

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

МЕТОДИЧНІ ВКАЗІВКИ
ДО ЛАБОРАТОРНИХ РОБІТ З ДИСЦИПЛІН
«ОСНОВИ ТЕОРІЇ ІНТРОСКОПІЇ», «ТЕОРІЯ ТА ТЕХНІКА ІНТРОСКОПІЇ»
для студентів денної форми навчання спеціальностей
7.091002, 8.091002 – біотехнічні та медичні апарати і системи,
7.090804, 8.090804 – фізична та біомедична електроніка
та напряму 6.051402 – біомедична інженерія

ЗАТВЕРДЖЕНО
Кафедрою біомедичних
електронних пристроїв і
систем
Протокол № від

Харків 2010

Методичні вказівки до лабораторних робіт з дисциплін «Основи теорії інтроскопії», «Теорія та техніка інтроскопії» для студентів спеціальностей 7.091002, 8.091002 – біотехнічні та медичні апарати і системи, 7.090804, 8.090804 – фізична та біомедична електроніка та напряму 6.051402 – біомедична інженерія / Упоряд. Головенко В.М., Авер'янова Л.О., Євдокименко С.О. – Харків, ХНУРЕ, 2009.- с.

Упорядники: В.М.Головенко
 Л.О.Авер'янова
 С.О.Євдокименко

ЗМІСТ

ВСТУП	5
Тема 1. ПРИНЦИПИ ПОБУДОВИ ТА ФУНКЦІОНУВАННЯ ПІДСИСТЕМИ ФОРМУВАННЯ ІНТРОСКОПІЧНОГО ЗОБРАЖЕННЯ	7
Лабораторна робота №1. Дослідження параметрів та властивостей цифрових зображень	7
1.1 Мета роботи	7
1.2 Підготовка до виконання роботи	7
1.3 Опис досліджуваних зображень	21
1.4 Порядок виконання роботи	21
1.5 Зміст звіту	28
1.6 Контрольні запитання	28
Тема 2. ПРИНЦИПИ ФУНКЦІОНУВАННЯ ПІДСИСТЕМИ АНАЛІЗУ ІНТРОСКОПІЧНОГО ЗОБРАЖЕННЯ	29
Лабораторна робота №2. Вивчення математичних операцій поелементного перетворення зображень	29
2.1 Мета роботи	29
2.2 Підготовка до виконання роботи	30
2.3 Порядок виконання роботи	44
2.4 Зміст звіту	46
2.5 Контрольні запитання	46
Лабораторна робота №3. Синтез та візуалізація тестових зображень у системі MATLAB	47
3.1 Мета роботи	47
3.2 Підготовка до виконання роботи	47
3.3 Порядок виконання роботи	63
3.4 Зміст звіту	68
3.5 Контрольні запитання	68
Лабораторна робота №4. Створення графічних препаратів діагностичних зображень у системі MATLAB	69
4.1 Мета роботи	69
4.2 Підготовка до виконання роботи	69
4.3 Порядок виконання роботи	83
4.4 Зміст звіту	84
4.5 Контрольні запитання	84
Лабораторна робота №5. Морфологічні перетворення та структурний аналіз зображень в MATLAB	85
5.1 Мета роботи	85
5.2 Підготовка до виконання роботи	85
5.3 Порядок виконання роботи	100
5.4 Зміст звіту	101
5.5 Контрольні запитання	101

ТЕМА 3. ПРИНЦИПИ ФУНКЦІОНУВАННЯ ПІДСИСТЕМИ РЕКОНСТРУКЦІЇ ТОМОГРАФІЧНОГО ЗОБРАЖЕННЯ	102
Лабораторна робота №6. «Вивчення методів реконструкції та перетворення томографічних зображень»	102
6.1 Мета роботи	102
6.2 Підготовка до виконання роботи	102
6.3 Порядок виконання роботи.	118
6.4 Зміст звіту	119
6.5 Контрольні запитання	119
ТЕМА 4. ПРИНЦИПИ ФУНКЦІОНУВАННЯ ПІДСИСТЕМИ ВІЗУАЛІЗАЦІЇ ЗОБРАЖЕННЯ	120
Лабораторна робота №7. «Візуалізація медичних інтроскопічних зображень у стандарті DICOM»	120
7.1 Мета роботи	120
7.2 Підготовка до виконання роботи	120
7.3 Порядок виконання роботи.	133
7.4 Зміст звіту	134
7.5 Контрольні запитання	134
ПЕРЕЛІК ПОСИЛАНЬ	135

ВСТУП

Медичні інтроскопічні системи є спеціалізованими технічними засобами візуалізації структури людського організму та відображення фізіологічних процесів у ньому. В медичних інтроскопічних системах застосовані новітні науково-технічні досягнення електроніки, радіотехніки, прикладної математики та комп'ютерних технологій. Інтроскопічні системи дозволяють отримати найціннішу діагностичну інформацію, не порушуючи цілісності організму людини, тому у всьому світі ці засоби діагностики посідають чільне місце в системі охорони здоров'я населення. Зростає необхідність підготовки спеціалістів технічного профілю, здатних розробляти та експлуатувати найскладніші медичні інтроскопічні системи.

Основними напрямками медичної інтроскопії є ультразвукова (УЗ) інтроскопія, рентгенівська комп'ютерна томографія (РКТ), емісійна томографія (ЕТ) та томографія на основі ефекту ядерного магнітного резонансу (МРТ).

Суть та задачі дисциплін. Суть дисциплін «Теорія і техніка інтроскопії», «Основи теорії інтроскопії» полягає у вивченні фізико-математичних, технічних та інформаційних основ інтроскопії, методів та засобів отримання зображень внутрішніх частин біологічного об'єкту, непрозорих для видимого світла, побудови апаратів і систем, призначених для інтроскопічних досліджень. Розглядаються фізичні основи отримання рентгенівського або гамма- випромінювання, ультразвукових хвиль, їх проходження скрізь тіло людини, методи детектування енергетичного випромінювання або відлучених ультразвукових хвиль, ефект ядерного магнітного резонансу та засоби зображення розподілення ядер водню у тілі людини, методи формування медичного інтроскопічного зображення, математичні методи реконструкції зображення, алгоритми реконструкції, побудова інтроскопічних апаратів і систем, функціональні та принципові схемні рішення цих пристроїв.

Для вивчення теорії та техніки медичної інтроскопії необхідні базові знання з фізики, вищої математики, біофізики, електротехніки та електроніки.

За результатами навчання студенти мають знати:

- принципи отримання інтроскопічних зображень;
- теоретичні основи взаємодії різних видів випромінювання з біологічним середовищем;
- методи отримання проєкцій та розрізів біологічного об'єкту;
- теоретичні методи реконструкції зображення зрізів за проєкційними даними;
- структурні схеми типових інтроскопічних систем.

Робоча програма дисциплін «Теорія і техніка інтроскопії», «Основи теорії інтроскопії» передбачає лекційні, практичні та лабораторні заняття. Виконання лабораторних робіт дозволяє студентові під керівництвом викладача ознайомитись із принципами формування інтроскопічного зображення, реалізовувати окремі функції та режими роботи інтроскопічних систем за допомогою сучасних програмних засобів для аналізу, синтезу та перетворення зображень. Студенти, які успішно виконали лабораторний практикум, можуть самостійно здійснювати імітаційні експерименти, набувати навичок дослідницької роботи у галузі медичної інтроскопії, відпрацьовувати методики аналізу та програмної обробки діагностичних зображень.

Порядок виконання лабораторних робіт. На початку першого заняття всі студенти повинні ознайомитися з правилами техніки безпеки і розписатися про це в журналі обліку виконання лабораторних робіт. Студенти, які не ознайомилися з правилами техніки безпеки, до виконання лабораторних робіт не допускаються.

Кожній лабораторній роботі (ЛР) повинна передувати самостійна підготовка студентів, у процесі якої потрібно вивчити методичні вказівки до лабораторної роботи, конспект лекцій та рекомендовані літературні джерела. Перед початком ЛР викладач перевіряє підготовленість студентів до виконання конкретної лабораторної роботи, де студенти мають знати мету і порядок виконання роботи.

Результати виконання ЛР відображаються у звіті, який повинен містити: назву лабораторної роботи, мету роботи, структурні схеми, графіки, таблиці з даними, тексти програм та результати їх виконання, а також висновки з результатами аналізу отриманих даних.

До початку наступної ЛР студент повинен надати викладачеві повністю оформлений звіт про попередню роботу та захистити її. Залік з ЛР студент отримує після співбесіди з викладачем за темою виконання робіт.

ТЕМА 1. ПРИНЦИПИ ПОБУДОВИ ТА ФУНКЦІОНУВАННЯ ПІДСИСТЕМИ ФОРМУВАННЯ ІНТРОСКОПІЧНОГО ЗОБРАЖЕННЯ

Підсистема формування зображення (ПФЗ) є найбільш важливим елементом будь-якої медичної інтроскопічної системи. В загальному випадку на ПФЗ покладено наступні функції: формування аналогового зображення; перетворення аналогового зображення у цифрове; попередня обробка цифрового зображення; запис, архівація та/або передача цифрового зображення. Кожна з систем медичної інтроскопії може відрізнитись від іншої способом реалізації ПФЗ. Проте основним результатом роботи всіх медичних інтроскопічних систем є формування діагностичного зображення. Воно має найбільш адекватно відображати внутрішню будову та властивості структур організму людини, які виявляються внаслідок певної фізичної взаємодії у процесі дослідження.

Інтроскопічне зображення є документом, носієм інформації, який є обов'язковим при здійсненні багатьох клінічних дій: при плануванні та проведенні хірургічного втручання та променевої терапії, при проведенні ортопедичної корекції, протезування, медичної експертизи тощо. Тому отримання якомога більш інформативного, достовірного та придатного для клінічного аналізу інтроскопічного зображення є основною задачею при розробці та технічному удосконаленні медичних інтроскопічних систем.

ЛАБОРАТОРНА РОБОТА №1

«Дослідження параметрів та властивостей цифрових зображень»

1.1 Мета роботи

Вивчення процесу формування цифрового зображення, його основних параметрів, способів запису та архівації.

1.2 Підготовка до виконання роботи

При підготовці до ЛР необхідно опрацювати теоретичний матеріал щодо структури та основних функцій підсистеми формування зображення у складі медичної інтроскопічної системи, проаналізувати особливості отримання цифрового зображення та опису його основних параметрів.

1.2.1 Основні властивості цифрового зображення.

Аналогове зображення є неперервною функцією дійсних змінних $I(x, y)$, де I – це інтенсивність (яскравість) в точці з координатами (x, y) . Аналогове зображення можна відобразити та обробити на комп'ютері лише після його переведення у цифровий формат. Процедура оцифрування зображення полягає у заміні неперервної множини значень яскравості аналогового зображення на множину квантованих значень яскравості, кожному з яких ставиться у відповідність двійкове число. Аналогове зображення перетворюється на впорядкований масив точок, де кожна точка, крім координат, описується цифровим кодом, який містить інформацію про яскравість та колір.

Для оцифрування аналогового зображення треба виконати його поділ на ділянки з індивідуальними рівнями яскравості. Це відбувається у два етапи: здійснюється просторова дискретизація та квантування яскравості (рис.1.1).

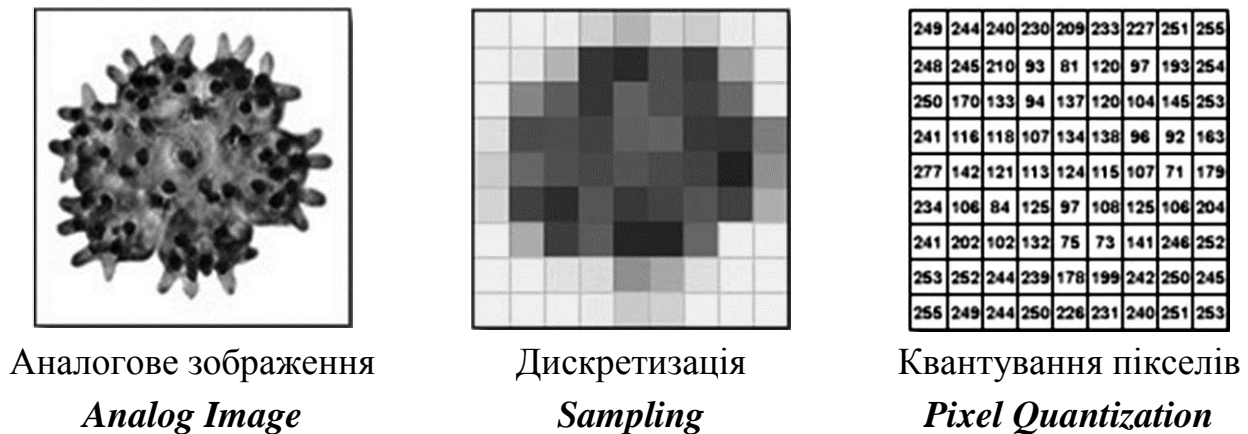


Рисунок 1.1 – Етапи створення цифрового зображення біооб'єкту

За **способом просторового формування** цифрові зображення можуть бути векторними та растровими. Векторне зображення формується із застосуванням набору графічних примітивів. Растрове ж зображення формується як двомірний масив елементів, де кожен елемент містить інформацію про градацію кольору. Растр – форма представлення зображення у вигляді впорядкованої множини дискретних елементів (пікселів). Піксел (**picture element**) – це найменша частинка, елемент зображення.

Растрове цифрове зображення представляється прямокутною (квадратною) матрицею (m, n) пікселів, які структуровані у декартовій системі координат та набувають певного значення інтенсивності. Щодо цифрових зображень зазвичай застосовують піксельну систему координат

(рис.1.2), у якій зображення поділене на окремі дискретні елементи, впорядковані зверху донизу (у рядки з номерами r) та зліва направо (у стовпці з номерами c). Значення піксельних координат є цілими числами: $r \in [1, m]$; $c \in [1, n]$.

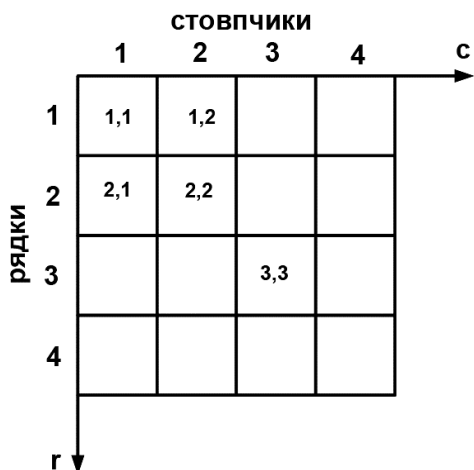


Рисунок 1.2 – Піксельна система координат

Піксельна координатна система дозволяє встановити зв'язок між матричним представленням зображення та його візуалізацією. Наприклад, дані пікселя у п'ятому рядку та другому стовпці запам'ятовуються у матриці як (5,2). Матричні індекси надалі використовуються для доступу до значень конкретних пікселів при цифровій обробці зображення. Добуток величин $m \times n$ визначає розміри цифрового зображення в пікселях (*Pixel Dimensions*).

При формуванні зображення насамперед необхідно визначити його **просторове розрізнення** (*Resolution*). Цей параметр зображення визначає кількість пікселів в одиниці довжини (табл.1.1) та впливає на детальність відтворення початкового зображення після дискретизації.

Таблиця 1.1 – Відповідність значень просторового розрізнення зображення

Розрізнення, пікселів/дюйм (dpi)	Відповідність		Розмір пікселя, мм
	пікселів/см	пікселів/мм	
200	78,74	7,87	0,127
300	118,11	11,81	0,085
600	236,22	23,62	0,042

Просторове розрізнення залежить від апаратних можливостей формуючої (скануючої) системи, насамперед від кількості давачів та відстані між ними. Ці параметри визначають максимальну кількість відліків та відстань між ними по

координаті – інтервал дискретизації (рис.1.3). Сучасні скануючі системи досягли апаратного розрізнення 9600 dpi (для порівняння: зоровий аналізатор людини може розрізнити фрагменти зображення не детальніше, ніж 6 відліків/мм, що становить 150 dpi).

При формуванні растрового цифрового зображення застосовується ортогональна структура розташування відліків по координаті (рис.1.4).

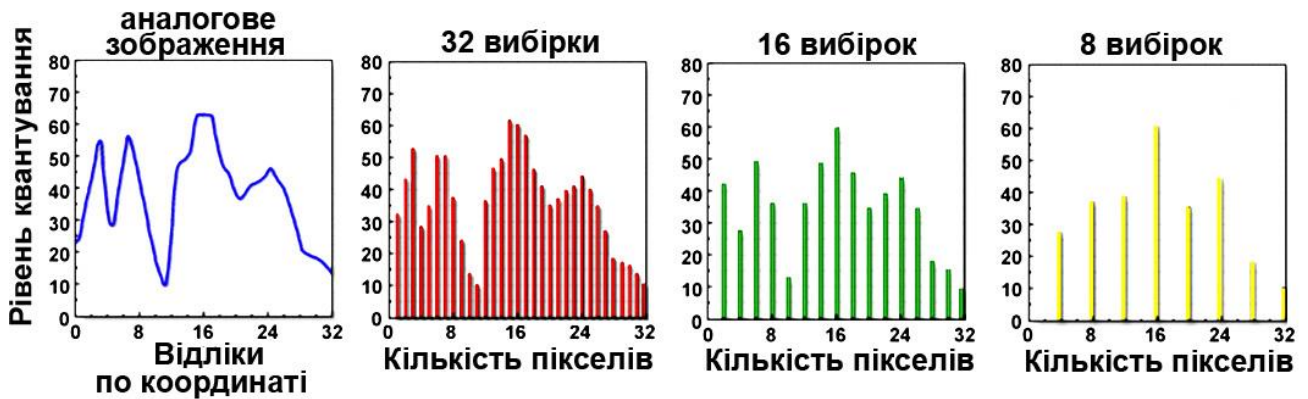


Рисунок 1.3 – Зміна точності відтворення рядка зображення в залежності від кількості вибірок значень яскравості

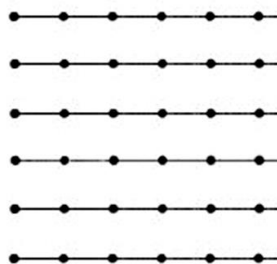


Рисунок 1.4 – Ортогональне розташування відліків при дискретизації зображення

В залежності від задачі кількість вибірок даних при формуванні цифрового зображення може бути меншою, ніж кількість можливих відліків по координаті (наприклад, при доступному апаратному розрізненні сканера 1200 dpi текст сканується з меншим розрізненням 200 dpi). Результат такого «прорідження» (рис.1.3) призводить до зниження точності відтворення зображення за рахунок збільшення інтервалу дискретизації. За певних умов розрізнення зображеного об'єкта стає неможливим, його інформаційний зміст помітно втрачається (рис.1.5).

Точність відтворення початкового зображення залежить від співвідношення між фізичними розмірами певних елементів зображення та обраним інтервалом його дискретизації. Його визначають за критерієм Найквіста-Котельникова:

частота дискретизації f_D має задовольняти нерівності:

$$f_D > 2f_B,$$

де f_B - верхня гранична просторова частота зображення, яке дискретизується.

Це підтверджує теорема дискретизації Шеннона: цифровий пристрій повинен використовувати інтервал дискретизації не більший, ніж половина розміру найменшого видимого елемента зображення. Практично це означає, що просторове розрізнення зображення є прийнятним, якщо будь-яка його видима деталь розділяється під час дискретизації як мінімум на дві ділянки. Елементи зображення, розмір яких менший за інтервал дискретизації, мають більшу просторову частоту, отже не можуть бути точно відображеними у цифровому форматі.

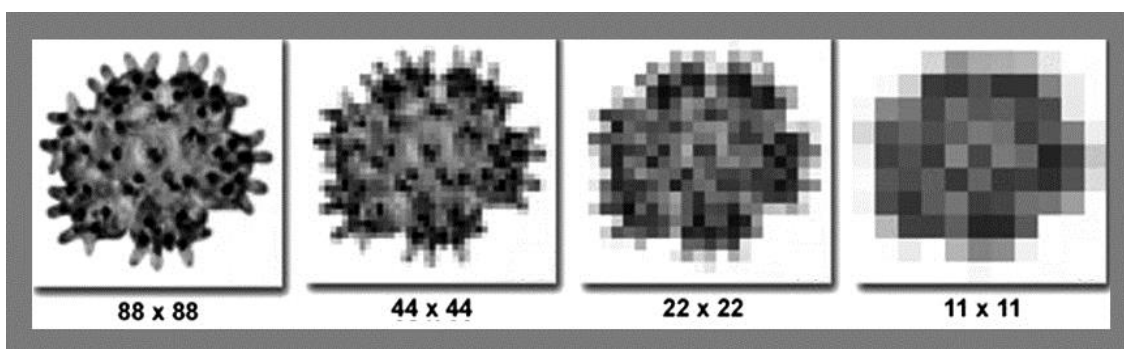


Рисунок 1.5 – Погіршення відтворення деталей зображення біооб'єкту при збільшенні інтервалу дискретизації

При порушенні критерія Найквіста-Котельникова після просторової дискретизації на зображенні можуть з'являтися спотворення. При **субдискретизації** відстань між пікселями є більшою за деталі аналогового зображення. В результаті на цифровому зображенні з'являються деталі, більш низькочастотні, ніж на аналоговому зразку. При $f_D \approx 1,5 f_B$ вони помітно спотворюють зображення. Як результат виникають викривлення текстури, зменшується чіткість зображення. Так, у зразках, що містять періодично повторювані ділянки, виникає криволінійний муаровий візерунок (рис.1.6а). Якщо ж початкове зображення містить нахилені лінії, то після дискретизації межі цих ліній стають зубчастими (рис.1.6 б). На цифрових медичних

зображеннях такі спотворення можуть призвести до виникнення несправжніх контурів та похибок відтворення меж анатомічних структур.

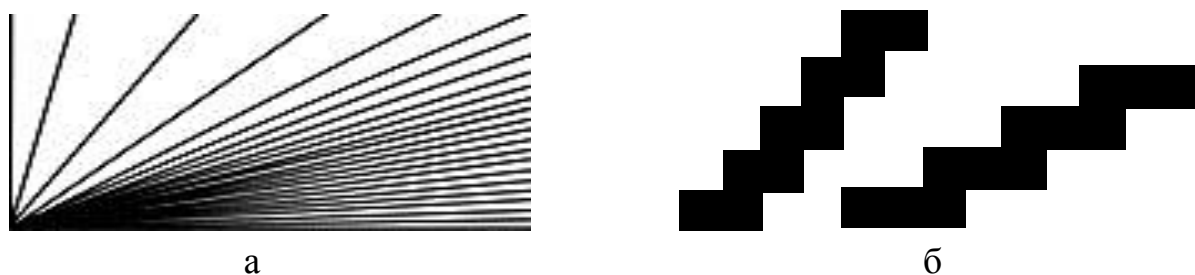


Рисунок 1.6 – Спотворення зображення ліній:
а - «муар»; б – «східці»

Піксели цифрового зображення задають не тільки його просторові характеристики, але й містять інформацію про яскравісні та кольорові параметри. Цих параметрів кожен піксел набуває в результаті **квантування за рівнями яскравості**. Діагностичний сигнал, зафіксований детекторами інтроскопічної системи, використовується для визначення величини «інтенсивність зображення», яка плавно змінюється у певному діапазоні, проте в процесі квантування має перетворитись в набір цілих числових значень – рівнів яскравості (*levels of brightness*).

Квантування за рівнем яскравості полягає у знаходженні для кожного відліку яскравості найближчого за значенням рівня квантування (з доступного набору рівнів квантування). Кількість рівнів квантування (діапазон градацій яскравості) N_{KB} пов'язана з кількістю b двійкових розрядів (біт) аналогово-цифрового перетворювача (АЦП) співвідношенням $N_{KB} = 2^b$.

Так, при квантуванні з розрядністю 8 біт зображення буде утворене $N_{KB} = 2^8 = 256$ значеннями (градаціями) яскравості, які представляються відповідними 8-бітними двійковими кодами у діапазоні від «00000000» (чорний) до «11111111» (білий).

Отже, числове значення (рівень яскравості) кожного піксела в цифровому зображенні являє собою усереднену інтенсивність початкового зображення на інтервалі дискретизації (рис.1.1, 1.7).

Кількість рівнів квантування N_{KB} є другим параметром цифрового зображення, який значно впливає на розрізнення деталей зображення.

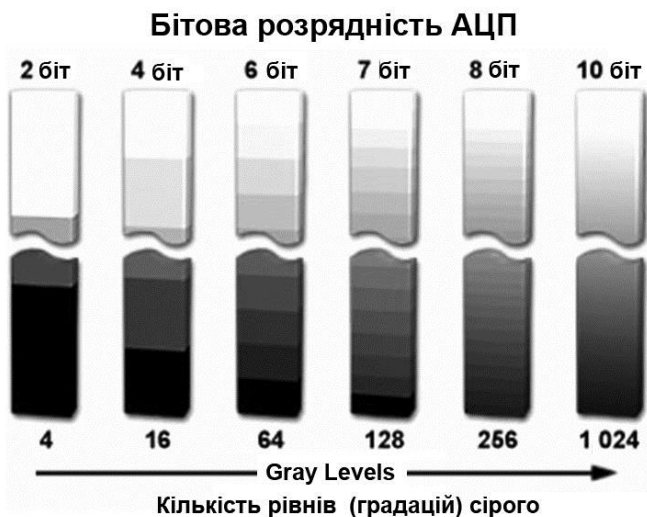


Рисунок 1.7 - Співвідношення бітової розрядності АЦП та кількості рівнів квантування (для сірого кольору)

Результат квантування зображення при зміні бітової розрядності показано на рис.1.8: видно, що при однаковому просторовому розрізненні зменшення рівнів квантування яскравості призводить до значної втрати детальності та інформативності зображення. На зображенні можуть виявлятися спотворення - виникають несправжні контури на межі переходу від одного рівня сірого до іншого або втрачаються справжні контури зображення.

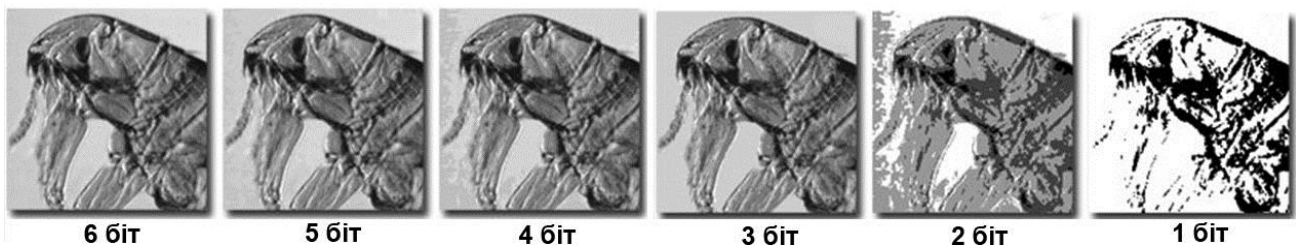


Рисунок 1.8 – Вигляд зображення біооб'єкту при зміні бітової розрядності кольору

Розподіл яскравості пікселів, що формують цифрове зображення, може бути графічно представлений у вигляді гістограми рівнів яскравості кольору. **Гістограма** має вигляд стовпчикової діаграми, де кожен стовпчик відображає кількість пікселів певного рівня яскравості, позначеного на осі абсцис (рис.1.9). Гістограма відображає діапазон та розподіл яскравості для зображення в цілому. Крім того, за даними гістограми можна визначити розміри деталей зображення за яскравістю їх пікселів. Найбільш часто гістограма застосовується для аналізу та зміни контрастності зображення. **Контрастність** – це ступінь розрізнення ділянок зображення при порівнянні їх за яскравістю. Величина контрастності визначається як інтервал градацій між найбільш темною та найбільш світлою ділянками зображення. Малоконтрастне

зображення має невеликий діапазон градацій, отже звужену, локальну гістограму (рис.1.9 б), тоді як більш контрастні зображення мають гістограму у всьому діапазоні градацій (рис.1.9 а,в).

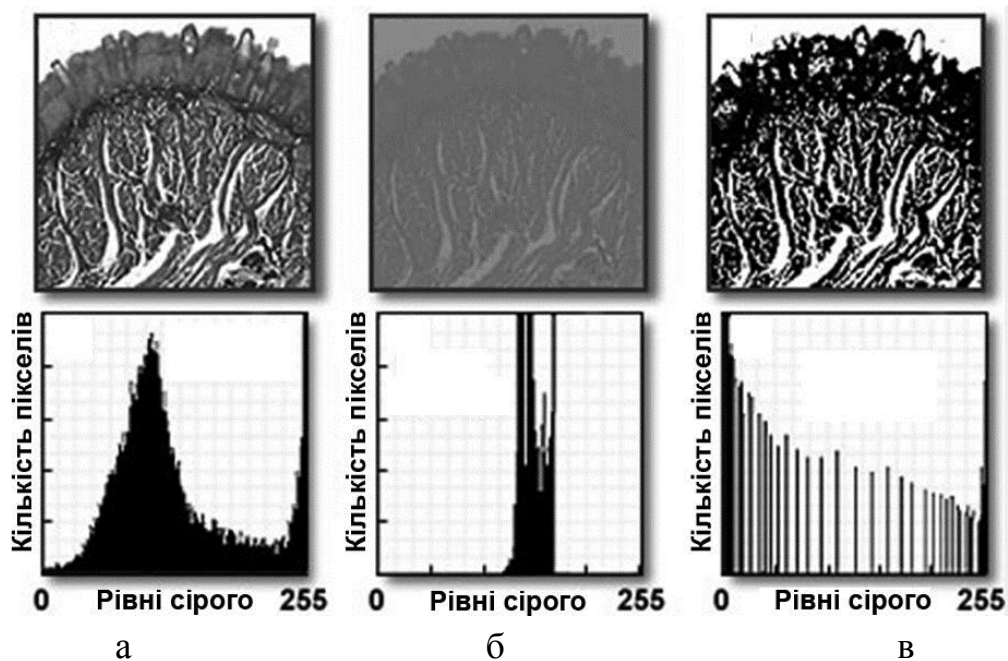


Рисунок 1.9 – Гістограми зображень біопрепарату різної контрастності

Третій параметр зображення, який впливає на розрізнення деталей цифрового зображення, є його **колірна складова**. Колір може бути представлений трьома параметрами:

- тон кольору (*hue*);
- насиченість кольору (*saturation*);
- яскравість (*brightness*), інтенсивність (*intensity*) або світлота (*lightness*).

При зміні **тону** колір плавно змінюється в межах спектру – від фіолетового до червоного. При зміні **насиченості** (колірності) колір плавно змінюється від ахроматичного (0% кольору, тобто колір змінюється на градацію сірого) до максимального значення (100% насиченості даного кольору). При зміні **яскравості** (світлоти) будь-який колір змінюється від найтемнішого (0% - чорний) до найсвітлішого (100% - білий).

Цифрові зображення мають наступні режими відображення кольору: **бінарний** (*Bitmap*), **напівтоновий** (*Grayscale*), **палітровий** (*Indexed Color*), **повноколірний** (*Full Color*). Вибір режиму відображення кольору в кожному конкретному випадку залежить від того, наскільки повно та без втрат має бути представлена візуальна інформація при заданій глибині кольору. Чим

складніше режим відображення кольору, тим більший обсяг пам'яті та обчислювальні можливості необхідні для обробки зображення.

У бінарному режимі (*Bitmap mode*) для позначення кольору пікселів застосовуються лише два значення - «0» (чорний) та «1» (білий). Тому бінарні зображення ще називають однобітними (глибина кольору - 1 біт/піксель) (рис.1.10).

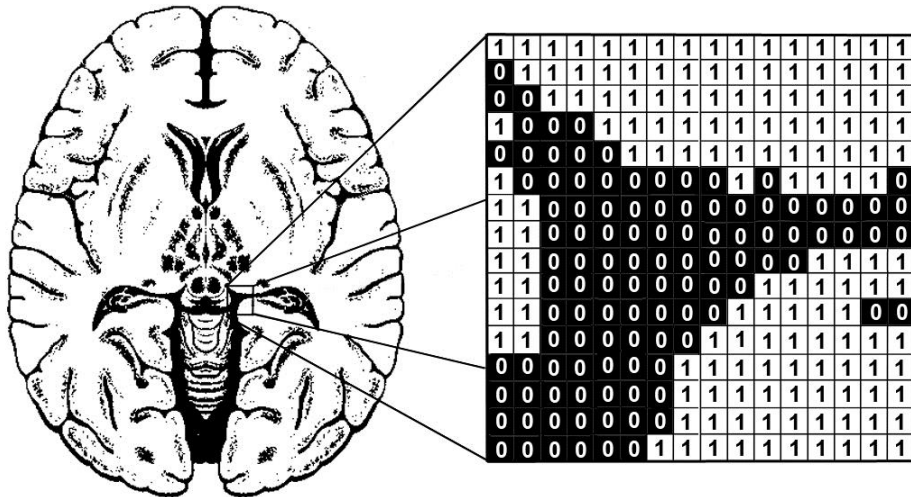


Рисунок 1.10 –
Фрагмент матриці
бінарного
зображення

Якщо задача потребує формування цифрового зображення, яке має містити лише оконтурені області та лінії, обирається саме бінарний режим. Зазвичай бінарне зображення можна отримати при обробці повноколірних, палітрових або напівтонових зображень методами бінаризації з пороговим обмеженням яскравості та виключенням даних про тон та насиченість кольору. При цьому надлишкова інформація про об'єкт відкидається, натомість чітко виявляються ті його морфологічні особливості, які становлять інтерес при конкретному дослідженні (рис.1.11). Бінарні зображення є найзручнішими для передачі даних, бо потребують найменшого обсягу пам'яті та затрат часу на обробку.

У напівтоновому режимі (*Halftone mode*) піксели набувають різних відтінків одного кольору (наприклад, сірого *Grayscale*). Діапазон градацій цього кольору визначається виходячи з розрядності цифрового коду (8, 16 або 32 біт). На практиці найчастіше застосовуються 8-бітні напівтонові зображення, які містять піксели з градаціями сірого у межах «00000000» (чорний) та «11111111» (білий). Весь діапазон градацій складає 256 рівнів сірого (глибина кольору – 8 біт на піксел). На рис.1.12 показано фрагмент 8-бітного напівтонового зображення та значення градацій сірого для пікселів вибраного фрагменту матриці. Більшість діагностичних зображень, отриманих за допомогою інтроскопічних систем, формуються саме у напівтоновому режимі.

Напівтонові зображення займають відносно невеликий обсяг пам'яті та ефективно застосовуються з метою отримання різноманітних діагностичних параметрів.

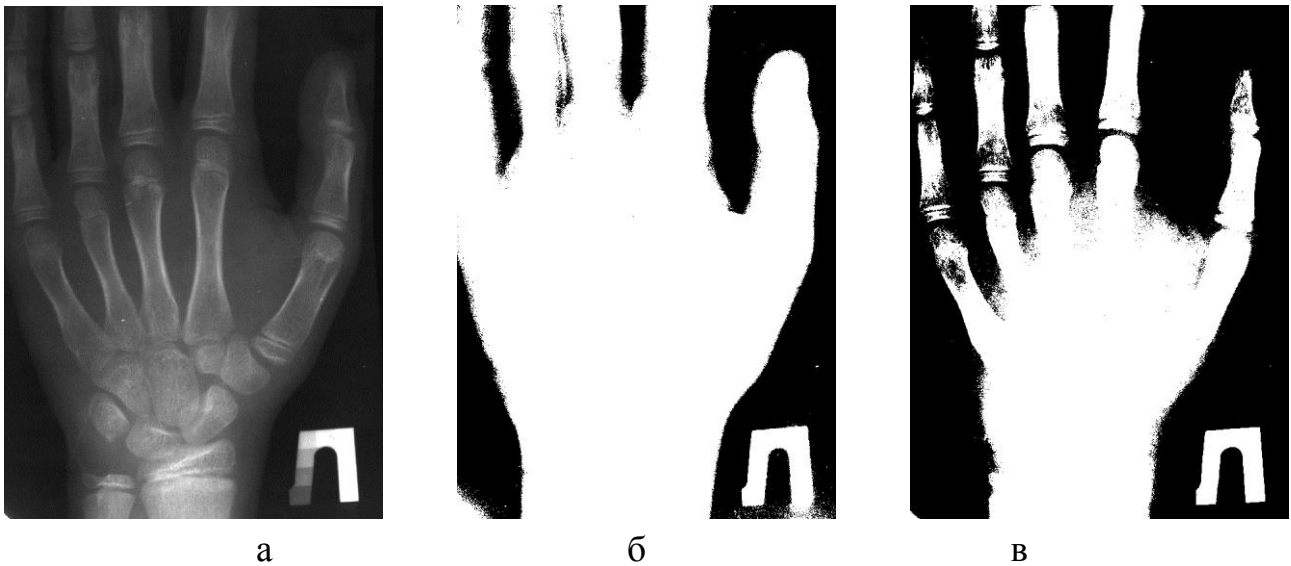


Рисунок 1.11 – Бінаризація напівтонового зображення:

а – початкове зображення – рентгенограма кисті (рівні яскравості 0...255);

б – виділення зображення кисті (пороговий рівень яскравості 32);

в – виділення зображень кісток пальців (пороговий рівень яскравості 48)

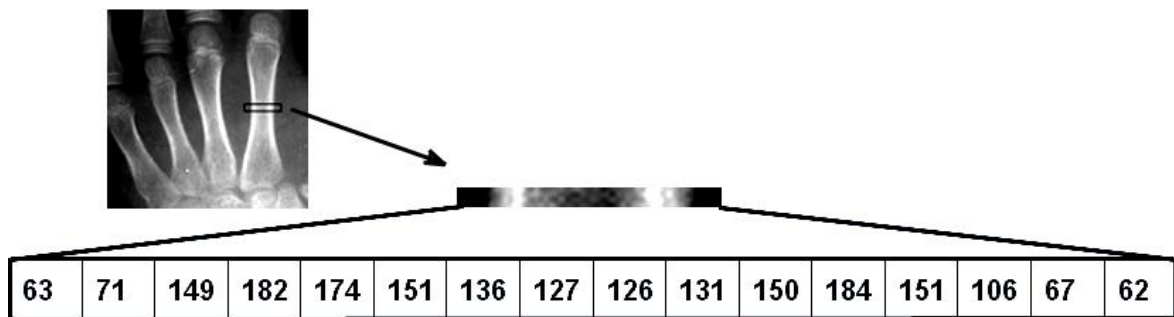


Рисунок 1.12 - Фрагмент матриці 8-бітного напівтонового зображення

У палітровому режимі (*Indexed Color*) формуються такі кольорові зображення, де кожен піксел набуває значення, яке відповідає певному зразку кольору з обраної карти кольорів (палітри). Колір кожного зразку кодується 4- або 8-бітним числом. Отже, палітра може містити до 256 зразків кольору та зберігається разом із зображенням. Палітровий режим можна застосовувати для позначення певних зон інтересу на діагностичних зображеннях (наприклад, кровоносні судини, дихальні шляхи тощо) або для створення псевдокольорових

зображень на основі напівтонових (карти розподілу температури, накопичення радіонуклідів тощо). Палітровий режим дозволяє відобразити кольорове зображення без надлишкової інформації, що полегшує обробку, зберігання та передачу зображення.

У повноколірному режимі (*Full Color*) кожен піксел зображення безпосередньо (без звернення до палітри) містить інформацію про колірні складові. Кольорові зображення формуються подібно до напівтонових, але для цього застосовується вже не один, а декілька окремих колірних каналів. Кожний такий колірний компонент квантується окремо, потім отримані компоненти об'єднуються, а колір кожного пікселя кодується двійковим числом, яке містить інформацію про всі компоненти кольору.

Точність відображення кольору залежить від бітової глибини кольору, яка визначає кількість доступних відтінків. Якщо зображення формується трьома 8-бітними колірними складовими (кожна складова має $2^8=256$ градацій кольору), тоді загальна глибина кольору пікселя становить $3 \times 8=24$ біти, тобто кількість доступних кодових комбінацій складає 2^{24} або більше 16 млн кольорів. Для такого зображення запис даних про колір здійснюється у 24-розрядному двійковому коді, а при записі додаткової інформації таке зображення може досягти розрядності до 56 біт/піксел.

В залежності від параметрів колірних складових кольорові цифрові зображення можуть формуватись за різними колірними моделями. **Колірна модель** – це засіб опису кольору. Найбільш поширені колірні моделі, що застосовуються для відтворення цифрових зображень, наведені у табл.1.2.

Таблиця 1.2 – Колірні моделі, їх складові та параметри

Назва моделі	Колірні складові	Розрядність, біт/піксел	Глибина кольору, біт	Застосування
RGB	Red – червоний Green – зелений Blue – синій	8, 16, 32	24, 48, 96	Генерація зображень
СМУК	Cyan – голубий Magenta – пурпурний Yellow – жовтий K – ключовий (чорний)	8	32	Друк зображень
Lab	Luminosity – яскравість a – від зеленого до червоного b – від синього до жовтого	8, 16	24, 48	Обробка зображень

Колірні моделі **HSI**, **HSL**, **HSB** (тон, насиченість, інтенсивність/світлота/яскравість) часто застосовуються при обробці кольорових зображень завдяки тому, що кольори надаються у формі, максимально адаптованій для зорового сприйняття. При роботі з цифровим зображенням кольори можуть визначатися наступним чином (табл.1.3):

- для моделі **RGB** вказуються числові значення градації кожної зі складових;
- для моделі **СМЯК** значення градацій змінюються у межах 0÷100%;
- для моделі **HSB** тон змінюється в межах 0÷360⁰, решта складових 0÷100 %;
- для моделі **Lab** яскравість в межах 0÷100%, складові **a** та **b** від -128 до +127.

Таблиця 1.3 – Опис основних кольорів в різних колірних моделях

колір	RGB			СМЯК				HSB			Lab		
	R	G	B	C,%	M,%	Y,%	K,%	H, ⁰	S,%	V,%	L,%	a	b
червоний	255	0	0	0	99	100	0	0	100	100	54	81	70
зелений	0	255	0	63	0	100	0	120	100	100	88	-79	81
синій	0	0	255	88	77	0	0	240	100	100	30	68	-112
білий	255	255	255	0	0	0	0	0	0	100	100	0	0
чорний	0	0	0	75	68	67	90	0	0	0	0	0	0

Гістограма кольорового зображення відображає комбінацію гістограм колірних складових, причому можуть бути побудовані гістограми для кожної зі складових у відповідності з колірною моделлю. На рис.1.13 показані гістограми RGB-зображення біопрепарату, підфарбованого у малиновий колір.

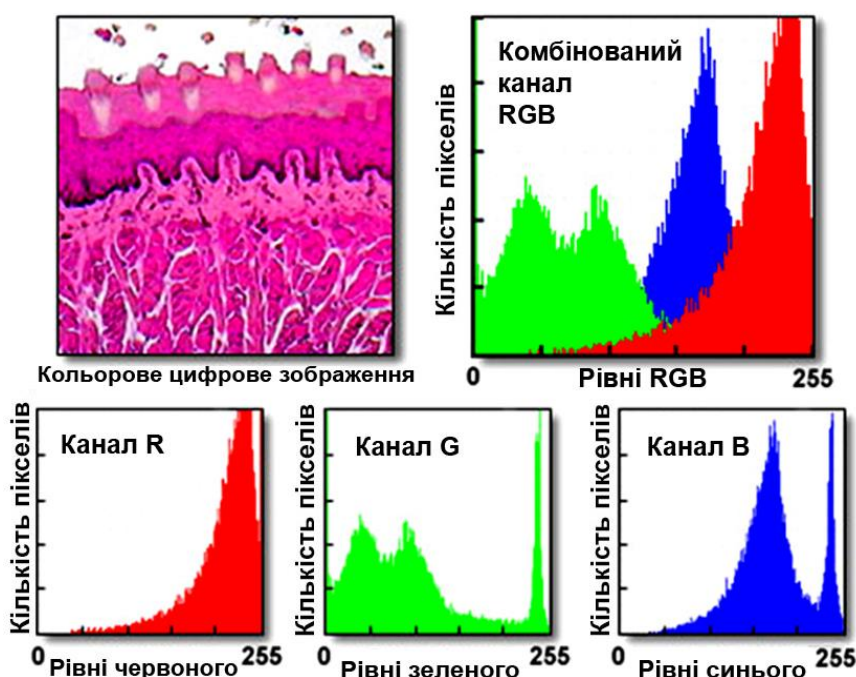


Рисунок 1.13 – Гістограми кольорового зображення біопрепарату

1.2.2 Зберігання цифрових зображень

Зберігання високодетальних діагностичних цифрових зображень потребує використання великих обсягів пам'яті комп'ютера. Тому важливо визначитись з оптимальним форматом і розміром графічного файлу для його подальшої програмної обробки. При зростанні кількості цифрових зображень, які потребують архівування, необхідно обрати раціональну та беззбиткову схему стиснення графічних файлів.

Для розрахунку мінімального обсягу графічного файлу в байтах необхідно кількість пікселів помножити на бітову розрядність та поділити це число на 8 (кількість біт в байті). Наприклад, для 8-бітного зображення у градаціях сірого розміром 800×600 пікселів мінімальний обсяг файлу становитиме 480 Кб.

Найчастіше цифрові медичні зображення записуються у графічних форматах BMP, TIFF, JPEG. В табл.1.4 показана залежність обсягу файлу цифрового зображення від розміру матриці пікселів, формату файлу та бітової розрядності кольору. Кожен з основних графічних форматів має своє призначення, переваги та обмеження (табл.1.5). При записі медичних цифрових зображень важливо якомога повніше зберегти початкову інформацію, для цього найкращим є формат TIFF. Формат BMP дозволяє найшвидше провести детальний програмний аналіз та обробку зображення, але потребує максимального обсягу пам'яті.

Таблиця 1.4 – Обсяги графічних файлів з різними параметрами зображення

Розмір, пікс	Обсяг пам'яті для запису зображень, Кб			
	Grayscale (8 біт)	BMP (24 біт)	JPEG (24 біт)	TIFF (24 біт)
256 × 256	66	198	22	199
512 × 512	262	786	52	787
800 × 600	480	1440	75	1441
1024 × 768	786	2358	104	2359
1600 × 1200	1920	5760	161	5761

Для створення ілюстративних матеріалів, остаточного запису зображення після обробки, його зберігання та передачі по мережі раціонально обрати файлові формати JPEG, PDF та ін., що дозволяє створювати електронні архіви та швидко отримувати дані для візуального аналізу.

Таблиця 1.5 – Формати графічних файлів та їх властивості

Формат	Повна назва	Основні риси	Колірні режими	Призначення	Обмеження
BMP	Bitmap	Стандартний Windows Device Independent	RGB Indexed Color Grayscale Bitmap	Обробка зображення без декодування, зберігання без втрат	Великий обсяг файлу
TIFF	Tagged-Image File Format	Універсальний cross-platform cross-application	Всі режими	Обробка зображення, поточне зберігання без втрат, друк	Великий обсяг файлу
JPEG	Joint Photographic Experts Group	Основний формат для цифрового фото	CMYK RGB Grayscale	Відображення та прикінцеве зберігання тональних зображень	Втрати при стисненні даних
PDF	Portable Document Format	Універсальний cross-platform cross-application	Всі режими	Відображення та прикінцеве зберігання, в т.ч. для www, друк	Неефективний для конкретних задач

При зберіганні великої кількості графічних файлів необхідно зменшити їх обсяг шляхом стиснення. При цьому можна застосовувати беззбиткові схеми стиснення, які не погіршують деталізацію та кольорову гаму зображення. Проте застосовується стиснення з втратою інформації, після якого поновити початковий зміст зображення неможливо. У табл.1.6 наведені параметри загальноживаних схем стиснення графічних файлів.

Таблиця 1.6 – Основні схеми стиснення графічних файлів

Схема	Повна назва	Підтримка форматів	Збереженість інформації	Ефективне застосування
RLE	Run Length Encoding	BMP 4-біт, 8-біт	Без втрат	Для бітових зображень
LZW	Lempel-Zif-Welch	TIFF, PDF, GIF	Без втрат	Для зображень з великими ділянками одного кольору
JPEG	Joint Photographic Experts Group	JPEG, TIFF, PDF	Втрати інформації	Для тонових зображень
ZIP		TIFF, PDF	Без втрат	Для зображень з великими ділянками одного кольору

1.3 Опис досліджуваних зображень

В роботі досліджуються цифрові зображення реальних біологічних об'єктів, медичні цифрові зображення, отримані за допомогою інтроскопічних систем, тестові зображення та графічні препарати, отримані за допомогою універсальних комп'ютерних програм для графічного редагування, а також довідкові анатомічні зображення.

1.4 Порядок виконання роботи

Робота виконується із застосуванням універсального графічного редактора Adobe Photoshop, який дозволяє створювати растрові зображення, отримувати та аналізувати графічні файли різних форматів, досліджувати та змінювати їх параметри, зберігати отриману інформацію. Головне вікно програми Adobe Photoshop наведено на рис.1.14.

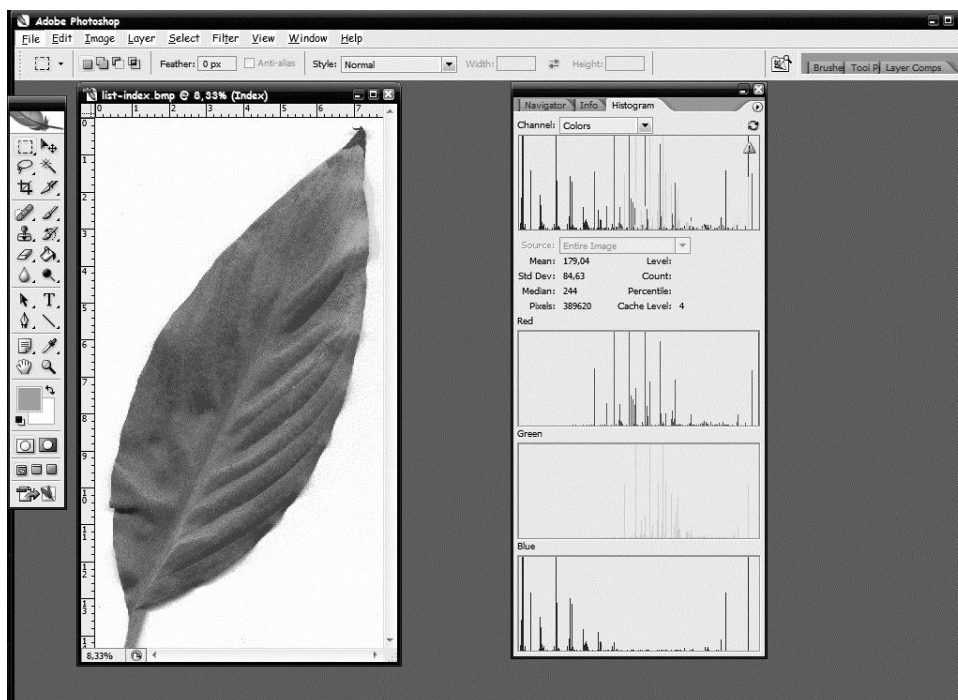


Рис.1.14 – Головне вікно програми Adobe Photoshop

1.4.1 Дослідження просторового розрізнення цифрового зображення

1.4.1.1 Дослідити тестові цифрові зображення **сітка.bmp** та **сектори.bmp** з метою виявлення можливості просторового розрізнення деталей різного розміру (рис.1.15). Визначити параметри файлів за допомогою команди **Image**→**Image size** (рис.1.16). На зображенні **сітка.bmp** накреслена сукупність ліній та проміжків різної ширини. Визначити її у пікселях.

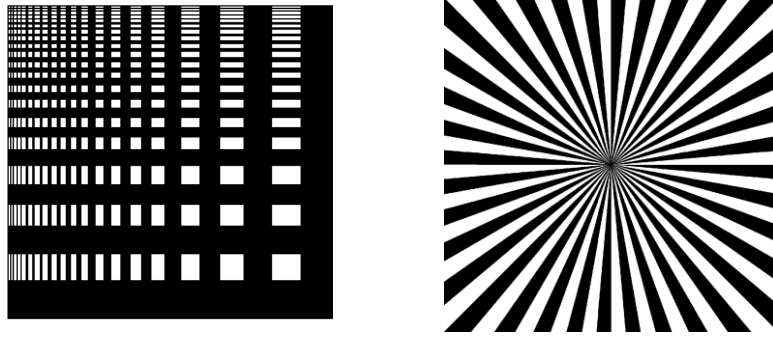


Рисунок 1.15 – Тестові зображення **сітка.bmp** та **сектори.bmp**

Проаналізувати розподіл пікселів зліва направо. Перші 50 значень записати як матрицю-рядок, де 0 - чорний та 1 - білий. Зменшуючи просторове зображення (36 dpi, 18 dpi, 9 dpi), записати три копії файлу та їх гістограми (команда **Window**→**Gistogram**). Визначити, які лінії зникли або змінили товщину при зменшенні просторового розрізнення. На зображенні **сектори.bmp** по колу зображені чорні та білі сектори з кутовим розміром 5° . Зменшуючи просторове зображення (36 dpi, 18 dpi, 9 dpi), записати копії файлу та їх гістограми. Виявити, при якому розрізненні з'являються спотворення зображення у вигляді муару та східців. Виміряти величину зони спотворень при розрізненні 36 dpi, 18 dpi, 9 dpi. Пояснити зв'язок між зміною просторового розрізнення та спотвореннями за допомогою критерію Найквіста-Котельникова. Прокоментувати зміну гістограм зображень.

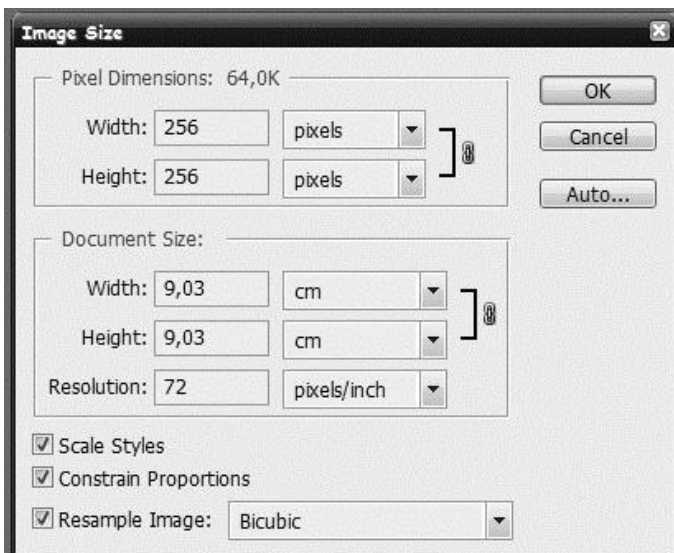


Рисунок 1.16 – Діалогове вікно визначення просторових параметрів зображення

1.4.1.2 Дослідити повноколірне RGB-зображення біооб'єкту, задане за варіантом (вибрати з папки **RGB-bmp**). Знижуючи просторове розрізнення,

виявити, які деталі біооб'єкту будуть втрачатись. Виявити граничне просторове розрізнення для даного зображення.

1.4.2 Дослідження квантування зображення за рівнями яскравості

Дослідити зображення морської зірки **зірка.jpeg** у градаціях сірого. За допомогою команди **Image→Adjustment→Posterize** (рис.1.17) зменшувати бітову розрядність сірого кольору з 8 біт до 1 біт (256, 128, 64, 32, 16, 8, 4 та 2 рівня сірого), кожного разу зберігаючи отримане зображення та його гістограму. Зробити висновки щодо зміни детальності відображення біооб'єкту.

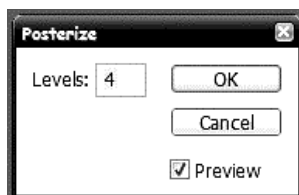


Рисунок 1.17 – Вікно вибору кількості рівнів кольору

1.4.3 Дослідження режимів відображення кольору зображення

Досліджується тестове зображення **RGB-сірі смуги.bmp**, створене у повноколірному режимі RGB (глибина кольору 8 біт/канал). На зображенні яскравість фрагментів змінюється від 0 до 100% з кроком 10% (рис.1.18).

Відкрити вікно відображення гістограм (**Window→Gistogram**) у режимі **All Channels View**. Підвести курсор до кожного зі стовпчиків гістограми та визначити рівень даної градації кольору, заповнити рядок табл.1.7 «RGB, од.».



Рисунок 1.18 – Тестове зображення **RGB-сірі смуги.bmp**

Змінити колірний режим тестового зображення з повноколірного на індексний за допомогою команди **Image→Mode→Indexed Color**. Зберегти отриманий файл як **Index-сірі смуги.bmp**. Визначити за гістограмою рівні градацій кольору тестового зображення у режимі **Indexed Color** та заповнити рядки табл.1.7 «Indexed,%, од.». Відкрити початкове зображення **RGB-сірі смуги.bmp** та змінити колірний режим з повноколірного на градації сірого командою **Image→Mode→Grayscale**. Зберегти отриманий файл як **Gray-сірі смуги.bmp**. За гістограмою визначити рівні градацій кольору тестового зображення у режимі **Grayscale** та заповнити рядки табл.1.7 «Grayscale,%, од.».

Таблиця 1.7 – Градації кольору тестового зображення **RGB-сірі смуги.bmp**

Режим		Яскравість фрагментів										
RGB	%	0	10	20	30	40	50	60	70	80	90	100
	од.	0										255
Indexed	%	0										
	од.	0										255
Grayscale	%	0										
	од.	0										255

Порівняти результати досліду та визначити різницю між градаціями яскравості відносно початкового зображення. За наявності такої різниці здійснити яскравісну корекцію зображення за допомогою команди **Image**→**Adjustments**→**Curves**. У діалоговому вікні **Curves** відображається графік, який показує співвідношення між вхідним (**Input**) та вихідним (**Output**) рівнями кольору (рис.1.19). За допомогою курсору знайти нове положення точок графіку, за яким розподіл градацій на перетвореному зображенні повернеться до заданих опорних значень (див.табл.1.7, рядок «RGB, од.»). Після натискання кнопки **ОК** зафіксувати новий розподіл градацій. Зберегти отримане зображення під новим ім'ям та, проаналізувавши його гістограму, пересвідчитись у правильності проведеної корекції.

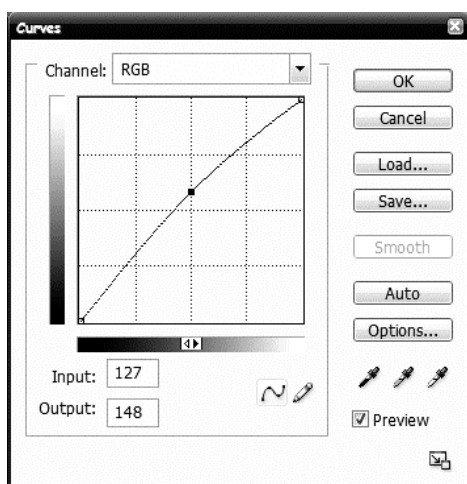



Рисунок 1.19 – Діалогове вікно **Curves** для корекції розподілу градацій

Всі отримані зображення та їх гістограми, а також дані табл.1.7 навести у звіті. Надати рекомендації щодо можливості застосування даних перетворень при роботі з біомедичними зображеннями.

1.4.4 Дослідження параметрів кольору зображення в різних колірних моделях та створення кольорового зображення

1.4.4.1 Визначення складових кольору у різних колірних моделях. Відкрити діалогове вікно вибору кольору **Color Picker**, двічі натиснувши кнопку вибору кольору  на панелі інструментів. У вікні **Color Picker** (рис.1.20) обрати модель **HSB** та вписати параметри складових кольору, заданого за варіантом (табл.1.8).

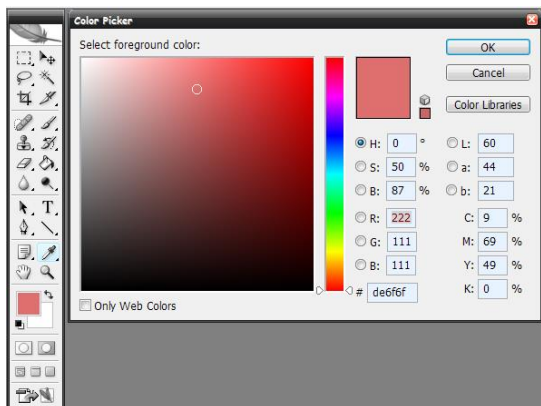


Рисунок 1.20 – Діалогове вікно вибору кольору **Color Picker** та панель інструментів Adobe Photoshop

До табл.1.8 записати значення складових даного кольору в інших колірних моделях (**RGB, CMYK, Lab**).

Таблиця 1.8 – Відповідність кольорів у різних колірних моделях

Варіант	HSB			RGB			CMYK				Lab		
	H, ⁰	S,%	B,%	R	G	B	C,%	M,%	Y,%	K,%	L,%	a	b
1	0	50	100										
2	60	50	100										
3	120	50	100										
4	180	50	100										
5	240	50	100										
6	300	50	100										
7	0	100	50										
8	60	100	50										
9	120	100	50										
10	180	100	50										

1.4.4.2 Колоризація зображення у відповідності з анатомічними даними

Створити кольорове зображення з наданого за варіантом напівтонового інтроскопічного зображення, записаного в режимі **Indexed Color**. Початкове

зображення являє собою перетин грудної клітини людини, побудований при рентгенівському томографічному дослідженні (рис.1.21). Ділянки, представлені певними градаціями сірого кольору на цьому зображенні, дають інформацію про відповідні анатомічні структури, що потрапили до зони томографічного сканування.

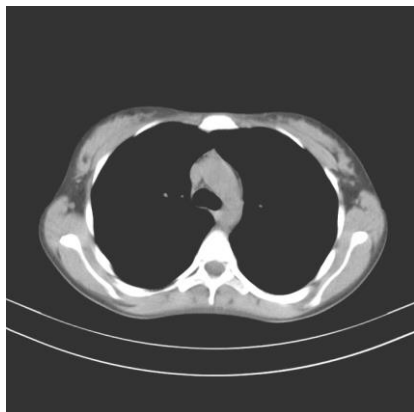


Рисунок 1.21 – Зображення перетину грудної клітини, створене при дослідженні на рентгенівському комп'ютерному томографі

Ознайомитись з кольоровими анатомічними зображеннями грудної клітини та визначити кольори, якими позначені основні шари тканин та органи (жирова тканина, м'язи, легені, кістки тощо). Співставити томографічне зображення з анатомічним та встановити відповідність зображених структур. Визначити, які градації сірого мають бути замінені на кольорові для позначення основних анатомічних структур на заданій томограмі. При розфарбовуванні **Grayscale-** або **Indexed Color-**зображення має бути переведене в колірний режим **RGB**. Далі на зображенні виділяються зони певної яскравості, до яких надалі застосовується операція зміни кольору (заливка, розфарбовування, зміна насиченості кольору).

Створити псевдокольорове томографічне зображення шляхом заливки виділених областей визначеними кольорами за допомогою інструментів:

- криволінійного виділення **Magic Wand Tool**  (виділення за градацією «чарівна паличка»), **Lasso Tool**  (виділення захватом «ласо»);
- визначення кольору **Eyedropper Tool**  (точкове визначення кольору «піпетка»), **Set Foreground Color**  (вибір кольору);
- нанесення кольору **Brush Tool**  (зафарбовування стилем «пензель») та **Paint Bucket Tool**  (заливка кольором виділеної ділянки).

Створити кольорове томографічне зображення шляхом зміни насиченості кольорів. Кожен з рівнів сірого може бути представлений як будь-який колір з нульовою насиченістю. Якщо скористатись інструментами для виділення певних областей зображення, подальшу його колоризацією можна здійснити шляхом виведення кольору на бажаний рівень насиченості. Для цього треба увійти в режим зміни балансу кольорів: **Image**→**Adjustment**→**Color Balance**. На екрані з'явиться модальне вікно **Color Balance** з регуляторами насиченості кольору (рис.1.22). В інтерактивному режимі обирається бажана комбінація колірних складових, а режим попереднього перегляду **Preview** дасть можливість спостерігати результат корекції кольору.

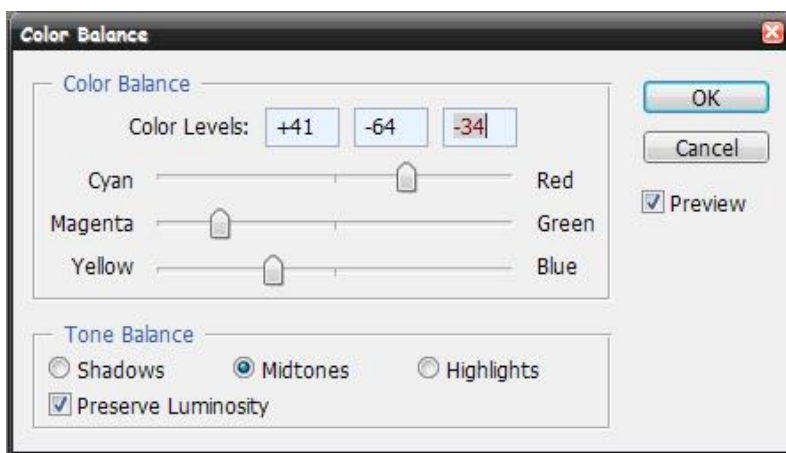


Рисунок 1.22 – Вікно зміни балансу колірних складових **Color Balance**

Анатомічно адекватна колоризація томографічних зображень полегшує їх аналіз при вивченні особливостей томографічної візуалізації організму людини.

1.4.5 Дослідження параметрів цифрового зображення при записі

Відкрити повноколірне RGB-зображення біооб'єкту, задане за варіантом. Створити та зберегти його копії у різних форматах (**BMP**→**TIFF** без стиснення, **BMP**→**JPEG** максимальної якості). Для кожного з форматів дослідити зміну обсягу файлу зображення в залежності від обраного колірного режиму (**RGB** →**Indexed Color**, **RGB** → **Grayscale**). Заповнити табл.1.9 та зробити висновки щодо виявлених особливостей запису файлу у різних форматах.

1.4.6 Дослідження параметрів цифрового зображення при стисненні

Скористатись отриманими за п.1.4.5 файлами та дослідити зміну їх обсягу при застосуванні різних схем стиснення. Для цього здійснити збереження початкового файлу даного формату із застосуванням всіх доступних опцій стиснення. Заповнити табл.1.10 та зробити висновки щодо переваг та обмежень кожної зі схем стиснення графічних файлів при зберіганні біомедичних зображень.

Таблиця 1.9 – Обсяг графічного файлу при записі в різних форматах та колірних режимах

Колірні режими	Обсяг файлу, Кб		
	BMP	TIFF	JPEG
RGB			
Indexed Color			
Grayscale			

Таблиця 1.10 – Зміна обсягу графічних файлів при застосуванні стиснення

Колірні режими	Обсяг файлу після стиснення, Кб							
	BMP	TIFF			JPEG			
	RLE	LZW	ZIP	JPEG	12	8	4	0
RGB								
Grayscale								

1.5 Зміст звіту

У звіті наводиться:

- назва роботи;
- мета роботи;
- зображення та їх гістограми, таблиці та текстові пояснення;
- висновки.

1.6 Контрольні запитання

1. Чим відрізняється цифрове зображення від аналогового?
2. Чим визначається просторове розрізнення цифрового зображення?
3. Як визначити граничну частоту дискретизації зображення?
4. Які спотворення зображення спостерігаються при субдискретизації?
5. Як відбувається квантування зображення за рівнями яскравості?
6. Яку інформацію містить гістограма цифрового зображення?
7. Якими параметрами представляється колір зображення?
9. Які основні режими відображення кольору в медичних зображеннях?
10. Що таке колірні моделі та якими складовими вони характеризуються?
11. Які особливості зберігання цифрових медичних зображень?
12. Які схеми стиснення застосовуються щодо медичних зображень?

ТЕМА 2. ПРИНЦИПИ ФУНКЦІОНУВАННЯ ПІДСИСТЕМИ АНАЛІЗУ ІНТРОСКОПІЧНОГО ЗОБРАЖЕННЯ

Підсистема аналізу зображення (ПАЗ) є інтелектуальною складовою інтроскопічної системи, яка реалізує регламентовані розробниками операції з перетворення отриманих діагностичних зображень. ПАЗ є комп'ютерним засобом, який втілює розроблені для даної інтроскопічної системи методи та алгоритми цифрової обробки зображень, насамперед, з метою їх об'єктивної діагностичної оцінки та отримання кількісних параметрів візуалізованих анатомічних об'єктів.

Кожна медична задача потребує розробки окремого алгоритму отримання діагностичної інформації, тому розробник програмного забезпечення має правильно обирати та застосовувати існуючі стандартні методи обробки зображень. Розробка ж нових методів потребує перевірки правильності результатів їх роботи шляхом застосування фантомних та тестових зображень із заздалегідь відомими параметрами.

Застосування загальновідомих універсальних програмних засобів з розробки методів та окремих процедур обробки зображень дає можливість здійснити апробацію нового, створюваного для даної інтроскопічної системи програмного засобу та перевірити правильність його роботи.

В цьому розділі вивчаються основні методи та алгоритми цифрової обробки зображень, реалізовані в спеціалізованій обчислювальній системі – матричній лабораторії MATLAB. Програмний пакет системи MATLAB для обробки зображень Image Processing Toolbox реалізує основні операції зі створення зображення, змін його параметрів, перетворення та зберігання зображення, яке має застосовуватись для подальшої діагностичної оцінки.

ЛАБОРАТОРНА РОБОТА №2

«Вивчення математичних операцій поелементного перетворення зображень»

2.1 Мета роботи

Вивчення елементарних операцій перетворення матриць та векторів, які застосовуються при роботі з масивами графічних даних, синтезі тестових зображень та аналізі реальних медичних зображень із застосуванням комп'ютерної математичної системи MATLAB.

2.2 Підготовка до виконання роботи

При підготовці до ЛР необхідно опрацювати теоретичний матеріал щодо способів формального опису дискретних зображень та їх перетворень із застосуванням апарату матричної алгебри. Виконанню роботи має передувати ознайомлення з синтаксисом основних операторів матричної алгебри, особливостями реалізації основних операцій над багатовимірними векторами та матрицями засобами спеціалізованої обчислювальної системи MATLAB. Необхідно ознайомитись з основними операціями з формування та перетворення масивів даних, представлених у вигляді багатовимірних числових матриць, які надалі застосовуються для відтворення цифрових зображень.

Матрична форма представлення дискретного зображення дозволяє ефективно застосовувати комп'ютерні засоби обробки даних для задач перетворення зображення. Перетворення може виконуватись над матрицею елементів зображення в цілому, над групою елементів або окремо над кожним елементом.

Застосування методу поелементної обробки масиву графічних даних дозволяє швидко та гнучко вирішувати досить багато прикладних задач покращення якості та аналізу зображень. Суть методу поелементної обробки полягає у наступному.

Хай $P_1(i, j) = P_{1_{i,j}}$ та $P_2(i, j) = P_{2_{i,j}}$ - значення яскравості зображення у точці (i, j) до обробки та після обробки відповідно. Поелементна обробка передбачає, що існує функціональна однозначна залежність між цими значеннями яскравості

$$P_{2_{i,j}} = f_{i,j}(P_{1_{i,j}}),$$

яка дозволяє скористатись відомим початковим значенням яскравості для розрахунку нового значення яскравості у даній точці зображення. Вид та параметри перетворюючої функції $f_{i,j}$ залежать від координат, тобто перетворення зображення може бути неоднорідним. Проте, в багатьох випадках застосовується однорідна поелементна обробка зображення, при якій до всіх

точок зображення застосовується однакове перетворення, тобто $P_2 = f(P_1)$.

До найбільш розповсюджених процедур поелементної обробки зображення відносяться лінійне контрастування зображення, порогова обробка, бінаризація, перетворення гістограм, геометричні перетворення – зсув, розтягнення/ушільнення, дзеркальне відбиття, проєктивні перетворення тощо. Ці процедури потребують змін значень яскравості точок на зображенні. Для цього необхідно організувати програмний доступ до кожної точки зображення як до елемента матриці пікселів шляхом зазначення індексів (i, j) - номерів рядка та стовпця. Надалі до кожної точки зображення застосовується дія перетворюючої функції $f_{i,j}$, яка математично реалізується у матричній формі.

2.2.1 Найпростіші операції над векторами та матрицями в MATLAB

2.2.1.1 Формування масиву, матриці, вектора. Набори чисел, які формують матрицю, називають **масивами**. Всьому масиву присвоюється одне ім'я, а доступ до окремих елементів масиву здійснюється за цілочисельним індексом (номером елемента в масиві). Масиви бувають одновимірними (для доступу до елемента використовується один індекс) або багатовимірними, зокрема двовимірними.

Для формування **одновимірного масиву**, який складається з кількох чисел, застосовується операція конкатенації. Результатом конкатенації двох об'єктів є новий об'єкт, отриманий шляхом почергового «підчеплення» другого об'єкту до першого. В системі MATLAB операція конкатенації позначається квадратними дужками — []. Наприклад, вираз

$$\mathbf{a1} = [1\ 2\ 3] \text{ або } \mathbf{a1} = [1, 2, 3]$$

формує змінну **a1**, яка є одновимірним масивом з трьох елементів (дійсних чисел 1, 2, 3). Для доступу до певного елемента одновимірного масиву потрібно після його імені вказати у круглих дужках індекс (номер) цього елемента. Наприклад, третій елемент масиву **a1** позначається як **a1(3)**. Якщо потрібно змінити значення певного елемента в сформованому масиві, можна застосувати операцію присвоєння, наприклад **a1(3)=7**.

Двовимірний масив зазвичай представляють як набір чисел, впорядкований у вигляді прямокутної таблиці, тоді для доступу до певного її елемента вказуються два індекси – номер **рядка** та номер **стовпця**, на перетині

яких знаходиться даний елемент. Двовимірний масив характеризується кількістю рядків та стовпців. Наприклад, складемо масив **a3**, у якому елементи розташовані у два стовпці та три рядки:

a3=[1 2; 3 4; 5 6]

a3=

1 2

3 4

5 6

При формуванні двовимірного масиву із застосуванням конкатенації для розділення рядків застосовується крапка з комою. Подібно до одновимірного масиву, двовимірний масив можна створити, індивідуально прописуючи його елементи з відповідною індексацією:

a3(1,1) = 1

a3(1,2) = 2

a3(2,1) = 3

a3(2,2) = 4

a3(3,1) = 5

a3(3,2) = 6

Вираз у круглих дужках (індекси – номер рядка та, через кому, номер стовпця) застосовується і для доступу до окремих елементів двовимірного масиву.

Будь-який рядок або стовпець матриці є одновимірними масивами, проте вони відрізняються способом упорядкування елементів: у рядку елементи впорядковані горизонтально, а в стовпці – вертикально. Тому масиви першого типу називають **вектор-рядками**, а другого типу – **вектор-стовпцями**.

Вектор-рядок можна розглядати як матрицю з кількістю рядків, рівною 1, а матрицю-стовпець – як матрицю з кількістю стовпців, рівною 1.

В MATLAB за допомогою операції конкатенації вектор-стовпець задається так:

a4=[1; 2; 3]

a4=

1

2

3

Точка з комою автоматично задає перехід на новий рядок, тому система автоматично розпізнає «геометрію» цього масиву та розташовує його елементи вертикально.

Загалом, при створенні матриць із застосуванням конкатенації розділювачі елементів кома (пробіл), або кома з крапкою визначають зовсім різні результати. В першому випадку здійснюється «горизонтальна» конкатенація («зчеплення» вздовж рядків), а в другому – «вертикальна» («зчеплення» вздовж стовпців).

Наприклад, операція «горизонтальної» конкатенації, здійснена над двома матрицями **A1** та **A2** з однаковою кількістю рядків, створює більш «широку» матрицю **C**

$$C = [A1 , A2] \quad ,$$

у якій до останнього елементу кожного рядка матриці **A1** «причіпляється» відповідний рядок матриці **A2**. При цьому кількість стовпців буде визначатись як сума кількості стовпців матриць **A1** та **A2**.

На відміну від попередньої, операція

$$C = [A1 ; A2]$$

з'єднує матриці **A1** та **A2** вертикально (до стовпчика з **A1** приєднується відповідний стовпчик з **A2**). В результуючій матриці **C** кількість рядків буде дорівнювати сумі кількості рядків матриць **A1** та **A2**.

При роботі з матрицями (особливо з метою відтворення зображення) може потребуватись визначення їх розмірів. Для визначення розмірів двовимірного масиву в обох напрямках в MATLAB застосовується функція **size**. Так, для наведеного вище масиву **a3** отримуємо:

```
size(a3)
ans =
     3     2
```

де у відповіді **ans** першим вказується число рядків, а другим – число стовпців.

Опис всіх масивів, які використовуються у поточному сеансі роботи (ім'я масиву, розмір, тип даних), можна отримати, застосовуючи оператор **whos**. Наведемо приклад отримання опису трьох масивів з різними параметрами:

whos			
Name	Size	Bytes	Class
a1	1×3	24	double array
a3	3×2	48	double array
a4	3×1	24	double array

Система MATLAB дозволяє не тільки задати масив у явному вигляді, але й автоматично сформувати його з чисел, які знаходяться у заданому діапазоні. При цьому вказується крок приросту для визначення проміжних значень. Для цього застосовується операція формування діапазону значень, яка позначається **двокрапкою (:)**. Наприклад, задано скласти одновимірний масив **diap1** з усіх дійсних чисел від 3,7 включно до 8,947 з приростом 0,3. Записуємо операцію як

diap1 = 3.7 : 0.3 : 8.947 .

При виконанні цієї дії всі елементи масиву **diap1** будуть вираховані та виведені на екран. Якщо очікується великий масив, який немає потреби переглядати, виведення всіх чисел на екран можна заздалегідь заблокувати, якщо в кінці наведеного виразу поставити кому з крапкою.

Найчастіше операція формування діапазону значень застосовується для формування діапазону цілих чисел:

diap2 = 4 : 2 : 26 .

Якщо приріст значення дорівнює одиниці, його можна не вказувати:

diap3 = 2:45 .

Операція «діапазон значень» може бути застосована і щодо індексації. Наприклад, необхідно отримати значення елементів матриці, для яких перший індекс відомий, а другий – змінюється у всьому доступному для нього діапазоні (отримані елементи належатимуть певному рядку матриці):

```

a = [ 1 2 ; 3 4 ; 5 6 ];
b = a(2,:)
b =
    3 4
  
```

Як результат отримаємо змінну **b**, яка позначає одновимірний масив, що містить всі елементи другого рядка двовимірного масиву **a**. Так простий вираз **a(2,:)** дозволяє вибрати з матриці **a** весь її другий рядок, наприклад, для подальших перетворень.

Одним з таких перетворень може бути видалення рядка (або стовпця), для чого, окрім операції «діапазон значень» застосовується поняття «порожній масив», який позначається квадратними дужками **[]**. Наступний вираз дозволить видалити з матриці весь другий рядок:

$$\mathbf{a}(2,:) = []$$

Операція «діапазон значень» дозволяє також вибірково виділити з масиву певну послідовність елементів. Наприклад, з одновимірного масиву

$$\mathbf{a} = [1 \ 6 \ 3 \ 9 \ 2];$$

необхідно виділити елементи з другого по четвертий. Для цього запишемо індексний вираз

$$\mathbf{b} = \mathbf{a}(2:4)$$

який вказує на елементи з індексами у заданому діапазоні від 2 до 4. Цим ми виконуємо групову операцію, яка записує в змінну **b** послідовно другий, третій та четвертий елементи масиву **a**. Формується новий одновимірний масив **b**:

$$\mathbf{b} = [6 \ 3 \ 9]$$

Наведені вище операції дозволяють вирішити ряд більш складних операцій, наприклад, вибірка елементів з масиву в довільному порядку. Так, з даного масиву **a** можна сформувати одновимірний масив **d**, в якому першим елементом буде п'ятий елемент масиву **a**, другим – третій елемент масиву **a**, третім – перший елемент масиву **a**. Вся ця операція здійснюється за наступним виразом:

$$\mathbf{b} = \mathbf{a}([5 \ 3 \ 1])$$

Послідовність написання набору індексів визначає порядок вибірки елементів з масиву **a** для формування нового масиву **d**.

Аналогічно, скориставшись операцією індексації, можна здійснити перестановку елементів в матриці. Наприклад, треба поміняти місцями стовпці в матриці **a** розміром 3x3:

$$\mathbf{a} = [\mathbf{1\ 2\ 3}; \mathbf{4\ 5\ 6}; \mathbf{7\ 8\ 9}];$$

так, щоби останній стовпець став першим, другий – останнім, а перший – другим. Ось операція, яка вирішує цю задачу:

$$\mathbf{a} = \mathbf{a}(:, [\mathbf{3\ 1\ 2}]) .$$

Отримаємо змінену матрицю **a** :

$$\begin{array}{ccc} 3 & 1 & 2 \\ 6 & 4 & 5 \\ 9 & 7 & 8 \end{array} .$$

Подібні операції можна застосовувати не тільки для перестановки рядків або стовпців, але й видаляти частину з них, а інші – дублювати. Наприклад, з отриманої вище зміненої матриці **a** розміром 3x3, треба отримати матрицю **c** розміром 4x5. При цьому перший рядок матриці **a** не використовується, а другий та третій продубльовані у наступному порядку: **2 3 3 2** . Далі кожен з рядків «розтягується» за рахунок дублювання першого та другого стовпців з подальшою перестановкою стовпців у порядку **2 1 3 2 1** . Задача вирішується наступною операцією:

$$\mathbf{c} = \mathbf{a}([\mathbf{2\ 3\ 3\ 2}], [\mathbf{2\ 1\ 3\ 2\ 1}]) .$$

$$\begin{array}{ccccc} 4 & 6 & 5 & 4 & 6 \\ 7 & 9 & 8 & 7 & 9 \\ 7 & 9 & 8 & 7 & 9 \\ 4 & 6 & 5 & 4 & 6 \end{array} .$$

2.2.1.2 Взаємні перетворення векторів та матриць. У ряді випадків потрібно отримати матриці з вектор-стовпців та вектор-рядків та навпаки.

Розглянемо приклад. Створимо вектор-стовпець, застосовуючи операцію конкатенації

$$\mathbf{v} = [\mathbf{1} ; \mathbf{2}]$$

або операцію індексації

$$\mathbf{v}(1,1) = \mathbf{1};$$

$$\mathbf{v}(2,1) = \mathbf{2};$$

Тепер за допомогою виразу

$$\mathbf{v} = \mathbf{v}(:, [\mathbf{1} \ \mathbf{1}])$$

здійснюємо перетворення змінної \mathbf{v} з вектора-стовпця на матрицю 2×2 :

$$\begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$$

Тут можна зазначити, що матрицю отримано дублюванням початкового вектора-стовпця. Якщо ж необхідно продублювати вектор-стовпець не двічі, а \mathbf{N} разів, в нагоді стане функція **ones**, яка породжує масив будь-якої розмірності, причому всі його елементи мають значення одиниці. Так, наступний вираз

$$\mathbf{U} = \mathbf{ones}(10)$$

породжує одновимірний масив \mathbf{U} , який складається з 10 одиниць.

Застосуємо цю функцію для отримання з вектора-стовпця матриці 2×10 :

$$\mathbf{v} = \mathbf{v}(:, \mathbf{ones}(10))$$

в результаті вектор-стовпець \mathbf{v} «розмножиться», даючи 10 однакових стовпців.

Розглянемо ще деякі особливості операцій з перетворення матриць.

Наприклад, задані матриця \mathbf{A} розміром 3×2 та вектор-стовпець \mathbf{a} з 6 елементів:

$$\mathbf{A} = [\mathbf{4} \ \mathbf{2} ; \mathbf{1} \ \mathbf{6} ; \mathbf{5} \ \mathbf{3}],$$

$$\mathbf{a} = [\mathbf{1} \ \mathbf{7} \ \mathbf{2} \ \mathbf{4} \ \mathbf{8} \ \mathbf{1}]$$

Для матриці **A**:

- вираз **A(2,1)** вказує на один елемент (у другому рядку та першому стовпці);
- вираз **A(:,2)** вказує на весь другий стовпець;
- вираз **A(:, :)** вказує на елементи всієї матриці **A**, тобто це і є матриця **A**;
- вираз **A(:)** формує набір з усіх елементів матриці **A**, впорядкований лінійно по стовпцях (тобто вектор-стовпець): спочатку вибираються елементи першого стовпця, за ними – другого стовпця, третього, тощо. До речі, в пам'яті комп'ютера всі масиви зберігаються саме таким чином.

Тепер спробуємо здійснити над заданими масивами операцію **a(A(:))**. Тут елементи матриці **A** впорядковуються стовпець за стовпцем та мають виконувати роль індексів для входу в масив **a**. Саме у такій послідовності з матриці **a** вибираються її елементи та формується масив-результат:

```
a(A(:))  
ans =  
    [ 4 1 8 7 1 2 ] .
```

Можна задати ще одну операцію над матрицями, застосувавши вираз **a(A)**. Результат її нам поки невідомий. Система **MATLAB** дасть таку відповідь:

```
a(A)  
ans =  
  
    4    7  
    1    1  
    8    2 .
```

Видно, що з одновимірного масиву **a** отримана матриця 3×2 . **MATLAB** трактує такий вираз як породження двовимірного масиву тієї ж структури, що і матриця **A**, яка надає індекси для вибірки значень з одновимірного масиву **a**.

Таким чином, як індекси можна застосовувати не тільки лінійні діапазони, але й геометрично структуровані набори чисел.

2.2.1.3 Багатовимірні масиви в системі **MATLAB**

Багатовимірними називаються масиви з розмірністю більше двох. Для індексації елементів таких масивів потребується три чи більше номерів, які вказують на положення певного елемента вздовж декількох напрямків впорядкування. Наприклад, створимо тривимірний масив **D** розміром $2 \times 3 \times 2$, який містить самі одиниці:

D=ones(2,3,2);

Потім змінимо значення двох елементів, вказуючи їх індекси (рис.1.25):

D(2,2,1)=7; D(1,2,2)=5;

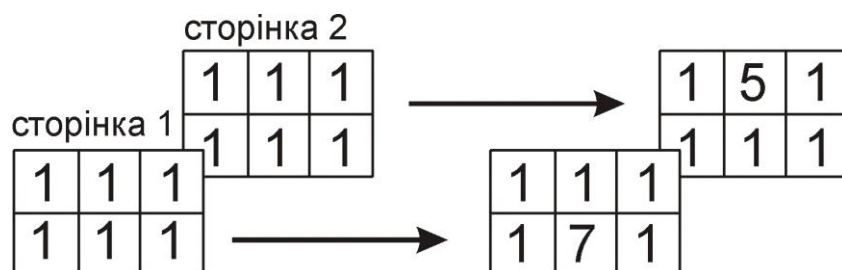


Рисунок 1.25 – Заміна значень елементів у тривимірній матриці 2×3×2

Вміст оновленого масиву проглянемо за допомогою таких операцій MATLAB:

D

D(:, :, 1) =

1 1 1
1 7 1

D(:, :, 2) =

1 5 1
1 1 1

Масив подається у вигляді послідовності матриць-сторінок. Зберігання цього масиву в пам'яті відбувається також по сторінках (стовпець за стовпцем). Якщо зважати на таку послідовність, можна звертатись до елементів масиву, вказуючи для елемента єдиний (поточний) індекс. Так, для заданого масиву **D** справедливі наступні рівності:

D(3)==1; ... D(4)==7; ... D(7)==1; ... D(9)=5; ...

Операції з масивами розмірності 4 та вище аналогічні до щойно розглянутих операцій з тривимірними масивами.

2.2.2 Функції для роботи з масивами

В системі MATLAB декілька функцій призначені для роботи з масивами. Одну з таких функцій **ones** (формування масиву з одиниць) ми вже розглянули раніше. До цієї ж групи функцій належать: функція **zeros** (формування масиву з нулів), функція **magic** (формування спеціальної квадратної матриці з цілих чисел, сума яких по рядках та стовпцях однакова).

Функція **rand** дозволяє згенерувати масив з випадкових елементів, значення яких формуються за рівномірним законом на інтервалі від **0** до **1**. Наприклад, сформуємо масив випадкових чисел розміром 3×3 :

A = rand(3)

A =

```
0.9501  0.4860  0.4565
0.2311  0.8913  0.0185
0.6068  0.7621  0.821  .
```

Якщо задати функцію **rand** з двома аргументами (наприклад **(2,3)**), отримаємо матрицю з випадкових чисел розміром 2×3:

B = rand(2,3)

B =

```
0.4447  0.7919  0.7382
0.6154  0.9218  0.1763  .
```

Функцію **rand** можна задати і з більшою кількістю скалярних аргументів – у цьому випадку будуть генеруватись багатовимірні масиви випадкових чисел.

При роботі з багатовимірними масивами операція конкатенації застосовується з деякими обмеженнями. В багатовимірних масивах конкатенацію за допомогою квадратних дужок не можна застосовувати для породження нового виміру в матрицях (за винятком переходу від розмірності 1 до розмірності 2). Для цього застосовують спеціальну функцію **cat**. Розглянемо приклад. Хай задані два двовимірні масиви (матриці) розміром 3×2:

A1 = [1 2 3 ; 4 5 6]

A2 = [7 8 9 ; 3 2 1]

Конкатенація цих матриць вздовж різних напрямків дає такі результати:

cat(1,A1,A2)

породжує масив [1 2 3 ; 4 5 6 ; 7 8 9 ; 3 2 1], тобто маємо той же результат, як від [A1 ; A2] – відбувається зчеплення матриць в стовпчик (розмірність 3×4);

cat(2,A1,A2)

породжує масив [1 2 3 7 8 9 ; 4 5 6 3 2 1], що збігається з результатом операції [A1 , A2] – виконується зчеплення матриць в рядок (розмірність 6×2).

A3 = cat(3,A1,A2)

породжує з двовимірних масивів A1 и A2 тривимірний масив A3 розміром 3×2×2. Результат цієї дії надається у наступному вигляді:

```
ans =  
1  2  3  7  8  9  
4  5  6  3  2  1  
A3(:,:,1) =  
    1    2    3  
    4    5    6  
A3(:,:,2) =  
    7    8    9  
    3    2    1
```

Багатовимірні масиви можна перетворювати інакше, застосовуючи функцію **reshape(A,m,n,p,...)**, яка створює багатовимірний масив розміром $m \times n \times p$, сформований з елементів масиву A. Конструкції **reshape(A,[m n p ...])** та **reshape(A,m,n,p,...)** рівносильні.

Розглянемо приклад. Задано перетворити матрицю A розміром 3×4 в іншу матрицю розміром 2×6 елементів, застосовуючи функцію **reshape** в різних формах.

```
A=[1 4 7 10;2 5 8 11;3 6 9 12 ]
```

```
B = reshape(A,2,6)
```

```
A =
```

```
1  4  7 10
2  5  8 11
3  6  9 12
```

```
B =
```

```
1  3  5  7  9 11
2  4  6  8 10 12
```

```
B = reshape(A,2,[])
```

```
B =
```

```
1  3  5  7  9 11
2  4  6  8 10 12
```

Існує ряд функцій, які дозволяють здійснювати прості розрахунки елементів масивів: **sum**, **prod**, **max**, **min** и **sort**.

Функцію **sum** найчастіше застосовують до одновимірних масивів (розрахунок суми всіх елементів). Для матриць ця функція підсумовує елементи в стовпцях. Наприклад:

```
B = [ 1 2 3 ; 4 5 6 ; 7 8 9 ]
```

```
sum(B)
```

```
[ 12 , 15, 18 ]
```

Для багатовимірних масивів ця функція дозволяє обрати напрямок підсумовування елементів.

Функція **prod** виконує операцію множення (принцип, аналогічний **sum**):

```
B = [ 1 2 3 ; 4 5 6 ; 7 8 9 ]
```

```
prod(B)
```

```
[ 28 , 80, 162 ]
```

Функції **max** та **min** призначені для пошуку максимального та мінімального елементу в масиві, причому в одновимірних масивах ці функції визначають єдиний екстремальний елемент, а в матрицях – породжують набір екстремальних елементів для кожного стовпця. Наприклад:

```
B = [ 1 2 3 ; 4 5 6 ; 7 8 9 ]  
max(B)  
[ 7, 8, 9 ]
```

2.2.3 Обчислення, які здійснюються над масивами

Масиви є найбільш розповсюдженими та цінними типами даних в системі MATLAB, тому важливо розглянути всі особливості обчислень з ними.

2.2.3.1 Арифметичні операції . Операції додавання та віднімання масивів здійснюються поелементно та позначаються традиційними символами. Наприклад, до масиву **A** треба додати масив **B**:

```
A = [ 1 1 1; 2 2 2; 3 3 3 ]; B = [ 0 0 0; 7 7 7; 1 2 3 ]  
A + B  
== [ 1 1 1 ; 9 9 9 ; 4 5 6 ];
```

Матриці, що додаються або віднімаються, повинні мати однакові розміри. Виключенням є операція додавання скаляру. В цьому випадку виконується генерація матриці того ж розміру, що і у матричного доданку, причому всі елементи згенерованої матриці однакові (дорівнюють скаляру). Наприклад:

```
A + 5  
== [ 6 6 6 ; 7 7 7 ; 8 8 8 ];
```

тобто перед виконанням по елементного матричного додавання генерується безіменна матриця [**5 5 5; 5 5 5; 5 5 5**], яка і додається до матриці **A**.

Операції множення, ділення, піднесення до степеня в системі MATLAB мають особливий формат. Традиційні символи «*» та «/» зарезервовані за операціями лінійної алгебри:

- символ ***** позначає перемноження матриць (а також вектор-рядків та вектор-стовпців) за відомим правилом «рядок на стовпець»;
- символ **/** , а також **** позначає операцію «розв'язання систем лінійних рівнянь».

Для позначення поелементних операцій застосовуються комбінації «крапка та традиційний символ». Так, поелементне перемноження масивів позначається `.*` та дає наступний результат:

$$\mathbf{A} .* \mathbf{B} \\ == [0 0 0 ; 14 14 14 ; 3 6 9];$$

Відповідно, вираз $\mathbf{A}.^2$ дозволяє піднести в квадрат кожен елемент матриці \mathbf{A} .

При поелементному множенні матриці на скаляр виконується множення кожного елемента матриці на скаляр:

$$\mathbf{A} .* 4 \\ == [4 4 4 ; 8 8 8 ; 12 12 12];$$

Для поелементного ділення застосовуються комбінації символів `./` та `.\`, які позначають «праве» та «ліве» по елементні ділення. Вираз $\mathbf{A}./\mathbf{B}$ спричиняє формування результату у вигляді матриці з елементами $\mathbf{A}(\mathbf{k},\mathbf{m})/\mathbf{B}(\mathbf{k},\mathbf{m})$, а вираз $\mathbf{A}.\mathbf{B}$ надає матрицю з елементами $\mathbf{B}(\mathbf{k},\mathbf{m})/\mathbf{A}(\mathbf{k},\mathbf{m})$.

Обидві матриці, задіяні в операції ділення повинні мати однакові розміри. У випадку операції ділення зі скаляром виконується генерація матриці з елементами, рівними скаляру, причому розміри згенерованої матриці розширюється до розмірів матриці-операнда. Наприклад, операція

$$6 ./ \mathbf{A} \\ == [6 6 6 ; 3 3 3 ; 2 2 2];$$

призводить до поелементного ділення скаляра **6** на кожен з елементів заданої матриці \mathbf{A} .

2.3 Порядок виконання роботи

Запустити програму MATLAB. Програмувати завдання в новому вікні «New» чи безпосередньо в головному вікні MATLAB. Завдання зберігати у вигляді **m**-файлу зі своїм ім'ям за допомогою опції **Save As**.

Запустити програму, записану в **m**-файлі командою «Run» (пункт меню «Debug»). Результати виконаних перетворень, представлених в головному вікні MATLAB, зберегти у вигляді текстового файлу в директорії, вказаній викладачем (**Inbox**).

Безпосереднє виконання запрограмованих операцій можна здійснити як в головному вікні MATLAB, так і в новому вікні «New». Для цього виділити за допомогою лівої кнопки «миші» створений фрагмент програми, а при натисканні правої кнопки вибрати команду «Evaluate Selection». Результати виконання програми з'являться в головному вікні MATLAB.

2.3.1 Робота з елементами масивів

Створити вектор-стовпець **a**, вектор-рядок **b** та два квадратних масиви **A** та **B** довільних цілих чисел, застосовуючи операцію конкатенації. Розмірність масивів задається викладачем.

Організувати програмний доступ до кожного з елементів масиву.

Змінити вміст масивів, замінюючи значення певних елементів.

Визначити параметри (геометрію, розмірність) створених масивів за допомогою операторів «size» та «whos».

Організувати «горизонтальну» та «вертикальну» конкатенації раніше створених масивів **A** та **B**. Перевірити, чи можливо об'єднати масиви з різною кількістю рядків або стовпців (неквадратні масиви)?

Сформувати діапазон чисел з вибраним кроком за допомогою операції формування діапазону чисел.

Застосувати операцію «діапазон» для захвату (виділення) останнього рядка однієї зі створених раніше матриць. Видалити вказаний рядок, застосувавши операцію «порожній масив». Поміняти місцями стовпці та рядки матриці у визначеному Вами порядку.

Зі сформованого раніше одновимірного масиву (вектор-рядок **b**) за допомогою операції «діапазон» виділити групу елементів, вказаних викладачем.

Для створеного раніше вектора-стовпця **a** та квадратної матриці **A** виконати операції **a(A(:))** та **a(A)**. Сформувати новий масив.

Результати перетворень вивести в головному вікні MATLAB та зберегти.

Проаналізувати проведені операції та отримані дані з точки зору поелементного перетворення матриць цифрових зображень.

2.3.2 Робота з багатовимірними масивами та функціями їх обробки

Створити два тривимірних масиви одиниць та нулів. Змінити значення деяких елементів цих масивів на довільні числа. Прочитати вміст змінених масивів.

За допомогою функцій **rand** та **magic** сформувати два масиви, розмірність яких задасть викладач. За допомогою функції **cat** здійснити конкатенацію

створених масивів вздовж різних напрямків. Результати перетворень вивести в головному вікні MATLAB та зберегти.

Для створених тривимірних матриць здійснити операції **sum**, **prod**, **max**, **min**, **fliplr**, **flipud**, **rot90**, **sort** (див.додаток А).

Створити багатовимірний масив цілих чисел розмірністю $m \times n$. За допомогою операції **reshape** перетворити його в масив розмірністю $n \times m$.

Проаналізувати проведені операції та отримані дані. Результати перетворень зберегти у текстовому файлі.

2.3.3 Виконання обчислень над масивами

Сформувати два різних масиви однакової розмірності. Виконати операції їх додавання, віднімання, множення, ділення та піднесення до степеня. Проаналізувати отриману інформацію. Дані зберегти у текстовому файлі.

2.4 Зміст звіту

У звіті наводиться:

- назва роботи;
- мета роботи;
- тексти завдань, програм та результати їх виконання;
- висновки.

2.5 Контрольні запитання

1. Як формуються вектор-стовпці, вектор-рядки та матриці у MATLAB?
2. Як здійснюється операція конкатенації з масивами чисел?
3. Для чого застосовується операція «діапазон»?
4. Які операції застосовуються для перетворення елементів масивів?
5. Як здійснюються арифметичні операції над матрицями та їх елементами?
6. Вкажіть синтаксис основних операторів перетворень елементів масивів

ЛАБОРАТОРНА РОБОТА №3

«Синтез та візуалізація тестових зображень у системі MATLAB»

3.1 Мета роботи

Вивчення особливостей представлення зображень у прикладному програмному пакеті MATLAB Image Processing Toolbox. Синтез тестових зображень для апробації методів і засобів реконструкції зображень в медичних комп'ютерних томографічних системах.

3.2 Підготовка до виконання роботи

При підготовці до ЛР необхідно ознайомитись з основними засобами машинної графіки у системі MATLAB, принципами формування, перетворення та зберігання зображень різного формату за допомогою пакету **Image Processing Toolbox**. Ознайомитись з координатними системами машинної графіки та особливостями обробки послідовностей діагностичних зображень.

3.2.1 Реалізація принципів машинної графіки у системі MATLAB

Сучасні інтегровані середовища для моделювання та виконання програм цифрової обробки зображень містять потужні засоби для візуалізації медичних зображень. Так, у систему MATLAB інтегровано пакет **Image Processing Toolbox (IPT)**, розроблений фірмою MathWorks. Цей пакет містить спеціалізовані програмні засоби для обробки зображень. Вони мають відкриту архітектуру та дозволяють організовувати взаємодію з апаратурою для цифрової обробки сигналів, а також підключати стандартні драйвери.

Система MATLAB та її пакет IPT є зручними інструментами для розробки, дослідження та моделювання методів та алгоритмів обробки зображень. Пакет IPT дозволяє вирішувати задачі обробки зображень двома шляхами. Перший полягає у самостійній програмній реалізації методів та алгоритмів, а другий – дозволяє моделювати розв'язання задачі за допомогою «готових» функцій, які реалізують найбільш відомі методи та алгоритми обробки зображень. Застосування цих стандартних функцій має значні переваги при вирішенні навчальних та дослідницьких задач, оскільки дозволяє гнучко змінювати всі параметри у програмах обробки зображень. Перед безпосереднім застосуванням стандартних функцій пакету IPT необхідно ретельно дослідити кожен з них та визначити, який саме метод та з якими параметрами реалізує задана функція.

3.2.2. Опис цифрових зображень в системі MATLAB

Цифрове зображення, яке є предметом вивчення та дослідження, в системі MATLAB представлено певним типом в залежності від режиму відображення кольору (табл.3.1).

Бінарне зображення запам'ятовується як логічний масив. За домовленістю змінна **BW** надалі буде позначати бінарне зображення.

Індексні зображення представлені масивом значень та палітрою зображення. Значення пікселів масиву є індексами палітри. За домовленістю змінна **X** буде застосовуватись як назва масиву індексного зображення.

Таблиця 3.1 – Типи зображень в MATLAB та їх опис

Тип зображення	Опис	
Бінарне (Binary)	Логічний масив, який містить тільки нулі та одиниці (зображується чорний та білий колір відповідно). Також існують дворівневі зображення, утворені пікселями тільки двох рівнів інтенсивності, не обов'язково 0 та 1.	
Індексне (Indexed)	Масив в форматі logical , uint8 , uint16 , single або double , значення пікселів якого є індексами з палітри. Відомі як псевдокольорові зображення.	
	формат single або double	діапазон значень [1, p]
	формат logical , uint8 або uint16	діапазон значень [0, p-1]
Напівтонове (Grayscale)	Масив $m \times 3$ в форматі uint8 , uint16 , int16 , single або double . Значення пікселів описують значення інтенсивності (яскравості). Відомі як яскравісні зображення.	
	формат single або double	діапазон значень [0, 1]
	формат uint8 (1 байт)	діапазон значень [0,255]
	формат uint16 (2 байт)	діапазон [0, 65535]
	формат int16	діапазон [-32768, 32767].
Повноколірне (Truecolor)	Масив $m \times n \times 3$ в форматі uint8 , uint16 , single або double . Значення пікселя дорівнює значенню інтенсивності. Відомі як RGB -зображення.	
	формат single або double	діапазон значень [0, 1]
	формат uint8	діапазон значень [0,255]
	формат uint16	діапазон [0, 65535]

Палітра являє собою масив розмірністю $m \times 3$, представлений у форматі **double**, що містить значення з плаваючою точкою в діапазоні $[0,1]$. Кожен рядок палітри описує червону, зелену та синю складові певного кольору. Колір кожного пікселя визначається за відповідним значенням X як індексу палітри. Палітра, як правило, запам'ятовується разом з індексним зображенням та автоматично зчитується разом з ним при застосування функції **imread**. Після зчитування матриці зображення та палітри в робочій простір MATLAB система сприймає їх як окремі змінні, проте між ними існує чіткий взаємозв'язок, який визначається відповідно до формату матриці зображення (див.табл.3.1). Якщо матриця зображення надається у форматі **single** або **double**, то вона містить значення від 1 до p , де p – довжина палітри (значення 1 відповідає першому рядку в палітрі, 2 – другому і т.д.). Якщо матриця зображення представлена в форматі **uint8** або **uint16**, перший рядок палітри задає значення 0. На рис.3.1 показана структура індексного зображення.

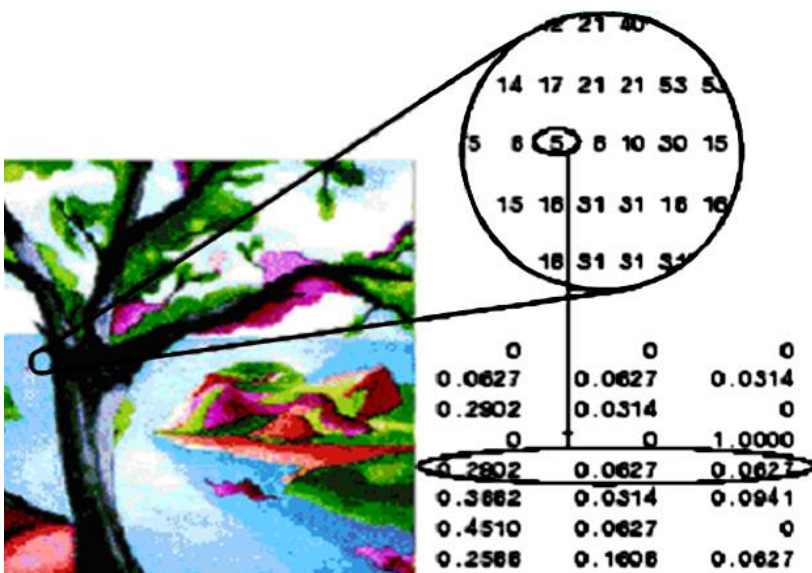


Рисунок 3.1 – Матриця пікселів індексного зображення та співвідношення значень її пікселів з палітрою

Матриця зображення надається разом з палітрою у форматі **double**, де 5 – значення пікселя (індекс кольору) є посиланням на п'ятий рядок палітри (колір зі складовими $R=0.2802$, $G=0.0627$, $B=0.067$). В системі MATLAB існує ряд спеціальних функцій для роботи з палітрою.

Напівтонові зображення представляються в вигляді матриці даних, значення елементів якої відповідають інтенсивності (яскравості) в деякому діапазоні. Система MATLAB запам'ятовує напівтонове зображення в вигляді матриці, кожний елемент якої відповідає одному пікселю зображення. За домовленістю напівтонові зображення будемо позначати змінною **I**. Матриця

даних може записуватись в форматах **uint8**, **uint16**, **int16**, **single** або **double** (див.табл.3.1). Для матриць, представлених в форматі **single** або **double**, застосовується напівтонова палітра, де інтенсивність 0 відображається чорним кольором, а інтенсивність 1 – білим. Для матриць, представлених в форматі **uint8**, **uint16** або **int16** значення інтенсивності **intmin(class(I))** відповідає чорному кольору, а значення **intmax(class(I))** – білому кольору. На рис.3.2 показано напівтонове зображення, записане в форматі **double**.

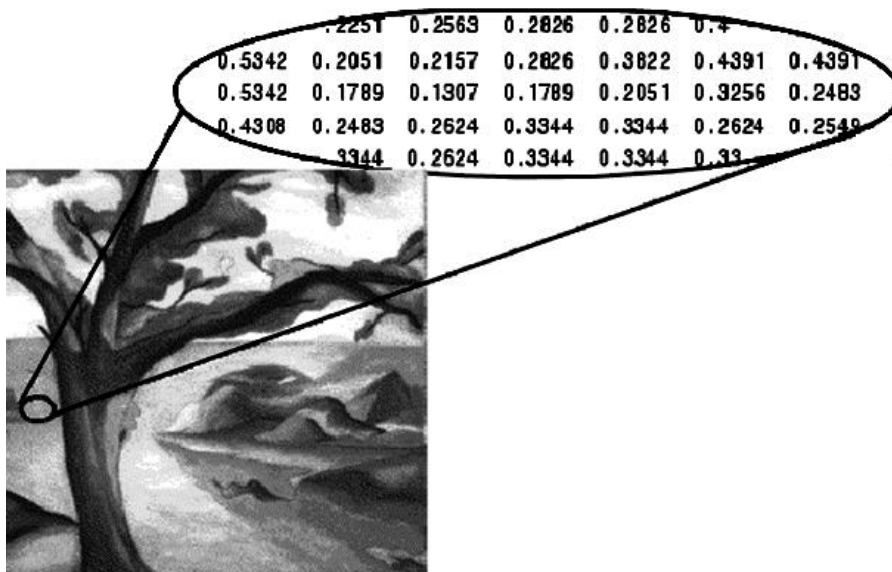


Рисунок 3.2 –
Фрагмент матриці
пікселів
напівтонового
зображення

Повноколірні зображення в системі MATLAB запам'ятовуються в вигляді масиву даних розмірності $m \times n \times 3$. В цьому масиві зберігаються всі три компоненти кольору (R,G,B) для кожного пікселя. Таким чином, повноколірні зображення не використовують палітру. Повноколірні зображення можуть бути представлені в форматі **uint8**, **uint16**, **single** або **double**. В форматі **single** або **double** значення кожної колірної компоненти задається числом в діапазоні від 0 до 1. Так, піксель з колірними компонентами (0,0,0) матиме чорний колір, а з компонентами (1,1,1) – білий. Значення «чистих» колірних складових будуть записуватись у наступних позиціях: червона – (x, y, 1), зелена – (x, y, 2), синя – (x, y, 3), тобто значення кожної колірної компоненти розташовується «вздовж» одного з трьох напрямків масиву даних. Наприклад, червона, зелена та синя складові пікселя (10,5) запам'ятовуються відповідно як **RGB(10,5,1)**, **RGB(10,5,2)** и **RGB(10,5,3)**. На рис.3.3 зображено фрагмент тривимірного масиву чисел у форматі **double**, який задає кольори пікселів RGB-зображення. В системі MATLAB визначення кольору пікселя, наприклад, з координатами (2,3), означає знаходження всіх трьох колірних складових, які записані в

позиціях (2,3,1:3). Для прикладу, червона складова цього пікселя (2,3,1) міститиме значення 0.5176, зелена (2,3,2) – значення 0.1608 та синя – значення 0.0627. Тоді «змішаний» колір пікселя (2, 3) може бути записаний набором чисел 0.5176 0.1608 0.0627 .

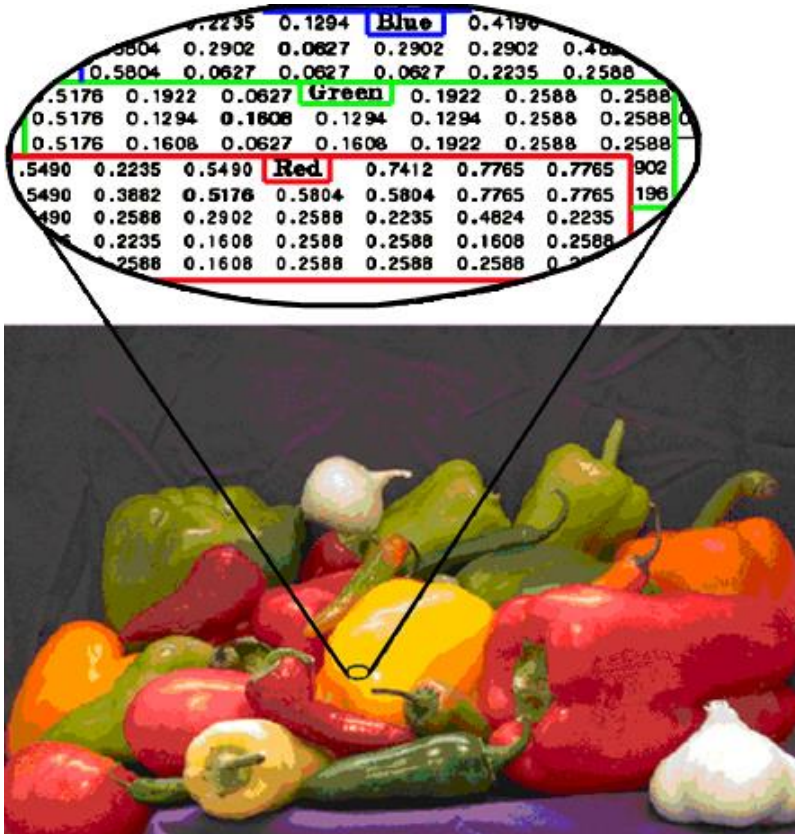


Рисунок 3.3 – Колірні складові повноколірного зображення

3.2.3 Перетворення типів зображень. При обробці цифрових зображень виникає необхідність зміни режиму їх відображення, що потребує перетворення їх типу. Наприклад, при здійсненні кольорової корекції індексного зображення попередньо необхідно перетворити його в повноколірне зображення. При обробці цього повноколірного зображення MATLAB буде здійснювати дії над значеннями інтенсивності зображення. Якщо ж попередньо не перевести індексне зображення в повноколірне, операції корекції будуть застосовані до індексів кольорів у матриці зображення, а не до справжніх значень кольорів. Це призведе до спотворення початкового зображення та отримання результату, який не піддається жодній інтерпретації. Перетворення зображення з одного формату в інший також можуть дати результат, відмінний від початкового. Так, перетворення індексного зображення в напівтонове призведе до повної втрати інформації про колір, а результуюче зображення сформується вже як напівтонове, з перерахунком значень пікселів в градації одного кольору.

В табл.3.2 наведені всі функції перетворення типів зображень, реалізовані в пакеті Image Processing Toolbox.

Таблиця 3.2 – Функції перетворення типів зображень в пакеті IPT

Функція	Опис
dither	Дифузійне псевдозмішування (dithering) для перетворення напівтонового зображення в бінарне або перетворення повноколірного зображення в індексне
gray2ind	Перетворення напівтонового зображення в індексне зображення
grayslice	Перетворення напівтонового зображення в індексне з використанням багаторівневої установки порогів
im2bw	Перетворення напівтонового, індексного або повноколірного зображення в бінарне зображення з використанням яскравісних порогів
ind2gray	Перетворення індексного зображення в напівтонове
ind2rgb	Перетворення індексного зображення в повноколірне
mat2gray	Перетворення матриці даних в напівтонове зображення з проведенням масштабування даних
rgb2gray	Перетворення повноколірного зображення в напівтонове
rgb2ind	Перетворення повноколірного зображення в індексне

Наведені вище перетворення виконуються у точній відповідності до синтаксису MATLAB. Наприклад, при перетворенні напівтонового зображення в повноколірне здійснюється з'єднання трьох копій початкової матриці $RGB = \text{cat}(3, I, I, I)$; вздовж трьох вимірів. Результуюче повноколірне зображення ідентифікує червону, зелену та синю складові матриці. Проте при візуалізації отримане зображення матиме вигляд напівтонового.

Відмітимо також, що при виводі в одному графічному вікні зображень різного типу з різними колірними картами не можна використовувати функцію **imshow**, тому що в цьому разі у всьому вікні встановлюватимуться атрибути кольору, задані при останньому виводі. Цього недоліку позбавлена функція **subimage**, яка спільно з функцією **subplot** дозволяє в одному вікні розміщувати зображення різного типу.

Наведемо кілька прикладів застосування функцій перетворення типів зображень (табл.3.2).

Функція **gray2ind** створює палітрове зображення **X** з палітрою **map**, перетворюючи початкове напівтонове зображення **I**. Палітра **map** містить **n** градацій сірого кольору, рівномірно розподілених в діапазоні [0, 1.0]. За замовчуванням **n** дорівнює 64.

Для створення палітри всередині функції **gray2ind** відбувається звертання до функції **gray(n)**. Якщо **n** менше або дорівнює 256, то отримане палітрове зображення **X** має формат представлення даних **uint8**, якщо **n** більше за 256 – формат даних визначається як **double**. В наступному прикладі напівтонове зображення **I** перетворюється на палітрове **X**, в якому палітра **map** містить 16 напівтонів:

```
clear;close all
I=imread('cameraman.tif');whos I
[X,map]=gray2ind(I,16);whos X
whos map
subplot(1,2,1);subimage(I);
subplot(1,2,2);subimage(X,map);
```

На перетвореному зображенні можуть бути помітними «плями» певної градації сірого кольору, що пов'язано з недостатньою кількістю рівнів сірого, заданою в прикладі.

За необхідності можна «відсікати» частину градацій кольору, встановлюючи декілька порогів яскравості за допомогою функції **X=grayslice(I,n)**, яка також перетворює напівтонове зображення на палітрове зі встановленням порогу відсікання **n**, що може дорівнювати $1/n, 2/n, \dots, (n-1)/n$:

```
clear;close all
I=imread('cameraman.tif');
X=grayslice(I,32);
subplot(1,2,1);subimage(I);
subplot(1,2,2);subimage(X,jet(32))
```

Функція **BW=im2bw(I,level)**, дозволяє отримати бінарне зображення шляхом перетворення напівтонового зображення з подальшим відсіканням градацій за вказаним порогом яскравості **level**, що регулюється в межах [0, 1.0]. Якщо яскравість пікселя менша за рівень **level**, вона набуває нульового значення (чорний колір), якщо більша за рівень – піксель стане білим.

Для бінаризації палітрового та повноколірного зображень застосовуються функції **BW=im2bw(X,map,level)** та **BW=im2bw(RGB,level)** відповідно. Наведемо приклад реалізації бінаризації па літрового зображення:

```
clear;close all
X=imread('circuit.tif'); BW=im2bw(X,0.2);
subplot(1,2,1);imshow(X);
subplot(1,2,2);imshow(BW)
```

Функція **I=ind2gray(X,map)** перетворює палітрове зображення на напівтонове, причому клас даних в перетвореному зображенні стане **double**:

```
clear;close all
[X,map]=imread('trees.tif'); I=ind2gray(X,map);
subplot(1,2,1);subimage(X,map);
subplot(1,2,2);subimage(I)
```

Функція **I=rgb2gray(RGB)** дозволяє перетворити повно колірне зображення на напівтонове. Вхідне зображення може бути представлене класами даних **uint8**, **uint16** або **double**, перетворене зображення **I** матиме той же клас, що і вхідне.

```
clear;close all
RGB=imread('autumn.tif'); I=rgb2gray(RGB);
subplot(1,2,1);subimage(RGB);
subplot(1,2,2);subimage(I)
```

Поряд з існуючими стандартними функціями перетворень існують також інші функції, які виконують перетворення типів зображень тільки в тій області, яка визначена користувачем. Наприклад, функція області інтересу повертає як результат бінарне зображення, яке застосовується як маска при фільтрації зображень або при інших операціях.

3.2.4 Перетворення колірних моделей

Пакет Image Processing Toolbox представляє колір як сукупність RGB-значень для повноколірних та індексних зображень. Функції пакету IPT дозволяють перетворювати значення кольору та переходити до інших колірних моделей (HSB, CMYK тощо). Функції обробки зображень працюють тільки з даними в форматі RGB, тому зображення, надане в іншій колірній моделі, має

бути переведене в RGB-формат, а після обробки його можна повернути в початковий колірний простір.

3.2.5 Перетворення класів зображень

Операції з перетворення зображень можуть супроводжуватись зміною розрядності двійкового числа (тобто кількості біт в коді даних), яким описані параметри кожного пікселя даного зображення. В системі MATLAB існує можливість перетворення даних зображення з формату **uint8** та **uint16** в формат **double** з використанням функції **double**. Однак перетворення класів призводить до змін у програмній інтерпретації даних зображення. Тому, щоб після перетворення класу дані сприймались як матриця зображення, необхідно здійснити деякі дії, наприклад, масштабування. Для реалізації перетворення класів зображень може бути застосована одна з вбудованих функцій програми **im2uint8**, **im2uint16**, **im2int16**, **im2single** або **im2double**. Ці функції автоматично масштабують та перетворюють початкові дані в зображення заданого класу. Наприклад, команда перетворення даних зображення RGB, представлених у форматі подвійної точності в діапазоні [0,1] в RGB- зображення класу **uint8** с діапазоном даних [0,255] має вигляд:

RGB2 = im2uint8(RGB1);

Слід відзначити, що при проведенні перетворень класів може статись втрата інформації про зображення. Наприклад, напівтонове зображення класу **uint16** може відображати 65536 різних відтінків сірого, а напівтонове зображення класу **uint8** відображає тільки 256 відтінків сірого. Тому при проведенні перетворень напівтонового зображення з класу **uint16** в клас **uint8** (тут можна застосувати функцію **im2uint8**) деякі дані про напівтони початкового зображення будуть втрачені. Втрати інформації будуть ще більшими при перетворенні напівтонового зображення з класу **uint16** в бінарний варіант. Однак, перетворення даних та втрати інформації, які їх супроводжують, бувають виправданими з огляду на ті цілі та задачі, заради яких вони виконуються.

3.2.6 Формати зображень в IPT

В пакеті IPT використовуються різноманітні формати зображень та файлів, що її представляють (табл.3.3). Ці формати можуть бути прочитані функцією **imread**.

Таблиця 3.3 – Формати зображення Image Processing Toolbox в MATLAB 6.5

Формат	Глибина кольору, біт	Особливості файлів зображень
'bmp'	1,4,8 або 24	Windows Bitmap – нестиснені файли, 4-bit та 8-bit – файли з RLE-стисненням
'cur'	1,4 або 8	Нестиснені файли
'hdf'	8 або 24	8-bit зображення з колірною картою та без неї, 24-bit зображення
'ico'	1, 4 або 8	Нестиснені файли
'jpeg'	8 або 24	Файли jpeg з різним ступенем стиснення
'pcx'	1, 8 або 24	Формат програми Paint Brush
'png'	1, 2, 4, 8, 16, 24 або 48	1-, 2-, 4-, 8- та 16-bit напівтонові зображення, 8- та 16-bit палітрові зображення, 24- та 48-bit повноколірні зображення
'tiff'	1, 8, 16, 24 або 48	1-, 8- та 24-bit нестиснені файли; 1-, 8-, 16- та 24-bit с packbits-стисненням; 1-bit зображення с CCITT-стисненням; 16- bit палітрові та 48- bit повноколірні
'xwd'	1 або 8	1-bit та 8-bit ZPixmap; XYBitmaps; 1-bit XYPixmap файлы

3.2.7 Робота з послідовностями зображень

В медичній практиці існують задачі, які потребують одночасної роботи з декількома зображеннями. Це можуть бути задачі обробки та аналізу томографічних зображень. Сукупність цих зображень називають послідовністю.

Послідовності зображень можуть зберігатись у вигляді N-вимірного масиву. Наприклад, масив розмірністю $m \times n \times r$ може зберігати r двовимірних зображень (напівтонових або бінарних). Це продемонстровано на рис.3.4. Масиви розмірністю $m \times n \times 3 \times r$ можуть зберігати повноколірні зображення, кожне з яких складається з трьох складових.

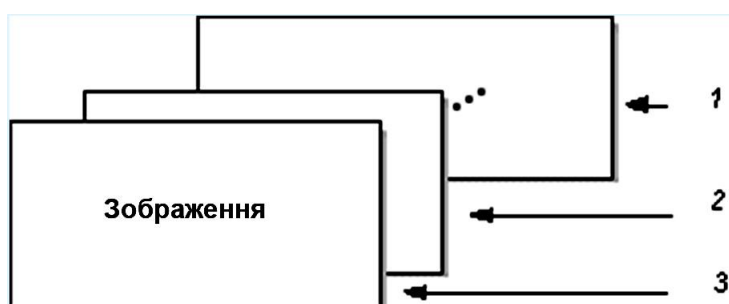


Рисунок 3.4 – Багатовимірний масив послідовності зображень

Багато функцій пакета можуть обробляти багатовимірні масиви та, відповідно, можуть обробляти і послідовності зображень. Наприклад, якщо багатовимірний масив обробити за допомогою функції **imtransform**, то обробка буде здійснена щодо всіх складових цього масиву. Результат буде аналогічний тому, якщо б кожна складова оброблювалася б окремо.

3.2.8 Представлення зображення в IPT

Основною структурною одиницею даних в MATLAB є масив. Для представлення зображення у вигляді масиву даних, його елементи мають містити інформацію про певні значення інтенсивностей кольору. MATLAB зберігає більшість зображень у вигляді двовимірного масиву (тобто матриці), в якому кожний елемент відповідає одному пікселю створеного зображення. Наприклад, якщо зображення складається з 200 рядків та 300 рядків, то воно в системі MATLAB запам'ятовується як матриця з розмірністю 200×300.

Деякі зображення, наприклад, повноколірні, представляються у вигляді тривимірних масивів, де записуються, відповідно, червона, зелена та синя складові. Таке представлення зображень спрощує їх обробку та дозволяє в повній мірі використовувати можливості системи MATLAB щодо обробки зображень.

3.2.8 Системи координат в IPT

В пакеті IPT зображення задаються в різних координатних системах. У більшості випадків різні методи описують розміщення пікселів на зображенні за допомогою піксельної системи координат, опис якої наведений у лабораторній роботі №1. Піксельна координатна система аналогічна до матричної координатної системи, яка використовується в MATLAB. Матричні індекси в MATLAB застосовуються для доступу до значень конкретних пікселів. Наприклад, код **I(2,15)** надає значення яскравості пікселя з 2-горядка та 15-го стовпця зображення **I**.

В піксельній координатній системі піксель є дискретною ділянкою, яка однозначно визначається парою координат, наприклад (5,2). Тоді позначення координат в ній виду (5.3,2.2) не має сенсу. Коли ж така форма позначення пікселя є необхідною, застосовується просторова система координат, де локалізація елементів зображення описується значеннями x и y (на відміну від значень r та c , як в піксельній системі координат). Змінна y збільшується донизу (рис.3.5).

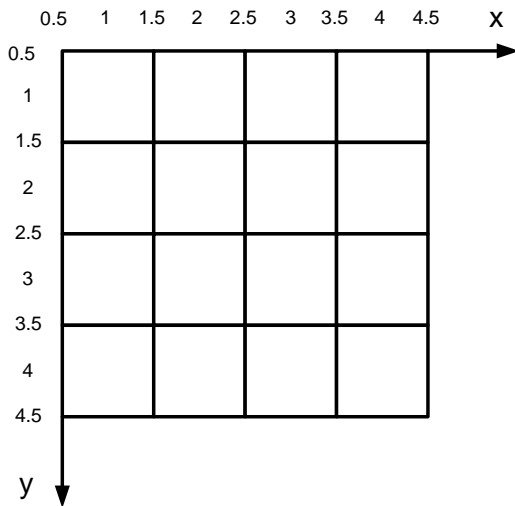


Рисунок 3.5 - Просторова система координат

Просторова система координат в деяких випадках відповідає піксельній координатній системі. Наприклад, в обох системах координат для всіх пікселів зображення центри пікселів співпадають. Однак є і різниця. Якщо в піксельних координатах верхній лівий кут зображення має координати (1,1), то в просторовій системі ці індекси дорівнюють (0.5,0.5). Просторова система координат є безперервною на відміну від дискретної піксельної системи.

За замовчуванням просторова система координат сприймається як піксельна. Наприклад, координати пікселя в 5 рядку та 3 стовпці мають просторові координати $x=3$, $y=5$ (відмітимо, що порядок координат при цьому зворотний). Деякі функції переважно працюють з просторовими координатами, інші – з піксельними. Іноді ж виникає потреба працювати і з нестандартними системами просторових координат. Для визначення просторової координатної системи потрібно описати дані зображення у вигляді **XData** та **Ydata**. Ці властивості будуть описані як двохелементний вектор та задаватимуть діапазон координат зображення, що відображається. За замовчуванням для зображення **A** дані **XData** представляються діапазоном $[1 \text{ size}(A,2)]$, а **YData** – діапазоном $[1 \text{ size}(A,1)]$. Значення цих векторів є координатами центрів пікселів. В дійсності ж діапазон координат, що відображаються, дещо більший. Якщо **XData** дорівнює $[1 \ 200]$, то діапазон вздовж осі x зображення насправді складає $[0.5 \ 200.5]$. Продемонструємо процедуру візуалізації зображення з використанням **XData** и **YData**:

```
A = magic(5);
x = [19.5 23.5];
y = [8.0 12.0];
image(A,'XData',x,'YData',y), axis image, colormap(jet(25))
```

3.2.9 Зчитування даних зображень з графічних файлів

Функція **imread** дозволяє зчитувати зображення з графічних файлів різних форматів та з різною глибиною кольору. Більшість форматів для запам'ятовування значень пікселів використовує 8 біт на піксель. Якщо ці дані зчитуються в пам'ять, то система MATLAB запам'ятовує їх в форматі **uint8**. Для файлових форматів PNG и TIFF, які підтримують 16-бітні дані, система MATLAB запам'ятовує зображення в форматі **uint16**. Наприклад, наведемо програмний код, який зчитує зображення в робочий простір MATLAB у форматі RGB:

```
RGB = imread('filename.format');
```

За допомогою наступного коду в робочий простір MATLAB зчитується індексне зображення з палітрою **colormap** в вигляді двох окремих змінних:

```
[X,map] = imread('filename.format');
```

В цьому прикладі функція **imread** при зчитуванні використовує той формат, який вказаний в назві файлу. Також формат можна вказати в функції **imread** як аргумент. Система MATLAB підтримує різні формати графічних файлів: BMP, GIF, JPEG, PNG, TIFF и др. Для отримання іншої інформації, наприклад, відносно глибини кольору, див. опис функцій **imread** та **imformats**.

3.2.10 Візуалізація зображень в пакеті IPT

Індексні зображення візуалізуються за допомогою функції **imshow** або **imtool**, які використовують опис матриці зображення та палітри. Тоді для представлення індексного зображення в робочому просторі застосовуються змінна з назвою **X** для представлення матриці індексного зображення та змінна **map** для представлення палітри:

```
imshow(X,map)
```

Для кожного пікселя в **X** функція застосовує при відображенні той колір, який записаний у відповідному рядку палітри **map**. Якщо матриця даних зображення представлена в форматі **double**, значення першого відліку відображається першому рядку в **colormap**, значення другого відліку – другим рядком тощо. Однак, якщо дані матриці зображень представлені в форматі **uint8** або **uint16**, то значення нульового відліку відображається першим рядком палітри, значення першого відліку – другим рядком і т.д. Це автоматично

визначається функцією **imshow**. Якщо кількість кольорів в палітрі більша, ніж на зображенні, тоді зайві кольори палітри функція ігнорує. Якщо палітра містить менше кольорів, ніж є на зображенні, функція встановлює для всіх пікселів зображення, значення яких вийшли за діапазон палітри, значення останнього кольору в палітрі. Наприклад, при візуалізації зображення, яке представлено в форматі **uint8** та містить 256 кольорів, застосовується палітра з 16 кольорів. Тоді всі пікселі, значення яких більші за 15, будуть відображатись останнім кольором палітри.

Напівтонові зображення візуалізуються за допомогою функції **imshow** з описом назви матриці зображення як аргументу. Далі будуть використовуватись різні імена (**I**) для представлення напівтонових зображень в робочому просторі **imshow(I)**. Ця функція відображає зображення з масштабованими значеннями інтенсивностей, представлених відповідними індексами в палітрі. Якщо параметр **I** представлений в форматі **double**, пікселі зі значенням 0.0 відображаються як чорні, пікселі зі значенням 1.0 – як білі, а пікселі з проміжними значеннями відображаються різними відтінками сірого. Якщо матриця зображення **I** представлена в форматі **uint8**, тоді пікселі зі значеннями 255 відображаються білими. Якщо матриця зображення **I** представлена в форматі **uint16**, тоді пікселі зі значеннями 65535 також відображаються білими.

Напівтонові зображення, як і індексні, які використовують палітру RGB, в більшості випадків не потребують опису палітри. Система MATLAB відображає напівтонові зображення за допомогою напівтонової палітрової системи, де $R=G=B$. За замовчуванням, число відліків сірого в палітрі дорівнює 256 для системи з представленням 24 біти на піксель.

В деяких випадках дані зображення можуть відображатись у вигляді напівтонових зображень, а їх діапазон візуалізації може відрізнитись від прийнятих в програмі діапазонів (тобто $[0,1]$ – для масивів звичайної або подвійної точності, $[0,255]$ - для масивів в форматі **uint8**, $[0,65535]$ – для масивів в форматі **uint16** або $[-32767,32768]$ для масивів в форматі **int16**. Наприклад, при фільтрації напівтонових зображень, значення даних можуть виходити за діапазон початкового зображення. Для відображення даних з нестандартним діапазоном у вигляді зображення, в функції **imshow** необхідно описати межі діапазону:

imshow(I,'DisplayRange',[low high])

При використанні порожньої матриці ([]) для відображення діапазону, функція масштабує дані автоматично, встановлюючи межі діапазону в відповідності з мінімальним та максимальним значеннями масиву.

Розглянемо приклад, коли в результаті фільтрації напівтонового зображення були отримані дані з нестандартним діапазоном. В прикладі для візуалізації такого зображення застосовується опція автоматичного масштабування:

```
I = imread('testpat1.tif');  
J = filter2([1 2;-1 -2],I);  
imshow(J,[]);
```

Бінарні зображення в системі MATLAB представляються в форматі **logical**. Пікселі зі значенням 0 відображаються як чорні, а пікселі зі значенням 1 – як білі. Для того, щоб програма інтерпретувала зображення як бінарне, необхідно, щоб це зображення було представлено в форматі **logical**. Для візуалізації бінарних зображень застосовується функція **imshow** з описом матриці зображення як аргументу. Розглянемо приклад зчитування бінарного зображення в робочий простір MATLAB з наступною візуалізацією зображення. Далі будемо використовувати назву **BW** для представлення бінарних зображень в робочому просторі:

```
BW = imread('circles.tif');  
imshow(BW)
```

Повноколірні зображення, які також називають RGB-зображеннями, представляють значення кольорів безпосередньо, без застосування палітри. Для кожного пікселя (r,c) на зображенні колір представляється триплетом ($r,c,1:3$). При візуалізації повноколірних зображень використовується функція **imshow** з описом матриці зображення як аргументу. Розглянемо приклад зчитування повноколірного зображення в робочий простір MATLAB та його наступної візуалізації. Далі назву **RGB** будемо використовувати для представлення повноколірних зображень в робочому просторі:

```
RGB = imread('flowers.tif');  
imshow(RGB)
```

Система, яка використовує представлення 24 біт на піксель, може відображати повноколірні зображення безпосередньо, тобто по 8 біт на запис червоної, зеленої та синьої складових відповідно. В системах з меншою кількістю кольорів, функція **imshow** відображає зображення за допомогою методу моделювання кольорів (**dithering**). Якщо трапилось, що при візуалізації кольорового зображення, воно відображається як чорно-біле, тоді необхідно перевірити, чи не є це зображення індексним. Для індексних зображень необхідно описати палітру та пов'язати її з зображенням.

3.2.11 Збереження зображень в IPT

Для запису створеного зображення на диск застосовується функція **imwrite** з розширеннями, що використовуються в MATLAB. Так, функція **imwrite(S, filename, fmt)** записує в файл с іменем **filename** бінарне, напівтонове або повноколірне зображення **S**. Функція **imwrite(X, map, filename, fmt)** записує в файл с іменем **filename** палітрове зображення **X** з палітрою **map**. Формат файлу визначається параметром **fmt**. Параметри **filename** та **fmt** є рядками. Якщо рядок **filename** не містить інформацію про путь, то функція **imwrite** записує файл в поточну папку.

При записі зображень в файли форматів TIFF, JPEG, HDF можна вказати ряд додаткових параметрів, які впливають на спосіб збереження зображень. Для цього в функції **imwrite** після параметрів **filename** та **fmt** передається одна або кілька пар параметрів **Parameter, Value**. Якщо запис здійснюється в JPEG-файли, то можна вказувати показник якості стисненого зображення. Для цього **Parameter** має бути рядком **'Quality'**, а **Value** – числом, яке визначає ступінь стиснення зображення. Цей показник може приймати значення в діапазоні [0, 100]. Чим менше значення цього показника, тим вище ступінь стиснення, але гірше якість зображення. При записі зображень в TIFF-файл можна використовувати наступні додаткові параметри (табл. 3.4). Якщо початкове зображення має формат представлення даних **double**, то перед записом в файл дані зображення автоматично перетворюються в формат **uint8**.

Приклад збереження зображення **'flowers.tif'** на диску E у вказаній папці в форматі **bmp** :

```
clear;close all
RGB = imread('flowers.tif');
imshow(RGB)
imwrite(RGB,'E:\Matlab-зображення до лр\flowers.bmp')
```

Таблиця 3.4 – Параметри запису зображення в TIFF-файл

Параметр	Можливі значення Value	Значення Value за замовчуванням
‘Compression’	‘none’ - без стиснення; ‘packbits’ - використовувати метод стиснення Packbits; ‘ccitt’ - використовувати метод стиснення CCITT	‘ccitt’ - для бінарних зображень; ‘packbits’ - для інших типів зображень
‘Description’	Будь-який рядок. Цей рядок знаходиться в полі ImageDescription структури, яку повертає iminfo	‘
‘Resolution’	Розрізнення в точках на дюйм	72

Записане зображення для перевірки належить переглянути на вказаному диску в заданому форматі.

3.3 Порядок виконання роботи

Запустити програму MATLAB. За допомогою функції **help images** ознайомитись з можливостями пакету **Image Processing Toolbox**. Проаналізувати інформацію, представлену в основному вікні MATLAB.

3.3.1 Синтез тестових зображень

Застосувати наведений нижче приклад m-файлу для синтезу кількох нових тестових напівтонових зображень (зображення в градаціях сірого), застосовуючи операції **конкатенації** та «**діапазон**». Ці зображення надалі будуть застосовуватись для інтегральних перетворень та реконструкцій:

```
close all % Закрити всі зображення (очистити екран)
I1=zeros(100,100);
I1(1:100,41:60)=1; % Формування 1-го тестового зображення
subplot(1,3,1);imshow(I1) % Вивід 1-го зображення в першому вікні з трьох
I2=ones(100,100);
I2(41:60,1:100)=0; % Формування 2-го тестового зображення
subplot(1,3,2);imshow(I2) % Вивід 2-го зображення в другому вікні з трьох
I3=zeros(100,100);
I3(20:80,20:80)=0.8;
```

```

I3(20:30,70:80)=0.5;
I3(20:30,20:30)=0.5;
I3(70:80,70:80)=0.5;
I3(70:80,20:30)=0.5;
I3(40:60,40:60)=0.3;
I3(51:69,51:69)=1;
I3(31:49,31:49)=1; % Формування 3-го тестового зображення
subplot(1,3,3);imshow(I3); % Вивід 3-го зображення в третьому вікні з трьох

```

Результати формування заданих тестових зображень показані на рис.3.6.

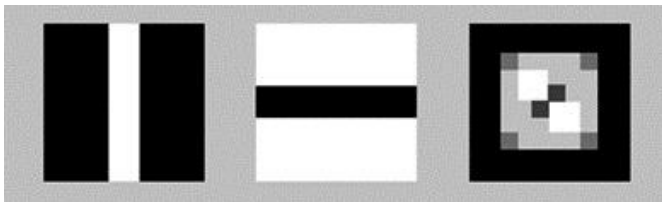


Рисунок 3.6 – Найпростіші тестові зображення

Створений m-файл зберегти під своїм ім'ям в своїй робочій папці. Тестові зображення зберегти за допомогою операції «Copy Figure» з вікна «Edit» у вигляді текстового файлу.

3.3.2 Формування тестового фантомного зображення зрізу голови людини

Для створення методів реконструкції зображень в MATLAB вбудовано програмний фантом зрізу голови людини у вигляді структури, побудованої з 10 еліпсоїдів, які моделюють мозок та його морфологічні особливості. Доступ до фантому здійснюється функцією

P=phantom(def,n),

де **n** – параметр, скаляр, що визначає число рядків и рядків фантома (якщо **n** – відсутній, за замовчуванням від дорівнює 256);

def – рядкова величина, яка визначає тип генерації фантома. Вона може приймати значення: ‘Shepp-Logan’ – тестове зображення, яке широко застосовується в рентгенівських томографічних дослідженнях, та ‘Modified Shepp-Logan’ (за замовчуванням) – варіант фантома з підвищеним контрастом для кращого візуального сприйняття.

За допомогою функції **P=phantom(E,n)** можна створити користувацький модифікований фантом, де кожний рядок матриці **E** визначає параметри еліпса на зображенні. В табл. 3.5 описані стовпці матриці, яка формує фантом голови.

Таблиця 3.5 – Параметри формування фантому голови людини

№ стовпця	Параметр	Призначення
1	A	Додавання значення до інтенсивності еліпса
2	a	Довжина горизонтальної півосі еліпса
3	B	Довжина вертикальної півосі еліпса
4	x0	x - координата центра еліпса
5	y0	y - координата центра еліпса
6	Phi	Кут (в градусах) між горизонтальною піввіссю еліпса та x – віссю зображення

Після виконання прикладу з формування фантома, виконати його модифікації згідно наведеної нижче програми. Результати перетворень (рис.3.7) та створені програми зберегти як m- та текстовий файли.

close all

P=phantom(512);

subplot(1,3,1);imshow(P)

```
E = [ 1 .69 .92 0 0 0
      -7 .6624 .8740 0 -.0184 0
      -2 .1100 .3100 .22 0 -18
      -2 .1600 .4100 -.22 0 18
      .1 .2100 .2500 0 .35 0
      .1 .0460 .0460 0 .1 0
      .1 .0460 .0460 0 -.1 0
      .1 .0460 .0230 -.08 -.605 0
      .1 .0230 .0230 0 -.606 0
      .1 .0230 .0460 .06 -.605 0 ]; % Параметри матриці E
```

P1 = phantom(E,128);

subplot(1,3,2);imshow(P1)

```
E1 = [ 1 .69 .92 0 0 0
       -6 .6624 .8740 0 -.0184 0
       -2 .1100 .3100 .22 0 -18
       -2 .1600 .4100 -.22 0 18
       .1 .2100 .2500 0 .35 0
       .1 .0460 .0460 0 .1 0
       .1 .0460 .0460 0 -.1 0
```

```

    .1 .0460 .0230 -.08 -.605  0
    .1 .0230 .0230  0  -.606  0
    .1 .0230 .0460 .06  -.605  0 ];
    P2 = phantom(E1,256);
subplot(1,3,3);imshow(P2)
whos P
whos P1
whos P2 % Вивід інформації про структуру файлів

```

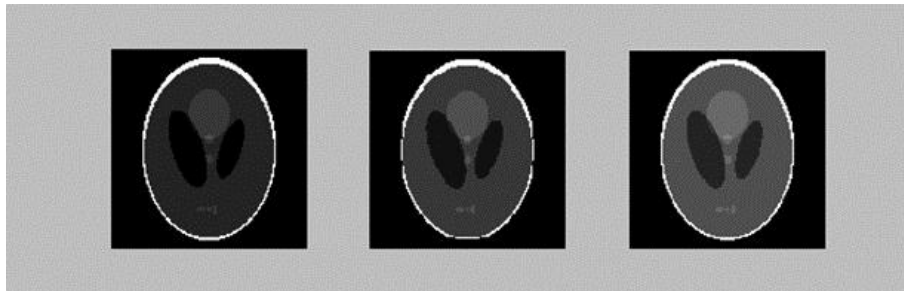


Рис.3.7 – Результати формування фантомів зрізу голови людини

3.3.3 Вивід вбудованих в MATLAB медичних зображень

За допомогою операції **imshow** можна вивести на екран монітора медичні зображення, наприклад:

```

close all
imshow spine.tif
whos

```

Це зображення являє собою елементи кісткової структури хребта. Даний фрейм буде використовуватися надалі при програмній обробці зображень.

Вивести на екран всі кадри вбудованої в MATLAB відеопослідовності магнітно-резонансних зображень **mri** (рис.3.8) за допомогою функції **montage(X,map)** – вивід k-кадров палітрового зображення **X** розміром $m \times n \times 1 \times k$ з використанням колірної карти **map**:

```

close all
load mri; % Загрузка відеопослідовності mri
mov=montage(D,map); % Вивід на екран цієї відеопослідовності
whos

```

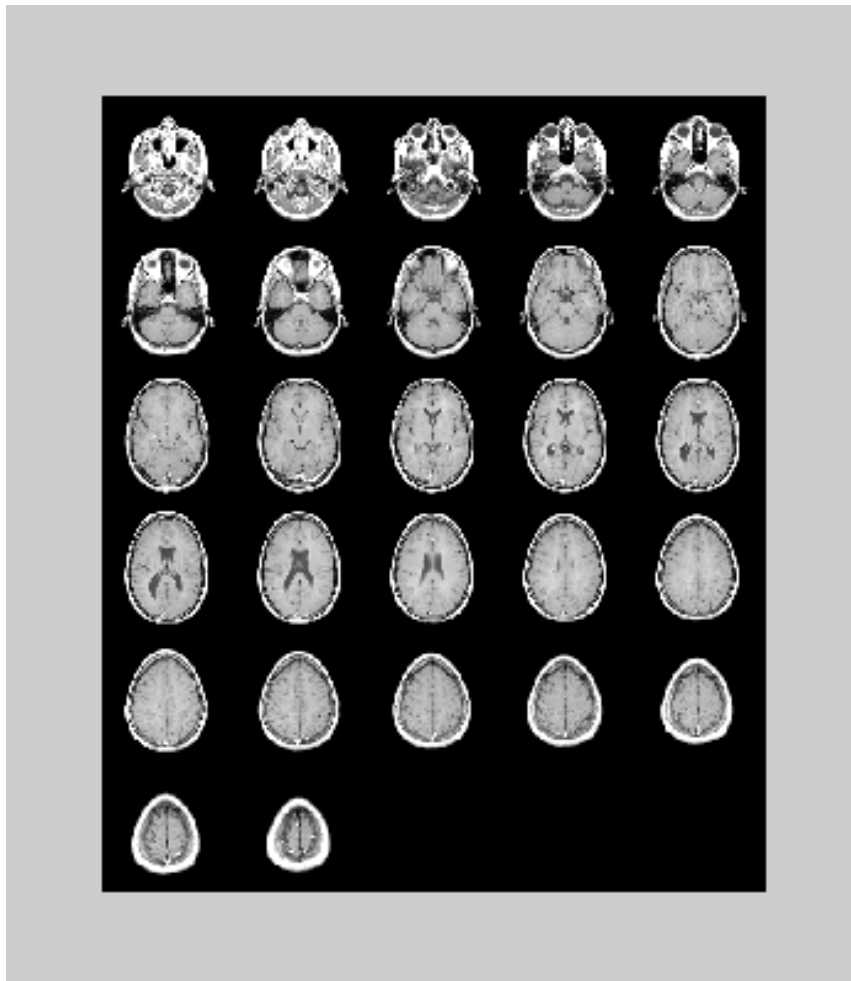


Рисунок 3.8 – Вивід 27 кадрів магнітно-резонансного зображення у вигляді послідовності зрізів

При завантаженні вказаної послідовності MATLAB видає повідомлення, що дане зображення надто велике та автоматично масштабується до визначеної кількості елементів.

Вивести один з кадрів отриманого вище зображення, скориставшись функцією читання зображення **imread('mri.tif',frame)** з файлу **'mri.tif'** під номером **frame** (вибрати вказаний викладачем кадр зображення).

```
close all  
I=imread('mri.tif',1);  
imshow(I)
```

З наведеного далі списку вбудованих в MATLAB файлів за допомогою операції **imread** (повний синтаксис функції наведений в додатку **help**) переглянути та вибрати файл, який надалі буде змінюватись за допомогою

інтегральних перетворень в наступних лабораторних роботах. Для доступу до інформації про параметри зображення застосувати функцію **whos**.

Список вбудованих файлів зображень з розширенням **.jpg** :

football.jpg greens.jpg

Список файлів зображень з розширенням **.tif** :

**autumn.tif board.tif cameraman.tif canoe.tif cell.tif circbw.tif
circles.tif forest.tif kids.tif lily.tif logo.tif m83.tif moon.tif mri.tif
paper1.tif pout.tif shadow.tif spine.tif tire.tif trees.tif**

Отриману інформацію зберегти у вигляді **.m** – та текстових файлів в своїй папці під своїм ім'ям.

3.4 Зміст звіту

У звіті наводиться:

- назва роботи;
- мета роботи;
- тексти завдань, програм та результати роботи з зображеннями;
- висновки.

3.5 Контрольні запитання

1. Які типи зображень використовуються в IPT MATLAB?
2. Як відбувається перетворення зображень?
3. Які формати зображень використовуються в IPT MATLAB?
4. Які системи координат застосовуються для опису зображень?
5. Як здійснюються операції зчитування даних для зображень?
6. Які особливості візуалізації бінарних, напівтонових та повноколірних зображень?
7. Які функції збереження зображень використовуються в IPT MATLAB?

ЛАБОРАТОРНА РОБОТА №4

«Створення графічних препаратів діагностичних зображень у системі MATLAB»

4.1 Мета роботи

Вивчення основних операцій MATLAB з перетворення діагностичних зображень та процедур отримання графічних препаратів для морфологічного аналізу біооб'єкту

4.2 Підготовка до виконання роботи

При підготовці до виконання ЛР слід ознайомитись зі способами отримання інформації про елементи зображення (визначення яскравості пікселів та відстані між ними). У медичній інтроскопії параметри елементів зображення ототожнюються з певними біофізичними параметрами досліджуваних анатомічних структур. Так, рентгенівські зображення та томограми відображають розподіл щільності тканин; ультразвукові зображення – розподіл значень акустичного імпедансу; емісійні томографічні зображення – розподіл концентрації радіофармпрепаратів, зафіксованих тканинами; магніто-резонансні томографічні зображення – розподіл концентрації ядер протонів водню в тканинах. Якщо дослідження займають певний проміжок часу, можна визначити не тільки просторові параметри анатомічних структур, але й часові параметри фізіологічних процесів в них: час поздовжньої та поперечної релаксації протонів водню, напрям та швидкість кровотоку, швидкість процесів виведення радіоізотопів з організму тощо. У всіх цих випадках основним джерелом інформації є інформація про пікселі в досліджуваній ділянці зображення – значення градації сірого кольору та координати. Додатково визначаються часові параметри шляхом формування послідовності зображень.

Першою стадією процесу діагностичної оцінки зображення є визначення морфологічних ознак зображених об'єктів (меж, розмірів, форми, взаємного розташування тощо). Далі необхідно отримати кількісне вираження цих ознак у величинах довжин, площ, кутів тощо. Для цього зображення мають аналізуватись на наявність певних закономірностей у розташуванні пікселів (контурів, ділянок певної яскравості, їх величини тощо). Виявлення цих особливостей можливе при здійсненні певних операцій над пікселями зображення. Так, визначення площі однорідної структури можна здійснити шляхом підрахунку кількості пікселів відповідної яскравості. Таку інформацію

можна отримати з гістограми зображення, для якої кількість пікселів різних градацій вже розрахована, треба лише вказати значення градації у зоні інтересу.

Обриси анатомічних структур виявляються при здійсненні сегментації зображення за певними рівнями яскравості. Тут можна застосувати порогову обробку зображення, бінаризацію тощо. Визначення лінійних розмірів об'єктів можна проводити за профілями розподілу яскравості пікселів у певному напрямку, визначаючи на них критичні точки, які відповідають анатомічним орієнтирам при визначенні меж об'єкту.

Операції, які передбачають перетворення початкового зображення, мають в результаті сформувати нове зображення (графічний препарат), що може аналізуватись окремо від оригіналу з метою виявлення певних специфічних діагностичних ознак.

4.2.1 Операції над пікселями зображення, реалізовані в MATLAB

Вивід інформації про піксель в зоні інтересу є стандартною операцією, реалізованою у всіх програмних засобах сучасних інтроскопічних систем. В пакеті IPT є можливість підключити функцію **pixval on** для відображення параметрів пікселя в поточному графічному вікні. В нижній частині вікна виводяться координати пікселя (номер рядка та стовпця) та його колірні атрибути. Також в цьому режимі можна виміряти евклідову відстань між пікселями. Початковий піксель фіксується натисканням лівої клавіші миші, після чого її курсор можна перемістити до другого пікселя – кінцевого. За курсором буде тягнутись пряма лінія, а під вікном буде відображатись відстань між пікселями (довжина прямої лінії). При відпусканні лівої клавіші миші позначення протяжки прямої лінії та відображення відстані між пікселями зникають. Прямокутник з даними про піксель можна переміщувати в будь-яке зручне місце. Функція **pixval off** відключає режим представлення пікселя. На рис.4.1 показаний результат роботи функції **pixval on** у наступному прикладі.

```
close all
```

```
P=phantom;% Функція фантома голови
```

```
imshow(P) % Вивід зображення
```

```
pixval on % Вивід інформації про піксель
```

Процедура виводу параметрів пікселів кольорового зображення має вид:

```
close all
```

```
RGB = imread('flowers.tif');
```

```
imshow(RGB)
```

```
pixval on
```

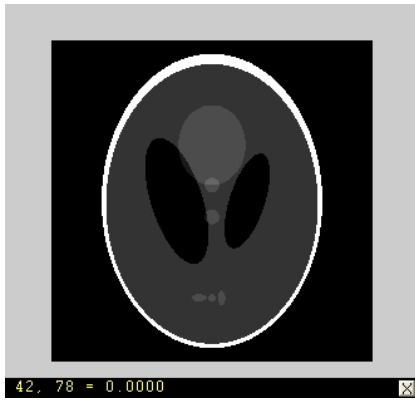


Рисунок 4.1 – Вивід інформації про пікселі та відстань між ними

Інформацію про яскравості пікселів можна отримати за допомогою функції **imread**. Вона повертає значення червоної (R), зеленої (G) та синьої (B) складових для вказаного пікселя зображення (двічі натиснути ліву клавішу миші в позиції курсору – у вікні MATLAB з’явиться інформація про параметри пікселя):

close all

RGB = imread('flowers.tif');

imshow(RGB)

imread % Визначення значення яскравості пікселя

Функції **P=imread(I)** **P=imread(X,map)** **P=imread(RGB)** виводять на екран напівтонове, палітрове або повноколірне зображення, відповідно, та переходять в інтерактивний режим очікування вибору пікселя.

Зміна яскравості медичних зображень, представлених в градаціях шкали сірого, може покращити умови візуального аналізу.

Застосовується спосіб зміни яскравості всього зображення шляхом підбору деяких параметрів усередненого за яскравістю зображення у вигляді функціонала

$$K = \sum_i \sum_j \left(\frac{I_{ij} - I_{\min}}{I_{\max} - I_{\min}} \right)^a,$$

де K – параметр яскравості, підшукується при аналізі зображення,

a – параметр, що обирається із залежності $K = f(a)$.

Цей спосіб зміни яскравості зображення в шкалі сірого можна проілюструвати наступною програмою, в якій засобами MATLAB коригується зображення фантома голови людини.

```

clear; close all
I=phantom(256);
figure, imshow(I); title('Image start');
[N M]=size(I); % Визначення розмірів матриці зображення
R=0.3; % Вибір параметра інтенсивності зображення
Imin=min(min(I)); % Визначення елемента масиву з min яскравістю
Imax=max(max(I)); % Визначення елемента масиву з max яскравістю
a=[0:.01:1]; % Формування масиву a
for i=1:length(a); % Цикл для визначення значення a, близького до R
    K(i)=sum(sum(((I-Imin)./(Imax-Imin)).^a(i)))/(N*M);
    if abs(K(i)-R)<0.01;
        st=a(i);
    end;
end;
figure,plot(a,K); % Побудова графіка  $K = f(a)$ 
grid;
% Формування зображення зі зміненою яскравістю
Iout=((I-Imin)./(Imax-Imin)).^st; figure,imshow(Iout);
% Визначення середнього значення інтенсивності зображення
AV=mean(mean(Iout));
disp(AV); % Вивід середнього значення інтенсивності

```

Результати здійснених перетворень зображень представлені на рис.4.2, додаткова інформація виводиться у вікні MATLAB.

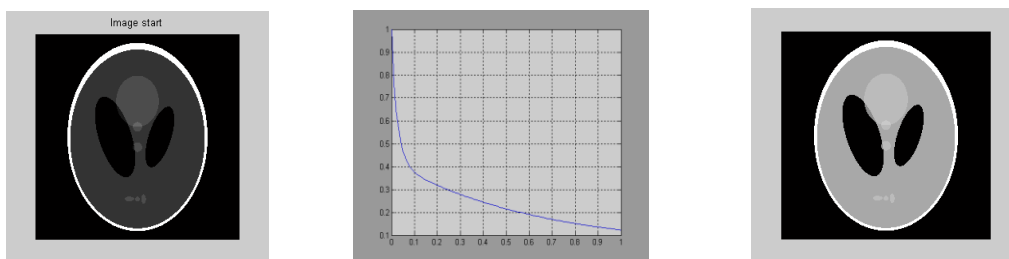


Рисунок 4.2 – Результати перетворень тестового зображення та графік $K = f(a)$

4.2.2 Побудова контурного зображення

Побудова контурного зображення є однією з необхідних процедур, яка застосовується при аналізі медичних зображень з метою виділення структур об'єкту, визначення параметрів патологічних осередків, які спостерігаються на

зображенні та ін. В системі MATLAB є функція **imcontour**, призначена для побудови контурного графіка напівтонового зображення з автоматичним встановленням осей, при якому їх орієнтація та формат відповідали зображенню, що аналізується. Зазначена функція будує лінії рівного рівня, які отримуються шляхом перетинання поверхні зображення сукупністю паралельно розташованих січних площин. Проекції ліній рівного рівня на площину являють собою контурний графік аналізованого зображення.

Наведемо приклад формування контуру зображення фантома голови:

```
clear; close all  
P=phantom;  
figure;  
subplot(1,2,1); imshow(P); subplot(1,2,2);  
imcontour(P)
```

Результати визначення контурів елементів, які формують зображення фантома голови, представлені на рис.4.3.

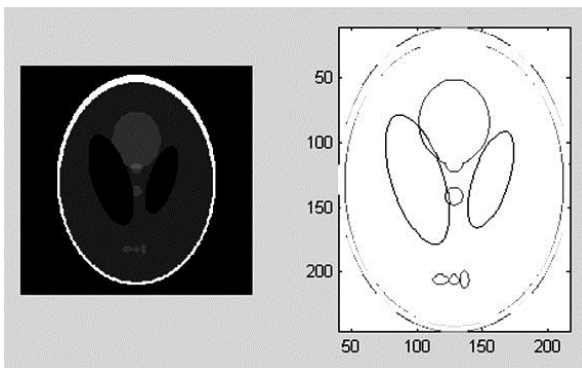


Рисунок 4.3 – Формування контурного зображення фантома голови

При побудові контурного графіка кольорового зображення, можна скористатись послідовним перетворенням кольорового зображення в напівтонове з наступним виділенням контурів за певною кількістю рівнів шкали сірого. Наведемо приклад побудови контурного зображення кольорового зображення, що знаходиться в директорії, вказаній в функції **imread**:

```
clear;close all  
% Зчитування кольорового зображення з файлу із вказаним повним ім'ям та розширенням  
VM=imread('C:\Documents and Settings\user\Мои документы\...  
Мои рисунки\VM.jpg');
```

```

subplot(1,3,1);subimage(VM);
% Перетворення кольорового зображення в напівтонове
IVM=rgb2gray(VM);
subplot(1,3,2);subimage(IVM);
subplot(1,3,3);
% Вивід контурного зображення із вказаною кількістю рівнів шкали сірого
imcontour(IVM,3)

```

4.2.3 Виділення меж об'єктів та сегментація зображень

Виділення меж між різними за яскравістю ділянками напівтонового медичного зображення дає можливість сегментувати зображення, виявляти межі об'єктів в області інтересу за різними напрямками (по горизонталі, вертикалі та в інших напрямках).

В MATLAB є функція **BW=edge(I, method)**, яка призначена для виділення меж на початковому напівтоновому зображенні **I**. Дана функція повертає бінарне зображення **BW** такого ж розміру, як і початкове **I**. Піксель **BW(r, c)** має значення 1, якщо піксель **I(r,c)** належить межі. Для виявлення меж можуть застосовуватись кілька методів. Кожен з них задається в параметрі **method** у вигляді одного з наступних рядків: 'sobel', 'prewitt', 'roberts', 'log', 'zerocross', 'canny'. Якщо параметр **method** при виклику функції опущений, то за замовчуванням він набуває значення 'sobel'.

Для кожного з методів визначення меж можна задати додаткові параметри. Для цього використовується одна з функцій:

BW=edge(I, method, thresh), BW=edge(I, method, thresh, P) ,

де параметр **thresh** задає поріг для визначення належності пікселя межі, а в параметрі **P** передаються настройки, специфічні для кожного з методів.

Якщо при виклику функції параметр **thresh** опущений, то значення порога обирається автоматично. Отримати значення порога можна, визначивши кінцевий параметр **thresh: [BW, thresh]=edge(I, method,...)** .

Розглянемо застосування функції **edge** щодо найбільш часто вживаного методу виділення меж – методу Собеля. Функція **BW=edge(I, 'sobel', thresh)** для визначення меж використовує фільтрацію початкового зображення **I** фільтром Собеля: піксель вважається таким, що належить межі, якщо відповідний йому піксель результату фільтрації має значення, більше за **thresh**. Для даного метода можна вказати додатковий параметр **direction:**

BW=edge(I, 'sobel', thresh, direction), він визначає, які межі будуть виявлятися.

Параметр **direction** може набувати значень:

'horizontal' - виділення горизонтальних меж;

'vertical' - виділення вертикальних меж;

'both' - виділення меж за всіма напрямками (дане значення використовується за замовчуванням, коли параметр **direction** не визначений).

Наведемо приклад реалізації операції з виділення меж структур на зображенні фантома голови (рис.4.4).

```
clear; close all
```

```
I=phantom;
```

```
imshow(I);
```

```
% Виділення меж методом Собеля та вивід результату на екран
```

```
BW=edge(I, 'sobel', 0.01);
```

```
figure, imshow(BW);
```



Рисунок 4.4 – Виділення меж структур на зображенні фантома голови людини функцією **edge**

Сегментація зображень реалізується у наступний спосіб. Зображення розбивається на блоки, що не перекриваються. Кожен блок за допомогою певного критерію перевіряється на однорідність. Якщо блок неоднорідний, то він розбивається на блоки меншого розміру, кожен з яких, в свою чергу, перевіряється на однорідність. Процес завершується тоді, коли жоден з блоків не може бути розділений, тобто або блоки однорідні, або ж їх розміри досягли гранично малих величин. В результаті роботи алгоритму отримуємо зображення у вигляді набору однорідних блоків різного розміру.

Для сегментації напівтонових зображень методом розділення застосовується функція **qtdecomp**. Функція **qtdecomp** розбиває кожен блок на чотири блоки однакового розміру, що не перекриваються. На першому кроці алгоритму блоком вважається все зображення. Найменшим за розміром є блок, який не можна розділити на 4 блоки однакового розміру. Таким чином, в функції **qtdecomp** рекомендується використовувати зображення з розмірами,

рівними 2^n . В цьому випадку найменший блок буде складатись з одного пікселя. При реалізації алгоритму розділення використовуються структури даних, основані на квадродеревах. Цей алгоритм називають декомпозицією.

Функція $A=qtdecomp(I)$ здійснює сегментацію напівтонового зображення I методом розділення та розміщує результат в розрідженому масиві A (тип даних MATLAB - **sparse**). Розріджений масив A конструюється наступним чином. Елементом матриці $A(r,c)$, які відповідають координатам лівих верхніх кутів блоків на початковому зображенні I , присвоюються значення, які визначають розміри кожного блока. Таким чином, більшість елементів матриці дорівнює нулю. Тому для зберігання квадродерева застосовується розріджений масив, який ефективно використовує пам'ять. Для даної функції критерієм однорідності блока є рівність всіх пікселів блоку один одному. Функція $A=qtdecomp(I, threshold)$ працює аналогічно наведеній вище, але в ній блок вважається однорідним, якщо різниця між максимальним та мінімальним значеннями пікселів блоку менша за параметр **threshold**.

Наведемо приклад реалізації процедури сегментації зображення тестового зображення. Результат зображений на рис.4.5 .

```
clear;close all
I4=zeros(128,128);
I4(20:108,20:108)=0.8;
I4(20:30,98:108)=0.5;
I4(98:108,20:30)=0.5;
I4(98:108,98:108)=0.5;
I4(20:30,20:30)=0.5;
I4(46:85,46:85)=1;
I4(84:98,84:98)=1;
I4(31:45,31:45)=1;
subplot(1,2,1);imshow(I4);
S=qtdecomp(I4,0.1);subplot(1,2,2);
imshow(S)
```

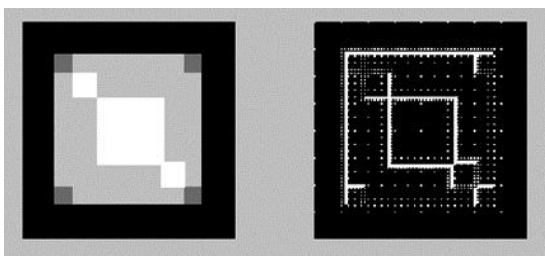


Рисунок 4.5 – Результат сегментації тестового зображення

За допомогою функції **qtgetblk** можна вивести всі блоки, отримані в результаті сегментації наведеного вище зображення. Інформація про реалізацію такої дії відображається в основному вікні MATLAB:

```
[vals,r,c]=qtgetblk(I3,S,4);  
vals(:,:,1)  
vals(:,:,2)  
vals(:,:,3)  
vals(:,:,4)  
vals(:,:,5)  
vals(:,:,6) ...
```

4.2.4 Побудова гістограми та профілів розподілу яскравості

Гістограмний аналіз діагностичного зображення дає можливість визначити «внесок» кожної з градацій шкали сірого. Для цього застосовуються функції **imhist(I, n)** та **imhist(BW, n)**, які будують гістограми яскравостей пікселів відповідно напівтонового та бінарного зображень в поточному робочому вікні. Гістограма складається з **n** стовпців. Значення **n** при виклику функції можна не вказувати, тоді будуть застосовані значення за замовчуванням: **n=256** для напівтонового зображення та **n=2** для бінарного зображення. Під рисунком гістограми виводиться шкала яскравостей. Наведемо приклад побудови гістограми магніторезонансного зображення голови людини (1-й зріз). Результат наведений на рис.4.6.

```
close all  
I=imread('mri.tif',1);  
subplot(1,2,1);imshow(I);subplot(1,2,2);  
imhist(I)
```

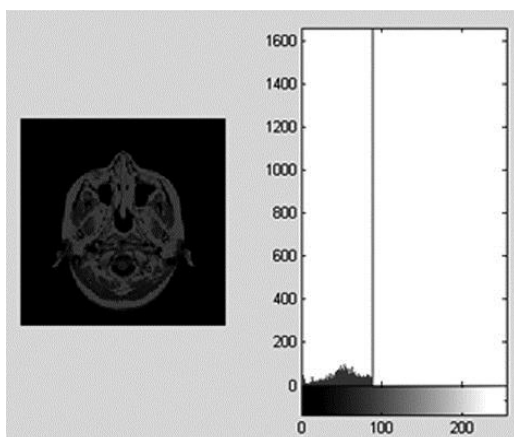


Рисунок 4.6 – Гістограма магніторезонансного зображення зрізу голови людини

Одним зі способів покращення зорового сприйняття зображення є зміна його контрастності, яку можна здійснити шляхом вирівнювання (еквалізації) гістограми. Цей спосіб корекції зображення реалізується за допомогою функції **histeq**, яка має ряд форм, опис яких наводиться нижче.

Функція **histeq(Is,hgram)** перетворює початкове напівтонове зображення **Is** таким чином, щоб гістограма яскравостей пікселів отриманого нового напівтонового зображення **Id** приблизно відповідала гістограмі, яку задає вектор **hgram**. Кількість елементів, вказана в **hgram**, задає число стовпців гістограми, а значення кожного елемента - відносну висоту кожного стовпця. Значення елементів вектора **hgram** має бути в діапазоні [0, 1]. Функція **hgram** автоматично масштабує значення елементів **hgram** так, щоб сума значень елементів в гістограмі дорівнювала кількості пікселів зображення. Гістограма кінцевого зображення **Id** буде краще відповідати заданій гістограмі **hgram** в тому випадку, коли кількість стовпців **hgram** набагато менша за кількість градацій яскравості початкового зображення **Is**.

Функція **Id=histeq(Is,n)** перетворює початкове напівтонове зображення **Is** таким чином, щоб отримане напівтонове зображення **Id** мало гістограму яскравостей пікселів, близьку до рівномірної. Рівномірна гістограма **hgram** створюється з **n** стовпців як **hgram=ones(1, n)*prod(size(Is))/n**. Чим менше **n** в порівнянні з кількістю градацій яскравості в зображенні **Is**, тим більш рівномірною стає гістограма яскравостей пікселів отриманого в результаті зображення **Id**. За замовчуванням значення **n=64**, цей параметр можна не вказувати при виклику функції. Формат отриманого зображення **Id** співпадає з форматом початкового зображення **Is**. Приклад вирівнювання гістограми медичного зображення ілюструється такою послідовністю дій із застосування функції **histeq**:

```
clear;close all
I=imread('mri.tif',1); J=histeq(I); figure;
subplot(2,2,1);imshow(I);subplot(2,2,2);imhist(I);
subplot(2,2,3);imshow(J);subplot(2,2,4);imhist(J);
```

Профіль розподілу яскравості у перетині зображення відображається як крива залежності яскравості від координати в заданому напрямку. За такою кривою можна обчислювати геометричні розміри об'єкту на медичному зображенні (наприклад, так можна визначити розмір перетину трубчастої кістки за її рентгенограмою тощо). В MATLAB є функція **improfile**, яка обчислює

значення яскравості (так званий профіль яскравості) вздовж путі, який складається з одного чи кількох відрізків прямої. Операція застосовується щодо градацій напівтонових зображень або значень R-, G-, B-складових (профілі кольорних складових) у повноколірних зображеннях.

Функції **c=improfile** та **c=improfile(n)** дозволяють в інтерактивному режимі на зображенні в поточному вікні задавати путь, вздовж якого обчислюється профіль яскравості, Параметр **n** задає кількість точок, в яких обчислюються значення яскравості чи кольори пікселів. Ці **n** точок розташовані вздовж заданого путі на однаковій відстані. Якщо параметр **n** не заданий, то він встановлюється приблизно рівним кількості пікселів, які знаходяться на перетині з лінією путі. Для напівтонових зображень формується результат – параметр **c**, який є одновимірним масивом, в який записуються **n** значень яскравості, а для повноколірних зображень параметр **c** записується як двовимірний масив **nx3**.

Координати відрізків, які формують путь, задаються натисканням лівої клавіші миші без фіксації. В разі помилки натисканням клавіш **Backspace** або **Delete** можна видалити попередньо задану точку. Натискання правої клавіші миші або подвійне натискання лівої клавіші задає кінцеву точку останнього відрізка путі та завершує процес вибору пікселів. Завершити процес вибору пікселів можна і без вказування на останній піксель, для цього достатньо натиснути клавішу **Enter**. Коли путь заданий, в масив-результат повертаються значення градацій кольору тих пікселів, які належать обраному профілю.

Якщо параметри **c** при виклику функцій **improfile** не вказані, тоді профіль у вигляді графіку виводиться на екран в нове вікно. При цьому, якщо путь складається з одного відрізка, то графік - двовимірний, його координата X співпадає з заданим відрізком. Якщо в путі більш ніж один відрізок, то графік будується як тривимірний, на осях X та Y якого вказуються просторові координати зображення. Для повноколірних зображень в одних координатах одночасно виводяться 3 графіки: червоним, зеленим та синім кольором для R-, G- та B-складових відповідно.

В функціях **c=improfile(S, xi, yi)** та **c=improfile(S, xi, yi, n)** для напівтонового або повноколірного зображення **S** путь задається явно за допомогою векторів **xi, yi** однакової довжини, які містять значення координат кінцевих точок відрізків путі в просторовій системі координат. Наведемо приклад реалізації побудови яскравісного профілю фантому голови (рис.4.8). Тут при побудові яскравісного профілю застосовується білінійна інтерполяція даних '**bilinear**', хоча можливі також інші методи: '**nearest**' – застосовується

значення найближчого пікселя (параметр встановлений за замовчуванням, тому він може бути опущений при виклику функції); **'bicubic'** – застосовується інтерполяція по бікубічній поверхні.

I=phantom;

imshow(I);

% Побудова профілю яскравості зображення вздовж прямої в центрі

% зображення зліва направо.

figure, improfile(I, [128 128], [1 256], 'bilinear');

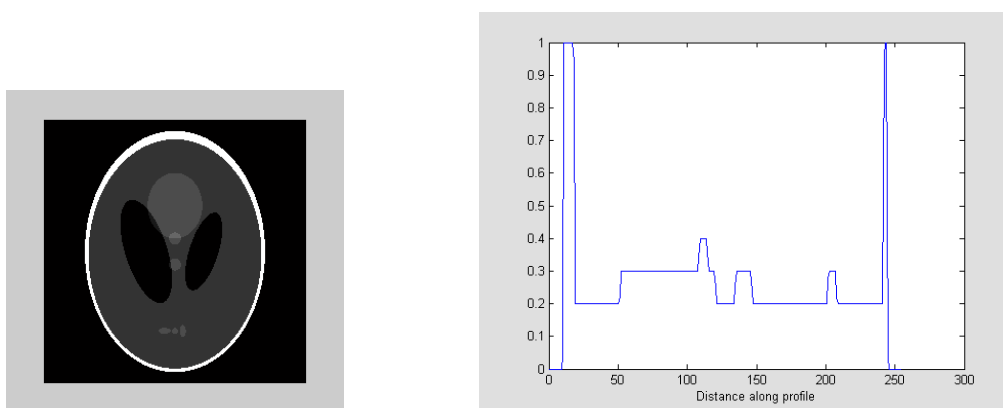


Рисунок 4.8 – Побудова яскравісного профілю перетину фантому голови

4.2.5 Геометричні операції над зображеннями

Ряд функцій Image Processing Toolbox реалізують типові геометричні операції над зображеннями (кадрування зображення, зміна його розмірів, поворот зображення).

Операція кадрування застосовується для відокремлення зон інтересу з більшого зображення. Це може бути корисним при архівуванні результатів досліджень, яке потребує зменшення розмірів графічних файлів. Кадрування здійснюється за допомогою функцій **D=imcrop(S)** та **Xd=imcrop(Xs, map)**. При цьому на екран виводиться бінарне, напівтонове або повноколірне зображення **S** та палітрове зображення **Xs**, після чого програма очікує на позначення прямокутника, який обмежує фрагмент, що вирізається з метою створення кадру певних розмірів. Для позначення положення одного з кутів обмежувального прямокутника слід натиснути ліву кнопку миші. Положення протилежного кута прямокутника буде задано, якщо, не відпускаючи кнопку миші, перемістити курсор, а потім відпустити кнопку. Коли потрібно позначити квадратний фрагмент, слід одночасно тримати натиснутою клавішу **Shift**.

Функція **imcrop** повертає зображення, обмежене заданим прямокутником. Якщо вихідні параметри не задані, наприклад **imcrop(I)**, то результат відображається в новому графічному вікні. Якщо не визначені вхідні параметри, наприклад, **D=imcrop**, то функція **imcrop** оперує із зображенням в поточному графічному вікні. Застосовуючи функції **D=imcrop(S, rect)** та **Xd=imcrop(Xs, map, rect)**, можна явно описати параметри обмежувального прямокутника: **rect** - вектор з чотирьох елементів: [**x_{min} y_{min} w h**], які задають положення лівого верхнього кута (**x_{min} y_{min}**) прямокутника, а також його ширину (**w**) та висоту (**h**) в просторових координатах.

Наступний приклад ілюструє застосування функції **imcrop** для обробки кольорового зображення **'flowers.tif'**: тут можна виділити необхідний фрагмент зображення за допомогою лівої клавіші миші. Отриманий фрагмент можна зберегти як нове зображення **I2**.

```
clear;close all  
I=imread('flowers.tif');  
I2=imcrop(I) % Операція кадрування зображення  
imshow(I2)
```

Операція зміни розмірів зображення часто застосовується для детального аналізу певних анатомічних областей. Зміна розмірів зображення здійснюється за допомогою функції **imresize**, яка створює нове зображення будь-якого типу, що відрізняється від початкового в деяке число разів. Для зміни розмірів застосовується один з наперед визначених методів інтерполяції, який задається у вхідному параметрі **method** у вигляді одного з наступних рядків: **'nearest'**, **'bilinear'**, **'bicubic'**. Так, функція **D=imresize(S, m, method)** створює нове зображення **D**, розміри якого в **m** разів відрізняються від розмірів початкового зображення **S**. Якщо **m** належить діапазону від 0 до 1, то зображення **D** менше за **S**, якщо ж коефіцієнт **m** більше 1, то **D** більше **S**.

Функція **D=imresize(S, [mrows ncols], method)** створює нове зображення **D** з розмірами **mrows** × **ncols**. Коли розміри отриманого зображення менші за розміри початкового, і при цьому застосовується інтерполяція по білінійній чи бікубичній поверхні, початкове зображення попередньо фільтрується фільтром нижніх частот з метою усунення ефекту ступінчастості. За замовчуванням застосовується усереднюючий фільтр з розміром маски 11x11 пікселів. Можна задати довільний розмір маски фільтра **n**×**n**, визначивши додатковий параметр

n в функції: **D=imresize(..., method, n)**. Якщо **n** дорівнює 0, то попередня фільтрація не виконується.

Тип та формат представлення даних отриманого масштабованого зображення співпадають з типом та форматом початкового зображення.

Приклад реалізації зміни розмірів зображення з інтерполяцією по білінійному закону ілюструється такою процедурою:

```
clear;close all  
I1=imread('lily.tif');  
imshow(I1)  
I2=imresize(I,0.3,'bilinear')  
figure;imshow(I2)
```

Операція повороту зображення на визначений кут є однією зі стандартних маніпуляцій з зображенням. При інтроскопічних дослідженнях нерідко трапляються випадки порушення стандартного позиціонування пацієнта відносно скануючої системи. Тоді «вирівнювання» застосовується вже не до пацієнта, а до отриманого зображення шляхом його повороту. Для цього використовуються функції виду:

```
D=imrotate(S, angle, method)  
D=imrotate(S, angle, method, crop) ,
```

де функція **D=imrotate(S, angle, method)** створює зображення **D**, яке відповідає повернутому початковому зображенню **S**.

При повороті застосовується один з наперед визначених методів інтерполяції, який задається у вхідному параметрі **method** одним з наступних рядків: **'nearest'**, **'bilinear'**, **'bicubic'**. Кут повороту **angle** задається в градусах. Додатні значення даного параметра відповідають повороту проти годинникової стрілки, а від'ємні – за годинниковою стрілкою.

В загальному випадку кількість пікселів отриманого після повороту зображення **D** є більшою або дорівнює кількості пікселів початкового зображення **S**. Значення пікселів зображення **D**, для яких відсутні відповідні їм пікселі в зображенні **S**, встановлюються в 0, що, наприклад, для напівтонових зображень відповідає чорному кольору. Функція **D=imrotate(S, angle, method, crop)** має додатковий параметр **'crop'**, який вказує на те, що слід повернути

зображення з розмірами, які дорівнюють розмірам **S**. В цьому випадку для кадрування обирається центральна частина повернутого зображення **D**.

Тип та формат представлення даних отриманого масштабованого зображення співпадають з типом та форматом початкового зображення.

Далі наводиться приклад реалізації повороту зображення на 30° з наступною інтерполяцією білінійного типу.

```
clear;close all
```

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

```
I2=imrotate(I,30,'bilinear')
```

```
imshow(I2)
```

4.3 Порядок виконання роботи

Запустити програму MATLAB. За допомогою функції **help images** ознайомитись з можливостями пакету **Image Processing Toolbox** (див. п.4.2).

4.3.1 Здійснити операції з пікселями тестових зображень. Зверніться до тестових зображень, сформованих Вами в попередній лабораторній роботі, та виведіть інформацію про параметри пікселів з обраних елементів зображення (див.п.4.2.1). Проведіть вимірювання відстаней між окремими фрагментами зображення. Зазначені операції застосуйте як до напівтонового, так і до кольорового зображень. Отриману інформацію збережіть як текстові файли у Вашій робочій папці.

4.3.2 Змінити яскравість тестових зображень. Виконайте зміни яскравості створених Вами фантомних зображень за процедурою, описаною в п. 4.2.1. Виведіть інформацію про значення середньої яскравості зображення. Обравши значення параметра інтенсивності зображення **R**, виведіть зображення зі зміненою яскравістю, більш зручне для зорового аналізу.

4.3.3. Побудувати контурне зображення за п.4.2.2. Утворіть контурні зображення фантомних зображень, розроблених вами у попередній лабораторній роботі, та надайте їх у градаціях шкали сірого. Побудуйте контурні зображення, користуючись підготовленими раніше кольоровими зображеннями. Контури відобразити заданим типом лінії рівня та кольором для її зображення.

4.3.4 Виконати виділення меж та сегментацію напівтонових фантомних зображень за п.4.2.3. Обравши значення параметра **tresh**, виділити структури,

з яких формуються аналізовані зображення. Повторіть зазначені операції для автоматичного вибору значення порога виділення та для різних типів фільтрів.

Для здійснення сегментації тестових напівтонових зображень приведіть їх матриці до розміру 2^n , де $n > 5$. Виконайте сегментацію зображень з обраним порогом **threshold** та виведіть інформацію про реалізацію проведеної сегментації в основному вікні програми.

4.3.5 Побудувати гістограми та профілі яскравості зображення за п.4.2.4. Гістограма будується для заданого викладачем зображення (наприклад, певний знімок з послідовності магніто-резонансних зображень, фантомні зображення голови тощо). Побудуйте яскравісні профілі перетинів зображень, вказаних викладачем.

4.3.6 Здійснити геометричні операції над зображеннями за п.4.2.5. Для вказаних викладачем тестових зображень виконайте операції кадрування, зміни розмірів та повороту.

Інформацію, отриману в ході виконання лабораторної роботи, зберегти у вигляді **.m** - файлів, текстових файлів та графічних файлів з їх розширеннями, в своїй папці та під своїм ім'ям.

4.4 Зміст звіту

У звіті наводяться:

- назва роботи;
- мета роботи;
- тексти завдань, програм та результати виконання операцій над зображеннями за п.п. 4.3.1 – 4.3.6;
- висновки.

4.5 Контрольні запитання

1. Які операції з пікселями зображень реалізовані в MATLAB?
2. Як отримуються контурні зображення в MATLAB?
3. Як і для чого здійснюється виділення меж та сегментація зображень?
4. Що зображають та як будуються гістограми зображень та профілі їх яскравості?
5. Які основні геометричні операції над зображеннями доступні в MATLAB?

ЛАБОРАТОРНА РОБОТА №5

«Морфологічні перетворення та структурний аналіз зображень в MATLAB»

5.1 Мета роботи

Вивчення методів морфологічних перетворень біомедичних зображень. Аналіз структурних елементів зображень із застосуванням засобів морфологічного аналізу.

5.2 Підготовка до виконання роботи

При підготовці до виконання ЛР слід ознайомитись з морфологічними операціями щодо бінарних та напівтонових зображень, методами оцінки параметрів діагностичного зображення. Також потрібно ознайомитись з функціями просторових перетворень та цілочисельними арифметичними операціями над зображеннями, реалізованими в системі MATLAB.

5.2.1 Морфологічні операції над бінарними зображеннями

Морфологічні перетворення використовують при аналізі діагностичних зображень з метою пошуку на них структурних елементів певного типу, виявлення закономірностей їх розташування. Комп'ютерний морфологічний аналіз зображень дає можливість автоматизувати процес знаходження об'єктів, їх відокремлення, визначення розмірів, підрахунку тощо.

Методи розпізнавання образів містять у собі морфологічні операції, які застосовуються для зміни форми об'єктів. Ці операції є **нелінійними**, вони базуються на врахуванні зв'язності пікселів в деяких аналізованих блоках зображення. В MATLAB прийнято вважати, що пікселі зі значенням 1 відносяться до об'єктів, а зі значенням 0 – до фону.

Пояснимо поняття *зв'язності пікселів*. *Квартетом сусідів* пікселя **P** називають 4 оточуючих його пікселя **x1**, **x3**, **x5** та **x7**.

x4	x3	x2
x5	P	x1
x6	x7	x8

Відповідно, *октетом сусідів* називають всі 8 пікселів, що оточують піксель **P**. Об'єкт називають 4-зв'язним, якщо квартет сусідніх пікселів містить 1. Відповідно, об'єкт називають 8-зв'язним, якщо октет оточуючих його пікселів

містить 1. Поняття зв'язності застосовується і до фону: 4-зв'язність фону автоматично означає 8-зв'язність об'єктів і навпаки.

Розглянемо приклад зв'язності об'єктів в матриці зображення. Якщо об'єкти даного зображення розглядати як 4-зв'язні, то в ньому будуть два таких об'єкти, а якщо розглядати як 8-зв'язні, то буде тільки один об'єкт.

0	0	1	0	0	0	0	0	0	0
1	1	1	1	1	1	0	1	1	0
0	0	1	1	1	0	1	1	1	1
0	0	1	0	0	0	1	0	1	1
0	0	0	0	0	0	1	1	1	0

0	0	1	0	0	0	0	0	0	0
1	1	1	1	1	1	0	1	1	0
0	0	1	1	1	0	1	1	1	1
0	0	1	0	0	0	1	0	1	1
0	0	0	0	0	0	1	1	1	0

В MATLAB операція оцінки зв'язності бінарних об'єктів здійснюється за допомогою функції `bwlabel(BW,n)`. Для вищенаведеної матриці зображення операція визначення зв'язності об'єктів матиме вигляд:

```
clear,close all;
```

```
BW=[0 0 1 0 0 0 0 0 0 0;
    1 1 1 1 1 1 0 1 1 0;
    0 0 1 1 1 0 1 1 1 1;
    0 0 1 0 0 0 1 0 1 1;
    0 0 0 0 0 1 1 1 0];
```

```
[L,num]=bwlabel(BW,4) %Представлення матриці зв'язаних об'єктів, num —  
представляє їх кількість, n – набуває значення 4 або 8
```

```
[L,num]=bwlabel(BW,8)
```

```
num =
```

```
2
```

```
num =
```

```
1
```

Виконання необхідної морфологічної операції в MATLAB здійснюється за допомогою функції `BWd=bwmorph(BWd, operation)`, яка створює нове бінарне зображення `BWd`, оброблюючи початкове бінарне зображення `BWs` морфологічним фільтром. Тип застосованого морфологічного фільтра визначається параметром `operation` (табл.5.1).

Таблиця 5.1. Параметри функції **operation**

Параметр	Морфологічна операція	
"erode" <i>Ерозія</i>	Заміна значень граничних пікселів об'єкта на 0. Одноразове застосування <i>ерозії</i> призводить до зменшення товщини граничного шару на 1 піксель («виїдання» межі об'єкту зовні)	
"dilate" <i>Нарощування</i>	Заміна значень пікселів фону, які межують з об'єктом, на 1. Одноразове застосування <i>нарощування</i> додає зовні об'єкту шар товщиною в 1 піксель	
"open" <i>Відкриття</i>	Послідовне застосування <i>ерозії</i> та <i>нарощування</i> . Призводить до з'єднання областей фону, раніше роз'єднаних вузькими ділянками пікселів об'єктів	
"close" <i>Закриття</i>	Послідовне застосування <i>нарощування</i> та <i>ерозії</i> . Призводить до видалення невеликих за площею фрагментів фону всередині об'єктів	
"tophat" <i>"Верх капелюха"</i>	Віднімання від початкового зображення результату операції <i>Відкриття</i> цього зображення	
"bothat" <i>"Низ капелюха"</i>	Віднімання початкового зображення від результату його <i>Закриття</i>	
"clean" <i>Вичищення</i>	Ізольовані пікселі зображення (всі їх 8 сусідів дорівнюють 0) замінюються з 1 на 0	
"fill" <i>Заповнення</i>	Ізольовані пікселі фону (всі їх 8 сусідів дорівнюють 1) замінюються з 0 на 1	
"diag" <i>Діагональ</i>	Знищення 8-зв'язності фону додаванням необхідної кількості одиниць у фрагменти об'єктів, зв'язаних тільки по діагоналі	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$
"bridge" <i>Місток</i>	З'єднання пікселів об'єкту, роз'єднаних фрагментом фону товщиною в 1 піксель	$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$
"hbreak" <i>Розрив «Н»</i>	Видалення центрального пікселя в конфігураціях, схожих на літеру "Н"	$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$

Продовження табл.5.1

<p>"remove" <i>Видалення</i></p>	<p><i>Видалення</i> внутрішніх пікселів: в 0 встановлюються пікселі об'єкту, у яких 4 сусідніх по горизонталі та вертикалі пікселя були рівні 1, тобто також були пікселями об'єкту. Не обнуленими залишаються тільки пікселі межі об'єкту</p>	
<p>"majority" <i>Більшість</i></p>	<p><i>Ерозія та нарощування</i> за перевагою в околі пікселів фону або об'єкту: якщо в околі пікселя розміру 3x3 знаходиться 5 або більше пікселів об'єкту, то цей піксель встановлюється в 1, в протилежному випадку – в 0.</p>	
<p>"skel" <i>Скелет</i></p>	<p>Побудова остову об'єкту: виконується <i>ерозія</i> за умови збереження 8-зв'язності остову. В декілька дій можна побудувати остов у вигляді зв'язної лінії, яка проходить по середині об'єкту (мінімальна товщина лінії – 1 піксель)</p>	
<p>"shrink" <i>Стиснення</i></p>	<p>Виконується <i>ерозія</i> за умови збереження 8-зв'язності замкнених ділянок остову. За декілька дій об'єкти, які не містять дірок, перетворюються на точки, а об'єкти з дірками "стискаються" до 8-зв'язних замкнених ділянок остова, які проходять посередині та вздовж зовнішньої межі об'єкту</p>	
<p>"thin" <i>Зменшення товщини</i></p>	<p>Виконується <i>ерозія</i> за умови збереження 8-зв'язності замкнених ділянок остову. За кілька дій об'єкти, які не містять дірок, перетворюються на одну чи кілька зв'язних ліній з мінімальною кількістю розгалужень остову, а об'єкти з дірками "стискаються" до 8-зв'язних замкнених ділянок остову, які проходять посередині між зовнішньою межею об'єкту та межами дірок</p>	
<p>"thicken" <i>Збільшення товщини</i></p>	<p><i>Нарощування</i> об'єкту за умови збереження 4-зв'язності ділянок фону. Виконується побудова остову фону (результат зворотній результату операції 'thin')</p>	
<p>"spur" <i>Відгалуження</i></p>	<p>Стирання відгалужень об'єкту товщиною в 1 піксель (коли тільки один сусідній піксель встановлено в 1, а інші - в 0). За кілька дій над зображенням остову на зображенні залишаються лише замкнені ділянки остову</p>	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$

Наведемо приклад застосування до бінарного зображення сукупності кіл **circles.tif** трьох морфологічних операцій – видалення внутрішніх пікселів кіл; побудови остову сукупності кіл; зменшення зі збереженням остову (рис.5.1):

```
clear;close all
BW1=imread('circles.tif'); BW2=bwmorph(BW1,'remove');
BW3=bwmorph(BW1,'skel',Inf); BW4=bwmorph(BW1,'thin',8);
iptsetpref('ImshowBorder','tight'); figure;
subplot(2,2,1); subimage(BW1);subplot(2,2,2); subimage(BW2);
subplot(2,2,3); subimage(BW3);subplot(2,2,4); subimage(BW4);
```

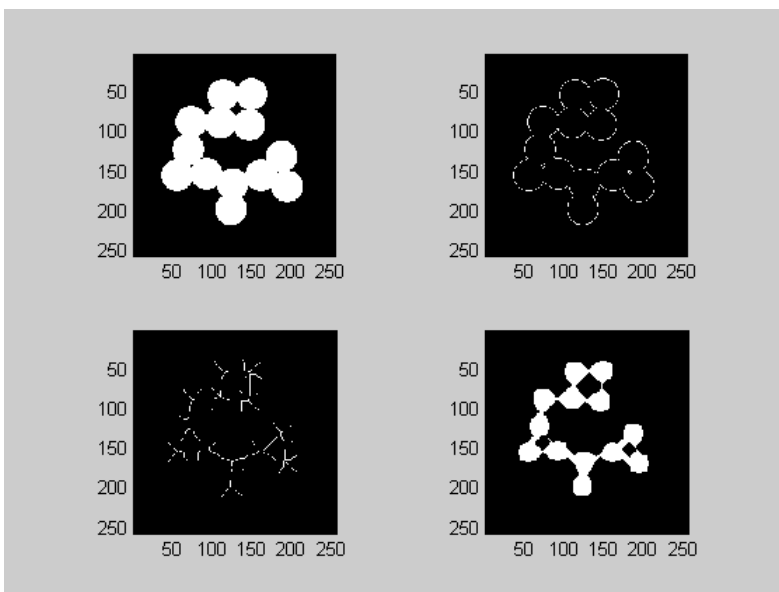


Рисунок 5.1 – Морфологічні перетворення бінарного зображення **circles.tif**

Функція **total=bwarea(BW)** дає можливість оцінити площу об'єктів бінарного зображення. Параметр **total** є числом, значення якого приблизно залежить від кількості пікселів в зображенні. Наведемо приклад оцінки площі (як кількості пікселів), яку займає квадрат з одиниць розміром 10×10 , всередині якого «вирізаний» квадратний отвір з 6×6 нулів, тобто зображення, яке складається з одиниць, формують 64 пікселі:

```
clear;close all
I1=ones(10,10);
I1(3:8,3:8)=0;
whos
total=bwarea(I1)
```

Name	Size	Bytes	Class
I1	10x10	800	double array

Grand total is 100 elements using 800 bytes

total =

64.5000

Функція **bwperim(BW,n)** дозволяє побудувати периметр всіх структур, які формують зображення. Піксель належить периметру, якщо його значення дорівнює 1 та існує як мінімум один нульовий сусідній піксель. Параметр **n** може дорівнювати 4 або 8. Наведемо приклад побудови периметра зображення сукупності кіл **circles.tif** (рис.5.2) :

```
clear;close all
```

```
I1=imread('circles.tif');
```

```
perimetr=bwperim(I1,8);
```

```
figure;
```

```
subplot(1,2,1);imshow(I1);
```

```
subplot(1,2,2);imshow(perimetr);
```

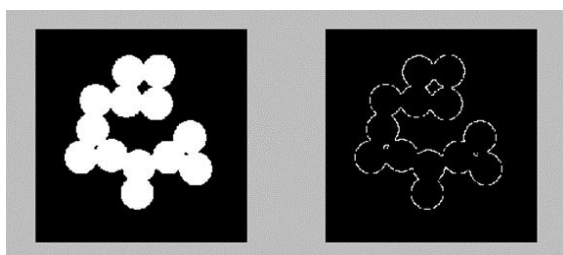


Рисунок 5.2 – Формування зображення з пікселів, розташованих по периметру

Функція **BWd=bwselect(BWs, c, r, n)** застосовується для виділення об'єктів бінарного зображення в MATLAB. Функція створює нове бінарне зображення **BWd**, яке містить об'єкт початкового зображення **BWs**, а всередині об'єкту вказується піксель **c** з координатами **(r, c)**. Піксель, з якого починається виділення об'єкта, називають *початковим*. Якщо параметри **r** та **c** є векторами однакової довжини, то із зображення **BWs** на зображення **BWd** переносяться всі об'єкти з координатами початкових пікселів **(r(k), c(k))**. Параметр **n** для всіх розглянутих функцій **bwselect** задає критерій зв'язності для пікселів об'єкту. Цей параметр може дорівнювати 4 або 8 (8 - значення за замовчуванням).

Функція **BWd=bwselect(BWs, n)** виводить зображення **BWs** на екран та надає можливість інтерактивно вказати початкові пікселі. Все об'єкти, які містять хоча б один з відмічених пікселів, переносяться із зображення **BWs** на зображення **BWd**. Координати початкових пікселів задаються одноразовим натисканням на ліву клавішу миші. Попередній заданий початковий піксель

можна видалити, якщо натиснути клавішу **Backspace** або **Delete**. Останній з початкових пікселів задається подвійним натисканням лівої клавіші миші або одноразовим натисканням правої клавіші миші. Натискання клавіші **Enter** завершує процес виділення початкових пікселів без додавання ще одного початкового пікселя. Зразу ж після натискання клавіші **Enter** або вибору останнього початкового пікселя створюється нове бінарне зображення **BWd**.

Функція **[BWd, idx]=bwselect(...)** додатково повертає вектор **idx**, який дає можливість визначити координати всіх пікселів всіх об'єктів зображення **BWd**. Координату (**r, c**) пікселя за значенням **idx(k)** можна обчислити так:

```
c=fix((idx(k)+rows-1)/rows);  
r=idx(k)-rows*(c-1),
```

де **rows** – кількість рядків у початковому зображенні.

Отримане нове зображення має формат представлення даних **uint8**.

Приклад виокремлення елементів зображення в інтерактивному режимі.

```
clear;close all
```

```
BW1 = imread('text.tif');
```

```
BW2 = bwselect(BW1,4);
```

```
imshow(BW1)
```

```
figure, imshow(BW2)
```

Після виводу початкового зображення **BW1** за допомогою курсору виділяються окремі фрагменти тексту. Після натискання **Enter** створюється нове бінарне зображення **BW2** (рис.5.3).

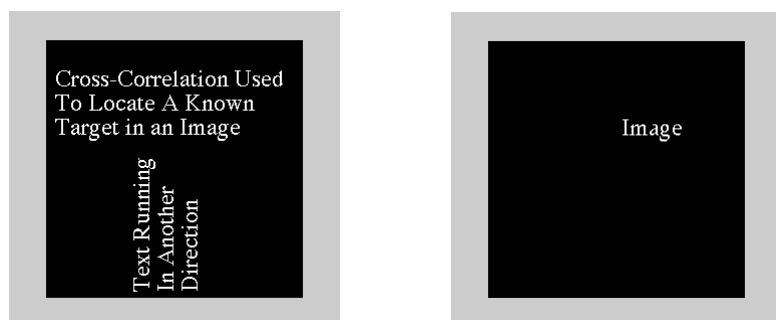


Рисунок 5.3 – Результат виокремлення об'єкту початкового зображення

При роботі із текстурованими зображеннями повне виділення елементів зображення, яке містить текстуру, можна здійснити функцією **BW=imextendedmax(I, H)** Ця функція здійснює розширене перетворення за

максимумами, які є локальними в перетворенні за **H** максимумами. Параметр **H** представлений невід'ємним скаляром. Локальні максимуми є зв'язними компонентами пікселів з однаковими значеннями яскравостей **I**, зовнішні межі яких утворені з пікселів, значення яких менші за **I**.

За замовчуванням функція **imextendedmax** використовує 8-зв'язні сусідні елементи для двовимірних зображень та 26-зв'язні для тривимірних зображень.

Нижче наведений приклад складного текстурованого зображення - мікрорентгенограми препарату кісткової тканини (рис.5.4а) та результату виділення його максимальних компонент (рис.5.4б).

clear; close all

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

```
BW = imextendedmax(I,40);
```

```
imshow(I, figure, imshow(BW))
```

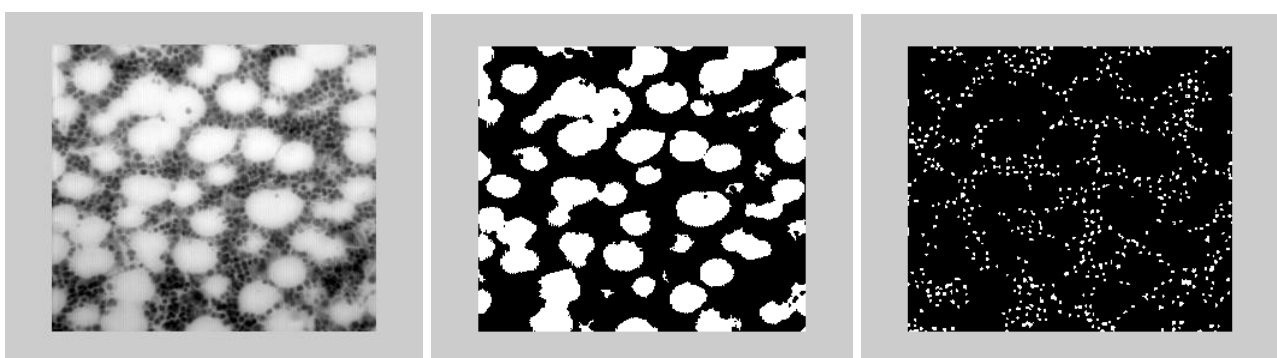
Подібну операцію перетворення зображення, тільки за мінімумами, здійснює функція **BW=imextendedmin(I,h)**, де параметр **h** представлений невід'ємним скаляром. Локальні мінімуми є зв'язними компонентами пікселів з однаковими значеннями яскравостей **I**, чий зовнішні межі утворюються пікселами, значення яких більші за **I**. Результат показаний на рис.5.4в.

clear; close all

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

```
BW = imextendedmin(I,10);
```

```
imshow(I, figure, imshow(BW))
```



а

б

в

Рисунок 5.4 – Виділення елементів текстурованого зображення:
а – початкове зображення; б – елементи з максимальною яскравістю (**imextendedmax**); в – елементи з мінімальною яскравістю (**imextendedmin**)

5.2.2 Створення морфологічних структурних елементів

Морфологічні структурні елементи застосовуються при структурному аналізі зображень. Вони можуть мати форму певних геометричних фігур.

Формування структурних елементів певного виду здійснюється функцією **SE=strel(shape, parameters)**, яка створює структурний елемент **SE**, тип якого описується в параметрі *shape*. В залежності від параметра форми *shape*, структурний елемент **strel** може мати ряд додаткових параметрів. Синтаксис написання функції залежить від форми структурного елемента. В табл.5.2 вказані всі типи форм структурних елементів, доступні для формування функцією **strel**.

Таблиця 5.2 – Типи форм структурних елементів

Плоскі структурні елементи		Неплоскі структурні елементи
'arbitrary'	'pair'	'arbitrary'
'diamond'	'periodicline'	'ball'
'disk'	'rectangle'	
'line'	'square'	
'octagon'		

Для всіх форм за виключенням форми, представленої параметром **'arbitrary'**, структурні елементи створюються за допомогою сімейства технологій, які відомі як технології *декомпозиції структурних елементів*.

При здійсненні морфологічних операцій застосовується апроксимація, ступінь якої задається параметром **N**. Апроксимація пришвидшує морфологічні операції. Апроксимація структурних елементів здійснюється, якщо параметр **N>0**. Структурні елементи, які не застосовують апроксимацію (**N=0**), є непридатними при розв'язанні деяких задач.

Наведемо приклади створення різних структурних елементів:

```
se1=strel('square', 11)    % квадрат розміром 11×11
se2=strel('line', 10, 45)  % лінія довжиною 10, кут 450
se3=strel('disk', 15)     % диск, радіус 15
se4=strel('ball', 15, 5)  % шар, радіус 15, висота 5
```

За допомогою комбінації функцій **strel** та **imerode** можна здійснити ерозію бінарного зображення з метою отримання інформації про будову та

зв'язки структурних елементів діагностичного зображення. Наведемо приклад структурних перетворень зображення сукупності кіл (рис.5.1). Результат перетворення показаний на рис.5.5.

```
clear;close all
BW=imread('circles.tif'); se=strel('rectangle',[10 16]);
subplot(1,2,1); imshow(BW);
BW1=imerode(BW,se);
subplot(1,2,2); imshow(BW1);
```

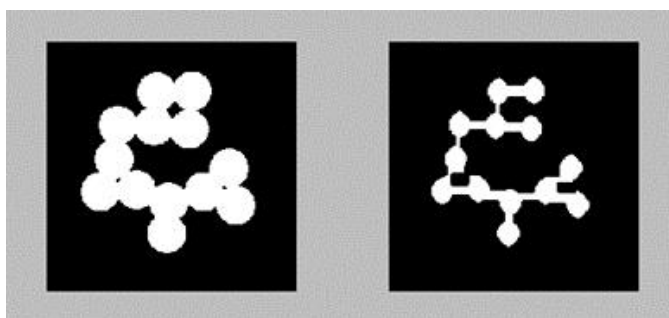


Рисунок 5.5 – Ерозія зображення з перетворенням в структурні елементи у вигляді прямокутників розміром 10×16 пікселів

Зворотну операцію – розширення елементів зображення виконує функція **IM2=imdilate(IM, SE)**. Аргумент **SE** являє собою структурний елемент об'єкту або масив структурних елементів об'єкту, який повертає функція **strel**. Якщо масив **IM** логічний, а структурні елементи плоскі, функція **imdilate** виконує бінарне нарощення; в інших випадках виконується напівтонове нарощення. Коли **SE** є масивом структурних елементів об'єкту, функція **imdilate** виконує багаторазове нарощення початкового зображення, використовуючи послідовно кожний структурний елемент **SE**. Функція **IM2=imdilate(IM, NHOOD)** виконує операцію нарощення зображення **IM**, де **NHOOD** є матрицею нулів та одиниць, яка описує окіл структурних елементів. Синтаксично це еквівалентно виразу **imdilate(IM, strel(NHOOD))**. Функція **imdilate** визначає центральний елемент околу за виразом **floor((size(NHOOD)+1)/2)**.

Нижче наведений приклад збільшення площі елементів зображення з оцінкою її відносної зміни (рис.5.6):

```
clear;close all
BW=imread('circles.tif'); se=strel('rectangle',[10 16]);
subplot(1,2,1); imshow(BW);
BW1=imdilate(BW,se);
subplot(1,2,2); imshow(BW1);
%Оцінка відносної зміни площ елементів зображення
```

```
increase=(bwarea(BW1)-bwarea(BW))/bwarea(BW);
```

```
increase
```

```
increase =  
0.4573
```

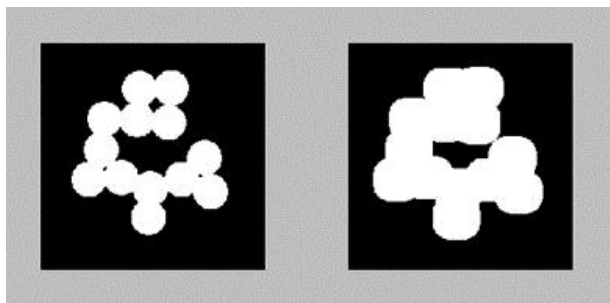


Рисунок 5.6 – Збільшення площі елементів зображення за допомогою функції **imdilate**

Більшість розглянутих морфологічних операцій, застосованих до бінарних зображень, можуть розповсюджуватись і на обробку напівтонових зображень. Розглянемо приклад операцій дилатації та ерозії (табл.5.1) для напівтонового зображення фантому голови (рис.5.7).

```
clear; close all
```

```
L=phantom(256);
```

```
figure, imshow(L);
```

```
title('start image');
```

```
se=strel('square',3);
```

```
%Операція дилатації
```

```
Ldil=imdilate(L,se);
```

```
figure, imshow(Ldil);
```

```
title('dilate image');
```

```
% Операція ерозії
```

```
Ler=imerode(L,se);
```

```
figure, imshow(Ler);
```

```
title('erode image');
```

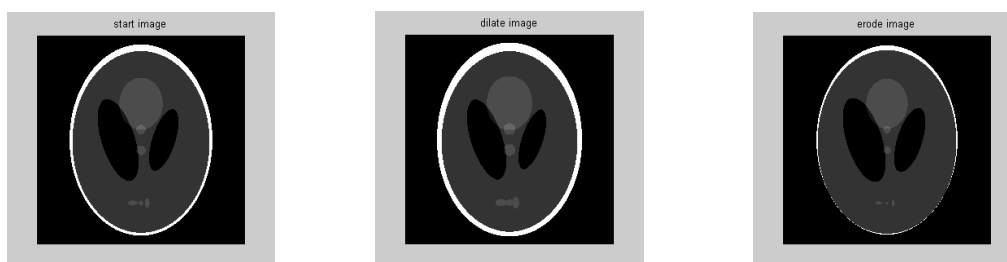


Рисунок 5.7 – Результати дилатації та ерозії зображення фантому

5.2.3 Метричні перетворення зображень

Для здійснення евклідових метричних перетворень на бінарному зображенні **BW** в MATLAB застосовується функція **D=bwdist(BW)**. Функція **bwdist** застосовує стійкий алгоритм обчислення евклідової відстані між пікселями, визначений для двовимірних масивів. Для кожного пікселя на зображенні **BW** результат обчислень визначається числом, яке є відстанню між поточним пікселем та найближчим ненульовим пікселем зображення **BW**. Функція **bwdist** застосовує метрику евклідового простору за замовчуванням. Зображення **BW** може мати довільну розмірність. Розмірність масиву **D** збігається з розміром масиву **BW**.

Функція **[D,L]=bwdist(BW)** також здійснює обчислення для найближчих сусідніх пікселів та формує результат як матрицю міток **L**, розмір якої такий, як і в масивів **BW** та **D**. Кожний елемент матриці **L** містить в собі лінійні індекси найближчих ненульових пікселів **BW**.

Функція **[D,L]=bwdist(BW, METHOD)** здійснює метричні перетворення на зображенні із застосуванням параметру **METHOD**, який визначає альтернативну просторову метрику при зображенні об'єктів (табл.5.3). Параметр **METHOD** може не вказуватись.

Таблиця 5.3 – Значення параметру **METHOD** при визначенні відстаней в двовимірному просторі

Значення	Спосіб обчислення відстані між пікселями (x1,y1) та (x2,y2)
'chessboard' “шахматна дошка”	$\max(x_1 - x_2 , y_1 - y_2)$
'cityblock' “міські квартали”	$ x_1 - x_2 + y_1 - y_2 $
'euclidean' евклідова відстань (за замовчуванням)	$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
'quasi-euclidean' квазіевклідова відстань	$ x_1 - x_2 + (\sqrt{2} - 1) y_1 - y_2 $, коли $ x_1 - x_2 > y_1 - y_2 $ $(\sqrt{2} - 1) x_1 - x_2 + y_1 - y_2 $ в інших випадках

Початкове зображення **BW** має бути числовим або логічним нерозрідженим масивом. **D** та **L** – матриці-копії розмірністю як у масиву **BW**.

Наведемо приклад евклідового метричного перетворення з визначенням відстаней між пікселями.

```
clear;close all
```

```
bw=zeros(5,5); bw(2,2)=1; bw(4,4)=1;
```

```
bw
```

```
[D,L]=bwdist(bw)
```

```
bw =
```

```
0 0 0 0 0
0 1 0 0 0
0 0 0 0 0
0 0 0 1 0
0 0 0 0 0
```

```
D =
```

```
1.4142  1.0000  1.4142  2.2361  3.1623
1.0000  0        1.0000  2.0000  2.2361
1.4142  1.0000  1.4142  1.0000  1.4142
2.2361  2.0000  1.0000  0        1.0000
3.1623  2.2361  1.4142  1.0000  1.4142
```

```
L =
```

```
7 7 7 7 7
7 7 7 7 19
7 7 7 19 19
7 7 19 19 19
7 19 19 19 19
```

Тут в матриці найближчих сусідів **L** значення 7 та 19 представляють позиції ненульових елементів за лінійною матричною індексацією.

Наведемо приклад двовимірних метричних перетворень з метою віднаходження та розмежування елементів (рис.5.8) для кожної з доступних просторових метрик (табл.5.3).

```
clear;close all
```

```
bw=zeros(200,200); bw(50,50)=1; bw(50,150)=1;
```

```
bw(150,100)=1;
```

```
D1=bwdist(bw,'euclidean');
```

```

D2=bwdist(bw,'cityblock');
D3=bwdist(bw,'chessboard');
D4=bwdist(bw,'quasi-euclidean');
figure
subplot(2,2,1), subimage(mat2gray(D1)), title('Euclidean')
%Включення режиму збереження поточного графіка та побудова контуру
зображення
hold on, imcontour(D1)
subplot(2,2,2), subimage(mat2gray(D2)), title('City block')
hold on, imcontour(D2)
subplot(2,2,3), subimage(mat2gray(D3)), title('Chessboard')
hold on, imcontour(D3)
subplot(2,2,4), subimage(mat2gray(D4)), title('Quasi-Euclidean')
hold on, imcontour(D4)

```

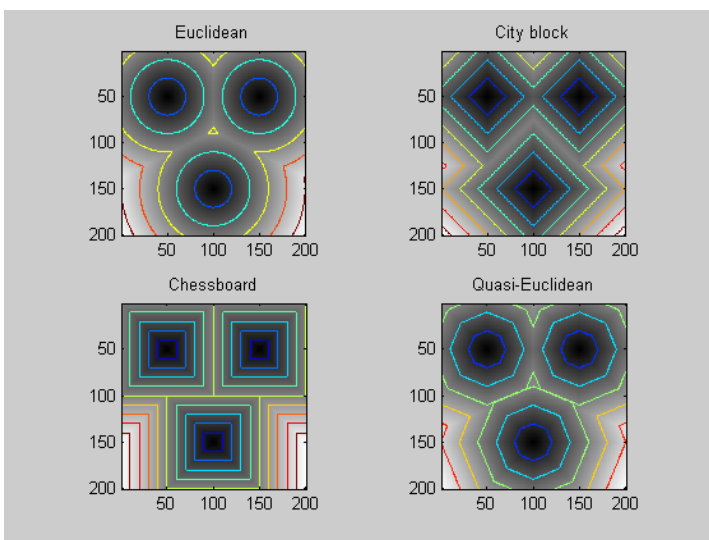


Рисунок 5.8 – Приклади двовимірних метричних перетворень

Інший спосіб метричних перетворень зображення полягає у застосуванні методу маркерного вододілу, який дозволяє за певним алгоритмом розмежувати об'єкти, які можуть бути дотичними один до одного. Метод вододілу реалізує один з удосконалених підходів до сегментації об'єктів, який дозволяє досягти більш стабільних результатів, наприклад, побудови нерозривних меж. При перетвореннях за цим методом на зображенні визначаються «водозбірні області» та «лінії вододілу» шляхом обробки локальних областей за їх яскравісними характеристиками.

Метод вододілу реалізується за допомогою функції $L=\text{watershed}(A)$, яка обчислює матрицю міток, що ідентифікує водозбірні області у початковій

матриці **A**. Матриця **A** може бути будь-якої розмірності. Елементи матриці **L** – це цілі числа, більші чи рівні 0. Елементи, помічені нулями, не належать жодній водозбірній області. Вони називаються пікселями лінії вододілу. Елементи, помічені одиницями, належать першій водозбірній області, елементи, помічені двійкою – другій тощо. Функція **L=watershed(A,CONN)** визначає зв'язність околу пікселя для реалізації алгоритму маркерного вододілу. Параметр **CONN** набуває таких значень: 4 та 8 у двовимірному випадку – 6, 18, 26. За замовчуванням реалізований алгоритм маркерного вододілу використовує 8-зв'язні околиці для двовимірних та 26-зв'язні околиці для тривимірних зображень. Наведемо приклад побудови бінарного зображення двох з'єднаних кругів, з подальшим перетворенням зображення та розмежуванням (рис.5.9).

%Створення бінарного зображення

```
center1=-10;
center2=-center1;
dist=sqrt(2*(2*center1)^2);
radius=dist/2*1.4;
lims=[floor(center1-1.2*radius) ceil(center2+1.2*radius)];
[x,y]=meshgrid(lims(1):lims(2));
bw1=sqrt((x-center1).^2+(y-center1).^2)<=radius;
bw2=sqrt((x-center2).^2+(y-center2).^2)<=radius;
bw=bw1|bw2;
figure, imshow(bw,'n'), title('bw')
```

%Обчислення метричного перетворення доданку до бінарного зображення

```
D=bwdist(~bw);
figure, imshow(D,[],'n'), title('Distance transform of ~bw')
```

%3.Доповнення метричного перетворення та перетворення значень пікселів, які не належать об'єктам на значення -Inf

```
D=-D;
D(~bw)=-Inf;
```

%Обчислення алгоритму маркерного вододілу та візуалізація результату у вигляді індексного зображення

```
L=watershed(D);
rgb=label2rgb(L,'jet',[.5 .5 .5]);
figure, imshow(rgb,'n'), title('Watershed transform of D');
```

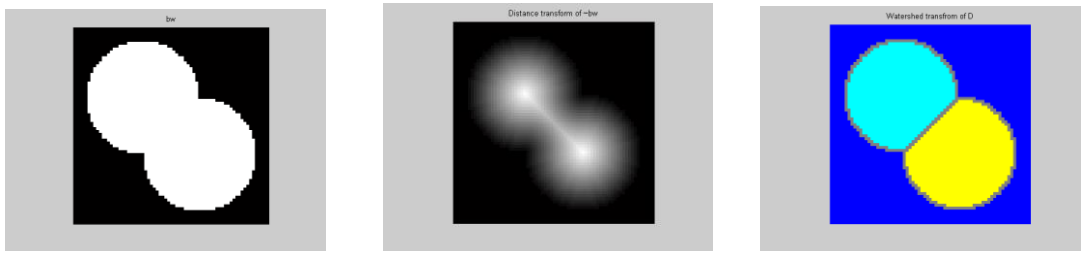


Рисунок 5.9 – Реалізація методу маркерного вододілу елементів зображення

5.3 Порядок виконання роботи

Запустити програму MATLAB. Ознайомитись з реалізацією морфологічних операцій в пакеті IPT. Виконати лабораторні завдання. Провести аналіз отриманої інформації за пп.5.3.1-5.3.3. Результати перетворень зберегти у вигляді графічних, m- та текстових файлів в особисті папки директорії **Inbox**.

5.3.1 Морфологічні операції над бінарними зображеннями

Вибрати бінарне зображення зі списку вбудованих в MATLAB зображень.

Провести морфологічні операції за допомогою функції **bwmorph**, де для параметра **operation** задати різні опції з табл.5.1.

Оцінити площу об'єктів на обраному бінарному зображенні, застосовуючи функцію **bwarea**.

Створити програму для побудови периметра бінарного зображення, отримати та задокументувати результати перетворень.

Провести виділення об'єктів бінарного зображення **'text.tif'** в інтерактивному режимі та представити отриманий результат виокремлення об'єкту.

Здійснити повне виділення елементів зображення за допомогою функцій **imextendedmax** та **imextendedmin**.

5.3.2. Створення морфологічних структурних елементів

За прикладами, наведеними в п.5.2.2, здійснити структурні перетворення (**imerode**, **imdilate**) над обраним бінарним зображенням із застосуванням різноманітних структурних елементів.

Виконати морфологічні перетворення над створеним раніше напівтоновим зображенням (фантом зрізу голови людини).

5.3.3. Метричні перетворення елементів зображення

Створити логічну матрицю, розмірність якої буде задана викладачем, розмістити в ній кілька елементів (логічних одиниць). Побудувати матрицю, яка відображає евклідову відстань між цими елементами, а також оцінити цю відстань за допомогою функції **bwdist** з використанням різних значень параметру **METHOD**.

Створити бінарне зображення, яке складається з двох кіл, із заданими значеннями радіусів та відстані між центрами цих об'єктів. Здійсніть метричні перетворення зображення методом маркерного вододілу з метою виділення межі дотику цих об'єктів.

5.4 Зміст звіту

У звіті наводяться:

- назва роботи;
- мета роботи;
- тексти завдань, програм та результати виконання операцій над зображеннями за п.п. 5.3.1 – 5.3.3;
- висновки.

5.5 Контрольні запитання

1. Які типи морфологічних перетворень застосовуються при роботі з діагностичними зображеннями?
2. Як оцінити площу об'єктів на зображенні?
3. Які методи застосовуються для виділення структур на зображенні?
4. Як і для чого виконується повне виділення елементів зображення, яке містить текстуру?
5. Як формуються морфологічні структурні елементи при аналізі зображення?
6. Для чого застосовуються метричні перетворення зображення?
7. Як можна обчислити евклідову відстань між елементами зображення?
8. В чому полягає суть методу маркерного вододілу?

ТЕМА 3. ПРИНЦИПИ ФУНКЦІОНУВАННЯ ПІДСИСТЕМИ РЕКОНСТРУКЦІЇ ТОМОГРАФІЧНОГО ЗОБРАЖЕННЯ

Підсистема реконструкції томографічного зображення (ПРТЗ) є невід'ємною складовою будь-якої медичної томографічної системи. Вона дозволяє перетворити масив проєкційних даних, отриманих при томографічному скануванні, у масив графічних даних для візуалізації зображення із застосуванням інтегральних перетворень та операцій двовимірної реконструкції.

ЛАБОРАТОРНА РОБОТА №6

«Вивчення методів реконструкції та перетворення томографічних зображень»

6.1 Мета роботи

Вивчення методів отримання проєкційних даних та реконструкції томографічного зображення із застосуванням прямого та зворотного інтегрального перетворення Радона та операцій частотної фільтрації реконструйованого зображення.

6.2 Підготовка до виконання роботи

При підготовці до ЛР необхідно опрацювати теоретичний матеріал щодо особливостей методів реконструкції та перетворення зображень, отриманих за проєкційними даними при скануванні об'єкту.

6.2.1 Реконструкція зображень за даними паралельно-променевого томографічного сканування

В комп'ютерній томографії реконструкція зображення здійснюється за допомогою інтегрального перетворення Радона, яке обчислює проєкції зображення вздовж певних напрямів.

Проєкція двовимірної функції $f(x, y)$ на вісь x' вздовж лінії L , що знаходиться на відстані s від початку начала координат (рис.6.1) являє собою лінійний інтеграл

$$R_{\theta}(x') = \int_{-a}^a f(x' \cos \theta - y' \sin \theta, x' \sin \theta + y' \cos \theta) dy',$$

де положення осей x' та y' визначається поворотом на кут θ проти годинникової стрілки:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix}.$$

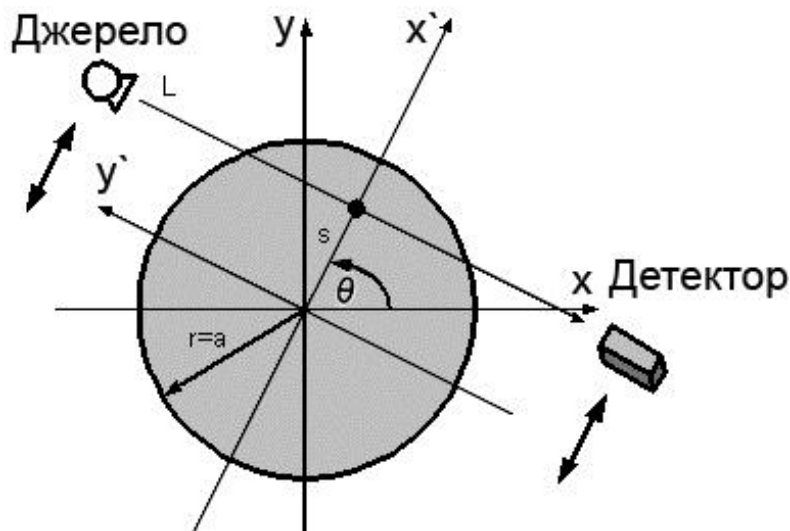


Рисунок 6.1 – Принцип реєстрації паралельно-променевих проекційних даних у томографічній системі першого покоління

В системі MATLAB початок координат в системі x',y' відповідає центральному пікселю зображення \mathbf{I} в піксельній системі координат. Центральний піксел \mathbf{I} можна визначити як $\text{floor}((\text{size}(\mathbf{I}) + 1)/2)$.

Отримання проекційних даних. Функція $\mathbf{R}=\text{radon}(\mathbf{I}, \text{theta})$ застосовує перетворення Радона до напівтонового зображення \mathbf{I} та розташовує результат у матриці проєкцій \mathbf{R} . Перетворення Радона – це процес обчислення проєкцій зображення на осі, що задаються кутами в градусах відносно горизонталі проти годинникової стрілки. Ці кути передаються в параметрі theta . Якщо theta - скаляр, то \mathbf{R} є вектором-стовпцем, який містить перетворення Радона для кута theta . Якщо theta - вектор, то \mathbf{R} є матрицею, в якій кожен стовпець є перетворенням Радона для одного з кутів, що містяться у векторі theta . Якщо при виклику функції параметр theta опущений, то до theta записуються значення кутів проєктування від 0 до 179° з кроком в 1° .

Функція $\mathbf{R}=\text{radon}(\mathbf{I}, \text{theta}, \mathbf{n})$ виконує перетворення Радона над напівтоновим зображенням \mathbf{I} . Значення кожної проєкції обчислюються в \mathbf{n} точках,

а матриця **R** має **n** рядків. Якщо при виклику функції параметр **n** опущений, то він встановлюється рівним $2*\text{ceil}(\text{norm}(\text{size}(\mathbf{I}) - \text{floor}((\text{size}(\mathbf{I}) - 1)/2) - 1)) + 3$.

Якщо додатково визначити вихідний параметр **xp**: **[R, xp]=radon(...)**, то до нього вносять значення координат, при яких обчислювались значення проекції. Значення в **k**-му рядку **R** відповідають координаті **xp(k)**. Матриця проекцій **R** може розглядатись як напівтонове або палітрове зображення та має формат представлення даних **double**.

Наведемо приклад операції з визначення проекційних даних тестового об'єкту для двох кутів проектування (0^0 та 45^0). Результат показаний на рис.6.2.

```
clear; close all
I = ones(100,100);
I(25:75, 25:75) = 0;
imshow(I)
[R,xp] = radon(I,[0 45]);
figure; plot(xp,R(:,1)); title('R_{0^o} (x\prime)')
figure; plot(xp,R(:,2)); title('R_{45^o} (x\prime)')
```

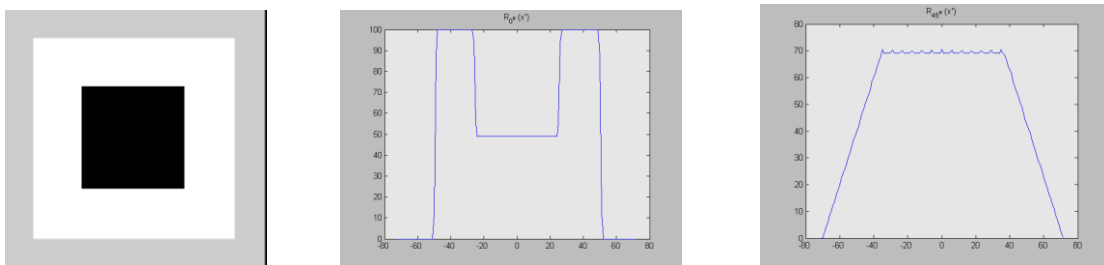


Рисунок 6.2 – Результати проектування тестового об'єкту для кутів 0^0 та 45^0

Далі наводиться приклад візуалізації проекційних даних для фантома Шеппа-Логана з 9 ракурсів, крок кута проектування 20^0 (рис.6.3).

```
clear; close all
P = phantom(256);
imshow(P)
theta = 0:20:160; [R3,xp] = radon(P,theta);
figure, imagesc(theta,xp,R3); colormap(hot); colorbar
xlabel('\theta'); ylabel('x\prime');
```

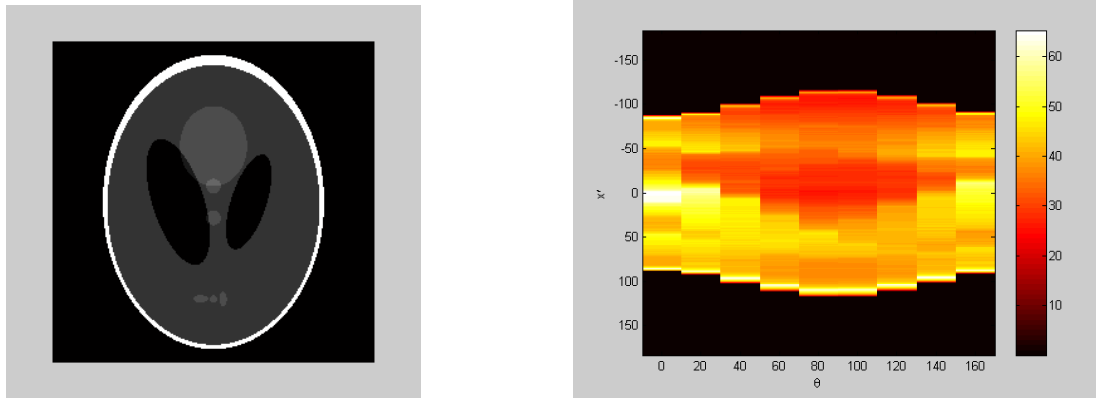



Рисунок 6.3 – Візуалізація проєкційних даних фантома зрізу голови людини

Реконструкція зображення за паралельно-променевими проєкційними даними здійснюється в MATLAB за допомогою функції $I = \text{iradon}(P, \theta)$, яка виконує реконструкцію зображення I за його проєкційними даними, що містяться в масиві P . Рядки P є даними паралельно-променевих проєкцій. В функції **iradon** центр обертання є центральною точкою проєкцій та визначається за виразом $\text{ceil}(\text{size}(P, 1)/2)$.

Функція **iradon** застосовує алгоритм фільтрації зворотних проєкцій для виконання інверсного перетворення Радона. Фільтр проєктується безпосередньо в частотній області та помножується на функцію перетворення Фур'є проєкцій. Для прискорення обчислень функції перетворення Фур'є здійснюються спеціальні перетворення над проєкціями.

Функція $I = \text{iradon}(P, \theta, \text{interp}, \text{filter}, d, n)$ містить опис параметрів, які застосовуються при зворотних перетвореннях Фур'є. Існує також можливість точного визначення деяких комбінацій останніх чотирьох аргументів. Для відсутніх параметрів функція **iradon** за замовчуванням встановлює деякі значення. Параметр **interp** визначає тип інтерполяції, який застосовується у зворотній проєкції. Доступні такі опції інтерполяції:

- 'nearest' – інтерполяція по найближчому околу;
- 'linear' - лінійна інтерполяція (за замовчуванням);
- 'spline' - сплайнова інтерполяція.

Параметр **filter** описує тип фільтра, що застосовується для частотної фільтрації. Параметр **filter** є рядком, що задає кілька стандартних фільтрів:

'**Ram-Lak**' - фільтр Рама-Лака (за замовчуванням). Частотний відклик цього фільтру дорівнює $|f|$. Одним з недоліків фільтра Рама-Лака є те, що він чутливий до шуму, присутньому на проєкціях. Через це він застосовується в комбінаціях з іншими фільтрами.

'**Shepp-Logan**' - фільтр Шепа-Логана, помножений на фільтр Рама-Лака через фазову функцію.

'**Cosine**' - косинусний фільтр, помножений на фільтр Рама-Лака через косинусну функцію.

'**Hamming**' - фільтр Хеммінга, помножений на фільтр Рама-Лака через вікно Хеммінга.

'**Hann**' - фільтр Ханна, помножений на фільтр Рама-Лака через вікно Ханна.

Параметр **d** є скаляром в діапазоні (0, 1] та застосовується для модифікації фільтра при масштабуванні по частотній осі. За замовчуванням він дорівнює 1. Коли **d** менший 1, тоді фільтр стискає частотний діапазон до [0, d], нормує частоти; всі частоти, більші значення d, прирівнюються до 0.

Параметр **n** є скаляром, що описує число рядків та стовпців у зображенні, що відтворюється. Коли параметр **n** не описаний, тоді розміри визначаються виходячи з довжини проєкцій: **n=2*floor(size(P, 1)/(2*sqrt(2)))**.

Після визначення параметра **n**, функція **iradon** відновлює зображення, не змінюючи масштабу даних. Якщо проєкції були обчислені за допомогою функції **radon**, то розміри відтвореного та початкового зображень можуть не збігатися. Функція **[I, h]=iradon(...)** повертає частотний відклик фільтра у вектор **h**. Всі початкові та результуючі аргументи повинні бути представлені в форматі **double**.

Наведемо приклад реалізації зворотного перетворення Радона для різних кутів проектування (рис.6.4).

```
clear; close all  
P = phantom(256);  
imshow(P)  
theta1 = 0:10:170; [R1,xp] = radon(P,theta1);  
theta2 = 0:5:175; [R2,xp] = radon(P,theta2);  
theta3 = 0:2:178; [R3,xp] = radon(P,theta3);  
figure, imagesc(theta3,xp,R3); colormap(hot); colorbar  
xlabel('\theta'); ylabel('x\prime');  
I1 = iradon(R1,10);  
I2 = iradon(R2,5);  
I3 = iradon(R3,2);  
imshow(I1)  
figure, imshow(I2)  
figure, imshow(I3)
```

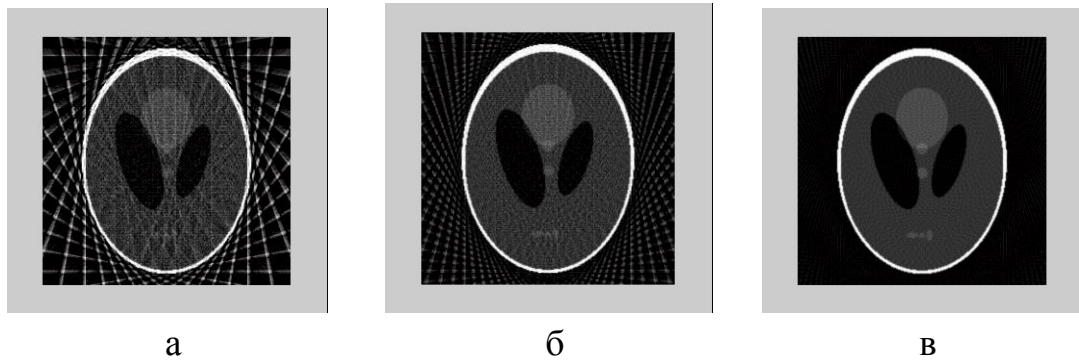


Рисунок 6.4 – Результати реконструкції фантома голови при різній кількості кутових проекційних даних: а – 18 проекцій; б – 36 проекцій; в – 90 проекцій

Результат проведених реконструкцій фантома показує, що чим менший крок кута зворотного проектування у діапазоні $0^{\circ} - 180^{\circ}$, тим більш якісне зображення отримується в результаті зворотного перетворення Радона. Найменші деталі зображення фантому (сірі еліпси) починають задовільно відтворюватись при реконструкції за 36 проекціями (рис.6.4 б). При невеликій кількості проекцій на реконструйованому зображенні помітні артефакти зворотного проектування (білі лінії). Для зменшення їх впливу на однорідність отриманого зображення фантому застосуємо процедуру фільтрації (рис.6.5).

```
clear; close all
P=phantom(128);
R=radon(P, 0:179);
I=iradon(R, 0:179, 'nearest', 'Hann');
imshow(P),figure, imshow(I)
```



Рисунок 6.5 – Реконструкція зображення фантома голови із застосуванням фільтра Ханна

6.2.2 Реконструкція зображень за проекційними даними віялово-променевого томографічного сканування

Отримання проекційних даних. Сучасна рентгенівська комп'ютерна томографія реалізує дослідження із застосуванням досить широкого віялового пучка рентгенівського випромінювання спільно з дуговим або прямолінійним детектором (сенсором). Це дозволяє одночасно реєструвати випромінювання, що пройшло крізь об'єкт, у всіх точках його проектування на матрицю детекторів (рис.6.6).

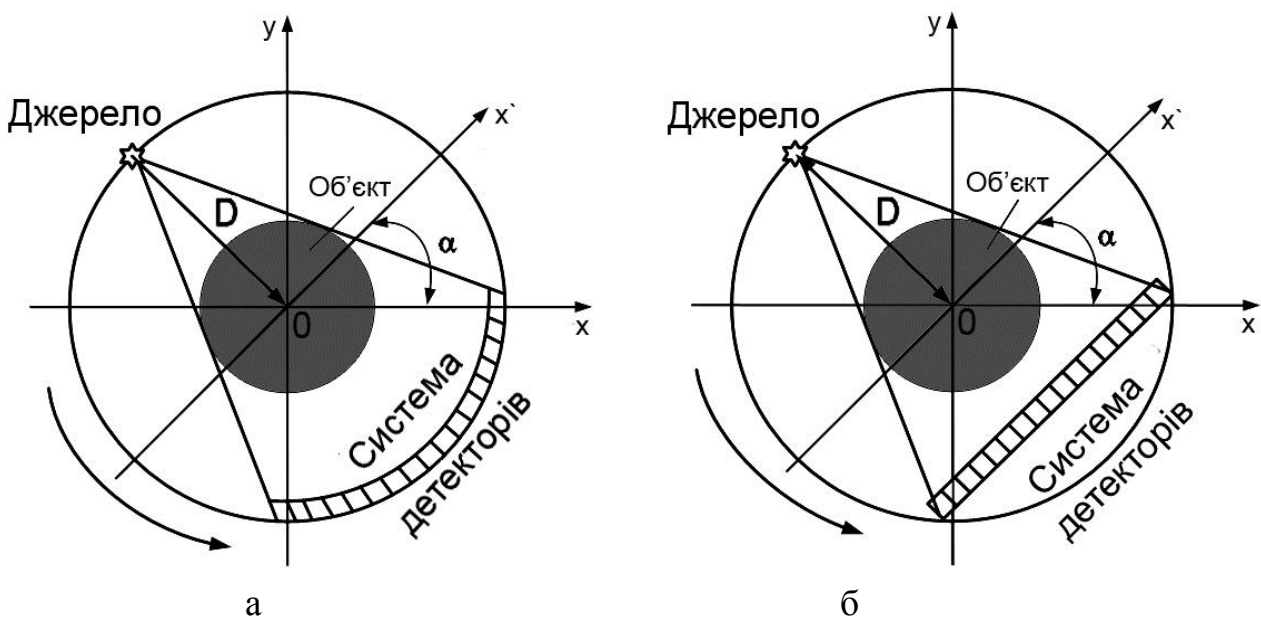


Рисунок 6.6 – Принцип реєстрації віялово-променевих проекційних даних у сучасній томографічній системі:

а – дугова система детекторів; б – лінійна система детекторів

В MATLAB існує функція $\mathbf{F}=\text{fanbeam}(\mathbf{I}, \mathbf{D})$, яка на основі зображення \mathbf{I} обчислює дані \mathbf{F} віялово-променевих перетворень (синограм). Параметр \mathbf{D} являє собою відстань (в пікселях) від вершини до центру обертання скануючої системи. Кожен стовпець \mathbf{F} містить дані про елемент (сенсор) для реєстрації віялово-променевої проекції; ці дані відповідають вибірці елементарних обертань. Розмір сенсора задає крок за кутом в кутовому просторі детектуючої системи. Кутовий оберт в просторі еквівалентний діапазону відліків [0:359].

Елементи масиву \mathbf{I} мають подаватись у форматі **double**, **logical** або іншому форматі цілих чисел. Всі інші початкові та результуючі числові дані мають представлятись у форматі **double**. Початкові дані не повинні бути розрідженими.

Функція **F=fanbeam(..., param1, val1, param1, val2, ...)** описує параметри, які визначають особливості формування віялово-променевих проєкцій; перелік цих параметрів наведений у табл.6.1. Значення, які застосовуються за замовчуванням, наведені у фігурних дужках {}.

Таблиця 6.1 – Параметри функції **fanbeam**

Параметр	Опис
'FanRotationIncrement'	Додатний дійсний скаляр, що описує приріст кута повороту при формуванні віялово-променевих проєкцій. За замовчуванням дорівнює 1.
'FanSensorGeometry'	Текстовий рядок, що описує розміщення сенсорів. 'arc' – сенсори розміщені по колу на відстані D від центра повороту; 'line' – сенсори розміщені рівномірно вздовж лінії, крайня точка розміщена на відстані D від центра повороту;
'FanSensorSpacing'	Додатний дійсний скаляр, що описує розміщення віяла променів. Інтерпретація значень залежить від установок 'FanSensorGeometry' : якщо 'arc' – значення визначається як відліки у кутовому просторі. За замовчуванням дорівнює 1; якщо 'line' – значення визначається у лінійному просторі в пікселях

Функція **[F, sensor_positions, fan_rotation_angles]=fanbeam(...)** повертає інформацію про розміщення віялово-променевих сенсорів та кути повороту. Коли опція **'FanSensorGeometry'** дорівнює **'arc'**, то параметр **sensor_positions** містить інформацію про дуговий сенсор, що подається як кількість кутових відліків. Коли опція **'FanSensorGeometry'** дорівнює **'line'**, тоді параметр **sensor_positions** містить інформацію про розміщення лінійного сенсора, яка подається у пікселях. Параметр **fan_rotation_angles** відображає кути повороту системи сканування.

Наведемо приклад отримання проєкційних даних з тестового зображення у вигляді квадрата при різному розташуванні сенсорів. Прикінцева частина програми здійснює порівняння отриманих даних з даними кутових радонівських перетворень. Результати перетворень відображуються після виконання програми.

```

clear; close all
I = ones(100); imshow(I);
D = 200; %Відстань від джерела до центра обертання системи, в пікселях
dtheta = 45; %Кут проектування
% Розрахунок віялових проєкційних даних для дугової геометрії сенсорів
[Farc,FposArcDeg,Fangles] = fanbeam(I,D,...
    'FanSensorGeometry','arc',...
    'FanRotationIncrement',dtheta);
% Перетворення кутового напрямку проектування в лінійну відстань вздовж осі x`
FposArc = D*tan(FposArcDeg*pi/180);
% Розрахунок віялових проєкційних даних для лінійної геометрії сенсорів
[Fline,FposLine] = fanbeam(I,D,...
    'FanSensorGeometry','line',...
    'FanRotationIncrement',dtheta);
% Розрахунок перетворення Радона для даної проєкції
[R,Rpos]=radon(I,Fangles);
% Відображення проєкційних даних для різної геометрії розташування сенсорів
при заданому куті проектування та їх порівняння з перетворенням Радона
figure
idx = find(Fangles==45);
plot(Rpos,R(:,idx),...
    FposArc,Farc(:,idx),...
    FposLine,Fline(:,idx))
legend('Radon','Arc','Line')

```

Наведемо приклад отримання віялово-променевих проєкційних даних (синограм) від зображення фантому зрізу голови людини (рис.6.7)

```

clear; close all
iptsetpref('ImshowAxesVisible','on')
ph = phantom(128);
imshow(ph)
%Формування віялово-променевих проєкцій фантома Шеппа-Логана для D=250
пікселів
[F,Fpos,Fangles] = fanbeam(ph,250);
figure

```

```

imshow(F,[],'XData',Fangles,'YData',Fpos,...
       'InitialMagnification','fit')
axis normal
xlabel('Rotation Angles (degrees)')
ylabel('Sensor Positions (degrees)')
colormap(hot), colorbar

```

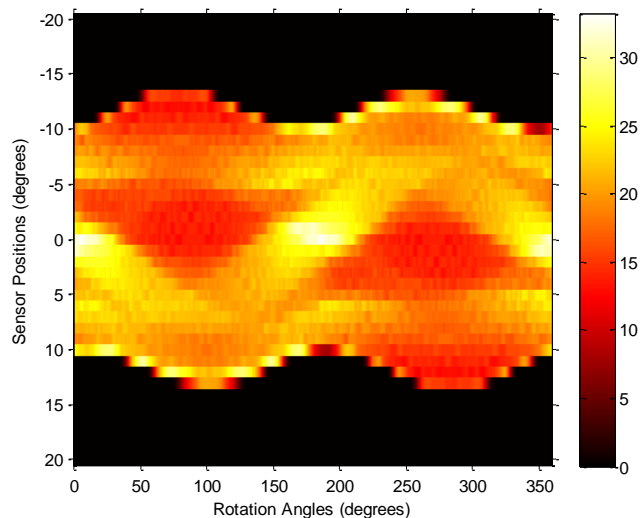
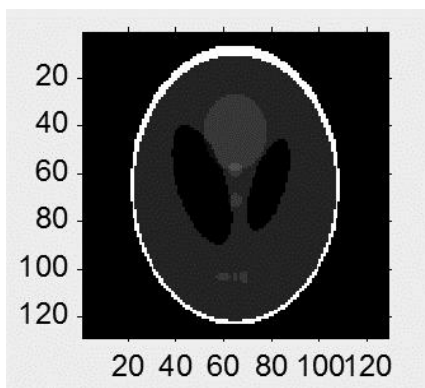


Рисунок 6.7 – Синограми фантомного зображення зрізу голови людини при віялово-променевому томографічному скануванні

Реконструкція зображення за паралельно-променевими проєкційними даними. Функція $I=ifanbeam(F, D)$ відновлює зображення I з проєкцій, які задані в двовимірному масиві F . Кожен стовпець масиву F містить дані віялово-променевих проєкцій при повороті системи на один градус.

В функції **ifanbeam** центр повороту є точкою центра проєкцій та визначається як $\text{ceil}(\text{size}(F,1)/2)$. Кути обертання мають однакові прирости та перекривають весь діапазон $[0:359]$ градусів. D є відстанню від вершини до центра обертання.

Функція $I=ifanbeam(\dots, \text{param1}, \text{val1}, \text{param2}, \text{val2}, \dots)$ містить опис параметрів, які визначають особливості реконструкції зображення за результатами віялово-променевого сканування. Опис процедур реконструкції наведений в табл.6.2. Значення за замовчуванням подаються у дужках $\{\}$. Всі початкові числові аргументи мають бути представлені в форматі **double**. Результуючі аргументи також будуть надаватись в форматі **double**.

Функція **ifanbeam** перетворює віялово-променеві дані в паралельно-променеві проекції та застосовує їх в алгоритмі фільтрації зворотних проекцій для виконання зворотного перетворення Радона. Фільтр працює повністю в частотній області.

Таблиця 6.2 – Опис параметрів функції **ifanbeam**

Параметр	Опис
'FanCoverage'	Рядок, що описує кутовий діапазон, який застосовується при обертанні випромінювача: {'cycle'} – обертання на 360^0 ; 'minimal' – мінімальний діапазон обертання, необхідний для реконструкції зображення об'єкта.
'FanRotationIncrement'	Див. опис функції fanbeam (табл.6.1)
'FanSensorGeometry'	Див. опис функції fanbeam (табл.6.1)
'FanSensorSpacing'	Див. опис функції fanbeam (табл.6.1)
'Filter'	Рядок, що описує назву фільтра. Див. функцію iradon (п.6.2.1)
'FrequencyScaling'	Скаляр в діапазоні (0,1], який змінює фільтр шляхом масштабування частотної осі. Див. функцію iradon (п.6.2.1)
'Interpolation'	Рядок, що описує метод інтерполяції. Див. функцію iradon (п.6.2.1)
'OutputSize'	Додатний скаляр, що описує кількість рядків та стовпців у реконструйованому зображенні Якщо 'OutputSize' не заданий, функція ifanbeam визначає розмір зображення автоматично. Якщо 'OutputSize' заданий, тоді функція ifanbeam реконструює області зображення, не змінюючи масштаб даних. <i>Примітка:</i> якщо проекції обчислені за допомогою функції fanbeam , реконструйоване зображення не може мати ті ж розміри, що і початкове зображення.

Наведемо приклад програми для реконструкції фантомного зображення зрізу голови людини.


```

clear; close all
ph = phantom(128);
d = 100;
F = fanbeam(ph,d);%Отримання віялово-променевих проекційних даних
I = ifanbeam(F,d);%Реконструкція зображення за віялово-променевими даними
imshow(ph), figure, imshow(F),figure,imshow(I);

```

6.2.3 Обчислення віялово-променевих проекцій за паралельно-променевими томографічними даними

Для обчислення віялово-променевих даних на основі паралельно-променевих синограмм застосовується функція **F=para2fan(P, D)**. Кожен стовпець **P** містить вибірку даних одного паралельно-променевого сенсора, яка відповідає певному куту повороту. Параметр **D** задає відстань в пікселях від центра повороту до центра сенсора. Передбачається, що сенсори відображаються за допомогою одного пікселя. Паралельно-променеве кутове обертання можливе в межах діапазону [0,180] градусів. Віялово-променеві кутові обертання здійснюються в діапазоні [0,360]. Взаємні перетворення між віялово-променевими и паралельно-променевими проекціями здійснюються з урахуванням просторового розміщення сенсорів.

Функція **I=para2fan(..., param1, val1, param2, val2, ...)** містить опис параметрів, які визначають різноманітні опції при перетворенні паралельно-променевих проекцій на віялово-променеві. Значення параметрів за замовчуванням наводяться у дужках {}. Всі початкові числові аргументи мають бути представлені в форматі **double**. Результат також буде надаватись в форматі **double**.

Таблиця 6.3 – Опис параметрів функції **para2fan**

Параметр	Опис
'FanCoverage'	Див. опис функції ifanbeam (табл.6.2)
'FanRotationIncrement'	Додатний дійсний скаляр, що описує приріст кутів повороту віялово-променевих проекцій. Якщо 'FanCoverage' має значення 'cycle', тоді 'FanRotationIncrement' дорівнює 360. Якщо 'FanRotationIncrement' не описаний, то він встановлюється таким, як при паралельно-променевих кутових переміщеннях.

Продовження табл.6.3

Параметр	Опис
'FanSensorGeometry'	Див. опис функції fanbeam (табл.6.1)
'FanSensorSpacing'	Див. опис функції fanbeam (табл.6.1) Якщо опція ' FanSensorSpacing ' не визначена, за замовчуванням встановлюється найменше значення ' ParallelSensorSpacing '. Якщо опція ' FanSensorGeometry ' визначена як ' arc ', тоді ' FanSensorSpacing ' визначається як $180/\pi * \text{ASIN}(\text{PSPACE}/D)$, де PSPACE дорівнює значенню параметра ' ParallelSensorSpacing '. Якщо опція ' FanSensorGeometry ' визначена як ' line ', то ' FanSensorSpacing ' дорівнює $D * \text{ASIN}(\text{PSPACE}/D)$.
'Interpolation'	Текстовий рядок, що описує тип інтерполяції: 'nearest' – у найближчому околі; {'linear'} – лінійна; 'spline' – шматкові кубічні сплайни; 'pchip' - шматкові кубічні ермітові сплайни; 'cubic' – аналогічна 'pchip'
'ParallelCoverage'	Текстовий рядок, що описує діапазон обертання: 'cycle' - паралельні дані перекривають 360° ; {'halfcycle'} – паралельні дані перекривають 180°
'ParallelRotation Increment'	Додатний дійсний скаляр, що описує приріст кута при паралельно-променевому скануванні, що вимірюється як кількість приростів PAR_ROT_INC в діапазоні [0,180] градусів. Значення $180/\text{PAR_ROT_INC}$ має бути цілим числом. Якщо опція ' ParallelRotationIncrement ' не визначена, прирости задаються як для віялово-променевого сканування
'ParallelSensor Spacing'	Додатний дійсний скаляр, що описує просторове розташування сенсорів (в пікселях). Діапазон розміщення сенсорів залежить від кутового розміру віялового пучка та обчислюється як $[D * \sin(\min(\text{FAN_ANGLES})), D * \sin(\max(\text{FAN_ANGLES}))]$ Якщо опція ' ParallelSensorSpacing ' не описана, розташування приймається рівномірним та залежить від кута «віяла» та його дискретизації.

Приклад обчислення віялово-променевих даних на основі паралельно-променевих синограмм, отриманих із зображення фантому (див.рис.6.7):

```

clear;close all
ph = phantom(128);
%Генерація паралельно-променевих синограм шляхом перетворення Радона
theta = 0:180;
[P,xp] = radon(ph,theta);
imshow(theta,xp,P,[],'n'), axis normal
title('Parallel-Beam Projections')
xlabel('\theta (degrees)')
ylabel('x''')
colormap(hot), colorbar
%Перетворення паралельно-променевих даних на віялово-променеві
[F,Fpos,Fangles] = para2fan(P,100);
figure, imshow(Fangles,Fpos,F,[],'n'), axis normal
title('Fan-Beam Projections')
xlabel('\theta (degrees)')
ylabel('Sensor Locations (degrees)')
colormap(hot), colorbar

```

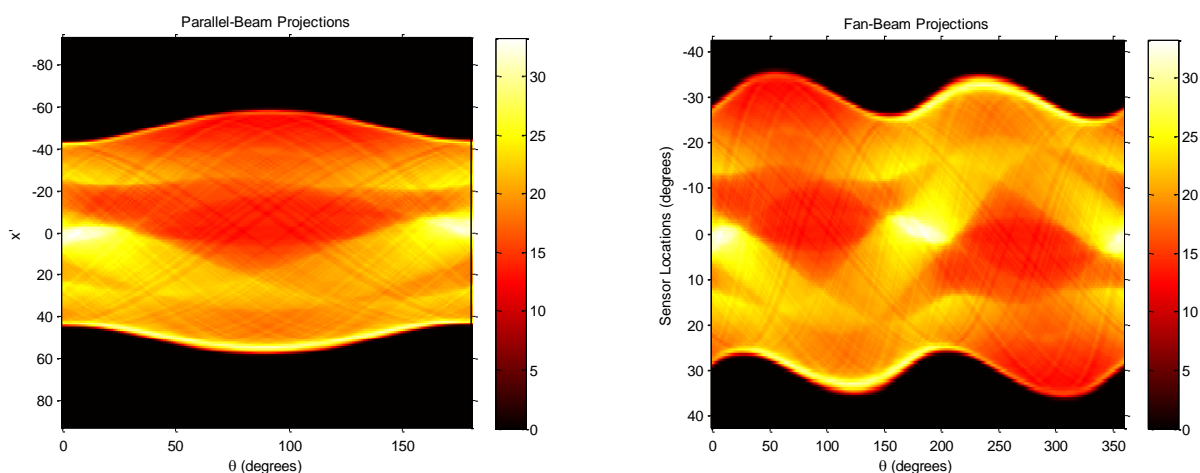


Рисунок 6.8 – Синограма паралельно-променевих проекційних даних фантому зрізу голови людини та її перетворення для віялово-променевих даних

6.2.4 Перепакування віялово-променевих проекцій у паралельно-променеві томографічні дані

У сучасних рентгенівських томографічних системах швидкість отримання результатів досліджень забезпечується поєднанням кількох складових:

застосуванням віялової системи сканування при спіральному обході тіла пацієнта, а також швидкодією систем обробки даних, реконструкції та візуалізації томограм. Перевагою віялової системи сканування є зменшення часу сканування, однак обчислювальні операції найбільш швидко здійснюються за паралельною схемою реконструкції. Тому оптимальним методом реконструкції зображень є використання віялово-променевих проєкцій з їх перетворенням таким чином, щоб вони відповідали паралельній схемі реконструкції. Потім можна скористатись алгоритмом відновлення зображення за цією паралельною схемою реконструкції. Таке перетворення називається «перепакуванням».

В MATLAB є функція $P=fan2para(F,D)$, яка обчислює паралельно-променеві проєкції (синограми) на основі даних сканування віяловим пучком F . Кожний рядок F містить дані про віялову протяжність кутів та їх переміщення. Параметр D є відстанню від вершини віяла до центру обертання.

Функція **fan2para** працює з проєкціями для відповідного кута повороту системи. Кути повороту системи сканування змінюються дискретно в діапазоні $[0,360]$ градусів. Результуючі кути обчислюються в діапазоні $[0,180]$ з аналогічними кутовими приростами. Функція $P=fan2para(..., param1, val1, param2, val2, ...)$ описує параметри, що визначають різноманітні опції перетворення для функції **fan2para** (табл.6.4).

Таблиця 6.4 – Опис параметрів функції **fan2para**

Параметр	Опис
'FanCoverage'	Див.табл. 6.2
'FanRotationIncrement'	Див.табл.6.3. За замовчуванням дорівнює 1
'FanSensorGeometry'	Див.табл.6.1
'FanSensorSpacing'	Див.табл.6.1.
'Interpolation'	Див.табл.6.3
'ParallelCoverage'	Див.табл.6.3
'ParallelRotation Increment'	Див.табл.6.3. Якщо параметр не визначений, приріст дорівнює приросту при точково-променевих кутах повороту
'ParallelSensorSpacing'	Див.табл.6.3 . Якщо параметр не визначений, розміщення буде рівномірним та мінімальне розміщення залежить від кутів віяла та вказує на ранг, що визначається на основі цих кутів

Вираз `[P,parallel_locations,parallel_rotation_angles]=fan2para(...)` повертає паралельно-променеве розташування сенсорів в `parallel_locations` та кут повороту в `parallel_rotation_angles`.

Приклад перепакуння віялово-променевих проекційних даних в паралельно-променеві:

```
clear;close all
ph = phantom(128);
theta = 0:179;
%Генерація паралельно-променевих проекцій за допомогою перетворення
Радона та перетворення їх у віялово-променеві дані
[Psynthetic,xp] = radon(ph,theta);
imshow(Psynthetic,[],...
    'XData',theta,'YData',xp,'InitialMagnification','fit')
axis normal
title('Synthetic Parallel-Beam Data')
xlabel('\theta (degrees)')
ylabel('x''')
colormap(hot), colorbar
%Отримання паралельно-променевих даних
Fsynthetic = para2fan(Psynthetic,100,'FanSensorSpacing',1);
[Precovered,Ploc,Pangles] = fan2para(Fsynthetic,100,...
    'FanSensorSpacing',1,...
    'ParallelSensorSpacing',1);
figure
imshow(Precovered,[],'XData',Pangles,...
    'YData',Ploc,'InitialMagnification','fit')
axis normal
title('Recovered Parallel-Beam Data')
xlabel('Rotation Angles (degrees)')
ylabel('Parallel Sensor Locations (pixels)')
colormap(hot), colorbar
```

Результат виконання програми показаний на рис.6.9.

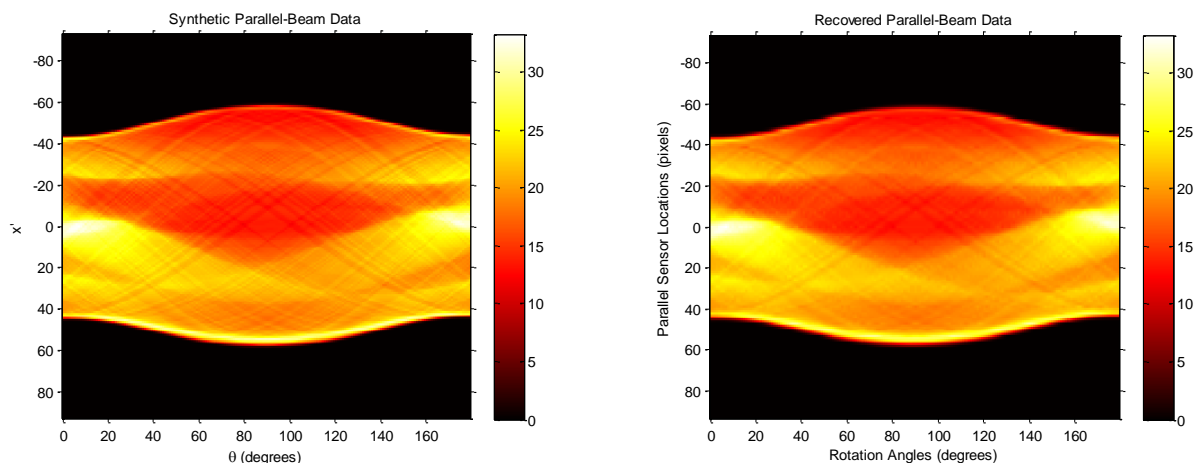


Рисунок 6.9 – Порівняння синтезованих паралельно-променевих проєкційних даних з перепакованими даними за допомогою функції **fan2para**

6.3 Порядок виконання роботи

Запустити програму MATLAB. Ознайомитись з реалізацією методів реконструкції томографічних зображень в пакеті IPT. Виконати лабораторні завдання. Провести аналіз отриманої інформації за пп.6.3.1-6.3.4. Результати перетворень зберегти у вигляді графічних, m- та текстових файлів в особисті папки директорії **Inbox**.

6.3.1 Реконструкція тестових зображень за даними паралельно-променевого сканування

Здійснити пряме перетворення Радона створеного у попередніх роботах тестового зображення. Отримати проєкційні дані для обраних кутів проектування. Проаналізувати отримані результати.

Отримати проєкційні дані для розробленого раніше фантома голови людини. Здійснити реконструкцію зображення за проєкційними даними із застосуванням функції **iradon** спільно з різними типами фільтрів та для різної кількості кутів проектування. Проаналізувати результати проведених перетворень.

6.3.2 Реконструкція тестових зображень за даними віялово-променевого сканування

Для створених у попередніх роботах тестових та фантомних зображень отримати віялово-променеві проєкційні дані при різній геометрії розташування сенсорів (лінійна, дугова). За допомогою функції **ifanbeam** виконати реконструкцію тестових зображень. Проаналізувати отримані результати.

6.3.3 Обчислення віялово-променевих проекцій за паралельно-променевими томографічними даними

За допомогою перетворення Радона згенерувати паралельно-променеві синограми для раніше розроблених тестових та фантомних зображень. Отримати їх віялово-променеві синограми.

6.3.4 Перепакування віялово-променевих проекцій у паралельно-променеві томографічні дані

За допомогою перетворення Радона згенерувати паралельно-променеві синограми для раніше розроблених тестових та фантомних зображень. Отримати їх віялово-променеві синограми. Перепакувати ці дані у паралельно-променеві. Порівняти результати цих перетворень.

Результати виконаних завдань зберегти у вигляді окремих файлів.

6.4 Зміст звіту

У звіті наводяться:

- назва роботи;
- мета роботи;
- тексти завдань, програм та результати виконання операцій над зображеннями за п.п. 6.3.1 – 6.3.4;
- висновки.

6.5 Контрольні запитання

1. Які методи реконструкції зображень застосовуються у томографії?
2. Як впливає кількість проекційних даних на якість зображень, що реконструюються?
3. Які існують методи отримання віялово-променевих даних?
4. Як виконується реконструкція зображень у віялово-променевій томографії?
5. Для чого і як реалізуються взаємні перетворення та перепакування променевих даних у сучасній комп'ютерній томографії?
6. Що таке синограма томографічного зображення?

ТЕМА 4. ПРИНЦИПИ ФУНКЦІОНУВАННЯ ПІДСИСТЕМИ ВІЗУАЛІЗАЦІЇ ЗОБРАЖЕННЯ

Підсистема візуалізації зображення (ПВЗ) є кінцевою ланкою будь-якої медичної інтроскопічної системи, що надає лікарю діагностичне зображення у наочному вигляді. Масив графічних даних для візуалізації зображення застосовується також для формування необхідних ракурсів, вибірки послідовностей зображень, побудови тривимірних реконструкцій анатомічних об'єктів тощо. Цифрові діагностичні зображення надаються для аналізу у спеціальних файлових форматах, які містять супроводжуючу текстову інформацію. Ця службова інформація має бути захищеною від стороннього втручання. Подальша обробка отриманих графічних даних з метою кількісної діагностичної оцінки має здійснюватись виключно спеціалістом.

ЛАБОРАТОРНА РОБОТА №7

«Візуалізація медичних інтроскопічних зображень у стандарті DICOM»

7.1 Мета роботи

Вивчення програмних засобів візуалізації медичних інтроскопічних зображень. Ознайомлення з процесами формування, аналізу та перетворення файлів медичних зображень у стандарті DICOM.

7.2 Підготовка до виконання роботи

При підготовці до ЛР необхідно ознайомитись з методами представлення та обробки медичної інформації в стандарті DICOM як із застосуванням можливостей пакету Image Processing Toolbox MATLAB, так і в сучасних програмах перегляду медичних зображень (в'юверах), що постачаються у складі сучасних медичних інтроскопічних систем.

7.2.1 Подання медичних зображень у стандарті DICOM

DICOM (Digital Imaging and Communications in Medicine) – це промисловий стандарт для передачі радіологічних зображень та іншої медичної інформації між комп'ютерами, який базується на стандарті Open System Interconnection (OSI), розробленому Міжнародною організацією зі стандартів ISO. Стандарт DICOM дозволяє організувати цифровий зв'язок між різноманітними діагностичними та терапевтичними системами з

використанням відкритих мереж за стандартними протоколами. Цей стандарт забезпечує інтеграцію DICOM-сумісних апаратів і систем (томографічних систем, ультразвукових сканерів, мікроскопів, робочих станцій, серверів, архівів тощо) в єдиний інформаційний простір незалежно від особливостей їх технічної реалізації конкретною фірмою-виробником. Стандарт DICOM є універсальним засобом телекомунікаційної медицини, який забезпечує територіально відокремленим клінічним підрозділам швидкий доступ до медичної інформації (рис.7.1).

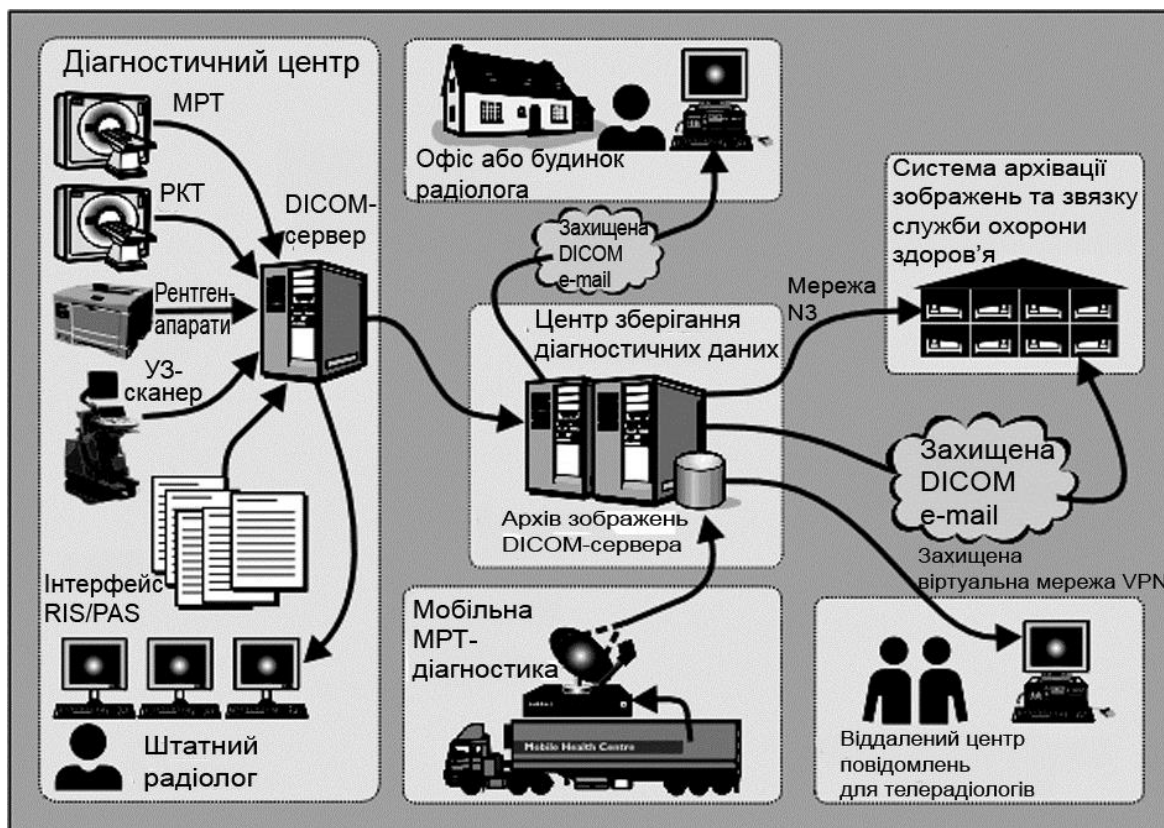


Рисунок 7.1 – Сучасна схема обміну інформацією у стандарті DICOM

Стандарт DICOM версії 3.0 (1993 р.) призначений для передачі медичних зображень, отриманих за допомогою різних методів променевої та іншої діагностики (усього 29 діагностичних методів). Стандарт DICOM надає цифрове зображення разом із супроводжуючою його інформацією: ідентифікаційні дані пацієнта, умови проведення дослідження, положення пацієнта в момент отримання зображення та ін.(рис.7.2). Складне DICOM-зображення є елементом серії зображень, отриманих при дослідженні одного й того ж пацієнта. Компонентами такого зображення можуть бути кілька кривих, які задають область інтересу, накладень растрових текстів для опису даних,

таблиць (або формул) перетворення пікселів при візуалізації, таблиць перетворення внутрішніх параметрів зберігання пікселів в універсальну зовнішню форму.

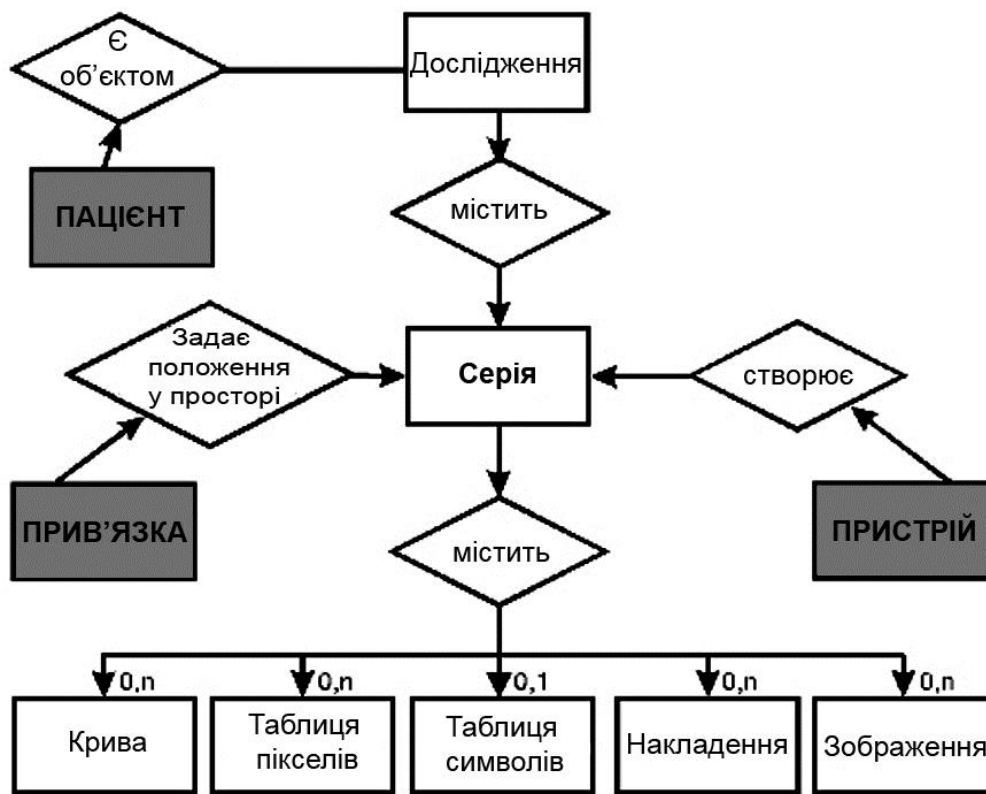


Рисунок 7.2 – Схема формування зображення в DICOM

Описова інформація в DICOM поділяється на декілька груп із закодованими даними. Наведемо основні описові параметри.

Група 0008 – Загальна інформація

(0008,0020) Study Date	<i>Дата огляду пацієнта</i>
(0008,0021) Series Date	<i>Дата серії зображень</i>
(0008,0022) Acquisition Date	<i>Дата отримання зображень</i>
(0008,0030) Study Time	<i>Час огляду пацієнта</i>
(0008,0031) Series Time	<i>Час початку запису серії зображень</i>
(0008,0032) Acquisition Time	<i>Час отримання зображень</i>
(0008,0060) Modality	<i>Вид дослідження (КТ, МРТ, УЗД...)</i>
(0008,0061) Modalities	<i>Список всіх досліджень пацієнта</i>
(0008,0082) Institution Code Sequence	<i>Код клінічної установи</i>
(0008,0090) Referring Physician's Name	<i>Ім'я лікаря, що направив пацієнта</i>
(0008,0092) Referring Physician's	<i>Адреса лікаря</i>

(0008,0094) Ref. Physician's Telephone	<i>Телефон лікаря</i>
(0008,0116) Responsible Organization	<i>Відповідальна установа</i>
(0008,1010) Station Name	<i>Назва робочої станції</i>
(0008,1030) Study Description	<i>Опис дослідження</i>
(0008,1032) Procedure Code Sequence	<i>Код процедури</i>
(0008,103E) Series Description	<i>Опис серії</i>
(0008,1050) Performing Physic. Name	<i>Імена лікуючих спеціалістів</i>
(0008,1060) Name of Phys. Reading Study	<i>Імена радіологів, які проводили опис</i>
(0008,1080) Admitting Diagn. Descript.	<i>Опис попереднього діагнозу</i>
(0008,1084) Admit. Diagn. Code Sequence	<i>Код попереднього діагнозу</i>
(0008,1090) Manufacturer's Model Name	<i>Назва моделі апарату від виробника</i>
(0008,2218) Anatomic Region Sequence	<i>Послідовність анатомічних областей</i>

Група 0010 – Дані про пацієнта

(0010, 0010) Patient's Name	<i>Ім'я пацієнта</i>
(0010, 0020) Patient's ID	<i>Ідент.дані пацієнта (паспорт тощо)</i>
(0010, 0030) Patient's Birth Date	<i>Дата народження пацієнта</i>
(0010, 0032) Patient's Birth Time	<i>Час народження пацієнта</i>
(0010, 0040) Patient's Sex	<i>Стать пацієнта</i>
(0010, 0050) Patient's Insur.Plan Code	<i>Код страховки пацієнта</i>
(0010, 0101) Patient's Prim.Language	<i>Код рідної мови пацієнта</i>
(0010, 1010) Patient's Age	<i>Вік пацієнта</i>
(0010, 1020) Patient's Size	<i>Зріст пацієнта</i>
(0010, 1030) Patient's Weight	<i>Вага пацієнта</i>
(0010, 1040) Patient's Address	<i>Адреса пацієнта</i>
(0010, 1080) Military Rank	<i>Військове звання</i>
(0010, 2000) Medical Alerts	<i>Медичні перестороги, протипоказання</i>
(0010, 2110) Contrast Allergies	<i>Алергія на контраст</i>
(0010, 2150) Country of Residence	<i>Країна проживання пацієнта</i>
(0010, 2154) Patient's Tel. Numbers	<i>Номери телефонів пацієнта</i>
(0010, 2160) Ethnic Group	<i>Етнічна група, раса</i>
(0010, 2160) Occupation	<i>Професія</i>
(0010, 2160) Smoking Status	<i>Чи курить пацієнт?</i>
(0010, 21C0) Pregnancy Status	<i>Дані про вагітність</i>
(0010, 21F0) Pat. Religious Preference	<i>Дані про релігію</i>
(0010, 4000) Patient's Comments	<i>Коментарі щодо пацієнта</i>

Група 0018 – Фізичні умови отримання зображень

(0018,0015) Body Part Examined	<i>Досліджувана частина тіла</i>
(0018,0040) Cine Rate	<i>Частота кадрів</i>
(0018,0050) Slice Thickness	<i>Товщина шару сканування</i>
(0018,0060) KVP	<i>Пікова анодна напруга, кВ</i>
(0018,0087) Magnetic Field Strength	<i>Напруженість магнітного поля</i>
(0018,1042) Contrast/Bolus Start Time	<i>Час початку введення контрасту</i>
(0018,1043) Contrast/Bolus Stop Time	<i>Час закінчення введення контрасту</i>
(0018,1044) Contrast/Bolus Total Dose	<i>Загальна кількість контрасту</i>
(0018,1045) Syringe Counts	<i>Кількість ін'єкцій</i>
(0018,1046) Contrast Flow Rate	<i>Швидкість розповсюдження контрасту</i>
(0018,1088) Heart Rate	<i>Частота серцевих скорочень</i>
(0018,1164) Imager Pixel Spacing	<i>Розбивка пікселів при відображенні</i>
(0018,1210) Convolution Kernel	<i>Ядро зортки</i>
(0018,5100) Patient Position	<i>Координати розташування пацієнта</i>
(0018,5101) View Position	<i>Координати точки огляду</i>
(0018,5210) Image Transform. Matrix	<i>Матриця перетворення зображення</i>
(0018,5212) Image Translation Vector	<i>Вектор зміщення зображення</i>

Група 0028 – Параметри зображення

(0028,0002) Samples per Pixel	<i>Кількість кольорних градацій на піксель</i>
(0028,0004) Photometric Interpretation	<i>Режим відображення кольору</i>
(0028,0008) Number of Frames	<i>Кількість кадрів</i>
(0028,0010) Rows	<i>Кількість рядків (висота)</i>
(0028,0011) Columns	<i>Кількість стовпців (ширина)</i>
(0028,0012) Planes	<i>Проекції</i>
(0028,0030) Pixel Spacing	<i>Відстань між пікселями</i>
(0028,0031) Zoom Factor	<i>Коефіцієнт збільшення</i>
(0028,0100) Bits Allocated	<i>Кількість виділених біт на піксель</i>
(0028,1050) Window Center	<i>Центр вікна яскравості</i>
(0028,1050) Window Width	<i>Ширина вікна яскравості</i>
(0028,1090) Recommended View.Mode	<i>Бажаний режим перегляду</i>
(0028,2110) Lossy Image Compression	<i>Стиснення зображення з втратами</i>

При описі DICOM-зображення загалом застосовується близько 1700 термінів, але зазвичай кількість описових параметрів не перевищує 100.

Користувачу DICOM надаються такі робочі засоби: редактор для DICOM-файлів і зображень (Editor for DICOM files and images), засоби перегляду зображень (Images viewer), довідкова система (DICOM help system), DICOM-перетворювач (DICOM converter) тощо.

7.2.2 Робота з DICOM-файлами у системі MATLAB

Зчитування метаданих з DICOM-файлу. Зчитування метаданих (даних про дані) з DICOM-файлу можна здійснити, скориставшись відповідними опціями програм для перегляду DICOM-файлів. В системі MATLAB для цього застосовується функція **dicominfo**. Ця функція повертає інформацію у вигляді MATLAB-структури, поля якої містять опис частин метаданих. Далі структуру метаданих, отриману із застосуванням **dicominfo**, можна використати при зчитуванні даних за допомогою функції **dicomread**.

Розглянемо приклад зчитування метаданих з доступного системі графічного DICOM-файлу КТ-дослідження **CT-MONO2-16-ankle.dcm**.

```
clear; close all;
```

```
info = dicominfo('CT-MONO2-16-ankle.dcm')
```

```
info =
```

```
Filename: [1x75 char]  
FileModDate: [1x20 char]  
FileSize: 525436  
Format: 'DICOM'  
FormatVersion: 3  
Width: 512  
Height: 512  
BitDepth: 16  
ColorType: 'grayscale'  
FileMetaInformationGroupLength: 192  
FileMetaInformationVersion: [2x1 uint8]  
MediaStorageSOPClassUID: [1x25 char]  
MediaStorageSOPInstanceUID: [1x50 char]  
TransferSyntaxUID: '1.2.840.10008.1.2'  
ImplementationClassUID: '1.2.840.113619.6.5'  
ImplementationVersionName: '1_2_5'  
SourceApplicationEntityTitle: 'CTN_STORAGE'  
IdentifyingGroupLength: 414  
ImageType: [1x20 char]  
SOPClassUID: [1x25 char]  
SOPInstanceUID: [1x50 char]  
StudyDate: '1993.04.30'
```

SeriesDate: '1993.04.30'
ContentDate: '1993.04.30'
StudyTime: '11:27:24'
SeriesTime: '11:27:24'
ContentTime: '11:27:24'
Modality: 'CT'
ConversionType: 'WSD'
Manufacturer: 'GE MEDICAL SYSTEMS'
InstitutionName: 'JFK IMAGING CENTER'
ReferringPhysicianName: [1x1 struct]
StationName: 'CT01OC0'
StudyDescription: 'RT ANKLE'
PhysicianReadingStudy: [1x1 struct]
OperatorName: [1x1 struct]
ManufacturerModelName: 'GENESIS_ZEUS'
PatientGroupLength: 18
PatientName: [1x1 struct]
AcquisitionGroupLength: 10
SoftwareVersion: '03'
RelationshipGroupLength: 134
StudyInstanceUID: [1x48 char]
SeriesInstanceUID: [1x48 char]
SeriesNumber: 365
InstanceNumber: 1
ImagePresentationGroupLength: 168
SamplesPerPixel: 1
PhotometricInterpretation: 'MONOCHROME2'
Rows: 512
Columns: 512
BitsAllocated: 16
BitsStored: 16
HighBit: 15
PixelRepresentation: 1
SmallestImagePixelValue: 0
PixelPaddingValue: 0
WindowCenter: 1024
WindowWidth: 4095
RescaleIntercept: -1024
RescaleSlope: 1
RescaleType: 'US'
PixelDataGroupLength: 524296

Результат виконання цієї операції показує, що функція **dicominfo** надає найдетальнішу інформацію про файли формату DICOM. Вона необхідна для максимально точної інтерпретації графічних даних з огляду на потреби та особливості конкретної клінічної процедури.

Зчитування даних зображення з DICOM-файлу. Для зчитування даних з DICOM-файлу застосовується функція **dicomread**. При застосуванні функції **dicomread** необхідно вказати назву файлу як аргумент функції.

Розглянемо приклад зчитування DICOM-файлу, доступного в MATLAB:

```
clear; close all;  
I = dicomread('CT-MONO2-16-ankle.dcm');
```

Також можна застосувати **dicomread** для отримання даних палітрового зображення:

```
clear; close all  
[X, map] = dicomread('US-PAL-8-10x-echo.dcm');  
montage(X, map);
```

Інформація про структуру багатоканального файлу з'являється після виконання даної процедури.

Перегляд зображень з DICOM-файлів. Для перегляду даних зображення, які отримані з DICOM-файлу застосовуються відомі функції додатку IPT для візуалізації зображень **imshow** або **imshow** (рис.7.3). Зауважимо, що хоча дані зображення в DICOM-файлі представлені як 16-бітні дані без знаку, існує необхідність проведення їх автомасштабування для покращення візуального сприйняття зображення.

```
info = dicominfo('CT-MONO2-16-ankle.dcm');  
I = dicomread(info);  
imshow(I, 'DisplayRange', [])
```



Рисунок 7.3 – Перегляд DICOM-файлу

Аналогічно можна переглянути файл палітрового зображення:

```
clear;close all
[X, map] = dicomread('US-PAL-8-10x-echo.dcm');
montage(X, map);
imshow('US-PAL-8-10x-echo.dcm');
```

Після виконання даних операцій MATLAB надасть графічні файли, за якими можна переглянути структурні особливості вказаних зображень.

Запис даних зображення або метаданих в DICOM-файл. Для запису даних зображення або метаданих в файл в форматі DICOM застосовується функція **dicomwrite**.

Розглянемо приклад запису зображення **I** в DICOM-файл **ankle.dcm**.

```
clear; close all;
I = dicomread('CT-MONO2-16-ankle.dcm');
dicomwrite(I,'D:\matlab\work\ankle.dcm')
```

Тут вказаний шлях запису інформації, який зазвичай вказується при виконанні лабораторної роботи.

Запис метаданих спільно з даними зображення. При записі даних зображення в DICOM-файл функція **dicomwrite** автоматично включає мінімальні установки полів метаданих з інформацією, що стосується створеного об'єкту інформації. Функція **dicomwrite** підтримує такі типи зображень:

- вторинного захвату (за замовчуванням) ;
- МРТ;
- комп'ютерна томографія.

Функція **dicomwrite** може також записувати інші типи DICOM-даних (наприклад, дані рентгенодіагностики, радіотерапії, ядерної медицини) в файл. Для отримання більш детальної інформації см. опис функції **dicomwrite** у вікні **Help**. Також в DICOM можна записувати метадані. Розглянемо приклад застосування функції **dicomwrite** для запису важливої інформації як структури метаданих в новий DICOM-файл.

```
clear;close all
info = dicominfo('CT-MONO2-16-ankle.dcm');
I = dicomread(info);
dicomwrite(I,'D:\1234\ankle.dcm',info);
```


Зазначимо, що дані, які записуються в файл не ідентичні метаданим, які отримуються при виконанні функції **dicominfo**. Запис метаданих в файл може спричинити оновлення деяких полів. Розглянемо приклад порівняння метаданих у початковому та новому файлах.

info.SOPInstanceUID %Добування інформації з поля **SOPInstanceUID** даних про універсальний ідентифікатор DICOM-файлу

ans =

1.2.840.113619.2.1.2411.1031152382.365.1.736169244

Тепер зчитуємо метадані з нового DICOM-файлу з використанням функції **dicominfo** (розглянемо поле **SOPInstanceUID**). Відмітимо різницю між записаними та зчитаними значеннями.

info2 = dicominfo('C:\matlab\tmp\ankle.dcm');

info2.SOPInstanceUID

ans =

1.2.841.113411.2.1.2411.10311244477.365.1.63874544

Видалення конфіденційної інформації з DICOM-файлу. При застосуванні DICOM-файлу в процесі тренінгу або презентації з етичних міркувань необхідно видалити конфіденційну інформацію про пацієнта. В MATLAB існує можливість видалення такої інформації з DICOM-файлу. Для цього застосовують функцію **dicomanon**, що працює з анонімним файлом. Функція **dicomanon** створює новий набір метаданих та записує їх в файл.

В DICOM-стандарті зображення можуть бути представлені серіями. При записі зображень з метаданими в DICOM-файл, функція **dicomwrite** може формувати серії зображень за замовчуванням. Для створення нової серії необхідно встановити новий DICOM-ідентифікатор в полі метаданих **SeriesInstanceUID**. Розглянемо приклад створення нової серії: при записи модифікованого зображення в DICOM-файл існує можливість модифікувати зображення та запустити нову серію. Зчитуємо зображення з DICOM-файлу в робочий простір MATLAB.

I = dicomread('CT-MONO2-16-ankle.dcm');

Для перегляду зображень застосуємо стандартні функції візуалізації - **imshow** або **imshow**. Оскільки дані DICOM-зображення представлені в 16-бітному вигляді без знаку, необхідно провести автомасштабування.

```
imshow(I,'DisplayRange',[])
```

Зчитуємо метадані з цього ж DICOM-файлу.

```
info = dicominfo('CT-MONO2-16-ankle.dcm');
```

Для ідентифікації серії зображень переглянемо значення поля SeriesInstanceUID.

```
info.SeriesInstanceUID
```

```
ans =
```

```
1.2.840.113619.2.1.2411.1031152382.365.736169244
```

Проведемо видалення всієї текстової інформації з зображення (рис.7.4). Для цього необхідно знайти мінімальні і максимальні значення пікселів на зображенні. Пікселі, за допомогою яких представлена текстова інформація, набувають максимального значення інтенсивності. Для їх видалення їм треба надати мінімальне значення яскравості.

```
max(I(:))
```

```
ans =
```

```
4080
```

```
min(I(:))
```

```
ans =
```

```
32
```

```
Imodified = I;
```

```
Imodified(Imodified == 4080) = 32;
```

Візуалізуємо оброблене зображення.

```
imshow(Imodified,'DisplayRange',[])
```

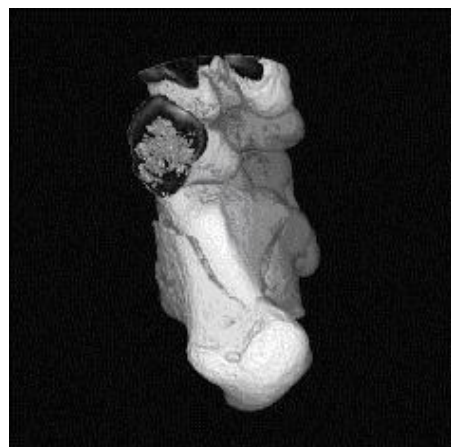


Рисунок 7.4 – Генерація нового (анонімного) зображення в форматі DICOM

Згенеруємо новий DICOM-ідентифікатор. Для цього застосуємо функцію **dicomuid**. Далі запишемо нові ідентифікатор та зображення як нову серію.

uid = dicomuid

uid =

1.3.6.1.4.1.9590.100.1.1.56461980611264497732341403390561061497

Функція **dicomuid** гарантує генерацію індивідуального ідентифікатора. Встановимо нові значення поля SeriesInstanceUID в метадані, які пов'язані з початковим DICOM-файлом.

info.SeriesInstanceUID = uid;

Запишемо модифіковане зображення в новий DICOM-файл, описавши змінену структуру метаданих як аргумент. Через те, що були встановлені нові значення SeriesInstanceUID, зображення будуть записані як частина нової серії.

dicomwrite(Imodified,'ankle_newseries.dcm',info);

Для перевірки цих операцій можна переглянути зображення та метадані поля SeriesInstanceUID в новому файлі, застосовуючи функцію **imshow**.

7.2.3 Спеціалізовані програмні засоби для роботи з DICOM-файлами

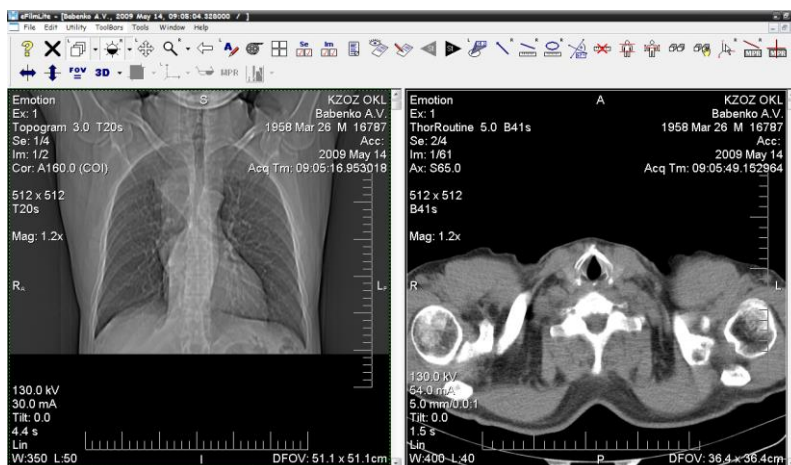
Сучасні медичні комп'ютерні системи обладнані програмним забезпеченням для формування, перегляду та передачі файлів з діагностичними зображеннями у форматі DICOM. Насамперед це програми-в'ювери, які застосовуються для візуалізації результатів томографічних досліджень.

Структура DICOM-в'юверів передбачає наявність:

- браузера для пошуку зображень (доступ до джерела зображення, ідентифікаційних даних пацієнта та даних про час та режим обстеження);
- засобів маркування анатомічних орієнтирів (лінії сканування, кути, контури, площини реконструкції тощо);
- засобів навігації та елементів управління опціями програми.
- засобів вимірювання довжин, кутів, площ, яскравості, тощо;
- засобів інтерактивного маніпулювання та корекції зображення;
- засобів, що забезпечують спеціальні режими візуалізації (зміна вікна відображення, тривимірна реконструкція, фільтрація, сегментація тощо);
- засобів архівування, роздрукування та передачі зображень.

Інтерфейс DICOM-в'юверів як правило за структурою подібний до стандартного графічного Windows-додатку: містить рядок головного меню,

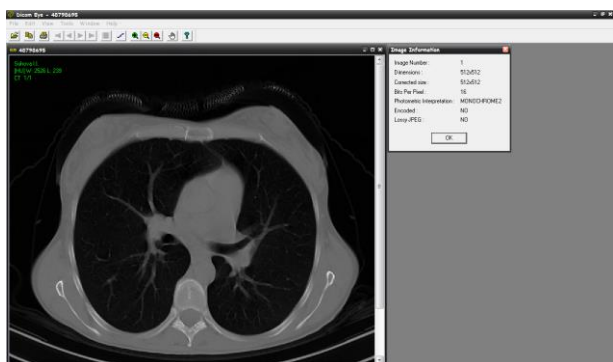
екранний простір розділений на зону зображення та «інструментальну» зону. Головні вікна деяких DICOM-в'юверів наведені на рис.7.5.



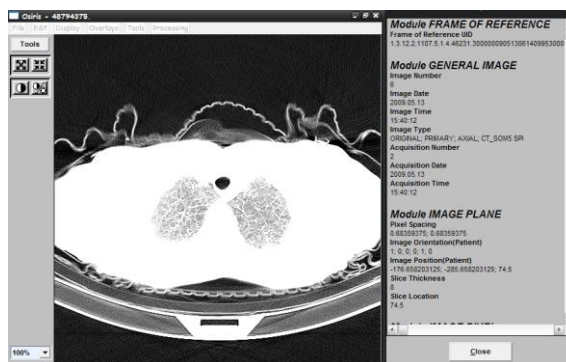
а



б



в



г

Рисунок 7.5 – Головні вікна DICOM-в'юверів:
 а – програма **eFilmLite 1.9.3**; б – програма **Siemens AG Syngo fastView**;
 в – програма **Etiam Dicom Eye 2.3.6**; г – програма **Osiris**

Інтерактивний режим роботи із зображенням реалізується з використанням маніпулятора «миша», причому у різних програмах результат дії елементів управління «миші» при обробці зображень є неоднаковим. Для уточнення опцій графічного режиму слід ознайомитись з довідковою інформацією за допомогою функції **Help**.

7.3 Порядок виконання роботи

7.3.1 Вивчення функцій ITP MATLAB для роботи з DICOM-файлами

Запустити програму MATLAB. Ознайомитись з реалізацією процедур зчитування, перетворення та запису даних зображення та метаданих DICOM-файлів у пакеті IPT. Виконати лабораторні завдання (див.п.7.2.2). Провести аналіз отриманої інформації. Результати перетворень зберегти у вигляді графічних, m- та текстових файлів в особисті папки директорії **Inbox**.

7.3.2 Вивчення структури та функціональних можливостей DICOM-в'юверів

Запустити надані для вивчення програми-в'ювери DICOM. Проаналізувати структуру, порядок роботи та функціональні можливості цих програм при роботі з медичними зображеннями. Дані внести до табл.7.1.

Таблиця 7.1 – Результати аналізу функціональних можливостей в'юверів DICOM

Опція програми	в'ювер 1	в'ювер 2	в'ювер 3
<i>Візуальний аналіз зображення:</i>	(назва)	(назва)	(назва)
Масштабування	+	+	+
Інтерактивна лупа	+	-	+
Поворот та відображення	+	+	+
Показ сукупності кадрів	+	-	-
Тривимірний показ	+	-	-
Синтез додаткових проєкцій	+	+	-
Регулювання «яскравість/контраст»	+	+	+
Вибір вікна яскравості	+	+	-
Інвертування	+	+	+
Сегментація	-	+	-
Субтракційний режим
Фільтрація			
Реконструкція			
Видалення тексту			
Додаткові опції:			

Опція програми	в'ювер 1	в'ювер 2	в'ювер 3
Кількісний аналіз зображення:			
Визначення координат та яскравості пікселя			
Вимірювання довжини			
Вимірювання кута			
Вимірювання діаметру			
Визначення параметрів у виділеній області			
Побудова профілю яскравості за напрямком			
Побудова ізоліній			
Додаткові опції:			

Провести порівняння технічних можливостей цих програм та скласти рекомендації щодо галузей застосування цих програм (перегляд, клініко-діагностичний аналіз, дослідницька робота, навчання тощо).

Результати виконаних завдань зберегти у вигляді окремих файлів.

7.4 Зміст звіту

У звіті наводяться:

- назва роботи;
- мета роботи;
- тексти завдань, програм та результати виконання операцій над зображеннями;
- результати порівняльного аналізу програм-в'юверів DICOM;
- висновки.

7.5 Контрольні запитання

1. Що таке стандарт DICOM та яке його призначення?
2. Яка структура даних про зображення у DICOM-файлі?
3. Які параметри вказуються як метадані у DICOM-файлі?
4. Які основні функції виконують програми-в'ювери DICOM?
5. Які типові опції реалізовані у програмах-в'юверах DICOM?
6. Які з відомих вам функцій IPT MATLAB застосовані у в'юверах?

ПЕРЕЛІК ПОСИЛАНЬ

1. Прэтт У. Цифровая обработка изображений. – М.: Мир, 1982. – Кн.1–312с.: Кн.2 – 480с.
2. Физика визуализации изображений в медицине. Под ред. С.Уэбба – М.: Мир, 1991.
3. В.Дьяконов. MATLAB 6: Учебный курс. СПб.: Питер, 2001. – 592 с.
4. Р.Гонсалес, Р.Вудс. Цифровая обработка изображений. – М.: Техносфера, 2005. – 1072 с.
5. Роджерс Д. Алгоритмические основы машинной графики. – М.: Мир, 1989. – 512 с.
6. Климов А.С. Форматы графических файлов. К.: НИПФ “ДиаСофт Лтд.”, 1995. – 480 с.
7. Хермен Г., Восстановление изображений по проекциям: Основы реконструктивной томографии. – М.: Мир, 1983. – 352с.
8. Бейтс Р., Мак-Доннел М., Восстановление и реконструкция изображений. М.: Мир, 1989. – 336с.
9. Натгерер Ф. Математические аспекты компьютерной томографии. – М.: Мир, 1990. – 288 с.
10. Тихонов А.Н., Арсенин В.Я., Тимонов А.А. Математические задачи компьютерной томографии. – М.: Наука, 1987. – 150 с.
11. Фор А. Восприятие и распознавание образов. – М.: Машиностроение, 1989. – 272с.

МЕТОДИЧНІ ВКАЗІВКИ
ДО ЛАБОРАТОРНИХ РОБІТ З ДИСЦИПЛІН
«ОСНОВИ ТЕОРІЇ ІНТРОСКОПІЇ», «ТЕОРІЯ ТА ТЕХНІКА ІНТРОСКОПІЇ»
для студентів денної форми навчання спеціальностей
7.091002, 8.091002 – біотехнічні та медичні апарати і системи,
7.090804, 8.090804 – фізична та біомедична електроніка
та напряму 6.051402 – біомедична інженерія

Упорядники: ГОЛОВЕНКО Валерій Михайлович
АВЕР'ЯНОВА Лілія Олександрівна
ЄВДОКИМЕНКО Сергій Олександрович

Відповідальний випусковий А.І.Бих

Редактор

План 2010 поз.

Підп. до друку 00.00.00. Формат 60x84 1/16. Спосіб друку – ризографія

Умов. друк. арк. Облік вид. арк. Тираж прим.

Зам. № Ціна договірна

ХНУРЕ Україна. 61166, Харків, просп. Леніна, 14

Надруковано в учбово-виробничому
видавничо-поліграфічному центрі ХНУРЕ
61166, Харків, просп. Леніна, 14