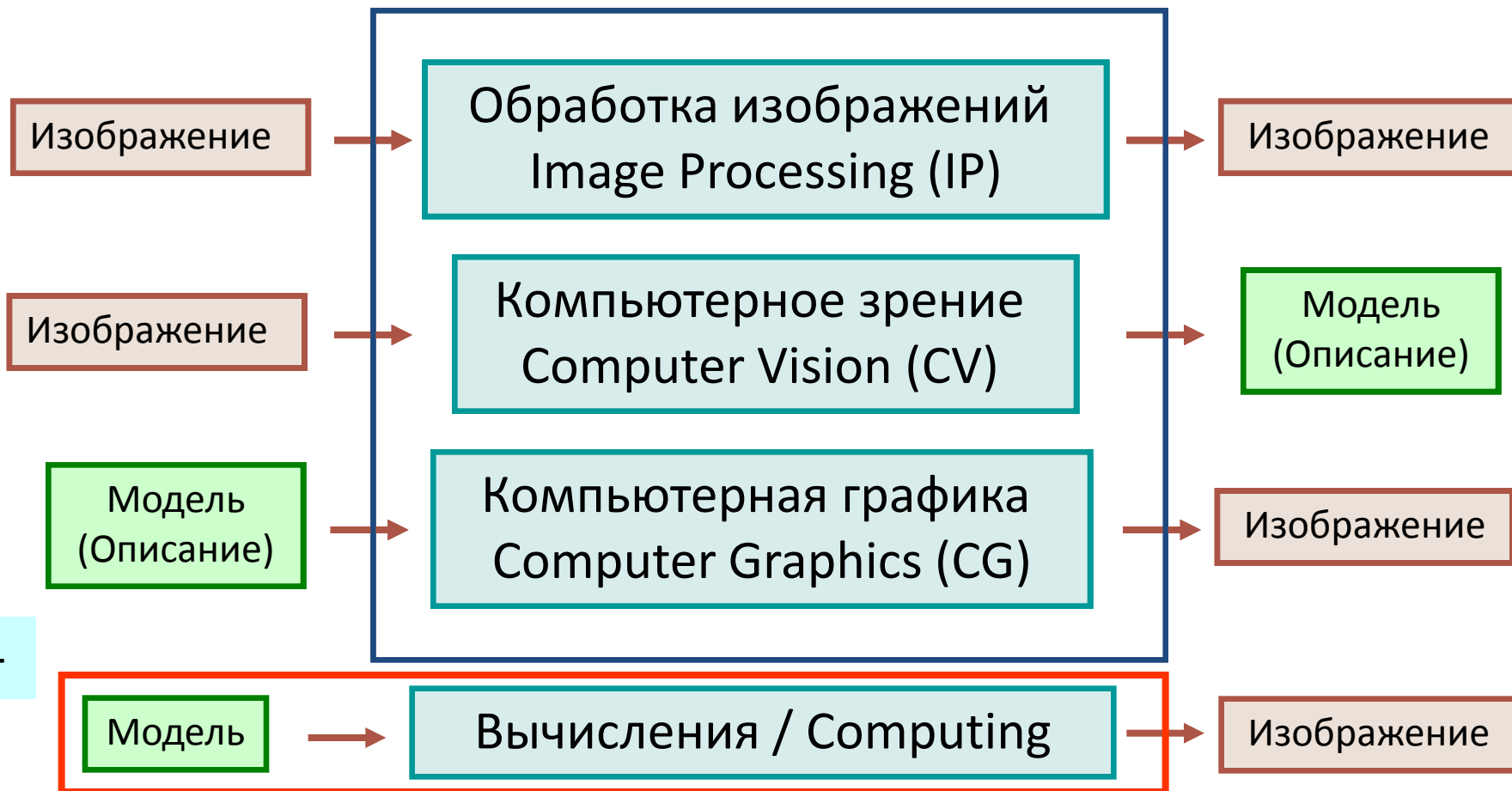


# АЛГОРИТМЫ МАШИННОЙ ГРАФИКИ: I



Цыганов А.В. 2010.

# Машинная или компьютерная графика в широком смысле Visual Computing



# Обработка изображений Image Processing

Изображение



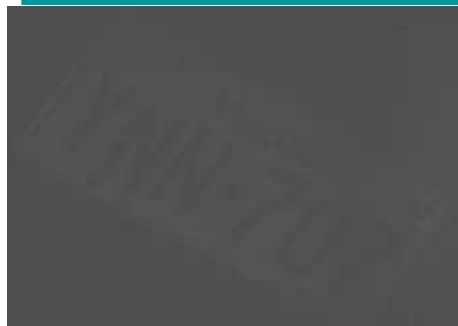
Обработка изображений  
Image Processing



Изображение



Захват кадра из видеопотока



Выделение области для распознавания



Контрастирование области распознавания



Замена цвета для усиления контраста между номером и подложкой



# Компьютерное зрение Computer Vision

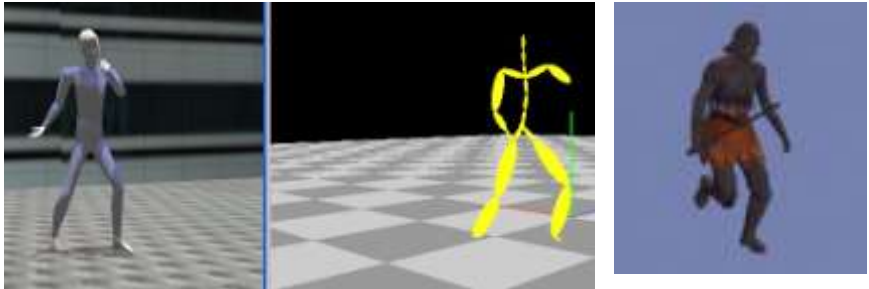
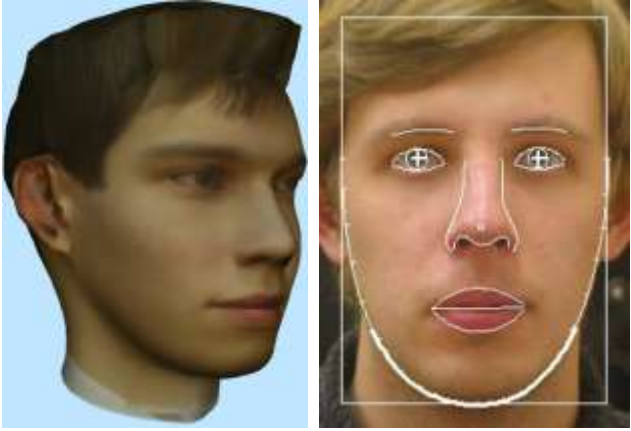
Изображение



Компьютерное зрение  
Computer Vision

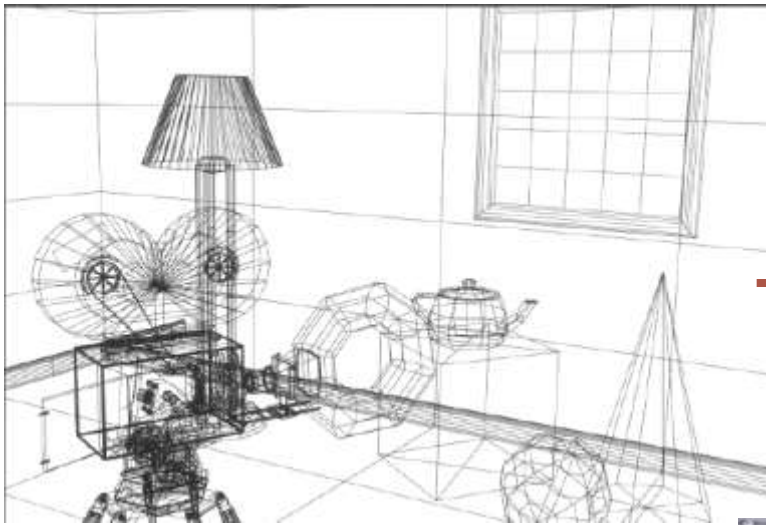
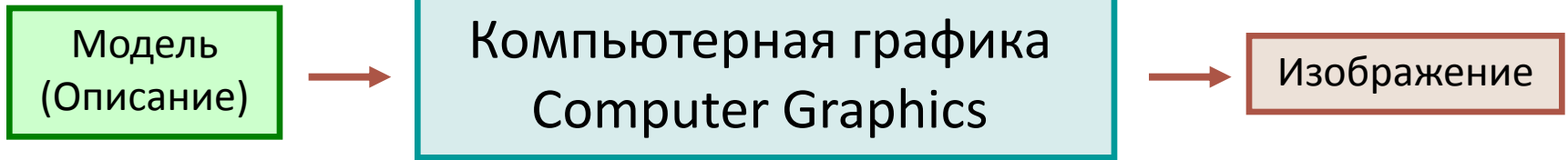


Модель  
(Описание)

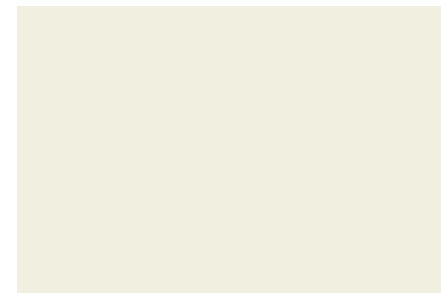
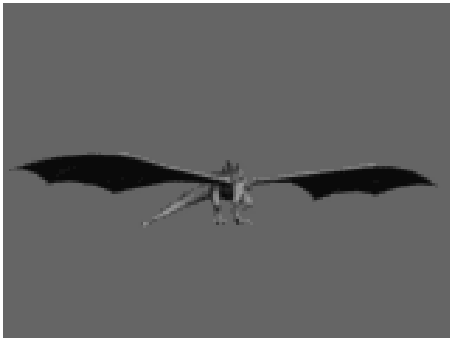


Захват движения и анимация персонажа

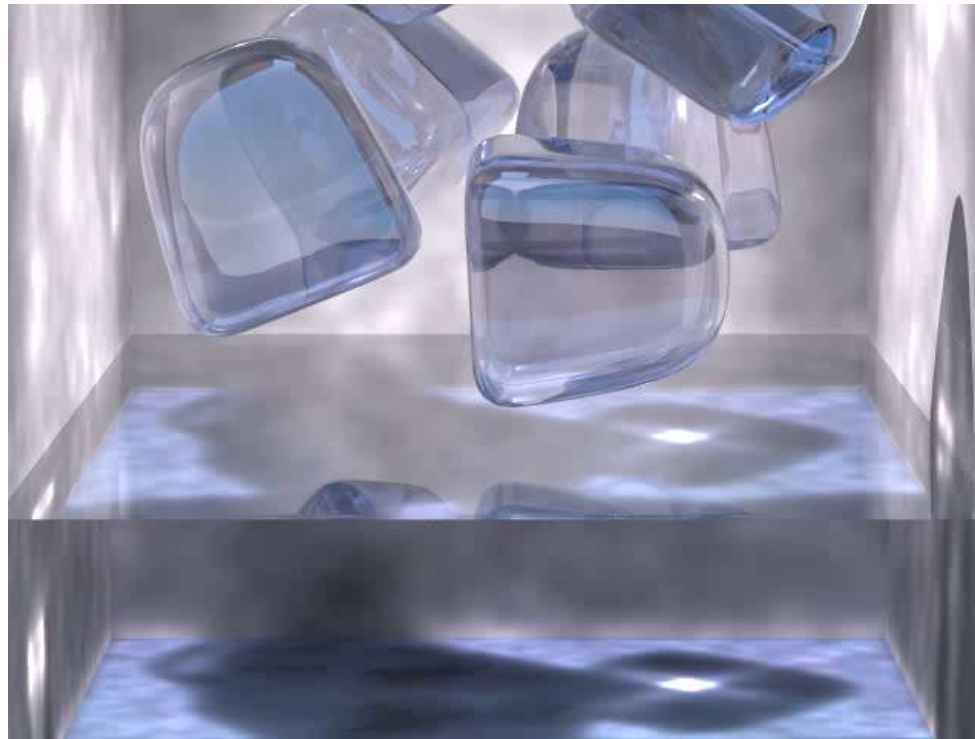
# Компьютерная графика Computer Graphics



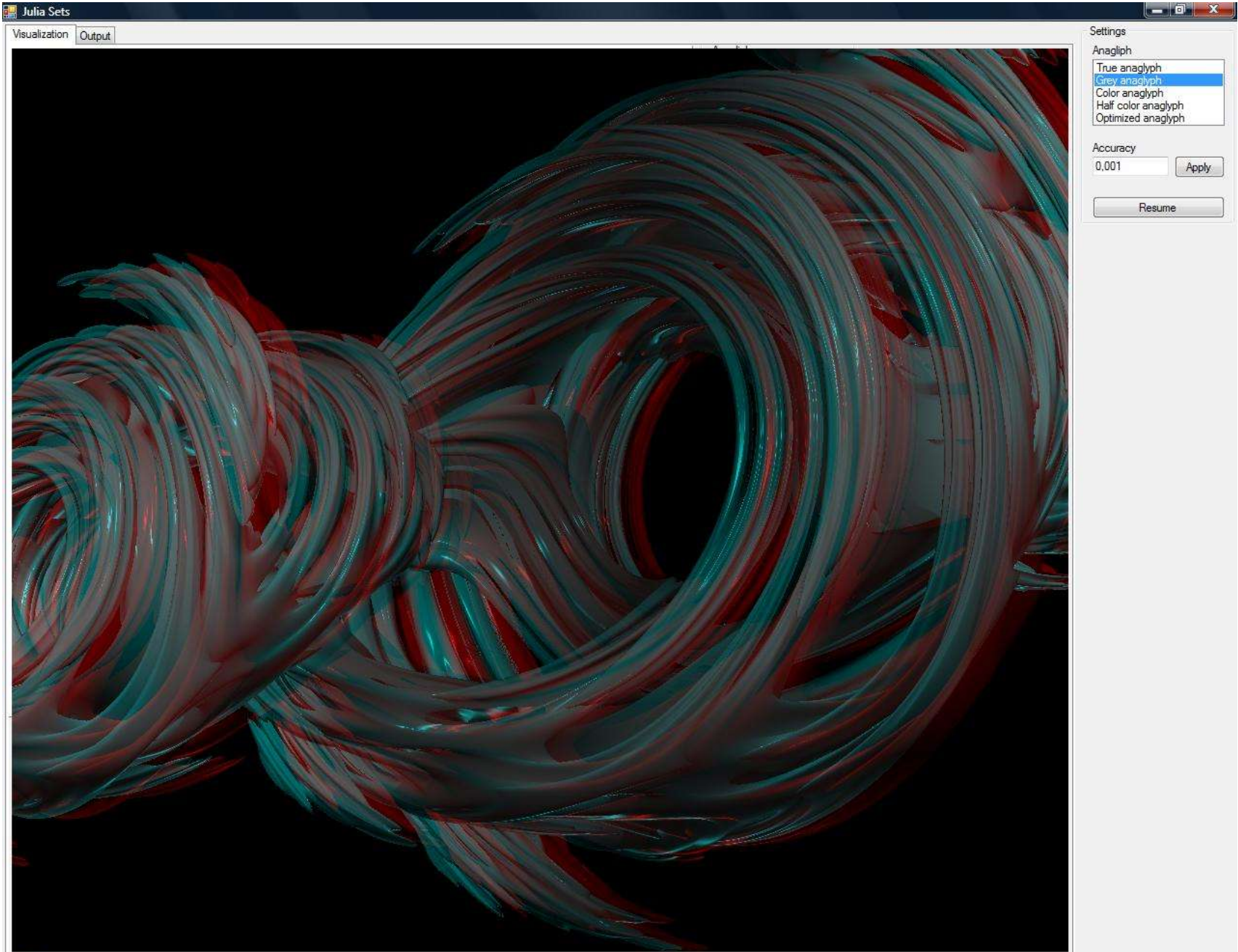
Графический  
конвейер



# Visualization \ Visual Computing

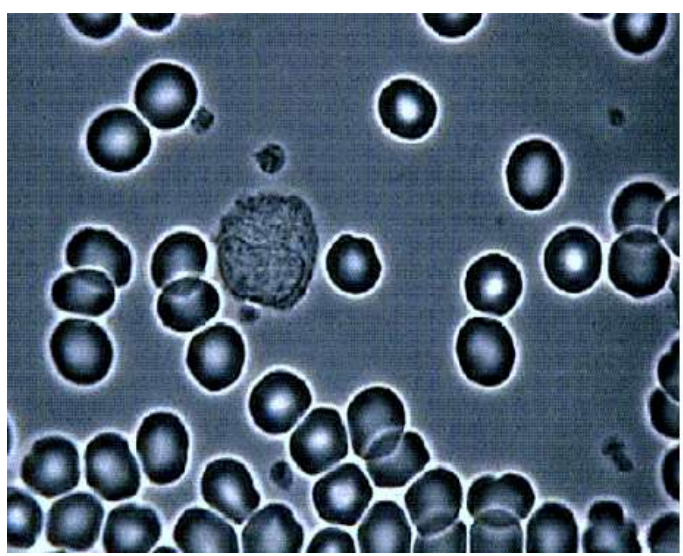









# Зачем это надо?



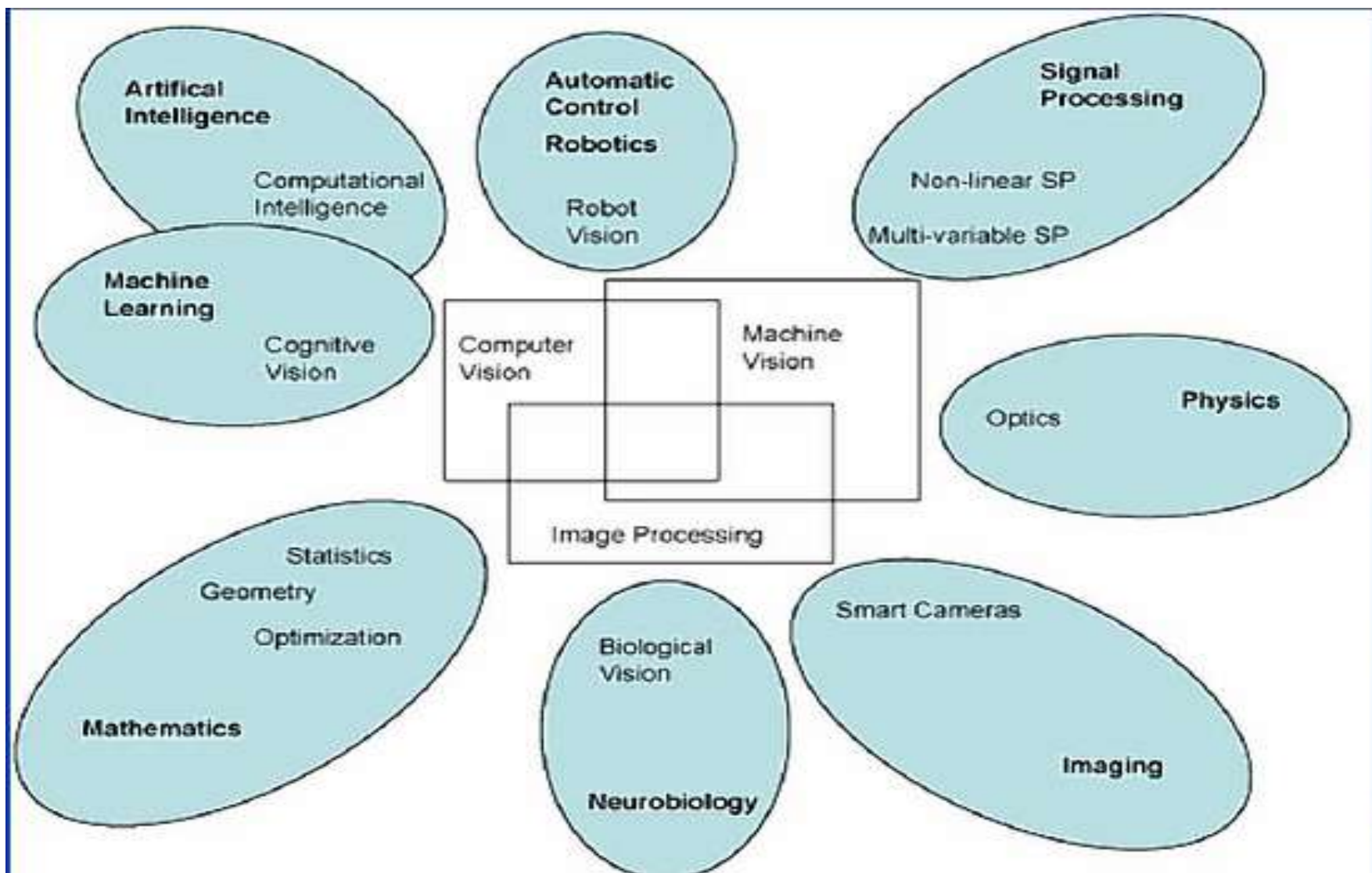
# Неразрушающая диагностика

- Поиск и анализ дефектов без разрушения объекта исследования
-  Автоматизированный поиск дефектов по изображениям



Автоматический поиск трещин в асфальте по ИК изображениям

# Смежные дисциплины



- **Физика** – требуется понимание и учет различных физических сдвигения объекта и формирования его изображения (оптика, механика, теория сплошных сред, теория горения и взрыва и т.д.)
- **Математика** – геометрия, алгебра, численные методы, мат. статистика, фракталы, вейвлеты и т.д.
- **Анализ и обработка данных** – методы обработки одномерных данных обобщаются на двумерные (изображения) и трехмерные (видео) объекты.
- **Психология, физиология, нейробиология** – особенности восприятия, оптические иллюзии – например шрифты.
- **Искусственный интеллект** - в частности решает задачи планирования и принятия решений для систем управления механизмов, использующих оптические сенсоры .....

# Изображение

**Изображение оптическое** – картина, получаемая в результате прохождения через оптическую систему лучей, распространяющихся от объекта, и воспроизводящая его контуры и детали.

*Физический энциклопедический словарь.*

# Обработка изображений

Функция интенсивности (яркости) канала изображения:

$$I = g(x, y), \{x \in [x_0, x_1], y \in [y_0, y_1]\}$$

В компьютере используется ее дискретное представление:

$$I = g(i, j), \{i = \overline{1, n}, j \in \overline{1, m}\}$$

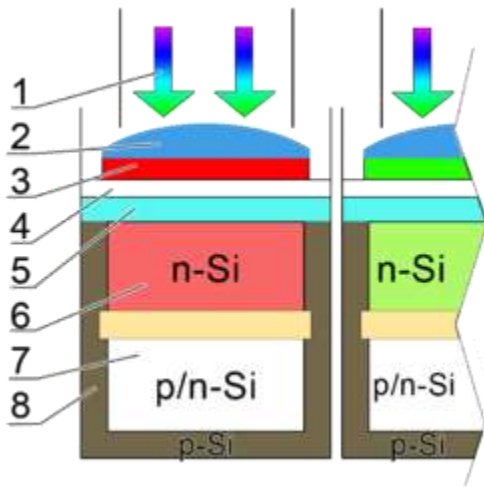
**Семейство методов и задач, где входной и выходной информацией являются изображения.**

Примеры :

- Устранение шума в изображениях
- Улучшение качества изображения
- Усиления полезной и подавления нежелательной (в контексте конкретной задачи) информации

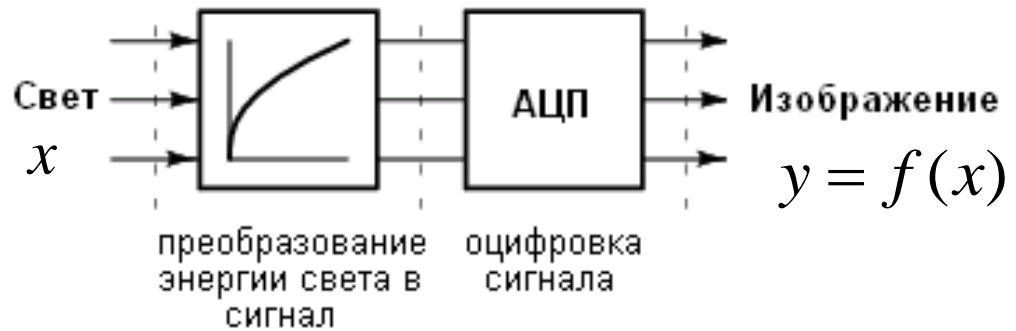
# Цифровое изображение

- Свет, падая на светочувствительный элемент матрицы ПЗС (прибор с зарядовой связью, CCD-Charge-Coupled Device), преобразуется в электрические сигналы, зависящие от интенсивности света → **проблема воссоздания цвета!**
- Сигналы оцифровываются, превращаются в массив чисел



Пример субпиксела  
3 — R-красный светофильтр субпикселя,

фрагмент фильтра Байера;



$x$  — характеристика яркости света

$y$  — яркость пикселя изображения

# Причины потери качества изображения

- Ограниченный диапазон чувствительности ПЗС-матрицы
- “Плохая” функция передачи ПЗС-матрицы
- “Плохая” освещенность





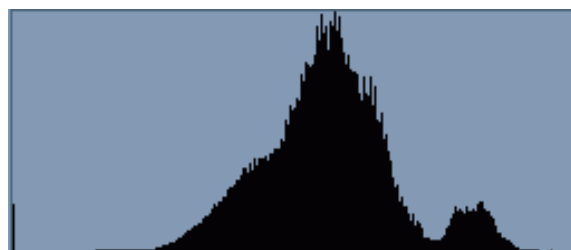
**Гистограмма** – это график распределения интенсивности в изображении.

На горизонтальной оси - шкала яркостей тонов от белого до черного, на вертикальной оси - число пикселей заданной яркости.



0

255



0

255

# Коррекция яркости/контраста изображения

Что может не устраивать в полученном изображении:

- Узкий или смещенный диапазон яркостей  
(узкий диапазон - тусклое изображение, «пересвеченное» изображение)
- Концентрация яркостей вокруг определенных значений, неравномерное заполнение диапазона яркостей

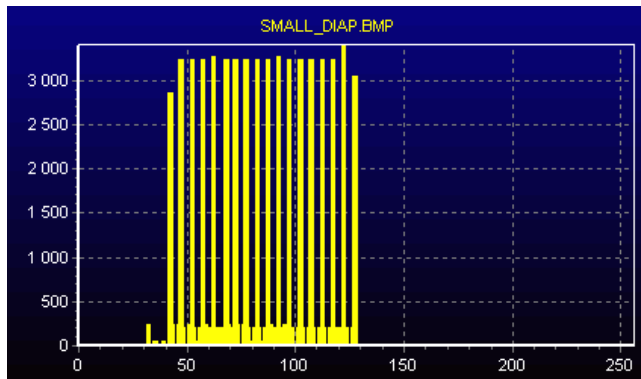
Коррекция - к изображению применяется преобразование яркостей, компенсирующее нежелательный эффект:

$$f^{-1}(y) = x$$

$y$  – яркость пиксела на исходном изображении,  
 $x$  – яркость пиксела после коррекции.

# Линейная коррекция

Компенсация узкого диапазона яркостей – линейное растяжение:



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

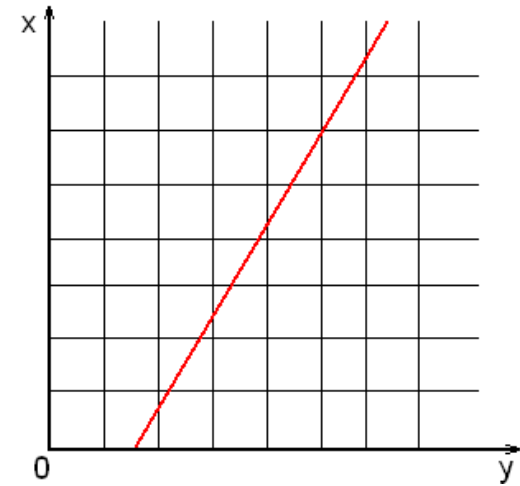
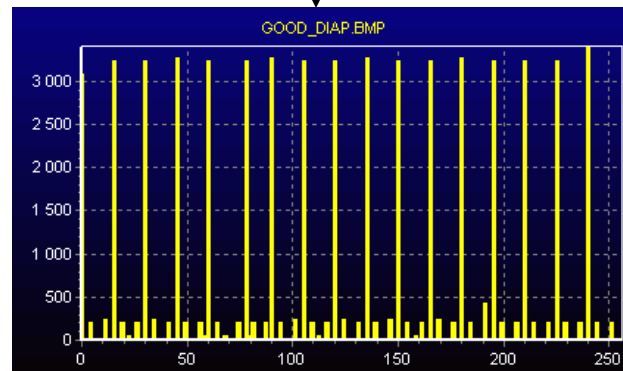
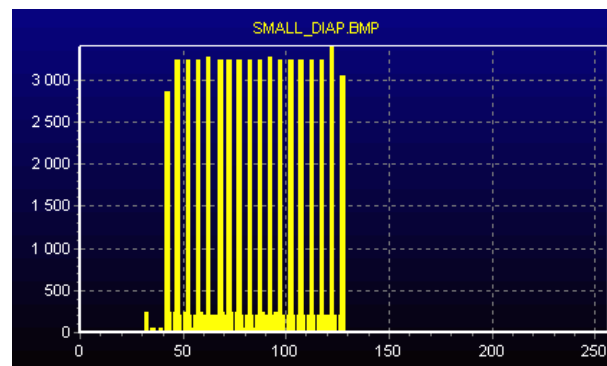


График функции  $f^{-1}(y)$

# Линейная коррекция.

Компенсация узкого диапазона яркостей – линейное растяжение:



# Линейная коррекция.

Линейное растяжение – «как AutoContrast в Photoshop»



*Линейная коррекция не помогает, если в изображении уже представлены все интенсивности!*



# Нелинейная коррекция

## Нелинейная компенсация недостаточной контрастности

Часто применяемые функции:

- **Гамма-коррекция**

- Изначальная цель – коррекция для правильного отображения на мониторе.

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

- **Логарифмическая**

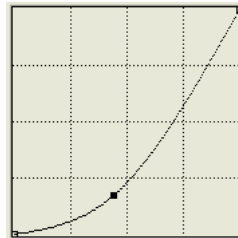
- Цель – сжатие динамического диапазона при визуализации данных

$$y = c \cdot \log(1 + x)$$

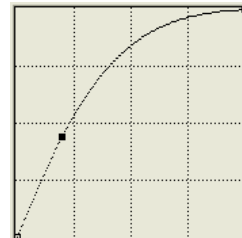
# Гамма-коррекция

Гамма-коррекция (коррекция яркости монитора изменением напряжения).

Преобразование вида:  $y = c \cdot x^\gamma$

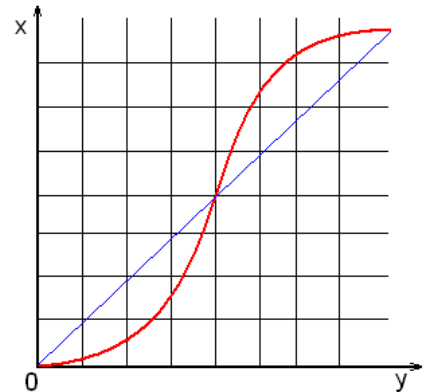


$\gamma > 1$



$\gamma < 1$

Графики функции  $f^{-1}(y)$



Возможный график функции  $f^{-1}(y)$



# Нелинейная коррекция.

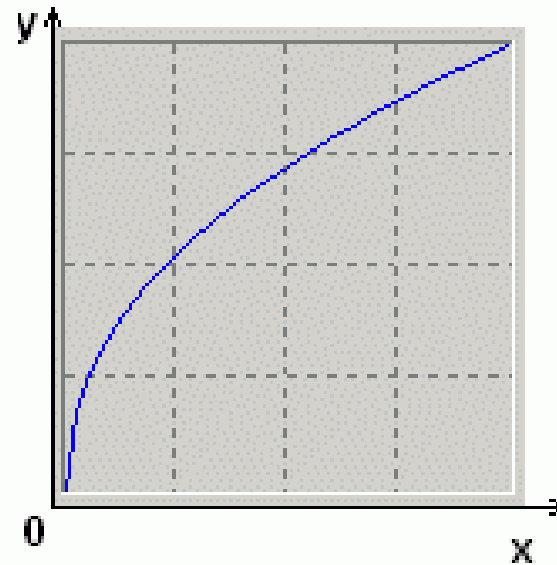
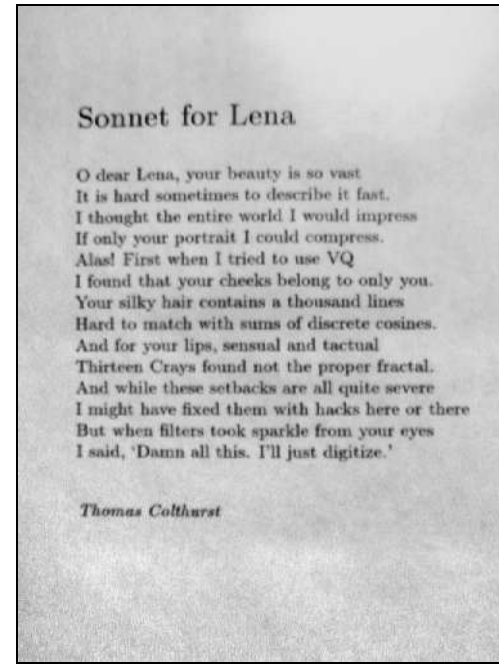
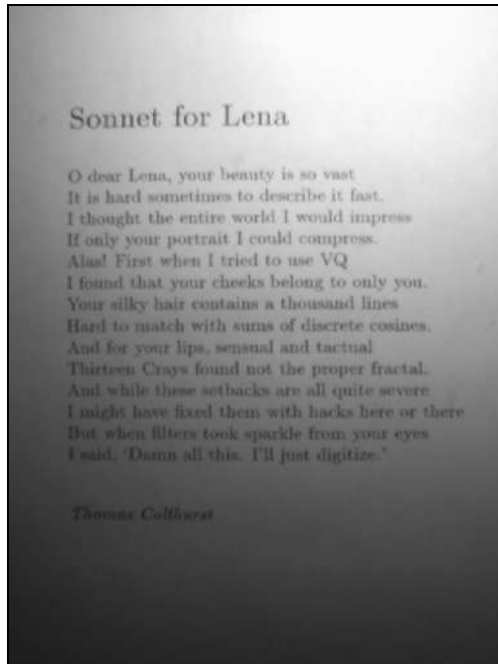


График функции  $f^{-1}(y)$

*Растянуты низкие и сжаты высокие интенсивности*

# Компенсация разности освещения

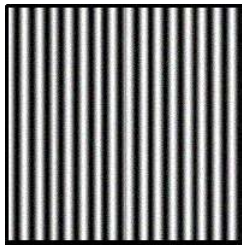


# Компенсация разности освещения

Изображение факторизуется на произведение

$$I(i, j) = l(i, j) \cdot f(i, j)$$

плавных изменения  $l(i, j)$  яркости относятся к освещению,  
резкие  $f(i, j)$  - к объектам.



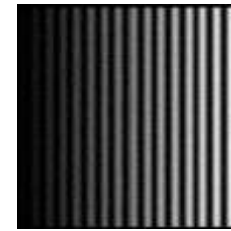
объект

$f(i, j)$



освещение

$l(i, j)$



Изображение  
освещенного  
объекта

$I(i, j)$

# Выравнивание освещения. Алгоритм

- **Алгоритм**

- Получить компонент освещения путем низкочастотной фильтрации  $G$  изображения

$$l'(i, j) = I(i, j) * G$$

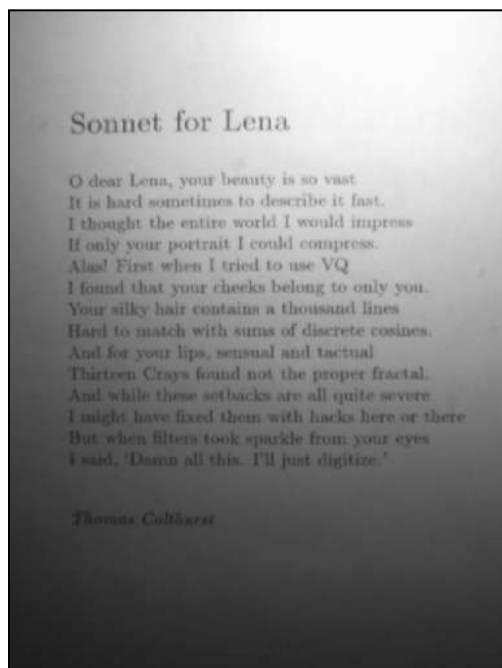
- Восстановить изображение по формуле

$$f'(i, j) = \frac{I(i, j)}{l'(i, j)}$$

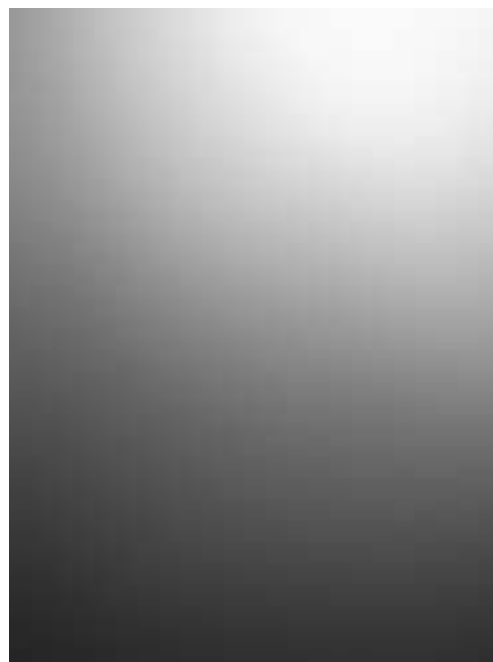


# Компенсация разности освещения.

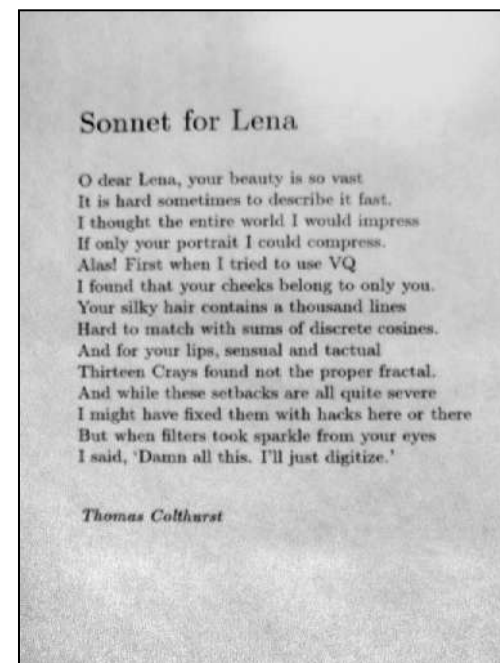
## Пример



/

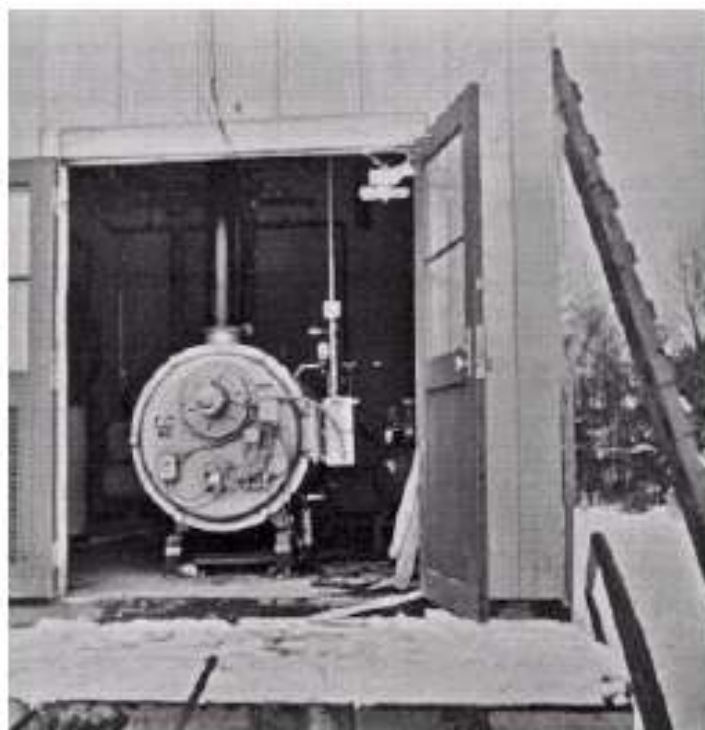


=



Gauss 14.7 пикселей

## Выравнивание освещения.



## *Цветовая коррекция - 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»



# «Серый мир»



Один источник  
света слева



Один источник  
света спереди



Два источника  
света: слева и  
спереди

# «Идеальный отражатель»

- Предположение:
  - Наиболее яркие области изображения относятся к бликам на поверхностях, модель отражения которых такова, что цвет блика = цвету освещения;  
(дихроматическая модель)
- Метод
  - Обнаружить максимумы по каждому из каналов:

$$R_{\max}, G_{\max}, B_{\max}$$

- Масштабировать яркости пикселов:

$$R * \frac{255}{R_{\max}}; \quad B * \frac{255}{B_{\max}}; \quad G * \frac{255}{G_{\max}};$$

# Цветовая коррекция изображений

- Растяжение контрастности (“autolevels”)
  - Идея – растянуть интенсивности по каждому из каналов на весь диапазон;
- Метод:
  - Найти минимум, максимум по каждому из каналов:
  - Преобразовать интенсивности:

$$R_{\min}, R_{\max}, G_{\min}, G_{\max}, B_{\min}, B_{\max}$$

$$(R - R_{\min}) * \frac{(255 - 0)}{(R_{\max} - R_{\min})};$$

$$(G - G_{\min}) * \frac{(255 - 0)}{(G_{\max} - G_{\min})};$$

$$(B - B_{\min}) * \frac{(255 - 0)}{(B_{\max} - B_{\min})};$$

# Растяжение контрастности (“autolevels”)



Эффект изменения  
цветовой гаммы  
изображения

# Коррекция с опорным цветом

- Предположение
  - Пользователь указывает целевой цвет (например: белый, серый, черный) вручную;
- Источники для указания целевого цвета:
  - Знание реального цвета
  - Хорошая фотография этой же сцены
- Метод
  - Преобразовать по каждому из каналов цвета по формуле:

$$R * \frac{R_{dst}}{R_{src}};$$

$$G * \frac{G_{dst}}{G_{src}};$$

$$B * \frac{B_{dst}}{B_{src}};$$

Здесь мы использовали коррекцию по белому цвету, зная априори что снег - “белый”, а небо – “светлое”



Исходное изображение



Результат коррекции



## Пример изменения цвета отдельной области.

Мы выделили область желтого цвета и скорректировали только ее, указав в качестве опорных цветов **розовый** и **красный** из коллекции стандартных цветов





## *Борьба с шумом - примеры*

# Причины и примеры шума изображения

- Подавление и устранение шума
  - Причины возникновения шума:
    - Несовершенство регистрирующих приборов
    - Хранение и передача изображений с потерей данных



Шум фотоаппарата

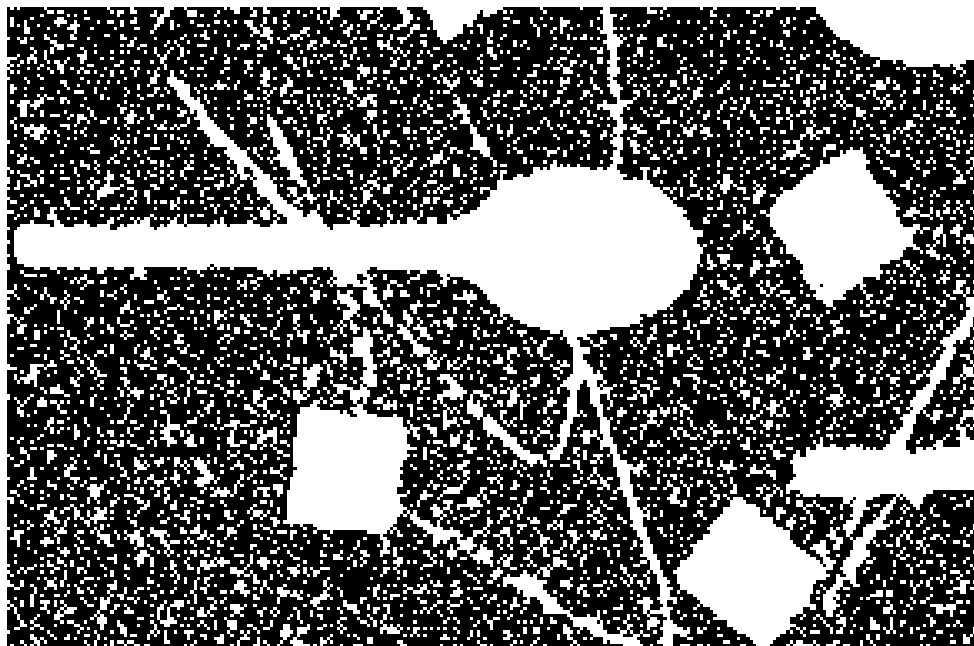


Сильное сжатие JPEG

# Шум в бинарных изображениях

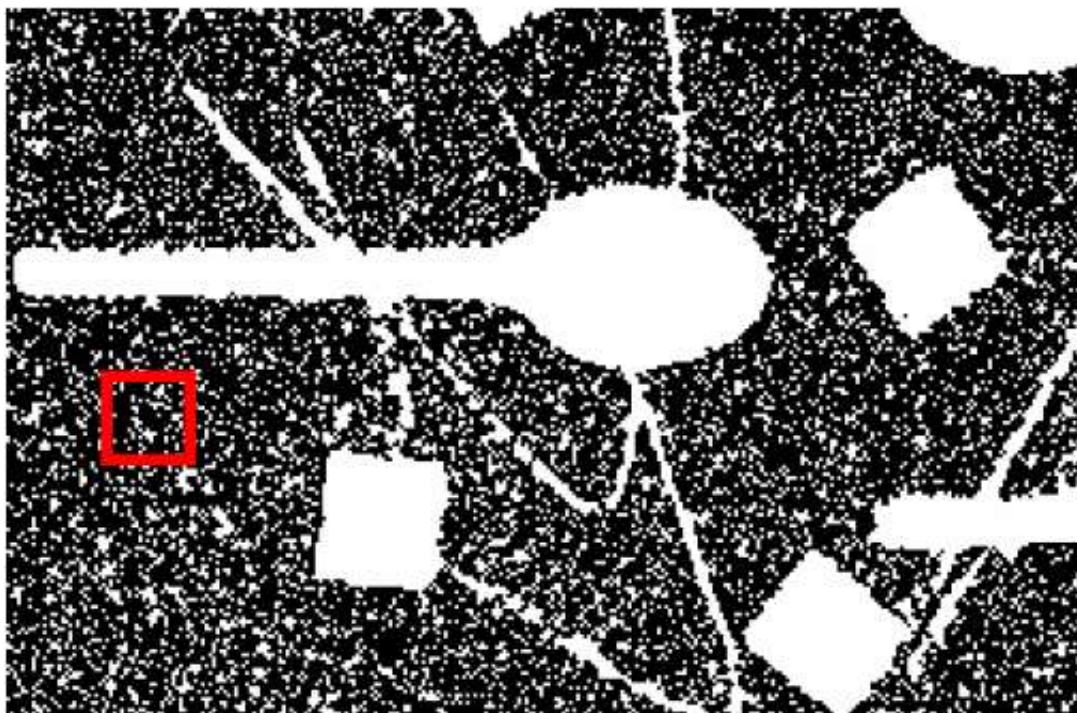
Бинарное изображение – изображение, пиксели которого принимают всего два значения (0 и 1).

Пример бинарного изображения с сильным шумом:



По одному пикселю невозможно определить – шум или объект?

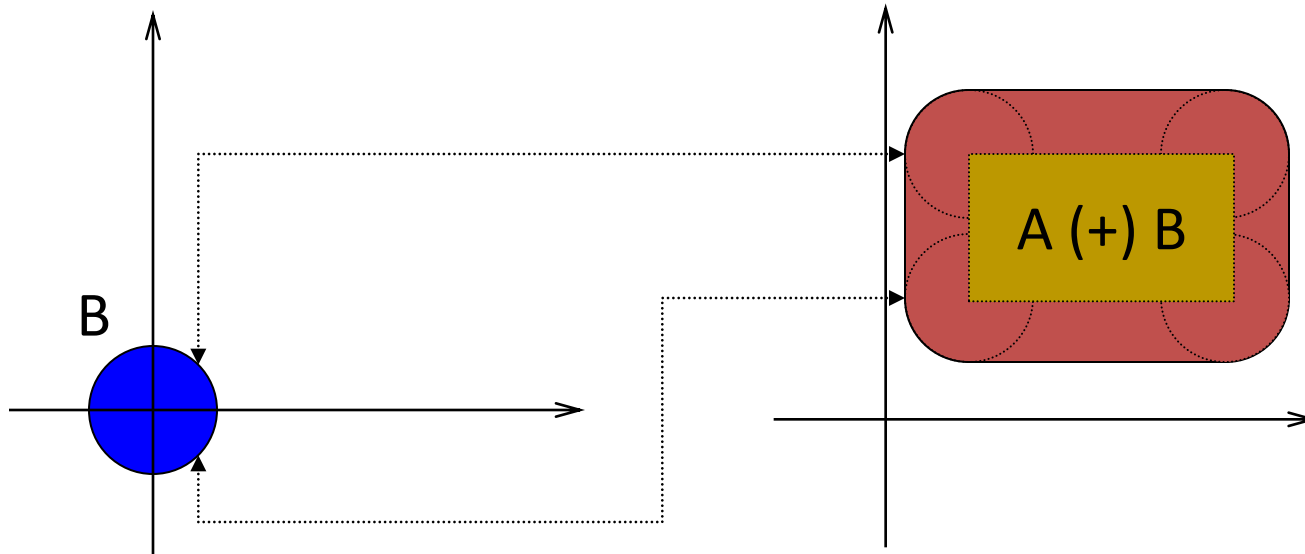
Нужно рассматривать окрестность пикселя - **операции математической морфологии.**



# Расширение

Расширение (dilation)

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



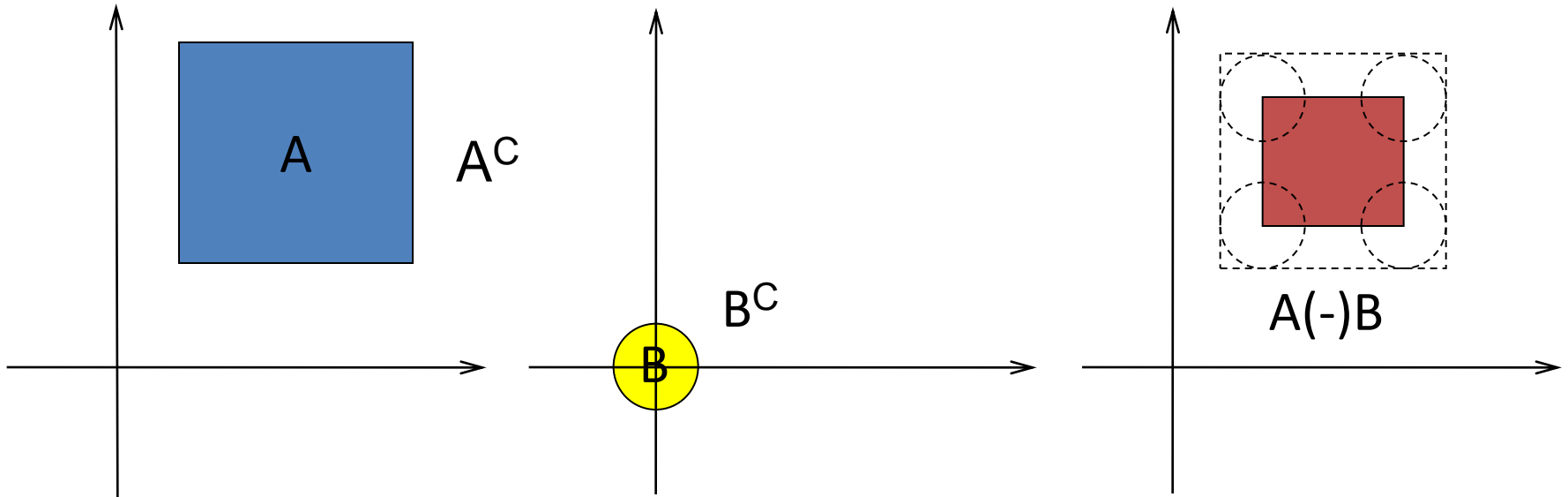
Множество **A** обычно является объектом обработки, а множество **B** (называемое структурным элементом) – инструментом.

# Сужение

Сужение (erosion)

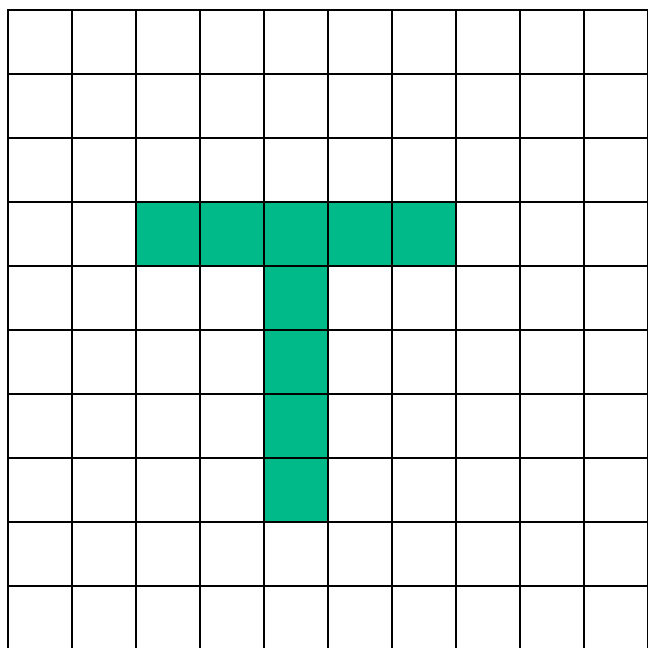
$A (-) B = (A^C (+) B)^C$ , где  $A^C$  – дополнение  $A$

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

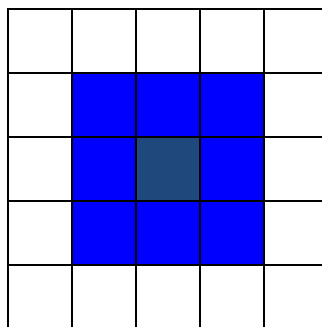


# Дискретные операции морфологии

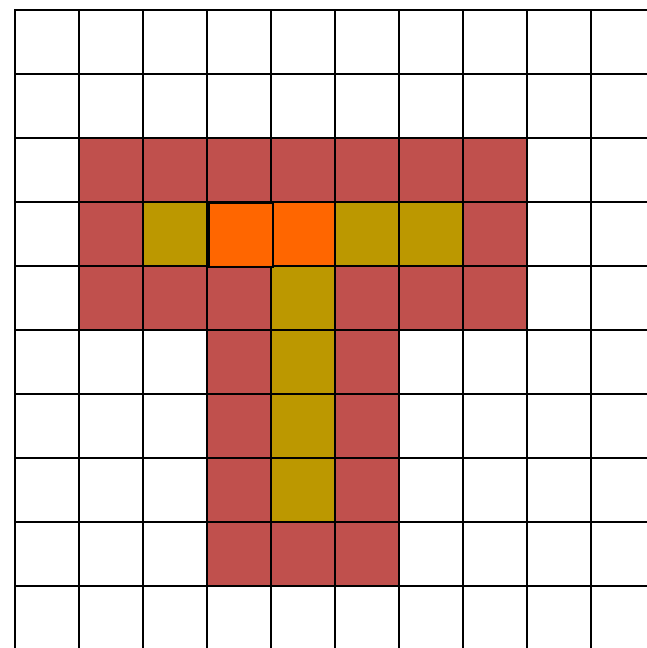
A



B



A(+)<sub>B</sub>



# Свойства морфологических операций

Коммутативный закон

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

Ассоциативный закон

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

Две основных операции морфологии используют элементы структурирования:

- erosion (эрозия)
- dilatation (расширение)

Более сложные операции морфологии:

- opening (открытие)
- closing (заккрытие)
- morphological gradient (градиент)
- top hat (цилиндр)
- black hat (Эффект черной шляпы)



образ



Erosion  $I(-)B$



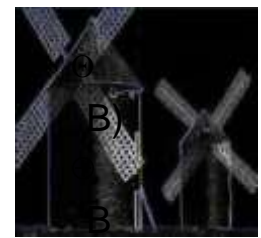
Dilatation  $I\oplus B$



Opening  $I\circ B = (I(-)B)(+)B$



Closing  $I\bullet B = (I(+)B)(-)B$



Grad(I) =  $(I(+)B) - (I(-)B)$



Top Hat  $(I) = I - (I(-)B)$



Black Hat  $(I) = (I(+)B) - I$



# Применения сужения к бинарному изображению с сильным шумом



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



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



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

## Плохой объект для морфологии

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



## Операция «свертка» (convolution)

Свертка двумерной функции  $f$  по функции  $g$  в непрерывном и дискретном случае.

$$\langle f * g \rangle (i, j) = \sum_{l=n_0}^{n_1} \sum_{k=m_0}^{m_1} f(i-l)(j-k) \cdot g(l, k)$$

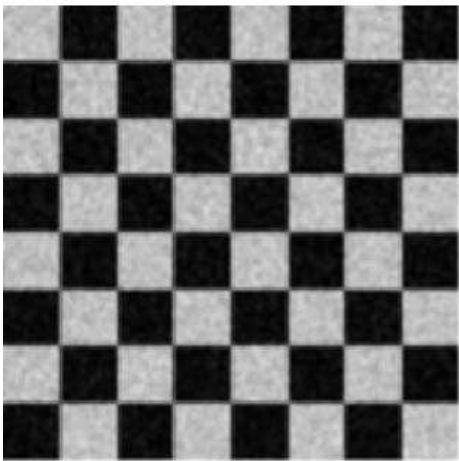
Свертка изображения по какой-либо функции называется применением **фильтра** к изображению.

## Усреднение (box filter)

Операция усреднения значения каждого пикселя – свертка по **константной** функции:

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

Результат применения:



# Медианный фильтр

Устранение шума в полутоновых, цветных и бинарных изображениях с помощью медианного фильтра - выбор медианы среди значений яркости пикселей в некоторой окрестности.

**Определение медианы:**

$A_i, i = \overline{1, n}$ ; - отсортированный набор чисел,

$A_{[n/2]}$  – медиана набора.

**Медианный фильтр радиусом  $r$  – выбор медианы среди пикселей в окрестности  $[-r, r]$ .**

# Пример очистки изображения с помощью медианного фильтра

Фильтр с окрестностью 3x3



# Фильтр Гаусса (gaussian blurring)

Свертка по функции Гаусса:

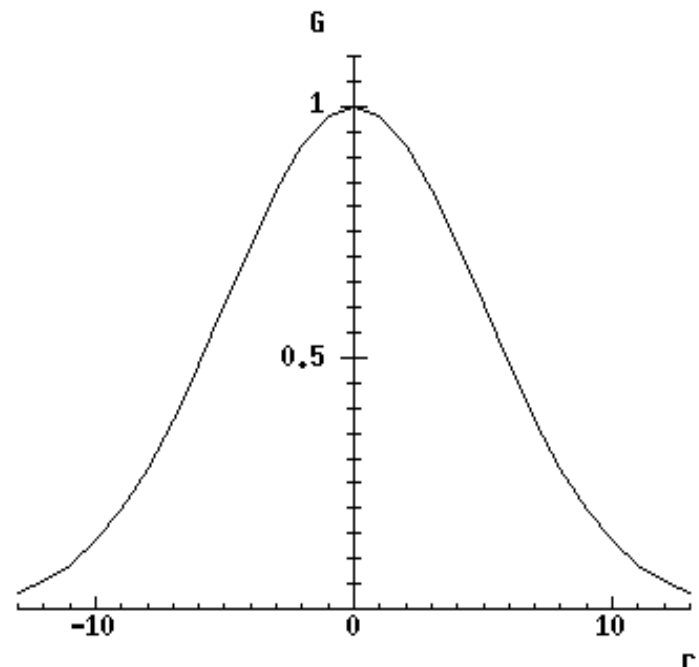
$$I'(i, j) = \sum_{l=-n}^n \sum_{k=-m}^m I(i-l)(j-k) \cdot \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{d^2}{2\sigma^2}}$$

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

Параметр  $\sigma$  задает степень размытия.

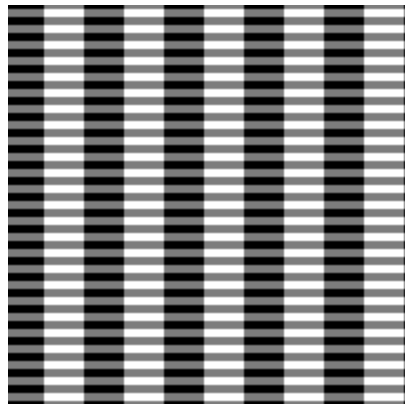
На графике функция с

$$\sigma = 5$$

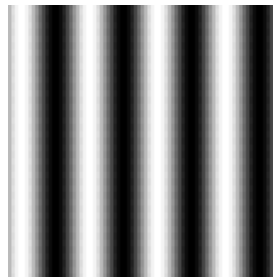


# Фильтр Гаусса (gaussian blurring)

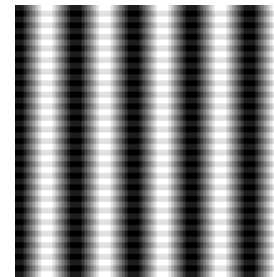
Результаты свертки по функции Гаусса и по константной функции (усреднения).



Исходное изображение



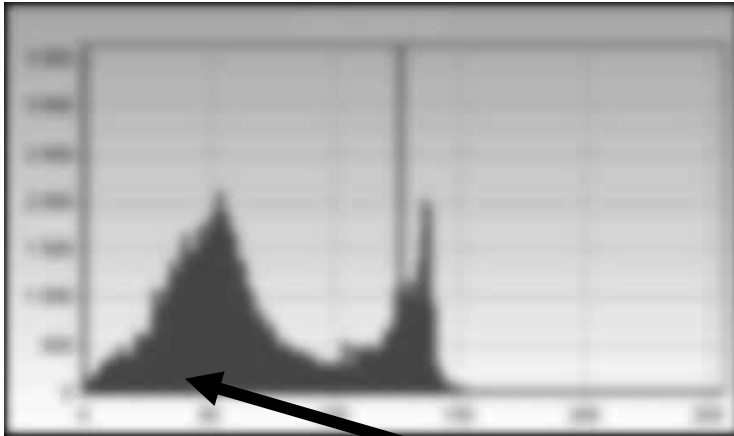
Фильтр Гаусса с  
Sigma = 4



Усреднение по 49  
пикселям (7x7)

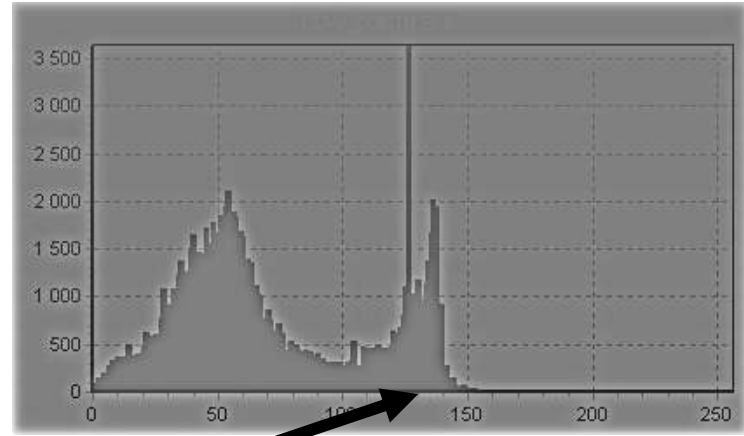


# Преобразование Фурье

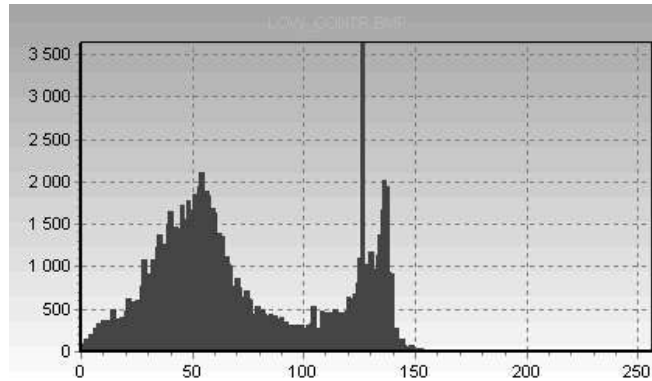


Низкие частоты

+

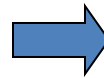


Высокие частоты



# «Продвинутые» фильтры

- Примеры таких фильтров (см. лек. А. Лукина, G&MLab ВМК МГУ)



## Художественные фильтры – “акварель”.



## 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
- Адаптивные фильтры – меньше портят детали, зависят от большего числа параметров. Иногда изображение становится менее естественным.
- «Продвинутые» фильтры – лучшее сохранение деталей, меньше размытие. Часто сложны в реализации и очень медленные.

## Как бороться с шумом аппаратуры?

Предположим, что прибор (камера), которой производится измерение (съемка) заметно «шумит». Обычно **шум измерительной аппаратуры** моделируется как **нормально распределенная случайная величина с нулевым средним** -

$$Err(i, j);$$

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

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

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

Raw and jpeg от Fuji

# Примеры шумоподавления



Зашумленные  
изображения



Усреднение по  
10  
изображениям

# Примеры шумоподавления



Усреднение по 9  
пикселям (3x3)

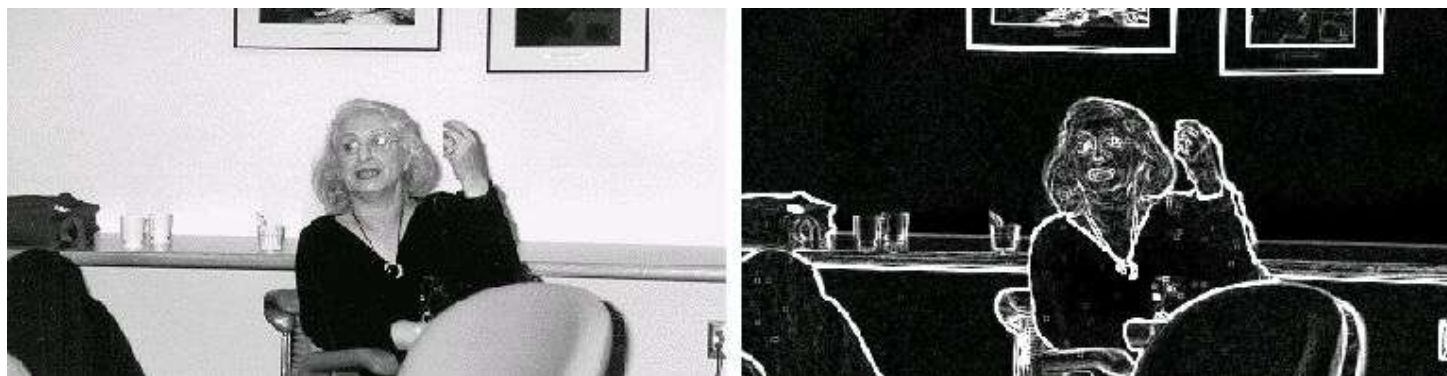


Медианный фильтр (3x3)

# Выделение контуров объекта

Рассмотрим выделение краев (границ)

*Край (edge) – резкое изменение яркости на изображении, часто соответствует границам объектов на изображении.*

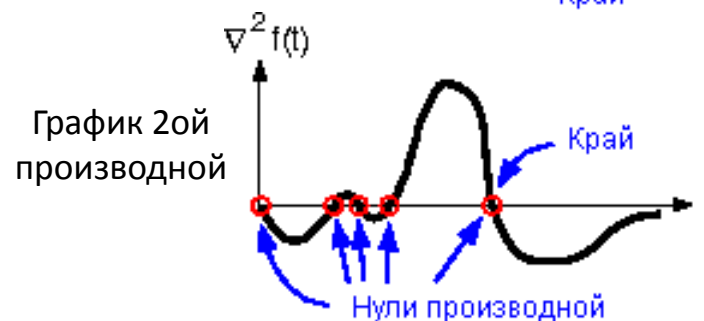
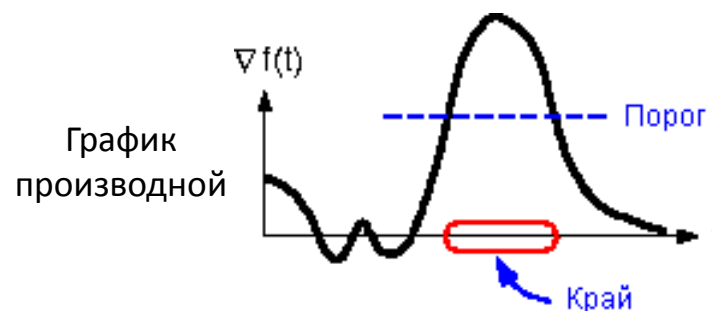
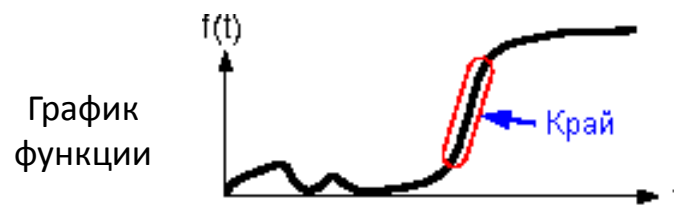




# Выделение точек контура

Нас интересуют области **резкого изменения яркости** –  
нахождение таких областей  
можно организовать на основе  
анализа первой и второй  
производной изображения.

Интерес - анализ примитивов для  
распознавания объектов.



## Выделение точек контура. Градиент

Наибольшее изменение функции происходит в направлении ее градиента. Величина изменения измеряется абсолютной величиной градиента.

$$\nabla I(x, y) = \left( \frac{\partial I}{\partial x}(x, y), \frac{\partial I}{\partial y}(x, y) \right);$$

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

**Приближенное вычисление градиента:**

$$|\nabla I(x, y)| \cong \left| \frac{\partial I}{\partial x}(x, y) \right| + \left| \frac{\partial I}{\partial y}(x, y) \right|$$

# Приближения (маски) Робертса, Превитта и Собеля

Семейство методов основано на приближенном вычислении градиента, анализе его направления и абсолютной величины.

**Свертка по функциям:**

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

Робертса

Превитта

Собеля

Математический смысл – приближенное вычисление производных по направлению.

# Выделение точек контура.

Примеры применения операторов подчеркивания краев:



Робертса



Превитта



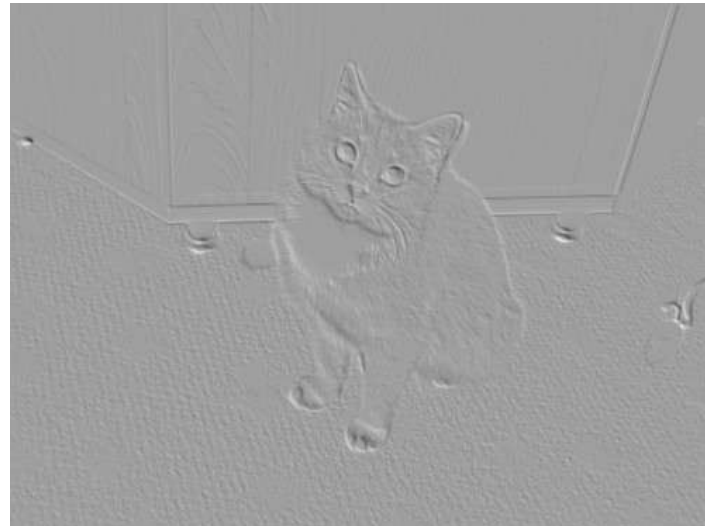
Собеля

# *Спецэффекты*

# Тиснение

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

Фильтр + сдвиг яркости, нормировка...



## Цифровой негатив



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



## Перенос/поворот



Перенос:

$$x(k; l) = k + 50; y(k; l) = l;$$

Поворот:

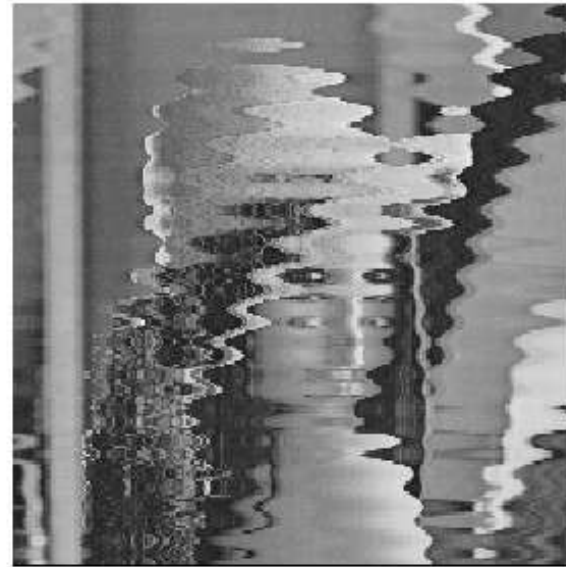
$$x(k; l) = (k - x_0) \cos(\mu) + (l - y_0) \sin(\mu) + x_0;$$

$$y(k; l) = (k - x_0) \sin(\mu) + (l - y_0) \cos(\mu) + y_0;$$

$$x_0 = y_0 = 256.5 \text{ (центр поворота)}, \mu = \pi/6$$



## «Волны»



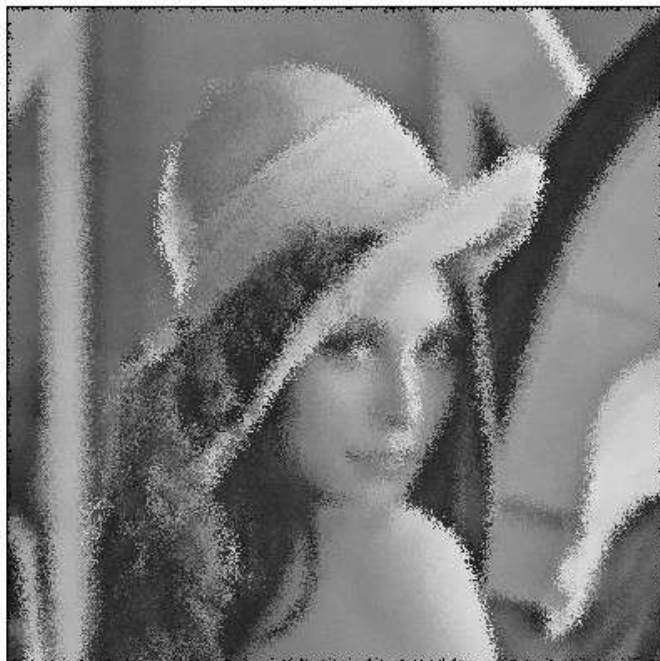
Волны 1:

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

Волны 2:

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

## Эффект «стекла»



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

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

1. <http://graphics.sc.msu.su/courses/> - курсы "Введение в компьютерную графику" Баяковского Ю.М. и Шикина Е.В. для ф-та ВМиК МГУ, материалы лаборатории Graphics & Media Lab при МГУ, в т.ч. <http://graphics.cs.msu.su/courses/cg/assigns/2005/hw2/index.html>  
<http://graphics.cs.msu.su/courses/cg99/notes/lect5/notes05.htm> ...
2. <http://www.intuit.ru/shop/product-2493469.html> - Иванов Д.В., Карпов А.С., Кузьмин Е.П., Лемпицкий В.С., Хропов А.А, Алгоритмические основы растровой машинной графики. БИНОМ. Лаборатория знаний, Интернет-университет информационных технологий - ИНТУИТ.ру, 2007.
3. Порев В.Н. Компьютерная графика. –СПб.: БХВ-Петербург, 2002. –432с.
4. Никулин Е.А. Компьютерная геометрия и алгоритмы машинной графики, . –СПб.: БХВ-Петербург, . –СПб.: БХВ-Петербург 2003б-560 с.
5. Шикин Е.В., Боресков А.В. Компьютерная графика. Полигональные модели. –М.: ДИАЛОГ-МИФИ, 2001.-464с.
6. Color& Shading (из книги Linda Shapiro& George Stockman "Computer Vision") (pdf, 1.23MB, eng.)  
<http://graphics.cs.msu.su/courses/cg01b/ch6.pdf>
7. JohnC. Russ. "Image Processing. Handbook"
8. Хилл Ф. OpenGL. Программирование компьютерной графики. – С.Пб: Питер, 2002. 1088с.
9. Gary Bradski and Adrian Kaehler. Learning OpenCV/ Published by O'Reilly Media, Inc., 2008. -577pp.
10. Open Source Computer Vision Library. *Reference Manual*. Copyright © 1999-2001 Intel Corporation. Issued in U.S.A. Order Number: 123456-001 (<http://developer.intel.com>)