

**ЛІТЕРАТУРА**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ  
УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ

**Кафедра біотехнічних систем**



# **МЕТОДИЧНІ ВКАЗІВКИ**

**до виконання лабораторних робіт  
з дисципліни**

## **ОБРОБКА БІОМЕДИЧНИХ ЗОБРАЖЕНЬ**

**для студентів напряму підготовки  
6.051402 – Біомедична інженерія  
спеціальності 7.05140201, 8.05140201 –  
Біомедична інженерія**

**НАВЧАЛЬНО-МЕТОДИЧНА**

**ТЕРНОПІЛЬ, 2016**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
імені ІВАНА ПУЛЮЯ

**Кафедра біотехнічних систем**

## **МЕТОДИЧНІ ВКАЗІВКИ**

**до виконання лабораторних робіт  
з дисципліни**

## **ОБРОБКА БІОМЕДИЧНИХ ЗОБРАЖЕНЬ**

**для студентів напряму підготовки  
6.051402 – Біомедична інженерія  
спеціальності 7.05140201, 8.05140201 –  
Біомедична інженерія**

*Розглянуто на засіданні  
кафедри біотехнічних систем*

*протокол № \_\_\_\_ від \_\_\_\_\_  
2016 р.*

*Затверджено на засіданні  
методичної комісії факультету  
прикладних інформаційних  
технологій та електроінженерії*

*протокол № \_\_\_\_ від \_\_\_\_\_  
2016 р.*

**ТЕРНОПІЛЬ, 2016**

Методичні вказівки до виконання лабораторних робіт з дисципліни “Обробка біомедичних зображень” для студентів напряму підготовки 6.051402 – Біомедична інженерія спеціальності 7.05140201, 8.05140201 – Біомедична інженерія. / Паляниця.: Ю.Б. – Тернопіль: ТНТУ, 2016 – с.

Призначені для полегшення засвоєння дисципліни “ Обробка біомедичних зображень” і контролю знань студентів. Складається з урахуванням модульної системи навчання, рекомендацій до самостійної роботи і індивідуальних завдань, тем практичних та лабораторних занять, тестів, екзаменаційних питань, типової форми та вимог для комплексної перевірки знань з дисципліни.

## ЗМІСТ

ВСТУП.....	
ПРАВИЛА ТЕХНІКИ БЕЗПЕКИ ПРИ ВИКОНАННІ ЛАБОРАТОРНИХ РОБІТ.....	
Лабораторна робота № 1. Основні функції обробки зображень в системі Matlab 8	
Лабораторна робота № 2. Морфологічний аналіз зображень в системі Matlab..	10
Лабораторна робота № 3. Фільтрація зображень в системі Matlab.....	17
Лабораторна робота № 4. Сегментація зображень в системі Matlab методом k-середніх .....	27
Лабораторна робота № 5. Застосування фільтра Габора.....	32
Лабораторна робота № 6. Тривимірні візуалізація зображень в системі Matlab.....	36
Лабораторна робота № 7. Автоматизоване визначення локалізації пухлини легенів в системі Matlab .....	41
Лабораторна робота № 8. Знаходження контурів елементів зображення в системі Matlab .....	46

## ВСТУП

Лабораторні роботи виконуються для закріплення теоретичних знань, отриманих студентами під час лекційних занять та самопідготовки. Метою виконання лабораторних робіт є вироблення у студентів навичок та вмінь самостійної роботи при створенні алгоритмів опрацювання зображень.

Метою посібника є допомога студентам при підготовці та виконанні лабораторних робіт за 8 темами, які ввійшли до цього видання. На початку збірника подані основні вимоги техніки безпеки при виконанні лабораторних робіт. Ознайомлення з ними студентів проводиться на вступному лабораторному занятті з обов'язковим підписом кожного студента у відповідному журналі. Далі наведено опис кожної лабораторної роботи.

Навчальний посібник забезпечує можливість студентам самостійно підготуватись до виконання лабораторної роботи.

## ПРАВИЛА ТЕХНІКИ БЕЗПЕКИ ПРИ ВИКОНАННІ ЛАБОРАТОРНИХ РОБІТ

Лабораторні роботи з курсу проводяться в лабораторіях кафедри радіотехніки Тернопільського національного технічного університету імені Івана Пулюя. Дотримання правил техніки безпеки є обов'язковою умовою виконання лабораторних робіт. Для забезпечення цієї вимоги кожен студент на вступному занятті повинен ознайомитися з вимогами правил техніки безпеки, про що вказує відмітка у відповідному журналі та отримати допуск до виконання лабораторних робіт у викладача.

Забороняється заходити в лабораторію без дозволу викладача або когось із допоміжного персоналу кафедри. Не дозволяється знаходитись в лабораторії у верхньому одязі. Студенти, які тимчасово не задіяні на роботі із комп'ютерами чи макетами, повинні знаходитися в місці, вказаному викладачем. Всі зайві предмети, які не стосуються до виконання даної лабораторної роботи необхідно прибрати з робочого місця.

Забороняється без дозволу викладача вмикати лабораторну установку, користуватись приладами, які не використовуються для виконання даної лабораторної роботи, а також переносити прилади з місця на місце. При виявленні несправності негайно повідомити про це викладача, або когось із допоміжного персоналу кафедри, хто знаходиться в лабораторії. Загалом необхідним є обов'язкове виконання всіх правил техніки безпеки та пожежної безпеки, які передбачені державними стандартами та інструкцією по університету.

Порушення правил техніки безпеки може призвести до нещасних випадків і веде за собою адміністративну та кримінальну відповідальність. Студент, який порушив правила техніки безпеки в лабораторії не допускається до занять. Допуском до подальшого виконання лабораторних робіт є відповідальність за причинену шкоду та повторне проходження інструктажу по техніці безпеки у зав. лабораторіями кафедри радіокомп'ютерних систем з відповідною відміткою про це в журналі.

**Строго дотримуйтесь цих правил - це запорука вашої безпеки  
та безпеки ваших одногрупників.**

# Основні функції обробки зображень в системі Matlab

## ЛАБОРАТОРНА РОБОТА №1

Мета роботи: ознайомлення з представленням зображень різних типів і класів, набуття навичок виконання операцій перетворення зображень, запису і зчитування зображень, набуття навичок основних градаційних перетворень зображень, ознайомлення з видами та параметрами просторових шумів у зображеннях

### Завдання до лабораторної роботи №1

1. Ознайомитися з роботою та параметрами функцій відкривання зображень `imread`. Завантажити в робочу область Матлаб зображення, що зберігаються у файлах різних форматів. Визначити розмір матриці зображення, отримати інформацію про тип зображення та наявність палітр за допомогою функції `imfinfo`.

2. Сформувати матрицю розмірністю 128x128 випадкових чисел за допомогою функції `rand` в діапазонах від 0 до 50, від 0 до 256, від 0 до 512, від 0 до 1, від -128 до 128. Вивести матрицю на екран як зображення, скориставшись функцією `imshow` для випадків:

- 1) використання параметрів яскравості за замовчуванням;
- 2) використання трьох рівнів яскравості;
- 3) використання рівнів яскравості у вигляді порожнього масиву;
- 4) використання рівнів яскравості від 20 до 30;
- 5) використання рівнів яскравості від 0.4 до 0.6. Вивести в кожному вікні шкалу кольорів за допомогою функції `colorbar`.

3. Для повнокольорового зображення вивести окремо матриці яскравостей кожного кольору. Поміняти місцями червоний та синій кольори, вивести результат. Замінити один з кольорів випадковими числами, вивести результат.

\*Запропонувати концепцію техніки оцінки важливості компонентів кольору для загального сприйняття всього зображення.

4. За допомогою функції `im2bw` виконати перетворення повнокольорового зображення на бінарне для трьох різних типів порогу. Побудувати залежність кількості чорних та білих пікселів від порогу. Зробити висновки щодо суб'єктивної якості отриманих бінарних зображень.

\*Запропонувати концепцію техніки визначення оптимального порогу для перетворення.

5. Ознайомитися з функціями перетворення класів та типів зображень `im2uint8`, `mat2gray`, `im2double`. Навести приклади застосування цих функцій на власних зображеннях. Зробити висновки про особливості перетворень та про можливі втрати інформації.

6. Ознайомитися з функцією запису файлів зображень різних типів `imwrite`.

Записати одне і те саме зображення у файл формату jpeg з параметрами якості  $q=0, 5, 10, 20, 50, 75, 100$ . Побудувати залежність ступеня стиснення зображення від параметру якості.

### **Зміст звіту**

1. Тема.
2. Мета.
3. Обґрунтування вибраного зображення та необхідності обробки його.
4. Текст програми.
5. Результати.
6. Інтерпретація результатів роботи програми.
7. Висновки.

### **Використана література**

1. Гонсалес, Джевирс. Цифровая обработка сигналов: практический подход, 2-е издание. М.: Вильямс, 2009. – 992 с.:ил.
2. Прэтт У. Цифровая обработка изображений: - М.: Мир, 1982. – 790 с.
3. Дьяконов В.П., MATLAB 6.5 SP1/7/7 SP1/ Работа с изображениями и видеопотоками. – М.: СОЛОН-Пресс, 2010. – 400 с.



# Морфологічний аналіз зображень в системі Matlab

## ЛАБОРАТОРНА РОБОТА №2

Мета роботи: ознайомлення з можливостями морфологічного аналізу зображень засобами системи Matlab та набуття навиків контрастування, позбавлення фону та сегментації зображень.

### Хід роботи

*В якості вихідного зображення для опрацювання використано зображення, що імітує гістологічний матеріал в апертурі мікроскопа і являє собою зернисту структуру на предметному склі, як, наприклад, сепаровану (відділену на центрифuzі) еритроцитарну масу, при цьому необхідно виділити контури об'єктів з метою визначення наявності конгломератів (еритроцити, що склеїлися між собою).*

Очистимо робочий простір MATLAB, закриємо всі відкриті вікна перегляду і закриємо всі відкриті засоби Image Tools.

```
clear, close all, imtool close all
```

Зчитуємо і візуалізуємо півтонове зображення rice.png.

```
I = imread('rice.png');  
imshow(I)
```

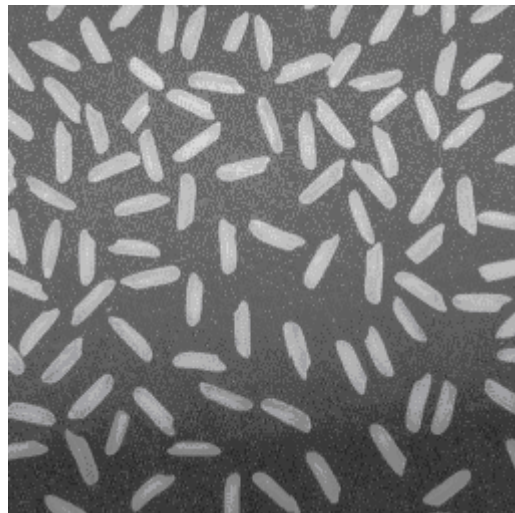


Рис. 1 - Півтонове зображення rice.png

Зображення характеризується нерівномірним засвіченням фону по полю зображення. Тому спочатку за допомогою операції морфологічного розкриття оцінимо інтенсивність фону. При виконанні операції морфологічного розкриття використовується структурний елемент.

У прикладі використовується функція `imopen`, яка виконує морфологічний розкриття. При цьому також використовується функція `strel` для створення структурного елементу в вигляді диска з радіусом 15.

```
background = imopen(I, strel('disk', 15));
```

Для перегляду і оцінки фону використовуються наступні команди:

```
figure, imshow(background)
```

Використовуємо команду `surf` для відображення поверхні фону. За допомогою команди `surf` створюється кольорова параметрична поверхня, яка дає можливість переглядати прямокутну область досліджуваного зображення. Функція `surf` працює з даними, які представлені в форматі `double`. Тому, перш ніж застосувати цю функцію, дані необхідно перевести в формат `double`.

```
figure, surf(double(background(1:8:end, 1:8:end))), zlim([0 255]);  
set(gca, 'ydir', 'reverse');
```

В даному прикладі відображається тільки кожен восьмий піксель по кожному напрямку. У наведеному прикладі існує також можливість установки масштабу та інших параметрів.

На основі цього можна проводити аналіз фону досліджуваного зображення.

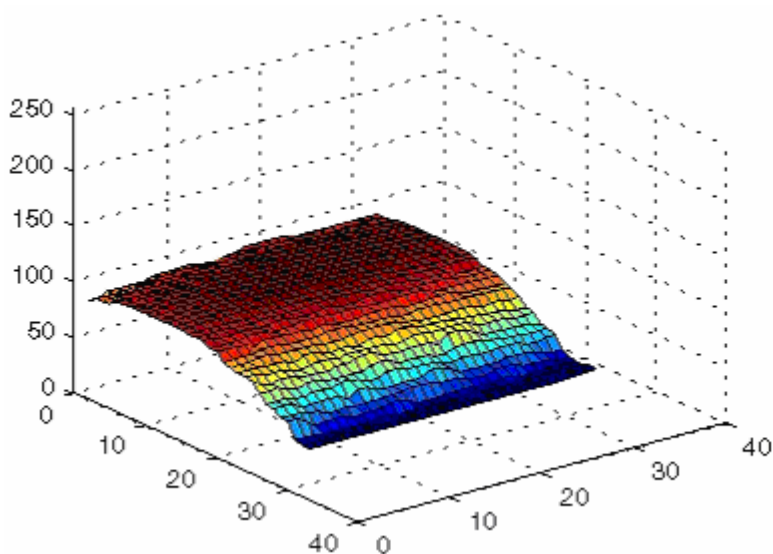


Рис. 2 - Фон зображення

Для створення зображення з більш рівномірним фоном, віднімемо зображення фону `background` з вихідного зображення `I`.

```
I2 = imsubtract(I, background);
```

Відобразимо отримане зображення з більш рівномірним фоном.

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

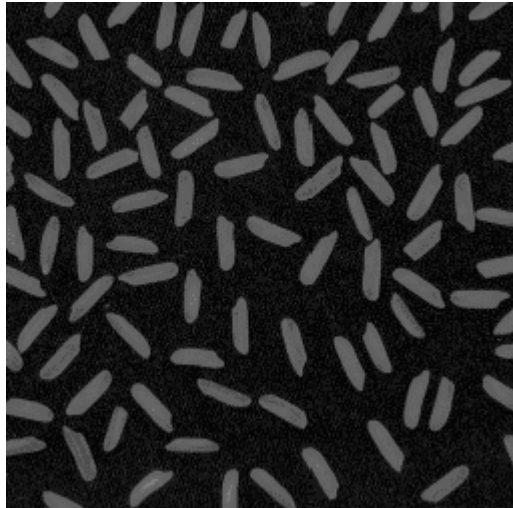


Рис. 3 – Зображення з рівномірним фоном

Після вирахування, отримане зображення матиме більш рівномірний фон і буде більш темним. Використовуємо функцію `imadjust` для підвищення контрасту зображення.

```
I3 = imadjust(I2);
```

Функція `imadjust` збільшує контраст зображення шляхом розтягування значень інтенсивностей динамічного діапазону. Відобразимо покращене зображення I3.

```
figure, imshow(I3);
```

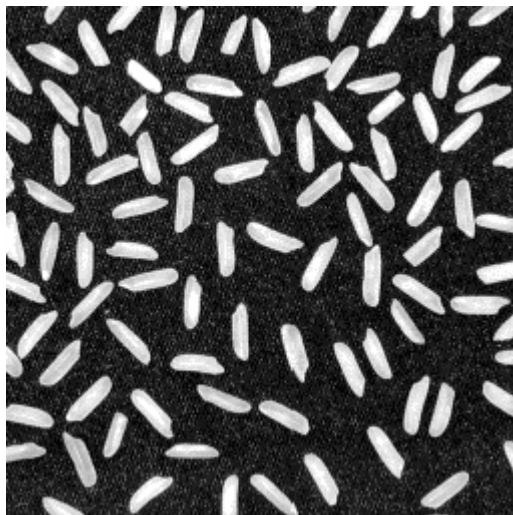


Рис. 4 – Зображення після корекції інтенсивності

Бінарне зображення можна створити використовуючи функцію `thresholding`. Однак функція `graythresh` автоматично визначає відповідний поріг, який використовується для перетворення напівтонового зображення в бінарне. Функція `im2bw` виконує це перетворення.

```
level = graythresh(I3);  
bw = im2bw(I3, level);
```

```
figure, imshow(bw)
```

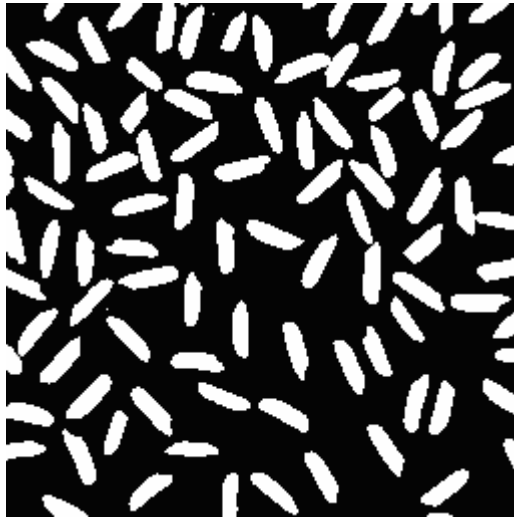


Рис. 5 – Бінарне зображення

Після перетворення зображення в бінарне, можна використовувати функцію `bwlabel` для визначення числа об'єктів (зерен) на зображенні. Функція `bwlabel` зазначає всі компоненти на бінарному зображенні `bw` і повертає їх число у вигляді значення `numObjects`.

```
[labeled,numObjects] = bwlabel(bw,4);
```

```
numObjects
```

```
ans =
```

```
101
```

Один з можливих шляхів відображення матриці міток полягає в використанні псевдоцветной індексного зображення. У псевдоцветной зображенні числа, які ідентифікують кожен об'єкт в матриці міток, відображаються різними кольорами, який пов'язаний з відповідною палітрою кольорів. Таке уявлення дозволяє розрізнити різні об'єкти.

Для перегляду матриці міток використовується функція `label2rgb`. Ця функція застосовується при створенні палітри, кольору фону і кольору кожного об'єкта зображення.

```
pseudo_color = label2rgb(labeled, @spring, 'c', 'shuffle');  
imshow(pseudo_color);
```

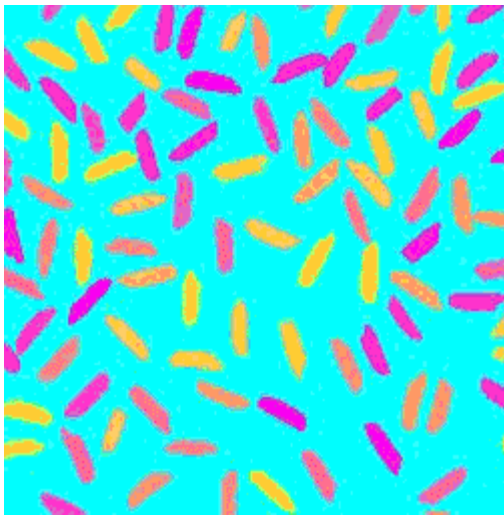


Рис. 6 – Представлення матриці міток у вигляді псевдоколірного зображення

Функція `regionprops` застосовується для вимірювання властивостей об'єктів в деякому околі зображення і представляє результат у вигляді масиву. Якщо застосувати це до зображення із зазначеними компонентами, то створюється структура для кожної компоненти.

У цьому прикладі використовується функція `regionprops` для створення структурованого масиву, що містить деякі основні властивості. Якщо встановити властивостями параметр `'basic'`, тоді функція `regionprops` повертає три вимірних параметри: площа, центр ваги (або центр мас) і обмежує прямокутник (локальна околиця). В даному випадку обмежувальний прямокутник є локальною околицею, яка буде показувати гранули.

```
graindata = regionprops(labeled,'basic')
graindata =

101x1 struct array with fields:
    Area
    Centroid
    BoundingBox
```

Для пошуку області з 51-м зазначеним компонентом, необхідно знайти поле `Area` field і його 51 елемент в структурі масиву `graindata`.

```
graindata(51).Area
ans =

140
```

Для пошуку найменш можливого прямокутника і центроїда (центру мас) деяких компонентів використовується наступний код:

```
graindata(51).BoundingBox, graindata(51).Centroid
ans =

107.5000    4.5000    13.0000    20.0000

ans =
```

Функції системи MATLAB можуть використовуватися для обчислення статистичних властивостей об'єктів. Спочатку використовується функція `max` для пошуку найбільшого зерна. (У нашому прикладі найбільшими є два зерна, які стикаються.)

```
max([graindata.Area])
ans =

404
```

Далі за допомогою функції `find` знайдемо ті компоненти, якими відзначені зерна з цією площею.

```
biggrain = find([graindata.Area]==404)
biggrain =

59
```

Середній розмір всіх зерен.

```
mean([graindata.Area])
ans =

175.0396
```

Побудуємо гистограму розподілу зерен рису за їх розміром (при цьому будемо використовувати 20 позицій при градації по осі площі). З гистограми видно, що основна частина зерен рису має площу від 150 до 250 пікселів.

```
hist([graindata.Area],20)
```

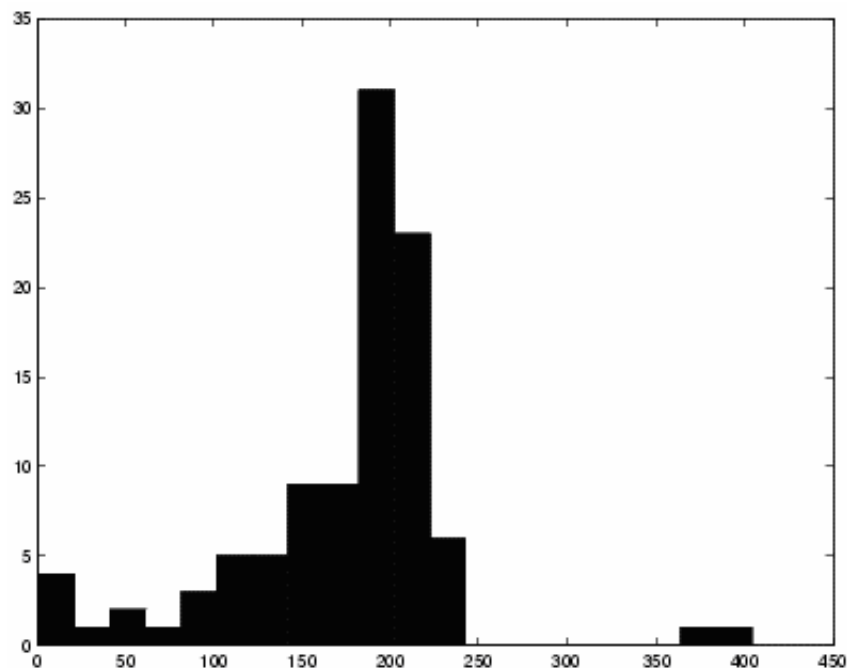


Рис. 7 – Гістограма зображення

## **Завдання до лабораторної роботи**

1. Повторити етапи, приведені в розділі «Хід роботи».
2. Оформити звіт.

### **Зміст звіту**

1. Тема.
2. Мета.
3. Обґрунтування вибраного зображення та необхідності обробки його.
4. Текст програми.
5. Результати.
6. Інтерпретація результатів роботи програми.
7. Висновки.

### **Використана література**

1. Гонсалес, Джевирс. Цифровая обработка сигналов: практический подход, 2-е издание. М.: Вильямс, 2009. – 992 с.:ил.
2. Прэтт У. Цифровая обработка изображений: - М.: Мир, 1982. – 790 с.
3. Дьяконов В.П., MATLAB 6.5 SP1/7/7 SP1/ Работа с изображениями и видеопотоками. – М.: СОЛОН-Пресс, 2010. – 400 с.
4. Обработка сигналов и изображений. Режим доступа: <http://matlab.exponenta.ru/imageprocess/book5/3.php>

# Фільтрація зображень в системі Matlab

## ЛАБОРАТОРНА РОБОТА №3

Мета роботи: вивчення функцій зашумлення і фільтрації зображення, набуття практичних навичок їх використання..

### КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Функція додавання шуму `imnoise`:

Синтаксис

`ID = imnoise(Is, type, params)`

Функція `imnoise` створює нове зображення `ID` шляхом додавання шуму до вихідного напівтонового зображення `Is`. Функція може додавати шум трьох типів, які задаються строковою константою `type`. Кількість і зміст параметрів `params` визначаються обраним типом шуму. Якщо параметри не вказані, то будуть використані значення за замовчуванням:

1) `ID = imnoise(Is, "gaussian", m,v)` додає до зображення `Is` гауссовий білий шум

з математичним очікуванням  $m$  і дисперсією  $v$ . За замовчуванням  $m = 0$ ,  $v =$

0.01;

2) `ID = imnoise(Is, "salt & pepper", d)` додає до зображення `Is` імпульсний

(точковий) шум. Параметр  $d$  визначає щільність шуму і дорівнює частці

перекручених пікселів. За замовчуванням  $d = 0.05$ ;

3) `ID = imnoise(Is, "speckle", v)` додає до зображення `Is` мультиплікативний шум

$ID = Is + n * Is$ , де  $n$  - рівномірно розподілений випадковий шум з математичним

очікуванням нуль і дисперсією  $v$ . За замовчуванням  $v = 0.04$ .

Завдання 1. Зашумити зображення гаусовим, імпульсним і мультиплікативним шумом.

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

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

```
>>S1=imnoise(I,'gaussian', 0,0.1);
```

```
>>S2=imnoise(I,'salt & pepper');
```

```
>>S3=imnoise(I,'speckle');
```

```
>>figure,subplot(1,3,1),subimage(S1);
```

```
>>Subplot(1,3,2),subimage(S2);
```

```
>>Subplot(1,3,3),subimage(S3)
```

Фільтрація зображення

Для виконання лінійної просторової фільтрації використовують метод



двоірної просторової згортки в локальній околиці оброблюваного елемента з лінійним оператором, який називається маскою або матрицею коефіцієнтів фільтра.

Алгоритм згортки полягає в тому, що маска сканує вихідне зображення.

На кожному кроці знаходиться сума добутків елементів маски і відповідних елементів вихідного зображення, і знайдене значення присвоюється одному елементу результуючого зображення. Досягнувши таким чином кінця рядка, маска зміщується на один рядок вниз, на початок рядка, і процес повторюється.

Є дві тісно пов'язані концепції, які необхідно розуміти при вчиненні лінійної просторової фільтрації. Перша - це кореляція, а друга - згортка.

Кореляція полягає в проходженні маски по зображенню. З точки зору механіки

процесу, згортка робиться так само, але маску треба повернути на  $180^\circ$  перед

проходженням по зображенню. Якщо зсувна маска є симетричною, то кореляція і згортка дають однакові результати.

Нижче наводяться функції, використовувані для фільтрації зображення.

Функція обчислення двовимірної згортки `conv2` 45

Синтаксис

$D = \text{conv2}(S, h, \text{shape})$

Функція виконує двовимірну просторову згортку зображення  $S$  з маскою  $h$ . Параметр `shape`, що визначає розмір результуючого зображення  $D$ , може приймати такі значення:

- 1) "full" - повнорозмірна згортка (за замовчуванням);
- 2) "same" - центральна частина розміру зображення  $S$ ;
- 3) "valid" - центральна частина розміру зображення  $S$  з вирахуванням

розміру маски.

Функція обчислення двовимірної лінійної фільтрації `filter2`

Синтаксис

$D = \text{filter2}(h, S, \text{shape})$

Результат обчислюється як кореляція масиву  $S$  двовимірним фільтром, коефіцієнти якого зведені в матрицю  $h$ . Як правило,  $S$  і  $D$  є напівтоновими зображеннями. Значення параметра `shape` такі ж, як у функції `conv2`. За замовчуванням `shape = "same"`. З точки зору виконання процесу, згортка робиться так само, але маску  $h$  треба повернути на  $180^\circ$  перед проходженням по зображенню  $S$ . Для цього можна використовувати функцію `rot90`.

Завдання 2. Виконати фільтрацію палітрових зображень, використовуючи маску

```

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

```

Функція обчислення медіанної фільтрації `medfilt2`

Синтаксис

`I D = medfilt2(I S ,[m,n], padopt)`

Функція виконує нелінійну фільтрацію, механізм виконання якої аналогічний лінійній фільтрації, але на кожному кроці сканування маски

розміру  $m \times n$  (по замовчуванню  $3 \times 3$ ) виконується наступна нелінійна операція:

пікселі зображення, що знаходяться під маскою, упорядковано і складають

упорядковану послідовність  $A$ . Піксель результуючого зображення  $I D$  ( $r,c$ ), де  $r$  і  $c$  - координати поточного становища центрального елемента маски,

присвоюється значення медіани послідовності  $A$ .

Параметр `padopt` визначає три можливі опції розширення меж

зображення: опція за замовчуванням `'zeros'` з нульовим розширенням, 46

`'symmetric'`, при якій зображення  $I S$  розширюється шляхом його дзеркального відображення через границі, і `'indexed'`, при якій  $I S$  розширюється значенням 1,

якщо  $I S$  має клас `double`, і значенням 0 в іншому випадку.

Завдання 3. Відфільтрувати зображення файлу `Athena.bmp`, зашумленого

імпульсним шумом.

```

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

```

Функція обчислення двовимірної АЧХ `freqz2`

Синтаксис

`freqz2(h,n1,n2)` - виводить на екран двовимірну АЧХ.

$[H, f1, f2] = \text{freqz2}(h, n1, n2)$  - формує матрицю  $H$  розміром  $n1 \times n2$ , яка є АЧХ на частотах, що містяться у векторах  $f1$  і  $f2$ , по двовимірному фільтру, коефіцієнти якого зведені в матрицю  $h$ .

Завдання 4. Вивід АЧХ високочастотного фільтра двома способами.

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

Функція задання маски зумовленої фільтром `fspecial`

Синтаксис

$h = \text{fspecial}(\text{type}, P1, P2)$

Функція повертає маску  $h$  зумовлену двовимірним лінійним фільтром, що задається рядком `type`. Залежно від типу фільтра, для нього можуть бути

визначені один або два додаткові параметри  $P1$ ,  $P2$ . Може бути заданий розмір

маски  $n$  (якщо  $n$  - вектор, то розмір маски  $n(1) \times n(2)$ , якщо  $n$  - скаляр, то розмір

маски -  $n \times n$ ),  $\sigma$  - середньоквадратичне відхилення розподілу Гаусса, яке використовується при формуванні маски  $h$ ; параметр  $a$ , керуючий

співвідношенням між центральним і граничними елементами маски, встановлюється в діапазоні  $[0, 1]$ , по замовчуванням дорівнює  $0.2$ .

Можливі варіанти функції `fspecial`:

1)  $h = \text{fspecial}(\text{'gaussian'}, n, \sigma)$  повертає маску  $h$  фільтра нижніх частот

Гаусса. За замовчуванням  $n$  дорівнює  $3 \times 3$ , а  $\sigma$  рівне  $0.5$ ;

2)  $h = \text{fspecial}(\text{'sobel'})$  повертає маску фільтра Собела для виділення горизонтальних границь, для виділення вертикальних границь досить транспонувати дану маску  $h$ ;

3)  $h = \text{fspecial}(\text{'prewitt'})$  повертає маску фільтра Превітта для виділення горизонтальних кордонів, для виділення вертикальних границь досить транспонувати дану маску  $h$ ;

4)  $h = \text{fspecial}(\text{'laplacian'}, a)$  повертає маску  $h$  ВЧ фільтр Лапласа. Розмір маски

$3 \times 3$ . За допомогою фільтра Лапласа можна виконувати поліпшення зображення, використовуючи формулу  $g(x, y) = f(x, y) + c \Delta^2 f(x, y)$ , де  $f(x, y)$  - це вихідне зображення;  $g(x, y)$  - покращене зображення, а параметр  $c$  дорівнює  $1$ , якщо центральний коефіцієнт маски позитивний, і  $c = -1$  в іншому випадку;  $\Delta^2 f(x, y) = d^2 f(x, y)/dx^2 + d^2 f(x, y)/dy^2$  - відфільтроване зображення фільтром Лапласа;

5)  $h = \text{fspecial}(\text{'log'}, n, \sigma)$  повертає маску  $h$  фільтра, аналогічного

послідовному застосуванню фільтрів Гауса і Лапласа, так званого Лапласіана -

гауссіана,  $n$  і  $\sigma$  за замовчуванням встановлюється рівним  $5 \times 5$  і 0.5

відповідно;

6)  $h = \text{fspecial('average', n)}$  повертає маску  $h$  усередненого НЧ фільтра.

За

замовчуванням розмір маски  $n$  встановлюється рівним  $3 \times 3$ ;

7)  $h = \text{fspecial('unsharp', a)}$  повертає маску  $h$  фільтра, що підвищує різкість

зображення. Розмір маски:  $3 \times 3$ .

Завдання 5. Поліпшити зображення, використовуючи фільтр високих частот

Лапласа.

1. Створити маску  $h$  фільтра високих частот Лапласа:

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

2. Виконати високочастотну фільтрацію зображення:

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

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

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

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

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

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

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

Завдання 6. Відфільтрувати зображення фільтром Собеля.

1. Створити маску фільтра Собеля для виділення горизонтальних границь

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

2. Створити маску фільтра Собеля для виділення вертикальних границь

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

3. Відфільтрувати зображення earth.bmp фільтрами Собеля.

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

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

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

```
>>F1=filter2(h,I); 48
```

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

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

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

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

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

Функція формування маски лінійного фільтра за бажаною АЧХ

fsamp2

Синтаксис

$h = \text{fsamp2}(f1, f2, N)$

Функція формує маску  $h$  лінійного двовимірного фільтра, ґрунтуючись на бажаній АЧХ двовимірного фільтра  $N$  для частот, переданих у векторах  $f1$  і  $f2$ .

Завдання 7. Підвищити різкість зображення за допомогою лінійного фільтра,

сформованого за бажаною АЧХ.

1. Отримати нормалізовані значення частот:

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

2. Обчислити бажану АЧХ -  $N$ , використовуючи так звану метрику міських кварталів:

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

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

3. Вивести на екран бажану АЧХ:

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

4. Сформувати маску фільтра за бажаною АЧХ:

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

5. Вивести на екран отриману АЧХ:

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

6. Прочитати зображення і вивести його на екран:

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

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

7. Відфільтрувати зображення:

```
>>I1=mat2gray(filter2(h,im2double(i)));
```

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

8. Контрастування результату фільтрації:

```
>>I2=imadjust(i1,[0 0.5],[]);
```

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

2.7. Функція формування маски лінійного фільтра методом перетворення частот `ftrans2`

Синтаксис:

$h = \text{ftrans2}(b)$

Функція формує маску  $h$  лінійного двовимірного фільтра використовуючи метод перетворення частот для трансформації одновимірного фільтра з коефіцієнтами  $b$ . Для перетворення частот використовується

спеціальна матриця трансформації  $t$ . За замовчуванням застосовується матриця Мак-Клеллана:

У функції  $h = \text{ftrans2}(b,t)$  можна вказати іншу матрицю трансформації.

Завдання 8. Здійснити розмиття зображення з метою придушення муару за допомогою лінійного двовимірного фільтра низької частоти (ФНЧ), сформованого з одновимірного ФНЧ методом перетворення частот:

1. Створити одновимірний ФНЧ 14 -го порядку з частотою зрізу 0.2:

```
>>B=fir1(14,0.2);% проектування фільтрів методом зважування  
>>Freqz(b,1,256)
```

2. Сформуувати двовимірний фільтр з одновимірного:

```
>>H=ftrans2(b);  
>>figure,freqz2(h)
```

3. Прочитати вихідне зображення і відфільтрувати отриманим фільтром:

```
>>[S,map]=imread('c:\Image\Athena.bmp');  
>>I=ind2gray(S,map);  
>>I=im2double(I);  
>>figure,imshow(I)  
>>I1=filter2(h,I);  
>>figure,imshow(I1)  
>>I2=mat2gray(I1);  
>>figure,imshow(I2)
```

Функція обчислення загальної нелінійної фільтрації `nlfilter`

Синтаксис

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

Функція `D = nlfilter(S,[mn],fun,P1,P2,...)` використовується для обробки бінарних і напівтонових зображень. Формує нове зображення, використовуючи

для фільтрації маску розміром `[mn]` і функцію, що здійснює фільтрацію,

`fun.P1,P2` - можливі параметри, передані в `fun`.

Завдання 9. Створити функцію з фільтрації імпульсного шуму для використання в якості аргументу `fun` у функції узагальненого нелінійного фільтра, який в цьому випадку буде використовуватися для видалення імпульсного шуму в зображенні. У функції фільтрація імпульсного шуму здійснюється операція усереднення з порогом, яка полягає в тому, центральному пікселю в межах маски присвоюється середнє значення яскравості всіх пікселів в межах маски в тому випадку, якщо різниця між початковим значенням центрального пікселя і середнім більше заданого

порогового значення. Для створення цієї функції з ім'ям 'AverageWithTh' в

редакторі m - файлів необхідно виконати наступні дії.

1. Створити новий m - файл командою головного меню:

File/New/m - file

2. Ввести команди функції:

```
function R=AverageWithTh(x, Th); % заголовочний рядок
[r c]=size(x);
n = r*c;
r = floor((r +1)/2); c = floor((c +1)/2);
s = sum(x(:))/n;
if(abs(x(r, c)-s))>Th
R = s;
else
R = x(r, c);
end;
end;
```

3. Створити новий підкаталог.

4. Зберегти в новому підкаталогові створену функцію в файл з однойменною назвою.

5. Прочитати зображення, зашумити імпульсним шумом і відфільтрувати його, використовуючи створену функцію:

```
>>[X,map]=imread('c:\Image\Mona.bmp');
>>I=im2double(ind2gray(X,map));
>>figure,imshow(I)
>>I1=imnoise(I,'salt & pepper');
>>figure,imshow(I1)
>>I2=nlfilter(I1,[3 3],'AverageWithTh',0.2);
>>figure,imshow(I2)
```

## 2. ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО РІШЕННЯ

Виконати наступні дії.

1. Прочитати палітрові зображення з файлу Earth.bmp, вивести на екран, перетворити в півтонове, додати імпульсний шум, відфільтрувати зашумлене

зображення медіанної фільтрацією і узагальненим нелінійним фільтром.

Зашумлені і відфільтровані зображення вивести у одному вікні для порівняння.

2. Виконати фільтрацію зображення з файлу Bigbird.bmp:

а) Відфільтрувати зображення за допомогою масок кругового градієнта:

Північ Північно-східний Східний Південно-східний

Південний Південно-західний Західний Північно-західний

Введення масок двовимірних лінійних фільтрів, побудова їх АЧХ, фільтрацію зображення з їх допомогою і вивід результату виконати в циклі; б)

Поліпшити зображення за допомогою масок лапласіановських фільтрів:

Введення масок і виведення результатів виконати в циклі.

3. Прочитати зображення з файлу `Technlgy.bmp`, вивести на екран, перетворити

в півтонове. Отримати маску фільтра Превітт. Виконати фільтрацію вихідного

напівтонового фрагмента маски фільтра Превітт окремо по горизонталі і по

вертикалі і разом на одному зображенні.

4. Прочитати кольорове зображення з файлу `'Vike.bmp'`, вивести на екран,

перетворити в півтонове. Взяти в якості АЧХ фільтра функцію відстані від

початку координат, сформувати по АЧХ маску фільтра і відфільтрувати

півтонування. Початкове півтонування та результати його обробки вивести на

екран.

5. Вивести АЧХ всіх фільтрів, створюваних функцією за заданням масок

зумовленого фільтра, в одному вікні з заголовками для кожного фільтра.

6. Зашумувати в циклі півтонування (вихідне взяти з файлу `'butterfly.bmp'` гаусовим шумом з різними дисперсіями: а) за замовчуванням (0.01), б)  $v = 0.5$

для математичного сподівання, рівного  $m = 0.5$ .

### 3. ПИТАННЯ

1. Які типи фільтрів створює функція з формування масок фільтрів `fspecial`?

2. У чому полягає алгоритм двовимірної згортки?

3. У яких функціях присутній алгоритм двовимірної згортки?

4. У чому відмінність алгоритму медіанної фільтрації від алгоритму фільтрації

за допомогою операції усереднення з порогом?

5. Які типи шумів формує функція по зашумленні зображень `imnoise`?

6. Для яких цілей можна використовувати функцію `freqz2`?

7. Яким чином можна сформувати маску лінійного фільтра за бажаною АЧХ?



8. Яка функція дозволяє сформувати двовимірний фільтр з одновимірного?

Завдання до лабораторної роботи №3

1. Повторити етапи, приведені в розділі «Хід роботи».
2. Оформити звіт.

### **Зміст звіту**

1. Тема.
2. Мета.
3. Обґрунтування вибраного зображення та необхідності обробки його.
4. Текст програми.
5. Результати.
6. Інтерпретація результатів роботи програми.
7. Висновки.

### **Використана література**

1. Гонсалес, Джевирс. Цифровая обработка сигналов: практический подход, 2-е издание. М.: Вильямс, 2009. – 992 с.:ил.
2. Прэтт У. Цифровая обработка изображений: - М.: Мир, 1982. – 790 с.
3. Дьяконов В.П., MATLAB 6.5 SP1/7/7 SP1/ Работа с изображениями и видеопотоками. – М.: СОЛОН-Пресс, 2010. – 400 с.
4. Обработка сигналов и изображений. Режим доступа: <http://matlab.exponenta.ru/imageprocess/book2/31.php>
5. Журавель, И.М. Краткий курс теории обработки изображений / И.М. Журавель. – М., 1999.
6. С.А.Свелеба, Ю.М.Корчак, Ю.М.Фургала. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Оптичні методи запису та обробки інформації»: Львівський національний університет імені Івана Франка, Львів, 2015.

# Сегментація зображень в системі Matlab методом k-середніх

## ЛАБОРАТОРНА РОБОТА №4

Мета роботи: ознайомлення з можливостями кластерного аналізу зображень засобами системи Matlab та набуття навиків перетворення колірному простору зображення та розбиття його на кластери за ознакою міри відстані у відповідному колірному просторі.

### Хід роботи

Розглянемо задачу, основною метою якої є автоматична сегментація на основі кластеризації (метод k-середніх) кольорових зображень, представлених в колірному просторі  $L^*a^*b^*$ .

#### Крок 1: Зчитування зображення.

Зчитуємо файл `hestain.png`, який містить гістологічний зріз зображення гемотоксину і еозину (H&E) в апертурі мікроскопа. Тут застосований метод фарбування по Граму для детального аналізу патологій.

```
clear all; clc; close all, rng('default'), format rational
he=imread('hestain.png'); %
figure, imshow(he), title('H&E зображення');
text(size(he, 2), size(he, 1)+15, ...
      'Image courtesy of Alan Partin, Johns Hopkins
      University', ...
      'FontSize', 7, 'HorizontalAlignment', 'right');
```

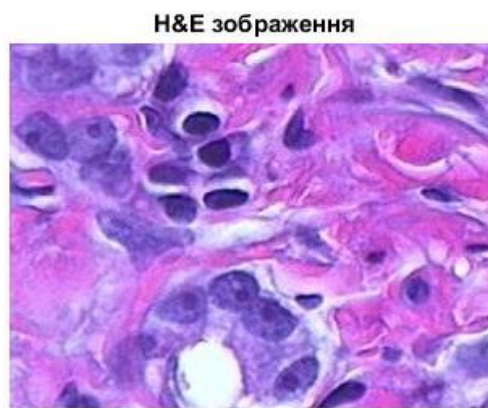


Image courtesy of Alan Partin, Johns Hopkins University

Рис. 1 - Зображення `hestain.png`

**Крок 2: Перетворення зображення з колірною системи RGB в колірну систему  $L^*a^*b^*$ .**

Яку кількість кольорів видно на зображенні, коли не брати до уваги можливість комбінації яскравостей? Насправді їх три: білий, блакитний і рожевий. Слід зазначити відмінності цих кольорів між собою. Кольорова

палітра  $L^*a^*b^*$  (вона ще відомо як CIELAB або CIE  $L^*a^*b^*$ ) дає можливість розрізняти ці візуальні відмінності.

Кольорова палітра  $L^*a^*b^*$  отримана на основі триколірних значень CIE XYZ. Простір  $L^*a^*b^*$  включає інформацію про значення інтенсивності 'L\*', значення колірності 'a\*', яке показує який колір вибраний на червоно-зеленій осі і значення колірності 'b\*', що показує який колір вибраний на блакитно-жовтій осі. Вся інформація про кольори міститься в значеннях 'a\*' і 'b\*'. Оцінити різницю між двома кольорами можна з використанням евклідової відстані (або ж підбором значення параметру 'distance' команди 'kmeans').

Перетворимо зображення в колірний простір  $L^*a^*b^*$  з використанням функцій `makecform` і `applycform`.

```
cform=makecform('srgb2lab'); % Створює структуру C
перетворення кольорів, яка описує типи перетворення колірних
просторів
lab_he=applycform(he, cform); % перетворює значення
інтенсивностей кольорів зображення I в той колірний простір,
який описано в структурі перетворення кольорів C
```

### **Крок 3: Класифікація кольорів у просторі 'a\*b\*' з використанням кластеризації (метод k-середніх).**

Кластеризація призводить до поділу об'єктів на групи. Кластеризація методом k-середніх призводить також до локалізації об'єктів в просторі. Пошук поділу, тобто який об'єкт до якого класу належить, відбувається на основі аналізу метричної відстані між об'єктами.

Далі на підставі інформації про кольори в просторі 'a\* b\*', кожному пікселю об'єкта присвоюється значення 'a\*' і 'b\*'. Використовуємо кластеризацію методом k-середніх для поділу об'єктів на кластери (кількість кластерів задати емпірично). Для цього використовуємо Евклідову метрику (або іншу).

```
ab=double(lab_he(:, :, 2:3));
nrows=size(ab, 1);
ncols=size(ab, 2);
ab=reshape(ab, nrows*ncols, 2);
nColors=4; % !!! Задати кількість кольорів розбиття,
добившись найкращого результату суб'єктивно !!!
[cluster_idx, cluster_center]=kmeans(ab, nColors, 'distance',
'sqEuclidean', ...
'Replicates', 3); % !!! Підбором значення параметру 'distance'
добитися найкращого результату суб'єктивно !!!
```

### **Крок 4: Присвоєння міток кожному пікселю зображення на основі методу k-середніх.**

Для кожного об'єкта на оригінальному документі метод k-середніх повертає індекс відповідного кластера. Значення параметра `cluster_center`, яке отримано в результаті застосування методу k-середніх буде використано при

подальшій демонстрації методу. Відзначимо пікселі, які містяться в `cluster_index`.

```
pixel_labels=reshape(cluster_idx, nrows,ncols);  
figure, imshow(pixel_labels, []), title('зображення відмічене  
кластерними індексами');
```

зображення відмічене кластерними індексами

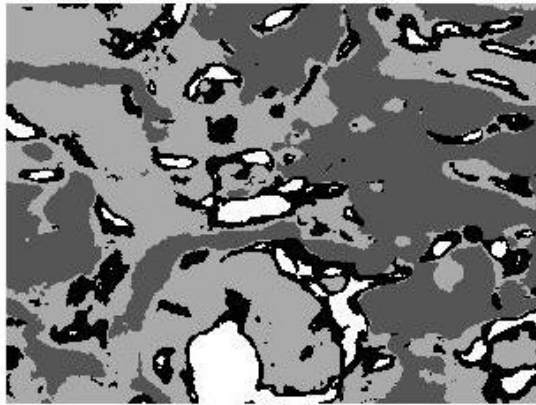


Рис. 2 - Зображення відмічене кластерними індексами

### Крок 5: Створення сегментованого зображення на основі кольорового.

Використовуючи параметр `pixel_labels`, можна розділити об'єкти на зображенні `hestain.png` по кольорах.

```
clusters= repmat(pixel_labels, [1 1 nColors]); % Дублюємо  
матрицю індексів кластерів nColors разів  
segmented_images=cell(1, nColors); % Масив комірок для  
сегментованого зображення (результат)  
for k=1 : nColors % Цикл виконується nColors разів (кількість  
кластерів)  
    clusters_tmp=clusters(:, :, k); % Тимчасова матриця-маска  
для k-го кластеру  
    clusters_tmp=uint8(clusters_tmp); % Перетворення типу до  
типу вхідного зображення he  
    clusters_tmp(clusters_tmp~=k)=0; % Обнулення всіх  
елементів належних іншим кластерам  
    clusters_tmp(clusters_tmp~=0)=1; % Елементи належні k-му  
кластеру рівні "1"  
    im=he; % Вхідне зображення в тимчасову змінну im  
    for i=1:3 % Для кожної з 3-х матриць R, G, B вхідного  
зображення  
        im(:, :, i) = im(:, :, i).*clusters_tmp; % Зважуємо з  
матрицею-маскою  
    end  
    segmented_images{k} = im; % Присвоюємо k-й комірці  
сегментоване зображення k-го сегменту  
    figure, imshow(im), title(strcat('Об'єкти в кластері', ...  
    ' ', num2str(k)));  
end
```

```
clear clusters_tmp i im
```

Объекти в кластері 1:



Рис. 3 - Объекти в кластері 1

Объекти в кластері 2:



Рис. 4 - Объекти в кластері 2

Объекти в кластері 3:



Рис. 5 - Объекти в кластері 3

Объекти в кластері 4:

Объекти в кластері 4



Рис. 6 - Объекти в кластері 4

## Завдання до лабораторної роботи №2

1. Повторити етапи, приведені в розділі «Хід роботи».
2. Оформити звіт.

### Зміст звіту

1. Тема.
2. Мета.
3. Обґрунтування вибраного зображення та необхідності обробки його.
4. Текст програми.
5. Результати.
6. Інтерпретація результатів роботи програми.
7. Висновки.

### Використана література

1. Гонсалес, Джевирс. Цифровая обработка сигналов: практический подход, 2-е издание. М.: Вильямс, 2009. – 992 с.:ил.
2. Прэтт У. Цифровая обработка изображений: - М.: Мир, 1982. – 790 с.
3. Дьяконов В.П., MATLAB 6.5 SP1/7/7 SP1/ Работа с изображениями и видеопотоками. – М.: СОЛОН-Пресс, 2010. – 400 с.
4. Обработка сигналов и изображений. Режим доступа: <http://matlab.exponenta.ru/imageprocess/book2/31.php>
5. Журавель, И.М. Краткий курс теории обработки изображений / И.М. Журавель. – М., 1999.

# Застосування фільтра Габора

## ЛАБОРАТОРНА РОБОТА №5

Мета роботи: ознайомлення з можливостями пакету прикладних програм системи Matlab та набуття навичок написання алгоритмів для створення фільтра Габора.

### Хід роботи

Розглянемо задачу, основною метою якої є одержання регулярностей, наявних в зображенні папілярного відбитку великого пальця людини.

#### Крок 1: Створення лістингу програми.

```
% Related links:
% https://en.wikipedia.org/wiki/Gabor_filter
% http://www.mathworks.com/

clear all; clc; close all, randn('state',0); format rational %edit
file

I=imread('Fingerprint1.jpg');
image_resize=imresize(I, [500,300]);
image_gray=rgb2gray(image_resize);
image_double=im2double(image_gray);
figure;
imshow(image_double);

grad = 55; % моє перетворення градусів
theta = (pi*grad)/180; % у радіани
%define the five parameters
% theta=10; %either 0 or pi/4 or pi/2 or 3pi/4
% theta визначає орієнтацію нормалі паралельних смуг функції
Габора в
% ?градусах?
lambda = 8; % довжину хвилі множника-косинуса
gamma=0.3; % коефіцієнт стиснення, що характеризує еліптичність
функції Габора
sigma=2.8; % Согласно правилу трёх сигм, практически все значения
экспоненты
%лежат в интервале
psi=0; % зсув фаз у градусах

sigma_x = sigma;
sigma_y = sigma/gamma;
```

```

nstds = 5;
xmax =
max(abs(nstds*sigma_x*cos(theta)),abs(nstds*sigma_y*sin(theta)));
xmax = ceil(max(1,xmax));
ymax =
max(abs(nstds*sigma_x*sin(theta)),abs(nstds*sigma_y*cos(theta)));
ymax = ceil(max(1,ymax));
xmin = -xmax; ymin = -ymax;
[x,y] = meshgrid(xmin:xmax,ymin:ymax);

x_theta=x*cos(theta)+y*sin(theta);
y_theta=-x*sin(theta)+y*cos(theta);

gb= exp(-
.5*(x_theta.^2/sigma_x^2+y_theta.^2/sigma_y^2)).*cos(2*pi/lambda*x
_theta+psi);

figure(2);
imshow(gb);
title(num2str(theta));
%imagesc(gb);
%colormap(gray);
%title('theta=...');

I2=imfilter(image_double,gb);
figure, imshow (I2);

```

## Крок 2: Відображення результату роботи програми.



Рис. 1 - Зображення відбитку пальця

Ядро згортки фільтра Габора має вигляд:



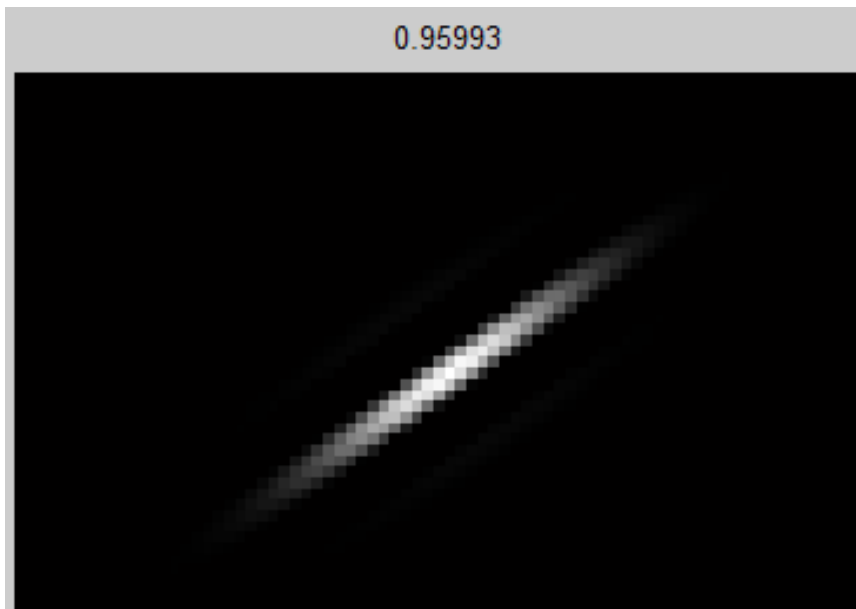


Рис. 2 - Ядро згортки фільтра Габора

Результат опрацювання зображення фільтром Габора має вигляд:



Рис. 2 - Результат опрацювання зображення фільтром Габора

### **Завдання до лабораторної роботи**

1. Повторити етапи, приведені в розділі «Хід роботи».
2. Оформити звіт.

## Зміст звіту

1. Тема.
2. Мета.
3. Обґрунтування вибраного зображення та необхідності обробки його.
4. Текст програми.
5. Результати.
6. Інтерпретація результатів роботи програми.
7. Висновки.

## Використана література

1. Гонсалес, Джевирс. Цифровая обработка сигналов: практический подход, 2-е издание. М.: Вильямс, 2009. – 992 с.:ил.
2. Прэйт У. Цифровая обработка изображений: - М.: Мир, 1982. – 790 с.
3. Дьяконов В.П., MATLAB 6.5 SP1/7/7 SP1/ Работа с изображениями и видеопотоками. – М.: СОЛОН-Пресс, 2010. – 400 с.
4. Обработка сигналов и изображений. Режим доступа: <http://matlab.exponenta.ru/imageprocess/book2/31.php>
5. Журавель, И.М. Краткий курс теории обработки изображений / И.М. Журавель. – М., 1999.

# Тривимірна візуалізація зображень в системі Matlab

## ЛАБОРАТОРНА РОБОТА №6

Мета роботи: ознайомлення з можливостями пакету прикладних програм системи Matlab та набуття навичок написання алгоритмів для 3D-відображення зрізів голови людини, одержаних із застосуванням магнітно-резонансної томографії.

### Хід роботи

Розглянемо задачу, основною метою якої є 3D-відображення зрізів голови людини, одержаних із застосуванням магнітно-резонансної томографії.

#### Крок 1: Створення лістингу програми.

```
% Related links:
% https://www.mathworks.com/help/matlab/visualize/techniques-for-visualizing-scalar-volume-data.html

clear all; clc; close all, randn('state',0); format long
%rational %edit file

% Задаємо вхідні змінні
current_path = '../13_MRI 3D/'; % Поточна директорія
my_dir = 'my_dir/'; % Результат роботи скрипту
mkdir(current_path, my_dir)
figCNT = 0; % Номер поточної фігури

%% load the data and transform the data array from 4-D to 3-D
load mri
D = squeeze(D);

%% Displaying Images of MRI Data
figure, figCNT=figCNT+1;
colormap(map)
image_num = 8;
image(D(:,:,image_num))
axis image
x = xlim;
y = ylim;
title('Displaying Images of MRI Data')
saveas(gcf, [current_path my_dir num2str(figCNT) '.bmp']);

%% Displaying a 2-D Contour Slice
cm = brighten(jet(length(map)), -.5);
figure, figCNT=figCNT+1;
colormap(cm)
contourslice(D, [], [], image_num)
```

```

axis ij
xlim(x)
ylim(y)
daspect([1,1,1])
title('Displaying a 2-D Contour Slice')
saveas(gcf, [current_path my_dir num2str(figCNT) '.bmp']);

%% Displaying 3-D Contour Slices
figure, figCNT=figCNT+1;
colormap(cm)
contourslice(D, [], [], [1,12,19,27],8);
view(3);
axis tight
title('Displaying 3-D Contour Slices')
saveas(gcf, [current_path my_dir num2str(figCNT) '.bmp']);

%% Applying an Isosurface to the MRI Data
figure, figCNT=figCNT+1;
colormap(map)
Ds = smooth3(D);
hiso = patch(isosurface(Ds,5),...
    'FaceColor',[1,.75,.65],...
    'EdgeColor','none');
isonormals(Ds,hiso)
title('Applying an Isosurface to the MRI Data')

% 3-D Visualization
% Adding Isocaps Show Cut-Away Surface

hcap = patch(isocaps(D,5),...
    'FaceColor','interp',...
    'EdgeColor','none');

% Defining the View
view(35,30)
axis tight
daspect([1,1,.4])

% Add Lighting
lightangle(45,45); % Підставити значення у відповідності до
варіанту завдання
lighting gouraud
hcap.AmbientStrength = 0.6;
hiso.SpecularColorReflectance = 0;
hiso.SpecularExponent = 50;
rotate3d

```

```
pause(10);  
saveas(gcf, [current_path my_dir num2str(figCNT) '.bmp']);
```

## Крок 2: Відображення результату роботи програми.

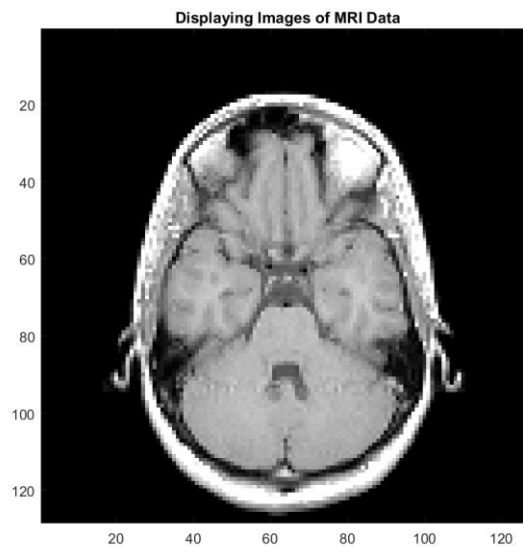


Рис. 1 – Зображення МРТ-зрізу голови

Підсвічене контрастування зрізу має вигляд:

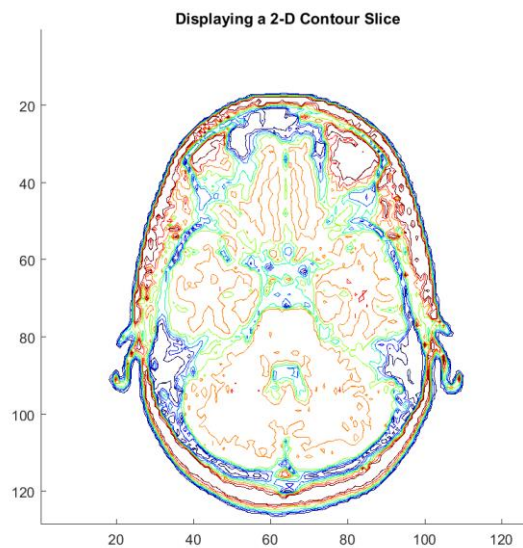


Рис. 2 - Підсвічене контрастування зрізу

Підсвічене контрастування кількох зрізів має вигляд:

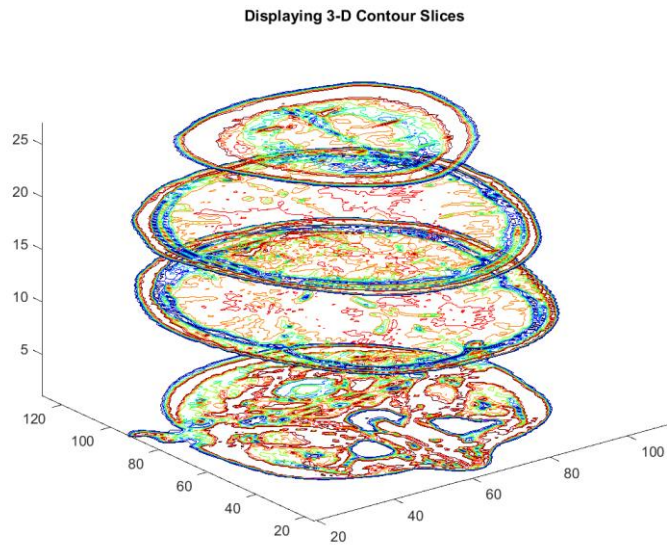


Рис. 3 - Підсвічене контрастування кількох зрізів

Тривимірна апроксимація голови людини за кількома МРТ-зрізами має вигляд:

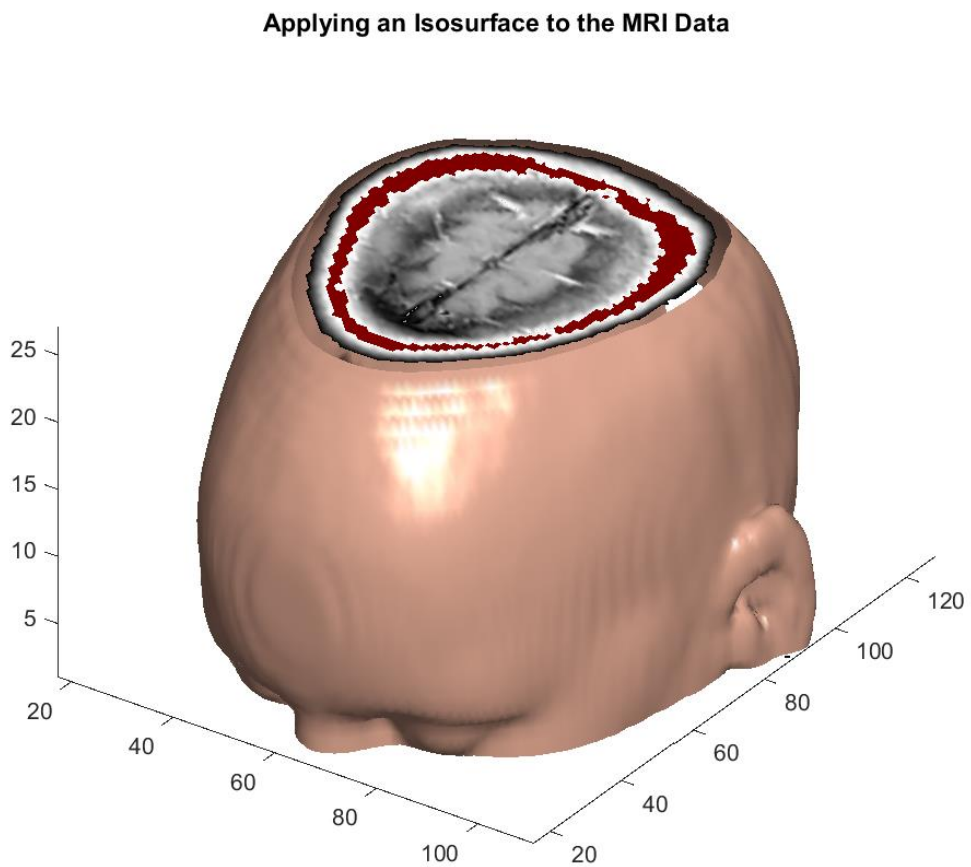


Рис. 4 - Тривимірна апроксимація голови людини за кількома МРТ-зрізами

## **Завдання до лабораторної роботи**

1. Повторити етапи, приведені в розділі «Хід роботи».
2. Оформити звіт.

### **Зміст звіту**

1. Тема.
2. Мета.
3. Обґрунтування вибраного зображення та необхідності обробки його.
4. Текст програми.
5. Результати.
6. Інтерпретація результатів роботи програми.
7. Висновки.

### **Використана література**

1. Гонсалес, Джевирс. Цифровая обработка сигналов: практический подход, 2-е издание. М.: Вильямс, 2009. – 992 с.:ил.
2. Прэйтт У. Цифровая обработка изображений: - М.: Мир, 1982. – 790 с.
3. Дьяконов В.П., MATLAB 6.5 SP1/7/7 SP1/ Работа с изображениями и видеопотоками. – М.: СОЛОН-Пресс, 2010. – 400 с.
4. Обработка сигналов и изображений. Режим доступа: <http://matlab.exponenta.ru/imageprocess/book2/31.php>
5. Журавель, И.М. Краткий курс теории обработки изображений / И.М. Журавель. – М., 1999.

# Автоматизоване визначення локалізації пухлини легенів в системі Matlab

## ЛАБОРАТОРНА РОБОТА №7

Мета роботи: ознайомлення з можливостями пакету прикладних програм системи Matlab та набуття навичок написання алгоритмів для автоматизованого визначення локалізації пухлини легенів.

### Хід роботи

Розглянемо задачу, основною метою якої є локалізація пухлини легенів на зображеннях зрізів, одержаних із застосуванням магнітно-резонансної томографії.

#### Крок 1: Створення лістингу програми.

```
% related links:
% www.mathworks.com

clear all; clc; close all, randn('state',0); format rational %edit
file

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
% Варіант №1
% UNCOMMENT NEXT LINE according to your task variant (if #1)
% file = 'ok1.jpg';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
% Варіант №2
% UNCOMMENT NEXT LINE according to your task variant (if #2)
% file = 'ok2.jpg';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%

imtool close all;
FontSize = 12;
initImage = imread(file);
[rows, columns] = size(initImage);

initImage = rgb2gray(initImage);
    initImage = medfilt2(initImage,[10 10]);

[B, A] = imhist(initImage);

C=A.*B;
D=A.*A;
E=B.*D;
```



```

n=sum(B);
Mean=sum(C)/sum(B);
var=sum(E)/sum(B)-Mean*Mean;
std= (var)^0.5;
thresholdValue = Mean+0.5*std;
bwImage = initImage > thresholdValue;
figure
imshow(bwImage)
title('binary image');

img_dil = imdilate(bwImage , strel('arbitrary', 20));
figure
imshow(img_dil);
title('dilated image');
bwImage = imerode(img_dil , strel('arbitrary', 20 ));
figure
imshow(bwImage);
title('eroded image');

bigMask = bwareaopen(bwImage, 2000);
finalImage = bwImage;
finalImage(bigMask) = false;

bwImage=bwareaopen(finalImage,55);
figure
imshow(bwImage)

labeledImage = bwlabel(bwImage, 8);
RegionMeasurements = regionprops(labeledImage, initImage, 'all');
Ecc = [RegionMeasurements.Eccentricity];
RegionNo = size(RegionMeasurements, 1);
allowableEccIndexes = (Ecc< 0.98);
keeperIndexes = find(allowableEccIndexes);
RegionImage = ismember(labeledImage, keeperIndexes);
bwImage=RegionImage;

figure
imshow(RegionImage)

%%%%%
clear labeledImage;
clear RegionMeasurements;
clear RegionNo;

labeledImage = bwlabel(bwImage, 8);
RegionMeasurements = regionprops(labeledImage, initImage, 'all');
figure
imshow(initImage);
title('Outlines', 'FontSize', FontSize);
axis image;
hold on;
boundaries = bwboundaries(bwImage);
numberOfBoundaries = size(boundaries, 1);

```

```

for k = 1 : numberOfBoundaries
    thisBoundary = boundaries{k};
    plot(thisBoundary(:,2), thisBoundary(:,1), 'r', 'LineWidth',
3);
end
hold off;
RegionMeas = regionprops(labeledImage, initImage, 'all');
RegionNo = size(RegionMeas, 1);

textFontSize = 14;
labelShiftX = -7;
RegionECD = zeros(1, RegionNo);

fprintf(1, 'Region number      Area      Perimeter      Centroid
Diameter\n');

for k = 1 : RegionNo

    RegionArea = RegionMeas(k).Area;
    RegionPerimeter = RegionMeas(k).Perimeter;
    RegionCentroid = RegionMeas(k).Centroid;
    RegionECD(k) = sqrt(4 * RegionArea / pi);
    fprintf(1, '#%2d          %11.1f %8.1f %8.1f %8.1f % 8.1f\n',
k, RegionArea, RegionPerimeter, RegionCentroid, RegionECD(k));
    text(RegionCentroid(1) + labelShiftX, RegionCentroid(2),
num2str(k), 'FontSize', textFontSize, 'FontWeight', 'Bold');
end

```

## Крок 2: Відображення результату роботи програми.



Рис. 1 – Бінаризоване зображення зрізу

Зображення з вирівняними границями пухлини має вигляд:



Рис. 2 - Зображення з вирівняними границями пухлини

Зображення уточнених країв основної маси пухлини має вигляд:



Рис. 3 - Уточнені краї основної маси пухлини

Зображення уточнених згладжених країв основної маси пухлини має вигляд:

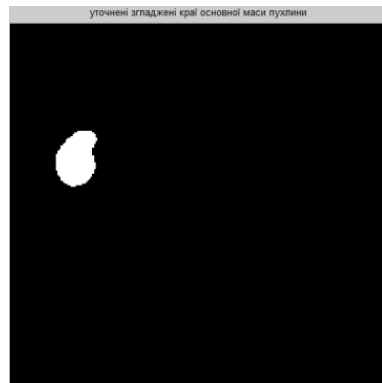


Рис. 4 - Уточнені загладженні краї основної маси пухлини

Зображення із видаленими мінорними конгломератами має вигляд:

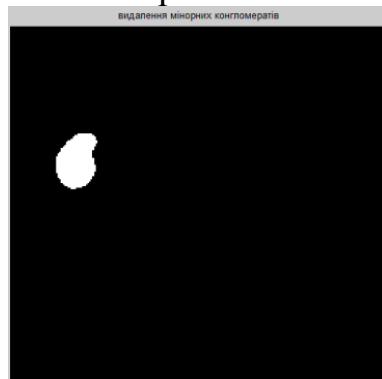


Рис. 5 - Видалення мінорних конгломератів

Зображення із видаленими границями пухлини має вигляд:

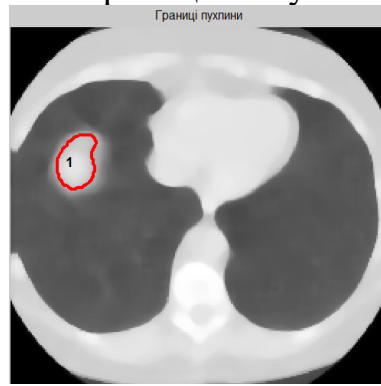


Рис. 6 - Границі пухлини

### **Завдання до лабораторної роботи**

1. Повторити етапи, приведені в розділі «Хід роботи».
2. Оформити звіт.

### **Зміст звіту**

1. Тема.
2. Мета.
3. Обґрунтування вибраного зображення та необхідності обробки його.
4. Текст програми.
5. Результати.
6. Інтерпретація результатів роботи програми.
7. Висновки.

### **Використана література**

1. Гонсалес, Джевирс. Цифровая обработка сигналов: практический поход, 2-е издание. М.: Вильямс, 2009. – 992 с.:ил.
2. Прэтт У. Цифровая обработка изображений: - М.: Мир, 1982. – 790 с.
3. Дьяконов В.П., MATLAB 6.5 SP1/7/7 SP1/ Работа с изображениями и видеопотоками. – М.: СОЛОН-Пресс, 2010. – 400 с.
4. Обработка сигналов и изображений. Режим доступа: <http://matlab.exponenta.ru/imageprocess/book2/31.php>
5. Журавель, И.М. Краткий курс теории обработки изображений / И.М. Журавель. – М., 1999.

# Знаходження контурів елементів зображення в системі Matlab

## ЛАБОРАТОРНА РОБОТА №8

Мета роботи: ознайомлення з можливостями програмного середовища Matlab та набуття навиків створення алгоритмів автоматизованого пошуку контурів елементів зображення.

### Хід роботи

Розглянемо задачу, основною метою якої є автоматизований пошук контурів елементів зображення задля визначення імовірності розвитку та прогресування дерматологічного онкологічного процесу (меланома).

#### Крок 1: Зчитування зображення.

Очищуємо робочий простір Matlab IDE.

```
% Блок коду для варіанту НЕПАРНИЙ
% Очищуємо робочий простір Matlab IDE
clear all; clc; close all, rng('default'), format rational %edit
file
```

#### Крок 2 Задаємо вхідні змінні.

```
% Задаємо вхідні змінні
current_path = '../image_processing_lab8/'; % Поточна директорія
my_dir = 'my_dir/'; % Результат роботи скрипту
mkdir(current_path, my_dir)
figCNT = 0; % Номер поточної фігури
```

#### Крок 3 Задати своє зображення розміром [50\*50] пікселів згідно варіанту завдання.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Задати своє зображення розміром [50*50] пікселів згідно варіанту
завдання
I = imread([current_path '\melanoma\test.png']);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure, figCNT=figCNT+1;
imshow(I, [])
title('initial image');
saveas(gcf, [current_path my_dir num2str(figCNT) '.bmp']);
```

initial image



Рис. 1 – Вхідне зображення

#### Крок 4: Конвертуємо вхідне зображення у бінарне

```
% Конвертуємо вхідне зображення у бінарне
BI = im2bw(I, graythresh(I));
figure, figCNT=figCNT+1;
imshow(BI, [])
title('initial double image');
saveas(gcf, [current_path my_dir num2str(figCNT) '.bmp']);
```

initial double image

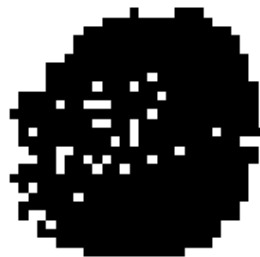


Рис. 2 – Бінаризоване зображення

#### Крок 5: Проводимо операцію нарощування зображення

```
% Проводимо операцію нарощування зображення
% % Примітка: замість команди imdilate()
% % зачтововується команда imerode(),
% % оскільки значимі елементи зображення "чорного" кольору
ROW = 2; % Радіус прямокутних структурних елементів (згідно
варіанту завдання)
COLUMN = 2; % Висота прямокутних структурних елементів (згідно
варіанту завдання)
se=strel('rectangle', [ROW COLUMN]) % Масив структурних елементів
об'єкта для нарощування
BI=imerode(BI, se);
figure, figCNT=figCNT+1;
imshow(BI, [])
```

```
title('Dilated double image');
saveas(gcf, [current_path my_dir num2str(figCNT) '.bmp']);
```

### Dilated double image

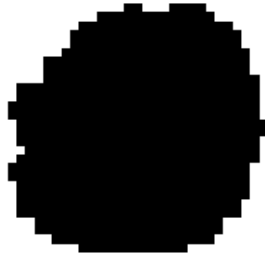


Рис. 3 – Нарощене бінаризоване зображення

### Крок 6: Відображаємо контури зображення

```
% Відображаємо контури зображення
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
% Розкоментувати лінійку у відповідності до номеру варіанту
завдання
% [B,L] = bwboundaries(BW,'noholes'); % Непарний номеру варіанту
завдання
[B,L] = bwboundaries(BI,8,'holes'); % Парний номеру варіанту
завдання
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%

figure, figCNT=figCNT+1;
imshow(label2rgb(L, @jet, [.5 .5 .5]))

hold on
for k = 2:length(B)
    boundary = B{k};
    plot(boundary(:,2), boundary(:,1), 'r', 'LineWidth', 2)
end
clear k boundary
title('initial image and boundaries');
saveas(gcf, [current_path my_dir num2str(figCNT) '.bmp']);
```

### initial image and boundaries

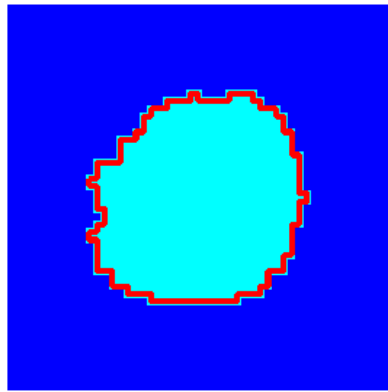


Рис. 4 – Контури зображення

Знаходимо контури зображення іншим способом.

### Крок 7: Очищуємо робочий простір Matlab IDE

```
% Очищуємо робочий простір Matlab IDE  
clear all; clc; close all, rng('default'), format rational %edit  
file
```

### Крок 8: Задаємо вхідні змінні

```
% Задаємо вхідні змінні  
current_path = '../image_processing_lab8/'; % Поточна директорія  
my_dir = 'my_dir/'; % Результат роботи скрипту  
mkdir(current_path, my_dir)  
figCNT = 0; % Номер поточної фігури
```

**Крок 9: Задати своє зображення розміром [50\*50] пікселів згідно варіанту завдання**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% Задати своє зображення розміром [50*50] пікселів згідно варіанту  
завдання  
I = imread([current_path '\melanoma\test.png']);  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
figure, figCNT=figCNT+1;  
imshow(I, [])  
title('initial image');  
saveas(gcf, [current_path my_dir num2str(figCNT) '.bmp']);
```



initial image



Рис. 5 – Вхідне зображення

### Крок 10: Конвертуємо зображення в тип "відтінки сірого"

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%  
% Задати своє зображення розміром [50*50] пікселів згідно варіанту  
завдання  
I = imread([current_path '\melanoma\test.png']);  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%  
figure, figCNT=figCNT+1;  
imshow(I, [])  
title('initial image');  
saveas(gcf, [current_path my_dir num2str(figCNT) '.bmp']);  
initial greyscale image
```

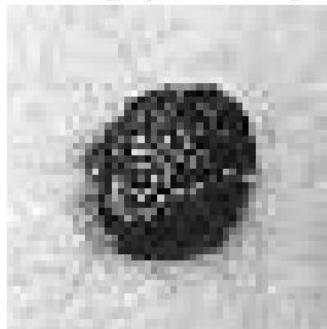


Рис. 6 – Конвертоване зображення в тип "відтінки сірого"

### Крок 11: Знайдемо контури зображення за допомогою фільтрів "Canny" та "Prewitt"

```
% Знайдемо контури зображення за допомогою фільтрів "Canny" та  
"Prewitt"  
GI1 = edge(I, 'Canny');  
  
GI2 = edge(I, 'Prewitt');  
  
figure, figCNT=figCNT+1;  
imshowpair(GI1,GI2, 'montage')  
title('Compare Canny and Prewitt techniques');  
saveas(gcf, [current_path my_dir num2str(figCNT) '.bmp']);  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%
```

```

% Розкоментувати лінійку у відповідності до номеру варіанту
завдання
% GI = GI1; title('edge detection by Canny techniques'); %
Непарний номеру варіанту завдання
GI = GI2; % Парний номеру варіанту завдання
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear GI1 GI2

```

### Compare Canny and Prewitt techniques

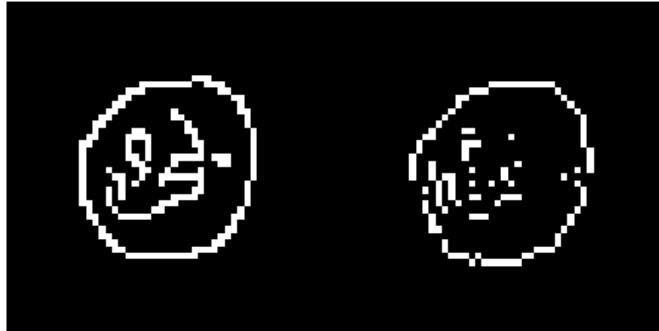


Рис. 7 – Знайдені контури зображення за допомогою фільтрів "Canny" та "Prewitt"

## Завдання до лабораторної роботи

1. Повторити етапи, приведені в розділі «Хід роботи».
2. Оформити звіт.

### Зміст звіту

1. Тема.
2. Мета.
3. Обґрунтування вибраного зображення та необхідності обробки його.
4. Текст програми.
5. Результати.
6. Інтерпретація результатів роботи програми.
7. Висновки.

## Використана література

1. Гонсалес, Джевирс. Цифровая обработка сигналов: практический поход, 2-е издание. М.: Вильямс, 2009. – 992 с.:ил.
2. Прэтт У. Цифровая обработка изображений: - М.: Мир, 1982. – 790 с.
3. Дьяконов В.П., MATLAB 6.5 SP1/7/7 SP1/ Работа с изображениями и видеопотоками. – М.: СОЛОН-Пресс, 2010. – 400 с.
4. Обработка сигналов и изображений. Режим доступа: <http://matlab.exponenta.ru/imageprocess/book2/31.php>
5. Журавель, И.М. Краткий курс теории обработки изображений / И.М. Журавель. – М., 1999.