



LOBACHEVSKY  
UNIVERSITY

Information Technology, Mathematics & Mechanics (ITMM) institute  
Software & Supercomputing Technology department

## CS255. Computer Graphics Introduction Course

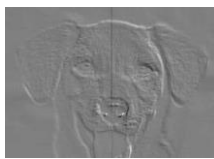
# Обработка изображений. Часть 1

Турлапов Вадим Евгеньевич  
проф. каф. МОСТ, ИТММ, ННГУ

# Techniques & Applications

## Some Techniques:

- **Image filtering**: correction and adjusting, special effect creation
- **Image binarization** (conversion to b & w)
- **Image Segmentation**
- **Morphological** processing of image



Dot filters

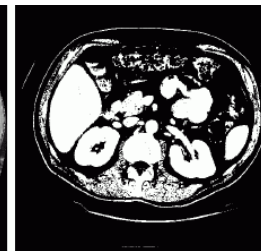
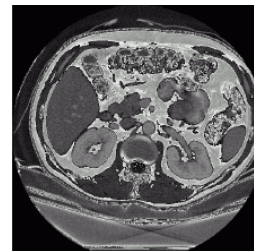
Emboss  
Transform



Matrix filters

## Some Applications:

- Computer vision, video-surveillance
- Feature detection, face detection
- Noise reduction and removal
- Medical including microscope 3D image processing
- Hyperspectral 3D Images

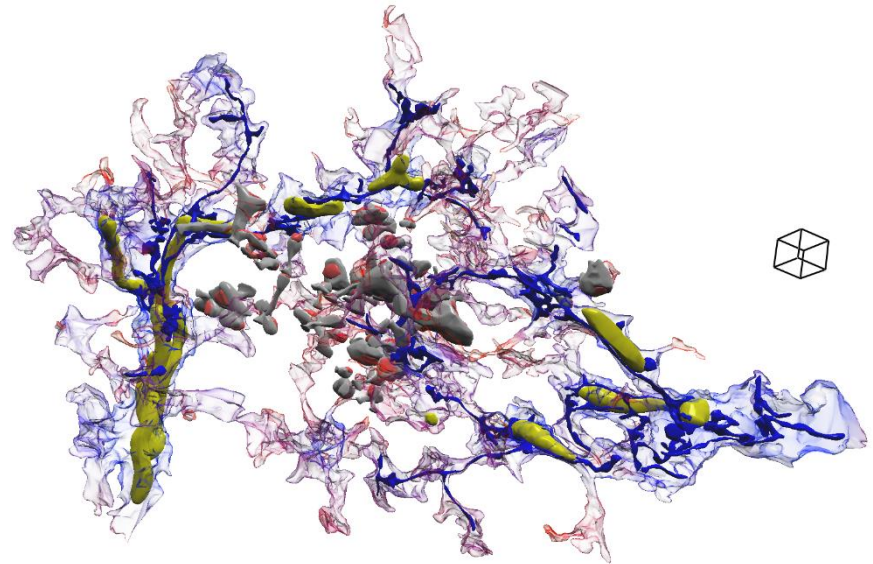
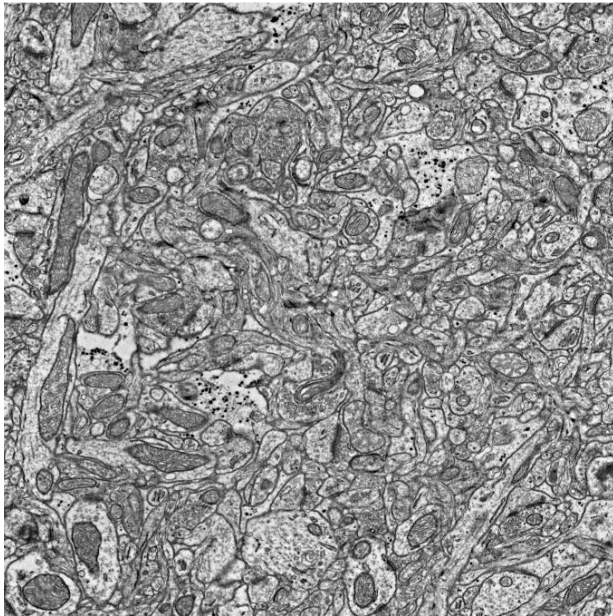


Medical image processing: filtering; binarization; morphological processing.

# Какие изображения мы обрабатываем:

## Изображения электронной микроскопии

---



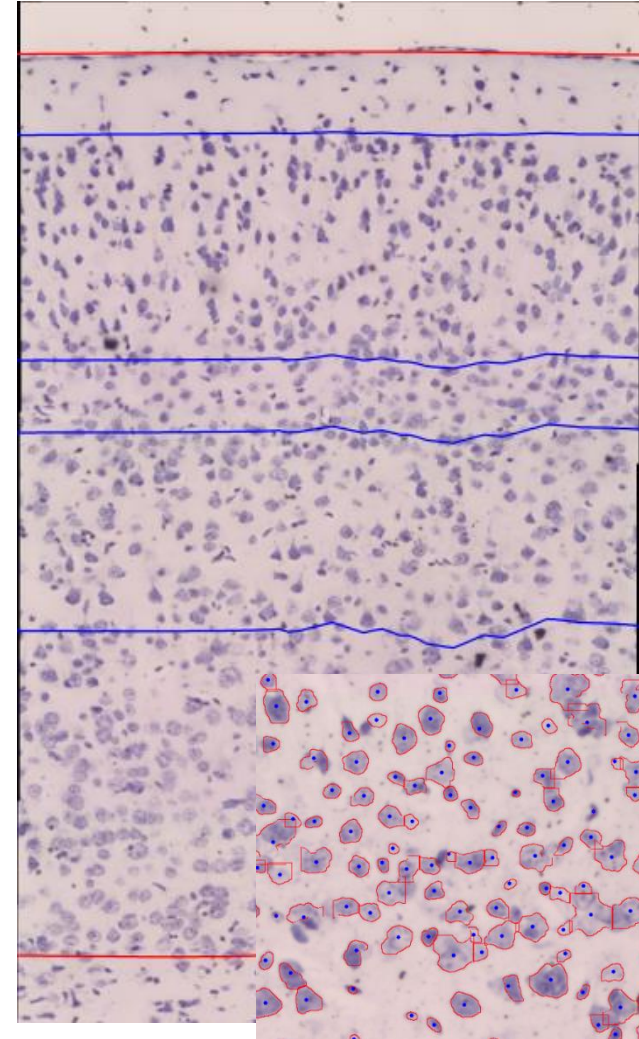
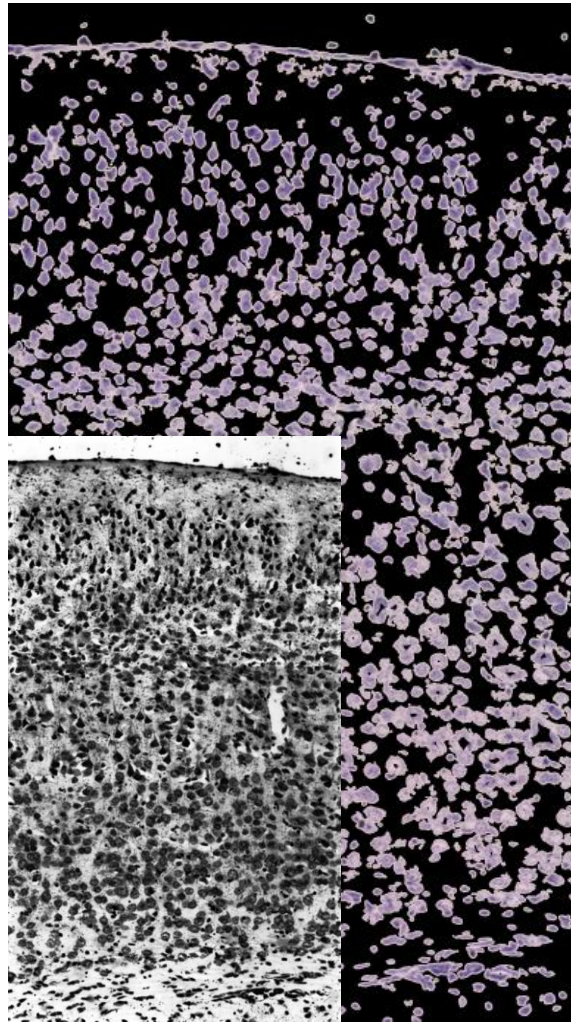
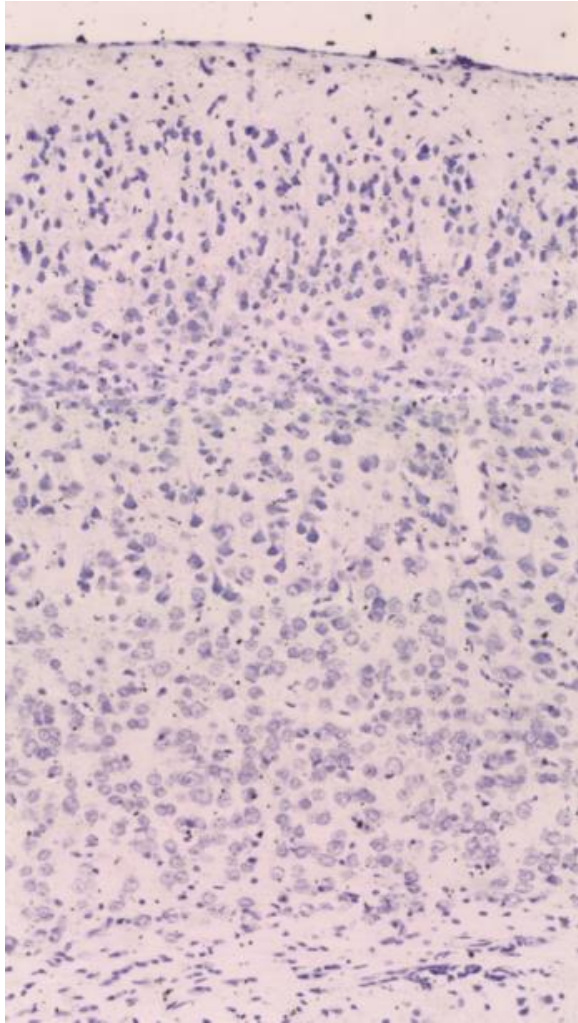
ILASTIK (<http://ilastik.org>) - user-friendly tool for interactive image classification, segmentation and analysis. It has automated (supervised) pixel- and object-level classification, automated and semi-automated object tracking, semi-automated segmentation and object counting without detection. Most analysis operations are performed lazily.

ILASTIK was developed by Heidelberg Uni., Germany under support of [Heidelberg Collaboratory for Image Processing](#) (HCI); HHMI (Howard Hughes Medical Institute) Janelia Farm Research Campus (<http://www.janelia.org/>); CellNetworks Excellence Cluster ([www.cellnetworks.uni-hd.de](http://www.cellnetworks.uni-hd.de))



# Какие изображения мы обрабатываем:

## Изображения оптической микроскопии



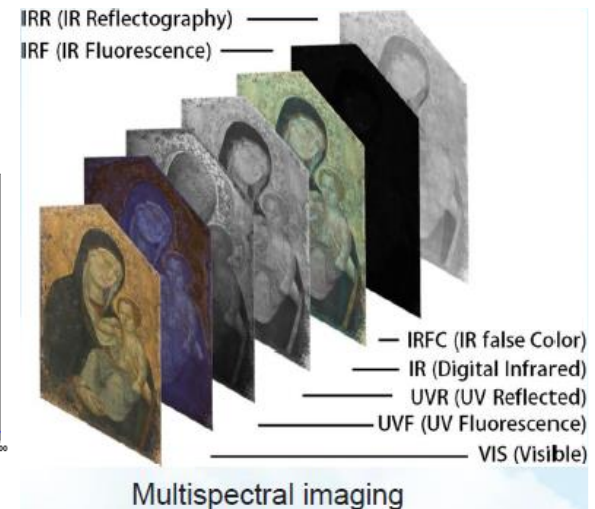
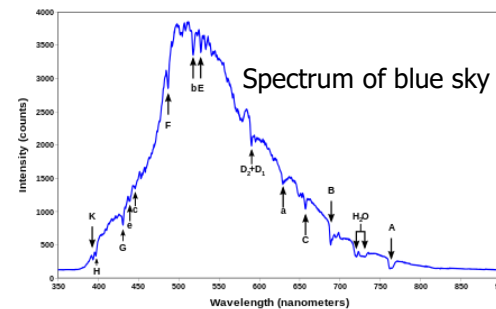
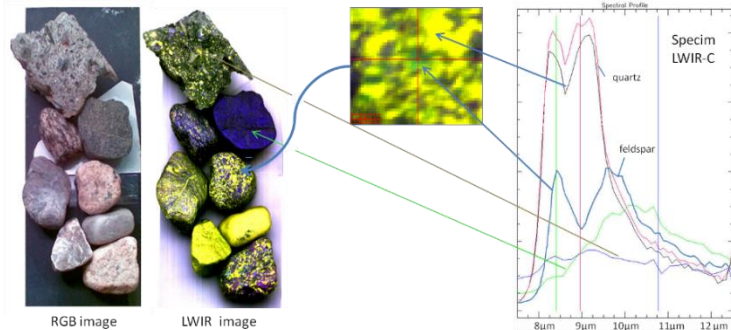
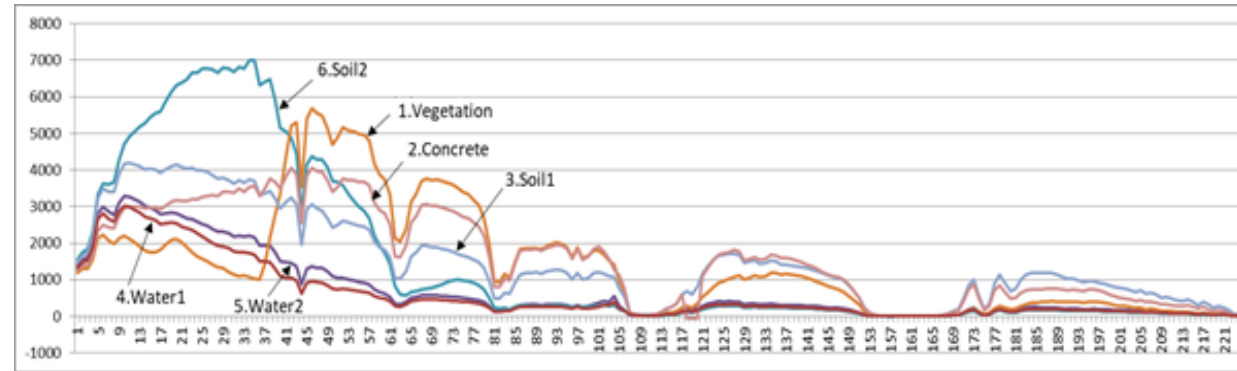


# Какие изображения мы обрабатываем: Гиперспектральные изображения (ГСИ)



## Spectral Signature

[https://en.wikipedia.org/wiki/Spectral\\_signature](https://en.wikipedia.org/wiki/Spectral_signature)



# Введение

## What are the multi- and hyper-spectral images (MSI, HSI)?

**Проблема:** комплексный анализ и мониторинг объектов окружающей среды на основе данных MSI и, особенно, гиперспектральных изображений (HSI, ГСИ)

### Основная область применения сегодня – ДЗ Земли.

Типичный диапазон спутниковых и авиационных спектрометров на примере AVIRIS: 360-2500нм (0.36-2.5мкм), 224 канала

Для сравнения: Основная часть собственного излучения кожи человека приходится на диапазон волн с длиной от 4 до 50 мкм.

**Ближний ультрафиолет NUV:** 350-400нм

**Видимый диапазон:** 400-700нм.

**Инфракрасный (IR)**

Near-infrared, **NIR:** 0,74-1,4 мкм

Short-wavelength infrared, **SWIR:** 1,4-3,0 мкм

Mid-wavelength infrared, **MWIR:** 3-8 мкм

Long-wavelength infrared, **LWIR:** 8-15 мкм

Far-wavelength infrared, **FIR:** 8-15 мкм

**CIE схема:**

**IR-A:** 700 нм — 1400 нм (0,7 мкм — 1,4 мкм)

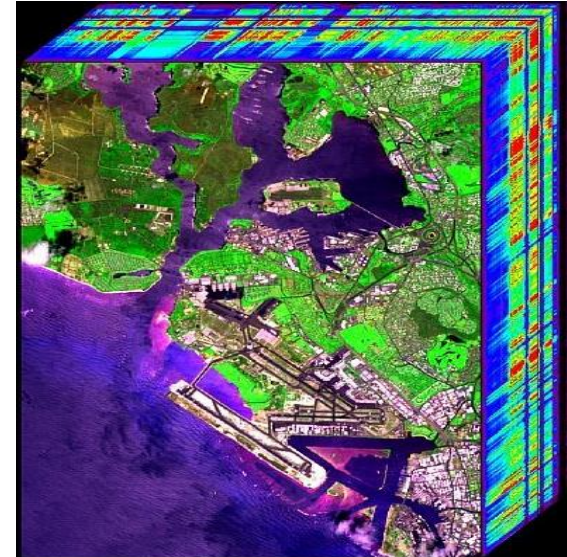
**IR-B:** 1400 нм — 3000 нм (1,4 мкм — 3 мкм)

**IR-C:** 3000 нм — 1 мм (3 мкм — 1000 мкм)

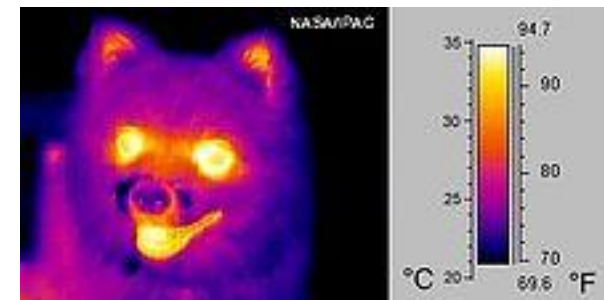
**ISO 20473 схема:**

**NIR** 0,78-3 мкм; **MIR** 3-50 мкм;

**FIR** 50-1000 мкм

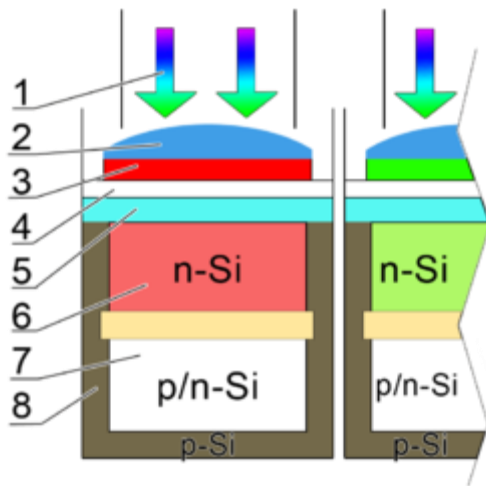


<https://directory.eoportal.org/web/eoportal/airborne-sensors/aviris>

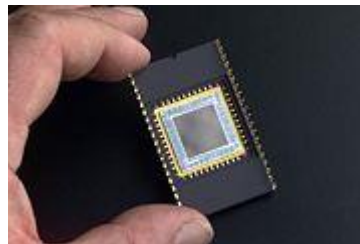
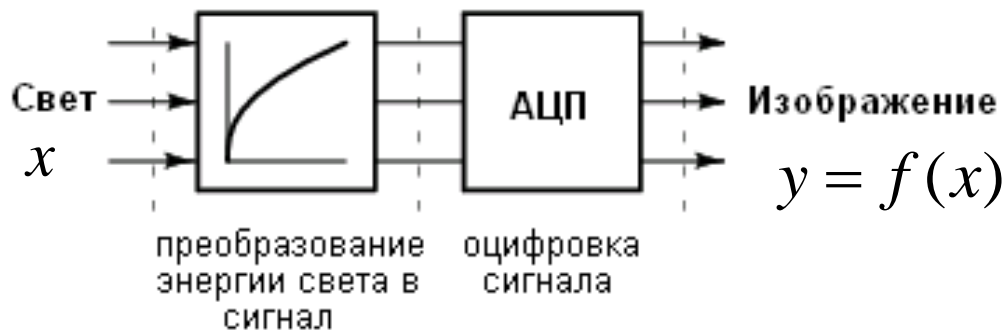


# Как получается цветное цифровое изображение в видео-регистрации

- ▶ 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 пиксела изображения

# Обработка изображений (IP)

---

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:

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

IP: 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
- ▶ Сегментация/Классификация? – попытки распознавания фрагментов



# Причины потери качества изображения

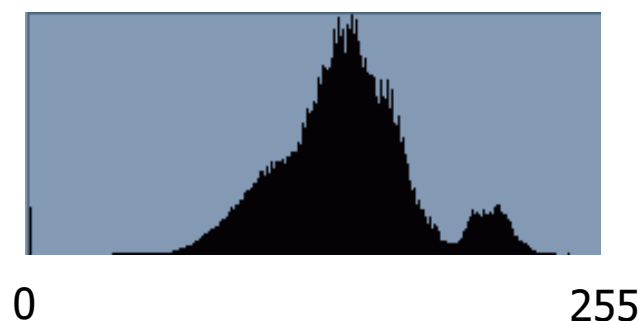
---

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



# Что такое гистограмма

**Гистограмма** — это график распределения интенсивности в изображении. На горизонтальной оси - шкала яркостей тонов от белого до черного, на вертикальной оси - число пикселей заданной яркости.



Вопрос: Сколько бинов на горизонтальной оси гистограммы?

# Точечные фильтры.

## 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:

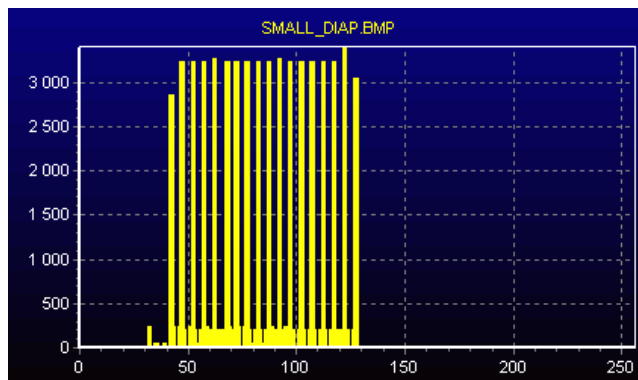
$$f^{-1}(y) = x$$

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



# Линейная коррекция

Компенсация узкого диапазона яркостей – линейное растяжение:



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

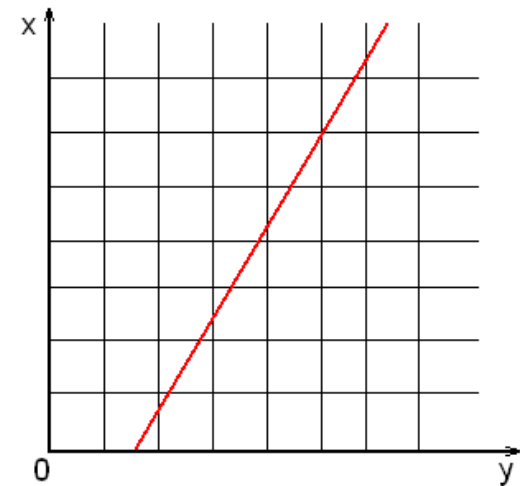
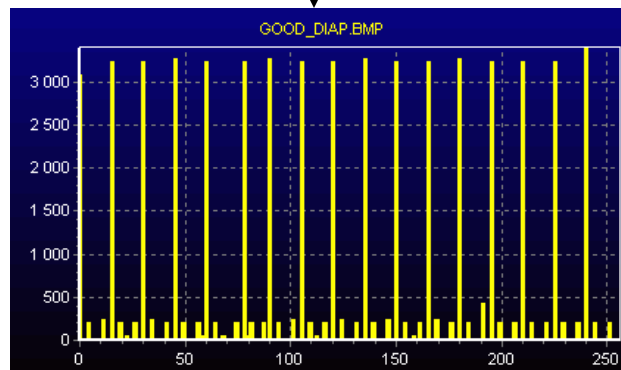
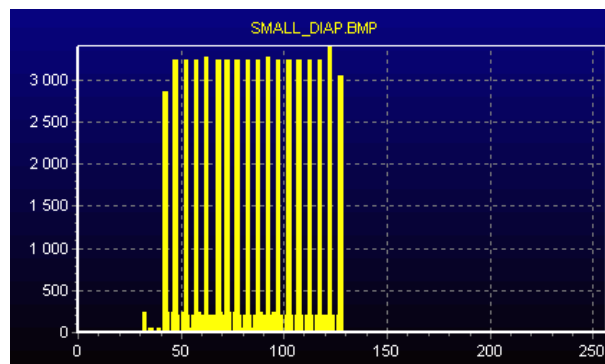


График функции  $f^{-1}(y)$

# Линейная коррекция. Результат

Компенсация узкого диапазона яркостей – линейное растяжение:



# Линейная коррекция. Пример

---

Линейное растяжение – «как AutoContrast в Photoshop»





# Линейная коррекция не всегда успешна

---

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.
- Логарифмическая
  - Цель – сжатие динамического диапазона при визуализации данных (связано с отображением HDR на обычный диапазон)

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

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

# Гамма-коррекция

Гамма-коррекция (коррекция яркости монитора изменением напряжения).

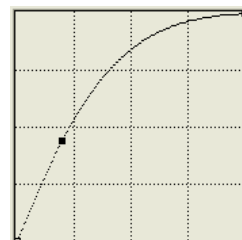
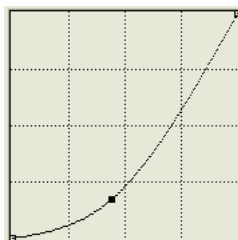
Так называют преобразование вида:  $y = c \cdot x^\gamma$



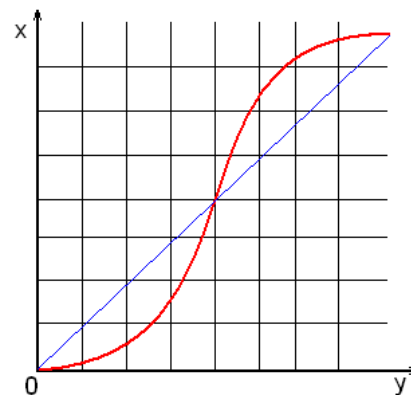
$\gamma > 1$



$\gamma < 1$



Графики функции  $f^{-1}(y)$



Возможный график функции  $f^{-1}(y)$

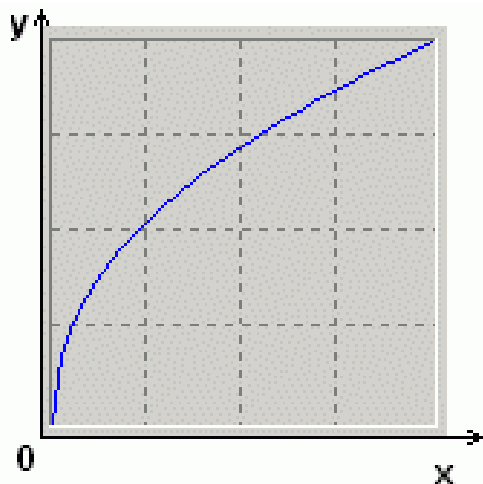


# Нелинейная коррекция. Пример



График функции  
 $f^{-1}(y)$

*Растянуты вдвое  
низкие и сжаты  
высокие  
интенсивности*



# Разд. 2. "Цветовая коррекция изображений". План

---

## Введение

1. Серый мир
  2. Коррекция "autolevels"
  3. Коррекция с опорным цветом
  4. Идеальный отражатель
  5. Статистическая цветокоррекция
- ▶ Изменение цветового баланса
    - ▶ Компенсация:
      - ▶ Неверного цветовосприятия камеры
      - ▶ Цветного освещения

# Гипотеза «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» - пример 1

---

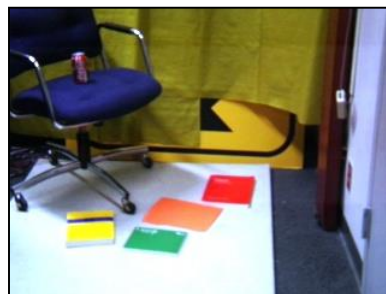
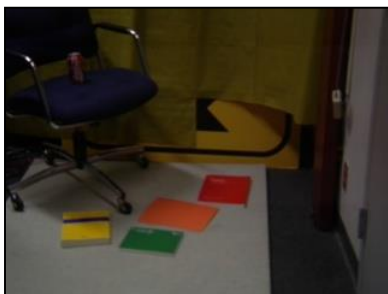


## «Серый мир» - пример 2

---



# «Серый мир» - пример 3



Один источник  
света слева



Один источник  
света спереди



Два источника  
света: слева и  
спереди

# Цветовая коррекция изображений

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

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

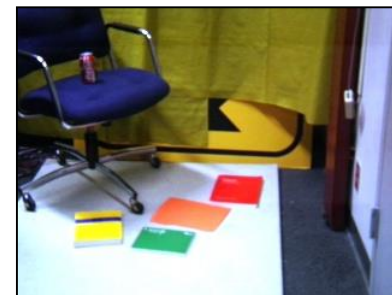
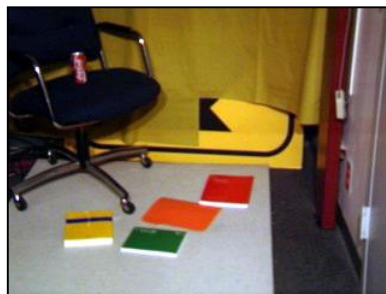
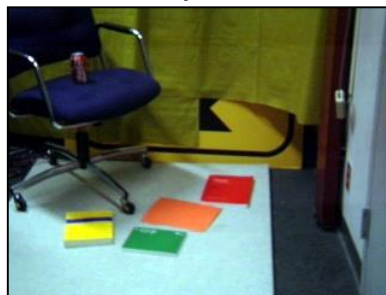
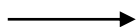
$$(R - R_{\min}) * \frac{(255 - 0)}{(R_{\max} - R_{\min})}; \quad (G - G_{\min}) * \frac{(255 - 0)}{(G_{\max} - G_{\min})};$$

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

# Растяжение контрастности ("autolevels")

Эффект изменения цветовой гаммы изображения

Серый мир





# Коррекция с опорным цветом

- ▶ Предположение
  - ▶ Пользователь указывает пиксел, цвет которого  $R_{src}, G_{src}, B_{src} > 0$ , но для которого известен правильный (целевой) цвет  $R_{dst}, G_{dst}, B_{dst} > 0$ .
  - ▶ Пользователь указывает целевой цвет вручную;
- ▶ Источники для указания целевого цвета:
  - ▶ Знание реального цвета
  - ▶ Хорошая фотография этой же сцены
- ▶ Метод
  - ▶ Преобразовать по каждому из каналов цвета по формуле:

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

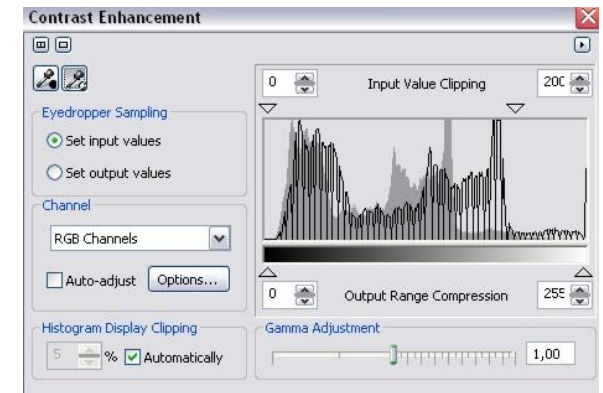
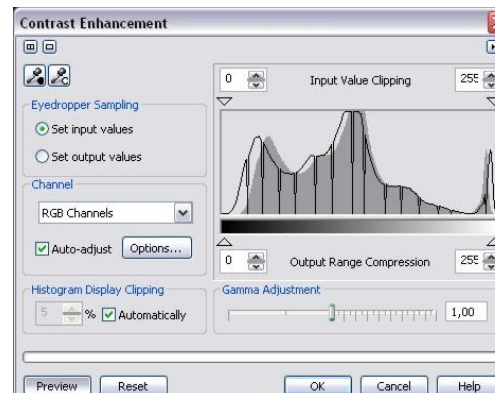
# Растяжение контрастности всех каналов ("autolevels")



**АВТО**



**По белому**



# Коррекция с опорным цветом. Примеры

## Коррекция по серому



# Гипотеза «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 пикселей:

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

# Статистическая цветокоррекция

Пример 1:



Пример 2:



Алгоритм состоит в том, что к каждому каналу каждого пиксела целевого изображения применяется следующее преобразование:

$$C_t^{new} = E_s + (C_t - E_t) \cdot \frac{\sigma_s}{\sigma_t}$$

где  $E$ ,  $\sigma$  – матожидание и среднеквадратическое отклонение исходного (s) и целевого (t) изображения;  $C_t$  – цвет целевого изображения. Лучше сделать в  $L^*a^*b^*$ .



# Устранение шума в бинарных изображениях.

---

## Матричные фильтры.

Операции математической морфологии  
в подавлении шума

## Разд.3. "Борьба с шумом"

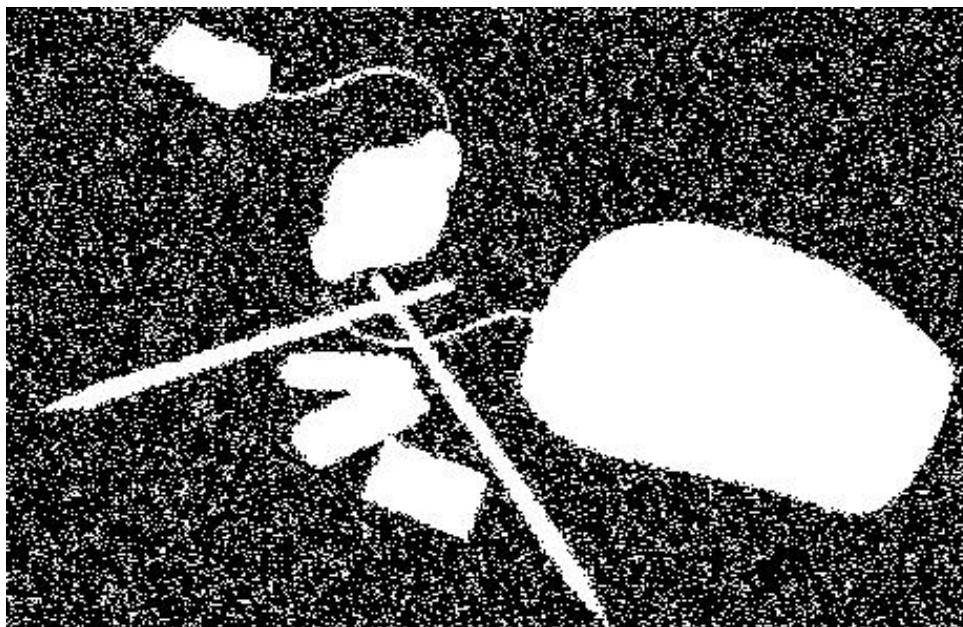
---

1. Шум в бинарных изображениях
2. Операции матморфологии. Расширение. Сужение.
3. Операции закрытия и раскрытия.
4. Примеры

# Шум в бинарных изображениях

---

Бинарное изображение – изображение, пиксели которого принимают всего два значения (0 и 1).  
Пример бинарного изображения с сильным шумом:



# Подавление и устранение шума

---

Устранение шума в бинарных изображениях

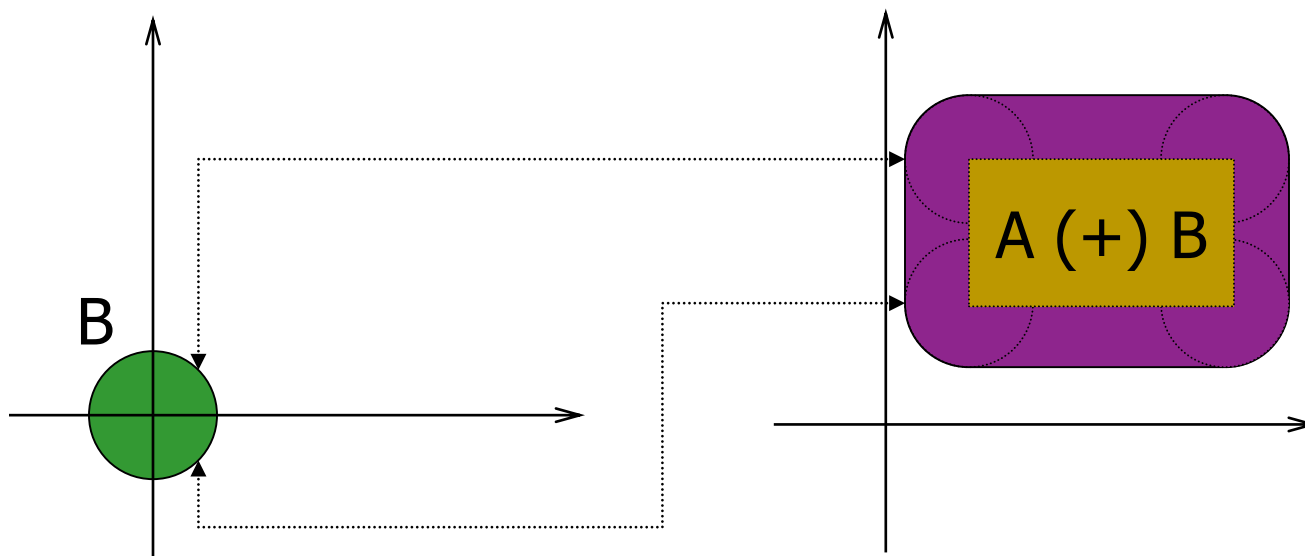
Широко известный способ - устранение шума с помощью операций математической морфологии:

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

# Операции матморфологии. Расширение

Расширение (dilation)

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



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

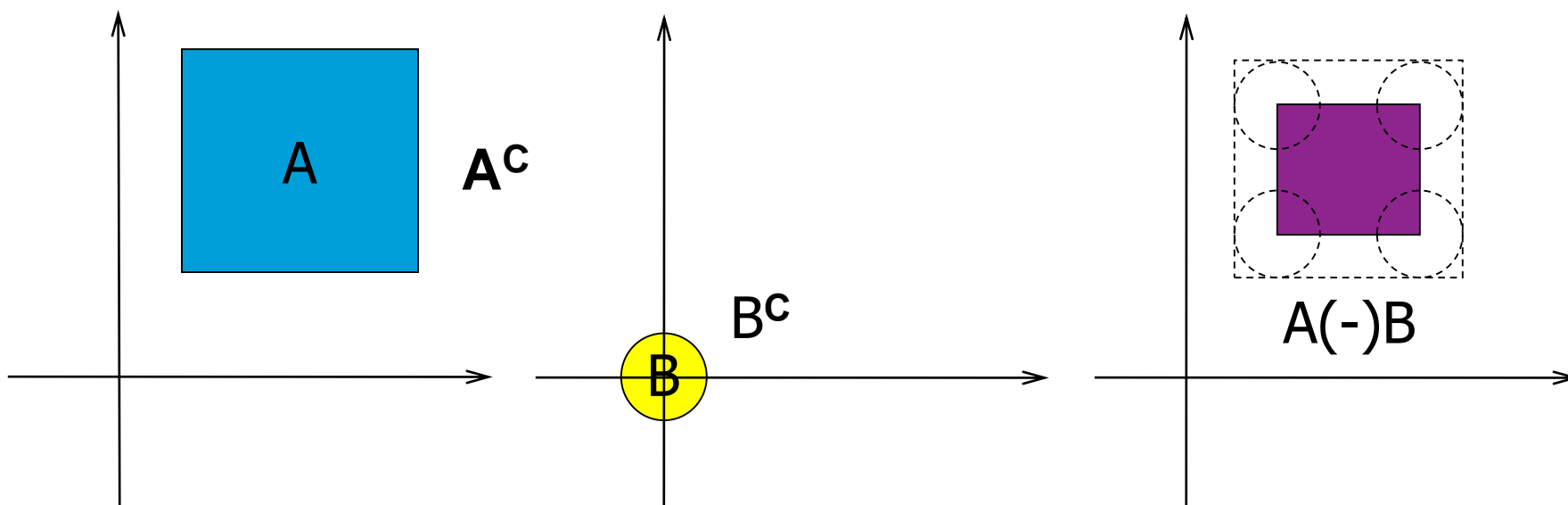


# Операции матморфологии. Сужение

Сужение (erosion)

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

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



# Свойства операции Dilation

## ► Коммутативность (Commutativity):

$$A \oplus B = B \oplus A$$

- Означает, что изображение и инструмент могут поменяться ролями

## ► Ассоциативность (Associativity):

$$I_1 \oplus (I_2 \oplus I_3) = (I_1 \oplus I_2) \oplus I_3$$

- Означает, что иногда мы можем разбить большой инструмент на совокупность маленьких:
- Если справедливо  $I_1 \oplus (I_2 \oplus I_3) = (I_1 \oplus I_2) \oplus I_3$
- То, большой структурный элемент  $B$

$$B = H_1 \oplus H_2 \oplus \dots \oplus H_n$$

может быть выполнен как:

$$A \oplus B = (\dots((A \oplus H_1) \oplus H_2) \oplus \dots \oplus H_n)$$

# Свойства Erosion

---

- ▶ Операция не коммутативна:

$$I(-)H \neq H(-)I$$

- ▶ Операция не ассоциативна, однако справедливо следующее равенство:

$$(I(-)H_1)(-)H_2 = I(-)(H_1 \oplus H_2)$$

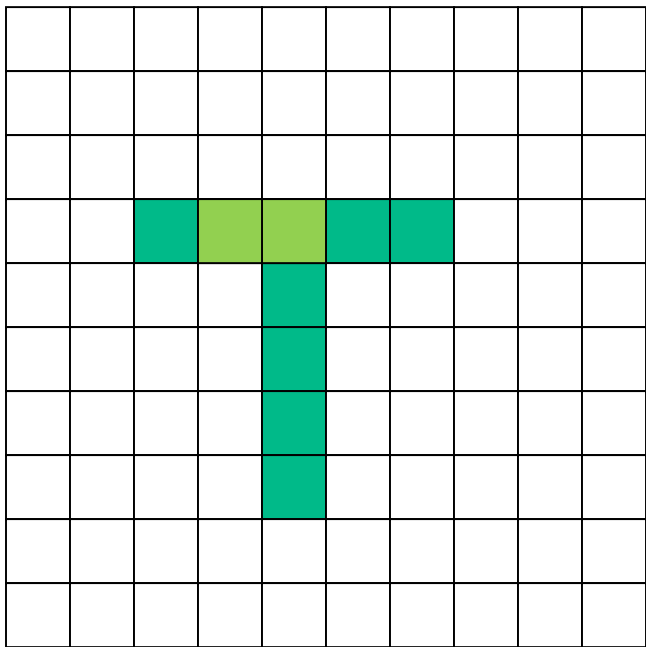
которое позволяет заменить последовательность сужений несколькими инструментами на сужение одним интегральным (первым, расширенным всеми остальными).

# Дискретные операции морфологии.

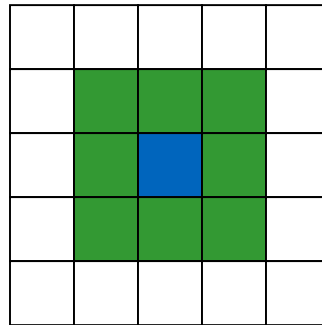
## Расширение

---

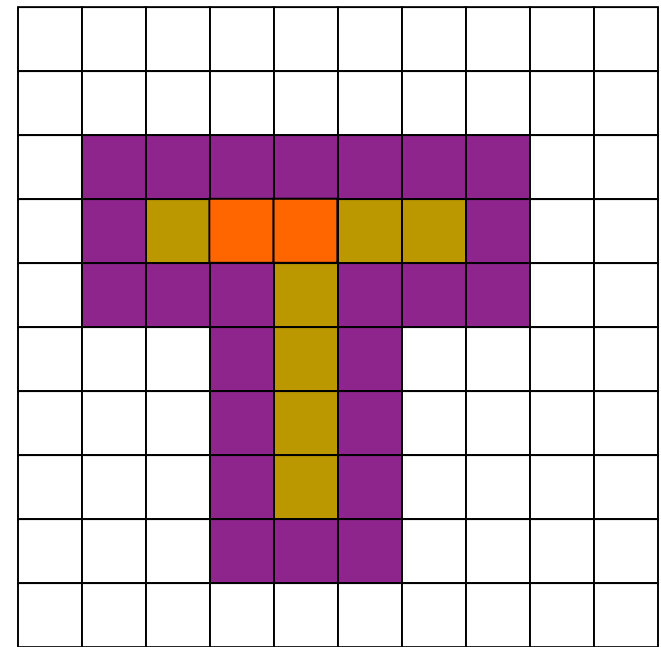
**A**



**B**



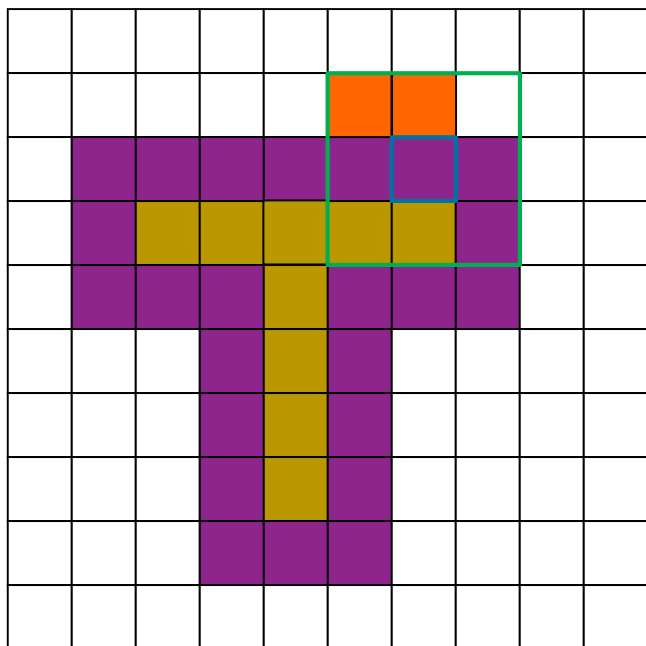
**A(+ )B**



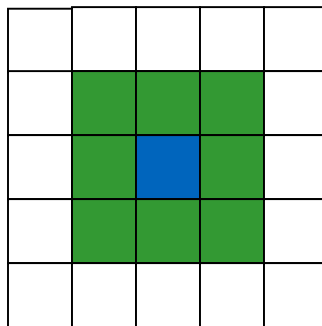
# Дискретные операции морфологии.

## Сужение

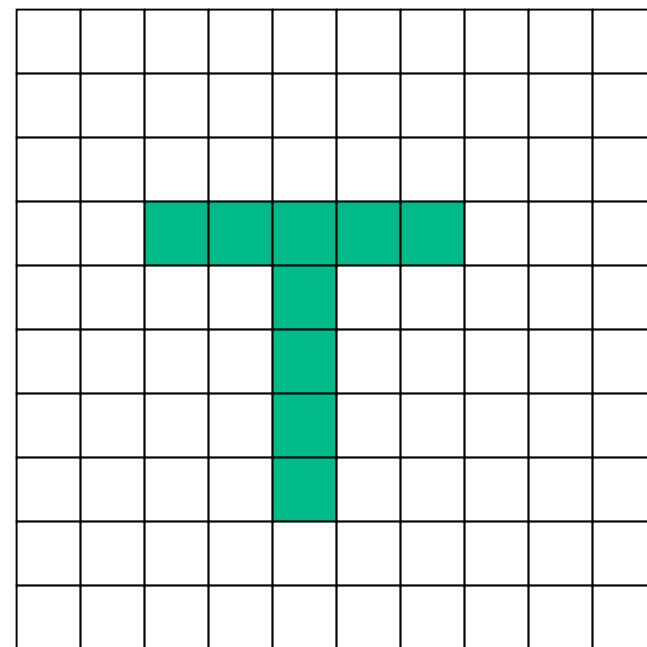
**A**



**B**



**A(-)B**





# Алгоритм морфологического расширения

```
void Dilation(BIT* source[ ], bool* mask[ ], BIT* result[ ])
{
    // Width, Height - размеры исходного и результирующего изображений
    // MW, MH - размеры структурного множества
    for(y = MH/2; y < Height - MH/2; y++)
        for(x = MW/2; x < Width - MW/2; x++)
        {
            BIT max = 0;
            for(j = -MH/2; j <= MH/2; j++)
                for(i = -MW/2; i <= MW/2; i++)
                    if((mask[i][j]) && (source[x + i][y + j] > max))
                    {
                        max = source[x + i][y + j];
                    }
            result[x][y] = max;
        }
}
```

**result[x][y] = max из пикселей, покрываемых маской**

# Алгоритм морфологического сужения

```
void Erosion(BIT* source[ ], bool* mask[ ], BIT* result[ ])
{
    // Width, Height - размеры исходного и результирующего изображений
    // MW, MH - размеры структурного множества
    for(y = MH/2; y < Height - MH/2; y++)
        for(x = MW/2; x < Width - MW/2; x++)
        {
            BIT min = MAXBIT;
            for(j = -MH/2; j <= MH/2; j++)
                for(i = -MW/2; i <= MW/2; i++)
                    if((mask[i][j]) && (source[x + i][y + j] < min))
                    {
                        min = source[x + i][y + j];
                    }
            result[x][y] = min;
        }
}
```

**result[x][y] = min из пикселей, покрываемых маской**

# Операции открытия и закрытия

---

Морфологическое открытие (opening)

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

Морфологическое закрытие (closing)

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

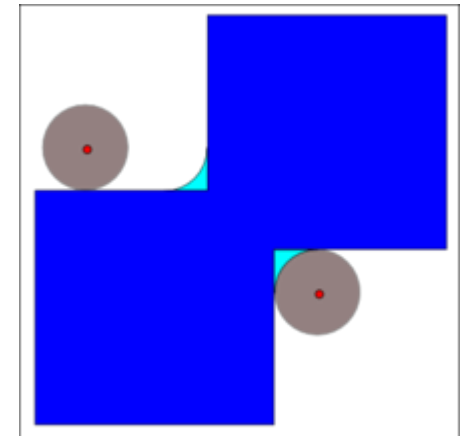
# Operations of opening and closing

## Morphological closing

► **close** (**A**, **B**) = (**A** (+) **B**) (-) **B** = **A** • **B**

### Closing Properties:

- Closing is **idempotent** (при повторном применении операции - тот же результат, что и при первом), that is,  $(A \bullet B) \bullet B = A \bullet B$ .
- Closing is **increasing**, that is, if  $A \subseteq C$ , then  $A \bullet B \subseteq C \bullet B$ .
- Closing is **extensive**, i.e.,  $A \subseteq A \bullet B$ .
- Closing is **translation invariant**.



The closing of the dark-blue shape by a disk, resulting in the union of the dark-blue shape and the light-blue areas (Wikipedia)

**Closing** удаляет небольшие дыры на переднем плане (шум, потерянные или желаемые к заполнению) объекты с переднего плана (обычно это темные объекты или отдельные пиксели) изображения, помещая их из фона на передний план.

# Operations of opening and closing

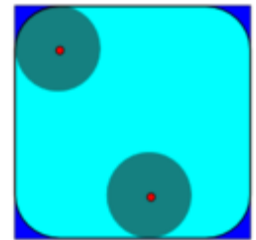
## Morphological opening

► **open**  $(A, B) = (A \ominus B) \oplus B$

### Opening Properties:

- Opening is **idempotent**, that is,  $(A \circ B) \circ B = A \circ B$ .
- Opening is **increasing**, that is, if  $A \subseteq C$ , then  $A \circ B \subseteq C \circ B$ .
- Opening is **anti-extensive**, i.e.,  $A \circ B \subseteq A$ .
- Opening is **translation invariant**.
- Opening and closing satisfy the duality  $A \bullet B = (A^c \circ B^c)^c$ , where  $\bullet$  denotes closing.

The opening of the dark-blue square by a disk, resulting in the light-blue square with round corners (Wikipedia)



**B** замечает внутреннюю часть границы **A**, не выходя за границу **A**.

**Opening** удаляет мелкие (шумные, лишние или нежелательные) объекты с переднего плана (обычно это яркие объекты или отдельные пиксели) изображения, помещая их на задний план.

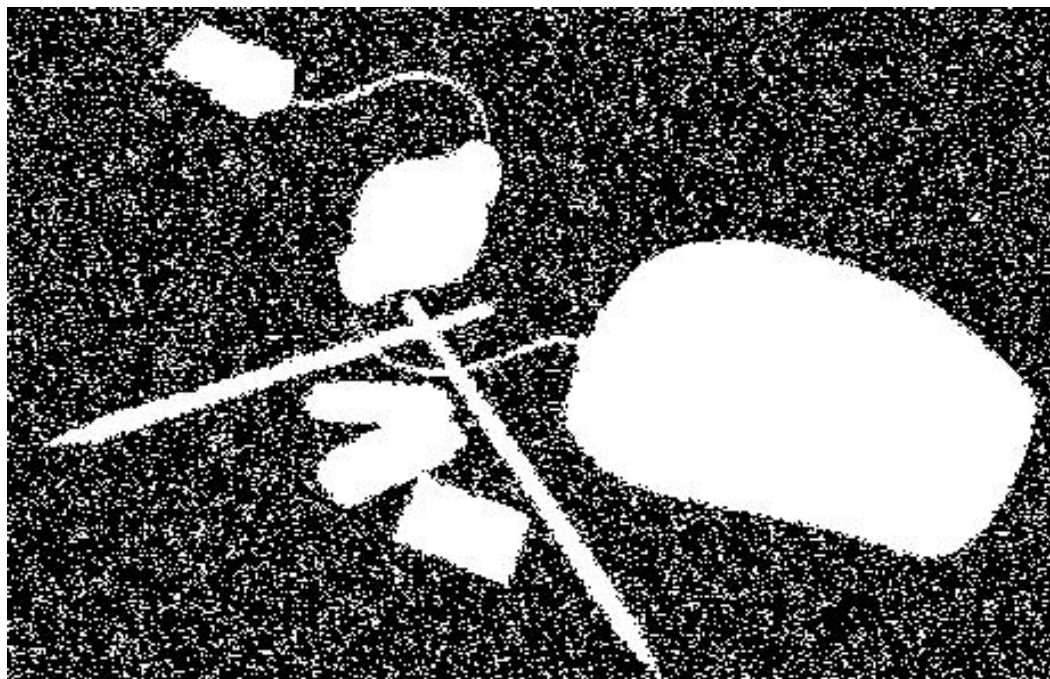
**Opening** можно использовать для поиска вещей, в которые может/не может поместиться конкретный структурный элемент (края, углы, ...).

See also: [https://en.wikipedia.org/wiki/Mathematical\\_morphology](https://en.wikipedia.org/wiki/Mathematical_morphology)

# Замечание

► Результат морфологических операций во многом определяется применяемым структурным элементом (множеством  $V$ ). Выбирая различный структурный элемент можно решать разные задачи обработки изображений:

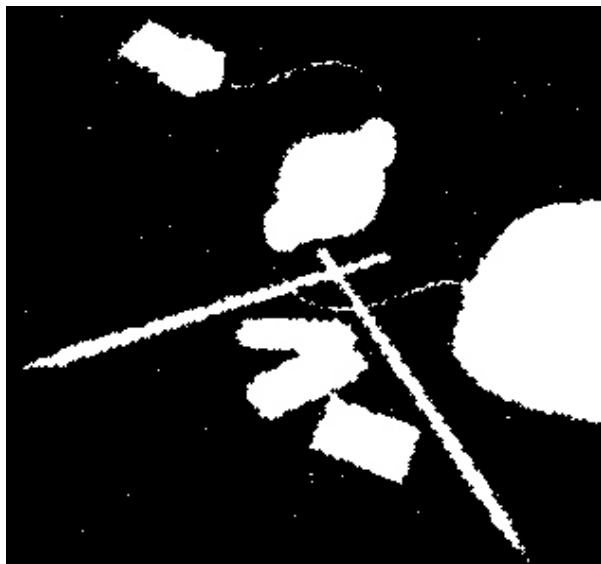
- Шумоподавление
- Выделение границ объекта
- Выделение скелета объекта
- Выделение дефектов регулярных структур (напр., сломанных зубьев на изображении шестерни)





# Применения сужения к бинарному изображению с сильным шумом

Сужение позволяет подавлять «белый» шум фона с потерями для площади объекта



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



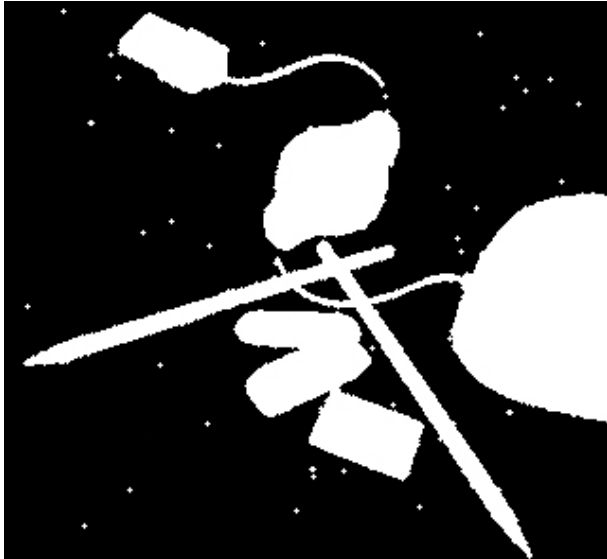
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & [1] & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



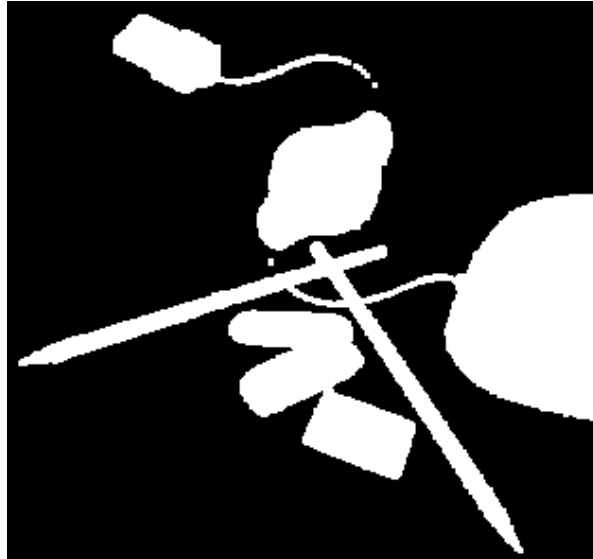
$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & [1] & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

# Применения открытия (A(-)B)(+)B к бинарному изображению с сильным шумом

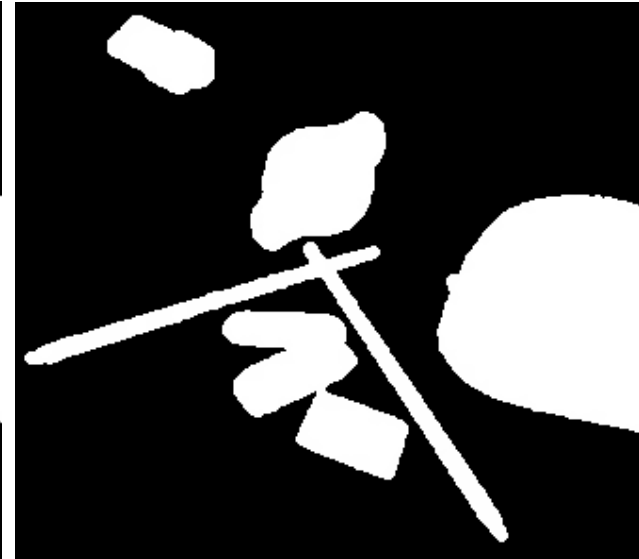
Открытие позволяет подавлять «белый» шум фона, если объекты без шума



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



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

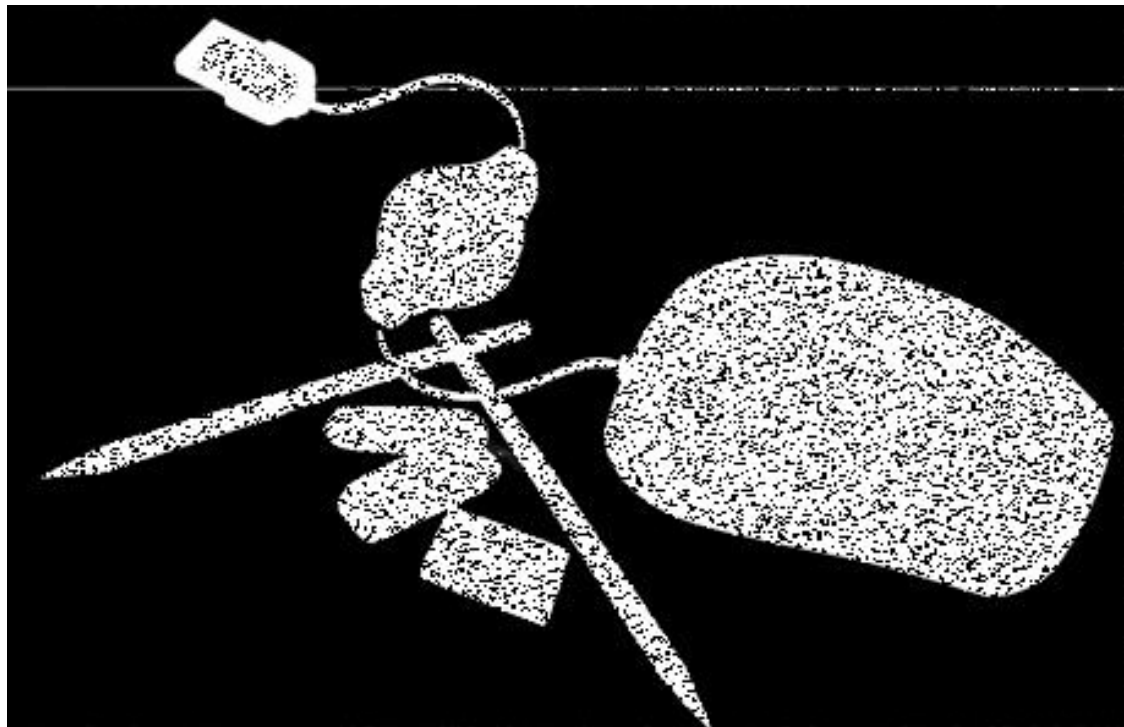


$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

# Шум в бинарных изображениях с дефектами объектов. Пример.

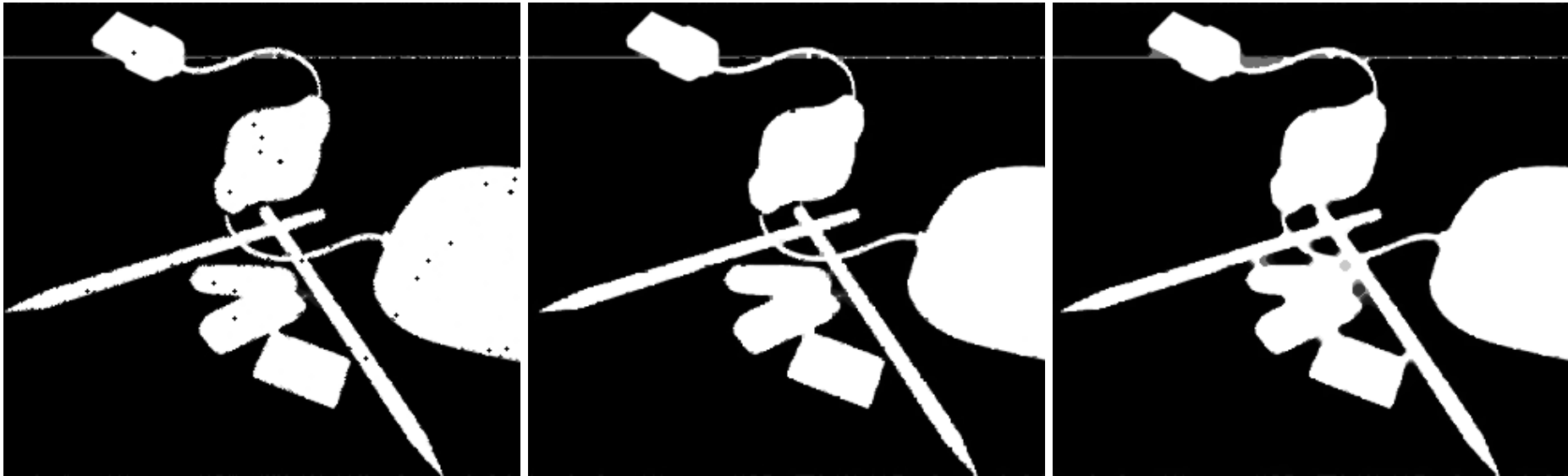
---

Пример бинарного изображения с дефектами распознаваемых объектов



# Применения закрытия (A(+))B(-)B к бинарному изображению с дефектами объектов

Закрытие позволяет подавлять «черный» шум на объектах, если фон без шума



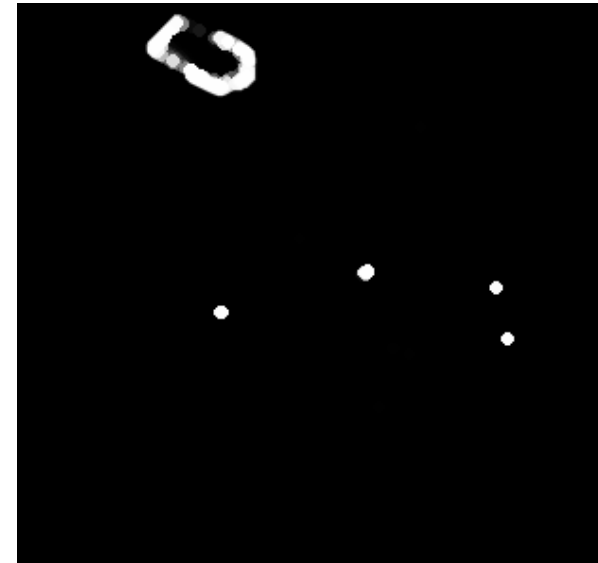
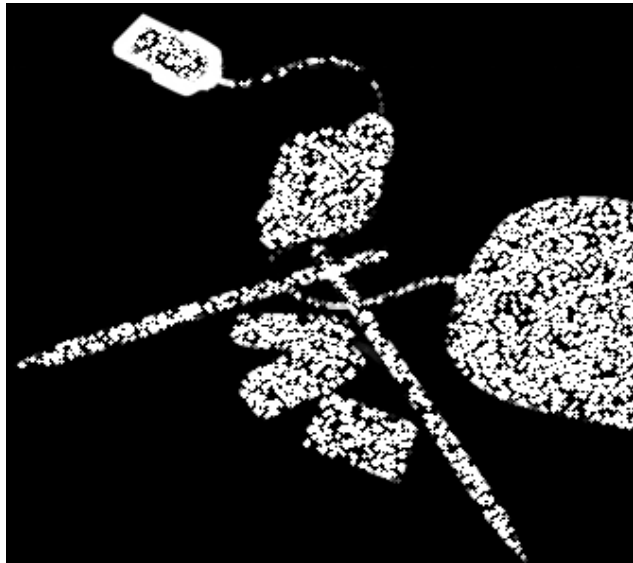
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

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

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

# Применение операции открытия (A(-)B)(+)B

Объекты полностью разрушены



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

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

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

# Свойства морфологических операций



образ



Erosion  $I(-)B$



Dilatation  $I \oplus B$

Структурный  
элемент

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



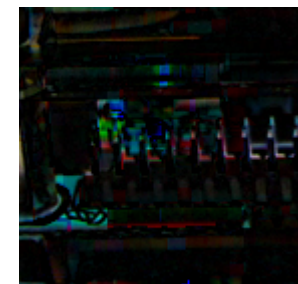
Opening  $I \circ B = (I(-)B)(+)B$     Closing  $I \bullet B = (I(+)B)(-)B$



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



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

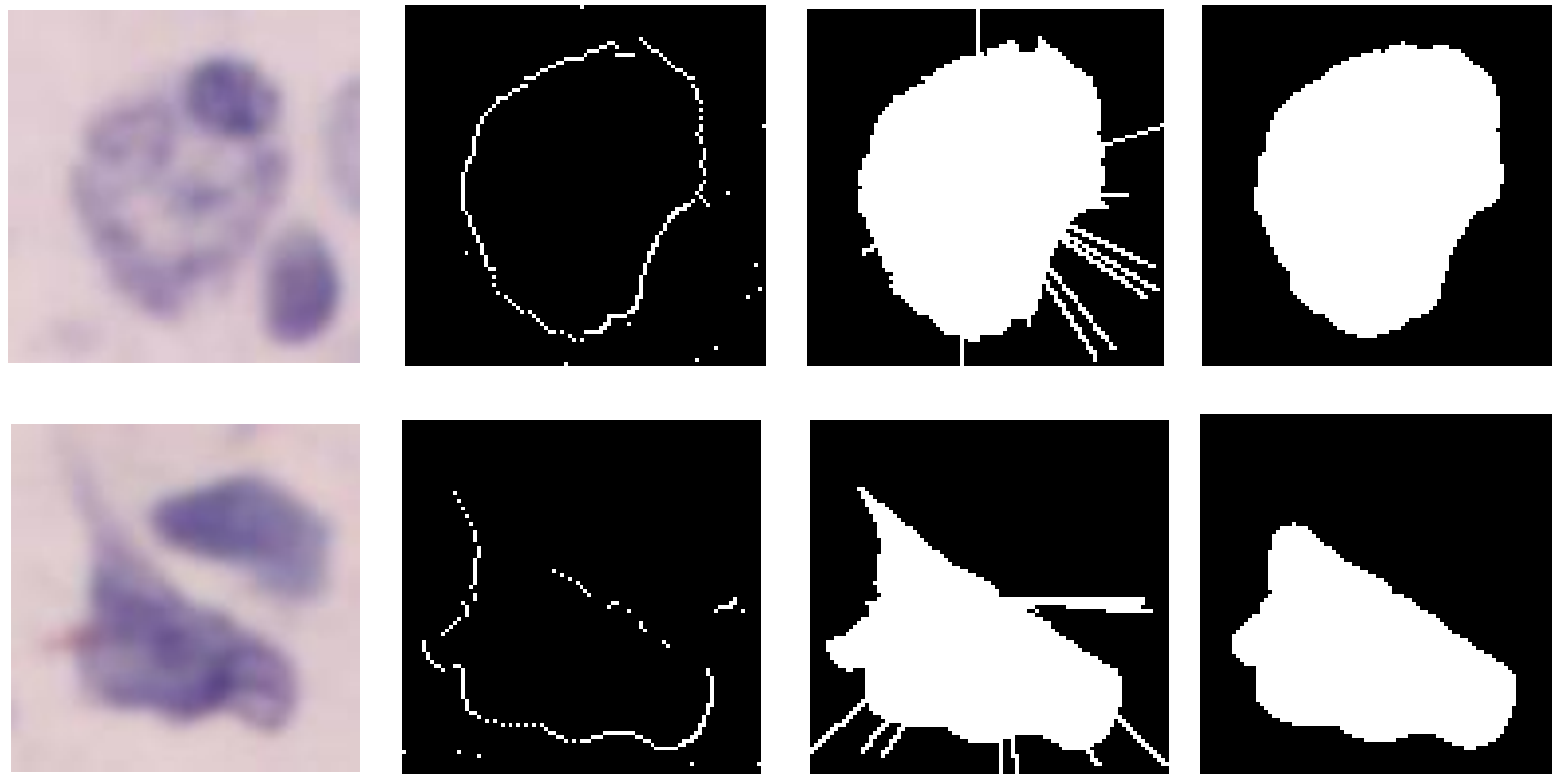


Black Hat (I) =  $(I \circ B) - I$



# Свойства операций открытия и закрытия

## Примеры применения Opening и Closing в регистрации клеток мозга

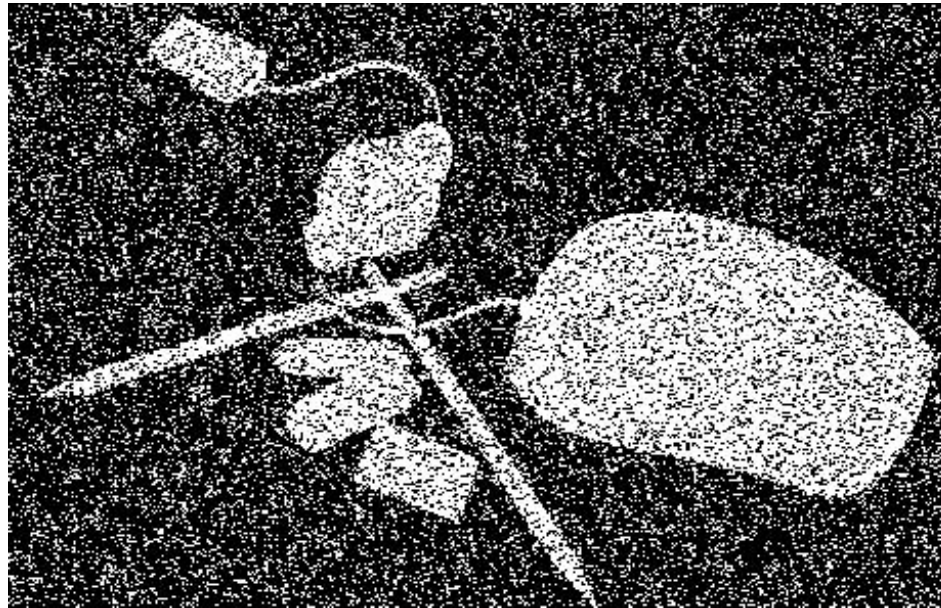


Примеры определения границы клеток (слева направо): исходное изображение клеток; найденные границы; регион, заполненный по найденным границам; заполненный регион после применения операции морфологического открытия

# Шум «соль и перец» - не лучший пример для морфологии

---

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



Шум «соль и перец»

# Источники

---

- ▶ Gary Bradski and Adrian Kaehler. Learning OpenCV/ Published by O'Reilly Media, Inc., 2008. - 577pp.
- ▶ Курсы и материалы лаборатории Graphics & Media Lab при ВМиК МГУ (<http://graphics.cs.msu.ru>)
- ▶ Ватолин Д.С. Сжатие изображений. :Изд. МГУ, 1999. -76с. ([local](#))
- ▶ 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>)
- ▶ Рекомендации для начинающих пользователей: Programming with intel IPP (integrated performance primitives) and intel OpenCV (open computer vision) under gnu linux: a beginner's tutorial.(j.Landre@iutlecreusot.u-bourgogne.fr)
- ▶ Intel® OPEN SOURCE COMPUTER VISION LIBRARY
- ▶ Image Analysis Cookbook 6.0. <http://www.reindeergraphics.com/foveaprotutorial.html>

## OpenCV

- ▶ \_ Open Computer Vision (OpenCV) sources, download site - <http://sourceforge.net/projects/opencvlibrary>
- ▶ \_ Open Computer Vision (OpenCV) mailing list and group - <http://groups.yahoo.com/group/OpenCV>