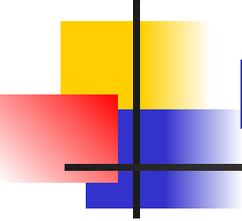


Обработка изображений

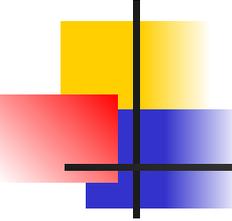
- Компьютерная графика
Computer graphics
- Компьютерное (машинное) зрение
Computer (machine) vision
- Обработка изображений
Image processing

Антон Конушин
ktosh@graphics.cs.msu.ru



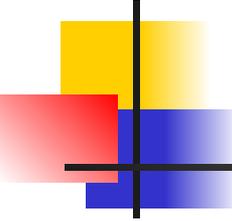
Цель лекции

- Рассказать о нескольких способах обработки изображений, которые могут пригодиться в «реальной жизни»
- Зачем обрабатывать?
 1. *Улучшение изображения для восприятия человеком*
 - цель – чтобы стало «лучше» с субъективной точки зрения человека
 2. *Улучшение изображения для восприятия компьютером*
 - цель – упрощение последующего распознавания
 3. *Развлечение (спецэффекты)*
 - цель – получить эстетическое удовольствие от красивого эффекта



План лекции

- Введение
- Коррекция контрастности/яркости изображения
- Коррекция цветового баланса изображения
- Подавление шума в изображениях
- Метрики качества
- Подчеркивание резких границ (краев) на изображении
- Спецэффекты



Изображение

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

Физический энциклопедический словарь.

Компьютерное представление изображения:

Функция интенсивности (яркости) канала

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

Используется дискретное представление

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



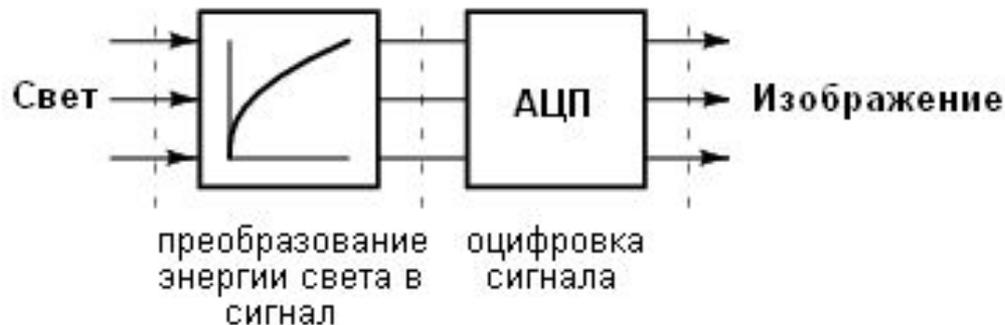
Обработка изображений

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

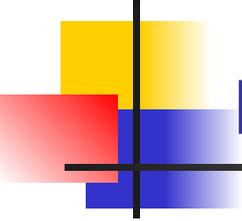
Как получается цифровое изображение?

- Свет, падая на светочувствительный элемент преобразуется в электрические сигналы
- Сигналы оцифровываются, превращаются в массив чисел



$$f(x) = y$$

x – характеристика яркости света
 y – яркость пиксела изображения



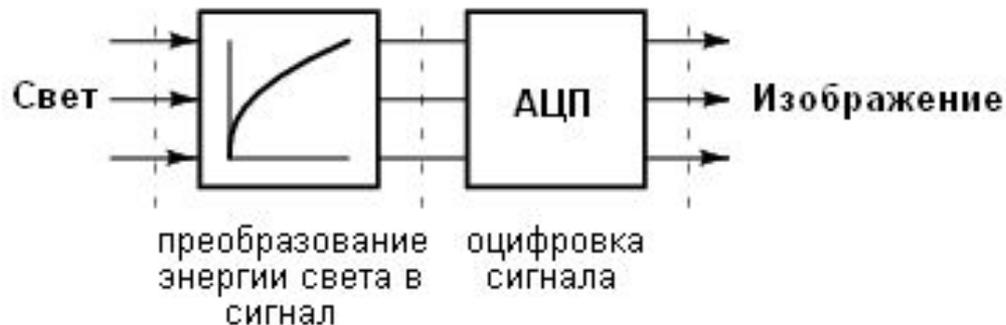
Почему оно может получиться плохо?

- Ограниченный диапазон чувствительности датчика
- «Плохой» функции передачи датчика



«Улучшение» изображения

- Изменение контраста изображения
 - Компенсация:
 - Ограниченного диапазона яркостей датчика
 - “Плохой” функции передачи датчика

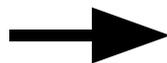


$$f(x) = y$$

x – характеристика яркости света
 y – яркость пиксела изображения

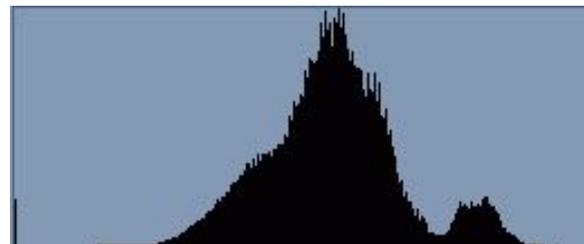
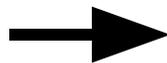
Что такое гистограмма?

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



0

255



0

255



Изменение контраста изображения

Что может не устраивать в полученном изображении:

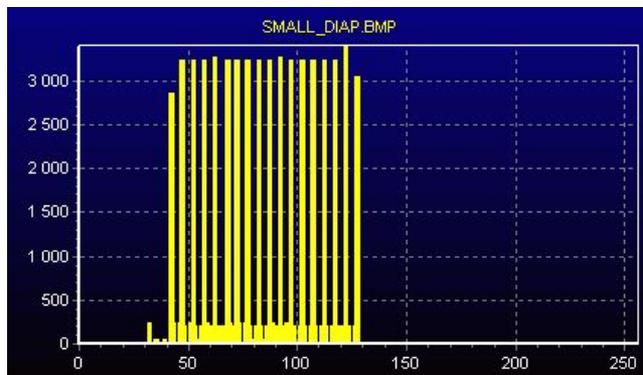
- Узкий или смещенный диапазон яркостей пикселей (тусклое или «пересвеченное» изображение)
- Концентрация яркостей вокруг определенных значений, неравномерное заполнение диапазона яркостей (узкий диапазон - тусклое изображение)

Коррекция - к каждому пикселю применяется преобразование яркостей, компенсирующий нежелательный эффект:

y – яркость пиксела на исходном изображении,
 $f^{-1}(y) = x$ – яркость пиксела после коррекции.

Линейная коррекция

Компенсация узкого диапазона яркостей – линейное растяжение:



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

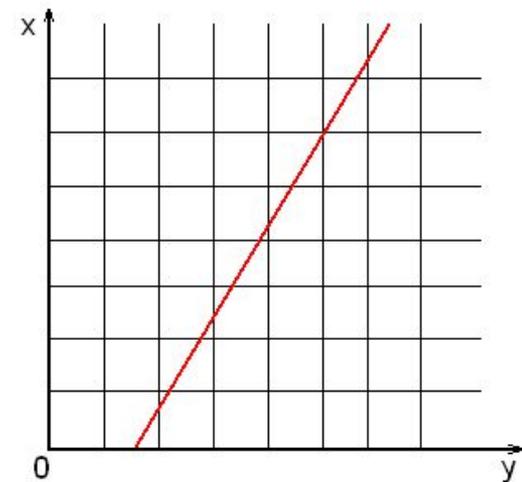
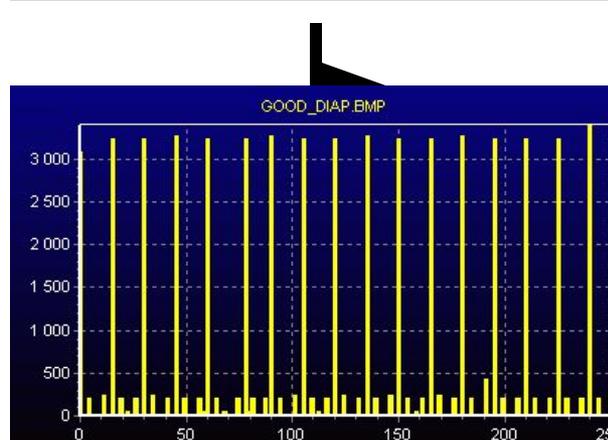
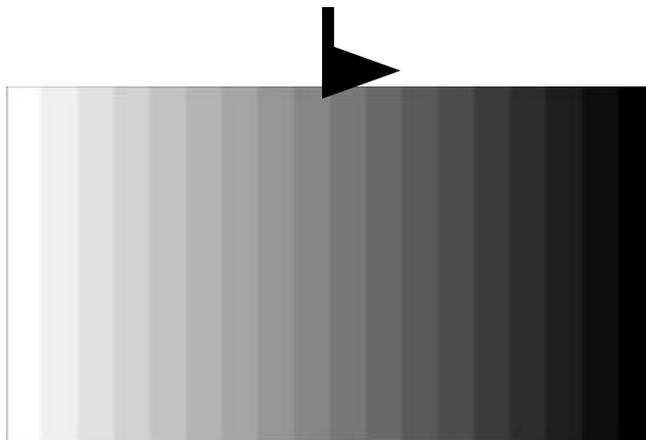
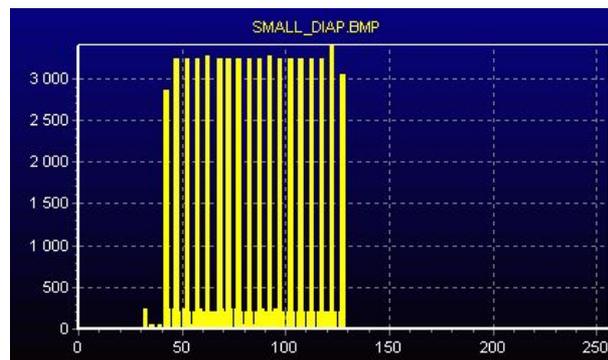
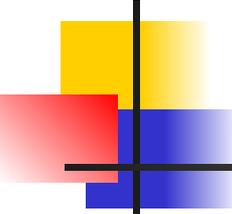


График функции $f^{-1}(y)$

Линейная коррекция

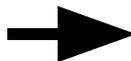
Компенсация узкого диапазона яркостей – линейное растяжение:

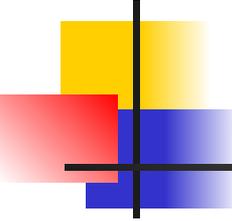




Линейная коррекция

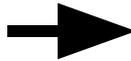
Линейное растяжение – «как AutoContrast в Photoshop»





Линейная коррекция

Линейная коррекция помогает не всегда!



Нелинейная коррекция

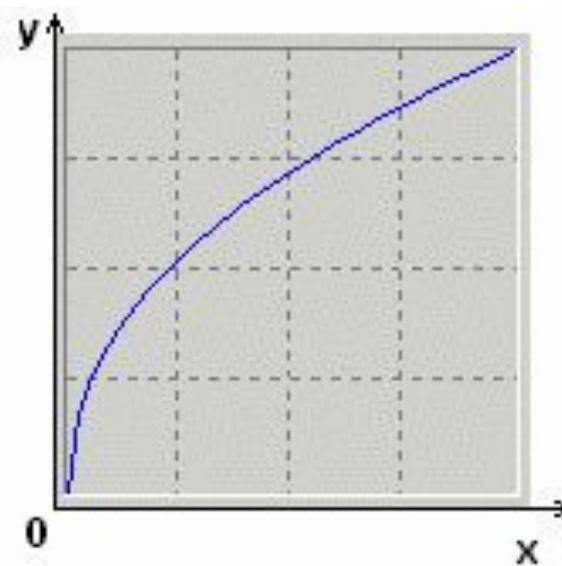
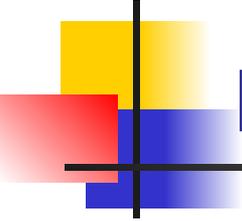


График функции $f^{-1}(y)$



Нелинейная коррекция

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

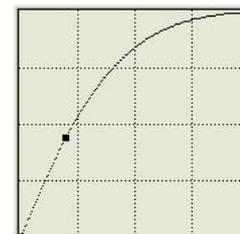
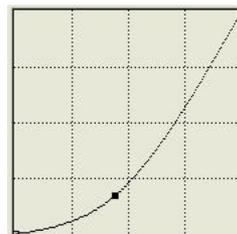
$$y = c \cdot \log(1 + x)$$

Гамма-коррекция

Гамма-коррекция

- Изначальная цель – коррекция для правильного отображения на мониторе. Так называют преобразование вида:

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



Графики функции $f^{-1}(y)$

Нелинейная коррекция

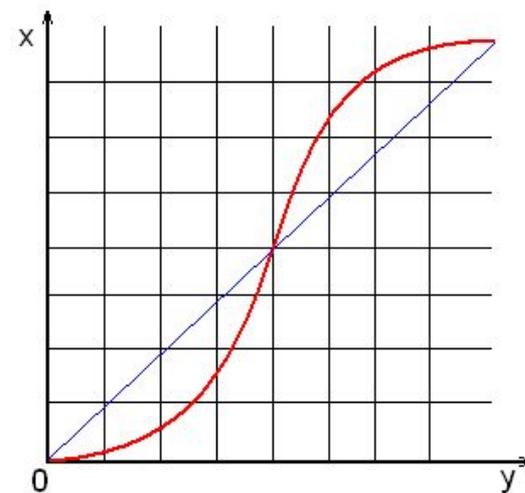
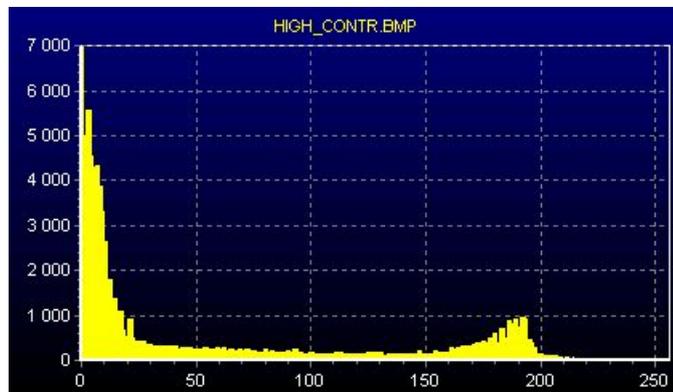
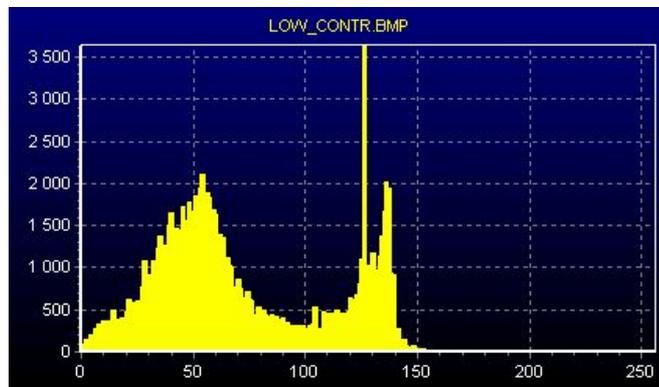
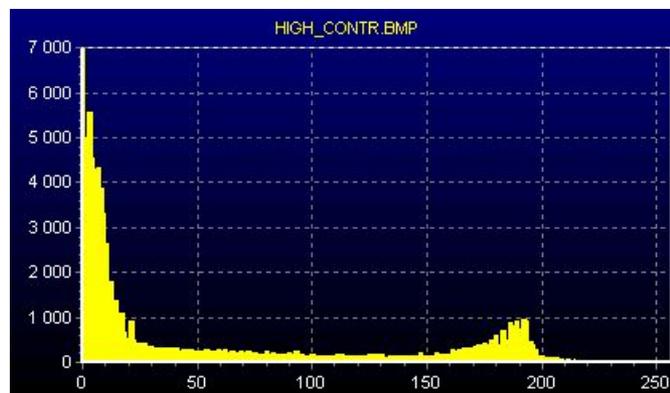
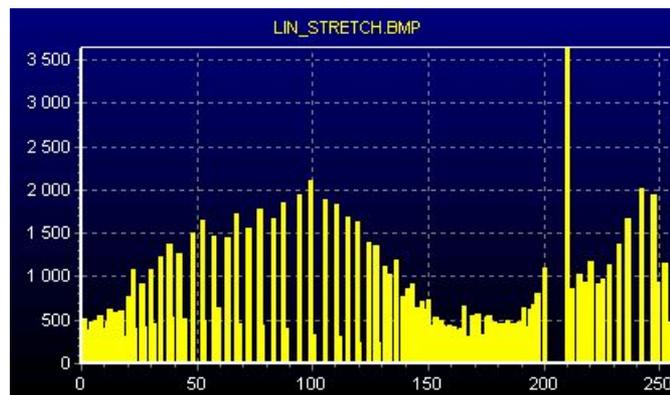


График функции $f^{-1}(y)$

Сравнение линейной и нелинейной коррекции





Компенсация разности освещения

Пример

Sonnet for Lena

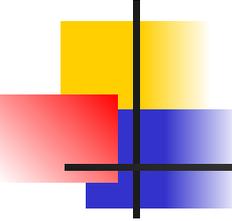
O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst



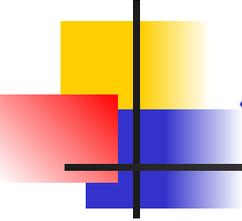
План лекции

- ✓ Введение
- ✓ Коррекция контрастности/яркости изображения
 - Коррекция цветового баланса изображения
 - Подавление шума в изображениях
 - Метрики качества
 - Подчеркивание резких границ (краев) на изображении
 - Спецэффекты

Цветовая коррекция изображений

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





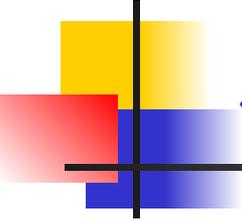
«Серый мир»

- 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:

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



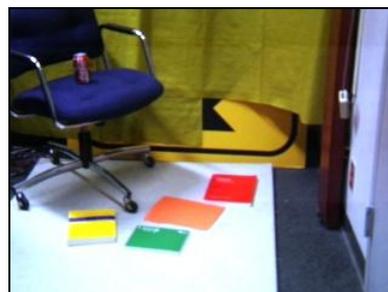
«Серый мир» - примеры

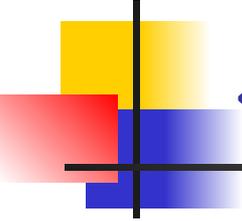


«Серый мир» - примеры



«Серый мир» - примеры





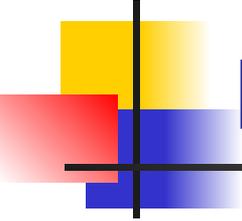
«Идеальный отражатель»

- 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:

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

- Преобразовать интенсивности:

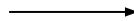
$$(R - R_{\min}) * \frac{(255 - 0)}{(R_{\max} - R_{\min})}; \quad (G - G_{\min}) * \frac{(255 - 0)}{(G_{\max} - G_{\min})};$$

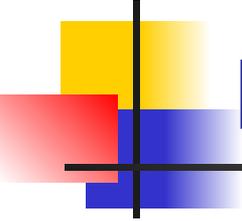
$$(B - B_{\min}) * \frac{(255 - 0)}{(B_{\max} - B_{\min})};$$

Растяжение контрастности всех каналов (“autolevels”)



Растяжение контрастности (“autolevels”)





Коррекция с опорным цветом

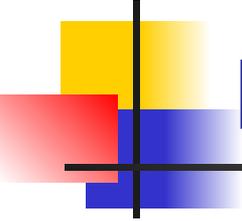
- Предположение
 - Пользователь указывает цвет вручную;
- Источник:
 - Априорные знания – «облака – белые»
 - Хорошая фотография этой же сцены
- Метод
 - Преобразовать по каждому из каналов цвета по формуле:

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

Коррекция с опорным цветом

- Примеры:





План лекции

- ✓ Введение
- ✓ Коррекция контрастности/яркости изображения
- ✓ Коррекция цветового баланса изображения
 - Подавление шума в изображениях
 - Метрики качества
 - Подчеркивание резких границ (краев) на изображении

Борьба с шумом изображения

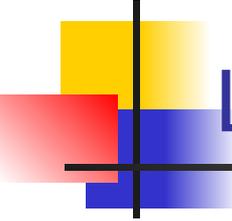
- Подавление и устранение шума
 - Причины возникновения шума:
 - Несовершенство измерительных приборов
 - Хранение и передача изображений с потерей данных



Шум фотоаппарата

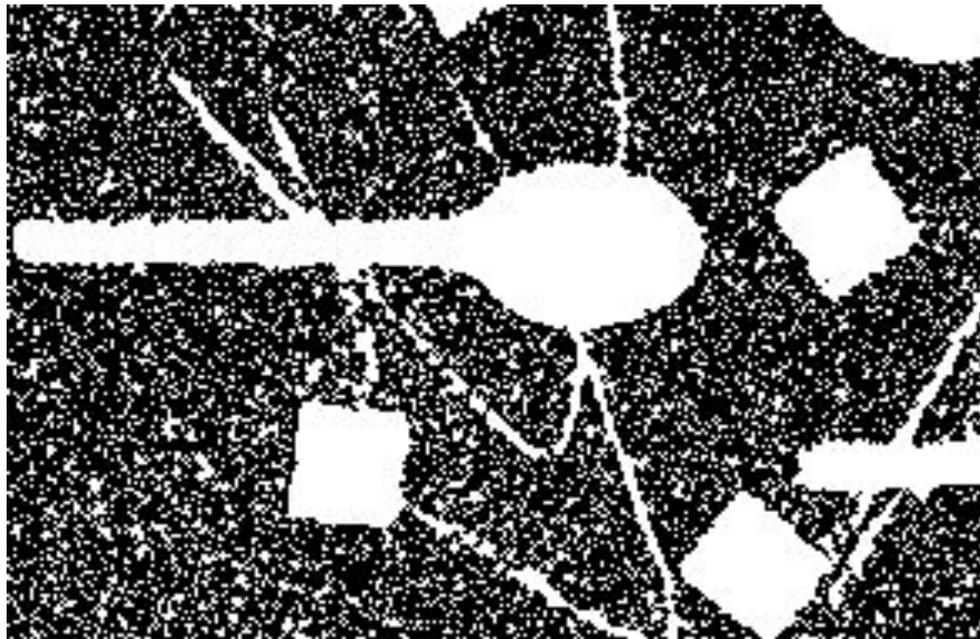


Сильное сжатие JPEG



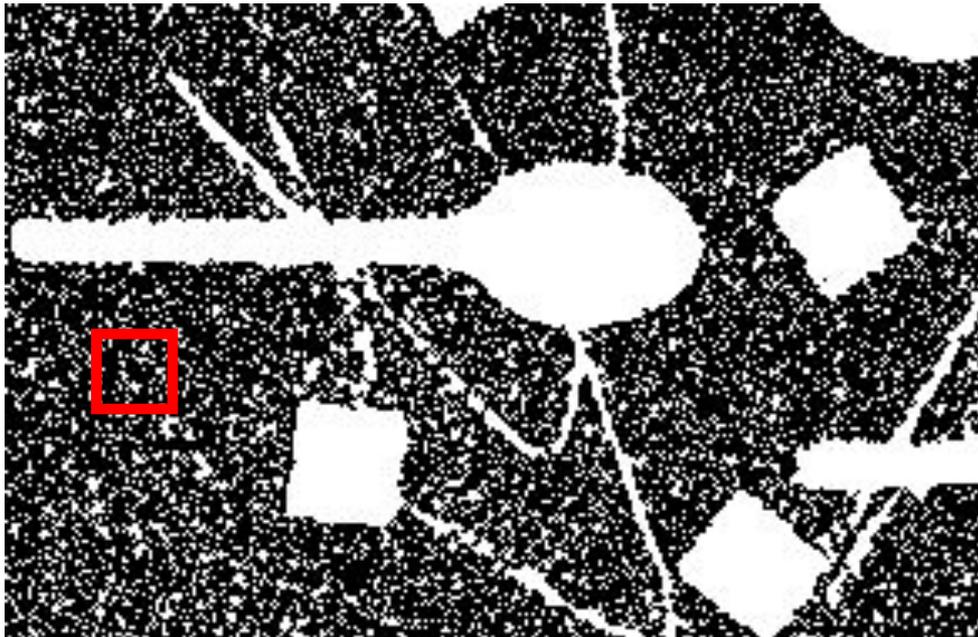
Шум в бинарных изображениях

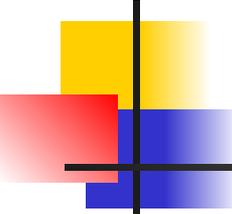
Пример бинарного изображению с сильным шумом



Шум в бинарных изображениях

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





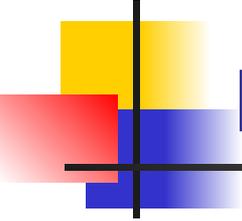
Подавление и устранение шума

Устранение шума в бинарных изображениях

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

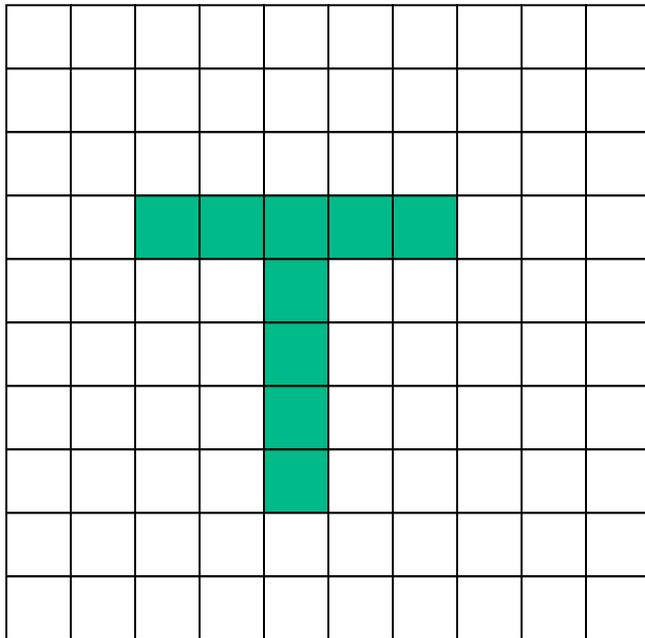
Широко известный способ - устранение шума с помощью операций математической морфологии:

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

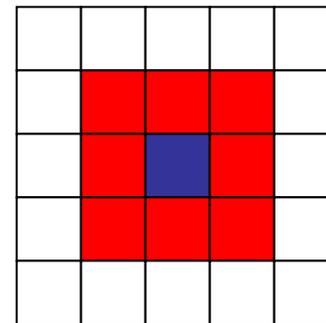


Математическая морфология

A

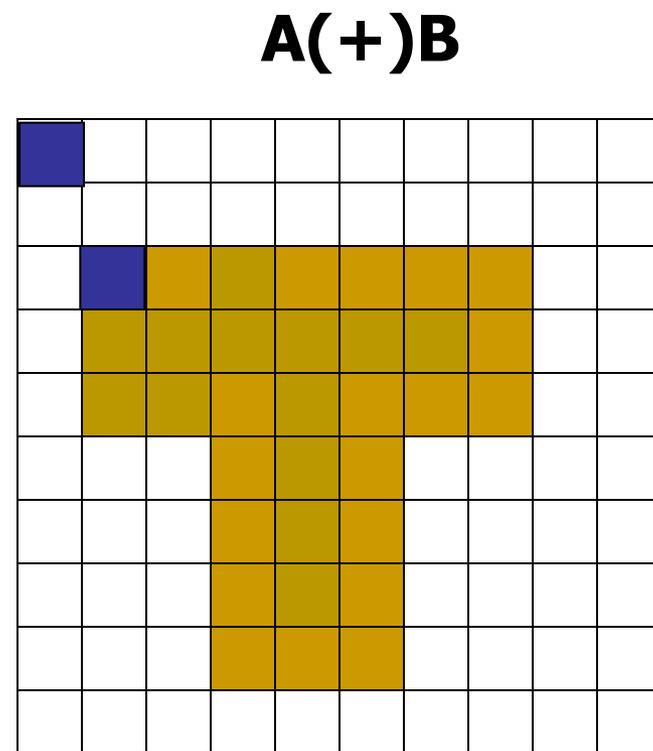
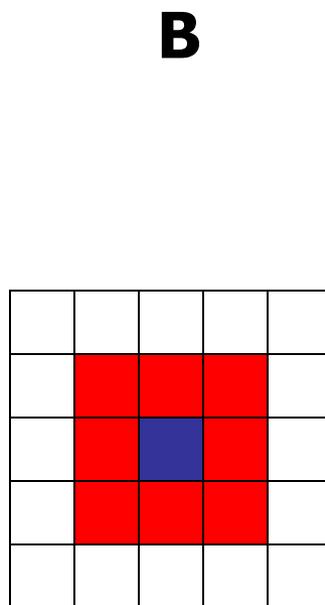
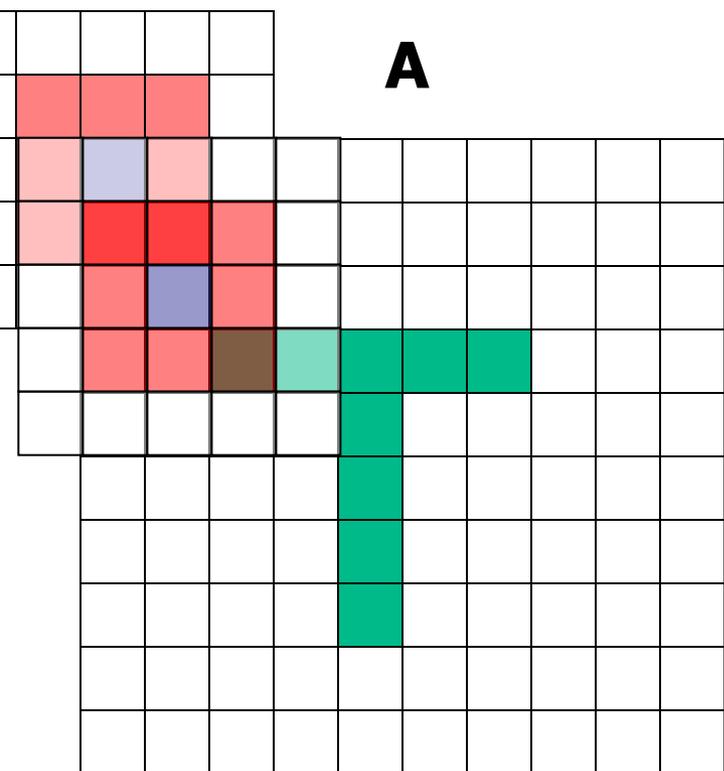


B



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

Расширение в дискретном случае

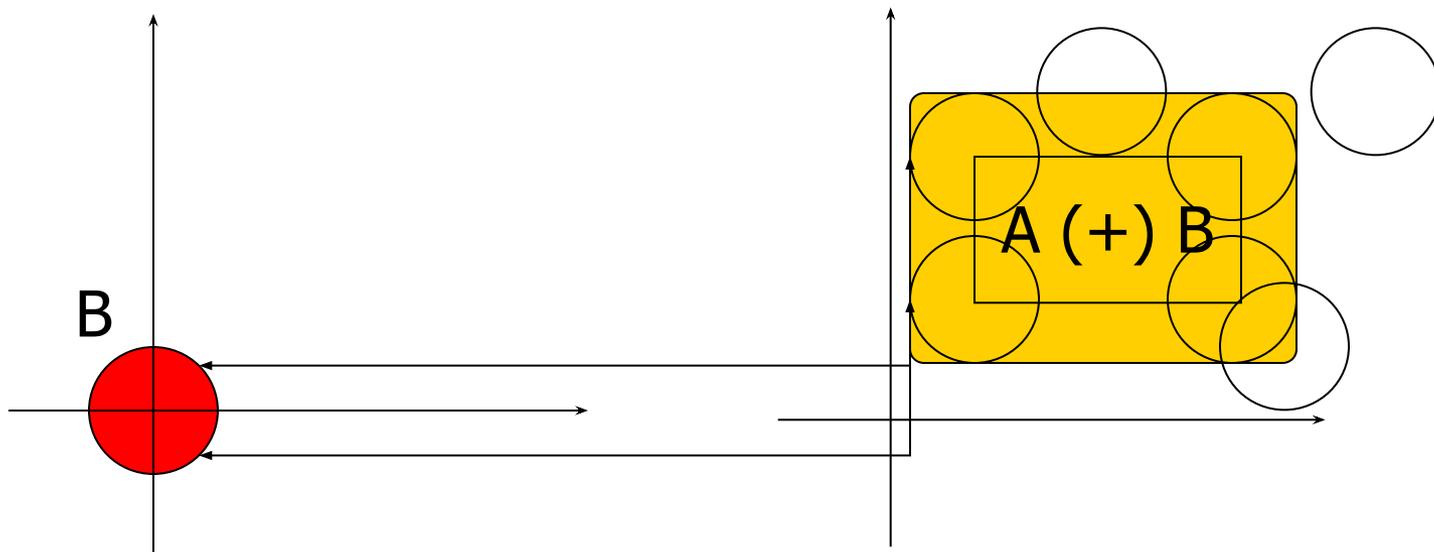


Операция «расширение» - аналог логического «или»

Операции математической морфологии

Расширение

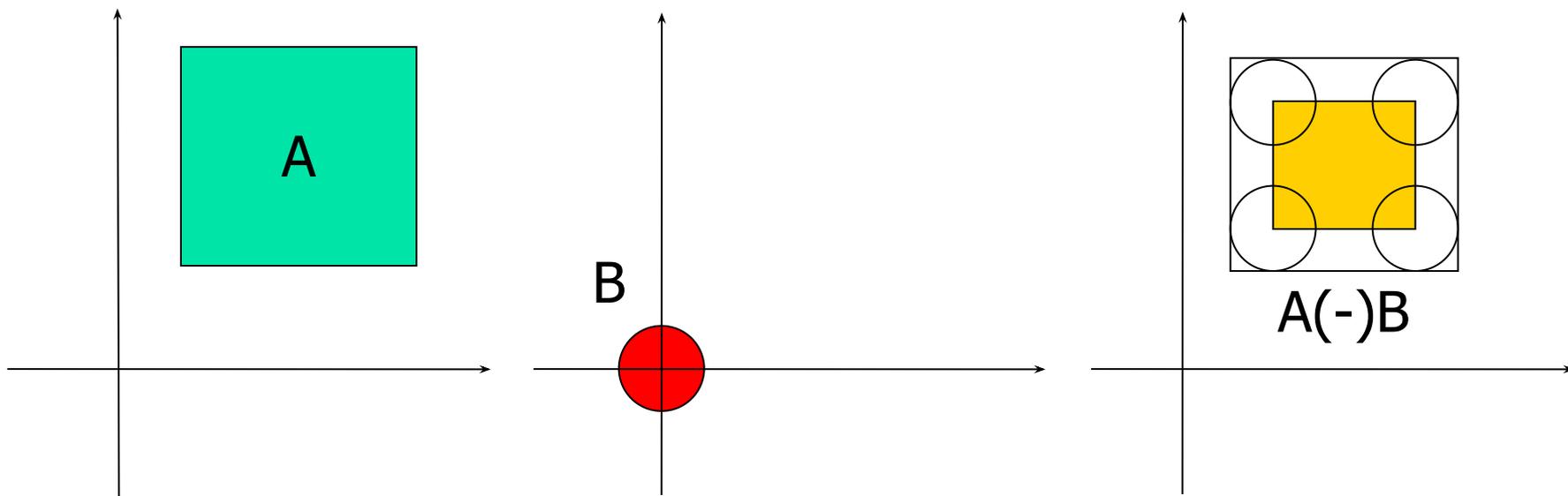
$$A (+) B = \{t \in \mathbb{R}^2: t = a + b, a \in A, b \in B\}$$

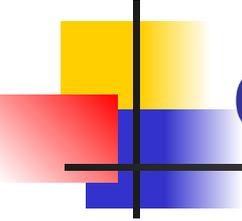


Операции математической морфологии

Сужение

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





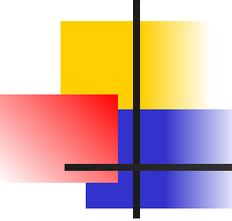
Свойства морфологических операций

Коммутативный закон

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

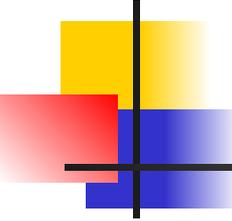
Ассоциативный закон

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



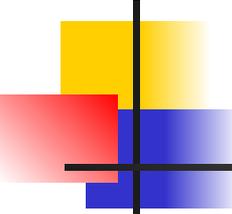
Алгоритм морфологического расширения

```
void Dilation(BIT* src[], bool* mask[], BIT* dst[])
{
    // W, H – размеры исходного и результирующего изображений
    // MW, MH – размеры структурного множества
    for(y = MH/2; y < H – MH/2; y++)
    {
        for(x = MW/2; x < W – 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]) && (src[x + i][y + j] > max))
                    {
                        max = src[x + i][y + j];
                    }
            }
            dst[x][y] = max;
        }
    }
}
```



Алгоритм морфологического сужения

```
void Erosion(BIT* src[], bool* mask[], BIT* dst[])
{
    // W, H – размеры исходного и результирующего изображений
    // MW, MH – размеры структурного множества
    for(y = MH/2; y < H – MH/2; y++)
    {
        for(x = MW/2; x < W – 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]) && (src[x + i][y + j] < min))
                    {
                        min = src[x + i][y + j];
                    }
            }
            dst[x][y] = min;
        }
    }
}
```



Операции раскрытия и закрытия

Морфологическое раскрытие (opening)

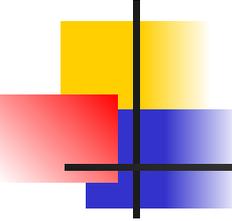
- **$\text{open}(A, B) = (A (-) B) (+) B$**

Морфологическое закрытие (closing)

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

Образовательные материалы по мат. морфологии
доступны по адресу:

<http://courses.graphicon.ru/main/cg/library>



Важное замечание

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

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

Применения сужения к бинарному изображению с сильным шумом



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

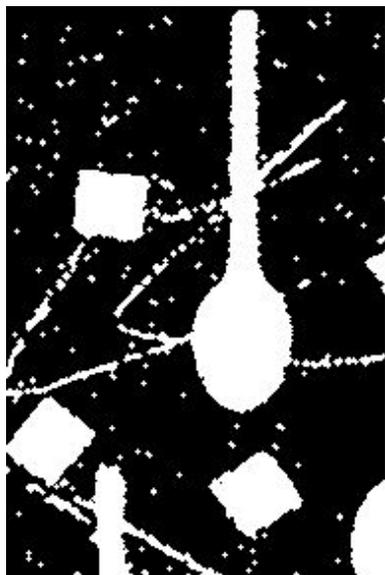


$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & [1] & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



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

Применения открытия к бинарному изображению с сильным шумом



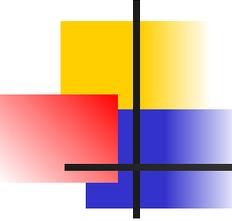
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



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



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



Сужение vs Открытие



Сужение



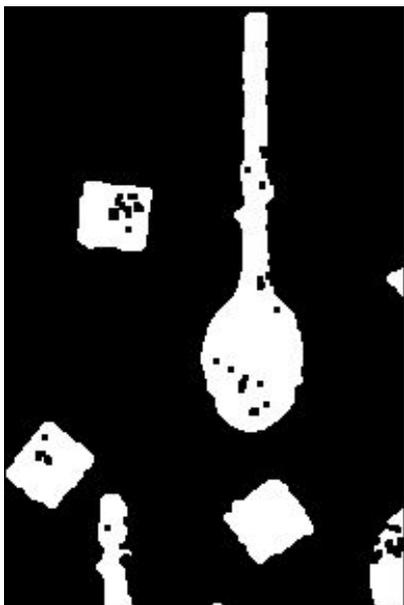
Открытие

Устранение шума в бинарных изображениях

Пример бинарного изображению с дефектами
распознаваемых объектов



Применения закрытия к бинарному изображению с дефектами объектов



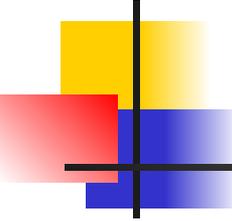
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



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



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

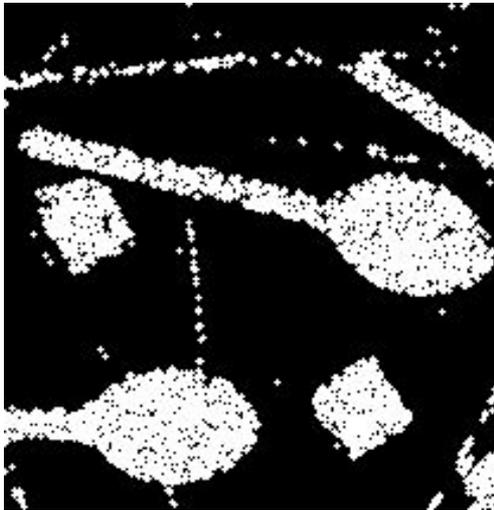


Не лучший пример для морфологии

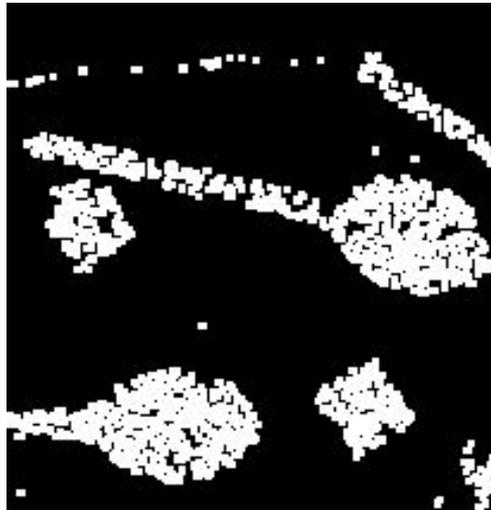
Не во всех случаях математическая морфология так легко убирает дефекты, как хотелось бы...



Применения операции открытия



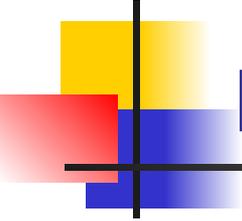
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



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

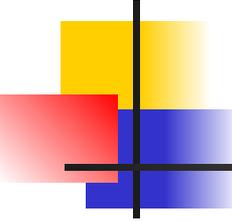


$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$



Подавление и устранение шума

- Устранение шума в полутоновых и цветных изображениях
 - Усреднение (box filter)
 - Фильтр Гаусса (gaussian blurring)
 - Медианный фильтр
 - Адаптивные фильтры



Операция «свертка» (convolution)

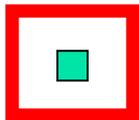
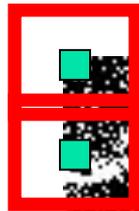
Свертка двумерной функции f по функции g в непрерывном и дискретном случае.

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

$g(l, k)$ – ядро (*kernel*) свертки или фильтра размером $(n_1-n_0) \times (m_1-m_0)$

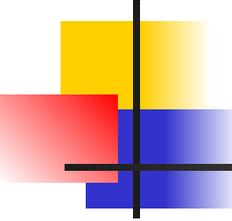
Часто, свертка изображения по какой-либо функции называется применением фильтра к изображению.

Применение фильтров



$g(l,k)$





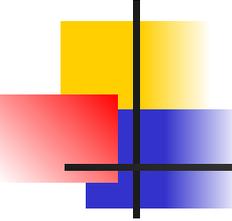
Свертка

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

```
// Обнулить изображение Dest[i][j]
...
// Выполнить свертку
for (i=0; i<Height; i++)           // Для каждого пикс. Dest[i][j]...
  for (j=0; j<Width; j++)
    for (l=-1; l<=1; l++)          // ...превратить его в ядро свертки
      for (k=-1; k<=1; k++)
        Dest[i+l][j+k] += Src[i][j] * Ker[l][k];    // и сложить
```

Подводные камни:

- Выход за границы массива
- Выход за пределы допустимого диапазона яркости пикселей
- Обработка краев.



Свойства фильтров

1. Результат фильтрации однотонного (константного) изображения – константное изображение. Его цвет равен

$$Dest = Src \cdot \sum_{k,p} Ker[k, p]$$

2. Следствие: чтобы фильтр сохранял цвет однотонных областей, нужно чтобы

$$\sum_{k,p} Ker[k, p] = 1$$

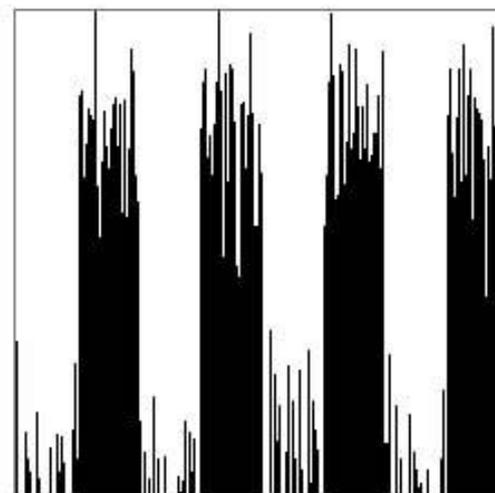
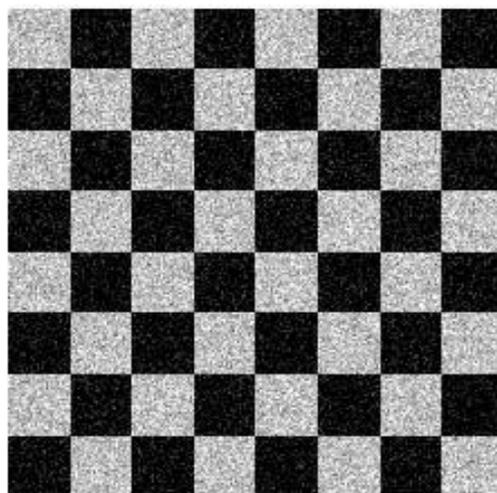
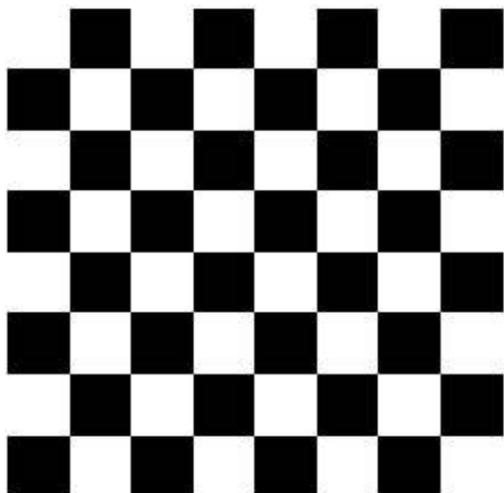
3. Следствие: если сумма коэффициентов фильтра равна нулю, то он переводит однотонные области в нулевые.

Устранение шума в полутоновых и цветных изображениях

Пример: изображение с равномерным шумом.

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

$Err(i, j)$ – нормально распределенная случайная величина.

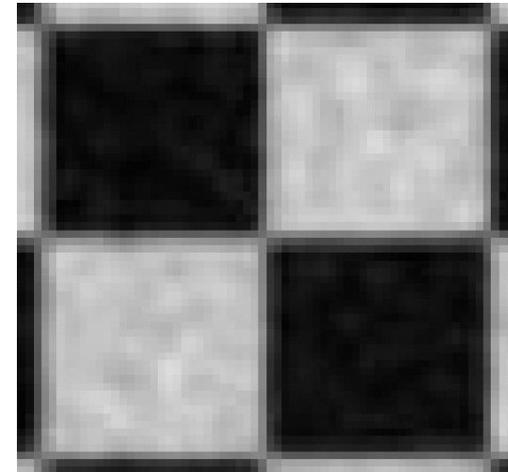
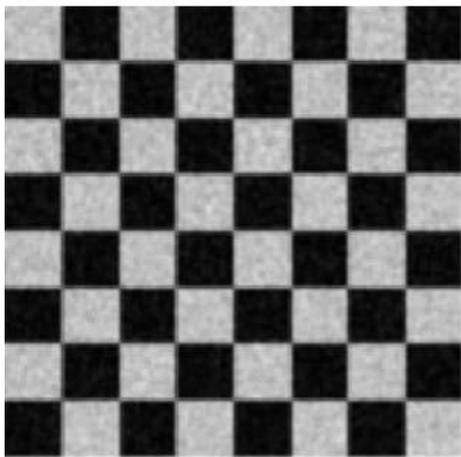


Усреднение (box filter)

Операция усреднения значения каждого пикселя – свертка по константной функции:

$$I'(i, j) = \sum_{l=-n}^n \sum_{k=-m}^m I(i+l)(j+k) \cdot \frac{1}{4nm}$$

Результат применения:



Фильтр Гаусса (gaussian blurring)

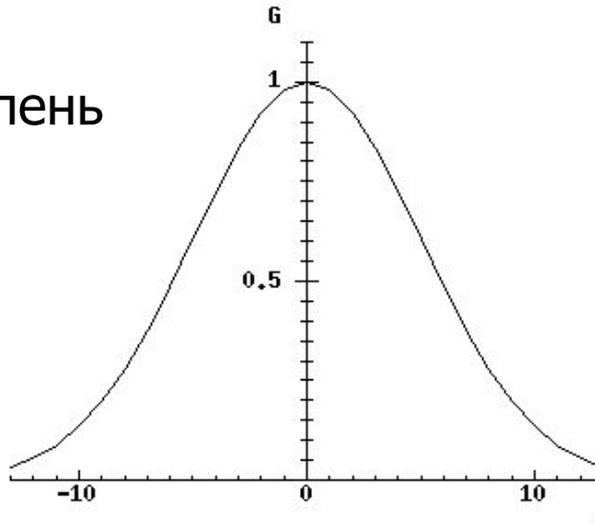
Свертка по функции:

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

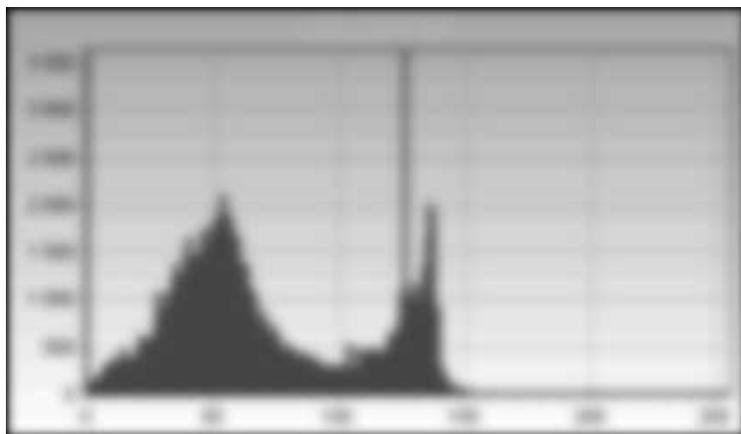
$$d = \sqrt{l^2 + k^2}$$

Параметр σ задает степень размытия.

На графике функция с $\sigma = 5$.

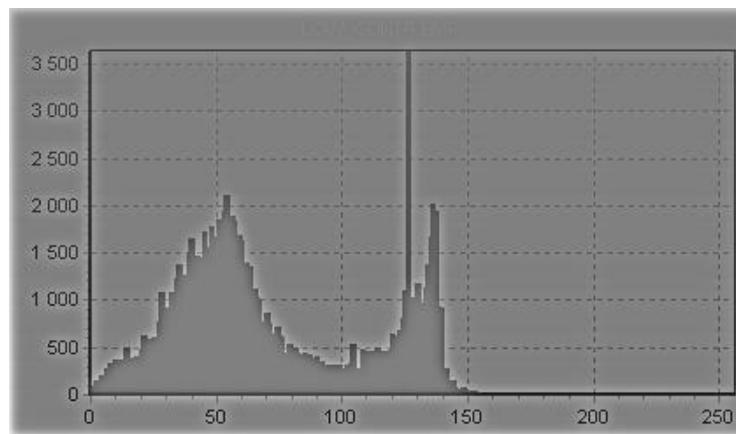


Маленькая экскурсия к Фурье

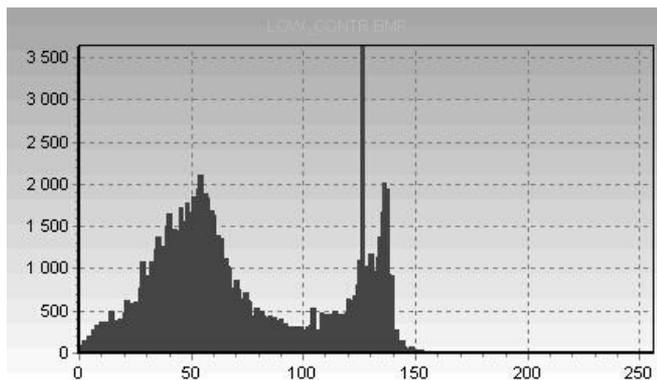


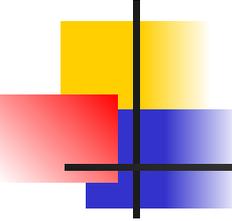
Низкие частоты

+



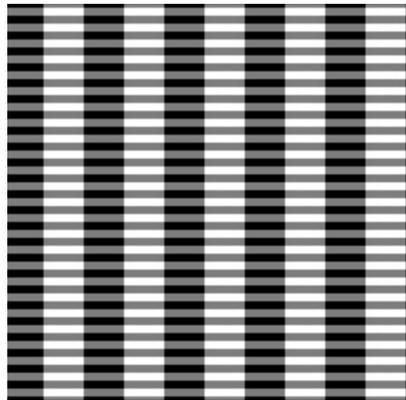
Высокие частоты



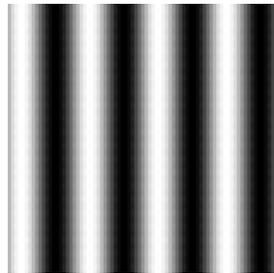


Фильтр Гаусса (gaussian blurring)

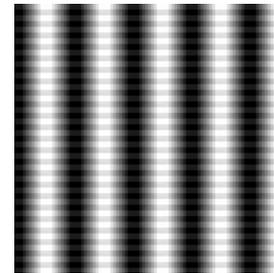
Результаты свертки по функции Гаусса и по константной функции (усреднения).



Исходное изображение

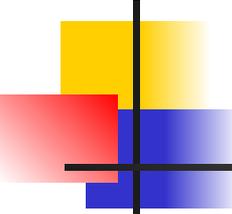


Фильтр Гаусса с
Sigma = 4



Усреднение по 49
пикселям (7x7)

Важное свойство фильтра Гаусса – он по сути является низкочастотным фильтром.



Подавление и устранение шума

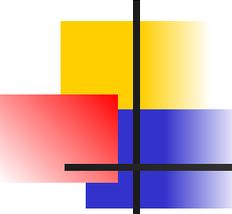
Устранение шума в полутоновых, цветных и бинарных изображениях с помощью медианного фильтра - выбор медианы среди значений яркости пикселей в некоторой окрестности.

Определение медианы:

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

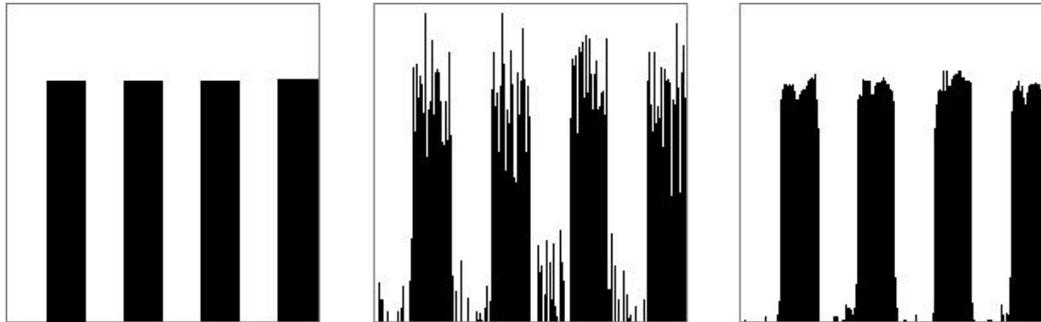
$A_{[n/2]}$ – медиана набора.

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

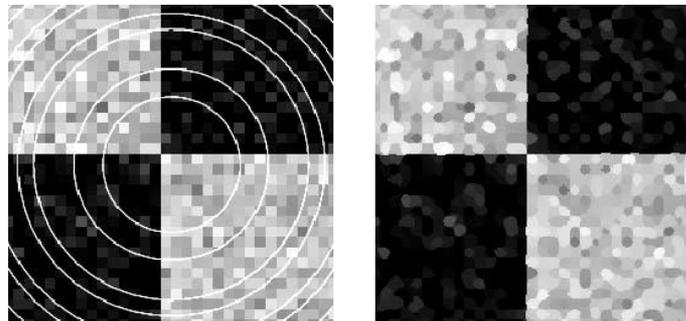


Медианный фильтр

Результат применения медианного фильтра с радиусом 5 пикселей.

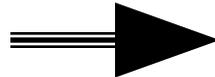
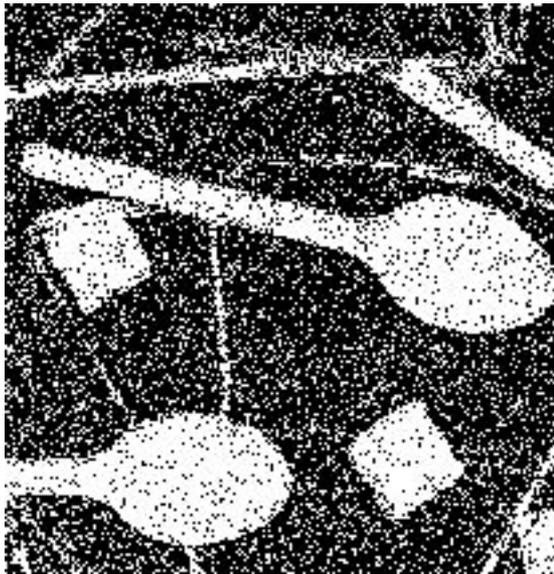


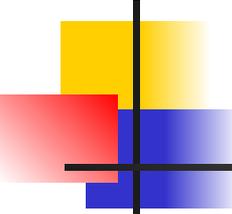
Результат применения медианного фильтра с радиусом в 7 пикселей к изображению с шумом и артефактами в виде тонких светлых окружностей.



Очистка изображения с помощью медианного фильтра

Фильтр с окрестностью 3x3



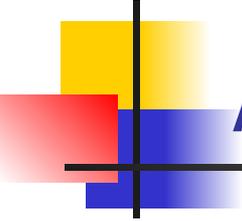


Быстрая реализация медианного фильтра

Медианный фильтр считается дольше, чем операция свертки, поскольку требует частичной сортировки массива яркостей окрестных пикселей.

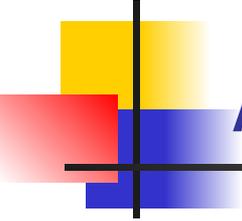
Возможности ускорения:

- Использовать алгоритмы быстрой сортировки
- Конкретная реализация для каждого радиуса (3x3, 5x5)
- Не использовать сортировку вообще – считать через гистограмму окрестности точки



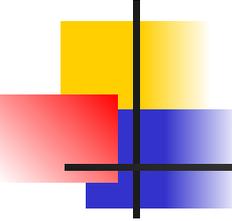
Адаптивные фильтры

- Чего бы хотелось?
 - Размывать шум, резкие границы – сохранять.
- Как бы этого добиться?
 - *Предположение:* перепады яркости из-за шума относительно перепадов на резких границах невелики
 - *Алгоритм:* При расчете новой яркости усреднять только по тем пикселям из окрестности, которые не сильно отличаются по яркости от обрабатываемого



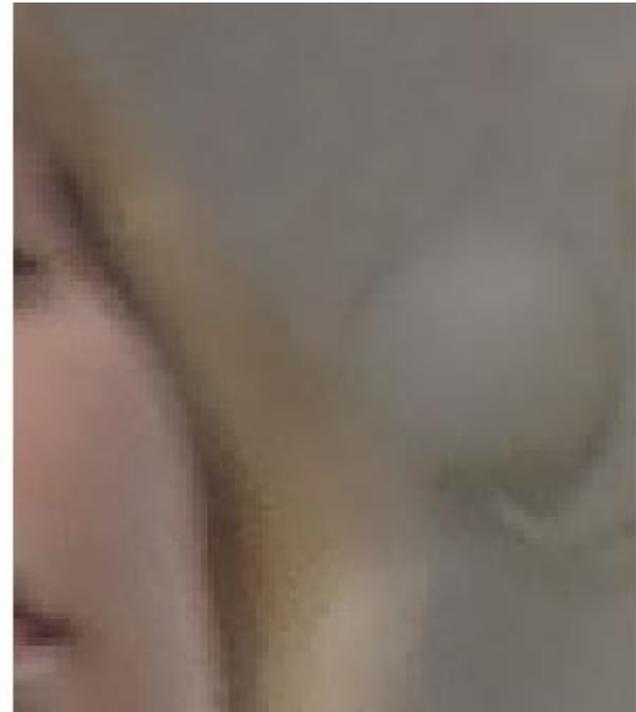
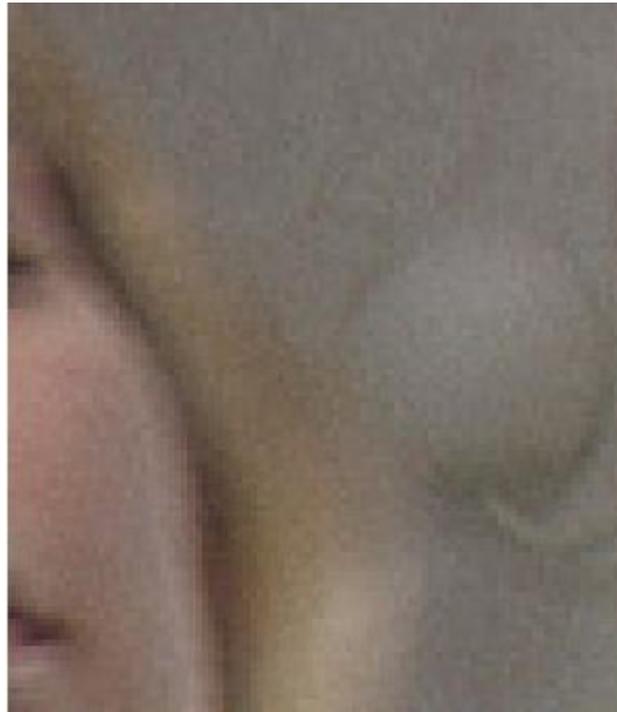
Адаптивный фильтр - пример

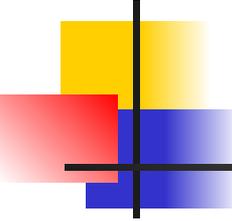
```
for (each pixel of the current video frame)
{
    GetRGB (source_pixel, r, g, b);
    tot_red = tot_green = tot_blue = 0;
    count_red = count_green = count_blue = 0;
    for (each pixel in the specified radius)
    {
        GetRGB (neighbour_pixel, r1, g1, b1);
        if (abs(r1-r) < Threshold)
            {tot_red += r1; count_red ++;}
        if (abs(g1-g) < Threshold)
            {tot_green += g1; count_green ++;}
        if (abs(b1-b) < Threshold)
            {tot_blue += b1; count_blue ++;}
    }
    destination_pixel = RGB (tot_red / count_red,
                            tot_green / count_green ,
                            tot_blue / count_blue );
}
```



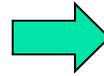
Адаптивные фильтры

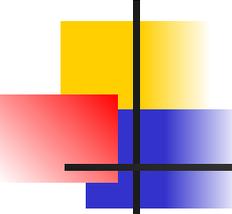
- Примеры таких фильтров:
<http://www.compression.ru/video/denoising/denoising.pdf>





«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
- 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.

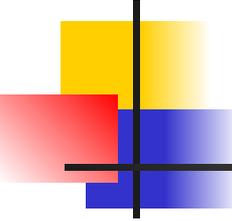
Что лучше? – зависит от конкретной задачи

Повышение резкости

Ядро свертки

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





Как бороться с шумом аппаратуры?

Предположим, камера, которой производится съемка заметно «шумит». Обычно шум измерительной аппаратуры моделируется как случайная нормально распределенная случайная величина с нулевым средним - $Err(i, j)$;

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

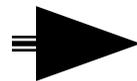
$$\bar{I}(i, j) = \frac{1}{N} \sum_{k=1}^N I_k(i, j);$$

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

Примеры шумоподавления

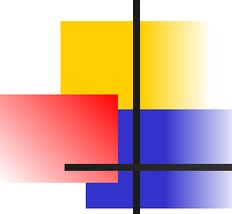


Зашумленные изображения



Усреднение по 10
изображениям

Так работают камеры в некоторых сотовых телефонах



Примеры шумоподавления



Исходное изображение



Испорченное
изображение

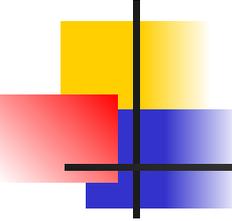
Примеры шумоподавления



Усреднение по 9
пикселям (3x3)



Медианный фильтр
(3x3)



Компенсация разности освещения

Пример

Sonnet for Lena

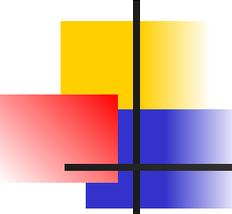
O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst



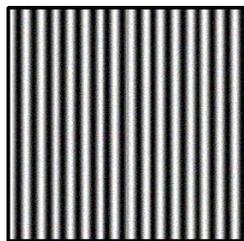
Компенсация разности освещения

Идея:

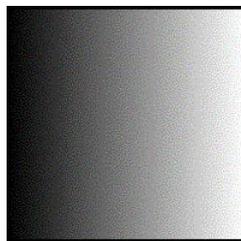
Формирование изображения:

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

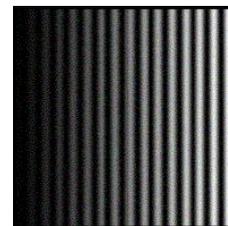
Плавные изменения яркости относятся к освещению,
резкие - к объектам.



объект $f(i, j)$



освещение $l(i, j)$



Изображение
освещенного
объекта $I(i, j)$

Выравнивание освещения

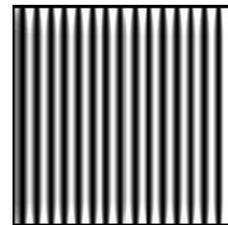
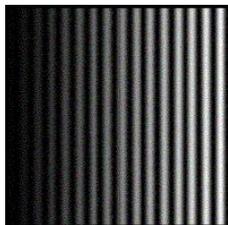
- Алгоритм

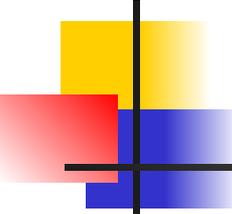
- Получить приближенное изображение освещения путем низкочастотной фильтрации

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

- Восстановить изображение по формуле

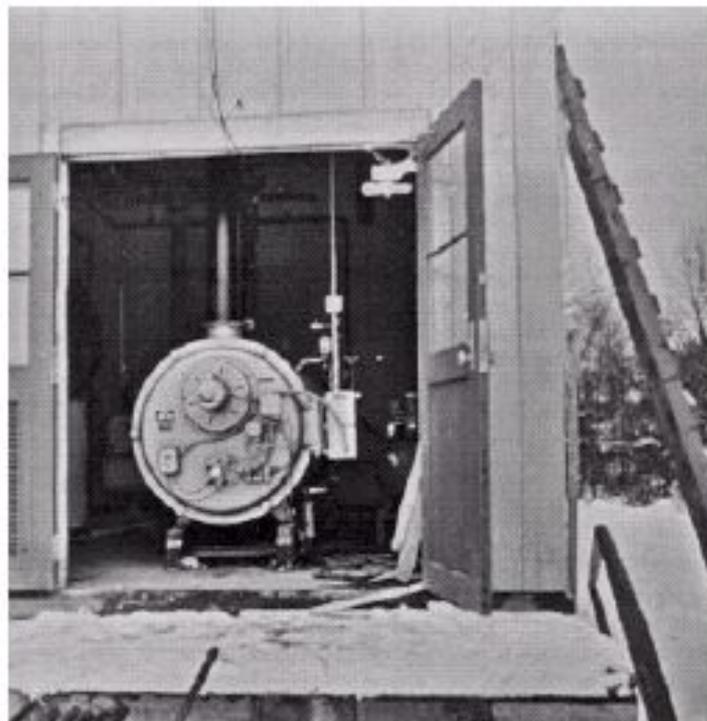
$$f'(i, j) = \frac{I(i, j)}{l'(i, j)}$$





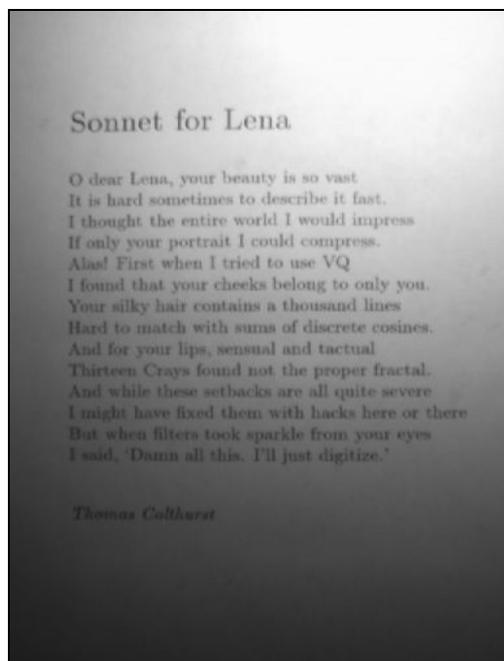
Выравнивание освещения

Пример



Компенсация разности освещения

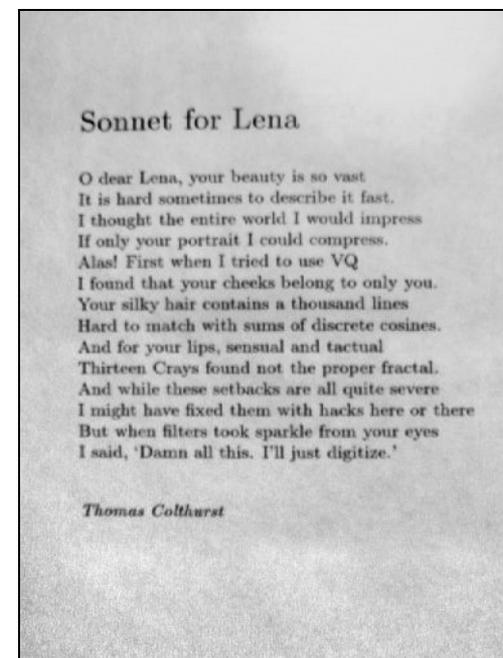
Пример



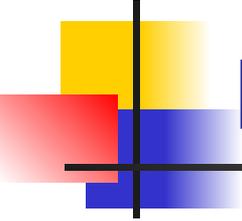
/



=



Gauss 14.7 пикселей



План лекции

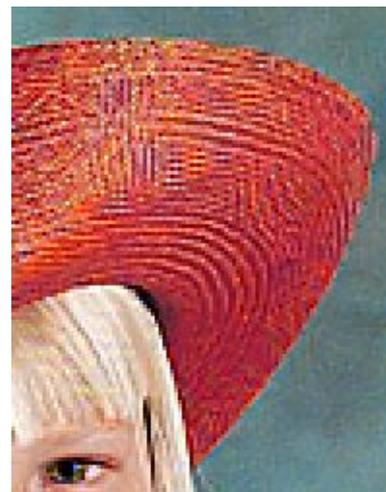
- ✓ Введение
- ✓ Коррекция контрастности/яркости изображения
- ✓ Коррекция цветового баланса изображения
- ✓ Подавление шума в изображениях
 - Метрики качества
 - Подчеркивание резких границ (краев) на изображении
 - Спецэффекты

Метрики качества

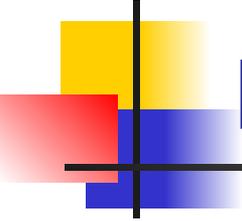
- Как измерить похожесть двух изображений?



исходное
изображение



искаженное
изображение



Метрики качества

- Среднеквадратичная ошибка (MSE)

$$MSE = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2$$

N – число пикселей

- Пиковое отношение сигнал/шум (PSNR)

$$PSNR_{dB} = 10 \lg \frac{M^2}{MSE}$$

M – максимальное значение пикселя

Метрики качества

- PSNR и MSE не учитывают особенности человеческого восприятия!



Оригинал

Далее будут использованы рисунки из статьи Wang, Bovik, Lu "WHY IS IMAGE QUALITY ASSESMENT SO DIFFICULT?"

Метрики качества

- У этих изображений одинаковые PSNR с оригиналом (примерно 25 dB)



Повышена контрастность



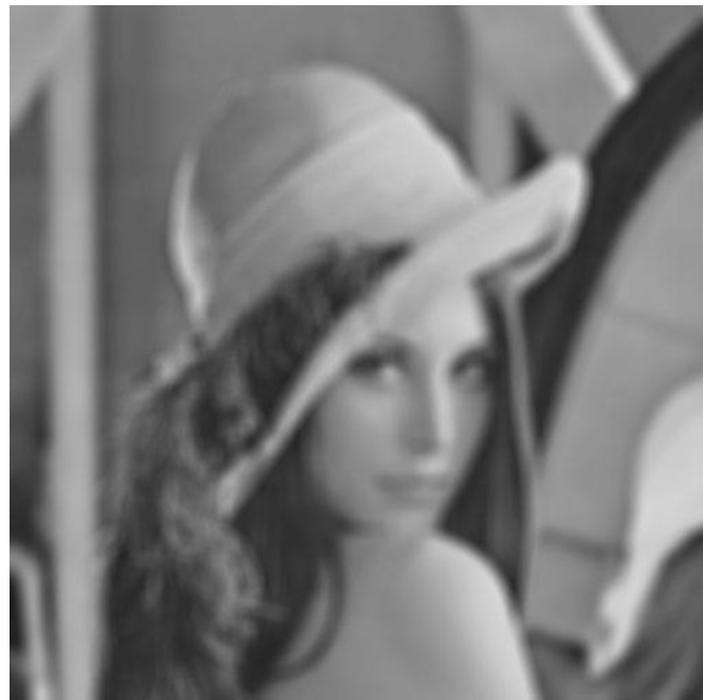
Добавлен белый гауссов шум

Метрики качества

- И у этих – тоже примерно 25 dB!



Добавлен импульсный шум



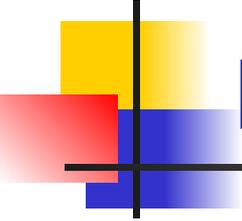
Размытие

Метрики качества

- И у этого – тоже!



Артефакт блочности после JPEG

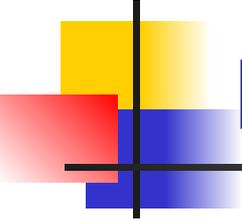


Метрики качества

- Вывод: PSNR не всегда отражает реальный видимый уровень искажений.
- Как улучшить?

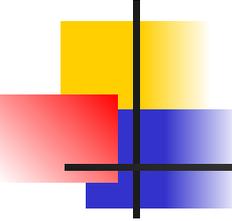
HVS
models
(human visual
system)

- Использовать функцию чувствительности глаза к различным частотам (CSF)
- Использовать свойство маскировки
- Использовать равномерные к восприятию цветовые пространства (CIE Lab, CIEDE2000)



План лекции

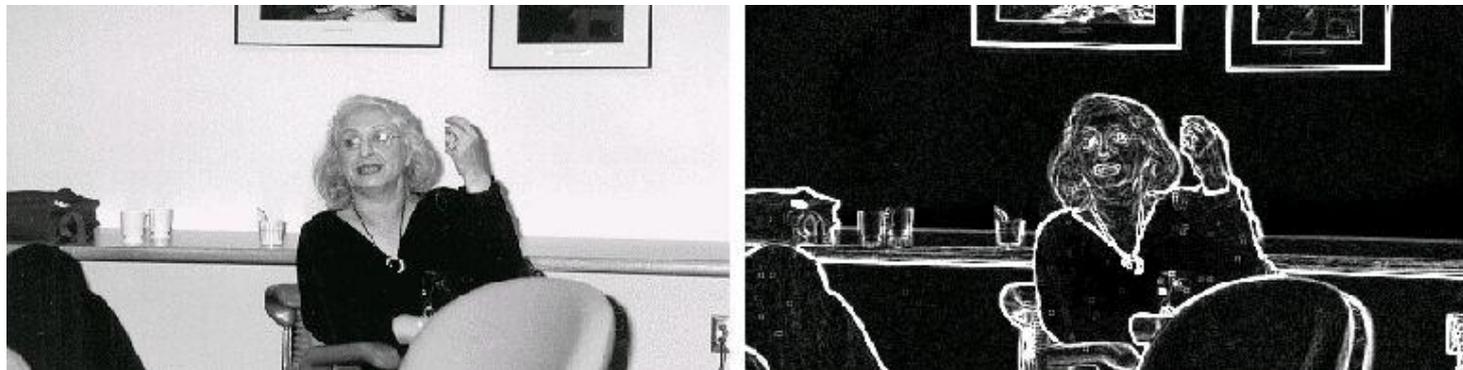
- ✓ Введение
- ✓ Коррекция контрастности/яркости изображения
- ✓ Коррекция цветового баланса изображения
- ✓ Подавление шума в изображениях
- ✓ Метрики качества
 - Подчеркивание резких границ (краев) на изображении
 - Спецэффекты

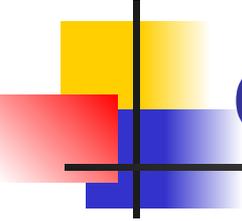


Подчеркивание контуров объекта

Рассмотрим подчеркивание краев (границ)

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





Операция оконтуривания объекта

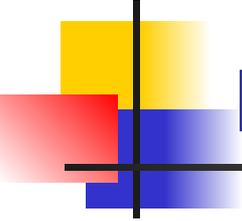
При работе с бинарными изображениями контуры объекта можно получить с помощью операций математической морфологии

Внутреннее оконтуривание

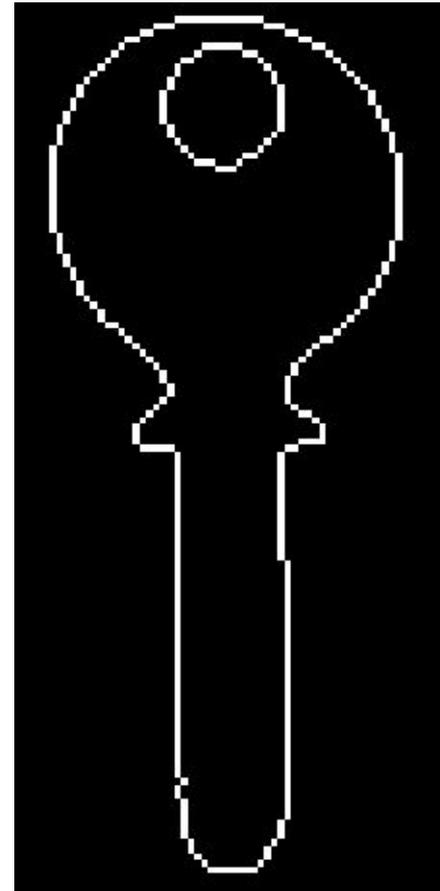
- $C_I = A - (A (-) B)$

Внешнее оконтуривание

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

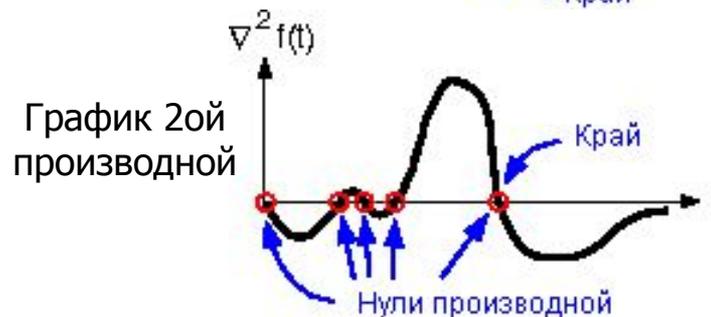
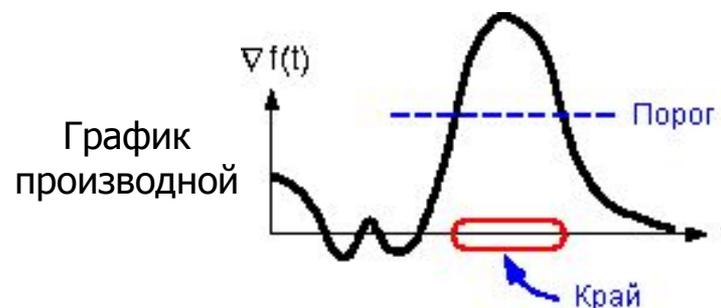
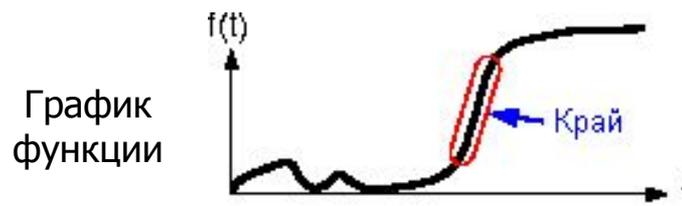


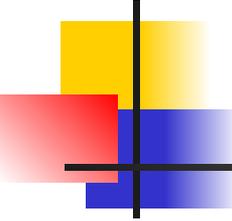
Пример оконтуривания объекта



Подчеркивание краев

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





Подчеркивание краев

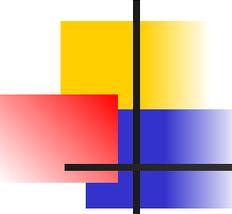
Известно, что наибольшее изменение функции происходит в направлении ее градиента. Величина изменения измеряется абсолютной величиной градиента.

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

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

Часто используется приближенное вычисление градиента:

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



Подчеркивание краев

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

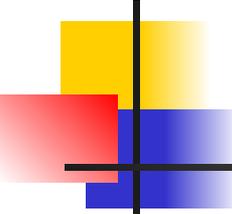
$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\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

Робертса

Превитт

Собея

Математический смысл – приближенное вычисление производных по направлению.



Подчеркивание краев

Примеры применения операторов подчеркивания краев:



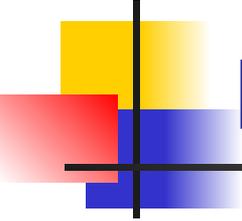
Робертса



Превитт

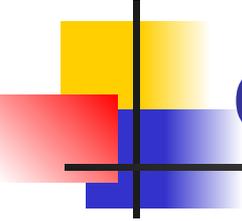


Собеля



План лекции

- ✓ Введение
- ✓ Коррекция контрастности/яркости изображения
- ✓ Коррекция цветового баланса изображения
- ✓ Подавление шума в изображениях
- ✓ Метрики качества
- ✓ Подчеркивание резких границ (краев) на изображении
 - Спецэффекты



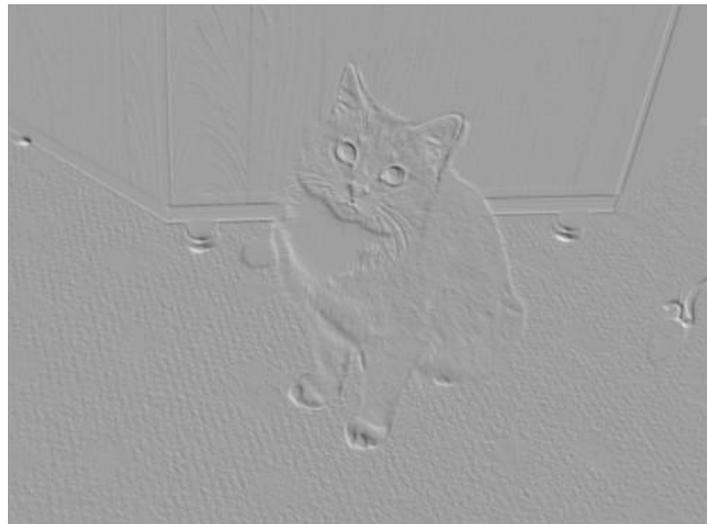
Спецэффекты

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

Тиснение

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

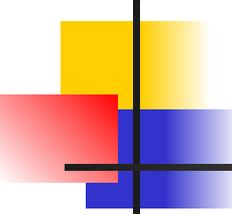
Фильтр + сдвиг яркости, нормировка...



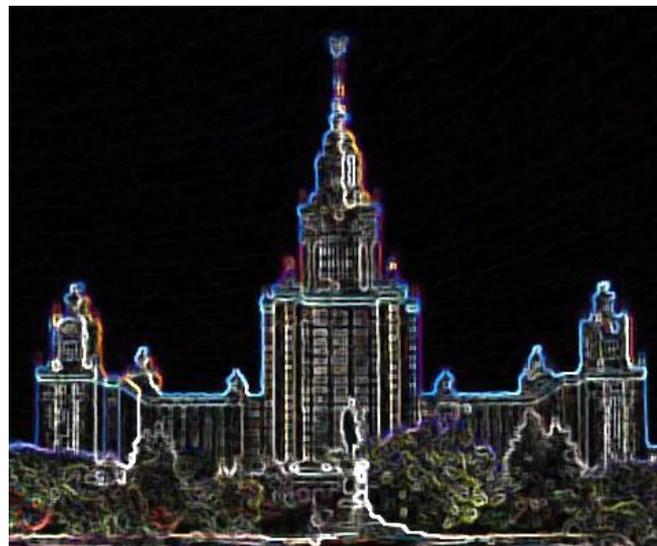
Цифровой негатив



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



Светящиеся края



Медианный фильтр + выделение краев + фильтр «максимума»

Перенос/поворот



Перенос:

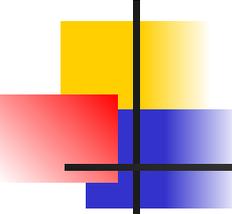
$$x(k; l) = k + 50; y(k; l) = l;$$

Поворот:

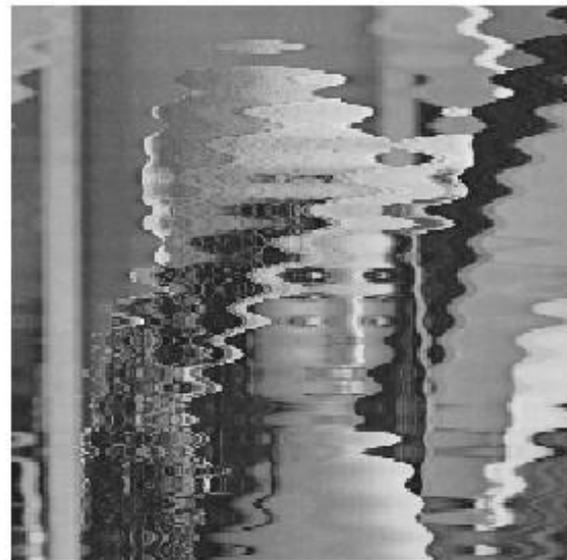
$$x(k; l) = (k \cdot x_0)\cos(\mu) + (l \cdot y_0)\sin(\mu) + x_0;$$

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

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



«Волны»

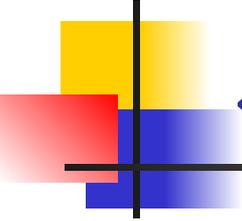


Волны 1:

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

Волны 2:

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



«Эффект стекла»



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