

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ**

**ЗБІРНИК МЕТОДИЧНИХ ВКАЗІВОК**

до лабораторних робіт з дисципліни

**ЦИФРОВА ОБРОБКА СУПУТНИКОВИХ ЗОБРАЖЕНЬ**

**Одеса – 2019**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

**ЗБІРНИК МЕТОДИЧНИХ ВКАЗІВОК**

до лабораторних робіт з дисципліни

**ЦИФРОВА ОБРОБКА СУПУТНИКОВИХ ЗОБРАЖЕНЬ**

Узгоджено  
на факультеті магістерської  
підготовки

Одеса – 2019

Збірник методичних вказівок до лабораторних робіт з дисципліни «Цифрова обробка супутникових зображень» для магістрів I курсу денної форми навчання за спеціальністю - 122 "Комп'ютерні науки та інформаційні технології", спеціалізацією "Комп'ютерний еколого-економічний моніторинг". / Гор'єв С.А., – Одеса, ОДЕКУ, 2019 р. – 88 с

Збірник методичних вказівок до лабораторних робіт з дисципліни «Цифрова обробка супутникових зображень» для магістрів I курсу денної форми навчання за спеціальністю - 122 "Компютерні науки та інформаційні технології", спеціалізацією "Компютерний еколого-економічний моніторинг". / Гор'єв С.А., – Одеса, ОДЕКУ, 2019 р. – 88 с

Підп. до друку  
Умовн. друк. арк.

Формат  
Тираж

Папір  
Зам. №

Надруковано з готового оригінал-макета

---

Одеський державний екологічний університет  
65016 Одеса, вул. Львівська, 15

---

## ЗМІСТ

Передмова .....	3
Лабораторна робота №1 «Основи роботи з зображенням в MatLab. Найпростіші операції з зображеннями».....	
Лабораторна робота №2 «Основи методів просторової обробки зображень. Перетворення яскравості і контрасту».....	34
Лабораторна робота № 3 «Просторова та частотна фільтрація зображень»	

*При изучении наук примеры полезнее правил.  
Исаак Ньютон*

## ПЕРЕДМОВА

Дисципліна “ Цифрова обробка супутникових зображень ” є вибірковою дисципліною підготовки магістрів за спеціальністю 122 Комп’ютерні науки і за освітньо-професійною програмою комп’ютерний еколого-економічний моніторинг відноситься до циклу професійної та практичної підготовки.

Мета дисципліни – підготовка майбутніх фахівців в галузі цифрової обробки зображень інформаційно-вимірювальних систем моніторингу навколишнього середовища.

Завдання дисципліни полягає у вивченні теоретичних, методичних основ та практичного застосування методів і засобів цифрової обробки супутникових зображень.

В результаті вивчення дисципліни студент повинен:

*Знати:*

- основних визначень і понять комп’ютерної обробки зображень та математичного апарата опису сигналів зображень і цифрових фільтрів,
- методів і способів цифрової обробки супутникових зображень заснованих на фільтрації, відновленні та геометричних перетвореннях.
- основних фізичних принципів одержання аерокосмічної інформації про навколишнє середовище

*Вміти:*

- застосовувати існуючі засоби цифрової обробки супутникових зображень,
- розробляти нові алгоритми цифрової обробки супутникових зображень,
- використовувати основні методи аналізу й обробки супутникових зображень.

Вивчення даної дисципліни базується на знаннях, які одержані з дисциплін “ Фізика ”, “ Вища математика ”, “ Цифрова обробка і передача інформації ”, “ Супутниковий моніторинг ”.

Мета методичних вказівок – надати студентам допомогу у вирішенні практичних задач при проектування та розробка інформаційних систем метеорологічних вимірювань.

У даних методичних вказівках приводяться теоретичні відомості, необхідні для виконання лабораторних робіт, а також мета, завдання і

порядок виконання кожної роботи. Приведені також вимоги до оформлення звіту з кожної лабораторної роботи.

При виконанні лабораторної роботи кожен студент відповідає на теоретичні питання і, потім, після одержання допуску, практично виконує роботу.

Під час проведення лабораторних робіт проводиться модульний контроль, який складається з одного практичного модуля. Максимальна оцінка ЗМ-П1 – 50 балів.

Критерії оцінювання в балах практичних модулів:

Визначення	Бали ЗМ-П1
відмінне виконання лише з незначною кількістю помилок	23-25
вище середнього рівня з кількома помилками	21-22
в загальному правильна робота з певною кількістю грубих помилок	19-20
непогано, але зі значною кількістю помилок	17-18
виконання задовольняє мінімальні критерії	6-16
виконання не задовольняє мінімальні критерії	1-5

Збірник методичних вказівок складається з 3 практичних робіт.

### ПЕРЕЛІК ПОСИЛАНЬ

- 1.Перелигін Б.В. Супутниковий моніторинг: Конспект лекцій – Одеса: Екологія, 2008. – 130 с.
- 2.Перелигін Б.В. Одержання, передача, прийом і надання локаційної космічної інформації: Навчальний посібник – Одеса: Екологія, 2006. – 88 с.
- 3.Лімонов О.С. Цифрова обробка і передача інформації: Конспект лекцій. - Одеса: ОДЕКУ, 2013. – 116 с.
- 4.Лімонов О.С. Цифрова обробка сигналів: Конспект лекцій – Одеса: ОДЕКУ, 2011. – 122 с.
- 5.Дьяконов В.П. MATLAB 6.5 SP1/7 + Simulink 5/6. Обработка сигналов и проектирование фильтров – М.: СОЛОН-ПРЕСС, 2009. – 577 с.
- 6.Гонсалес, Р. Цифровая обработка изображений в среде MATLAB /Р. Гонсалес, Р. Вудс, С. Эддинс. – М.: Техносфера, 2006. – 616 с.
- 7.Гонсалес, Р. Цифровая обработка изображений – М.: Техносфера, 2012. – 1104 с.
- 8.Цифровая обработка изображений в информационных системах /И. С. Грузман, В. С. Киричук [и др.]. – Новосибирск: Изд-во НГТУ, 2002. – 352 с.

## Лабораторна робота №1

### Основи роботи з зображенням в MatLab. Найпростіші операції з зображенням»

**Мета роботи:** вивчення можливостей пакета Image Processing Toolbox. Вивчення основних прикладів роботи з зображеннями. Отримання практичних навичок у складанні програм обробки зображень.

**Завдання на підготовку до лабораторної роботи.** Під час проведення лабораторної роботи студент повинен:

**знати:**

- основні характеристики детермінованих та стохастичних сигналів;
- цифрові методи обробки сигналів;
- будову і роботу програмного пакета MATLAB;

**уміти:**

- використовувати ручне введення та конструювання інструментальних засобів пакета MATLAB;
- використовувати можливості пакета Signal Processing Toolbox;
- досліджувати та вимірювати детерміновані та ймовірнісні характеристики сигналів;
- зберігати та візуалізувати інформацію в програмному пакеті MATLAB.

### Теоретичні відомості.

Пакет Image Processing Toolbox (IPT) являє собою набір функцій для розширення можливостей системи MATLAB при роботі із цифровими зображеннями. Даний пакет підтримує різні операції для обробки зображень, такі як:

- 1) просторове перетворення;
- 2) нелінійна й лінійна фільтрація;
- 3) аналіз і поліпшення зображень;
- 4) відновлення зображень;
- 5) стиск зображень.

#### Подання цифрових зображень.

Монохромне (чорно-біле) зображення (Grayscale) можна визначити як двовимірну функцію  $f(x, y)$ , де  $x$  й  $y$  представляють просторові координати, а амплітуда для кожної пари координат  $f(x, y)$  є інтенсивністю або яскравістю зображення в крапці із заданими координатами. Кольорові зображення формуються комбінацією декількох монохромних. Наприклад, у



системі RGB кольорове зображення складається з комбінацій трьох монохромних компонентів R - червоної, G - зеленої й B - синьої.

В аналогових зображеннях, наприклад, зображеннях на фотографіях, або на картинах, намальованих художником, кожен крапку можна характеризувати безперервними просторовими координатами  $(x, y)$  і відповідною інтенсивністю або кольорами.

Перетворення аналогового зображення в цифрове вимагає подання координат й інтенсивностей деякими дискретними значеннями. Для оцифрування виконуються операції:

- просторова дискретизація координат, при якій безперервні координати замінюються кінцевою безліччю відліків;
- квантування по амплітуді, при якому інтенсивності в кожній отриманій крапці замінюються квантованими значеннями (округленими до деяких еталонних).

Результатом дискретизації й квантування є матриця чисел, що складає з елементів зображення (пікселів) і що має  $M$  рядків й  $N$  стовпців. Це значить, що зображення має розмір  $M \times N$ .

Цифрові зображення, завантажені в середовище MATLAB, містяться в прямокутну систему координат, де дискретні координати  $(x, y)$  замінюються на  $(r, c)$  для позначення рядків (row) і стовпців (column). Початок цієї системи відповідає верхньому лівому куту зображення, координати якої  $(r, c) = (1, 1)$ . При цьому значення координат можуть мінятися в такий спосіб  $r = 1, 2, \dots, M; c = 1, 2, \dots, N$ .

Математично таке цифрове зображення описується матрицею, вид якої наведений нижче.

$$f(r,c) = \begin{bmatrix} f(1,1) & f(1,2) & \dots & f(1,N) \\ f(2,1) & f(2,2) & \dots & f(2,N) \\ \vdots & \vdots & \dots & \vdots \\ f(M,1) & f(M,2) & \dots & f(M,N) \end{bmatrix}$$

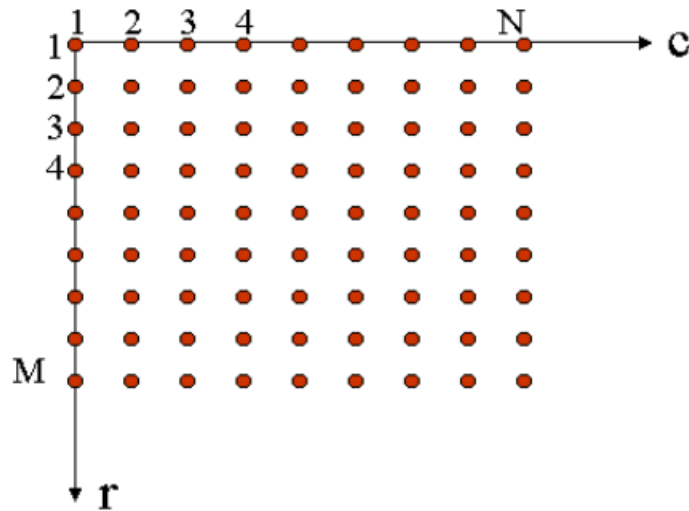


Рисунок 1.1 - Координатна угода в системі MATLAB

Координати пікселів зображень є цілими числами, однак значення пікселів (інтенсивність, яскравість), незважаючи на те, що вони проквантовані, можуть приймати дискретні значення в різних діапазонах числових значень. Це визначається *класом даних*, що використовується для подання зображень.

**Використовувані в MATLAB класи даних наведені нижче:**

- uint8*** Цілі числа без знака в інтервалі  $[0, 255]$  (1 байт на число);
- uint16*** Цілі числа без знака в інтервалі  $[0, 65\ 535]$  (2 байти на число);
- uint32*** Цілі числа без знака в інтервалі  $[0, 4\ 294\ 967\ 295]$  (4 байти на число);
- int8*** Цілі числа зі знаком в інтервалі  $[-128, 127]$  (1 байт на число);
- int16*** Цілі числа зі знаком в інтервалі  $[-32\ 768, 32\ 767]$  (2 байти на число);
- single*** Речовинні числа із плаваючою комою одинарної точності в діапазоні, приблизно, від  $-10^{38}$  до  $10^{38}$  (4 байти на число);
- double*** Речовинні числа із плаваючою комою подвійної точності в діапазоні, приблизно, від  $-10^{308}$  до  $10^{308}$ ;
- logical*** Значення логічного 0 або 1 (1 байт на елемент);

*Типи зображень*

Пакет розширень IPT може працювати із зображеннями наступних типів:

- ***напівтонові*** (чорно-білі, монохромні, ***grayscale***) зображення. Це матриця, елементи якої представлені числовими значеннями.

Якщо елементи цього зображення належать класу ***uint8*** або ***uint16***, то вони представляються цілими числами в інтервалах  $[0, 255]$  й  $[0, 65\ 535]$  відповідно.

Якщо зображення використовує клас **double**, то інтенсивність пікселів представляється речовинними числами із плаваючою крапкою подвійної точності. При цьому домовлено, що інтенсивність пікселів лежить у діапазоні  $[0, 1]$ .

– **двійкові** (бінарні, *bitmap*) зображення є логічними масивами, що складаються з  $1$  й  $0$ . Масив елементів з  $1$  й  $0$  інших класів, наприклад, **uint8**, не є двійковим зображенням. Для перетворення числових масивів у логічні використовується функція **logical**:

$$G = \text{logical}(A),$$

де  $A$  - числовий масив, що складається з  $1$  й  $0$ ,  $G$  - логічний масив з тими ж елементами.

Якщо масив  $A$  має елементи відмінні від  $0$  й  $1$ , то ця функція перетворить всі його відмінні від  $0$  елементи в логічні  $1$ , а всі нульові в логічний  $0$ .

### Конвертування класів даних і типів зображень

Перетворення (конвертування) зображень із одних класів і типів в інші класи й типи є найпоширенішою дією в IPT. Застосовуючи конвертування типів даних, варто пам'ятати про діапазони значень величин кожного класу, наведених вище.

Наприклад, нехай  $A$  — масив класу **uint8**. Масив подвійної точності  $B$  генерується командою

$$B = \text{double}(A).$$

Таке перетворення використовується досить часто, оскільки MATLAB припускає, що операнди числових операцій є речовинними числами з подвійною точністю.

Якщо  $C$  – це масив класу **double**, елементи якого лежать в інтервалі  $[0, 255]$ , але серед них можуть зустрічатися дробові числа, то для їхнього виключення його можна перетворити в масив **uint8** у такий спосіб:

$$D = \text{uint8}(C).$$

Якщо масив класу **double** мав елементи зі значеннями поза інтервалом  $[0, 255]$  і він був конвертований у клас **uint8** описаним вище способом, то MATLAB перетворить всі негативні величини (менші  $0$ ) в  $0$ , всі величини, більші  $255$  — в  $255$ , а у всіх інших елементів *відкидаються дробові частини*.

Це означає, що перед перетворенням масивів **double** в **uint8** необхідно зробити підходящу перенормування (перемасштабування) його елементів.

Перетворення будь-яких числових даних у логічні приводить до логічного масиву, у якому коштують логічні  $1$  скрізь на місці ненульових елементів вхідного масиву й  $0$  на місці тих елементів, які дорівнюють нулю.

Уточнити інформацію про розміри масиву й клас зображення  $f$  можна за допомогою команди **whos f**:

```
>> whos f
```

Name	Size	Bytes	Class
f	3x3	72	double array

В IPT є спеціальні функції, які реалізують *перенормування* (перемасштабування) при конвертуванні одних класів і типів зображень в інші. Функція **im2uint8** спочатку розпізнає клас даних на вході й робить всі необхідні перетворення, щоб вихідне зображення мало правильний тип даних. Як приклад розглянемо зображення *f* розміру **2x2** класи **double**, що може бути результатом деяких проміжних обчислень:

```
f =  
-0.5  0.5  
 0.75 1.5
```

Виконавши перетворення

```
>> g = im2uint8(f)
```

одержимо результат

```
g =  
128 128  
191 255
```

Із приклада видно, що функція **im2uint8** обнуляє всі негативні значення вхідного зображення, ставить число **255** на місце величин, більших **1**, і множить інші значення на **255**, після чого округляє результат до *найближчого цілого числа*.

Перетворення довільних масивів **double** у перенормовані масиви (зображення) **double** зі значеннями в інтервалі **[0, 1]** виконується за допомогою функції **mat2gray**, що має наступний синтаксис:

$$g = \text{mat2gray}(A, [A_{\min}, A_{\max}]),$$

де зображення *g* має значення пікселів в інтервалі від 0 (чорний) до 1 (білий). Це відбувається в такий спосіб:

- всі елементи, менші або рівні *A<sub>min</sub>*, обнуляє;
- всі елементи, більш або рівні *A<sub>max</sub>*, замінюються на **1.0**.
- інші елементи відображаються пропорційними значеннями в інтервалі **[0, 1.0]**.

```
f =  
 1  2  
 3  4
```

```
>> g=mat2gray(f, [1, 4])
```

```
g =  
    0    0.3333  
0.6667    1.0000
```

При виконанні команди без параметрів [*Amin*, *Amax*]

***g* = mat2gray(*A*),**

створюється новий масив класу **double** у якому значення *Amin* й *Amax* – це дійсні максимум і мінімум масиву *A*.

```
f1 =  
    0    3    2  
    6    0    2  
    5    0    4
```

```
>> g1=mat2gray(f1)
```

```
g1 =  
    0    0.5000    0.3333  
    1.0    0    0.3333  
    0.8333    0    0.6667
```

Функція **im2double** перетворить вхідний масив класів **logical**, **uint8** або **uint16** у клас **double** з діапазоном **[0, 1]**. Якщо вхідний масив був класу **double**, то функція **im2double** залишає його без змін.

```
>> f2=uint8([15 3 25; 66 0 200; 51 40 49])
```

```
f2 =  
    15    3    25  
    66    0   200  
    51   40    49
```

```
>> g2=im2double(f2)
```

```
g2 =  
    0.0588    0.0118    0.0980  
    0.2588    0    0.7843  
    0.2000    0.1569    0.1922
```

Із приклада видно, що вхідний масив класу **uint8** перемасштабується розподілом кожного його елемента на **255**. У випадку перемасштабування масиву класу **uint16** кожен елемент буде ділитися на **65535**.

Для одержання двійкових зображень із напівтонових в IPT використовується функція **im2bw**, що має синтаксис

$g = \text{im2bw}(f, T)$ .

Результатом буде двійкове зображення  $g$ , отримане з напівтонового зображення  $f$ , перетворене по порозі  $T$ . Значення всіх елементів  $f$ , менших  $T$ , стають логічними  $0$ , а всі інші – логічними  $1$ . Значення порога  $T$  повинне перебувати в інтервалі  $[0, 1]$  незалежно від класу вхідного зображення. Вихідний масив автоматично буде логічним. Якщо поріг не вказувати

$g = \text{im2bw}(f)$ ,

те за замовчуванням він приймається  $T = 0.5$ .

Якщо вхідне зображення було класу *uint8*, то *im2bw* спочатку ділить його елементи на **255**, а потім застосовує заданий поріг або поріг, прийнятий за замовчуванням.

Логічний (двійковий) масив можна перетворити в числовий за допомогою кожної із чотирьох функцій:

- *im2uint8*;
- *im2uint16*;
- *mat2gray*;
- *im2double*.

### Основні прийоми роботи із зображеннями

Однією з найчастіших операцій по роботі із зображеннями є їхнє зчитування із графічних файлів з метою завантаження в робочий простір MATLAB. Ця операція реалізується функцією *imread*:

$A = \text{imread}('filename')$ .

Тут *filename* – це рядок символів, що утворюють повне ім'я файлу зображення, що завантажує, (включаючи будь-яке розширення).

Для випадку, коли графічний файл включений у додаток Image Processing Toolbox, досить вказати тільки ім'я файлу і його розширення, наприклад:

$A = \text{imread}('image.bmp');$

При цьому матричній змінній  $A$  привласнюється значення матриці, що відповідає зображенню, що зберігається у файлі з ім'ям *image.bmp*. Крапка з коми наприкінці командного рядка забороняє (придушує) вивід результату. Для даної команди забороняється вивід на екран значень масиву  $A$ .

У випадку, коли зображення розташоване в конкретній папці, необхідно в явному виді вказати повний шлях до цієї директорії, наприклад:

$A = \text{imread}('D:\Image\image.jpeg');$

Таблиці 1 наведені основні формати графічних файлів, які можуть бути прочитані функцією *imread* (або записані функцією *imwrite*).

Таблиця 1 - Основні формати графічних файлів, підтримувані MATLAB

Формат зображення	Розшифровка скорочення	Припустимі скорочення	Глибина кольору (біт)
TIFF	Tag Image File Format	.tif, .tiff	1, 8, 24
JPEG	Joint Photographic Experts Group	.jpg, .jpeg	.8, 24
GIF	Graphics Interchange Format	.gif	
BMP	Windows Bitmap	.bmp	1, 4, 8, 24
PNG	Portable Network Graphics	.png	1, 8, 24

Для виводу на дисплей зображення, що перебуває в робочому просторі, використовується функція *imshow*, що має наступний синтаксис:

*imshow (A, G),*

де *A* - матриця виведеного зображення, *G* - число градацій яскравості, використовуване для відображення зображення *A*. Якщо *G* відсутній, за замовчуванням використовується 256 градацій яскравості.

Команда *imshow(f, [low high])*

дозволяє вивести зображення в якому всі пікселі з інтенсивністю не більше числа *low* відображаються чорними, а пікселі з інтенсивністю не менше числа *high* відображаються білими. Якщо значення [*low high*] не задаються, а квадратні дужки залишаються порожніми, то пікселі з мінімальною інтенсивністю відображаються чорними, а пікселі з максимальною інтенсивністю - білими, тобто вивід у графічне вікно виробляється з максимальним контрастом.

Наприклад

```
>> f = imread('F:/Images/boat.jpg');
```

```
figure, imshow(f)
```

Результатом виконання останньої команди буде вивід на екран вікна із завантаженим зображенням.



Рисунок 1.2 – Вікно виводу зображення на екран

Інша функція додатка **imtool** служить для запуску інструмента Image Tool, що являє собою інтегроване середовище для візуалізації зображень і виконання деяких операцій обробки й аналізу зображень. Інструментарій Image Tool забезпечує відображення зображень всіх типів, які підтримуються системою MATLAB, а також доступ до інших засобів навігації й аналізу зображень, наприклад, до Pixel Region tool, Image Information tool, Adjust Contrast tool й ін.

Для збереження зображення у файл використовується функція **imwrite**, що має формат:

***imwrite (A, 'filename' ),***

де  $A$  – матриця зображення, **'filename'** – ім'я файлу, що зберігає, із вказівкою розширення, підтримуваного системою MATLAB. Якщо **'filename'** не містить інформацію про шляху до папки, то збереження виробляється в поточну папку.

Найчастіше виникає необхідність збереження зображення на диску точно в тій формі, у якому воно відображено в графічному вікні на екрані. Уміст вікна зображень можна експортувати на диск за допомогою команди **Export** у спадаючому меню **File** у вікні зображень.

Для того, щоб одержати інформацію про розмір зображень, необхідно використати функцію **size (A)**, що повертає розмір зображення  $A$  в виді вектора із двох елементів  $M$  й  $N$ . Приклад:

```
>> size (A)
ans =
    500    500
```



Для того, щоб одержувати значення розмірів використати для подальших обчислень варто застосовувати функцію **size** у наступному виді:

```
>> [M N] = size (A);
```

### Подання зображень у цифровій формі.

Для здійснення цифрової обробки зображень необхідно виконати перетворення безперервного (аналогового) сигналу зображень у цифровий масив. Таке перетворення включає виконання двох перетворень. Перше перетворення представляє заміну реального безперервного зображення набором відліків у дискретні моменти часу, таке перетворення називається *дискретизацією*. Друге - це перетворення безперервної безлічі значень сигналу зображення в безліч квантованих значень, таке перетворення називається *квантуванням*.

### *Дискретизація зображень*

Просторова дискретизація зображення припускає формування безперервної функції в дискретних відліках (отчетах) просторових координат. Нехай функція  $f_i(x,y)$  описує вхідне безперервне зображення нескінченних розмірів. В ідеальній системі дискретизації зображення просторові відліки вихідного зображення виходять шляхом перемноження цієї функції із просторовою що дискретизує функцією, що складається з нескінченного числа  $\delta$ - функцій Дирака, заданих у вузлах ґрат із кроком  $(\Delta x, \Delta y)$ :

$$s(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(x - n\Delta x, y - m\Delta y),$$

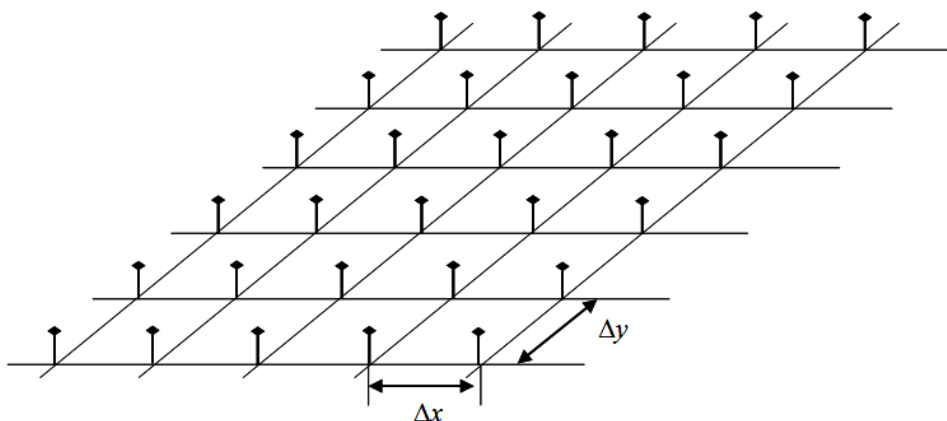


Рисунок 1. 3 - Функція просторової дискретизації зображення.

Помножимо безперервне зображення на функцію просторової дискретизації :

$$f(x,y) = f_i(x,y) * s(x,y)$$

Відповідно до дискретизуючої функції одержуємо:

$$f(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f_I(n\Delta x, m\Delta y) \delta(x - n\Delta x, y - m\Delta y).$$

*Дискретизація* – це перетворення безперервного сигналу в послідовність чисел (відліків), тобто подання цього сигналу по якому-небудь скінченномірному базисі. Найбільш зручним з погляду організації обробки й природним способом дискретизації є подання сигналів у вигляді вибірки їхніх значень (відліків) в окремих, регулярно розташованих крапках. Такий спосіб називають *растрированием*, а послідовність вузлів, у яких беруться відліки – *растром*. Інтервал, через який беруться значення безперервного сигналу, називається *кроком дискретизації*. Зворотна кроку величина називається *частотою дискретизації*,

Таким чином, дискретизація зображення по просторовим змінним переводить його в таблицю вибірових значень. Розмірність таблиці (число рядків і стовпців) визначається геометричними розмірами вихідної прямокутної області й вибором кроку дискретизації.

Елемент таблиці, отриманий шляхом дискретизації, називається *піксель (picture element)*. Кроки дискретизації повинні вибиратися досить малими для того, щоб погрішність дискретизації була незначна й цифрове подання зберігало основну інформацію. З фізичної точки зору вибір кроку дискретизації диктується шириною просторового спектра зображення. Чим більше ширина спектра  $W$ , тим менше крок дискретизації.

### **Квантування зображень**

При цифровій обробці зображень безперервний динамічний діапазон значень яскравості ділиться на ряд дискретних рівнів. Ця процедура називається *квантуванням*. При цьому відбувається перетворення безперервної змінної  $X$  у дискретну змінну  $X_{kv}$ , що приймає кінцеву безліч значень  $\{r_1, \dots, r_L\}$ . Ці значення називаються рівнями квантування. У загальному випадку перетворення виражається східчастою функцією. Якщо яскравість  $X$  відліку зображення належить інтервалу  $(d_j, d_{j+1}]$  (тобто, коли  $d_j < X \leq d_{j+1}$ ), то вихідний відлік замінюється на рівень квантування  $r_j$ , де  $j=1, \dots, L$  - пороги квантування. При цьому покладається, то динамічний діапазон значень яскравості обмежений і дорівнює  $[d_1, d_{L+1}]$ .

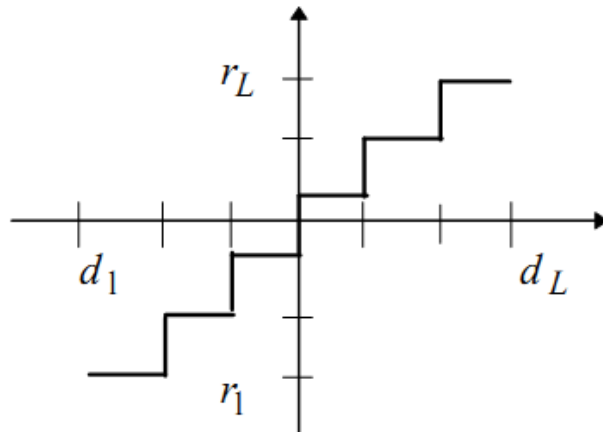


Рисунок 1. 4 - Функція, що описує квантування

Основне завдання при цьому складається у визначенні значень порогів  $d_j$  і рівнів  $r_j$  квантування. Найпростіший спосіб рішення цього завдання складається в розбивці динамічного діапазону на однакові інтервали. Однак таке рішення не є найкращим. Якщо значення інтенсивності більшості отсчетов зображень згруповані, наприклад, в «темній» області й число рівнів обмежене, то доцільно квантувати нерівномірно. В «темній» області треба квантувати частіше, а в «світлій» рідше. Це дозволить зменшити помилку квантування  $\varepsilon = X - X_{kv}$ , а

середньоквадратична помилка дорівнює  $E(\varepsilon^2) = \frac{\Delta^2}{12}$  де  $\Delta = (d_{L+1} - d_1) / L$ ,

У системах цифрової обробки зображень прагнуть зменшити число рівнів і порогів квантування, тому що від їхньої кількості залежить обсяг інформації, необхідний для кодування зображення. Однак при відносно невеликому числі рівнів на квантованому зображенні можлива поява помилкових контурів. Вони виникають внаслідок стрибкоподібної зміни яскравості проквантованого зображення й особливо помітні на пологих ділянках її зміни. Помилкові контури значно погіршують візуальна якість зображення, тому що зір людини особливо відчутно саме до контурів. При рівномірному квантуванні типових зображень потрібно не менш 64 рівнів.

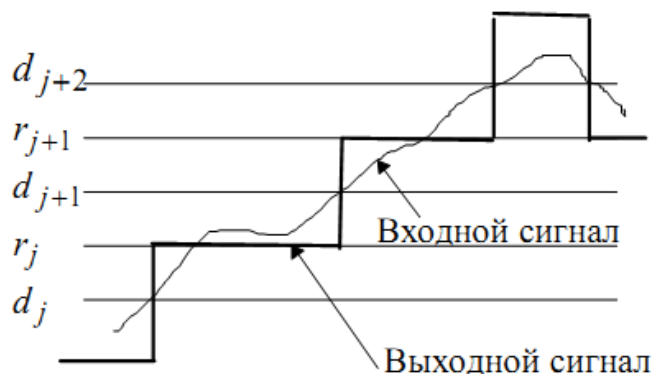


Рисунок 1.5 - Механізм виникнення помилкових контурів



Рисунок 1.6 - Результати рівномірного квантування відповідно на 256 й 14 рівнів квантування.

**Функції, використовувані для моделювання процесів дискретизації й квантування зображення**

Функція обробки блоків зображення *blkproc*

Синтаксис

$$D = \text{blkproc}(S, [m \ n], \text{fun}, P1, P2)$$

Функція  $D = \text{blkproc}(S, [m \ n], \text{fun}, P1, P2)$  формує нове зображення  $D$ , пікселі якого є результатом обробки функцією  $\text{fun}$  кожного не блоку, що перекриває, розміру  $m \times n$  вихідного зображення  $S$ .  $P1, P2$  – передані у функцію  $\text{fun}$  додаткові параметри. Вихідне зображення тимчасово доповнюється стовпцями нулів праворуч і рядками нулів знизу так, щоб розміри зображення були кратні відповідним розмірам блоку (рис.1.7). Функція використовується для обробки бінарних і напівтонових зображень.

Приклад створення функцію по імітації дискретизації зображення:

```
function disk (Im,s)
```

```
Imd = blkproc(Im, [m n], 'mean2(x)*ones(size(x))');
```

```
figure,imshow(Imd)
```

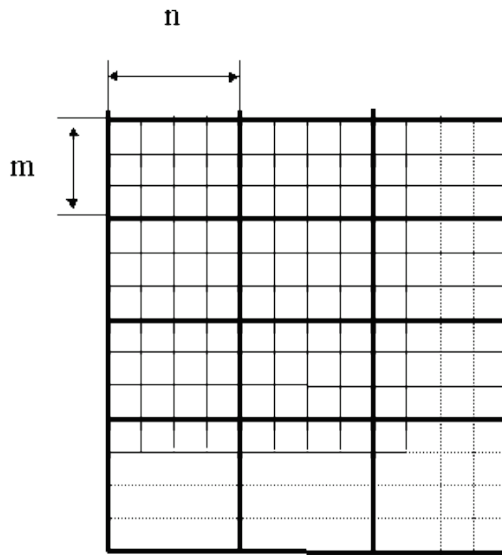
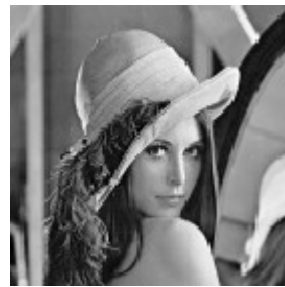


Рисунок 1.7 - Розбивка зображення на блоки

розрешення 512x512 пікселів



розрешення 128x128 пікселів



розрешення 64x64 пікселів



розрешення 32x32 пікселів



Рисунки

к 1.8 - Дискретизація зображення з різним дозволом.

У рядок **fun** міститься вираження, припустимо синтаксисом MatLab. Це вираження інтерпретується при виконанні кожного блоку.

*Гістограма розподілу яскравості зображення*

Гістограма цифрового зображення – це дискретна функція, що описує частоту появи (імовірність) рівня сірого в зображенні, представлена у вигляді графіка.

По осі абсцис відкладаються номери градацій рівнів сірого по зростанню (значення інтенсивності), а по осі ординат - кількість пікселів, що мають даний рівень сірого (частоту появи даної інтенсивності).

Гістограма може свідчити про загальну яскравість і контраст зображень, тому є коштовним методом як кількісної, так й якісної обробки зображення.

Гістограма - опис частоти появи значень окремих елементів зображення незалежно від значень інших елементів, що є дискретним аналогом щільності розподілу ймовірності.

Математична процедура обчислення гістограми можна описати:

$$h(k) = \frac{1}{M \cdot N} \cdot \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \delta(k - x_{m,n})$$

де  $\delta$  - функція Кронекера.

Гістограма може бути визначена не тільки для всього вихідного зображення але й окремих його ділянок, у цьому випадку вона називається локальною.

Функція побудови гістограми *imhist*

Синтаксис

*imhist(I,n)* – виводить у поточне вікно гістограму яркостей пікселів напівтонового зображення. Гістограма складається з *n* стовпців. За замовчуванням *n = 256* для напівтонового зображення й *n = 2* для бінарного зображення. Під малюнком гістограми виводиться шкала яркостей;

Перетворення яскравості на базі гістограми зображення

Гістограма – це функція розподілу яркостей напівтонового зображення. Нехай *L* – максимально можливе значення яскравості (звичайно 255). Обчислити гістограму означає побудувати масив *H* розміром в 256 елементів, значення яких дорівнюють кількості пікселів з яскравістю, рівної номеру елемента *H(i)*, тобто *i*. Обчислення виконуються проходом по всіх пікселіах зображення й додаванням одиниці до значення відповідного елемента масиву *H*. У гістограмі типового зображення яскравість пікселів розподілена нерівномірно (рис. 1.9). У результаті часто на темних ділянках зображень деталі виявляються погано помітними.

изображение с пониженной яркостью и контрастом

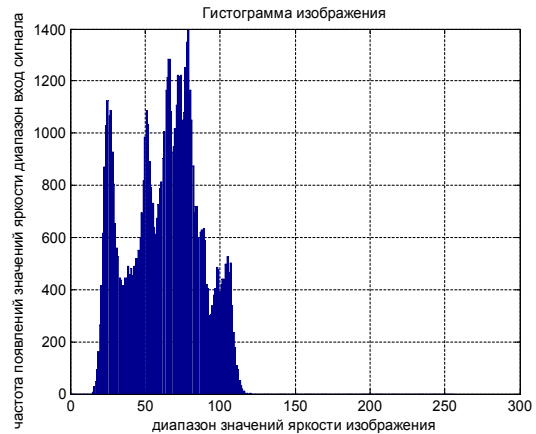
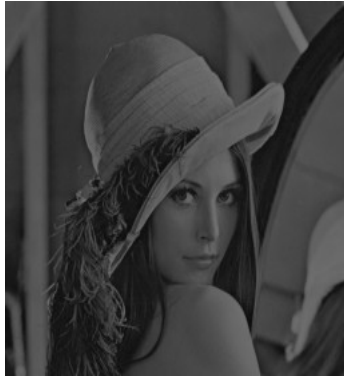


Рисунок 1.9 - Вихідне зображення і його гистограма

Розроблено ряд методів поліпшення зображень на базі зміни гистограми. Найбільш простим з них є еквалізація (лінійне вирівнювання) гистограми. Це перетворення прагне змінити розподіл яркостей зображення, роблячи графік гистограми максимально близьким до горизонтальній прямій (рис. 1.10).

*Алгоритм еквалізації (лінійного вирівнювання) гистограми.*

Крок 1. Обчислити гистограму вихідного зображення  $f(x, y)$ , записати її в масив  $H$ . Обчислити число пікселів  $N$ .

Крок 2. Масив  $H$  нормалізувати так, щоб сума всіх елементів стала дорівнює 255:

$$H(j) = (255/N) * H(j)$$

Крок 3. Обчислити кумулятивну гистограму, що підсумує розподіл яскравості від  $0$  до  $i$

$$Sum(i) = \sum_0^i H(j).$$

Крок 4. Обчислити нові значення яскравості кожного пікселу з координатами  $(x, y)$  по формулі

$$g(x, y) = Sum(f(x, y)).$$

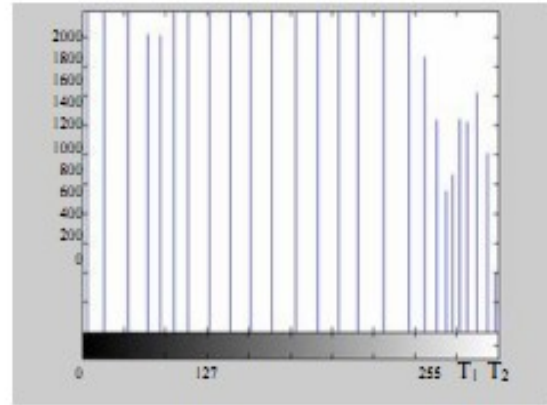


Рисунок 1.10 - Приклад вирівнювання гістограми зображення,

На рис. 1.10 показаний результат вирівнювання гістограми зображення. Якщо зображення має багато темних ділянок і мале число ділянок з високою яскравістю або навпаки, вирівнювання гістограми коректує яскравість зображення, рівняючи інтегральні площі ділянок з різними яркостями.

### Стандартні статистики зображення.

Функція обчислення середнього значення елементів матриці *mean2*

Синтаксис

$$m = \text{mean2}(S)$$

Функція  $m = \text{mean2}(S)$  обчислює середнє значення елементів матриці  $S$ . Дана функція еквівалентна функції  $\text{mean}(S(:))$ .

Функція *std2* обчислює середньоквадратичне відхилення.

Синтаксис

$d = \text{std2}(S)$  – обчислює середньоквадратичне відхилення елементів матриці  $S$ . Дана функція еквівалентна функції  $\text{std}(S(:))$ .

### Код програми ПО Matlab.

*% ОСНОВИ РОБОТИ З ЗОБРАЖЕННЯМИ В MATLAB*

*close all; %Видалення графічних вікон*

*clear*

*Завантаження картинки в робочу область, перегляд і перетворення масиву чисел*

*A=imread('C:\Documents and Settings\Admin\Moї документи\Moї документи\MATLAB\Image\pic\Portrait\_w.bmp');% завантаження картинки в робочу область Matlab whos A, % Видає основну інформацію про зображення*

*%Функція imshow(I, n) виводить на екран напівтонове зображення I,*

*використовуючи*

*%при виводі n рівнів сірого. Якщо при виклику функції опустити параметр n,*



*%те коли MATLAB запущений у графічному режимі TrueColor, для виводу напівтонового зображення використовується 256 градацій сірого figure,imshow(A), set(gca,'FontName','Arial Cyr','FontSize',10),title('Оригінал'), pixval on, % інтерактивне визначення яскравості окремих пікселів*



Рисунок 1.11 - Вхідне зображення приклада програми Matlab.

*figure,imshow(A), set(gca,'FontName','Arial Cyr','FontSize',10),title('Оригінал'), % Функція impixelregion. Засоби перегляду локального масиву пікселів відкривають окреме вікно, що містить значення пікселів деякої локальної околиці даного зображення. impixelregion,*



Рисунок 1.12 - Засоб перегляду локального масиву пікселів функція impixelregion.

```

%Функція mat2gray створює з матриці чисел напівтонове зображення,
здійснюючи необхідні операції масштабування для приведення до виду,
прийнятому для напівтонових зображень, масштабуючи значення елементів
матриці A, що перебувають у діапазоні [amin,amax], до діапазону [0, 1]
Ad=mat2gray(A);
whos Ad,
%Для одержання двійкових зображень із напівтонових у використається
функція im2bw, що має синтаксис
Ag= im2bw(Ad);
whos Ag,
figure,imshow(Ag), set(gca,'FontName','Arial Cyr','FontSize',10),title('двійкове,
дворівневе зображення') ,

```

двоичное, двухуровневое изображение



Рисунок 1.13 - Приклад перетворення у дворівневе зображення.

Формування вектора-рядка перетину зображення й відображення графіка лінії перетину

```

Test=imread('C:\Documents and Settings\Admin\Moї документи\Moї
документи\MATLAB\Image\pic\Yfq_0.5-5.0.bmp');% завантаження картинки
в робочу область Matlab

```

```
whos Test,
```

```

%Виходячи із властивості зображення Test застосовуємо Функцію
I=rgb2gray(RGB) створює напівтонове зображення I, перетворює R, G, B
складовим пікселам повнокольорового зображення у відповідні їм значення
яскравості.

```

```

Test_w=rgb2gray(Test) ;
figure,imshow(Test_w), ,set(gca,'FontName','Arial Cyr','FontSize',10), title('мест
зображення відтінки сірого') ,

```

*pixel on, % визначаємо номер рядка, що хочемо відобразити на графіку профілю перетину зображення*

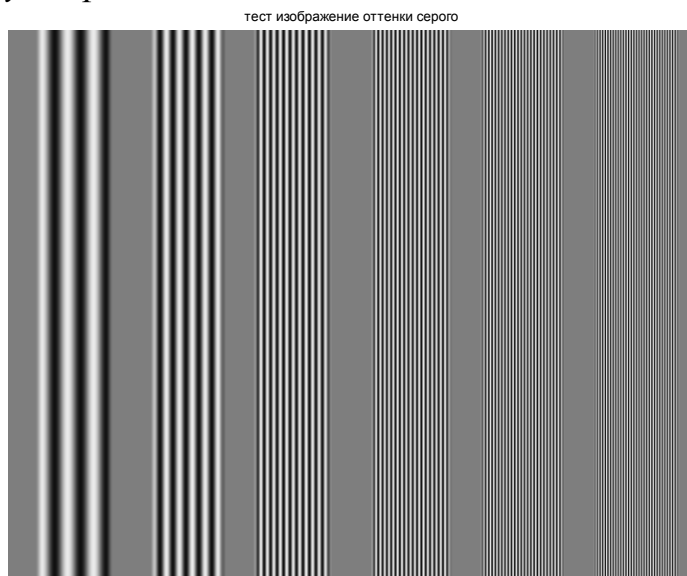


Рисунок 1.14 - Вихідне зображення приклада програми Matlab.

*figure, plot(Test\_w(256,:)), grid on, set(gca,'FontName','Arial Cyr','FontSize',10),  
title('линия профілю перетину зображення'),  
xlabel('пиксели'), ylabel('значення яркості '),*

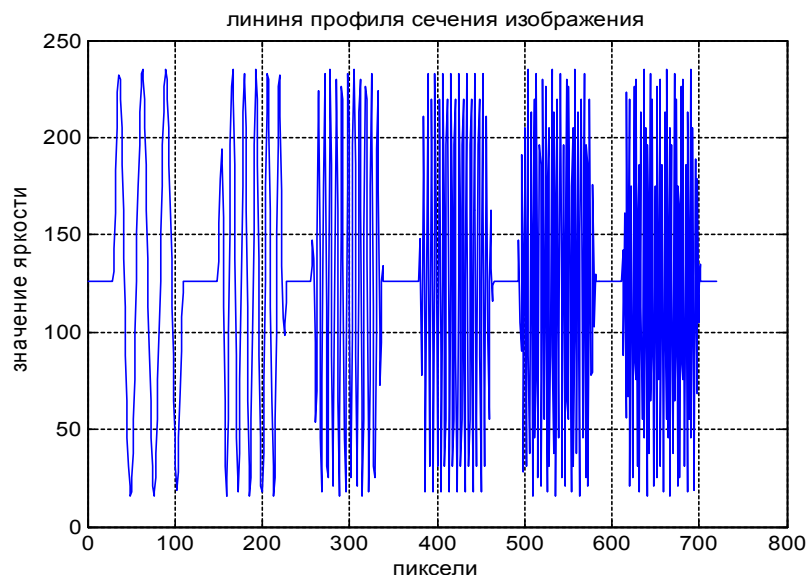


Рисунок 1.15 - Профіль перетину вихідного зображення.

*Дискретизація як фактор, що визначає просторовий дозвіл зображення*  
*L=imread('C:\Documents and Settings\Admin\Мої документи\Мої документи\MATLAB\Image\pic\Lenna\_(test\_image).bmp'); %завантаження картинки в  
робочу область Matlab  
whos L, %Вивдає основну інформацію про зображення*

*Lenna\_w=rgb2gray(L) ;% яркостна складова зображення Lenna дозволу 512x512 пікселів*

*% D\_4 яркостна складова зображення 128x128 наведена до вихідного розміру 512x512 що еквівалентно зменшенню числа отсчетов дискретизації зображення в 4-разу (тобто зменшенню частоти дискретизації)*

*% Функція D=blkproc(S, [m n], fun) формує нове зображення D, пікселі якого є результатом обробки функцією fun кожного із блоків, що не перекриваються, розміру вихідного зображення S.*

*% Вихідне зображення S розбивається на блоки, що не перекриваються, пікселів. Блоки розташовуються впритул друг до друга, починаючи з лівого верхнього кута зображення.*

*D\_4=blkproc(Lenna\_w, [4 4], 'mean2(x)\*ones(size(x))'); % зображення з дозволом 128x128 пікселів*

*D\_8=blkproc(Lenna\_w, [8 8], 'mean2(x)\*ones(size(x))'); % зображення з дозволом 64x64 пікселів*

*D\_16=blkproc(Lenna\_w, [16 16], 'mean2(x)\*ones(size(x))'); % зображення з дозволом 32x32 пікселів*

*figure,imshow(D\_8,[]), set(gca,'FontName','Arial Cyr','FontSize',10),title('зображення з дозволом 64x64 пікселів'), figure,*

*subplot(2,2,1),imshow(Lenna\_w,[]),set(gca,'FontName','Arial Cyr','FontSize',10),title('дозвіл 512x512 пікселів'),*

*subplot(2,2,2),imshow(D\_4,[]),set(gca,'FontName','Arial Cyr','FontSize',10),title('дозвіл 128x128 пікселів'),*

*subplot(2,2,3),imshow(D\_8,[]),set(gca,'FontName','Arial Cyr','FontSize',10),title('дозвіл 64x64 пікселів'),*

*subplot(2,2,4),imshow(D\_16,[]),set(gca,'FontName','Arial Cyr','FontSize',10),title('дозвіл 32x32 пікселів'),*

разрешение 512x512 пикселей



разрешение 128x128 пикселей



разрешение 64x64 пикселей



разрешение 32x32 пикселей



Рисунок 1.16 - Зображення з різним дозволом

### Квантування, зміна градація яскравості

```
% зберігаючи незмінним число отсчетов дискретизації будемо змінювати  
число рівнів квантування з 256 до 2-х  
Max=max(Lenna_w(:)); Min=min(Lenna_w(:));  
% Вихідне зображення має число рівнів квантування 256 [0 255]  
%функція Y = fix(X) повертає значення з усіканням дробової частини числа;  
%Використаємо функцію для огрубіння рівнів квантування, що  
еквівалентно зменшенню рівнів квантування  
QLen_4=fix(Lenna_w/4)*4; %число рівнів квантування 256/4=64  
QLen_8=fix(Lenna_w/8)*8; %число рівнів квантування 256/8=32  
QLen_16=fix(Lenna_w/16)*16; %число рівнів квантування 256/16=16  
QLen_32=fix(Lenna_w/32)*32; %число рівнів квантування 256/32=8  
QLen_64=fix(Lenna_w/64)*64; %число рівнів квантування 256/64=4  
QLen_128=fix(Lenna_w/128)*128; %число рівнів квантування 256/128=2  
figure,imshow(QLen_16,[]), set(gca,'FontName','Arial  
Cyr','FontSize',10) ,title('зображення з дозволом 512x512 пікселів і рівнів  
квантування 16') ,  
figure,  
subplot(2,2,1),imshow(Lenna_w,[]),set(gca,'FontName','Arial  
Cyr','FontSize',10),title(' рівнів квантування 256') ,  
subplot(2,2,2),imshow(QLen_16,[]),set(gca,'FontName','Arial  
Cyr','FontSize',10),title(' рівнів квантування 16') ,
```

```
subplot(2,2,3),imshow(QLen_64,[]),set(gca,'FontName','Arial
Cyr','FontSize',10),title('рівнів квантування 4') ,
subplot(2,2,4),imshow(QLen_128,[]),set(gca,'FontName','Arial
Cyr','FontSize',10),title('рівнів квантування 2') ,
```

уровней квантования 256



уровней квантования 16



уровней квантования 4



уровней квантования 2



/  
Рисунок 1.17 - Зображення з різними рівнями квантування амплітуди.

Статистичні характеристики зображення

```
L_2=blkproc(Lenna_w, [2 2], 'mean2(x)'); %% зображення з дозволом 256x256
пікселів
```

```
L2n=L_2*0.5/255;
```

```
figure,imshow(L2n),set(gca,'FontName','Arial Cyr','FontSize',10),
title('зображення зі зниженою яскравістю й контрастом') ,
```

```
[N M]=size(L2n); %%розмір зображення
```

изображение с пониженной яркостью и контрастом

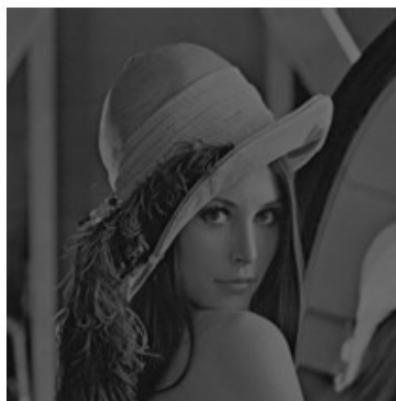


Рисунок 1.18 - Вхідне зображення зі зниженою яскравістю й контрасту приклада програми Matlab

```

%Визначення максимального й мінімального значення рівня яскравості
зображення
Max=max(L2n(:)*254); min=min(L2n(:)*254);
disp(['максимальне значення яскравості зображення L2n, Max='
mat2str(round(Max))]);
disp(['мінімальне значення яскравості зображення L2n, min='
mat2str(round(min))]);
%Середнє значення яскравості зображення або середнє арифметичне
елементів масиву
mean=mean(L2n(:)*254);
disp(['Середнє значення яскравості зображення L2n, mean='
mat2str(round(mean))]);
% Стандартне відхилення значень яскравості зображення або
% середньоквадратичне відхилення елементів масиву
std=std(L2n(:)*254);
disp(['Стандартне відхилення яскравості зображення L2n, std='
mat2str(round(std))]);
% Гістограма - відображає число можливих рівнів яскравості N, на які
розбивається діапазон яскравості (за замовчуванням він дорівнює 256), що
лежать у діапазоні яркостей [0 255] для напівтонового зображення.
Гістограма відповідає дискретній функції щільності розподілу ймовірності
випадкового процесу
HL=imhist(L2n);
figure, bar(HL),grid on,set(gca,'FontName','Arial Cyr','FontSize',10),
title('Гістограма зображення'),xlabel('діапазон значень яскравості
зображення'),
ylabel('частота появлених значень яскравості діапазон вхід сигналу'),

```

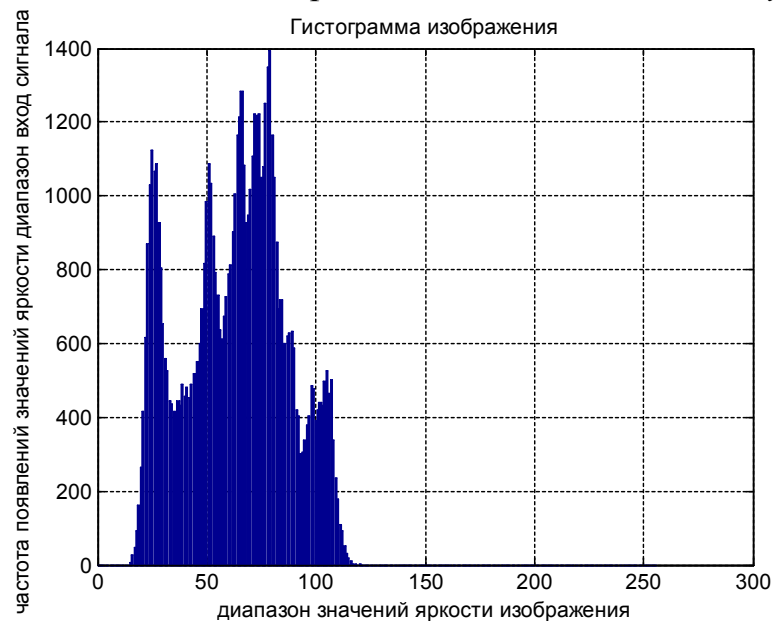


Рисунок 1.19 - Гістограма вихідного зображення

```

% функція розподілу ймовірностей вихідного зображення, що
апроксимується гістограмою розподілу або кумулятивної гістограми
зображення
CHL=cumsum(HL)./(N*M);
figure, stairs(CHL),grid on,set(gca,'FontName','Arial Cyr','FontSize',10), title('
кумулятивна гістограма зображення') ,xlabel('діапазон значень яскравості
зображення') ,
ylabel('імовірність появ діапазону яскравості '),

%Рівномірне перетворення гістограми змінює значення інтенсивностей
%пикселовів результуючого зображення в діапазоні [0 1]
for c=1:N;
    for r=1:M;
        L_2h(c,r)=CHL(ceil(L2n(c,r)*254+eps));
    end
end

% Задамо мінімальне й максимальне значення яскравості перетвореного
зображення в діапазоні [m_tr M_tr]
M_tr=240; m_tr=10;
for c=1:N;
    for r=1:M;
        L_2h_tr(c,r)=((M_tr-m_tr)*CHL(ceil(L2n(c,r)*254+eps))+m_tr)/255;
    end
end
MaxLh_tr=max(L_2h_tr(:)); minLh_tr=min(L_2h_tr(:));
whos L_2h, %Видає основну інформацію про зображення
figure,imshow(L_2h),set(gca,'FontName','Arial Cyr','FontSize',10), title('
зображення рівномірно перетвореної гістограмою') ,

```

изображение равномерно преобразованной гистограммой



Рисунок 1.20 - Зображення з вирівняної гістограмою.



```

HLh=imhist(L_2h);
figure, bar(HLh),grid on,set(gca,'FontName','Arial Cyr','FontSize',10), title('
рівномірно перетворена гістограма зображення'),xlabel('діапазон значень
яскравості зображення') ,
ylabel('частота появ значень яскравості діапазон вхід сигналу'),
MaxLh=max(L_2h(:)*254); minLh=min(L_2h(:)*254);
disp(['максимальне значення яскравості зображення рівномірної
гістограми L_2h, MaxLh=' mat2str(round(MaxLh))]);
disp(['мінімальне значення яскравості зображення рівномірної гістограми
L_2h, minLh=' mat2str(round(minLh))]);

```

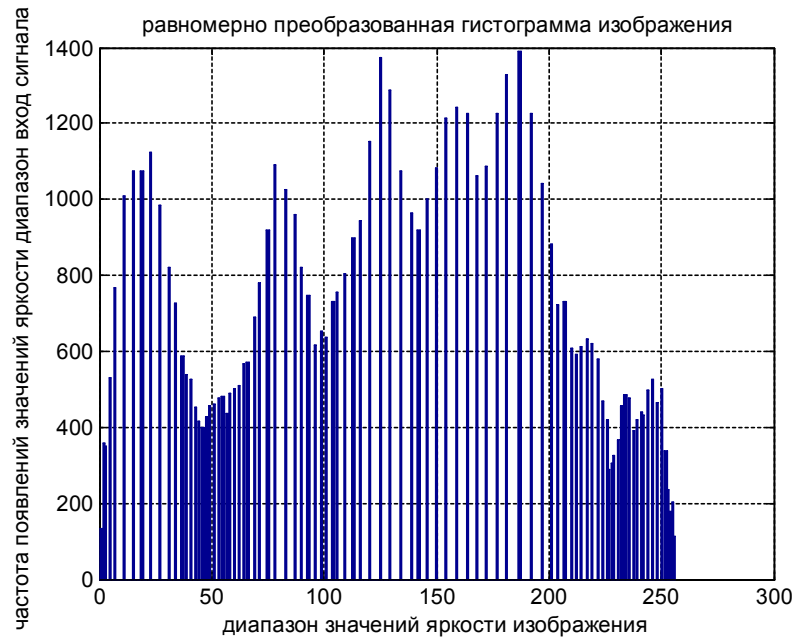


Рисунок 1.21 - Рівномірно вирівняна гістограма зображення.

```

figure,
subplot(1,2,1),imshow(L2n),set(gca,'FontName','Arial Cyr','FontSize',10),title('
зображення зі зниженою яскравістю й контрастом') ,
subplot(1,2,2),imshow(L_2h),set(gca,'FontName','Arial Cyr','FontSize',10),title('
зображення рівномірно перетвореної гістограмою') ,

```



Рисунок 1.22 - Вхідне зображення й зображення з вирівняної гистограмою

### **Порядок проведення лабораторної роботи**

1. Зчитати зображення у відповідності з варіантом завдання до робочої області MATLAB за допомогою команди `imread`:

Зображення, що відповідають варіантам 1-8, знаходяться в підкаталогах MATLAB; .

2. За допомогою команди `size` вивести розмір зображення, присвоївши змінній `M` значення, що відповідає кількості рядків зображення, а змінній `N` - значення, що відповідає кількості стовпчиків. `[M,N]=size(i);`
3. Підрахувати об'єм зображення в байтах.
4. Перевірити правильність розрахунку за допомогою команди `whos`.
5. Вивести зображення на екран за допомогою команди `imshow`:
6. За допомогою засобів перегляду визначити значення пікселів.
7. Змінити дозволу зображення для значень 128x128, 64x64, 32x32 пікселів.
8. Змінити кількість рівнів квантування зображення на 32, 16, 4. Вивести одночасно на екран результати роботи, для цього скористатись командою `subplot`.
9. Визначити гистограму зображення й зробити її перетворення.
10. Зберегти на жорсткий диск до потокового каталогу зображення у форматі JPEG з налаштуваннями якості `q = 50, 20, 5`. Для цього застосувати команду `imwrite`:

Переглянути збережені зображення за допомогою стандартних засобів операційної системи.

## Варіанти завдання

Варіант	Файл зображення
1	Portrait_w.bmp; YP.bmp; Lenna_(test_image).bmp
2	face.gif; Yfq_0.5-4.43-5.5.bmp; pout.jpg
3	clown.gif; boat.bmp; palzt.jpg
4	Portrait_w.bmp; pollen.tif; FR.bmp
5	Lenna_(test_image).bmp; gory.jpg; H_h.bmp
6	zelda.bmp; valley.bmp; 31.bmp
7	Portrait_w.bmp; pollen.tif; H_h.bmp
8	clown.gif; boat.bmp; palzt.jpg

### **Порядок оформлення звіту та його представлення й захист.**

Результати проведеної лабораторної роботи оформлюються протоколом. Він повинен містити такі дані:

- 1) тема практичної роботи;
- 2) позначка практичної роботи;
- 3) короткі теоретичні відомості;
- 4) порядок проведення практичної роботи;
- 5) зображення відповідно до п. 1-9;
- 6) висновки.

### **Запитання для самоперевірки:**

1. Як представляється зображення в системі MATLAB?
2. Які команди використовують для зчитування зображення до робочої області та виведення зображення на екран? Наведіть їх синтаксис.
3. Які формати зображення підтримує система MATLAB?
4. Які класи даних використовують у системі MATLAB?
5. Чому необхідно використовувати команди figure та subplot?
6. У чому полягає відмінність між результатами виконання команд imshow(i) і imshow(i,[])?

### **Перелік використаної літератури**

1. Гонсалес, Р. Цифрова обробка зображень у середовищі MATLAB /Р. Гонсалес, Р. Вудс, С. Эддинс. ? М.: Техносфера, 2006. ? 616 с.
2. Сергиенко А.Б. Цифрова обробка сигналів. Учень для вузів. - С-Пб.: Питер, 2002. - 608 с.

3. Дияконів В.П. MATLAB 6.5 SP1/7 + Simulink 5/6. Обробка сигналів і проектування фільтрів - М.: СОЛОНІЙ^-ПРЕСС, 2009. - 577 с.
4. Лімонов О.С. Цифрова обробка сигналів: Конспект лекцій. - В "ТЕС", Одеса, 2011. - 115 с.
5. Лазарєв Ю. Моделювання процесів і систем в MATLAB. Навчальний курс. Київ, 2005. . - 512 с.

## Лабораторна робота № 2.

### "Основи методів просторової обробки зображень. Перетворення яскравості і контрасту"

**Мета роботи** – вивчення функцій амплітудних перетворень зображень і придбання практичних навичок їх використання.

**Завдання на підготовку до лабораторної роботи.** Під час проведення лабораторної роботи студент повинен:

**знати:**

- основні характеристики та можливості пакета Signal Processing Toolbox;
- цифрові методи обробки сигналів та зображень;
- будову і роботу програмного пакета MATLAB;

**уміти:**

- використовувати ручне введення та конструювання інструментальних засобів пакета MATLAB;
- використовувати можливості пакета Signal Processing Toolbox;
- досліджувати та вимірювати амплітудні та ймовірнісні характеристики зображень;
- зберігати та візуалізувати інформацію в програмному пакеті MATLAB.

### Теоретичні відомості.

#### *Градаційні перетворення*

Яскравість і контрастність зображення є найважливішими характеристиками зображень. *Яскравість* (brightness) характеризує інтенсивність випромінюваного або відбитого деталями зображення світла і є енергетичною характеристикою. Одиницею виміру яскравості служить кд/м<sup>2</sup>. Деякі значення яскравості для орієнтування наведені нижче:

- сонце -  $1,5 \cdot 10^9$  кд/м<sup>2</sup>;
- нитка лампи накаливання -  $5 \cdot 10^6$  кд/м<sup>2</sup>;
- полум'я сірника -  $5 \cdot 10^3$  кд/м<sup>2</sup>;
- екран телевізора - 50...100 кд/м<sup>2</sup>.

*Яскравість* цифрового зображення вимірюється інтенсивністю його пікселів і залежить від класу дані зображення: для зображень класу *double* вона лежить у діапазоні 0...1, для зображень класу *uint8* – 0...255. Білим ділянкам зображення відповідає максимальна яскравість, чорним ділянкам - мінімальна.

*Контрастність* (contrast) у загальному випадку характеризує діапазон зміни яскравості різних ділянок зображення, тобто його динамічний діапазон. Контраст є безрозмірною величиною, а для його виміру існують кілька виражень. Для оцінки зображень, завантажених у середовище MATLAB доцільно використати наступне вираження, що дає нормоване значення контрасту:

$$K = (V_{max} - V_{min}) / V_{0max}$$

Тут  $V_{0max}$  максимально досяжна яскравість пікселів у графічному вікні MATLAB. При цьому значення контрасту лежать у діапазоні 0...1.

Градаційні або амплітудні перетворення зображень пов'язані з перетворенням інтенсивності (яскравості) пікселів. Оскільки вони виробляються в площині зображень їх можна віднести до найпростішої обробки цифрового зображення в просторовій області, тобто до найпростішої просторової фільтрації [1].

Процеси в просторовій області можна позначити рівнянням

$$g(x, y) = T[f(x, y)], \quad (1)$$

де  $f(x, y)$  – вхідне зображення,  $g(x, y)$  – вихідне (оброблене) зображення, а  $T$  – деякий оператор (перетворення) над  $f$ , що визначений у деякій околиці крапки з координатами  $(x, y)$ . (Крім того, оператор  $T$  може обробляти послідовність зображень, наприклад, він може підсумувати  $K$  вхідних зображень для придушення шуму). Як околиця крапки з координатами  $(x, y)$  використовується квадратна або прямокутна область із центром у крапці  $(x, y)$ , що найчастіше називають маскою (рис. 2.1).

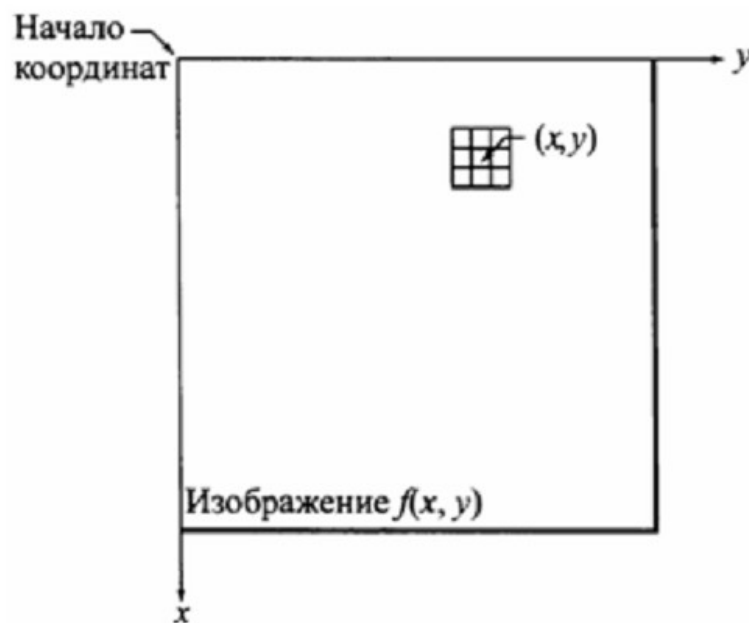


Рисунок 2.1 — Околиця (маска)  $3 \times 3$  пікселів навколо крапки  $(x, y)$  зображення

Центр заданої маски переміщається від пікселя до пікселю, починаючи, наприклад, з верхнього лівого кута, і на своєму шляху накриває різні околиці зображення. Перетворення  $T$  застосовується в кожній крапці  $(x, y)$ , даючи в результаті оброблене значення  $g$  для даної крапки. У процесі обчислень на кожному кроці використовуються тільки піксели покриті маскою із центром  $(x, y)$ .

У випадку градаційні перетворень маска має розмір  $1 \times 1$  піксел. При цьому інтенсивність обробленого пікселя  $g$  залежить тільки від інтенсивності вихідного пікселя, а перетворення  $T$  є функцією перетворення інтенсивностей. У цьому випадку вираження (1) можна записати в простій формі

$$s = T(r),$$

де  $s$  – яскравість пікселів вихідного зображення  $g(x, y)$ ,  $r$  – яскравість пікселів вхідного зображення  $f(x, y)$ ,  $T$  – функція перетворення інтенсивностей.

#### Амплітудні перетворення зображень

Амплітудні перетворення відносять до крапкових процесів. Крапкові процеси - це алгоритми, які змінюють значення елементів у зображенні. Алгоритми крапкових процесів сканують зображення елемент за елементом, здійснюючи перетворення елементів зображення.

#### Контраст зображення

Важливою характеристикою зображення є інтенсивність. Існує верхня границя інтенсивності  $J_{max}$ , що визначає величину припустимої інтенсивності реального зображення й нижня границя інтенсивності, що може бути більше нуля  $J_{min} > 0$ . У цьому випадку різниця  $Dj = J_{max} - J_{min}$  визначає діапазон значень інтенсивності. Для характеристики відносної зміни півтонів у сусідніх крапках використовують величину контрасту. Нехай інтенсивність зображення в крапці  $(X1, Y1)$  становить  $J1 = J(X1, Y1)$ , а в крапці  $(X2, Y2)$  дорівнює  $J2 = J(X2, Y2)$ . Знайдемо  $J_{max} = \max(J1, J2)$  і  $J_{min} = \min(J1, J2)$ . Тоді за контраст приймається наступна величина:

$$K = (J_{max} - J_{min}) / J_{max}$$

Підвищення контрасту робить відображення вхідного діапазону яркостях зображення у вихідний діапазон яркостях. Найбільш простим і розповсюдженим способом є відображення вхідного діапазону яркостях у максимально припустимий діапазон яркостях за допомогою лінійної функції (мал. 1). Така операція дає більшу різницю в яркостях сусідніх пікселів, що полегшує виділення перепадів яркостях й, як правило, більше комфортне зорове сприйняття.

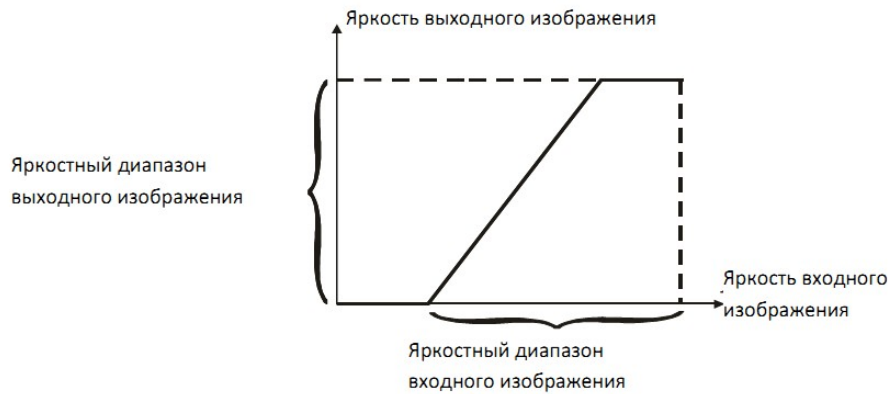


Рисунок 2.2 — Підвищення контрасту є відображення вхідного діапазону яркостях у максимально припустимий діапазон яркостях.

Для одержання негатива напівтонового зображення  $I$  необхідно знайти значення максимального рівня сірого  $Max$  й одержати негатив  $NI$  по формулі:  $NI = Max - I$ .

Функції, використовувані для зміни контрасту зображення

Функція контрастування зображення з гамма-корекцією *imadjust*

Синтаксис

$$ID = imadjust(IS, [low, high], [bottom top], gamma)$$

Функція  $ID = imadjust(IS, [low, high], [bottom top], gamma)$  створює напівтонове зображення  $ID$  шляхом контрастування вихідного напівтонового зображення  $IS$ . Значення яскравості в діапазоні  $[low, high]$  перетворюються в значення яскравості в діапазоні  $[bottom, top]$ . Значення яскравості, менші  $low$ , приймають значення  $bottom$ , а значення яскравості, більші  $high$ , приймають значення  $top$ . Значення  $top, bottom, low, high$  повинні лежати в діапазоні  $[0, 1]$ . Якщо в якості другого або третього параметрів передати порожній вектор  $[]$ , то за замовчуванням буде використатися вектор  $[0, 1]$ .

Параметр *gamma* визначає форму кривої характеристики передачі рівнів яскравості. Якщо *gamma* менше 1, то характеристика передачі рівнів буде опуклою й результуюче зображення буде світліше, ніж вихідне. Якщо *gamma* більше 1, то характеристика передачі рівнів буде ввігнутою й результуюче зображення буде темніше, ніж вихідне. За замовчуванням *gamma* дорівнює 1, що відповідає лінійній характеристиці передачі рівнів і відсутності гамма-корекції. Характеристика передачі рівнів для різних значень *gamma* наведені на рис. 2.3.



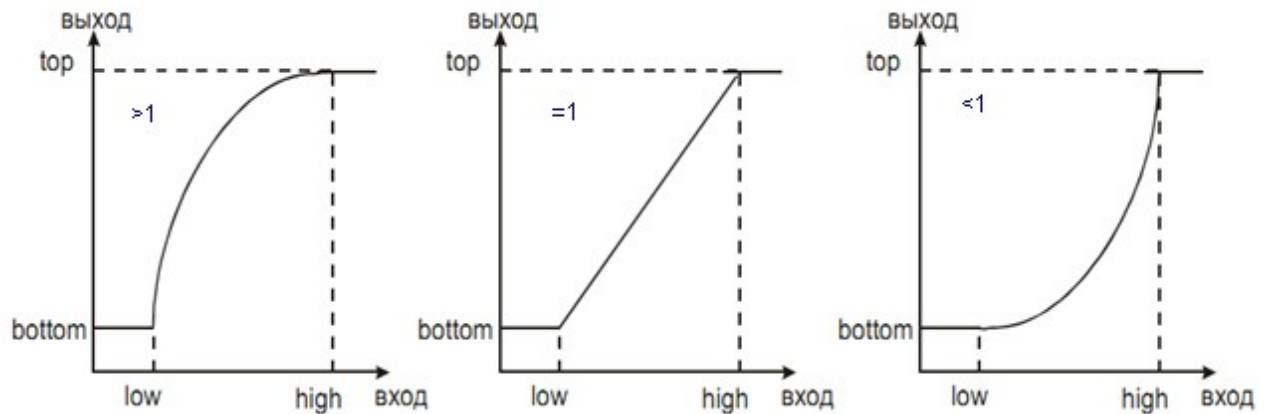


Рисунок 2.3 — Характеристика передачі рівнів функції *imadjust*

*Приклад.*

RGB зображення *I* читається з файлу й відображається на екрані (мал. 3 а). Для аналізу діапазону яркостей вихідного зображення *I* будується гістограма яркостей пікселів за допомогою функції *imhist(I)* (мал.3 б). По гістограмі видно, що піксели зображення *I* мають яскравості в діапазоні [0, 85], зображення *I* недостатньо контрастне. Функція *imadjust* “розтягує” вихідний діапазон яркостей на діапазон від мінімально можливої яскравості до максимально можливої. Результат перетворення виводиться на екран у нове вікно (мал. 3 в). Гістограма яркостей пікселів результуючого зображення показана на мал. 3 р. Результуюче зображення контрастніше, ніж вихідне.

*%Приклад демонструє контрастування зображення.*

*%Читання вихідного зображення й вивід його на екран.*

```
I=imread('gory.tif');
```

```
imshow(I);
```

*%Побудова гістограми вихідного зображення.*

```
figure, imhist(I);
```

*%Контрастування вихідного зображення.*

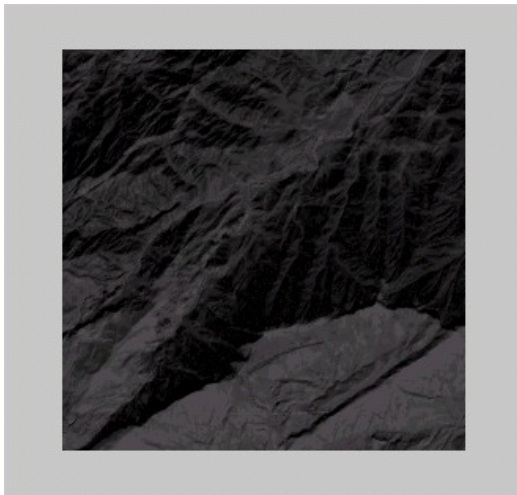
```
I=imadjust(I, [0 85]/255, [], 1);
```

*%Вивід перетвореного зображення на екран.*

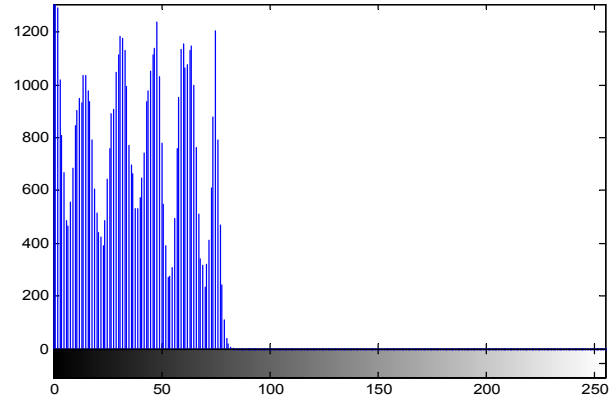
```
figure, imshow(I);
```

*%Вивід гістограми перетвореного зображення.*

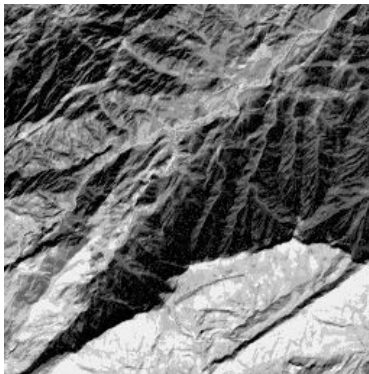
```
figure, imhist(I);
```



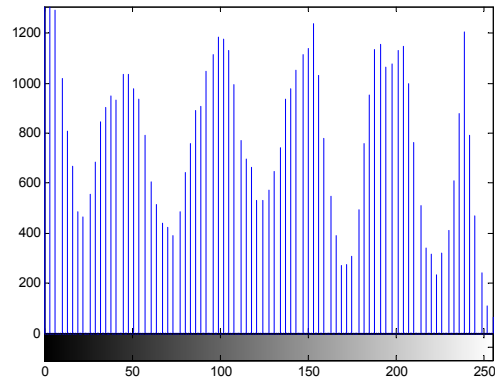
а



б



в



г

Рисунок 2.4 — Результат контрастування зображення функцією *imadjust*

Одним з найбільше часто застосовуваних в обробці зображень є перетворення, називане гама- корекцією.

Розглянемо приклад гама корекції зображення. Відзначимо, що при використанні функції *imadjust* діапазони даних вихідного й результуючого зображень описані у вигляді порожніх матриць. Якщо вони описані таким чином, тобто у вигляді порожніх матриць, то функція *imadjust* використає весь діапазон [0,1]. Результат застосування такого підходу показаний на зображенні.

```
[X,map] = imread('forest.tif')
I = ind2gray(X,map);
J = imadjust(I,[],[],0.5);
imshow(I)
figure, imshow(J)
```



гама=1



гама<1

Рисунок 2.5 — Зображення до й після застосування гама корекції

### Поняття гистограми. Видозміна гистограми

Гістограма є основою для численних методів просторової обробки зображень. Крім одержання корисної статистики про зображення, що втримується в гістограмі інформація також досить корисна й у таких завданнях, як стиск і сегментація зображень. Крім того, видозміна гістограми (гістограмна обробка) може бути успішно використане для поліпшення зображень. Гістограми досить прості як для програмного обчислення, так і для апаратної реалізації, що робить їхнім зручним інструментом для обробки зображень у реальному часі.

Гістограмою цифрового зображення з рівнями яскравості в діапазоні  $[0, L-1]$  – називається дискретна функція  $h(r_k)=n_k$  де  $r_k$   $k$ -й рівень яскравості, а  $n_k$  – число пікселів на зображенні, що мають яскравість  $r_k$ . Загальною практикою є нормалізація гістограми шляхом розподілу кожного з її значень на загальне число пікселів у зображенні, позначуване  $n$ . Тим самим, значення нормалізованої гістограмою будуть  $p(r_k)=n_k/n$  для  $k=0,1,\dots, L-1$ . Загалом кажучи,  $p(r_k)$  є оцінка ймовірності появи пікселів зі значенням яскравості  $r_k$ . Помітимо, що сума всіх значень нормалізованої гістограми дорівнює одиниці.

### Гістограма яскравості й робота з нею

Функції перетворення зображень, засновані на інформації, що витягається з гістограм яскравості, відіграють ключову роль при обробці зображень, чиненої при рішенні завдань поліпшення зображень, їхнього стиску, сегментації й опису.

**Гістограмою цифрового зображення** є число можливих рівнів яскравості  $L$ , що лежать у діапазоні  $[0, G]$ , називається дискретна функція  $h(r_k)=n_k$ , де  $r_k$  – це  $k$ -ий рівень яскравості з інтервалу  $[0, G]$ , а  $n_k$  – число пікселів зображення, рівні яскравості яких дорівнює  $r_k$ . Індеси в MATLAB починаються з 1, а не з 0.

Часто буває зручно працювати з нормованими гістограмами, які виходять розподілом елементів  $h(r_k)$  на загальне число пікселів зображення  $n$ :

$$p(r_k) = n_k/n \text{ при } k = 1, 2, \dots, L.$$

З погляду теорії ймовірностей, число  $p(r_k)$  – це ймовірність (частота) появи (присутності) рівня інтенсивності  $r_k$  у даному зображенні.

Функція **imhist** служить побудови гістограм розподілу яскравості й використовується у формі

$$h = \text{imhist}(f, b),$$

де  $h$  – гістограма зображення,  $f$  – вхідне зображення,  $b$  – число відображуваних градацій шкали яскравості (якщо не зазначено, то за замовчуванням  $b = 256$ ).

Щоб одержати нормовану гістограму, треба розділити отриману гістограму на загальне число пікселів зображення:

$$p = \text{imhist}(f, b) / \text{numel}(f).$$

**Numel(f)** дає число елементів масиву  $f$ , тобто число пікселів зображення.

Гістограми розподілу яркостей типового зображення природних об'єктів мають явно виражену нерівномірність, тому контраст такого зображення часто виходить низьким. **Еквалізація гістограми** має на увазі таке перетворення вихідного зображення, при якому його гістограма розподілу яскравості відповідала б рівномірному закону розподілу. Іншими словами, дане перетворення породжує зображення, рівні яскравості якого є рівноімовірними й покривають весь інтервал  $[0, 1]$ . Результатом цього процесу є більша контрастність вихідного зображення.

Еквалізація гістограми реалізована в пакеті Image Processing Toolbox функцією **histeq**:

$$g = \text{histeq}(f, nlev),$$

де  $f$  – вхідне зображення,  $nlev$  – число рівнів яскравості для вихідного зображення  $g$ . Параметр  $nlev$  за замовчуванням має значення 64, тому для щирої реалізації методу гістограмою еквалізації рекомендується використати  $nlev=256$ .

**Гістограмна еквалізація** відбувається перетворенням, що є адаптивним у тому розумінні, що воно залежить від гістограми вихідного зображення. Однак якщо функція перетворення вже обчислена, то вона не буде мінятися, поки не зміниться саме зображення. Обчислити функцію перетворення гістограми, побудовану на основі гістограмною еквалізації, можна ґрунтуючись на тім, що вона є кумулятивною сумою величин нормованої гістограми:

```
% побудова функції перетворення гістограми
hnorm = imhist(f) ./ numel(f); % нормуємо гістограму
sdf = cumsum(hnorm); % обчислюємо функцію кумулятивного розподілу
% CDF (Cumulative Distribution Function)
x = linspace(0, 1, 256); % задаємо масштаб графіка по осі X
figure; plot(x, sdf); % будуємо графік
```

```

axis([0 1 0 1]);
set(gca, 'xtick', 0:0.2:1)           %наносимо мітки на осі X
set(gca, 'ytick', 0:0.2:1)           %наносимо мітки на осі Y
xlabel ('Input intensity','fontsize', 9) %напис на осі X
ylabel ('Output intensity','fontsize', 9) %напис на осі Y
text(0.18, 0.5, 'Transformation function', 'fontsize', 9) %текст на графіку

```

### **Код програми ПО Matlab.**

```

%АМПЛІТУДНІ ПЕРЕТВОРЕННЯ ЗОБРАЖЕННЯ В MATLAB
close all; %Видалення графічних вікон
clear
F=imread('C:\MATLAB\Image\pic\mona.bmp');
whos F, %Видає основну інформацію про зображення
F_w=rgb2gray(F); %яркостна складова зображення Lenna дозволу 512x512
пікселів
F_w=F;
figure,imshow(F_w), set(gca,'FontName','Arial
Cyr','FontSize',10),title('Оригінал'),

```

Оригенал



Рисунок 2.6 — Вхідне зображення

```

%Визначення максимального й мінімального значення рівня яскравості
%зображення
Max=max(F_w(:)); min=min(F_w(:));
disp(['максимальне значення яскравості зображення F_w, Max='
mat2str(round(Max))]);
disp(['мінімальне значення яскравості зображення F_w, min='
mat2str(round(min))]);
максимальне значення яскравості зображення F_w, Max=255
мінімальне значення яскравості зображення F_w, min=0
% Одержати негатив напівтонового зображення
NF_w = double(Max)-double(F_w);
figure,subplot(1,2,1),subimage(F_w);

```

```
subplot(1,2,2),subimage(im2uint8 (NF_w/255)), set(gca,'FontName','Arial
Cyr','FontSize',10),title('Негатив') ; % im2uint8 (S)
```

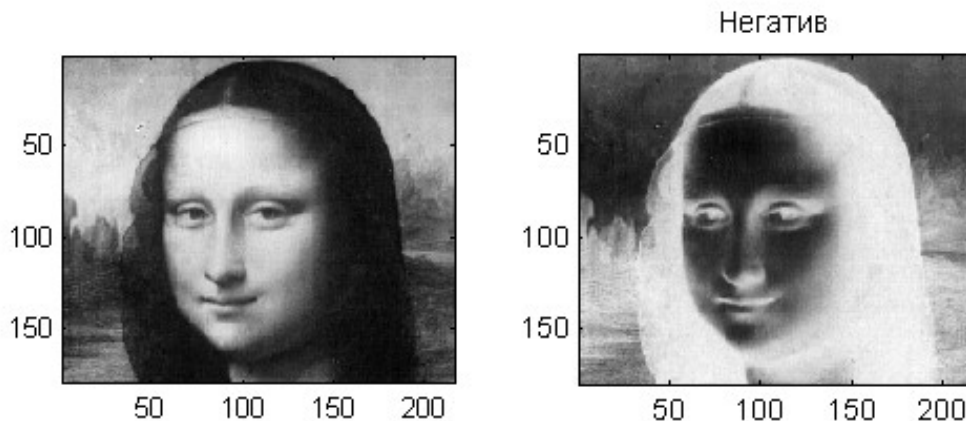


Рисунок 2.7 — Вхідне зображення і його негатив

*% Гістограма - відображає число можливих рівнів яскравості  $N$ , на які розбивається діапазон яскравості (за замовчуванням він дорівнює 256), що лежать у діапазоні яркостей  $[0\ 255]$  для напівтонового зображення. Гістограма відповідає дискретній функції щільності розподілу ймовірності випадкового процесу*

```
F=imread('C:\Documents and Settings\Сергій\Мої документи\MATLAB\Image\pic\pout.jpg'); %pout.jpg
```

```
whos F, %Виває основну інформацію про зображення
```

```
HF=imhist(F);
```

```
figure, bar(HF),grid on,set(gca,'FontName','Arial Cyr','FontSize',10),
title('Гістограма зображення'),xlabel('діапазон значень яскравості
зображення'),
```

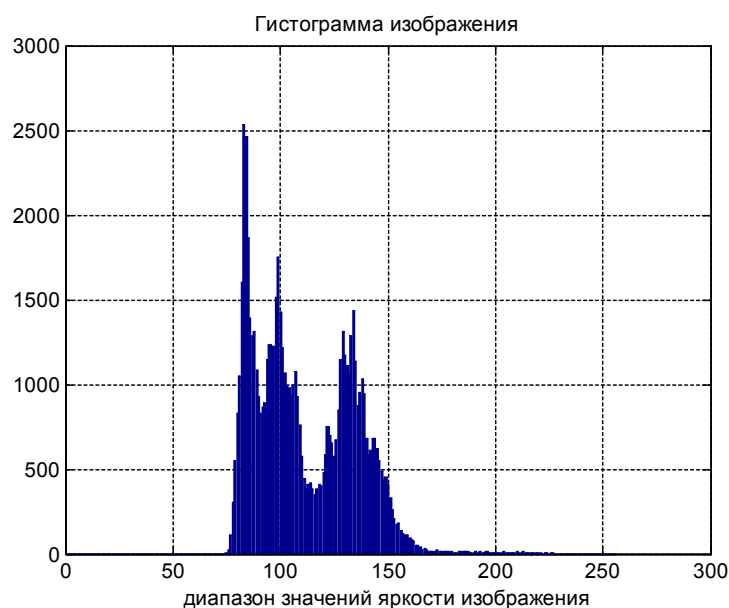


Рисунок 2.8 — Гістограма вхідного зображення

```

%Эквалізація гістограми
F_e=histeq(F,252);
whos F_e, %Вивдає основну інформацію про зображення
HFe=imhist(F_e);
figure, bar(HFe),grid on,set(gca,'FontName','Arial Cyr','FontSize',10),
title('Гистограмма перетвореного зображення'),xlabel('діапазон значень
яскравості зображення') ,

```



Рисунок 2.9 — Гістограма вихідного зображення після еквалізації.

```

figure,imshow(F_e),set(gca,'FontName','Arial Cyr','FontSize',10), title('
зображення рівномірно перетвореної гістограмою') ,
figure,
subplot(1,2,1), bar(HF),grid on,set(gca,'FontName','Arial Cyr','FontSize',10),title('
гістограма вихідні зображення') ,
subplot(1,2,2),bar(HFe),grid on,set(gca,'FontName','Arial
Cyr','FontSize',10),title(' гистограмма перетвореної зображення ') ,
%Використання функції subimage для відображення декількох зображень
figure,
subplot(1,2,1), subimage(F),set(gca,'FontName','Arial
Cyr','FontSize',10),title('вихідне зображення') ,
subplot(1,2,2), subimage(F_e),set(gca,'FontName','Arial Cyr','FontSize',10),title('
зображення перетвореної гістограмою ') ,

```

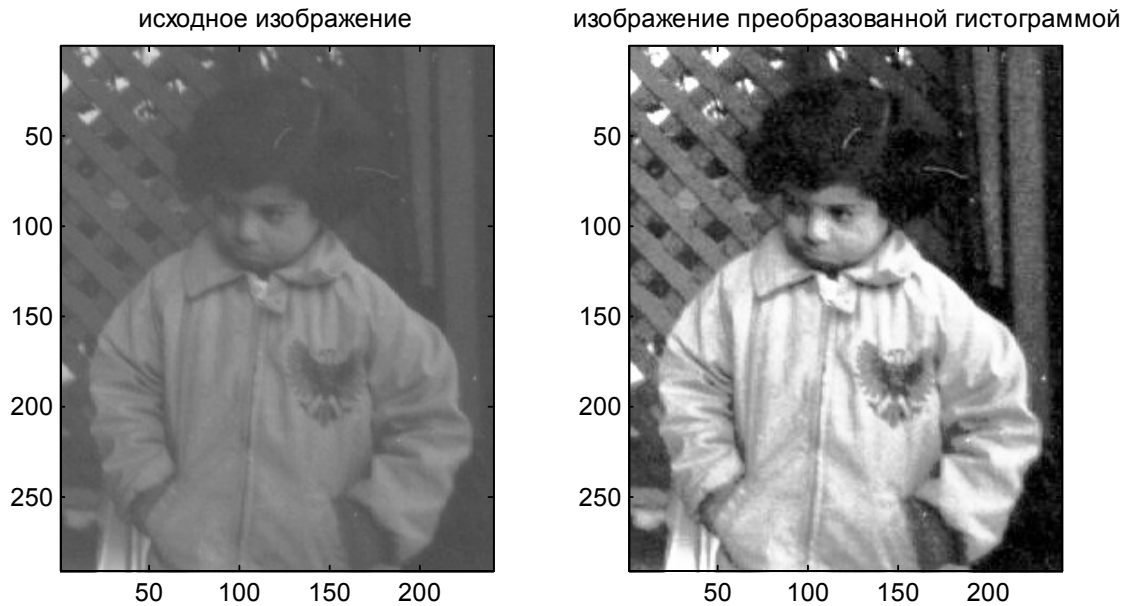


Рисунок 2.10 — Вхідного зображення й зображення після еквалізації.

### Порядок проведення лабораторної роботи

1) Створити й налагодити програму, що вирішує наступні завдання:

а) завантажити зображення 1 за допомогою функції `imread`. Якщо з'ясовано, що зображення кольорове (складається із трьох компонентів), перетворити його в напівтонове за допомогою функції `rgb2gray`;

б) за допомогою функції `size` визначити піксельний розмір зображення у вигляді вектора  $[M, N]$ ;

в) визначити максимальне  $I_{max}$  і мінімальне  $I_{min}$  значення яркостей пікселів зображення й обчислити контраст  $DO$ ;

д) вивести завантажене зображення в графічне вікно за допомогою функції `imshow`, при цьому сформувати пояснювальні написи: **Original**,  $Mx$ ,  $I_{max}=xxx(r1,c1)$ ,  $I_{min}=xxx(r2,c2)$ ,  $DO=xxx$ . Для того, щоб не змінювати вихідне зображення при виводі на екран варто використати функцію `imshow` без параметрів;

е) використовуючи функцію `imadjust` змінити контраст зображення, перетворивши мінімальне й максимальне значення яскравості до значень, що відповідають варіанту завдання. Параметр `gamma` вибрати виходячи з умов одержання найкращої інформативності;

ж) визначити максимальне  $I_{max}$  і мінімальне  $I_{min}$  значення яркостей пікселів і контраст  $DO$  обробленого зображення;

з) вивести отримане зображення в графічне вікно за допомогою функції `imshow`, при цьому сформувати пояснювальні написи: **Rezult1**,  $Mx$ ,  $I_{max}=xxx(r1,c1)$ ,  $I_{min}=xxx(r2,c2)$ ,  $DO=xxx$ ;



- и) перетворити отримане зображення в негативне;
- к) визначити максимальне *I<sub>max</sub>* і мінімальне *I<sub>min</sub>* значення яркостей пікселів і контраст *Д<sub>0</sub>* обробленого зображення;
- л) вивести отримане зображення в графічне вікно за допомогою функції `imshow`, при цьому сформувавши пояснювальні написи: *Negative, Mx, I<sub>max</sub>=xxx(r1,c1), I<sub>min</sub>=xxx(r2,c2), Д<sub>0</sub>=xxx*.
- м) зберегти отримані зображення на диск.

2) Створити й налагодити програму, що вирішує наступні завдання:

- а) завантажити зображення 2, зазначене в таблиці з варіантами завдань;
- б) побудувати гістограму розподілу яркостей вихідного зображення;
- в) вивести вихідне зображення і його гістограму в одне вікно екрана розміром 1x2, сформувавши написи, що *пояснюють*, Original;
- г) виконати еквалізацію гістограми вихідного зображення;
- д) вивести перетворене зображення і його гістограму в одне вікно екрана розміром 1x2, сформувавши написи, що *пояснюють*, Rezalt;
- е) побудувати й вивести у вікно на екрані характеристику перетворення гістограми (послідовність команд для виконання завдання е) оформити у вигляді М-функції);
- ж) вивести всі отримані зображення й гістограми в одне вікно, розміром 2x2, сформувавши написи, що *пояснюють*;
- з) зберегти всі отримані зображення у файл на диску.

**Варіанти завдання**

Варіант	Файл зображення
1	Portrait_w.bmp; YP.bmp; Lenna_(test_image).bmp
2	face.gif; Yfq_0.5-4.43-5.5.bmp; pout.jpg
3	clown.gif; boat.bmp; palzt.jpg
4	Portrait_w.bmp; pollen.tif; FR.bmp
5	Lenna_(test_image).bmp; gory.jpg; H_h.bmp
6	zelda.bmp; valley.bmp; 31.bmp
7	Portrait_w.bmp; pollen.tif; H_h.bmp
8	clown.gif; boat.bmp; palzt.jpg

**Порядок оформлення звіту та його представлення й захист.**

Результати проведеної лабораторної роботи оформлюються протоколом. Він повинен містити такі дані:

- 1) тема практичної роботи;
- 2) позначка практичної роботи;
- 3) короткі теоретичні відомості;
- 4) лістинги всіх програм з обов'язковими коментарями;

- 5) отримані на шкірному етапі роботи зображення;
- 6) висновки.

### **Запитання для самоперевірки:**

1. За допомогою якої команди можна здійснити завантаження зображення? Опишіть синтаксис цієї команди.
2. Яким образом вивести кілька зображень у графічне вікно MATLAB?
3. За допомогою яких команд можна одержати інформацію про завантажене зображення?
4. Типи зображень в MATLAB. Напівтонові й двійкові зображення.
5. Поясніть необхідність і суть градаційних перетворень зображень.
6. Дайте визначення яскравості й контрасту зображення.
7. Для чого служить функція *imadjust*? Який синтаксис вона має?
8. Як впливає на сприйняття зображень гамма-характеристика?
9. Якими способами можна одержати негатив вихідного зображення? Опишіть їх.
10. Що таке гістограма яскравості? Використовуючи яку функцію можна її одержати?
11. Нормована гістограма й способи її одержання.
12. Для чого використовується еквалізація гістограми й у чому вона полягає?

## Лабораторна робота № 3.

### Просторова та частотна фільтрація зображень.

**Мета роботи:** ознайомитись зі способами представлення зображень в області просторових частот; ознайомитись з просторовими методами фільтрації та покращення зображень, ознайомитись з частотними методами фільтрації зображень, використовуючи систему моделювання MATLAB з набором інструментів Image Processing Toolbox; проаналізувати призначення та принцип дії різних фільтрів та методів їх створення в системі MATLAB.

**Завдання на підготовку до лабораторної роботи.** Під година проведення лабораторної роботи студент винний:

**знати:**

- основні характеристики та можливості пакета Signal Processing Toolbox;
- цифрові методи обробки сигналів та зображень;
- будову й роботу програмного пакета MATLAB;

**уміти:**

- способи представлення зображень в області просторових частот та частотними методами фільтрації зображень; ;
- використовувати можливості пакета Signal Processing Toolbox;
- досліджувати різноманітні методи покращення і обробки зображень та з'ясувати їх ефективність для різних типів зображень;
- зберігати та візуалізувати інформацію в програмному пакеті MATLAB.

### Теоретичні відомості.

Лінійна фільтрація зображень може здійснюватися як у просторової, так й у частотній області. При цьому вважається, що «низьким» частотам відповідає основний зміст зображення – фон й великорозмірні об'єкти, а «високим» частотам – дрібно розмірні об'єкти, дрібні деталі великих форм і шумовий компонент. Традиційно для переходу в область просторових частот використовуються методи, засновані на перетворенні Фур'є.

Перетворення Фур'є дозволяє представити практично будь-яку функцію або набір даних у вигляді комбінації таких тригонометричних функцій, як синус і косинус, що дозволяє виявити періодичні компоненти в даних й оцінити їхній внесок у структуру вихідних даних або форму функції.

Виділяють три основні форми перетворення Фур'є: інтегральне перетворення Фур'є, ряди Фур'є й дискретне перетворення Фур'є.

Основна відмінність між трьома формами перетворення Фур'є полягає в

тім, що якщо інтегральне перетворення Фур'є визначено по всій області визначення функції, то ряд і дискретне перетворення Фур'є визначені тільки на дискретній безлічі крапок, нескінченному для ряду Фур'є й кінцевому для дискретного перетворення. Як видно з визначень перетворення Фур'є, найбільший інтерес для систем цифрової обробки сигналів представляє дискретне перетворення Фур'є. Дані, одержувані із цифрових носіїв або джерел інформації, являють собою впорядковані набори чисел, записані у вигляді векторів або матриць.

Використання Фур'є-спектра зображення для виконання операцій фільтрації обумовлено, насамперед, високою продуктивністю таких операцій. Як правило, виконання прямого й зворотного двовимірного перетворення Фур'є й множення на коефіцієнти Фур'є-спектра фільтра займає менше часу, чим виконання двовимірної згортки вихідного зображення.

Алгоритми фільтрації в частотній області ґрунтуються на теоремі про згортку. У двовимірному випадку перетворення згортки виглядає в такий спосіб:

$$G(u, v) = F(u, v) H(u, v),$$

де  $G(u, v)$  - Фур'є-спектр результату згортки (відфільтрованого зображення),  $H(u, v)$  - Фур'є-спектр фільтра  $F(u, v)$  - Фур'є образ вихідного зображення. Тобто в частотній області двовимірна згортка заміняється заелементним перемножуванням образів вихідного зображення й відповідного фільтра.

### Лінійна фільтрація в просторовій області

При операції фільтрації в просторовій області значення пікселів вихідного зображення обчислюються шляхом застосування деякого алгоритму, що використовує значення розглянутого пікселу й пікселів у його околиці й фільтра. Лінійна фільтрація це операція, при якій значення пікселю на виході являє собою лінійну комбінацію значень пікселів в околиці вхідного пікселу.

Для виконання лінійної просторової фільтрації використовують метод двовимірної просторової згортки локальної околиці оброблюваного елемента з лінійним оператором, що називається маскою або матрицею коефіцієнтів фільтра.

Алгоритм згортки полягає в тім, що маска сканує вихідне зображення. На кожному кроці перебуває сума добутків елементів маски й відповідних елементів вихідного зображення, і знайдене значення

привласнюється одному елементу результуючого зображення. Досягши, таким чином, кінця рядка, маска зміщується на один рядок униз, у початок рядка, і процес повторюється.

Є дві тісно зв'язані концепції, які необхідно розуміти при здійсненні лінійної просторової фільтрації. Перша – це кореляція, а друга – згортка. Кореляція складається в проходженні маски по зображенню. З погляду механіки процесу, згортка робиться так само, але маску треба повернути на  $180^\circ$  перед проходженням по зображенню. Якщо звинута маска, що, є симетричної, то кореляція й згортка дають однакові результати.

Прямокутна область розміром  $m \times n$ , на якій задана імпульсна характеристика  $w(s,t)$ , називається маскою або ядром фільтра. Розглянемо, як співвідносяться координати зображення й імпульсної характеристики фільтра на прикладі.

Відгук фільтра  $g(y, x)$  обчислюється як сума добутків отсчетов зображення на відповідні отсчети поверненої імпульсної характеристики. Ця операція виконується для кожного відліку зображення.

Якщо імпульсна характеристика фільтра симетрична, тобто  $w(s,t)=w(-s,-t)$ , те замість згортки можна виконувати кореляцію:

$$g(x, y) = f(x, y) \otimes w(x, y) = \sum_{t=-m/2}^{m/2} \sum_{s=-n/2}^{n/2} f(x+s, y+t) \cdot w(s, t)$$

Ця операція не вимагає відбиття імпульсної характеристики, а являє собою обчислення в ковзній по зображенню масці фільтра добутків отсчетов зображення на відповідні коефіцієнти фільтра і їхнє підсумовування (відповідно до рисунку 3.1)

Для обробки крайових ефектів необхідно збільшити зображення по рядках і стовпцям на  $n/2$  отсчетов ліворуч і праворуч і на  $m/2$  отсчетов зверху й знизу. При цьому розширення може виконуватися різними способами: доповненням нулями, повторенням граничних елементів, періодичним повторенням  $(x \bmod N, y \bmod M)$  або дзеркальним відбиттям граничних елементів. Розмір вихідного зображення при цьому зберігається рівним розміру вхідного зображення.

Нехай зображення  $f(x,y)$  і маска  $w$  мають такий вид, як показано на наступному рисунку 1 (а). Початок зображення перебуває у верхньому лівому куті матриці  $f(x,y)$ . Для здійснення кореляції треба помістити праву нижню крапку  $w(x,y)$  так, щоб вона збіглася з початком  $f(x,y)$ . Оскільки є крапки, які не перекриваються, то думаємо, що функція  $f$  дорівнює нулю скрізь, де це необхідно, щоб гарантувати наявність відповідних крапок на  $f$  при проходженні над нею маски  $w$  (рис.1 б). Першим значенням кореляції є сума заелементних добутків двох функцій у положенні на рис.1 в. У цьому випадку сума добутків дорівнює 0. Потім

маска  $w$  переміщається на один крок вправо й процес обчислення повторюється. Дійшовши до правого краю, маска зрушується на один пікселів униз і повертається в крайнє ліве положення. Перші 16 кроків будемо одержувати нулі, які містяться в крапку розширеного зображення, що збігає із центром маски.

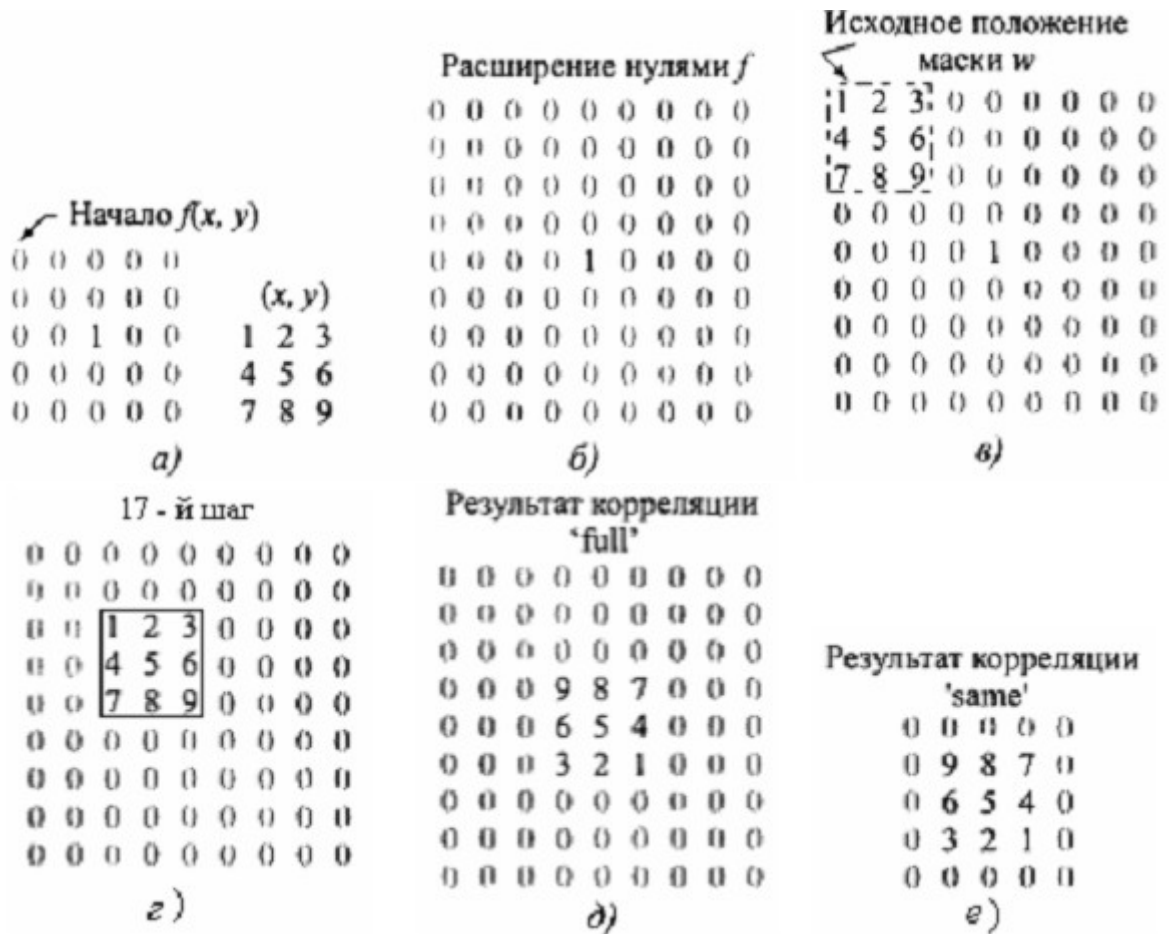


Рисунок 3.1 – Алгоритм лінійної просторової фільтрації

На 17 – м кроці дев'ятка (права нижня крапка маски  $w$ ) збіжиться з одиницею (див. рис.1 г) і ми одержимо перше ненульове значення  $9 \times 1 = 9$  кореляції. Продовжуємо цей процес доти, поки  $w$  повністю не пройде  $f$ . Ми одержимо результат, показаний на рис.1 е.

Таким чином, для здійснення кореляції треба переміщати  $w(x,y)$  по всіх можливих положеннях, так щоб хоч один пікселів маски перекривався з пікселями зображення  $f(x,y)$ . Останнє обчислення виробляється, коли права нижня крапка маски сполучається із правою нижньою крапкою розширеного зображення  $f$ . Відзначимо, що якби ми зафіксували  $w$ , а переміщали б  $f$ , те результат був би іншим, так що порядок операції тут має значення.

Мітка '**full**' над результатом кореляції на мал.1 д є прапором, що пропонує застосування кореляції з розширенням зображення, продемонстрованим вище методом. Можна використати й іншу опцію,

'same' (ри.1 е), коли обчислюється кореляція, розмір якої збігається з розміром вихідного зображення  $f$ . Ці обчислення також використовують розширення нулями, але при цьому початкова позиція центральної крапки маски (це крапка з оцінкою 5) сполучається з початком  $f$ . Останнє обчислення виробляється, коли центральна крапка маски сполучається із правою нижньою крапкою  $f$  (нерозширеного). У результуючому зображенні додані нулі вбираються, і результат збігається по розмірах з нерозширеним вихідним зображенням.

Для здійснення згортки необхідно спочатку повернути  $w(x,y)$  на  $180^\circ$  навколо центра й зробити ту ж процедуру, що й при обчисленні кореляції. Вихідне розширене зображення й положення на ньому поверненої маски при прапорі 'full' показано на наступному малюнку ліворуч, у середині показане вихідне положення маски при прапорі 'same', результат згортки із прапором 'same' показаний праворуч.

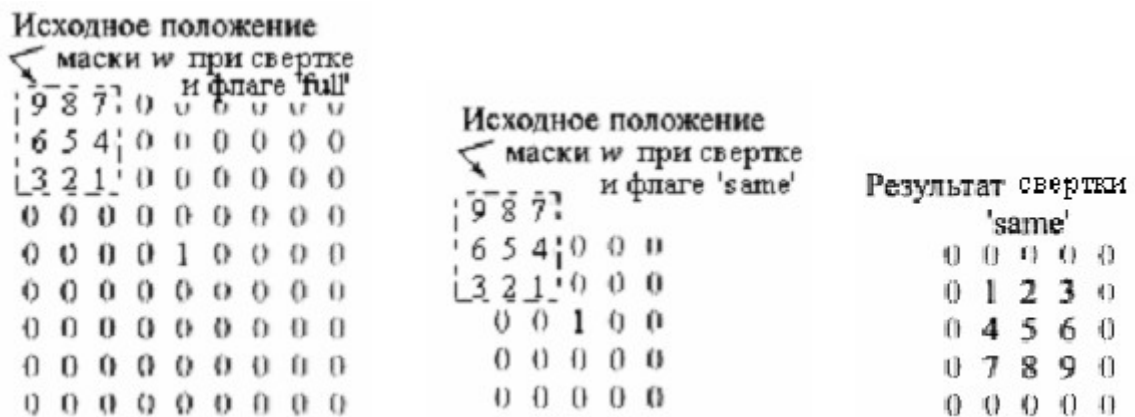


Рисунок 3.2 – Алгоритм лінійної просторової фільтрації як згортка.

Згортка дає однаковий результат незалежно від того, яка із двох функцій піддається переміщенню. У пакеті ІРТ при реалізації процедур згортки й кореляції завжди переміщається фільтруюча маска. Помітимо також, що результати кореляції й згортки виходять друг із друга поворотом на  $180^\circ$ .

Функція  $f$  на попередніх малюнках була дискретним одиничним імпульсом, тобто вона дорівнює  $1$  тільки в одній крапці, а у всіх інших вона дорівнює  $0$ . З наведених малюнків ясно видно, що згортка просто «копіює» маску  $w$  у те місце, де був одиничний імпульс. Це проста властивість копіювання є фундаментальною концепцією теорії лінійних систем і пояснює необхідність повороту однієї з функцій на  $180^\circ$  при виконанні операції згортки. Якщо маска є симетричною, то кореляція й згортка дають однакові результати.

У пакеті ІРТ лінійна просторова фільтрація реалізована функцією *imfilter*, що має наступний синтаксис:

$g = \text{imfilter}(f, w, \text{filtering\_mode}, \text{boundary\_options}, \text{size\_options})$ ; де  $f$  — це вхідне зображення,  $w$  — фільтруюча маска,  $g$  — результат фільтрації. Параметр **filtering\_mode** (мода фільтрації) визначає, що робить фільтр, кореляцію ('*corr*') або згортку ('*conv*'). Опція **boundary\_options** (спосіб розширення границь) відповідає за розширення границь, причому розміри розширення визначаються розмірами фільтра. Цей параметр приймає числове значення, яким варто розширити зображення. За замовчуванням цей параметр дорівнює нулю (без апострофів). Він може також приймати наступні текстові значення:

'**replicate**' Розмір зображення збільшується повторенням величин на його бічних границях

'**symmetric**' Розмір зображення збільшується шляхом дзеркального відбиття через границі

'**circular**' Розмір зображення збільшується періодичним повторенням двовимірної функції

Опція **size\_options** (опція розміру) — це або '**full**', або '**same**', зміст яких уже порозумівався раніше. При опції '**full**', вихід має розміри розширеного зображення, а при опції '**same**' вихід має ті ж розміри, що й вхід (опція за замовчуванням).

Найчастіше функція **imfilter** застосовується у вигляді команди  $g = \text{imfilter}(f, w, \text{'replicate'})$ ; Ця форма команди використовується при реалізації в IPT стандартних лінійних просторових фільтрів.

Методику просторової фільтрації проілюструємо прикладами.

**Приклад.** Вихідне зображення класу uint8, показане на мал. 3

```
f=imread('C:\Documents and Settings\pic\ipexcheckerboard_thumbnail.png');%  
завантаження зображення в робочу область Matlab  
whos f, %Виділення основну інформацію про зображення  
f1=f(1:32,1:32);  
figure, imshow(f1,'InitialMagnification','fit') ,title('Оригінал '),  
impixelregion,  
% Функція impixelregion. Засоби перегляду локального масиву пікселів  
відкривають окреме вікно, що містить значення пікселів деякої локальної  
околиці даного зображення.  
% InitialMagnification - Управляє початковим збільшенням зображення,  
% 'fit' масштабує зображення так, щоб воно вписувалося у вікно цілком.  
w = ones(5)/5^2; %% імпульсна характеристика фільтра, що фільтрує  
маска  
gd = imfilter(f1,w);% лінійна просторова фільтрація  
figure, imshow(gd, 'InitialMagnification','fit'); title('просторова фільтрація '),  
impixelregion,
```



```
gd_s = imfilter(f1, w, 'symmetric');%Розмір зображення збільшується
повторенням величин на його бічних границях
figure, imshow(gd_s, 'InitialMagnification','fit'); title('просторова фільтрація
symmetric '),
impixelregion,
```

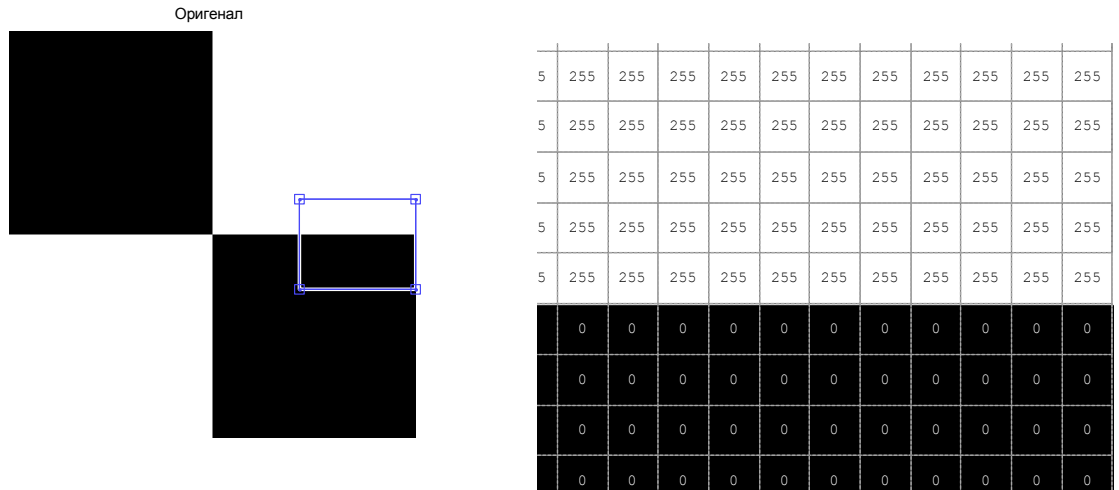


Рисунок 3.3 – Вихідне зображення й засіб перегляду, що містить значення пікселів деякої локальної околиці даного зображення.

Нижче показаний результат застосування до нього простого фільтра розміру  $5 \times 5$ , що пропорційний фільтру усереднення. Фільтр усереднення вийде, якщо одиниці розділити на  $5^2$ . був би іншим. Згортка з фільтром  $w$  дає ефект розмиття вихідного зображення. Оскільки фільтр симетричний, ми могли використати моду кореляції, задану в *imfilter* за замовчуванням. Тут також використалася гранична опція, прийнята за замовчуванням, тобто розширення зображення нулями (чорними кольорами). Як й очікувалося, границі між білими й чорними областями стали розмитими. Той же ефект мав місце на границях зображення, до яких примикають білі області, що цілком зрозуміло в силу розширення границь чорними кольорами.



Рисунок 3.4 – Просторова фільтрація вихідне зображення з розширенням зображення нулями й засіб перегляду локальної околиці

З розмиття границь можна впоратися, якщо використати опцію *'replicate'* (див. слід. рисунок). Видно, що тепер границі фільтрованого зображення виглядають так, як й очікувалося. Еквівалентний результат виходить із опцією *'symmetric'*. Однак, якщо застосувати опцію *'circular'*, те одержимо такий же результат як на попередньому малюнку праворуч. Виявляється та ж проблема, що й при розширенні нулями. Цьому не варто дивуватися, адже при періодичному повторенні вихідного зображення чорні квадрати примикають до білого.

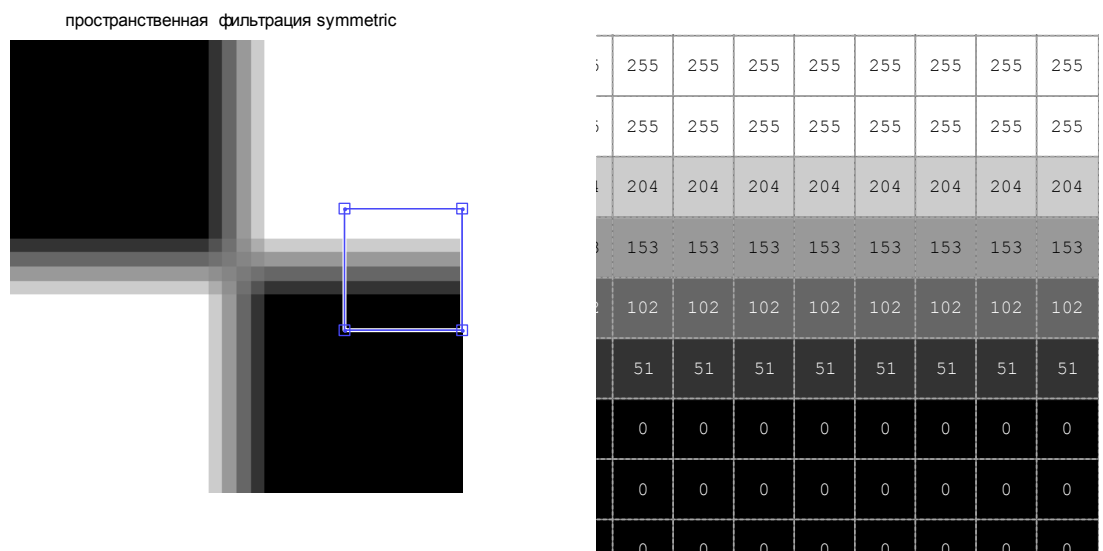


Рисунок 3.5 – Просторова фільтрація вихідне зображення з розширенням зображення повторенням величин на його бічних границях і засіб перегляду локальної околиці

Фільтр  $w$ , що ми використали в попередньому прикладі, створений у вигляді квадратної матриці. Для виконання операцій фільтрації в MatLab є кілька стандартних просторових фільтрів. Їх можна одержати з функції *fspecial*, що генерує маску фільтр  $w$  при виконанні команди  $w=fspecial('type', parameters)$ , де *'type'* позначає тип фільтра, а в аргументах *parameters* задаються параметри обраного фільтра.

Можливі типи фільтрів:

- *'average'* усереднюючий фільтр

$h=fspecial('average', hsize)$

*hsize* може бути вектором, що задає число рядків і стовпців в *h* або скаляром, у цьому випадку *h* є квадратною матрицею. За замовчуванням *hsize* матриця [3 3].

- *'disk'* круглий усереднюючий фільтр

*h=fspecial('disk',radius)*

повертає круговий фільтр усереднення, уписаний у квадратну матрицю зі стороною  $2*radius+1$ . За замовчуванням *radius* дорівнює 5.

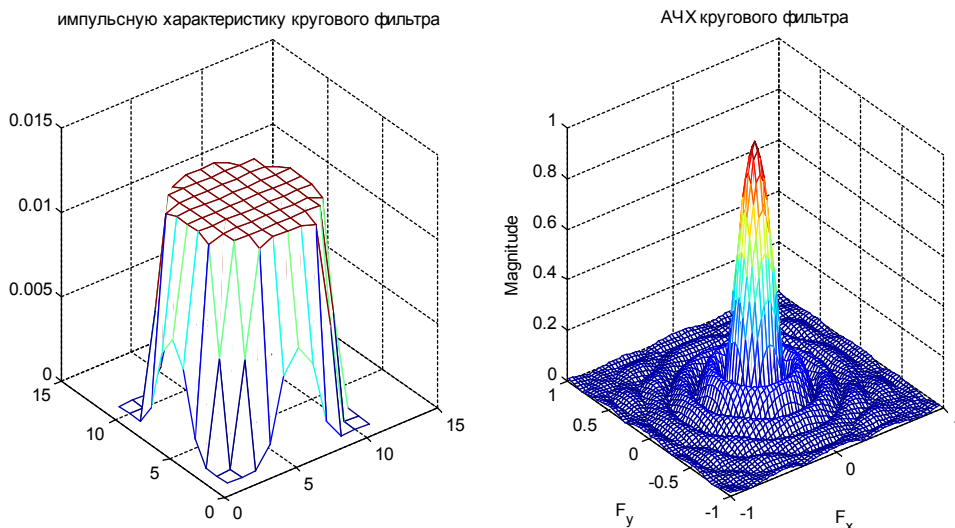


Рисунок 3.6 – Імпульсна характеристика й амплітудно-частотна характеристика кругового фільтра задана функцією *fspecial*, що генерує маску фільтра.

- *'gaussian'* Gaussian низькочастотний фільтр

*h=fspecial('gaussian',hsize,sigma)*

повертає гауссовий фільтр розміру *hsize* зі стандартними відхиленням *sigma* (позитивний). *hsize* може бути вектор, що задає число рядків і стовпців в *h* або скаляр. За замовчуванням *hsize* дорівнює [3 3], а *sigma* дорівнює 0.5.

*%Функція h\_g=fspecial('gaussian',n,0.8) повертає маску (імпульсну характеристику) h фільтра нижніх частот Гаусса.*

*% Розмір маски визначає параметр n.*

*n=9;*

*h\_g=fspecial('gaussian',n,0.8);*

*figure,*

*subplot(1,2,1), mesh(h\_g),title('імпульсну характеристику фільтра Гаусса'),*

*subplot(1,2,2),freqz2(h\_g),title('АЧХ фільтри Гаусса'),%Функція обчислення двовимірної АЧХ*

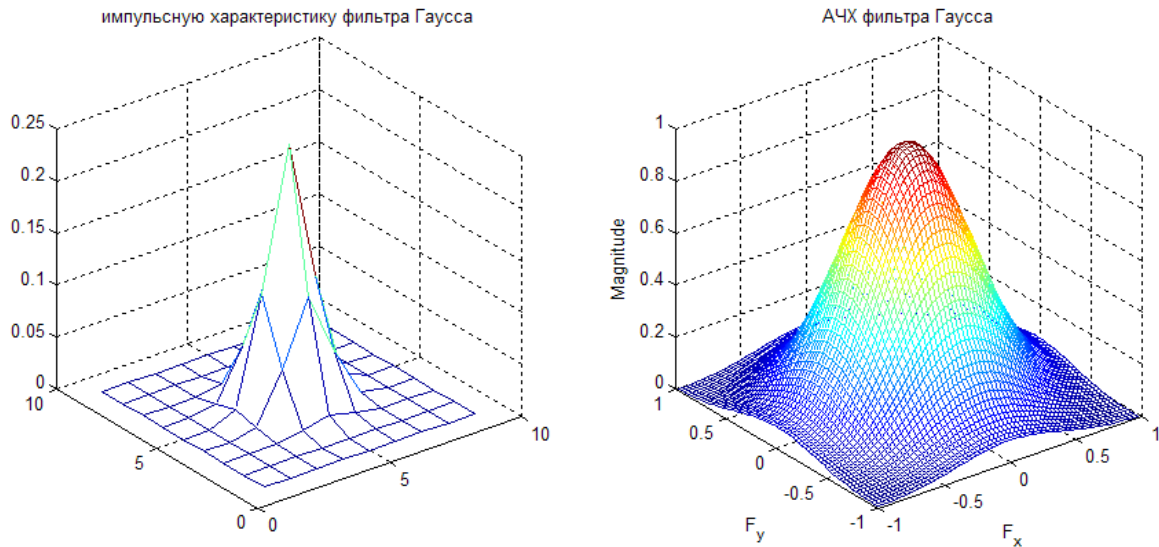


Рисунок 3.7 – Імпульсна характеристика й амплітудно-частотна характеристика фільтра Гаусса задана функцією *fspecial*, що генерує маску фільтра.

- *'laplacian'* фільтр, що апроксимує 2D Laplacian оператор

$h = \text{fspecial}('laplacian', \alpha)$

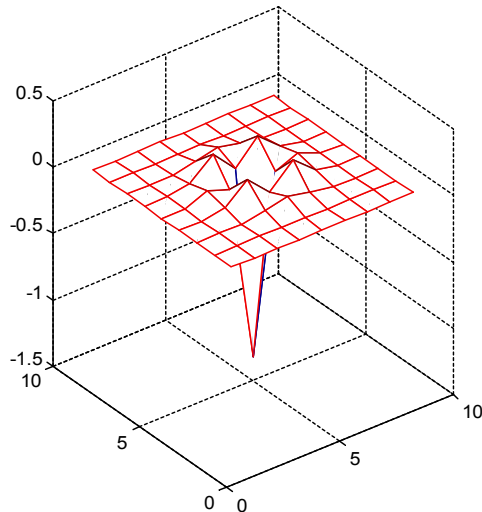
повертає 3 на 3 фільтр, що апроксимує форму двовимірного оператора Лапласа. Параметр *alpha* контролює форму Лапласа й повинен бути в діапазоні від **0,0** до **1,0**. За замовчуванням *alpha* **0.2**.

- *'log'* фільтр Laplacian Gaussian

$h = \text{fspecial}('log', \text{hsize}, \text{sigma})$

повертає осесиметричний фільтр розміром *hsize* зі стандартним відхиленням *sigma* (позитивний). *hsize* може бути вектор, що задає число рядків і стовпців в *h* або скаляром. За замовчуванням це *hsize* **[5 5]**, а *sigma* дорівнює **0.5**.

импульсную характеристику фільтра Лапласа-Гаусса



АЧХ фільтра Лапласа-Гаусса

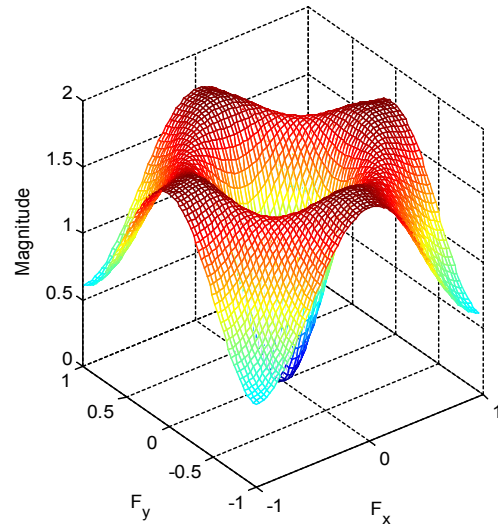


Рисунок 3.8 – Імпульсна характеристика й амплітудно-частотна характеристика фільтра Лапласа-Гаусса задана функцією *fspecial*, що генерує маску фільтра.

- **'sobel'** Sobel фільтр виділення горизонтальних границь,  $h=[1\ 2\ 1;0\ 0\ 0;-1\ -2\ -1]$ . Для виділення вертикальних границь використовується транспонований фільтр  $h^*$ .

$h=fspecial('sobel')$

- **'unsharp'** фільтр нерізкого маскуваня

$h=fspecial('unsharp',alpha)$

повертає 3 на 3 фільтр згладжування. *fspecial* створює фільтр із негативного фільтра лапласіана з параметром *alpha*. *alpha* задає форму фільтра й повинна бути в діапазоні від 0,0 до 1,0. За замовчуванням *alpha* дорівнює 0.2.

$h_u = \begin{bmatrix} -0.1667 & -0.6667 & -0.1667 \\ -0.6667 & 4.3333 & -0.6667 \\ -0.1667 & -0.6667 & -0.1667 \end{bmatrix}$

імпульсну характеристику фільтр нерезкого маскування

АЧХ кругового фільтр нерезкого маскування

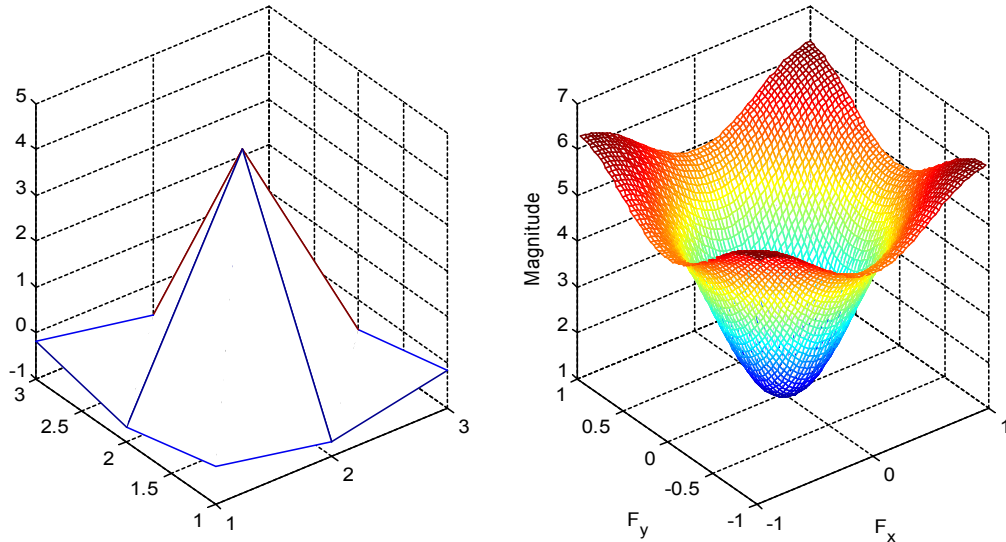


Рисунок 3.9 – Імпульсна характеристика й амплітудно-частотна характеристика фільтра нерезкого маскування задана функцією *fspecial*, що генерує маску фільтра.

Після створення фільтра з *fspecial* фільтрація виконується функцією *imfilter*.

**Приклад** ілюструє застосування фільтра нерезкого маскування для чорно-білого зображення. Фільтр нерезкого маскування підкреслює краї й дрібні деталі в зображенні, роблячи їх більше чіткими.

```
I=imread('moon.jpg');  
h=fspecial('unsharp',0.2);  
I2=imfilter(I,h);  
figure,  
subplot(1,2,1), imshow(I),title('вихідне зображення');  
subplot(1,2,2), imshow(I2), title('Фільтрація unsharp') ,
```

исходное изображение



Фильтрация unsharp



Рисунок 3.10 – Підвищення чіткості зображення застосування фільтра нерізкого маскування задана функцією *fspecial('unsharp')*.

### **Просторова лінійна фільтрація зображень. Просторові методи покращення зображень.**

**Відновлення зображення** – одне із завдань обробки зображень. Метою відновлення є реконструкція зображення, що раніше було перекручено або зіпсоване процесами, інформація про які апріорі відома. Тому методи відновлення засновані на моделюванні процесів перекручування й застосування зворотних процесів для одержання вихідних зображень.

#### *Шуми і їхні класи*

Шум є однією з основних причин зображень, що приводять до перекручування. Основні джерела шуму на цифровому зображенні пов'язані із процесами формування зображення в датчику й у пристрої відображення, а також із процесами пов'язаними з перетворенням і передачею зображення по каналах передачі. Значення просторового шуму є випадковими величинами, які, проте, можна описати математично.

Існують кілька важливих типів шумів, кожний з яких має специфічні просторові характеристики. Основною характеристикою шуму є функція щільності розподілу ймовірності (PDF, Probability Density Function):

- Гауссов (нормальний) шум;
- шум Релея;
- шум Эрланга (гама шум);
- експонентний шум;
- рівномірний шум;
- імпульсний шум.

У системі MATLAB (Image Processing Toolbox) існує можливість формування й накладення на зображення шумів. Для цього використовується убудована функція *imnoise*, що призначена для додавання в зображення певного типу шуму із заданими характеристиками. За допомогою цієї функції можна сформувати й накласти на зображення три типи шуму:

- 1) *gaussian* – гауссовий білий шум (нормальний). Має рівномірний спектр у всій смузі просторових частот і проявляється на зображенні у вигляді характерних сторонніх вкраплень;
- 2) *salt & pepper* – імпульсний шум, що проявляється у вигляді включених або виключених пікселів;
- 3) *speckle* – мультиплікативний шум, що вводиться в зображення в результаті перемножування зображення й рівномірного шуму з нульовим середнім значенням.

Функція *imnoise* має наступний синтаксис:

$g = \text{imnoise}(f, \text{type}, \text{parameters}),$

де  $f$  - вихідне зображення,  $\text{type}$  – тип шуму, що задає,  $\text{parameters}$  - аргументи, які

дозволяють задати додатково параметри шуму. Наприклад:  $g = \text{imnoise}(f, \text{'gaussian'}, m, \text{var})$  додає до зображення  $f$  гауссовий білий шум із середнім значенням  $m$  і відхиленням (дисперсією)  $\text{var}$  (за замовчуванням  $m=0$ ,  $\text{var}=0.01$ ).

$g = \text{imnoise}(f, \text{'salt \& pepper'}, d)$ , де  $f$  - вихідне зображення,  $d$  – щільність шуму. Додає імпульсний шум у вигляді чорних і білих пікселів. Щільність шуму  $d$  приблизно дорівнює відсотку зображення, ушкодженого шумом. Число шумових пікселів можна оцінити по формулі  $d * \text{numel}(f)$ . За замовчуванням  $d=0.05$ .

$g = \text{imnoise}(f, \text{'speckle'}, \text{var})$  додає до зображення  $f$  мультиплікативний шум по формулі  $g = f + n * f$ , де  $n$  – рівномірно розподілений шум із середнім значенням рівним  $0$  і відхиленням (дисперсією)  $\text{var}$  (за замовчуванням  $\text{var}=0.04$ ).

*Придушення шуму методами просторової фільтрації*

Процеси при фільтрації в просторовій області можна позначити рівнянням

$$g(x, y) = T[f(x, y)],$$

де  $f(x, y)$  – вхідне зображення,  $g(x, y)$  – вихідне (оброблене) зображення, а  $T$  – деякий оператор (перетворення) над  $f$ , що визначений у деякій околиці крапки з координатами  $(x, y)$ . Як околиця крапки з координатами  $(x, y)$  використовується квадратна або прямокутна область із центром у крапці  $(x, y)$ , що найчастіше називають маскою (рис. 3.11).



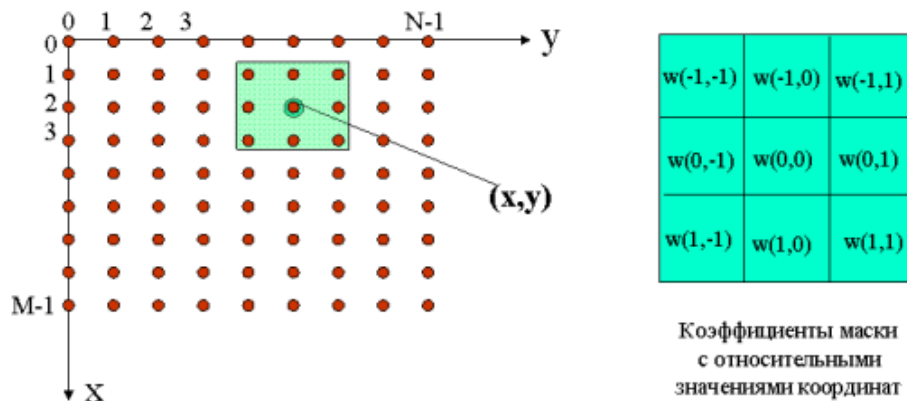


Рисунок 3.11 - Принцип просторової фільтрації

**Процес фільтрації заснований на переміщенні маски фільтра від крапки до крапки вихідного зображення й складається з послідовності наступних дій:**

- 1) визначення центральної крапки з координатами  $(x, y)$ ;
- 2) здійснення лінійної або нелінійної операції, що використовує значення пікселів, що покривають маскою в околиці навколо центральної крапки  $(x, y)$ ;
- 3) призначенні результату цієї операції «відгуком» у цій крапці;
- 4) повторення всього процесу для кожної крапки зображення.

Відгук фільтра по масці  $m \times n$  визначається вираженням

$$R = w_1 Z_1 + w_2 Z_2 + \dots + w_{mn} Z_{mn} = \sum_{i=1}^{mn} w_i Z_i$$

де  $w_i$  – коефіцієнти маски;  $Z_i$  -значення яскравості пікселів, що покривають маскою;  $m, n$  – розміри маски.

Для придушення високочастотних шумів, які проявляються у вигляді крапок з розмірами порівнянними з розмірами пікселів зображення, часто використовують однорідний усереднюючий фільтр, маска якого, має розміри  $3 \times 3$  ( функція яка формує маску  $3 \times 3$  -  $w = \text{fspecial}('average', [3 \ 3])$  )

. Відгук, формований таким фільтром буде дорівнює

$$R = \frac{1}{9} \sum_{i=1}^9 Z_i$$

Звідси треба, що на кожному кроці обробки яскравість центрального пікселя під маскою буде замінятися середнім значенням яскравості пікселів, що покривають маскою. Оскільки яскравість пікселів, що містять шум характерна різкими викидами, вона буде замінена на середні значення по околиці, що приведе до зниження шумової складової в зображенні.

Лінійні фільтри, що згладжують. Для зменшення шумів широко застосовуються НЧ фільтри, оскільки шум являє собою ВЧ сигнал. Зокрема, для НЧ фільтрації застосовується усереднення сигналу в масці, наприклад, при  $n=m=3$ :

$$h_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}; h_2 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}; h_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

Нормування необхідна для того, щоб привести значення відгуку фільтра до діапазону вхідних даних. коефіцієнт, Що Нормує, визначається з умови рівності одиниці суми всіх коефіцієнтів КИХ.

### Нелінійна просторова фільтрація. Придушення шумів.

У результаті застосування лінійних фільтрів, що згладжують, відбувається придушення шумів, але одночасно розмиваються границі між областями з різною амплітудою сигналу. Для зменшення «смазу» границь розроблені різні нелінійні фільтри. Як і лінійні КИХ фільтри, нелінійні фільтри працюють у ковзному вікні. Але, при лінійній фільтрації обчислюється лінійна комбінація отсчетов сигналу, а при нелінійній фільтрації виконуються нелінійні перетворення отсчетов сигналу в обумовлену маскою фільтра околиці елементів.

*Сігма-фільтр* призначен для придушення шумів у зображенні зі збереженням контурів (різких границь областей) [1]. Центральний елемент маски заміщається зваженим середнім значенням, обчисленим тільки по тимі амплітудам отсчетов, значення яких попадають в  $\pm k\sigma$  - область щодо яскравості центрального елемента.  $\sigma$  вибирається або як СКО шуму, що придушує, або як СКО в масці, або встановлюється рівної СКО, отриманому по всьому зображенню:

$$g(x, y) = \sum_{s, t \in S} h(s, t) \cdot f(x-s, y-t),$$

де **S-околицю** становлять ті значення координат маски, у яких виконується накладена умова:

$$S = \{(s, t) : |f(x-s, y-t) - f(x, y)| \leq k\sigma\},$$

$h(s, t)$  КИХ лінійного подібного що згладжує фільтрую

При  $k=2$  діапазон замінних значень становить  $\pm \sigma 2$ , у випадку нормального розподілу шуму ймовірність влучення амплітуди за межі діапазону дорівнює 4,55%.

**Приклад** ілюструє застосування лінійних фільтрів, що згладжують, для придушення гауссового білого шуму, але одночасно розмиваються границі між областями з різною амплітудою сигналу. Для зменшення «смазу»

границь розроблені різні нелінійні фільтри, які виконують нелінійні перетворення отсчетов сигналу обумовленою маскою фільтра околиці елементів зображення, *сігма-фільтр*.

```
% шумоподавітель Гаусіва білого шуму усередняючим фільтром і сігма-фільтром.  
close all; %Видалення графічних вікон  
clear  
A=imread('C:\MATLAB\Image\pic\Portrait_w.bmp'); %завантаження картинки в робочу область Matlab  
whos A, %Видає основну інформацію про зображення  
A1=im2double(A)*0.9+0.05;  
whos A1, %Видає основну інформацію про зображення  
figure,imshow(A1),title('Оригінал') ,  
v=0.01; %дисперсія білого шуму  
A_nois=imnoise(A1,'gaussian',0,v);% додаємо гауссовий білий шум, із середнім значенням і дисперсії-v  
figure,imshow(A_nois),title('оригінал+білий шум') ,  
  
%визначення однорідної область обробки n(1)-координата у,  
%n(2)-координата x, для визначення статистичних характеристик шуму  
n=input('n='),  
Ad=A_nois(0+n(1):30+n(1),0+n(2):30+n(2));%виділяємо область обробки.  
d=std2(Ad),% середньоквадратичне відхилення.  
m=mean2(Ad), % середнє значення  
  
% Сігма-фільтр для придушення шумів у зображенні.  
Th=input('Th='),  
% Th=2.5*v^0.5; % задається поріг для завдання маскки усереднення  
%використання сігма-фільтра sig_filt(x,Th) з порогом Th.  
A_sigm=nlfilter(A_nois,[7 7],'sig_filt',Th);  
figure, imshow(A_sigm),title('сигма фільтр') ,  
  
n=[7 7];  
h_a=fspecial('average',n) ;%маска(імпульсну характеристику)фільтра нижніх частот, усереднюючий фільтр  
  
% визначаємо номер рядка, що хочемо відобразити на графіку профілю перетину зображення  
ys=150;  
figure,  
subplot(3,1,1), plot(A_nois(ys,:)),grid on,title(' Перетин зашумленого зображення') ,
```

```

subplot(3,1,2),plot(A_aver(ys,:)),grid on,title('Перетин зображення
усереднюючий фільтр '),
subplot(3,1,3),plot(A_sigm(ys,:)),grid on,title('Перетин зображення сігма-
фільтр '),
figure,
subplot(3,1,1),plot(A_nois(ys,:)),grid on,xlim([180 200]),title('Перетин
зашумленого зображення'),
subplot(3,1,2),plot(A_aver(ys,:)),grid on,xlim([180 200]),title('Перетин
зображення усереднюючий фільтр '),
subplot(3,1,3),plot(A_sigm(ys,:)),grid on,xlim([180 200]),title('Перетин
зображення сігма-фільтр '),

```

```

figure,
subplot(2,2,1),subimage(A1),title('Оригінал'),
subplot(2,2,2),subimage(A_nois),title('оригінал+білий шум'),
subplot(2,2,3),subimage(A_aver),title('усереднюючий фільтр'),
subplot(2,2,4),subimage(A_sigm),title('сігма фільтр'),

```

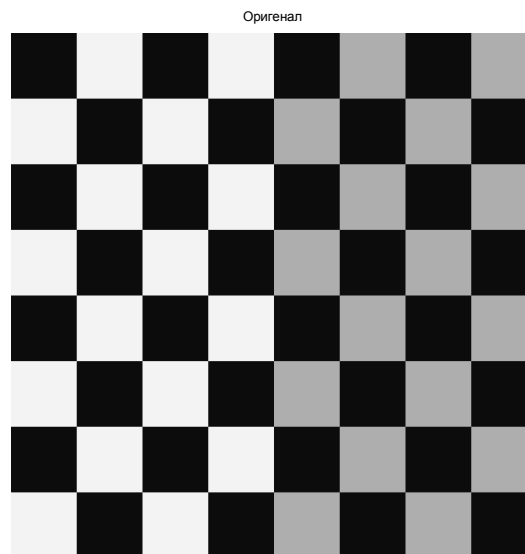


Рисунок 3.12 - Вихідне тестове зображення для порівняльної оцінки лінійного й нелінійного алгоритму згладжування гауссового шуму.

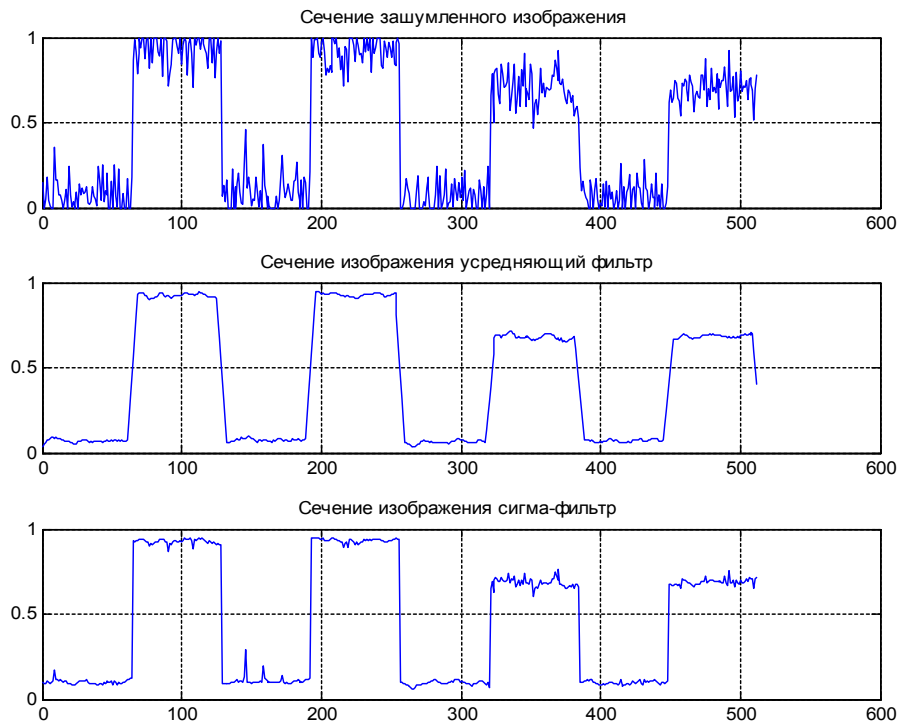


Рисунок 3.13 - Порівняльний аналіз рядка перетину зображення для лінійного й нелінійного згладжування шуму.

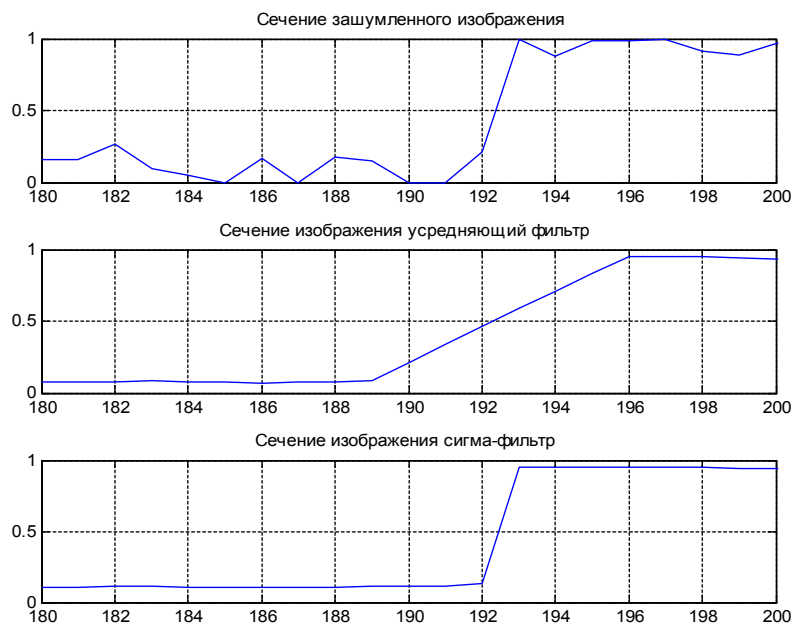


Рисунок 3.14 - Порівняльний аналіз рядка перетину зображення, для оцінки розмиття границь зображення лінійного й нелінійного згладжування шуму.

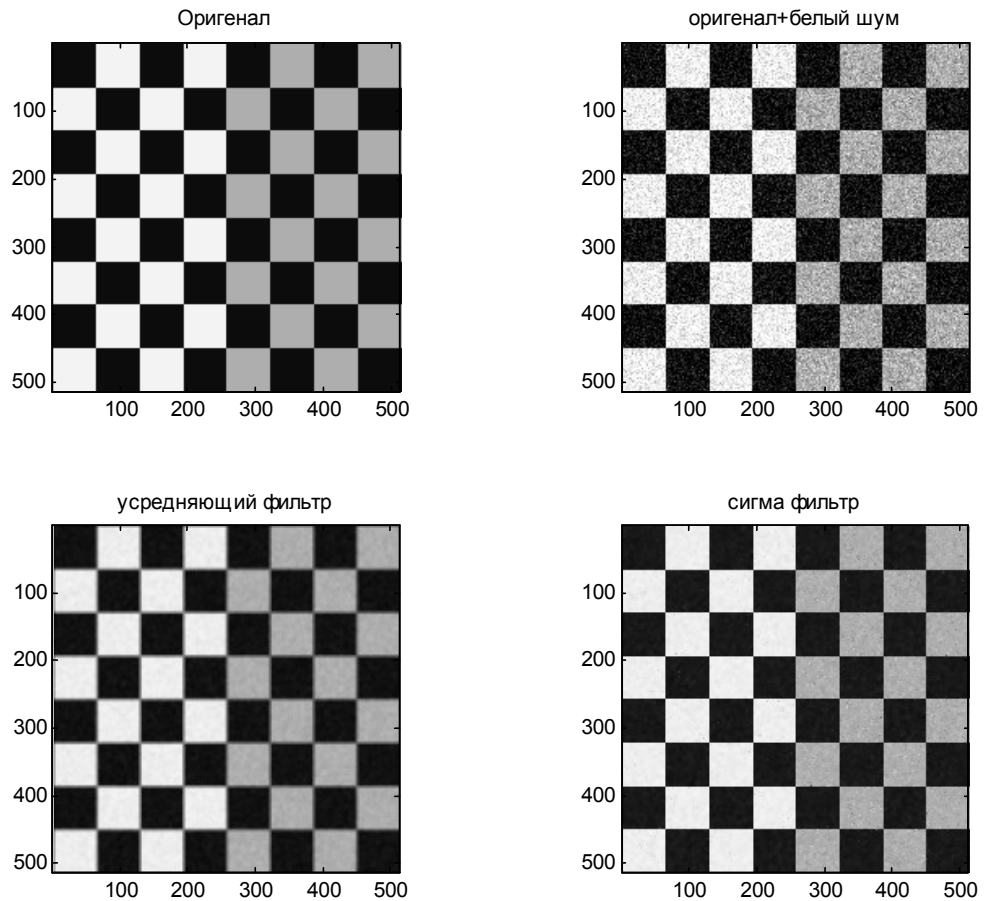


Рисунок 3.15 - Порівняльний аналіз зашумленого зображення, для оцінки лінійного й нелінійного згладжування шуму.

На рис. 15 наведений приклад обробки зображення монтажної плати з додаванням гауссового білого шуму з дисперсією  $\sigma^2=0.01$  при розмаху яскравості 1 і результат видалення шуму лінійним фільтром, що згладжує, по *околиці*  $S=[7\ 7]$  і нелінійного сігма-фільтра по *околиці*  $S=[7\ 7]$ .

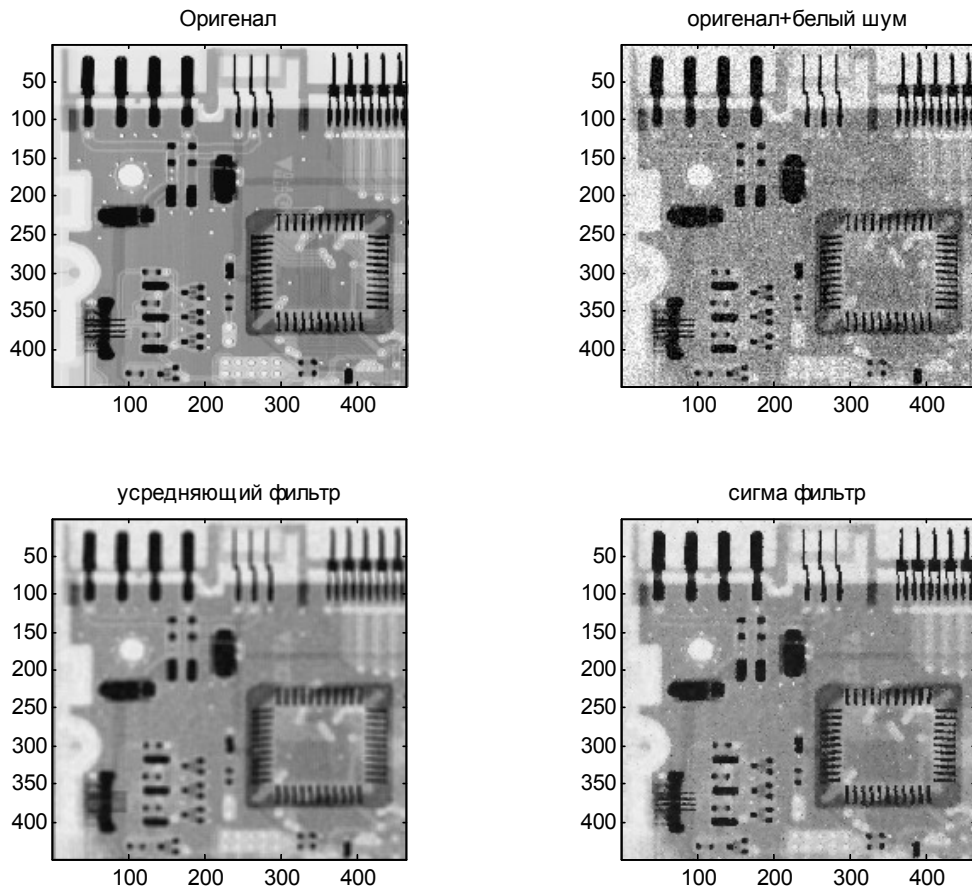


Рисунок 3.16 - Порівняльний аналіз зашумленого реального зображення монтажно́ї плати, для оцінки лінійного й нелінійного згладжування білого шуму

Зображення після усереднюючого фільтра очищено від шуму, але одночасно із цим воно стало занадто розпливчастим і перекрученим (зверніть увагу на те, як виглядають штирьки рознімання вгорі зображення). Зображення після сігма-фільтра шум також відсутній, але воно істотно чіткіше.

### ***Медіанний фільтр***

Фільтри, засновані на порядкових статистиках, також ставляться до нелінійних фільтрів. Найбільш ефективними по сукупності впливів: згладжування шуму на однорідних ділянках зображення, збереження стрибків зміни яскравості, мінімального перекручування форми границі, придушення імпульсного шуму, обчислювальної ефективності є медіанний фільтр.

*Медіанні фільтри.* Найбільш відомим з фільтрів, оснований на порядкових статистиках, є медіанний фільтр. Дія цього фільтру полягає в

заміні значення в точці зображення на медіану значень яскравості в околиці цієї крапки:

$$\hat{g}(x, y) = \underset{(s, t) \in S_{xy}}{\text{med}} \{f(s, t)\}$$

При обчисленні медіани значення в самій точці (тобто в центрі околиці) також ураховується. Широка популярність медіанних фільтрів обумовлена тим, що смороду прекрасно пристосовані для придушення деяких видів випадкових шумів, і при цьому приводять до меншого розмивання в порівнянні з лінійними фільтрами, що згладжують, того ж розміру. Медіанні фільтри особливо ефективні при наявності як біполярного, так й уніполярного імпульсного шуму. Насправді, застосування медіанних фільтрів дає відмінні результати для зображень, які перекручені шумом цього типу.

*Адаптивні медіанні фільтри.* Медіанні фільтри добрі працюють доти, поки просторова щільність імпульсного шуму невелике. Адаптивна медіанна фільтрація допомагає впоратися з імпульсним шумом, імовірності якого перевищують вказані значення. Додаткова переважиль адаптивного медіанного фільтра полягає в тому, що такий фільтр «намагається зберегти деталі» в областях, спотворених не імпульсним шумом. Звичайний медіанний фільтр такою властивістю не володіє. Подібно всім розглянутим дотепер фільтрам, адаптивний медіанний фільтр здійснює обробку в прямокутній околиці  $S_{xy}$ . Однак, на відміну від цих фільтрів, адаптивний медіанний фільтр змінює (збільшує) розміри околиці  $S_{xy}$  під година роботи відповідно до наведеного нижче умовами. Будемо пам'ятати про ті, що відгук фільтра являє собою однину, що заміщає значення елемента зображення в тій точці  $(x, y)$ , що є центром околиці  $S_{xy}$ , у сучасний момент.

Введемо наступні позначення:

$Z_{min}$  – мінімальне значення яскравості в  $S_{xy}$ ;

$Z_{max}$  — максимальне значення яскравості в  $S_{xy}$  ;

$Z_{med}$  — медіана значень яскравості в  $S_{xy}$ ;

$Z_{xy}$  – значення яскравості в точці  $(x, y)$ ;

$S_{max}$  – максимальний припустимий розмір  $S_{xy}$ .

Алгоритм адаптивної медіанної фільтрації складається із двох областей, позначених нижче як область  $A$  и область  $B$ , і його дія полягає в наступному.

Область А:	Область Б:
$A1 = Z_{med} - Z_{min};$	$B1 = Z_{xy} - Z_{min};$
$A2 = Z_{med} - Z_{max};$	$B1 = Z_{xy} - Z_{max};$
якщо $A1 > 0$ й $A2 < 0$ , перейти до області Б;	якщо $B1 > 0$ й $B2 < 0$ , результат дорівнює $Z_{xy}$ ;



інакше збільшити розмір околиці; якщо розмір околиці $\leq S_{max}$ повторити область $A$ ; інакше результат дорівнює $Z_{xy}$	інакше результат дорівнює $Z_{med}$
---	-------------------------------------

Для розуміння того, як працює цей алгоритм, необхідно пам'ятати, що його застосування переслідує три основні цілі: видалити біполярний імпульсний шуми, забезпечити згладжування шумів інших типів, а також звести до мінімуму такі спотворення, як надмірне стоншення або стовщення границь об'єктів. Значення  $Z_{min}$  й  $Z_{max}$  сприймаються алгоритмом статистично як значення «імпульсних» складових шуму, навіть якщо смороду не рівні найменшим і найбільшому можливим значенням яскравості на зображенні.

З урахуванням останнього зауваження ми бачимо, що область  $A$  алгоритму має на меті визначити, чи є медіана  $Z_{med}$  імпульсом («чорним» або «білим») чи ні. Якщо умова  $Z_{min} < Z_{med} < Z_{max}$  виконана, те в силу зазначених у попередньому абзаці причин  $Z_{med}$  не може бути імпульсом. У цьому випадку ми переходимо до області  $B$  и перевіряємо, чи є імпульсом значення  $Z_{xy}$  у тій точці, що відповідає центру околиці (нагадаємо, що ми будемо відгук фільтра в цій точці). Якщо умови  $B1 > 0$  й  $B2 < 0$  виконані, те  $Z_{min} < Z_{xy} < Z_{max}$ , і значення  $Z_{xy}$  не є імпульсним з тих же причин, що й вище. У цьому випадку алгоритм дає на виході незмінене значення  $Z_{xy}$ . Збереження значень у таких крапках «проміжного рівня» яскравості мінімізує перекручування, внесені обробкою зображення. Якщо одне з розумів  $B1 > 0$  й  $B2 < 0$  порушено, те або  $Z_{xy} = Z_{min}$ , або  $Z_{xy} = Z_{max}$ . В обох випадках значення є екстремальним, і алгоритм дає на виході значення медіани  $Z_{med}$ , що, як треба з результатів роботи області  $A$ , не є значенням імпульсного шуму. Остання операція відповідає дії звичайного медіанного фільтра. Відмінність полягає в тім, що звичайний медіанний фільтр заміняє значення в кожній точці на значення медіани по відповідній області. Це приводить до зайвих спотворень деталей на зображенні.

**Приклад** ілюструє застосування медіанний фільтр для придушення імпульсного шуму, однак при цьому розмиваються границі між областями з різною амплітудою сигналу. Для зменшення «змазу» границь пропонується адаптивний медіанний фільтр.

*%шумоподавітель імпульсного шуму.*

*close all; %Видалення графічних вікон*

*clear*

*A=imread('C:\MATLAB\Image\pic\Portrait\_w.bmp');%завантаження картинки в робочу облость Matlab*

*whos A, %Видає основну інформацію про зображення*

*A1=im2double(A); %перетворення даних класу uint8 у клас double*

```

whos A1, %Виває основну інформацію про зображення
figure,imshow(A1),title('Оригінал') ,

A_nois=imnoise(A1,'salt & pepper',0.2);% додаємо імпульсний шум, із
щільністю ймовірності 0.2
figure,imshow(A_nois),title('оригінал+імпульсний шум') ,
A_med=medfilt2(A_nois,[5 5],'symmetric'); %медіанний фільтр
figure,imshow(A_med),title('медіанний фільтр') ,
A_amed=adpmedian(A_nois,7);% адаптивний медіанний фільтр
figure,imshow(A_amed),title('адаптивний медіанний фільтр') ,

figure,
subplot(2,2,1),subimage(A1),title('Оригенал') ,
subplot(2,2,2),subimage(A_nois),title('оригінал+імпульсний шум') ,
subplot(2,2,3),subimage(A_med),title('медіанний фільтр') ,
subplot(2,2,4),subimage(A_amed),title('адаптивний медіанний фільтр') ,

```

Наведений результату обробки зображення "Портрет" з додаванням імпульсного шуму із щільністю шум  $d=0.2$  (тобто відсоток зображення підданого цьому шуму). При цьому приблизно  $d \cdot \text{numel}(f)$  пікселів будуть зіпсовані.

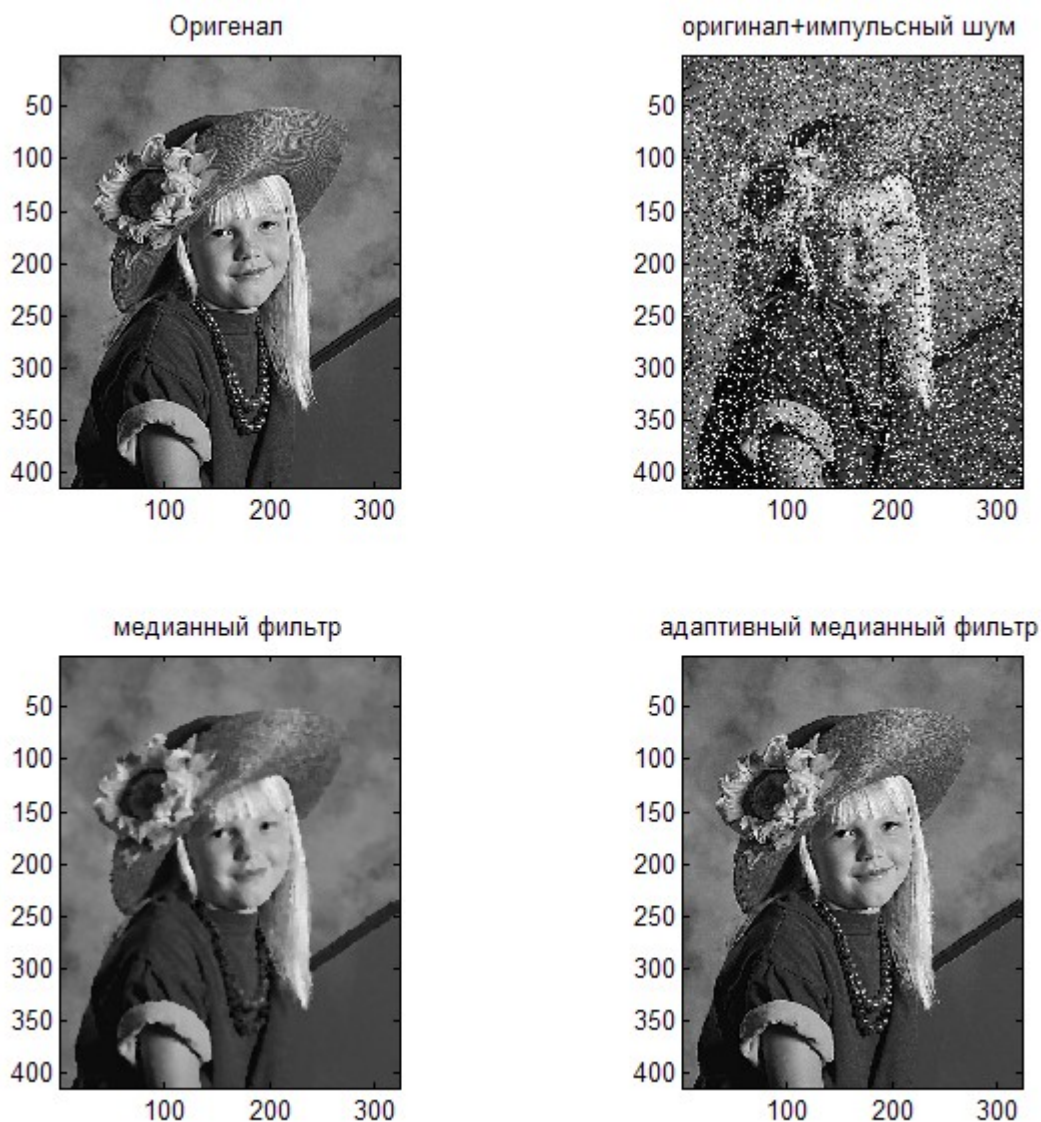


Рисунок 3.17 - Порівняльний аналіз зашумленого імпульсним шумом реального зображення, для оцінки медіанної й адаптивно медіанної фільтрації в околиці  $S=[5 \ 5]$ .

Широка популярність медіанних фільтрів обумовлена тим, що смороду прекрасно пристосовані для придушення деяких видів випадкових шумів, і при цьому приводять до меншого розмивання в порівнянні з лінійними фільтрами, що згладжують, того ж розміру. Медіанні фільтри особливо ефективні при наявності як біполярного, так й уніполярного імпульсного шуму. Насправді, застосування медіанних фільтрів дає відмінні результати для зображень, які перекручені шумом цього типу.

## Двовірне дискретне перетворення Фур'є

Звичайне математичне подання зображення є функцією двох просторових змінних:  $F(x,y)$ . Значення функції в певній крапці  $(x,y)$  являє собою яскравість зображення в цій крапці. Це подання зображень у просторовій області. Термін перетворення ставиться до альтернативного математичного подання зображення. Наприклад, перетворення Фур'є є поданням зображень у вигляді суми комплексних експонент. Це називається частотною областю. Перетворення корисні для широкого кола завдань, у тому числі згортки, поліпшення функції виявлення й стиски.

Перетворення Фур'є це подання зображень у вигляді суми комплексних експонент різної величини, частоти й фази. Перетворення Фур'є відіграє важливу роль у широкому спектрі додатків обробки зображень, включаючи поліпшення, аналіз, відновлення й стиск зображень.

Якщо  $f(m,n)$  є функцією двох дискретних просторових змінних  $m$  й  $n$ , то двовимірне перетворення Фур'є від  $f(m,n)$  визначається співвідношенням

$$F(\omega_1, \omega_2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m,n) e^{-j\omega_1 m} e^{-j\omega_2 n}$$

Змінні  $\omega_1$  й  $\omega_2$  це змінні частоти  $F(\omega_1, \omega_2)$  часто називають поданням у частотній області  $f(m,n)$ .

$F(\omega_1, \omega_2)$  є комплексною функцією, періодичної в  $\omega_1$  й  $\omega_2$  з періодом  $2\pi$ . У зв'язку з періодичністю звичайно відображається діапазон

$-\pi \leq \omega_1, \omega_2 \leq \pi$ .  $F(0,0)$  являє собою суму всіх значень  $f(m,n)$ . Із цієї причини,  $F(0,0)$  часто називають постійною складовою або DC компонентом перетворення Фур'є.

Зворотне перетворення являє собою операцію, що здійснюється над перетвореним зображенням і відтворює вихідне зображення. Зворотне двовимірне перетворення Фур'є

$$f(m,n) = \frac{1}{4\pi^2} \int_{\omega_1=-\pi}^{\pi} \int_{\omega_2=-\pi}^{\pi} F(\omega_1, \omega_2) e^{j\omega_1 m} e^{j\omega_2 n} d\omega_1 d\omega_2$$

Грубо говорячи, ця рівність означає, що  $f(m,n)$  можна представити у вигляді суми нескінченного числа комплексних експонент (синусоїд) з різними частотами. Амплітуди й фазовий внесок на частотах  $(\omega_1, \omega_2)$  для створення зображення має вигляд  $F(\omega_1, \omega_2)$ .

Знаходячись в основі методів лінійної фільтрації, перетворення Фур'є забезпечує значну гнучкість при розробці й реалізації алгоритмів фільтрації при рішенні завдань поліпшення, відновлення й стиснення зображень. Перетворення Фур'є також лежить у фундаменті великої кількості інших важливий практичних застосувань.

Двомірне дискретне перетворення Фур'є. Нехай  $f(x, y)$ , при  $x = 0, 1, 2, \dots, M-1$  і  $y = 0, 1, 2, \dots, N-1$ , означає зображення  $M \times N$ . Двомірне дискретне перетворення Фур'є (DFT, Discrete Fourier Transform) зображення  $f$ , яку записується  $F(u, v)$ , задається рівнянням:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

при  $u = 0, 1, 2, \dots, M-1$  і  $v = 0, 1, 2, \dots, N-1$ . Мі могли б розкласти експоненту на синуси і косинуси від змінних  $u$  і  $v$  з відповідними частотами (змінні  $x$  та  $y$  зникнуть після додавання). *Частотною областю* називається координатна система, що задає аргументи  $F(u, v)$  частотними змінними  $u$  та  $v$ . Тут можна виявити аналогію із задаванням аргументів  $f(x, y)$  просторовими змінними  $x$  та  $y$ . Прямокутну область розміром  $M \times N$ , яка задається при  $u = 0, 1, 2, \dots, M-1$  і  $v = 0, 1, 2, \dots, N-1$ , прийнято називати *частотним прямокутником*. Видно, що частотний прямокутник має ті ж розміри, що й початкове зображення.

Зворотне дискретне перетворення Фур'є. Зворотне дискретне перетворення Фур'є задається рівняннями:

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$$

при  $x = 0, 1, 2, \dots, M-1$  й  $y = 0, 1, 2, \dots, N-1$ . Таким чином, знаючи  $F(u, v)$ , можна відновити  $f(x, y)$  з допомогою зворотного ДПФ. Величини  $F(u, v)$  у цих рівняннях прийнято називати коефіцієнтами розкладання Фур'є.

У деяких визначеннях коефіцієнт  $1/MN$  поміщається в пряме перетворення, а в інших — у зворотне. Для сумісності з реалізацією перетворення Фур'є в системі MATLAB ми будемо вважати, що коефіцієнт  $1/MN$  розташований у формулах зворотного перетворення, як це наведений вище. Оскільки індекси масивів у MATLAB починаються з  $1$ , а не з  $0$ , у MATLAB формули  $F(1, 1)$  та  $f(1, 1)$  відповідають математичним величинам  $F(0, 0)$  і  $f(0, 0)$ , які знаходяться в прямому і зворотному перетвореннях.

Значення перетворення Фур'є в качану координат частотної області (тобто величина  $F(0, 0)$ ) називається коефіцієнтом або компонентою DC перетворення Фур'є.

### Функції частотної фільтрації.

Для виконання прямого перетворення Фур'є в пакеті MATLAB використовується функція `fft2`, що має наступний синтаксис:

$$F = \text{fft2}(f)$$

де  $f$  – вихідне зображення,  $F$  – перетворення Фур'є двовимірної функції  $f(x, y)$ .

$$F = \text{fft2}(f, MROWS, NCOLS).$$

Доповнює матрицю  $f$  нулями до розміру  $MROWS$ -на- $NCOLS$  перед перетворенням.

Для одержання спектра зображення необхідно виконати наступну команду:

$$S = \text{abs}(F)$$

Команда **abs** обчислює абсолютну величину (модуль) комплексної функції  $F$ .

Для зсуву початку координат Фур'є-перетворення в центр частотної області використовується функція **fftshift** з наступним синтаксисом:

$$Fc = \text{fftshift}(F)$$

де  $Fc$  – центроване Фур'є-перетворення.

Міняє порядок виводу **fft**, **fft2** й **fftn**, щоб нульова трансформанта була поміщена в центр масиву. Для векторів, **fftshift(F)** міняє місцями ліву й праву половини  $F$ . Для матриць, **fftshift(F)** міняє місцями перший квадрант із третім і другою із четвертим.

Функції проектування фільтра **fsamp2**, **fwind1** й **fwind2** і перетворення Фур'є **fft2** створюють фільтри й спектр зображення, засновані на матриці необхідної частотної характеристики.

Можна створити відповідну матрицю необхідної частотної характеристики, використовуючи функцію **freqspace**, що повертає точну, з рівномірно розподіленими значеннями частоти матрицю частотної характеристики будь-якого розміру. Якщо ви створюєте необхідну матрицю, використовуючи значення частоти, відмінні від розрахованих **freqspace**, частотно-фазова характеристика може стати нелінійною.

### **freqspace**

$$[F1, F2] = \text{freqspace}(n)$$

Повертає 2D вектори частот  $F1$  й  $F2$  для  $n$ -на- $n$  матриці.

$$[F1, F2] = \text{freqspace}(m, n)$$

Повертає 2D вектори частот для  $m$ -на- $n$  матриці.

Для 2-х векторів і непарного  $n$ ,  $F = (-1+1/n:2/n:1-1/n)$ . Для 2-х векторів і парного  $n$ ,

$$F = (-1:2/n:1-2/n).$$

$$[X1, Y1] = \text{freqspace}(\dots, 'meshgrid')$$

Еквівалентна  $[F1, F2] = \text{freqspace}(\dots); [F1, F2] = \text{meshgrid}(F1, F2);$

$$F = \text{freqspace}(n)$$

Повертає 1D вектор частот  $F$ , припускаючи  $n$  рівномірно розподілених

крапок навколо окружності. Для 1D вектора,  $F=(0:2/n:1)$ .

$F=freqspace(N,'whole')$

Повертає всі  $n$  дорівнює розташованих крапок. У цьому випадку  $F=(0:2/n:2*(n-1)/n)$ .

### Фільтрація в частотній області.

Фільтрація в частотній області має вельми просту концепцію. Основою лінійної фільтрації в частотній й просторовій області є теорема про згортку, якові можна сформулювати так:

$$f(x,y) * h(x, y) \Leftrightarrow H(u,v) F(u,v)$$

і у зворотну сторону

$$f(x,y) h(x, y) \Leftrightarrow H(u,v) * F(u,v)$$

Отут символ «\*» позначає операцію згортки двох функцій, а вирази по обидві сторони від подвійних стрілок визначають відповідні парі при перетворенні Фур'є. Наприклад, перший вираз означає, що згортку двох просторових функцій можна отримати, якщо обчислити зворотне перетворення Фур'є від добутку прямих перетворень Фур'є цих двох функцій. Навпаки, пряме перетворення Фур'є згортки двох просторових функцій дає добуток їх прямих перетворень Фур'є. Аналогічним чином можна прокоментувати й другу твердження.

Відповідно до теореми про згортку, тієї ж результат можна отримати в частотній області, помноживши  $F(u,v)$  на  $H(u,v)$  — перетворення Фур'є просторового фільтру. Прийнято називати  $H(u,v)$  *передаточною функцією фільтру*.

**Основні кроки фільтрації в частотній області.** Дії, що обговорювалися в попередньому параграфі, можна формалізувати у вигляді наступної покрокової процедури з використанням функцій MATLAB. Через  $f$  позначимо вихідне зображення, а через  $g$  - результат фільтрації. Передбачається, що передавальна функція  $H(i,v)$  має ті ж розміри, що й вихідне зображення.

1. Отримати параметри розширення з допомогою *paddedsizе*:

$$PQ = paddedsizе(size(f));$$

2. Побудувати перетворення Фур'є з розширенням:

$$F = fft2(f, PQ(1), PQ(2));$$

3. Згенерувати функцію фільтру  $H$  розміром  $PQ(1) \times PQ(2)$  одним з описаних далі методів. Якщо він був центрувань, до використання його у фільтрації слід виконати команду  $H=fftshift(H)$ .

4. Помножити перетворення Фур'є на передаточну функцію фільтра:

$$G=H.*P;$$

5. Знайти дійсну частину зворотного перетворення Фур'є від  $G$ :  
 $g = \text{real}(\text{ifft2}(G));$
6. Вирізати верхній лівий прямокутник початкових розмірів:  
 $g = g(1:\text{size}(f,1), 1:\text{size}(f,1));$

Ця процедура фільтрації схематично змальована на рис. 2.1. Попередня стадія обробки може складатися з визначення розмірів зображення, обчислення параметрів розширення й генерації фільтру. Завершальна стадія обробки полягає у виділенні дійсної частини результату, обрізання зображення до початкового розміру і його конвертації в клас `unit8` або `unit16` для збереження на диску.

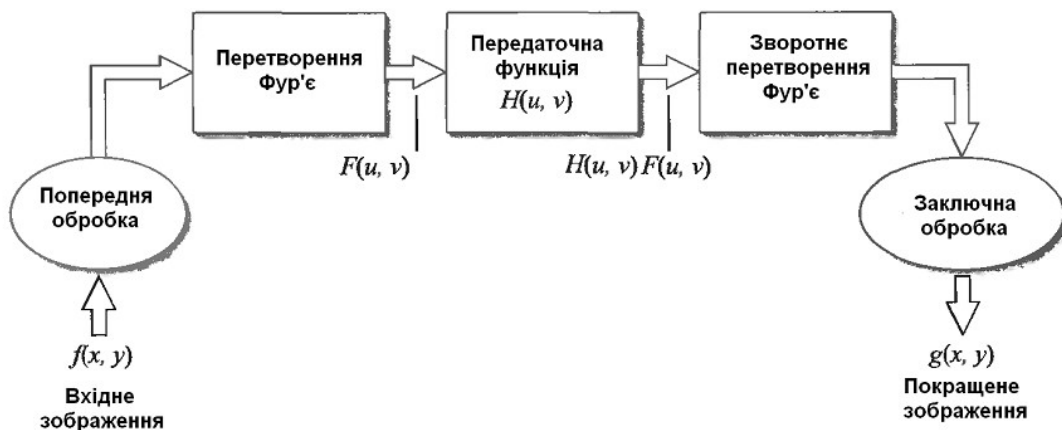


Рисунок 3.18 - Основні кроки фільтрації в частотній області

Передаточна функція фільтра  $H(u, v)$  на мал. 2.1 множитися на дійсну і уявну частини  $F(u, v)$ . Якщо функція  $H(u, v)$  була дійсною, те фазова частина виразу не міняється, що видно з фазового рівняння, оскільки при множенні дійсної і уявної частини комплексного числа на одне і ті ж дійсне число фазовий кут не міняється. Такі фільтри прийнято називати фільтрами з нульовим зрушенням фази.

#### Візуалізація перетворення Фур'є

Щоб проілюструвати це, розглянемо функцію  $f(m, n)$ , рівну  $1$  у прямокутній області й  $0$  поза її (мал. Для спрощення схеми,  $f(m, n)$  показана, як безперервна функція, хоча змінні  $m$  й  $n$  дискретні.

*%Будується дискретне зображення прямокутної області.*

```
f=zeros(32,32);
```

```
f(5:24,13:17)=1;
```

```
figure, imshow(f,'InitialMagnification','fit'),title('Оригінал') ,
```

```
figure, imshow(f,'InitialMagnification',1000)
```

*%InitialMagnification - Управляє початковим збільшенням зображення, відображуваного imshow.*

*%'fit' - масштабуйте зображення так, щоб воно вписувалося у вікно цілком.*



```

%Обчислюється перетворення Фур'є зображення
% Отримати параметри розширення з допомогою paddedsize:
PQ = paddedsize(size(f));
% Побудувати перетворення Фур'є з розширенням:
F = fft2(f,PQ(1),PQ(2)); AF=abs(F);
% Функція freqspace розраховує вектор рівномірно розташованих значень
частот
% нормалізовані значення частот у діапазоні  $f_n = [-1, 1]$ ,  $f = [-fs/2 fs/2]$ 
[fx,fy] = freqspace(size(F));
figure, mesh(fx,fy, fftshift(AF)),xlabel('fx');ylabel('fy'),zlabel('амплит'),
title('спектр зображення, модуль ДПФ-2D'),

```

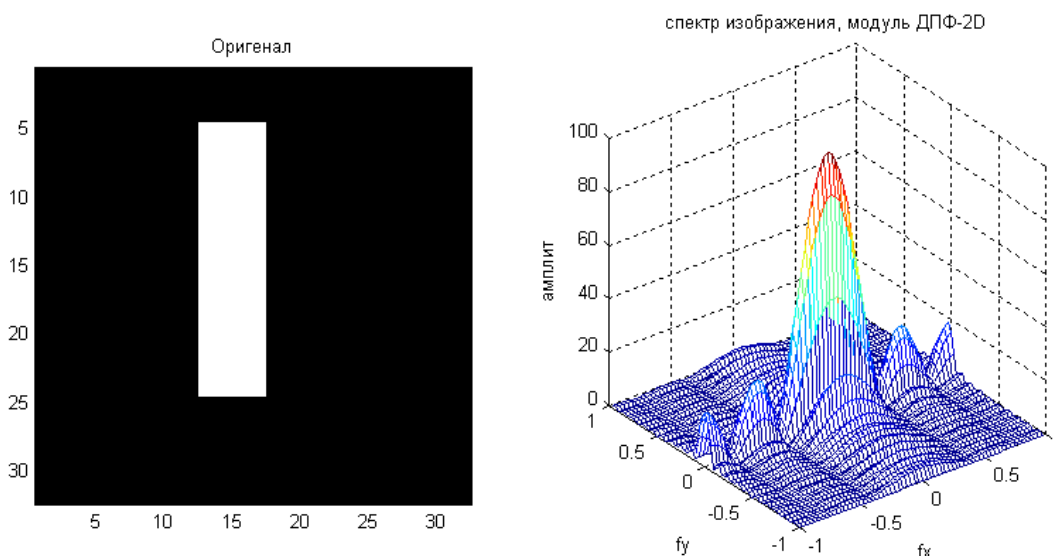


Рисунок 3.19 - Зображення прямокутної функції і його спектр

На рис. 17, показана величина перетворення Фур'є  $F(u,v)$  цієї прямокутної функції. Такий вид подання є розповсюдженим способом для візуалізації перетворення Фур'є.

Пік у центрі малюнка  $F(0,0)$  являє собою суму всіх значень в  $f(m,n)$ . Графік показує, що  $F(u,v)$  має більше енергії при високих горизонтальних частотах, чим при високих вертикальних. Це відбиває той факт, що горизонтальні перетини  $f(m,n)$  уже вертикальних.

### Проектування лінійних фільтрів у частотній області

Якщо ядро фільтра велике, фільтрація в просторовій області вимагає більших обчислювальних витрат. Більше ефективними стають методи фільтрації в частотній області.

Image Processing Toolbox забезпечує проектування двовимірних фільтрів з кінцевою імпульсною характеристикою (КИХ) (finite impulse

response FIR). КИХ фільтри мають кінцеву довжину реакції на крапковий вплив. Всі функції проектування фільтрів у пакеті створюють КИХ фільтри.

КИХ фільтри мають кілька особливостей, які роблять їх ідеальними для обробки зображень у середовищі MATLAB:

- КИХ фільтри легко представити у вигляді матриці коефіцієнтів.
- Двовимірні КИХ фільтри є природним продовженням одномірних КИХ-фільтрів.
- Відомі надійні методи проектування КИХ фільтрів.
- КИХ фільтри легко реалізовані.
- КИХ фільтри можуть бути спроектовані з лінійною фазовою характеристикою, що забезпечує відсутність перекручувань зображення.

### **Метод перетворення частоти. Функція *ftrans2***

Метод перетворення частоти перетворить одномірний КИХ фільтр у двовимірний КИХ фільтр. Метод перетворення частоти зберігає більшість характеристик одномірних фільтрів, зокрема, смугу пропускання й характеристики пульсації. Цей метод використовує матрицю перетворення, набір елементів, що визначає частоту перетворення.

Функція *ftrans2* реалізує метод перетворення частоти. Матриця перетворення McClellan цієї функції (за замовчуванням) створює фільтри з майже круговою симетрією.

$h=ftrans2(b)$  - Розраховує 2D КИХ-фільтр  $h$ , що відповідає 1D КИХ-фільтру  $b$  за допомогою перетворення McClellan.

Метод перетворення частоти в цілому дає дуже гарні результати, тому що простіше створити одномірний фільтр із особливими характеристиками, чим відповідний двовимірний фільтр. У **прикладі** спроектований оптимальний з постійними пульсаціями одномірний КИХ фільтр і він використаний для створення двовимірного фільтра з аналогічними характеристиками. Форма двовимірної частотної характеристики добре відповідає одномірній.

### **Приклад**

```
% Проектування 2D лінійних фільтрів у частотній області
% Создание 1D фільтра.
% b = remez(n,f,a)-Повертає вектор b довжини n+1, що містить
коєфіцієнти нерекурсивного (FIR) фільтра,
% частотна характеристика якого задається векторами f й a.
% f - вектор пар частотних крапок, що лежать у діапазоні від 0 до 1 (1
відповідає частоті Найквіста, тобто половині частоти дискретизації).
% Частоти повинні утворювати неубутну послідовність.
% a - вектор, що містить значення бажаної АЧХ для частот з вектора f.
b=remez(10,[0 0.4 0.6 1],[1 1 0 0]);
```

```

% Перетворення 1D в 2D ядро фільтра.
h=ftrans2(b);
% Обчислення частотної характеристики -freqz.
%Повертає вектор значень комплексної частотної характеристики h,
%обчислених на частотах, заданих (у радіанах на відлік) вектором w.
%Використає 64 частотних крапок, рівномірно розподілених уздовж повної
%одиничної окружності на комплексній z-площині.
%Повертає вектор, що, частот w має довжину 64 і містить значення,
%рівномірно розподілені в діапазоні від 0 до 2 $\pi$  радіан на відлік.
[H,w]=freqz(b,1,64,'whole');
figure, plot(w/pi-1,fftshift(abs(H))),grid, xlabel('нормована частота *pi
rad/sampi'),ylabel ('АЧХ'),title('АЧХ 1D цифрового фільтра прототипу')
% Графіки частотної характеристики фільтра 2D
% виводить у поточне вікно двовимірну АЧХ, розмір матриці АЧХ [32 32] .
figure,freqz2(h,[32 32]) ,title('АЧХ 2D цифрові фільтри')
colormap([0 0 0])
figure,
subplot(1,2,1),plot(w/pi-1,fftshift(abs(H))),grid, xlabel('нормована частота *pi
rad/sampi'),ylabel ('АЧХ'),title('АЧХ 1D цифрового фільтра прототипу')
subplot(1,2,2),freqz2(h,[32 32]) ,title('АЧХ 2D цифрові фільтри') ,colormap([0
0 0])

```

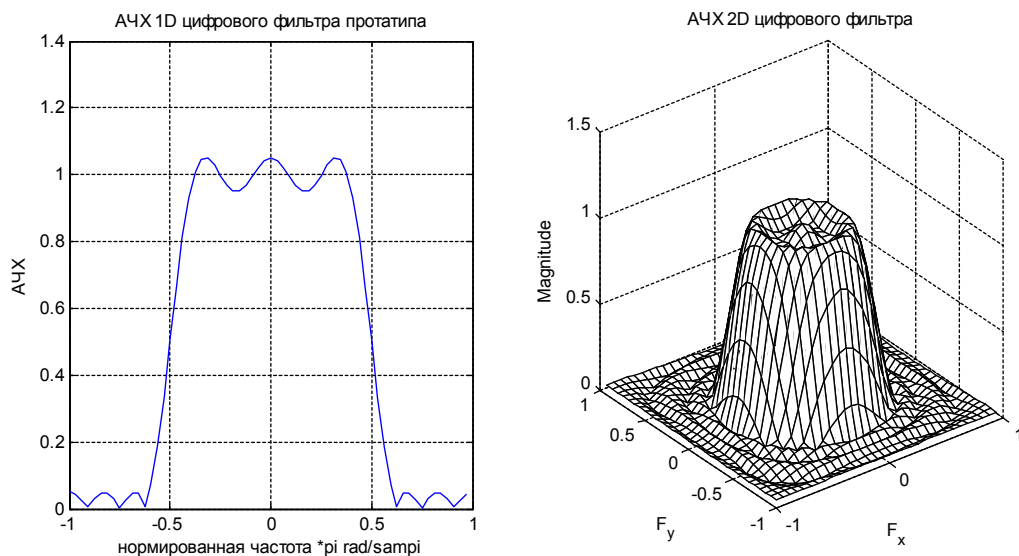


Рисунок 3.20 - 1D й 2D частотні характеристики фільтра

### Метод дискретизації частоти. Функція *fsamp2*

Метод дискретизації частоти створює фільтр, заснований на необхідній частотній характеристиці. Для матриці крапок, які визначають форму частотної характеристики, цей метод створює фільтр, частотна характеристика якого проходить через ці крапки. Частота дискретизації не

накладає обмежень на поводження АЧХ між заданими крапками, тому, звичайно характеристика має нерівномірності в цих областях.

Функція *fsamp2* розраховує методом дискретизації частоти двовимірний КИХ фільтр.

***h=fsamp2(HD)***

Розраховує двовимірний КИХ-фільтр із частотною характеристикою ***HD*** і повертає коефіцієнти фільтра в матрицю ***h***. *fsamp2* повертає ***h*** у формі, що підходить для використання з функцією *filter2*. Фільтр ***h*** має частотну характеристику, що проходить через крапки в ***HD***. Якщо ***HD*** масив *m-на-n*, то ***h*** також масив *m-на-n*. ***HD*** являє собою матрицю, що містить необхідну частотну характеристику в рівновіддалених крапках між ***-1,0*** й ***1,0*** по ***x*** и ***v*** осях, де ***1.0*** відповідає половині частоти дискретизації або  **$\pi$**  радіан. Для точного результату, використовуйте частоти крапок, що повертають функцією *freqspace* при створенні ***HD***.

***h=fsamp2(F1,F2,HD,[m n])***

Розраховує *m-на-n* ***FIR*** фільтр, що відповідає відгуку ***HD*** у крапках векторів ***F1*** й ***F2***. У результаті фільтр забезпечує мінімальну середньоквадратичну помилку наближення до необхідної характеристики. Для досягнення найкращих результатів повинне побут принаймні достатня кількість крапок *m-на-n* для апроксимації необхідної характеристики.

*fsamp2* повертає фільтр ***h*** із частотною характеристикою, що проходить через крапки вхідної матриці ***HD***. У **прикладі** створюється 11 на 11 фільтр, використовуючи *fsamp2* і будується графік частотної характеристики спроектованого фільтра.

### Приклад

```
% Бажана частотна характеристика фільтра 2D
HD=zeros(11,11);HD(4:8,4:8)=1;
%Вибір контрольних крапок у частотній області.
[f1,f2]=freqspace(11,'meshgrid');
%Побудова графіка необхідного фільтра.
figure, mesh(f1,f2,HD),axis([-1 1 -1 1 0 1.2]),colormap(jet(64)),
xlabel('wx *pi rad/sampi'),ylabel ('wy *pi rad/sampi'),title('АЧХ 2D цифрові
фільтри прототипу')
%Проектування фільтра методом дискретизації частоти.
%Функція h=fsamp2(H) формує маску лінійного двовимірного КИХ-
фільтра,
% ґрунтуючись на бажаній АЧХ двовимірного фільтра, переданої в матриці
H.
%Значення АЧХ у матриці H відповідають частотам по осях абсцис й
ординат,
```

```

% рівномірно розподіленим у діапазоні [-1, 1], де 1 відповідає половині
частоти дискретизації
h=fsamp2(HD);
% Побудова графіка спроектованої характеристики.
%% freqz2 виводить у поточне вікно двовимірну АЧХ, розмір матриці АЧХ
[32 32]
figure,freqz2(h,[32 32]),axis([-1 1 -1 1 0 1.2])
xlabel('wx *pi rad/sampi'),ylabel ('wy *pi rad/sampi'),title('АЧХ 2D проектовані
цифрові фільтри')

```

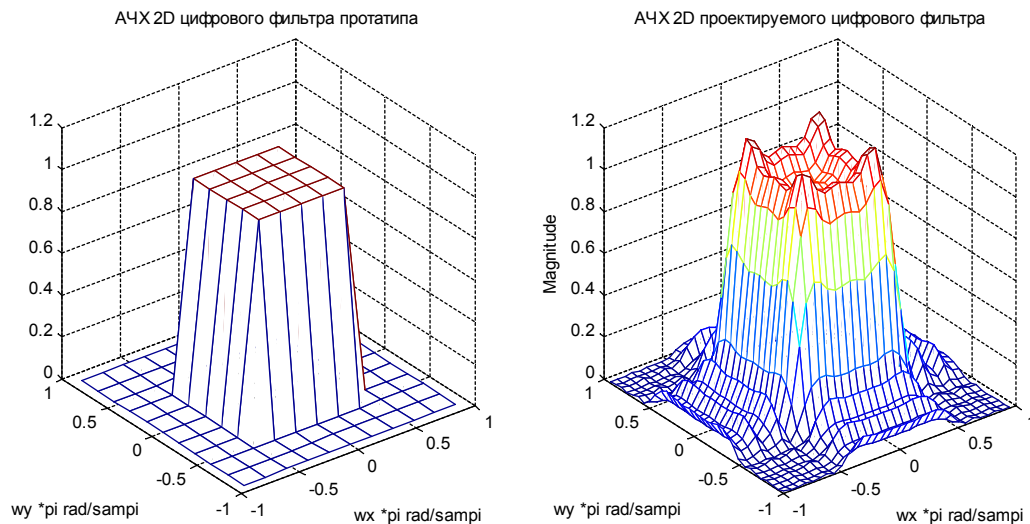


Рисунок 3.21 - Необхідна (ліворуч) і фактична (праворуч) двовимірні частотні характеристики фільтра

На реальній частотній характеристиці з'явилися нерівномірності (пульсації) у порівнянні з необхідною частотною характеристикою. Ці пульсації є основною проблемою при проектуванні методом дискретизації частоти. Вони максимальні в зонах різких переходів частотної характеристики.

### Формування маски фільтра по бажаній АЧХ із застосуванням 1-D вікна – **fwind1**

Метод вікна припускає множення ідеальної імпульсної характеристики з функцією вікна для створення відповідного фільтра, що згладжує ідеальну імпульсну характеристику. Як метод дискретизації частоти, метод вікна створює фільтр, частотна характеристика якого наближається до бажаної частотної характеристики. Віконний метод, однак, як правило, дає кращі результати, чим метод дискретизації частоти.

Функція **fwind1** конструює двовимірний фільтр за допомогою двовимірного вікна, що створюється з одного або двох одновимірних вікон.

**$h=fwind1(HD,win)$**

Розраховує двовимірний КИХ-фільтр  $h$  із частотної характеристики  $HD$ . *fwind1* повертає  $h$  у формі, що підходить для використання з функцією *filter2*. *fwind1* використає одновимірне вікно *win* для формування двовимірного вікна із приблизно круговою симетрією за допомогою методу Huang. Можна використати вікна з Signal Processing Toolbox наступних типів: *boxcar*, *hamming*, *hanning*, *bartlett*, *blackman*, *Kaiser*. Якщо *length* дорівнює  $N$ , масив  $h$  має розмір  $n$ -на- $n$ .  $HD$  являє собою матрицю, що містить потрібні отсчеты в рівновіддалених крапках між  $-1.0$  й  $1.0$  (нормовані частоти, де  $1.0$  відповідає половині частоти дискретизації, або  $\pi$  радіан) по  $x$  и  $y$  осях. Для точного результату, використовуйте частоти крапок, що повертають функцією *freqspace* при створенні  $HD$ .

$$h=fwind1(HD,win1,win2)$$

Використається два одновимірних вікна *win1* й *win2* для створення роздільного двовимірного вікна. Якщо довжина (*win1*) –  $n$  і довжина (*win2*) –  $m$ , то  $h$  масив  $m$ -на- $n$ .

$$h=fwind1(F1,F2,HD,...)$$

Дозволяє вказати бажану частотну характеристику  $HD$  при будь-яких частотах (нормовані вектори  $F1$  й  $F2$ ) уздовж осей  $x$  и  $y$ . Довжина вікна контролює розмір отриманого фільтра, як описувалося вище.

У прикладі *fwind1* створює фільтр  $11$  на  $11$  з необхідної частотної характеристики  $HD$ . Використається вікно *hamming* для створення одновимірного вікна, що *fwind1* потім поширює на двовимірне вікно.

## Приклад

```
%Бажана частотна характеристика
Hd=zeros(11,11); Hd(4:8,4:8)=1;
%Вибір контрольних крапок у частотній області.
[f1,f2]=freqspace(11,'meshgrid');
%Побудова графіка необхідного фільтра.
figure, mesh(f1,f2,Hd), axis([-1 1 -1 1 0 1.2]), colormap(jet(64))
xlabel('wx *pi rad/sampi'),ylabel ('wy *pi rad/sampi'),title('АЧХ 2D цифрові
фільтри протатипа')
%Згладжування методом вікна й визначення ядра фільтра.
h=fwind1(Hd,hamming(11));
%Побудова графіка згладженого фільтра.
figure, freqz2(h,[32 32]), axis([-1 1 -1 1 0 1.2])
xlabel('wx *pi rad/sampi'),ylabel ('wy *pi rad/sampi'),title('АЧХ 2D проєктовані
цифрові фільтри')
figure,
subplot(1,2,1),mesh(f1,f2,Hd),axis([-1 1 -1 1 0 1.2]),colormap(jet(64)),
```

```

xlabel('wx *pi rad/sampi'),ylabel ('wy *pi rad/sampi'),title('АЧХ 2D цифрові
фільтри прототипу')
subplot(1,2,2),freqz2(h,[32 32]),axis([-1 1 -1 0 1.2])
xlabel('wx *pi rad/sampi'),ylabel ('wy *pi rad/sampi'),title('АЧХ 2D проєктовані
цифрові фільтри')

```

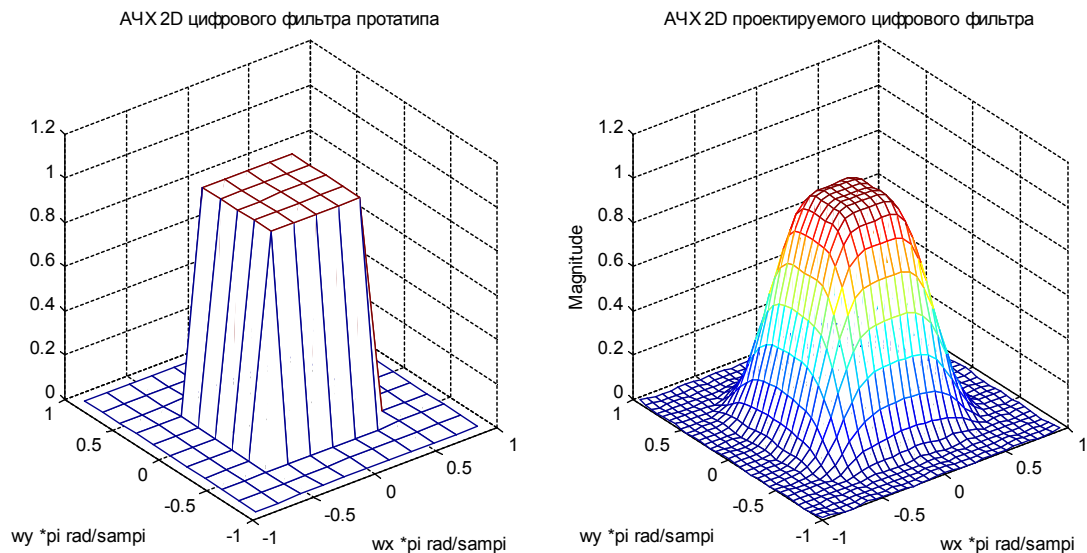


Рисунок 3.22 - Необхідна двовимірна частотна характеристика фільтра (ліворуч) і згладжена (праворуч)

### Обчислення частотної характеристики фільтра

Функція *freqz2* обчислює частотну характеристику двовимірного фільтра, використовуючи швидке перетворення Фур'є (функцію *fft2*).

$[H, Fx, Fy] = \text{freqz2}(h, Nx, Ny)$

Повертає матрицю  $H$  –  $(Nx \times Ny)$ , що є частотної характеристикою вектори частот  $Fx$  (розмірність  $Nx$ ),  $Fy$  (розмірність  $Ny$ ).  $h$  це 2D КИХ фільтр.  $Fx, Fy$  нормалізовані в діапазоні  $-1.0$   $+1.0$ , де  $1.0$  відповідає половині частоти дискретизації.

$[H, Fx, Fy] = \text{freqz2}(h, [Ny Nx])$

Той же результат

$[H, Fx, Fy] = \text{freqz2}(h, N)$

Використає  $[Ny Nx] = [N N]$ .

$[H, Fx, Fy] = \text{freqz2}(h)$

Використає  $[Nx Ny] = [64 64]$ .

$H = \text{freqz2}(h, f1, f2)$

Повертає АЧХ КИХ-фільтра  $h$  на значеннях частоти в  $F_x$  й  $F_y$ . Ці значення частоти повинні бути в діапазоні від  $-1.0$  до  $1.0$ , де  $1.0$  відповідає половині частоти дискретизації, або  $\pi$  радіан).

### Приклад

```
%Розглянемо КИХ фільтр, Його імпульсна характеристика (ядро) 3 на 3  
елементи  
 $h=[0.1667\ 0.6667\ 0.1667; 0.6667\ -3.3333\ 0.6667; \dots$   
 $0.1667\ 0.6667\ 0.1667];$   
%Обчислимо й відобразимо частотну характеристику у вікні 64-на-64  
відліку.  
 $figure, freqz2(h)$ 
```

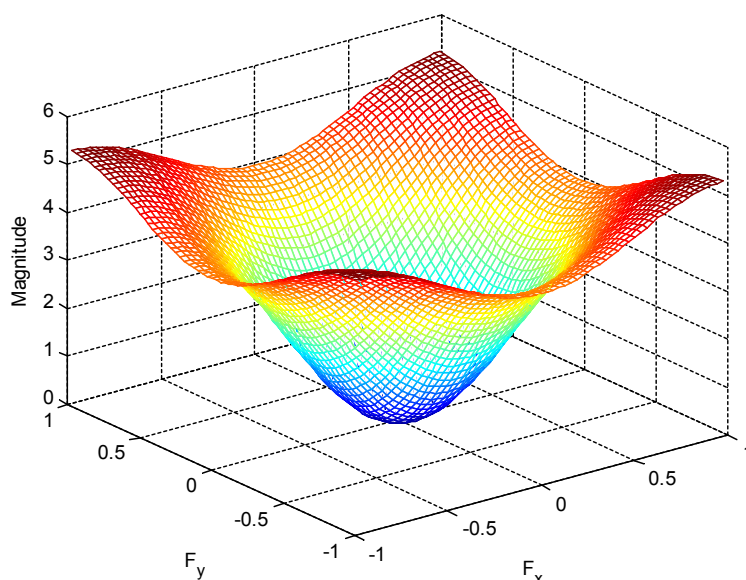


Рисунок 3.23 - АЧХ двовимірного фільтра

Для простого  $m$  на  $n$  виходу, як показано вище, **freqz2** використовують двовимірне швидке перетворення Фур'є – функцію **fft2**. Ви можете також визначити вектори довільних крапок частоти, але в цьому випадку **freqz2** використає більше повільний алгоритм.

Перетворення Фур'є імпульсної характеристики лінійного фільтра дає частотну характеристику фільтра. Функція **freqz2** обчислює й відображає АЧХ фільтра. Частотна характеристика фільтра Gaussian показує, що цей фільтр низьких частот (рис. 3.24).



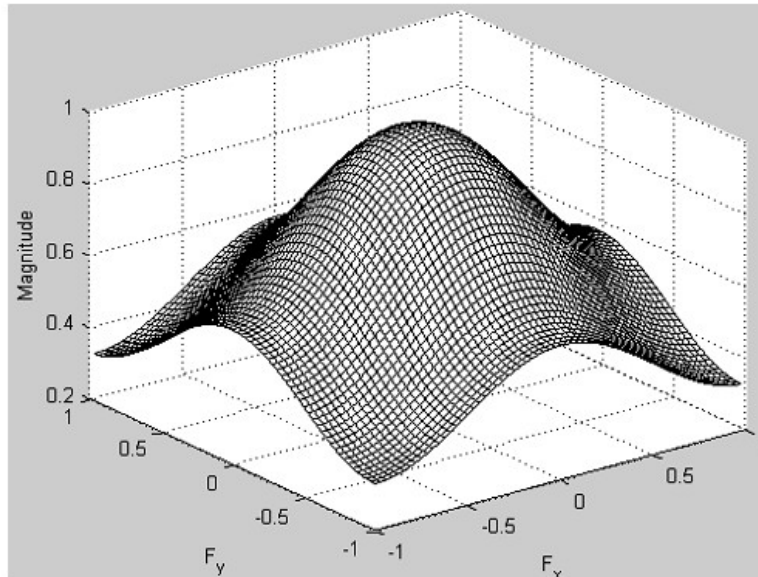


Рисунок 3.24 - Частотна характеристика фільтра Gaussian

### Приклад

*Будується частотна характеристика гауссова фільтра*

*`h=fspecial('gaussian');`*

*`freqz2(h)`*

### Порядок проведення лабораторної роботи

1) Створити й налагодити програму, що вирішує наступні завдання:

а) завантажити зображення 1 за допомогою функції `imread`. Якщо з'ясовано, що зображення кольорове (складається із трьох компонентів), перетворити його в напівтонове за допомогою функції `rgb2gray`;

б) вивести завантажене зображення в графічне вікно за допомогою функції `imshow`, при цьому сформувані пояснювальні написи: `Original`. Для того, щоб не змінювати вихідне зображення при виводі на екран варто використати функцію `imshow` без параметрів;

в) використовуючи функцію `imnoise` додати до зображення нормальний гауссовський шум з параметрами , узятими за замовчуванням;

г) вивести отримане зображення в графічне вікно за допомогою функції `imshow`, при цьому сформувані пояснювальні написи: `Original+Gaussian, var=xxx`;

2) Зрівняти результати впливу шуму на вихідне зображення. Зробити виводи.

- 3) Створити й налагодити програму, що вирішує наступні завдання:
- а) завантажити зображення 1 за допомогою функції `imread`. Якщо з'ясовано, що зображення кольорове (складається із трьох компонентів), перетворити його в напівтонове за допомогою функції `rgb2gray`;
  - б) вивести завантажене зображення в графічне вікно за допомогою функції `imshow`, при цьому сформувавши пояснювальні написи: `Original`. Для того, щоб не змінювати вихідне зображення при виводі на екран варто використати функцію `imshow` без параметрів;
  - в) використовуючи функцію `imnoise` додати до зображення високочастотний шум типу «сіль і перець» із щільністю, зазначеної в стовпці d1,2 таблиці з варіантами завдань;
  - г) вивести отримане зображення в графічне вікно за допомогою функції `imshow`, при цьому сформувавши пояснювальні написи: `Original+Noise d=xxx`;
  - д) сформувавши медіанний фільтр із маскою розмірами  $m \times n = 3 \times 3$  і виконати нелінійну фільтрацію з метою усунення високочастотного шуму;
  - е) вивести отримане зображення в графічне вікно за допомогою функції `imshow`, при цьому сформувавши пояснювальні написи: `Rezalt , median mxn`. Оцінити результат;
  - ж) повторити пункти д) і е) завдання для медіанного фільтра розмірами  $m \times n = 5 \times 5$  й  $7 \times 7$ ;
- з) зберегти отримані зображення у файл.
- 4) Проаналізувати зображення, виведені в графічні вікна, зробити виводи.

### Варіанти завдання

Варіант	Файл зображення
1	Portrait_w.bmp; YP.bmp; Lenna_(test_image).bmp
2	face.gif; Yfq_0.5-4.43-5.5.bmp; pout.jpg
3	clown.gif; boat.bmp; palzt.jpg
4	Portrait_w.bmp; pollen.tif; FR.bmp
5	Lenna_(test_image).bmp; gory.jpg; H_h.bmp
6	zelda.bmp; valley.bmp; 31.bmp
7	Portrait_w.bmp; pollen.tif; H_h.bmp
8	clown.gif; boat.bmp; palzt.jpg

### Порядок оформлення звіту та його представлення й захист.

Результати проведеної лабораторної роботи оформлюються протоколом. Він повинен містити такі дані:

- 1) тема практичної роботи;
- 2) позначка практичної роботи;
- 3) короткі теоретичні відомості;
- 4) лістинги всіх програм з обов'язковими коментарями;
- 5) отримані на шкiрному етапі роботи зображення;
- 6) висновки.

### **Запитання для самоперевірки:**

1. У чому полягає просторова фільтрація?
2. Які методи просторової фільтрації вам відомі?
3. У чому полягає робота фільтру усереднення?
4. Що лежить в основі рангової фільтрації?
5. Який принцип дії медіанного фільтру?
6. Який алгоритм роботи адаптивного медіанного фільтру?
7. Які команди системи MATLAB реалізують методи просторової фільтрації?
8. Що таке двовимірне ДПФ та який його фізичний зміст?
9. Чому дорівнює ДПФ на качану координат? Який його фізичний зміст?
10. Які команди MATLAB виконують фільтрацію зображення в частотній області?