

# Компьютерная обработка изображений

## Лекция 6: Фильтрация шумов.

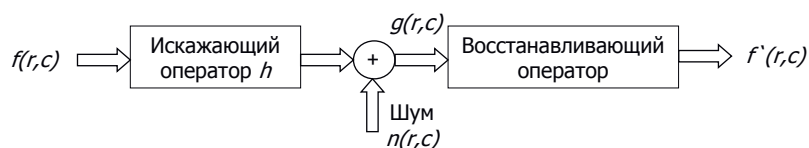
Сафонов И.В., Крыжановский К.А., Егорова М.А.

2011

1

## Восстановление и улучшение изображений

Модель процесса искажения/восстановления изображения



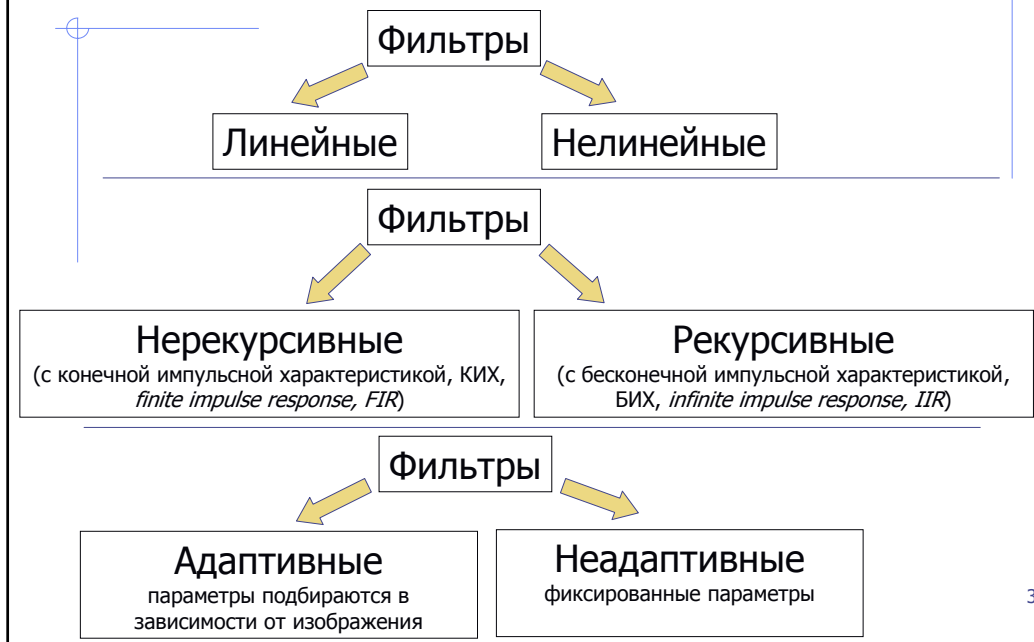
Если  $h$  линейный оператор, а шум аддитивный, то искажение в пространственной области может быть представлено в виде:  $g(r,c) = h(r,c) * f(r,c) + n(r,c)$   
в частотной области:  $G(u,v) = H(u,v)F(u,v) + N(u,v)$

Все алгоритмы повышения качества (коррекции) изображений разбиваются на две группы:

- алгоритмы восстановления изображений (*image restoration*) – искажающий оператор  $h$  и модель шума  $n$  известны, используются строгие математические подходы для нахождения  $f'(r,c)$ ;
- алгоритмы улучшения изображений (*image enhancement*) –  $h$  и  $n$  не известны точно, используются эвристические подходы для повышения качества  $f'(r,c)$  с точки зрения человеческого восприятия или решаемой задачи.

2

## Классификация фильтров



3

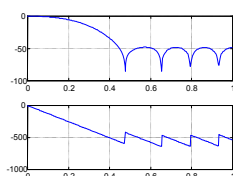
## Фильтрация шумов

На практике модель шума, как правило, неизвестна достаточно точно. Поэтому алгоритмы улучшения изображений используются гораздо чаще алгоритмов восстановления.

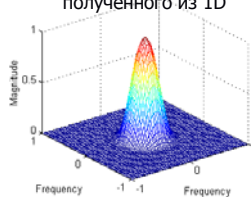
Существует огромное множество различных фильтров для подавления различных типов шумов (*noise reduction, noise suppression*). Ниже рассмотрена только небольшая часть способов фильтрации для пояснения базовых концепций.

Также необходимо помнить, что для фильтрации изображений могут быть применены фильтры, которые используются для фильтрации 1D сигналов и изучались в курсе ЦОС. Эти фильтры могут быть применены к строкам, а затем к столбцам изображения, или трансформированы в 2D фильтр, например, с помощью преобразования Мак-Клеллана (*McClellan transform*).

АЧХ и ФЧХ 1D КИХ фильтра



АЧХ 2D фильтра, полученного из 1D



Исходное изображение



Результат фильтрации

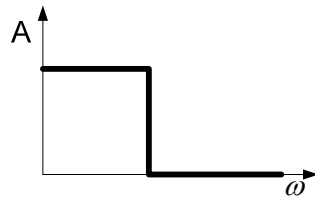


4

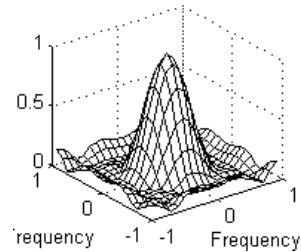
## Линейные ФНЧ: АЧХ

Свертка реализует линейный фильтр низких частот (ФНЧ, *Low-pass Filter, LPF*) в том случае, когда сумма элементов ядра свертки равна 1, а все элементы ядра являются неотрицательными.

АЧХ одномерного идеального ФНЧ



АЧХ двумерного ФНЧ



Высокие частоты – это различного рода помехи, текстуры, контурные перепады (границы). Низкие частоты – это плавные изменения яркости. Линейные ФНЧ называют также размывающими (*blurring*) и сглаживающими (*smoothing*) фильтрами.

5

## Линейные ФНЧ: Усредняющий (spatial)

Алгоритм работы усредняющего фильтра заключается в замене значения яркости в текущем пикселе на среднюю яркость вычисленную в окрестности данного пиксела. К недостаткам его можно отнести сильное уменьшение резкости изображения. Достоинство – такие фильтры имеют очень быстрые схемы вычислений.

Примеры ядер свертки:

$$\frac{1}{5} \times \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \xleftarrow{\text{box filters}} \frac{1}{25} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

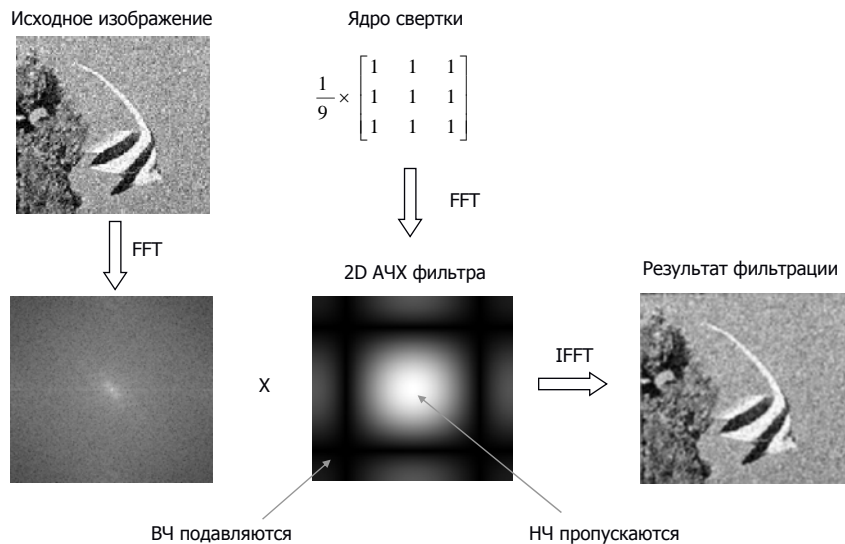
Пример вычисления ФНЧ в пространственной области

$$I = \begin{bmatrix} 5 & 7 & 8 & 2 & 1 \\ 3 & 2 & 4 & 1 & 3 \\ 6 & 4 & 6 & 2 & 0 \\ 1 & 0 & 1 & 7 & 3 \end{bmatrix} \quad H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad I' = \begin{bmatrix} 5 & 7 & 8 & 2 & 1 \\ 3 & 2 & 4 & 1 & 3 \\ 6 & 4 & 6 & 2 & 0 \\ 1 & 0 & 1 & 7 & 3 \end{bmatrix} \quad I'' = \begin{bmatrix} 5 & 7 & 8 & 2 & 1 \\ 3 & 2 & 4 & 1 & 3 \\ 6 & 4 & 6 & 2 & 0 \\ 1 & 0 & 1 & 7 & 3 \end{bmatrix}$$

6

## Линейные ФНЧ: Усредняющий (frequency)

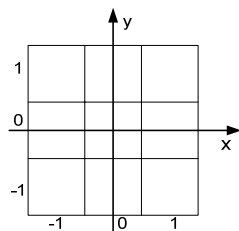
Пример вычисления ФНЧ в частотной области



7

## Линейные ФНЧ: Гауссиан

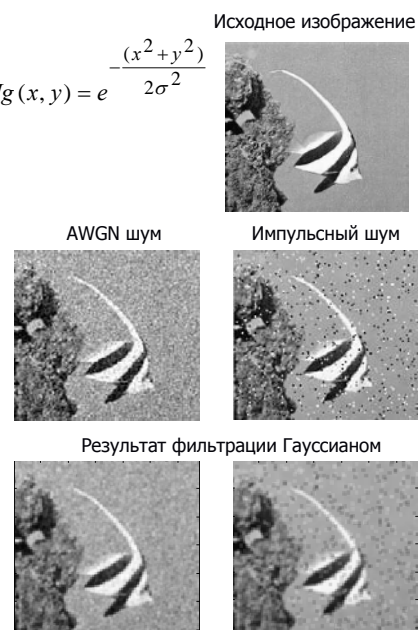
Гауссиан:  $H(x, y) = \frac{Hg(x, y)}{\sum_x \sum_y Hg(x, y)}$ ,  $Hg(x, y) = e^{-\frac{(x^2 + y^2)}{2\sigma^2}}$



Пример Гауссиана:

$$\frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Линейные ФНЧ предназначены для подавления аддитивного белого шума. Импульсные шумы подавляются ими недостаточно хорошо.



8

## Усредняющий фильтр с порогом

Для подавления импульсных шумов существует простая модификация усредняющего фильтра: замена значения яркости на среднее производится в том случае, если разность между значением яркости и средним превышает порог. Такой фильтр является нелинейным.

$$I'(r,c) = \begin{cases} I_c(r,c), & \text{если } |I_c(r,c) - I(r,c)| \geq T \\ I(r,c), & \text{иначе} \end{cases}$$

$I$  - исходное изображение,  
 $I_c$  - результат свертки со сглаживающей маской,  
 $I'$  - новое изображение  
 $T$  - пороговое значение

Исходное изображение



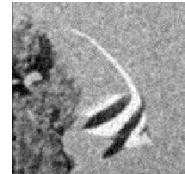
AWGN шум



Импульсный шум



Результат фильтрации усредняющим фильтром с порогом

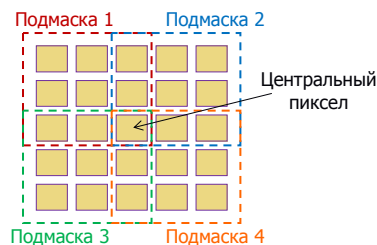


9

## Фильтр Кувахары

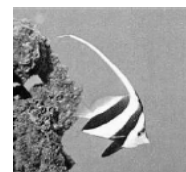
Нелинейный нерекурсивный фильтр Кувахары (*Kuwahara*) предназначен для подавления аддитивных шумов с меньшим размытием границ, чем в случае линейных ФНЧ.

Размер маски:  $K=J=4L+1$ ,  $L=1, 2, 3 \dots$   
 Например, 5x5, 9x9, 13x13.



1. Маска разбивается на 4 области размером:  $[(J+1)/2] \times [(K+1)/2]$ .
2. Для каждой подмаски вычисляется математическое ожидание  $m_i$  и дисперсия  $\sigma_i^2$ .
3. Центральному пикселу  $x$  присваивается значение  $m_i$ , дисперсия которого  $\sigma_i^2$  минимальна.

Исходное изображение



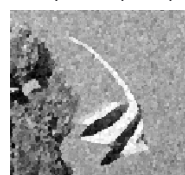
AWGN шум



Импульсный шум



Результат фильтрации фильтром Кувахары



10

## Билатеральный фильтр (1)

Нелинейный нерекursивный билатеральный (*bilateral*) фильтр предназначен для подавления аддитивных шумов с предохранением границ (*edge preserving filter*):

$$I_f(r, c) = \frac{\sum_{i=-S/2}^{S/2} \sum_{j=-S/2}^{S/2} I(r+i, c+j) \times v(i, j) \times w(I(r+i, c+j) - I(r, c))}{\sum_{i=-S/2}^{S/2} \sum_{j=-S/2}^{S/2} v(i, j) \times w(I(r+i, c+j) - I(r, c))}$$

$v(i, j)$  – пространственное ядро весов размера  $S \times S$

$w(x)$  – функция весов в зависимости от фотометрического расстояния. Также называется функцией по диапазону и edge-stop функцией.

Традиционный билатеральный фильтр использует Гауссианы для обеих весовых функций:

$$v(i, j) = \exp\left(-\frac{i^2 + j^2}{2\sigma_D^2}\right) \quad w(x) = \exp\left(-\frac{x^2}{2\sigma_R^2}\right)$$

Исходное изображение



AWGN шум



Импульсный шум

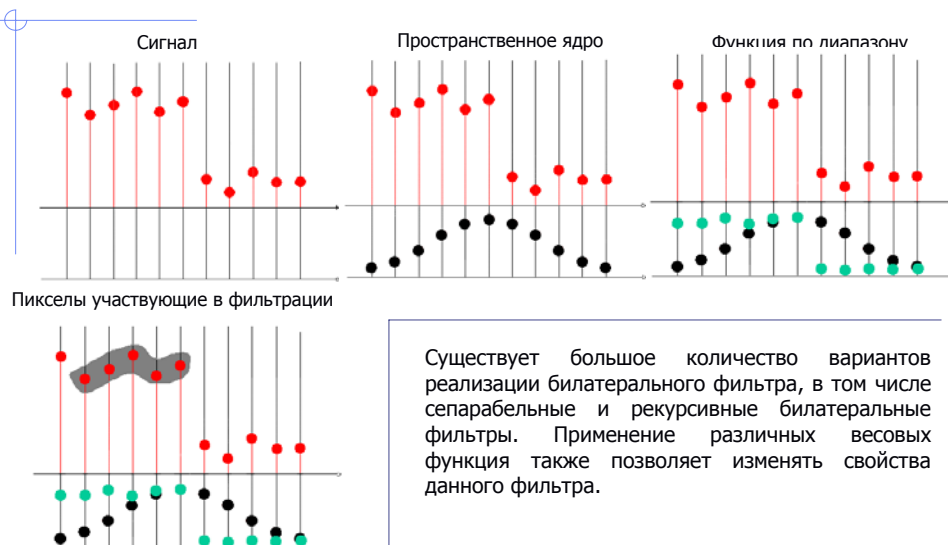


Результат фильтрации билатеральным фильтром



11

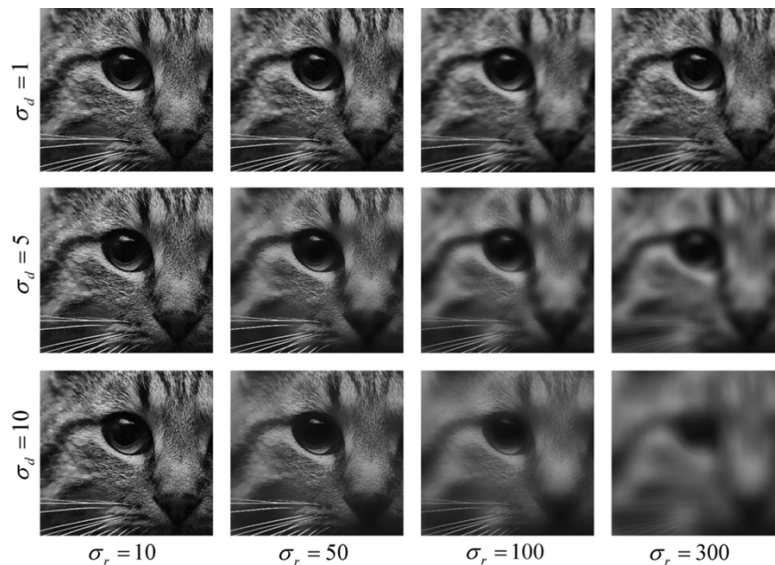
## Билатеральный фильтр (2)



Существует большое количество вариантов реализации билатерального фильтра, в том числе сепарабельные и рекурсивные билатеральные фильтры. Применение различных весовых функций также позволяет изменять свойства данного фильтра.

## Билатеральный фильтр (3)

Влияние параметров билатерального фильтра на его работу



13

## Билатеральный фильтр (4)

Другие варианты весовых функций  $w(x)$ :

- Волна Эндрю (Andrew's wave): 
$$g(x) = \begin{cases} \sin(\pi x / \sigma) / (\pi x \sigma) : |x| \leq \sigma \\ 0 : |x| > \sigma \end{cases}$$

- Эл Фалла Форд (El Fallah Ford): 
$$g(x) = \frac{1}{\sqrt{1 + (x / \sigma)^2}}$$

- Минимакс Хубера (Huber's minimax): 
$$g(x) = \begin{cases} 1 / \sigma : |x| \leq \sigma \\ \text{sign}(x) / x : |x| > \sigma \end{cases}$$

- Лоренциан (Lorentzian): 
$$g(x) = \frac{2}{2\sigma^2 + x^2}$$

- Туки (Tukey's bi-weight): 
$$g(x) = \begin{cases} 0.5 \times (1 - (x / \sigma)^2)^2 : |x| \leq \sigma \\ 0 : |x| > \sigma \end{cases}$$

- Плоская (Flat): 
$$g(x) = \begin{cases} 1 / \sigma : |x| \leq \sigma \\ 0 : |x| > \sigma \end{cases}$$

- Косинусная: 
$$g(x) = \begin{cases} \cos(\frac{\pi x}{2\sigma}) : |x| \leq \sigma \\ 0 : |x| > \sigma \end{cases}$$

Если данные функции преобразовать в двумерные, то их можно использовать как пространственные ядра

14

## Фильтр нелокального усреднения (1)

Нелинейный нерекursивный фильтр нелокального усреднения (*non-local means*, *NL-means*) предназначен для подавления аддитивных шумов с предохранением границ и текстур (*edge and texture preserving filter*):

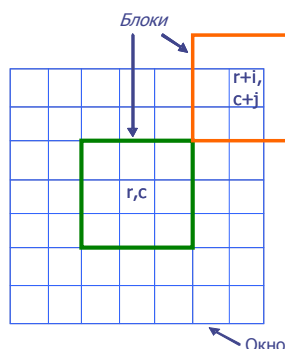
$$I_f(r, c) = \frac{\sum_{i=-S/2}^{S/2} \sum_{j=-K/2}^{K/2} I(r+i, c+j) \times w(I, r+i, c+j)}{\sum_{i=-S/2}^{S/2} \sum_{j=-K/2}^{K/2} w(I, r+i, c+j)}$$

$I$  – исходное изображение,  
 $I_f$  – результирующее изображение,  
 $S$  и  $K$  – размеры скользящего окна, в пределах которого происходит усреднение пикселей исходного изображения с весами, задаваемыми функцией сходства  $w(I, r+i, c+j)$  между блоками размера  $M \times L$ :

$$w(I, r+i, c+j) = e^{-\frac{\sum_{k=-M/2}^{M/2} \sum_{n=-L/2}^{L/2} (I(r+i+k, c+j+n) - I(r+k, c+n))^2 \cdot e^{-(k^2 + n^2) / 2\sigma^2}}{h^2}}$$

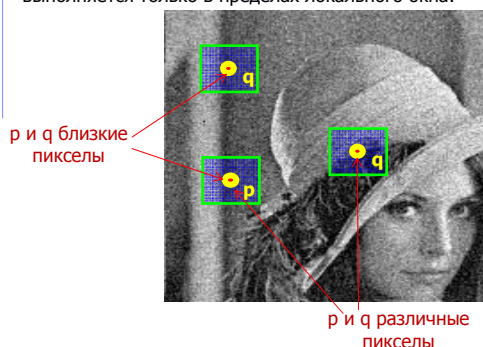
$h$  и  $\sigma$  – параметры фильтрации.

Следует отметить, что возможны и другие варианты функции сходства.



## Фильтр нелокального усреднения (2)

Общей идеей фильтра нелокального усреднения является поиск похожих друг на друга блоков изображения по всему изображению и усреднение значений по этим блокам. Однако поиск по всему изображению требует гигантских вычислительных затрат, поэтому выполняется только в пределах локального окна.



Несмотря на поиск сходных блоков только в пределах локального окна, NL-means фильтр требует значительных вычислительных затрат. По этой причине существует несколько вариантов "быстрых" NL-means за счет быстрого сравнения блоков. Также NL-means может быть эффективно распараллелен, например для работы на GPU.

Исходное изображение



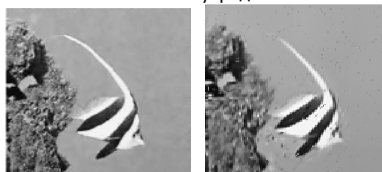
AWGN шум



Импульсный шум



Результат фильтрации фильтром нелокального усреднения





## Обобщенный билатеральный фильтр

Между билатеральным фильтром и NL-means фильтром много общего. Различие в том, что в билатеральном фильтре ищется сходство между значениями отдельных пикселей, а в NL-means сходство между блоками некоторого фиксированного размера. Несмотря на название и первоначальную идею в реализации NL-means фильтр, как правило, является локальным. Размер блока влияет на способность к предохранению различных типов текстур. Существует несколько вариантов обобщения этих фильтров в единый фильтр. Например два таких:

$$Y_{f+}(r, c) = \frac{\sum_{i=-S/2}^{S/2} \sum_{j=-S/2}^{S/2} Y(r+i, c+j) \times V(i, j) \times W(\sum_k a_k D_k(P_k(r+i, c+j), P_k(r, c)))}{\sum_{i=-S/2}^{S/2} \sum_{j=-S/2}^{S/2} V(i, j) \times W(\sum_k a_k D_k(P_k(r+i, c+j), P_k(r, c)))}$$

$$Y_{f \times}(r, c) = \frac{\sum_{i=-S/2}^{S/2} \sum_{j=-S/2}^{S/2} Y(r+i, c+j) \times V(i, j) \times W(\prod_k a_k D_k(P_k(r+i, c+j), P_k(r, c)))}{\sum_{i=-S/2}^{S/2} \sum_{j=-S/2}^{S/2} V(i, j) \times W(\prod_k a_k D_k(P_k(r+i, c+j), P_k(r, c)))}$$

$S$  – размер пространственного ядра,  $V(i, j)$  – пространственное ядро,  $W(x)$  – весовая функция,  $D_k(P_k(r+i, c+j), P_k(r, c))$  – функция расстояния (сходства) между блоками размера  $k$  на  $k$ ,  $P_k(r, c)$  – блок пикселей,  $a_k$  – веса блоков разного размера,  $(r, c)$  – координаты центра блока.

Оба этих фильтра превосходят по критерию PSNR как билатеральный, так и NL-means. Однако данные обобщенные фильтры имеют крайне высокую вычислительную сложность и большое количество параметров, которые сложно оптимально настроить.

## Ранговая фильтрация

Статистические порядковые (*order-statistic*) или ранговые фильтры являются нелинейными нерекурсивными фильтрами, в которых пиксели в окрестности некоторого пикселя упорядочиваются по возрастанию, и в качестве результата выбирается  $k$ -е по порядку значение. Иногда  $k$  называют рангом.

Окрестность пикселя задается с помощью структурного элемента (маски, апертурного фильтра). Текущий пиксел соответствует центральному элементу маски. Ненулевые элементы маски указывают на пиксели участвующие в составлении упорядоченного ряда.

Примеры структурных элементов:

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

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

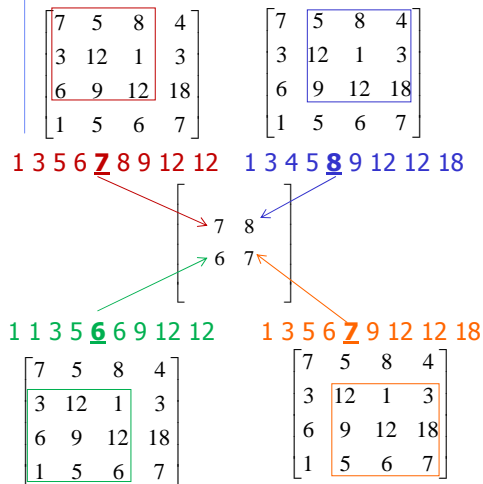
$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Основной идеей порядковой фильтрации является то, что поврежденные шумом пиксели будут находиться по краям отсортированного ряда, а в центре ряда окажутся неповрежденные значения. Это справедливо особенно для импульсного шума.

Важнейшим частным случаем порядкового фильтра является медианный фильтр, в котором результатом является медиана ряда, т.е. результирующее значение берется из середины отсортированного ряда. Если в маске четное количество ненулевых элементов, то в качестве медианы берут среднее двух центральных элементов отсортированного ряда.

## Медианный фильтр

$$I_f(r, c) = \text{med}_{(s, t) \in S_{rc}} \{I(s, t)\}$$



Исходное изображение



AWGN шум



Импульсный шум



Результат фильтрации медианным фильтром



19

## Модификации медианного фильтра

Существует несколько модификаций медианного фильтра, которые можно обобщить на ранговые фильтры в целом.

Например, существует несколько вариантов взвешенного медианного фильтра. В одном из простейших способов (предложен Brownrigg) значение в маске указывает сколько раз соответствующий пиксел необходимо учесть в отсортированном ряде.

Изображение	Маска	
$\begin{bmatrix} 7 & 5 & 8 & 4 \\ 3 & 12 & 1 & 3 \\ 6 & 9 & 12 & 18 \\ 1 & 5 & 6 & 7 \end{bmatrix}$	$\begin{bmatrix} 0 & 2 & 0 \\ 2 & 1 & 2 \\ 0 & 2 & 0 \end{bmatrix}$	1) 1 1 3 3 <u>5</u> 5 9 9 12 2) 1 3 3 8 <u>8</u> 12 12 12 12 3) 5 5 6 6 <u>9</u> 12 12 12 12 4) 1 1 6 6 <u>9</u> 9 12 18 18

20

## Фильтр усеченного среднего

Положительные черты порядкового и усредняющего фильтров объединены в нелинейном фильтре усеченного среднего (*trimmed mean*).

Шаги фильтрации:

1. Из рассмотрения удаляются  $d/2$  наименьших и  $d/2$  наибольших значений яркости  $I$  в окрестности  $S_{xy}$
2. Оставшиеся значения усредняются

$$I_f(r, c) = \frac{1}{KL - d} \sum_{(s, t) \in S_{rc}} I_r(s, t)$$

где  $I_f$  – результат фильтрации  
 $S_{rc}$  – окрестность пиксела  $(r, c)$  размером  $K \times L$   
 $d$  – количество пикселей из окрестности, которые не участвуют в усреднении,  $d \in [0, KL - 1]$   
 $I_r$  – значения пикселей из окрестности, участвующие в усреднении

$d = 0 \Rightarrow$  усредняющий фильтр  
 $d/2 = (KL - 1) \Rightarrow$  медианный фильтр

Исходное изображение



AWGN шум



Импульсный шум



Результат фильтрации фильтром усеченного среднего



21

## Сравнение фильтров по PSNR

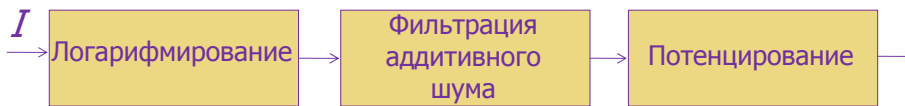
Несмотря на ряд недостатков критерия PSNR, принято сравнивать алгоритмы для фильтрации шумов по этому критерию. Для AWGN и импульсного шума PSNR достаточно хорошо коррелирует с человеческим восприятием, хотя иногда наибольшие значения PSNR соответствуют изображениям размытым сильнее, чем этого бы хотелось. Для большинства приведенных выше примеров работы фильтров параметры были подобраны с целью максимизации PSNR.

Фильтр\Шум	Гаусса	Импульсный	Гаусса+Импульсный
Без фильтра	20,205	18,1951	16,4604
Фильтр Гаусса	25,1465	24,1405	22,7913
Медианный фильтр	23,8991	26,8375	23,6356
Усредняющий фильтр с порогом	22,134	28,624	21,1638
Фильтр Кувахары	22,9473	24,9451	21,719
Билатеральный фильтр	25,7038	23,1969	22,7039
Фильтр нелокального усреднения	32,7541	28,2687	24,7914

22

## Гомоморфная фильтрация

Гомоморфная фильтрация применяется для подавления мультипликативного шума:  $I' = I * N$



В результате логарифмирования мультипликативный шум преобразуется в аддитивный:

$$\ln I = \ln I' + \ln N$$

Далее удаляется шумовая составляющая, используя алгоритмы для фильтрации аддитивного шума.

Затем производится операция потенцирования для преобразования сигнала к исходному диапазону.

23

## Адаптивная фильтрация AWGN

Для известной искажающей функции и модели шума существует теория адаптивной винеровской фильтрации (фильтра минимизации среднеквадратического отклонения). Для AWGN предложен частный случай адаптивного фильтра Винера.

Для фрагмента изображения размером  $M \times N$ , вычисляют среднее  $\mu$  и дисперсию  $\sigma^2$ .

$$I_n(r, c) = \mu + \frac{\sigma^2 - V^2}{\sigma^2} (I(r, c) - \mu)$$

где  $V$  — это дисперсия шума для всего изображения.

Если ее нельзя определить из априорных сведений, тогда полагают либо равной дисперсии всего изображения, либо средней из всех  $\sigma_f^2$ .

$M \times N$  — не должно быть слишком маленьким.

Исходное изображение



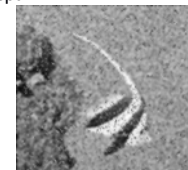
AWGN шум



Импульсный шум



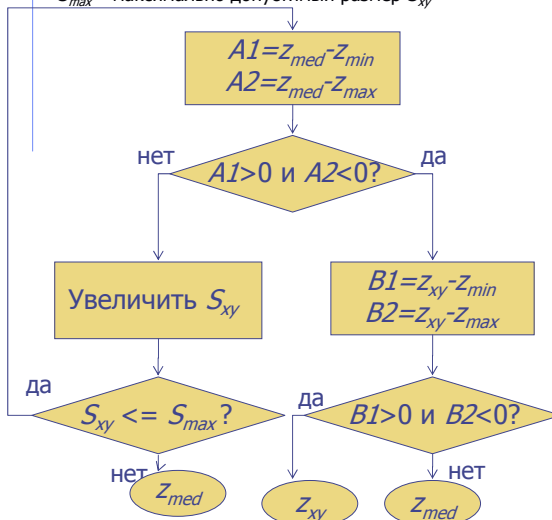
Результат фильтрации адаптивным фильтром Винера



24

## Адаптивный медианный фильтр

$Z_{min}$ ,  $Z_{max}$  – минимальное, максимальное значение яркости в  $S_{xy}$   
 $Z_{med}$  – медиана значений яркости в  $S_{xy}$   
 $Z_{xy}$  – значение яркости в  $(x, y)$   
 $S_{max}$  – максимально допустимый размер  $S_{xy}$



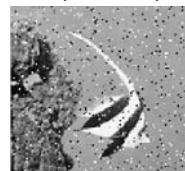
Исходное изображение



AWGN шум



Импульсный шум



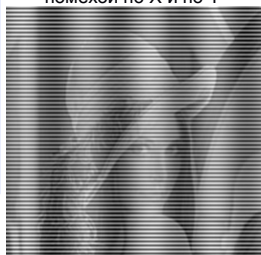
Результат фильтрации адаптивным медианным фильтром



25

## Подавление периодической помехи

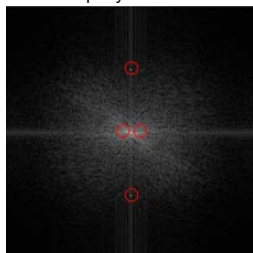
Изображение с sin помехой по X и по Y



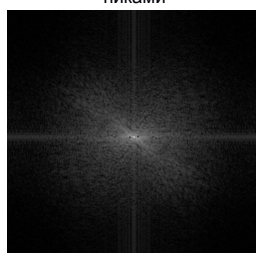
Восстановленное изображение



Спектр. Sin помеха образует пики.



Спектр с обнуленными пиками



Иногда в процессе регистрации или передачи изображений они повреждаются периодическим шумом. Муар также может проявляться в виде периодической регулярной помехи.

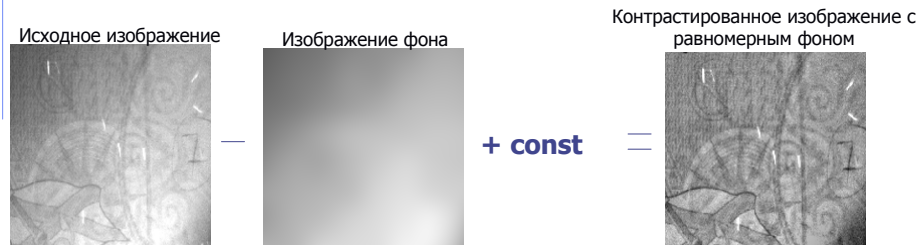
Такой шум на амплитуде спектра образует резкие пики. Подавлять его целесообразно режекторным фильтром в частотной области.

26

## Выравнивание фона

Неравномерный фон (низкочастотный тренд) может мешать выделению на изображении объектов интереса с помощью алгоритмов сегментации.

Общий подход к выравниванию фона (*background equalization*)



Существует несколько способов оценки фона изображения:

1. Иногда фон можно просто снять отдельно от объектов переднего плана.
2. С помощью фильтра низких частот цифровым или оптическим способом (расфокусировка оптической системы).
3. С помощью разбиения изображения на непересекающиеся прямоугольники, оценки фона для каждого прямоугольника, в результате чего получается изображение фона, но меньшего размера чем исходное изображение, масштабирование изображения фона до размеров исходного изображения с помощью билинейной или бикубической интерполяции.