

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Таныгин Максим Олегович  
Должность: и.о. декана факультета фундаментальной и прикладной информатики  
Дата подписания: 21.09.2023 13:12:44  
Уникальный программный ключ:  
65ab2aa0d384efe84806a4c08cedd6c479e411a

## МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра программной инженерии



УТВЕРЖДАЮ:

проректор по учебной работе

Локтионова О.Г.

2022 г.

## ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ ФУНКЦИЙ СРЕДЫ MATLAB

Методические указания для выполнения лабораторных работ по  
дисциплине «Методы и алгоритмы обработки изображений»  
для студентов направления подготовки 09.03.04  
«Программная инженерия»

Курск 2022



## ФИЛЬТРАЦИИ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ ФУНКЦИЙ СРЕДЫ MATLAB

**Цель и задачи лабораторного занятия (лабораторной работы):** изучение функций, позволяющий реализовать процесс зашумления;

освоение методов фильтрации изображений;  
приобретение практических умений и навыков их использования.

**Планируемые результаты обучения (формируемые знания, умения, навыки и компетенции):**

*Код и наименование индикатора достижения компетенции, закрепленного за дисциплиной: ПК – 2.2*

*Проводит публичную защиту выполненной работы*

**Необходимые материально-техническое оборудование и материалы:**

1. Класс ПЭВМ - Athlon 64 X2-2.4; Cel 2.4, Cel 2.6, Cel 800.
2. Мультимедиа центр: ноутбук ASUS X50VL PMD T2330/14"/1024Mb/ 160Gb/ сум-ка/проектор inFocus IN24+ .
3. Экран мобильный Draper Diplomat 60x60
4. Доступ в сеть Интернет.

**Шкала оценивания и критерии оценивания выполненной лабораторной работы:**

Форма контроля	Минимальный балл		Максимальный балл	
	балл	примечание	балл	примечание
Лабораторная работа № 1 (Работа с файлами различных типов изображений в среде MATLAB)	2	Выполнил, но «не защитил»	4	Выполнил и «защитил»

**План проведения лабораторного занятия (лабораторной работы)**

## 1. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ

Выполнить следующие действия.

1. Прочитать палитровое изображение из файла Earth.bmp, вывести на экран, преобразовать в полутоновое. Преобразовать изображение, наложив на него гауссов, импульсный и мультипликативный шум. Вывести на экран зашумленные изображения. Отфильтровать зашумленное изображение медианной фильтрацией и обобщенным нелинейным фильтром. Зашумленное и отфильтрованные изображения вывести в одном окне для сравнения.

2. Выполнить фильтрацию изображения из файла Bigbird.bmp:

а) Отфильтровать изображение с помощью масок кругового градиента:

север			северо-восток			восток			юго-восток		
1	1	1	1	1	1	-1	1	1	-1	1	1
1	-2	1	-1	-2	1	-1	-2	1	-1	-2	1
-1	-1	-1	-1	-1	1	-1	1	1	1	1	1
юг			юго-запад			запад			северо-запад		
-1	-1	-1	1	-1	-1	1	1	-1	1	1	-1
1	-2	1	1	-2	1	1	-2	-1	1	-2	-1
1	1	1	1	1	1	1	1	-1	1	1	-1

Ввод масок двумерных линейных фильтров, построение их АЧХ, фильтрацию изображения с их помощью и вывод результата выполнить в цикле;

б) Улучшить изображение с помощью масок фильтров Лапласа вида:

0	1	0	-1	-1	-1	1	-2	1
1	-4	1	-1	8	-1	-2	4	-2
0	1	0	-1	-1	-1	1	-2	1

Ввод масок и вывод результатов выполнить в цикле.

3. Прочитать изображение из файла Technlgy.bmp, вывести на экран, преобразовать в полутоновое. Получить маску фильтра Превитт. Выполнить фильтрацию исходного полутонового фрагмента

маской фильтра Превиитт отдельно по горизонтали и по вертикали и вместе на одном изображении.

4. Прочитать цветное изображение из файла 'Vike.bmp', вывести на экран, преобразовать в полутоновое. Взять в качестве АЧХ фильтра функцию расстояния от начала координат, сформировать по АЧХ маску фильтра и отфильтровать полутоновое изображение. Исходное полутоновое изображение и результаты его обработки вывести на экран.

5. Вывести АЧХ всех фильтров, создаваемых функцией по заданию масок predefinedного фильтра, в одном окне с заголовками для каждого фильтра.

6. Зашумить в цикле полутоновое изображение (исходное взять из файла 'butterfly.bmp' гауссовым шумом с разными дисперсиями: а) по умолчанию (0.01); б)  $\nu = 0.5$  для математического ожидания, равного  $m = 0.5$ .

## 2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Изображения, полученные на выходе оптико–электронных преобразователей, искажены помехами. Это затрудняет визуальный анализ изображений, а также автоматическую компьютерную обработку. При обработке изображений помехами являются и некоторые области самого изображения. Например, при анализе объектов на сложном фоне, фон тоже представляет собой помеху.

При цифровой обработке изображений необходимо устранять геометрические искажения изображений, подавлять шумы различной природы, производить апертурную коррекцию [1, 2, 7, 8].

*Ослабление действия помех достигается **фильтрацией**.*

*Фильтрация изображений производится в пространственной и частотной областях.*

При пространственной фильтрации преобразование выполняется непосредственно над значениями отсчетов изображения. Результатом фильтрации является оценка полезного сигнала изображения.

В некоторых случаях *признаком полезного сигнала являются резкие перепады яркости*. Однако, как правило, частота этих перепадов относительно невелика. В этом случае свойства сигнала

проявляются не только в отдельной точке, но и при анализе ее окрестности [3, 5, 7, 8].

### Функция добавления шума **imnoise**

#### Синтаксис

**g = imnoise(f, type, parameters),**

где **f** - исходное изображение.

Функция **imnoise** моделирует искажение изображения некоторым шумом и создает новое изображение **g** путем *добавления шума к исходному изображению f*.

Функция может добавлять шум трех типов, которые задаются строковой константой **type**. Количество и смысл параметров **parameters** определяются выбранным *типом шума*. Если параметры не указаны, то будут использованы значения по умолчанию:

1) **ID = imnoise(Is, „gaussian”, m,v)** добавляет к изображению **Is** гауссов белый шум с математическим ожиданием **m** и дисперсией **v**. По умолчанию  $m = 0$ ,  $v = 0.01$ ;

2) **ID = imnoise(Is, „salt & pepper”, d)** добавляет к изображению **Is** импульсный (точечный) шум. Параметр **d** определяет плотность шума и равен доле искаженных пикселей. По умолчанию  $d = 0.05$ ;

3) **ID = imnoise(Is, „speckle”,v)** добавляет к изображению **Is** мультипликативный шум **ID = Is+n\*Is**, где **n** – равномерно распределенный случайный шум с математическим ожиданием ноль и дисперсией **v**. По умолчанию  $v = 0.04$ .

Возможность смоделировать шум является ключевым моментом восстановления изображений. Будем предполагать, что шум не зависит от координат пикселей изображения. Рассмотрим различные виды шумов в пространственной области, характеризующихся своими функциями плотности распределения вероятностей, которые представлены в таблице 1.

Таблица 1

#### Характеристики различных видов шумов

Имя	Функция плотности распределения	Математическое ожидание и дисперсия
-----	---------------------------------	-------------------------------------

Равномерная	$p(z) = \begin{cases} \frac{1}{b-a}, & a \leq z \leq b \\ 0, & \text{иначе} \end{cases}$	$m = \frac{a+b}{2}, \sigma^2 = \frac{(b-a)^2}{12}$
Гауссова	$p(z) = \frac{1}{\sqrt{2\pi}b} e^{-(z-a)^2/2b^2}$	$m = a, \sigma^2 = b^2$
«Соль и перец»	$p(z) = \begin{cases} P_a, & z = a \\ P_b, & z = b, (b > a) \\ 0, & \text{иначе} \end{cases}$	$m = aP_a + bP_b$ $\sigma^2 = (a-m)^2 P_a + (b-m)^2 P_b$
Релея	$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b}, & z \geq a \\ 0, & z < a \end{cases}$	$m = a + \sqrt{2\pi b/4},$ $\sigma^2 = \frac{b(4-\pi)}{4}$
Экспоненциальная	$p(z) = \begin{cases} ae^{-az}, & z \geq 0 \\ 0, & z < 0 \end{cases}$	$m = \frac{1}{a}$ $\sigma^2 = \frac{1}{a^2}$

На рисунке 1 приведены гистограммы сгенерированных шумов, имеющих указанные выше функции плотности распределения.

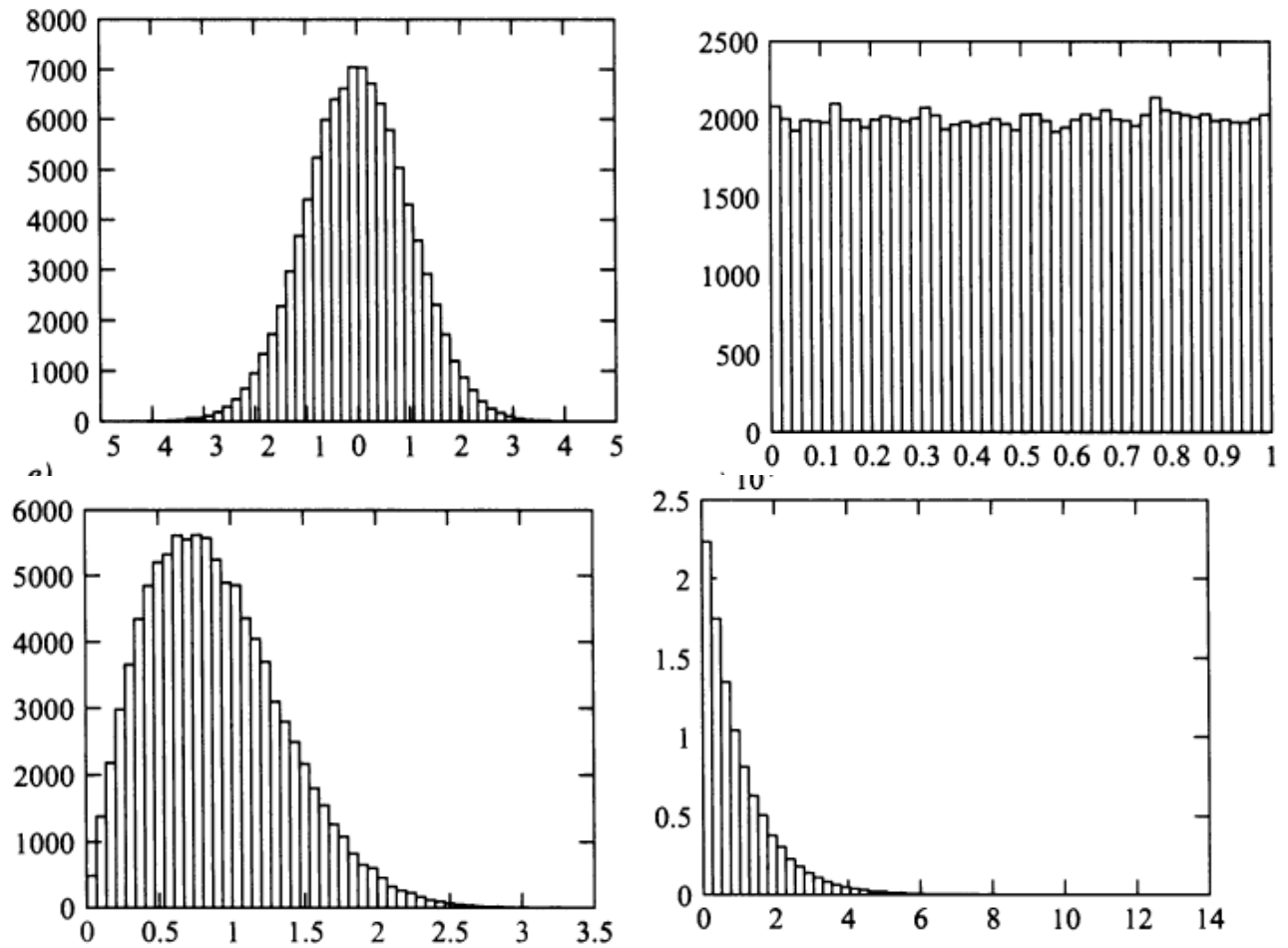


Рисунок 1. Гистограммы различных видов шумов:

а) Гауссов шум; б) равномерный шум; в) шум Релея; г) экспоненциальный шум.

**Задание 1.** Преобразовать изображение, наложив на него гауссов, импульсный и мультипликативный шум.

```
>> [S,map] = imread('c:\Image\Mona.bmp');
>> I = ind2gray(S,map);
>> S1 = imnoise(I,'gaussian',0,0.1);
>> S2 = imnoise(I,'salt & pepper');
>> S3 = imnoise(I,'speckle');
>> figure,subplot(1,3,1),subimage(S1);
>> subplot(1,3,2),subimage(S2);
>> subplot(1,3,3),subimage(S3)
```



## **Фильтрация изображения**

Для выполнения линейной *пространственной фильтрации* используют метод двухмерной пространственной свертки локальной окрестности обрабатываемого элемента с линейным оператором, которая называется *маской* или *матрицей коэффициентов фильтра*.

*Алгоритм свертки заключается в том, что маска сканирует исходное изображение. На каждом шаге находится сумма произведений элементов маски и соответствующих элементов исходного изображения, и найденное значение присваивается одному элементу результирующего изображения. Достигнув конца строки, маска смещается на одну строку вниз, в начало строки, и процесс повторяется.*

Имеется две тесно связанные концепции, которые необходимо понимать при совершении линейной пространственной фильтрации. Первая – это корреляция, а вторая – свертка.

*Корреляция заключается в прохождении маски по изображению.*

*Свертка делается так же, но маску надо повернуть на 180° перед прохождением по изображению.*

*Если сдвигаемая маска является симметричной, то корреляция и свертка дают одинаковые результаты.*

Ниже приводятся функции, используемые для фильтрации изображения.

### **Функция вычисления двумерной свертки conv2**

#### **Синтаксис**

**D = conv2(S,h, shape)**

Функция выполняет *двумерную пространственную свертку* изображения **S** с маской **h**. Параметр **shape**, определяющий размер результирующего изображения **D**, может принимать следующие значения:

- 1) „full” – *полноразмерная свертка* (по умолчанию);
- 2) „same” – *центральная часть размера изображения S*;
- 3) „valid” – *центральная часть размера изображения S с вычетом размера маски.*

## Функция вычисления двумерной линейной фильтрации `filter2`

### Синтаксис

**D = filter2(h,S, shape)**

Результат вычисляется как *корреляция массива S двумерным фильтром*, коэффициенты которого сведены в матрицу *h*. Как правило, *S* и *D* являются полутоновыми изображениями. Значения параметра *shape* такие же, как у функции `conv2`. По умолчанию *shape* = „same“. С точки зрения выполнения процесса, свертка делается так же, но маску *h* надо повернуть на 180° перед прохождением по изображению *S*. Для этого можно использовать функцию **rot90**.

**Задание 2.** Выполнить фильтрацию палитрового изображения, используя маску

$$h = \begin{vmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{vmatrix}.$$

```
>> [x,map] = imread('c:\image\mona.bmp');
>> I = ind2gray(x,map);
>> I = im2double(I);
>> h = [1 1 1;1 -2 1;-1 -1 -1];
>> h1 = rot90(h,2);
>> I1 = filter2(h1,I);
>> figure,imshow(I)
>> figure,imshow(I1)
```

## Функция вычисления медианной фильтрации `medfilt2`

### Синтаксис

**ID = medfilt2(IS,[m,n], padopt)**

Функция **medfilt2** выполняет *нелинейную фильтрацию*, механизм выполнения которой аналогичен линейной фильтрации. На каждом шаге сканирования маски размера  $m \times n$  (по умолчанию  $3 \times 3$ ) выполняется следующая нелинейная операция: пиксели изображения, находящиеся под маской, *сортируются и составляют упорядоченную последовательность A*. Пикселю результирующего изображения **ID** (**r,c**), *присваивается значение медианы*

последовательности  $A$ , где  $r$  и  $c$  – координаты текущего положения центрального элемента маски,

Параметр **padopt** определяет три возможные опции расширения границ изображения:

- 1) опция по умолчанию **'zeros'** с нулевым расширением;
- 2) **'symmetric'**, при которой изображение  $IS$  расширяется путем его зеркального отражения через границы;
- 3) **'indexed'**, при которой  $IS$  расширяется значением 1, если  $IS$  имеет класс `double`, и значением 0 в противном случае.

**Задание 3.** Отфильтровать изображение файла `Athena.bmp`, зашумленного импульсным шумом.

```
>> [S,map] = imread('c:\Image\Mona.bmp');
>> I = ind2gray(S,map);
>> S = imnoise(I,'salt & pepper');
>> figure,imshow(S,[]);
>> D = medfilt2(S);
>> figure,imshow(D,[]);
```

### Функция вычисления двумерной АЧХ `freqz2`

#### Синтаксис

**freqz2(h,n1,n2)** – выводит на экран двумерную АЧХ.

**[H,f1,f2] = freqz2(h,n1,n2)** – формирует матрицу  $H$  размера  $n1 \times n2$ , которая является АЧХ на частотах, содержащихся в векторах **f1** и **f2**, по двумерному фильтру, коэффициенты которого сведены в матрицу **h**.

**Задание 4.** Вывод АЧХ высокочастотного фильтра двумя способами.

```
>> h = [-1,-1,-1;-1,9,-1;-1,-1,-1];
>> figure,freqz2(h);
>> title(,,'1 variant')
>> [H,f1,f2] = freqz2(h);
>> figure,mesh(f1,f2,abs(H))
>> title(,,'2 variant')
```

## Функция задания маски предопределенного фильтра **fspecial**

Синтаксис

**h=fspecial (type, P1, P2)**

Функция возвращает маску **h** предопределенного двумерного линейного фильтра, задаваемого строкой **type**. В зависимости от типа фильтра, для него могут быть определены один или два дополнительных параметра **P1, P2**. Может быть задан размер маски **n** (если **n** – вектор, то размер маски  $n(1) \times n(2)$ , если **n** – скаляр, то размер маски –  $n \times n$ ), **sigma** – среднеквадратичное отклонение распределения Гаусса, которое используется при формировании маски **h**; параметр **a**, управляющий соотношением между центральным и граничными элементами маски, устанавливается в диапазоне  $[0, 1]$ , по умолчанию равен 0.2.

Возможные варианты функции **fspecial**:

1) **h = fspecial ('gaussian', n,sigma)** возвращает маску **h** фильтра нижних частот Гаусса. По умолчанию **n** равно  $3 \times 3$ , а **sigma** равно 0,5;

2) **h = fspecial ('sobel')** возвращает маску фильтра Собела для выделения горизонтальных границ, для выделения вертикальных границ достаточно транспонировать данную маску **h**;

3) **h = fspecial ('prewitt')** возвращает маску фильтра Превитта для выделения горизонтальных границ, для выделения вертикальных границ достаточно транспонировать данную маску **h**;

4) **h = fspecial ('laplacian', a)** возвращает маску **h** ВЧ фильтр Лапласа. Размер маски  $3 \times 3$ . С помощью фильтра Лапласа можно выполнять улучшение изображения, используя формулу  $g(x, y) = f(x, y) + c \nabla^2 f(x, y)$ , где  $f(x, y)$  – это исходное изображение;  $g(x, y)$  – улучшенное изображение, а параметр **c** равен 1, если центральный коэффициент маски положителен, и  $c = -1$  в противном случае;

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}$$
 – отфильтрованное изображение

фильтром Лапласа;

5) **h = fspecial ('log',n,sigma)** возвращает маску **h** фильтра, аналогичного последовательному применению фильтров Гаусса и Лапласа, так называемого лапласиана – гауссиана, **n** и **sigma** по умолчанию устанавливается равным  $5 \times 5$  и 0.5 соответственно;

6) **h = fspecial ('average', n)** возвращает маску **h** усредняющего НЧ фильтра. По умолчанию размер маски **n** устанавливается равным  $3 \times 3$ ;

7) **h = fspecial ('unsharp', a)** возвращает маску **h** фильтра, повышающего резкость изображения. Размер маски:  $3 \times 3$ .

**Задание 5.** Улучшить изображение, используя фильтр высоких частот Лапласа.

1. Создать маску **h** фильтра высоких частот Лапласа:

```
>> h = fspecial('laplacian',0);
```

2. Выполнить высокочастотную фильтрацию изображения:

```
>> [S,map] = imread('c:\Image\Mona.bmp');
```

```
>> I = im2double(ind2gray(S,map));
```

```
>> figure,imshow(I)
```

```
>> I1 = filter2(h,I);
```

```
>> figure,imshow(I1)
```

```
>> I2 = I - I1;
```

```
>> figure,imshow(I2)
```

**Задание 6.** Отфильтровать изображение фильтром Собеля.

1. Создать маску фильтра Собеля для выделения горизонтальных границ

```
>> h = fspecial('sobel');
```

2. Создать маску фильтра Собеля для выделения вертикальных границ

```
>> h1 = h';
```

3. Отфильтровать изображение **earth.bmp** фильтрами Собеля.

```
>> [X,map] = imread('c:\Image\earth.bmp');
```

```
>> I = ind2gray(X,map);
```

```
>> figure,imshow(I)
```

```
>> F1 = filter2(h,I);
```

```
>> figure,imshow(F1)
```

```
>> F2 = filter2(h1,I);
```

```
>> figure,imshow(F2)
```

```
>> F3=F1+F2;
```

```
>> figure,imshow(F3)
```

## Функция формирования маски линейного фильтра по желаемой АЧХ fsamp2

### Синтаксис

**h = fsamp2(f1,f2,H)**

Функция формирует *маску h линейного двумерного фильтра*, основываясь на желаемой АЧХ двумерного фильтра H для частот, передаваемых в векторах f1 и f2.

**Задание 7.** Повысить резкость изображения с помощью линейного фильтра, сформированного по желаемой АЧХ.

1. Получить нормализованные значения частот:

```
>> [f1,f2] = freqspace(15,'meshgrid');
```

2. Вычислить желаемую АЧХ - H, используя так называемую метрику городских кварталов:

```
>> dist = abs(f1)+abs(f2);
```

```
>> H = dist/max(dist(:));
```

3. Вывести на экран желаемую АЧХ:

```
>> figure, mesh(f1,f2,H), colormap(cool(32));
```

4. Сформировать маску фильтра по желаемой АЧХ:

```
>> h = fsamp2(f1,f2,H,[5 5]);
```

5. Вывести на экран получившуюся АЧХ:

```
>> figure, colormap(cool(32)), freqz2(h);
```

6. Прочитать изображение и вывести его на экран:

```
>> [i,map] = imread('c:\Image\athena.bmp');
```

```
>> figure, imshow(i,map);
```

7. Отфильтровать изображение:

```
>> i1 = mat2gray(filter2(h,im2double(i)));
```

```
>> figure, imshow(i1);
```

8. Контрастирование результата фильтрации:

```
>> i2 = imadjust(i1,[0 0.5],[]);
```

```
>> figure, imshow(i2);
```

## Функция формирования маски линейного фильтра методом преобразования частот ftrans2

### Синтаксис

**h = ftrans2(b)**

Функция формирует маску  $h$  линейного двумерного фильтра, используя метод преобразования частот для трансформации одномерного фильтра с коэффициентами  $b$ . Для преобразования частот используется специальная матрица трансформации  $t$ . По умолчанию применяется матрица Мак–Клеллана:

$$t = \frac{1}{8} \begin{bmatrix} 1 & 2 & 1 \\ 2 & -4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

В функции  $h = \mathbf{ftrans2}(b,t)$  можно указать иную матрицу трансформации.

**Задание 8.** Выполнить размытие изображения с целью подавления муара с помощью линейного двумерного фильтра низкой частоты (ФНЧ), сформированного из одномерного ФНЧ методом преобразования частот:

1. Создать одномерный ФНЧ 14-го порядка с частотой среза 0.2:  
`>> b = fir1(14,0.2); % проектирование фильтров методом взвешивания`

`>> freqz(b,1,256)`

2. Сформировать двумерный фильтр из одномерного:

`>> h = ftrans2(b);`

`>> figure,freqz2(h)`

3. Прочитать исходное изображение и отфильтровать полученным фильтром:

`>> [S,map] = imread('c:\Image\Athena.bmp');`

`>> I = ind2gray(S,map);`

`>> I = im2double(I);`

`>> figure,imshow(I)`

`>> I1 = filter2(h,I);`

`>> figure,imshow(I1)`

`>> I2 = mat2gray(I1);`

`>> figure,imshow(I2)`

**Функция вычисления общей нелинейной фильтрации**  
**nlfilter**

**Синтаксис**

**D = nlfilter(S,[m n],fun,P1,P2, ...)**

Функция **D = nlfiter(S,[m n],fun,P1,P2, ...)** используется для обработки бинарных и полутоновых изображений. Формирует новое изображение, используя для фильтрации маску размером [m n] и функцию, осуществляющую фильтрацию, **fun, P1,P2** – возможные параметры, передаваемые в **fun**.

**Задание 9.** Создать функцию по фильтрации импульсного шума для использования в качестве аргумента **fun** в функции обобщенного нелинейного фильтра, который в этом случае будет использоваться для удаления импульсного шума в изображении. В функции фильтрация импульсного шума осуществляется операцией усреднения с порогом, которая состоит в том, что центральному пикселю в пределах маски присваивается среднее значение яркости всех пикселей в пределах маски в том случае, если разница между исходным значением центрального пикселя и средним больше заданного порогового значения. Для создания этой функции с именем 'AverageWithTh' в редакторе m-файлов необходимо выполнить следующие действия.

1. Создать новый m-файл командой главного меню:

File/New/m-file

2. Ввести команды функции:

```
function R = AverageWithTh(x,Th); % заголовочная строка
```

```
[r c] = size(x);
```

```
n = r*c;
```

```
r = floor((r+1)/2); c = floor((c+1)/2);
```

```
s = sum(x(:))/n;
```

```
if (abs(x(r,c)-s))>Th
```

```
R = s;
```

```
else
```

```
R = x(r,c);
```

```
end;
```

```
end;
```

3. Создать новый подкаталог.

4. Сохранить в новый подкаталог созданную функцию в файл с одноименным названием.

5. Прочитать изображение, наложить на изображение импульсный шум и отфильтровать его, используя созданную функцию:

```
>> [X,map] = imread('c:\Image\Mona.bmp');
```

```
>> I = im2double(ind2gray(X,map));
```

```
>> figure,imshow(I)
```



```
>> I1 = imnoise(I,'salt & pepper');  
>> figure,imshow(I1)  
>> I2 = nlfiter(I1,[3 3],'AverageWithTh',0.2);  
>> figure,imshow(I2)
```

### 3. ПРИМЕРЫ ВЫПОЛНЕНИЯ ЗАДАНИЙ

Задание 1:

Прочитать палитровое изображение из файла, вывести на экран,

```
>> Image = imread('C:\Images\Blagai','jpg');  
>> [X map] = rgb2ind(Image,255);  
>> figure,subimage(Image),title('Исходноеизображение');
```

Рисунок 3.1 – Команда для считывания исходного изображения

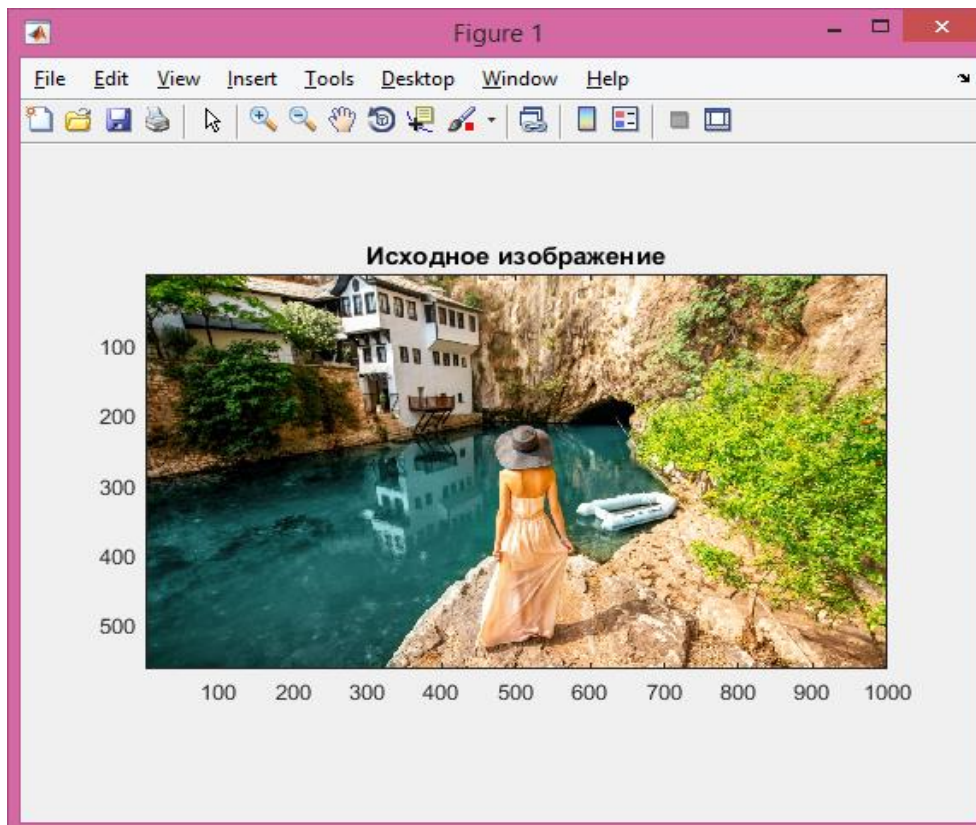


Рисунок 3.2 –Исходное изображение

Преобразовать изображение в полутоновое, наложить на него гауссов, импульсный и мультипликативный шум. Вывести на экран зашумленные изображения.

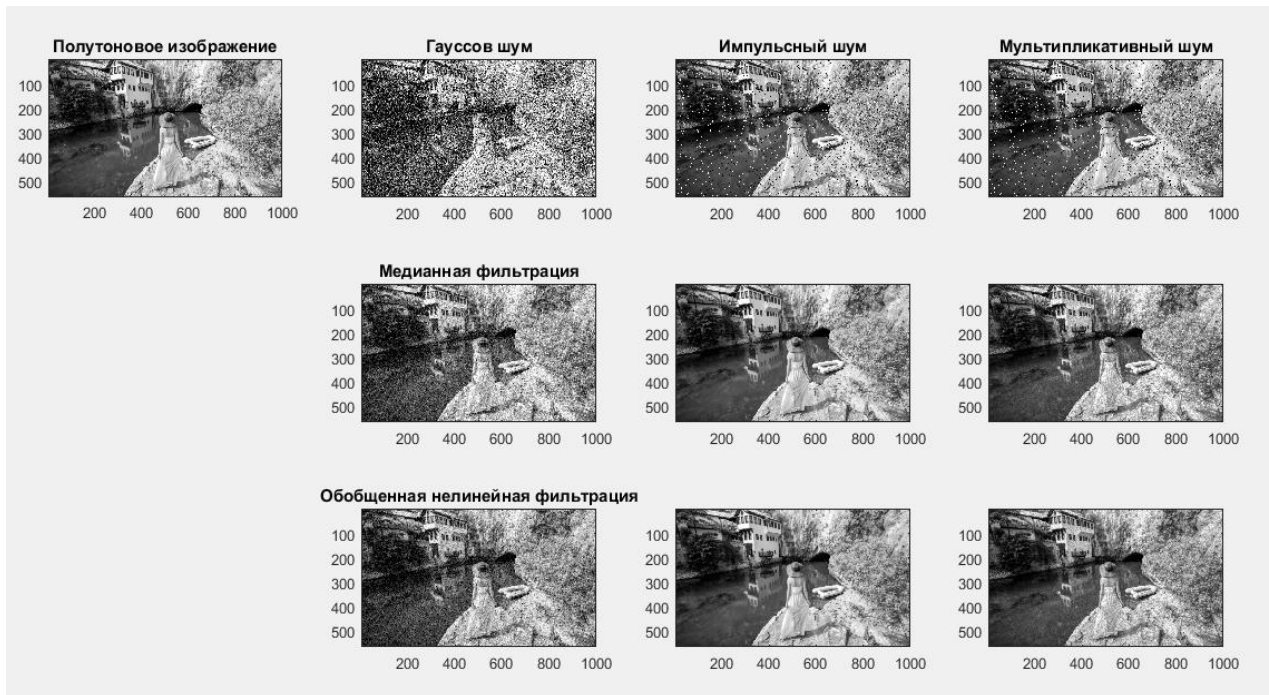


Рисунок 3.3 –Результат наложения на исходное изображение гауссового, импульсного и мультипликативного шумов и использования медианной и обобщенной нелинейной фильтрации

2. Выполнить фильтрацию изображения из файла Bigbird.bmp:

a) Отфильтровать изображение с помощью масок кругового градиента:

север	северо-восток	восток	юго-восток
1 1 1	1 1 1	-1 1 1	-1 1 1
1 -2 1	-1 -2 1	-1 -2 1	-1 -2 1
-1 -1 -1	-1 -1 1	-1 1 1	1 1 1
юг	юго-запад	запад	северо-запад
-1 -1 -1	1 -1 -1	1 1 -1	1 1 -1
1 -2 1	1 -2 1	1 -2 -1	1 -2 -1
1 1 1	1 1 1	1 1 -1	1 1 -1

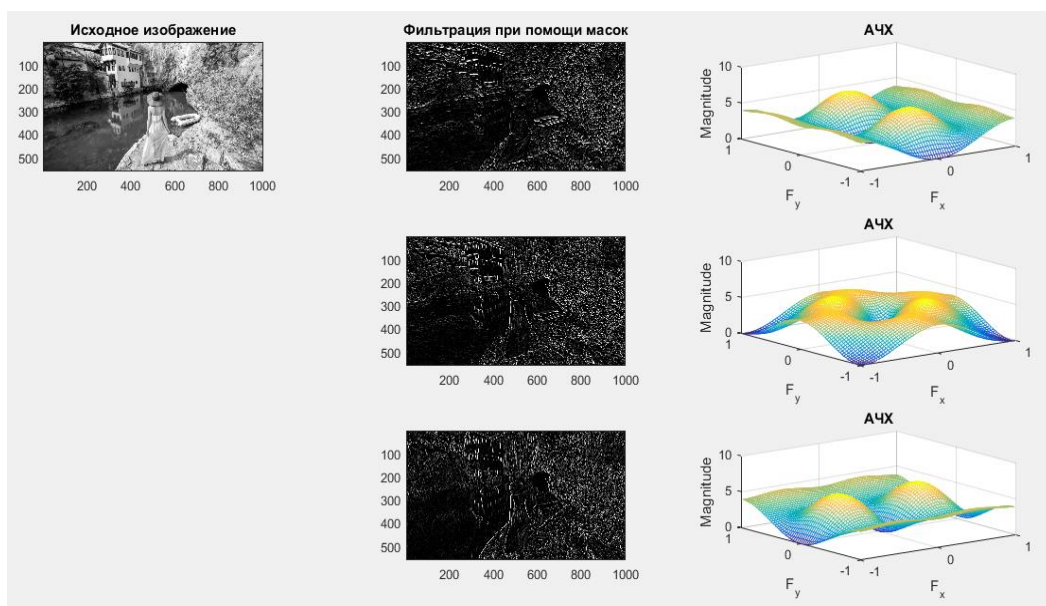


Рисунок 3.4 – Результат фильтрации исходного изображения с помощью масок кругового градиента

б) Улучшить изображение с помощью масок фильтров Лапласа вида:

$$\begin{array}{ccc}
 0 & 1 & 0 \\
 1 & -4 & 1 \\
 0 & 1 & 0
 \end{array}
 \quad
 \begin{array}{ccc}
 -1 & -1 & -1 \\
 -1 & 8 & -1 \\
 -1 & -1 & -1
 \end{array}
 \quad
 \begin{array}{ccc}
 1 & -2 & 1 \\
 -2 & 4 & -2 \\
 1 & -2 & 1
 \end{array}$$

Ввод масок и вывод результатов выполнить в цикле.

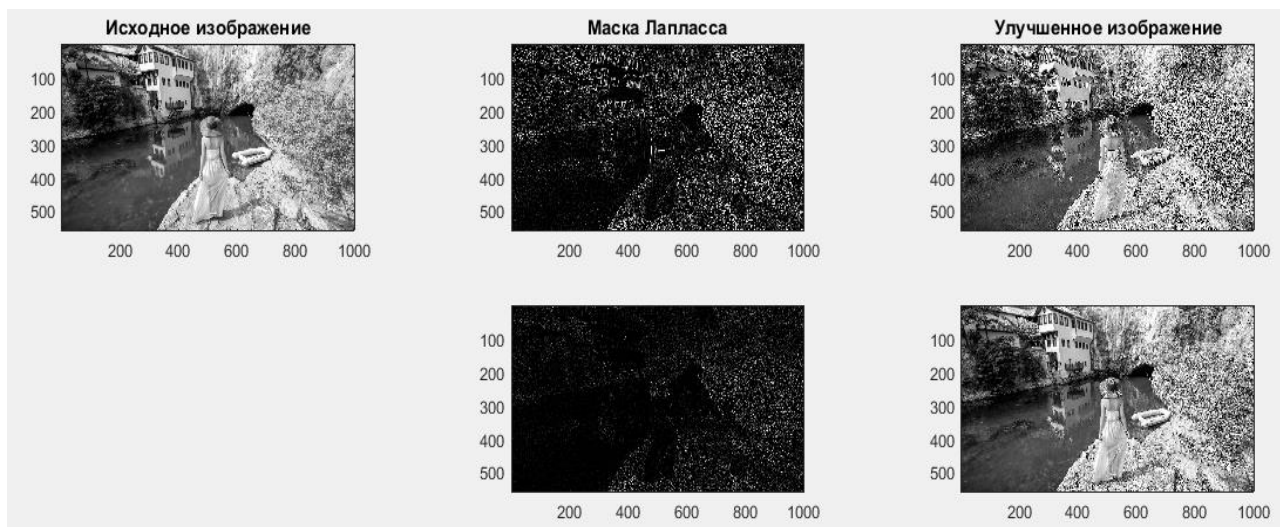


Рисунок 3. 5 – Результат улучшения исходного изображения с помощью масок фильтров Лапласа

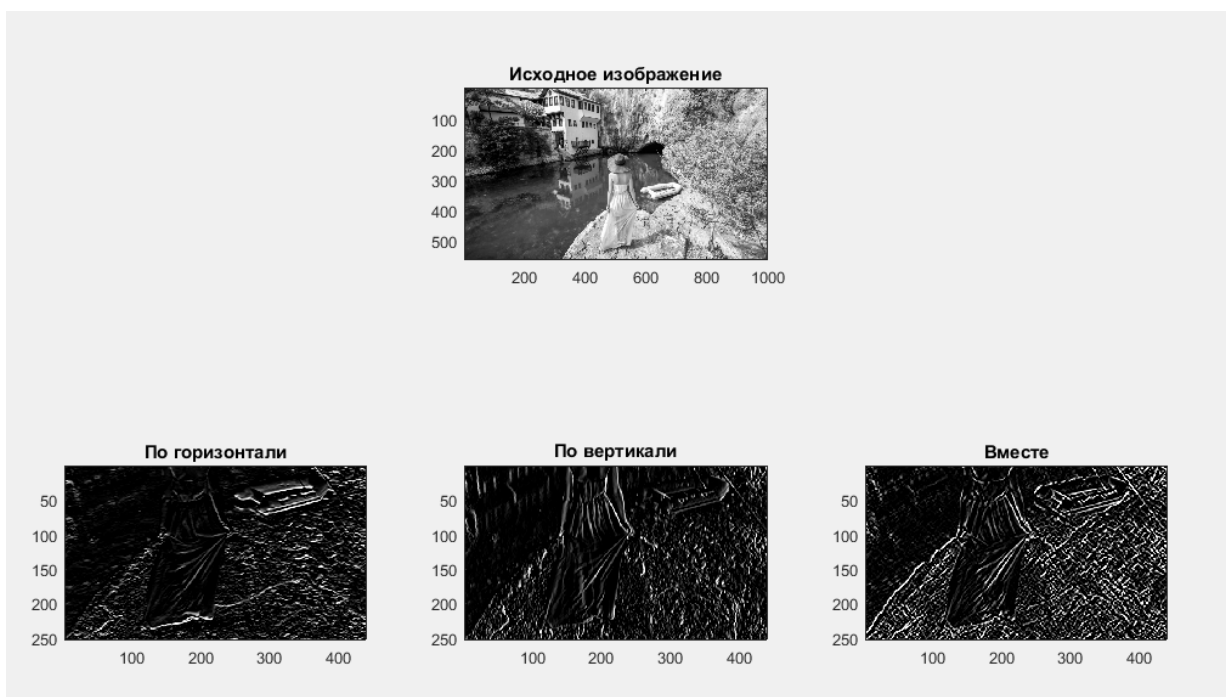


Рисунок 3.6 –Результат фильтрации исходного полутонового фрагмента маской фильтра Превитт

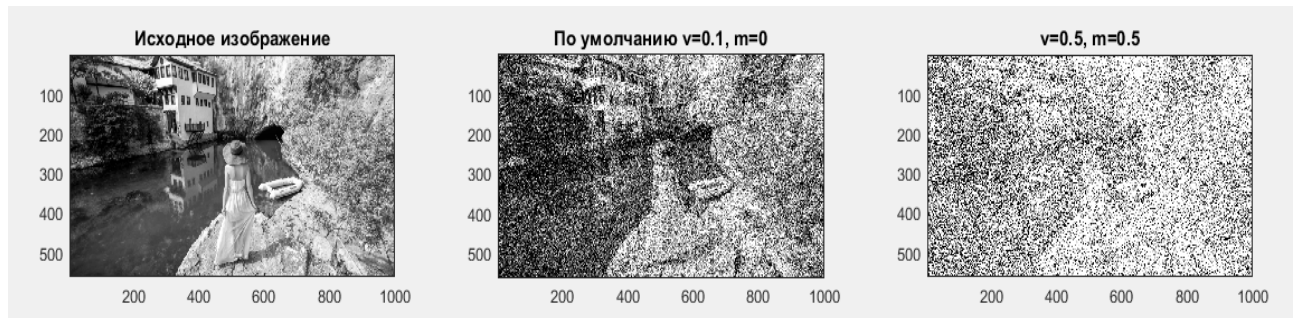


Рисунок 3.7 – Результат наложения шума на полутоновое изображение гауссовым шумом с разными дисперсиями

#### 4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие типы фильтров создает функция по формированию масок фильтров **fspecial**?
2. В чем заключается алгоритм двумерной свертки?
3. В каких функциях присутствует алгоритм двумерной свертки?
4. В чем отличие алгоритма медианной фильтрации от алгоритма фильтрации с помощью операции усреднения с порогом?
5. Какие типы шумов формирует функция по зашумлению изображений **imnoise**?
6. Для каких целей можно использовать функцию `freqz2`?
7. Каким образом можно сформировать маску линейного фильтра по желаемой АЧХ?
8. Какая функция позволяет сформировать двумерный фильтр из одномерного?
9. Какая функция добавляет к изображению **Is** *гауссов белый шум* с математическим ожиданием **m** и дисперсией **v**.
10. Какая функция добавляет к изображению **Is** мультипликативный шум?

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Томакова, Р. А. Методы и алгоритмы цифровой обработки изображений : учебное пособие для студентов всех форм обучения по направлениям подготовки 09.03.03 Прикладная информатика; 09.03.04 «Программная инженерия», 09.04.04 «Программная инженерия» (профиль «Разработка информационно-вычислительных систем») / Р. А. Томакова, Е. А. Петрик ; Юго-Зап. гос. ун-т. - Курск : Университетская книга, 2020. - 310 с. - Загл. с титул. экрана. – Текст : электронный.

3. Яне, Бернд. Цифровая обработка изображений : [учебное пособие] / пер. с англ. А. М. Измайловой. - М. : Техносфера, 2007. - 584 с. : ил. + 1 эл. опт. диск (CD-ROM). - (Мир цифровой обработки. XI. 06). - Библиогр.: с. 575-583 (221 назв.). - ISBN 978-5-94836-1 22-2 : 285.00 р. - Текст : непосредственный.

4. Красильников, Н. Н. Цифровая обработка 2D- и 3D-изображений : учебное пособие / Н. Н. Красильников. - СПб. : БХВ-Петербург, 2011. - 608 с. : ил. - ISBN 978-5-9775-07 00-4 : 372.50 р. - Текст : непосредственный.

5. Томакова, Римма Александровна. Интеллектуальные технологии сегментации и классификации биомедицинских изображений : монография / Р. А. Томакова, С. Г. Емельянов, С. А. Филист ; Юго-Западный государственный университет. - Курск : ЮЗГУ, 2012. - 222 с. - Текст : электронный.

6. Томакова, Римма Александровна . Теоретические основы и методы обработки и анализа микроскопических изображений биоматериалов : монография / Р. А. Томакова, С. А. Филист, С. Г. Емельянов ; МИНОБРНАУКИ РФ, Юго-Западный государственный университет. - Курск : ЮЗГУ, 2011. - 202 с. - Текст : электронный.

7. Методы цифровой обработки изображений : учебное пособие / А. Е. Архипов, С. В. Дегтярев, С. С. Садыков, С. Н. Середа, В. С. Титов. - Курск : КГТУ, 2002 - Ч. 2. - 115 с. - Текст : непосредственный.

9. Технологии сетей связи. Особенности кодирования цифровых факсимильных сообщений : учебное пособие / А. И. Атакищев [и др.] ; Министерство образования Российской Федерации, Курский государственный технический университет. - Курск : КурскГТУ,



2002. - 159 с. :ил.табл. - Имеется электрон. аналог. - ISBN 5-7681-0111-X : 62.00 р. - Текст : непосредственный.

10. Гонсалес, Р. Цифровая обработка изображений: практические советы : монография / Р. Гонсалес, Р. Вудс ; пер. П. А. Чочиа, Л. И. Рубанова. – 3-е изд., испр. и доп. – Москва : Техносфера, 2012. – 1104 с. :– (Мир цифровой обработки).– URL: <https://biblioclub.ru/index.php?page=book&id=233465> (дата обращения: 23.03.2022).

11. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. - М. : Техносфера, 2006. - 1072 с. - (Мир цифровой обработки). - ISBN 5-94836-028-8 : 394.66 р. - Текст : непосредственный.

12. Фисенко, В.Т. Компьютерная обработка и распознавание изображений / В.Т. Фисенко, Т.Ю. Фисенко. – СПб.: СПбГУ ИТМО, 2008. – 192 с.

13. Гонсалес, Р. Цифровая обработка изображений в среде Matlab / Р. Гонсалес, Р. Вудс, С. Эддинс. – М.: Техносфера, 2006. – 616 с.