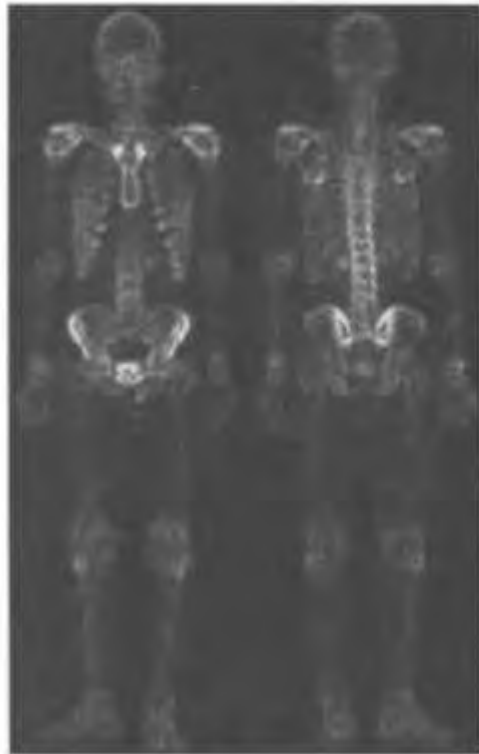


4

Применение
оператора
Собела – для
улучшения
выступающих
краев

Комбинирование методов пространственного улучшения

Сглаживание
градиента
усредняющим
фильтром 5x5



5

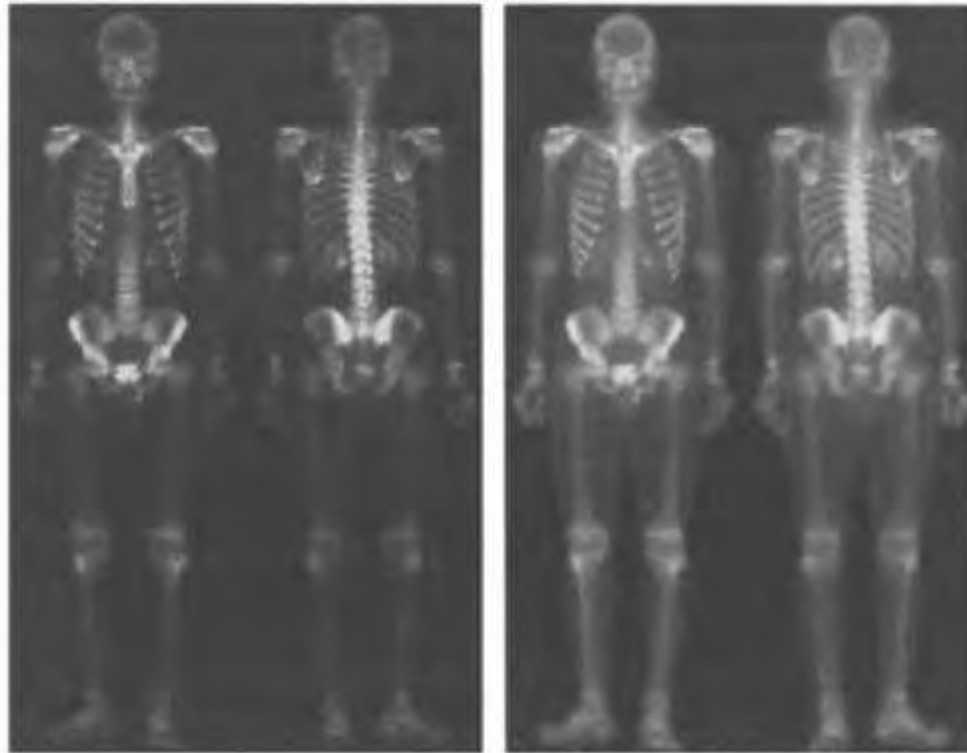


6

Перемножение
изображений 3 и
5 (с повышенной
резкостью и
сглаженного)

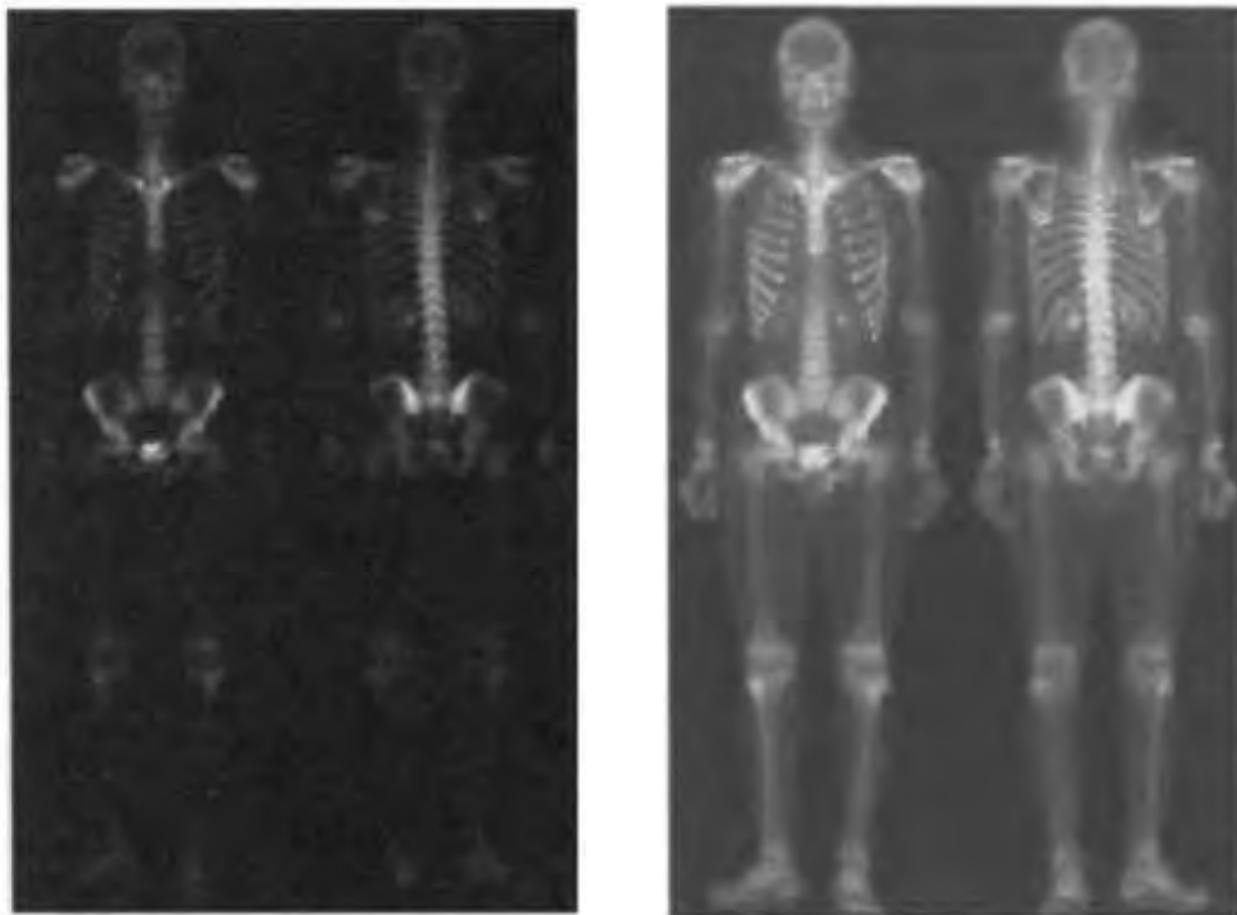
Комбинирование методов пространственного улучшения

Повышение
резкости
сложением
изображений
1 и 6



Применение
градиентной
 γ -коррекции
($\gamma=0,5$ $c=1$)

Было – стало



Предварительная обработка изображений

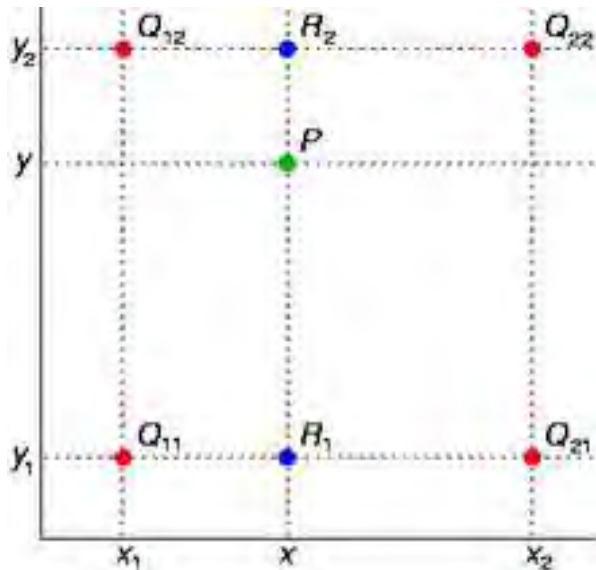
Примеры задач

- увеличение фрагментов
- геометрические преобразования
- повышение контрастного разрешения
- выделение наиболее информативных фрагментов
- улучшение качества изображений

Увеличение цифровых изображений

- Интерполяция по ближайшему соседу
- Метод дублирования пикселей – частный случай
 - Эффект пикселизации
- Билинейная интерполяция

Билинейная интерполяция



$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$

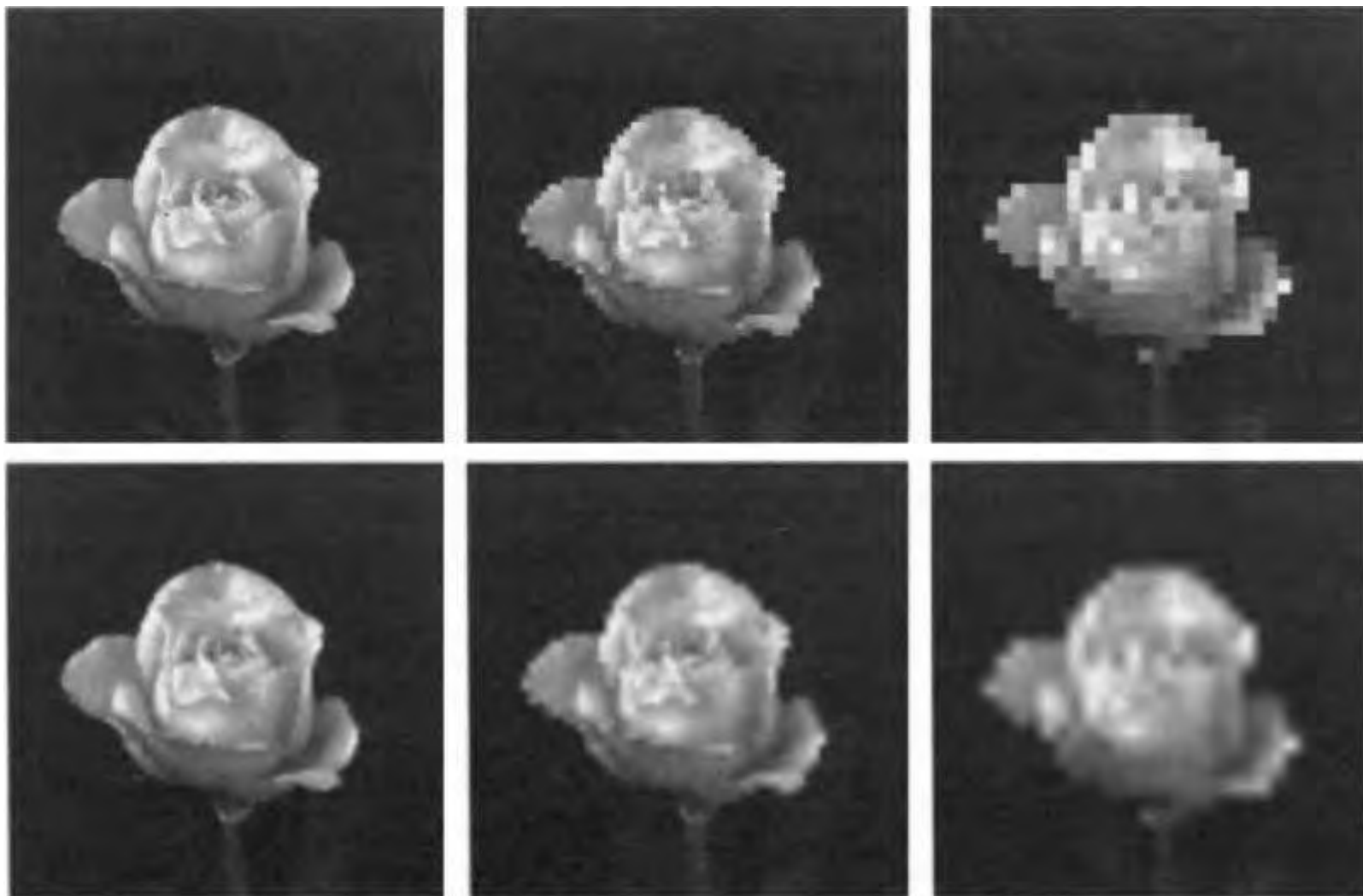
$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2).$$

Четыре красные точки представляют собой известные значения функции. Значение в зеленой точке должно быть интерполировано.

Пример увеличения



Результаты: по ближайшему соседу и билинейная



Недостаток метода билинейной интерполяции

Исходное изображение $W \times H$ пикселей

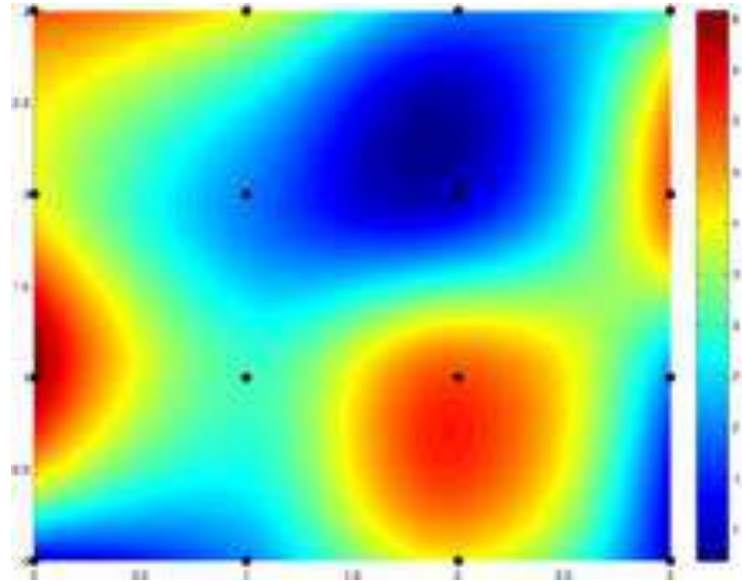
Увеличение в N раз

Результирующее изображение не $NW \times NH$ пикселей, а $(N(W-1)+1) \times (N(H-1)+1)$ пикселей.

Интерполяция с большим числом соседей

- 3-D графика
- Обработка медицинских изображений

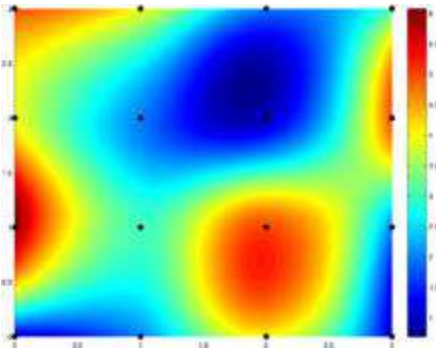
Бикубическая интерполяция



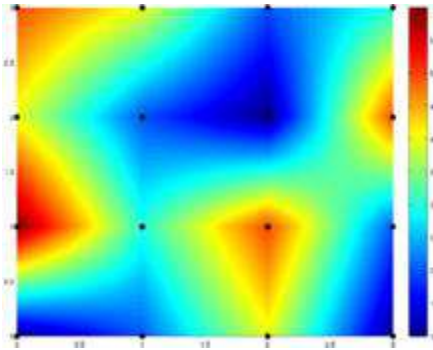
Сетка $[0,3] \times [0,3]$

16 соседей

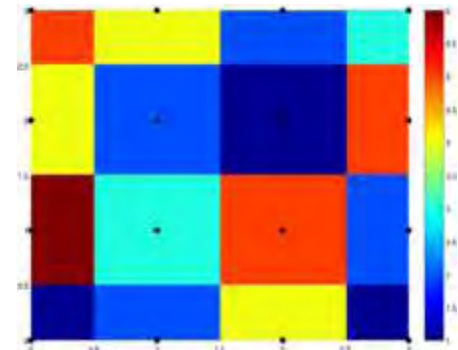
Сравнение



Бикубическая



Билинейная



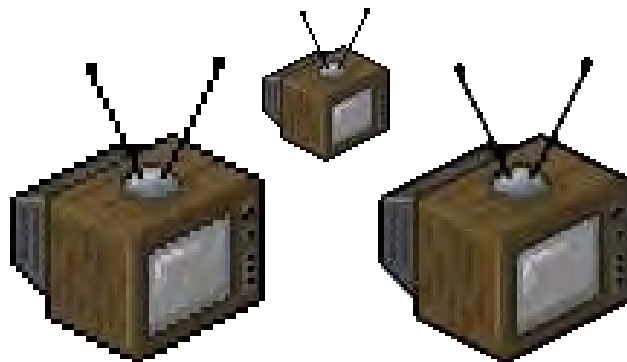
Ближайшего
соседа

Алгоритмы масштабирования

Алгоритм	Изображение
(Исходные изображения)	
Super-xBR 4x	
SuperEagle	
SuperSal	
Sal 2x	
Scale 2x	

Изображение, увеличенное при помощи

метода ближайшего соседа (слева) и 2xSaI (справа)



EPX/Scale2x/AdvMAME2x

- EPX («Eric's Pixel eXpansion», пиксельное увеличение Эрика) — алгоритм, разработанный Эриком Джонстоном из [LucasArts](#) приблизительно в 1992 году

такие как AdvMAME2x и Scale2x, разработанные приблизительно в 2001 году, имеют более эффективную, но функционально идентичную, реализацию:

```
  A      --\ 1 2
C R B   --/ 3 4
  D
1=R; 2=R; 3=R; 4=R;
Если C==A и C!=D и A!=B => 1=A
Если A==B и A!=C и B!=D => 2=B
Если B==D и B!=A и D!=C => 4=D
Если D==C и D!=B и C!=A => 3=C
```

Семейство $hq3x$



Изображение, увеличенное в 3
раза методом ближайшего
соседа

Изображение, полученное при
использовании алгоритма $hq3x$.

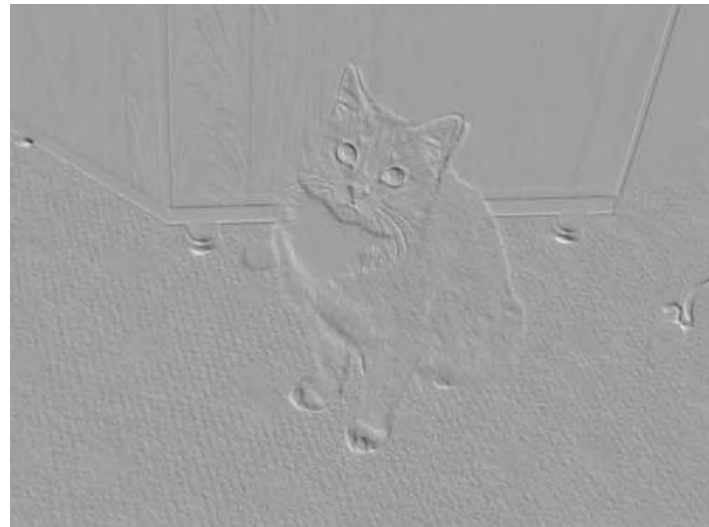
Спецэффекты

- Рассмотрим
 - Тиснение
 - Негатив
 - «Своящиеся» края
 - Геометрические эффекты
 - Перенос/поворот
 - Искажение
 - «Эффект стекла»

Тиснение

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Фильтр + сдвиг яркости, нормировка...

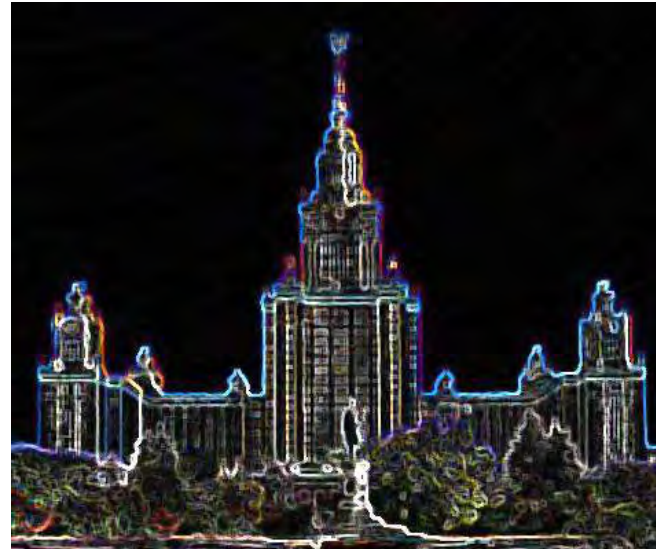


Цифровой негатив



$$R' = 255 - R; \quad G' = 255 - G; \quad B' = 255 - B;$$

Светящиеся края



Медианный фильтр + выделение краев + фильтр «максимума»

«Волны»



Волны 1:

$$x(k; l) = k + 20\sin(2\pi l / 128); y(k; l) = l;$$

Волны 2:

$$x(k; l) = k + 20\sin(2\pi k / 30); y(k; l) = l;$$

«Эффект стекла»



$$x(k; l) = k + (\text{rand}(1, 1) - 0.5) * 10;$$
$$y(k; l) = l + (\text{rand}(1, 1) - 0.5) * 10;$$

Современные методы

- Сейчас методы фильтрации изображений продолжают активно развиваться
- Можно выделить подходы:
 - Вейвлет и т.д. представления
 - Разреженные представления
 - Марковские случайные поля

Резюме лекции

- Линейная фильтрация (свёртка) изображения позволяет решать целый ряд задач – шумоподавление, повышение резкости, оценка градиента
- Выделение краёв изображения в простом случае можно достичь поиском локальных максимумов градиента яркости
-
- Всё это открытые задачи и сейчас активно продолжают исследоваться!

Частотные методы улучшения изображений

Лекция 6

Частотная обработка изображений

- Базируется на Фурье-анализе
- Частично подходы частотной обработки аналогичны подходам пространственной обработки
- Частично подходы частотной обработки дополняют подходы пространственной обработки

Главное научное достижение Фурье

Изложено в мемуарах в 1807

Полностью опубликовано в 1822 в книге «Аналитическая теория тепла»

В 1878 переведена на английский Фрименом

Метод Фурье состоял в представлении функций в виде тригонометрических рядов Фурье



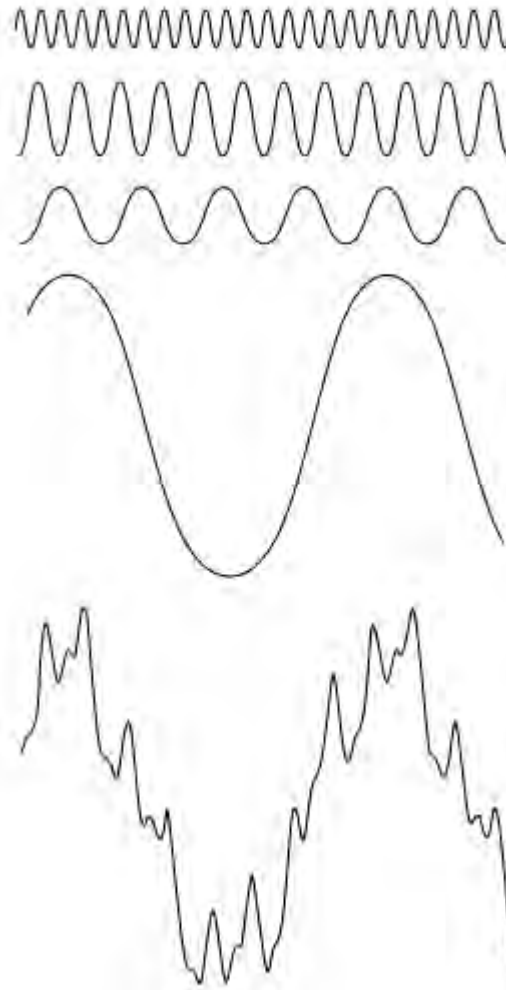
Жан Батист Жозеф Фурье
(1768 — 1830)

Результат Фурье

Любая периодическая функция может быть представлена в виде суммы синусов и/или косинусов различных частот, умноженных на некоторые коэффициенты.

Сумма – ряд Фурье

Нижняя функция – сумма четырех верхних



Результат Фурье

Если функция **не** является **периодической**, но **площадь под графиком её модуля конечна**, она может быть выражена в виде интеграла от синусов и/или косинусов, умноженных на некоторую весовую функцию.

Преобразование Фурье

Характерная особенность

Функция, заданная как рядом, так и преобразованием Фурье, может быть **полностью восстановлена** при помощи процедуры обращения.

Идея частотных преобразований

Область определения функции



Область Фурье



Область определения функции

Где применяется

- В начале 60-х Революция в области обработки сигналов
 - ЭВМ
 - БПФ
- Обработка сигналов
- Обработка звука
- Форматы звуковых файлов (например, MP3 и JPEG)
- Обработка изображений
- Медицинская диагностика
- Шифрование
- Средства электронной связи
- ...

Применение к обработке изображений

Изображения – функции конечной протяженности



Преобразование Фурье

Одномерный случай

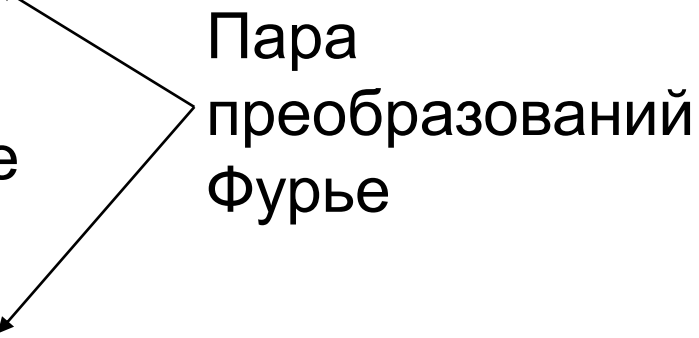
- Прямое Фурье-преобразование (Фурье-образ)

$$F(u) = \int_{-\infty}^{+\infty} f(x) e^{-i2\pi ux} dx$$

- Обратное Фурье-преобразование

$$f(x) = \int_{-\infty}^{+\infty} F(u) e^{i2\pi ux} du$$

Пара преобразований Фурье



Фурье-пара

$F(u)$ $f(x)$



Двумерный случай

- Прямое Фурье-преобразование (Фурье-образ)

$$F(u, v) = \int_{-\infty-\infty}^{+\infty+\infty} \int f(x, y) e^{-i2\pi(ux+vy)} dx dy$$

- Обратное Фурье-преобразование

$$f(x, y) = \int_{-\infty-\infty}^{+\infty+\infty} \int F(u, v) e^{i2\pi(ux+vy)} du dv$$

Дискретная форма

Прямое дискретное преобразование Фурье

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-i2\pi ux/M}, \quad u = 0, 1, 2, \dots, M-1$$

Обратное дискретное преобразование Фурье

$$f(x) = \sum_{u=0}^{M-1} F(u) e^{i2\pi ux/M}, \quad x = 0, 1, 2, \dots, M-1$$

Частотная и временная область

$$e^{i\theta} = \cos \theta + i \sin \theta$$

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) [\cos 2\pi ux / M - i \sin 2\pi ux / M], \quad u = 0, 1, 2, \dots, M - 1$$

Область значений переменной u – **частотная область**

Каждый из M элементов $F(u)$ – **частотная компонента** преобразования

Область значений переменной x – **временная область**

Каждый из M элементов $f(x)$ – **временная компонента**

Фурье-спектр, энергетический спектр и фаза

Представление в полярных координатах $F(u) = |F(u)|e^{-i\varphi(u)}$

Модуль или спектр Фурье-преобразования

$$|F(u)| = [R^2(u) + I^2(u)]^{1/2}$$

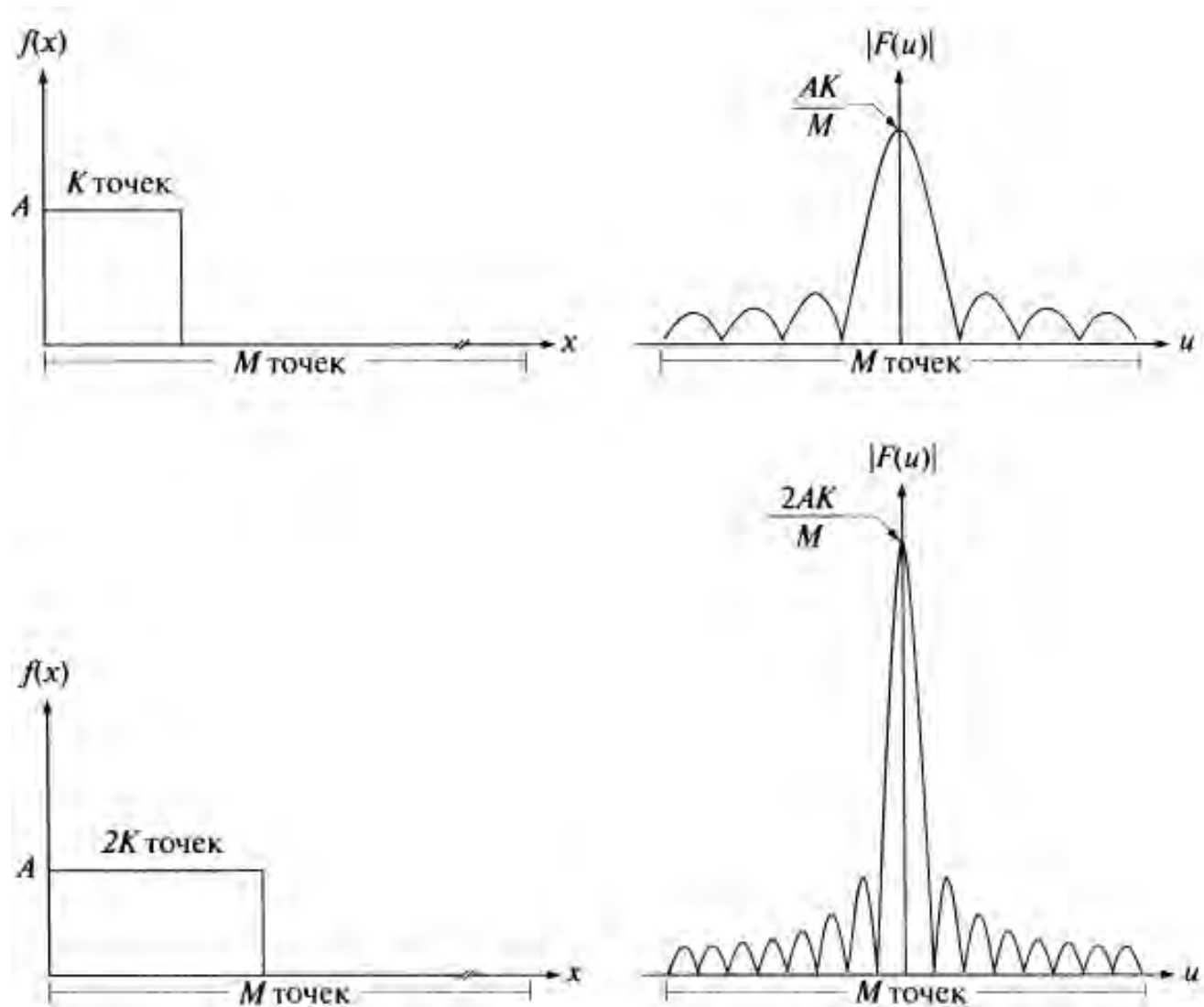
Энергетический спектр или спектральная плотность

$$P(u) = |F(u)|^2 = R^2(u) + I^2(u)$$

Фаза или фазовый спектр Фурье-преобразования

$$\varphi(u) = \operatorname{arctg} \left[\frac{I(u)}{R(u)} \right]$$

Пример. Фурье-спектры



Взаимосвязь между шагом дискретизации и частотными интервалами

$$\Delta u = \frac{1}{M\Delta x} = \frac{1}{X}$$

частотное
разрешение

$$M\Delta u = \frac{1}{\Delta x}$$

диапазон
частот

$$\Delta x = \frac{1}{M\Delta u} = \frac{1}{U}$$

Двумерное ДПФ и его обращение

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi(ux/M + vy/N)}$$

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{i2\pi(ux/M + vy/N)}$$

- u, v – переменные преобразования или частотные переменные
- x, y – переменные изображения или пространственные переменные

Фурье-спектр, энергетический спектр и фаза

$$|F(u, v)| = [R^2(u, v) + I^2(u, v)]^{1/2}$$

$$P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v)$$

$$\varphi(u, v) = \operatorname{arctg} \left[\frac{I(u, v)}{R(u, v)} \right]$$

Взаимосвязь между шагом дискретизации и частотными интервалами

$$\Delta u = \frac{1}{M\Delta x} \quad \Delta \nu = \frac{1}{N\Delta y}$$

частотное разрешение

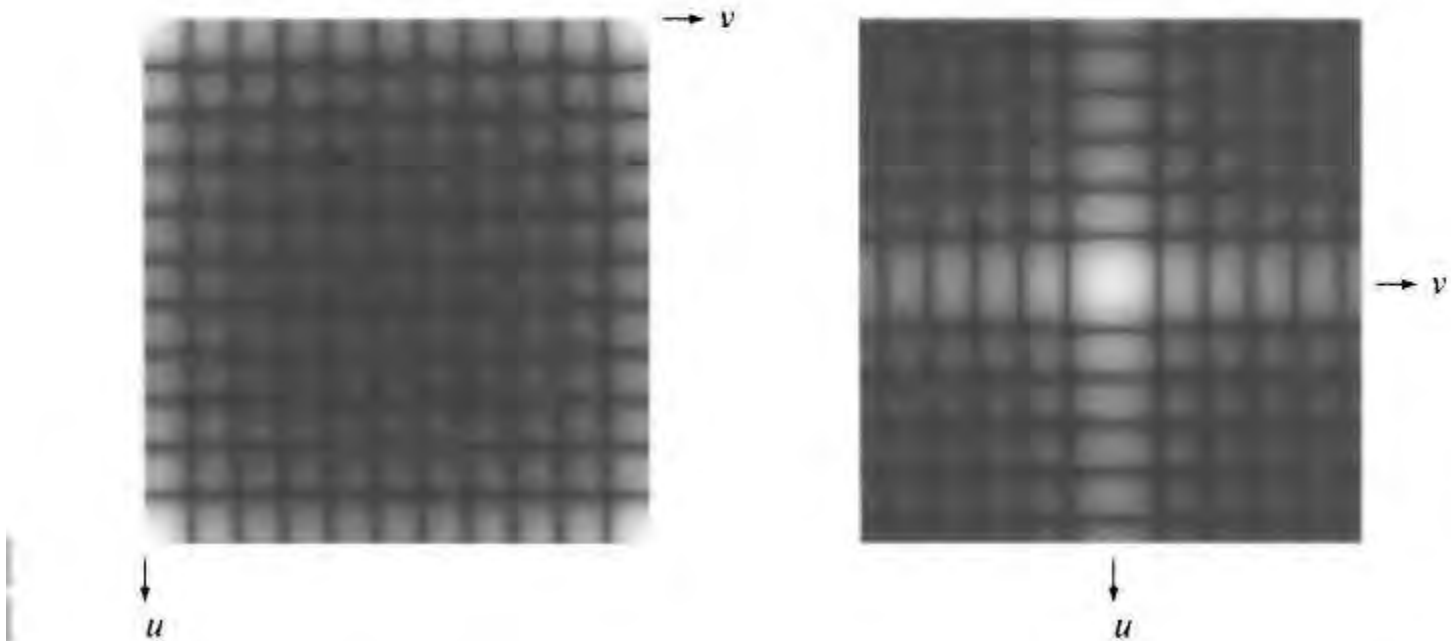
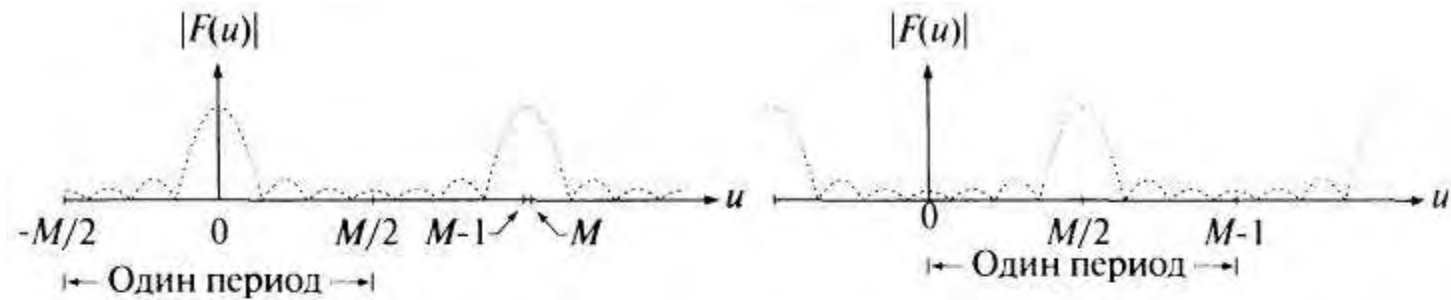
Фурье-образ для действительных функций будет симметрично сопряжённым

$$F(u, v) = F^*(-u, -v)$$

$$|F(u, v)| = |F(-u, -v)|$$

Периодичность

$$\mathfrak{F}\left[f(x, y)(-1)^{x+y}\right] = F(u - M/2, v - N/2)$$



Постоянная составляющая спектра

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi(ux/M + vy/N)}$$

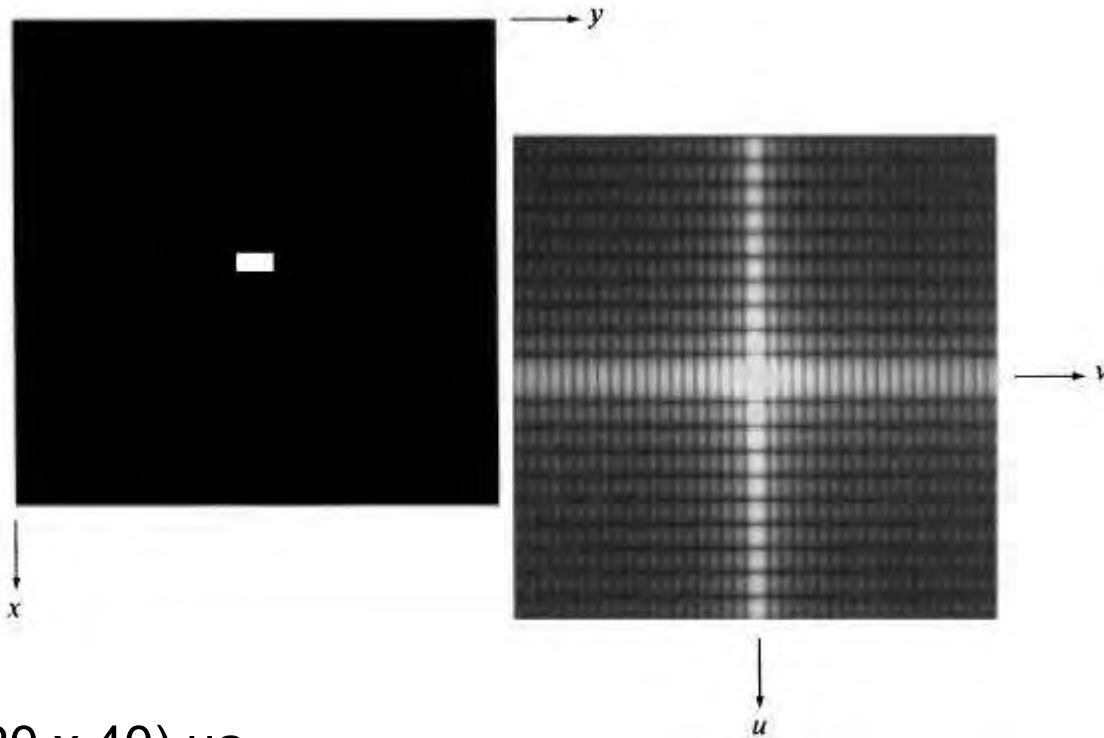
$$F(0, 0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

Среднее значение яркости на изображении

Основные свойства частотной области

- Постоянная частотная составляющая ($u=v=0$) совпадает со средней яркостью изображения
- Низкие частоты (вблизи начала координат) соответствуют медленно меняющимся компонентам изображения
- Высокие – быстро меняющимся

Пример. Центрированный спектр простой двумерной функции



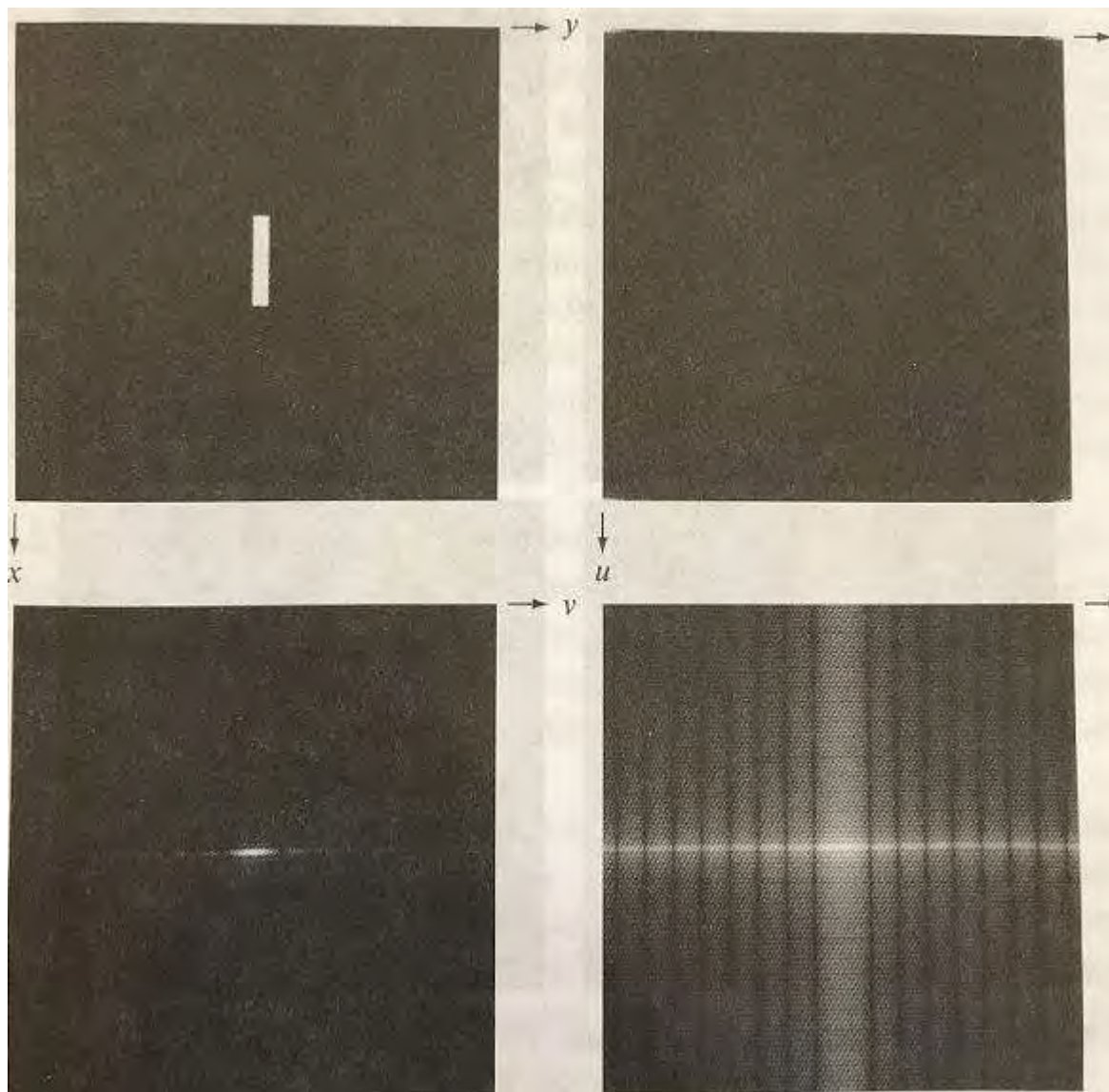
Белый (20 x 40) на
черном (512 x 512)

$$* (-1)^{x+y}$$

Центрированный спектр (после
логарифмического
преобразования яркости)

Изображение → Спектр

Изображение



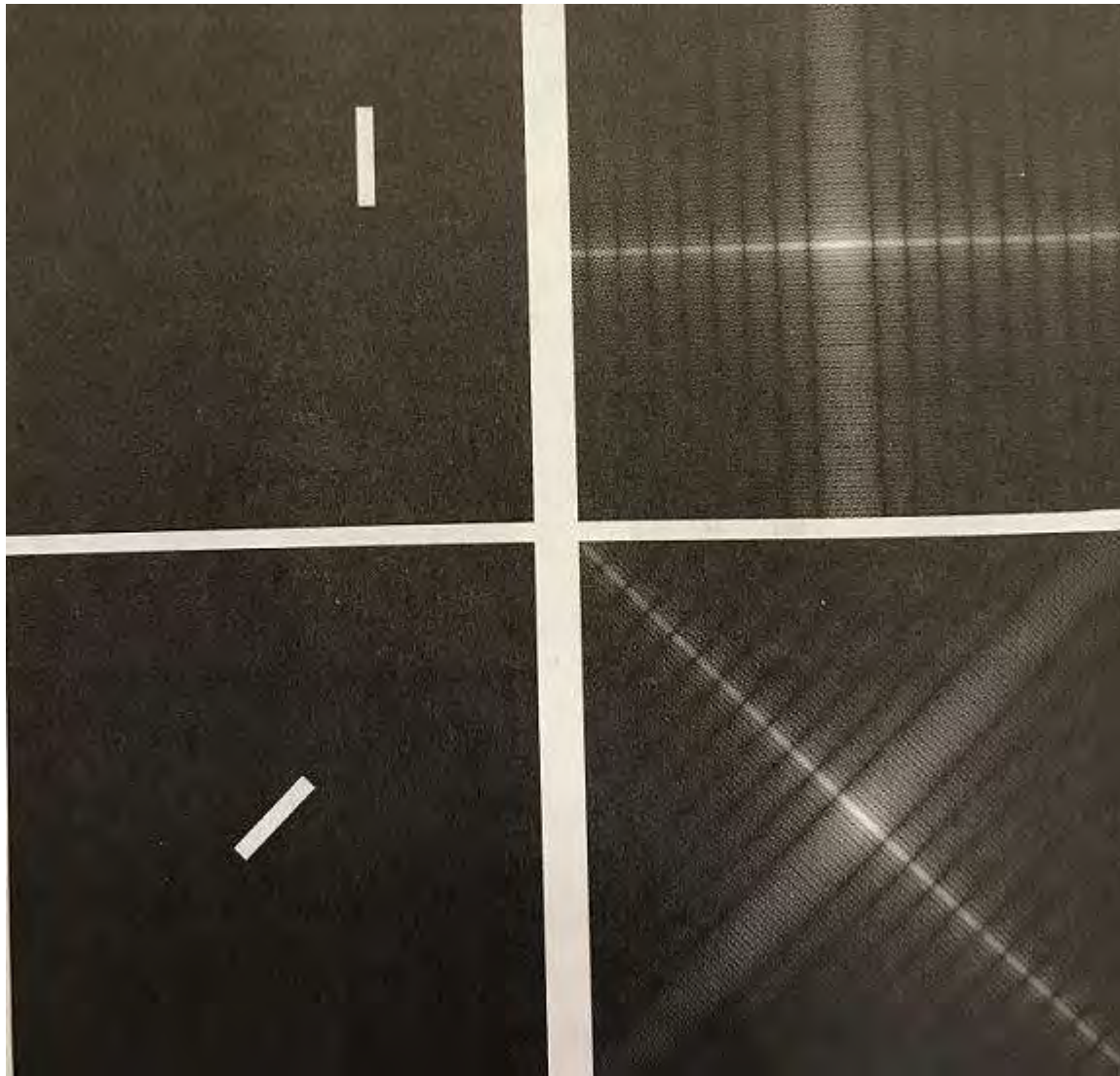
Спектр

Яркие
пятна в
углах

Центрирован
ный спектр

Результат логарифмических
преобразований

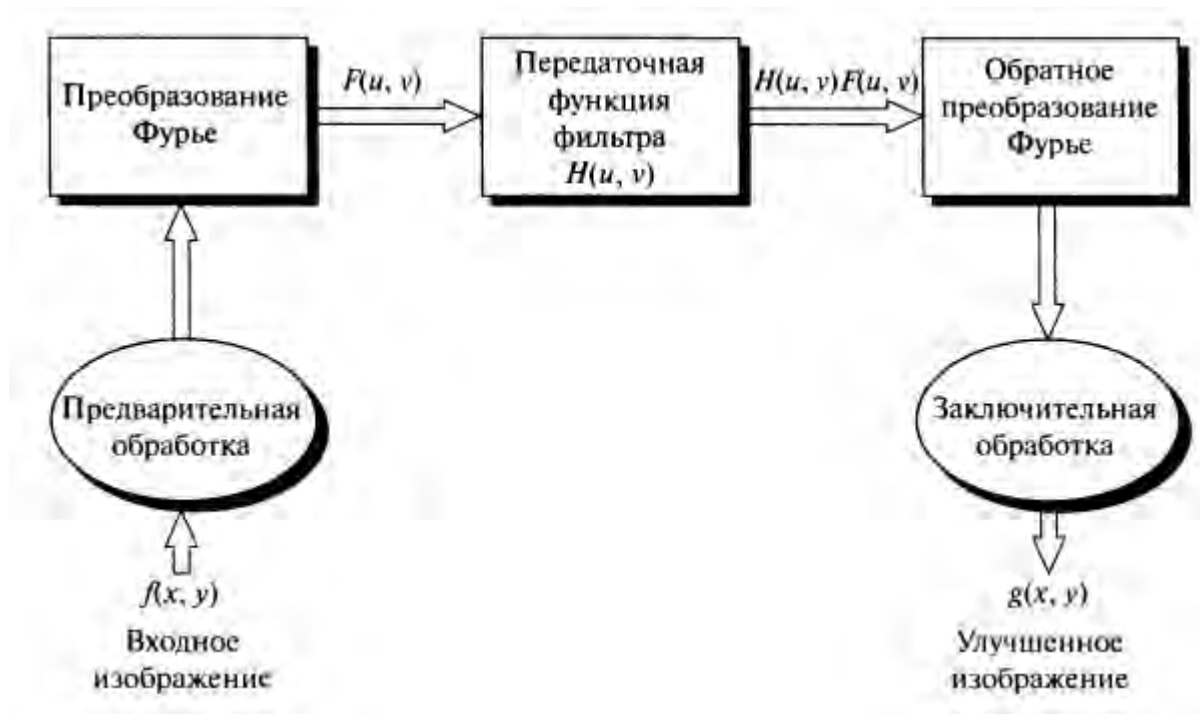
Сдвинутый и повернутый прямоугольник



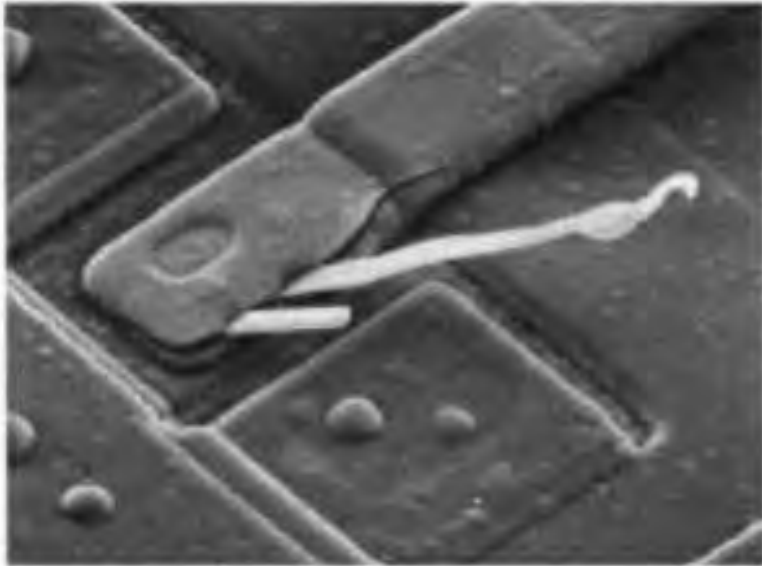
Процедура фильтрации в частотной области

1. Изображение * $(-1)^{x+y}$
2. Вычисляется прямое ДПФ $F(u,v)$
3. Функция $F(u,v)$ * функцию фильтра $H(u,v)$
4. Вычисляется обратное ДПФ
5. Выделяется вещественная часть
6. * $(-1)^{x+y}$

Основные этапы фильтрации в частотной области



Пример. Изображение и его Фурье-спектр



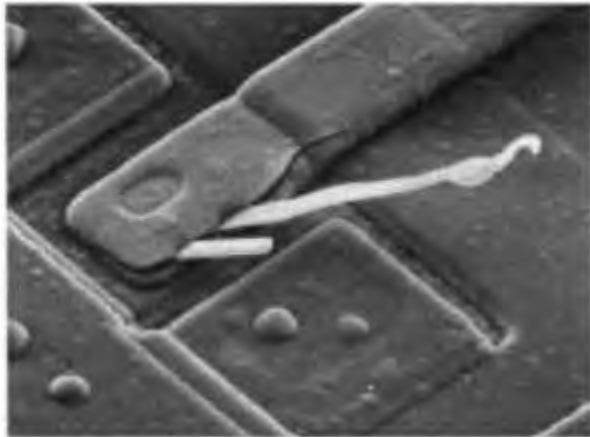
Увеличенное в 2500 раз изображение интегральной схемы, полученное сканирующим электронным микроскопом

Brockhouse Institute for Material Research

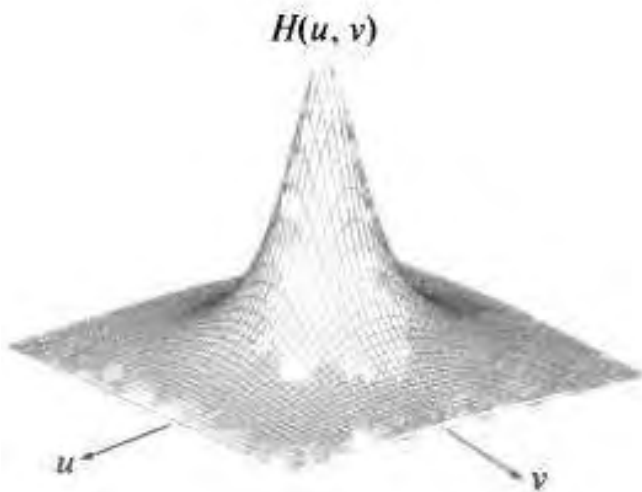
Основные фильтры

Фильтр пробка – узкополосный режекторный фильтр –
обнуление среднего

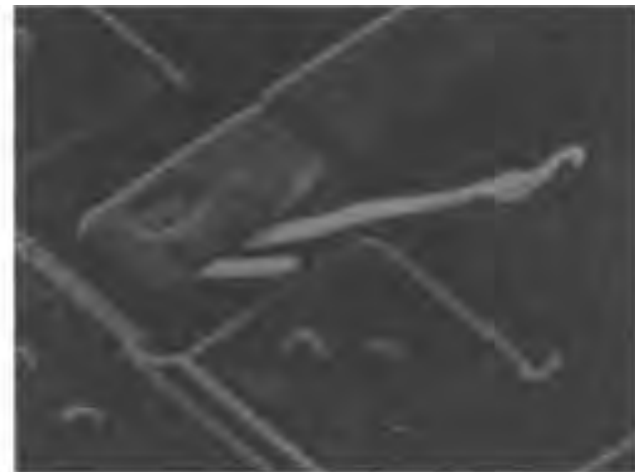
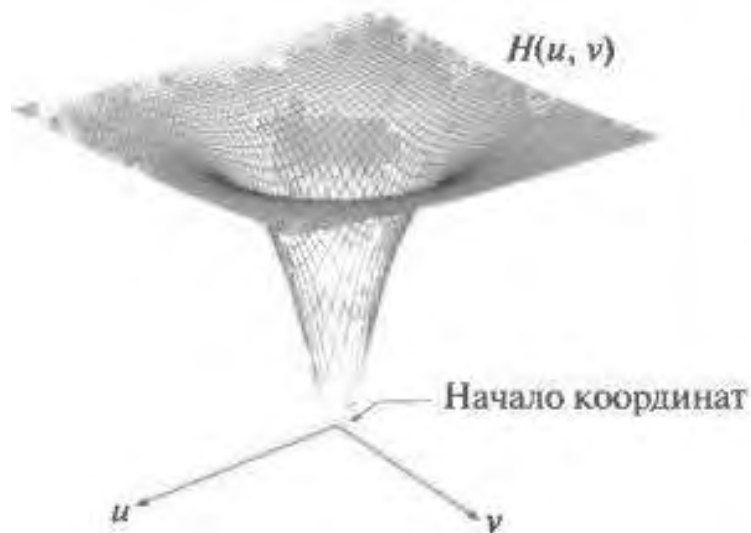
$$H(u, v) = \begin{cases} 0, & \text{при } (u, v) = (M/2, N/2) \\ 1, & \text{в другом случае} \end{cases}$$



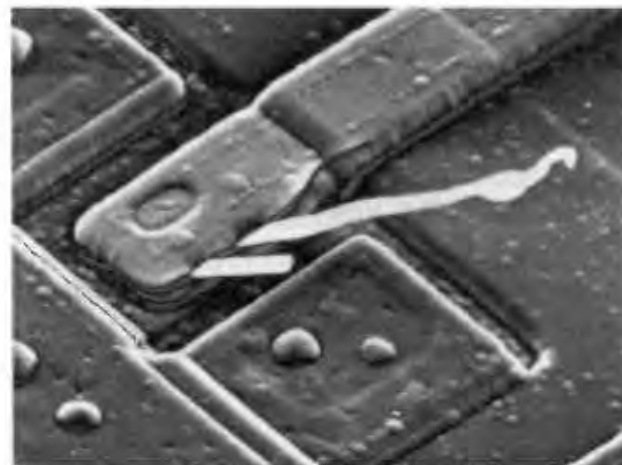
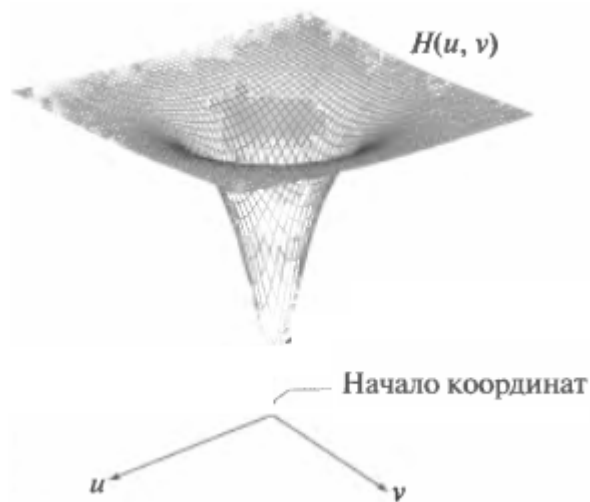
Низкочастотный фильтр



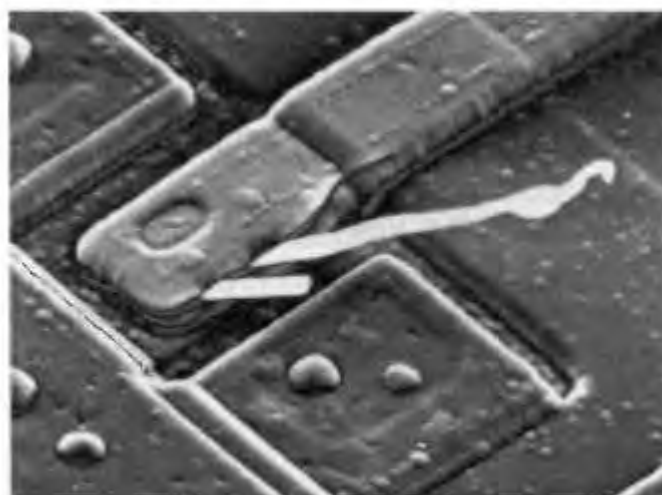
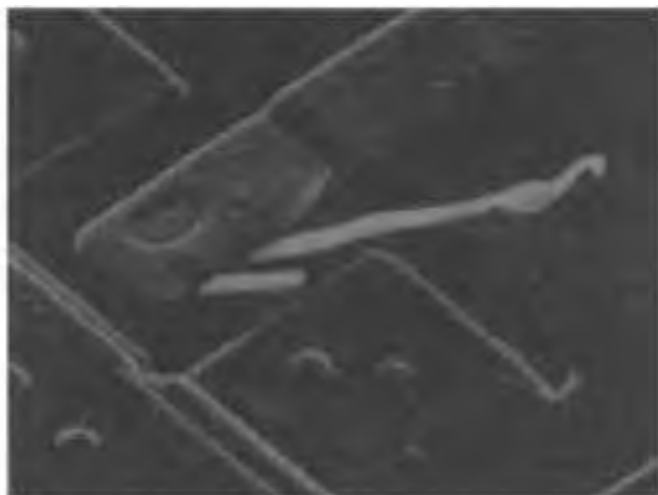
Высокочастотный фильтр



Высокочастотный фильтр с добавлением к передаточной функции константы, равной половине высоты фильтра



Результаты высокочастотной фильтрации изображения



Модификация фильтра добавлением к передаточной функции константы, равной половине высоты фильтра

Теорема о свёртке

Соответствие между фильтрацией в частотной и пространственной областях

$$f(x, y) * h(x, y) \Leftrightarrow F(u, v)H(u, v)$$

$$f(x, y)h(x, y) \Leftrightarrow F(u, v) * H(u, v)$$

Дискретная свёртка

$$f(x, y) * h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(x - m, y - n)$$

Сравнение фильтров

- Частотные почти всегда нагляднее
- Пространственные много меньше размера

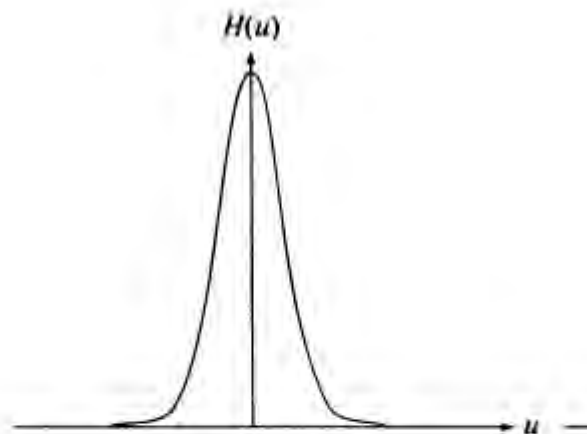
Частотный → Прообраз (пространственный) → Маска меньшего размера

Получение пространственного фильтра из частотного

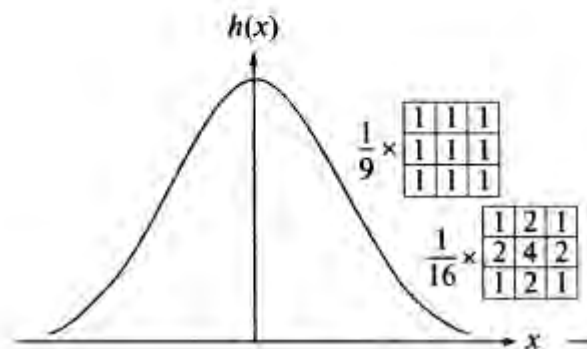
1. Функция фильтра $H(u, v) * (-1)^{u+v}$
2. Вычисляется обратное ДПФ
3. Вещественная часть $* (-1)^{x+y}$

Фурье-пара гауссова фильтра

$$H(u) = Ae^{-u^2/2\sigma^2}$$



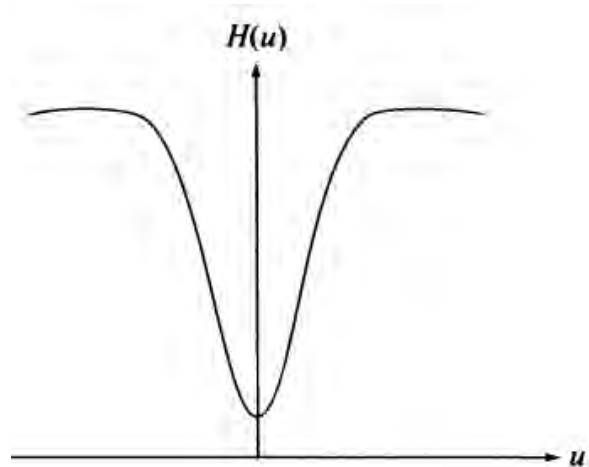
$$h(x) = \sqrt{2\pi}\sigma Ae^{-2\pi^2\sigma^2 x^2}$$



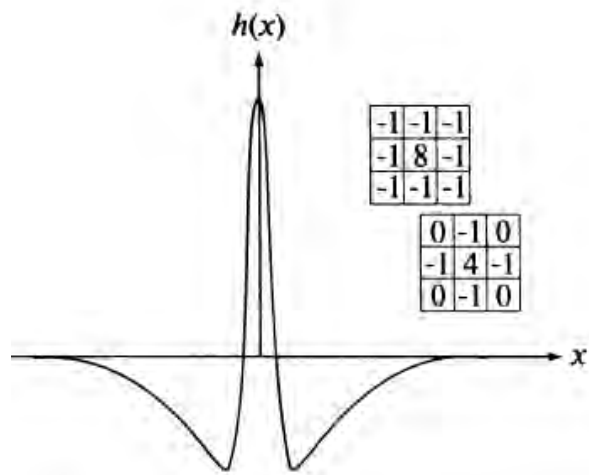
Низкочастотный

Фурье-пара гауссова фильтра

$$H(u) = Ae^{-u^2/2\sigma^2} - Be^{-u^2/2\sigma_2^2}$$

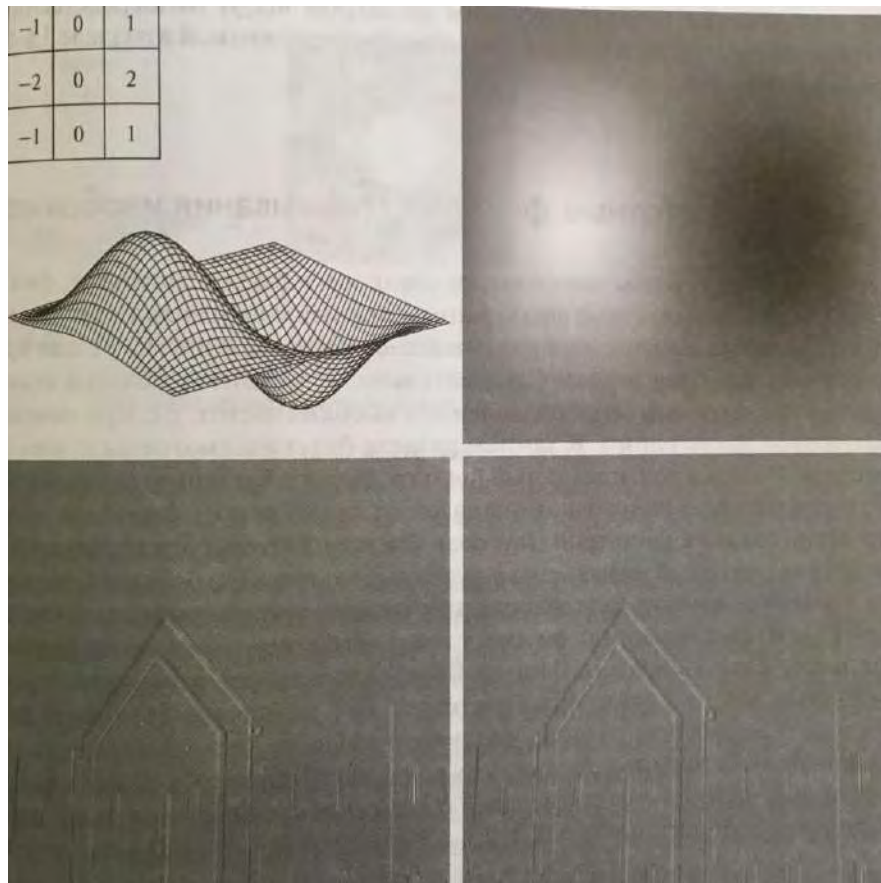


$$h(x) = \sqrt{2\pi}\sigma Ae^{-2\pi^2\sigma^2x^2} - \sqrt{2\pi}\sigma_2 B e^{-2\pi^2\sigma_2^2x^2}$$



Высокочастотный

Пространственная и частотная маски



Частотная
фильтрация

Пространственная
фильтрация

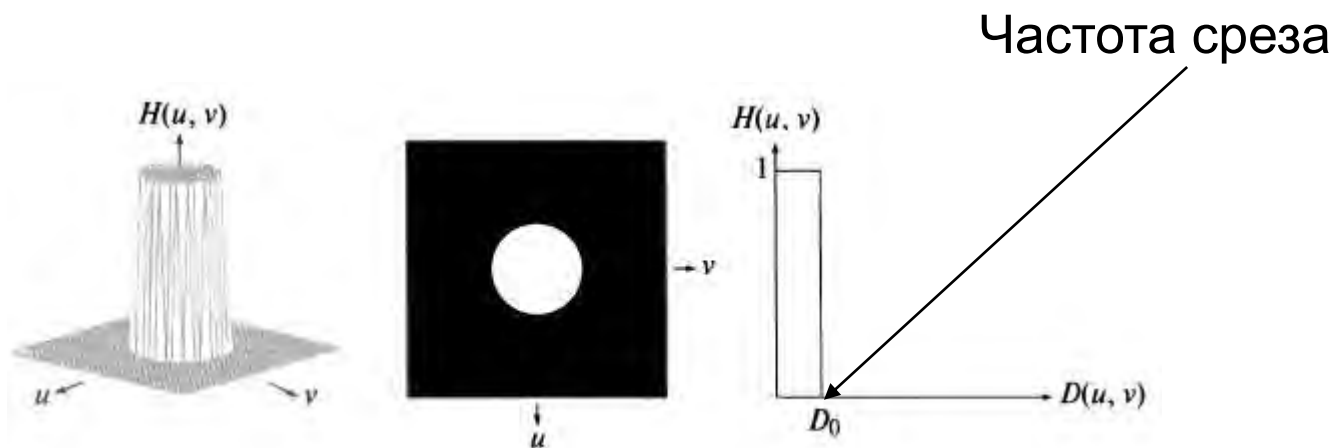
Сглаживающие низкочастотные фильтры

- Идеальные фильтры низких частот – очень резкий
- Фильтр Баттерворта – переходный (зависит от порядка)
- Гауссов фильтр – очень гладкий

Идеальные фильтры низких частот (ИФНЧ)

$$H(u, v) = \begin{cases} 1, & \text{при } D(u, v) \leq D_0 \\ 0, & \text{при } D(u, v) > D_0 \end{cases}$$

$$D(u, v) = \left[(u - M/2)^2 + (v - N/2)^2 \right]^{1/2}$$



Величины для сравнение низкочастотных фильтров

Полная энергия

$$P_T = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} P(u, v)$$

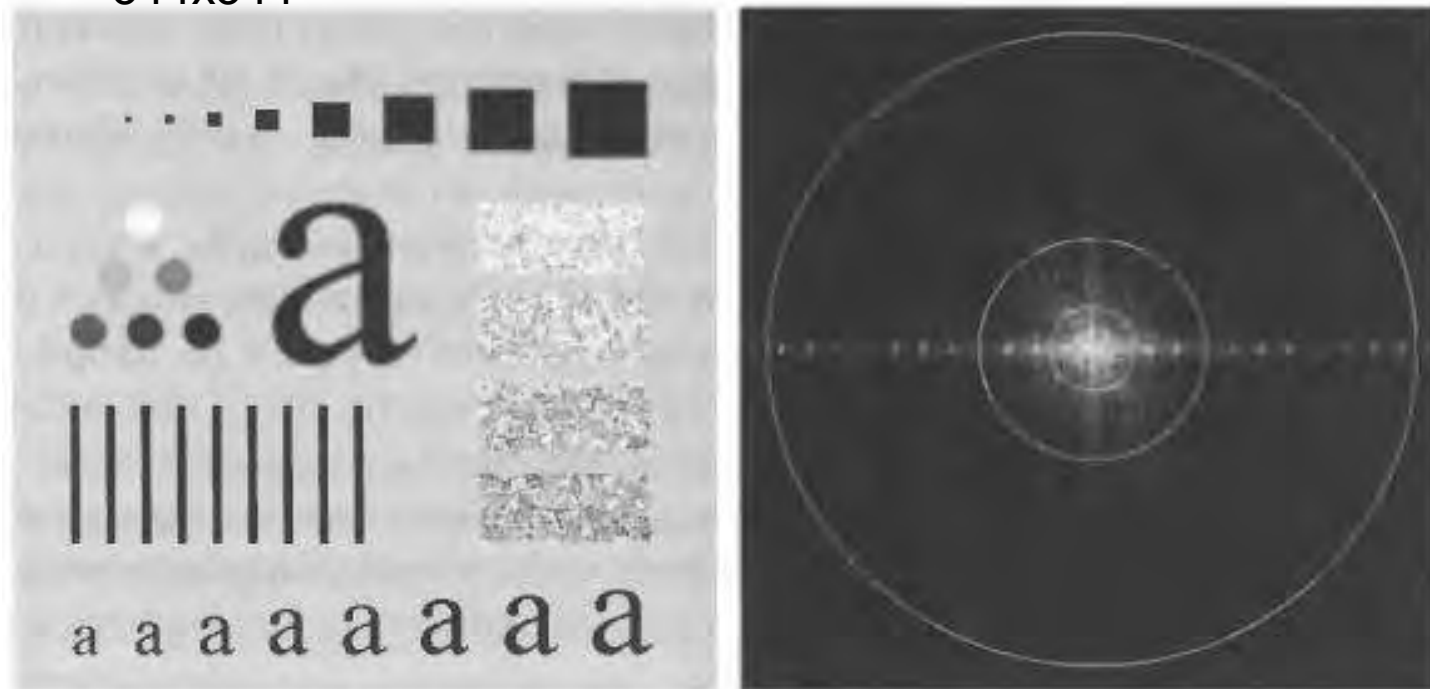
$$P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v)$$

$$\alpha = 100 \left[\sum_u \sum_v P(u, v) / P_T \right]$$

Частота $r(\alpha)$ – радиус круга с центром в центре частотного прямоугольника, содержащего α процентов энергии спектра

Пример. Энергия изображения как функция расстояния от центра ДПФ

344x344



$R = 5, 15, 30, 80, 230$

$\alpha = 92.0, 94.6, 96.4, 98.0, 99.5$

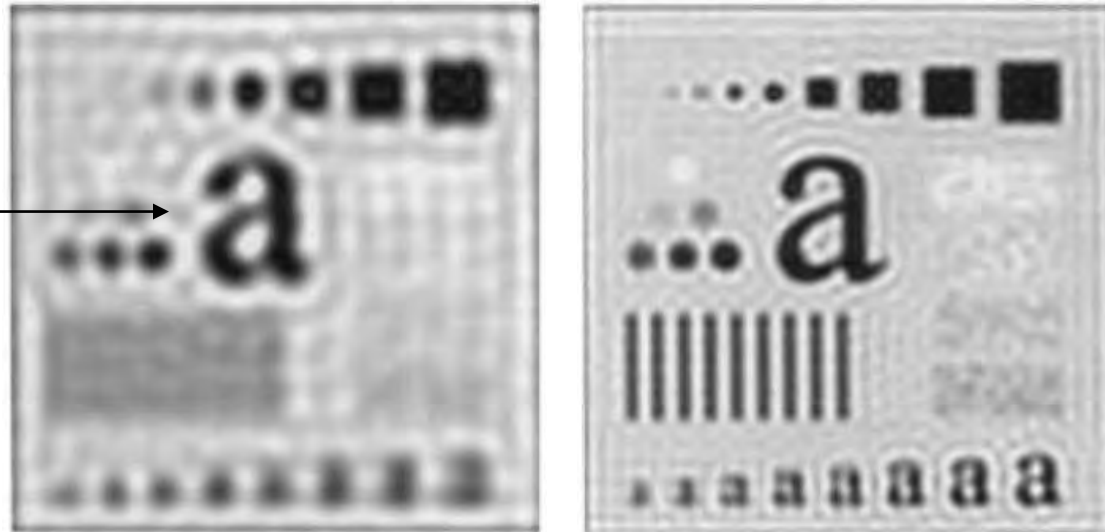
Результаты фильтрации



$$R = 5, \alpha = 92$$

Результаты фильтрации

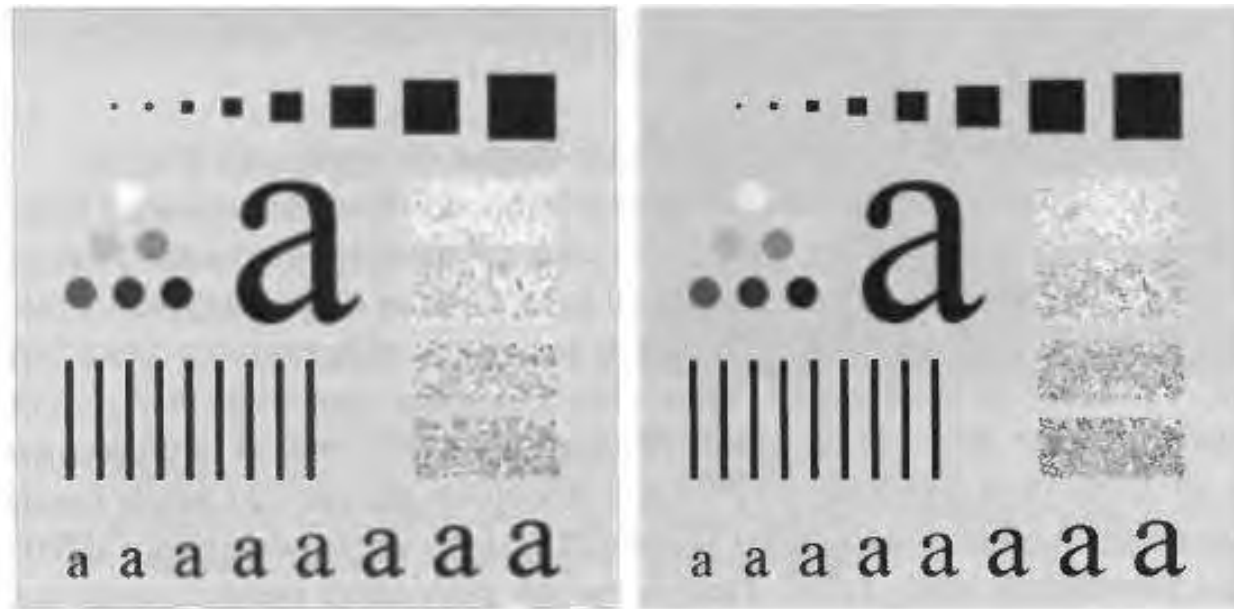
Звон
(явление
Гиббса)



$$R = 15, \alpha = 94.6$$

$$R = 30, \alpha = 96.4$$

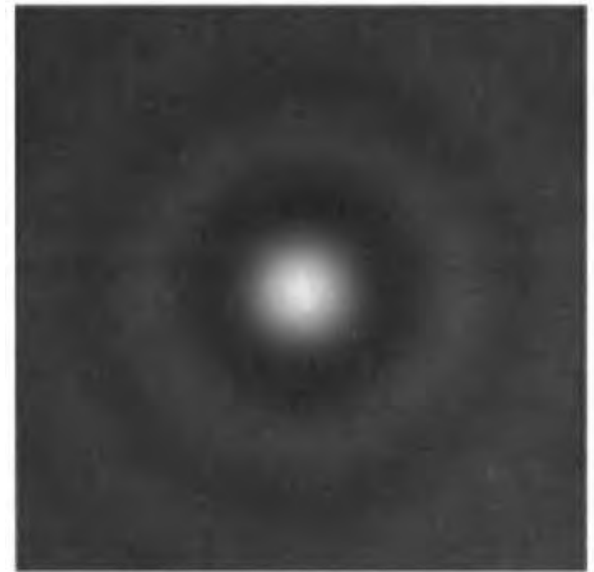
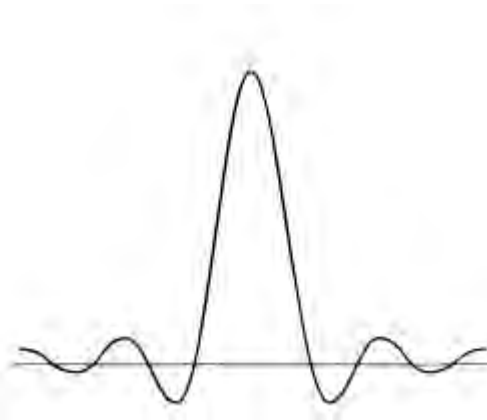
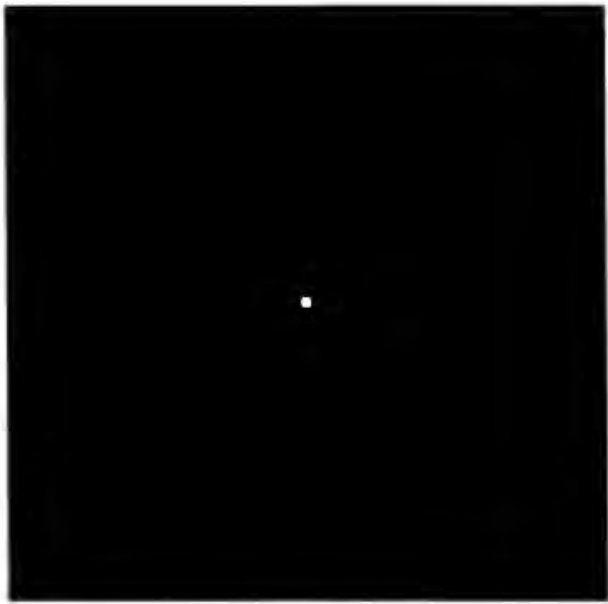
Результаты фильтрации



$R = 80, \alpha = 98$

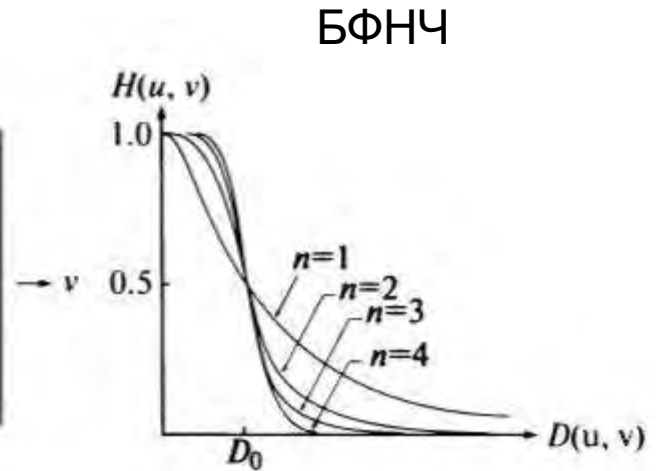
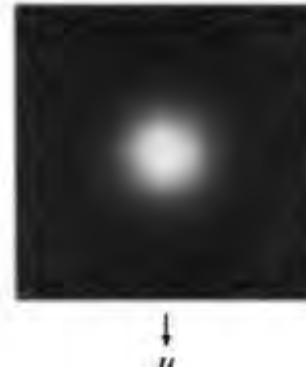
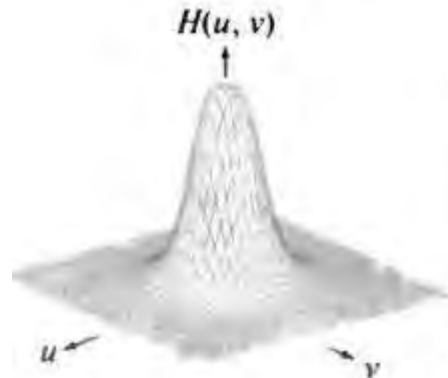
$R = 230, \alpha = 99.5$

Откуда берется звон



$R = 5$, $M \times N = 1000 \times 1000$

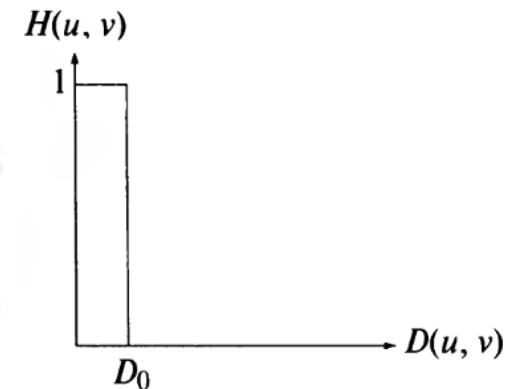
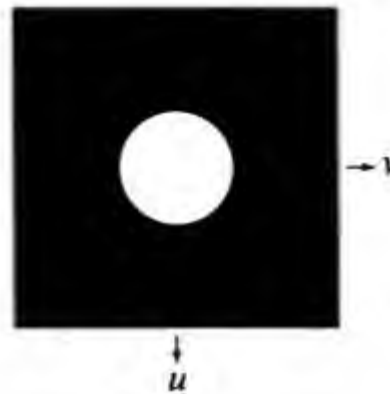
Фильтры низких частот Баттерворта (БФНЧ)



$$H(u, v) = \frac{1}{1 + [D(u, v) / D_0]^{2n}}$$

$$D(u, v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$

$$D(u, v) = D_0 \quad H(u, v) = 0,5$$



Результаты применения БФНЧ (n=2)



$$R = 5, \alpha = 92$$

Результаты применения БФНЧ (n=2)

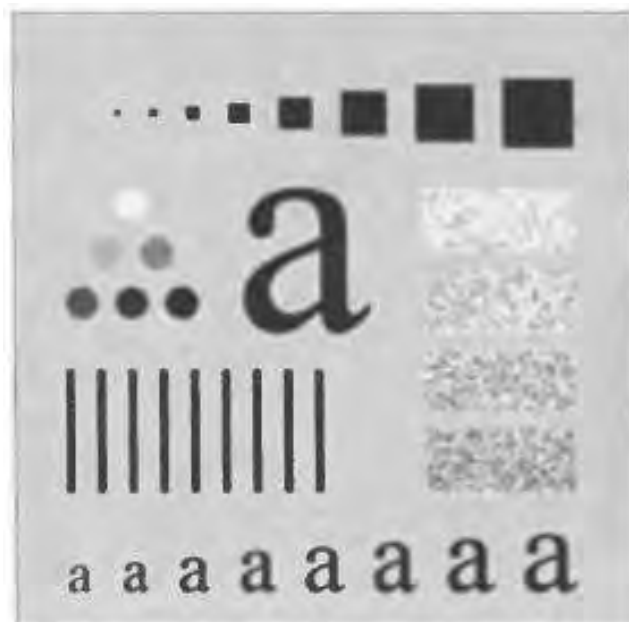


$R = 15, \alpha = 94.6$



$R = 30, \alpha = 96.4$

Результаты применения БФНЧ (n=2)

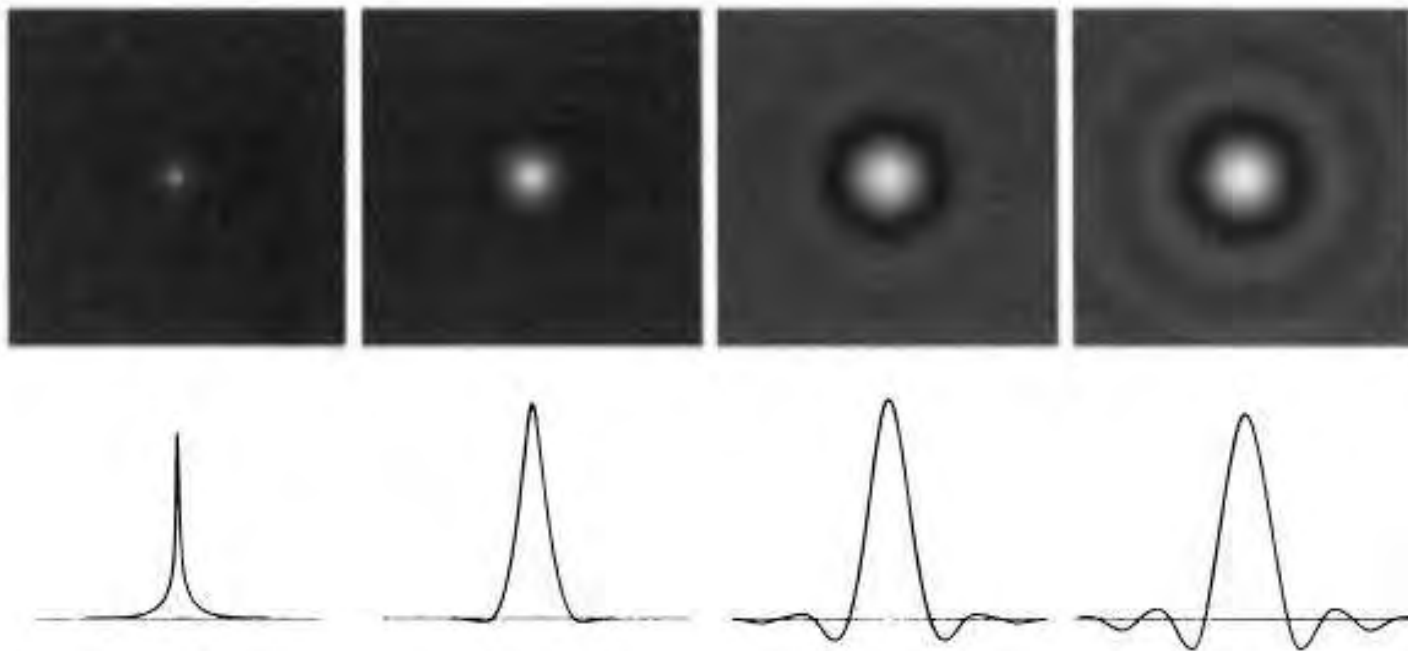


$R = 80, \alpha = 98$



$R = 230, \alpha = 99.5$

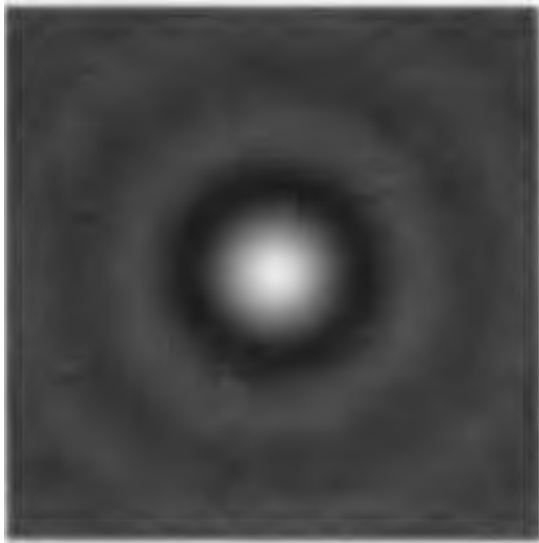
Пространственное представление БФНЧ и профили яркости



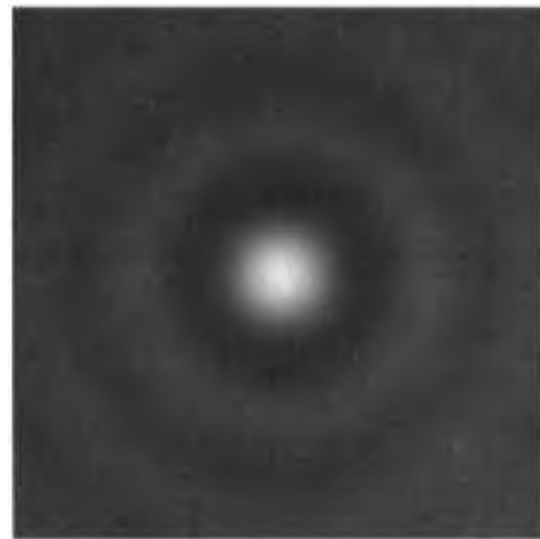
$$D_0 = 5$$

$$n = 1, 2, 5, 20$$

Сравнение БФНЧ ($n=20 \rightarrow \infty$) и ИФНЧ



БФНЧ ($n=20$)



ИФНЧ

$D_0=5$

Гауссовы фильтры низких частот (ГФНЧ)

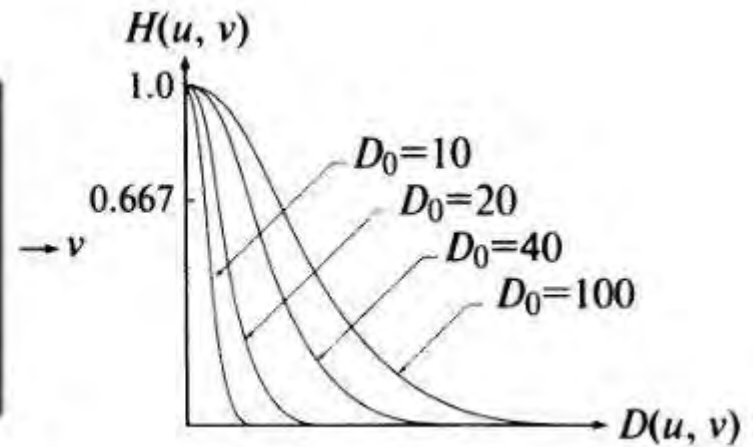
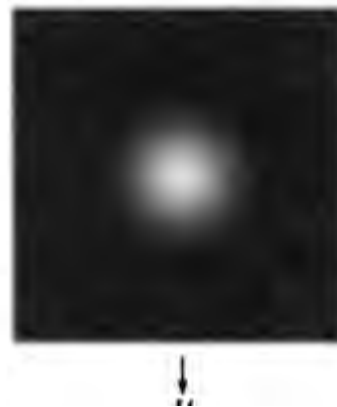
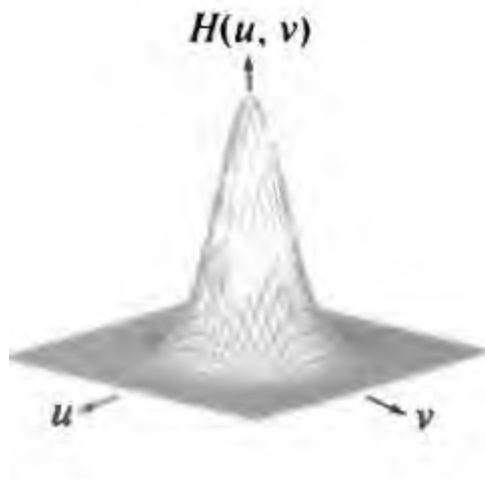
$$H(u) = Ae^{-u^2/2\sigma^2}$$

$$H(u, v) = e^{-D^2(u, v)/2\sigma^2}$$

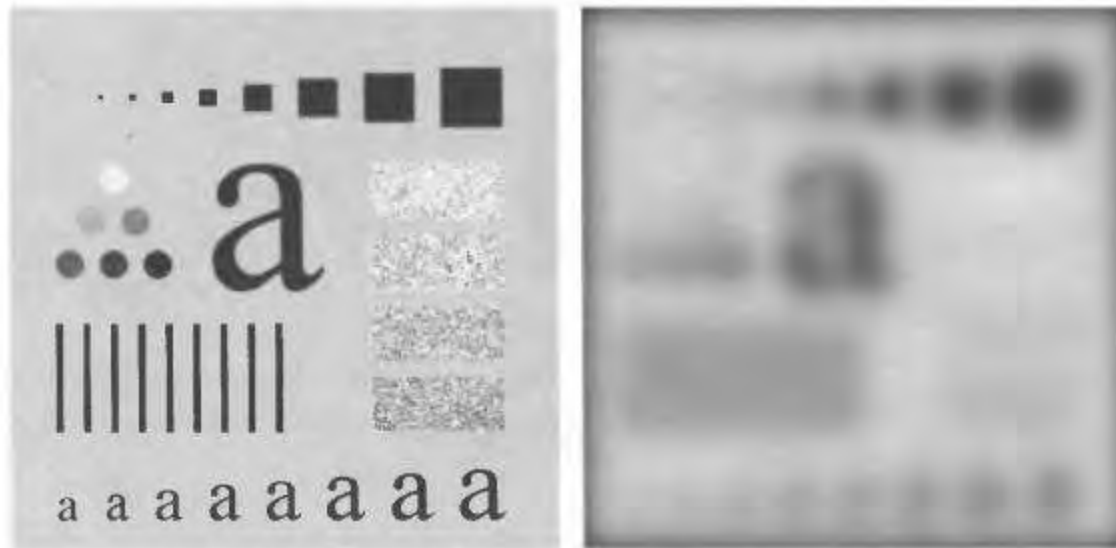
$$H(u, v) = e^{-D^2(u, v)/2D_0^2}$$

$$D(u, v) = D_0$$

$$H(u, v) = 0,607$$



Результаты применения ГФНЧ



$$R = 5, \alpha = 92$$

Результаты применения ГФНЧ

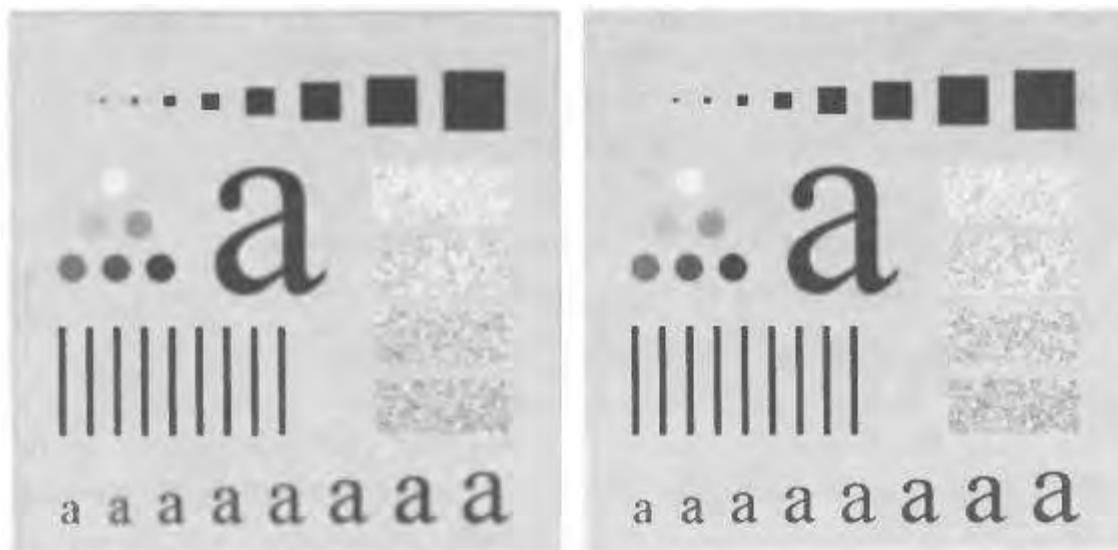


$R = 15, \alpha = 94.6$



$R = 30, \alpha = 96.4$

Результаты применения ГФНЧ



$R = 80, \alpha = 98$

$R = 230, \alpha = 99.5$

Сравнение БФНЧ и ГФНЧ ($D_0=15$)

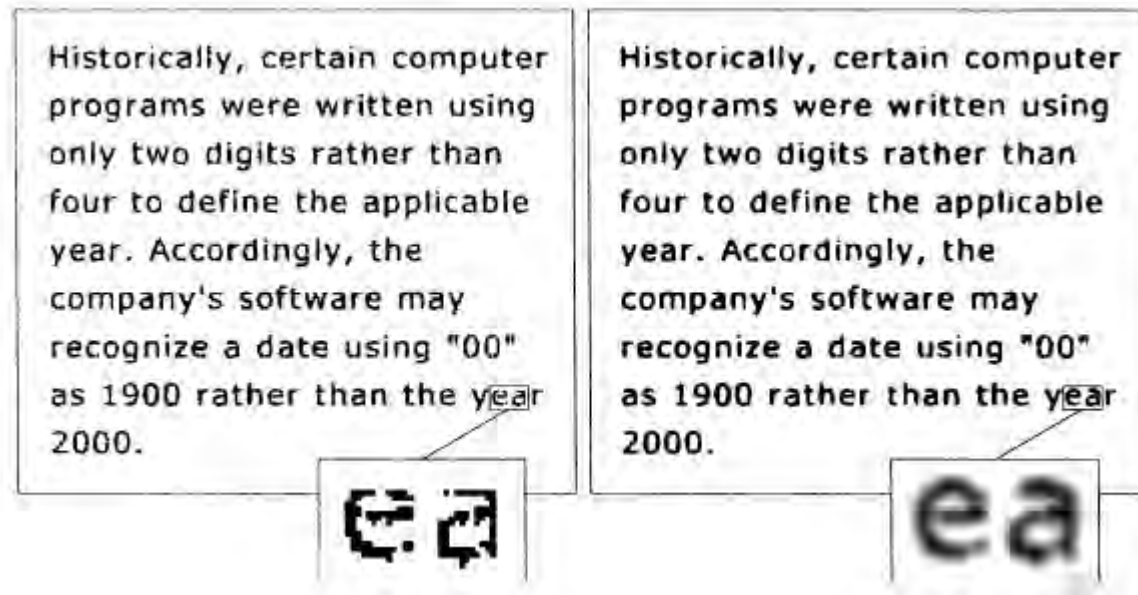


$R = 15, \alpha = 94.6$



$R = 15, \alpha = 94.6$

Примеры низкочастотной фильтрации в распознавании текста



ГФНЧ ($D_0=80$) размер изображения 444 x 508

Примеры низкочастотной фильтрации в полиграфии



ГФНЧ ($D_0=100$, $D_0=80$) размер изображения 1028 x 732

Частотные фильтры повышения резкости

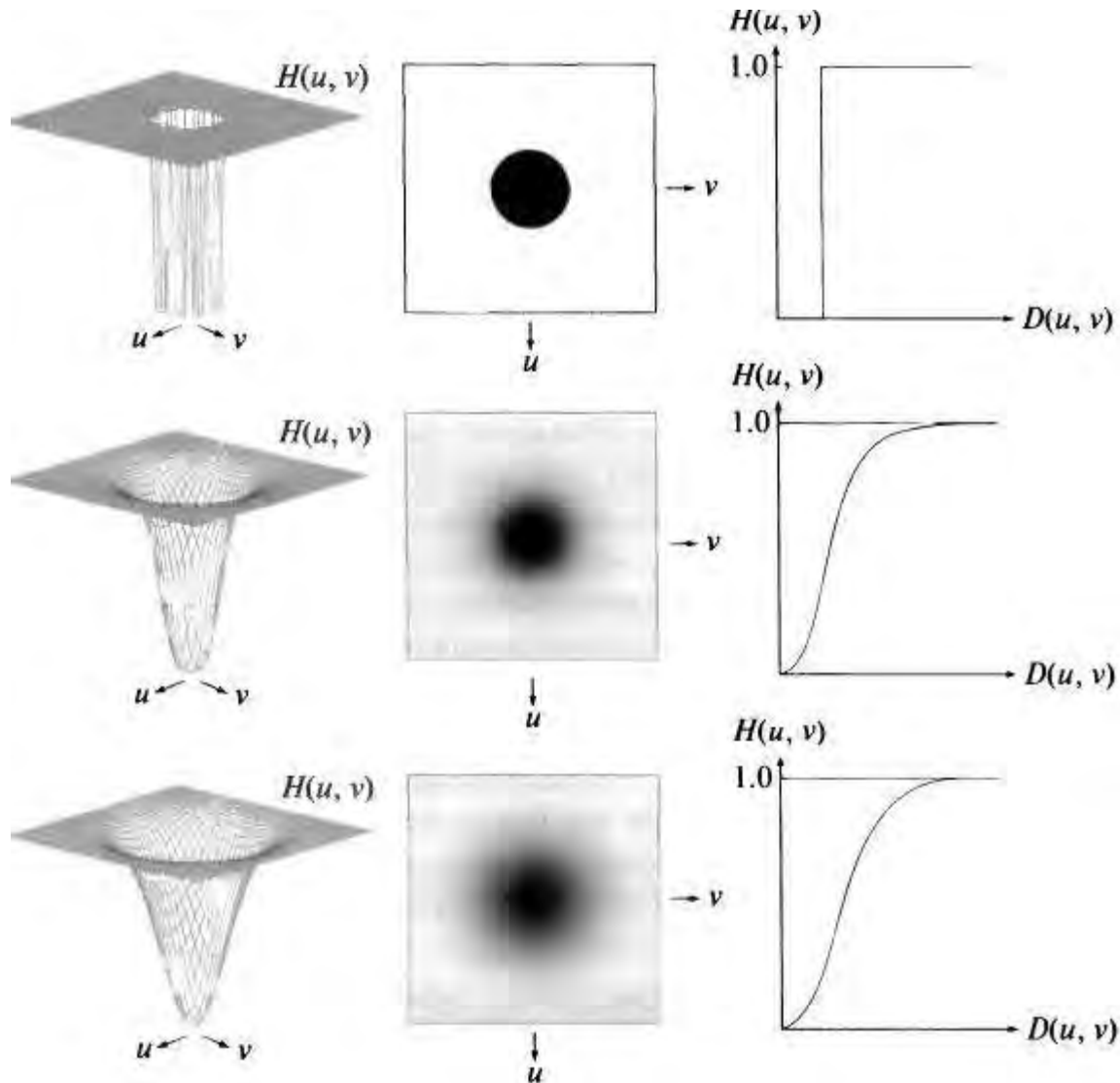
Рассматриваем центрально-симметричные фильтры нулевого фазового сдвига

Передаточная функция $H_{hp}(u, v) = 1 - H_{lp}(u, v)$

Высокочастотные фильтры

- Идеальные фильтры высоких частот – очень резкий
- Фильтр Баттерворта – переходный (зависит от порядка)
- Гауссов фильтр – очень гладкий

Высокочастотные фильтры

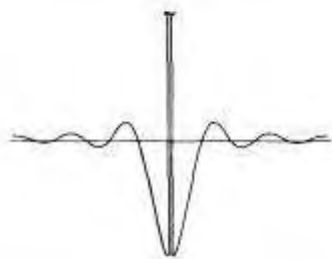


Идеальные фильтры низких частот (ИФВЧ)

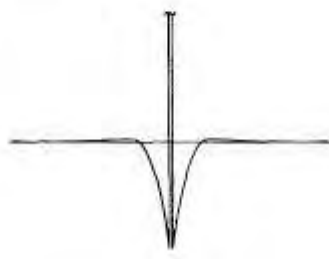
$$H(u, v) = \begin{cases} 0, & \text{при } D(u, v) \leq D_0 \\ 1, & \text{при } D(u, v) > D_0 \end{cases}$$

$$D(u, v) = \left[(u - M / 2)^2 + (v - N / 2)^2 \right]^{1/2}$$

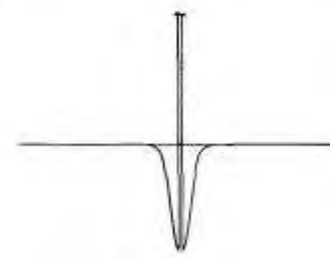
Представление в пространственной области



Идеальный



Баттерворта



Гауссов

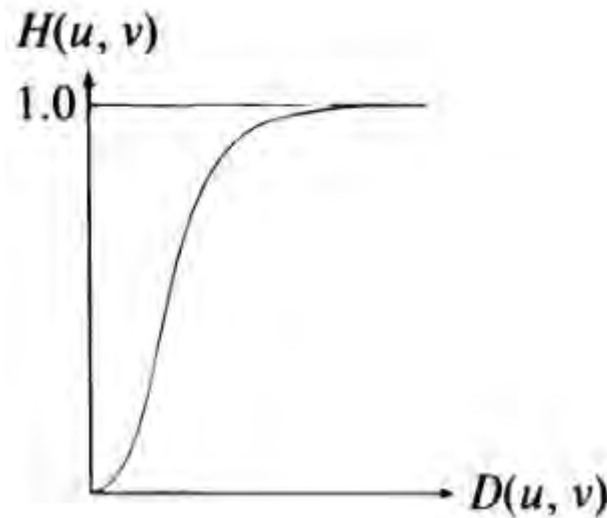
Результат применения ИФВЧ



$D_0 = 15, 30, 80$

Фильтры высоких частот Баттерворта (БФВЧ)

$$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}}$$



Результаты применения БФВЧ (n=2)



$D_0 = 15, 30, 80$

Гауссовы фильтры высоких частот (ГФВЧ)

$$H(u, v) = 1 - e^{-D^2(u, v) / 2D_0^2}$$

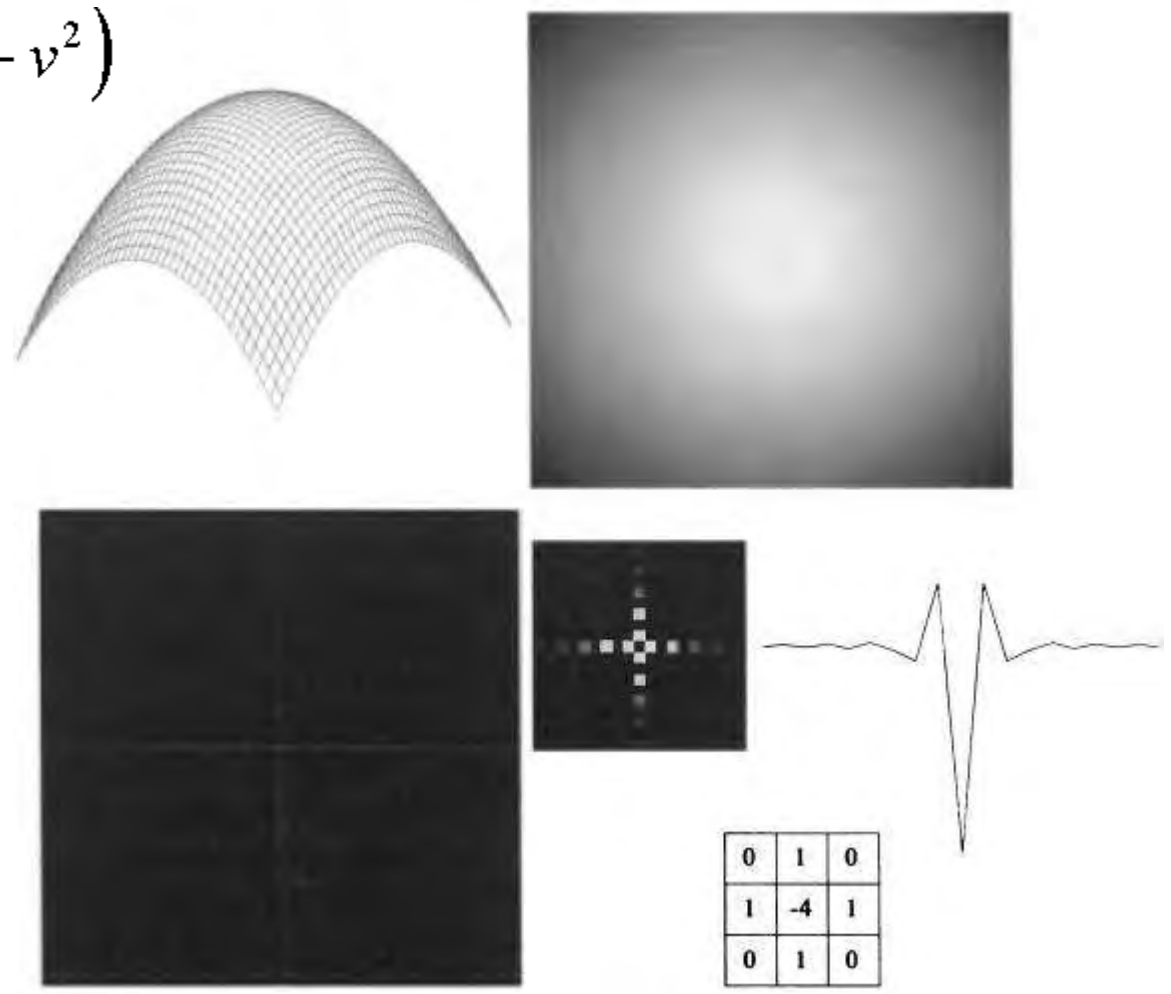
Результаты применения ГФВЧ



$D_0 = 15, 30, 80$

Лапласиан в частотной и пространственной областях

$$H(u, v) = -4\pi^2 (u^2 + v^2)$$

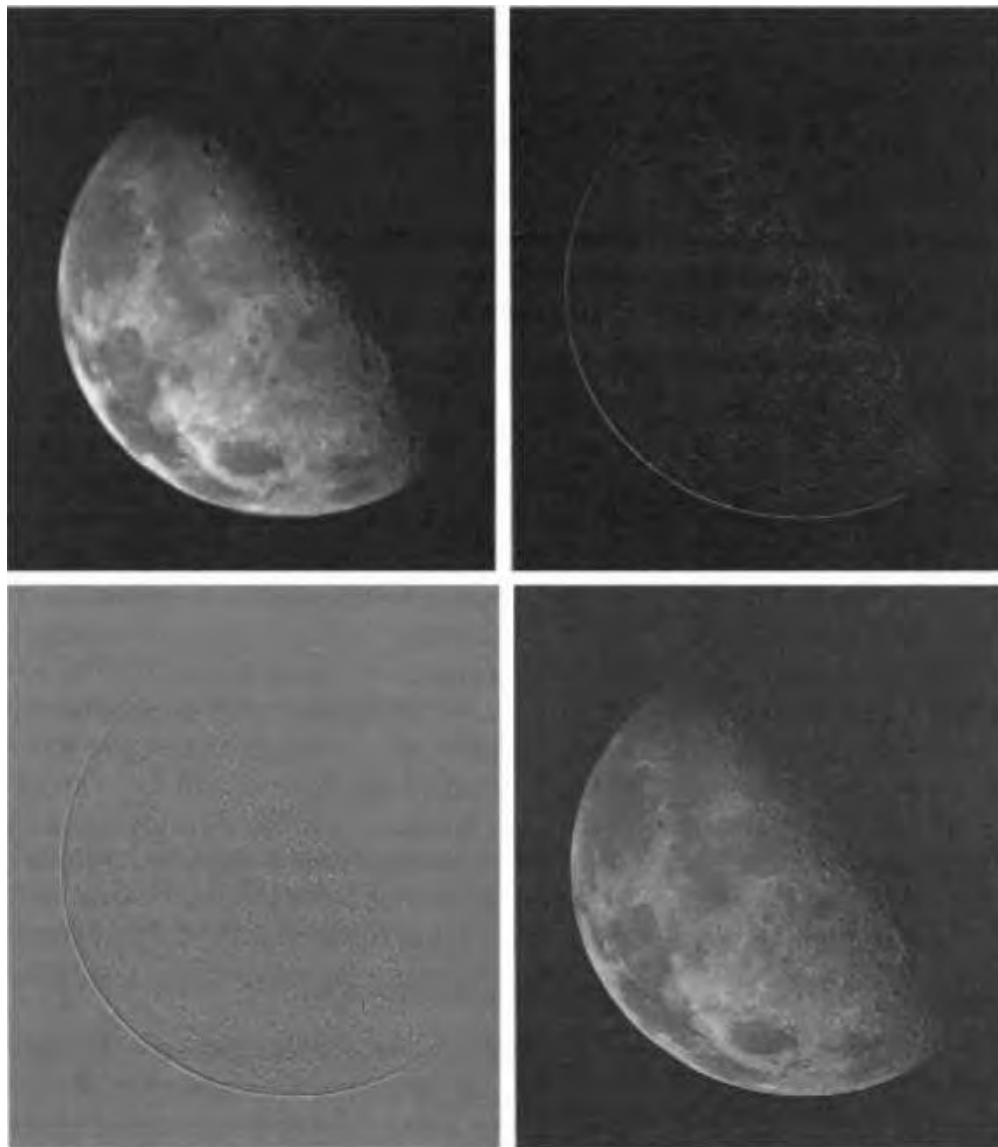


Для получения улучшенного изображения

вычитаем Лапласиан (изображение полученное с использованием частотного фильтра Лапласа) из оригинала

$$g(x, y) = f(x, y) - \nabla^2(x, y)$$

Применение лапласиана в частотной области



Вариации фильтров

- Нерезкое маскирование
- Высокочастотная фильтрация с подъемом частотной характеристики
- Фильтрация с усилением высоких частот

Нерезкое маскирование

нерезкое маскирование = оригинал – сглаженная копия

$$f_{hp}(x, y) = f(x, y) - f_{lp}(x, y)$$

Нерезкое маскирование в частотной области

$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

Высокочастотная фильтрация с подъемом частотной характеристики

$$f_{hb}(x, y) = Af(x, y) - f_{lp}(x, y)$$

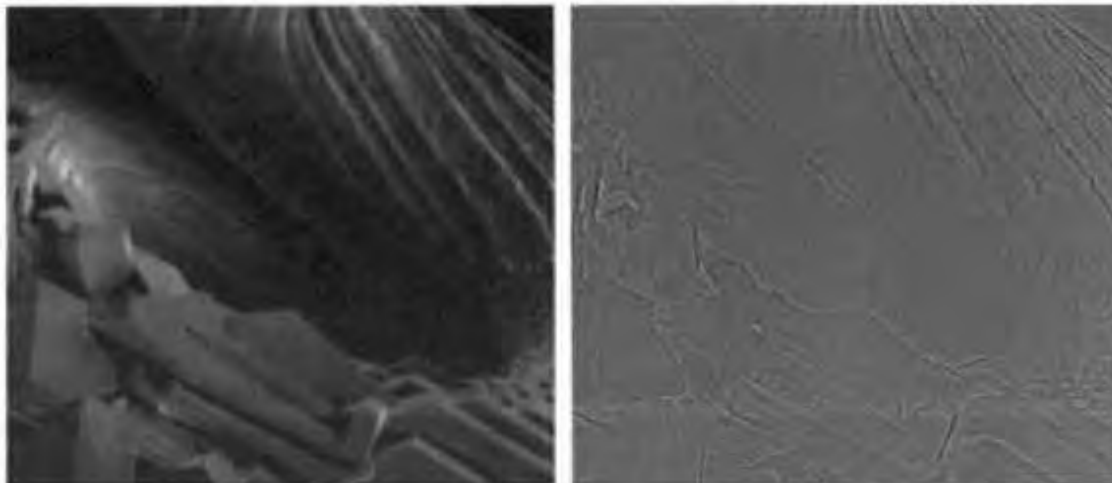
$$f_{hb}(x, y) = (A-1)f(x, y) + f(x, y) - f_{lp}(x, y)$$

$$f_{hb}(x, y) = (A-1)f(x, y) + f_{hp}(x, y)$$

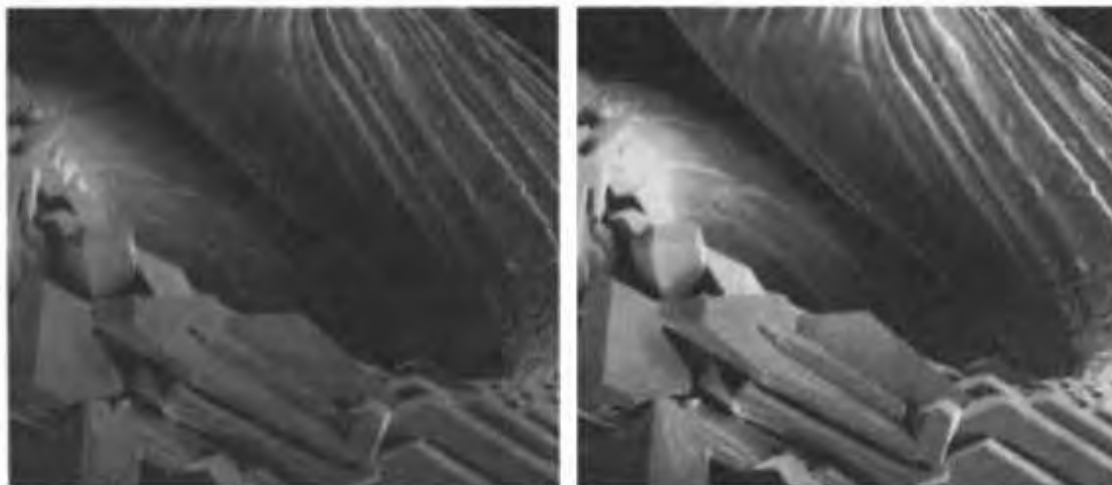
Фильтрация с подъемом частотной характеристики

$$H_{hb}(u, v) = (A - 1) + H_{hp}(u, v)$$

Результат ФВЧ с подъемом частотной характеристики



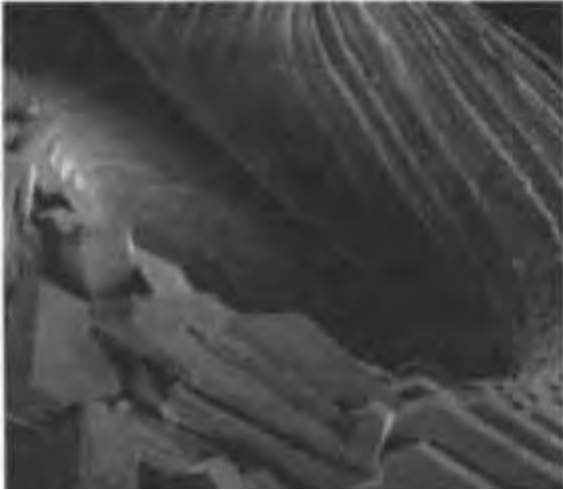
Лапласиан



$A=2.0$

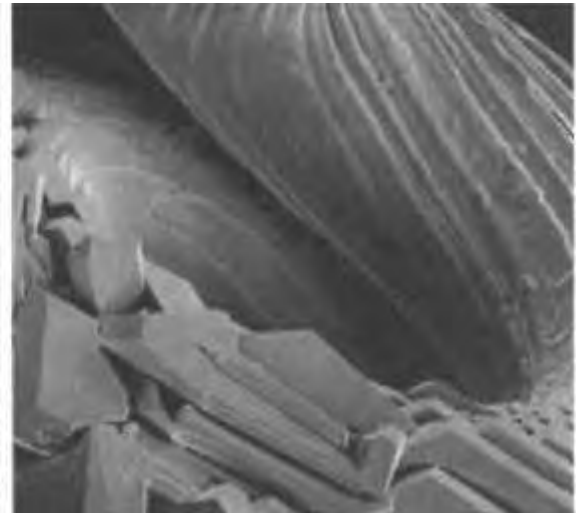
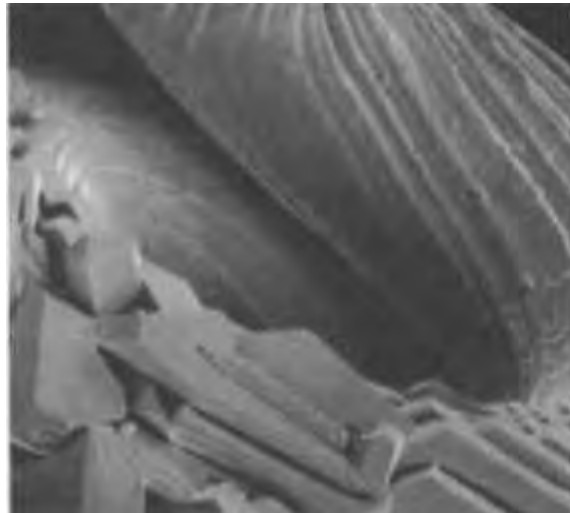
$A=2.7$

Сравнение частотного лапласиана с пространственным



частотный

пространственный



Результат применения фильтрации с усилением высоких частот



БФВЧ $n=2$
 $D_0=5\%$
ВЫСОТЫ

Усиление высоких частот +Эквализация гистограммы

Гомоморфная фильтрация

Сжатие яркостного диапазона и усиление контраста

Еще один вариант представления изображения

$$f(x, y) = i(x, y)r(x, y)$$

Сложности частотной обработки

$$\mathfrak{T}\{f(x, y)\} \neq \mathfrak{T}\{i(x, y)\}\mathfrak{T}\{r(x, y)\}$$

Рассмотрим величину

$$z(x, y) = \ln f(x, y) = \ln i(x, y) + \ln r(x, y)$$

Тогда

$$\mathfrak{T}\{z(x, y)\} = \mathfrak{T}\{\ln f(x, y)\} = \mathfrak{T}\{\ln i(x, y)\} + \mathfrak{T}\{\ln r(x, y)\}$$

ИЛИ

$$Z(u, v) = F_i(u, v) + F_r(u, v)$$

Применим фильтрацию

$$S(u, v) = H(u, v)Z(u, v) = H(u, v)F_i(u, v) + H(u, v)F_r(u, v)$$

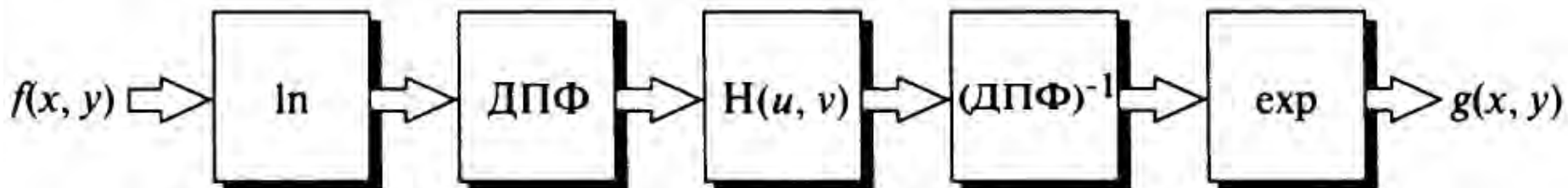
В пространственной области имеем

$$s(x, y) = \mathfrak{F}^{-1}\{S(u, v)\} = \mathfrak{F}^{-1}\{H(u, v)F_i(u, v)\} + \mathfrak{F}^{-1}\{H(u, v)F_r(u, v)\}$$

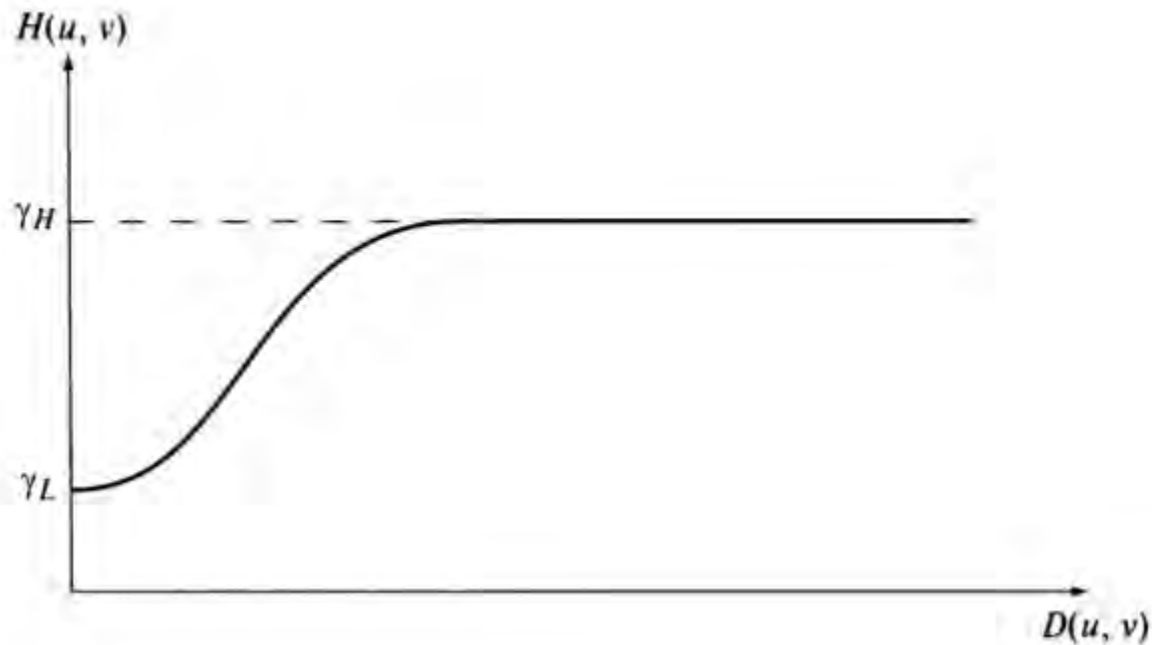
$$s(x, y) = i'(x, y) + r'(x, y)$$

$$g(x, y) = e^{s(x, y)} = e^{i'(x, y)} \cdot e^{r'(x, y)} = i_0(x, y) \cdot r_0(x, y)$$

Схема метода гомоморфной фильтрации



Профиль центрально-симметричной передаточной функции фильтра



$$\gamma_L < 1$$

$$\gamma_H > 1$$

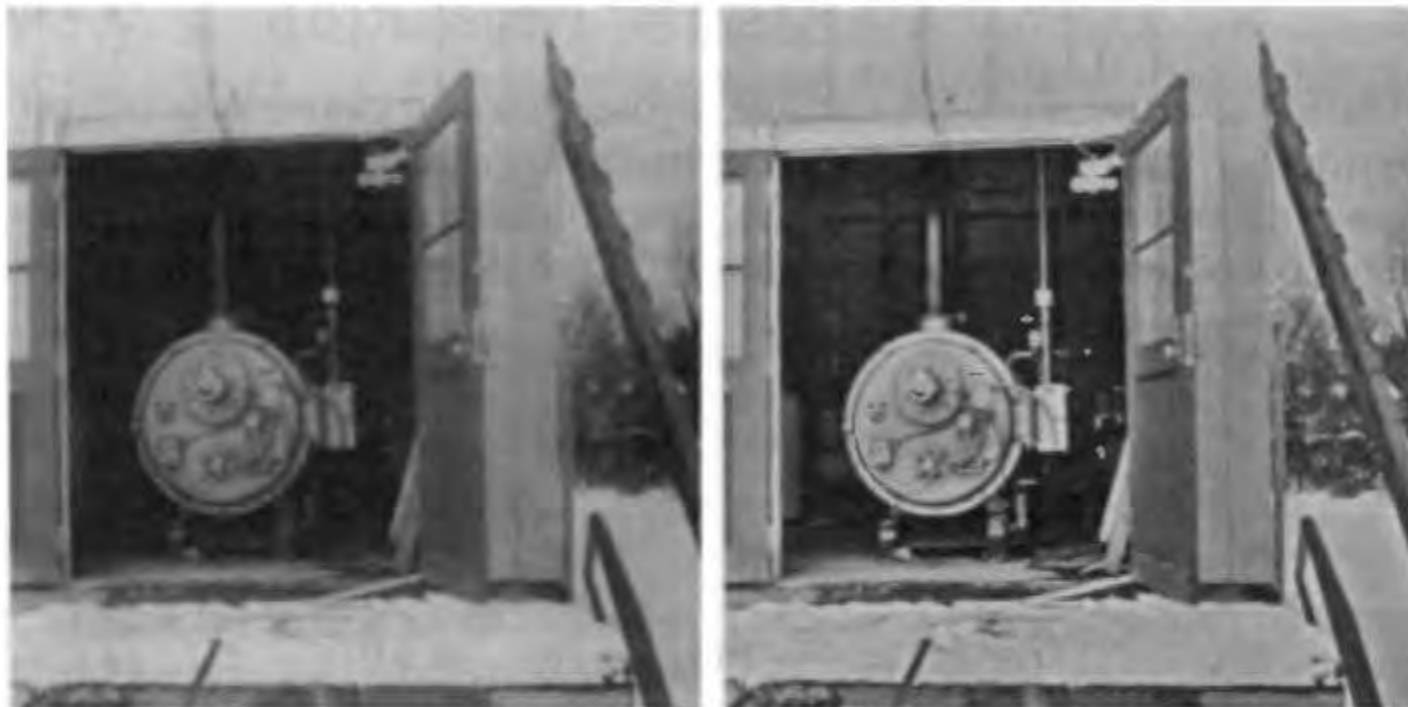
Аппроксимация модифицированным ГФВЧ

$$H(u, v) = (\gamma_H - \gamma_L) \left| 1 - e^{(-c(D^2(u, v)/D_0^2))} \right| + \gamma_L$$

Константа c управляет крутизной наклона

Похож на фильтр усиления высоких частот

Результат гомоморфной фильтрации



$$\gamma_L = 0.5 \quad \gamma_H = 2.0$$

РАБОТА С КОНТУРАМИ

Лекция 7

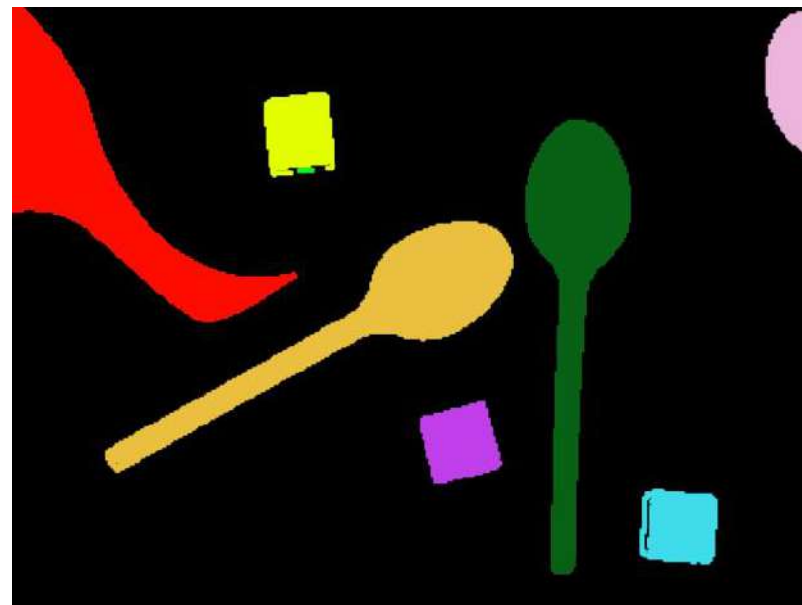
План

- **Прослеживание контуров**
- Связывание контуров
- Анализ контуров
- Метод активных контуров

Различные виды изображений: различные способы получения контуров и различные проблемы

- Бинарные
- Полутоновые
- Цветные

Поиск границ и контуров на бинарном изображении



Операция оконтуривания объекта

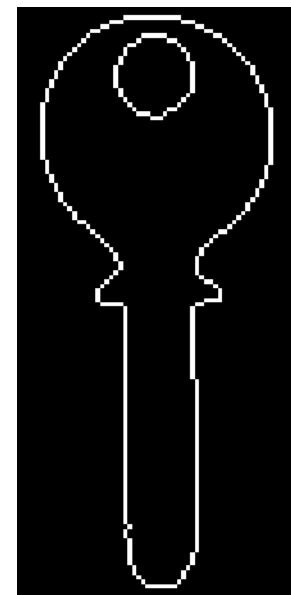
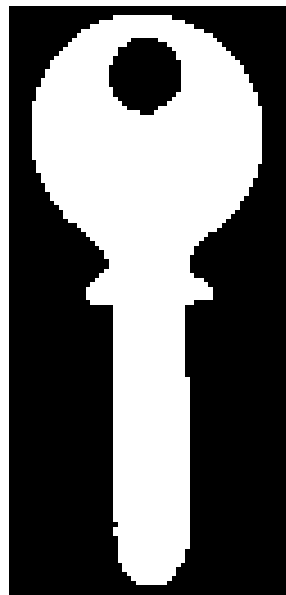
При работе с бинарными изображениями контуры объекта можно получить с помощью операций математической морфологии

Внутреннее оконтуривание

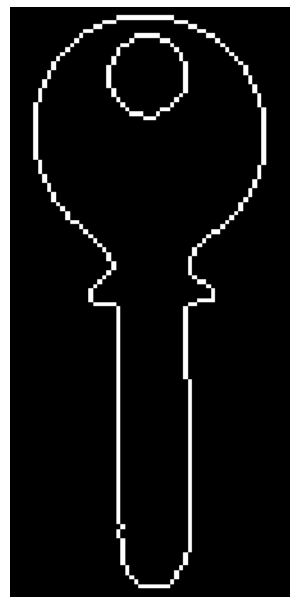
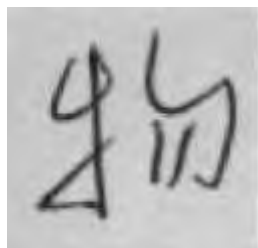
$$C_1 = A - (A (-) B)$$

Внешнее оконтуривание

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



Как отследить контур



Виды обработки при получении контуров

- **Локальная**
- **Глобальная**

Классификация пикселей для прослеживания контуров в бинарных изображениях

- Изолированный пиксел
- Вершина (начало или конец сегмента)
- Внутренний пиксел сегмента
- Угловая точка
- Стык

	1	2	3	4	5
1	1	0	0	0	1
2	0	1	0	1	0
3	0	0	1	0	0
4	0	0	1	0	0
5	0	0	1	1	1

Маркированное контурное изображение

Операции для алгоритма прослеживания сегментов

1. Прослеживание нового сегмента
2. Добавление внутреннего пиксела к сегменту
3. Завершение сегмента
4. Поиск стыка
5. Поиск угла

	1	2	3	4	5
1	1	0	0	0	1
2	0	1	0	1	0
3	0	0	1	0	0
4	0	0	1	0	0
5	0	0	1	1	1

Маркированное контурное изображение

Результат применения алгоритма для прослеживания контуров к маркированному изображению

	1	2	3	4	5
1	1	0	0	0	1
2	0	1	0	1	0
3	0	0	1	0	0
4	0	0	1	0	0
5	0	0	1	1	1

Segment ID	Length	List
1	3	(1,1)(2,2)(3,3)
2	3	(1,5)(2,4)(3,3)
3	3	(3,3)(4,3)(5,3)
4	3	(5,3)(5,4)(5,5)

Виды обработки при получении контуров

- Локальная
- **Глобальная**

Выделение границ - карты силы краев



Робертса

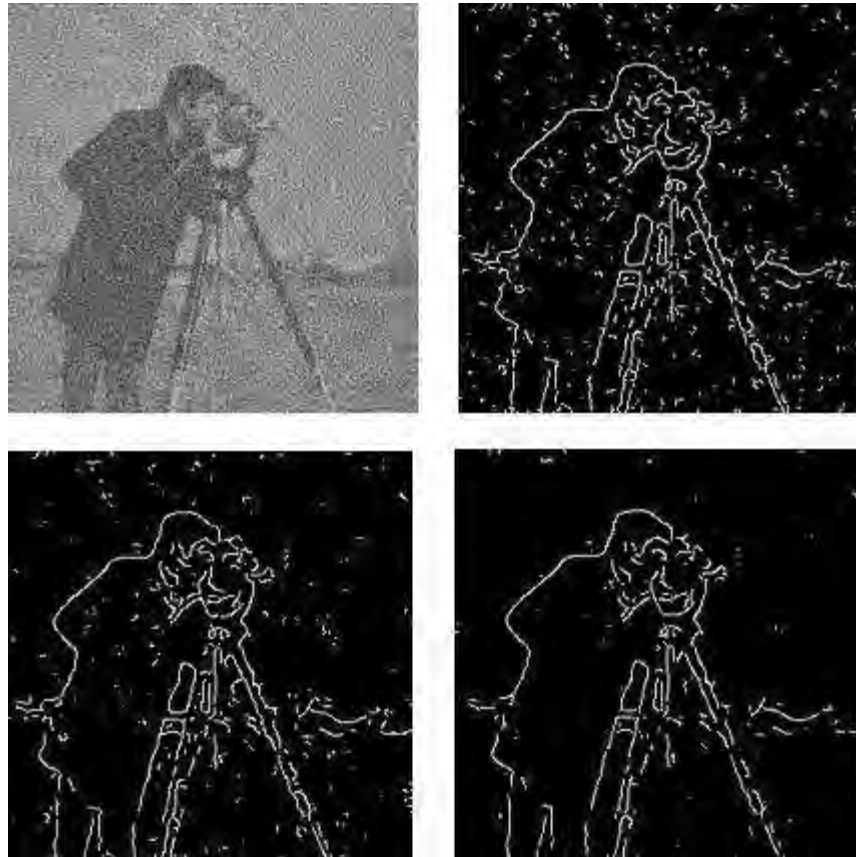


Превитт



Собеля

Методы, основанные на операторах выделения краёв



Выделение краёв

- Вычисление градиента – это еще не всё...



Исходное изображение



Карта силы краев

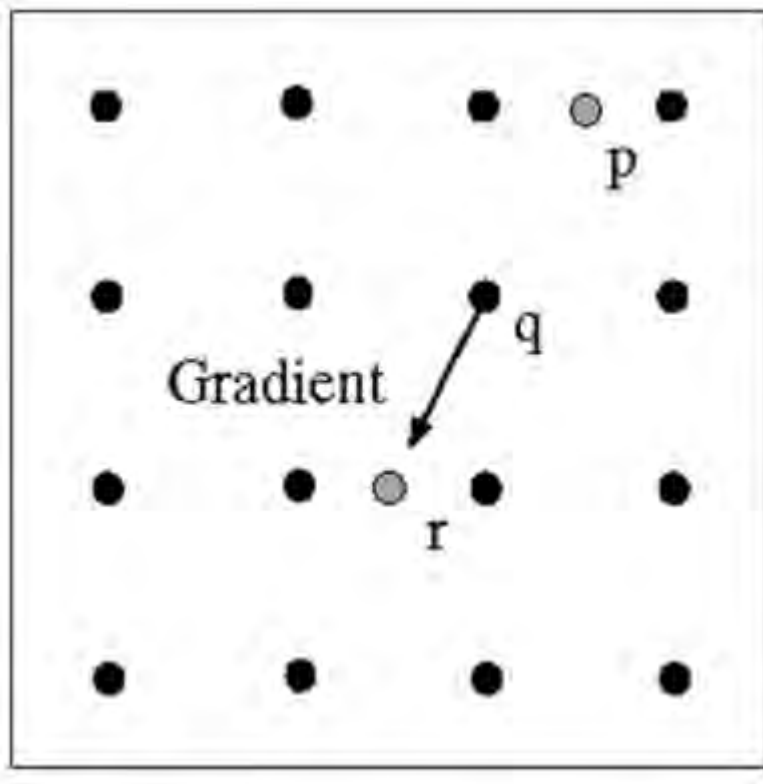
- Чего не хватает?
 - Точности – края «толстые» и размытые
 - Информации о связности

Детектор Canny

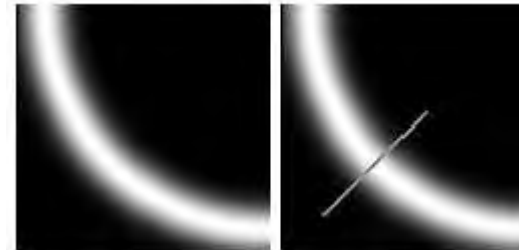
- Свертка изображения с ядром – размывание изображения фильтром Гаусса
- Поиск значения и направления градиента
- Подавление немаксимумов (Non-maximum suppression)
 - выделение локальных максимумов
 - утоньшение полос в несколько пикселей до одного пикселя
- Связывание краев и обрезание по порогу (гистерезис)
 - Определяем пороги: нижний и верхний
 - Верхний порог используем для инициализации кривых
 - Нижний порог используем для продолжения кривых

J.Canny, A Computational Approach To Edge Detection

Поиск локальных максимумов



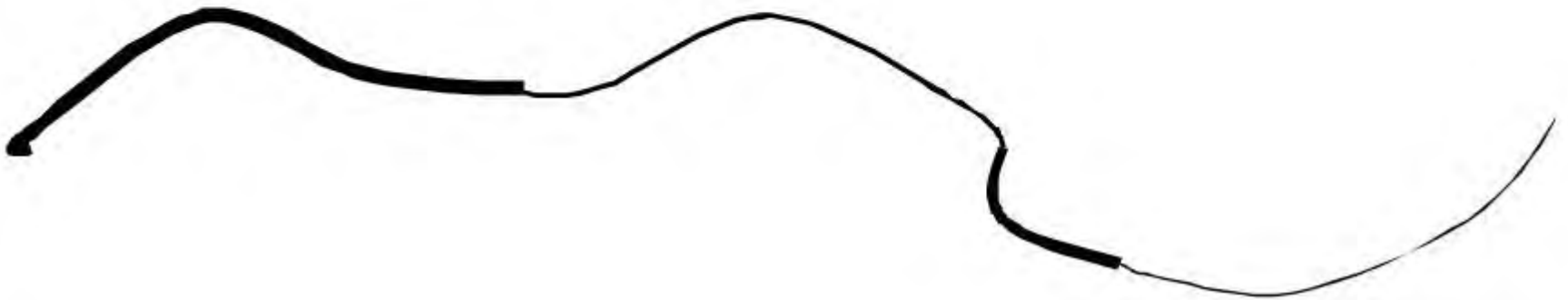
Максимум достигается в q , если значение больше p и r .



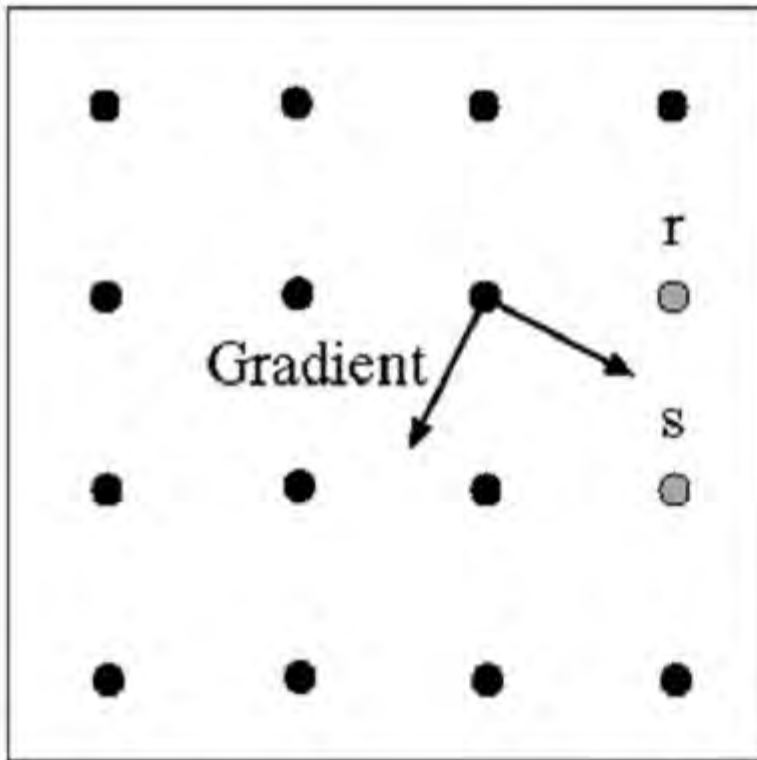
Source: D. Forsyth

Отсечение по порогу

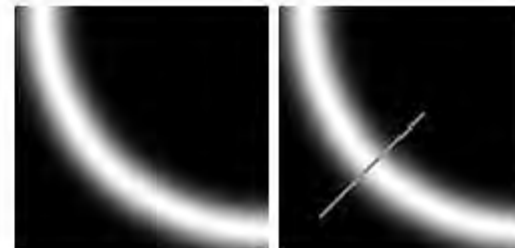
- Проверяем точку, чтобы значение градиента было выше порога
 - Используем **гистерезис**
 - Большой порог для начала построения кривой и низкий порог для продолжения края (связывания)



Связывание точек



Пусть отмеченная точка – край. Строим касательную к границе (нормаль к направлению градиента) и используем ее для предсказания новой точки (это либо s либо r).



Source: D. Forsyth

Эффект гистерезиса



Исходное изображение



Высокий порог
(сильные края)



Низкий порог
(слабые края)



Порог по гистерезису

Пример

Карта силы краев



Утончение краев



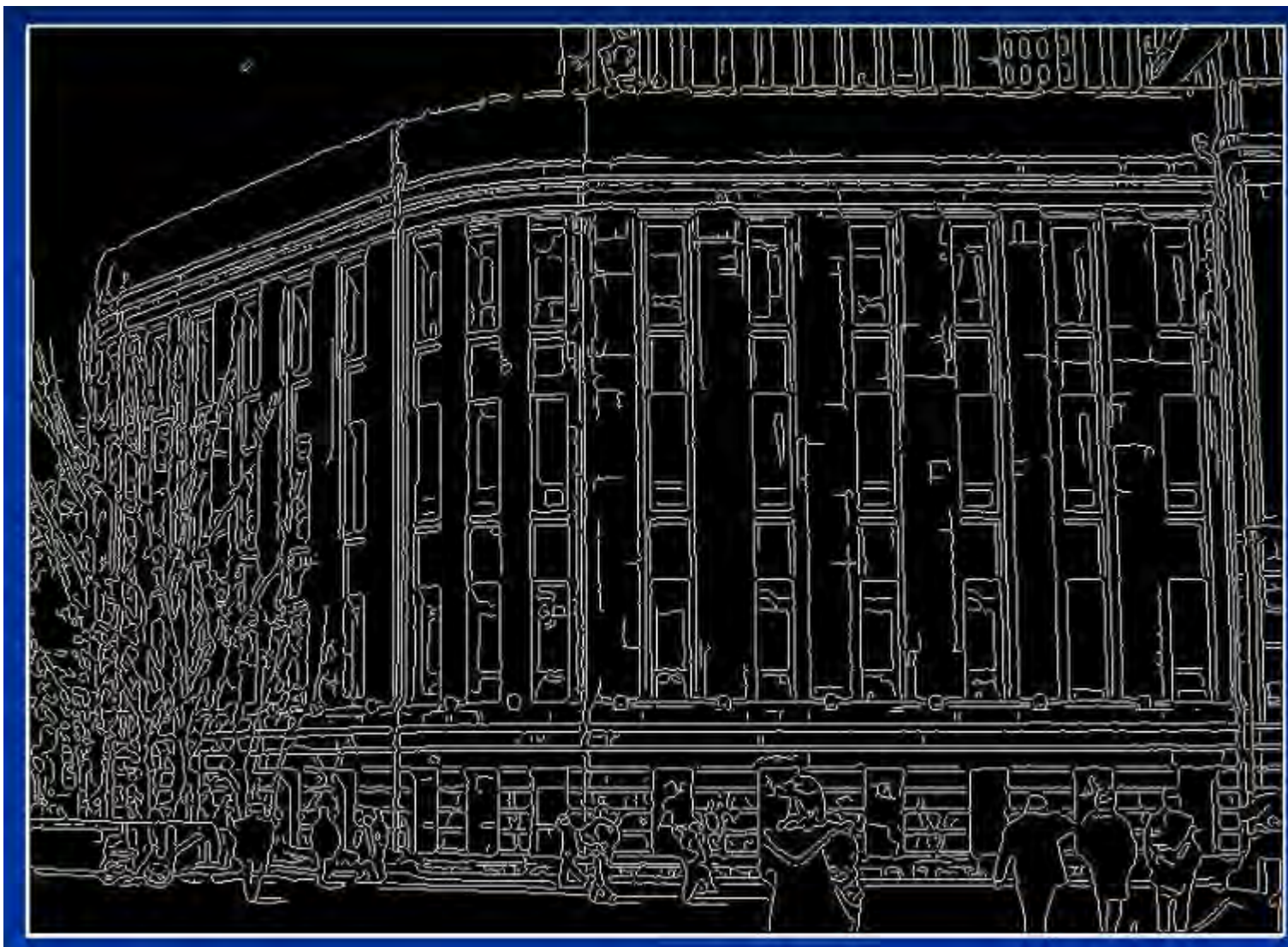
Обрезание по порогу



Удачный пример применения Sanny



Canny



Влияние σ (размер ядра размытия)



original

Canny with $\sigma = 1$

Canny with $\sigma = 2$

Выбор σ зависит от задачи

- большое σ - поиск крупных границ
- маленькое σ - выделение мелких деталей

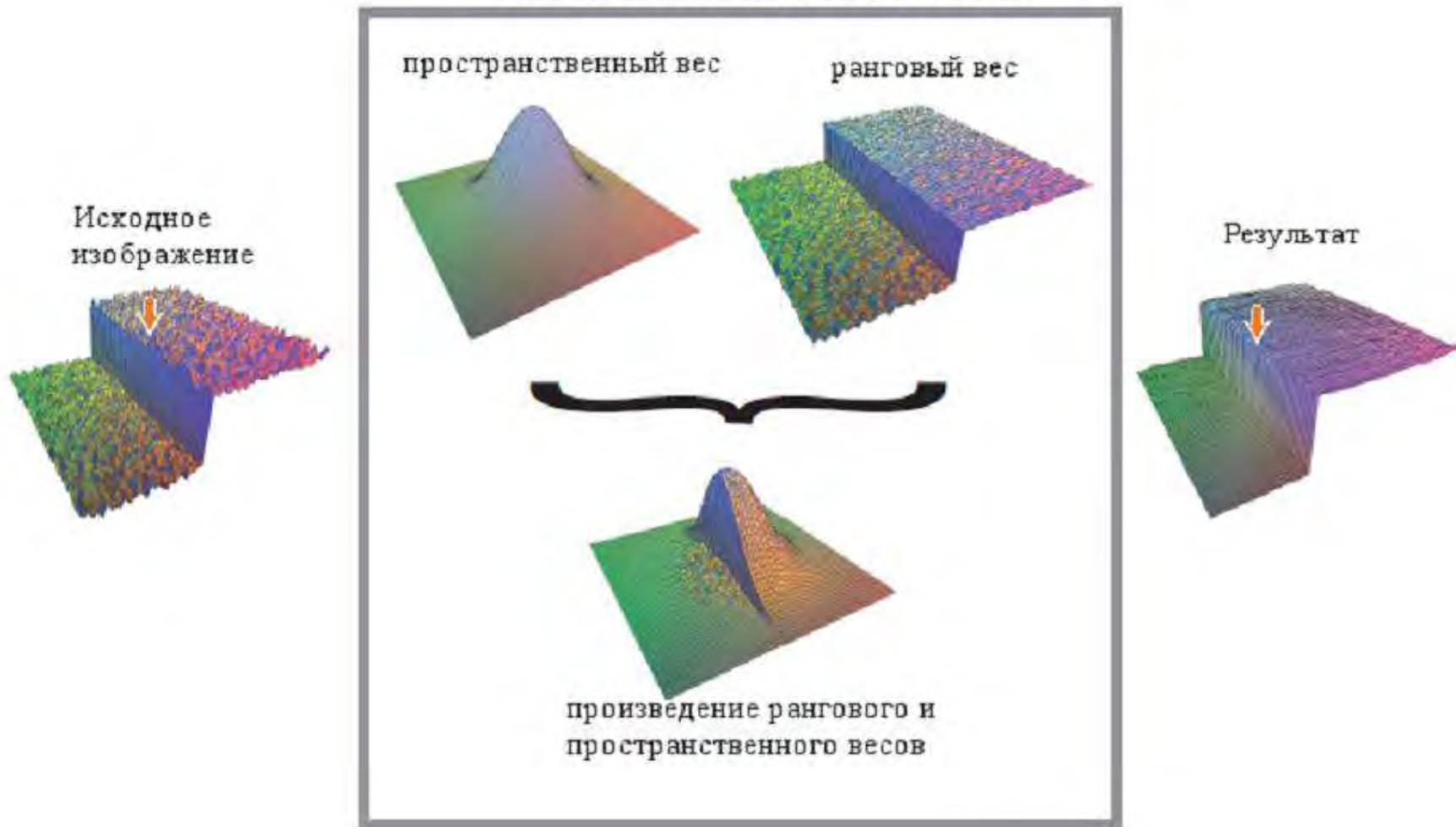
Source: S. Seitz

Сглаживание билатеральным фильтром с сохранением границ



Визуализация билатерального фильтра с сохранением границ

Вычисление веса выбранного пикселя

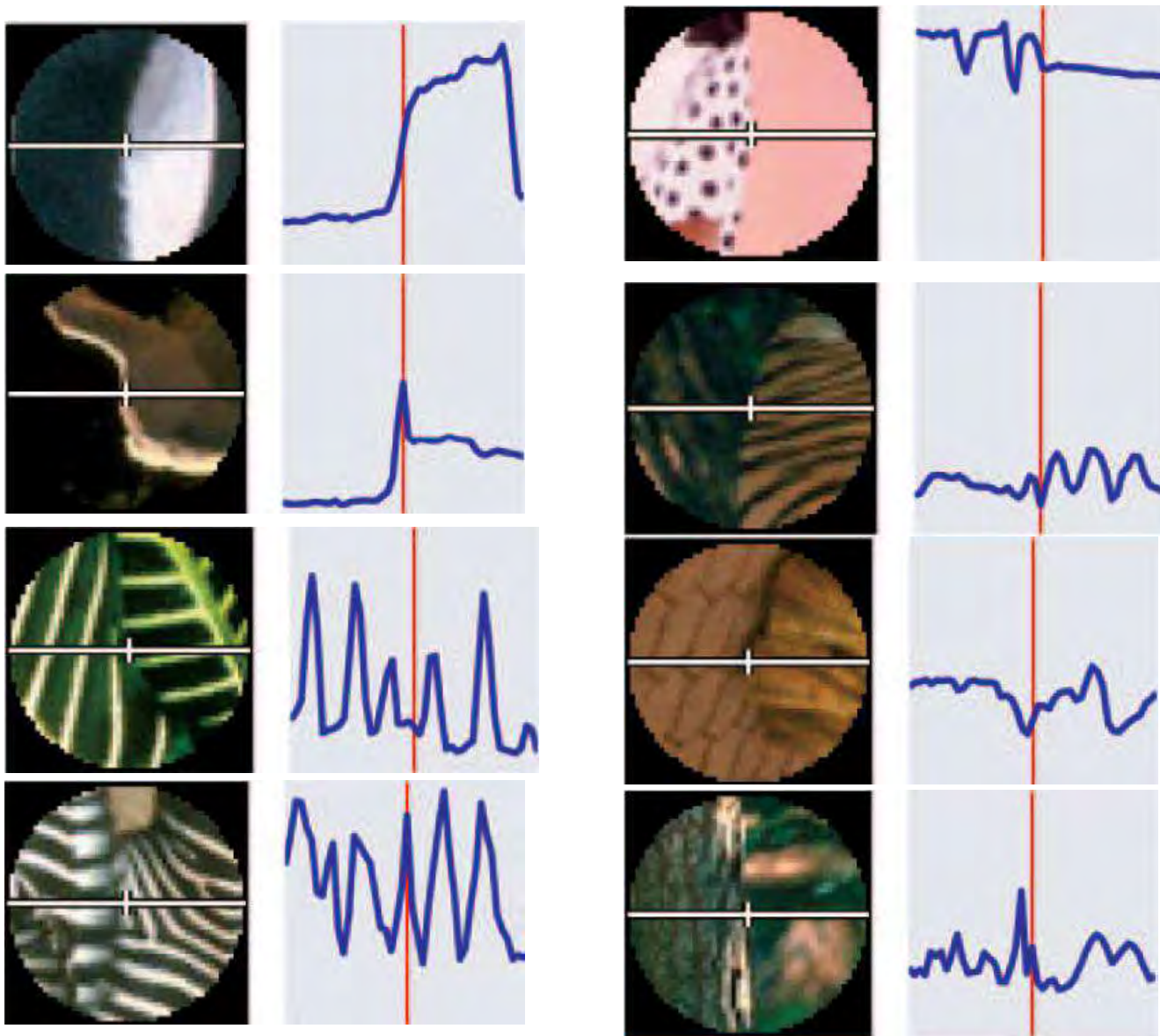


Билатеральный фильтр с сохранением границ

$$h(a_0) = k^{-1} \sum_{i=0}^{n-1} f(a_i) \times g(a_i) \times r(a_i)$$

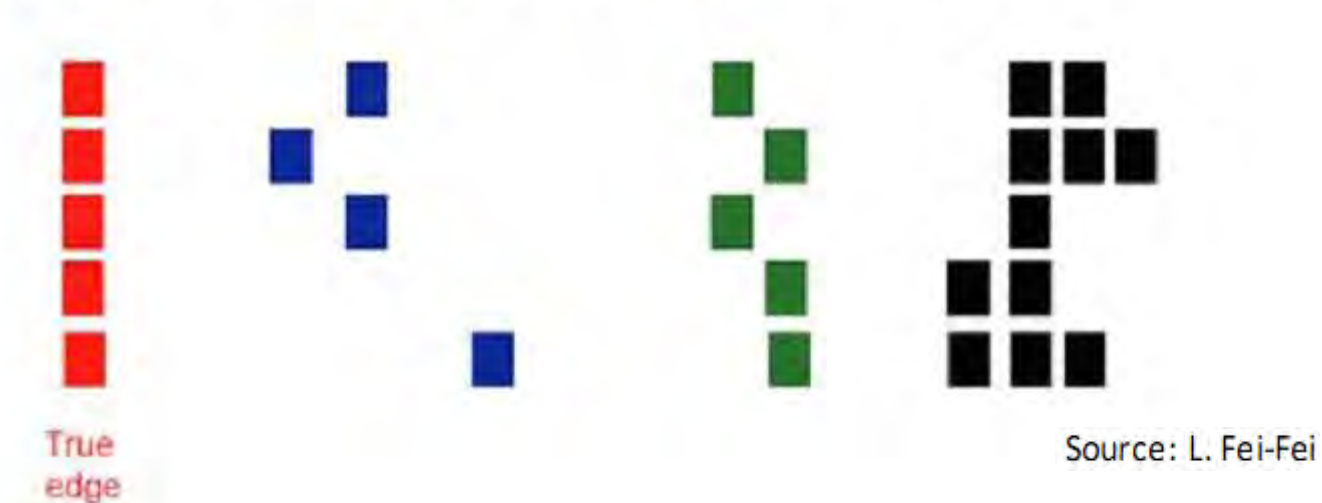
$$r(a_i) = e^{-\frac{(f(a_i) - f(a_0))^2}{2\sigma^2}} \quad g(x, y) = e^{-\frac{x^2 + y^2}{2t^2}}$$

Ограничения детектора



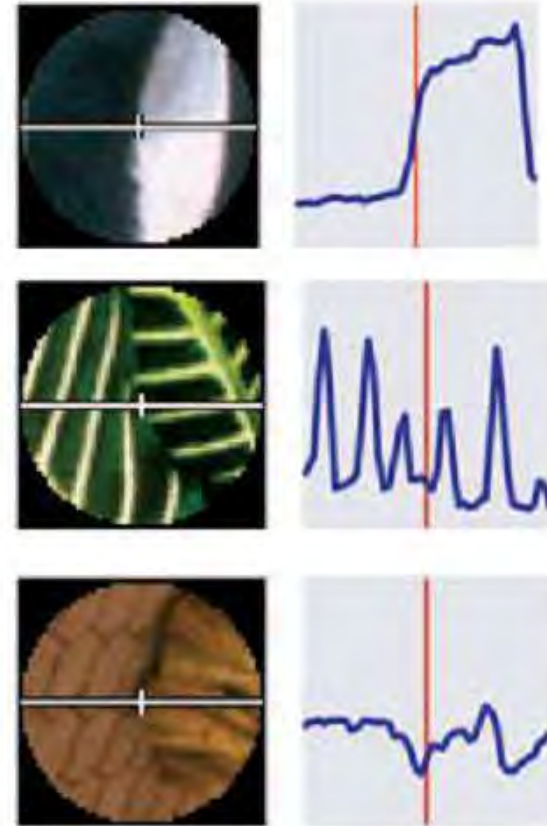
Критерии качества детектора

- **Good detection:** оптимальный детектор должен минимизировать ошибки 1 и 2го родов (ложные края и пропущенные края)
- **Точная локализация:** найденный край должен быть как можно ближе к истинному краю
- **Единственный отклик:** детектор должен выдавать одну точку для одной точки истинного края, т.е. локальных максимум вокруг края должно быть как можно меньше



Probability boundary (Pb) -детектор

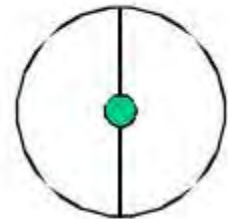
- Недостаточно находить градиент изображения
- Идея – обучить классификатор граница / не граница
- Будем использовать разные признаки:
 - Яркость
 - Цвет
 - Текстура
- «Probability boundary» (Pb) детектор



D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI 2004.

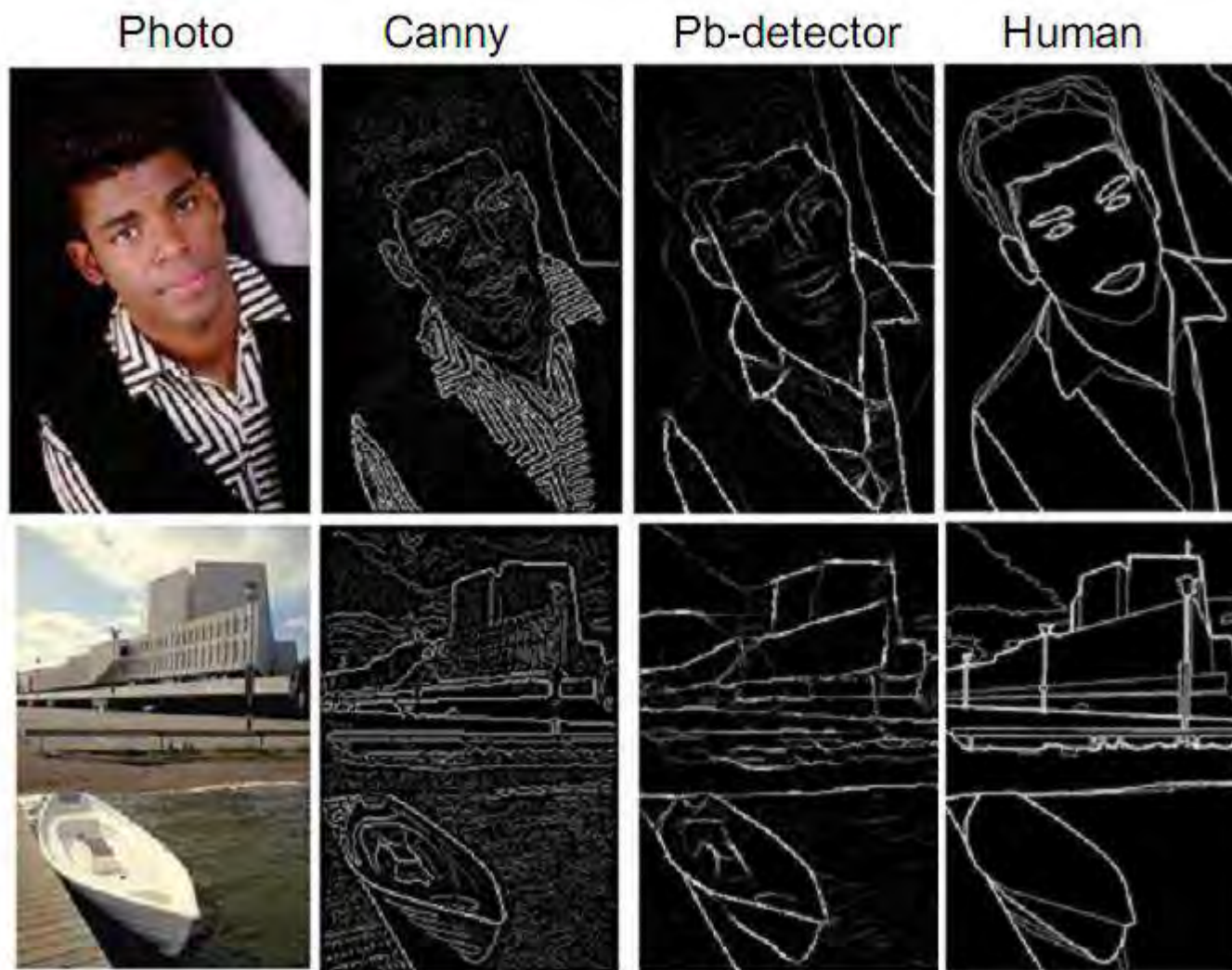
Схема Рb-детектора

- Берём круг с центром в исследуемом пикселе
- Выбираем ориентацию края
- Считаем градиент между признаками в левой и правой половине круга
- Обучаем классификатор края
- Требуется размеченная коллекция изображений



D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI 2004.

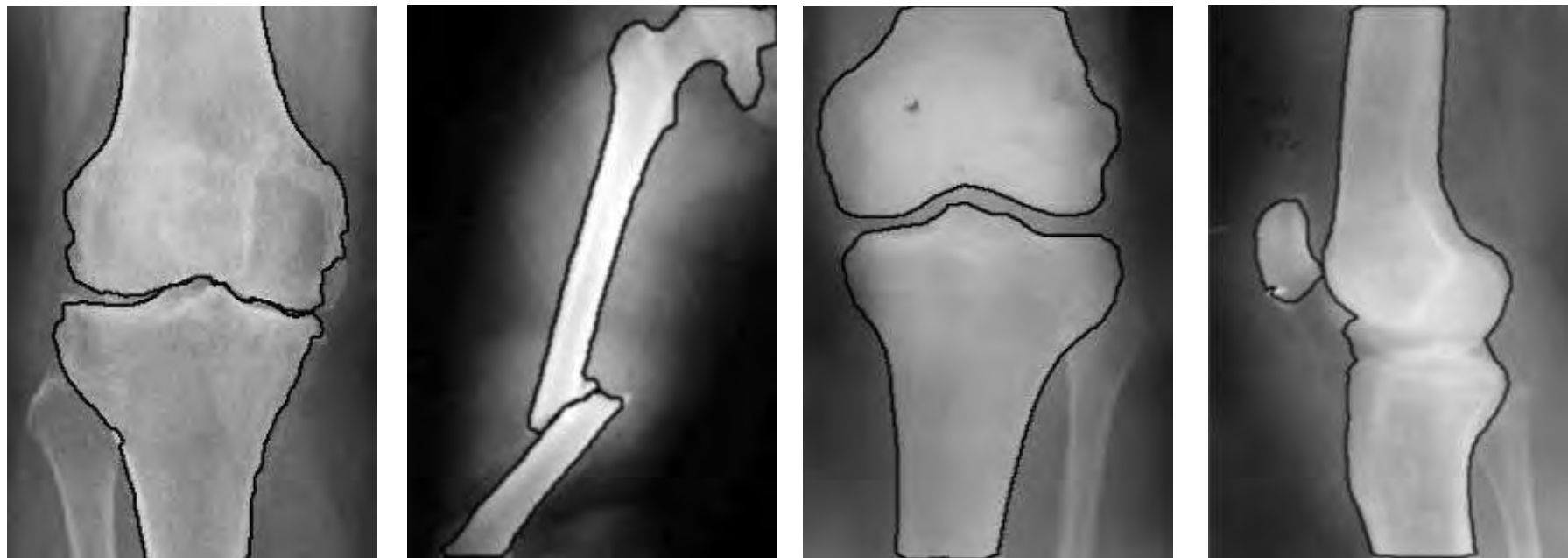
Результат Рb-детектора



План

- Прослеживание контуров
- **Связывание контуров**
- Анализ контуров
- Метод активных контуров

Детектирование контуров на медицинских изображениях



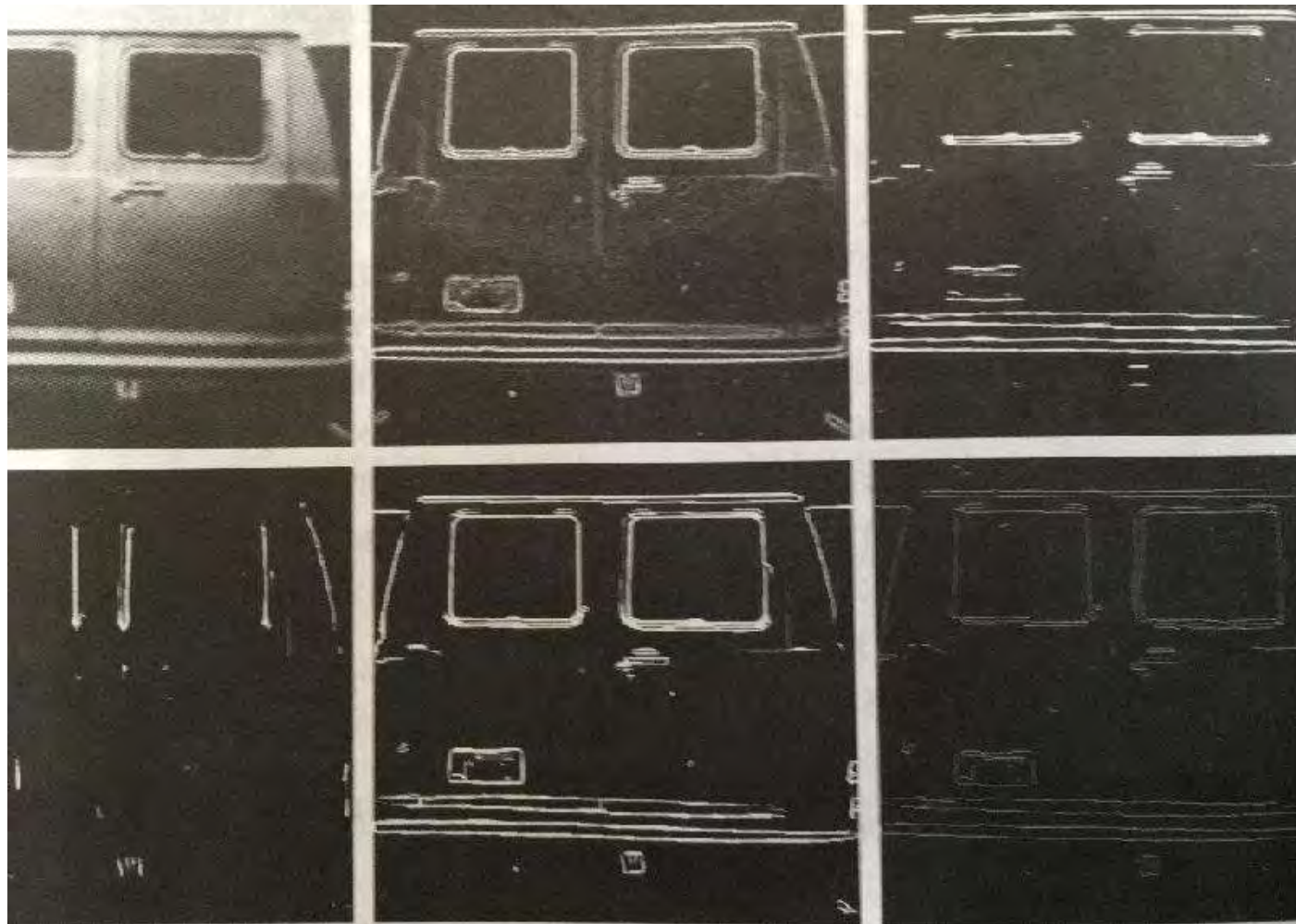
Работа Михайличенко Алексея (2016 магистратура ФИИТ)

Выделение ногтей и наложение текстуры



Работа Пака Владислава (2016 магистратура ФИИТ)

Связывание контуров с помощью локальной обработки



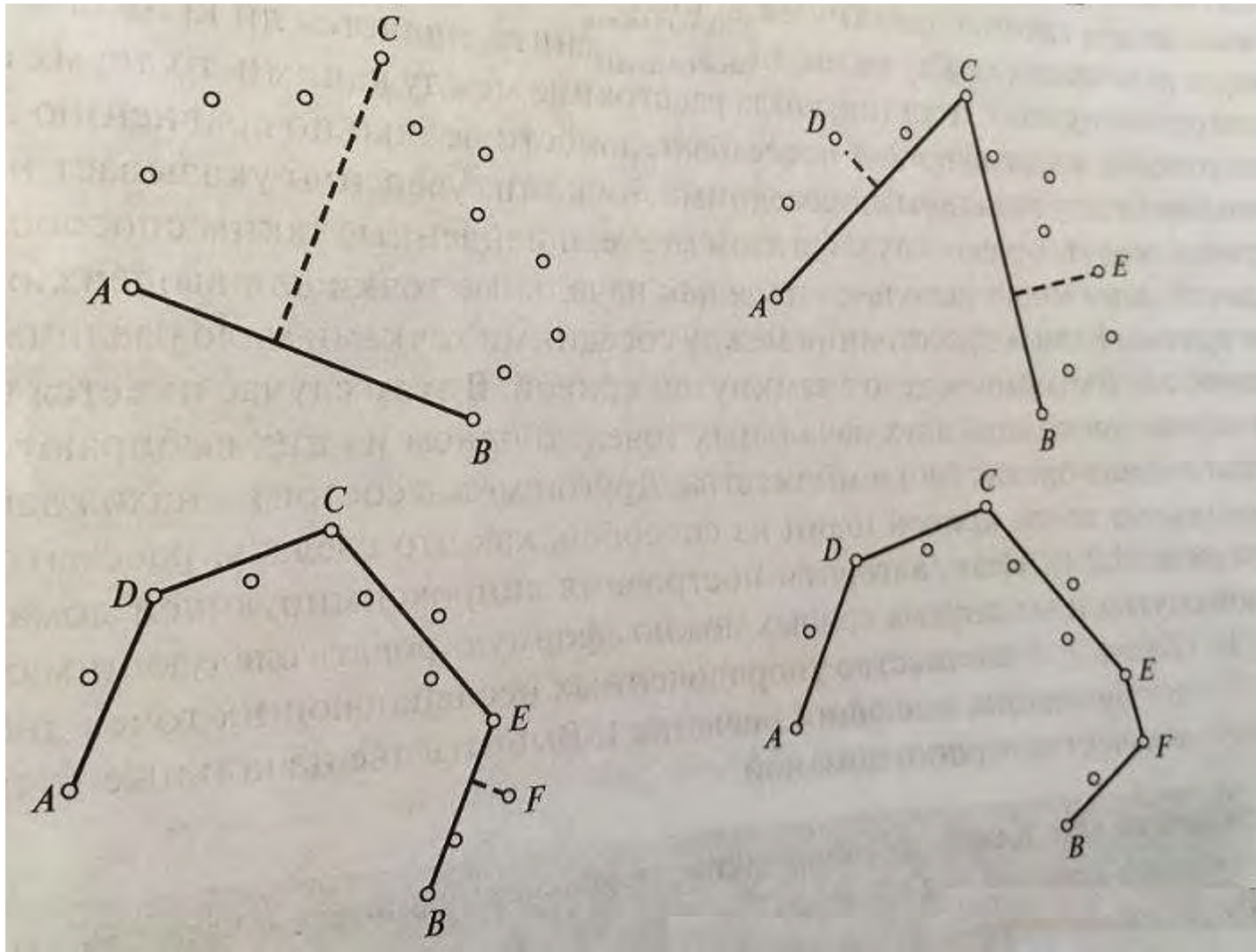
534x546

$E=30\%$ от макс
градиента

Заполнение
пропусков в 25
пикселей

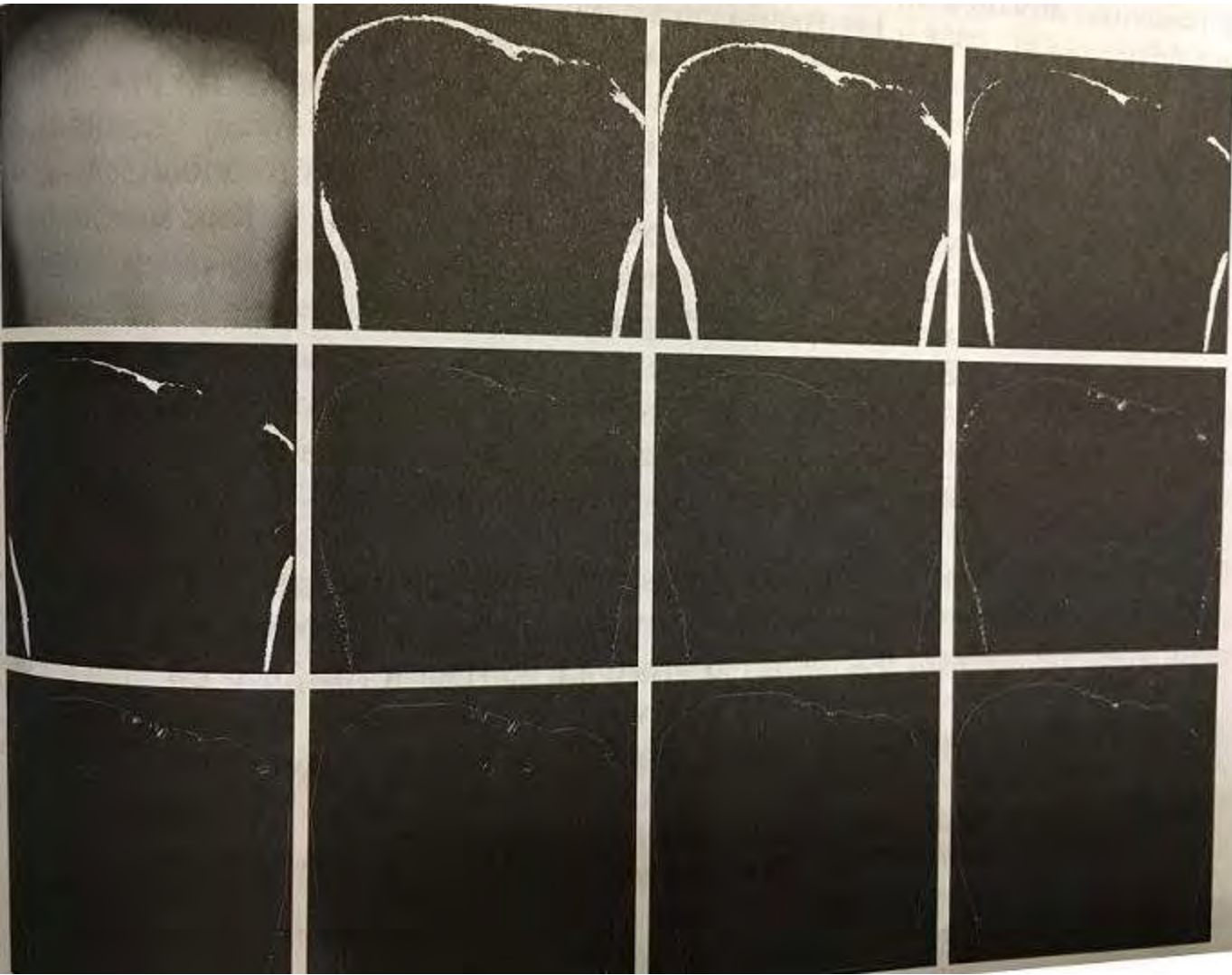
а) исходное изображение; б) модуль градиента; в) горизонтально связанные контурные пиксели; г) вертикально связанные контурные пиксели; д) результат ИЛИ к в) и г); е) результат морфологического утончения

Обработка в интересующей области



Итерационный алгоритм аппроксимации незамкнутой ломаной

Пример. Граница зуба



- а) исходное;
- б) градиентное;
- в) результат мажоритарной фильтрации;
- г) морфологическое утончение;
- д) морфологическая очистка;
- е) построение остова;
- ж) усечение отростков;
- з-к) аппроксимация границы;
- л-м) результаты сглаживания

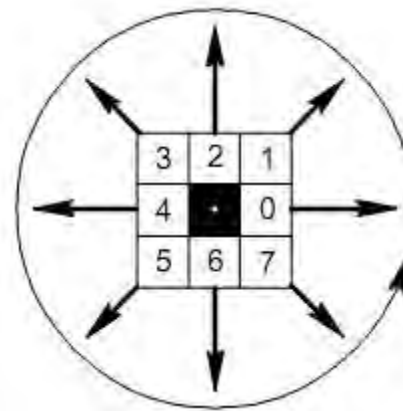
Локальная обработка полутонового изображения

Критерии сходства

$$|\nabla f(x, y) - \nabla f(x_0, y_0)| \leq E$$

$$|\alpha(x, y) - \alpha(x_0, y_0)| \leq A$$

обычно используют окрестность 3×3



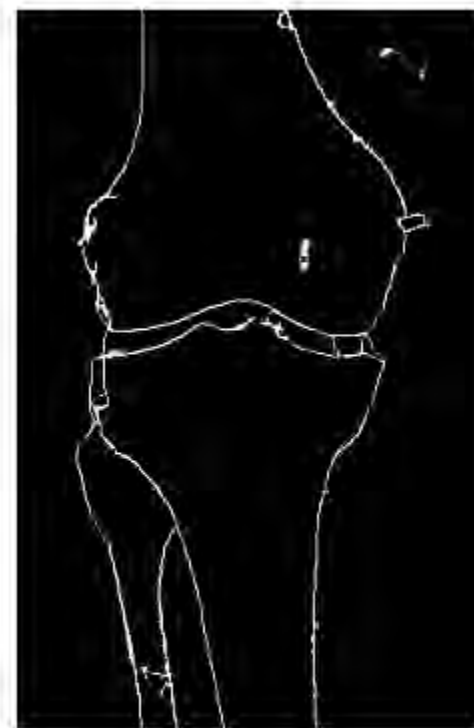
Пример устранения разрывов



a)



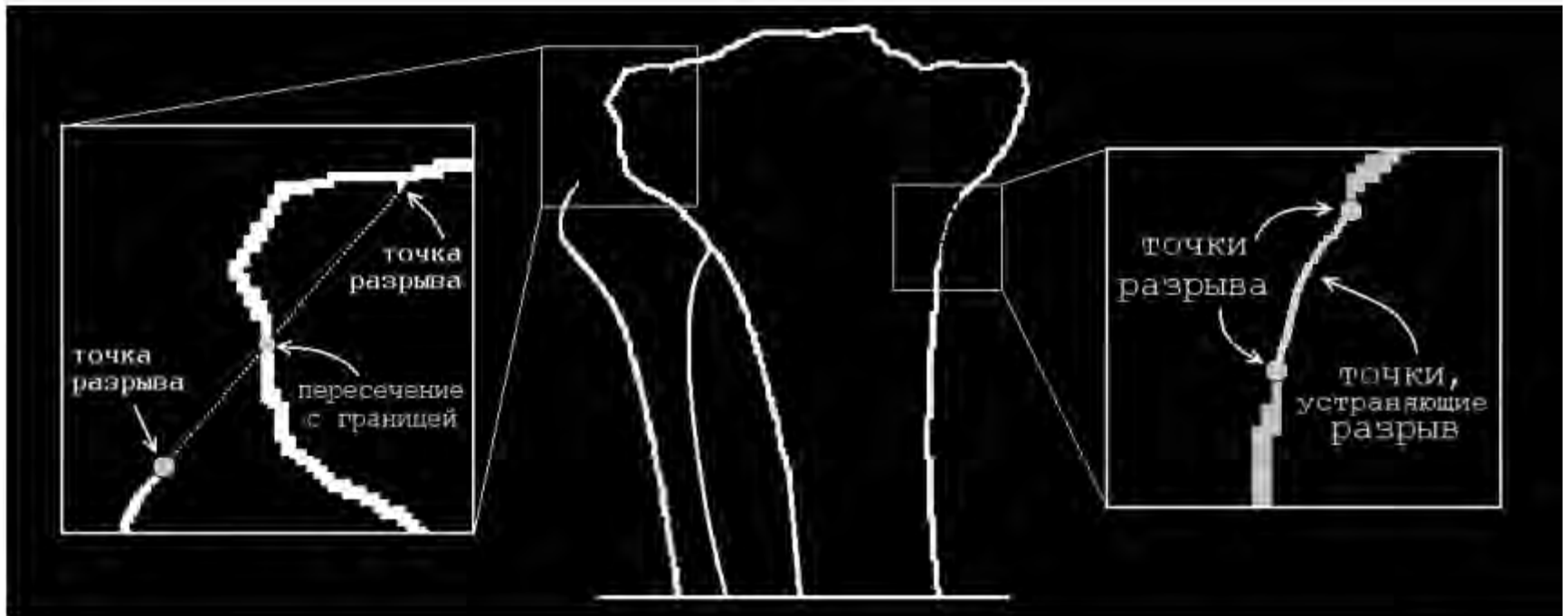
b)



c)

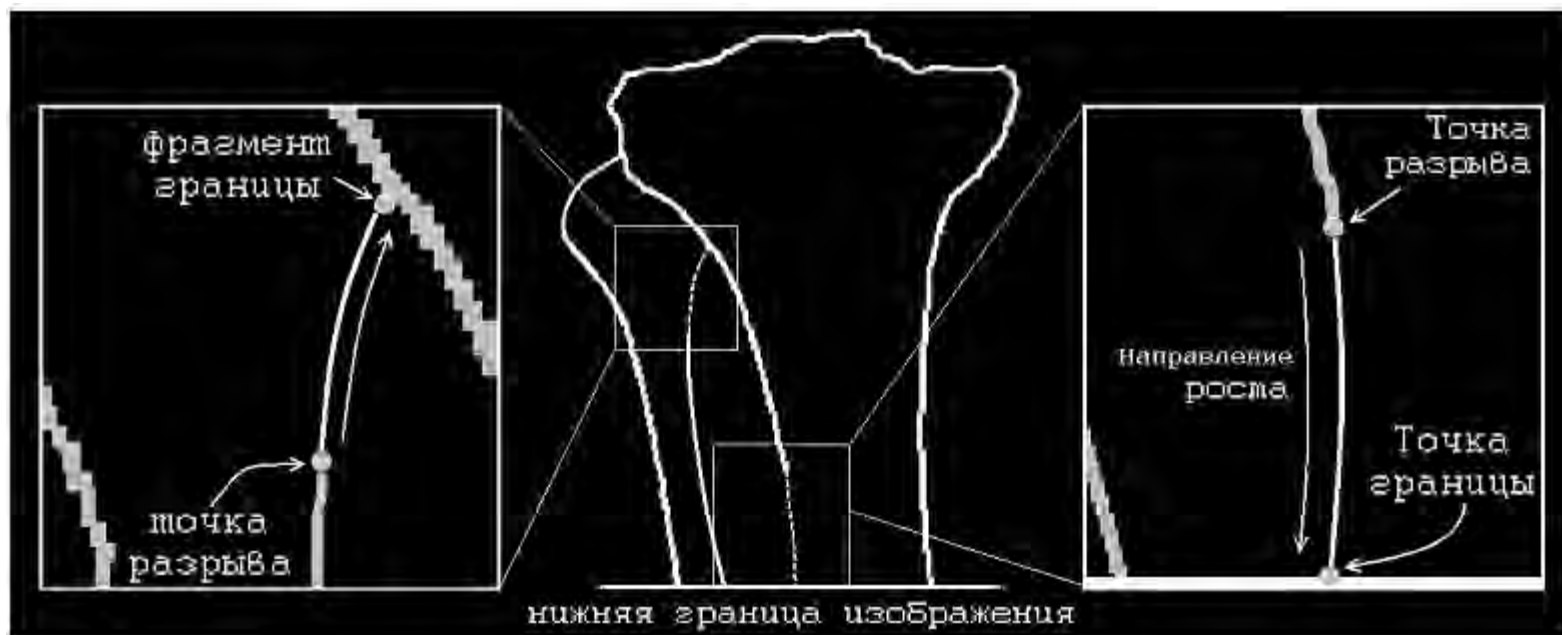
Процесс связывания границ: несколько итераций

Пример. Прослеживание контуров



Нарушение условия «не пересекает уже существующие границы на изображении» (слева) и пример устранения разрыва (справа)

Устранение разрывов



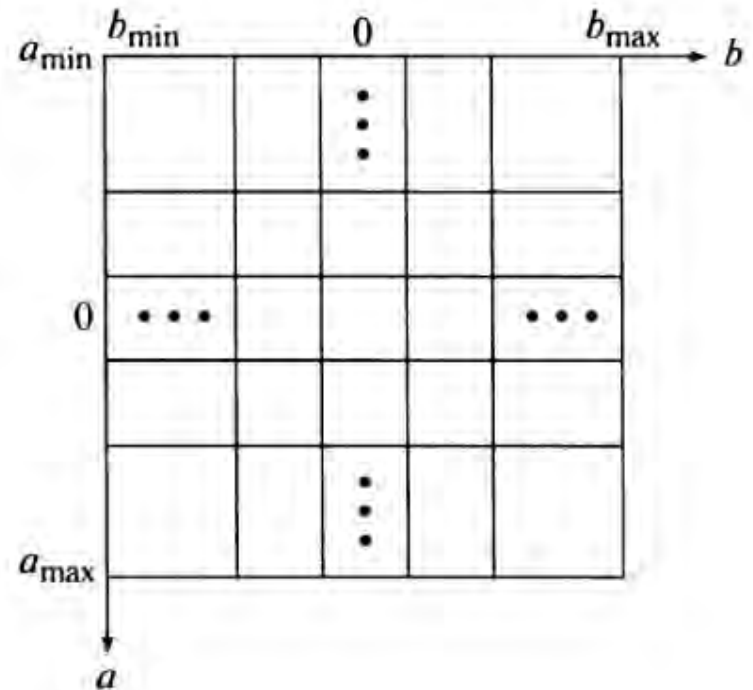
Процесс связывания для одиночной (непарной) точки разрыва

Основная идея преобразования Хафа

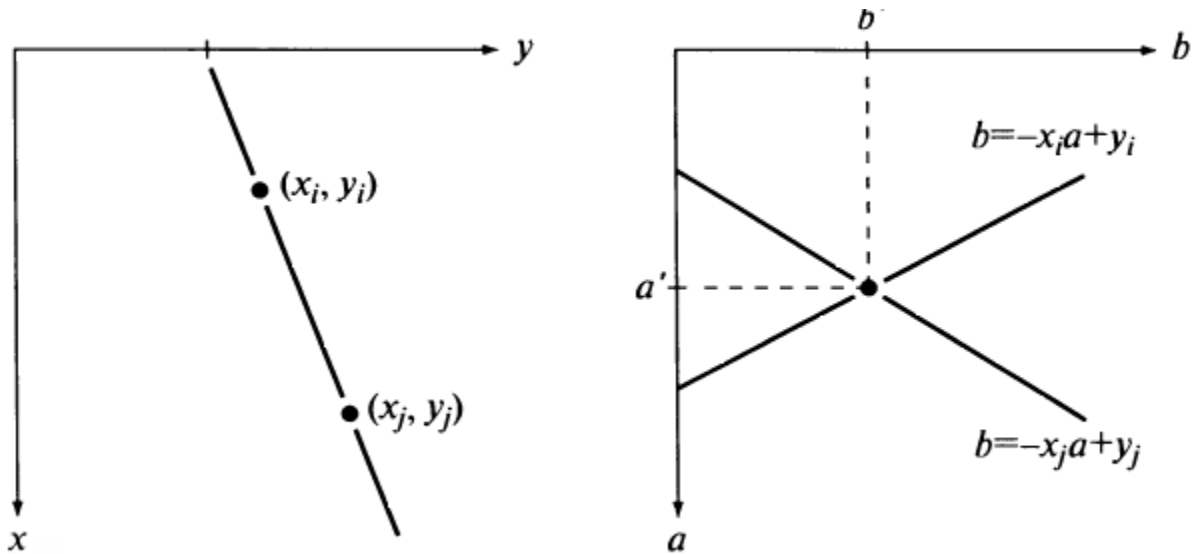
Семейство кривых на плоскости, заданное параметрическим уравнением:
 $F(a_1, a_2, \dots, a_n, x, y) = 0$;

a_1, a_2, \dots, a_n – параметры семейства кривых образуют фазовое пространство

- Аккумуляторный массив
- Список координат всех пикселей, которые внесли вклад в значение соответствующих элементов аккумуляторного массива



Глобальный анализ с помощью преобразования Хафа



(а) Плоскость $xу$. (б) Пространство параметров.

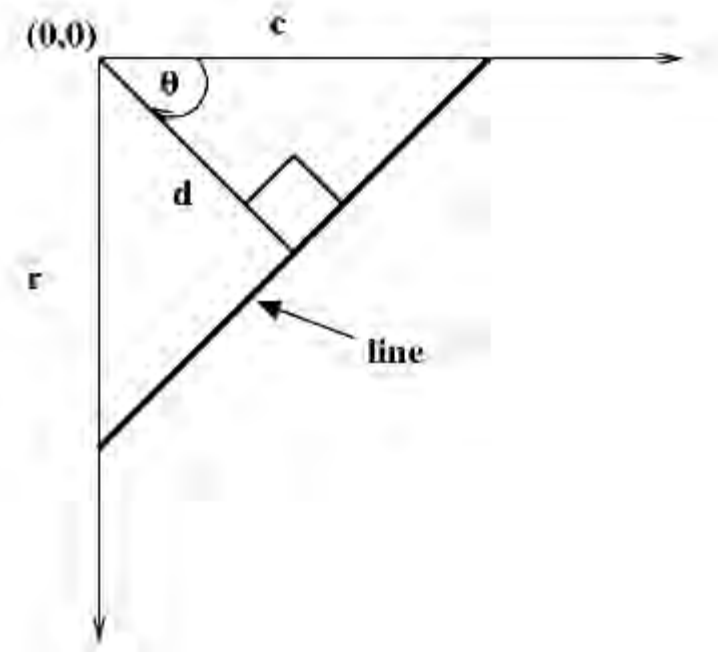
- Хаф [Hough, 1962]

Обнаружение отрезков прямых

$$d = c \cdot \cos\theta + r \cdot \sin\theta$$

$$d = x \cdot \cos\theta + y \cdot \sin\theta$$

$$F(d, \theta, x, y) = x \cos\theta + y \sin\theta - d$$

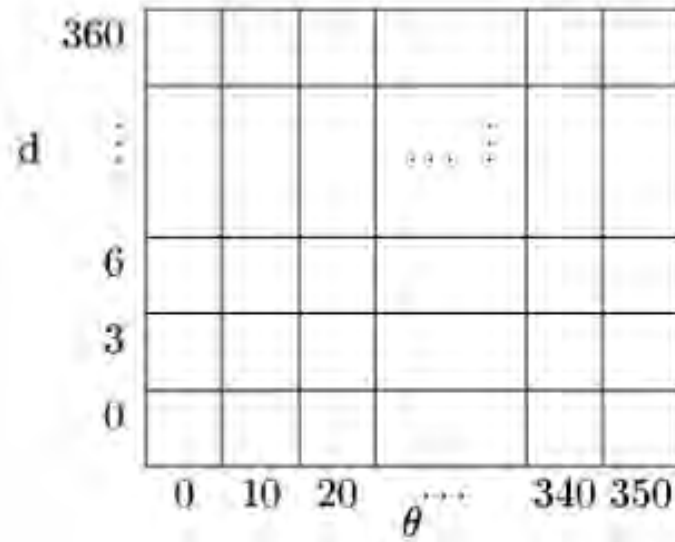


Разбиение плоскости параметров

$$x \cos\theta + y \sin\theta = d, \text{ где } \theta_i \leq \theta \leq \theta_{i+1}, d_i \leq d \leq d_{i+1}$$

O'Gorman & Clowes

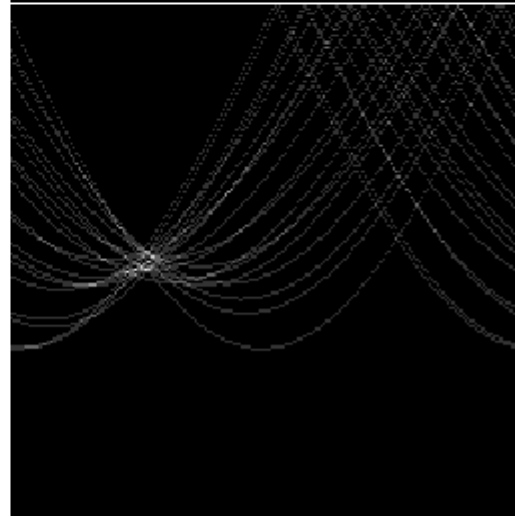
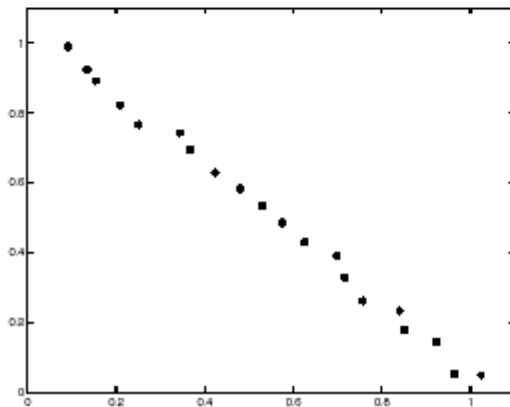
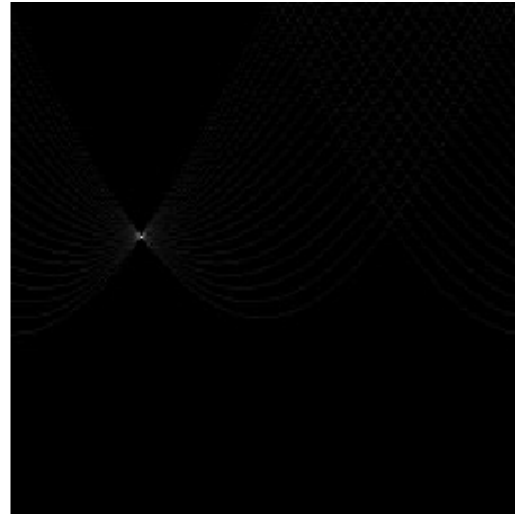
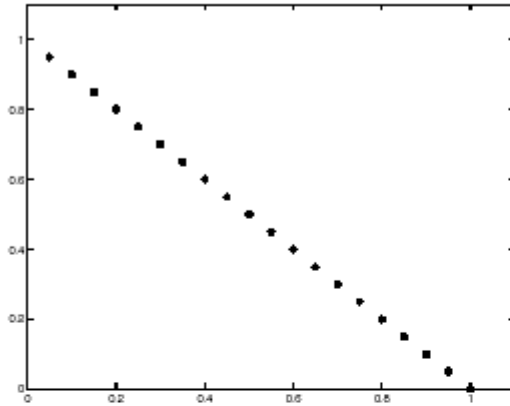
1976



Алгоритм поиска прямой на изображении

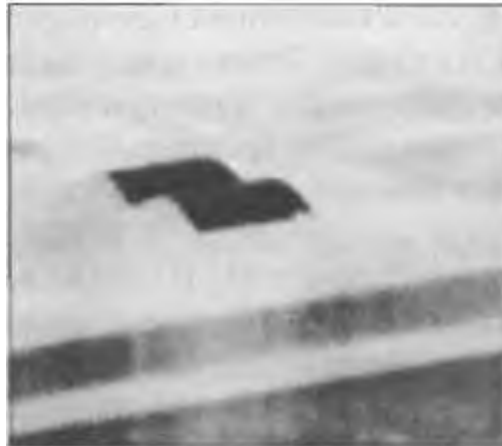
- обнулить счетчики всех ячеек;
- для каждой точки интереса;
-
- для каждой прямой, проходящей через данную точку;
-
- увеличить соответствующий счетчик;
- выбрать ячейку с максимальным значением счетчика;
- параметры прямой, проходящей через максимальное число точек принять равным координатам центра выбранной ячейки в фазовом пространстве.

Пример преобразования Хафа

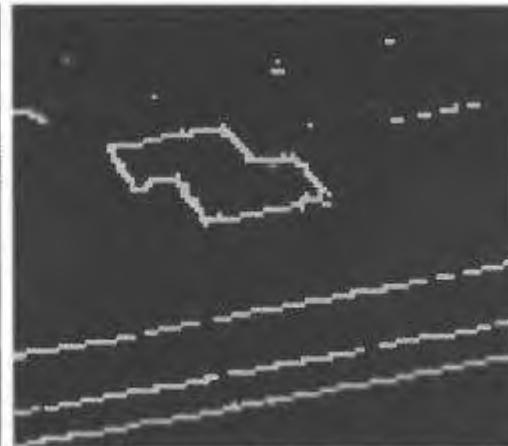


Пример

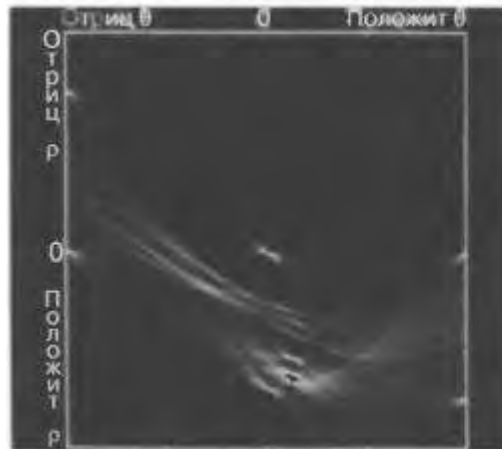
Исходное



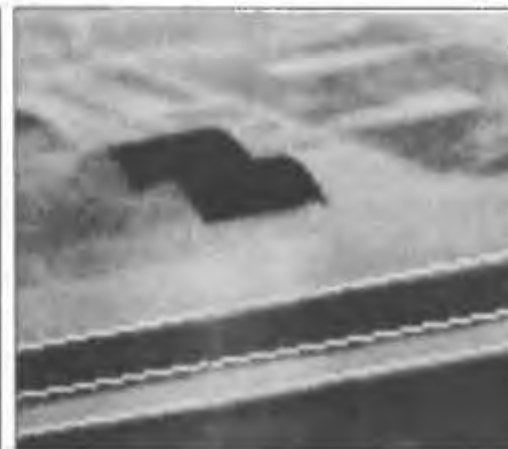
Градиентное



Хафа



Связанные
пиксели



Определитель прямых Барнса (1986)

Разработан для обнаружения прямых линий на сложных изображениях естественных сцен

1. Вычисление величины и направление градиента для каждого пиксела изображения
2. Точкам с достаточной большой величиной градиента присваиваются две метки, соответствующие двум различным дискретным значениям направления градиента. Результаты маркировки сохраняются в виде двух маркированных изображений

0 to 44, 45 to 90, 91 to 134, etc.

-22 to 22, 23 to 67, 68 to 112, etc.

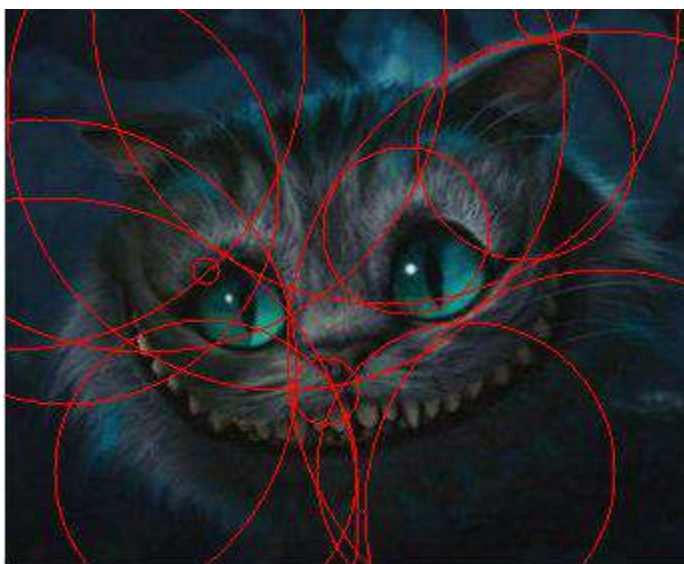
Определитель прямых Барнса (1986)

3. Поиск связных компонент на каждом маркированном изображении и вычисление длины отрезков каждой компоненты:
 1. Каждый пиксел принадлежит двум компонентам, по одной на каждом маркированном изображении
 2. Каждый пиксел используется для голосования за свою длиннейшую компоненту
 3. Счетчик каждой компоненты равен количеству проголосовавших за нее пикселов
 4. Выполняется выбор компонентов (отрезков прямых) с наибольшими значениями счетчиков

Чеширский кот не так уж и прямолинеен



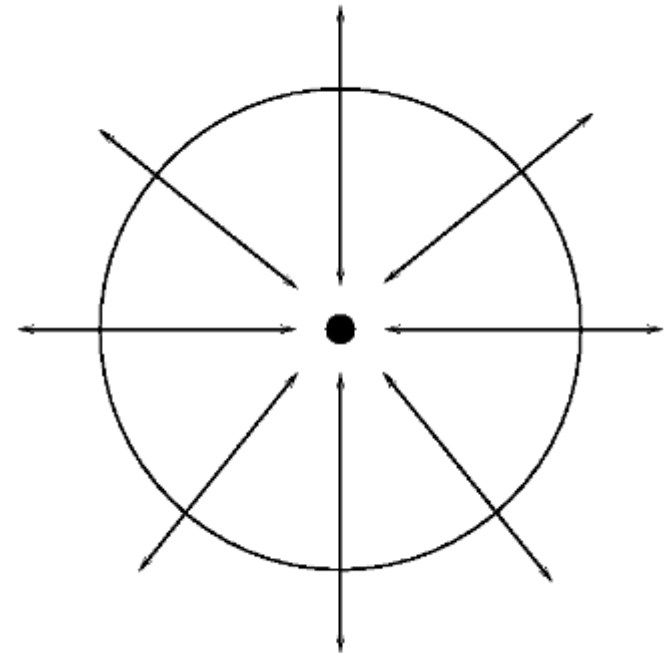
Чеширский кот любит окружности



Выделение окружностей на изображении

$$(x - a)^2 + (y - b)^2 = R^2$$

$$F(a, b, R, x, y) = (x - a)^2 + (y - b)^2 - R^2$$

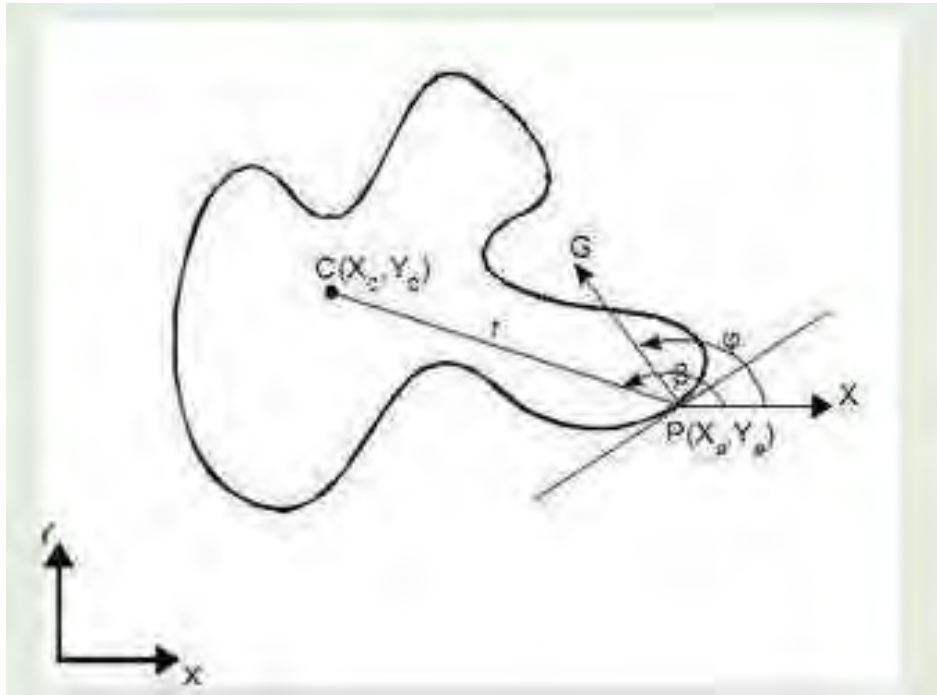


[Robyn Owens, "Computer Vision IT412, Lecture 6",
http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/OWENS/LECT6/node3.html](http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/OWENS/LECT6/node3.html), 1997

Обобщенное преобразование Хафа (Ballard 1981)

1. Инициализация аккумуляторного массива $A[a]$ нулевыми значениями
2. Для каждого краевого пиксела определяется такой вектор a , что $f(x,a)=0$ и выполняется увеличение значения соответствующего элемента массива $A[a]=A[a]+1$
3. Локальные максимумы в аккумуляторном массиве A соответствуют вероятным кривым f на изображении

Вычисление параметров граничных точек для занесения их в R-таблицу

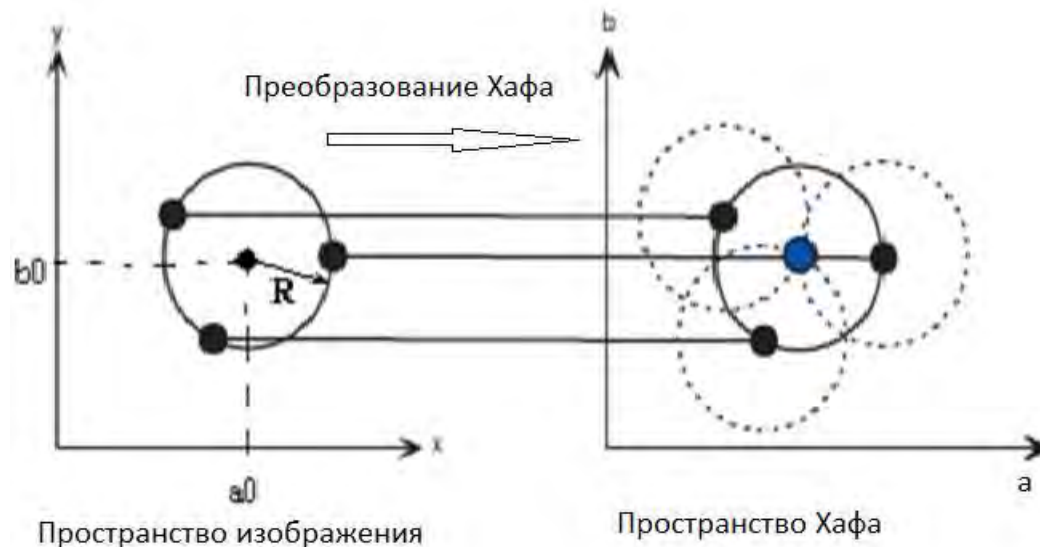


Выбирается произвольная точка отсчета для объекта и для каждого граничного пикселя объекта вычисляется направление градиента φ так же как длина r и направление β вектора, соединяющего этот граничный пиксель и точку отсчета

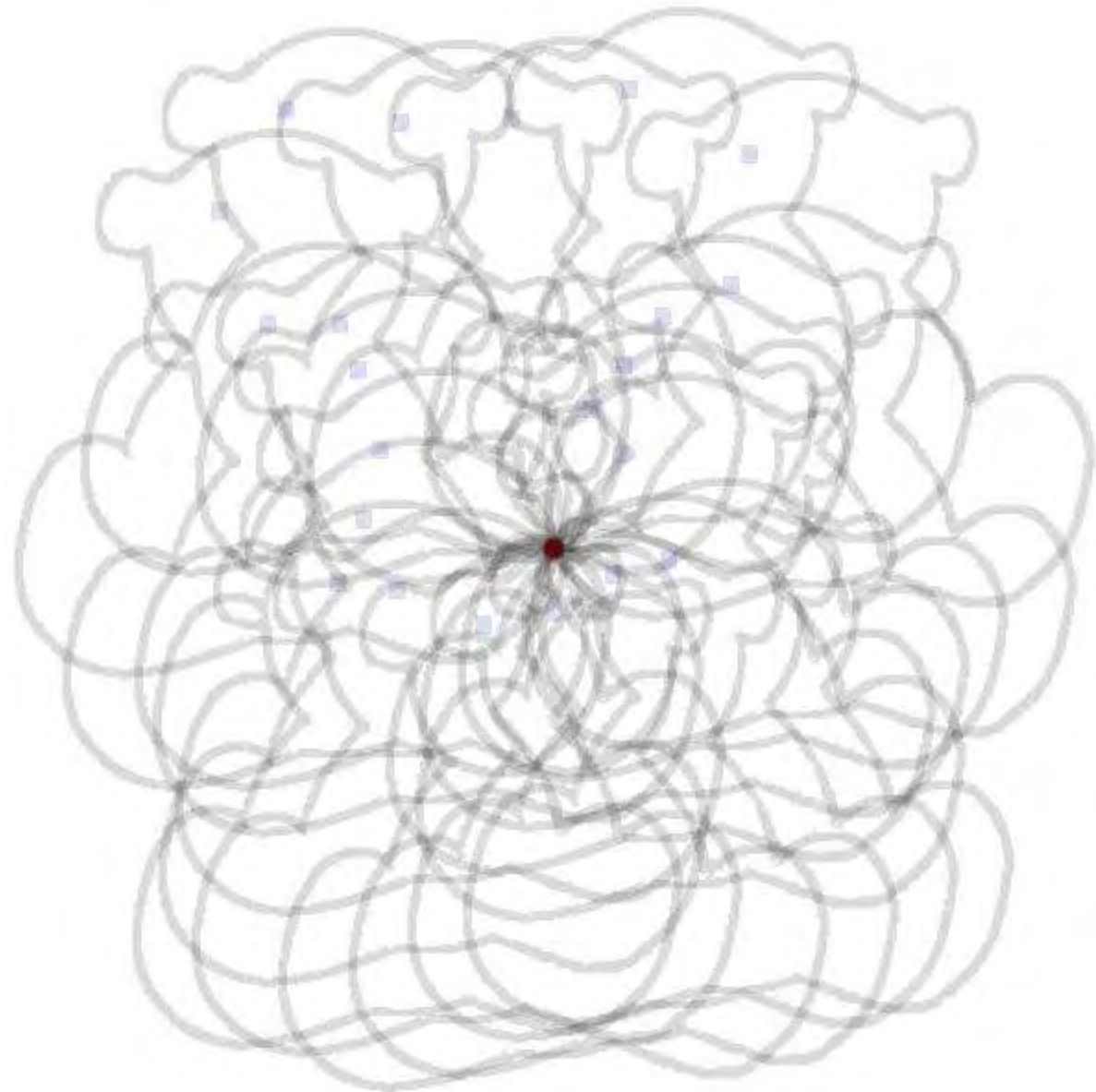
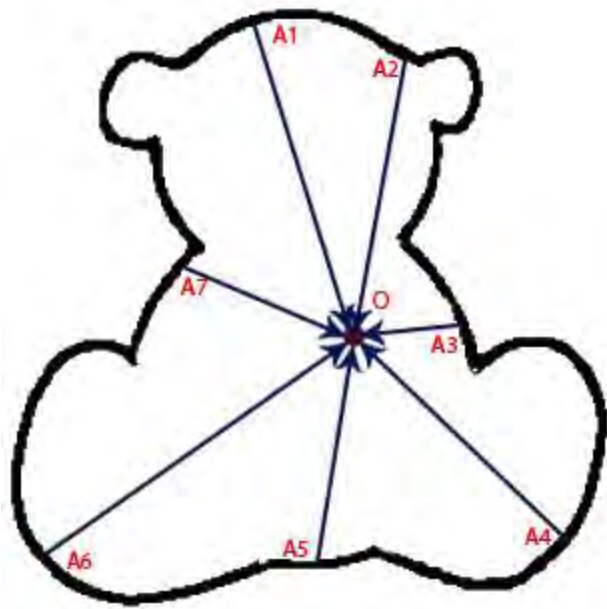
$$\begin{cases} x_p = x_c - r \cdot \cos(\beta) \\ y_p = y_c - r \cdot \sin(\beta) \end{cases}$$

Модифицированное преобразование Хафа для произвольных объектов методом окружностей

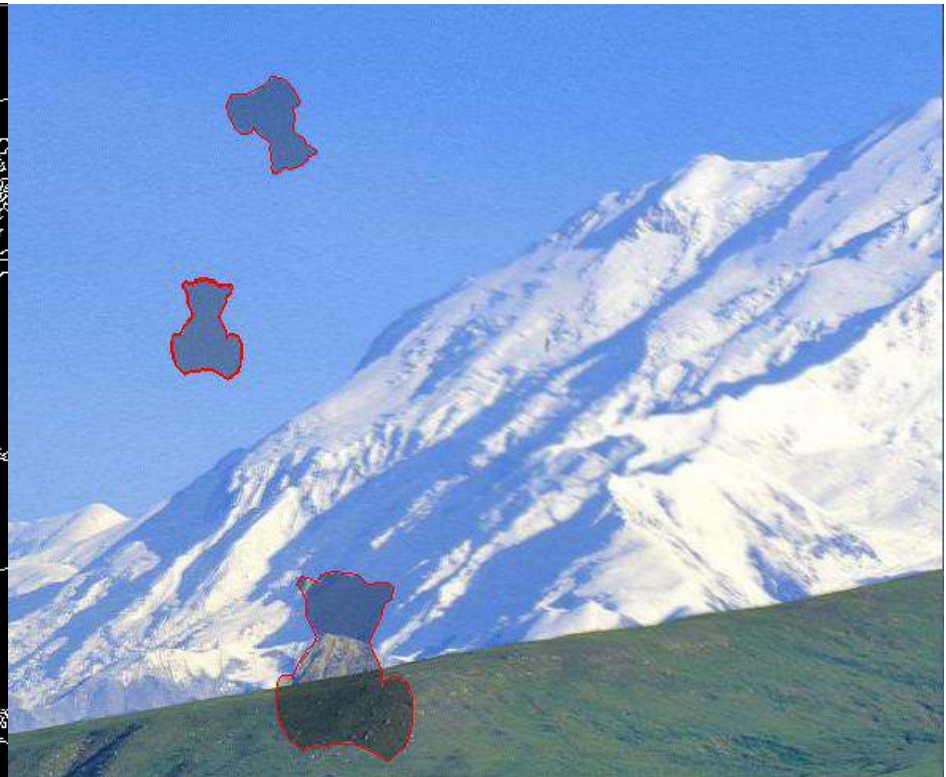
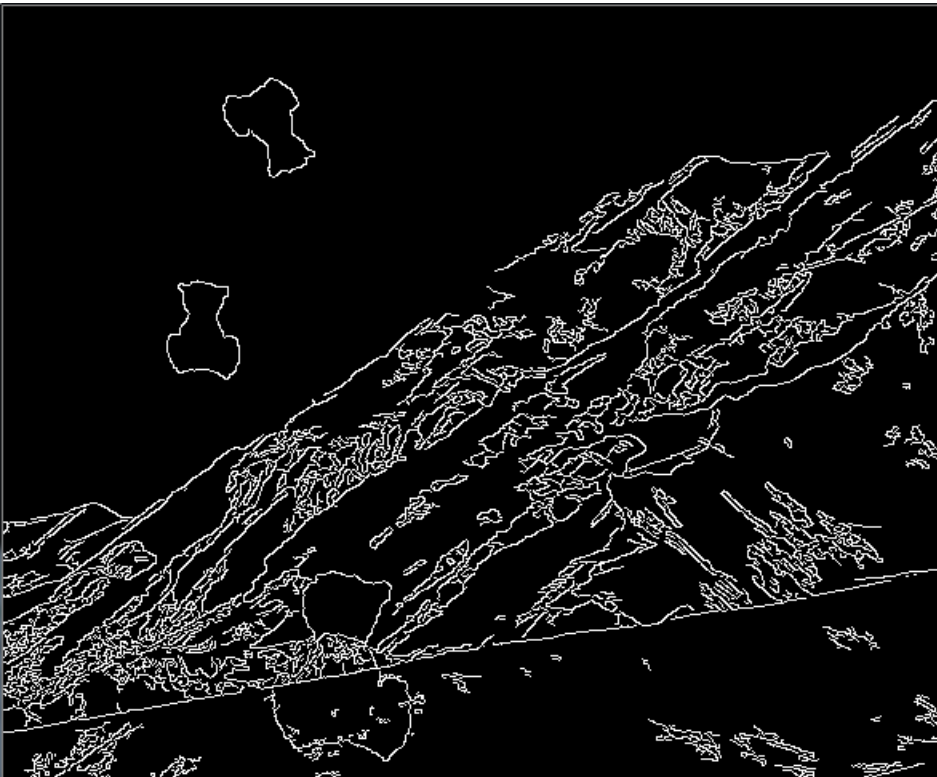
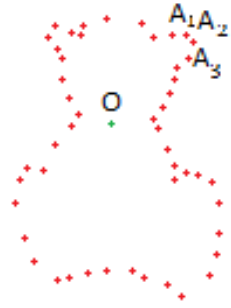
- Если известен радиус окружности, которую требуется детектировать, то число свободных параметров сокращается до двух – координаты центра окружности, то есть пространство Хафа является двухмерным.



Метод окружностей



Метод окружностей



Сравнение с точки зрения быстродействия

Размер изображения	Классическое преобразование Хафа	Метод "окружностей"
200·154	113853 ms	6304 ms
300·231	212492 ms	11026 ms
400·308	315489 ms	16593 ms

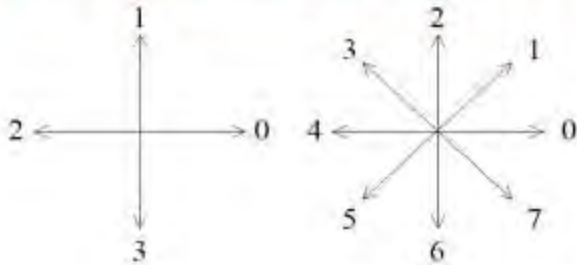
¶

План

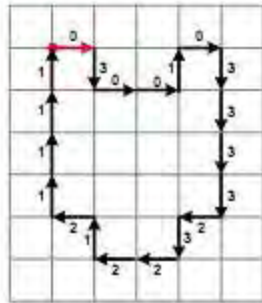
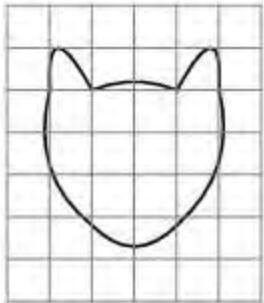
- Прослеживание контуров
- Связывание контуров
- **Анализ контуров**
- Метод активных контуров

Цепной код Фримана

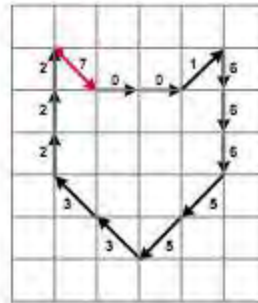
Нумерация направлений для 4-связного и 8-связного цепных кодов:



Пример:



А



Б

А: 03001033332322121111

Б: 70016665533222

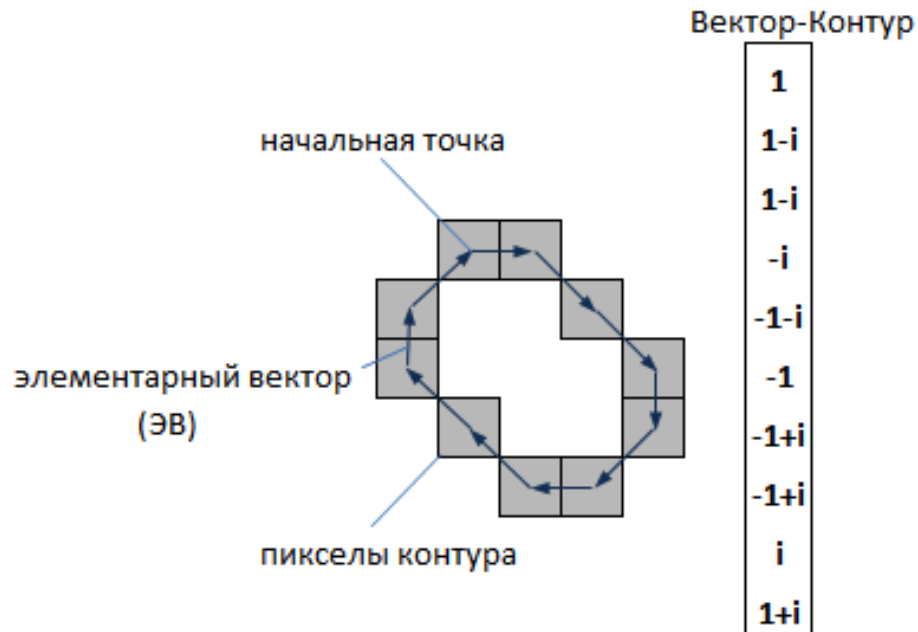


Инвариантность к выбору
начальной точки: минимальный код
70016665533222 -> 00166655332227



Инвариантность к повороту:
разности цифр кода
00166655332227 -> 01500706070051

Кодирование контура



$$\Gamma = (\gamma_0, \gamma_1, \dots, \gamma_{k-1})$$

Свойства контуров

- Сумма ЭВ замкнутого контура равна нулю.
- Контур-вектор не зависит от параллельного переноса исходного изображения.
- Поворот изображения на определенный угол равносильен повороту каждого ЭВ контура на тот же угол.
- Изменение начальной точки ведет к циклическому сдвигу ВК.
- Изменение масштаба исходного изображения можно рассматривать как умножение каждого ЭВ контура на масштабный коэффициент.

Скалярное произведение контуров

$$\eta = (\Gamma, N) = \sum_{n=0}^{k-1} (\gamma_n, \nu_n) \quad (1)$$

скалярное произведение элементарных контуров

$$(a + ib, c + id) = (a + ib)(c - id) = ac + bd + i(bc - ad) \quad (2)$$

скалярное произведение векторов

$$((a, b), (c, d)) = ac + bd \quad (3)$$

Нормированное скалярное произведение (НСП)

$$\eta = \frac{(\Gamma, N)}{|\Gamma||N|} \quad (4)$$

$$|\Gamma| = \left(\sum_{n=0}^{k-1} |\gamma_n|^2 \right)^{\frac{1}{2}} \quad (5)$$

максимально возможное значение модуля НСП = 1

в случае если









$$\Gamma = \mu N \quad (6)$$

При умножении комплексных чисел, их модули(длины) перемножаются, а аргументы(углы) – складываются.

Значит контур μN это тот же контур N , но повернутый и промасштабированный.

Масштаб и поворот определяется комплексным числом μ .

Свойства НСП

		НСП	$\text{Re}(\text{НСП})=\cos(a)$	$ \text{НСП} $
	x 	1	1	1
	x 	i	0	1
	x 	-1	-1	1
	x 	-i	0	1

Проблема

$$\Gamma = \mu N \quad (6)$$

равенство (6) достигается только если начальные точки контуров – совпадают

Взаимокорреляционная функция (ВКФ) двух контуров

$$\tau(m) = (\Gamma, N^{(m)}), \quad m = 0, \dots, k - 1 \quad (7)$$

Где $N^{(m)}$ — контур, полученный из N путем циклического сдвига его ЭВ на m элементов.

ВКФ является периодической, с периодом k

если $N = (n_1, n_2, n_3, n_4)$,
то $N(1) = (n_2, n_3, n_4, n_1)$,
 $N(2) = (n_3, n_4, n_1, n_2)$
...

Мера похожести двух контуров

$$\tau_{max} = \max \left(\frac{\tau(m)}{|\Gamma||N|} \right), \quad m = 0, \dots, k - 1 \quad (8)$$

Инвариантна

- переносу,
- масштабированию,
- вращению
- и сдвигу начальной точки

Автокорреляционная функция (АКФ)

$$v(m) = (\Gamma, \Gamma^{(m)}), \quad m = 0, \dots, k - 1 \quad (9)$$

Некоторые свойства АКФ

1. АКФ не зависит от выбора начальной точки контура
2. Модуль АКФ симметричен относительно центрального отсчета $k/2$

$$\text{АКФ}(0)=(n_1,n_1)+(n_2,n_2)+(n_3,n_3)+(n_4,n_4)$$

$$\text{АКФ}(1)=(n_1,n_2)+(n_2,n_3)+(n_3,n_4)+(n_4,n_1)$$

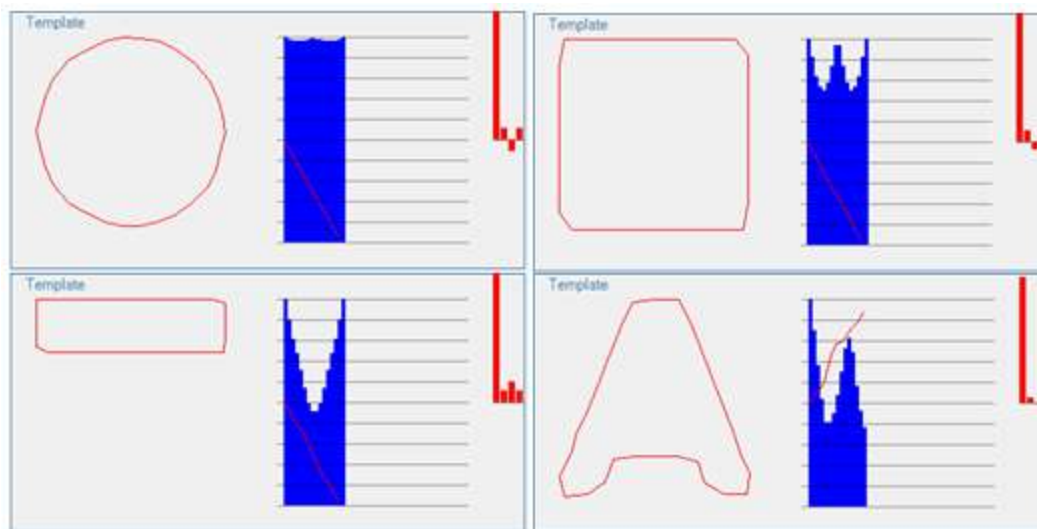
$$\text{АКФ}(2)=(n_1,n_3)+(n_2,n_4)+(n_3,n_1)+(n_4,n_2)$$

$$\text{АКФ}(3)=(n_1,n_4)+(n_2,n_1)+(n_3,n_2)+(n_4,n_3)$$

$$\text{АКФ}(4)=(n_1,n_1)+(n_2,n_2)+(n_3,n_3)+(n_4,n_4)$$

Некоторые свойства АКФ

3. Если контур имеет какую-либо симметрию относительно поворота, то аналогичную симметрию имеет его АКФ.
4. АКФ контура в определенном смысле можно считать характеристикой формы контура.



Некоторые свойства АКФ

5. Нормированная АКФ не зависит от масштаба, положения, вращения и выбора начальной точки контура.

Эквализация контуров

- Если исходный контур больше необходимого, то перебираем все его ЭВ, и считаем элементы N как сумму всех ЭВ, следующим образом:

```
Complex[] newPoint = new Complex[newCount];
```

```
for (int i = 0; i < oldCount; i++)  
newPoint[i * newCount / oldCount] += oldPoint[i];
```


Эквализация контуров

- Если же исходный контур меньше k , то производим интерполяцию:

```
Complex[] newPoint = new Complex[newCount];  
for (int i = 0; i < newCount; i++) {  
    double index = 1.0 * i * oldCount / newCount;  
    int j = (int)index;  
    double k = index - j;  
    newPoint[i] = oldPoint[j] * (1 - k) + oldPoint[j + 1] * k;  
}
```

Практическое применение контурного анализа

Общий алгоритм распознавания

- Предварительная обработка изображения — сглаживание, фильтрация помех, повышение контраста.
- Бинаризация изображения и выделение контуров объектов.
- Начальная фильтрация контуров по периметру, площади, коэффициенту формы, фрактальности и так далее.
- Приведение контуров к единой длине, сглаживание.
- Перебор всех найденных контуров, поиск шаблона, максимально похожего на данный контур.

Сложность

ВКФ требует вычислений порядка $O(k^2)$, где k – длина контура

Полное время поиска шаблона для *отдельно взятого контура* можно оценить как $O(k^2t)$, где t – число шаблонных контуров

Идентификации *всех контуров изображения* $O(n^2k^2t)$, где n – линейный размер изображения, k – длина контура, t – число шаблонов

Дескриптор контура

Для быстрого поиска шаблонов необходимо ввести некий дескриптор, характеризующий форму контура

В качестве дескриптора, описывающего форму контура можно выбрать АКФ

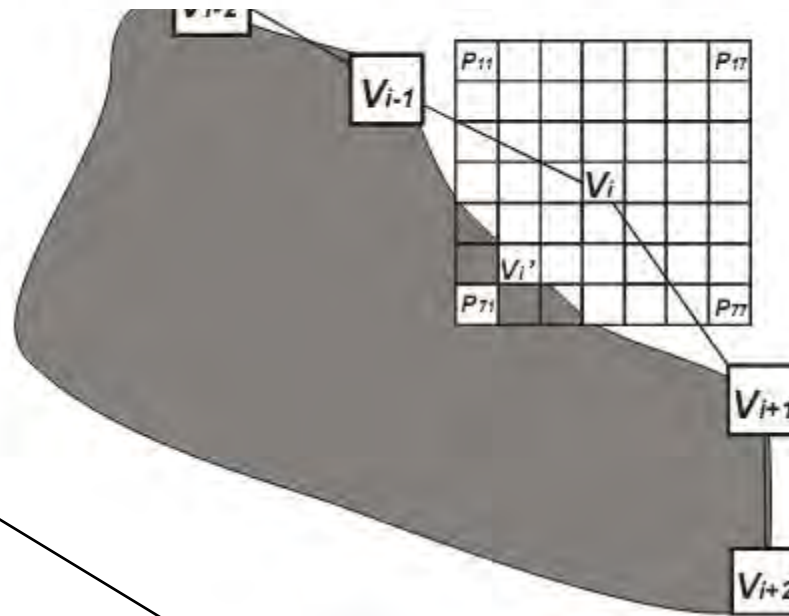
Если база шаблонов хранит их АКФ, то поиск шаблона для контура, путем сравнения АКФ, составит **$O(kt)$**

План

- Прослеживание контуров
- Связывание контуров
- Анализ контуров
- **Метод активных контуров**

Активный контур

$$V = \{v_1, \dots, v_n\}, \text{ где } v_i = (x_i, y_i), i = \{1, \dots, n\}$$



Энергетическая составляющая, зависящая от формы контура

Энергетическая составляющая, зависящая от свойств изображения (градиент)

$$E_i = a \cdot E_{\text{int}}(v_i) + b \cdot E_{\text{ext}}(v_i)$$

Внутренняя энергия

Внутренняя энергия = сглаживающая + распирающая

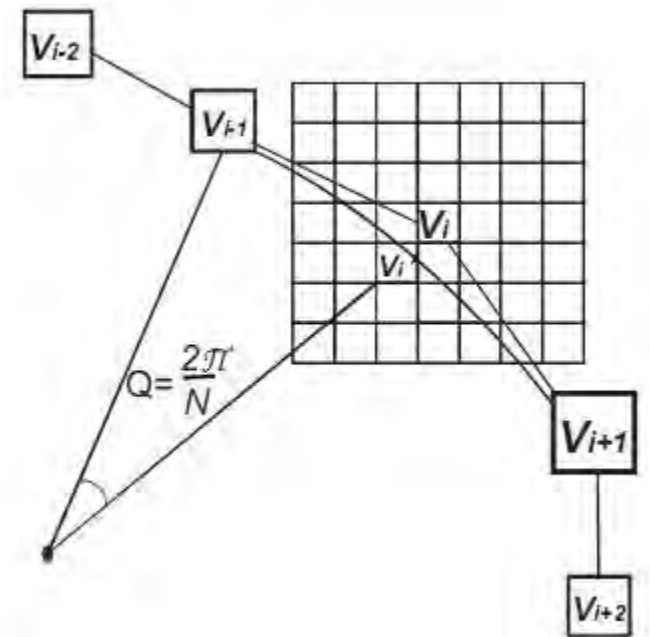
$$a \cdot E_{\text{int}}(v_i) = c \cdot E_{\text{con}}(v_i) + d \cdot E_{\text{bal}}(v_i)$$

Сглаживающая составляющая

$$e_{jk}(v_i) = \frac{1}{l(V)} \left\| p_{jk}(v_i) - \gamma(v_{i-1} + v_{i+1}) \right\|^2$$

$$\gamma = \frac{1}{2 \cos\left(\frac{2\pi}{n}\right)}$$

$$l(V) = \frac{1}{n} \sum_{i=1}^n \left\| v_{i+1} - v_i \right\|^2$$

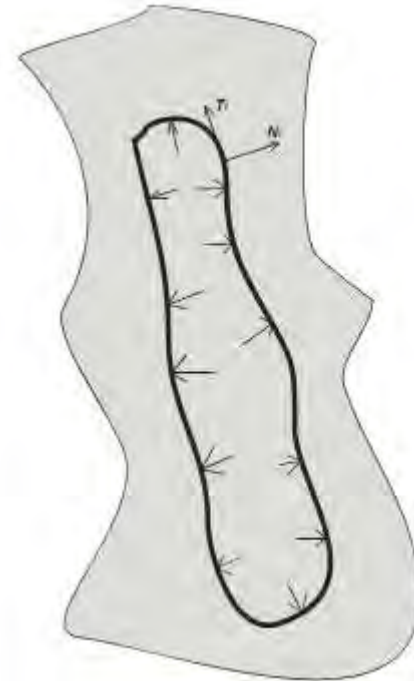


Распирающая составляющая

$$e_{jk}(v_i) = n_i \cdot (v_i - p_{jk}(v_i))$$

n_i — перпендикуляр к t_i

$$t_i = \frac{v_i - v_{i-1}}{\|v_i - v_{i-1}\|} + \frac{v_{i+1} - v_i}{\|v_{i+1} - v_i\|}$$



Внешняя энергия

Внешняя энергия = энергия изображения + энергия градиента

$$b \cdot E_{\text{ext}}(v_i) = m \cdot E_{\text{mag}}(v_i) + g \cdot E_{\text{grad}}(v_i)$$

Энергия изображения

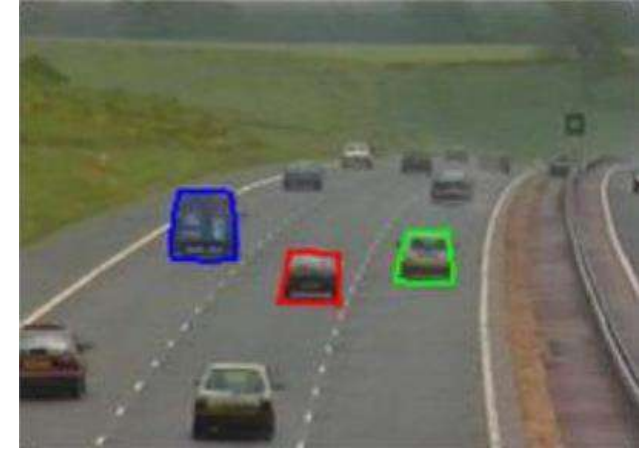
$$e_{jk} = I(p_{jk}(v_i))$$

Энергия градиента

$$e_{jk}(v_i) = -\left| \nabla I(p_{jk}(v_i)) \right|$$

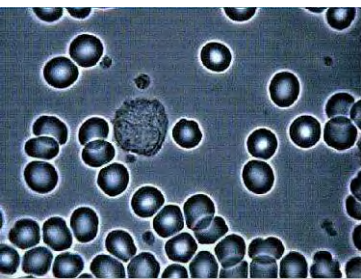
Дополнительные энергии

- Энергия схожести по яркости
- Энергия, соответствующая расстоянию от центра масс фигуры



Сегментация изображений

Лекция 8

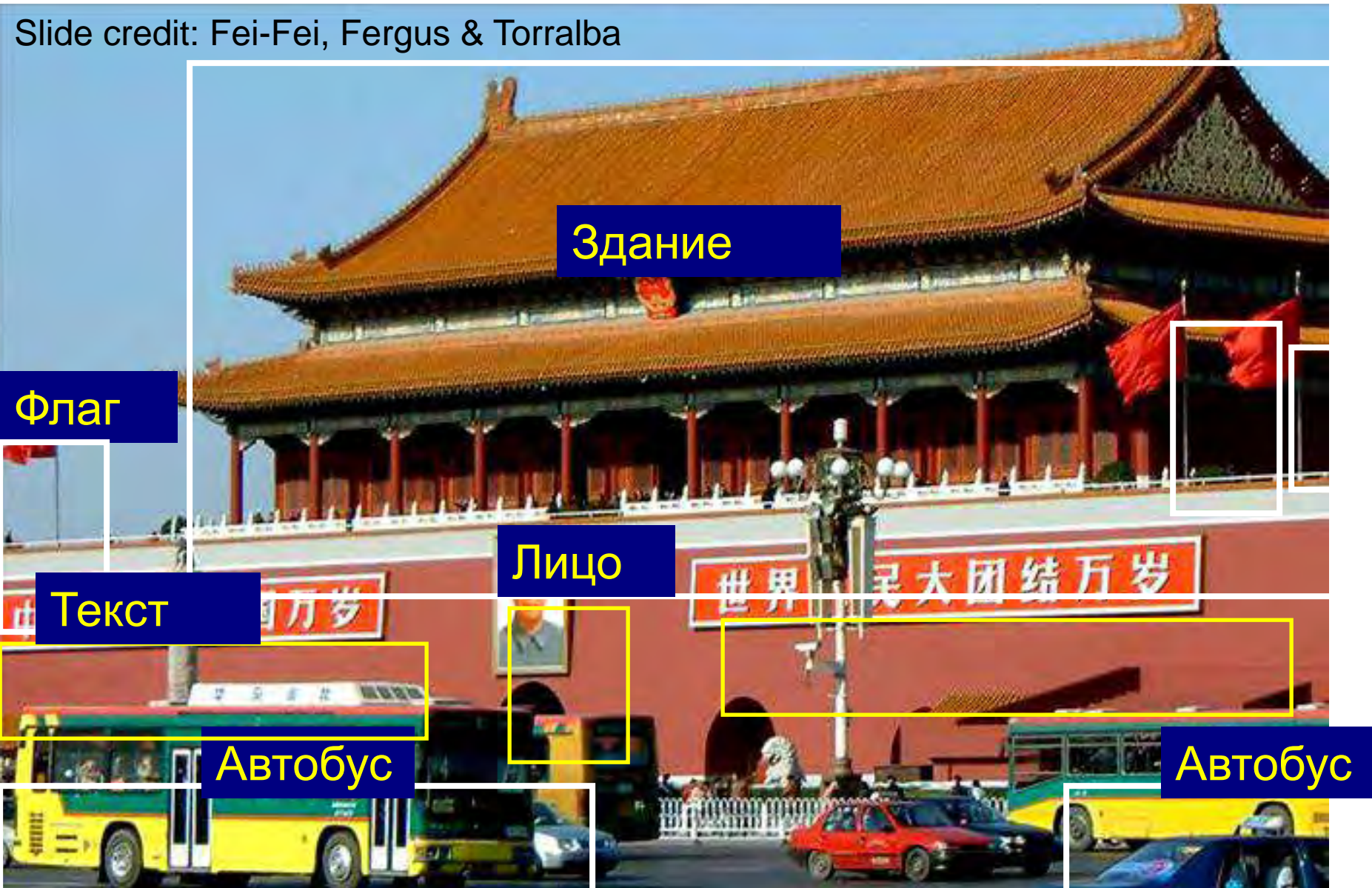


Сколько сегментов?



Такие сегменты?

Slide credit: Fei-Fei, Fergus & Torralba



Грубая сегментация

Небо

Строения

Автомобиль

Автомобиль

Где сегменты?

Голубое

наклонная

Ветер
справа
налево

Нежесткий,
движется

Жесткий,
движется

Мао

Жесткий,
движется

горизонтальный

Определение сегментации

- «Жесткая» сегментация
 - Разбиение изображения на неперекрывающиеся области, однородные по некоторому признаку и покрывающие всё изображение
- Формально:
 - Разбиение изображения на набор областей

$$S = \{S_i\}, i = \overline{1, N}$$

$$I = \bigcup_{i=1..N} S_i$$

$$\forall i, j = \overline{1, N} : i \neq j \quad S_i \cap S_j = \emptyset$$

$$\forall i = \overline{1, N}, P(S_i) = \text{истина}$$

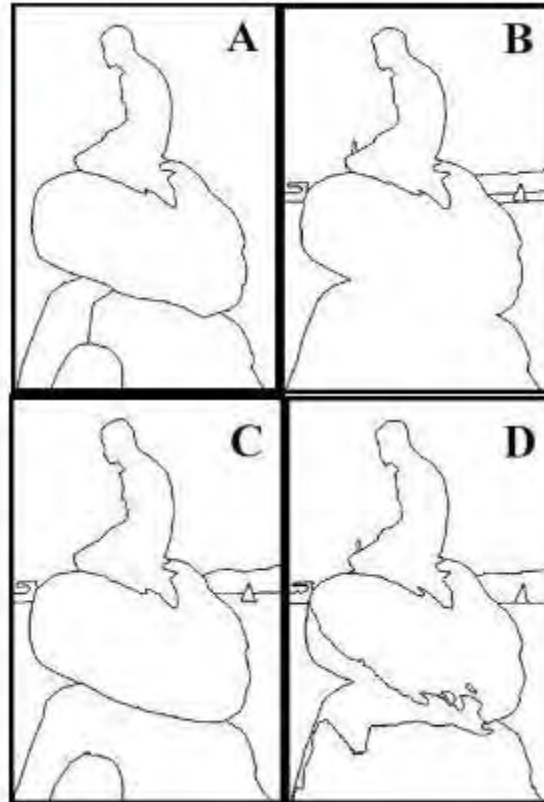
$$\forall i, j = \overline{1, N} : i \neq j \quad P(S_i \cup S_j) = \text{ложь}$$

Цель сегментации изображений

- Декомпозиция изображения на части
- Изменение формы описания изображения

Вопрос. Нужна ли информация из предметной области?

Варианты сегментации изображения. Какой верный?



Berkeley Segmentation Dataset

Оценка качества работы методов сегментации

- Целостность и однородность по некоторому признаку
- Отличие признака для смежных областей
- Отсутствие мелких отверстий внутри
- Гладкие границы

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

более 1000 изображений, отсегментированных вручную 30 разными людьми [18]

Критерии «похожести» пикселей

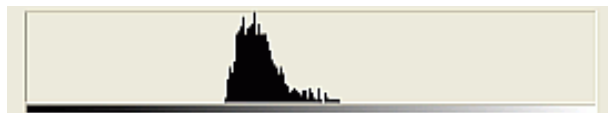
- По расстоянию
- По яркости
- По цвету
- По текстуре

Сегментация по цвету



Критерий однородности

- Гистограмма содержит не больше 1 значительного пика



- Отклонение любого пикселя от средней яркости $< T_{avg}$

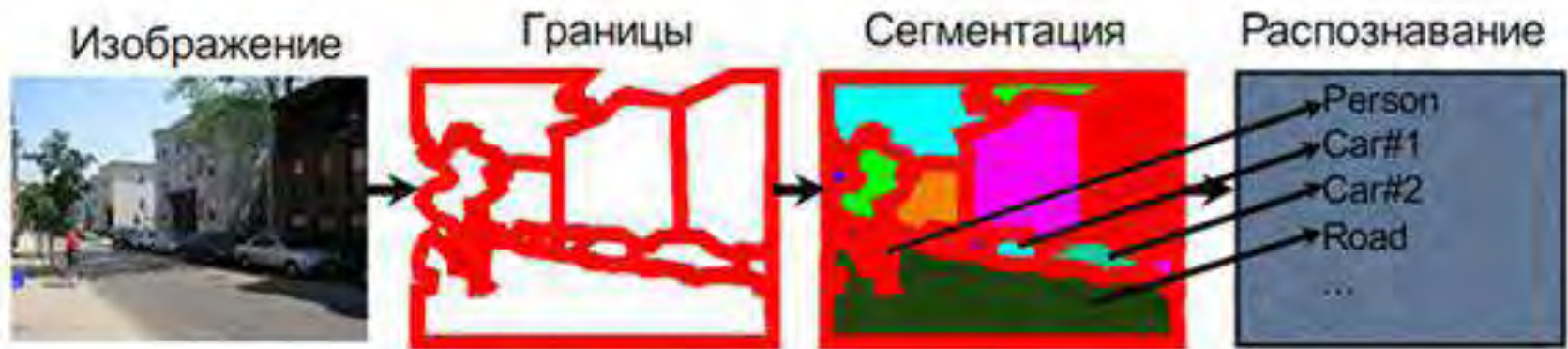
$$\forall p \in S \quad \left| I(p) - \frac{1}{N} \sum_{q \in S} I(q) \right| < T_{avg}$$

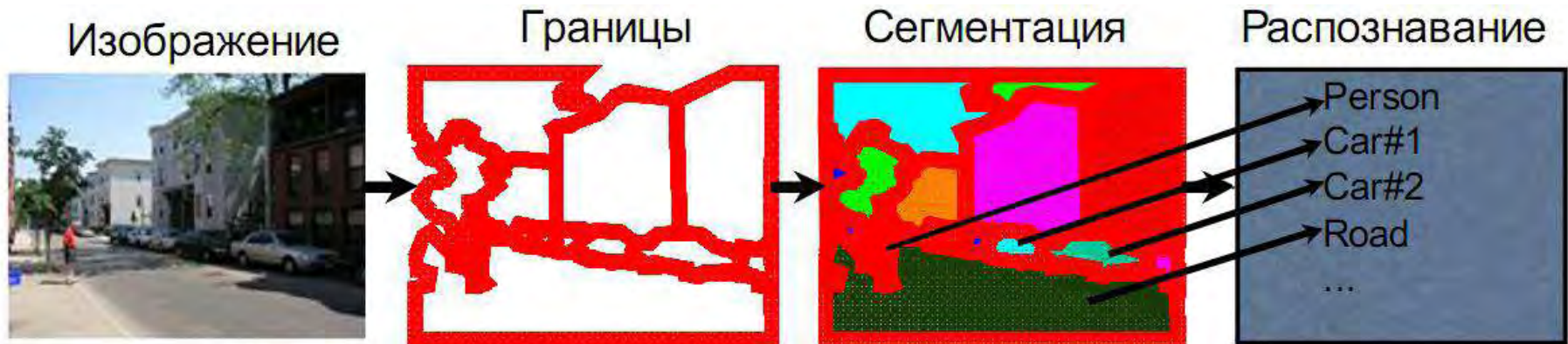
- Разница между соседними пикселями $< T_{diff}$

$$\forall p \in S, \forall q \in N(p) \quad |I(p) - I(q)| < T_{diff}$$

- «Слабая» граница между регионами (только для слияния)

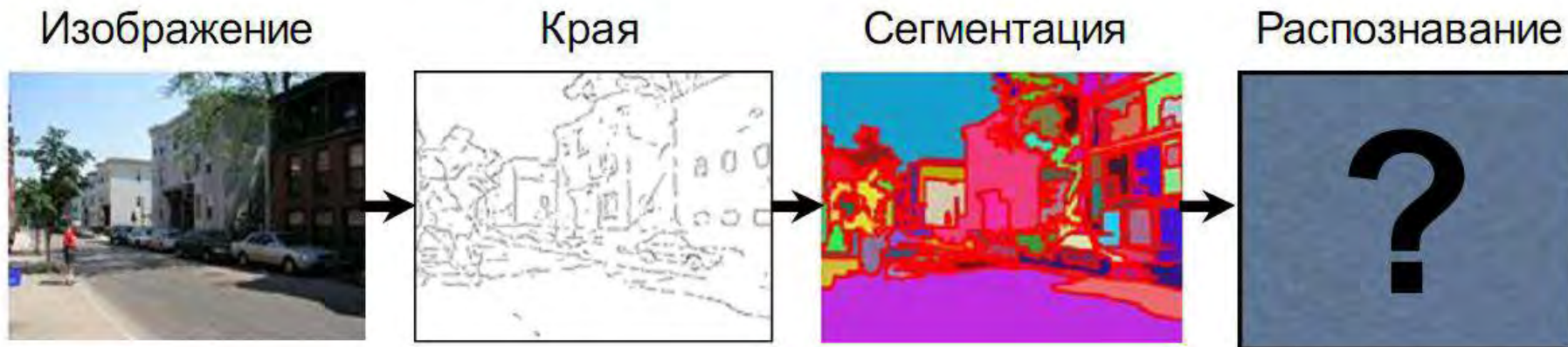
План





Теория ↑

Практика ↓



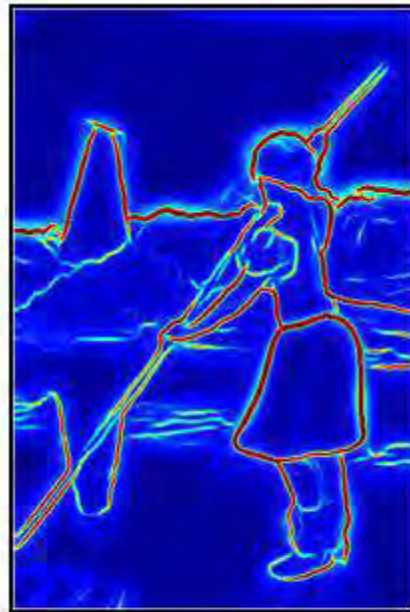
Два типа сегментации

- Автоматическая
 - Сегментация производимая без взаимодействия с пользователем
 - Картинка на входе, регионы на выходе
- Интерактивная
 - Сегментация, управляемая пользователем, допускающая и/или требующая ввода дополнительной информации
 - Пример – «волшебная палочка» в Photoshop

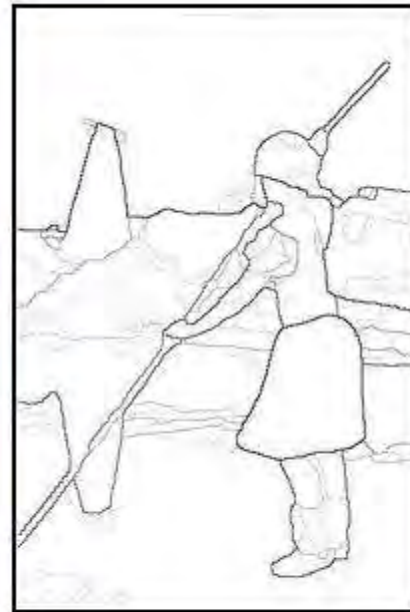
Автоматическая сегментация



Original Image



globalPb



Ultrametric Contour Map



Automatic Segmentation

Задачи автоматической сегментации делятся на два класса

- Выделение областей изображения с известными свойствами
- Разбиение изображения на однородные области

Семейства методов

- **Основанные на формировании однородных областей**
 - **без пространственных связей**
 - с учетом пространственных связей
- Основанные на поиске краев (края и области)
 - Canny
 - Рb-детектор краёв
- Методы на графах
 - Normalized cut
 - «Эффективный метод» Felzenszwalb & Huttenlocher
- Энергетические методы
 - Snakes
 - Методы уровня
 - ТурбоПиксели (TurboPixels)

Сегментация через поиск однородных областей

- **Сегментация без учёта пространственных связей**
 - Пороговая фильтрация
 - Кластеризация по цвету
 - К-средних
 - сдвиг среднего (Mean shift) и развитие
- Сегментация с учётом пространственных связей
 - Разрастание областей (region growing)
 - Слияние/разделение областей (region merging/splitting)
 - Методы водораздела

Этапы кластеризации цветового пространства

- Чтобы свести задачу сегментации к задаче кластеризации, достаточно
 - задать отображение точек изображения в некоторое пространство признаков
 - и ввести метрику (меру близости) на этом пространстве признаков
- Применение методов кластерного анализа
- Обычно после кластеризации точек - выделение связанных компонент

Кластеризация цветового пространства

Наиболее популярный метод кластеризации, используемый для сегментации изображений k -средних



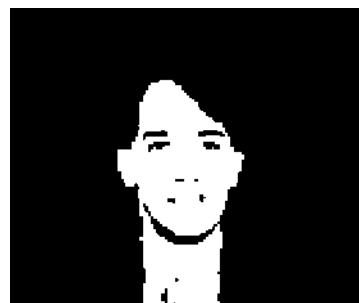
6 кластеров

Методы кластеризации плохо работают на зашумленных изображениях

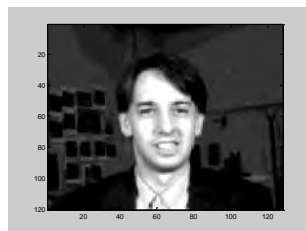
Сравнение k-средних с порогом по средней яркости

Чем отличается сегментация с помощью k-средних на 2 кластера от простейшей пороговой бинаризации по средней яркости изображения?

Пример:



k-средних



Порог по средней яркости

В причинах предлагается разобраться самостоятельно

Семейства методов

- **Основанные на формировании однородных областей**
 - без пространственных связей
 - **с учетом пространственных связей**
- Основанные на поиске краев (края и области)
 - Canny
 - Рb-детектор краёв
- Методы на графах
 - Normalized cut
 - «Эффективный метод» Felzenszwalb & Huttenlocher
- Энергетические методы
 - Snakes
 - Методы уровня
 - ТурбоПиксели (TurboPixels)

Сегментация через поиск однородных областей

- Сегментация без учёта пространственных связей
 - Пороговая фильтрация
 - Кластеризация по цвету
 - K-средних
 - сдвиг среднего (Mean shift) и развитие
- **Сегментация с учётом пространственных связей**
 - Разрастание областей (region growing)
 - Слияние/разделение областей (region merging/splitting)
 - Методы водораздела

Разрастание регионов (Region growing)

- Простая идея – начиная с некоторого “семени” обходить пиксели и присоединять к области пока выполняется условие однородности

Критерии присоединения к региону

- Близость точки к центру региона
- Близость к соседней точке, присоединенной к региону на предыдущем шаге
- Близость по некоторой статистике региона
- Стоимость кратчайшего пути от точки до центра региона

Существуют различные стратегии выбора зерен и выращивания регионов [14, 15, 16, 17]

Разрастание регионов

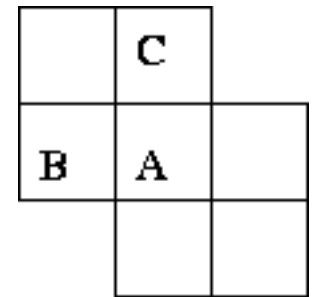
1. if $|I(A) - Cl_{avg}(B)| > \delta$ and $|I(A) - Cl_{avg}(C)| > \delta$ -
создаем новую область, присоединяем к ней пиксел A

2. if $|I(A) - Cl_{avg}(B)| \leq \delta$ xor $|I(A) - Cl_{avg}(C)| \leq \delta$ -
добавить A к одной из областей

3. if $|I(A) - Cl_{avg}(B)| \leq \delta$ and $|I(A) - Cl_{avg}(C)| \leq \delta$:

a) $|Cl_{avg}(B) - Cl_{avg}(C)| \leq \delta$ -
сливаем области B и C.

b) $|Cl_{avg}(B) - Cl_{avg}(C)| > \delta$ -
добавляем пиксел A к тому классу, отклонение от
которого минимально.



Алгоритм разрастания регионов 1

Среднее: 1

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

Пример $\delta = 1$

$$\forall p \in S \left| I(p) - \frac{1}{N} \sum_{q \in S} I(q) \right| \leq \delta$$

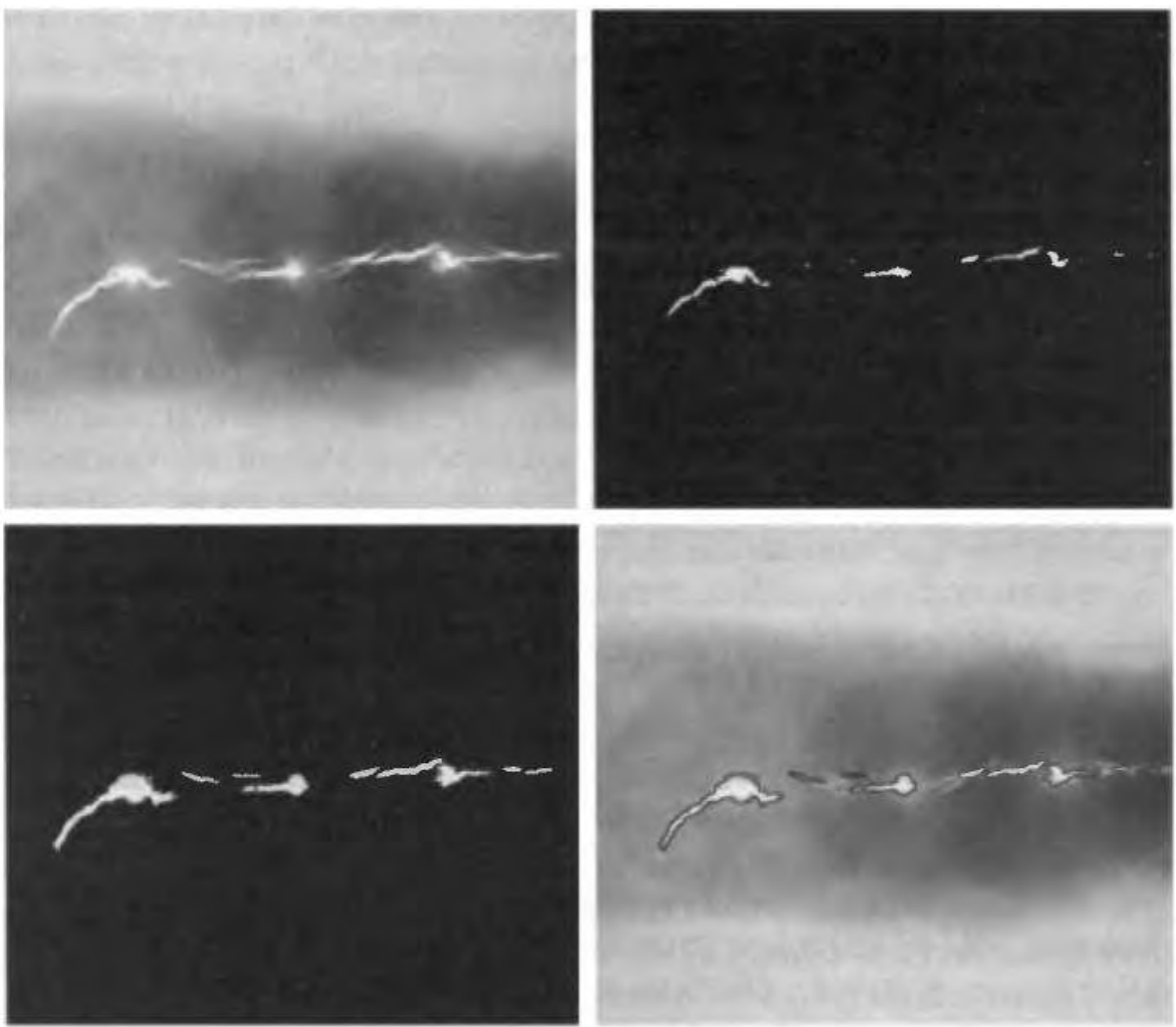
Алгоритм разрастания регионов 2

$$\forall p \in S \left| I(p) - \frac{1}{N} \sum_{q \in S} I(q) \right| \leq \delta$$

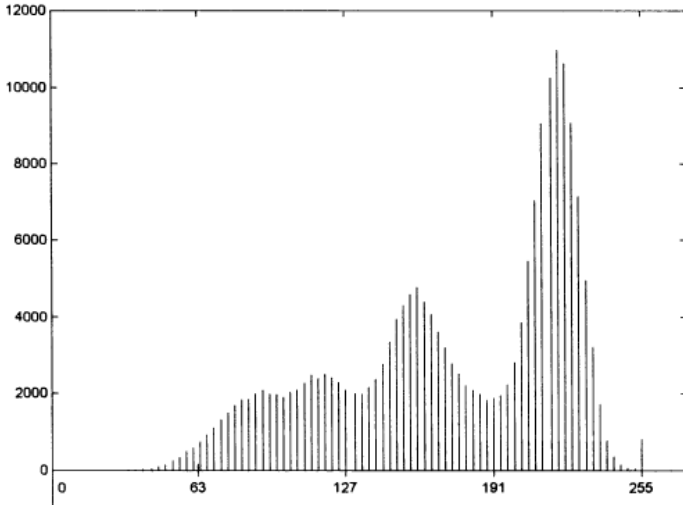
Пример $\delta = 1$



Сварочный шов с дефектами Центры кристаллизации



Гистограмма для сварного шва с дефектами



Границы дефектов, выявленных при сегментации

Результаты
выращивания
областей

Слияние областей

1. Первый шаг – каждый пиксель это отдельная область, поместить все области в стек
2. Пока стек не пуст
 - Взять область S из стека, для всех соседних областей S_i :
 - Проверить $S' = S \cup S_i$ на однородность
 - Если S' однородна -
 - Слить S и S_i , S' поместить в стек, S_i из стека удалить, перейти на 2
 - Если область не однородна
 - Пробуем другого соседа

Разделение областей

1. Первый шаг – всё изображение это одна область, поместить область в стек

2. Пока стек не пуст
 - Взять область S из стека
 - Проверить область на однородность
 - Если область неоднородна
 - разделить ее, новые области поместить в стек
 - Если область однородна
 - область больше не трогаем

Алгоритм разбиения (split)

Первое разбиение

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

Алгоритм разбиения (split)

Второе разбиение

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

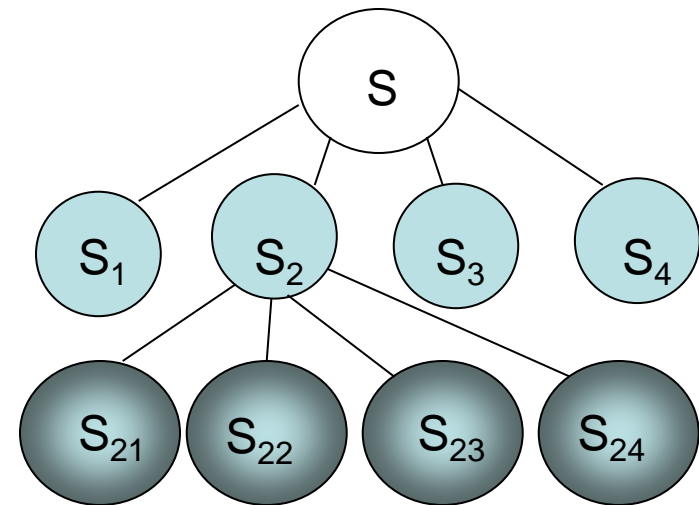
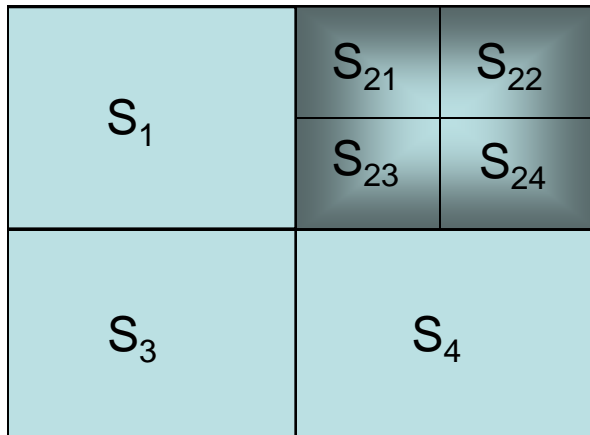
Алгоритм разбиения (split)

Третье разбиение

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

Правило разделения областей 1

- Распространенный вариант – на 4 части, как квадродерево



Просто реализовать, но границы получившихся областей вряд ли будут соответствовать границам объектов

Недостатки алгоритмов разбиения и слияния

- Разбиение
 - Может дать слишком много регионов
 - Если использовать квадродерево, границы скорее всего будут неверны
- Слияние
 - Долго работает, если начинать с индивидуальных пикселей
- **Вывод — нужен комбинированный метод**

Алгоритм разбиения/слияния (split and merge)

- Идея [4, 6] :
 - Сначала провести разбиение (дробление, пересегментация) на небольшие однородные области
 - Обычно используется принцип квадродерева
 - Затем слить между собой те из них, которые вместе не нарушат требование однородности
 - Продолжать до тех пор, пока остаются регионы которые можно объединить

Третье разбиение из алгоритм разбиения (split)

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

Слияние в алгоритме разбиения/слияния (split and merge)

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

Сравним результаты

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

разбиение/слияние

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

разрастание регионов

Сравнение результатов разделения-слияния и пороговой обработки



Исходное



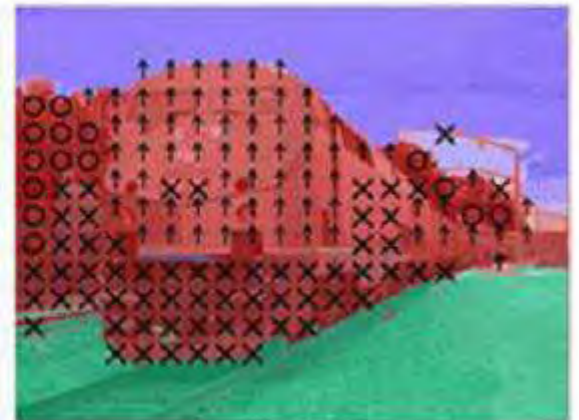
Разделение - слияние



Пороговая обработка

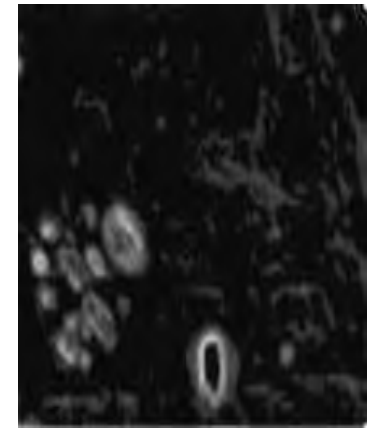
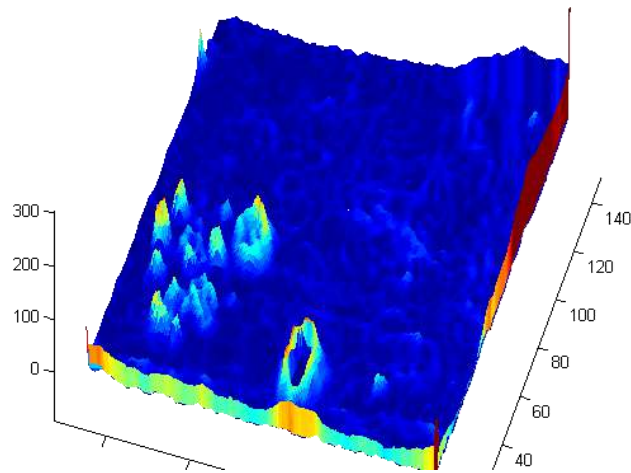
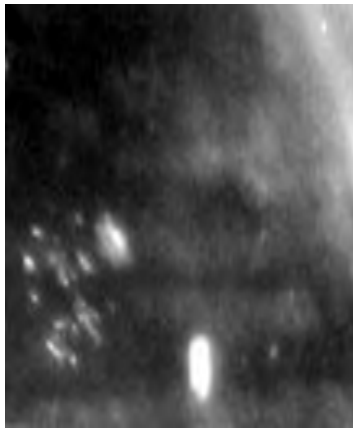
Суперпиксели

- Предобработка перед высокоуровневым анализом



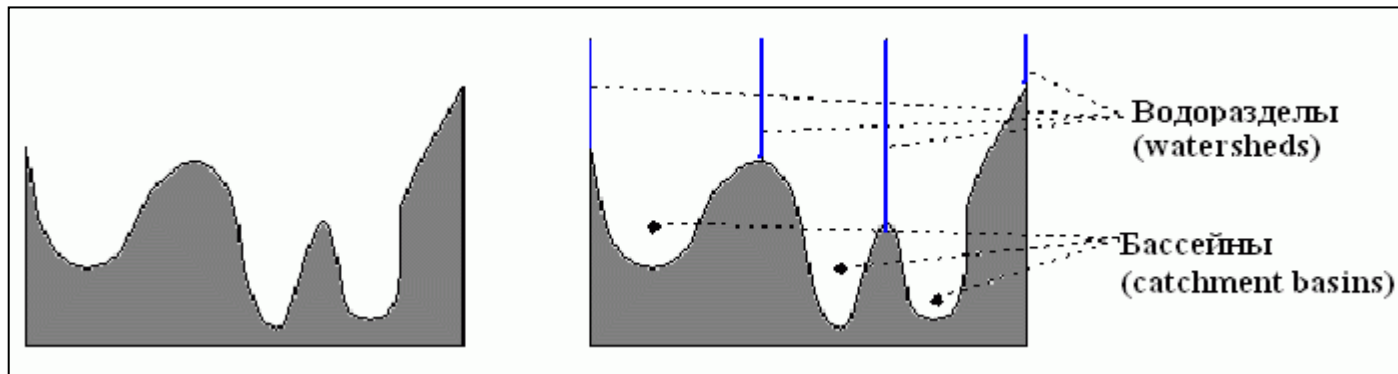
Алгоритмы водораздела (watershed)

Рассмотрим как карту высот ландшафта
– абсолютную величину градиента



На чём основаны алгоритмы водораздела

Область водораздела, бассейн (catchment basin): область, в которой поток из всех точек «стекает» к одной общей точке



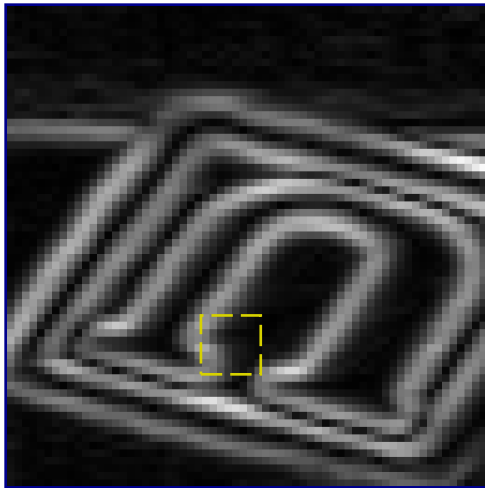
Основные понятия алгоритмов водораздела

1. точки локального минимума
2. точки на склоне, с которых вода скатывается в один и тот же локальный минимум
3. точки на гребне, с которой вода с более-менее равной вероятностью скатывается в более чем один минимум

Бассейн (водосбор) локального минимума

Линии водораздела

Алгоритм tobogganing

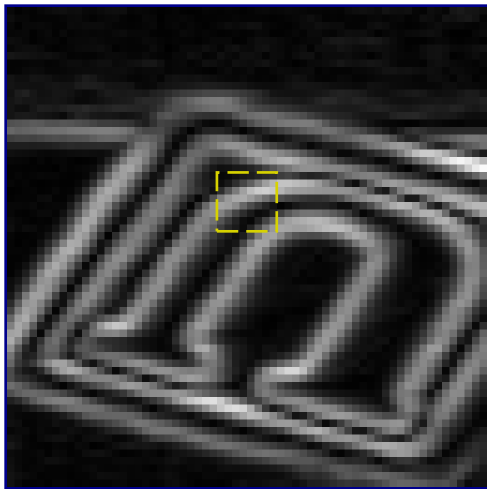


Идея

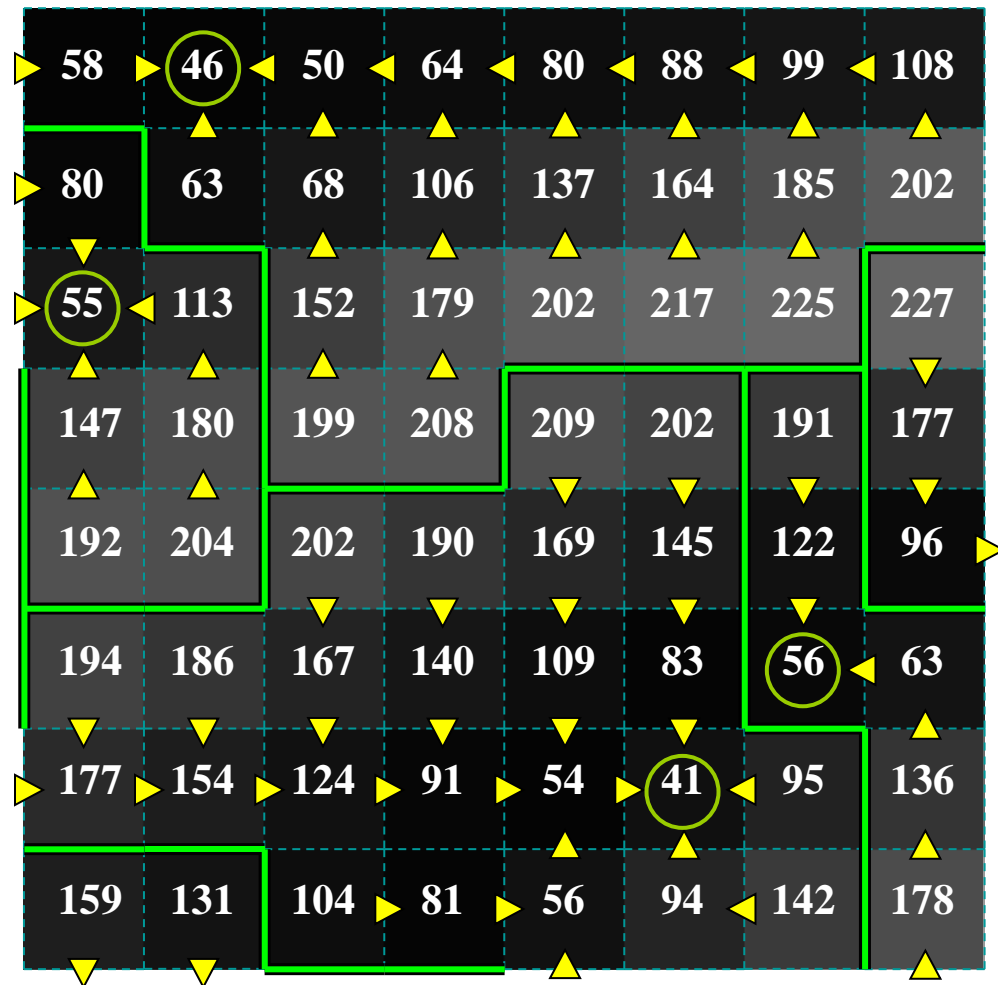
- Из каждого пикселя «спускаемся» в локальный минимум среди его соседей
- Спускаемся до тех пор, пока есть куда спускаться
- Пиксели «спустившиеся» в один минимум – одна область

58	46	50	64	80	88	99	108
80	63	68	106	137	164	185	202
55	113	152	179	202	217	225	227
147	180	199	208	209	202	191	177
192	204	202	190	169	145	122	96
194	186	167	140	109	83	56	63
177	154	124	91	54	41	95	136
159	131	104	81	56	94	142	178

Сегментация алгоритмом tobogganing



- Из каждого пикселя «спускаемся» в локальный минимум среди его соседей
- Спускаемся до тех пор, пока есть куда спускаться
- Пиксели «спустившиеся» в один минимум – одна область



Алгоритм «погружения» (immersion)

Начнем с самых «глубоких» (темных) пикселей
(они определяют начальные бассейны)

Для каждой яркости k :

Для каждой связной компоненты пикселей яркости k :

Если прилежит только к одному существующему бассейну

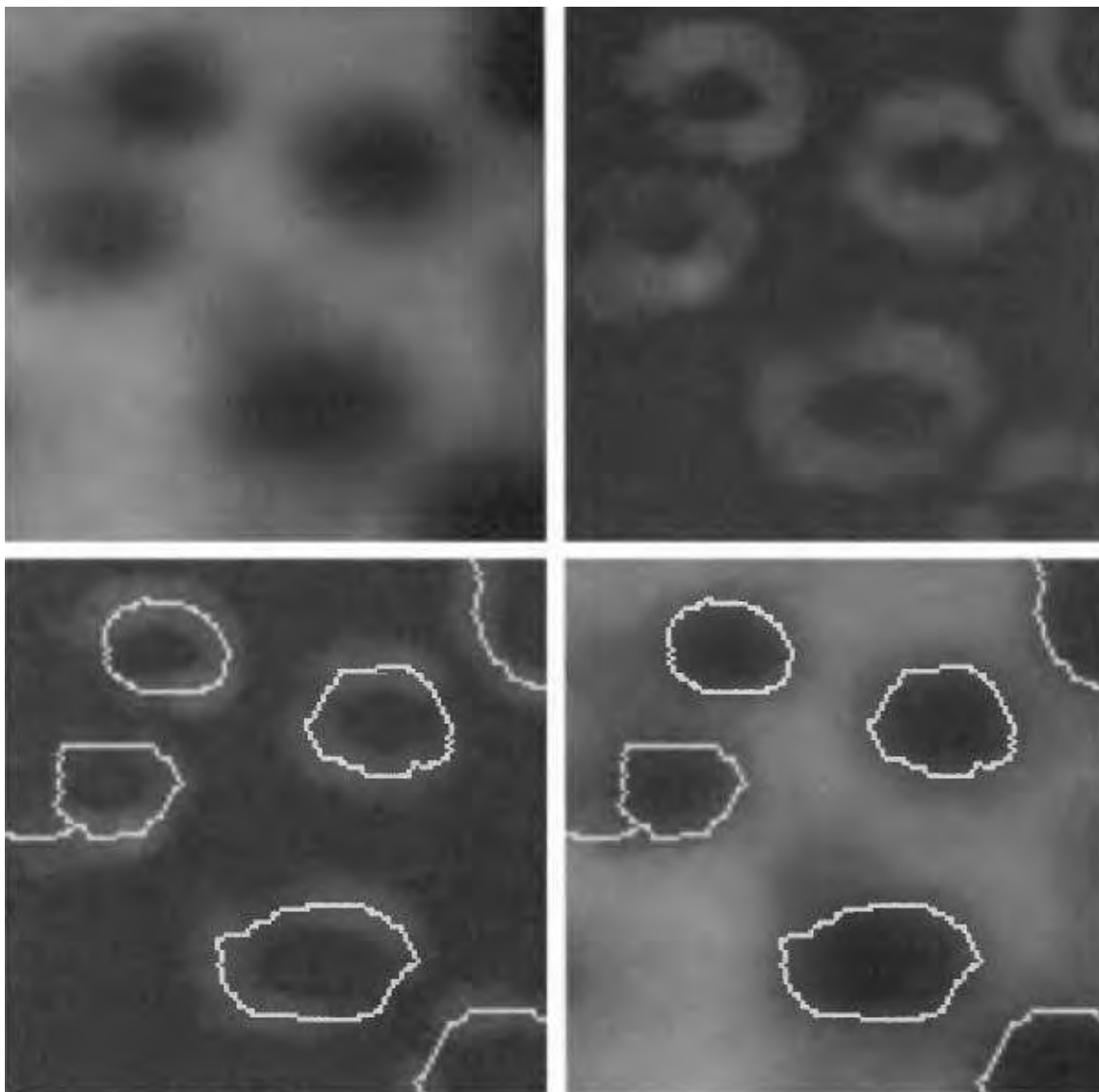
Добавить компоненту к бассейну

Иначе если прилежит более чем к одному существующему бассейну

Пометить как границу (водораздел)

Иначе – создать новый бассейн

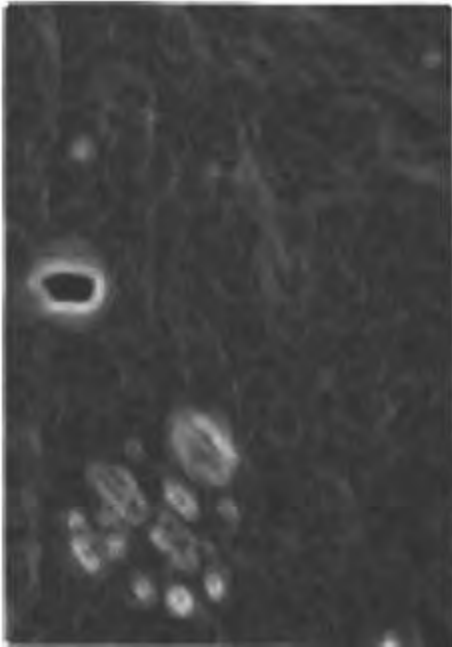
Иллюстрация работы алгоритма погружения



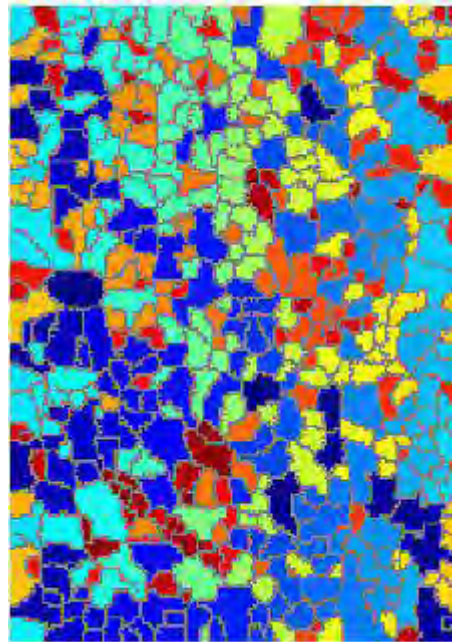
д.Бёше, СММ/Ecole
des Mines de Paris.

Проблемы алгоритмов водораздела

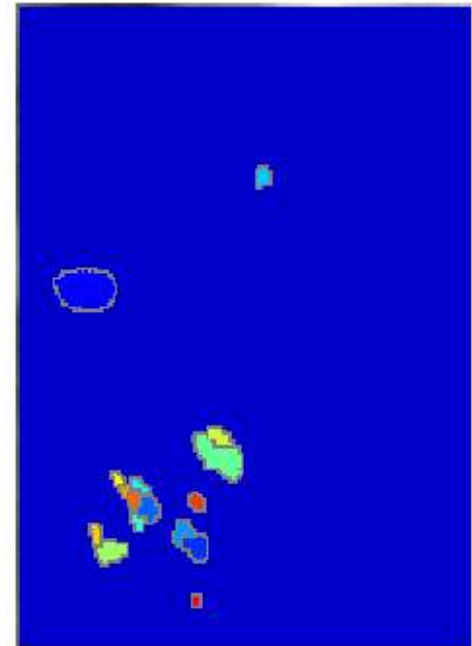
- Алгоритм дает множество небольших регионов
- Очень чувствителен к шуму – ищет все локальные минимумы
- Результат – избыточная сегментация



Абсолютная
величина
градиента

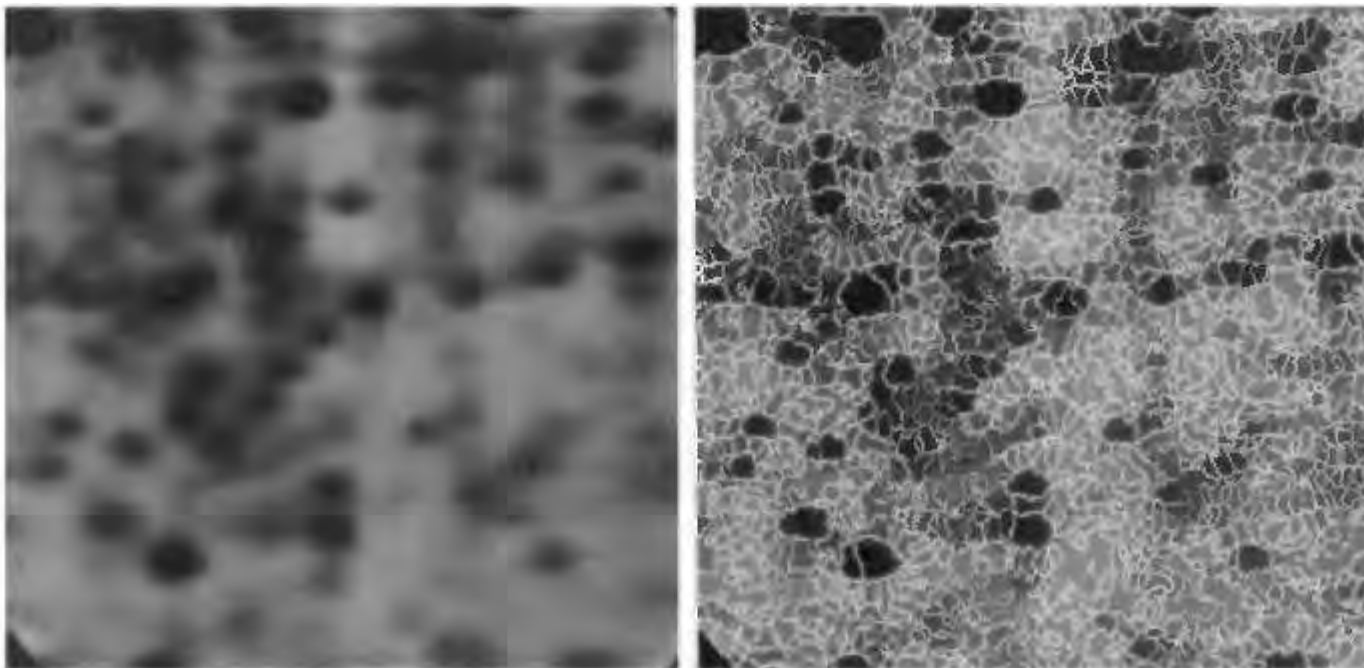


Сегментация по
данному
градиенту

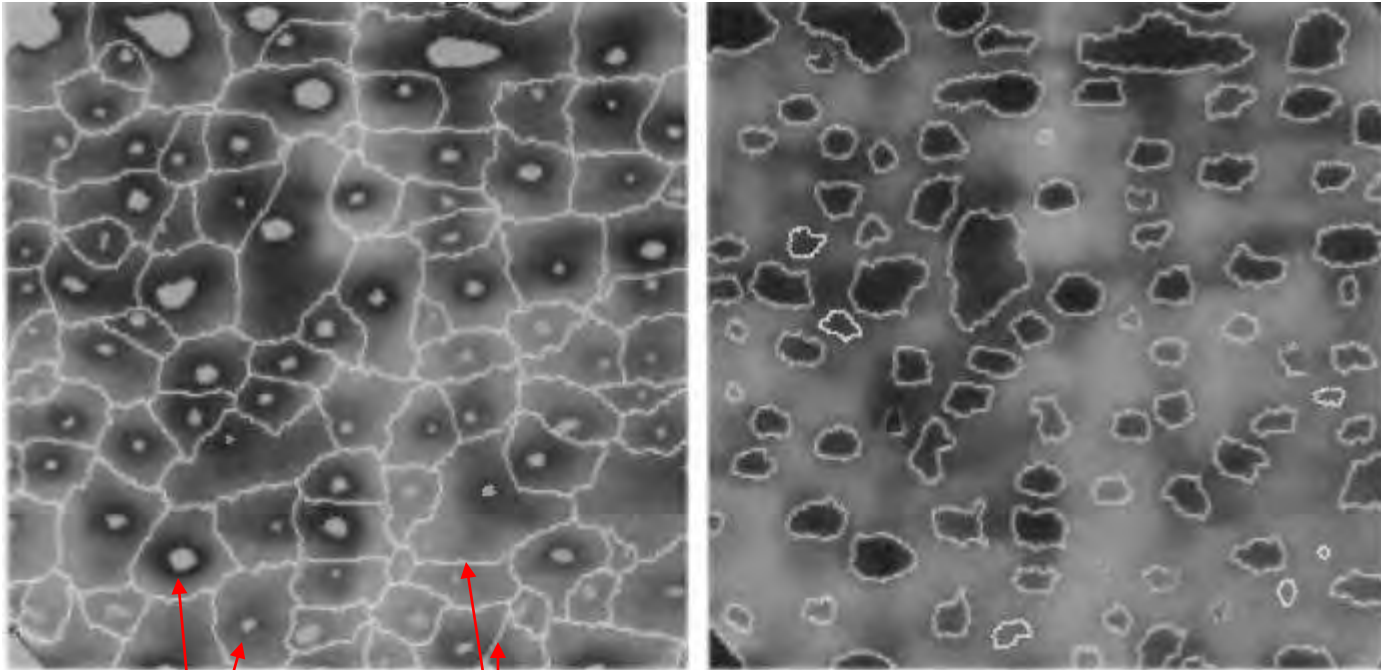


Градиент < 10
обращен в 0

Избыточная сегментация



Использование маркеров



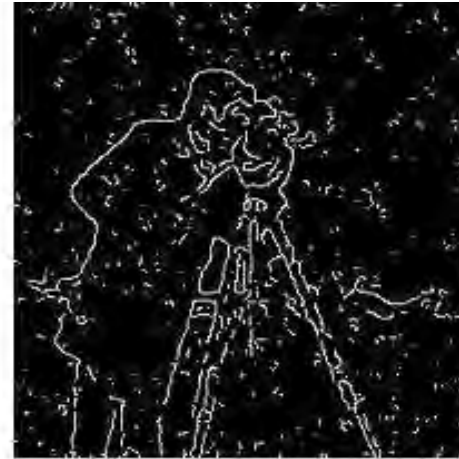
Внутренние
маркеры

Внешние маркеры

Семейства методов

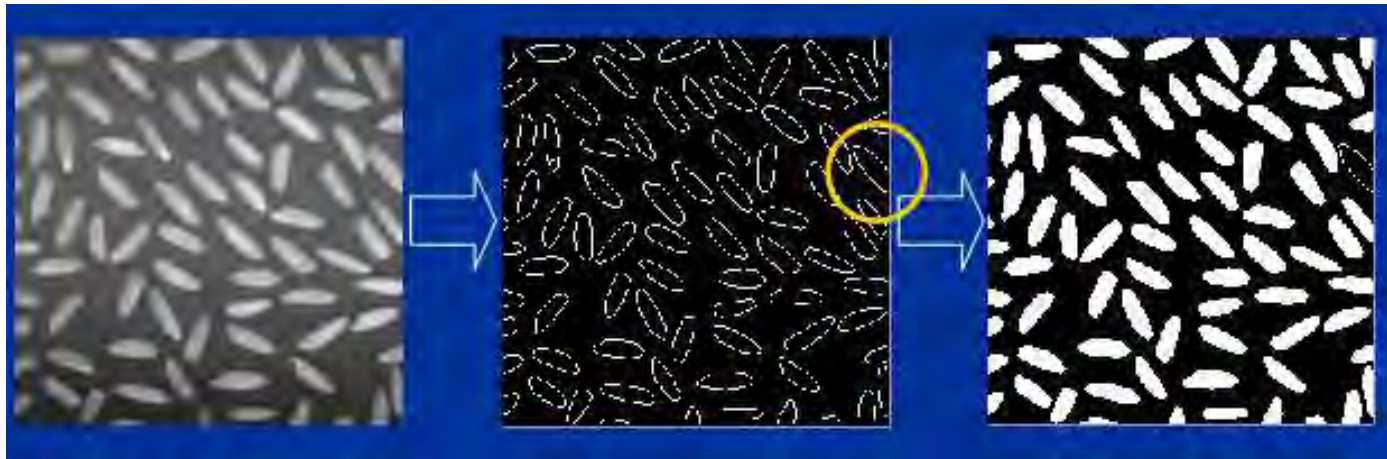
- Основанные на формировании однородных областей
 - без пространственных связей
 - с учетом пространственных связей
- **Основанные на поиске краев (края и области)**
 - Canny
 - Рb-детектор краёв
- Методы на графах
 - Normalized cut
 - «Эффективный метод» Felzenszwalb & Huttenlocher
- Энергетические методы
 - Snakes
 - Методы уровня
 - ТурбоПиксели (TurboPixels)

Методы, основанные на операторах выделения краёв



Сегментация посредством выделения контуров

1. Найдём все контура на изображении алгоритмом Canny или РВ-детектором.
2. Найдём все замкнутые контуры.
3. «Внутренности» замкнутых контуров являются искомыми однородными областями.

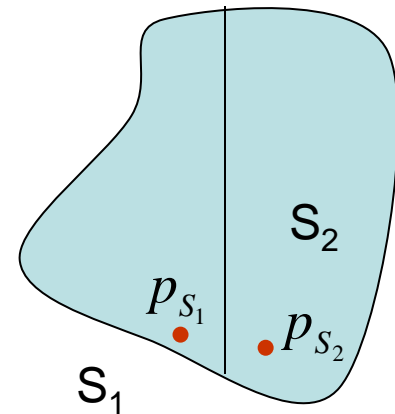


Алгоритм «фагоцита»

- Истаивание границ
 - Убирает слабые границы
- «Слабость границ» определяется по разности яркостей граничных пикселей

$$S(p_{S_1}, p_{S_2}) = |I(p_{S_1}) - I(p_{S_2})|$$

клетка способная захватывать и переваривать посторонние тела

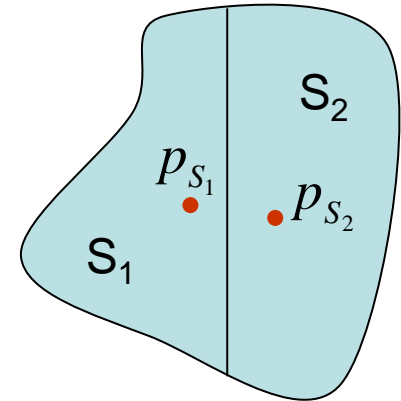


Алгоритм «фагоцита»

$$S(p_{S_1}, p_{S_2}) = |I(p_{S_1}) - I(p_{S_2})|$$

$$W(p_{S_1}, p_{S_2}) = \begin{cases} 1 & S(p_{S_1}, p_{S_2}) > T \\ 0 & \text{иначе} \end{cases}$$

$$W(S_1, S_2) = \sum_{p_{S_1} \in R_1 \wedge p_{S_2} \in R_2} W(p_{S_1}, p_{S_2})$$



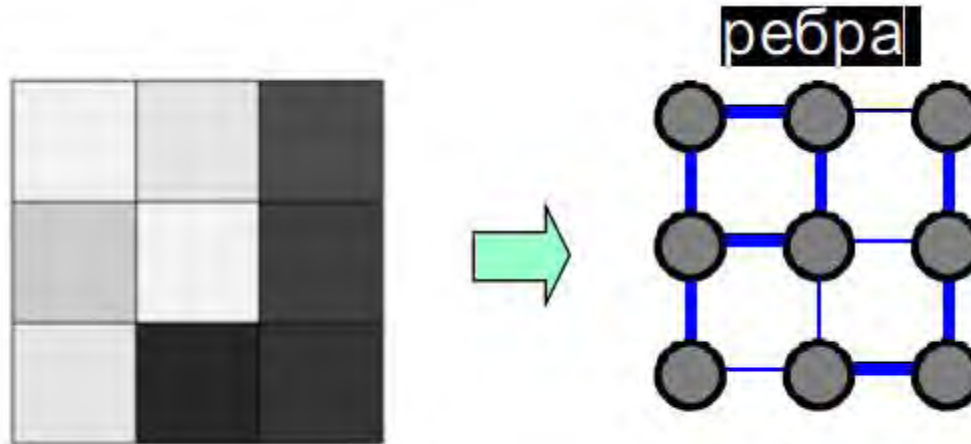
Слить две области если:

$$\frac{W(\text{граница})}{\text{кол - во Точек границы}} < T_3, \quad 0 < T_3 \leq 1$$

Семейства методов

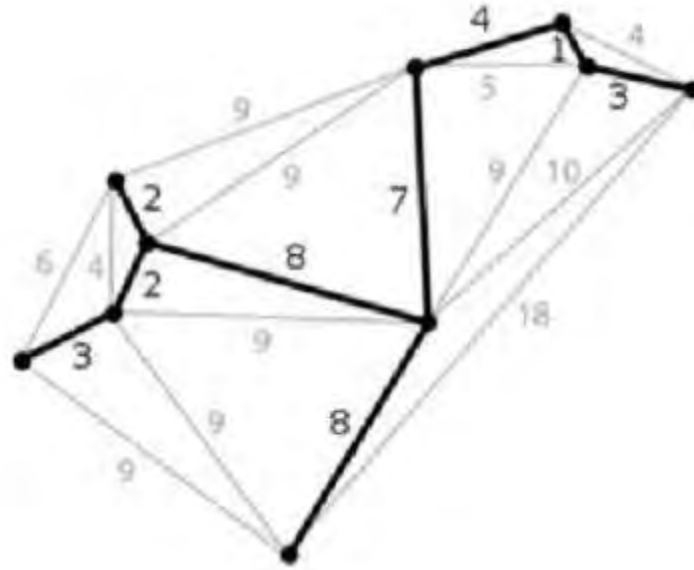
- Основанные на формировании однородных областей
 - без пространственных связей
 - с учетом пространственных связей
- Основанные на поиске краев (края и области)
 - Canny
 - Рb-детектор краёв
- **Методы на графах**
 - Normalized cut
 - «Эффективный метод» Felzenszwalb & Huttenlocher
- Энергетические методы
 - Snakes
 - Методы уровня
 - ТурбоПиксели (TurboPixels)

Представление в виде графа



- Изображение превращается во взвешенный неориентированный граф $G = (V, E)$
 - Вершины графа V – пиксели изображения
 - Ребра E – связи между соседними пикселями
 - Вес ребер пропорционален «похожести» пикселей

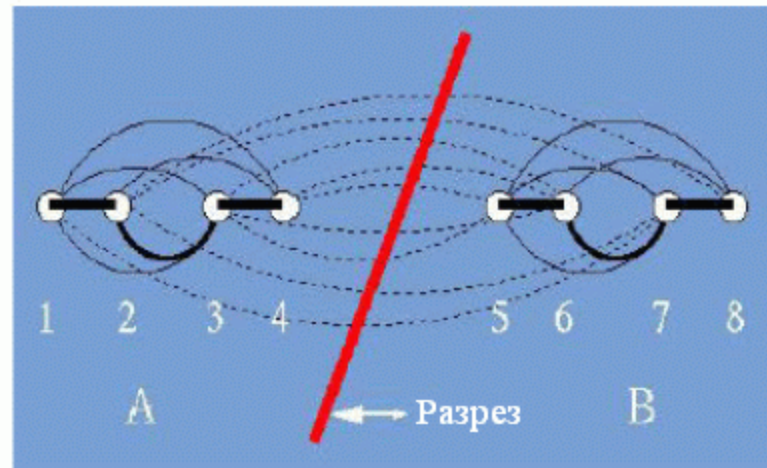
Minimum Spanning Tree (MST)



- Построим минимальное остовное дерево
 - Дерево, которое содержит все вершины графа, сумма весов ребёр которого минимальна
- MST для связанной компоненты (сегмента) содержит рёбра, обеспечивающие наибольшую «похожесть» пикселов в компоненте

Сегментация с помощью разрезов графа

- Создать граф
- Разрезать граф
- Каждую связную компоненту после разреза рассматривать как отдельную область

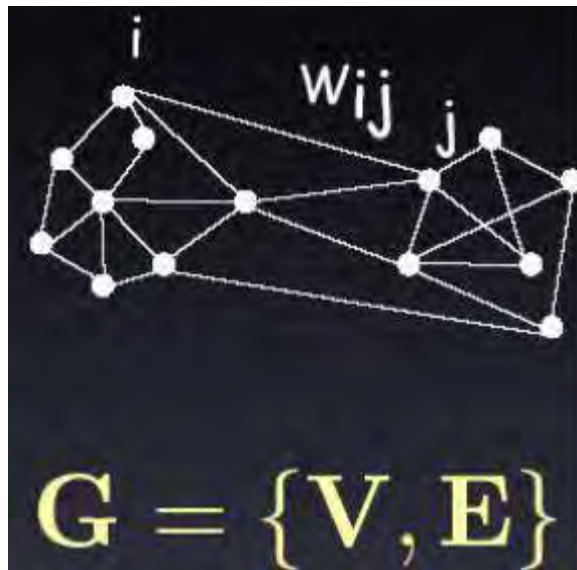


Разрез графа

- $G=(V,E)$
 - Непересекающиеся подмножества вершин A и B из V
 - Удаляем все ребра, связывающие A и B

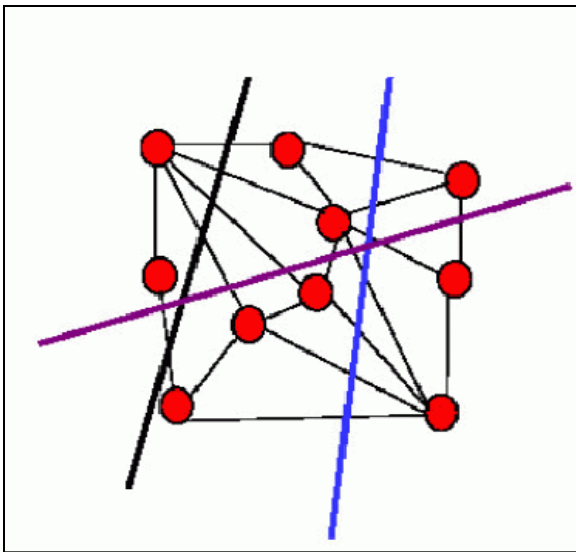
$$Cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

- $Cut(A, B)$ – мера «силы связности» множеств A и B



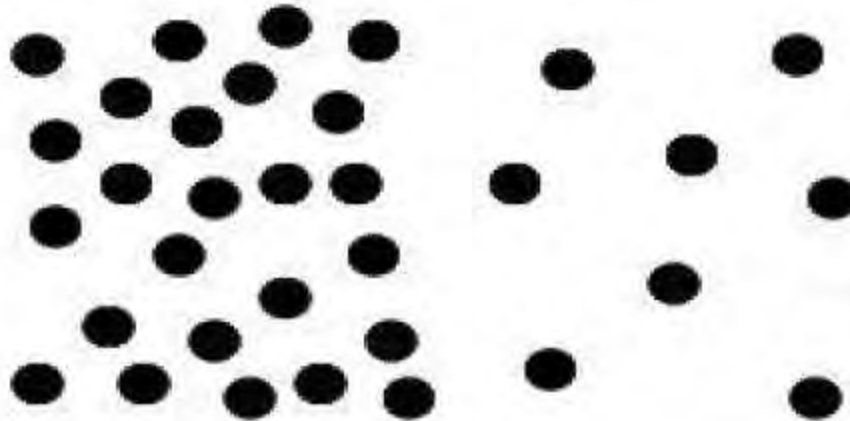
Минимальный разрез графа

- Если множества A и B не заданы заранее – разрезать граф можно по-разному:
 - **Минимальный разрез** – разрез, превращающий граф в несвязный, с минимальной суммой весов удаленных ребер



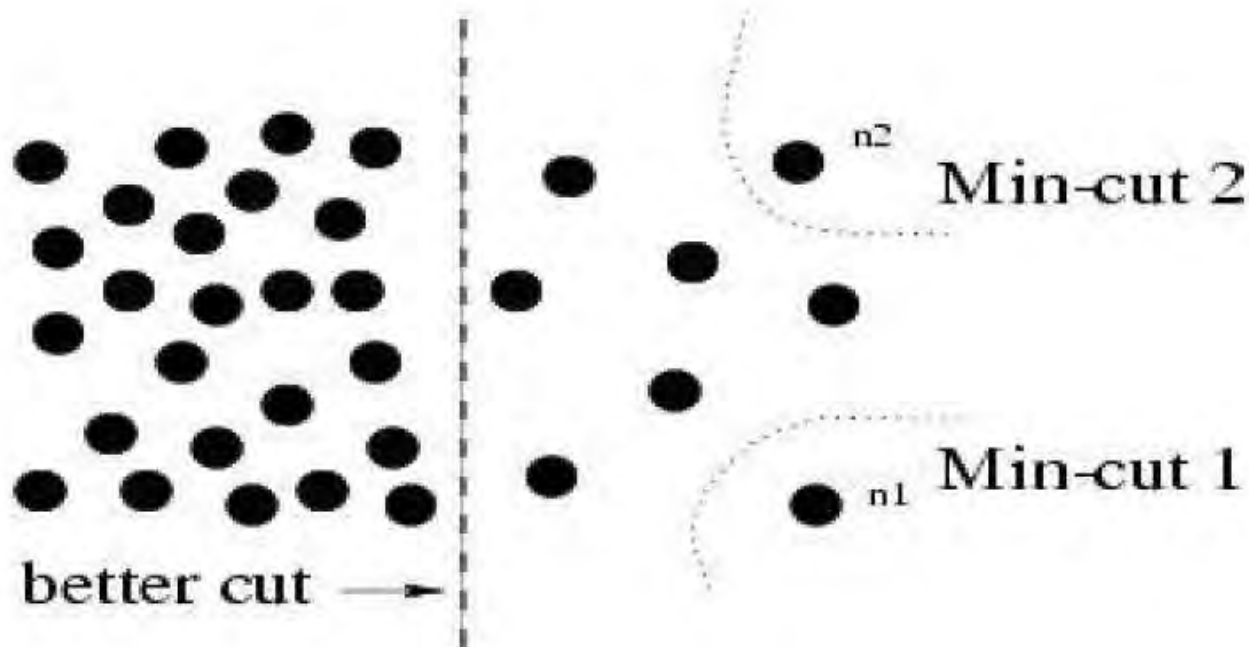
$$Cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

Как бы разбили вы?



Минимальный разрез хорош не всегда

- На данном рисунке вес ребер графа показан расстоянием между вершинами



Нормализованный разрез графа (Normalized cut)

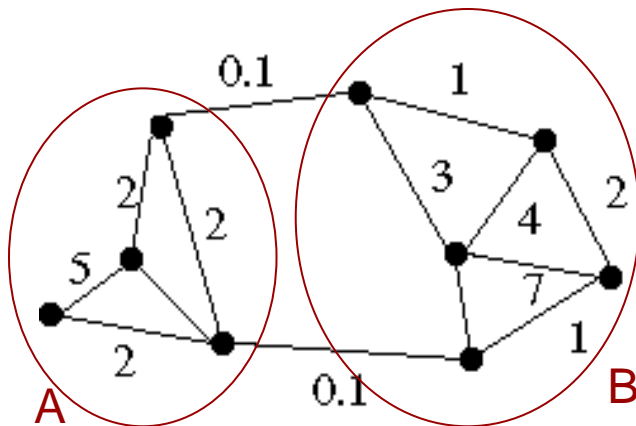
- **Другая мера разреза** – измеряет «похожесть» двух групп вершин, нормированную на «объем», занимаемый ими в графе

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

$$assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$$

Минимальный нормализованный разрез

- **Минимальный нормализованный разрез** – разрез, превращающий граф в несвязный, с минимальной величиной $NCut$
- *Как его найти?*



$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

$$assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$$

Алгоритм сегментации с помощью normalized cuts (по Ши)

1. Задать граф на изображении

Вес ребра (по Ши)

$$w(i, j) = e^{\frac{-\|F(i) - F(j)\|_2}{\sigma_I}} * \begin{cases} e^{\frac{-\|X(i) - X(j)\|_2}{\sigma_X}} & \text{if } \|X(i) - X(j)\|_2 < r \\ 0 & \text{otherwise} \end{cases}$$

$$- F(i) = I(i)$$

$$- F(i) = [v, v \cdot s \cdot \sin(h), v \cdot s \cdot \cos(h)](i)$$

where h , s , and v are the HSV values

Алгоритм сегментации с помощью normalized cuts (по Ши)

2. Рассчитать матрицы W и D

$$D = \begin{bmatrix} \sum_j w(1, j) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sum_j w(N, j) \end{bmatrix}, \quad W = \begin{bmatrix} w(1,1) & \cdots & w(1,N) \\ \vdots & \ddots & \vdots \\ w(N,1) & \cdots & w(N,N) \end{bmatrix}$$

Можно вывести что:

$$\text{MinNcut}(G) = \min_y \frac{y^t (D - W) y}{y^t D y}$$

NP-трудная задача



Если разрешить $y \in \mathfrak{R}$ задача сводится к задаче на собственные значения:

$$(D - W) y = \lambda D y$$

Алгоритм сегментации с помощью normalized cuts

3. Решить задачу $(D-W)y = \lambda Dy$, найти вектора с наименьшими собственными значениями
4. По вектору со вторым наименьшим с.з. разрезать граф на две части
5. Рекурсивно разбить получившиеся области, если требуется

Jianbo Shi and Jitendra Malik "Normalized Cuts and Image Segmentation",
IEEE PAMI, 2000

https://repository.upenn.edu/cgi/viewcontent.cgi?article=1101&context=cis_papers

<https://people.eecs.berkeley.edu/~malik/papers/SM-ncut.pdf>

Результат сегментации Ши



(a)



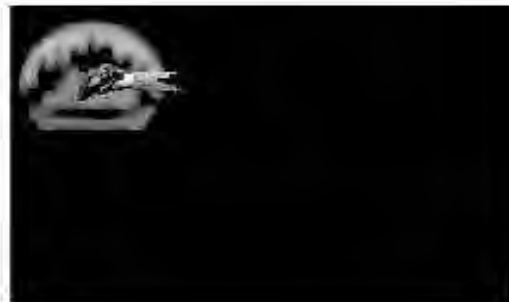
(b)



(c)



(d)

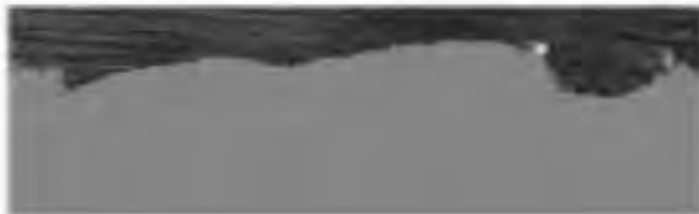


(e)



(f)

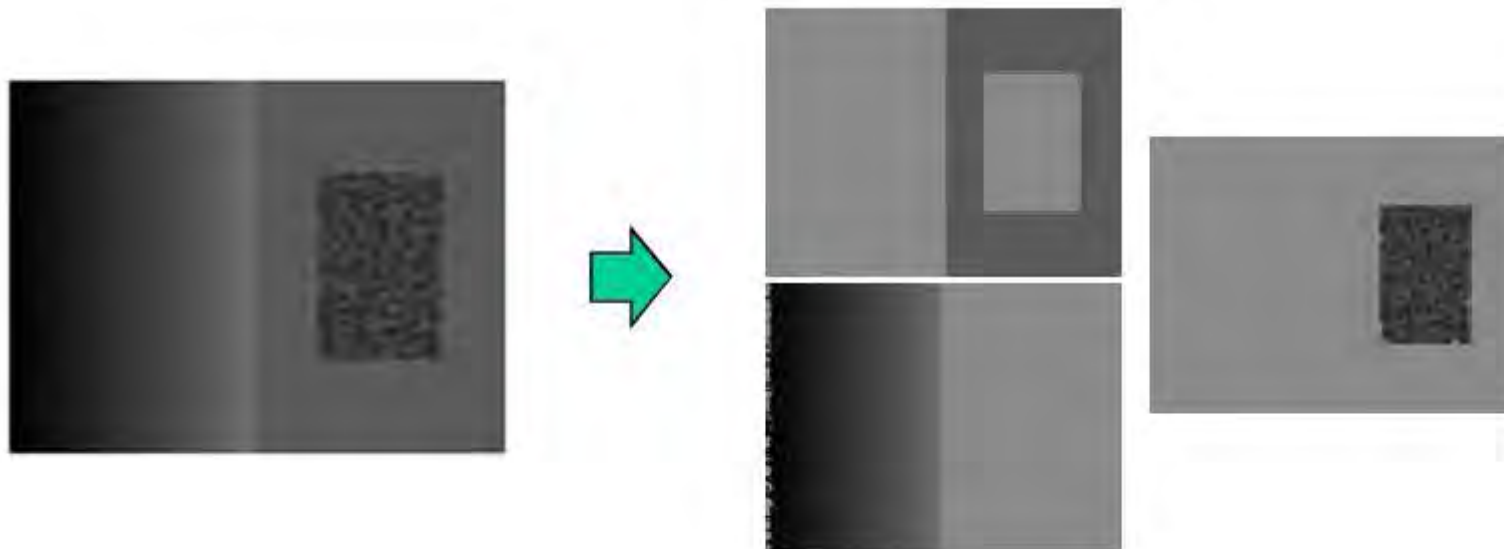
Результат сегментации Ши



Анализ алгоритма

- Метод требует хранения матрицы размером $n*n$, где n – число точек изображения, и потому в исходном виде неприменим к большим изображениям.
- Для данного метода предложены модификации [24, 25], позволяющие сократить сложность алгоритма и требования по памяти за счет аппроксимации матрицы расстояний. Такой подход дает выигрыш в скорости работы в 10-20 раз по сравнению с исходным методом.

«Эффективный метод»



- Идея: разница в интенсивности вдоль границы между областями должна быть существенной по сравнению с колебаниями интенсивности внутри одной из областей

P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. IJCV, 59(2):167–181, 2004.

Формализация

- «Внутренняя разница» в компоненте C :

$$Int(C) = \max_{e \in MST(C,E)} w(e) \quad \text{где } w(e) \text{ - вес ребра (мера различия двух пикселей)}$$

- Разница между областями:

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w((v_i, v_j))$$

- Предикат присутствия границы между областями:

$$D(C_1, C_2) = \begin{cases} \text{true} & \text{if } Dif(C_1, C_2) > MInt(C_1, C_2) \\ \text{false} & \text{otherwise} \end{cases}$$

где $MInt$ – минимальные колебания интенсивности по областям

$$MInt(C_1, C_2) = \min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2))$$

и регуляризационный параметр $\tau(C) = k/|C|$

где $|C|$ - размер области, k – коэффициент регуляризации

Алгоритм

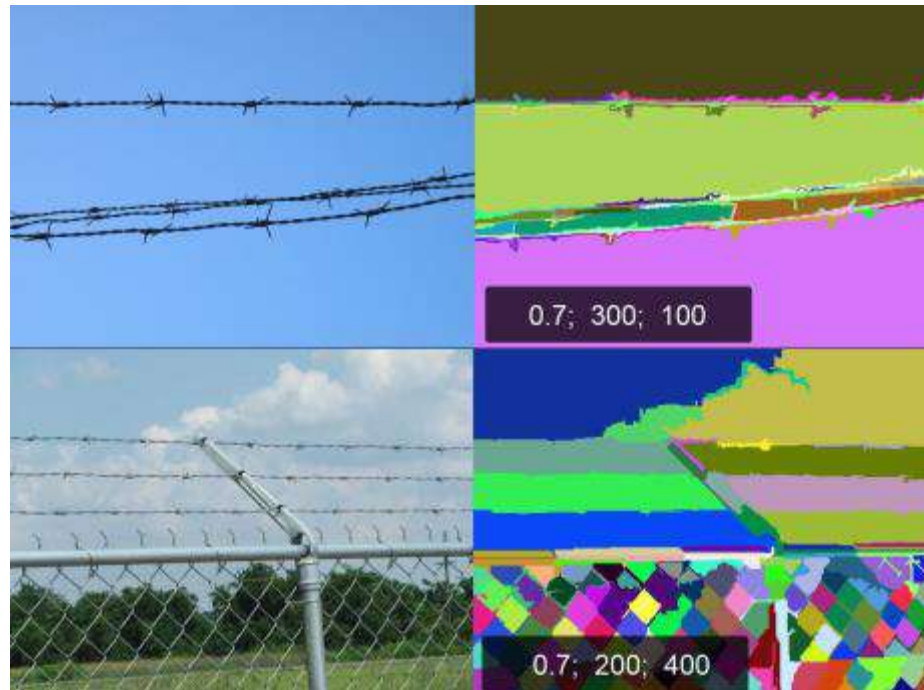
- **Схема «слияния регионов», но с учётом выбранных функций слияния:**
 - Сортируем все ребра по возрастанию веса
 - Инициализируем сегментацию максимальным разбиением (у каждого пиксела своя компонента)
 - Проходим по списку всех ребер (i,j)
 - Пусть C_i, C_j – компоненты, которым принадлежат вершины i, j
 - Тогда если не выполняется $D(C_i, C_j)$, тогда объединяем C_i и C_j

Сложность метода $O(N \log N)$

Результат



Как избавиться от сетки?

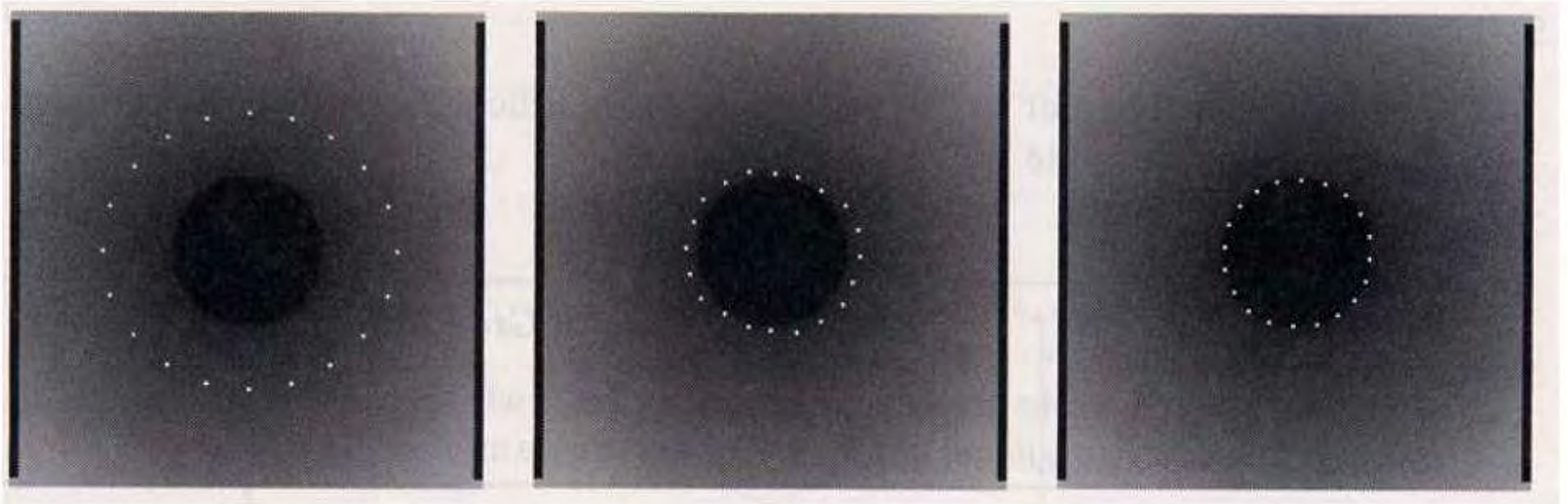


$$dist(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2}$$

Семейства методов

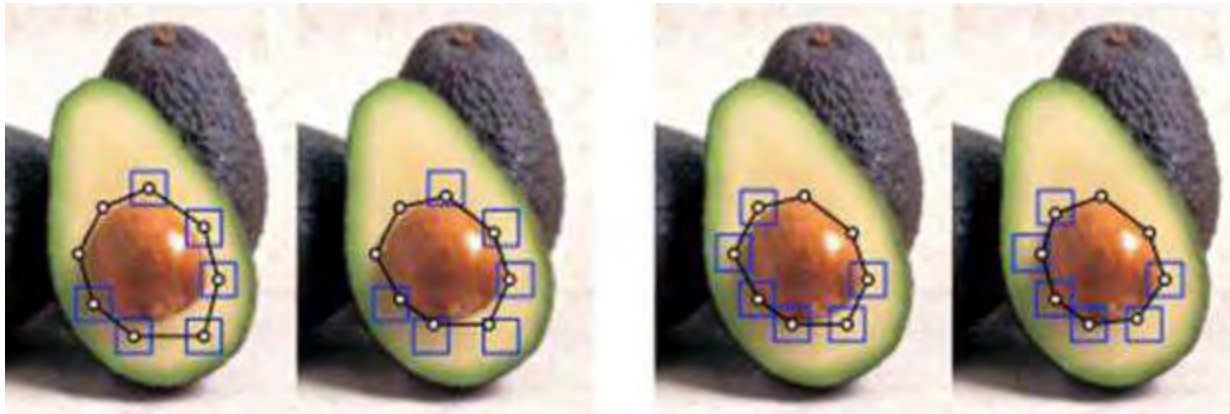
- Основанные на формировании однородных областей
 - без пространственных связей
 - с учетом пространственных связей
- Основанные на поиске краев (края и области)
 - Canny
 - Рb-детектор краёв
- Методы на графах
 - Normalized cut
 - «Эффективный метод» Felzenszwalb & Huttenlocher
- **Энергетические методы**
 - Snakes (активные контуры)
 - Методы уровня
 - ТурбоПиксели (TurboPixels)

Активный контур (Snakes)



M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. IJCV, 1988

Вариант алгоритма Snakes



Метод фиксации уровня (Level set method)

- введён американскими математиками Стэнли Ошером и Джеймсом Сетьяном в 1980-е годы.
- Он стал популярным во многих дисциплинах, таких как
 - компьютерная графика,
 - обработка изображений,
 - вычислительная геометрия,
 - оптимизация,
 - вычислительная гидродинамика и
 - вычислительная биофизика

Level Sets (Линии уровня)



Визуализация представления контура с помощью функции вложения

- Вместо явного представления контура зададим функцию вложения (embedding function) f
- $f > 0$ в области, $f < 0$ вне области
- Можем переформулировать задачу обновления контура как задачу обновления функции вложения

- Функция вложения позволяет моделировать изменения топологии области
- Работа с ней сводится к решению систем дифференциальных уравнений

Level set method

- Начальная маска. Это объект любой формы, находящийся внутри интересующего нас объекта. Чаще всего это прямоугольник, потому что его довольно просто задать.
- После определенного количества итераций функция модификации уровня меняется таким образом, чтобы ее пересечение с нулевой плоскостью давало интересующую нас кривую

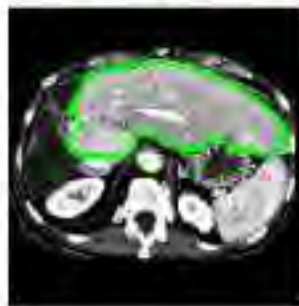
Входное изображение



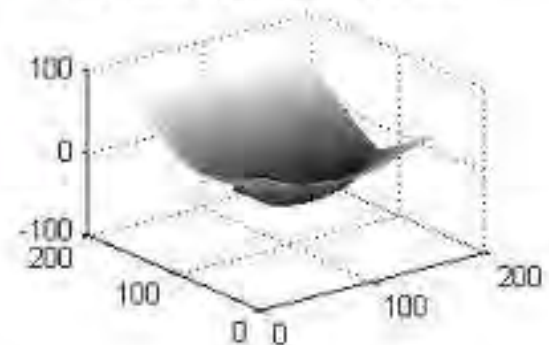
Маска



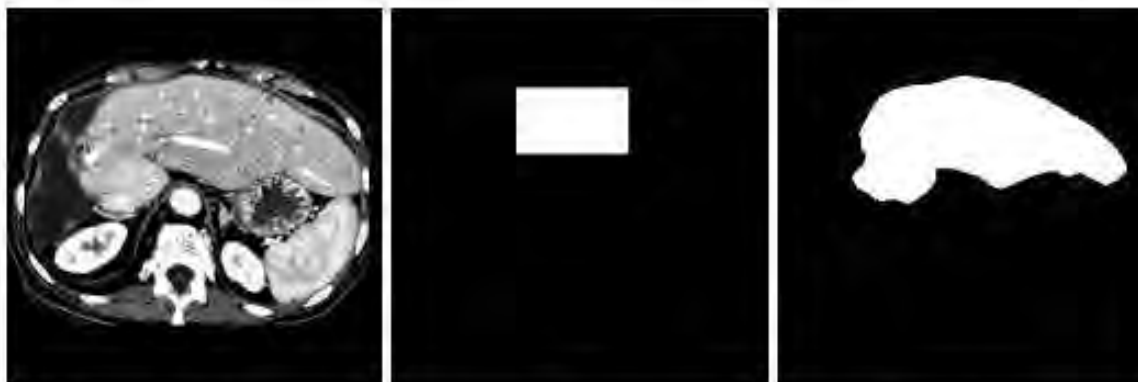
500 итераций



Функция фиксации уровня



Для медицинских изображений это работает вот так:

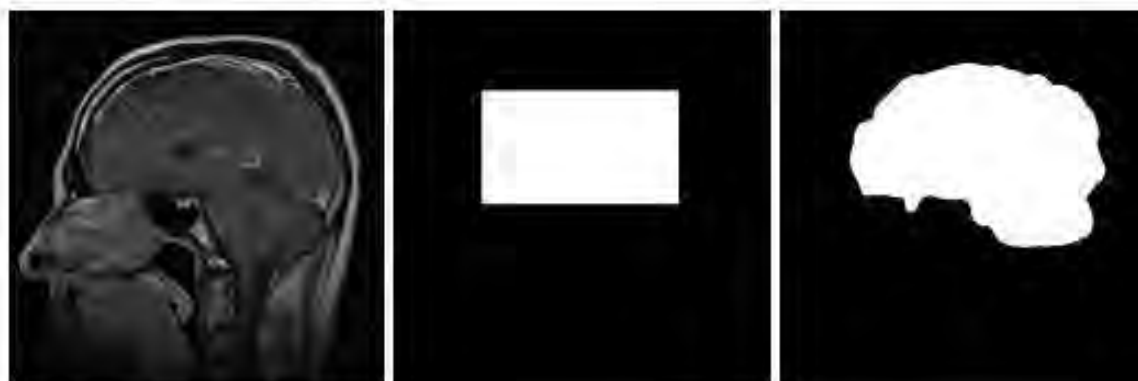


Входное изображение

Маска

Результат

Сегментация печени с параметрами $T = 180, \epsilon = 45, \alpha = 0.003$



Входное изображение

Маска

Результат

Сегментация мозга с параметрами $T = 45, \epsilon = 30, \alpha = 0.003$

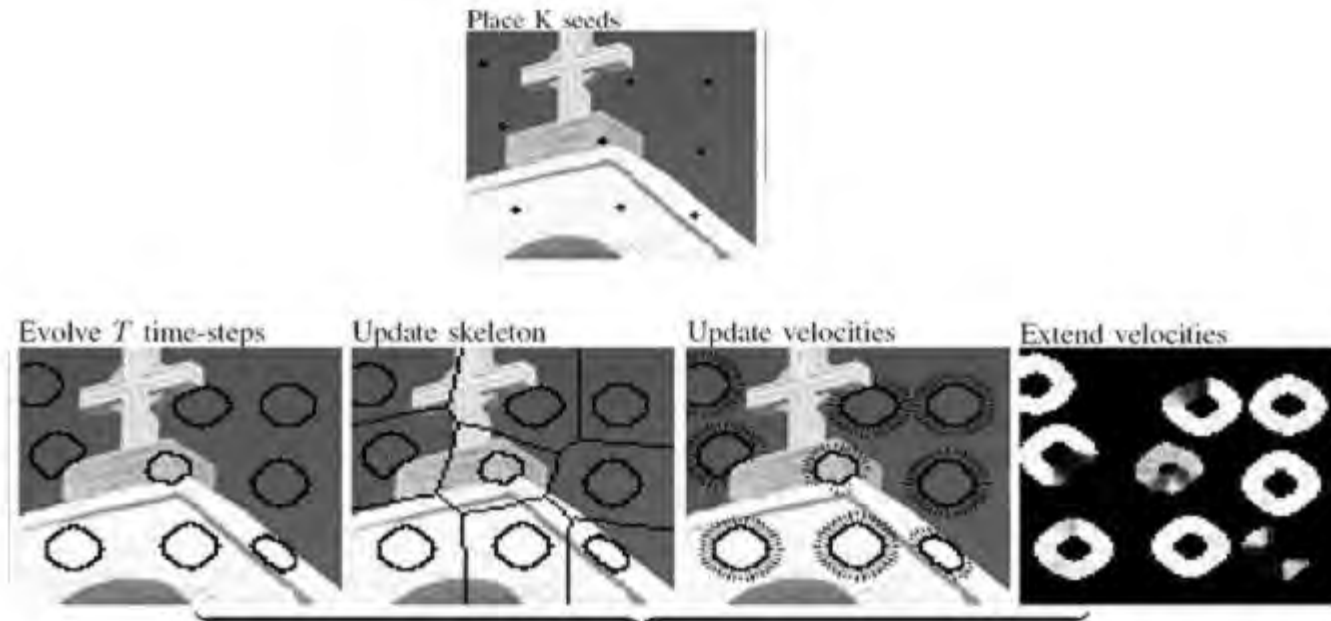
TurboPixel

- Алгоритм, специально нацеленный на сегментацию изображения на суперпиксели
- Даёт суперпиксели примерно одного
- размера, равномерно распределенные по изображению
- Использует подход линий уровня для сегментации
- Сложность $O(N)$, где N – пиксели
- При увеличении количества суперпикселей даже ускоряется



Alex Levinshtein, et. al, TurboPixels: Fast Superpixels Using Geometric Flows, PAMI 2009

Схема алгоритма



- Идея: скорость движения «контура» зависит от градиента, близости к предполагаемой границе региона и т.д.
- Благодаря этому суперпиксели «тормозятся» у краёв изображения и делят его на фрагменты похожего размера



SLIC Superpixel



Images segmented using SLIC into superpixels of size 64, 256, 1024 pixels (approximately).

Algorithm 1 SLIC superpixel segmentation

/ Initialization */*

Initialize cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ by sampling pixels at regular grid steps S .

Move cluster centers to the lowest gradient position in a 3×3 neighborhood.

Set label $l(i) = -1$ for each pixel i .

Set distance $d(i) = \infty$ for each pixel i .

repeat

/ Assignment */*

for each cluster center C_k **do**

for each pixel i in a $2S \times 2S$ region around C_k **do**

 Compute the distance D between C_k and i .

if $D < d(i)$ **then**

 set $d(i) = D$

 set $l(i) = k$

end if

end for

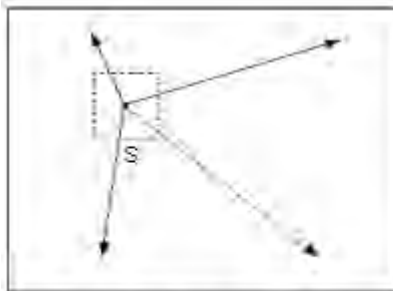
end for

/ Update */*

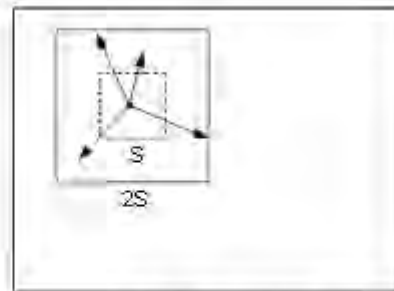
 Compute new cluster centers.

 Compute residual error E .

until $E \leq \text{threshold}$



(a) standard k -means searches the entire image

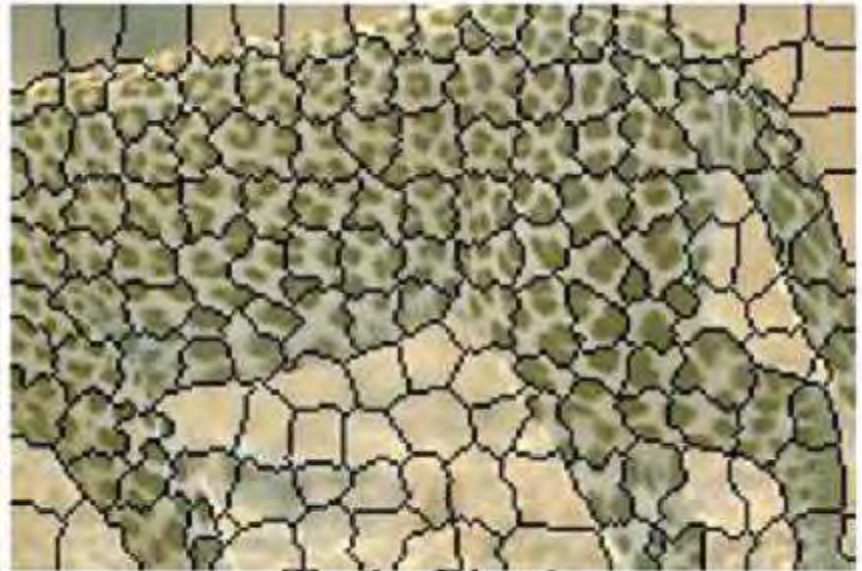


(b) SLIC searches a limited region










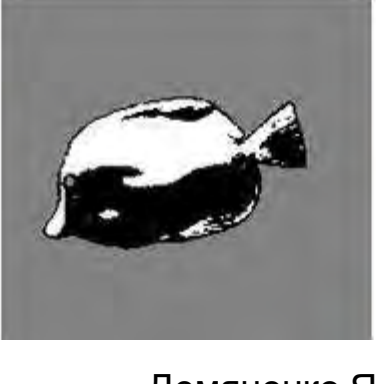


Сравнение



Neuts

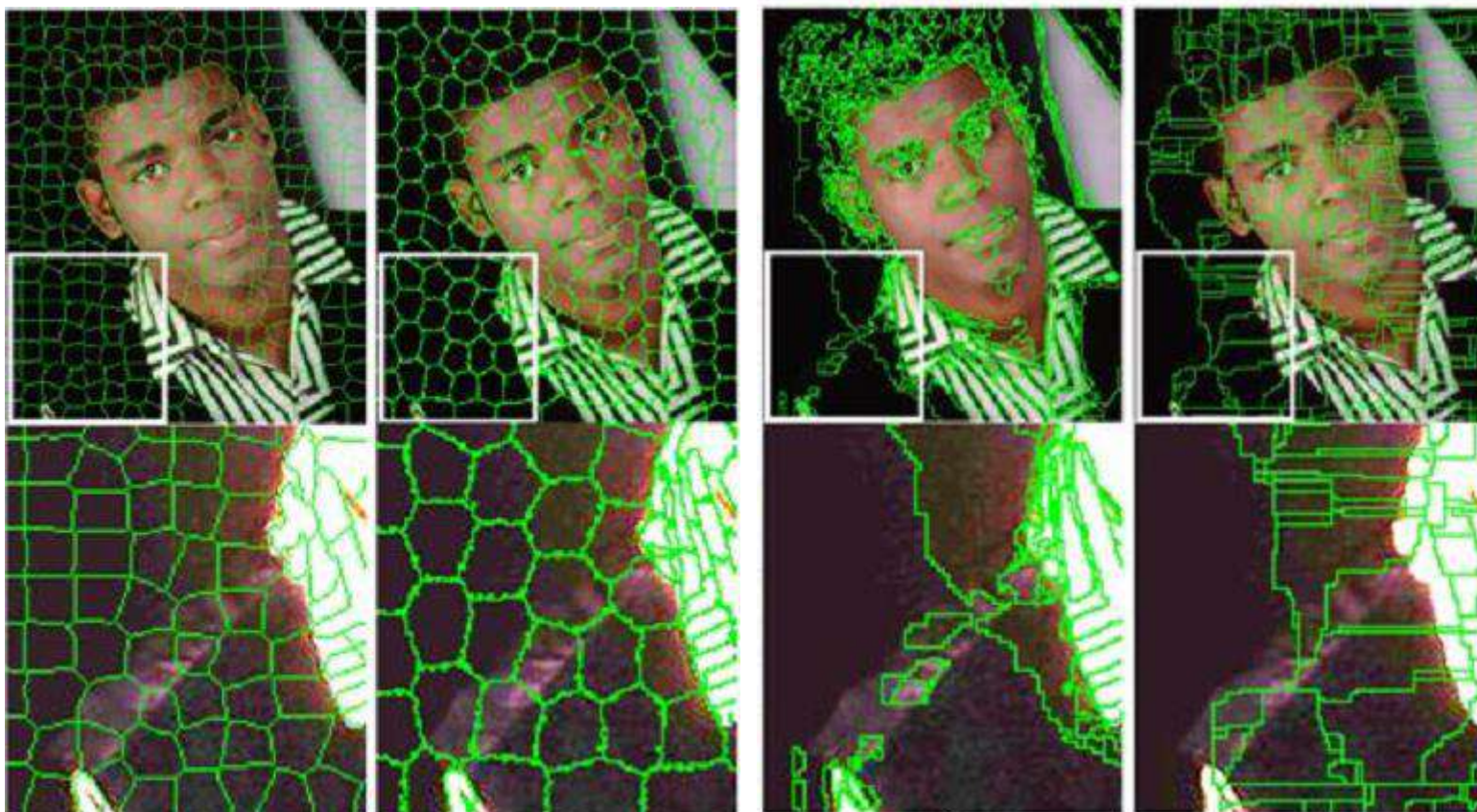


TurboPixels

Исходное изображение	Метод выращивания регионов	Метод нормальных разрезов	Метод водораздела
			
			
			

Исходное изображение	Метод выращивания регионов	Метод нормальных разрезов	Метод водораздела
			
			

Сравнение



TurboPixel

NCuts

Mean-shift

Watershed

Резюме лекции

- Задача сегментации – «разбора изображения» в общем случае является целью распознавания изображений
- Сегментация изображения на области по набору признаков – эффективная предобработка для решения других основных задач
- Понижение размерности задач (работа с областями, а не отдельными пикселями)
- Хорошая сегментация должна учитывать несколько признаков в совокупности
- Чаще всего используются в зависимости от задачи:
 - Методы на графах
 - Mean shift
 - Turbopixels

Литература

1. R. M. Haralick, L. G. Shapiro, "Image Segmentation Techniques," Computer Vision, Graphics, and Image Processing, Vol 29, No 1, 1985
2. K. S. Fu and J. k. Mui, "A Survey on Image Segmentation," Pattern Recognition, Vol. 13, 1981
3. N. R. Pal and S. K. Pal, "A Review on Image Segmentation Techniques," Pattern Recognition, Vol. 26, No 9, 1993
4. R. Jain, R. Kasturi, and B. G. Schunck, Machine Vision, 1995
5. L. Lucchese and S.K. Mitra "Color Image Segmentation: A State-of-the-Art Survey", 2001
6. S.-T. Bow, Pattern Recognition and Image Preprocessing, Marcel Dekker, Inc., New York, NY, 1992.
7. G. R. Cross and A. K. Jain, "Markov Random Field Texture Models," IEEE Trans. on Pattern Analysis and Machine Intelligence, 1983
8. S. German and D. German, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," IEEE Trans. on Pattern Analysis and Machine Intelligence, 1984
9. H. Digabel and C. Lantujoul, "Iterative Algorithms," Proc. of the 2nd European Symp. on Quantitative Analysis of Microstructures in Material Science, Biology and Medicine, 1977

Литература

10. M. Celenk, "Hierarchical Color Clustering for Segmentation of Textured Images," Proc. of the 29th Southeastern Symposium on System Theory, 1997
11. S. Ji and H. W. Park, "Image Segmentation of Color Image Based on Region Coherency," Proc. of ICIP'98
12. L. Shafarenko, M. Petrov, and J. Kittler, "Automatic Watershed Segmentation of Randomly Textured Color Images," IEEE Trans. on Image Processing, 1997
13. M. Barni, S. Rossi, and A. Mecocci, "A Fuzzy Expert System for Low Level Image Segmentation," EUSIPCO-96
14. A. Tremeau and N. Borel, "A Region Growing and Merging Algorithm to Color Segmentation," Pattern Recognition, 1997
15. Y. Kanai, "Image Segmentation Using Intensity and Color Information," SPIE – Visual Communications and Image Processing'98
16. B. Cramariuc, M. Gabbouj, and J. Astola, "Clustering Based Region Growing Algorithm for Color Image Segmentation," Int. Conf. on Digital Signal Processing, 1997
17. Y. Deng, B. S. Manjunath, and H. Shin, "Color Image Segmentation", CVPR 1999

Литература

18. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics David Martin Charless Fowlkes Doron Tal Jitendra Malik Department of Electrical Engineering and Computer Sciences
19. M. Jacob, M. Unser, "Design of Steerable Filters for Feature Detection Using Canny-Like Criteria," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 8, pp. 1007-1019
20. Image Segmentation By Learning Approach Horacio Andrés Legal-Ayala, Jacques Facon, 2003
21. Bayesian learning, global competition and unsupervised image Segmentation Guodong Guo and Songde Ma
22. Edge Detection Using steerable Filters and CNN, Atilla Ozmen and Emir Tufan Akman, 2002
23. Normalized Cuts and Image Segmentation - J. Shi, J. Malik (1997) University of California at Berkeley
24. Efficient Spatiotemporal Grouping Using the Nystrom Method (CVPR 2001) - Charless Fowlkes , Serge Belongie, Jitendra Malik
25. Efficient Graph Cuts for Unsupervised Image Segmentation using Probabilistic Sampling and SVD-based Approximation (ICCV 2003) - J. Keuchel, C. Schnorr, University of Mannheim, Germany

Литература

26. Image Segmentation by Nested Cuts (2000) - Olga Veksler, NEC Research Institute
27. A New Graph-Theoretic Approach to Clustering and Segmentation (CVPR03)
28. Fast Multiscale Image Segmentation (CVPR 2000) - Eitan Sharon, Achi Brandt Ronen Basriy, Dept. of Applied Math The Weizmann Inst. of Science, Israel
29. Segmentation and Boundary Detection Using Multiscale Intensity Measurements (CVPR 2001) - Eitan Sharon, Achi Brandt_, Ronen Basri
30. Texture Segmentation by Multiscale Aggregation of Filter Responses and Shape Elements (ICCV 2003) - Meirav Galun, Eitan Sharon, Ronen Basri, Achi Brandt
31. Learning Affinity Functions for Image Segmentation: Combining Patch-based and Gradient-based Approaches, Charless Fowlkes, David Martin, Jitendra Malik, 2003
32. Boosting Margin Based Distance Functions for Clustering, Tomer Hertz, Aharon Bar-Hillel , 2004
33. Learning to Detect Natural Image Boundaries Using Brightness and Texture, Charless Fowlkes, David Martin, Jitendra Malik, 2003

Литература

34. How Much Does Globalization Help Segmentation, Charles Fowlkes and Jitendra Malik, 2003
35. Bishop, C. M. Neural Networks for Pattern Recognition. Oxford, England: Oxford University Press, 1995

Интерактивная сегментация изображений

Лекция 9

Задача фотомонтажа

- Манипулирование отдельными объектами сцены с высоким визуальным качеством
 - Добавление нового объекта
 - Удаление объекта
 - Восстановление фона
- Композиция одного изображения из нескольких



Формирование изображения

- Мы представляем изображение как композицию объекта и фона, причём какие-то пиксели изображения есть результат смешения объекта и фона (следствие дискретизации)

$$I = \alpha * F + (1 - \alpha) * B$$



- Карта непрозрачности (opacity map) α из $[0, 1]$, где 0 - фон, 1 – непрозрачный объект
- Она же альфа-канал

Выделение объекта

- Задача маттирования (matting) объекта
 - Выделить интересующий пользователя объект на изображении
- Формально:
 - Для каждого пикселя изображения определить значения прозрачности α из $[0, 1]$, где 0 - фон, 1 – непрозрачный объект
 - Для всех пикселей с $\alpha > 0$ определить f – цвет пикселя объекта



Подход к решению

- С очевидностью, задача -- недоопределенная
 - Сцена состоит из множества объектов, какой из них нужен пользователю, неизвестно
- Поэтому в общем виде автоматически она решается пока только съемкой в специальных условиях

Chroma keying & Luma keying

- Chroma keying
 - Объект снимается на однородном фоне определенного цвета
 - Широко применяется в кино и на телевидении
 - Работает в real-time
 - Возможно извлечение теней, бликов, прозрачных элементов
-
- Luma keying
 - Альфа-канал получается путем применения фильтров яркости и контрастности к изображению, преобразованному к градациям серого

Chroma keying – пример

2003



Chroma keying

- фильмы, которые были полностью сняты на Chroma keying («Жизнь Пи», например)
- на 90% снятые на Chroma keying («Город грехов», «Аватар», «Алиса в стране чудес» и др)

Luma keying - пример

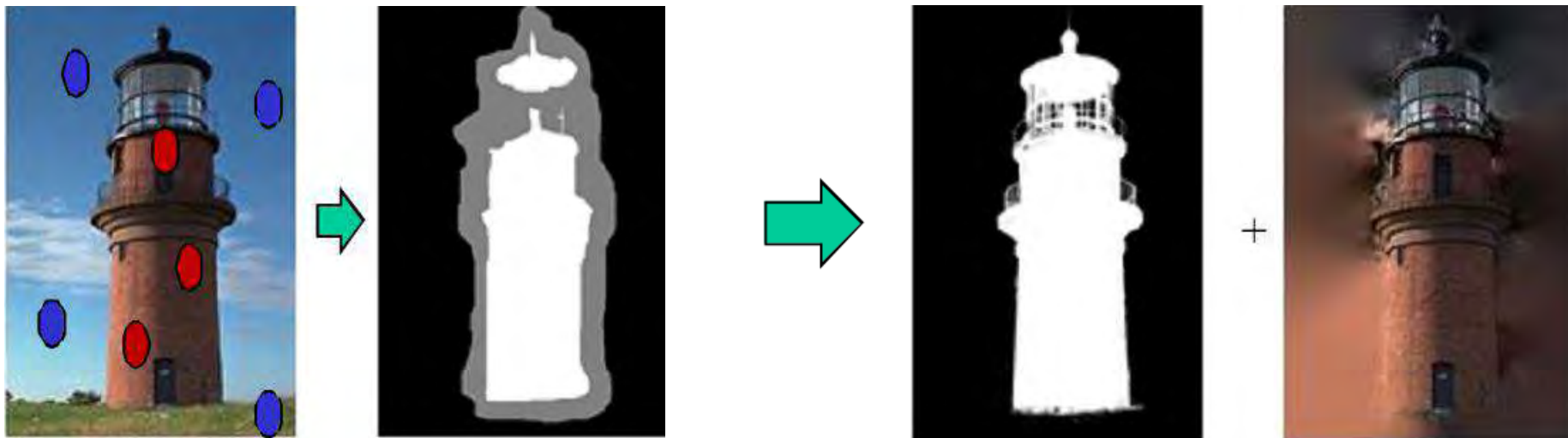


Luma keying - пример



Подход к решению

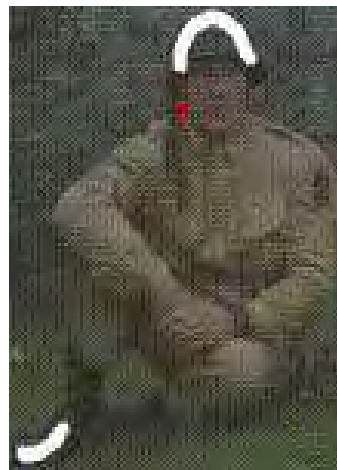
- Для изображений, полученных в обычных условиях, задача раскладывается в несколько более простых задач
 - Интерактивная «жесткая» сегментация объекта
 - Пользовательский ввод α из $\{0,1\}$
 - Мягкая сегментация объекта (собственно *маттирование*)
 - Результат жесткой сегментации используется как начальное приближение



Интерактивная сегментация – постановка задачи

- Рассматривается задача разбиения на 2 области - объект и фон
- На вход алгоритм получает:
 - исходное изображение
 - дополнительную информацию от пользователя:
 - ограничение на то, что некоторые конкретные пиксели обязательно должны принадлежать объекту (фону)
 - ограничивающий прямоугольник вокруг объекта
 - примерную границу объекта
- В процессе работы алгоритма пользователь может уточнять или дополнять входные данные.

Пример интерактивной сегментации изображения



Одна из основных проблем - отсутствие единой метрики качества
Субъективное сравнение результатов
Разный пользовательский ввод (объём?)

Тестирование методов на общей базе изображений

Например, Berkeley Segmentation Dataset насчитывает более 1000 изображений, отсегментированных вручную 30 разными людьми.

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

Алгоритмы интерактивной сегментации

Время до 2001

- **Magic wand**
- Intelligent scissors
- Intelligent Paint

После 2001

- Interactive graph cuts
- Lazy Snapping
- GrabCut
- Progressive Cut
- GrowCut

Magic Wand (<волшебная палочка>)

- ◆ Идея:

- Пользователь задает пиксель, запоминается его цвет C_1
- Все пиксели, цвет которых $\|C - C_1\| < \Theta$ - относятся к выделяемому региону



Эффективность Magic Wand



8 кликов мыши,



более 20 кликов,
причем некоторые части
олени остались
неотсегментированными

Алгоритмы интерактивной сегментации

Время до 2001

- Magic wand
- **Intelligent scissors**
- Intelligent Paint

После 2001

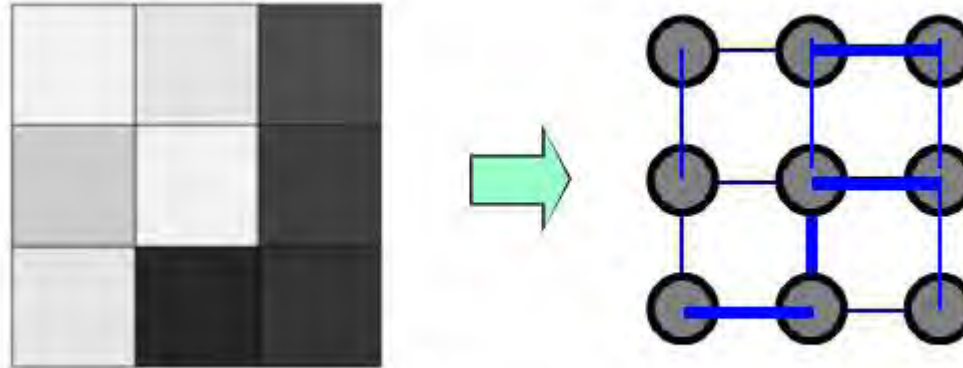
- Interactive graph cuts
- Lazy Snapping
- GrabCut
- Progressive Cut
- GrowCut

Intelligent Scissors (Умные ножницы) [3] 1996



Eric N. Mortensen, William A. Barrett, Intelligent scissors for image composition, Siggraph 1995

Принцип действия Intelligent Scissors



1. всё изображение трактуется как взвешенный неориентированный граф, каждая вершина которого соответствует пикселю изображения
2. вершины, соответствующие соседним пикселям связываются ребрами
3. на ребрах данного графа определяется весовая функция
4. значение этой функции мало на ребрах, соответствующих потенциальной границе на изображении.

Что учитывает Intelligent Scissors

$$l(p, q) = w_z \cdot f_z(q) + w_G \cdot f_G(q) + w_D \cdot f_D(p, q)$$

$f_z(q)$ учитывает локальные максимумы градиента

$f_G(q)$ учитывает силу градиента (чем он больше, тем f меньше),

$f_D(p, q)$ стимулирует более гладкие границы

Действия пользователя для Intelligent Scissors



- Пользователь указывает 2 точки на границе объекта.
- Граница объекта находится с помощью алгоритма нахождения пути минимальной стоимости в графе
- Алгоритм Дейкстры на основе динамического программирования

Плюсы и минусы Intelligent Scissors

- (+) Существуют очень быстрые реализации алгоритма
- (--) Плохо работает в сильно текстурированных областях (существует множество альтернативных минимальных путей) .

Модификация Intelligent Scissors

- Использование предварительной пересегментации [4]
- Или использование суперпикселей

Алгоритмы интерактивной сегментации

Время до 2001

- Magic wand
- Intelligent scissors
- **Intelligent Paint**

После 2001

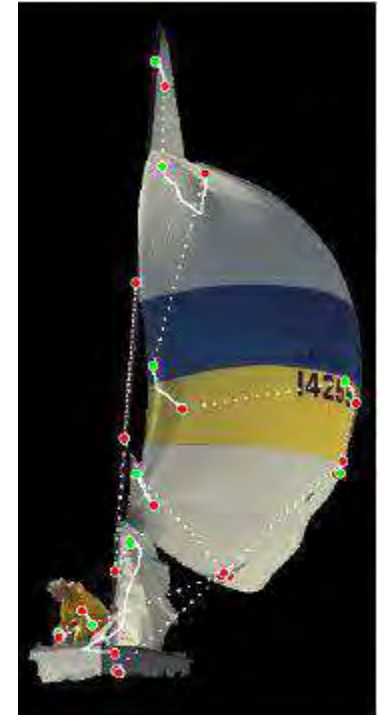
- Interactive graph cuts
- Lazy Snapping
- GrabCut
- Progressive Cut
- GrowCut

Intelligent Paint [5]



- ◆ Идея:
 - Предварительная иерархическая сегментация изображения
 - Анализ «мазков» пользователя для объединения регионов предварительной сегментации

Результаты работы Intelligent Paint



Коммерческие решения (не рассматриваем медицину)

- Magic Wand
(цветовая статистика)



- Magnetic Lasso
(Live Wire, Intelligent scissors)



Алгоритмы интерактивной сегментации

Время до 2001

- Magic wand
- Intelligent scissors
- Intelligent Paint

После 2001

- **Interactive graph cuts**
- Lazy Snapping
- GrabCut
- Progressive Cut
- GrowCut

Прорыв в данной области

- произошёл в 2000 г.
- с изобретением Юрием Бойковым и Мари-Пьер Джолли алгоритма GraphCut
- GraphCut стал де-факто эталонным алгоритмом интерактивной сегментации

Действия пользователя GraphCut

- Пользователь указывает жесткие ограничения на сегментацию – пиксели, принадлежащие объекту (О) и пиксели, принадлежащие фону (В)
 - семена объекта и фона



Yuri Boykov and Marie-Pierre Jolly. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D images. In International Conference on Computer Vision, (ICCV), vol. I, pp. 105-112, 2001.

GraphCut [6]

- всё изображение – граф
- вершины-пиксели + 2 терминальные вершины (исток и сток)
- вес ребра

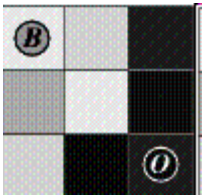
$$B_{p,q} = \exp\left(-\frac{\|C_p - C_q\|^2}{2\sigma^2}\right) * \frac{1}{dist(p,q)}$$

- Формулировка сегментации как минимизации энергии через разрез графа

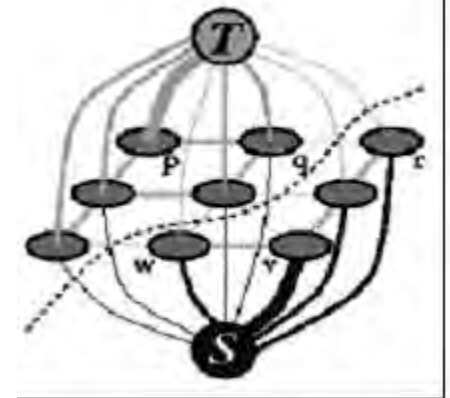
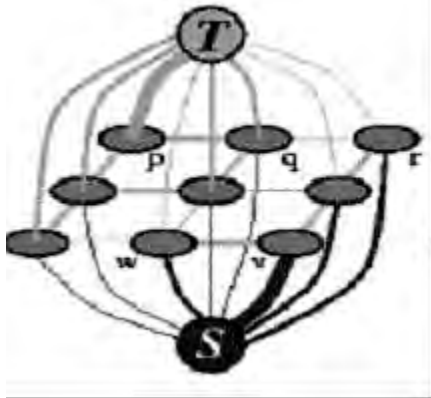
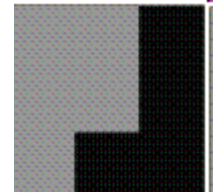
Иллюстрация алгоритма GraphCut

исходное изображение,
O, B - семена объекта и фона

финальная сегментация



Для нахождения самого минимального разреза авторы GraphCut-а разработали новый алгоритм, основанный на алгоритме Форда-Фалкерсона [7].

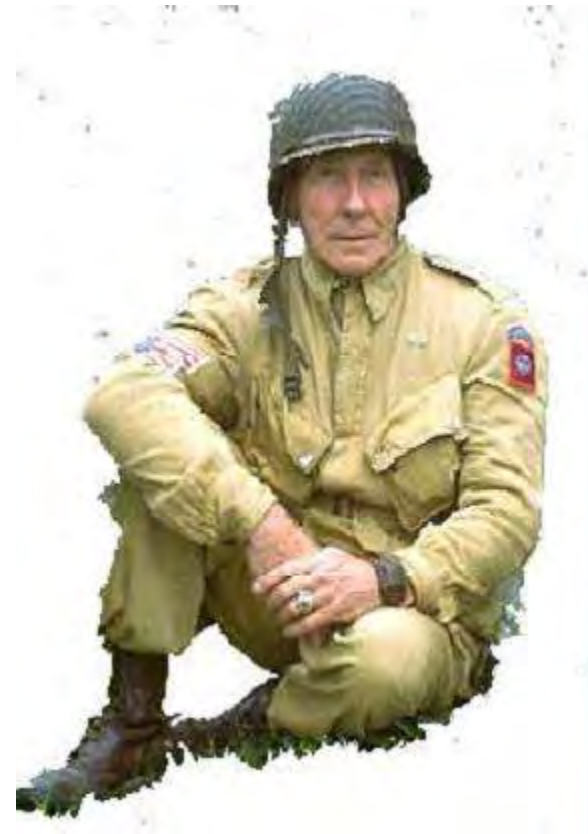


Улучшение алгоритма

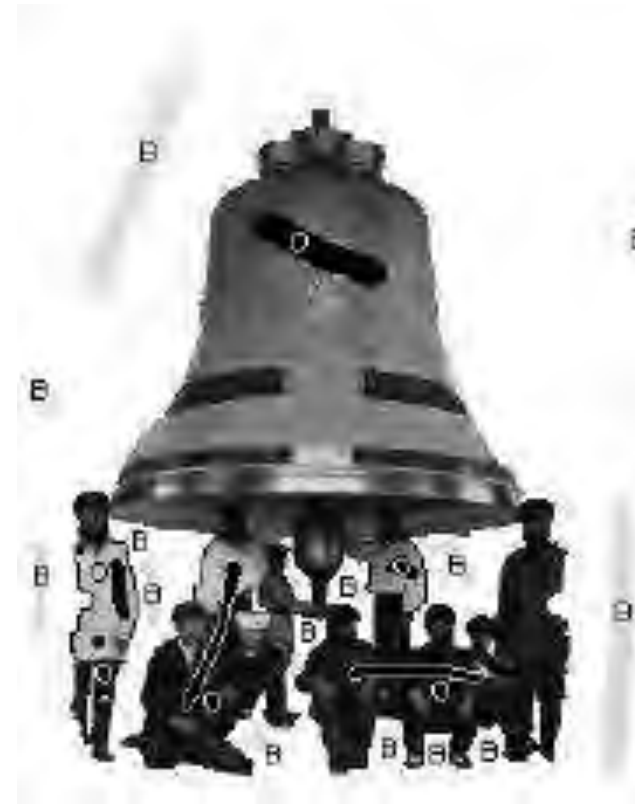
ребро	Вес	Для
$\{p, q\}$	$B_{\{p, q\}}$	$\{p, q\} \in N$
$\{p, S\}$	$\lambda * R_p(\text{"bkg"})$	$p \in P, p \notin O \cup B$
	∞	$p \in O$
	0	$p \in B$
$\{p, T\}$	$\lambda * R_p(\text{"obj"})$	$p \in P, p \notin O \cup B$
	0	$p \in O$
	∞	$p \in B$

цветовая
статистика
объекта и фона,
собранный на
основе
отмеченных
пользователем
семян объекта и
фона

Алгоритм GraphCut. Примеры



Пример работы алгоритма GraphCut



Алгоритмы интерактивной сегментации

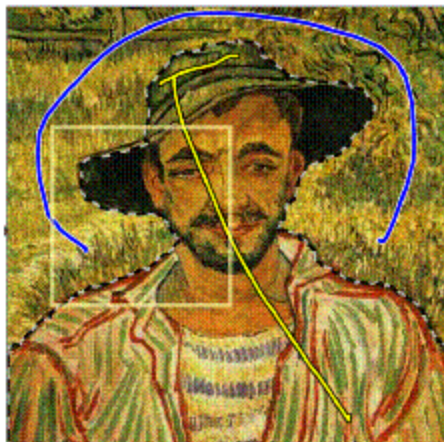
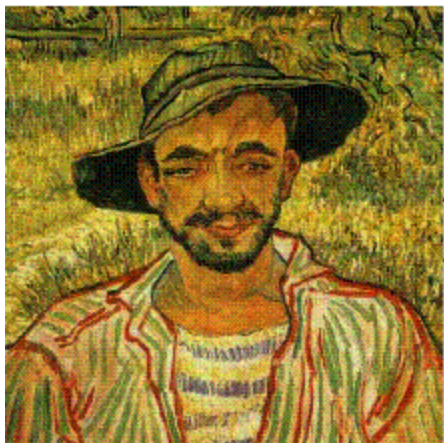
Время до 2001

- Magic wand
- Intelligent scissors
- Intelligent Paint

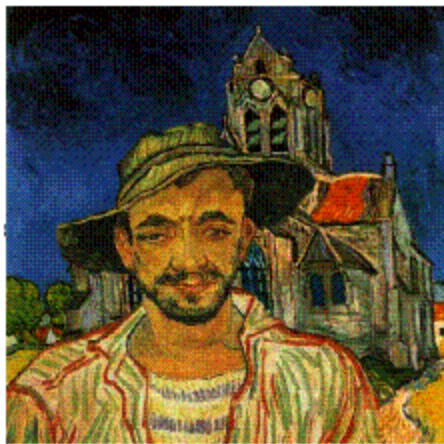
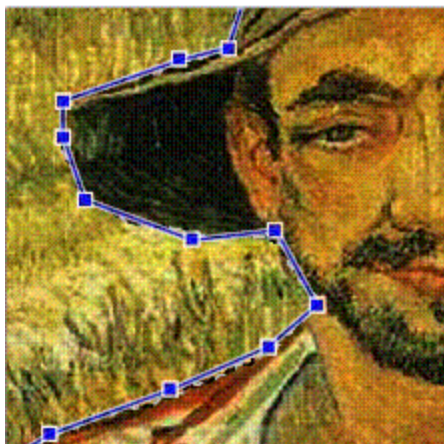
После 2001

- Interactive graph cuts
- **Lazy Snapping**
- GrabCut
- Progressive Cut
- GrowCut

Lazy Snapping [11]



(а) - исходное изображение;
(б) - пользователь указывает всего 3 линии (2 внутри объекта, 1 снаружи), все линии рисуются далеко от границы

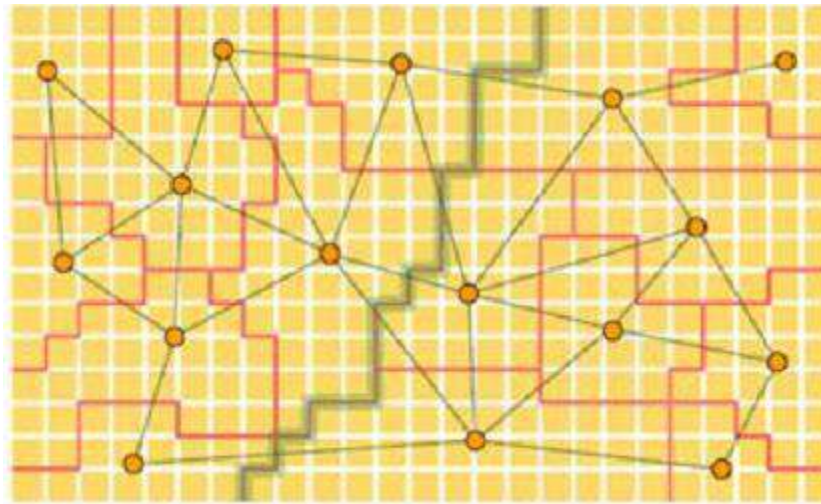


(в) - полигональное редактирование границы объекта;
(г) - композиция вырезанного изображения и ещё одной картины Ван Гога.

То же, что и Interactive GraphCuts, но на карте суперпикселей

Lazy Snapping - особенности

- Карта суперпикселей строится алгоритмом watershed



- В 10 раз быстрее чем на пикселях при сравнимом визуальном качестве
- Учитывается цвет фона и объектов.
- Выборка из размеченных пользователем областей кластеризуется k-средними на 64 кластера, вероятности считаются как отношение близости к объекту/фону

Lazy Snapping - редактирование границы

- получившаяся граница объекта преобразуется в полигон
- для редактирования границы объекта можно:
 - переместить, удалить 1 или несколько вершин полигона, добавить новую вершину;
 - использовать Overriding Brush в неправильно отсегментированной области (начало и конец мазка должны лежать на полигоне)



Результат Lazy Snapping



- Уточнение границ считается как задача поиска оптимального разреза в узкой полосе, отмеченной пользователем

Примеры сегментации с помощью Lazy Snapping



Алгоритмы интерактивной сегментации

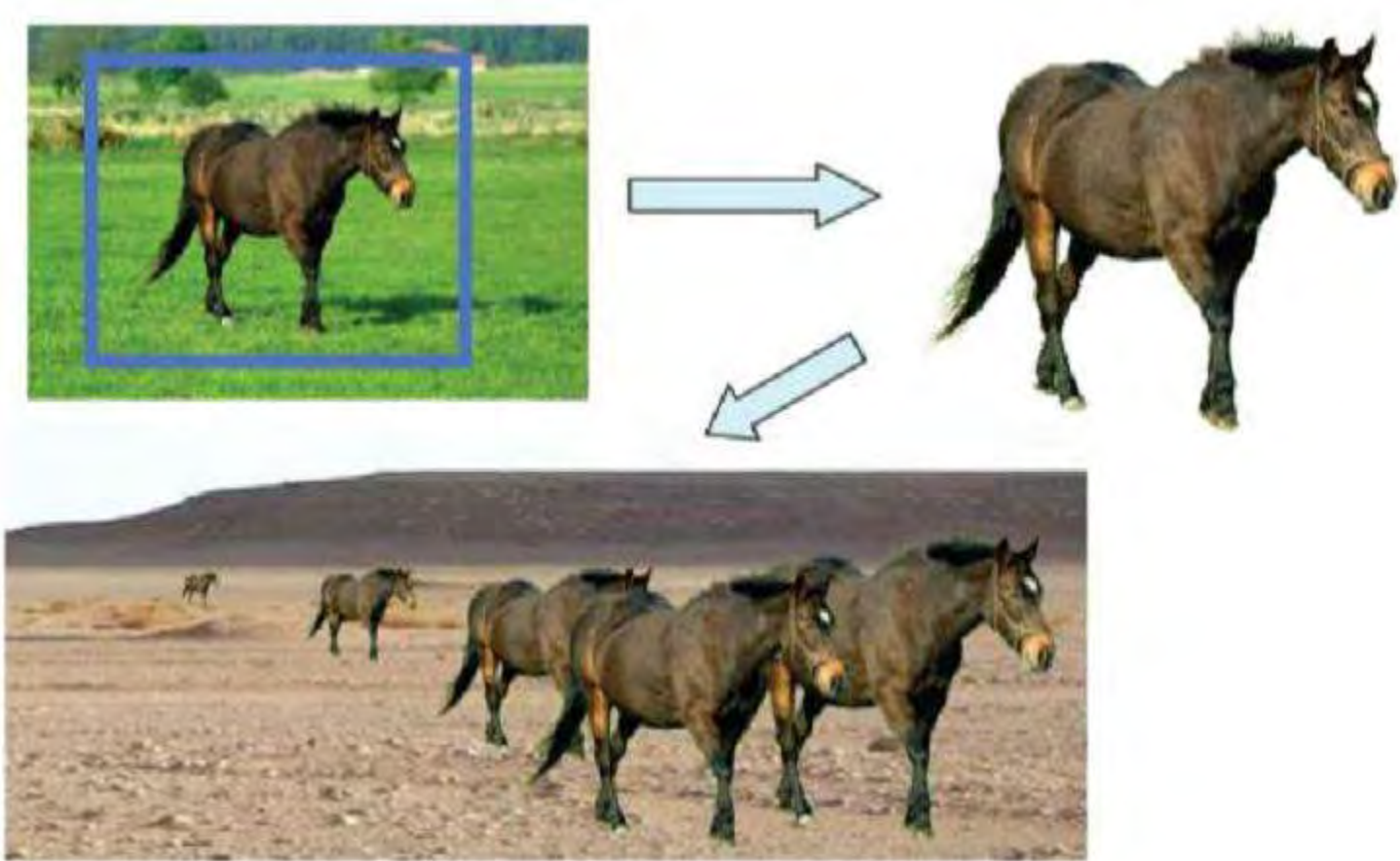
Время до 2001

- Magic wand
- Intelligent scissors
- Intelligent Paint

После 2001

- Interactive graph cuts
- Lazy Snapping
- **GrabCut**
- Progressive Cut
- GrowCut

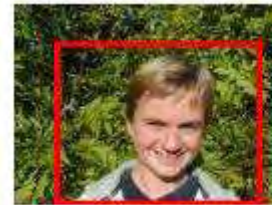
GrabCut [9]



C. Rother, V. Kolmogorov, and A. Blake. Grabcut – interactive foreground extraction using iterated graph cuts. Proc. ACM Siggraph, 2004.

Итеративная схема сегментации GrabCut :

- исходя из цветового распределения внутри и снаружи ограничивающего прямоугольника, строится первая цветовая статистика объекта и фона
- сегментация GraphCut-ом, использующим цветовую статистику
- уточнение цветовой статистики. Граф, в котором ищется минимальный разрез, перевзвешивается





Инициализация пользователя



К-средних для
обучения статистик
цветов



Graph cuts для
сегментации

- Цвет фона и объекта моделируется смесью гауссиан (обычно 5-8 компонент)
- Каждый пиксель объекта/фона сопоставляется компоненте смеси
- Затем уточняются параметры смеси

Slide by C. Rother

Примеры



Slide by C. Rother

Сложные примеры

Camouflage &
Low Contrast

Initial
Rectangle



Initial
Result



Fine structure



No telepathy



Slide by C. Rother

Сравнение

Magic Wand
(198?)

User
Input



Result



Regions

Intelligent Scissors
Mortensen and Barrett (1995)



Boundary

GrabCut



Regions & Boundary

Slide by C. Rother

Сравнение более поздних

Boykov and Jolly (2001)

User
Input

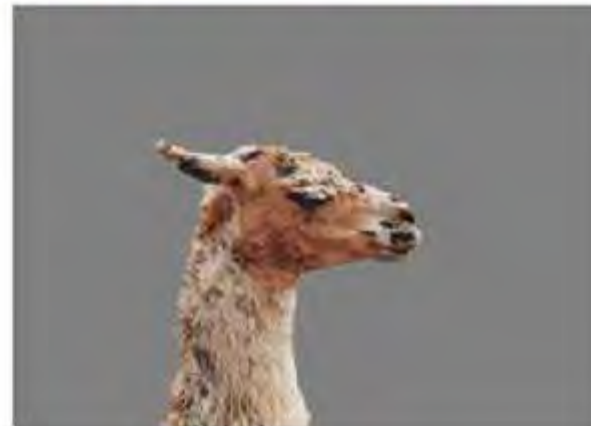


Result



Error Rate: 0.72%

GrabCut



Error Rate: 0.72%

Slide by C. Rother



Magic Wand
(198?)

Intelligent Scissors
Mortensen and
Barrett (1995)

Graph Cuts
Boykov and
Jolly (2001)

LazySnapping
Li et al. (2004)

GrabCut
Rother et al.
(2004)

Slide by C. Rother

GrabCut [9]

- GrabCut запатентован компанией Microsoft
- реализован в одной из рабочих версий графического редактора Microsoft Expression [10] (версия V4.0d)

Алгоритмы интерактивной сегментации

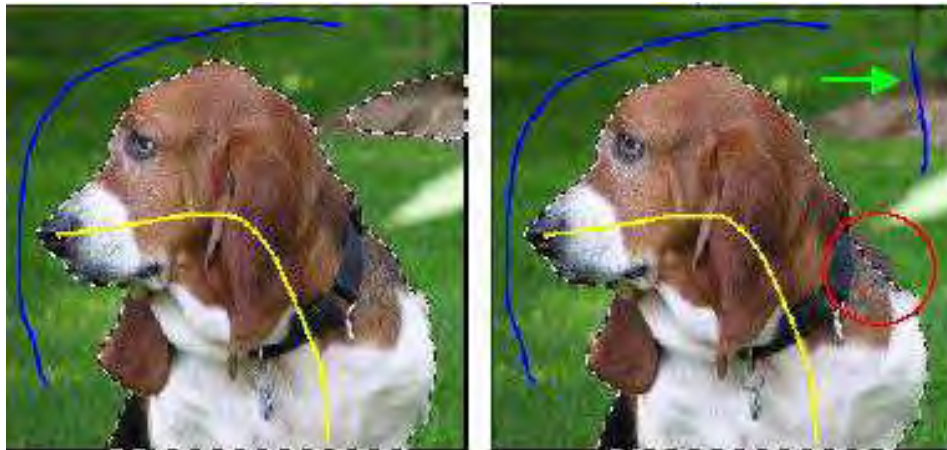
Время до 2001

- Magic wand
- Intelligent scissors
- Intelligent Paint

После 2001

- Interactive graph cuts
- Lazy Snapping
- GrabCut
- **Progressive Cut**
- GrowCut

Progressive Cut [13]

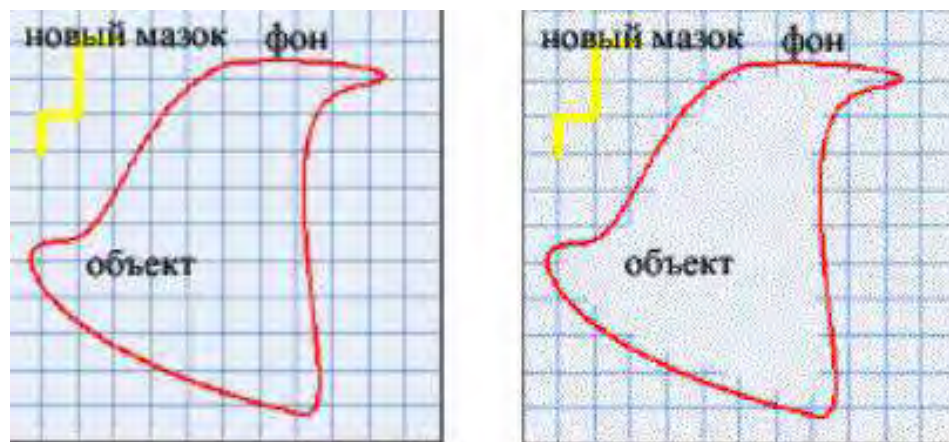


Типичное поведение GraphCut-а

Progressive Cut – намерение пользователя

Progressive Cut пытается понять намерение пользователя, лежащее под тем или иным дополнительным вводом.

1) анализируется, какой тип изменения ожидает пользователь

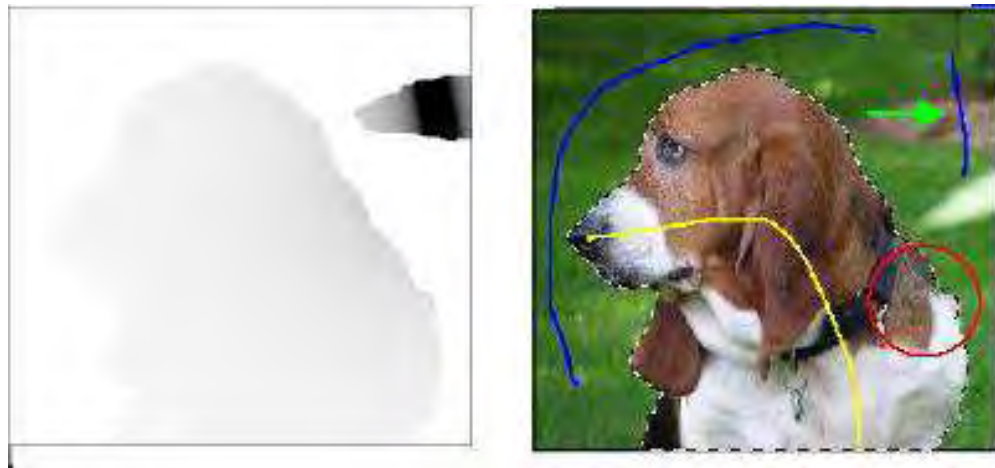


Построение уменьшенного графа на стадии редактирования изначальной сегментации

Progressive Cut - энергия намерения (intention energy)

2) пользователь обычно ожидает изменения в относительно небольшой области.

Это записывается в виде дополнительной энергии намерения (intention energy) , которая обратно пропорциональна расстоянию от данного пикселя до новых семян



Сравнение редактирования сегментации с помощью GraphCut и Progressive Cut



Сравнение редактирования сегментации с помощью GraphCut и Progressive Cut



Алгоритмы интерактивной сегментации

Время до 2001

- Magic wand
- Intelligent scissors
- Intelligent Paint

После 2001

- Interactive graph cuts
- Lazy Snapping
- GrabCut
- Progressive Cut
- **GrowCut**

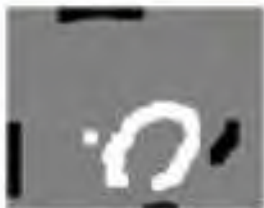
Параметры GrowCut [17]

- основывается на клеточных автоматах
- состояние каждой клетки - (L, Θ, C) ,
где L - метка ('объект', 'фон', 'неизвестно'),
 $\Theta \in [0,1]$ - сила клетки,
 C - вектор признаков клетки (RGB цвет)

Алгоритм GrowCut

- $g(x)$ – монотонно убывающая функция, изменяющаяся в диапазоне $[0, 1]$, которая описывает близость цветов двух пикселей (x обычно является нормой разницы цветов пикселей)
- на каждой итерации каждую клетку P 'атакуют' все её соседи Q
- если $\theta_q * g(\|C_p - C_q\|) > \theta_p$, клетка Q захватывает P

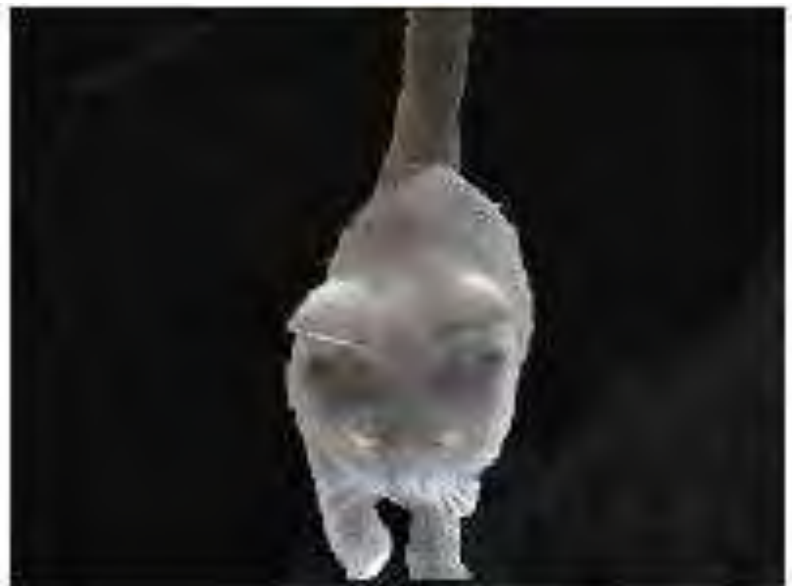
GrowCut. Процесс эволюции клеточного автомата



GrowCut

- Данный алгоритм реализован в виде одноименного плагина к Adobe Photoshop [19]
- легко реализуется на GPU [18]
- Официальный сайт алгоритма
www.growcut.com

Результат GrowCut



Достоинства и недостатки GrowCut

Плюсы

- прост
- данный алгоритм можно обобщить на случай более 2 сегментов
- время реакции алгоритма близко к нулю
- коррекцию можно выполнить по ходу алгоритма

Минусы

- недостаточно быстр для больших фото
- более рваная граница, чем у Graph Cuts



Развитие GrowCut

- В качестве развития данного алгоритма были предложены иерархическая версия, значительно ускоряющая базовый алгоритм, а также версия, налагающая на границы итоговых сегментов дополнительные ограничения на гладкость.

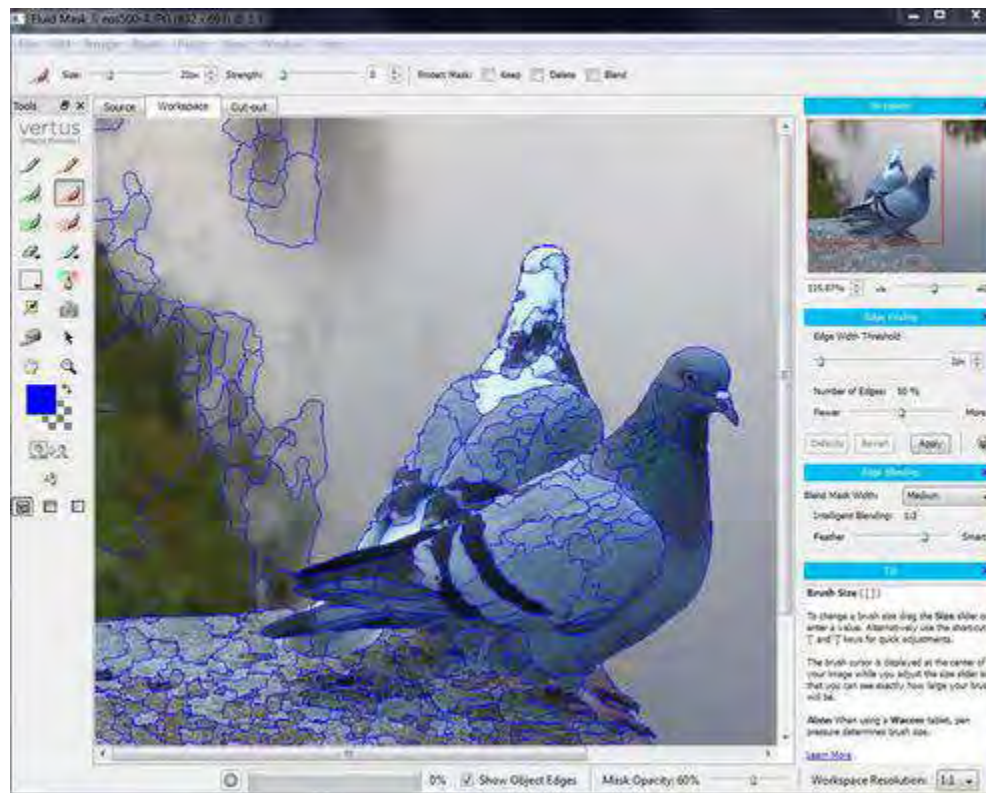
Заключение

- Magic Wand является самым примитивным инструментом сегментации, прост в реализации
- Intelligent Scissors: опорные точки в самой близи от границы объекта, высокая скорость алгоритма
- Intelligent Paint: придуман относительно давно, не реализован в коммерческих программах
- GrowCut: простота реализации, хороший алгоритм, очень маленькое время реакции на дополнительный ввод и высокая интерактивность, некоторая рваность границ, требователен к оперативной памяти

Заключение

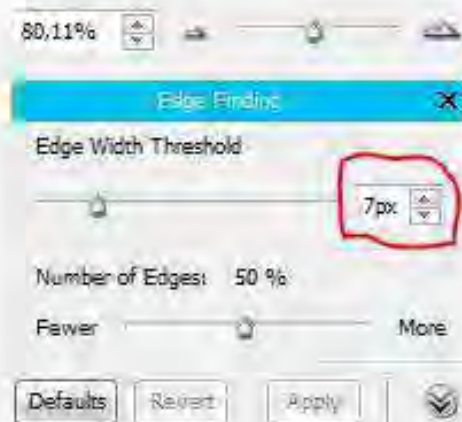
- GraphCut: упрощает действия пользователя, значительное время реакции, требует много оперативной памяти, запатентован
- GrabCut: самый удобный интерфейс (ограничивающий прямоугольник), в случаях сильного перекрытия цветовых распределений фона и объекта, выдаёт неправильный результат, и изображение приходится сегментировать также, как и в обычном GraphCut-е, сложнее для реализации, запатентован.
- Lazy Snapping, сохраняя достоинства GraphCut-а, значительно уменьшает время реакции на дополнительный ввод.
- Progressive Cut: упрощает процесс редактирования сегментации, изменение исходного графа может увеличить время реакции алгоритма

Коммерческий Vertus Fluid mask



Анализ цветowych переходов

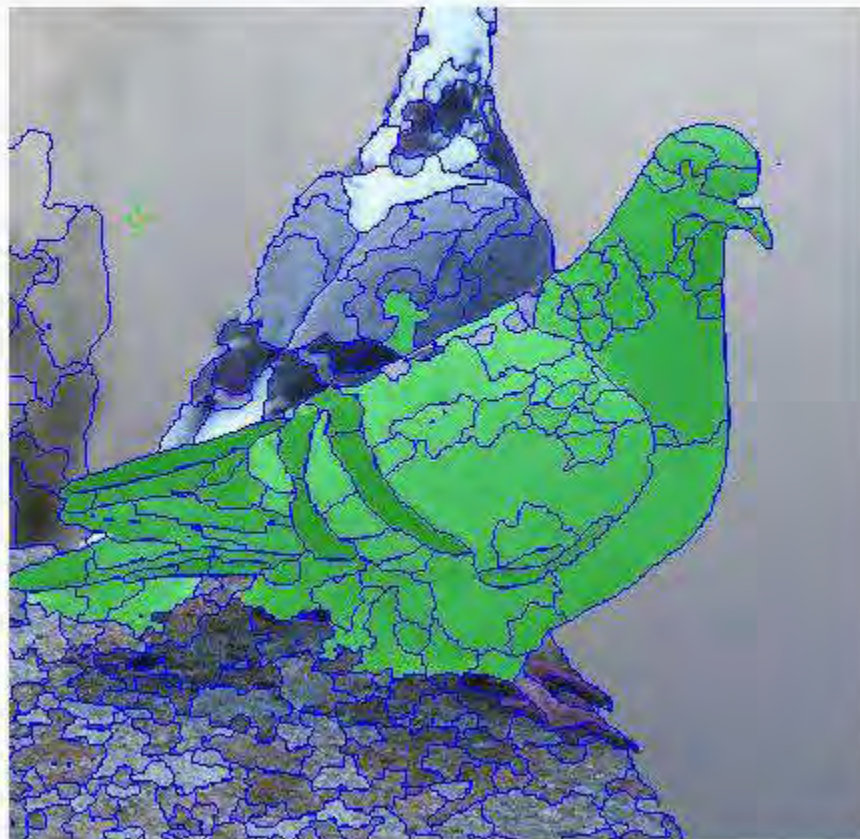
Смысл работы плагина в том, что он анализирует все цветowych переходы и показывает их. Плагин достаточно мощный, и можно исходя из изображения, настроить порог анализа. Если вырезаемый объект сливается с фоном, то лучше поставить минимальный порог, что бы плагин выделил как можно больше переходов(3).



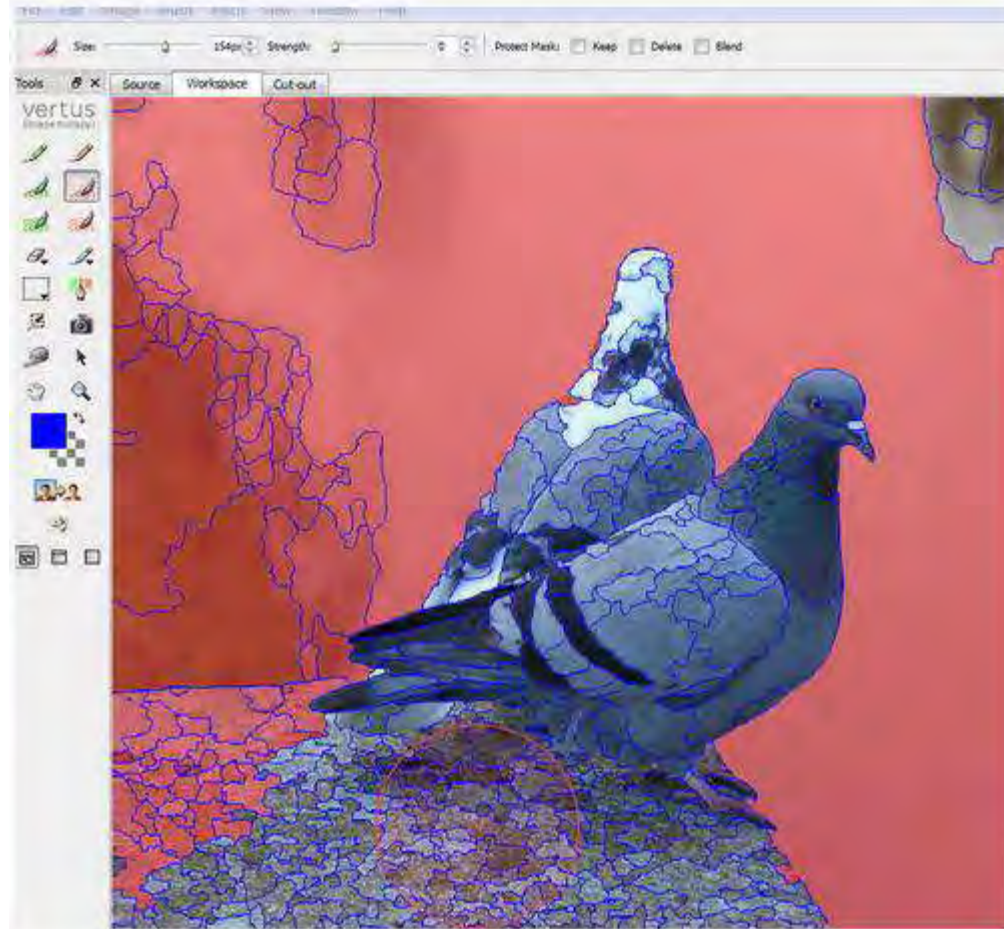
У наших голубей, есть места, которые сливаются и я выбрал маленький порог 2px.

Зелёное – то что нужно

Теперь берём красную кисточку, и закрашиваем всё, что нам не нужно. Можно пойти от обратного и закрасить зелёным, что нужно оставить(4).

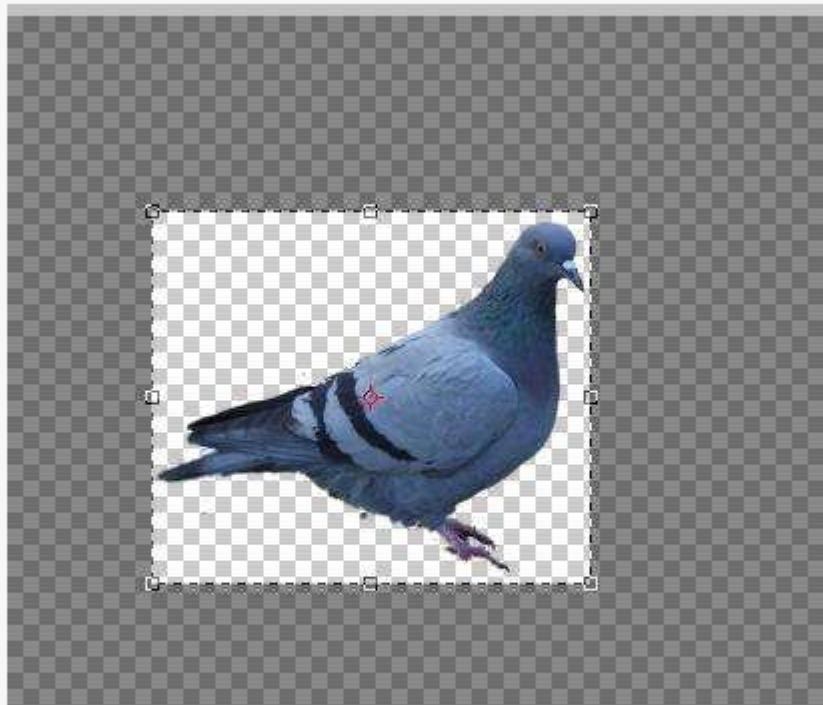


Или красным, то что не нужно



Демяненко Я.М., ЮФУ, 2021

Результат Vertus Fluid mask

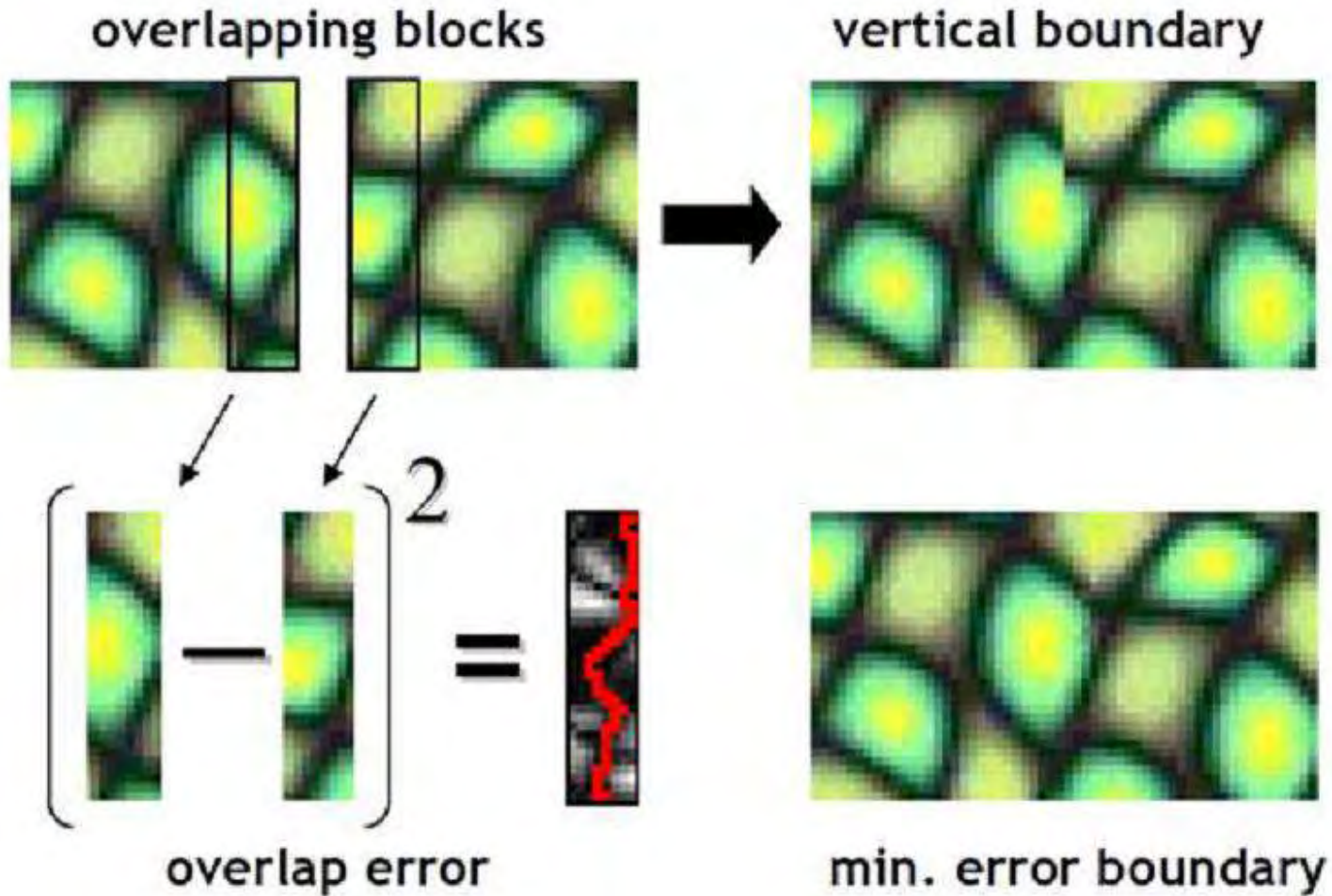


Теперь "ластиком" убираем остатки и "размытием" проходимся по краям(11).

Применение

- Композиция изображений
- Сшивка
- Создание текстур

GraphCut для сшивки



KWATRA, V., SCHODL, A., ESSA, I., TURK, G., AND BOBICK, A. Graphcut textures: image and video synthesis using graph cuts. SIGGRAPH, 2003

GraphCut для сшивки



Создание текстур



- «Накидываем» исходные экземпляры на изображение и сшиваем их с помощью разрезов графов.

- Перспективное изменение (масштабирование) по необходимости

Синтез текстур в жизни

Bush campaign digitally altered TV ad

President Bush's campaign acknowledged Thursday that it had digitally altered a photo that appeared in a national cable television commercial. In the photo, a handful of soldiers were multiplied many times.

This section shows a sampling of the duplication of soldiers.



Литература 1

2. Martin D., Fowlkes C., Tal D., Malik J. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. *Proceedings of 8th International Conference on Computer Vision*. 2001. 2. N. 1. P. 416-423.
3. Mortensen, E., and Barrett, W. 1995. Intelligent scissors for image composition. *Proc. ACM Siggraph*, 191-198.
4. Mortensen, E., and Barrett, W. 1999. Tobogan-based intelligent scissors with a four parameter edge model. In *Proc. IEEE Conf. Computer Vision and Pattern Recog.*, vol. 2, 452-458.
5. Reese, L. 1999 Intelligent Paint: Region-Based Interactive Image Segmentation. *Master's thesis*, Department of Computer Science, Brigham Young University, Provo, UT.
6. Y. Boykov and M.-P. Jolly. Interactive organ segmentation using graph cuts. In *Medical Image Computing and Computer-Assisted Intervention*, pages 276-286, 2000.
7. Boykov, Y., and Kolmogorov, V. 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 9, 1124-1137.
8. Boykov, Y., and Jolly, M.-P. 2001. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *Proc. Of the International Conference on Computer Vision*, vol. 1, 105-112.
9. Rother, C., Kolmogorov, V., and Blake, A. 2004. Grabcut - interactive foreground extraction using iterated graph cuts. *Proc. ACM Siggraph*.

Литература 2

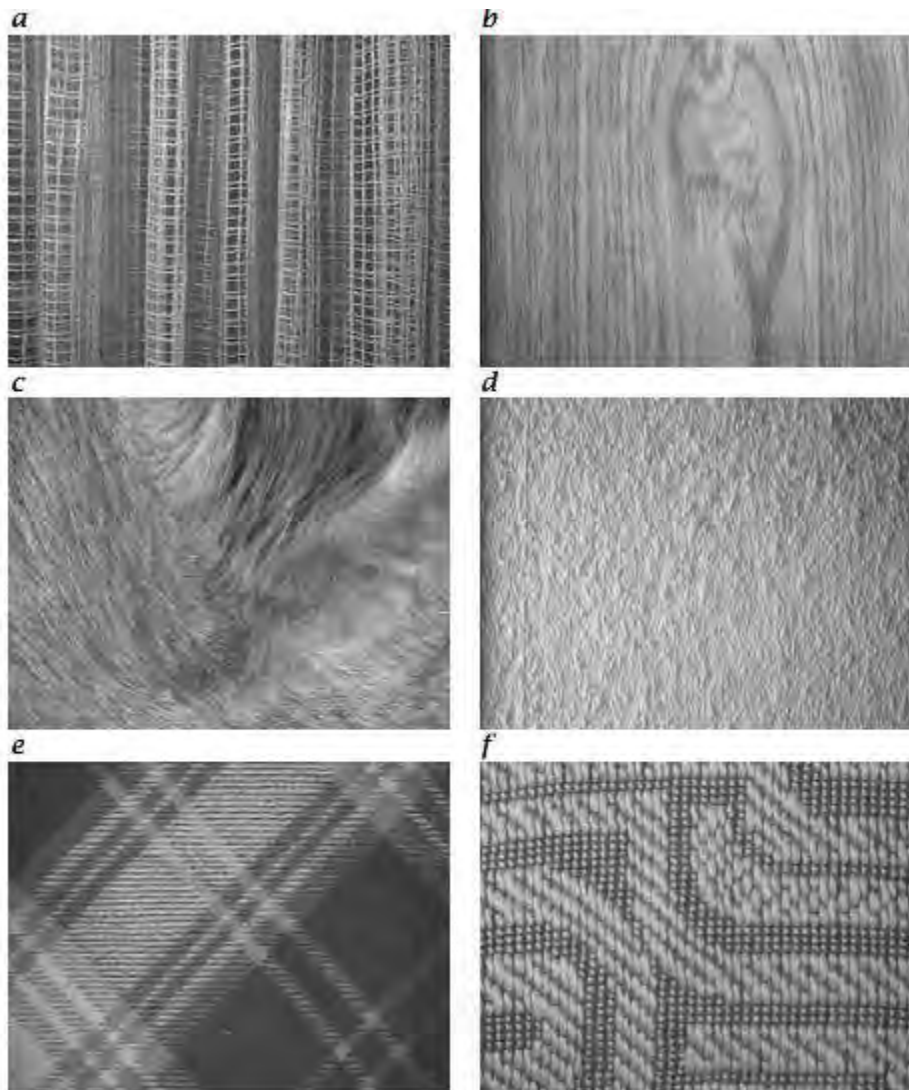
10. http://www.microsoft.com/products/expression/en/graphic_designer/default.mspx
11. Li, Y., Sun, J., Tang, C.-K., and Shum, H.-Y. 2004. Lazy Snapping, *Proc. ACM Siggraph*
12. Vincent, L., and Soille, P. 1991. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-13*, 6 (June), 583-598.
13. Wang, C., Yang, Q., Chen, M., Tang, X., Ye, Z.F. 2006. Progressive Cut. *Proc ACM SIG Multimedia*
14. Grady, L., and Funka-Lea, G. 2004. Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. In *ECCV Workshops CVAMIA and MMBIA*, 230-245.
15. Grady, L., Schiwietz, T., Aharon, S., and Westermann, R. 2005. Random walks for interactive organ segmentation in two and three dimensions: Implementation and validation. In *Proceedings of MICCAI 2005*, Springer, Palm Springs, CA, J. Duncan and G. Gerig, Eds., no. 2 in LNCS 3750, MICCAI Society, 773-780.
16. Grady, L. 2005. Multilabel random walker image segmentation using prior models. In *CVPR (1)*, 763-770.
17. Vezhnevets, V., and Konouchine, V., 2005. "GrowCut"- interactive multi-label N-D image segmentation by cellular automata, *Proc Graphicon*.
18. <http://www.sonycsl.co.jp/person/nielsen/PT/SHADERX5/shaderx5.html>
19. <http://www.growcut.com>
20. Talbot, J.F., and Xu, X., Implementing Grabcut, <http://students.cs.byu.edu/~jtalbot/research/Grabcut.pdf>

Текстуры

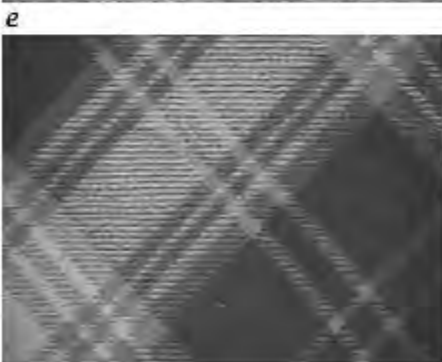
Лекция 10

Демяненко Я.М., ЮФУ, 2021

Что это?



Что отнести к текстуре?



a) занавеска

b) дерево

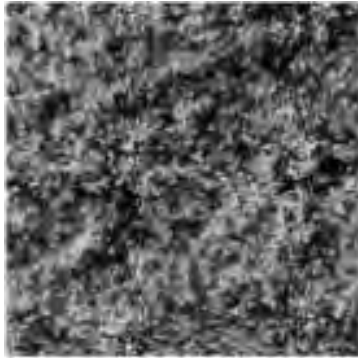
c) собачий мех

d) древесно-стружечная
бумага

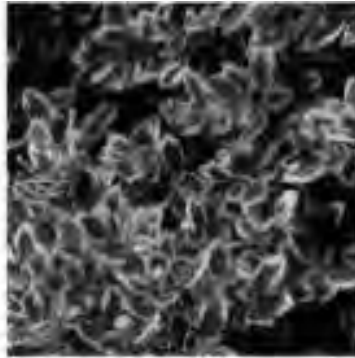
e) ткань

f) ткань

Природные текстуры



leaves



leaves



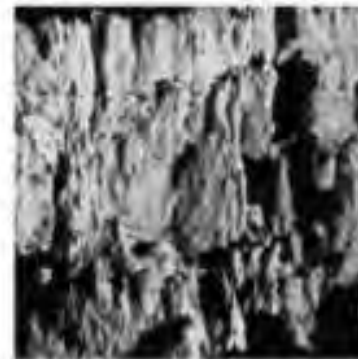
grass



brick

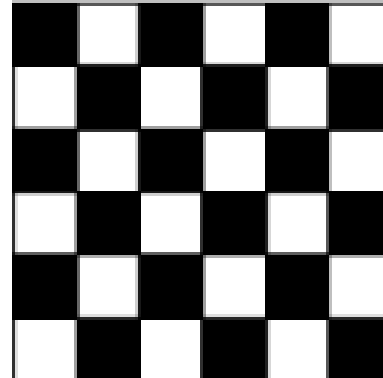
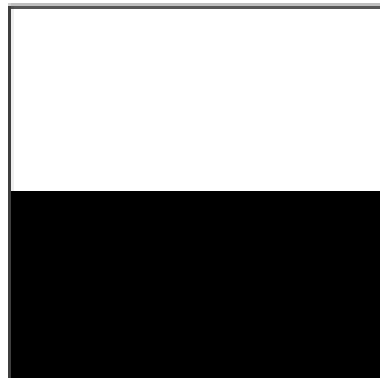


brick



stone

Виды текстур

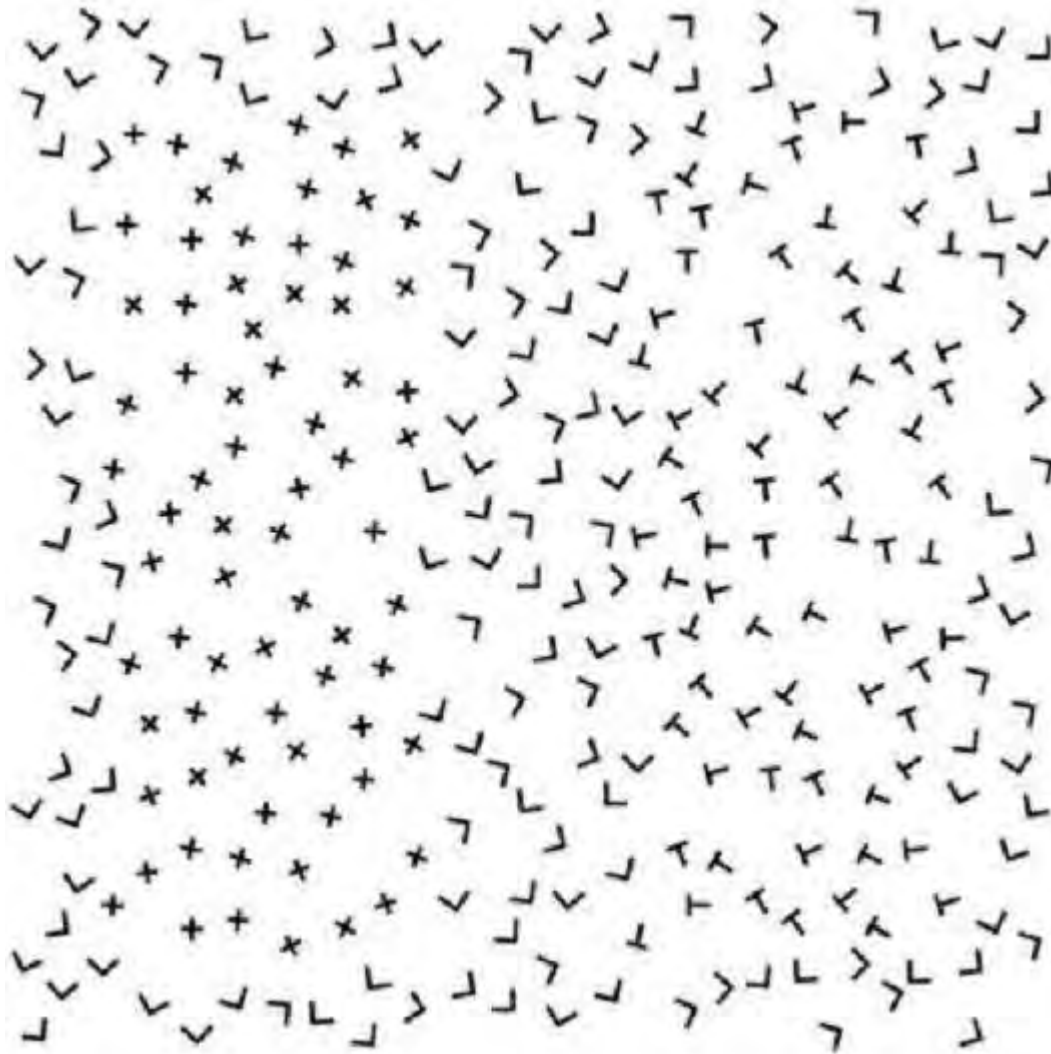


Признаки изображения

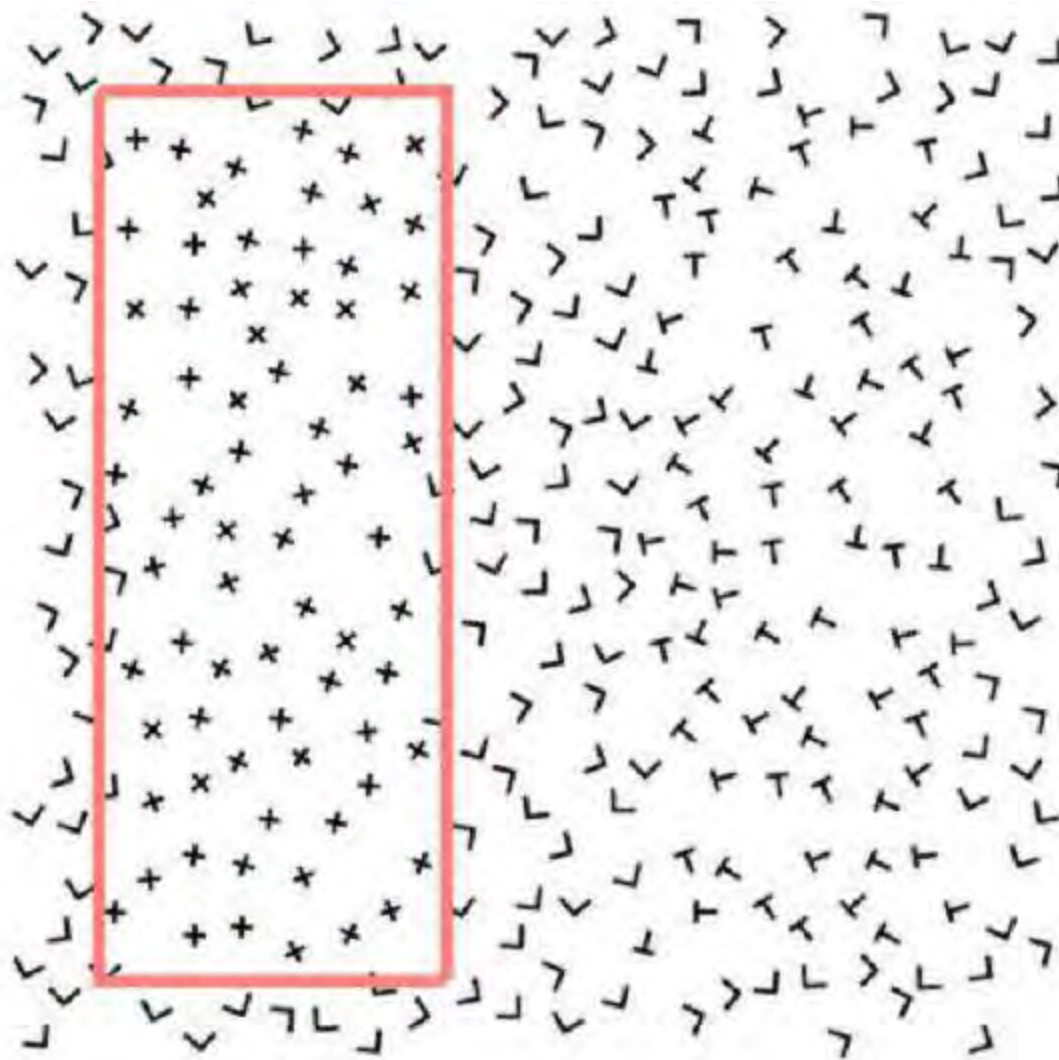
Какие признаки мы можем использовать для сравнения пикселей и регионов?

- Яркость
- Цвет
- ?

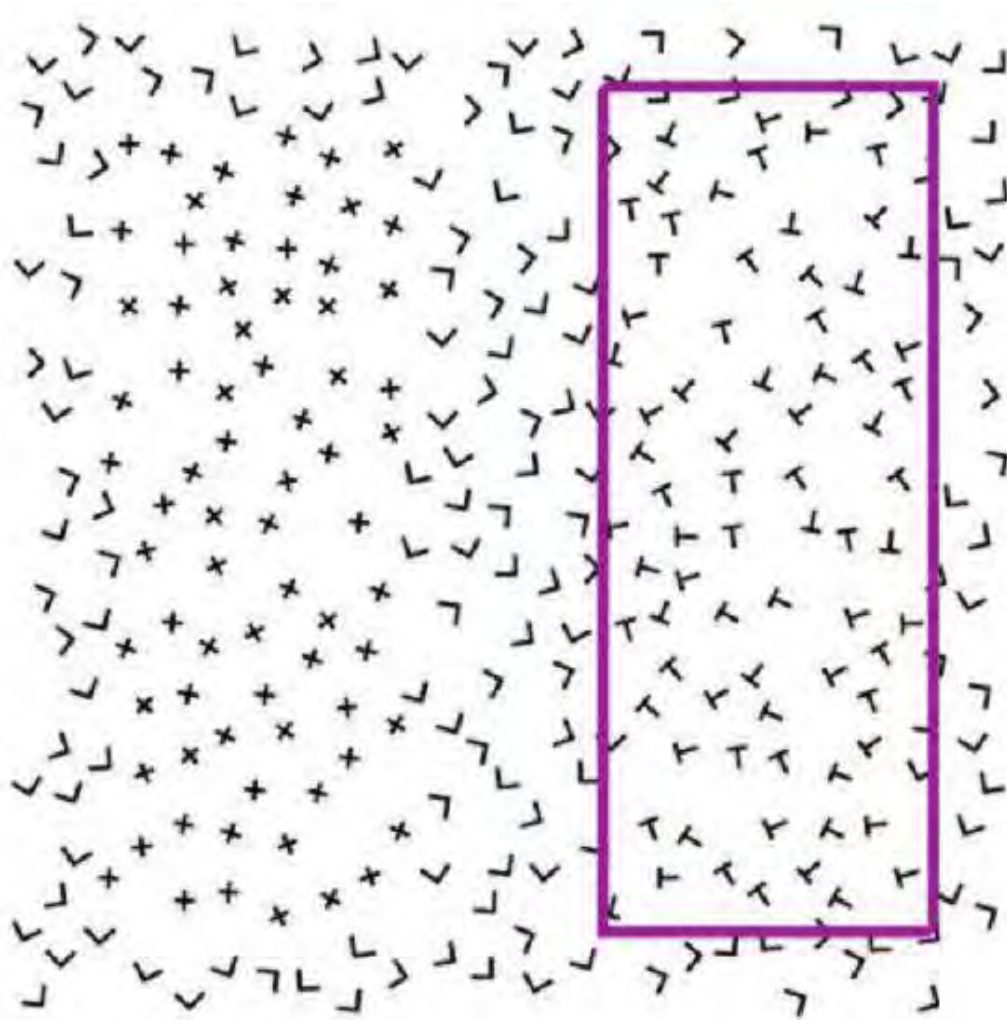
Видите отдельные области?



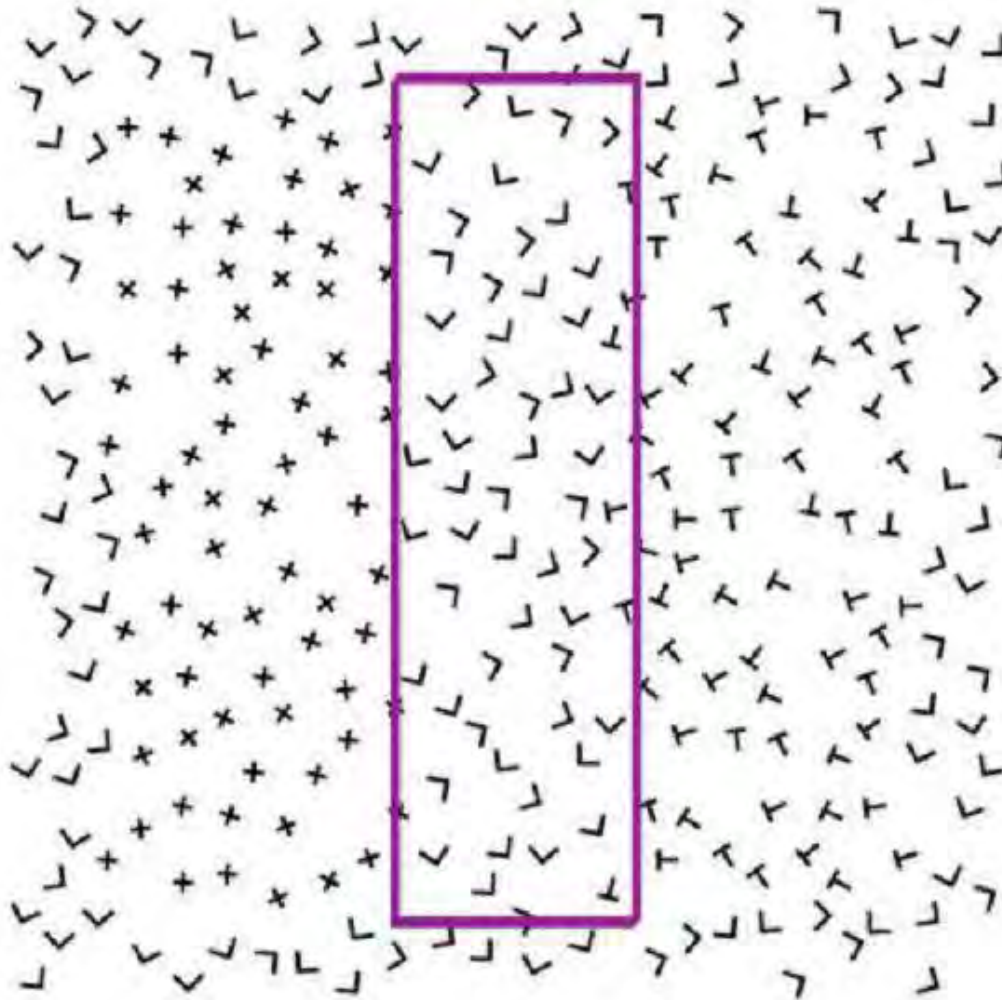
Область 1



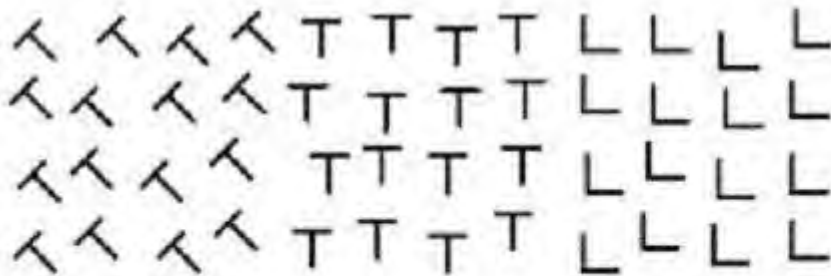
Область 2



Область 3



Текстура



(a)



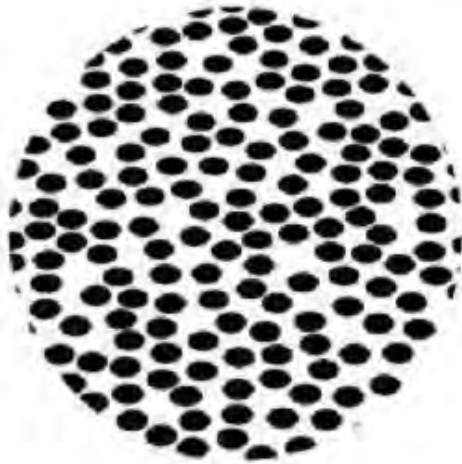
(b)

- **Текстура** — преимущественная ориентация элементов, составляющих материал (одно из определений)
- (b) - типичные примеры шаблонов (текстурных элементов) для исследований психофизиологического восприятия изображений
- Человек явно учитывает свойства текстуры фрагмента при распознавании изображений и выделении областей

Что ещё даёт текстура?

- распознавание изображений и выделение областей
- ???

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

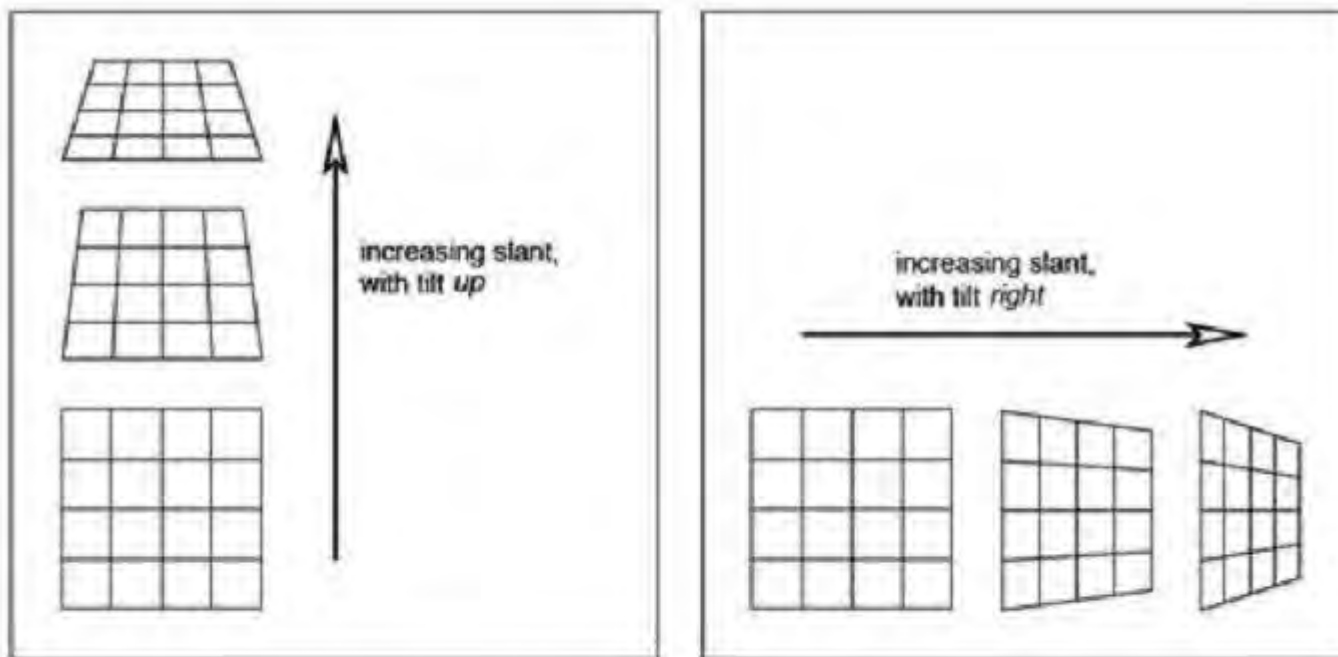
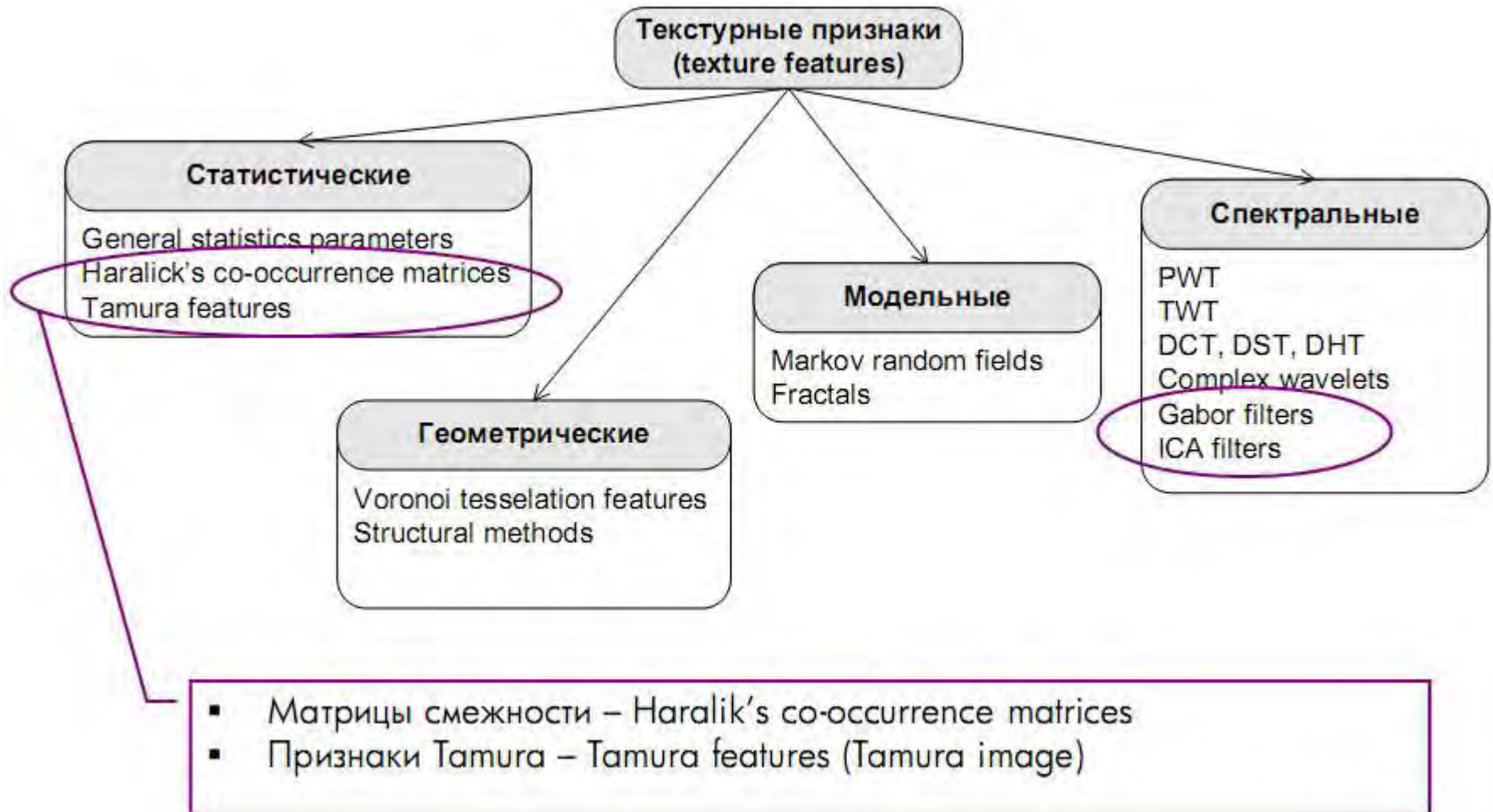


Figure 8.7. Surface orientation is often characterized in terms of *slant* and *tilt*.

Понятие текстуры при двух подходах текстурного анализа

- **Структурный подход:** текстура представляет собой множество примитивных текселов, расположенных в некотором регулярном порядке.
- **Статистический подход:** текстура является количественной характеристикой распределения значений интенсивности в области изображения.

Текстурные признаки



Один из вариантов структурного геометрического описания текстур

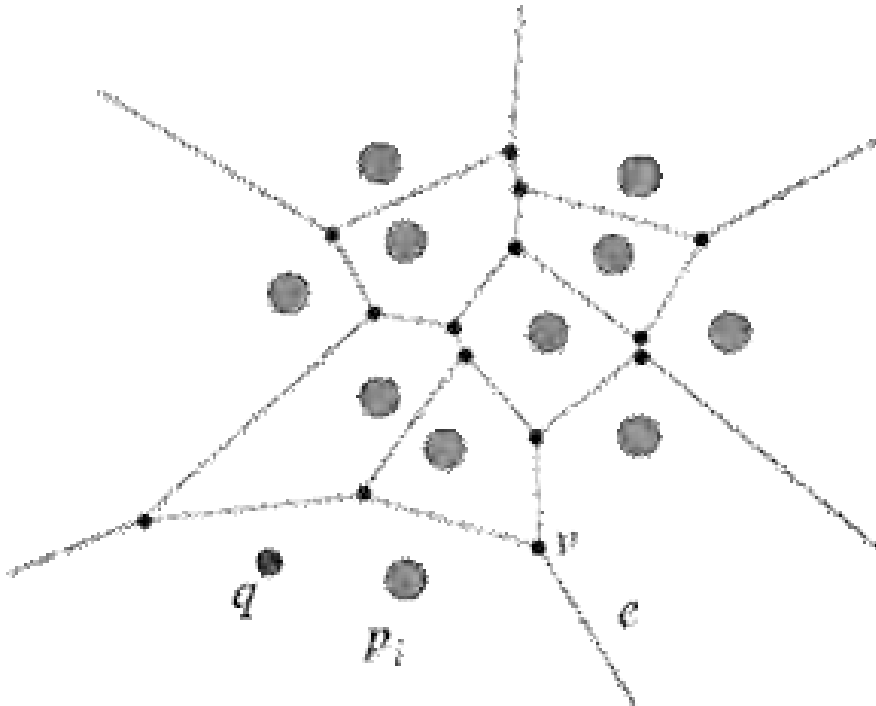
предложен (Tuceryan, Jain, 1990)



Текселы — области изображений, обнаруженных простыми операциями (наподобие пороговой бинаризации)

Характеристики пространственной взаимосвязи текселов определяются на основе **диаграммы Вороного**.

Диаграмма Вороного



Многоугольник Вороного:

$$V(P) = \bigcap_{Q \in S, Q \neq P} H^Q(P)$$

Алгоритм определения областей с однородной текстурой

1. Выделить текселы
2. Для них определить многоугольники Вороного
3. Вычислить признаки формы многоугольников
4. Сгруппировать многоугольники в кластеры

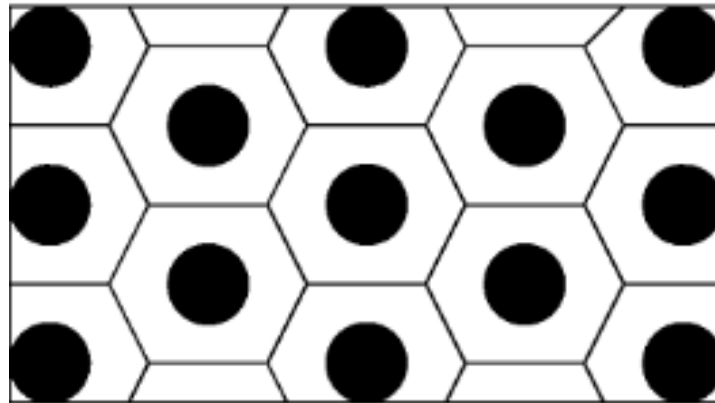


Диаграмма Вороного для множества круговых текселов

Статистический подход по сравнению со структурным:

- менее нагляден
- обладает вычислительными преимуществами
- может применяться как для сегментации, так и для классификации текстур

Количественные характеристики текстур

- Плотность и направление краёв
- Локальное двоичное разбиение
- Вычисление характерных признаков на основе матрицы вхождений (Характеристики Харалика)
- Энергетические текстурные характеристики Лавса
- Автокорреляция и спектр мощности
- Признаки Tamura
- Фильтры Габора

Плотность и направление краёв

- Использование детектора краёв на начальном этапе
- Вычисление количества краёв
- Определение направлений краёв

Плотность краёв

Количество краёв на единицу площади

$$F_{edgeness} = \frac{|\{p \mid Mag(p) \geq T\}|}{N}$$

где N – количество пикселей

$Mag(p)$ – модуль градиента для пиксела p

характеризует заполнение текстуры, но не её ориентацию

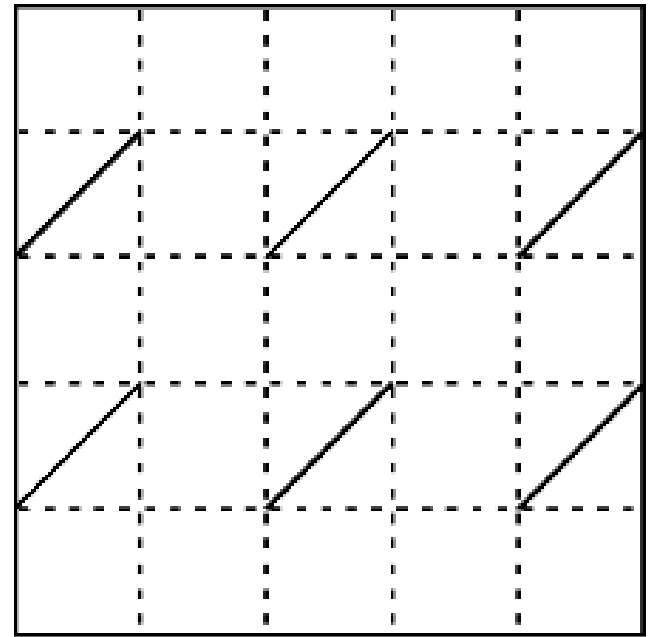
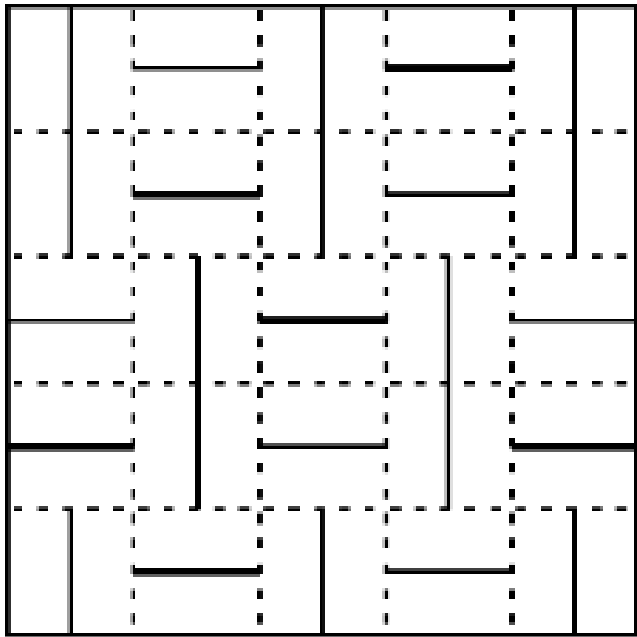
Расширение текстурной характеристики

- Учёт не только заполнения текстуры, но ещё и ориентации
- Нормированные гистограммы модуля и направления градиента с небольшим количеством разрядов(≤ 10)

$$F_{magdir} = (H_{mag}(R), H_{dir}(R))$$

Количественное описание текстуры области R

Текстурные характеристики



Плотность

1.0

0.24

Диаграмма
модуля

(0, 0.24, 0.76)

(0, 0.24, 0)

Диаграмма
градиента

(0.48, 0.52, 0)

(0, 0, 0.24)

Сравнение двух n-разрядных гистограмм

$$L_1(H_1, H_2) = \sum_{i=1}^n |H_1[i] - H_2[i]|$$

расстояние между гистограммами

Количественные характеристики текстур

- Плотность и направление краев
- Локальное двоичное разбиение
- Вычисление характерных признаков на основе матрицы вхождений (Характеристики Харалика)
- Энергетические текстурные характеристики Лавса
- Автокорреляция и спектр мощности
- Признаки Tamura
- Фильтры Габора

Локальное двоичное разбиение (LBP)

- Сравнение значение интенсивности каждого пиксела и восьми его соседей
- Результат проверки – двоичное число $b_1b_2b_3b_4b_5b_6b_7b_8$
- По этим числам строится гистограмма

Количественные характеристики текстур

- Плотность и направление краёв
- Локальное двоичное разбиение
- Вычисление характерных признаков на основе матрицы вхождений (смежности). Характеристики Харалика
- Энергетические текстурные характеристики Лавса
- Автокорреляция и спектр мощности
- Признаки Tamura
- Фильтры Габора

Матрицы вхождений (смежности) co-occurrence matrix

$C(i,j)$ – элемент матрицы – сколько раз встречалось i в отношении с j

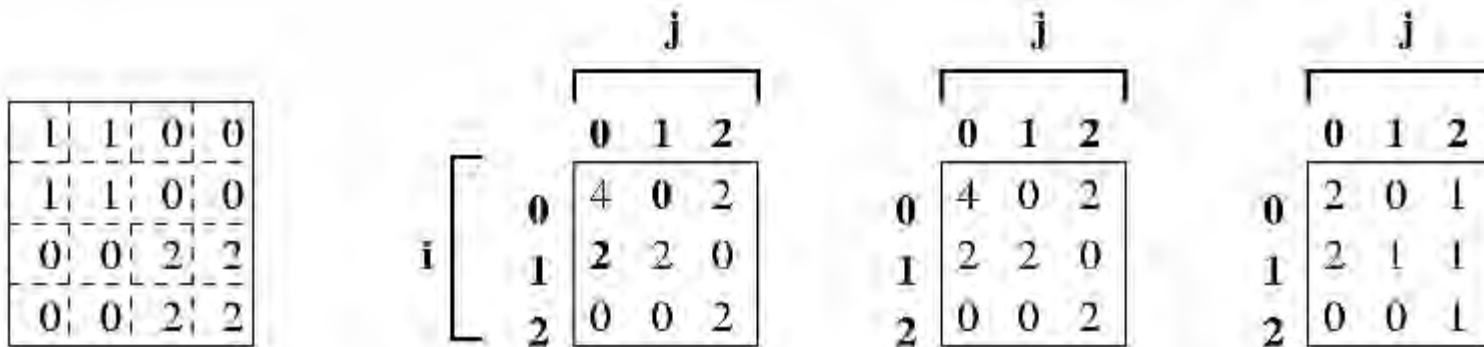
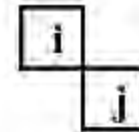
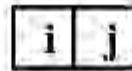


Image I

$C_{(0,1)}$

$C_{(1,0)}$

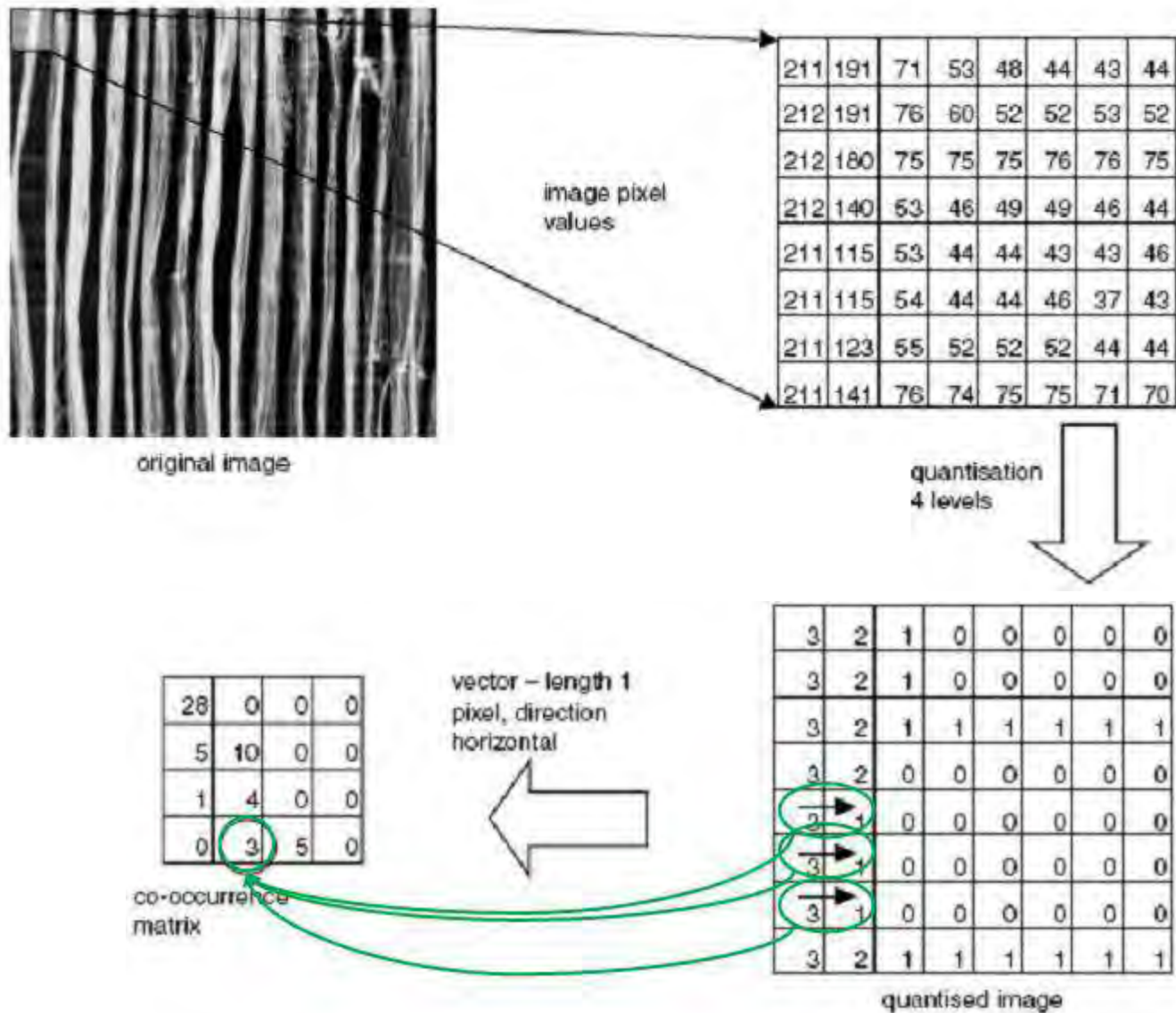
$C_{(1,1)}$



Полутоновая матрица вхождений

$$C(i, j) = \sum_{p=1}^N \sum_{q=1}^M \begin{cases} 1, \text{ если } I(p, q) = i, I(p + \Delta x, q + \Delta y) = j \\ 0, \text{ иначе} \end{cases}$$

$d=(\Delta x, \Delta y)$ – вектор пространственного отношения



Основные разновидности стандартной полутоновой матрицы вхождений Gray Level Co-occurrence Matrices (GLCM)

- Нормированная полутоновая матрица вхождений

$$N_d(i, j) = \frac{C_d(i, j)}{\sum_i \sum_j C_d(i, j)}$$

- Симметричная полутоновая матрица вхождений

$$S_d(i, j) = C_d(i, j) + C_{-d}(i, j)$$

Матрицы вхождений

- представляют свойства текстуры
- но не удобны для непосредственного применения при анализе изображений
- используются для вычисления числовых характерных признаков для более компактного представления текстуры

Характерные признаки

- **Энергия** (минимальна, когда все элементы равны)

$$\sum_i \sum_j N_d^2(i, j)$$

- **Энтропия** (максимальна, когда все элементы равны)

$$-\sum_i \sum_j N_d(i, j) \log_2 N_d(i, j)$$

- **Контраст** (мал, когда большие элементы вблизи главной диагонали)

$$\sum_i \sum_j (i - j)^2 N_d(i, j)$$

- **Однородность** (мала, когда большие элементы далеки от главной диагонали)

$$\sum_i \sum_j \frac{N_d(i, j)}{1 + |i - j|}$$

- **Корреляция**

$$\frac{\sum_i \sum_j (i - \mu_i)(j - \mu_j) N_d(i, j)}{\sigma_i \sigma_j}$$

Проблема

- Выбор вектора смещения d
- Zucker, Terzopoulos (1980) предложили использовать статистическую проверку χ^2 для выбора значений d

$$\max \chi^2(d) = \left(\sum_i \sum_j \frac{N_d^2(i, j)}{N_d(i)N_d(j)} - 1 \right)$$

$$N_d(i) = \sum_j N_d(i, j)$$

$$N_d(j) = \sum_i N_d(i, j)$$

Количественные характеристики текстур

- Плотность и направление краев
- Локальное двоичное разбиение
- Вычисление характерных признаков на основе матрицы вхождений (Характеристики Харалика)
- Энергетические текстурные характеристики Лавса
- Автокорреляция и спектр мощности
- Признаки Tamura
- Фильтры Габора

Энергетические текстурные характеристики Лавса (Laws)

- Обнаружение различных типов текстур с помощью локальных масок
- Энергетический подход, в котором оценивается изменение содержания текстуры в пределах окна фиксированного размера
- Для вычисления энергетических характеристик используется набор из девяти масок 5×5
- Энергетические характеристики каждого пиксела представляются вектором из 9 чисел

Векторы для вычисления масок

Для вычисления симметричного
взвешенного локального среднего
значения

$$L5 \text{ (Level)} : [1 \ 4 \ 6 \ 4 \ 1]$$

Для обнаружения краев

$$E5 \text{ (Edge)} : [-1 \ -2 \ 0 \ 2 \ 1]$$

Для обнаружения пятен

$$S5 \text{ (Spot)} : [-1 \ 0 \ 2 \ 0 \ -1]$$

Для обнаружения образа в виде ряби

$$R5 \text{ (Ripple)} : [1 \ -4 \ 6 \ -4 \ 1]$$

Получение двумерных масок

$$E5 \times L5 = E5L5$$

$$\begin{bmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 1 \end{bmatrix} \times [1 \ 4 \ 6 \ 4 \ 1] = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Метод Лавса (шаг 1)

Устранение влияния интенсивности освещения

По изображению перемещается окно и из каждого пиксела вычитается локальное среднее значение.

Формируется изображение, на котором средняя интенсивность каждой пиксельной окрестности близка к 0.

Размер окна зависит от типа изображений.

Для естественных сцен окно – 15x15

Метод Лавса (шаг 2)

Применение каждой из 16 масок 5x5

Формируются профильтрованные изображения

Энергетическая текстурная карта –

полноразмерное изображение

$$E_k[r, c] = \sum_{j=c-7}^{c+7} \sum_{i=r-7}^{r+7} |F_k[i, j]|$$

$F_k [i, j]$ – результат фильтрации пиксела k -ой маской

Метод Лавса (шаг 3)

Каждая симметричная пара карт заменяется усредненной картой

Например, можно усреднить:

E5L5 – горизонтальные края

L5E5 – вертикальные края

Среднее – края обоих типов

16 карт → 9 карт

Девять окончательных энергетических карт

L5E5/E5L5

E5R5/R5E5

L5S5/S5L5

S5S5

L5R5/R5L5

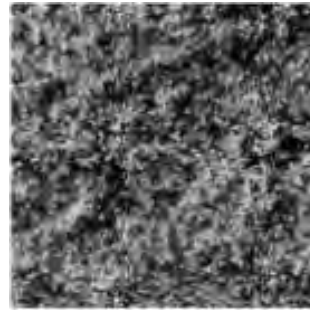
S5R5/R5S5

E5E5

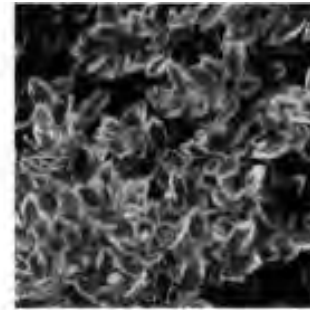
R5R5

E5S5/S5E5

Природные текстуры



leaves



leaves



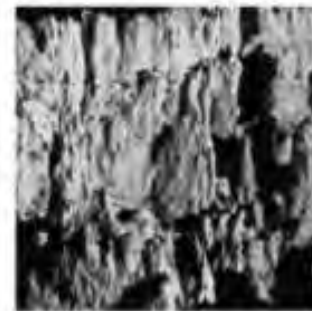
grass



brick



brick



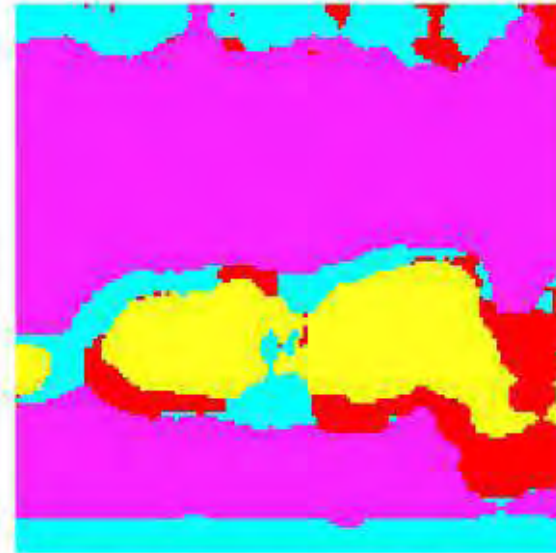
stone

Image	E5E5	S5S5	R5R5	E5L5	S5L5	R5L5	S5E5	R5E5	R5S5
Leaves1	250.9	140.0	1309.2	703.6	512.2	1516.2	187.5	568.8	430.0
Leaves2	257.7	121.4	988.7	820.6	510.1	1186.4	172.9	439.6	328.0
Grass	197.8	107.2	1076.9	586.9	410.5	1208.5	144.0	444.8	338.1
Brick1	128.1	60.2	512.7	442.1	273.8	724.8	86.6	248.1	176.3
Brick2	72.4	28.6	214.2	263.6	130.9	271.5	43.2	93.3	68.5
Stone	224.6	103.2	766.8	812.8	506.4	1311.0	150.4	413.5	281.1

Сегментация (характеристики Лавса)



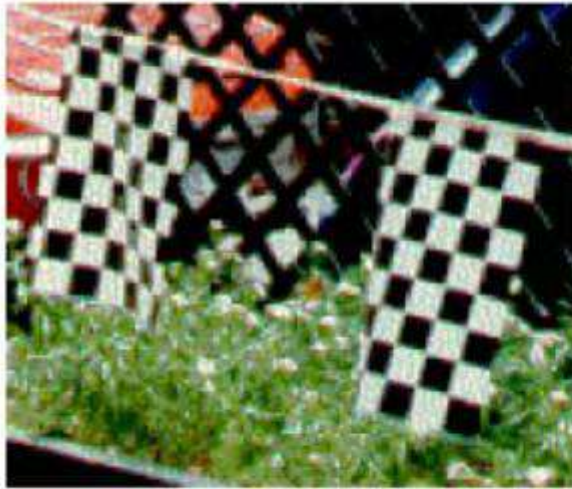
(a) Original image



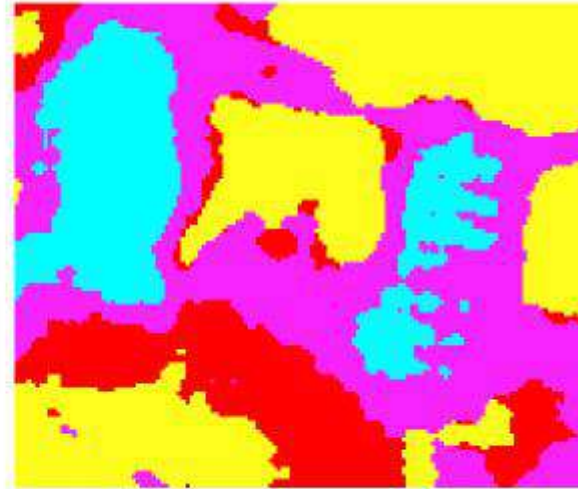
(b) Segmentation into 4 clusters

Region	E5E5	S5S5	R5R5	E5L5	S5L5	R5L5	S5E5	R5E5	R5S5
Tiger	168.1	84.0	807.7	553.7	354.4	910.6	116.3	339.2	257.4
Water	68.5	36.9	366.8	218.7	149.3	459.4	49.6	159.1	117.3
Flags	258.1	113.0	787.7	1057.6	702.2	2056.3	182.4	611.5	350.8
Fence	189.5	80.7	624.3	701.7	377.5	803.1	120.6	297.5	215.0
Grass	206.5	103.6	1031.7	625.2	428.3	1153.6	146.0	427.5	323.6
Small flowers	114.9	48.6	289.1	402.6	241.3	484.3	73.6	158.2	109.3
Big flowers	76.7	28.8	177.1	301.5	158.4	270.0	45.6	89.7	62.9
Borders	15.3	6.4	64.4	92.3	36.3	74.5	9.3	26.1	19.5

Сегментация (характеристики Лавса)



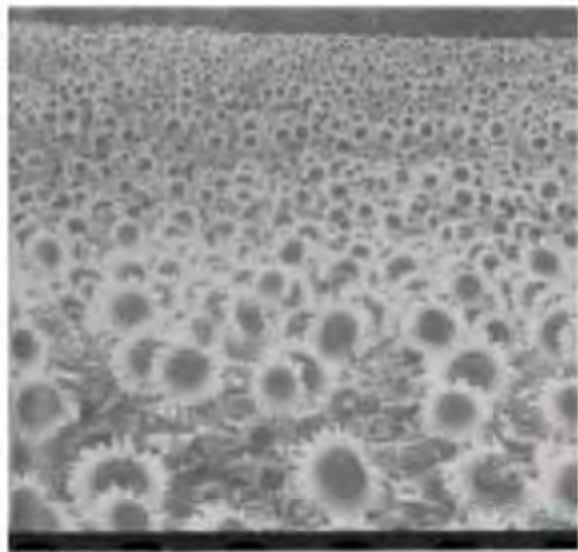
(c) Original image



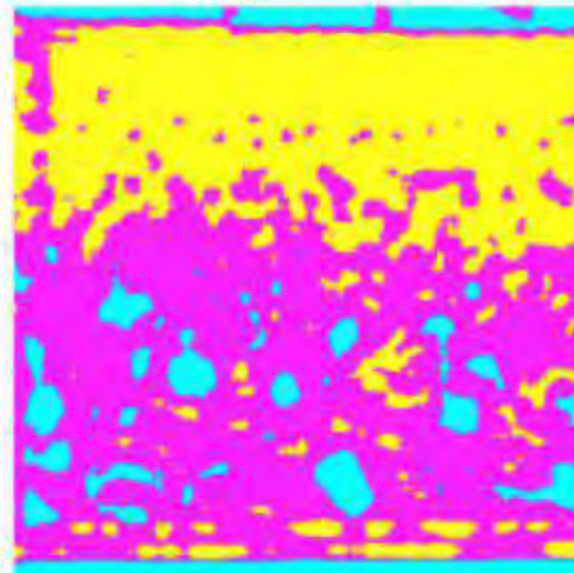
(d) Segmentation into 4 clusters

Region	E5E5	S5S5	R5R5	E5L5	S5L5	R5L5	S5E5	R5E5	R5S5
Tiger	168.1	84.0	807.7	553.7	354.4	910.6	116.3	339.2	257.4
Water	68.5	36.9	366.8	218.7	149.3	459.4	49.6	159.1	117.3
Flags	258.1	113.0	787.7	1057.6	702.2	2056.3	182.4	611.5	350.8
Fence	189.5	80.7	624.3	701.7	377.5	803.1	120.6	297.5	215.0
Grass	206.5	103.6	1031.7	625.2	428.3	1153.6	146.0	427.5	323.6
Small flowers	114.9	48.6	289.1	402.6	241.3	484.3	73.6	158.2	109.3
Big flowers	76.7	28.8	177.1	301.5	158.4	270.0	45.6	89.7	62.9
Borders	15.3	6.4	64.4	92.3	36.3	74.5	9.3	26.1	19.5

Сегментация (характеристики Лавса)



(e) Original image



(f) Segmentation into 3 clusters

Region	E5E5	S5S5	R5R5	E5L5	S5L5	R5L5	S5E5	R5E5	R5S5
Tiger	168.1	84.0	807.7	553.7	354.4	910.6	116.3	339.2	257.4
Water	68.5	36.9	366.8	218.7	149.3	459.4	49.6	159.1	117.3
Flags	258.1	113.0	787.7	1057.6	702.2	2056.3	182.4	611.5	350.8
Fence	189.5	80.7	624.3	701.7	377.5	803.1	120.6	297.5	215.0
Grass	206.5	103.6	1031.7	625.2	428.3	1153.6	146.0	427.5	323.6
Small flowers	114.9	48.6	289.1	402.6	241.3	484.3	73.6	158.2	109.3
Big flowers	76.7	28.8	177.1	301.5	158.4	270.0	45.6	89.7	62.9
Borders	15.3	6.4	64.4	92.3	36.3	74.5	9.3	26.1	19.5

Количественные характеристики текстур

- Плотность и направление краев
- Локальное двоичное разбиение
- Вычисление характерных признаков на основе матрицы вхождений (Характеристики Харалика)
- Энергетические текстурные характеристики Лавса
- Автокорреляция и спектр мощности
- Признаки Tamura
- Фильтры Габора

Автокорреляция и спектр мощности

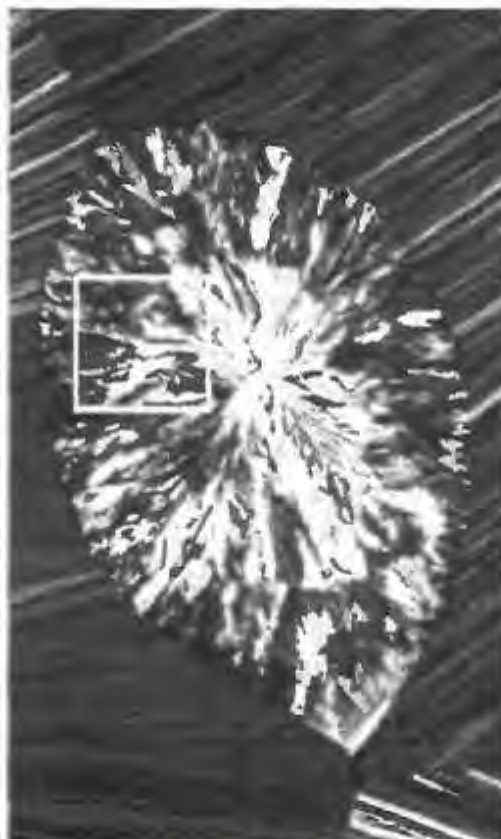
Используют

- для обнаружения повторяющихся образов текстурных элементов
- для описания мелкозернистости / грубости текстуры

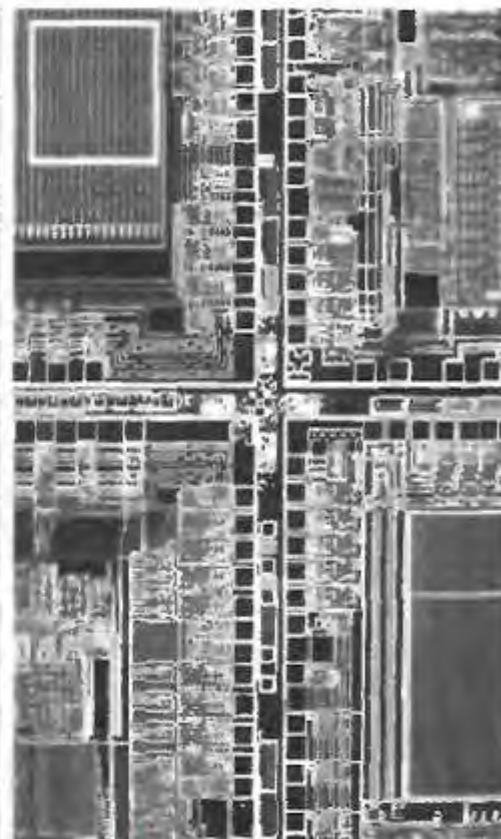
Текстура



Гладкая



Грубая



Периодическая

Автокорреляционная функция

$$\begin{aligned}\rho(dr, dc) &= \frac{\sum_{r=0}^N \sum_{c=0}^N I[r, c] I(r+dr, c+dc)}{\sum_{r=0}^N \sum_{c=0}^N I^2[r, c]} \\ &= \frac{I[r, c] \circ I_d[r, c]}{I[r, c] \circ I[r, c]}\end{aligned}$$

- В случае грубого текстурного образа автокорреляционная функция – медленно убывающая
- Автокорреляционная функция мелкозернистых текстур – быстро убывающая
- На графике автокорреляционных функций регулярных текстур присутствуют пики и впадины.

Количественные характеристики текстур

- Плотность и направление краев
- Локальное двоичное разбиение
- Вычисление характерных признаков на основе матрицы вхождений (Характеристики Харалика)
- Энергетические текстурные характеристики Лавса
- Автокорреляция и спектр мощности
- Признаки Tamura
- Фильтры Габора

Признаки Тамуры

Имеются 6 признаков, признанных существенными для зрительного восприятия в результате психологического эксперимента

Все они вычисляются по эвристическим процедурам и никак не масштабируются.

Tamura, H. Textural Features Corresponding to Visual Perception / H. Tamura, Sh. Mori, T. Yamawaki // IEEE Transactions on Systems, Man, and Cybernetics. – 1978. – June, Vol. SMC-8. – P. 460-473.

Признаки Тамуры

- **Зернистость** – это признак, связанный с расстоянием между заметными пространственными колебаниями оттенков серого, то есть с размером примитивных элементов (текселей), формирующих текстуру.
- **Контраст Тамуры** – это мера того, насколько сильно и резко может меняться цвет на изображении.
- **Направленность** – это признак, измеряемый с помощью гистограммы локальных направлений контуров.

Признаки Тамуры

- **Линейность** – это признак, показывающий, насколько прямолинейны контуры на изображении.
- **Регулярность** – это общая изменчивость первых четырёх признаков между различными частями изображения.
- **Грубость** – субъективная оценка грубости переходов на изображении

Признаки Tamura

Характеристики, существенные для зрительного восприятия:

- Зернистость (coarseness)
- Контрастность (contrast)
- Направленность (directionality)
- Линейность (line-likeness)
- Регулярность (regularity)
- Грубость (roughness)

Tamura image:

Coarseness-coNtrast-Directionality – точки в трехмерном пространстве CND

Признаки:

- Евклидово расстояние в 3D (QBIC)
- 3D гистограмма (Mars)

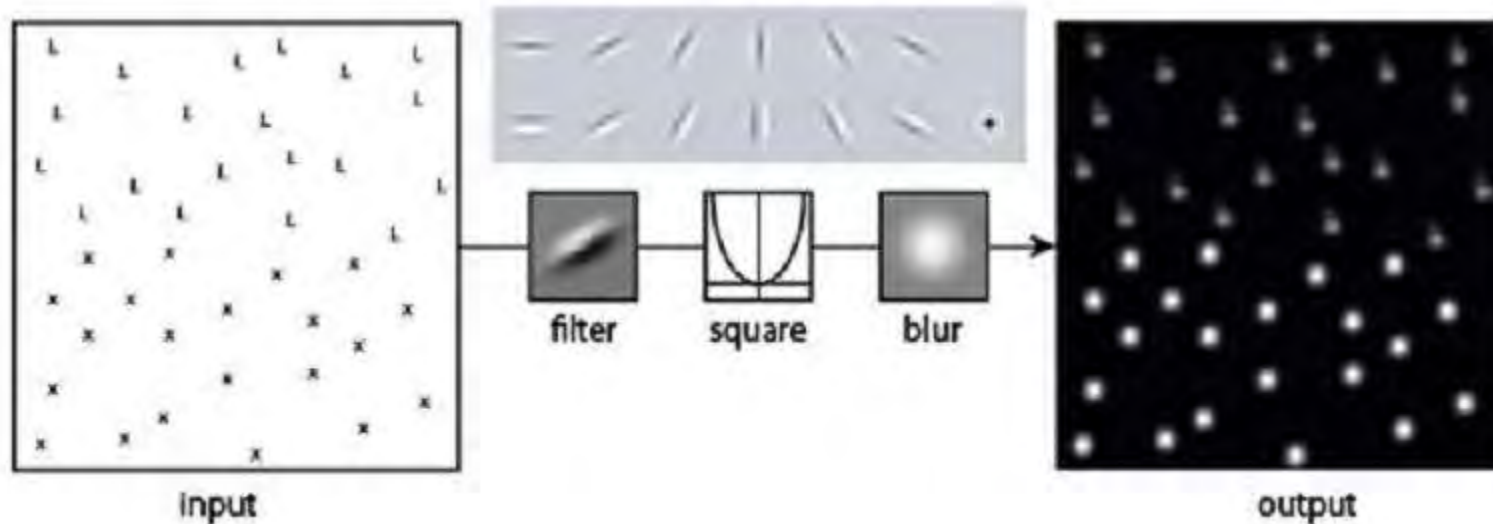
Количественные характеристики текстур

- Плотность и направление краев
- Локальное двоичное разбиение
- Вычисление характерных признаков на основе матрицы вхождений (Характеристики Харалика)
- Энергетические текстурные характеристики Лавса
- Автокорреляция и спектр мощности
- Признаки Tamura
- Фильтры Габора

Ещё один подход

- Определение структуры изображения с помощью блоков фильтров

Банки текстурных фильтров



- Выберем набор (банк) фильтров, каждый из которых чувствителен к краю определенной ориентации и размера
- Каждый пиксель изображения после обработки банком фильтров даёт вектор признаков
- Этот вектор признаков эффективно описывает локальную текстуру окрестности пикселя

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

Фильтры Габора

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

λ — длина волны

θ — ориентация

σ — сигма гауссиана

ψ — сдвиг фазы

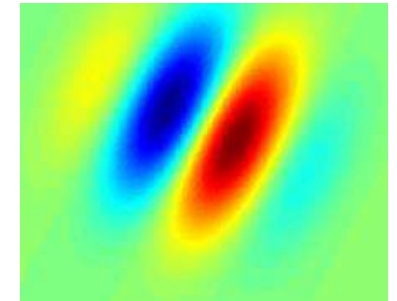
γ — коэффициент сжатия

или соотношение размеров
(aspect ratio),

«эллиптичность фильтра»

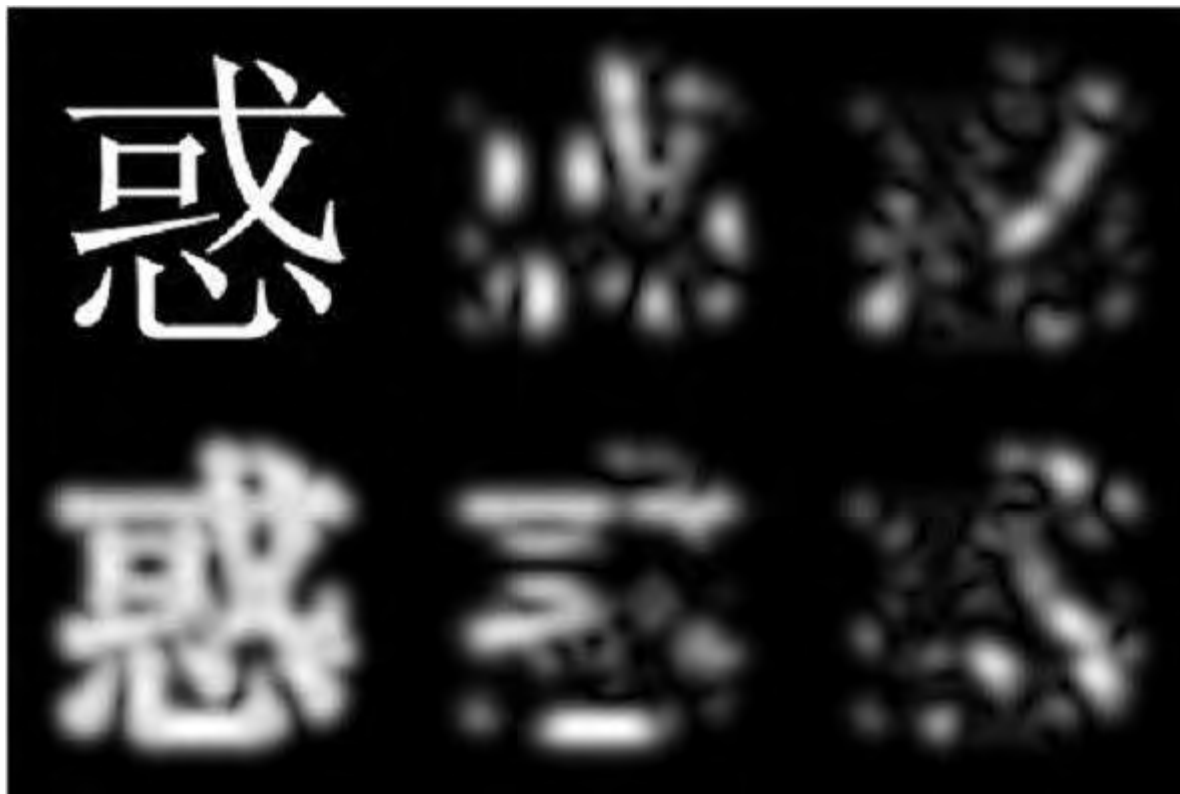
$$x' = x \cos(\theta) + y \sin(\theta)$$

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



- 2D фильтр Габора – ядро гауссиана, домноженное на синусоиду
- Предложены в 1947 Денисом Габором (нобелевским лауреатом), независимо переоткрыты в 1980 году
- Позволяет сделать банк фильтров, для выделения краёв разной ориентации, масштаба и положения в окрестности

Поиск краёв с помощью Габора



- Слева вверху – иероглиф
- 4 справа – применение фильтров Габора с ориентациями 0, 45, 90 и 135 градусов
- Слева внизу – совмещение результатов фильтрации

Фильтр Габора эффективен при обработке изображений со структурной избыточностью, имеющих квазипериодическую структуру.

Дактилоскопические изображения, изображения кристаллограмм и интерферрограмм.

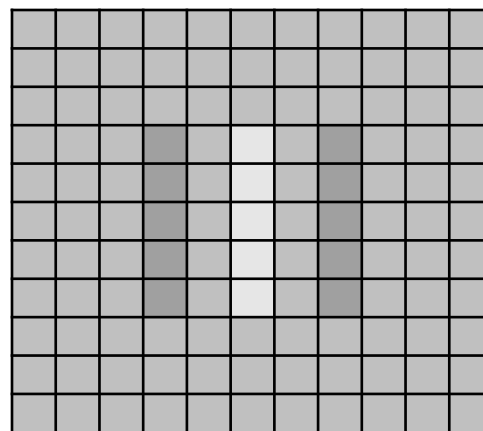
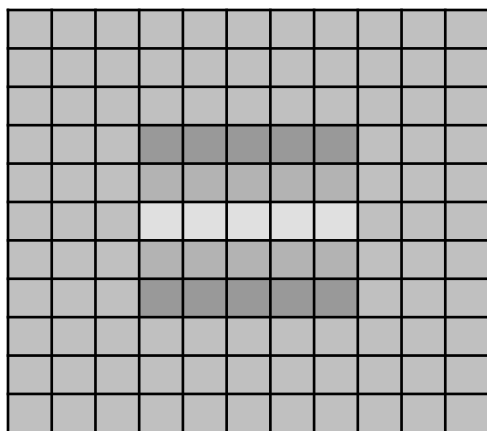
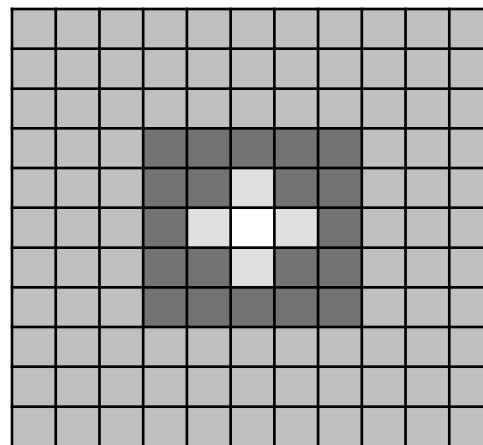
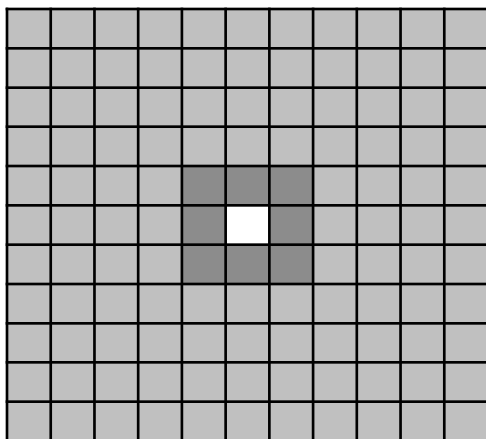
Окрас у зебр, различных кошачьих (тигры, рыси, дымчатые коты), птиц (тетерева), рыб (лепоринус полосатый).

Исходное
дактилоскопическое
изображение

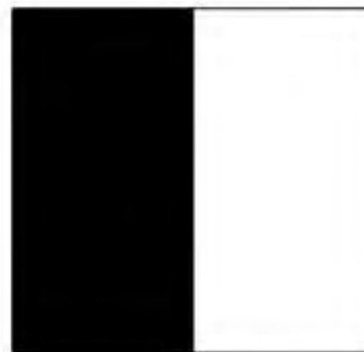


Дактилоскопическое
изображение,
обработанное
фильтром Габора

Признаки Хаара



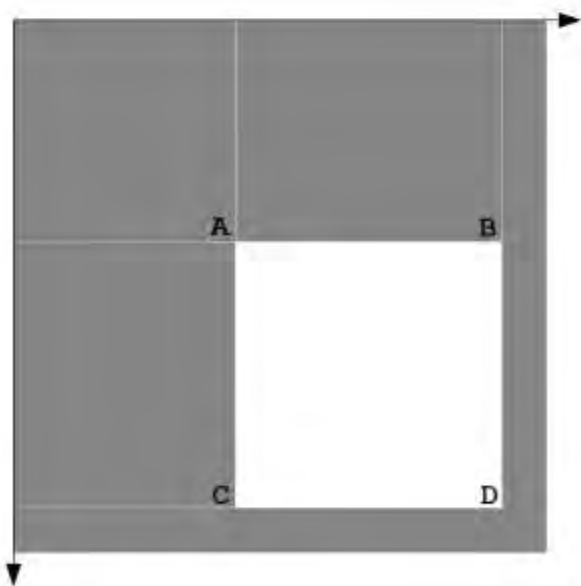
Признаки Хаара



Интегральное изменение яркости

Интегральное представление изображения

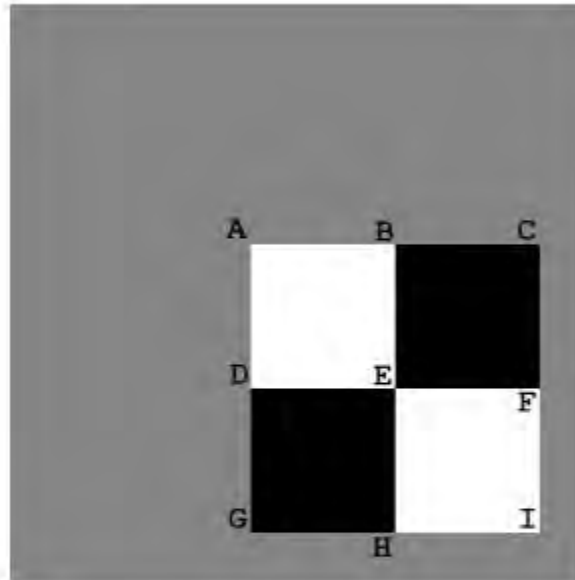
$$L(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j)$$



$$I_{ABCD} = L_D - L_B - L_C + L_A$$

$$L(x, y) = I(x, y) - L(x-1, y-1) + L(x, y-1) + L(x-1, y)$$

Интегральное представление



$$S_{ACGI} = L_A - 2L_B + L_C - 2L_D + 4L_E - 2L_F + L_G - 2L_H + L_I$$

Локализация инородных объектов на изображении



Нестеренко В.А.

Словарь текстурных элементов (текстонов)

- **Алгоритм:**

- Для каждого пикселя вычислим отклики по банку фильтров (вектор-признак)
- Составим обучающую выборку векторов-признаков
- Кластеризуем k-средними
 - К центров кластеров - Текстоны
 - Вместе они образуют словарь
- Теперь каждому пикселю изображения мы можем сопоставить номер текстона из словаря

- **Получили новые признаки:**

- Номер текстона для каждого пикселя
- Гистограмму частот текстонов для окрестности пикселя



Пример банка из 13 фильтров



Часть текстонов из словаря

J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and Texture Analysis for Image Segmentation," IJCV 2001

Поиск изображений

Лекция 11

История вопроса

- Термин «Content-based image retrieval» (CBIR) впервые был введен в употребление в 1992 году Т. Като
- Компания [Picsearch](#) выпустила первую публичную версию поиска по изображениям в сентябре 2001 года.
- В июле того же года запустил свой [поиск по картинкам Google](#).
- [Яндекс.Картинки](#) официально открылись в июле 2002 года, став первым российским поисковиком, ищущим изображения.
- В ноябре 2003 г. Yahoo! добавил справа от поисковой строки меню с опцией [поиска по изображениям](#) в том числе.

Яндекс 2015



Исходная картинка
379×567

Таких же картинок не найдено

Похожие картинки



Яндекс 2016





Яндекс 2017



Яндекс 2018

Яндекс

 Загруженная картинка 



Найти

Поиск **Картинки** Видео Карты Маркет Новости Переводчик Музыка Ещё

Моя лента Мои коллекции Темы > Мне нравится > Добавить карточку Ещё >

[← Вернуться назад](#)



Исходная картинка
379×567

Эта картинка в других размерах

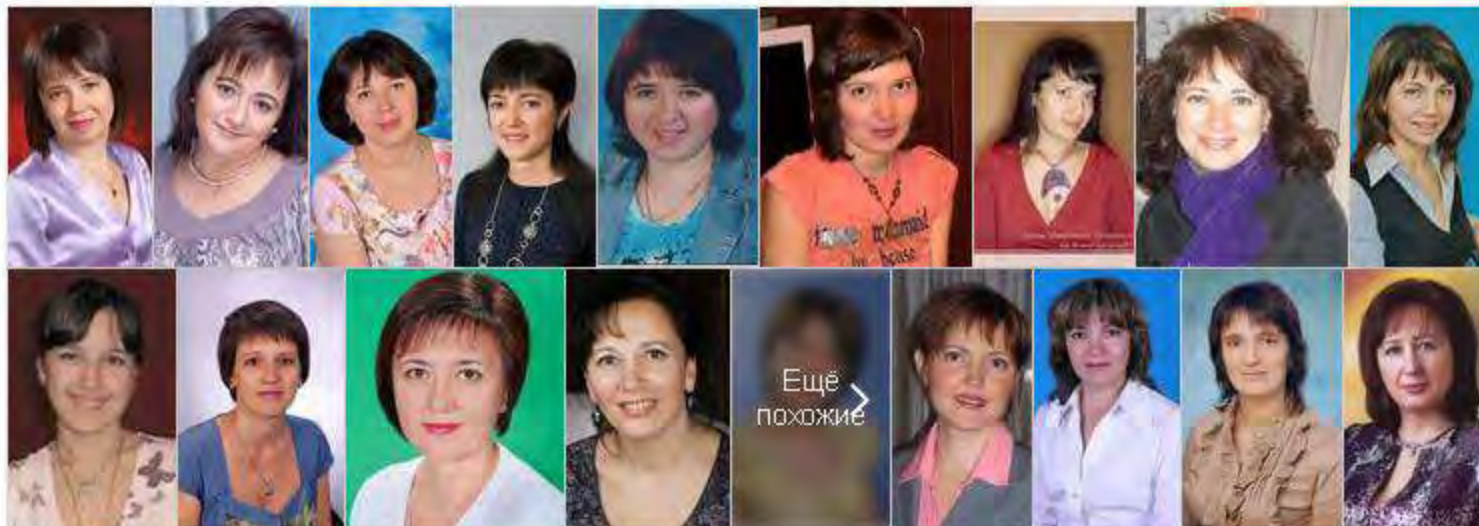
Средние	Маленькие
379×567	200×200
	200×200
	200×200
	180×240
	130×129
	100×100

Кажется, на картинке

женщина

Яндекс 2018 -- продолжение

Похожие картинки



Сайты, где встречается картинка



Демяненко Яна Михайловна, домашняя страница - Демяненко Яна Михайловна

staff.mmcs.sfedu.ru

Демяненко Яна Михайловна, домашняя страница - Демяненко Яна Михайловна



Исходная картинка
667x1000

Эта картинка в других размерах

379x567

376x376

Кажется, на картинке

женщина

яна демьяненко

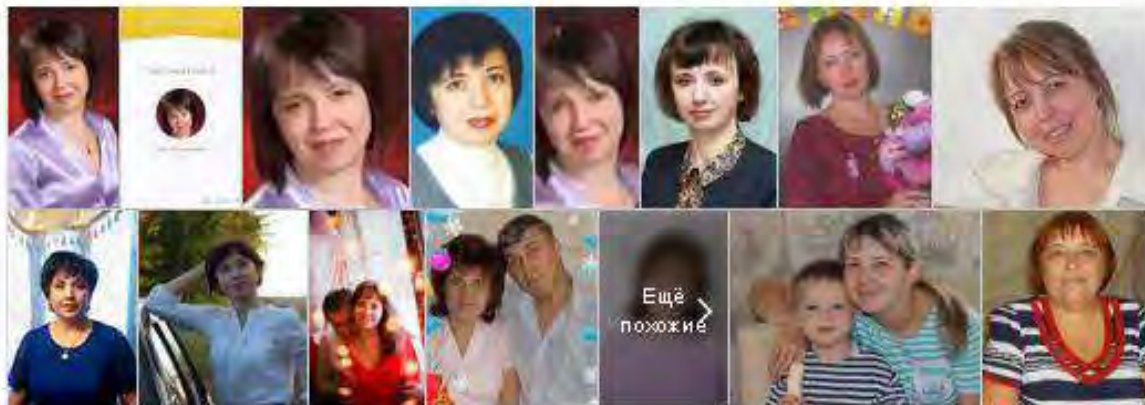
михайловна

самородова любовь анатольевна

маргарита колесникова белгород

Яндекс 2019

Похожие картинки



Ещё
похожие

Яндекс 2019 -- продолжение

Сайты, где встречается картинка



Демяненко Яна Михайловна, домашняя страница - Демяненко Яна Михайловна

staff.mmc.sfedu.ru

Демяненко Яна Михайловна, домашняя страница - Демяненко Яна Михайловна



Demyanenko Yana, homepage - Demyanenko Yana

staff.mmc.sfedu.ru

Demyanenko Yana, homepage - Demyanenko Yana



Research

sfedu.ru

Yana M. Demyanenko



Кафедра прикладной математики и программирования

sfedu.ru

Демяненко Яна Михайловна



4 курс ФИИТ) Компьютерная графика - ФИИТ, 4 курс - Форум мехмата ЮФУ

forum.mmc.sfedu.ru

4 курс ФИИТ) Компьютерная графика

Яндекс 2020

Яндекс

Загруженная картинка

Найти



Будьте в Плюсе



Регистрация

Войти

Поиск **Картинки** Видео Карты Маркет Новости Переводчик Эфир Кью Услуги Музыка Все

Моя лента Мои коллекции Темы Ещё



Исходный размер изображения: 200×200

Выбрать фрагмент

Это изображение в других размерах

200×198

Показать все размеры

Сайты, где встречается картинка



Yana Demyanenko, Ростов-на-Дону, Россия, ВКонтакте
118.face-base.online
Анализ страницы Yana Demyanenko, Вк



Вячеслав Цибулин, Ростов-на-Дону, Россия
lprofiles.ru
vyacheslav cibulin



Олег Хачумов
Вконтакте24.pф

Кажется, на картинке

женщина

Похожие картинки



Больше похожих

Яндекс 2021

Яндекс

Загруженная картинка x

Найти



Будьте в Плюсе



Регистрация

Войти

Поиск **Картинки** Видео Карты Маркет Новости Переводчик Кью Услуги Музыка Все

Размер

Ориентация

Цвет

Файл

Товары

Обои 2560x1440

На сайте



Пригласите в Instagram друзей с Facebook



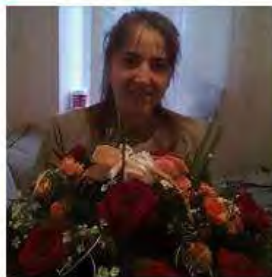
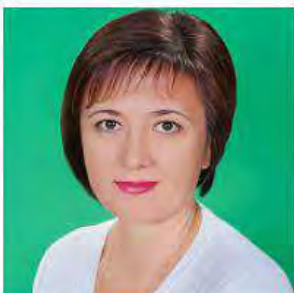
Yana Demyanenko

Пригласить



Alexey Kolesnikov

Пригласить



Яндекс 2021— а так?

Яндекс

Загруженная картинка x

Найти



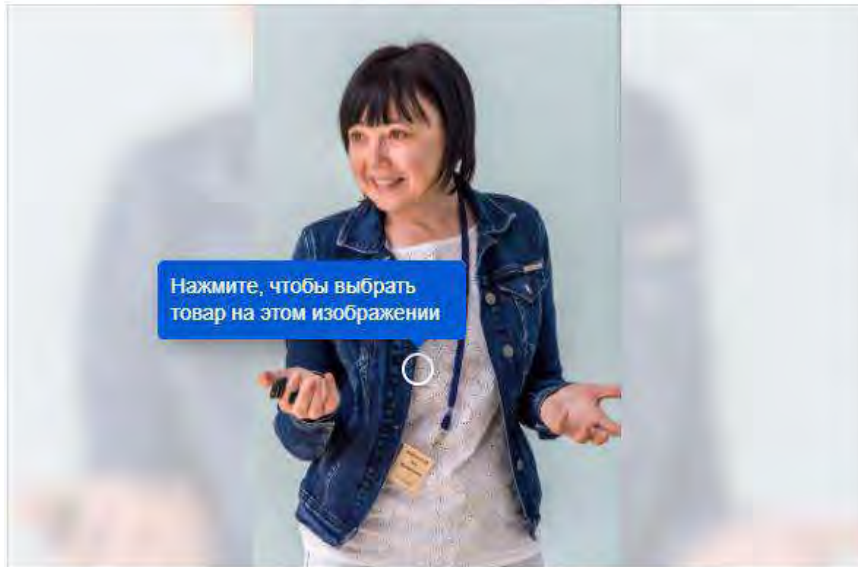
Будьте в Плюсе



Регистрация

Войти

Поиск **Картинки** Видео Карты Маркет Новости Переводчик Кью Услуги Музыка Все



Исходный размер изображения: 533x804

Выбрать фрагмент

Другие размеры изображения

Таких же изображений не найдено

Сайты с информацией про изображение



Валентина Воронина (Кобцева) OK.RU

ok.ru

461x338

59 лет

Кажется, на изображении

женщина

человек

андрева татьяна анатольевна



Похожие изображения



Больше похожих

Google 2014



Совет: введите описание изображения в строке поиска.



Размер изображения:
379 × 567

Изображения других размеров не найдены.

Похожие изображения - [Пожаловаться на картинки](#)





Размер изображения:
379 × 567

Изображения других размеров не найдены.

Google 2015

Совет: введите описание изображения в строке поиска.

Похожие изображения

[Пожаловаться на картинки](#)



Google 2016



Google 2017



Размер изображения:
379 × 567

Изображения других размеров не найдены.

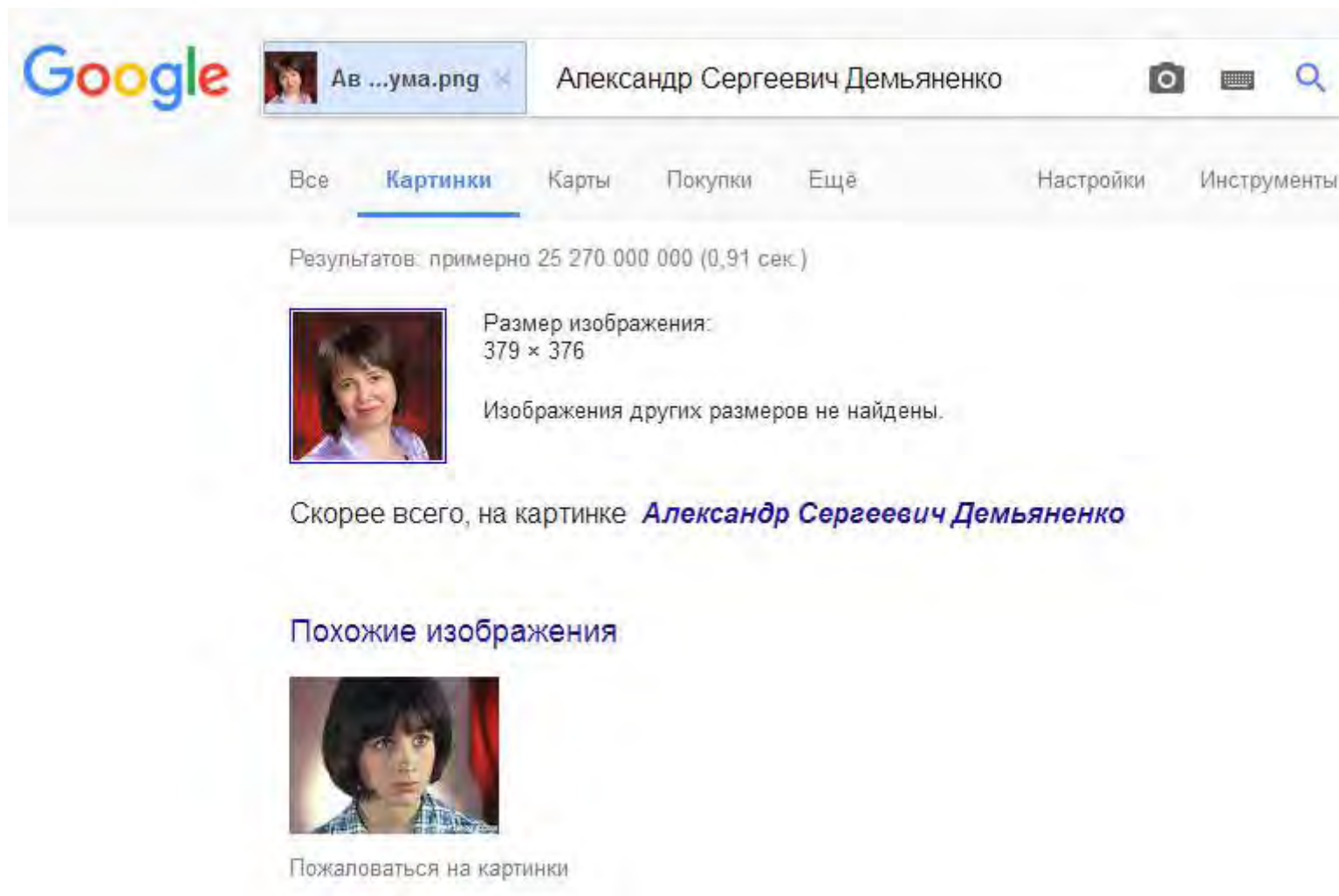
Скорее всего, на картинке *[Александр Сергеевич Демьяненко](#)*

Похожие изображения



[Пожаловаться на картинку](#)

Google 2017




Google

Ав ...ума.png × Александр Сергеевич Демьяненко

Все **Картинки** Карты Покупки Ещё Настройки Инструменты


Результатов: примерно 25 270 000 000 (0,91 сек.)

 Размер изображения:
379 × 376

Изображения других размеров не найдены.

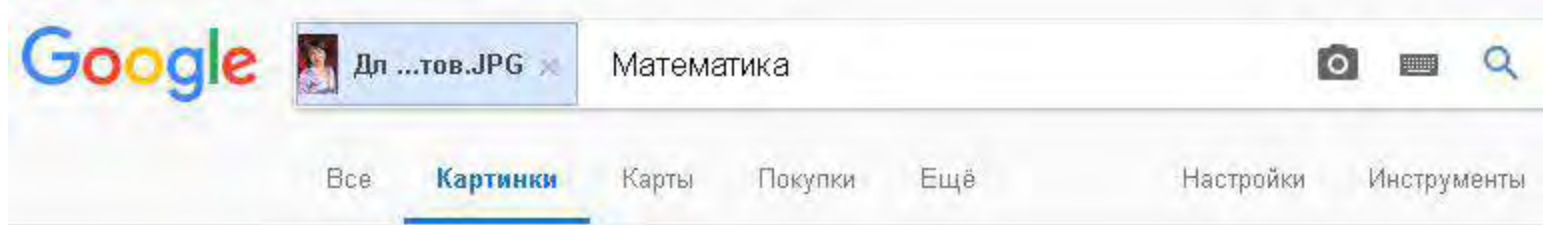
Скорее всего, на картинке ***Александр Сергеевич Демьяненко***

Похожие изображения



[Пожаловаться на картинки](#)

Google 2018



Результатов: примерно 3 (0,87 сек.)



Размер изображения:
379 × 567

Изображения других размеров не найдены.

Скорее всего, на картинке **Математика**

Похожие изображения



Google 2018 -- продолжение

Страницы с подходящими изображениями

Виталий Брагилевский (@_bravit) | Twitter



https://twitter.com/_bravit ▼

73 × 73 - The latest Tweets from Виталий Брагилевский (@_bravit). Преподаватель в #ОФУ, читаю курсы по ФП (Haskell, Idris) и теоретической информатике, пишу разное, иногда шучу (не очень). Ростов-на-Дону, Россия.

Белоконь (@Katamarinaki) | Twitter



<https://twitter.com/katamarinaki> ▼

73 × 73 - The latest Tweets from Белоконь (@Katamarinaki). русские оказались неуязвимы перед генетическим оружием. Ростов-на-Дону.

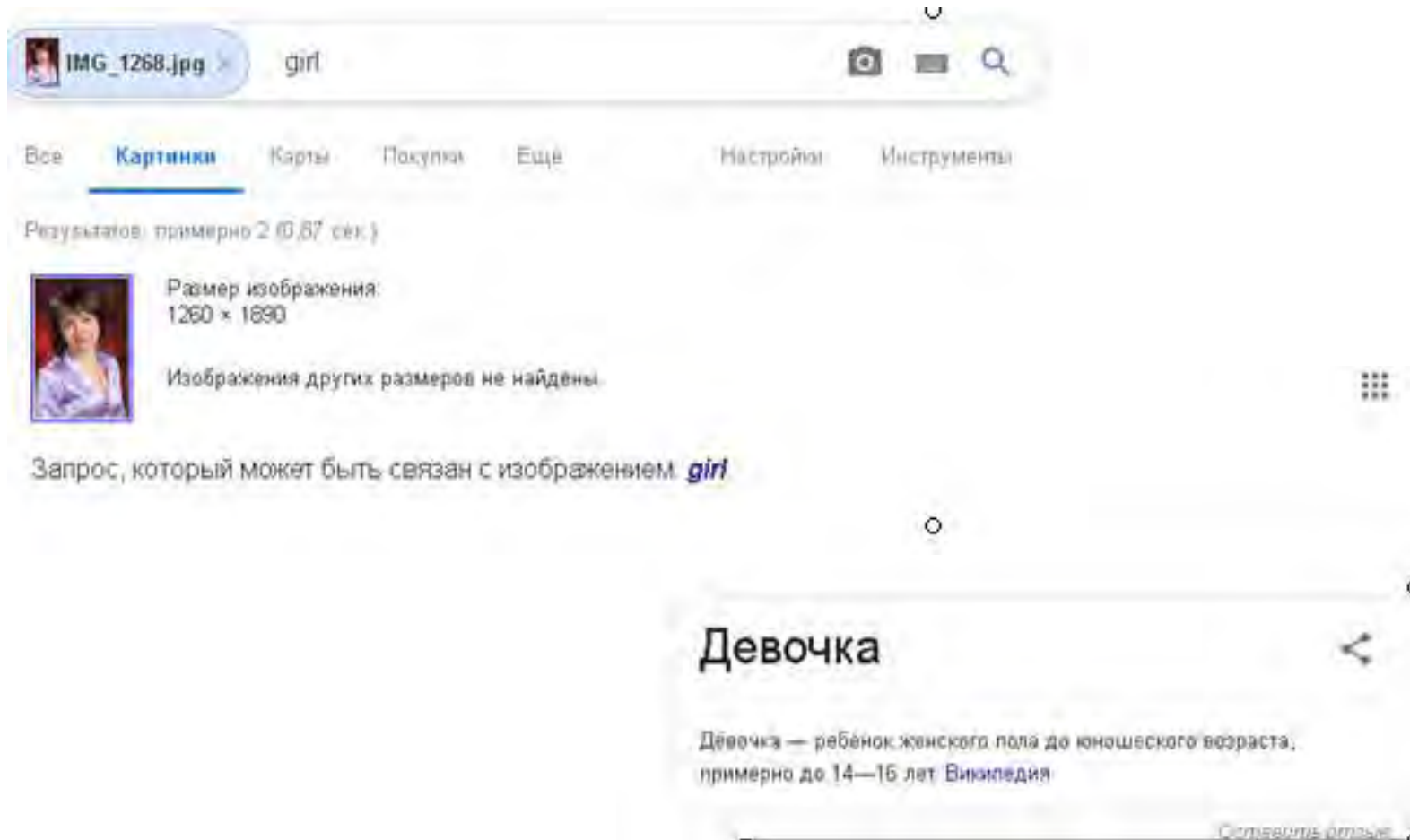
Распределение по научным руководителям 2016/2017 ...



docplayer.ru/48342556-Raspredelenie-po-nauchnyrn-rukovoditelyam-201... ▼

379 × 567 - Распределение по научным руководителям 2016/2017 Руководство для студентов направления «Фундаментальная информатика и информационные технологии» Содержание 1 Введение Научно-исследовательская работа.

Google 2019



The screenshot shows a Google search interface. At the top, the search bar contains the text "girl" and a small thumbnail of a girl's face labeled "IMG_1268.jpg". Below the search bar, navigation tabs include "Все", "Картинки", "Карты", "Покупки", "Ещё", "Настройки", and "Инструменты". The "Картинки" tab is selected. Below the tabs, it says "Результатов: примерно 2 (0,87 сек.)". A single image result is shown: a portrait of a young girl with dark hair, wearing a purple top. To the right of the image, the text reads "Размер изображения: 1260 x 1690" and "Изображения других размеров не найдены". Below the image, a suggestion says "Запрос, который может быть связан с изображением: [girl](#)". At the bottom, a knowledge panel is visible with the title "Девочка" and a definition: "Девочка — ребёнок женского пола до юношеского возраста, примерно до 14—15 лет. Википедия".

Google 2019 -- продолжение

Похожие изображения



[Пожаловаться на картинки](#)

Россия

● **Советский р-н, Ростов-на-Дону** - Из вашей истории поиска - Учитывать мое местоположение - Подробнее...


Google 2020

Google

ав ...рка.jpg hair design

Все Картинки Карты Покупки Ещё Настройки Инструменты

Результатов: примерно 3 (0,79 сек.)

 Размер изображения:
200 × 200


Изображения других размеров не найдены.

Запрос, который может быть связан с изображением: *hair design*

yandex.ru > ... > Парикмахерские ▾

[Hair design, парикмахерская, ул. Адмирала Октябрьского ...](#)

Hair design ★, ул. Адмирала Октябрьского, 12, Севастополь, Россия: ✓ фотографии, адрес и 📞 телефон, часы работы, фото и отзывы посетителей ...

 Похожие изображения



[Пожаловаться на картинки](#)

Google
2020 --
продол
жение

Страницы с подходящими изображениями

[docplayer.ru](#) > [48342556-Raspredelenie-po-nauchnym-...](#) ▾

[Распределение по научным руководителям 2016/2017 ...](#)



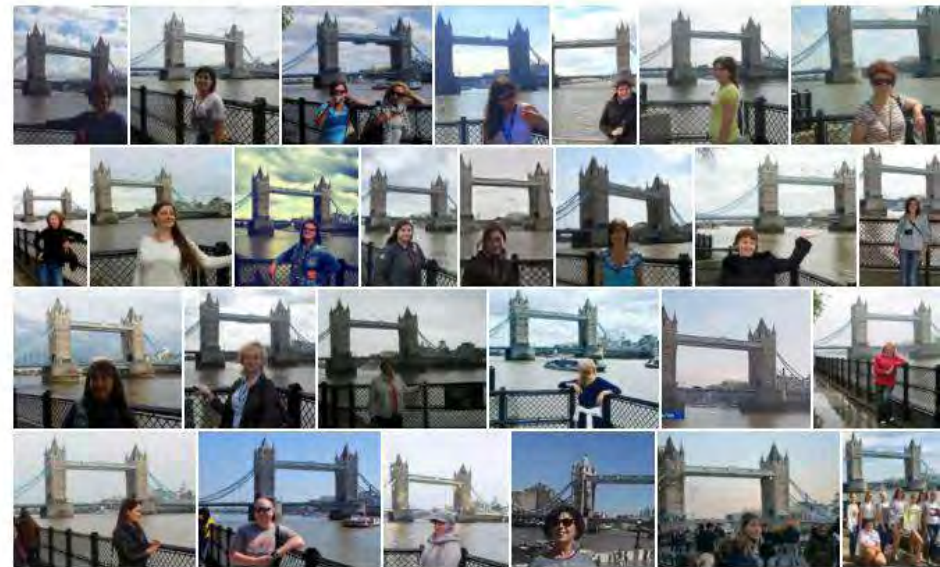
379 × 567 — [Распределение по научным руководителям 2016/2017](#) Руководство для студентов направления «Фундаментальная информатика и информационные ...»

2020



женщина

Похожие картинки




Исходный размер изображения: 504×618

Это изображение в других размерах

Таких же картинок не найдено

Сайты, где встречается картинка

 [Дмитрий Субботин, Пермь, Россия](#)

Результатов: примерно 303 (1,26 сек.)



Размер изображения: 504 x 618
Изображения других размеров не найдены.

Запрос, который может быть связан с изображением: tower bridge

ru.wikipedia.org > wiki > Тауэрский_мост

Тауэрский мост — Википедия

Тауэрский мост, или Тауэр-бридж (англ. **Tower Bridge**, дословный перевод — Башенный мост), — комбинированный подвесной (висячий) и разводной ...

en.wikipedia.org > wiki > Tow... Перевести эту страницу

Tower Bridge - Wikipedia

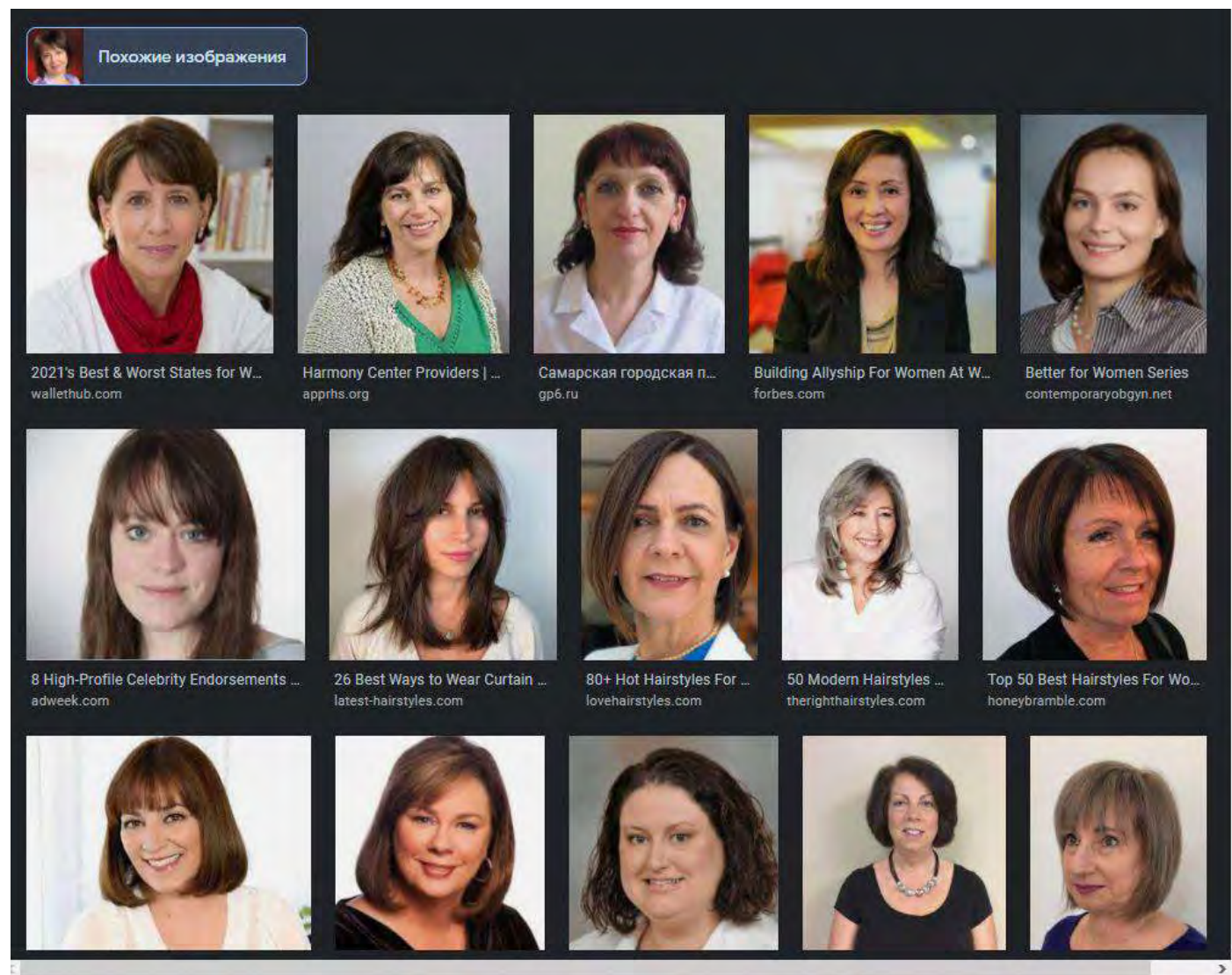
Tower Bridge is a combined bascule and suspension bridge in London, built between 1886 and 1894. The bridge crosses the River Thames close to the Tower ...

Похожие изображения



2020

Google 2021



Google 2021



Размер изображения:
533 × 804

Изображения других размеров не найдены.

Запрос, который может быть связан с изображением: **boyfriend**

<https://www.youtube.com> > w... > Перевести эту страницу

Justin Bieber - Boyfriend (Official Music Video) - YouTube

3 мая 2012 г. — Music video by Justin Bieber performing **Boyfriend**. © 2012 The Island Def Jam Music Group#VEVOCertified on July 11, ...

<https://context.reverso.net> > английский-русский > boy... >

Перевод "boyfriend" на русский - Reverso Context

Or laughing about how your **boyfriend** kissed your best friend. Или смеются о том, что твой парень поцеловал твою лучшую подругу.

Похожие изображения



Boyfriend



Переведено с английского языка. - В модном дизайне, в первую очередь в линиях прет-а-порте, бойфрендом является любой стиль женской одежды, который был модифицирован из соответствующей мужской одежды.

[Википедия \(Английский язык\)](#)

Оригинал описания >

Отзыв

Реверсивный поиск изображений

Основными целями использования такого поиска являются:

- поиск владельца или источника изображения;
- поиск такого же изображения, но в лучшем качестве;
- поиск персоны на изображении;
- поиск ресурсов, на которых опубликовано изображение.

Сервисы для реверсивного поиска

- **Google** – не первая компания, предложившая открытый универсальный реверсивный поиск изображений.
- До запуска гугловского проекта лидерами в этой области были, пожалуй, онлайн-сервисы канадской компании **Idee**.
 - Базовая технология данной службы – поисковая машина Piximilar, на базе которой и разрабатываются конкретные продукты. Соответствующий API на коммерческих условиях предлагается и для сторонних проектов.
- Лучшими универсальными машинами для реверсивного поиска изображений на сегодняшний день являются сервисы
 - Google Images
 - Google Lens
 - Bing
 - Яндекс изображения

Системы поиска изображений по содержанию

- Порядка 50 (подробный список) \approx 25 коммерческие + 25 открытые
- http://en.wikipedia.org/wiki/List_of_CBIR_Engines

Алгоритмы

- Масштабно-инвариантное преобразование признаков - для извлечения локальных особенностей изображения (SIFT)
- Максимально устойчивые экстремальные области – для находить соответствия между элементами изображения из двух изображений с разных точек зрения (MSER)
- Дерево словарного запаса – для классификации или поиска изображений (мешок слов)

Потенциальные области применения алгоритмов поиска по содержанию:

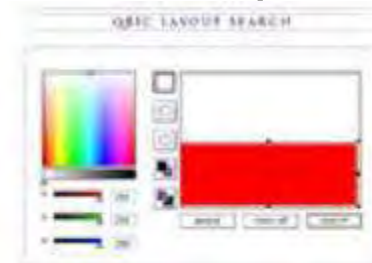
- Поиск изображений в сети интернет
- Каталогизация изображений произведений искусства
- Организация работы с архивами фотографических снимков
- Организация каталогов розничной продажи товаров
- Медицинская диагностика заболеваний
- Предотвращение преступлений и беспорядков
- Военно-оружейное применение
- Вопросы контроля за распространением интеллектуальной собственности
- Получение информации о местоположении удаленных зондов и географическое позиционирование
- Контроль за содержимым массивов изображений

Что запрашивает пользователь?

- Запрос в виде атрибутов/текстового описания изображения



- Запрос в виде некоторой характеристики содержимого
Гистограмма цветов

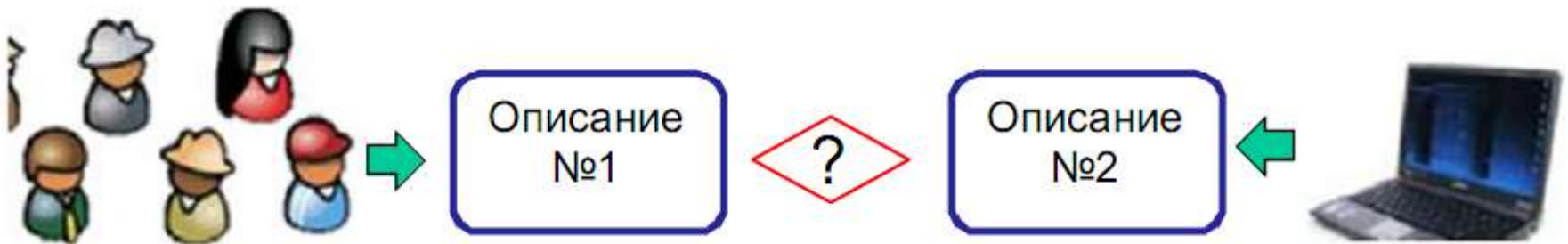


- Запрос в виде рисунка-наброска



Semantic Gap – Семантический Разрыв

- Запрос в виде изображения-примера («найди то же самое», «найди похожее изображение»)
- Что имел пользователь в виду?
- Что значит «похожее изображение»?
- «Семантический разрыв» – несовпадение информации, которую можно извлечь из визуальных данных, и интерпретацией тех же самых данных со стороны пользователя



Что значит похожее?

- Похожее по каким-то характеристикам, например, по цвету

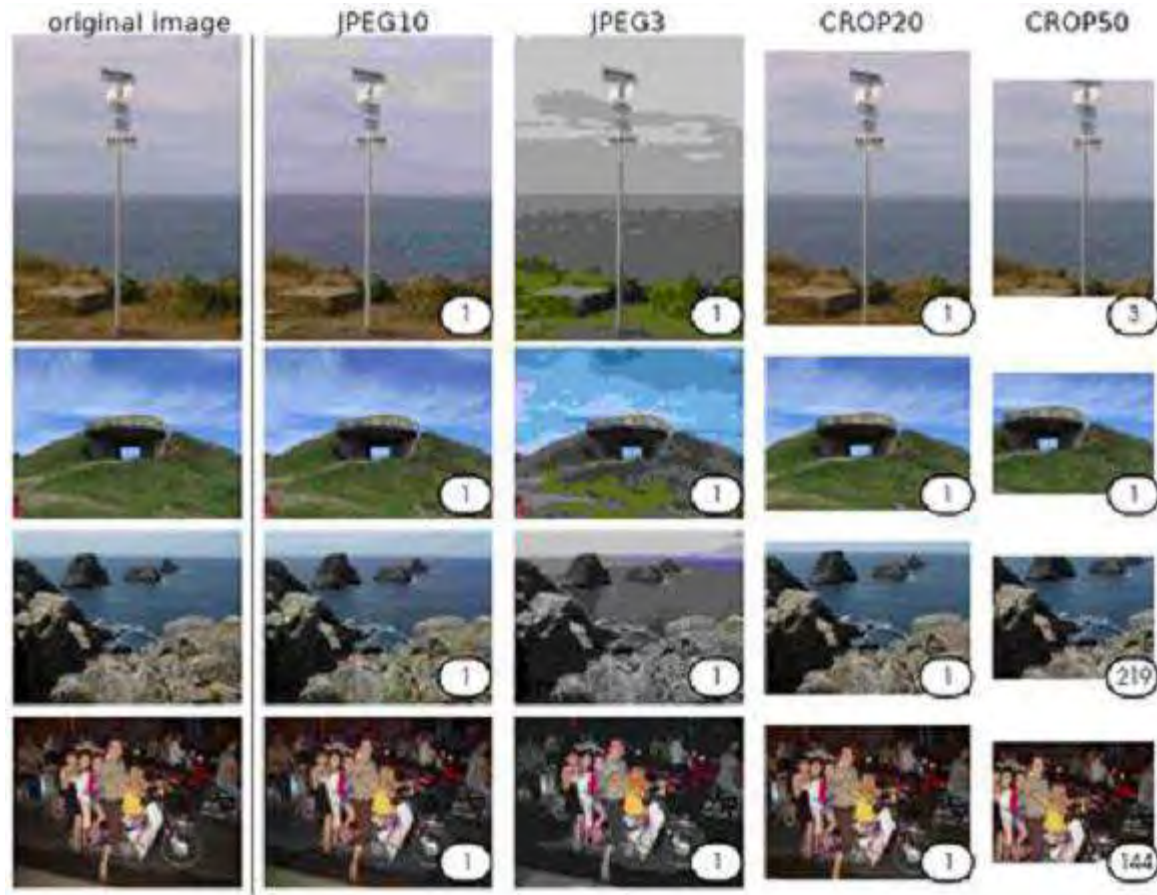


Запрос



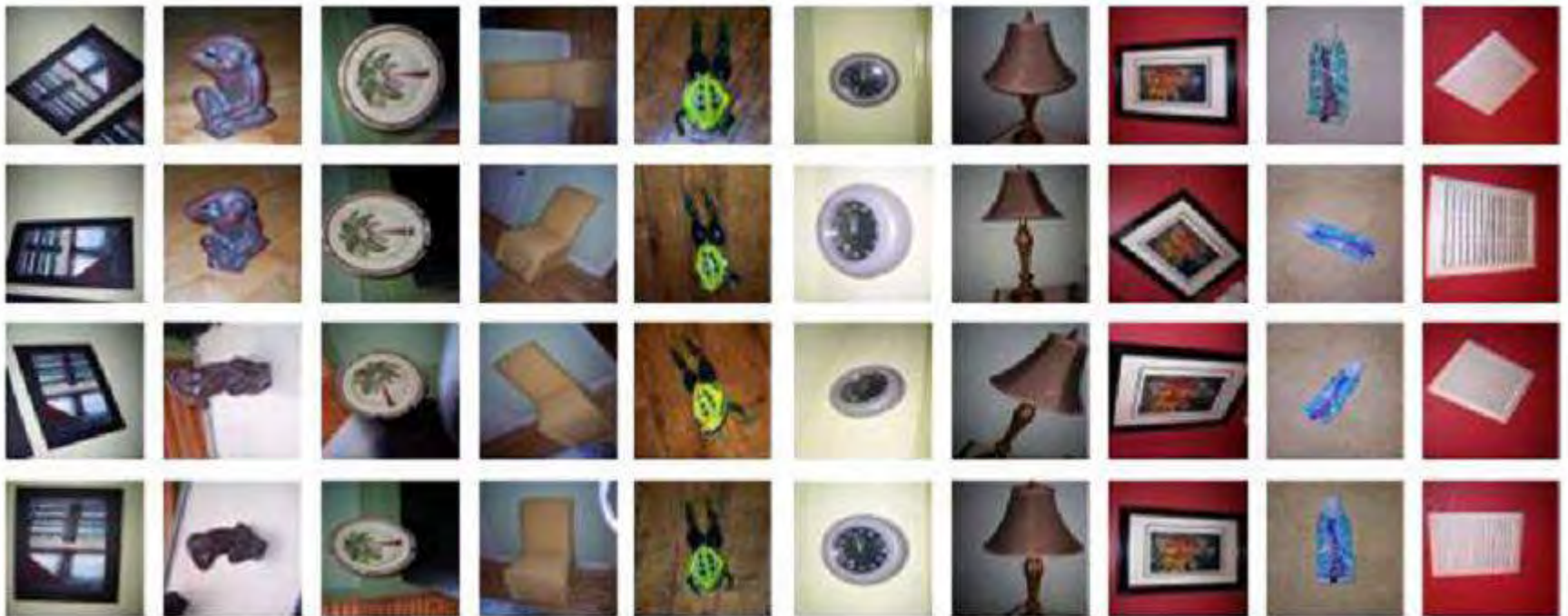
Что значит похожее?

- Полудубликаты (Near-duplicates) – слегка измененная версия изображения (ракурс, цвета)



Что значит похожее?

- Тот же самый объект или сцена («Object retrieval») Большие вариации ракурсов, фона, и т.д., чем при поиске полудубликатов



Что значит похожее?

- Похожие визуально по геометрии сцены с учетом ракурса (могут быть разные по назначению)



Кухня



Приемная



Бар



Автобус



Самолет



Зал

Что значит похожее?

- «Category-level classification» - изображения одного класса



Пример – банкетный зал.



Например, 256 классов из
базы Caltech 256

Анализ постановок задач



Визуальное подобие



Семантическое подобие

Общая схема поиска изображений



Тестовое изображение



Тестирование



Ищем ближайших соседей по выбранной метрике

Какие картинки более схожи между собой?

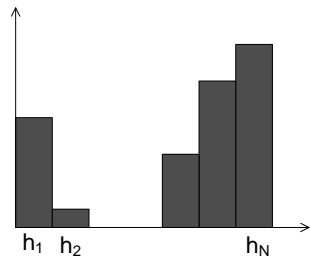


Характеристики сходства изображений

Основные группы:

1. Цветовое сходство
2. Текстурное сходство
3. Сходство формы
4. Дескрипторы изображений
5. Сходство объектов и отношений между объектами

Цвет



Мат. ожидание, дисперсия, 3-й момент:
для каждого цветового канала

$$F(I) = (h_1^I, h_2^I, \dots, h_N^I)$$

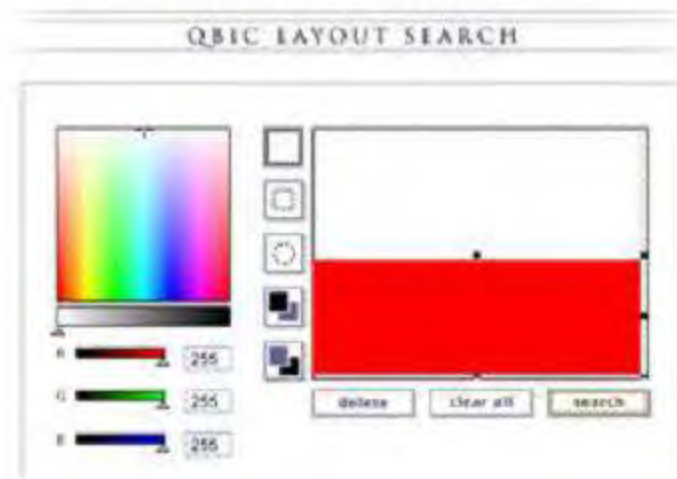
Метрики: L_1, L_2, L_∞

$$F(I) = (E_1^I, E_2^I, E_3^I, \\ \sigma_1^I, \sigma_2^I, \sigma_3^I, \\ s_1^I, s_2^I, s_3^I)$$

Метрики: $\sim L_1$

Stricker M., Orengo M. Similarity of Color Images. Proceedings of the SPIE Conference, vol. 2420, p. 381-392, 1995

QVICS: Пример использования



1) Vase of Flowers

Huijsum, Jan van Early 18th century



2) Seascape with Venice in the Distance

Cottet, Charles Circa 1896



3) Boats on a Sea Shore

Goyen, Jan Jozefsz van 1641



4) Avenue in a Park

Watteau, Antoine Circa 1715



5) Bird Perching on a Rose Twig

UNKNOWN 18th century



6) Old Woman with a Spindle

Watteau, Antoine 1710s



7) Interiors of the New Hermitage, The Room of Russian Sculpture

Premazzi, Luigi 1854

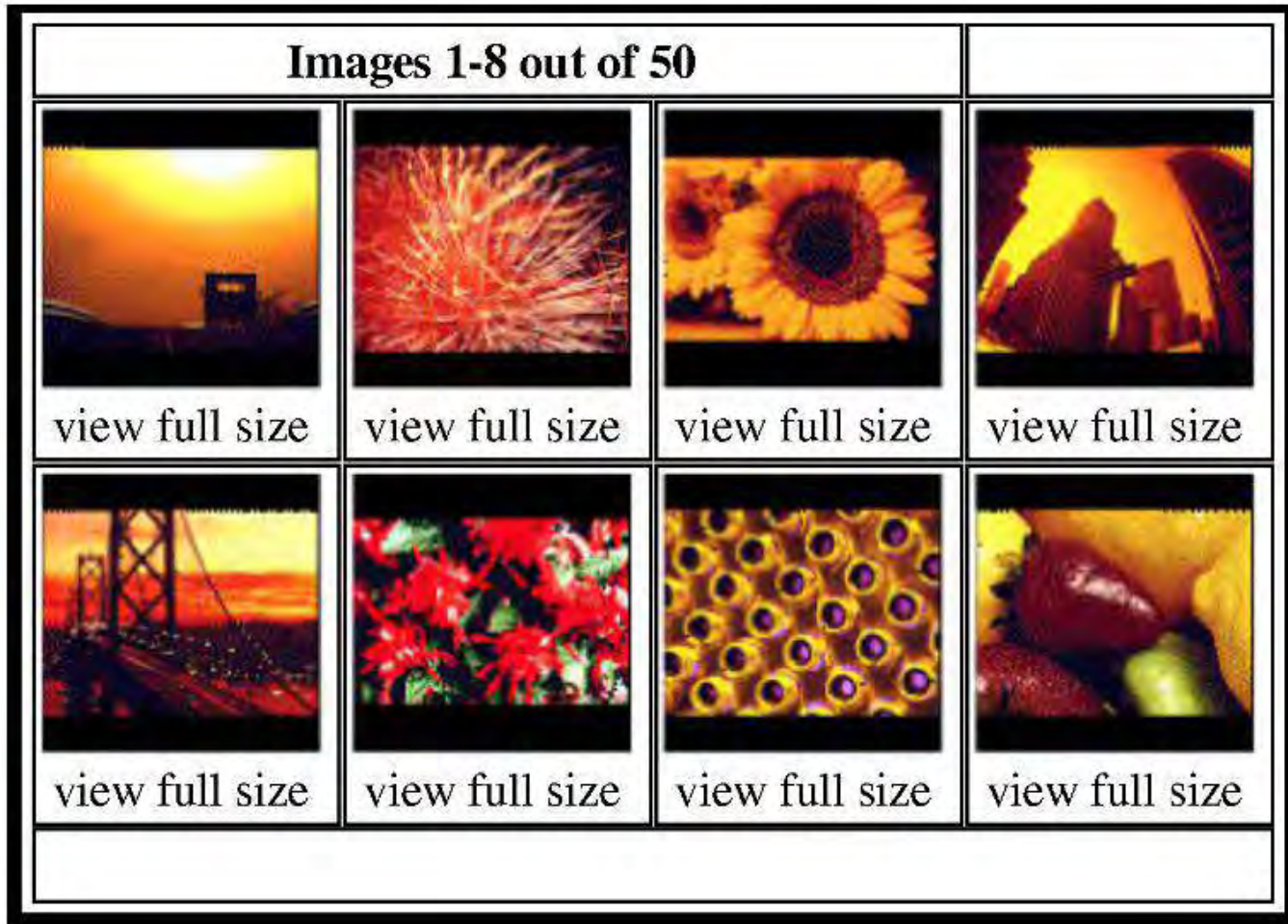


8) Allegory of George I, King of England

Vanloo, Carle (Charles-Andre) 1736

Качество работы системы понятно. Но первая!

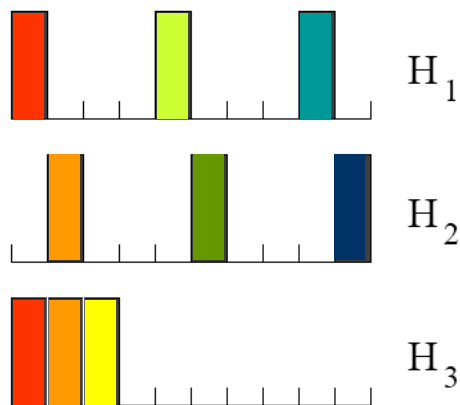
ЦВЕТОВОЕ СХОДСТВО



Результат выполнения запроса системой QBIC (1995):
40% красного, 30% жёлтого, 10% чёрного

Цветовые гистограммы – недостатки

1. Не учитывается схожесть цветов:



$$d(H_1, H_2) > d(H_1, H_3)$$

$$d(H_1, H_2) = \sqrt{(H_1 - H_2) \cdot A \cdot (H_1 - H_2)^T}$$

A – матрица с коэффициентами
«схожести» цветов

Niblack W., Barber R., et al. The QBIC project: Querying images by content using color, texture and shape. In IS&T/SPIE International Symposium on Electronic Imaging: Science & Technology, Conference 1908, Storage and Retrieval for Image and Video Databases, Feb. 1993

QBIC (1995)

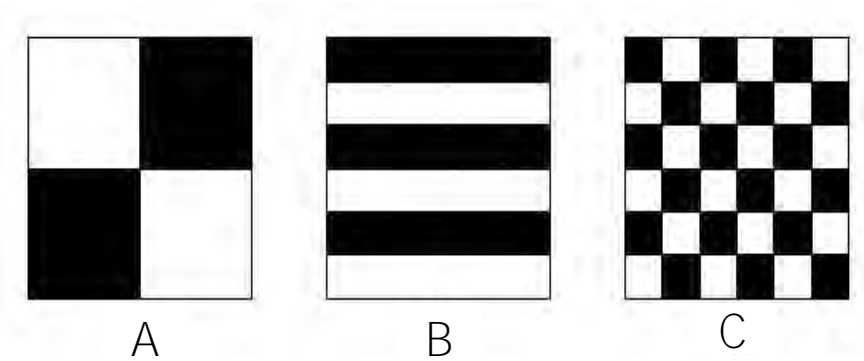
- «Query By Image Content»
- Самая первая система CBIR
- Изображения сравниваются по набору признаков
 - Цветовая гистограмма
 - Выделенные вручную объекты и признаки их формы (размер, площадь, количество)
- ~10000 изображений в базе

$$d_{hist}(I, Q) = (h(I) - h(Q))^T A(h(I) - h(Q))$$

M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: the QBIC system. IEEE Computer, 28(9):23–32, 1995.

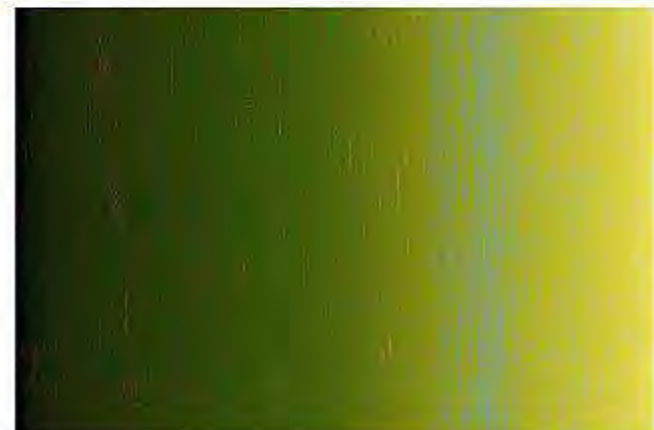
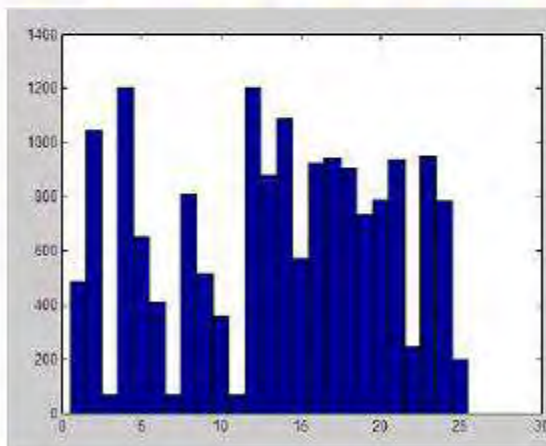
Цветовые гистограммы – недостатки

2. Не учитывается пространственное расположение цветов:

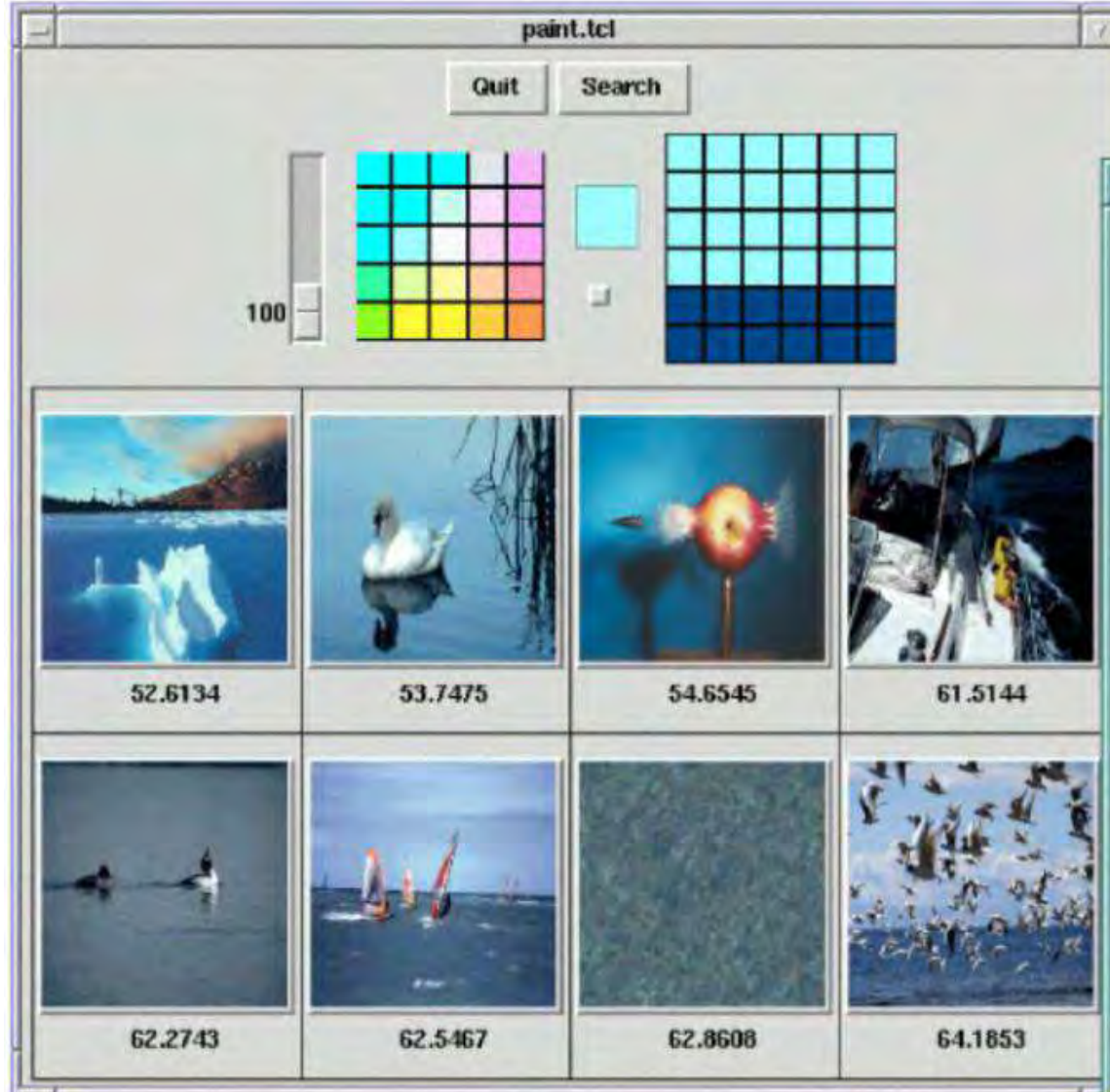


$$H_A = H_B = H_C$$

Чья гистограмма?



Запрос на основе закрашенной сетки











Мера расстояния на основе цветового макета (color layout) для закрашенной сетки

$$d_{gridded_color}(I, Q) = \sum_g \hat{d}_{color}(C^I(g), C^Q(g))$$

Способы представления цветов клеток сетки

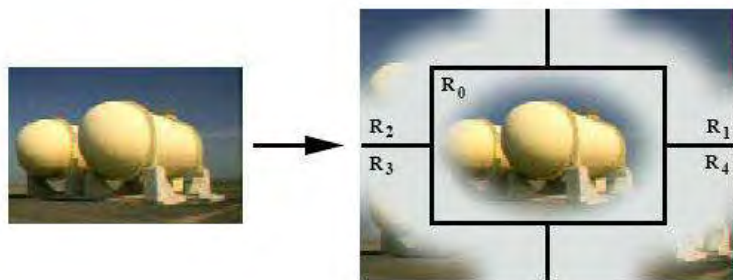
- Среднее значение цвета в пределах клетки сетки
- Среднее значение и среднеквадратическое отклонение цвета
- Многоразрядная цветовая гистограмма

Результаты поиска, выполненного системой QVIS с использованием меры расстояний на основе цветового макета

Images 1-8 out of 41				
				
view full size	view full size	view full size	view full size	
				
view full size	view full size	view full size	view full size	
Columns:				Rows:

Пространственное расположение цветов

- Разбиение изображения на фиксированные блоки
- «Нечеткие области»

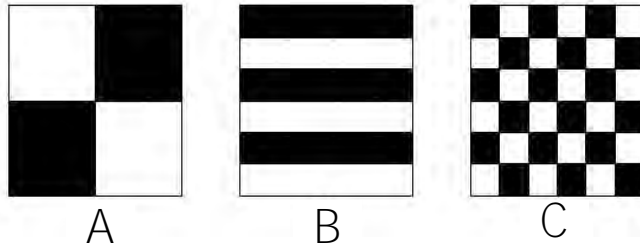


Stricker M., Dimai A. Spectral Covariance and Fuzzy Regions for Image Indexing. Machine Vision and Applications, vol. 10., p. 66-73, 1997

- Сегментация



Ещё одно решение проблемы



$$H_A = H_B = H_C$$



img239

```

{160, 160, 160, 0.2425520833333333,77,99};
{224, 224, 224, 0.2184895833333333,60,65};
{32, 32, 96, 0.2080208333333333,81,13};
{32, 96, 160, 0.1088541666666666,78,41};
{32, 96, 96, 0.0799479166666666,78,30};
{96, 96, 96, 0.0706770833333333,120,78};
{160, 160, 224, 0.0302604166666666,64,87};
{32, 32, 32, 0.0139583333333333,126,83};
{160, 224, 224, 0.0107291666666666,67,77};
    
```

$$f_i^A = (a_i, b_i, c_i, weight_i^A, x_i^A, y_i^A)$$

$i = 1..N$ – число цветов;

(a_i, b_i, c_i) – параметры цвета i ;

$weight_i$ – количество цвета i на изображении A ;

(x_i, y_i) – координаты центра цветового пятна.

Васильева Н., Новиков Б. Построение соответствий между низкоуровневыми характеристиками и семантикой статических изображений. Труды RCDL'2005.

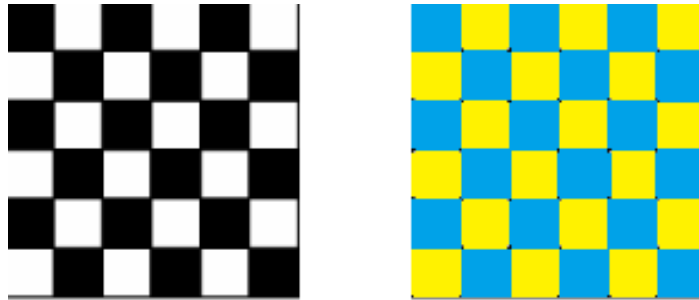
Характеристики сходства изображений

Основные группы:

1. Цветовое сходство
2. Текстурное сходство
3. Сходство формы
4. Дескрипторы изображений
5. Сходство объектов и отношений между объектами

Текстурное сходство

- Одинаковое пространственное распределение цветов
- Значения цветов изображений могут не совпадать



Рассматриваемые характеристики

- Представление текстуры -- Вектор текстурного описания
 - **Пятикомпонентный вектор Харалика (Энергия, Энтропия, Контраст, Однородность, Корреляция)**
 - Десятикомпонентный вектор, соответствующий энергетическим текстурным характеристикам Лавса
- Определение сходства с учетом выбранного представления текстуры

Матрицы вхождений (смежности) co-occurrence matrix

$C(i,j)$ – элемент матрицы – сколько раз встречалось i в отношении с j

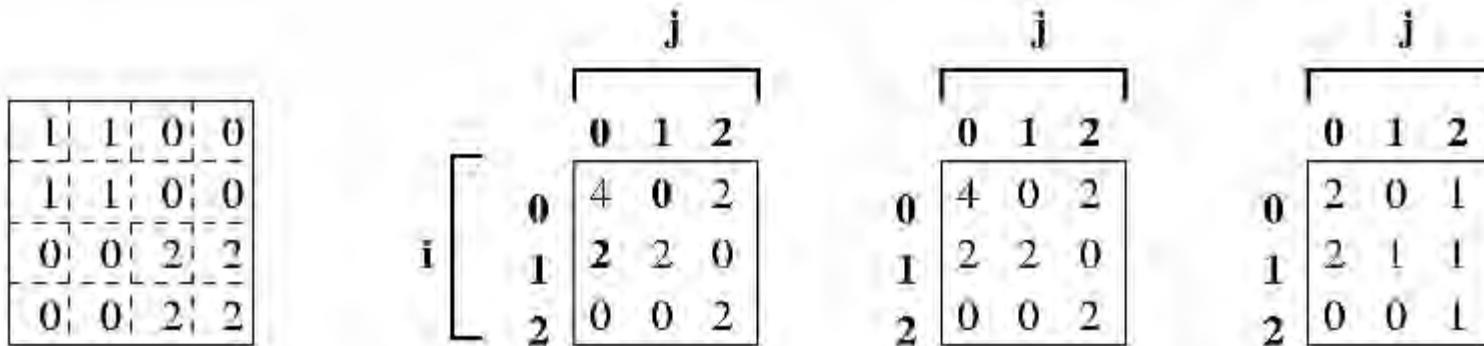
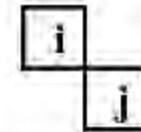
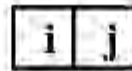


Image I

$C_{(0,1)}$

$C_{(1,0)}$

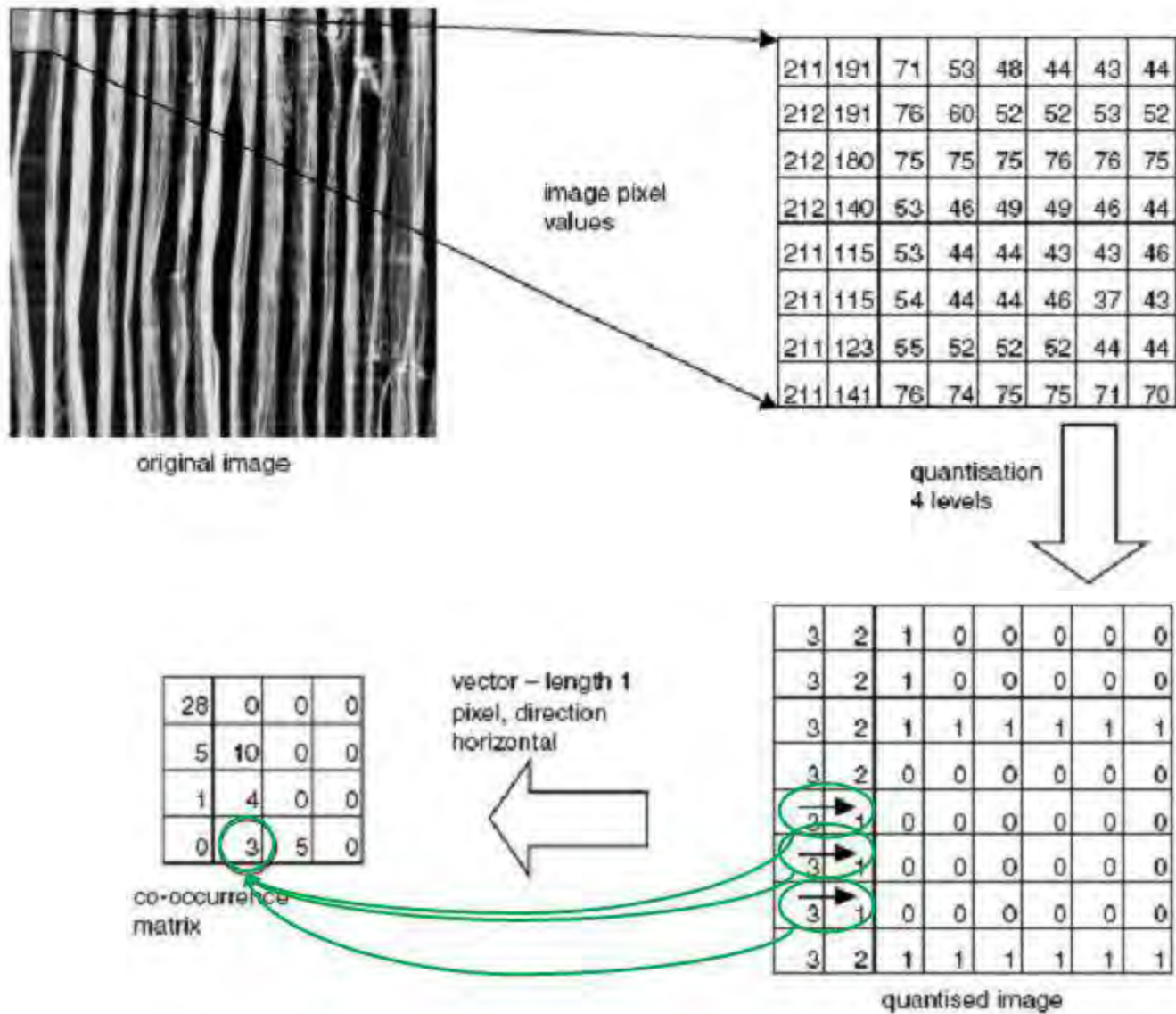
$C_{(1,1)}$



Полутоновая матрица вхождений

$$C(i, j) = \sum_{p=1}^N \sum_{q=1}^M \begin{cases} 1, & \text{если } I(p, q) = i, I(p + \Delta x, q + \Delta y) = j \\ 0, & \text{иначе} \end{cases}$$

$d=(\Delta x, \Delta y)$ – вектор пространственного отношения



Основные разновидности стандартной полутоновой матрицы вхождений Gray Level Co-occurrence Matrices (GLCM)

- Нормированная полутоновая матрица вхождений

$$N_d(i, j) = \frac{C_d(i, j)}{\sum_i \sum_j C_d(i, j)}$$

- Симметричная полутоновая матрица вхождений

$$S_d(i, j) = C_d(i, j) + C_{-d}(i, j)$$

Матрицы вхождений

- представляют свойства текстуры
- но не удобны для непосредственного применения при анализе изображений
- используются для вычисления числовых характерных признаков для более компактного представления текстуры

Матрицы смежности: характеристики Харалика

Статистические параметры, вычисленные по матрицам:

$$Energy = \sum_i \sum_j C^2(i, j) \quad - \text{минимален, когда все элементы равны}$$

$$Entropy = - \sum_i \sum_j C(i, j) \log_2 C(i, j) \quad - \text{мера хаотичности, максимален, когда все элементы равны}$$

$$Contrast = \sum_i \sum_j (i - j)^2 C(i, j) \quad - \text{мал, когда большие элементы вблизи главной диагонали}$$

$$Inverse \text{ Difference Moment} = \sum_i \sum_j \frac{C(i, j)}{1 + (i - j)^2} \quad - \text{мал, когда большие элементы далеки от главной диагонали}$$

$$Correlation = \frac{\sum_i \sum_j (i - \mu_i)(j - \mu_j) N_d(i, j)}{\sigma_i \sigma_j}$$

Проблема

- Выбор вектора смещения d
- Zucker, Terzopoulos (1980) предложили использовать статистическую проверку χ^2 для выбора значений d

$$\max \chi^2(d) = \left(\sum_i \sum_j \frac{N_d^2(i, j)}{N_d(i)N_d(j)} - 1 \right)$$

$$N_d(i) = \sum_j N_d(i, j)$$

$$N_d(j) = \sum_i N_d(i, j)$$

Рассматриваемые характеристики

- Представление текстуры -- Вектор текстурного описания
 - Пятикомпонентный вектор Харалика (Энергия, Энтропия, Контраст, Однородность, Корреляция)
 - **Девятикомпонентный вектор, соответствующий энергетическим текстурным характеристикам Лавса**
- Определение сходства с учетом выбранного представления текстуры

Энергетические текстурные характеристики Лавса (Laws)

- Обнаружение различных типов текстур с помощью локальных масок
- Энергетический подход, в котором оценивается изменение содержания текстуры в пределах окна фиксированного размера
- Для вычисления энергетических характеристик используется набор из девяти масок 5×5
- Энергетические характеристики каждого пиксела представляются вектором из 9 чисел

Векторы для вычисления масок

Для вычисления симметричного
взвешенного локального среднего
значения

$$L5 \text{ (Level)} : [1 \ 4 \ 6 \ 4 \ 1]$$

Для обнаружения краев

$$E5 \text{ (Edge)} : [-1 \ -2 \ 0 \ 2 \ 1]$$

Для обнаружения пятен

$$S5 \text{ (Spot)} : [-1 \ 0 \ 2 \ 0 \ -1]$$

Для обнаружения образа в виде ряби

$$R5 \text{ (Ripple)} : [1 \ -4 \ 6 \ -4 \ 1]$$

Получение двумерных масок

$$E5 \times L5 = E5L5$$

$$\begin{bmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 1 \end{bmatrix} \times [1 \ 4 \ 6 \ 4 \ 1] = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Метод Лавса (шаг 1)

Устранение влияния интенсивности освещения

По изображению перемещается окно и из каждого пиксела вычитается локальное среднее значение.

Формируется изображение, на котором средняя интенсивность каждой пиксельной окрестности близка к 0.

Размер окна зависит от типа изображений.

Для естественных сцен окно – 15x15

Метод Лавса (шаг 2)

Применение каждой из 16 масок 5x5

Формируются профильтрованные изображения

Энергетическая текстурная карта –

полноразмерное изображение

$$E_k[r, c] = \sum_{j=c-7}^{c+7} \sum_{i=r-7}^{r+7} |F_k[i, j]|$$

$F_k [i, j]$ – результат фильтрации пиксела k -ой маской

Метод Лавса (шаг 3)

Каждая симметричная пара карт заменяется усредненной картой

Например, можно усреднить:

E5L5 – горизонтальные края

L5E5 – вертикальные края

Среднее – края обоих типов

16 карт → 9 карт

Девять окончательных энергетических карт

L5E5/E5L5

E5R5/R5E5

L5S5/S5L5

S5S5

L5R5/R5L5

S5R5/R5S5

E5E5

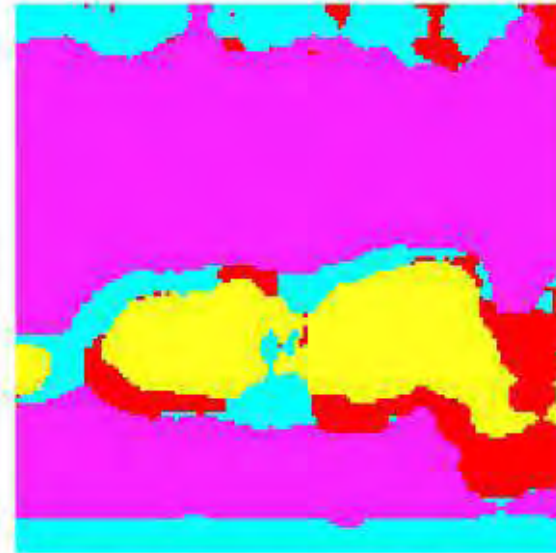
R5R5

E5S5/S5E5

Сегментация (характеристики Лавса)



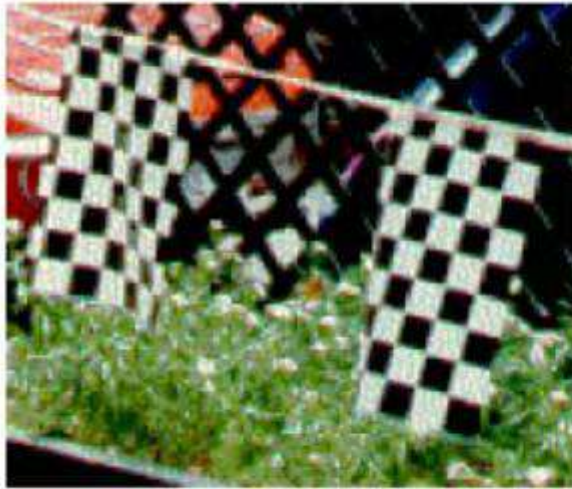
(a) Original image



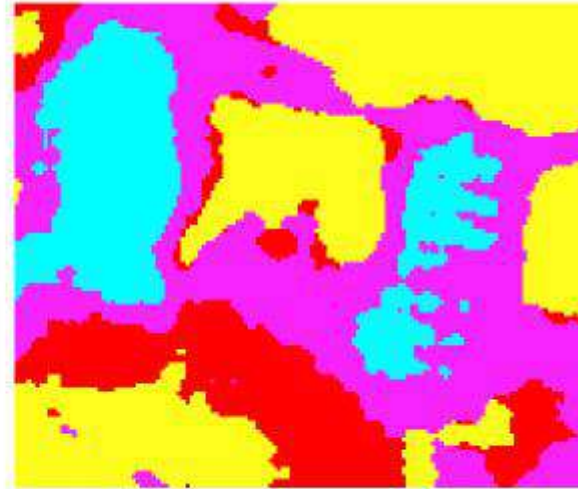
(b) Segmentation into 4 clusters

Region	E5E5	S5S5	R5R5	E5L5	S5L5	R5L5	S5E5	R5E5	R5S5
Tiger	168.1	84.0	807.7	553.7	354.4	910.6	116.3	339.2	257.4
Water	68.5	36.9	366.8	218.7	149.3	459.4	49.6	159.1	117.3
Flags	258.1	113.0	787.7	1057.6	702.2	2056.3	182.4	611.5	350.8
Fence	189.5	80.7	624.3	701.7	377.5	803.1	120.6	297.5	215.0
Grass	206.5	103.6	1031.7	625.2	428.3	1153.6	146.0	427.5	323.6
Small flowers	114.9	48.6	289.1	402.6	241.3	484.3	73.6	158.2	109.3
Big flowers	76.7	28.8	177.1	301.5	158.4	270.0	45.6	89.7	62.9
Borders	15.3	6.4	64.4	92.3	36.3	74.5	9.3	26.1	19.5

Сегментация (характеристики Лавса)



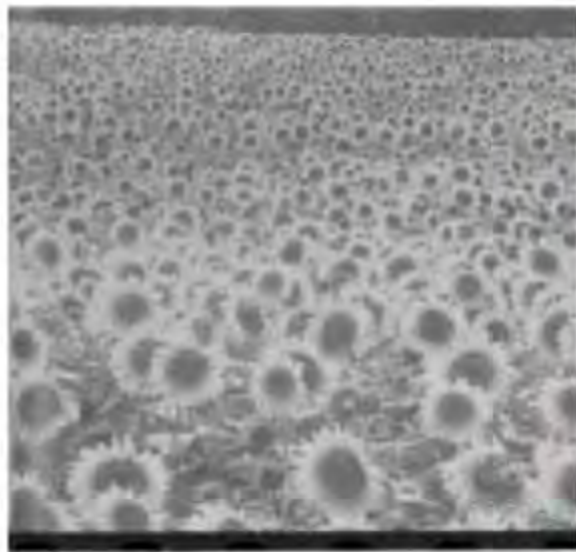
(c) Original image



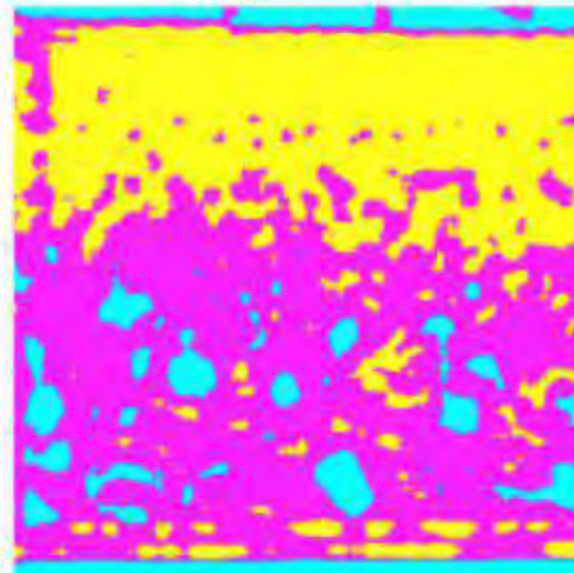
(d) Segmentation into 4 clusters

Region	E5E5	S5S5	R5R5	E5L5	S5L5	R5L5	S5E5	R5E5	R5S5
Tiger	168.1	84.0	807.7	553.7	354.4	910.6	116.3	339.2	257.4
Water	68.5	36.9	366.8	218.7	149.3	459.4	49.6	159.1	117.3
Flags	258.1	113.0	787.7	1057.6	702.2	2056.3	182.4	611.5	350.8
Fence	189.5	80.7	624.3	701.7	377.5	803.1	120.6	297.5	215.0
Grass	206.5	103.6	1031.7	625.2	428.3	1153.6	146.0	427.5	323.6
Small flowers	114.9	48.6	289.1	402.6	241.3	484.3	73.6	158.2	109.3
Big flowers	76.7	28.8	177.1	301.5	158.4	270.0	45.6	89.7	62.9
Borders	15.3	6.4	64.4	92.3	36.3	74.5	9.3	26.1	19.5

Сегментация (характеристики Лавса)



(e) Original image



(f) Segmentation into 3 clusters

Region	E5E5	S5S5	R5R5	E5L5	S5L5	R5L5	S5E5	R5E5	R5S5
Tiger	168.1	84.0	807.7	553.7	354.4	910.6	116.3	339.2	257.4
Water	68.5	36.9	366.8	218.7	149.3	459.4	49.6	159.1	117.3
Flags	258.1	113.0	787.7	1057.6	702.2	2056.3	182.4	611.5	350.8
Fence	189.5	80.7	624.3	701.7	377.5	803.1	120.6	297.5	215.0
Grass	206.5	103.6	1031.7	625.2	428.3	1153.6	146.0	427.5	323.6
Small flowers	114.9	48.6	289.1	402.6	241.3	484.3	73.6	158.2	109.3
Big flowers	76.7	28.8	177.1	301.5	158.4	270.0	45.6	89.7	62.9
Borders	15.3	6.4	64.4	92.3	36.3	74.5	9.3	26.1	19.5

Использование меры расстояния на основе энергетических текстурных характеристик Лавса

L5E5/E5L5

L5S5/S5L5

L5R5/R5L5

E5E5

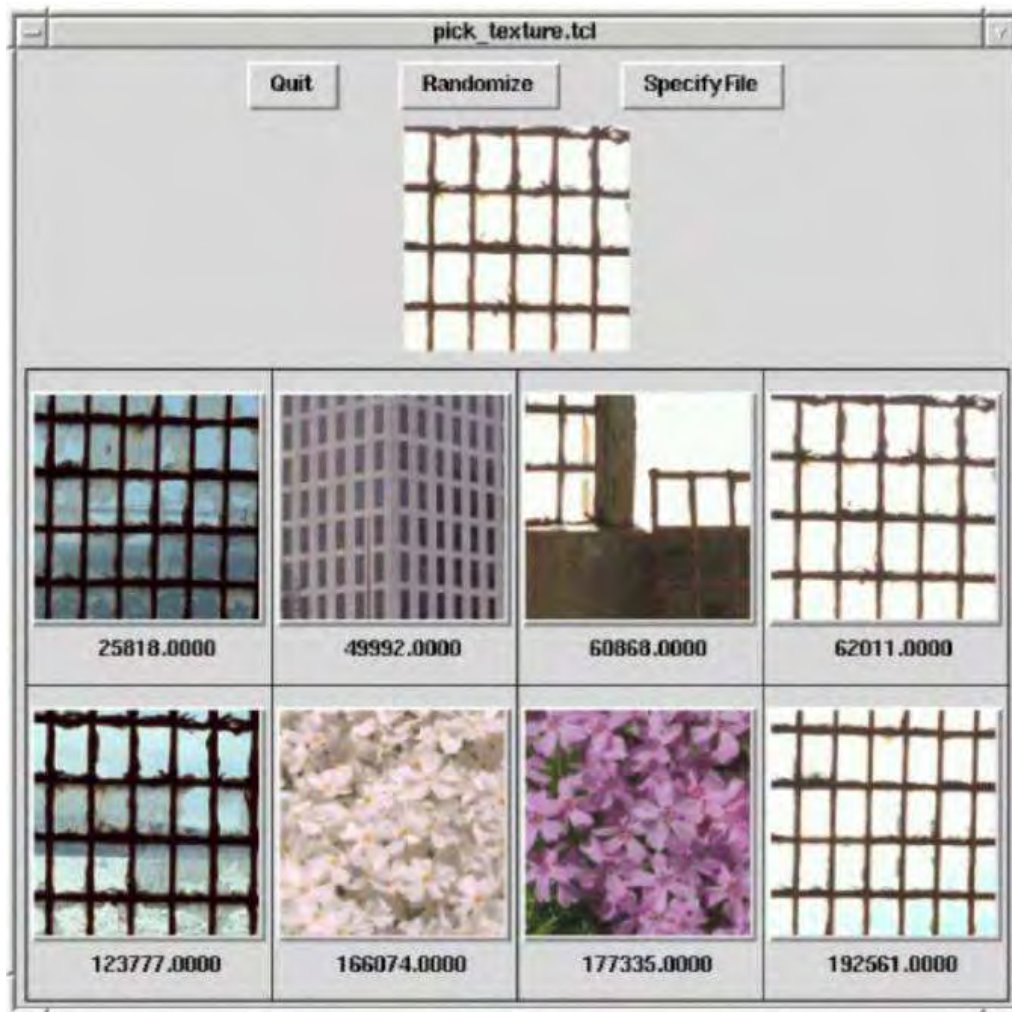
E5S5/S5E5

E5R5/R5E5

S5S5

S5R5/R5S5

R5R5



Рассматриваемые характеристики

- Представление текстуры -- Вектор текстурного описания
 - Пятикомпонентный вектор Харалика (Энергия, Энтропия, Контраст, Однородность, Корреляция)
 - **Девятикомпонентный вектор, соответствующий энергетическим текстурным характеристикам Лавса**
- Определение сходства с учетом выбранного представления текстуры

Признаки Тамуры

Имеются 6 признаков, признанных существенными для зрительного восприятия в результате психологического эксперимента

Все они вычисляются по эвристическим процедурам и никак не масштабируются.

Tamura, H. Textural Features Corresponding to Visual Perception / H. Tamura, Sh. Mori, T. Yamawaki // IEEE Transactions on Systems, Man, and Cybernetics. – 1978. – June, Vol. SMC-8. – P. 460-473.

Признаки Тамуры

- **Зернистость** – это признак, связанный с расстоянием между заметными пространственными колебаниями оттенков серого, то есть с размером примитивных элементов (текселей), формирующих текстуру.
- **Контраст Тамуры** – это мера того, насколько сильно и резко может меняться цвет на изображении.
- **Направленность** – это признак, измеряемый с помощью гистограммы локальных направлений контуров.

Признаки Тамуры

- **Линейность** – это признак, показывающий, насколько прямолинейны контуры на изображении.
- **Регулярность** – это общая изменчивость первых четырёх признаков между различными частями изображения.
- **Грубость** – субъективная оценка грубости переходов на изображении

Признаки Tamura

Характеристики, существенные для зрительного восприятия:

- Зернистость (coarseness)
- Контрастность (contrast)
- Направленность (directionality)
- Линейность (line-likeness)
- Регулярность (regularity)
- Грубость (roughness)

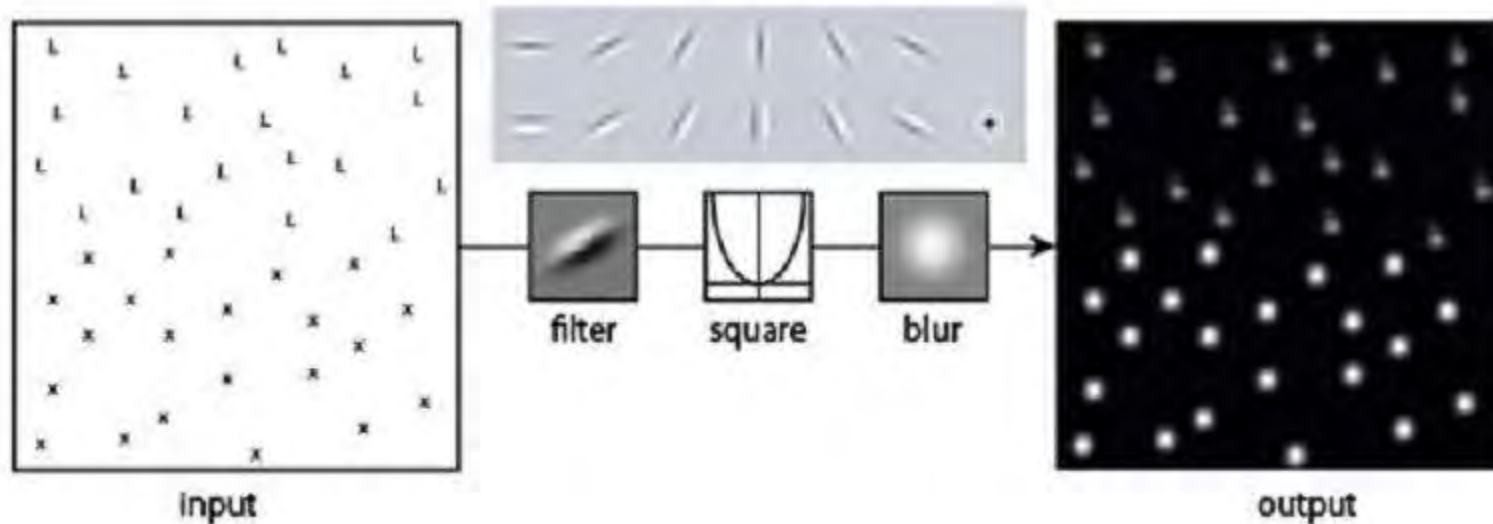
Tamura image:

Coarseness-coNtrast-Directionality –
точки в трехмерном пространстве CND

Признаки:

- Евклидово расстояние в 3D (QBIC)
- 3D гистограмма (Mars)

Банки текстурных фильтров



- Выберем набор (банк) фильтров, каждый из которых чувствителен к краю определенной ориентации и размера
- Каждый пиксель изображения после обработки банком фильтров даёт вектор признаков
- Этот вектор признаков эффективно описывает локальную текстуру окрестности пикселя

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

Банки текстурных фильтров

- **Фильтры Габора**
- Признаки Хаара

Фильтры Габора

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

λ — длина волны

θ — ориентация

σ — сигма гауссиана

ψ — сдвиг фазы

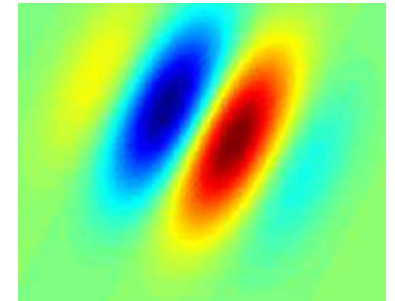
γ — коэффициент сжатия

или соотношение размеров
(aspect ratio),

«эллиптичность фильтра»

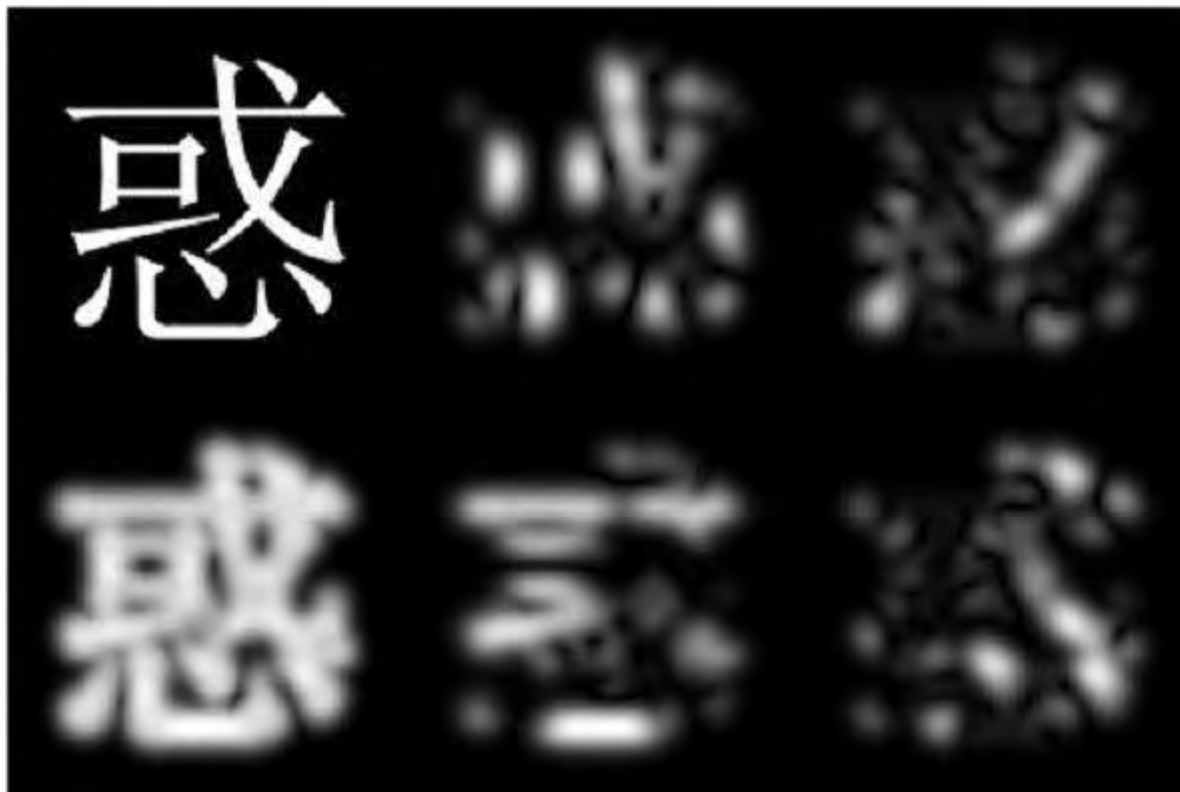
$$x' = x \cos(\theta) + y \sin(\theta)$$

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



- 2D фильтр Габора – ядро гауссиана, домноженное на синусоиду
- Предложены в 1947 Денисом Габором (нобелевским лауреатом), независимо переоткрыты в 1980 году
- Позволяет сделать банк фильтров, для выделения краёв разной ориентации, масштаба и положения в окрестности

Поиск краёв с помощью Габора



- Слева сверху – иероглиф
- 4 справа – применение фильтров Габора с ориентациями 0, 45, 90 и 135 градусов
- Слева внизу – совмещение результатов фильтрации

Фильтр Габора эффективен при обработке изображений со структурной избыточностью, имеющих квазипериодическую структуру.

Дактилоскопические изображения, изображения кристаллограмм и интерферрограмм.

Окрас у зебр, различных кошачьих (тигры, рыси, дымчатые коты), птиц (тетерева), рыб (лепоринус полосатый).

Исходное
дактилоскопическое
изображение

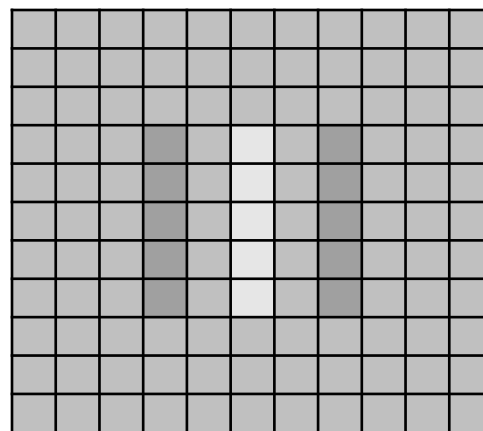
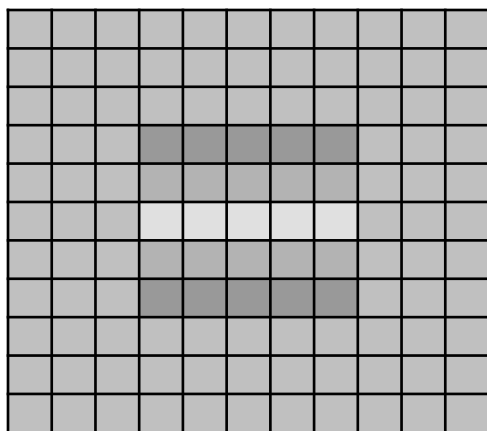
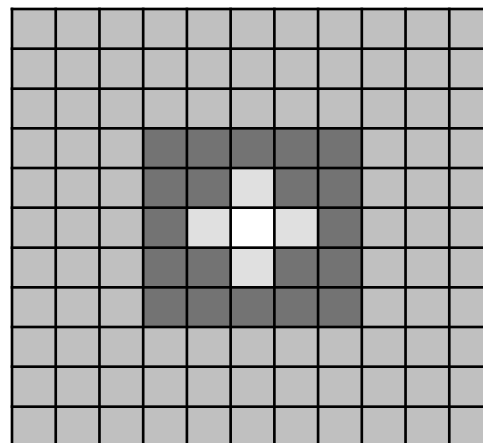
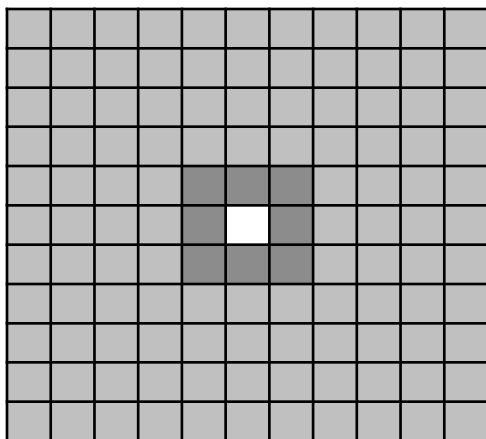


Дактилоскопическое
изображение,
обработанное
фильтром Габора

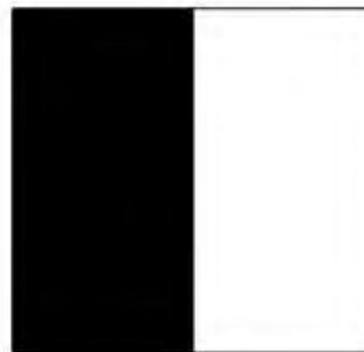
Банки текстурных фильтров

- Фильтры Габора
- **Признаки Хаара**

Признаки Хаара



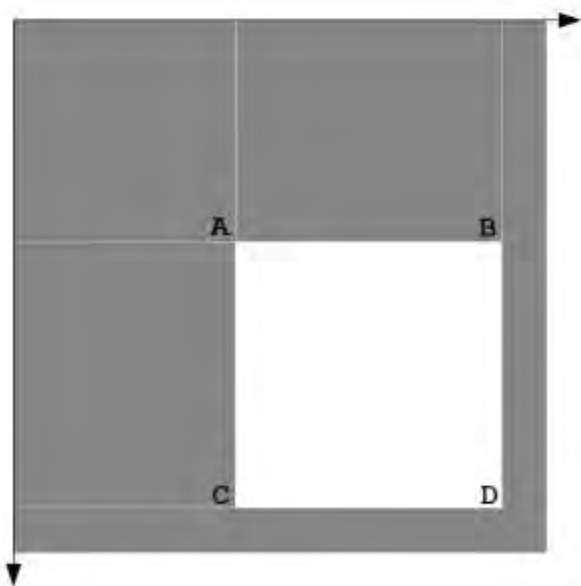
Признаки Хаара



Интегральное изменение яркости

Интегральное представление изображения

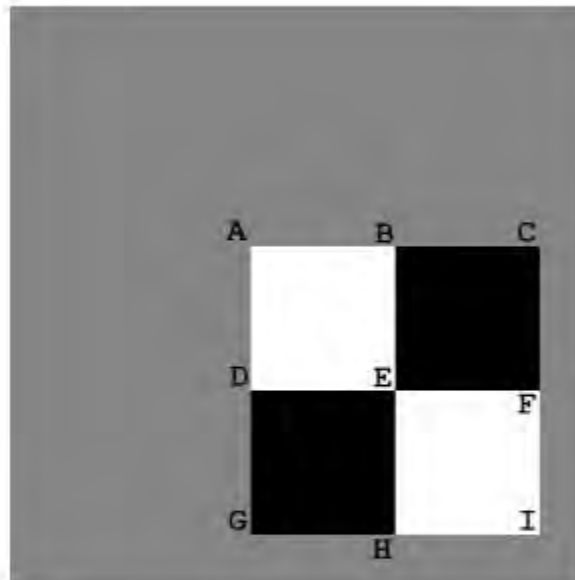
$$L(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j)$$



$$I_{ABCD} = L_D - L_B - L_C + L_A$$

$$L(x, y) = I(x, y) - L(x-1, y-1) + L(x, y-1) + L(x-1, y)$$

Интегральное представление



$$S_{ACGI} = L_A - 2L_B + L_C - 2L_D + 4L_E - 2L_F + L_G - 2L_H + L_I$$

Меры текстурного сходства

- Расстояние от выбранного образца

$$d_{pick_and_click}(I, Q) = \min_{i \in I} \|T(i) - T(Q)\|^2$$

- Обобщение на случай текстур характеристик на основе закрашенных сеток

$$d_{gridded_texture}(I, Q) = \sum_g \hat{d}_{texture}(T^I(g), T^Q(g))$$

- Расстояние между текстурными гистограммами

Характеристики сходства изображений

Основные группы:

1. Цветовое сходство
2. Текстурное сходство
3. Сходство формы
4. Дескрипторы изображений
5. Сходство объектов и отношений между объектами

Требования к признакам формы

- Инвариантность к параллельному переносу
- Инвариантность к изменению масштаба
- Инвариантность к повороту
- Устойчивость к незначительным изменениям формы
- Простота вычисления
- Простота сравнения

Сходство формы

- Сравнение двух областей:
 - Гистограммы формы
 - Методы сопоставления границ
 - Эскизное сопоставление
 - С помощью методов статистического распознавания образов

Гистограмма формы

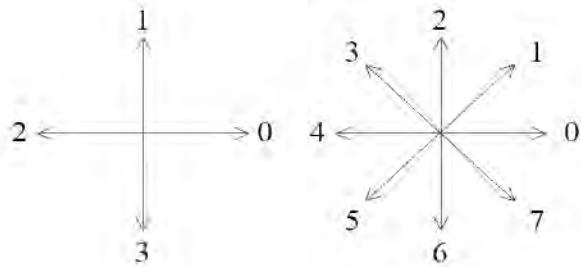
- Способы построения:
 - Проекционное сопоставление
 - По значению угла наклона касательной в каждом граничном пикселе области

Сопоставление границ

- Представление границ:
 - В виде последовательности символов
 - В виде многоугольника методом полигональной аппроксимации
 - Эластичное сопоставление

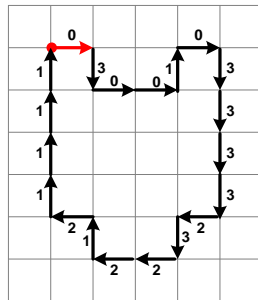
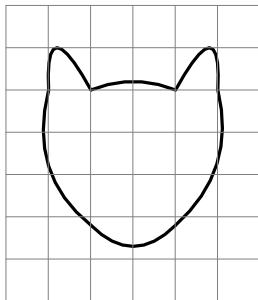
Цепные коды

Нумерация направлений для 4-связного и 8-связного цепных кодов:

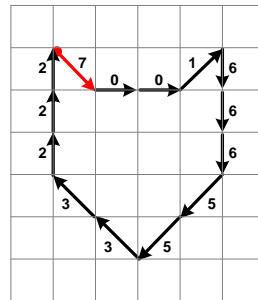


А: 03001033332322121111
 Б: 70016665533222

Пример:



А



Б



Инвариантность к выбору начальной точки: минимальный код



Инвариантность к повороту: разности цифр кода

Представление границ в виде последовательности символов (Коэффициенты Фурье)

$\langle \tilde{V}_0, V_1, \dots, \tilde{V}_{m-1} \rangle$ Набор точек

$$v_k = \frac{V_{k+1} - V_k}{|V_{k+1} - V_k|} \quad \text{Единичные вектора}$$

$$l_k = \sum_{i=1}^k |V_i - V_{i-1}|, \quad k > 0 \quad \text{Кумулятивные разности}$$

$$l_0 = 0$$

$$a_n = \frac{1}{L \left(\frac{n2\pi}{L}\right)^2} \sum_{k=1}^m (v_{k-1} - v_k) e^{-jn(2\pi/L)l_k}$$

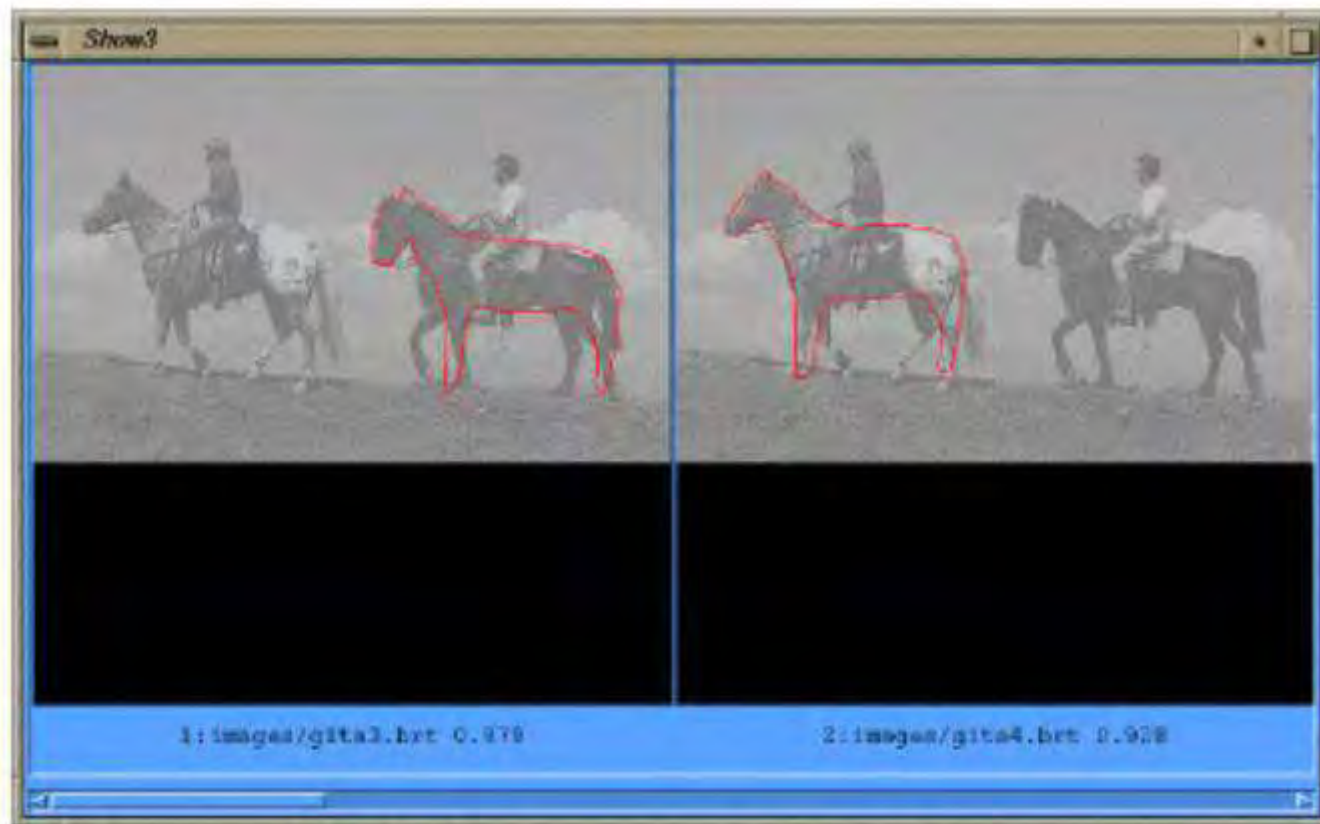
$$d_{\text{Fourier}}(I, Q) = \left[\sum_{n=-M}^M |a_n^I - a_n^Q|^2 \right]^{\frac{1}{2}}$$

В виде многоугольника методом полигональной аппроксимации

- Форма представляется в виде последовательности областей стыка

$$\langle X_i, Y_i, \alpha_i \rangle$$

Эластичное сопоставление



Описание формы на основе контура

Эластичное сопоставление



Эскизное сопоставление. Система ART MUSEUM

Предварительная обработка:

1. Уменьшение изображения до заданного размера и удаление шумов с помощью медианного фильтра
2. Обнаружение границ. Во-первых, с помощью глобального порога, затем с помощью локального порога. В результате получается очищенное контурное изображение.
3. На очищенном контурном изображении производится скелетизация и удаляются избыточные контуры. Полученное изображение ещё раз очищается от шумов, и мы получаем требуемое абстрактное представление.

Над эскизом производятся такие же операции обработки.

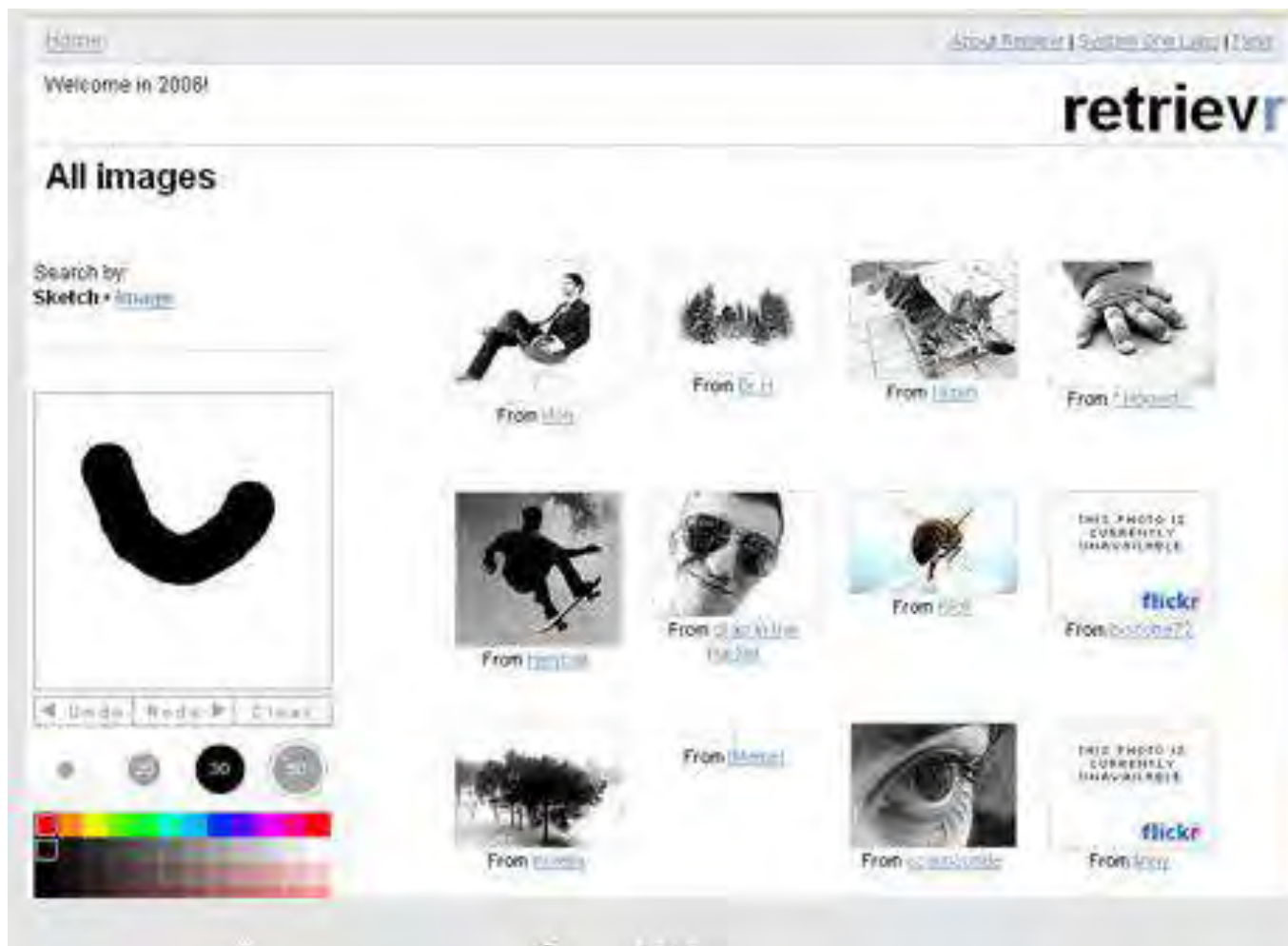
Эскизное сопоставление. Система ART MUSEUM

изображение делится на клетки, и для каждой клетки вычисляется корреляция с аналогичной клеткой изображения из базы данных

процедура выполняется несколько раз для разных значений сдвига линейного эскиза

Мера расстояния для эскизного сопоставления

$$d_{sketch}(I, Q) = \frac{1}{\sum_g \max_n [\hat{d}_{correlation}(shift_n(A^I(g)), L^Q(g))]}$$



Инвариантные моменты

Момент порядка $(p+q)$ двумерной непрерывной функций:

$$m_{pq} = \iint x^p y^q f(x, y) dx dy$$

Центральные моменты для $f(x,y)$ – дискретного изображения:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y), \quad \bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

Вектор признаков:

С использованием нормированных центральных моментов был выведен набор из 7 инвариантных к параллельному переносу, повороту и изменению масштаба моментов.

Гистограммы или моменты?



Stricker M., Orengo M. Similarity of Color Images. ... (3000 изображений)



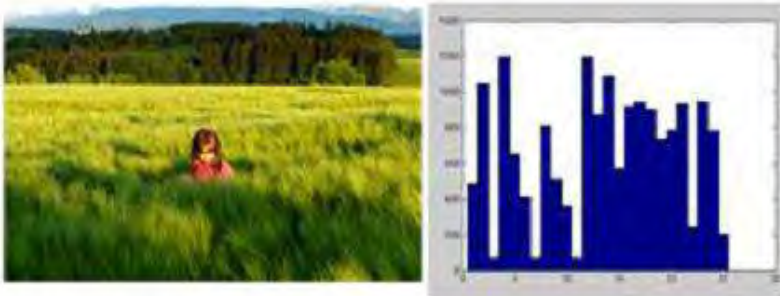
	Полнота	Точность
<u>ColorHist</u>	56,77 %	23,02 %
<u>ColorMoment</u>	55,98 %	25,06 %

Характеристики сходства изображений

Основные группы:

1. Цветовое сходство
2. Текстурное сходство
3. Сходство формы
4. Дескрипторы изображений
5. Сходство объектов и отношений между объектами

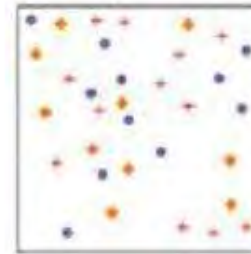
Дескрипторы изображений



Гистограммы цветов (1995)



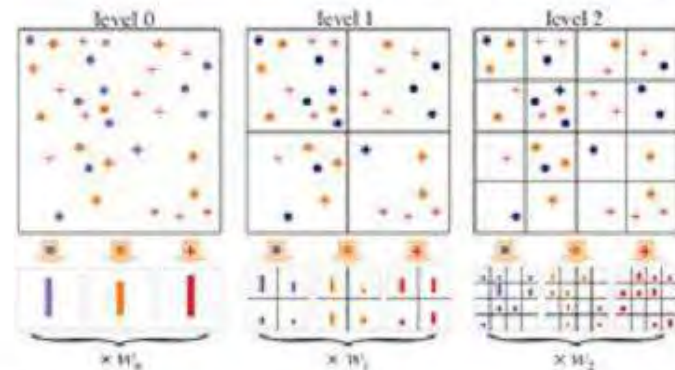
Мешок слов (2003)



Отдельные особенности (2003)

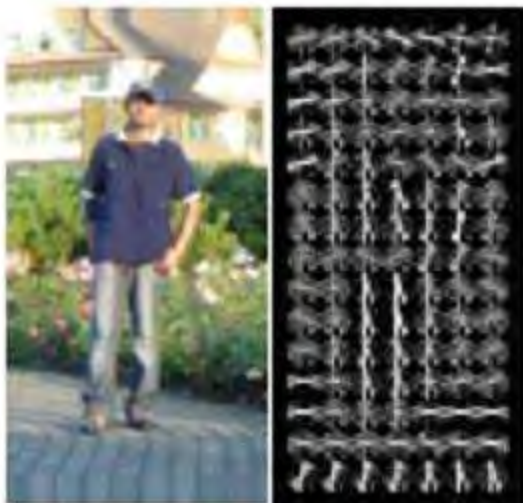


Гистограммы градиентов (2005)



Мешок слов и пирамида (2006)

Гистограммы градиентов



HOG (2003)

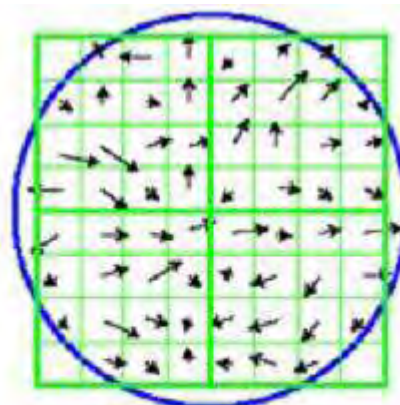


Image gradients

SIFT (2003)

Дескрипторы: Мотоциклы



Градиенты



Взвешенные + веса



Взвешенные - веса



Окно



Доминирующие + ориентации



Доминирующие - ориентации

Примеры работы



Применение дескрипторов



Запрос



Похожие по GIST + цвету

Дескриптор GIST вычисляется

1. Сверните изображение с помощью 32 фильтров Габора в 4 масштабах, 8 ориентациях, создавая 32 карты характеристик того же размера, что и входное изображение.
2. Разделите каждую карту функций на 16 регионов (сеткой 4x4), а затем усредните значения функций в каждом регионе.
3. Объедините 16 усредненных значений всех 32 карт функций, в результате получится дескриптор GIST размером $16 \times 32 = 512$.

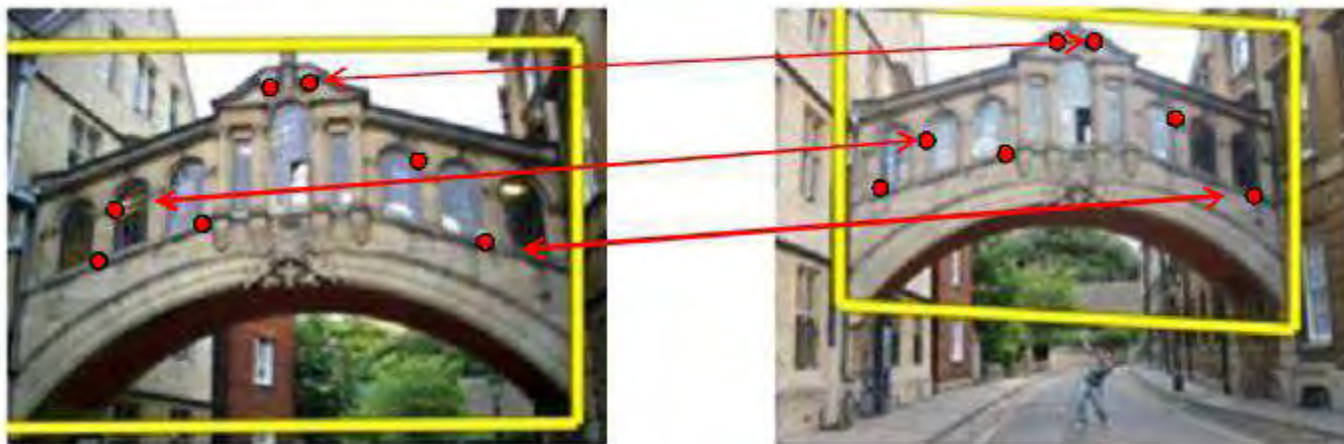
Интуитивно GIST суммирует информацию о градиенте (масштабы и ориентацию) для различных частей изображения, что дает приблизительное описание (суть) сцены.

Ограничения дескрипторов



- Как с этой задачей справятся дескрипторы?
- Не очень хорошо, т.к. размеры и ориентация объектов могут значительно меняться
- Может помочь сопоставление изображений по ключевым точкам

Геометрическое сопоставление



Характеристики сходства изображений

Четыре основные группы:

1. Цветовое сходство
2. Текстурное сходство
3. Сходство формы
4. Дескрипторы изображений
5. Сходство объектов и отношений между объектами

Обнаружение человеческих лиц

- Существующие алгоритмы обнаружения лиц можно разбить на две широкие категории:
 - Эмпирическое распознавание
 - Моделирование изображения лица
- Обнаружение элементов и особенностей которые характерны для изображения лица
 - Края
 - Яркость
 - Цвет
 - Характерная форма черт лица

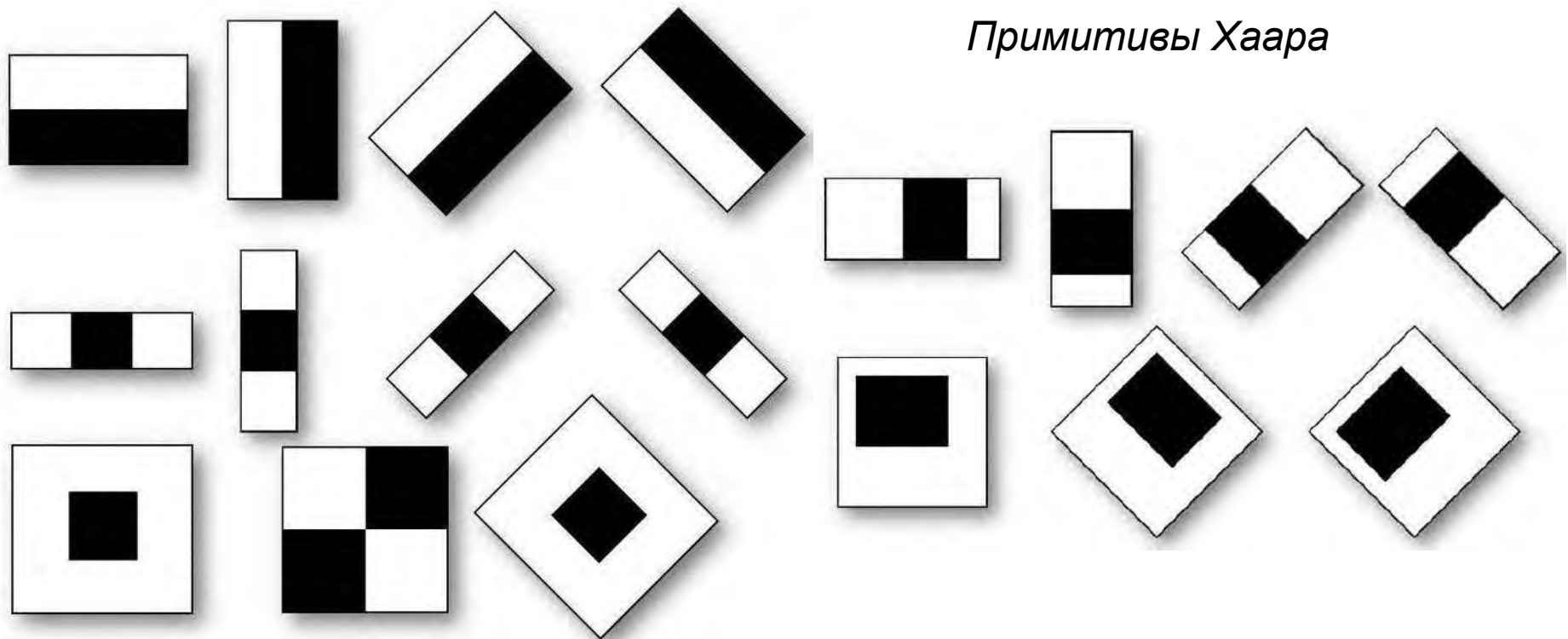
Обнаружение человеческих лиц, на основе нейронной сети (16000 изображений)



Система из университета Карнеги-Мелон

Метод Виолы — Джонса

- Предложен в 2001 году Paul Viola и Michael Jones
- Существует множество реализаций, в том числе в составе OpenCV (функция `cvHaarDetectObjects()`)





How-Old.net

HOW OLD DO I LOOK? #HowOldRobot



Sorry if we didn't quite get the age and gender right – [we are still improving this feature.](#)



Try Another Photo!

[Read the story behind this demo](#)

Обнаружение образов человеческих тел

Исходное RGB-изображение → Логарифмическое цветовое пространство

$$I = L(G)$$

$$R_g = L(R) - L(G)$$

$$B_y = L(B) - \frac{L(G) + L(R)}{2}$$

$$L(x) = 105 \log_{10}(x + 1 + n)$$

$$hue = atan(R_g, B_y)$$

$$texture = med_2(|I - med_1(I)|)$$

$$saturation = \sqrt{R_g^2 + B_y^2}$$

1. $texture < 5$, $110 < hue < 150$, $20 < saturation < 60$

2. $texture < 5$, $130 < hue < 170$, $30 < saturation < 130$

[Fleck, Forsyth, Bregler, 1996]

Признаки в системах поиска

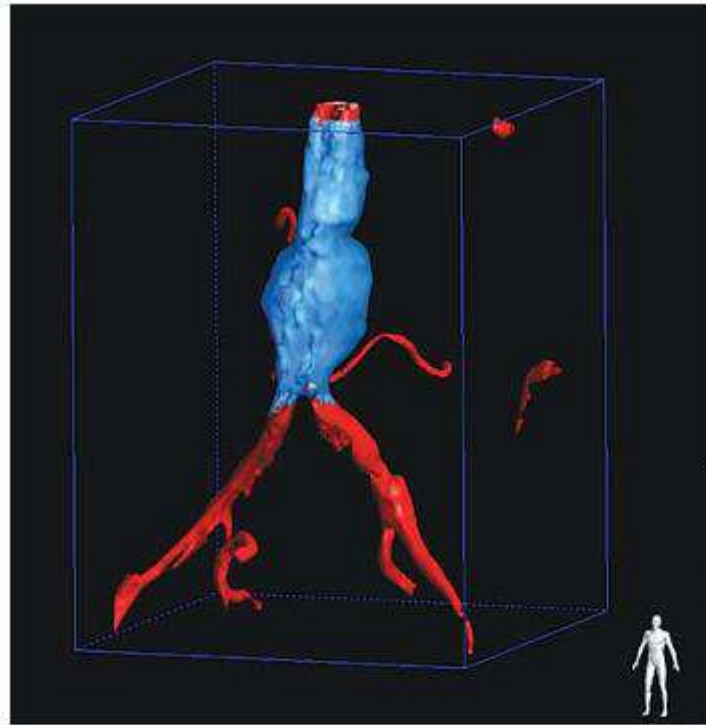
	Цвет	Текстура	Форма
QBIC	Гистограммы (HSV) $dist^2 = H_1 A H_2^T$	Tamura Image, Euclid dist	Геометрические для границ + моменты
VisualSEEk	Гистограммы (HSV), Color Sets, Location info		
Netra	Гистограммы (HSV), Color codebook, кластеризация	Фильтры Габора	Fourier-based (Фурье)
Mars	Гистограммы, HSV $dist = 1 - \sum_{i=1}^N \min(H_1(i), H_2(i))$	Tamura Image, 3D Histo	MFD (Фурье)

Заключение

- Большой выбор различных способов представления изображений
- Цвет: гистограммы или статистическая модель?
- Текстура: фильтры Габора, фильтры ICA
- Форма: дескрипторы Фурье, инвариантные моменты
- При сравнении изображений часто необходимо комбинировать различные признаки

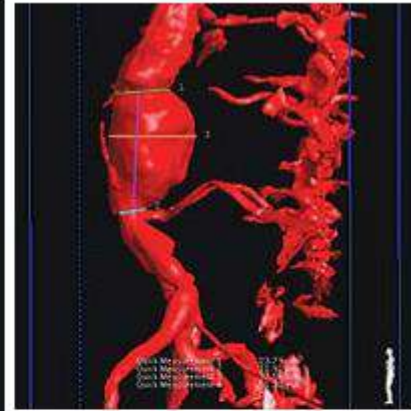
3D-реконструкция

Области применения 3D-реконструкции

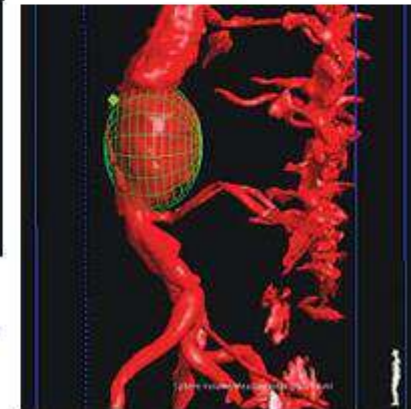


а

Трехмерная реконструкция аневризмы инфраренального сегмента аорты.

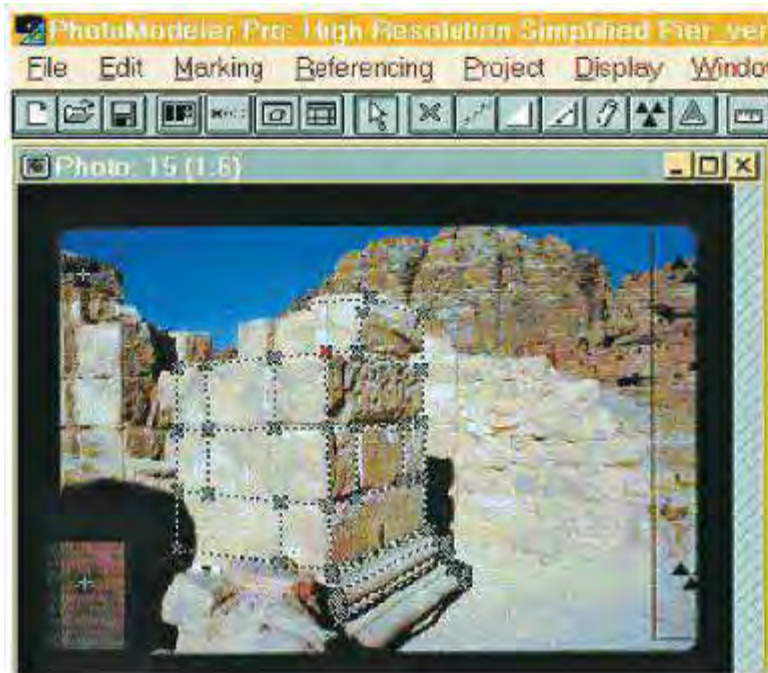


б



в

Области применения 3D-реконструкции



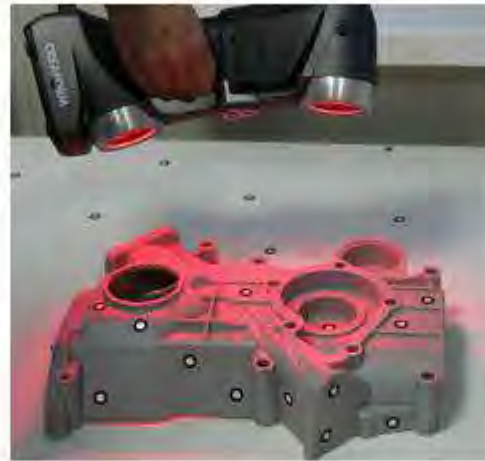
Построение трехмерной модели в автоматизированном режиме посредством программы PhotoModeler Pro³⁶ (слева). Справа: вид раскопа главного храма Петры³⁷.

Области применения 3D-реконструкции



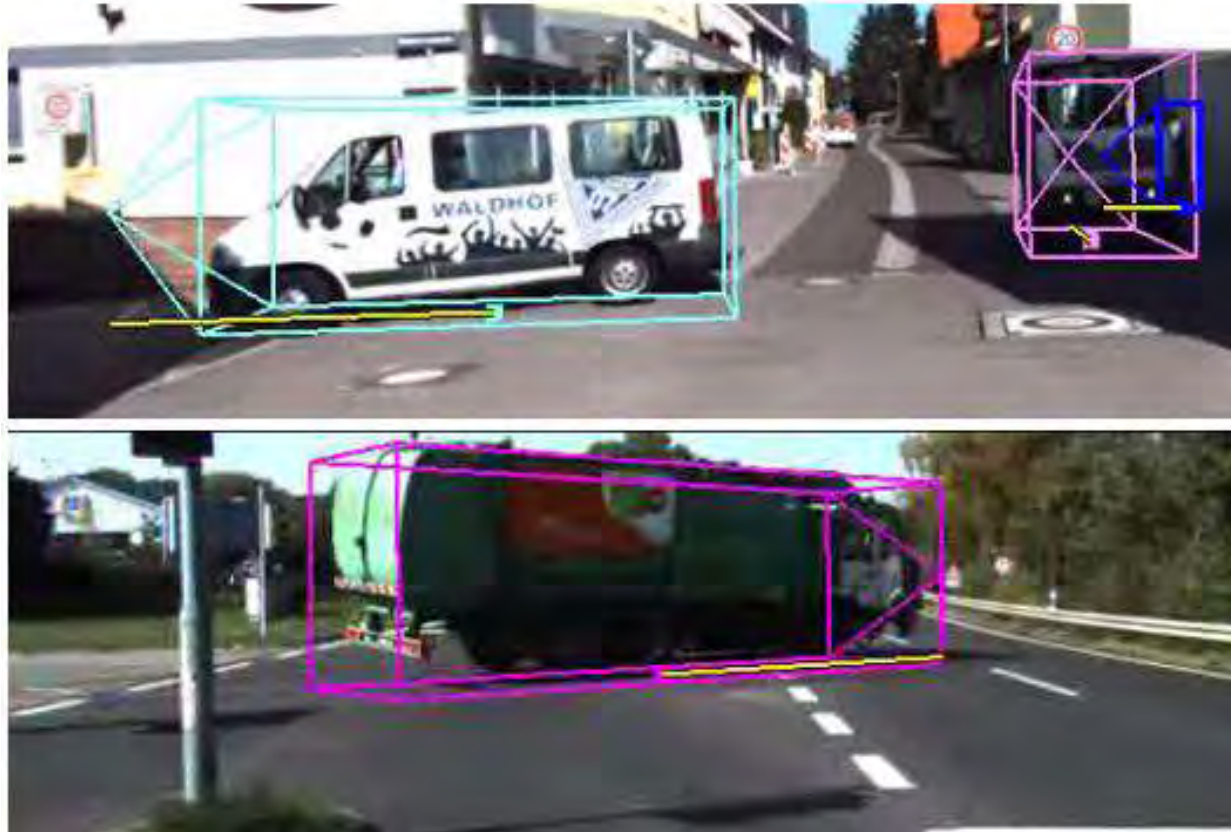
Виртуальная реконструкция Тамбова конца XVIII — начала XIX в.
Сопоставление визуальных источников панорамы города⁴³, фотографии Екатерининского института
конца XIX в. с виртуальной панорамой города

Области применения 3D-реконструкции

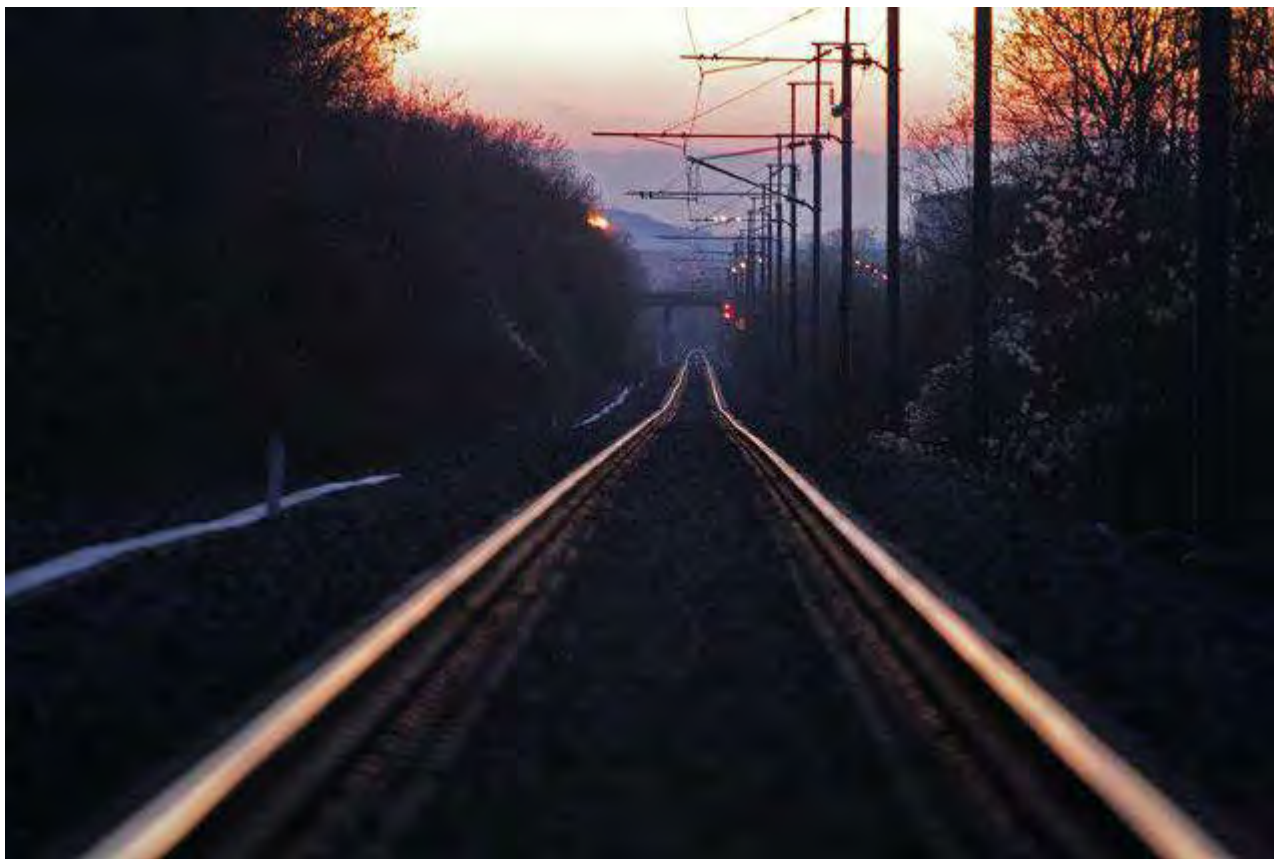


Использование реконструкции в производстве, для бесконтактного определения дефектов и в логистике

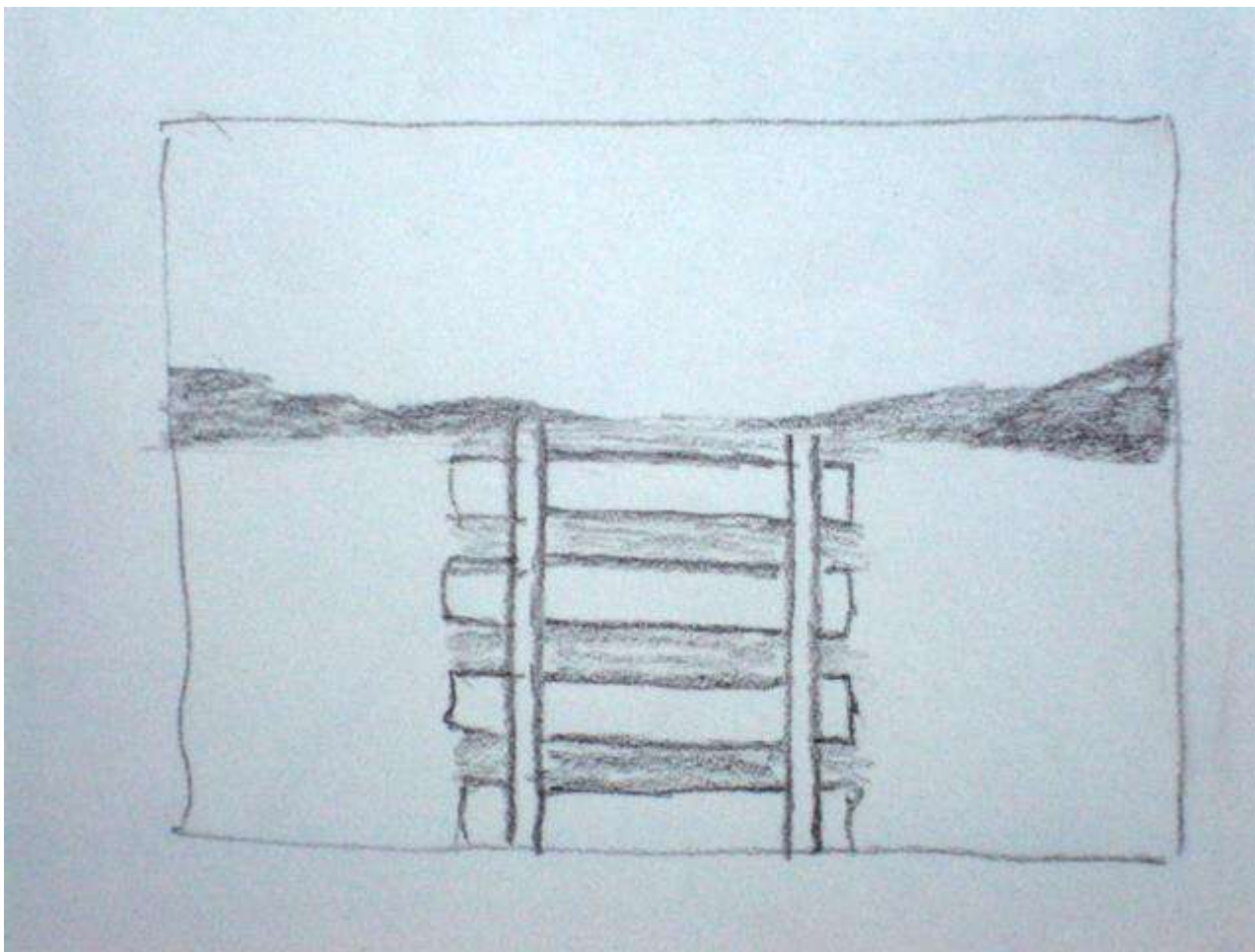
Области применения 3D-реконструкции



Что это?



А это?



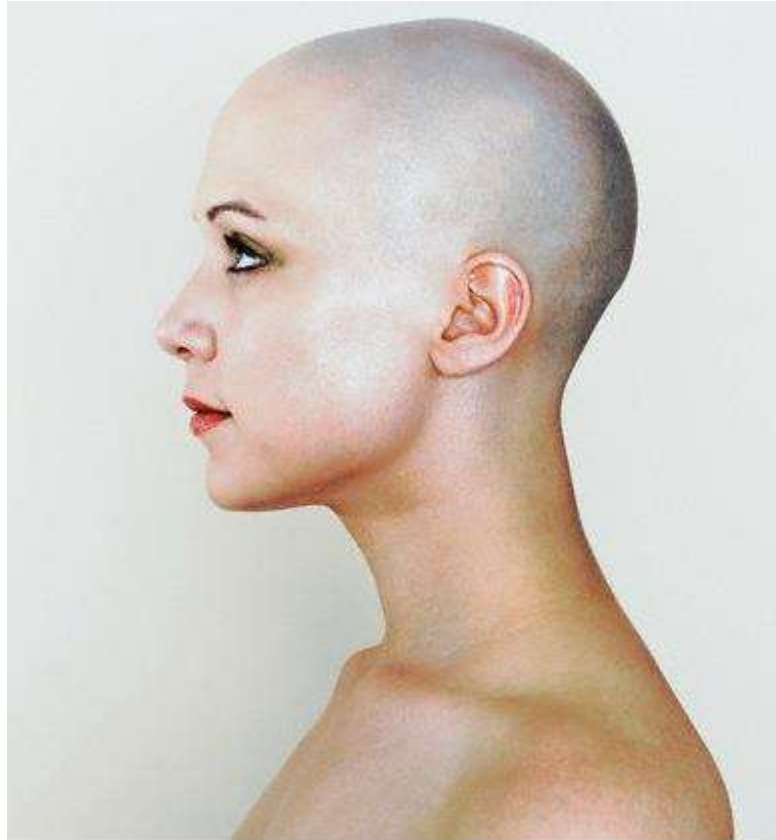
Демяненко Я.М., ЮФУ, 2021

Ну а это?



Демяненко Я.М., ЮФУ, 2021

Попытайтесь объяснить, как вы определяете, что это



Как же всё происходит?

- Инвариантность к переносу (дескрипторы, свёрточные сети, ...)
- Инвариантность к масштабу и повороту (SIFT, SURF и др.)
- Перспектива и поворот предмета в пространстве сцены -- ?

Восприятие глубины и пространства

- Все знают, что для восприятия глубины и пространства мы пользуемся бинокулярным зрением, у нас для этого два специальных глаза

Биноккулярное зрение

- Стереокамеры
- Дальномеры
- Kinect

Восприятие глубины и пространства

- забавный факт — в списке преимуществ бинокулярного зрения в Википедии
«It gives a creature a spare eye in case one is damaged»

Ещё один способ -- движение



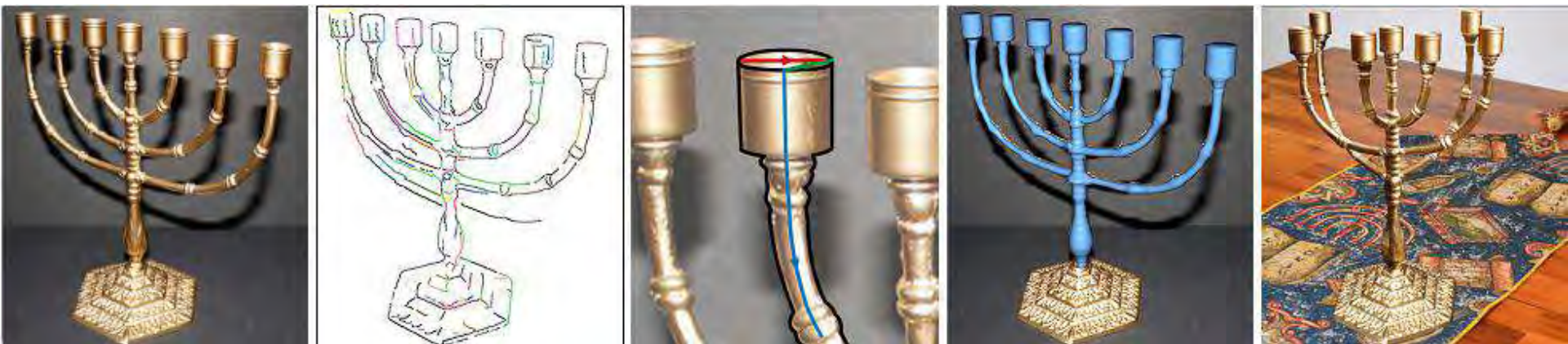
Параллакс движения

- Когда мы движемся, близкие предметы смещаются в поле зрения сильнее, чем далекие

Техника, которая официально называется «structure-from-motion»

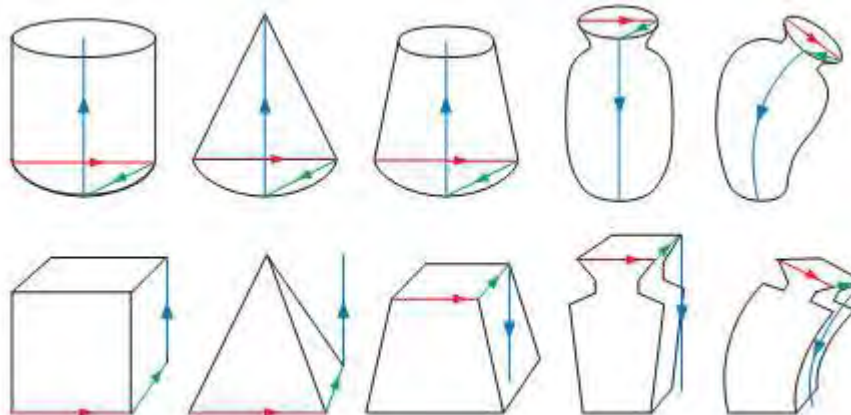
- Итак, мы собираемся двигаться, оценивать смещения предметов в кадре и на основании этого определять их расстояние от наблюдателя.

Частные случаи

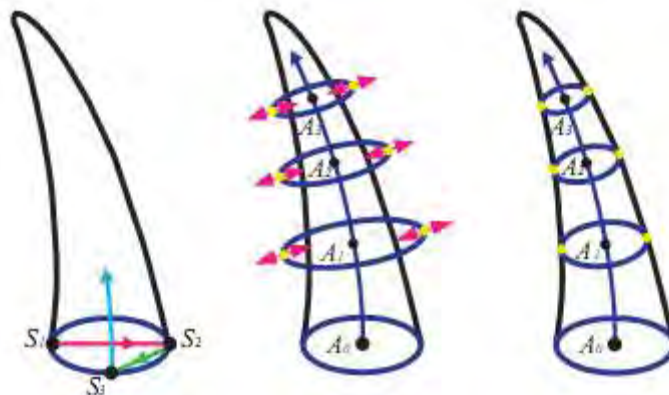


Получение редактируемых объектов из одиночных изображений

Получение редактируемых объектов из одиночных изображений



Получение редактируемых объектов из одиночных изображений









Все методы трёхмерной реконструкции можно сгруппировать в две категории:

- Активные
- Пассивные

Активные методы трёхмерной реконструкции

- Системы, реализующие активные методы, прямо взаимодействуют с объектами исследуемой сцены.
- Данное взаимодействие осуществляется через излучение с **заранее известными параметрами** в направлении объектов сцены с целью привнести в сцену **дополнительную информацию**, на основании которой будут работать реконструирующие алгоритмы, и последующее измерение отражённой части излучения.

Параметрами излучения могут являться

- тип излучения: электромагнитное излучение, в т.ч. радиоизлучение, инфракрасное, видимое, ультрафиолетовое, рентгеновское; инфразвук, звук в слышимом диапазоне, ультразвук;
- частотные и периодические параметры излучения;
- геометрические параметры излучения: форма пятен, размер пятен.

Примерами датчиков могут служить

- видео- и фотокамеры, работающих в видимых диапазонах, а также диапазонах, смежных с видимым;
- лазерные дальномеры;
- микрофонные матрицы;
- ультразвуковые датчики.

Пассивные методы трёхмерной реконструкции

- **Системы, реализующие пассивные методы, не взаимодействуют с объектами исследуемой сцены.**
- Основная идея данного подхода - измерение излучения, отражённого от объектов наблюдаемой сцены.
- **Необходимым условием работы данного подхода является наличие в сцене независимого источника излучения.**
- Выполнение данного условия наиболее характерно для светового излучения в видимом диапазоне, и обусловлено наличием естественных источников: солнца, луны, звёзд, а также наличием множества искусственных источников, имитирующих естественные.

Построение модели сцены

- осуществляется на основе **последовательности её изображений** (наборы снимков, видеозаписи) и **информации о взаимном расположении датчиков** в моменты съёмки (которая, впрочем, может быть извлечена из самой последовательности изображений).
- Значимым шагом является **компенсация искажений**, вносимых линзами приёмника. Так как геометрия линз приёмника постоянна, то вносимые искажения будут неизменными, а следовательно, будут легко поддаваться оценке и последующей компенсации.

Ключевой шаг - построение карты глубины

- Извлечение **уникальных точек** из каждого изображения, нахождения **соответствующих им точек**, извлечённых из остальных изображений набора и **последующая триангуляция** с использованием данных о взаимном расположении датчиков.

На основании карты глубины строится трёхмерная модель наблюдаемой сцены

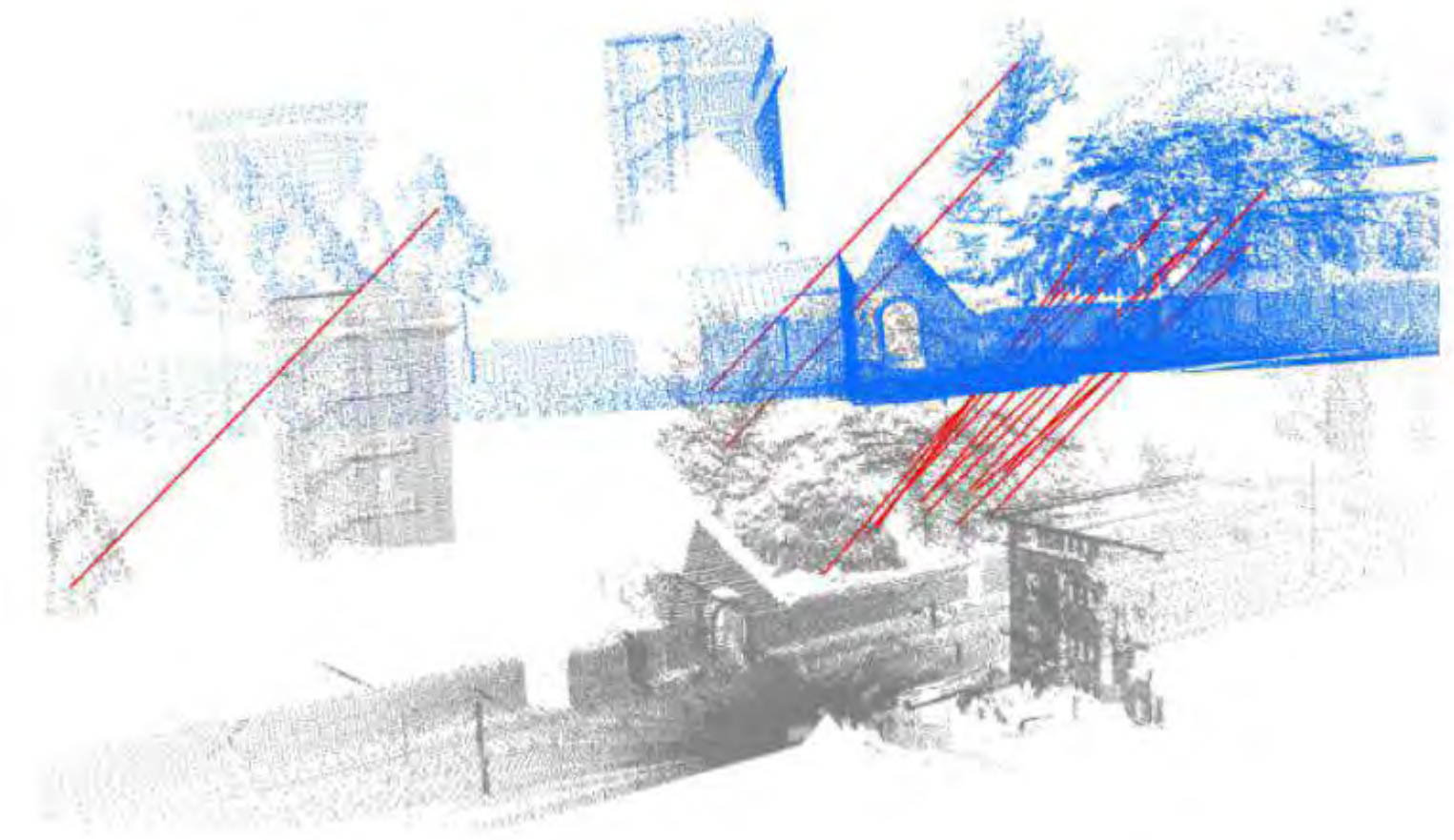
Для представления геометрии сцены используется один из следующих подходов

- Облако точек.
- Полигональное представление.
- Воксельное представление.

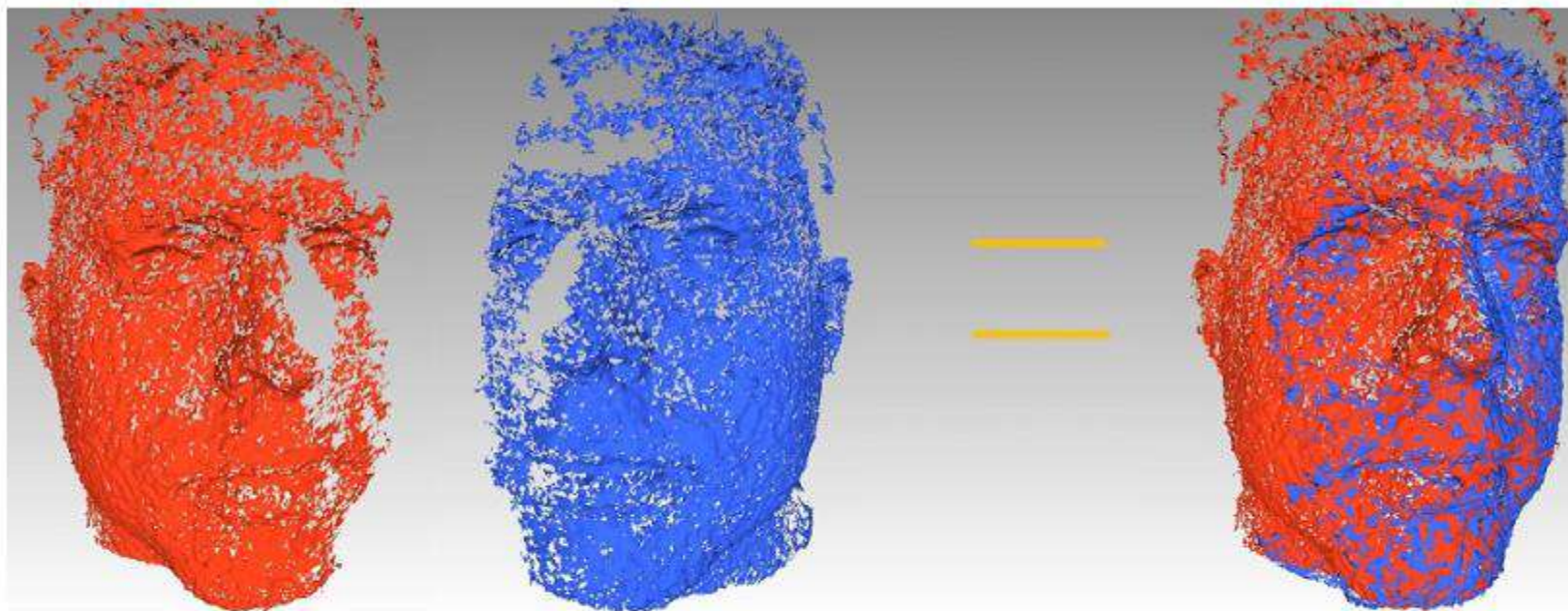
Облако точек

- Сырые геометрические данные, пригодные для последующей обработки: фильтрации, классификации.
- Группируя облака точек, полученные в результате обработки различных пар изображений, можно построить более точную и полную модель сцены.

Регистрация нескольких облаков точек



Пример регистрации



Полигональное представление

- Подход к представлению геометрических поверхностей, получивший наиболее широкое распространение в области визуализации в реальном времени (в т. ч. компьютерные игры, демонстрационные визуализации различных химических, физических процессов).
- Позволяет визуализировать модели, оценивать их геометрию (в т. ч. распознавать геометрические примитивы - плоскости, прямоугольники, сферы и т. д.) и производить передачу моделей в сторонние пакеты обработки моделей.

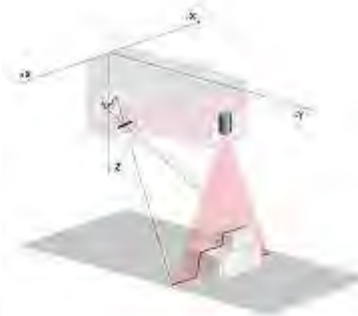
Воксельное представление

- Поверхность представляется множеством примитивных объектов - вокселей. В роли таких объектов зачастую выступают прямоугольные примитивы (чаще всего кубы), либо сферы.
- Такой подход позволяет дополнять и конкретизировать модель при поступлении новых данных. Подход широко применяется для трёхмерного реконструирования на основе данных медицинских сканирующих устройств (компьютерных, магнитно-резонансных томографов, УЗИ).

Существующие технологии 3D-реконструкции



Сканирующие лазерные
дальномеры



Методы оптической
триангуляции

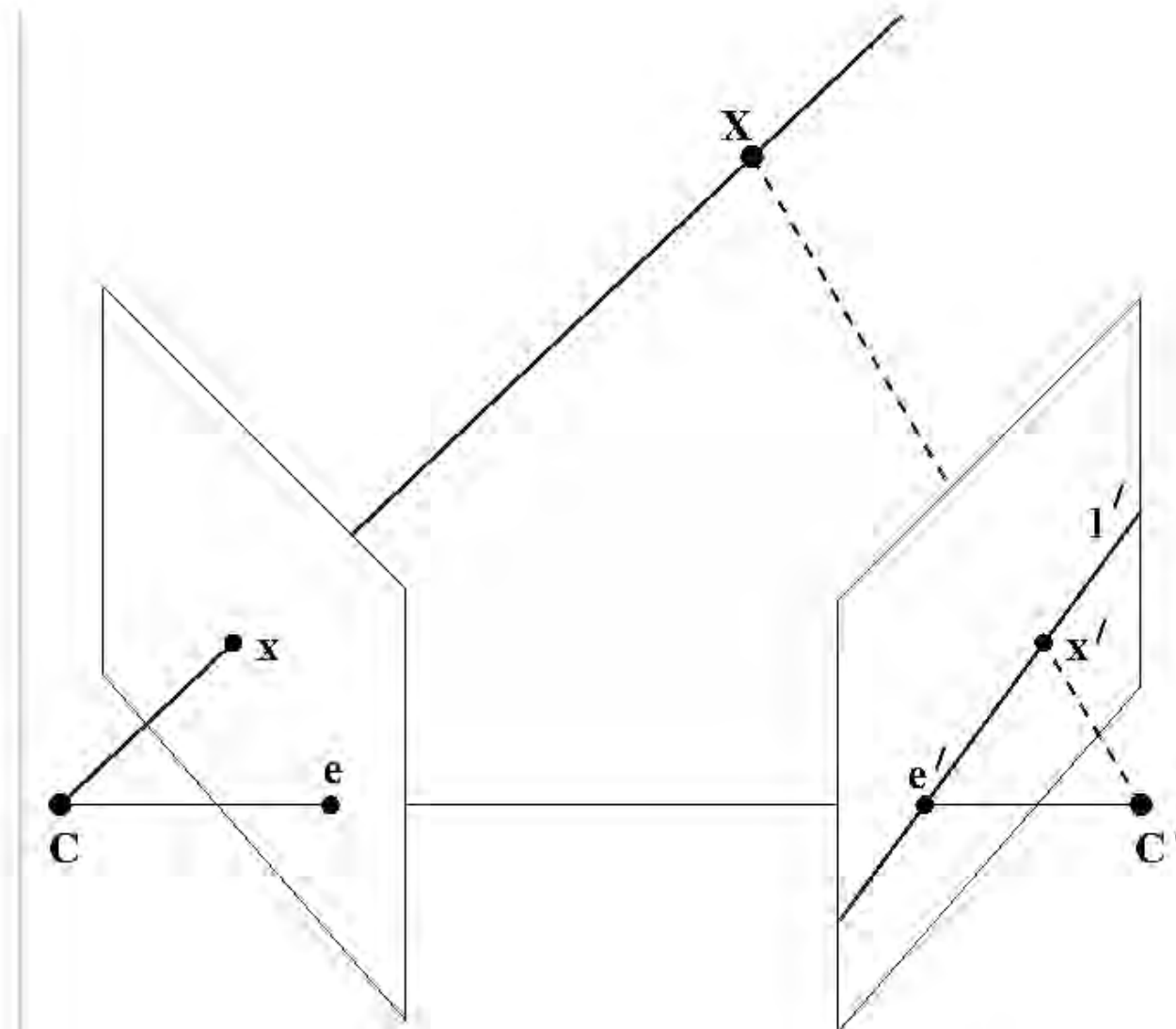


Активное стереовидение



Пассивное стереовидение

Эпилярная геометрия



Эпилярную геометрию используют для поиска стереопар

Демяненко Я.М., ЮФУ, 2021

Bundler: Structure from Motion (SfM) for Unordered Image Collections



Software written by [Noah Snavely](#)

Download Bundler from the [bundler_sfm repository on GitHub](#)

Bundler: Structure from Motion (SfM) for Unordered Image Collections

- Это достаточно простой и одновременно эффективный метод.
- Но, к сожалению, возникает проблема,— Bundler для корректной работы хочет знать модель камеры и ее внутренние параметры (в крайнем случае, если модель неизвестна, он требует указать фокусное расстояние).

Постановка задачи



Левый снимок



Правый снимок



a)



b)

Результат реконструкции. Вид с различных ракурсов:
сверху (a), из позиции камеры (b)

Корректировка стереопары

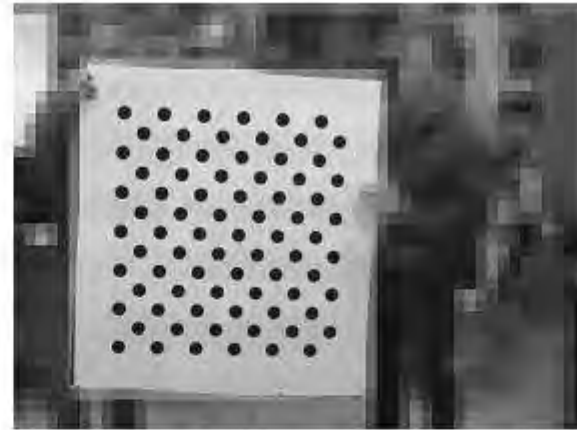
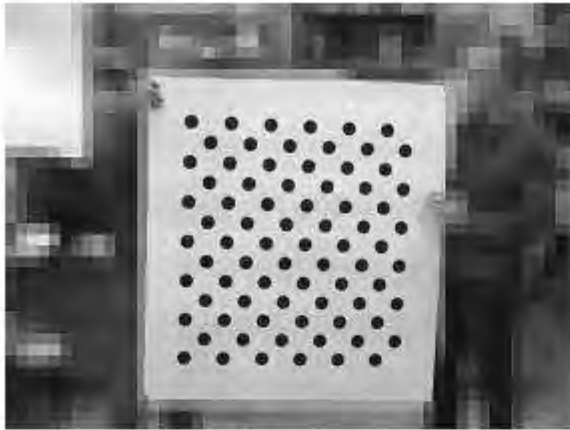


Исходная стереопара

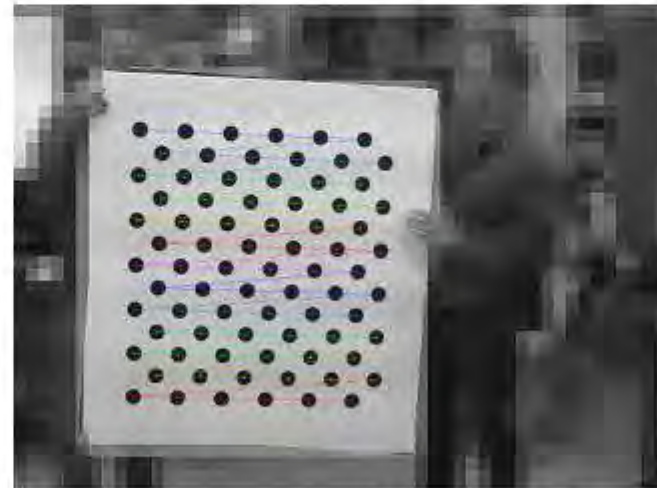
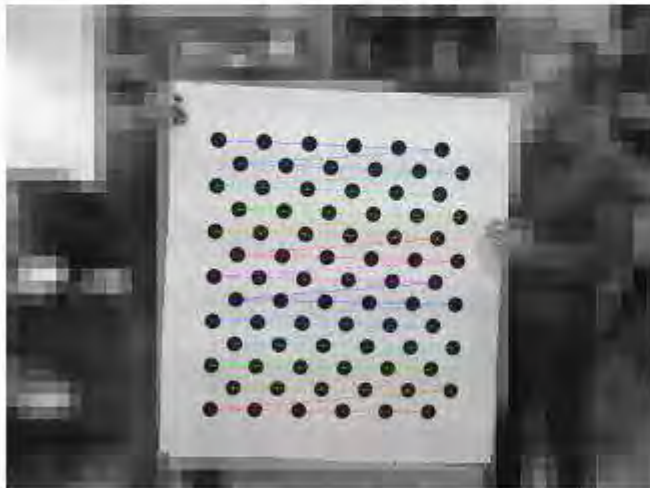


Откорректированная стереопара

Калибровка камеры



Снимки калибровочного шаблона с левой и правой камер



Найденные на этих изображениях характерные элементы

Подход в OpenCV

1. Взять два кадра, снятые с откалиброванной камеры.
2. Вместе с параметрами калибровки (матрицей камеры) положить их оба в функцию [stereoRectify](#), которая выпрямит (ректифицирует) эти два кадра — это преобразование, которое искажает изображение так, чтобы точка и ее смещение оказывались на одной горизонтальной прямой.



Подход в OpenCV

3. Эти ректифицированные кадры мы кладем в функцию [stereoBM](#) и получаем карту смещений (disparity map) — такую картинку в оттенках серого, где чем пиксель ярче, тем большее смещение он выражает.



Подход в OpenCV

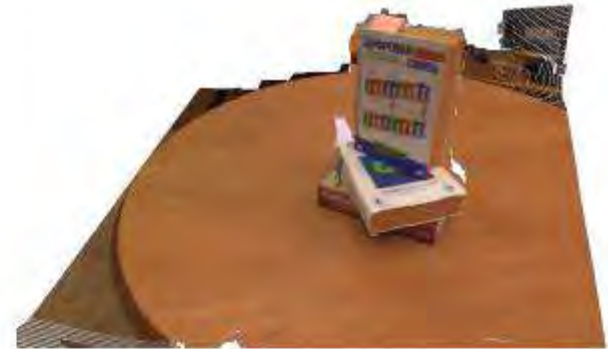
4. Полученную карту смещений кладем в функцию с говорящим названием [reprojectImageTo3D](#) (понадобится еще и матрица Q , которую в числе прочих мы получим на шаге 2).
5. Получаем наш трехмерный результат.



Левый снимок



Карта смещений



Результат
реконструкции



Левый снимок



Карта смещений

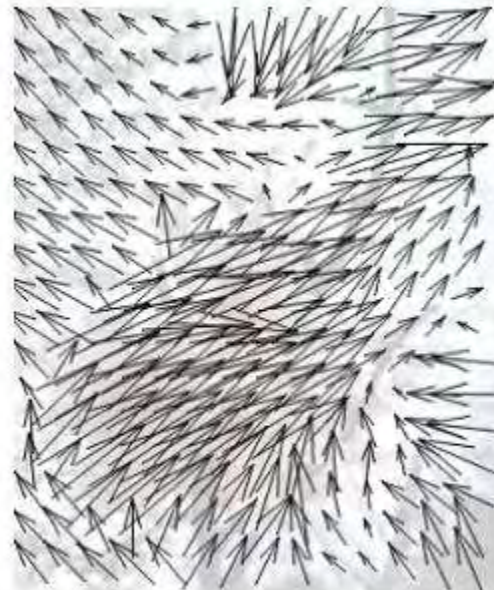


Результат
реконструкции

Полезные ссылки

- The KITTI Vision Benchmark Suite
http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=stereo
- Point Clouds Library
<http://pointclouds.org/>
- OpenCV, модуль calib3d
<http://docs.opencv.org/modules/calib3d/doc/calib3d.html>
- Insight3d
<http://insight3d.sourceforge.net/>

Пример оптического потока



3D-реконструкция



Входные данные



a)



b)

Входное изображение (a) и начальное положение кривой (b)

Поэтапная эволюция кривой

