Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут»

Цифрова обробка зображень

Методичні рекомендації

до виконання лабораторних робіт
для студентів спеціальності
7.05080302, 8.05080302 «Аудіо-, відео- та кінотехніка»
усіх форм навчання

Київ
2016

Цифрова обробка зображень [Текст] : метод, рекомендації до викон. лаборатор. робіт для студ. спеціальності 7.05080302, 8.05080302 «Аудіо-, відео- та кінотехніка» усіх форм навчання / Уклад.: В. С. Лазебний, П. В. Попович. - К.: НТУУ «КПІ», 2016. - 73 с.

Рекомендовано Вченою радою ФЕЛ
(Протокол №05/2016 від 30 травня 2016 р.)

НАВЧАЛЬНЕ ВИДАННЯ

Цифрова обробка зображень

Методичні рекомендації

до виконання лабораторних робіт
для студентів спеціальності
7.05080302, 8.05080302 «Аудіо-, відео- та кінотехніка»
усіх форм навчання

Укладачі: *Лазебний Володимир Семенович*, канд. техн. наук, доц.
Попович Павло Васильович, асист.

Відповідальний редактор *Ю. Г. Савченко,* д-р техн. наук, проф.

Рецензент *С.А. Найда*, д-р техн. наук, проф.

ВСТУП

Мета лабораторних занять:

вивчення методів та способів цифрової обробки зображень з застосуванням системи моделювання MATLAB.

При виконанні лабораторних робіт студентам пропонується:

- ознайомитись з принципами обробки зображень, що застосовуються у системі моделювання MATLAB та навчитись використовувати його засоби для обробки зображень;
- дослідити різноманітні методи покращення і обробки зображень та з'ясувати їх ефективність для різних типів зображень;
- ознайомитись зі способами представлення зображень в області просторових частот та частотними методами фільтрації зображень;
- проаналізувати призначення та принцип дії різних фільтрів та методів їх створення у системі MATLAB;
- ознайомитись із застосуванням дискретного косинусного перетворення для стиснення зображень у форматі JPEG та проаналізувати ефективність стиснення, що досягається для різних степенів якості отриманого зображення;
- закріпити знання з обробки статичних зображень, отримані при виконанні попередніх робіт та отримати навики розробки моделей для обробки зображень в середовищі моделювання SIMULINK.

Організація та проведення лабораторних занять

Специфіка виконання лабораторних робіт з даного курсу, а також вимоги техніки безпеки щодо роботи з електронними приладами вимагає, щоб до складу кожної бригади входило не менше двох осіб.

Лабораторні роботи виконують у два етапи. Перший етап -підготовчий - потребує попередньої проробки методичних вказівок та навчальних посібників, які вказано у списку літератури. Цей етап вимагає з'ясування завдання лабораторної роботи, відновлення теоретичних знань з її тематики. На першому етапі необхідно з'ясувати мету виконання роботи, вивчити теоретичне підґрунтя лабораторної роботи, ознайомитись з програмними засобами, що використовуватимуться для в ході її виконання.

Другий етап - виконання лабораторної роботи – вимагає застосування отриманих знань для роботи з набором інструментів Image Processing Toolbox, складання листингу або блок-схеми моделі в системі моделювання MATLAB відповідно до завдання лабораторної роботи, їх застосування для обробки зображень

, аналізу оброблених зображень з наступним формулюванням висновків щодо результатів виконання роботи. По успішному виконанні лабораторної роботи складають звіт про виконану роботу.

Звіт має обов'язково містити:

- назву лабораторної роботи;
- дату виконання роботи;
- прізвище та ініціали студента;
- шифр групи;
- мету лабораторної роботи;
- листинг або блок-схему моделі, що використовувалася;
- оформлені у вигляді графіків та рисунків результати;
- аналіз отриманих результатів;
- висновки за результатами роботи;
- список використаної літератури.

Звіт складають згідно з вимогами стандартів на технічну документацію, охайно та грамотно. Звіт підписує виконавець.

Оформлений звіт здають викладачеві після завершення лабораторного заняття. Студентам, які не відзвітували за попередню роботу, виконувати наступні лабораторні роботи не дозволяється.

З метою забезпечення завчасної підготовки студентів до лабораторних занять до їх відома доводиться графік виконання робіт на семестр (для кожної бригади).

ПЕРЕЛІК СКОРОЧЕНЬ

BLOB – Binary Large Object (бінарний великий об'єкт);

BLPF – Butterworth LowPass Filter (низькочастотний фільтр Баттерворта);

DFT – Descrete Fourier Transform (дискретне перетворення Фур'є);

FFT – Fast Fourier Transform (швидке перетворення Фур’є);

ILPF – Ideal Lowpass Filter (ідеальний низькочастотний фільтр);

JPEG – Joint Photographic Experts Group (Об’єднана Група Експертів з Фотографії);

MLE – Maximum-Likelihood Estimation (наближення по максимуму правдоподібності);

PSF – Point Spread Function (функція спотворення зображення/оптична функція);

АЧХ – амплітудно-частотна характеристика;

ВЧ – високі частоти;

ДКП – дискретне косинусне перетворення;

ДПФ – дискретне перетворення Фур'є;

НЧ – низькі частоти.

Лабораторна робота №1

ОСНОВИ РОБОТИ З ЗОБРАЖЕННЯМИ В MATLAB

Мета роботи: ознайомитись з принципами обробки зображень, що застосовуються у системі моделювання MATLAB з набором інструментів Image Processing Toolbox; вивчити основні команди пакету для зчитування, виведення та запису зображень.

Засоби виконання: пакет MATLAB 6.5/7.* зі встановленим набором інструментів Image Processing Toolbox версії 3.*/4.*.

Теоретичні відомості

Завантаження зображення. Для завантаження зображення в робочу область MATLAB використовується функція `imread` з наступним синтаксисом:

```
imread ('filename')
```

де `filename` – це рядок символів, що утворюють повне ім'я завантажуваного файлу зображення (включно з його розширенням). Наприклад, *командний рядок*:

```
>> f=imread ('chestxray.jpg');
```

присвоює зображення формату JPEG (див. табл. 1.1) з ім'ям `chestxray` матричній змінній `f`. Зауважимо, що символ `(')` використовується в якості обмежувача символьного рядка `filename`. Крапка з комою в кінці командного рядка означає інструкцію в системі MATLAB не відображати виведення для даної команди. Якщо крапка з комою відсутня, то MATLAB відображає (виводить) результат виконання операції в командному рядку. Символ запрошення (>>) позначає початок командного рядка, що з'являється у вікні MATLAB.

Якщо в імені файлу зображення не має інформації щодо шляху до даного файлу, то filename шукається у поточній папці. І якщо його там немає, то здійснюється пошук даного файлу у всіх папках, шляхи до яких наведено в шляху пошуку MATLAB. Найпростіший спосіб зчитати зображення з певної конкретної папки – це вказати повний чи відносний шлях до цієї паки в рядку filename.

Таблиця 1.1 – Деякі графічні формати, що розпізнається командами `imread` та `imwrite`, починаючи з MATLAB 6.5. Більш ранні версії підтримували не всі формати

Формат зображення	Розшифровка скорочення	Допустимі зображення
TIFF	Tagged Image Format File	.tif, .tiff
JPEG	Joint Photographic Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

Наприклад, команда

```
>> f=imread ('D:\myimage\chestxray.jpg');
```

зчитує зображення з папки `myimage` на диску `D:`, а команда

```
>> f=imread ('.\myimage\chestxray.jpg');
```

завантажує зображення з підпапки `myimage` поточної робочої папки. Поточна робоча папка MATLAB відображається в рядку інструментів робочого стола, де її легко змінити вручну.

Функція `size(f)` повертає розмір зображення, а саме число рядків і стовпчиків

```
>> size(f)
   ans=
 1024 1024
```

Ця функція буде особливо корисною при автоматичному визначенні розміру зображення, що виконується операцією

```
>> [M,N]=size(f)
```

При такій формі запису змінній `M` буде присвоєно число рядків зображення, а змінній `N` – число стовпчиків.

Функція `whos` повідомляє додаткову інформацію про масив. Наприклад, рядок

```
>> whos f
```

дає наступний результат

```
Name      Size              Bytes         Class
f         1024×1024         1048576       uint8 array
      Grand total is 1048576 elements using 1048576 bytes
```

Запис `uint8` означає один з класів MATLAB. Якщо вкінці команди `whos` поставити крапку з комою, то це ні до чого не призведе, тому крапку з комою можна не ставити.

Виведення зображення на дисплей. Зображення на дисплей комп'ютера можна вивести за допомогою функції `imshow`, яка має наступний синтаксис:

```
imshow(f,G)
```

де `f` – матриця зображення, а `G` – це число рівнів яскравості, що застосовується при відображенні цього зображення. Якщо аргумент `G` не використовується, то по замовчуванню застосовується 256 рівнів яскравості. Команда

```
imshow(f,[low high])
```

визначає, що всі пікселі зі значенням не більше числа `low` треба показувати чорними, а всі пік селі зі значенням не менше числа `high` – білими. Всі проміжні значення показуються з проміжною яскравістю з використанням числа рівнів, що прийнято за замовчуванням.

Запис в командному рядку

```
imshow(f,[ ])
```

встановлює для змінної `low` мінімальне значення масиву `f`, а змінній `high` його максимальне значення. Така форма функції `imshow` буває корисною в тому випадку, коли необхідно показати зображення, що має вузький динамічний діапазон значень пікселів, або коли серед них є додатні та від'ємні значення.

Функція `pixval` часто використовується для інтерактивного визначення значень яскравості окремих пікселів. Ця функція відображає курсор, що розміщений поверх зображення. Курсор переміщується по зображенню разом з мишкою, а під вікном зображення відображаються поточні координати курсору та значення інтенсивності в цій точці. При роботі з кольоровим зображенням разом з координатами відображається інтенсивність (яскравість) червоної, зеленої

та синьої компоненти кольорового пікселя. При натисканні та утриманні лівої клавіші мишки функція `pixval` показує евклідову відстань від початкового до поточного положення курсору. Синтаксична форма цієї команди має вигляд

```
pixval,
```

яка встановлює курсор на останнє виведене на дисплеї зображення. Натиснення кнопки X у вікні курсора відключає курсор на зображенні.

Збереження зображень. Зображення записуються на диск функцією `imwrite`, яка має наступний синтаксис:

```
imwrite(f,'filename')
```

При такому запису рядок `filename` має містити розширення , що підтримує система MATLAB. В іншому випадку необхідний формат можна задати явно за допомогою використання третього аргументу функції. Наприклад, наступна команда, записує матрицю `f` в зображення з форматом TIFF з ім'ям `patient10_run1`.

```
imwrite(f,'patient10_run1','tif')
```

або

```
imwrite(f,'patient10_run1.tif')
```

Якщо рядок `filename` не має інформації про шлях, то функція `imwrite` записує файл в поточну робочу папку.

Більш загальний синтаксис функції `imwrite`, що застосовується тільки для файлів JPEG, має наступний вигляд:

```
imwrite(f,'filename.jpg', 'quality', q)
```

де `q` – це ціле число в інтервалі від 0 до 100 (чим менше це число, ти вище спотворення при стисненні фалу у форматі JPEG).

Щоб зрозуміти ідею осягненої компресії (стиснення) та отримати іншу інформацію про файл зображення , можна використати функцію `iminfo`, яка має вигляд

```
iminfo filename
```

де `filename` – повне ім'я, наприклад bubbles25.jpg.

Часто буває необхідно зберегти зображення на диску точно в тому вигляді, в якому воно відображається на екрані. Особливо важливо це вміти робити для графіків та діаграм. Вміст вікна зображень можна експортувати на диск двома способами. Перший спосіб - це використати команду ***Export***, в меню ***File*** у вікні зображень. Потім користувач має змогу вибрати папку на диску, ім'я та формат файлу. Ще більші можливості для вибору параметрів експорту дає команда `print`:

```
print -fno -dfileformat -rresno -filename,
```

де *no* – це номер вікна необхідного зображення, *fileformat* – це один з допустимих параметрів зображення (див. табл. 1.1), *resno* – це роздільна здатність в dpi, а `filename` – це ім'я під яким користувач хоче зберегти зображення. Наприклад:

```
print -f1 -dtiff -r300 hi_res_rose
```

Ця команда розмітить побудований файл `hi_res_rose` в поточну папку.

Класи даних. Незважаючи на те, що роботу необхідно вести з цілочисленними координатами, значення самих пік селів необов'язково мають бути цілими числами. В табл. 1.2 перераховані різні класи даних, що підтримує система MATLAB та пакет IPT. Перші вісім записів, що наведені в таблиці, належать до числових класів даних. Дев'ятий запис – це клас символів, а останній запис належить до логічного класу даних.

Всі числові операції в MATLAB виконуються з подвійною точністю в класі даних *double*, який часто використовується в додатках для обробки зображень. Клас uint8 також зустрічається дуже часто, особливо при зчитуванні даних із запам'ятовуючих пристроїв, наприклад, найбільш поширених на практиці 8-бітних зображень. Ці два класи, клас logical, а також в меншій мірі клас uint16, утворюють першостепеневі класи даних, на яких буде сфокусована наша увага. Клас даних double вимагає 8 байт для представлення одного числа, клас uint8 використовує один байт на кожний елемент, типам uint16 та int16 необхідно по 2 байти, а типи uint32, int32 та single замають в пам'яті по 4 байти на кожний елемент. Символьний клас даних char зберігає букви та символи в кодуванні Unicode, в якій використовується два байти на елемент. Символьний рядок представляє собою масив 1×n символів класу char. Масив logical складається з елементів, що приймають тільки два значення 0 та 1, які зберігаються в пам'яті, наймаючи по одному байту кожне.

Таблиця 1.2 – Класи даних. Перші вісім класів називаються числовими, дев'ятий символьний клас, останній клас - логічний

Ім'я	Опис
double	Числа з плаваючою комою подвійної точності в діапазоні, приблизно, від -10^{308} до 10^{308} (8 байт на число).
uint8	Цілі без знаку в інтервалі [0, 255] (1 байт на число)
uint16	Цілі без знаку в інтервалі [0, 65535] (2 байти на число)
uint32	Цілі без знаку в інтервалі [0, 4294967295] (4 байти на число)
int8	Цілі зі знаком в інтервалі [-128, 127] (1 байт на число)
int16	Цілі зі знаком в інтервалі [-32768, 32767] (2 байти на число)
int32	Цілі зі знаком в інтервалі [-2147483648, 2147483647] (4 байти на число)
single	Числа з плаваючою комою, звичайної точності в діапазоні, приблизно, від -10^{38} до 10^{38} (4 байти на число).
char	Символи (букви та знаки) (2 байти на число)
logical	Значення 0 чи 1 (1 байт на число)

Хід виконання роботи

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

```
i = imread('filename.ext');
```

де filename – ім'я файлу; ext – його розширення.

Зображення, що відповідають варіантам 1-10, знаходяться у підкаталогах MATLAB; зображення, що відповідають варіантам 11-14, видаються викладачем і записуються до підкаталогу …\work системи 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 у байтах.
4. Перевірити правильність розрахунку за допомогою команди `whos`.
5. Вивести зображенне на екран за допомогою команди `imshow`:

```
imshow(i)
```

В чому полягає відмінність між результатами виконання команд `imshow(i)`и `imshow(i,[])`?

6. Вивести одночасно на екран результати роботи команд `imshow(i)`та `imshow(i,[])`. Для цього скористатись: а) командою `figure`, б) командою `subplot(i,j,k)`, *i* – кількість зображень виведених по горизонталі, *j* - кількість зображень виведених по вертикалі, *k* – номер активної області рисунка (вікна), куди буде виведено зображенне.

В чому відмінність у роботі цих двох команд?

7. Зберегти на жорсткий диск до поточного каталогу зображення у форматі JPEG з налаштуваннями якості q = 25, 5, 0. Для цього застосувати команду `imwrite`:

```
imwrite(i,'filename.jpg','quality',q)
```

Переглянути збережені зображення за допомогою стандартних засобів операційної системи.

Варіанти завдання

Варіант	Файл зображення
1	pout.tif
2	cameraman.tif
3	eight.tif
4	tire.tif
5	rice.png
6	circuit.tif
7	moon.tif
8	coins.png
9	liftingbody.png
10	cell.tif
11	building.tif
12	Moon Phobos.tif
13	pollen.tif
14	test_pattern.tif

Зміст звіту

Звіт про виконаннялабораторної роботи має містити:

- титульну сторінку;
- назву та мету роботи;

- листинги програми;
- зображення відповідно до п. 5-6;
- висновки відповідно до п. 4-6.

Орієнтовний перелік питань, що винесені на захист роботи:

1. Як представляється зображення у системі MATLAB?
2. Які команди використовують для зчитування зображення до рабочої 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`?

Лабораторна робота №2

ПРОСТОРОВІ МЕТОДИ ПОКРАЩЕННЯ ЗОБРАЖЕННЯ

Мета роботи: ознайомитись з просторовими методами покращення зображень, застосовуючи систему моделювання MATLAB з набором інструментів Image Processing Toolbox; проаналізувати ефективність розглянених методів для різних типів зображень.

Засоби виконання: пакет MATLAB 6.5/7.* зі встановленим набором інструментів Image Processing Toolbox версії 3.*/4.*.

Теоретичні відомості

Термін *просторова область* стосується площини зображення в цілому і методи з цієї категорії базуються на прямих маніпуляціях з пікселами зображень. Процеси у просторовій області можна визначити 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 певній околиці точки (x,y). Крім того, оператор T може обробляти послідовність зображень, наприклад, може додавати K вхідних зображень для притлумлення шуму.

Головний підхід до визначення просторової околиці навколо точки (x,y) полягає у використанні квадратної або прямокутної області з центром в точці (x,y), як показано на рис. 1.1. Центр заданої шаблонної підобласті переміщується від піксела до піксела, починаючи, наприклад, з верхнього лівого кута, і на своєму шляху він накриває різні околиці. Перетворення T застосовується у кожній точці (x,y), що в результаті дає вихідне (оброблене) значення g для даної точки. Під час обчислень використовується тільки пікселі всередині заданої околиці з центром в (x,y).

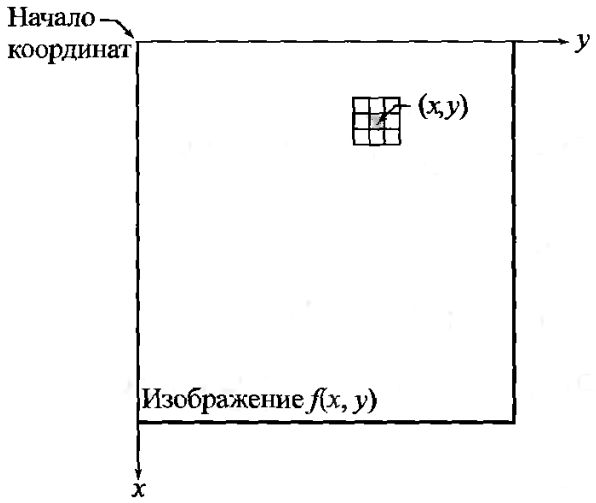

Рисунок 2.1 – Околиця розміром 3х3 навколо точки (*x*,*y*) зображення

Моделі шуму

Гаусів шум. Математична простота, характерна для роботи з моделями гауссового шуму (також називаного нормальним шумом) як у просторової, так й у частотній області, обумовила широке поширення цих моделей на практиці. Насправді ця простота виявляється настільки привабливої, що найчастіше гауссові моделі використаються навіть у тих ситуаціях, коли їхнє застосування виправдане, у найкращому разі, лише частково.

Функція щільності розподілу ймовірностей гауссової випадкової величини z задається виразом

$$p(z)=\frac{1}{\sqrt{2\pi\sigma}}e^{-(z-\mu)^2/2\sigma^2},$$

де *z* являє собою значення яскравості, μ – середнє значення випадкової величини *z*, σ – її средньоквадратичне відхилення. Квадрат средньоквадратичного відхилення σ^2 називається дисперсією величини *z*. Графік цієї функції представлений на рис. 1.2, а. Коли щільність розподілу випадкової величини *z* описується функцією (1.3), те приблизно 70% її значень попадають у діапазон [(μ - σ), (μ +σ)], і приблизно 95% – у діапазон [(μ - 2σ), (μ +2σ)].

Імпульсний шум. Функція щільності розподілу ймовірностей (біполярного) імпульсного шуму задається виразом

$$p(z)=\begin{cases} P_a & при\ z=a; \\ P_b & при\ z=b; \\ 0 & в\ інших\ випадках. \end{cases}$$

Якщо $b > a$, то пиксель із яскравістю b виглядає як світла крапка на зображенні. Пиксель із яскравістю a виглядає, навпаки, як темна крапка. Якщо одне зі значень імовірності (P_a або P_b) дорівнює нулю, то імпульсний шум називається уніполярним. Якщо жодна з імовірностей не дорівнює нулю, і особливо якщо вони приблизно рівні за величиною, імпульсний шум подібний до крупиць солі й перцю, випадково розсипаних по зображенню. Із цієї причини імпульсний шум називають також шумом типу «сіль і перець». Ми будемо використати обидві назви як взаємозамінні.

Значення імпульсів шуму можуть бути як позитивні, так і негативні. При оцифровці зображення звичайно відбувається масштабування (і обмеження) значень яскравості. Оскільки величина пов'язаних з імпульсним шумом спотворень як правило велике в порівнянні з величиною корисного сигналу, імпульсний шум після оцифровки приймає екстремальні значення, що відповідає появі абсолютно чорних і білих крапок на зображенні. Тому зазвичай передбачається, що значення a й b є «інтенсивними» у тому розумінні, що вони рівні мінімальному й максимальному значенням, які в принципі можуть бути присутнім в оцифрованому зображенні. У результаті негативні імпульси виглядають як чорні крапки на зображенні (перець). З тих же причин позитивні імпульси виглядають як білі крапки (сіль). Для 8-бітових зображень це означає, що a = 0 (чорне) і b = 255 (біле). На рис. 1.2, б представлений графік розподілу ймовірностей значень імпульсного шуму

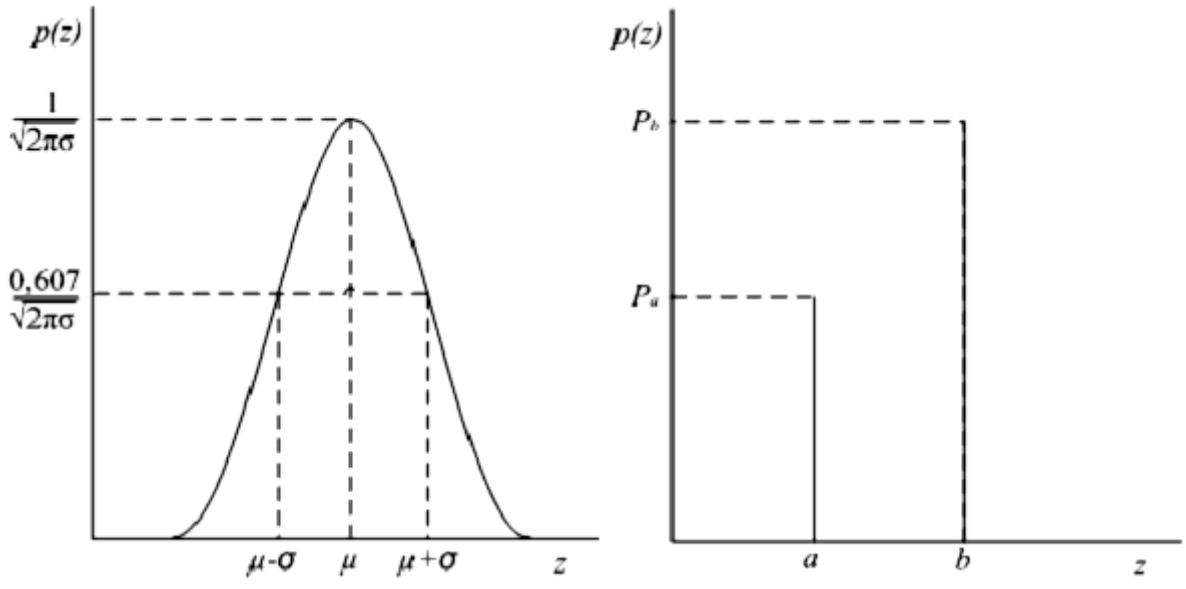

Рисунок 2.2 – Функції щільності розподілу імовірності: а – розподіл Гауса; б – імпульсний розподіл

Додавання шуму функцією **`imnoise`**

В пакеті є функція imnoise, яка моделює спотворення ізо-браженія деяким шумом. Синтаксис цієї функції має вигляд

g = imnoise(f, type, parameters)

де f — це вихідне зображення, а сенс аргументів type і parameters буде пояснений пізніше. Функція imnoise спочатку перетворить зображення в клас double в діапазоні [0,1]. Це слід мати на увазі перед завданням пари-метрів шуму. Наприклад, щоб додати шум з середнім 64 і дисперсією 400 до зображення класу uint8, слід стискувати середнє до величини 64/256, а дисперсію прирівняти до 400/(256) 2, після чого ці параметри можна підставляти у функцію imnoise. Розглянемо різні синтаксичні форми цієї функції.

g = imnoise(f, 'gaussian', m, var) додає до зображення f шум гауса з середнім m і дисперсією var. За умовчанням, m = 0 і var = 0.01.

g = imnoise(f, 'localvar', V) додає до зображення f локальний шум гауса з нульовим середнім, дисперсія якого в кожній точці зображення f задається матрицею V, розміру як в f.

g = imnoise(f, 'localvar', image_intensity, var) додає до зображення f гаусів шум з нульовим середнім, в якому локальна дисперсія шуму var являється функцією значень яскравості зображення f. Аргументи image_intensity і var є векторами однакової розмірності, а функція plot(image_intensity, var) будує графік залежності дисперсії var від яскравості image_intensity. Вектор image_intensity повинен містити нормовані значення яскравості в діапазоні [0,1].

g = imnoise (f, 'salt & paper', d) псує зображення f шумом «сіль і перець», де d – це щільність шуму (тобто відсоток зображення, схильного до цього шуму). При цьому приблизно d*numel(f) пікселів буде зіпсовано. За умовчанням, d = 0.05.

g = imnoise(f, "speckle", var) додає до f мультиплікативний шум по формулі g = f "+" n*f, де n - це рівномірно розподілений шум з нульовим середнім і дисперсією var. За умовчанням, var = 0.04.

g = imnoise(f, 'poisson') генерує пуассоновський шум, залежний від вихідних даних, замість додавання штучного шуму. Щоб узгоджуватися із статистиками Пуассона, пікселі зображень uint8 і uintl6 повинні відповідати числу фотонів (або іншим квантам інформації). Використовуються зображення з подвійною точністю, коли число фотонів на один піксел більше, ніж 65535 (але менше 10^{12}). Величини яскравості пікселів міняються в межах від 0 до 1 і відповідають числу фотонів, що ділиться на 10^{12}.

Огляд фільтрів

Усереднюючі фільтри. Фільтри, основані на обчисленні середнього арифметичного, називають середньоарифметичним. Він є найпростішим серед усереднюючих фільтрів. Нехай S_{xy} позначає прямокутну околицю (множина координат точок зображення) розмірами $m \times n$ із центром у точці (x,y). Процедура фільтрації припускає обчислення середнього арифметичного значення спотвореного зображення $g(x,y)$ по околиці S_{xy}. Значення відновленого зображення $\hat{f}$ в довільній точці (x,y) являє собою середнє арифметичне значень у точках, що належить околиці S_{xy}. Іншими словами:

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s,t).$$

Ця операція може бути реалізована у вигляді згортки з маскою, всі коефіцієнти якої рівні $1/mn$. Усереднюючий фільтр просто згладжує локальні варіації яскравості на зображенні. Зменшення шуму відбувається в результаті цього згладжування.

Медіанні фільтри. Найбільш відомим з фільтрів, основаних на порядкових статистиках, є медіанний фільтр. Дія цього фільтру полягає в заміні значення в точці зображення на медіану значень яскравості в околиці цієї точки:

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

При обчисленні медіани значення в самій точці (тобто в центрі околиці) також ураховується. Широка популярність медіанних фільтрів обумовлена тим, що вони 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 добре працюють доти, поки просторова щільність імпульсного шуму невелике (емпіричне правило – P_a й P_b не перевищують 0,2). Адаптивна медіанна фільтрація допомагає впоратися з імпульсним шумом, імовірності якого перевищують вказані значення. Додаткова перевага 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 час роботи відповідно до наведеного нижче умовами. Будемо пам'ятати про те, 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 А и область Б, і його дія полягає в наступному.

Область А:	Область Б:
$A1 = z_{med} - z_{min}$; $A2 = z_{med} - z_{max}$; якщо $A1 > 0$ й $A2 < 0$, перейти до області Б; інакше збільшити розмір околиці; якщо розмір околиці $\leq S_{max}$ повторити область А; інакше результат дорівнює z_{xy}	$B1 = z_{xy} - z_{min}$; $B1 = z_{xy} - z_{max}$; якщо $B1 > 0$ и $B2 < 0$, 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 шум, 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 вони не рівні найменшим і найбільшому можливим значенням яскравості на зображенні.

З урахуванням останнього зауваження ми бачимо, що область А 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, то в силу зазначених у попередньому абзаці причин z_{med} не може бути імпульсом. У цьому випадку ми переходимо до області Б и перевіряємо, чи є імпульсом значення z_{xy} у тій точці, що відповідає центру околиці (нагадаємо, що ми будуємо відгук фільтра в цій точці). Якщо умови $B1>0$ й $B2<0$ виконані, то $z_{min} < z_{xy} < z_{max}$, 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 точках «проміжного рівня» яскравості мінімізує перекручування, внесені обробкою зображення. Якщо одне з умов $B1 > O$ і $B2 < 0$ порушено, то 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 А, 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.

Фільтри, основані на виборі максимального й мінімального значення. Хоча медіанні фільтри, безумовно, належать до числа найбільше часто використовуваних в обробці зображень фільтрів, основаних на порядкових статистиках, це аж ніяк не єдиний приклад таких фільтрів. Медіана являє собою 50-ий процентиль упорядкованого набору чисел, однак використання інших статистичних характеристик надає багато інших можливостей. Наприклад, використання 100-го процентиля призводить до фільтра, основаного на виборі максимального значення (або фільтру максимуму), що задається виразом

$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\}.$$

Такий фільтр корисний при виявленні найбільш яскравих точок на зображенні. Крім того, оскільки уніполярний «чорний» імпульсний шум приймає мінімальні значення, застосування цього фільтра призводить до зменшення такого шуму, тому що в процесі фільтрації з околиці S_{xy} вибирається максимальне значення.

Використання 0-го процентиля призводить до використання фільтра, основаного на виборі мінімального значення (або фільтру мінімуму):

$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}.$$

Такий фільтр корисний при виявленні найбільш темних точок на зображенні. Крім того, застосування цього фільтра приводить до зменшення уніполярного «білого» імпульсного шуму внаслідок операції вибору мінімуму.

Хід виконання роботи

1. Зчитати зображення з файлу (відповідно до варіанту завдання), використовуючи команду *imread,* та вивести його на екран за допомогою команди *imshow*:

```
I=imread('pout.tif');
imshow(I)
```

2. Побудувати гістограму заданого зображення за допомогою команди *imhist:*

```
imhist(I)
```

3. Виконати покращення зображення за допомогою наступних команд:

histeq – еквалізація (вирівнювання) гістограми `I1=histeq(I);`

Зробити висновок про вплив операції еквалізаціїна початкове зображення.

imadjust - корекція контрасності та яскравості зображення

```
I2=imadjust(I, [low_in high_in],[low_out high_out],gamma);
```

Побудувати гістограми зображення для випадків:

а) γ=1, γ=0,5, γ =2,1 при порогах обмеження значення пікселя `[low_in high_in],[low_out high_out]` по замовчуванню `[0 1][0 1];`

б) `[0.2 0.6],[0 1]` при γ =1;

в) `[0 1],[0.2 0.6]` при γ =1.

Зробити висновок про вплив коефіцієнту гамма-корекції та порогів обмеження на характер зображення.

adapthisteq (тільки для MATLAB 7.*) – адаптивне вирівнювання гістограми по локальній області `I3=adapthisteq(I);`

Порівняти результати роботи функцій *histeq* та *adapthisteq*, зробити висновки щодо ефективності застосування цих двох методів.

4. Додати до початкового зображення шум за допомогою команди *imnoise*:

```
I_noise=imnoise(I,'вид шума');
```

Види шуму, які можна задати в Image Processing Toolbox:

- гаусівський шум ('gaussian');
- імпульсний шум ('salt & pepper');
- пуасонівський шум ('poisson');
- мультиплікативний шум('speckle').

Побудувати гістограми зашумлених зображень для наведених вище видів шуму та зробити висновки про характер розподілу яскравості пікселів залежно від виду шуму.

5. Для **парних** варіантів виконати фільтрацію зображень, зашумлених *гаусівським* шумом, за допомогою наступних видів просторових фільтрів:

- медіанного фільтру, як підвиду рангової фільтрації;

- адаптивного медіанного фільтру (виконати порівняння тільки з медіанним фільтром);
- фільтру усереднення (згладжувального фільтру).

Для **непарних** варіантів зробити фільтрацію зображень, зашумлених *імпульсним* шумом, за допомогою наступних видів просторових фільтрів:

- медіанного фільтру, як підвиду рангової фільтрації;
- адаптивного медіанного фільтру (виконати порівняння тільки з медіанним фільтром);
- фільтра максимуму;
- фільтра мінімуму.

Навести відфільтровані зображення, зробити висновки щодо ефективності роботи наведених фільтрів залежно від виду шуму.

а) Медіанна фільтрація здійснюється за допомогою команди *medfilt2:*

```
I_filt=medfilt2(I_noise, [n m]);
```

Де *n*×*m* – розмір фільтруючої маски (околиці). За замовчуванням *n=m=3.*

б) Адаптивна медіанна фільтрація виконується за допомогою команди *adpmedian:*

```
I_filt=adpmedian(I_noise, Smax);
```

Де `Smax` – максимальний розмір маски. Для порівняння дії фільтрів *medfilt2* та *adpmedian* встановити *Smax=n=m=7.*

Зробити висновки про ефективність застосування цих фільтрів.

в) Фільтри максимуму та мінімуму реализуються за допомогою команди рангової фільтрації *ordfilt2:*

```
I_filt=ordfilt2(I_noise, order, ones(n,m));
```

Де *n*×*m* – розмір фільтруючої маски (околиці). За замовчуванням *n=m=3.* Для фільтра мінімуму `order=1`, для фільтру максимуму – `order=n*m`.

г) Фільтр усереднення задається за допомогою команди формування стандартних фільтрів *fspecial* та застосування його до зображення виконується за командою *imfilter:*

```
filt=fspecial('average', [n m]);
I_filt=imfilter(I_noise, filt);
```

Перевірити вплив розміру маски на ступінь фільтрації та вид зображення, зробити відповідні висновки.

Варіанти завдання

Варіант	Файл зображення
1	pout.tif
2	cameraman.tif
3	eight.tif

4	tire.tif
5	rice.png
6	circuit.tif
7	moon.tif
8	coins.png
9	liftingbody.png
10	cell.tif
11	building.tif
12	Moon Phobos.tif
13	pollen.tif
14	test_pattern.tif

Зміст звіту

Звіт про виконання лабораторної роботи має містити:

- титульну сторінку;
- назву та мету роботи;
- листинги програми;
- графіки (гістограми) та зображення відповідно до п. 2-5;
- висновки відповідно до п. 3-5.

Зауваження: 1. Листинги програми – окремо для п. 1-3, п. 4 та п. 5 роботи. 2. Не забувати використовувати для виведення зображень та гістограм на екран команди *figure* и *subplot.*

Орієнтовний перелік питань, що винесені на захист роботи:

1. Що таке гістограма?
2. Що таке вирівнювання гістограми?
3. Що таке гамма-корекція?
4. В чому полягає метод локального покращення зображення?
5. В чому полягає просторова фільтрація?
6. Які методи просторової фільтрації вам відомі?
7. В чому полягає робота фільтру усереднення?
8. Що лежить в основі рангової фільтрації?
9. Який принцип дії медіанного фільтру?
10. Який алгоритм роботи адаптивного медіанного фільтру?
11. Які команди системи MATLAB реалізують методи просторової фільтрації?
12. Поясніть використання функції *imadjust* для покращення изображения.

Лабораторна робота №3

ДИСКРЕТНЕ ПЕРЕТВОРЕННЯ ФУР'Є. ЧАСТОТНА ФІЛЬТРАЦІЯ ЗОБРАЖЕНЬ

Мета роботи: ознайомитись зі способами представлення зображень в області просторових частот; 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 у системі MATLAB.

Засоби виконання: пакет MATLAB 6.5/7 зі встановленим набором інструментів Image Processing Toolbox версії 3.*/4.*.

Теоретичні відомості

Знаходячись в основі методів лінійної фільтрації, перетворення Фур'є забезпечує значну гнучкість при розробці і реалізації алгоритмів фільтрації при рішенні завдань поліпшення, відновлення і 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 $x = 0,1,2, ...,M - 1$ і $y = 0,1,2, ...,N - 1$, означає зображення *M*×*N*. Двомірне *дискретне перетворення Фур'є* (DFT, Descrete Fourier Transform) зображення *f*, яке 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 $u = 0,1,2,..., M - 1$ і $v = 0,1,2,..., N - 1$. Ми 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*×*N*, яка задається при $u = 0,1,2,..., M - 1$ і $y = 0,1,2,..., N - 1$, прийнято називати *частотним прямокутником*. Видно, що частотний прямокутник має ті ж розміри, що і 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 $x = 0,1,2,...,\ M-1$ и $y = 0,1,2,...,\ N-1$. Таким чином, знаючи *F(u,v)*, можна відновити *f(x,y)* з допомогою зворотного ДПФ. Величини *F(u,v)* в цих рівняннях прийнято називати *коефіцієнтами розкладання Фур'є*.

В деяких визначеннях коефіцієнт *1/MN* поміщається в пряме перетворення, а в інших — в 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 приведено 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*)* и *f(*0,0*)*, які знаходяться в прямому і зворотному перетвореннях.

Значення перетворення Фур'є у початку координат частотної області (тобто величина *F(*0,0*)*) називається коефіцієнтом або компонентою *dc* перетворення Фур'є.

Фільтрація в частотній області. Фільтрація в частотній області має вельми просту концепцію. Основою лінійної фільтрації в частотної і просторової області є теорема про згортку, яку можна сформулювати так:

f(x,y) h(x, y) <=> H(u,v) F(u,v)*

і в зворотну сторону

*f(x,y) h(x, y) <=> H(u,v) * F(u,v)*

Тут символ «*» позначає операцію згортки двох функцій, а вирази по обидві сторони від подвійних стрілок визначають відповідні пари 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 і друге твердження.

Відповідно до теореми про згортку, той же результат можна отримати в частотної 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 частотної 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(u,v)* має ті ж розміри, що і вихідне зображення.

1. Отримати параметри розширення з допомогою `paddedsize`:

```
PQ = paddedsize(size(f));
```

2. Побудувати перетворення Фур'є з розширенням:

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

3. Згенерувати функцію фільтру `H` розміром `PQ(1)xPQ(2)` одним з описаних далі методів. Якщо він був центрований, 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 = real(ifft2(G));
```

6. Вирізати верхній лівий прямокутник початкових розмірів:

```
g = g(1:size(f,1), 1:size(f,1));
```

Ця процедура фільтрації схематично змальована на рис. 2.1. Попередня стадія обробки може складатися з визначення розмірів зображення, обчислення параметрів розширення і 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.

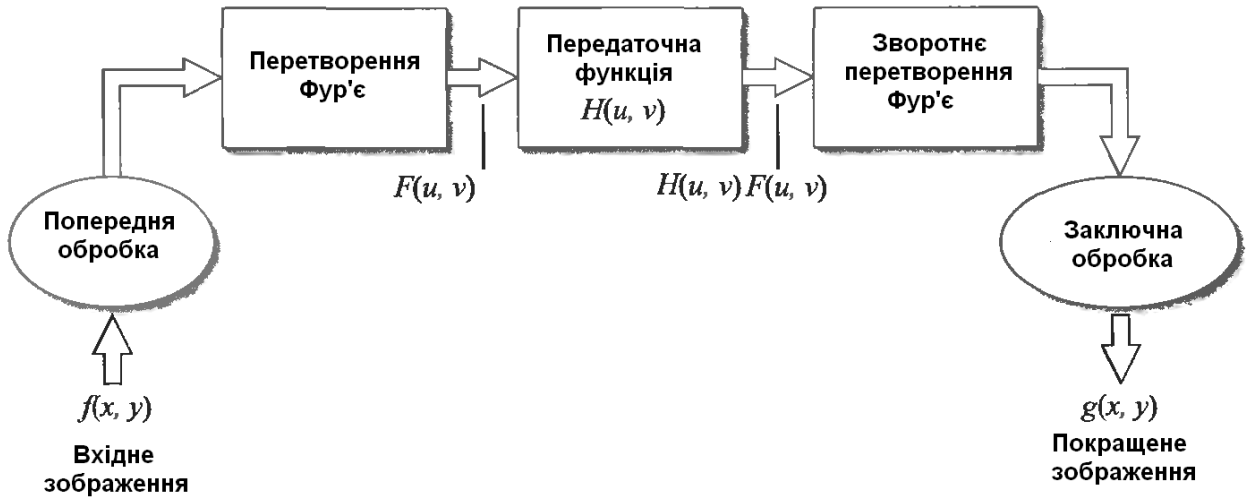

Рисунок 3.1 – Основні кроки фільтрації в частотній області

Передаточна функція фільтра *H(u,v)* на рис. 2.1 множиться на дійсну і уявну частини *F(u,v)*. Якщо функція *H(u,v)* була дійсною, то фазова частина виразу не міняється, що видно з фазового рівняння, оскільки при множенні дійсної і уявної частини комплексного числа на одне і те ж дійсне число фазовий кут не

міняється. Такі фільтри прийнято називати фільтрами з нульовим зрушенням фази.

М-функція для фільтрації в частотної області. Аргументами М-функції є зображення і передаточна функція фільтру, яка виконує всі необхідні процедури фільтрації і повертає відфільтроване і обрізане зображення.

Побудова фільтрів в частотної області по просторовим фільтрам. В загальному випадку фільтрація в просторової області є ефективнішою з обчислювальної точки зору, ніж фільтрація в частотної області, коли використовуються «малі» фільтри. Фільтрація з використанням алгоритму швидкого перетворення Фур'є FFT виконується швидше, ніж просторова реалізація цього процесу, коли функція має приблизно 32 точки, тобто їх число невелике. Тому корисно знати, як конвертувати просторові фільтри в еквівалентну частотну форму, щоб мати можливість порівняти ці два підходи.

Низькочастотні фільтри. Ідеальний низькочастотний фільтр (ILPF, Ideal Lowpass Filter) має передаточну функцію

$$H(u,v)=\begin{cases}1 & при \quad D(u,v)\le D_0\\ 0 & при \quad D(u,v)>D_0\end{cases}$$

де D_0 — це задане число >0, а $D(u,v)$ – відстань від центру фільтра до точки (u, v). Геометричне місце точок (u,v), для яких $D(u, v) = D_0$, являє собою коло.
Пам'ятаючи про те, що фільтр H множиться на перетворення Фур'є зображення, можна сказати, що ідеальний фільтр «зрізає» (множить на нуль) всі компоненти F, що знаходяться поза цим колом, і залишає незмінними (множить на 1) всі компоненти, що знаходяться усередині або на кордоні кола. Не дивлячись на те, що цей фільтр неможливо реалізувати на практиці в аналоговій формі за допомогою електронних компонент, його, безумовно, можна змоделювати на комп'ютері за допомогою заданої передаточної функції. Властивості ідеального фільтра часто бувають корисними при поясненні таких явищ, як *помилки перекриття.*

Низькочастотний фільтр Баттерворта (BLPF, Butterworth LowPass Filter) порядку n з обрізанням частот на відстані D_0 від початку координат має передаточну функцію

$$H(u,v)=\frac{1}{1+\left[D(u,v)/D_0\right]^{2n}}$$

На відміну від ідеального низькочастотного фільтру, функція фільтру BLPF не має розриву в пороговій точці D_0. Для фільтрів з гладкою передаточ-

ною функцією прийнято задавати частоту зрізання, яка визначається розташуванням точок, для яких функція *H(u,v)* менше певної долі її максимального значення. У попередньому рівнянні значення $H(u,v)= 0.5$ (тобто, 50% від максимального значення, яке дорівнює 1), коли $D(u,v) = D_0$.

Підвищення різкості при частотній фільтрації. На противагу низькочастотній фільтрації, яка призводить до розмиття зображень, високочастотна фільтрація підвищує різкість зображення, ослаблюючи низькі частоти і залишаючи високі частоти перетворення Фур'є відносно незмінними.

Основи високочастотної фільтрації. Маючи передаточну функцію $H_{lp}(u,v)$ низькочастотного фільтру, можна отримати передаточну функцію відповідного високочастотного фільтра за допомогою формули

$$H_{hp}(u,v) = 1 - H_{lp}(u,v).$$

Хід виконання роботи

1. Зчитати зображення з файлу (відповідно до варіанту завдання), використавши команду *imread*.
2. Отримати дискретне перетворення Фур'є від зображення:

```
F=fft2(I);%ДПФ від матриці зображення
F2=log(abs(F));%Для візуалізації ДПФ необхідно взяти його модуль у логарифмічній шкалі
imshow(F2[-5 15]); colormap(jet); colorbar
```

Проаналізувати отримане зображення і відповісти на питання, де знаходиться початок координат на отриманій картинці.

Для відцентровування результату ДПФ виконати наступну команду:

```
F1=fftshift(F);
```

Проаналізувати отримане зображенне, охарактеризувавши ділянки низьких та високих просторових частот.

3. Застосувати спеціальні фільтри до зображення, використавши команду *fspecial*:

```
H=fspecial('вид фільтру');
figure
freqz2(H);%Виводить АЧХ фільтру на екран
I2=imfilter(I,H);% Здійснюється фільтрація вихідного зображення
```

Отримати ДПФ відфільтрованого зображення та пояснити результат фільтрації, використовуючи представлення зображення у частотній та просторових областях.

Вид фільтру:

- 'gaussian' – гаусів НЧ фільтр;

- ‘average’ – фільтр усереднення НЧ;
- ‘laplacian’ – ВЧ фільтр Лапласа;
- ‘motion’ – фільтр, який створює ефект руху пікселів у визначеному напрямку.

4. Проектування фільтрів.

4.1. За заданою АЧХ за допомогою команди *fsamp2*:

```
[u,v]=freqspace(20,'meshgrid');
 %Створюємо маску фільтру ВЧ
 Hd=zeros(20);
 R=sqrt(u.^2+v.^2);Hd(R>0.15)=1;%0.15 – відносна частота зрізу
 figure
 H=fsamp2(Hd);%Створення фільтру по його масці
 freqz2(H);
 I2=imfilter(I,H);
```

Фільтр застосовувати до зображення *відповідно до номеру завдання*. Отримати ДПФ відфільтрованого зображення та пояснити результат фільтрації, використовуючи представлення зображення у частотній та просторовій областях.

Проаналізувати вплив на кінцеве зображення величини відносної частоти зрізу фільтру.

4.2. Повторити п.4.1, створивши фільтр НЧ. Для цього замінити команду *zeros* на *ones* та присвоїти *Hd(R>0.15)=0*.

4.3. Методом перетворення частот за допомогою команди *ftrans2*:

```
B=fir1(6,0.3);%Створення коеф. одновимірного ФНЧ 6-го порядку з відносною частотою зрізу 0,3
H=ftrans2(B);
figure
freqz2(H);
I2=imfilter(I,H);
```

Фільтр застосувати до зображення *відповідно до номеру завдання*. Отримати ДПФ відфільтрованого зображення та пояснити результат фільтрації, використовуючи представлення зображення у частотній та просторовій областях.

4.4. Повторити п.4.3, створивши фільтр ВЧ. Для цього у команду *fir1* внести наступні зміни:

```
B=fir1(6,0.3,'high');
```

Варіанти завдання

Варіант	Файл зображення
1	pout.tif
2	cameraman.tif
3	eight.tif

4	tire.tif
5	rice.png
6	circuit.tif
7	moon.tif
8	coins.png
9	liftingbody.png
10	cell.tif
11	building.tif
12	Moon Phobos.tif
13	pollen.tif
14	test_pattern.tif

Зміст звіту

Звіт про виконання лабораторної роботи повинен містити:

- титульну сторінку;
- назву та мету роботи;
- листинг програми;
- ДПФАЧХ та зображення відповідно до п. 2-4;
- висновки відповідно п. 2-4.

Орієнтовний перелік питань, які винесено на захист роботи:

1. Що таке двовимірне ДПФ та який його фізичний зміст?
2. Чому дорівнює ДПФ на початку координат? Який його фізичний зміст?
3. Чому відповідають області НЧ та ВЧ на зображенні?
4. Що таке центрування ДПФ і як воно здійснюється математично?
5. Як впливають фільтри НЧ та ВЧ на зображення?
6. Що буде із зображенням, якщо постійну складову спектру подавити?
7. Які команди MATLAB виконують фільтрацію зображення у частотній області?

Лабораторна робота №4

ДИСКРЕТНЕ КОСИНУСНЕ ПЕРЕТВОРЕННЯ І ЙОГО ЗАСТОСУВАННЯ ДЛЯ СТИСНЕННЯ ЗОБРАЖЕНЬ

Мета роботи: ознайомитись із застосуванням дискретного косинусного перетворення (ДКП) для стиснення зображень у форматі JPEG, використовуючи систему моделювання MATLAB з набором інструментів Image Processing Toolbox; проаналізувати ефективність стиснення, що досягається для різних степенів якості отриманого зображення.

Засоби виконання: пакет MATLAB 6.5/7 зі встановленим набором інструментів Image Processing Toolbox версії 3.*/4.*.

Теоретичні відомості

Дискретне косинусне перетворення

Дискретні косинусні перетворення представляють зображення у вигляді суми синусоїд з різною амплітудою і частотою. Функція dct2 в додатку Image Processing Toolbox реалізує двовимірні дискретні косинусні перетворення зображень. Одна з особливостей дискретного перетворення Фур'є полягає в тому, що деякі локальні ділянки зображення можна охарактеризувати невеликою кількістю коефіцієнтів дискретного перетворення Фур'є. Це властивість дуже часто використовується при розробці методів стискування зображень. Наприклад, дискретне косинусне перетворення є основою міжнародного стандарту, який використовується в алгоритмі стискування зображень з втратами JPEG. Назва формату “JPEG” складається з перших букв назви робочої групи, яка брала участь в розробці цього стандарту (Joint Photographic Experts Group).

Двомірне дискретне косинусне перетворення матриці A з розмірами *MXN* реалізується згідно з наступним вираженням

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N},$$

де $0 \le p \le M$ - 1 і $0 \le q \le N$ - 1;

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, & \text{якщо } p = 0; \\ \sqrt{\frac{2}{M}}, & \text{якщо } 1 \le p \le M-1. \end{cases} \qquad \alpha_q = \begin{cases} \frac{1}{\sqrt{N}}, & \text{якщо } q = 0; \\ \sqrt{\frac{2}{N}}, & \text{якщо } 1 \le q \le N-1. \end{cases}$$

Значення B_{pq} називають коефіцієнтами дискретного косинусного перетворення матриці A.

(Слід зазначити, що індекси матриці в MATLAB завжди починаються з 1, а не з 0. Тому елементи матриці, які представлені в MATLAB як A(1,1) і B(1,1), відповідатимуть елементам A00 і B00 з приведеної вище формули.) Зворотне дискретне косинусне перетворення реалізується згідно з виразами

$$A_{mn} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \alpha_p \alpha_q B_{pq} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N},$$

де $0 \le m \le M$ - 1 і $0 \le n \le N$ - 1;

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, & \text{якщо } p = 0; \\ \sqrt{\frac{2}{M}}, & \text{якщо } 1 \le p \le M-1. \end{cases} \qquad \alpha_q = \begin{cases} \frac{1}{\sqrt{N}}, & \text{якщо } q = 0; \\ \sqrt{\frac{2}{N}}, & \text{якщо } 1 \le q \le N-1. \end{cases}$$

Вираз зворотного дискретного косинусного перетворення може інтерпретуватися як представлення матриці A з розмірами *N*x*M* у вигляді суми *N*x*M* наступних функцій

$$\alpha_p \alpha_q \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \text{ де } 0 \le p \le M-1 \text{ и } 0 \le q \le N-1.$$

Ці функції називаються основними (базовими) функціями дискретного косинусного перетворення. Коефіцієнти дискретного косинусного перетворення B_{pq} можна розглядати як вагові при кожній базовій функції. Наприклад, для матриці з розміром 8x8 елементів існує 64 базових функції, що продемонстроване на зображенні.

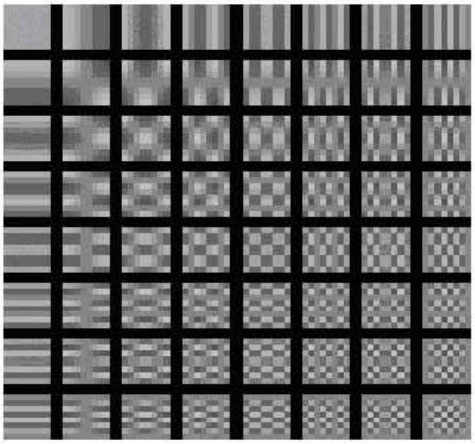

Рисунок 4.1 – 64 базові функції, які отримані для матриці з розмірами 8x8 елементів

Горизонтальні частоти збільшуються зліва направо, а вертикальні – зверху вниз

Матриця дискретних косинусних перетворень

Додаток Image Processing Toolbox пропонує дві різні дороги реалізації дискретних косинусних перетворень. Перший метод реалізований у функції dct2. Функція dct2 використовує швидке перетворення Фур'є для прискорення обчислень. Другий метод використовує матрицю дискретних косинусних перетворень, яка повертається функцією *dctmtx*. Матриця перетворень T формується згідно наступного виразу

$$T_{pq} = \begin{cases} \dfrac{1}{\sqrt{M}} & \text{при } p = 0,\ 0 \le q \le M-1; \\ \sqrt{\dfrac{2}{M}} \cos\dfrac{\pi(2q+1)p}{2M} & \text{при } 1 \le p \le M-1,\ 0 \le q \le M-1. \end{cases}$$

Для матриці A з розмірами *MxM*, *T*A* є матрицею з розмірами *MxM*, де кожен стовпець містить одномірне дискретне косинусне перетворення *A*. Двомірне дискретне косинусне перетворення A обчислюється як *B=T*A*T'*. Зворотне двомірне дискретне косинусне перетворення B обчислюється як *T'*B*T*.

Дискретні косинусні перетворення і стиснення зображень

В алгоритмі стискування зображень JPEG вихідне зображення розділяється на блоки з розмірами 8x8 або 16x16 елементів. Далі для кожного блоку обчислюється двовимірне дискретне косинусне перетворення. Коефіцієнти дискретних косинусних перетворень квантіруются, кодуються і передаються. Одержувач JPEG – даних декодує коефіцієнти дискретного косинусного перетворення,

обчислює зворотне двомірне дискретне косинусне перетворення у кожному блоці і далі поєднує їх разом в одне зображення.

Хід виконання роботи

1. Використовуючи зображення *відповідно до варіанту завдання,* представити його коефіцієнтами ДКП за допомогою команди *dct2:*

```
I = imread('cameraman.tif');
J = dct2(I);
imshow(I)
figure
imshow(log(abs(J)),[]), colormap(jet), colorbar
```

Зробити висновки про характер розподілу коефіцієнтів ДКП вздовж горизонтальної та вертикальної осей.

2. Застосувати дискретне косинусне перетворення (ДКП) для стиснення зображення *відповідно до варіанту завдання* у форматі JPEG. Як відомо, стисненню у форматі JPEG підлягає фрагмент (блок) зображення розміром 8×8. Тому для виконання поставленого завдання необхідно використати команду *dctmtx*, яка обчислює матрицю матрицу ДКП для блоку розміром $n \times n$, та команду *blkproc*, яка послідовно здійснює задані операції над фрагментами зображення розміром $n \times n$:

```
I = imread('cameraman.tif');
I = im2double(I);
T = dctmtx(8);
B = blkproc(I,[8 8],'P1*x*P2',T,T');
mask = [1   1   1   1   0   0   0   0
        1   1   1   0   0   0   0   0
        1   1   0   0   0   0   0   0
        1   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0];
B2 = blkproc(B,[8 8],'P1.*x',mask);
I2 = blkproc(B2,[8 8],'P1*x*P2',T',T);
imshow(I), figure, imshow(I2)
I3=I-I2;
figure, imshow(I3)
```

Пояснення. ДКП здійснюється послідовно шляхом перемноження матриці ДКП *(матриця T)* для блоку розміром 8×8 на блоки розміром 8×8 матриці зображення *(матриця I)*, а потім отриманий результат перемножується на транспоновану матрицю ДКП для блоку розміром 8×8 *(матриця T')*. Матриця *B2* – результат перемноження коефіцієнтів ДКП на маску, за

рахунок чого і здійснюється стиснення з втратами. Матриця *I2* – зображення, отримане в результаті операції стиснення. Матриця *I3* – помилка, яку вносить операція стиснення.

3. Повторити п.2 для маски з більшою та меншою кількістю ненульових елементів. Знайти оптимальний вид маски з точки зору степеня стиснення і якості зображення. Зробити висновок про вплив маски на степінь стиснення зображення і якість отриманого в результаті стиснення зображення.

Варіанти завдання

Варіант	Файл зображення
1	pout.tif
2	cameraman.tif
3	eight.tif
4	tire.tif
5	rice.png
6	circuit.tif
7	moon.tif
8	coins.png
9	liftingbody.png
10	cell.tif
11	building.tif
12	Moon Phobos.tif
13	pollen.tif
14	test_pattern.tif

Зміст звіту

Звіт про виконання лабораторної роботи має містити:

- титульну сторінку;
- назву та мету роботи;
- листинг програми;
- ДКП та зображення відповідно до п. 1;
- початкове зображення, стиснені зображення та графічні представлення помилки для різних варіантів представлення маски відповідно до п. 2-3;
- оптимальну маску та отримане зображення відповідно до п. 3;
- висновки відповідно до п. 1-3.

Орієнтовний перелік питань, що винесені на захист роботи:

1. Що таке ДКП та який його фізичний зміст?
2. Описати процедуру стиснення зображення у форматі JPEG.
3. Які два способи реалізації ДКП реалізовані у системі MATLAB та чим вони відрізняються?
4. Для чого призначена команда blkproc?
5. Навіщо необхідна матриця mask?
6. Як впливає кількість ненульових елементів матриці mask на стиснення зображення?

Лабораторна робота №5

МЕТОДИ РЕСТАВРАЦІЇ ЗОБРАЖЕНЬ

Мета роботи: ознайомитись з методами інверсної фільтрації та вінерівської фільтрації, що застосовуються для реставрації зображень, використовуючи систему моделювання MATLAB з набором інструментів Image Processing Toolbox; проаналізувати ефективність роботи розглянених методів фільтрації.

Засоби виконання: пакет MATLAB 6.5/7 зі встановленим набором інструментів Image Processing Toolbox версії 3.*/4.*.

Теоретичні відомості

Інверсна фільтрація

Найпростіший підхід до відновлення спотвореного зображення основується на побудові наближення вигляду:

$$\hat{F}(u,\upsilon) = \frac{G(u,\upsilon)}{H(u,\upsilon)},$$

після чого варто застосувати зворотне перетворення Фур'є до функції $\hat{F}(u,v)$ (нагадаємо, що $G(u, v)$ – це перетворення Фур'є спотвореного зображення). Цей підхід зручно називати інверсною фільтрацією. Такий метод можна виразити у вигляді наступного наближення:

$$\hat{F}(u,\upsilon) = F(u,\upsilon) + \frac{N(u,\upsilon)}{H(u,\upsilon)}.$$

Цей оманливо простий вираз говорить нам про те, що навіть якщо ми знаємо $H(u,v)$ точно, то ми не можемо відновити $F(u,v)$ (а отже, і оригінальне, неспотворене зображення $f(u,v)$), оскільки шумовий компонент є випадковою величиною, перетворення Фур'є якої $N(u, v)$ залишається невідомої. Крім того, на практиці звичайно є великі проблеми з функцією $H(u,v)$, що може мати нульові значення. Навіть якщо член $N(u, v)$ настільки малий, що ним можна знехтувати, ділення його на малі значення $H(u,v)$ може дати небажано велике збільшення до наближення $\hat{F}(u,v)$.

Типовий підхід до здійснення інверсної фільтрації базується на побудові частки $\hat{F}(u,v) = G(u,v)/H(u,v)$, у якої необхідно обмежити частотний діапазон

«малою» околицею початку відліку, а потім здійснити зворотне перетворення. Ідея полягає у тому, що нулі функції *H(u,v)* з меншим ступенем імовірності будуть розташовуватися «близько» від початку частотних координат, тому що амплітуда перетворення в цій області дорівнює найбільшому значенню цієї величини. Є множина варіацій на цю тему, при яких по-різному розбираються значення *(u,v)* функції *H* у нулі або біля нуля. Такий підхід іноді називається псевдоінверсною фільтрацією. Але, загалом, підходи, що базуються на інверсній фільтрації такого типу, не дуже точні.

Вінерівська фільтрація

Вінерівська фільтрація є одним із найстаріших і добре відомих підходів у лінійному відновленні зображень. Вінерівський фільтр шукає наближення $\hat{f}$, що мінімізує середньоквадратичне відхилення

$$e^2 = E(f - \hat{f})^2,$$

де E – оператор математичного очікування, а f – неспотворене зображення. Рішення цієї екстремальної задачі в частотній області виражається формулою

$$\hat{F}(u,\upsilon) = \left[\frac{1}{H(u,\upsilon)} \times \frac{|H(u,\upsilon)|^2}{H(u,\upsilon)|H(u,\upsilon)|^2 + S_\eta(u,\upsilon)/S_f(u,\upsilon)}\right] G(u,\upsilon),$$

де $H(u,v)$ – функція, що спотворює; $|H(u,v)|^2 = H^*(u,v)H(u,v)$; $H^*(u,v)$ – комплексно-спряжена функція $H(u,v)$; $S_\eta(u,v) = |N(u,v)|^2$ – енергетичний спектр шуму; $S_f(u,v) = |F(u,v)|^2$ – спектр неспотвореного зображення; частка $S_\eta(u, v)/S_f(u, v)$ називається енергетичним співвідношенням шум/сигнал. Видно, що якщо спектр шуму дорівнює нулю для всіх значень u й v, то це співвідношення також дорівнює нулю, і вінерівський фільтр зводиться до інверсного фільтра.

Визначимо дві корисні величини, які називаються середня енергія шуму й середня енергія зображення, відповідно,

$$\eta_A = \frac{1}{MN}\sum_u\sum_\upsilon S_\eta(u,\upsilon) \qquad f_A = \frac{1}{MN}\sum_u\sum_\upsilon S_f(u,\upsilon)$$

i

Тут M та N позначають вертикальний і горизонтальний розміри масивів зображення й шуму. Ці величини є скалярами, а їхня частка

$$R = \frac{\eta_A}{F_A}.$$

Сліпа деконволюція

Одна з найскладніших проблем, яка виникає при відновленні зображень, полягає в отриманні відповідних наближень функції спотворення зображення або оптичної функції (PSF, Point Spread Function) для використання в алгоритмах відновлення. Методи відновлення зображень, в яких не використовується інформація, що характеризує функцію PSF, називаються алгоритмами сліпої деконволюції.

Метод сліпої деконволюції, до якого було звернено увагу досліджувачів останні двадцять років, заснований на наближенні по максимуму правдоподібності (MLE, Maximum-Likelihood Estimation) - стратегії оптимізації при побудові наближень величин, перекручених випадковим шумом. Коротко можна сказати, що в інтерпретації MLE зображення вважається випадково вибраних з деякою певною ймовірністю з сімейства інших можливих випадкових величин. Функція правдоподібності виражається через функції *g(x, y)*, *f(x, y)* і *h (x, y)*, і завдання полягає в знаходженні максимуму функції правдоподібності. При сліпій деконволюції завдання оптимізації вирішується ітеративно при виконанні відповідних обмежень і за умови збіжності всієї процедури. Максимізуюча пара функцій *f (x, y)* і *h (x, y)* вважається відновленим зображенням і відповідною функцією PSF.

Хід виконання роботи

1. Застосувати до зображення *відповідно до варіанту завдання* размиття типу «гаусів шум»:

```
I = imread('xxxxxxx.xxx');
figure; imshow(I); title('Original Image');
H = fspecial('gaussian',7,10);
Blurred = imfilter(I,H,'symmetric','conv');
figure; imshow(Blurred); title('Blurred Image');
```

2. Відреставрувати зображення методом **сліпої деконволюції**, припускаючи, що інформація про спотворюючий оператор *H* не відома:

   а) спотворюючий оператор *H* на 4 пікселі менше по горизонталі та по вертикалі:

```
UNDERPSF = ones(size(H)-4);
[J1 H1] = deconvblind(Blurred,UNDERPSF);
figure;
imshow(J1);title('Deblurring with Undersized PSF');
```

б) спотворюючий оператор *H* на 4 пікселі більше по горизонталі и вертикалі:

```
OVERPSF = padarray(UNDERPSF,[4 4],'replicate','both');
[J2 H2] = deconvblind(Blurred,OVERPSF);
figure;imshow(J2);
title('Deblurring with Oversized PSF');
```

Зробити висновок про реставрацію зображення методом сліпої деконволюції, якщо інформація про спотворюючий оператор обмежена.

в) спотворюючий оператор *H* має початковий размер:

```
INITPSF = ones(size(H));
[J1 H1] = deconvblind(Blurred,INITPSF,N);
figure;
imshow(J1);title('Deblurring with Initial PSF');
```

Де N – кількість ітерацій застосування методу. Порівняти результати реставрації для кількості ітерацій N=5, 10, 20.

Візуально оцінити якість отриманого зображення по пунктам а), б) и в), зробити висновок про обмеження точності реставрації.

3. Застосувати до зображення *відповідно до варіанту* завдання розмиття типу «рух»:

```
I = imread('xxxxxxx.xxx');
figure;imshow(I);title('Original Image');
H = fspecial('motion',30,10);
Blurred = imfilter(I,H,'circular','conv');
figure; imshow(Blurred);
title('Blurred');
```

5. Відреставрувати зображення методом **інверсної фільтрації** за допомогою команди `deconvwnr`:

```
wnr1 = deconvwnr(Blurred,H);
figure;imshow(wnr1);
title('Restored');
```

Зробити висновок про якість реставрації зображення при впливі на нього розмиття.

6. Додати до розмитого зображення шум та повторити застосування команды `deconvwnr`:

```
noise = 0.1*randn(size(I));
BlurredNoisy = imadd(Blurred,im2uint8(noise));
figure;
imshow(BlurredNoisy);title('Blurred & Noisy');
wnr2 = deconvwnr(BlurredNoisy,H);
figure;
imshow(wnr2);
title('Inverse Filtering of Noisy Data');
```

Зробити висновок про якість реставрації зображення при впливі на нього спотворюючого оператору та шума.

7. Ввести контроль відношення шум-сигнал, при цьому застосування команди `deconvwnr` еквівалентно використанню **вінерівської фільтрації**:

```
NSR = sum(noise(:).^2) /sum(im2double(I(:)).^2);
wnr3 = deconvwnr(BlurredNoisy,H,2*NSR);
figure;
imshow(wnr3);
title('Restored with NSR');
```

Як змінилась якість відреставрованого зображення?

Варіанти завдання

Варіант	Файл зображення
1	pout.tif
2	cameraman.tif
3	eight.tif
4	tire.tif
5	rice.png
6	circuit.tif
7	moon.tif
8	coins.png
9	liftingbody.png
10	cell.tif
11	building.tif
12	Moon Phobos.tif
13	pollen.tif
14	test_pattern.tif

Зміст звіту

Звіт про виконання лабораторної роботи має містити:

- титульну сторінку;
- назву та мету роботи;
- листинг програми;
- зображення відповідно до п. 2-7;
- висновки відповідно до с п. 2-7.

Орієнтовний перелік питань, що винесені на захист роботи:

1. Що таке функція спотворення (функція розсіяння точки) та спотворюючий оператор?
2. Наведіть модель ввідновлення зображення.
3. В чому сенс інверсної фільтрації і які обмеження накладаються на результат відновлення?
4. Що таке вінерівська фільтрація і які її особливості?
5. В чому полягає метод сліпої деконволюції і коли він використовується?
6. Який з перерахованих вище трьох методів є ітеративним? Що це означає?

Лабораторна робота №6

ПЕРЕТВОРЕННЯ ХАФА. ВИДІЛЕННЯ КОНТУРІВ ТА ПІДКРЕСЛЕННЯ ЛІНІЙ НА ЗОБРАЖЕННІ

Мета роботи: ознайомитись з методами виділення контурів, що здійснюються детекторами Превіта та Канні, а також дослідити підкреслювання ліній у спотворених зображеннях, при цьому використати систему моделювання MATLAB з набором інструментів Image Processing Toolbox; проаналізувати ефективність роботи розглянених методів фільтрації.

Засоби виконання: пакет MATLAB 6.5/7 зі встановленим набором інструментів Image Processing Toolbox версії 3.*/4.*.

Теоретичні відомості

Детектор Превітта

Детектор Превітта використовує маску для числового приближення перших похідних G_x та G_y (табл. 6.1).

Таблиця 6.1 – Деякі маски детекторів і реалізовані ними формули наближення для перших похідних

-1	-1	-1
0	0	0
1	1	1
$G_x=(z_7+ z_8+ z_9) - (z_1+ z_2+ z_3)$		

-1	0	1
-1	0	1
1	0	1
$G_y=(z_3+ z_6+ z_9) - (z_1+ z_4+ z_7)$		

Його загальна форма виклику має вигляд

```
[g, t]=edge(f, 'prewitt', T, dir)
```

де `f` – це вхідне зображення, `T` – заданий поріг, а змінна `dir` позначає переважно напрямок для визначення межі, що має наступні значення: `'horizontal'`, `'vertical'`, чи `'both'` (за замовчуванням). `g` – логічний масив, в якому 1 знаходиться там, де знайдено межу, а 0 – там, де її нема. Вихідний параметр `t` не є обов'язковим. Якщо він заданий, то в нього записується значення порогу, використаного функцією `edge`. Якщо вхідний параметр `T` задано, то `t=T`. В ін-

шому випадку, якщо `T` не задано (або він пустий, тобто `[]`), то `t` автоматично рівний вибраному порогу, що і використовувався при визначенні межі. Одна з причин виключення вихідного параметра `t` полягає у заданні початкового значення для порогу.

Детектор Канні

Детектор Канні є найпотужнішим детектором, що закладений у функцію `edge`. Цей метод можна коротко описати наступним чином.

1. Зображення згладжується гаусовим фільтром із заданим стандартним відхиленням σ для скорочення шуму.
2. В кожній точці обчислюється градієнт $g(x,y)=\left[G_x^2+G_y^2\right]^{1/2}$ і напрям межі $\alpha(x,y)=arctg\left[G_y/G_x\right]$. Для знаходження G_x та G_y можна використовувати будь-який метод. Точки перепаду визначаються як точки локального максимуму градієнт.
3. Точки перепаду, визначені у пункті 2, спричинюють ріст гребенів на зображенні модуля градієнта. Потім алгоритм відслідковує верх цих гребенів і присвоює нульове значення точкам, які не лежать на гребні. В результаті на виході будується тонка лінія, а весь цей процес називається *не максимальним зменшенням*. Потім пікселі гребня підлягають пороговому обробленню з використанням двох порогів $T1$ та $T2$, причому $T1< T2$. Пікселі гребеня, величина яких більша $T2$, називаються «сильними», а пікселі, значення яких потрапляють в інтервал [$T1$, $T2$], називаються слабкими.
4. Алгоритм виконує з'єднання, додаючи до сильних пік селів слабкі, які 8-зв'язані з сильними.

   Синтаксис детектора Канні для визначення перепадів має вигляд

```
[g, t]=edge(f, 'canny', T, sigma)
```

де `T` – це вектор [$T1$, $T2$], що складається з порогів, описаних у п.3, а `sigma` позначає стандартне відхилення згладжу вального фільтру. Якщо параметр `t` є у вихідних аргументах, то він є вектором з двох компонент, куди записано два пороги, використані алгоритмом. Решта параметрів мають звичайний зміст, пояснений для іншого методу, включно з автоматичним визначенням порогів, якщо вектор `T` не заданий. Значення `sigma` по замовчуванню дорівнює 1.

Перетворення Хафа

В ідеальному випадку розглянуті вище методи мають визначати тільки пікселі, що належать краям та перепадам яскравості. Проте на практиці виділені пікселі рідко відносяться тільки до цієї категорії, це пов'язано з ярдом причин: вплив шуму, розриви країв через нерівномірне освітлення і інші фактори, що вносять хибні перепади яскравості у зображенні. Тому за процедурою визначення країв зазвичай йде процедура компонування виділених пікселів країв у справжні, змістовні лінії та крайові сегменти. Один з підходів до виконання подібних дій базується на перетворенні Хафа.

Маючи деяку множину точок зображення (звичайного двійкового), припустимо, що нам необхідно знайти підмножини цих точок, які лежать на прямих лініях. Один можливий підхід полягає у побудові довільних прямих, що проходять через задану пару точок, а потім у виявленні точок, що знаходяться близько до окремих прямих ліній. Проблема реалізації такої процедури пов'язана з необхідністю розгляду $n(n-1)/2 \approx n^2$ прямих, а потім у виконанні порядку $n\big(n(n-1)/2\big)/2 \approx n^3$ операцій порівняння всіх точок з кожною з цих прямих. Обчислювальна складність такого рішення дозволяє застосовувати його тільки в самих простих прикладних задачах.

Використовуючи перетворення Хафа, можливий інший підхід. Розглянемо будь-яку точку (x_i, y_i) і всі прямі, що проходять через неї. Всі такі прямі задовольняють рівняння $y_i = ax_i + b$ для будь яких значень *a* та *b*. Проте якщо переписати це рівняння у вигляді $b = -ax_i + y_i$ і розглянути відповідну площину *ab* (її називають областю параметрів), то воно задасть єдину пряму для кожної фіксованої координати пари (x_i, y_i). Більш того, іншій точці (x_j, y_j) відповідає своя пряма у області параметрів, і ці дві прямі перетинаються у деякій точці (a', b'), де a' - це кутовий коефіцієнт, а b' - точка перетину з віссю *y* прямої, що проходить через точки (x_j, y_j) та (x_i, y_i) на площині *xy*. Насправді, у кожної точки прямої є пряма у області параметрів, причому всі прямі перетинаються в точці (a', b'). Ці поняття ілюструє рис. 6.1.

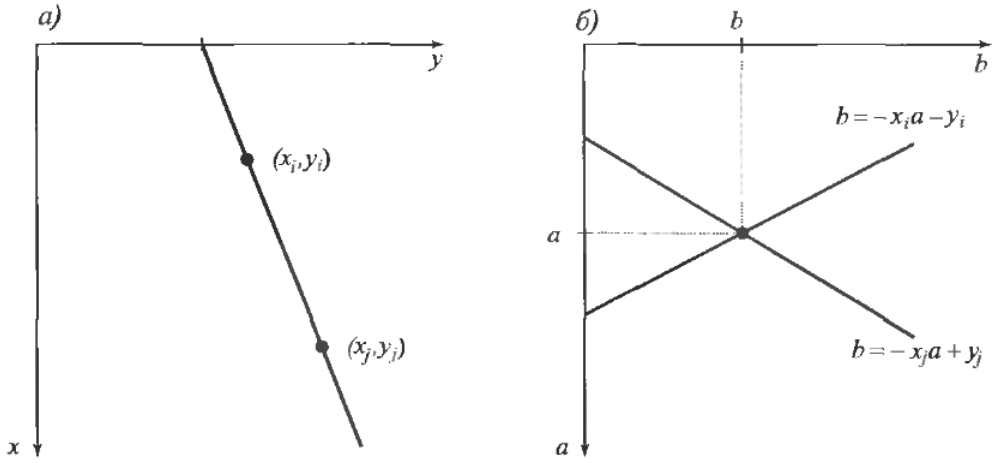

Рисунок 6.1 – а) площина *xy*, б) область параметрів *ab*

В принципі, можна побудувати на графіку всі параметричні прямі, що відповідають усім заданим точкам (x_i, y_i) зображення, а потім всі лінії зображення можна ідентифікувати за допомогою точок перетину параметричних прямих. При цьому виникає обчислювальна складність, пов'язана з тим, що число *a* (кутовий коефіцієнт) прямує до нескінченності, коли пряма наближена до вертикалі. Один із способів позбутись цієї проблеми полягає у представленні рівняння прямої за допомогою вектора нормалі:

$$x\cos\theta + y\sin\theta = \rho$$

На рис.6.2, а наведено геометрична інтерпретація параметрів θ та ρ. Горизонтальна пряма має $\theta = 0^{\circ}$, а параметр ρ рівний (додатній) координаті перетину з віссю *x*. Аналогічно, вертикальна пряма має $\theta = 90^{\circ}$, а ρ рівний координаті перетину з додатною піввіссю *y* чи $\theta = -90^{\circ}$, а ρ - координаті перетину з від'ємною піввіссю *y*. Кожна синусоїдальна крива на рис. 6.2, б відповідає сімейству прямих ліній, що проходять через деяку точку (x_i, y_i). Точка перетину кривих (ρ', θ') відповідає прямій, що проходить через обидві точки (x_i, y_i) та (x_j, y_j).

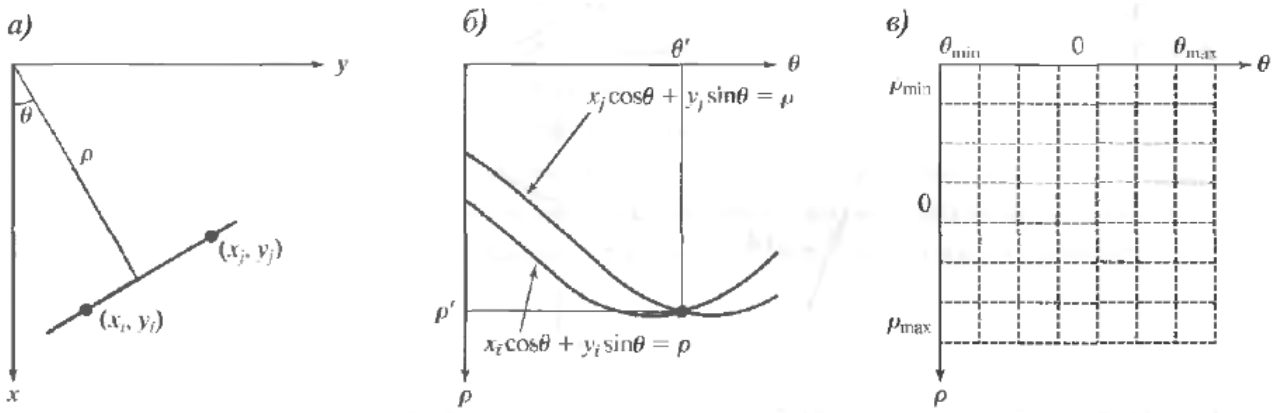

Рисунок 6.2 – а) Параметразація (ρ,θ) прямих на площині *xy*; б) синосуїди n площини $\rho\theta$. Точка перетину (ρ',θ') відповідає параметрам прямої, що з'єднує (x_i,y_i) та (x_j,y_j); в) розділення площини $\rho\theta$ на ділянки накопичення

Перевага перетворення Хафа з точки зору обчислювань полягає в можливості розбиття області параметрів $\rho\theta$ на так звані ділянки накопичення, як показано на рис.6.2, в, де $(\rho_{\min},\rho_{\max})$ та $(\theta_{\min},\theta_{\max})$ - передбачувані діапазони значень параметрів. За звичай ці значення лежать в інтервалах $-90^o \le \theta \le 90^o$ та $-D \le \rho \le D$, де D - це відстань між кутовими точками зображення. На ділянці з координатами (i, j) накопичуються значення $A(i, j)$ для квадрату в області параметрів, що відповідає точці (ρ_i,θ_j). На початку значення всіх ділянок накопичення дорівнюють нулю. Потім для кожної точки (x_k, y_k) вибраної множини на зображенні приймаємо параметр θ рівний кожному роздільному дискретному значенню θ_j у ділянках θ-осі і знаходимо відповідне для нього значення, знаходячи розв'язок рівняння $\rho = x_k \cos\theta + y_k \sin\theta$. Потім знайдені ведичини ρ округлюються до найближчого роздільного дискретного значення ρ_i із ділянок ρ-осі. Після цього значення відповідної ділянки накопичення збільшується на одиницю: $A(i, j) = A(i, j) + 1$. В кінці процедури ділянка $A(i, j)$ рівна числу Q, що означає, що Q точок площини *xy* лежать на прямій $x\cos\theta_j + y\sin\theta_j = \rho_i$. Точність потрапляння цих точок на пряму залежить від числа ділянок накопичення в області $\rho\theta$.

Приклад. Використання перетворення Хафа для виявлення ліній та зв'язування.

В даному прикладі ми використаємо функції `hough`, `houghpeaks` та `houghlines` для виявлення сегментів ліній у двійковому сегменті зображення f, що наведено на рис.6.3. Перед усім, ми обчислимо і покажемо перетворення

Хафа, використовуючи менший крок кутової дискретизації, ніж прийнятий за замовчуванням ($\Delta\theta = 0.5$, замість $\theta = 1$).

На рис.6.3, а наведено перетворення Хафа зі вказаним положенням знайдених максимумів. Нарешті, застосовуємо функцію `houghlines` для знаходження і зв'язування сегментів ліній, а знайдені сегменти розміщуємо на початковому двійковому зображенні за допомогою функцій `imshow, hold on, plot.`

```
>> lines= houghlines (f, theta, rho, r, c)
>> figure, imshow(f),hold on
>> for k=1:length(lines)
     xy=[lines(k).point1; lines(k).point2];
     plot(xy(:,2), xy(:,1), 'LineWidth', 4, 'Color', [.6 .6 .6]);
     end
```

На рис. 6.3, б наведено результуюче зображення з визначеними сегментами, що позначені жирними сірими лініями.

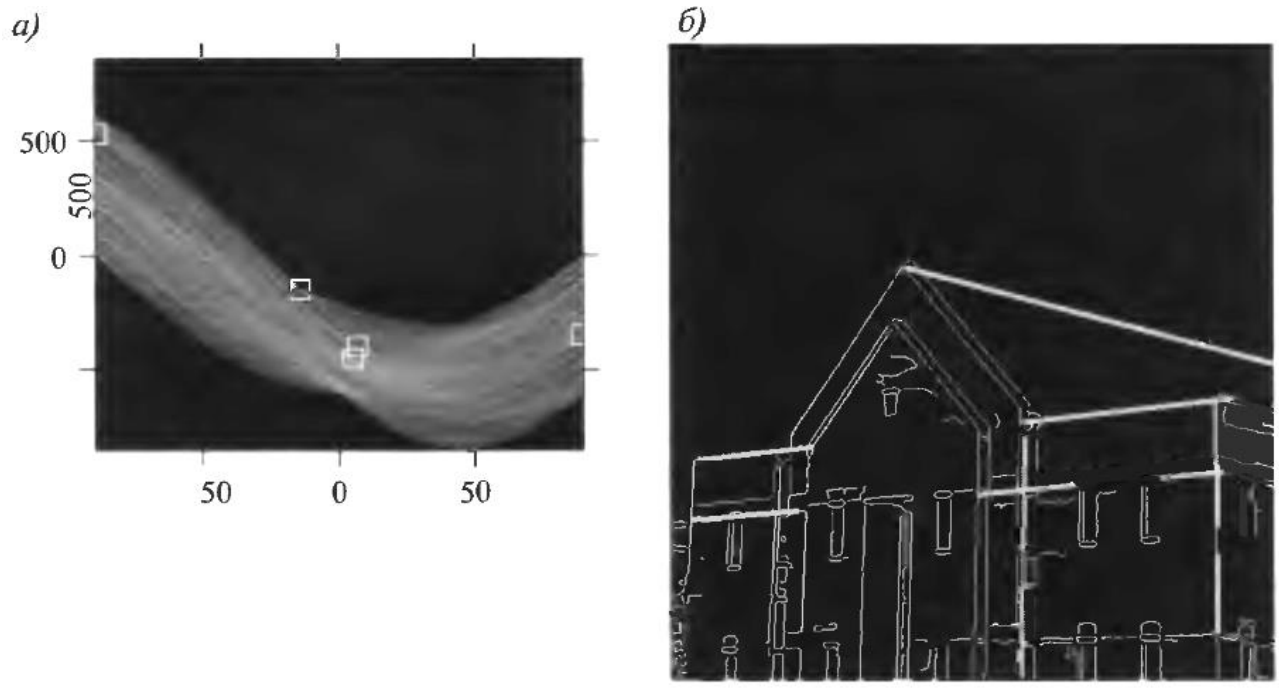

Рисунок 6.3 – а) перетворення Хафа з п'ятьма локальними максимумами;
б) сегменти ліній, що відповідають максимумам перетворення Хафа

Хід виконання роботи

1. Зчитати зображення з файлу (відповідно до варіанту завдання), використавши команду imread.

2. Відповідно до варіанту здійснити виділення контурів на зображенні за допомгою одного з методів: а) для парних варіантів – за допомогою детектора Превіта; б) для непарних – за допомгою детектора Канні.
3. Використовуючи наведені детектори, визначити оптимальне значення порогу детектування, при якому у бінарному зображенні будуть наявні тільки необхідні контури.
4. Використовуючи перетворення Хафа, здійснити підкреслювання ліній на зображенні (бінарному), яке містить виділені контури (краї). Для цього необхідно:
    a. Отримати перетворення Хафа від бінарного зображення;
    b. Визначити 5 локальних максимумів перетворення Хафа, які, скоріш за все, мають суттєвий зміст;
    c. Знайти і зв’язати сегменти ліній, у відповідності з отриманими локальними максимумами перетворення та розмістити ці лінії на початковому бінарному зображенні.

Варіанти завдання

Варіант	Файл зображення
1	pout.tif
2	cameraman.tif
3	eight.tif
4	tire.tif
5	rice.png
6	circuit.tif
7	moon.tif
8	coins.png
9	liftingbody.png
10	cell.tif
11	building.tif
12	Moon Phobos.tif
13	pollen.tif
14	test_pattern.tif

Зміст звіту

Звіт про виконання лабораторної роботи має містити:

- титульну сторінку;
- назву та мету роботи;

- листинг програми;
- результати виконання роботи;
- висновки.

Орієнтовний перелік питань, що винесені на захист роботи:

1. Як здійснюється виділення контурів?
2. Як здійснюється підкреслювання ліній?
3. В чому полягає метод виділення країв за допомогою детектору Превіта?
4. В чому полягає метод виділення країв за допомгою детектора Канні?
5. Для чого використовують перетворення Хафа?
6. Як здійснюється знаходження максимумів перетворення Хафа?
7. Як здійснюється виявлення ліній та їх зв’язування при відомих максимумах перетворення Хафа?

Лабораторна робота №7

ОБРОБКА ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ МОДЕЛЮВАННЯ В СЕРЕДОВИЩІ SIMULINK

Мета роботи: закріпити знання з обробки статичних зображень, отримані при виконанні попередніх робіт; отримати навики розробки моделей для обробки зображень в середовищі моделювання SIMULINK.

Засоби виконання: пакет SIMULINK 6.* зі встановленим набором блоків Video and Image Processing Blockset версії 1.*/2.*.

Теоретичні відомості

Загальні відомості

Програма Simulink є додатком до пакету MATLAB. При моделюванні з використанням Simulink реалізується принцип візуального програмування, відповідно до якого, користувач на екрані з бібліотеки стандартних блоків створює модель пристрою та здійснює розрахунки. При цьому, на відміну від класичних способів моделювання, користувачеві не потрібно досконально вивчати мову програмування та чисельні методи математики, а досить загальних знань потрібних при роботі на комп'ютері і, природно, знання тієї предметної області в якій він працює.

Simulink є досить самостійним інструментом MATLAB і при роботі з ним зовсім не потрібно знати сам MATLAB і решту його застосувань. З іншого боку доступ до функцій MATLAB і інших його інструментів залишається відкритим і їх можна використовувати в Simulink. Частина пакетів, що входять до його складу, має інструменти, які вбудовуються в Simulink (наприклад, LTI-Viewer програми Control System Toolbox - пакета для розробки систем управління). Є також додаткові бібліотеки блоків для різних областей застосування (наприклад, Power System Blockset - моделювання електротехнічних пристроїв, Digital Signal Processing Blockset - набір блоків для розробки цифрових пристроїв і т.д).

При роботі з Simulink користувач має можливість модернізувати бібліотечні блоки, створювати свої власні, а також складати нові бібліотеки блоків.

При моделюванні користувач може вибирати метод рішення диф-диференціальних рівнянь, а також спосіб зміни модельного часу (з фіксованим або змінним кроком). У ході моделювання є можливість стежити за процесами, що відбуваються в системі. Для цього використовуються спеціальні пристрої

спостереження, що входять до складу бібліотеки Simulink. Результати моделювання можуть бути представлені у вигляді графіків або таблиць.

Перевага Simulink полягає також у тому, що він дозволяє виконувати бібліотеки блоків за допомогою підпрограм написаних як на мові MATLAB, так і на мовах C + +, Fortran і Ada.

Запуск Simulink

Для запуску програми необхідно попередньо запустити пакет MATLAB.

Після відкриття основного вікна програми MATLAB потрібно запустити програму Simulink. Це можна зробити одним з трьох способів:

- натиснути кнопку (Simulink) на панелі інструментів командного вікна MATLAB;
- у командному рядку головного вікна MATLAB надрукувати Simulink і натиснути клавішу Enter на клавіатурі;
- виконати команду Open ... в меню File і відкрити файл моделі (mdl - файл).

Останній варіант зручно використовувати для запуску вже готової і налагодженої моделі, коли потрібно лише провести розрахунки і не потрібно додавати нові блоки в модель. Використання першого і другого способів призводить до відкриття вікна браузера розділів бібліотеки Simulink.

Створення моделі

Для створення моделі в середовищі SIMULINK необхідно послідовно виконати ряд дій:

1. Створити новий файл моделі за допомогою команди File / New / Model, або використовуючи кнопку на панелі інструментів (тут і далі, за допомогою символу "/", зазначено пункти меню програми, які потрібно послідовно вибрати для виконання зазначеної дії).

2. Розташувати блоки у вікні моделі. Для цього необхідно відкрити відповідний розділ бібліотеки (Наприклад, Sources - Джерела). Далі, вказавши курсором на потрібний блок і натиснувши на ліву клавішу "миші" - "перетягнути" блок у створене вікно. Клавішу миші потрібно тримати натиснутою.

Для видалення блоку необхідно вибрати блок (вказати курсором на його зображення і натиснути ліву клавішу "миші"), а потім натиснути клавішу Delete на клавіатурі.

Для зміни розмірів блоку потрібно вибрати блок, встановити курсор в один з кутів блоку і, натиснувши ліву клавішу "миші", змінити розмір блоку (курсор при цьому перетвориться на двосторонню стрілку).

3. Далі, якщо це потрібно, потрібно змінити параметри блоку, встановлені програмою "за замовчуванням". Для цього необхідно двічі клацнути лівою клавішею "миші", вказавши курсором на зображення блоку. Відкриється вікно редагування параметрів даного блоку. При завданні чисельних параметрів слід мати на увазі, що в якості десяткового роздільника повинна використовуватися точка, а не кома. Після внесення змін потрібно закрити вікно кнопкою OK.

4. Після установки на схемі всіх блоків із потрібних бібліотек потрібно виконати з'єднання елементів схеми. Для з'єднання блоків необхідно вказати курсором на "вихід" блоку, а потім, натиснути і, не відпускаючи ліву клавішу "миші", провести лінію до входу іншого блоку. Після чого відпустити клавішу. У разі правильного з'єднання зображення стрілки на вході блоку змінює колір. Для створення точки розгалуження в сполучній лінії потрібно підвести курсор до створюваного вузла і, натиснувши праву клавішу "миші", протягнути лінію. Для видалення лінії потрібно вибрати лінію (так само, як це виконується для блоку), а потім натиснути клавішу Delete на клавіатурі.

5. Після складання розрахункової схеми необхідно зберегти її у вигляді файлу на диску, вибравши пункт меню File / Save As ... у вікні схеми і вказавши папку та ім'я файлу. Слід мати на увазі, що ім'я файлу не повинне перевищувати 32 символів, має починатися з букви і не може містити символи кирилиці та спецсимволи. Ця ж вимога стосується й шляху файлу (до тої папки, в якій зберігається файл). При подальшому редагуванні схеми можна користуватися пунктом меню File / Save. При повторних запусках програми SIMULINK завантаження схеми здійснюється за допомогою меню File / Open ... у вікні браузера бібліотеки або з основного вікна MATLAB.

Бібліотека Video and Image Processing Blockset

Програмне зебезпечння MathWorks Video and Image Processing Blockset представляє собою бібліотеку Simulink блоків для моделювання, проектування та ррозробки систем спостереження, відслідковування та обробки відео сигналу у реальному часі.

Бібліотека Video and Image Processing Blockset містить типові блоки відеофільтрації та сучасні алгоритми обробки зображень, включно з двовимірною фільтрацією, статистичні функції, геометричні перетворення, морфологічні операції, детектори руху та засоби введення/виведення відео сигналів.

Основні можливості:

1) Проектування та моделювання систем відео обробки з цілочисленною, з плаваючою та фіксованою крапкою.
2) Спільне використання з Real-Time Workshop для генерації C-коду.
3) Імпорт/експорт відеопотоків.

4) Двовимірні фільтри, геометричні та інтегральні перетворення.
5) Типові перетворення кольорових ділянок, хроматична інтерполяція.
6) Класичні методи обробки зображень такі як, детектори меж, морфологія, статистика.

Доступ до бібліотеки Video and Image Processing Blockset можна отримати з командного рядка MATLAB ввівши viplib. Ця основна бібліотека включає 11 підбібліотек, що містять 52 блоки.

З роботою бібліотеки можна ознайомитись, запустивши демо версії певних процесів, ввівши відповідну команду у рядок MATLAB(див табл.1)

Таблиця 1 – Команди для запуску демо версій процесів

Процес	Process	Команда
Детектор руху	Motion detection	`vipmotion`
Запис нагляду	Surveillance recording	`vipsurveillance`
Зіставлення із зразком	Pattern matching	`vippattern`
Відеокомпресія	Video compression	`vipcodec`
Компресія зображення	Image compression	`vipimagecompression`
Показ гістограми	Histogram display	`viphistogram`
Визначення меж	Edge detection	`vipedge`
Визначення зміни зображення	Scene change detection	`vipscenechange`
Оцінка фокусу відео	Video focus assessment	`vipfocus`
Стабілізація відео	Video stabilization	`vipstabilize`
Зниження періодичних шумів	Periodic noise reduction	`vipstripes`
Вирівнювання гістограми	Histogram equalization	`viphisteq`
Корекція повороту	Rotation correction	`viphough`
Виділення ознак	Feature extraction	`vipspokes`
Підраховування об’єктів	Object counting	`vipstaples`
Виділення об’єкту та його переміщення	Object extraction and replacement	`vipobj`
Безперервна ротація зображення	Continuous image rotation	`viprotate`

Хід виконання роботи

1. Запустити MATLAB та викликати з панелі інструментів середовище візуального моделювання SIMULINK чи з командного рядка командою `simulink`
2. Створити нову модель, вибравши на панелі інструментів значок “Create new model”.
3. Ознайомитись з набором блоків для візуального моделювання у лівій частині вікна SIMULINK. Особливо звернути увагу на зміст блоків у розділі Video and Image Processing Blockset, проаналізувати призна-

чення блоків, що належать бібліотеці Analysis&Enhancement, Conversions, Filtering, Geometric Transformations, Sinks, Sources, Statistics, Transforms, Utilities. *Навести короткий опис функцій, що виконують блоки в цих бібліотеках.*

4. Відповідно до завдання на лабораторну роботу розробити модель обробки зображення.
   4.1 Накреслити на аркуші паперу орієнтовну блок-схему моделі, яка має містити:
   – *джерело* початкового зображення;
   – можливі *джерела шуму* чи завад (якщо це необхідно за умовою завдання);
   – відповідний до завдання *блок обробки* (наприклад, фільтр, блок ДКП, т.і.);
   – *блоки перегляду* початкового та обробленого зображень.
   4.2 Знайшовши відповідні блоки у бібліотеках Video and Image Processing Blockset (якщо необхідно, то і в інших бібліотеках SIMULINK), перетягнути їх до робочої області моделі та з'єднати між собою.
   4.3 Здійснити налаштування основних параметрів блоків моделі, натиснувши двічі по тілу відповідного блока; зберегти модель.
   4.4 Здійснити налаштування моделі перед її запуском (закладка Simulation – Configuration Parameters), запустити модель.
   4.5 Зробити висновки за результатами моделювання.

Завдання на лабораторну роботу

Варіант	Завдання	Файл зображення
1	Обробити зображення за допомогою фільтрів: НЧ (усереднюючого), ВЧ (Лапласа), руху.	pout.tif
2	Ввести в зображення імпульсний шум, обрабити зображення за допомогою медіанного фільтру, фільтра максимуму та мінімуму.	cameraman.tif
3	Побудувати гістограму зображення, виконати його еквалізацію, визначити максимальне та мінімальне значення яскравості зображення.	eight.tif
4	Побудувати гістограму зображення, виконати корекцію яскравості та контрастності	tire.tif

	зображення, визначити максимальне та мінімальне значення яскравості зображення.	
5	Ввести в зображення гаусів шум, обробити зображення за допомгою усереднюючого та медіанного фільтрів.	rice.png
6	Взяти ДКП від зображення, відсіяти частину коефіцієнтів (половину), виконати зворотнє перетворення та вивести результуюче зображення.	circuit.tif
7	Виділити краї на зображенні з використанням детекторів Превіта та Собеля.	moon.tif
8	Обрабити зображення за допомогою фільтра руху з трьома різними параметрами.	coins.png
9	Виділити краї на зображенні з використанням детекторів Канні та LoG.	liftingbody.png
10	Побудувати гістограму зображення, здійснити корекцію яскравості зображення (три різних параметри гамма), визначити максимальне, мінімальне та середнє значення яскравості зображення.	cell.tif
11	Виділити краї на зображенні з використанням детектора Канні та за допмогою перетворення Хафа виділити суттєві лінії на зображенні.	building.tif
12	Взяти ДКП від зображення, відсіяти частину коефіцієнтів (третину), виконати зворотнє перетворення та вивести результуюче зображення.	Moon Phobos.tif
13	Ввести до зображення імпульсний шум, обрабити зображення за допомогою усереднюючого та медіанного фільтрів	pollen.tif
14	Виділити краї на зображенні з використанням детектора Собеля та за допомгою перетворення Хафа виділити суттєві лінії на зображенні.	test_pattern.tif

Зміст звіту

Звіт про виконання лабораторної роботи повинен містити:

- титульний аркуш;
- назву та мету роботи;
- блок-схеми моделей відповідно до завдань 1-2;
- короткий опис функцій, що виконують блоки в моделях по завданням 1-2;
- віхідне, оброблене та спотворене (тільки для завдання 2.2) зображення;
- висновки по завданням 1-2.

Лабораторна робота №8

РОЗПІЗНАВАННЯ РЕЄСТРАЦІЙНИХ НОМЕРІВ З АВТОМОБІЛІВ ЗА ДОПОМОГОЮ MATLAB

Мета роботи: ознайомитись з методами обробки зображень, що дозволять розпізнати основні 3 цифри номера з картинки, що містить тільки номерний знак, використовуючи систему моделювання MATLAB з набором інструментів Image Processing Toolbox; проаналізувати ефективність роботи розглянених методів фільтрації.

Засоби виконання: пакет MATLAB 6.5/7 зі встановленим набором інструментів Image Processing Toolbox версії 3.*/4.*.

Теоретичні відомості

Передобробка

Зображення дуже зашумлене. Погана контрастність, різнорідна освітленість, шум заважають безпосередньо застосовувати механізми виявлення об'єктів.

Перший крок. Функція imadjust здійснює лінійне розтягнення гістограми зображення (рис. 8.1). При відсутності параметрів вона розтягує її так, щоб мінімальна та максимальна яскравості пікселів були 0 і 255 відповідно.

Рисунок 8.1 – Зображення після розтягнення гістограми

Другий крок. Від дрібного засмічення картинки шумом камери можна позбутися медіанної фільтрацією. Це призведе до зникнення точок на білому тлі біля цифр (рис. 8.2).

Рисунок 8.2 – Зображення після обробки з допомогою медіанної фільтрації

Третій крок. Для пошуку ліній необхідно використати фільтр «Laplacian of Gaussian» або log (рисю 8.3). Якщо швидкість є критичною - його роботу можна замінити більш швидким фільтром DoG (difference of Gaussian). Він є швидшим, але менш якісним.

Рисунок 8.3 – Зображення після обробки з фільтром Гауса

Четвертий крок. Тепер необхідно здійснити бінаризацію зображення, відокремлюючи фон від об'єкта (рис. 8.4).

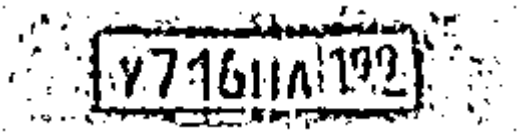

Рисунок 8.4 – Зображення після бінаризації

Після цього необхідно замкнути області рамки номера (якщо вона зіллється з фоном, то пошук цифр має відбуватися в цілій картинці, а не тільки в номері) (рис. 8.5). Зробити це можна з допомогою операції морфологічного розкриття, з маскою у вигляді диска радіуса 2.

Рисунок 8.5 – Зображення після операції морфологічного розкриття

Пошук областей, що представляють інтерес

Необхідно перебрати усі сегменти, в яких відбуватиметься пошук цифри.

Пошук самого номера. Функція regionprops обчислює задані характеристики областей, на які поділена картинка. У даному випадку знадобляться поля «Площа», «Зображення», «Орієнтація», «Рамка», «Повна площа (включаючи внутрішню частину)». Критерій, за яким шукається номер, підбирається емпіричним шляхом, виходячи з геометричних розмірів реєстраційного номера.

До параметрів, що представляють інтерес, відносяться: ratio - відношення ширини до висоти номери, Convex Area - повна площа об'єкта, включаючи дірки. Якщо номер підходить, в спеціальний масив записується картинка – вміст цього номера з початкового зображення.

Пошук цифр

Пошук цифр у цілому схожий на пошук номеру. Спочатку здійснюється передобробка зображення. Для цього вирізається область номера з початкового зображення для здійснення повторної обробки тільки цієї області. Це допоможе уникнути впливу стороннього шуму, освітлення та всього, що не належить даній області.

Дії , які необхідно здійснити - adjust, LoG, бінаризація (тепер уже по >150), морфологічне закриття для видалення дрібного сміття, морфологічне розкриття для усунення незамкненості. Область збільшується, щоб спростити евристику за площею.

Розрахунок параметрів і розпізнавання

Для кожного блоба (*BLOB*, Binary Large Object) розраховується його подовженість і площа. Якщо область досить велика (Area> 2200), і має певну подовженість (ratio <0.7 & & ratio> 0.1), то обчислюється коефіцієнт кореляції цієї області з кожної «ідеальною» цифрою. Ці цифри згенеровані заздалегідь. Далі вибирається найкращий збіг, і, якщо він більше деякого порога - приймається за хороший результат. Обрахування закінчується при наборі трьох цифер підряд. Регіони перебираються зліва направо, так що потрібні цифри вийдуть у потрібному порядку (рис. 8.6).

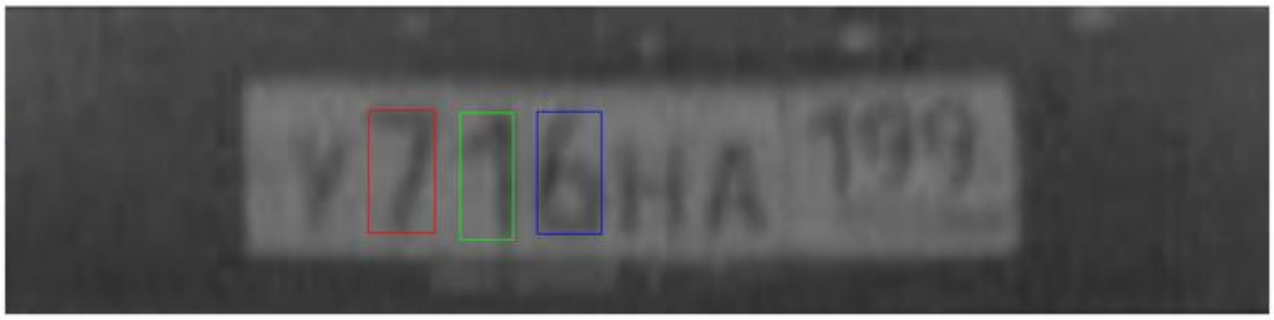

Рисунок 8.6 – Зображення після операції морфологічного розкриття і розпізнавання

Хід виконання роботи

1. Здійснити лінійне розтягнення гістограми зображення:
```
srcImod = imadjust(srcImod);
```
2. Виконати медіанну фільтрацію зображення:
```
srcImod = medfilt2(srcImod,[3 3]);
```
3. Використанb фільтр log:
```
filt = fspecial('log',[7 7], 0.3);
srcImod = imfilter(srcImod,filt);
```
4. Здійснити бінаризацію зображення:
```
srcImod = srcImod(:,:,:) < 150;
```

5. Замкнути область рамки номера:

```
srcImod = imopen(srcImod,strel('disk',2));
srcImod = (bwlabel(srcImod,8));
```

6. Перебрати всі сегменти, в яких відбуватиметься пошук цифр номера:

```
Idata = regionprops(srcImod,'Area','Image','Orientation'
,'BoundingBox','ConvexArea');
k = 1;
for i=1:length(Idata)
ratio = Idata(i).BoundingBox(3)/Idata(i).BoundingBox(4);
if Idata(i).ConvexArea > 500 && ratio < 7 && ratio > 2.5
bound = floor(Idata(i).BoundingBox);
numbers(k) = Idata(i);
numbers(k).Image = imcrop(srcI,bound);
k = k +1;
end
end
```

Далі в циклі за всіма знайденими сегментами шукати що-небудь, схоже на цифри:

```
count = k - 1;
k = 1;
for i=1:count
```

7. Здійснити пошук цифр:

```
srcImod = imrotate(numbers(i).Image, -
numbers(i).Orientation, 'bicubic','crop');
srcImod = imadjust(srcImod);
filt = fspecial('log',[7 7], 0.32);
srcIlog = imfilter(srcImod,filt);
srcImod = srcIlog;
srcImod = imresize(srcImod,[400 NaN]);
srcImod = srcImod (:,:,:) > 150;
sl = strel('disk',7);
srcImod = imopen(srcImod,sl);
sl = strel('disk',8);
srcImod = imclose(srcImod,sl);
```

8. Здійснити остаточне розпізнавання:

```
for i=1:length(probdigits)
ratio = probdigits(i).BoundingBox(3) / probdig-
its(i).BoundingBox(4);
if probdigits(i).Area > 2200 && ratio < 0.7 && ratio > 0.1;
ImageToTest = imresize(probdigits(i).Image,[64 64]);
c = zeros();
for j=1:10,  c(j)= corr2 (ImageToTest,idealnum(j).Image);  end
c = abs©;
maxid = 1;
for j=2:10, if c(j) > c(maxid), maxid = j; end,  end
if c(maxid)>0.55;
digits(k) = (maxid-1);
```

```
digareas(k).bound = PlateBounds + probdigits(i).BoundingBox;
k = k + 1;
end
if k==4, break, end
end
end
```

Варіанти завдання

Варіант	Файл зображення
1	007.bmp
2	136.bmp
3	322.bmp
4	393.bmp
5	482.bmp
6	523.bmp
7	586.bmp
8	667.bmp
9	702.bmp
10	739.bmp
11	750.bmp
12	836.bmp
13	853.bmp
14	957.bmp

Зміст звіту

Звіт про виконання лабораторної роботи має містити:

- титульну сторінку;
- назву та мету роботи;
- листинг програми;
- зображення відповідно до п. 1-8;
- висновки відповідно до с п. 1-8.

Орієнтовний перелік питань, що винесені на захист роботи:

1. Як здійснюється лінійне розтягнення гістограми зображення?
2. Як здійснюється бінаризація зображення?
3. Для чого використовується медіанна фільтрація?
4. З допомогою чого і з якою метою здійснюється пошук ліній?

5. Для чого здійснюється замикання області рамки номера?
6. В чому полягає пошук областей, в яких відбуватиметься пошук цифр?
7. Як здійснюється остаточний пошук цифр і які параметри важливі для нього?

СПИСОК ФУНКЦІЙ IMAGE PROCESSING TOOLBOX

Формати представлення даних

xyz2uint16 - перетворення даних кольору з формату XYZ в uint16
xyz2double - перетворення даних в XYZ-значення в форматі double
double - представлення елементів масиву у форматі double
uint8 - представлення елементів масиву у форматі uint8
im2double - представлення зображення масивом у форматі double
im2uint8 - представлення зображення масивом у форматі uint8
im2uint16 - представлення зображення масивом у форматі uint16
im2mis - представлення зображень в Java MemoryImageSource
im2java2d - представлення зображення в буферизоване зображення Java
im2java - перетворення даних в Java-зображення
lab2uint8 - перетворення даних з формату L*a*b* в uint8
lab2uint16 - перетворення даних L*a*b* в формат uint16
lab2double - перетворення даних L*a*b* в формат double

Визначення типу зображення

isbw - перевірити, чи є зображення бінарним
isgray - перевірити, чи є зображення напівтоновим
isind - перевірити, чи є зображення палітровим
isrgb - перевірити, чи є зображення повноколірним

Робота з графічними форматами файлів

imfinfo - читання з файлу інформації про зображення
imread - читання зображення із файлу
imwrite - запис зображення в файл

Установка і читання глобальних змінних IPT

iptsetpref - установка глобальних змінних IPT
iptgetpref - читання глобальних змінних IPT
getline - вибір ломаної лінії за допомогою мишки
getpts - вибір точок за допомогою мишки
getrect - вибір прямокутника за допомогою мишки

Виведення зображень на екран і захват їх з екрану

imshow - виведення зображень на екран
truesize - установка розмірів вікна для відображення зображень
subimage - виведення декількох зображень в одному вікні
colorbar - вивід на екран палітри
imcontour - побудова для зображення ліній рівня
immovie - створення відео послідовності
montage - вивід на екран всіх кадрів багатокадрового зображення
warp - накладання зображення на поверхню
zoom - масштабування зображення у вікні зображення
getimage - отримання зображення з графічного об'єкту
dicominfo - читання метаданих з DICOM-файлу
dicomread - читання DICOM зображень
dicomwrite - запис зображень в DICOM-файл
dicomuid - генерація ідентифікатора для DICOM-файлів
imview - відображення зображень в Image Viewer

Перетворення типів зображень

im2bw - бінаризація відсіканням по порогу яскравості
mat2gray - перетворення матриці чисел в півтонове зображення
rgb2gray - перетворення полноцветного зображення в півтонове
ind2gray - перетворення палітрового зображення в півтонове
gray2ind - перетворення півтонового зображення в палітрове
grayslice - перетворення півтонового зображення в палітрове відсіканням по декількох порогах
ind2rgb - перетворення палітрового зображення в повноцінне
dither - дифузійне псевдозмішування кольорів
rgb2ind - перетворення повноколірного зображення в палітрове
imapprox - зменшення кількості кольорів палітрового зображення
cmunique - пошук палітри мінімального розміру
cmpermute - зміна порядку кольорів в палітрі
label2rgb - перетворення матриці міток в RGB-зображення

Конвертація колірних систем

rgb2hsv - конвертація з RGB у HSV
hsv2rgb - конвертація з HSV у RGB
rgb2ntsc - конвертація з RGB у YIQ
ntscc2rgb - конвертація з YIQ у RGB
rgb2ycbcr - конвертація з RGB у YCbCr
ycbcr2rgb - конвертація з YCbCr у RGB
rgbplot - зображення компонентів RGB палітри (MATLAB Toolbox)
graythresh - обчислення глобального порогу зображення з використанням методу Отса
iccread - прочитування опису ICC

Геометричні перетворення зображень

imcrop - кадрування зображень
imresize - зміна розмірів зображення
imrotate - поворот зображення
checkerboard - створення шахово-образних зображень
findbounds - визначення кордонів при просторових перетвореннях
imtransform - застосування просторових перетворень зображень
makeresampler - створення структури, що повторюється
maketform - створення структури просторових перетворень (TFORM)
tformarray - застосування просторових перетворень для багатовимірних масивів
tformfwd - застосування прямих просторових перетворень
para2fan - обчислення віяльно-променевих проекцій на підставі паралельно-променевих даних томографій
tforminv - застосування зворотних просторових перетворень
fan2para - обчислення паралельно-променевих проекцій даних томографії з пучком, що розходиться
f**anbeam** - обчислення віяльно-променевих перетворень
f**liptform** - перестановка вихідних і результуючих даних в структурі TFORM
i**fanbeam** - обчислення інверсного віяльно-променевого перетворення
applycform - застосування перетворення простору кольорів
makecform - створення структури перетворення значень кольорів
whitepoint - опис повноколірного білого пікселя в просторі кольорів

Аналіз зображень

imhist - побудова гістограми
improfile - побудова профілю
impixel - визначення значення пікселя
pixval - управління режимом відображення значень пікселів
mean2 - обчислення середнього значення елементів матриці
std2 - обчислення середньоквадратичного відхилення елементів матриці
corr2 - обчислення коефіцієнтів кореляції між двома матрицями
xcorr2 - обчислення двовимірної взаємної кореляційної функції
imabsdiff - визначення відмітних ознак двох зображень
imadd - підсумовування двох зображень або підсумовування зображення і константи
imcomplement - доповнення зображень
imdivide - розділення двох зображень або ділення зображення на константу
imlincomb - обчислення лінійної комбінації двох зображень
immultiply - множення двох зображень або множення зображення на константу
imsubtract - віднімання двох зображень або віднімання константи із зображення
regionprops - визначення властивостей області зображення
cpstruct2pairs - конвертація cpstruct у найбільш важливі контрольні точки
cp2tform - виведення просторових перетворень між парою контрольних точок
cpcorr - визначення погоджених контрольних точок з використанням кросс-кореляції
cpselect - інструмент вибору контрольних точок
normxcorr2 - нормалізація двовимірної кросс-кореляції
deconvblind - поліпшення зображень з використанням зворотної згортки
deconvlucy - поліпшення зображень з використанням методу Люсі-Річардсона
deconvreg - поліпшення зображень з використанням регулярізаційної фільтрації
deconvwnr - поліпшення зображень з використанням фільтру Вінера
ippl - перевірка наявності бібліотеки функцій (Intel Performance Primitives Library (IPPL))

Поліпшення зображень

histeq - вирівнювання гістограми
imadjust - контрастування з гамма-корекцією
brighten - управління яскравістю палітри
imnoise - додавання шуму
roifill - заповнення областей інтересу
stretchlim - пошук кордонів підвищення контрасту зображення
edgetaper - віділення країв з використанням функції протяжності точок
otf2psf - перетворення оптичної функції у функцію протяжності точок
psf2otf - перетворення функції протяжності точок в оптичну функцію
adapthisteq - виконання контрастно обмеженої адаптивної еквалізації гістограми
decorrstretch - застосування декорреляціонного розтягування багатоканальних зображень

Фільтрація зображень

conv2 - згортка зображень
convn - згортка N–мірних сигналів
convmtx2 - обчислення матриці згортки
filter2 - двовимірна лінійна фільтрація
freqz2 - двовимірна АЧХ
fspecial - завдання маски зумовленого фільтру
fsamp2 - формування маски лінійного фільтру по бажаній АЧХ
ftrans2 - формування маски лінійного фільтру методом перетворення частот
fwind1 - формування маски лінійного фільтру по бажаній АЧХ з використанням одновимірного вікна
fwind2 - формування маски лінійного фільтру по бажаній АЧХ з використанням двовимірного вікна
blkproc - обробка блоків зображення
bestblk - визначення розміру блоку
nlfilter - узагальнений нелінійний фільтр
colfilt - оптимізована операція фільтрації
im2col - перетворення фрагментів зображення в стовпці
col2im - перетворення допоміжного зображення
ordfilt2 - рангова фільтрація
medfilt2 - медіанна фільтрація

wiener2 - адаптивна вінерівська фільтрація
roifilt2 - фільтрація областей інтересу
imfilter - фільтрація двовимірних і багатовимірних зображень
freqspace - визначення відгуку в двовимірній частотній області (MATLAB Toolbox)

Сегментація зображень

poly2mask - перетворення деякої області в маску
qtdecomp - сегментація методом розділення
qtgetblk - отримання блоків з квадро–дерева результатів сегментації
qtsetblk - заміна блок–результатів сегментації
edge - виділення границь
roipoly - завдання області інтересу за допомогою полігону
roicolor - бінаризація по заданим кольорах
watershed - алгоритм маркерного вододілу

Морфологічні операції над бінарним зображенням

uintlut - обчислення нових значень масиву на основі табличних перетворень
applylut - перетворення бінарного зображення за допомогою таблиці перекодування
bwboundaries - вистежування локальних границь на бінарному зображенні
bwmorph - морфологічні операції над бінарним зображенням
bwareaopen - відкриття бінарних площ (малих об'єктів)
bwdist - визначення періоду перетворення бінарних об'єктів
bwfill - заповнення областей фону
bwhitmiss - бінарні hit-miss операції
bwlabeln - установка мітки зв'язаних елементів в багатовимірних бінарних зображеннях
bwpack - упаковка бінарних зображень
bwperim - виділення границь бінарних об'єктів
bwselect - виділення об'єктів
bwtraceboundary - відслідковування контурів бінарних зображень
bwulterode - гранична ерозія
bwunpack - розпаковування бінарних зображень
conndef - відсутність зв'язності
dilate - нарощування бінарного об'єкту

erode - ерозія бінарного об'єкту
imbothat - виконання низькочастотної фільтрації
imclearborder - пригнічення світлової структури, яка пов'язана з краями зображення
imclose - закрити зображення
imdilate - розширення зображення
imerode - ерозія зображення
imextendedmax - максимальна тривалість перетворень
imextendedmin - мінімальна тривалість перетворень
imfill - заповнення областей зображення
imhmax - H-максимальні перетворення
imhmin - H-мінімальні перетворення
imimposemin - установка мінімуму
imopen - відкрити зображення
imreconstruct - морфологічне відновлення зображень
imregionalmax - максимум області
imregionalmin - мінімум області
imtophat - виконання високочастотної фільтрації
makelut - формування таблиці перекодування

Пошук об'єктів і обчислення їх ознак

bwlabel - пошук об'єктів
bwarea - обчислення площі об'єктів
bweuler - обчислення числа Ейлера
imfeature - обчислення ознак об'єктів

Перетворення Фур'є

fft2 - двовимірне БПФ
fftn - n–вимірне БПФ
ifft2 - зворотне двовимірне БПФ
ifftn - n– вимірне зворотне БПФ
fftshift - перегрупіровка вихідного масиву перетворення Фур'є

Дискретне косинусне перетворення

dct2 - двовимірне ДКП

idct2 - зворотне двовимірне ДКП
dctmtx - обчислення матриці коефіцієнтів ДКП

Перетворення Радону

radon - пряме перетворення Радону
iradon - зворотне перетворення Радону
phantom - створення модельного зображення голови

Створення і обробка структурних елементів

getheight - створення вертикальних структурних елементів
getneighbors - визначення місця розташування суміжних структурних елементів
getnhood - створення суміжних структурних елементів
getsequence - створення послідовності розкладених структурних елементів
isflat - повернення однакових структурних елементів
reflect - представлення структурних елементів через їх центр
strel - створення морфологічних структурних елементів
translate - перетворення структурних елементів

Операції з масивами

padarray - порожній масив

Демонстрация

dctdemo - демонстрація стиснення зображень на основі двовимірних дискретних косинусних перетворень
edgedemo - демонстрація виділення границь об'єктів зображення
firdemo - демонстрація двовимірній фільтрації зображень і проектування фільтрів
imadjdemo - демонстрація корегування яскравостей і еквалізації гістограми зображень
landsatdemo - демонстрація складноколірних зображень
nrfiltdemo - демонстрація фільтрації шумової складової
qtdemo - демонстрація розкладання квадродерев
roidemo - демонстрація обробки областей інтересу

ПЕРЕЛІК ЛІТЕРАТУРИ

1. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB. – Москва: Техносфера, 2006. – 616с.
2. Р. Гонсалес, Р. Вудс. Цифровая обработка изображений. – Москва: Техносфера, 2005. – 1072 с.
3. Грузман И.С., Киричук В.С., Косых В.П., Перетягин Г.И., Спектор А.А. Цифровая обработка изображений в информационных системах: Учебное пособие. – Новосибисрк: Изд-во НГТУ, 2000. – 168 с.
4. Гашников М.В., Глумов Н.И., Ильясова Н.Ю., Мясников В.В., Попов СБ., Сергеев В.В., Сойфер В.А.и др. Методы компьютерной обработки изображений / Под ред. В.А. Сойфера. – 2-е изд., испр. – М.: ФИЗМАТЛИТ, 2003. – 784 с.
5. Яне Б. Цифровая обработка изображений. – Москва: Техносфера, 2007. – 584с.

ЗМІСТ