

Міністерство освіти і науки України
Вінницький національний технічний університет

**Методичні вказівки
до виконання лабораторних робіт
з дисципліни
«Обробка біомедичних зображень та реконструкція
об'єктів»
для здобувачів вищої освіти за спеціальністю
163 – Біомедична інженерія**



Co-funded by the
Erasmus+ Programme
of the European Union



Розроблено в рамках проекту “Erasmus+ (CBHE) BioArt “Інноваційна мультидисциплінарна освітня програма зі штучних імплантів для біоінженерії для бакалаврів та магістрів”

Developed in the frame of project “Erasmus+ (CBHE) BioArt “Innovative Multidisciplinary Curriculum in Artificial Implants for Bio-Engineering BSc / MSc Degrees” (586114-EPP- 1-2017- 1-ES- EPPKA2-CBHE- JP).

Вінниця
ВНТУ
2020

Рекомендовано - кафедрою біомедичної інженерії Вінницького національного технічного університету Міністерства освіти і науки України (протокол засідання кафедри БМІ № 2 від 10.09.2019 р., оновлено, розглянуто зі змінами на засіданні кафедри БМІ № 2 від 09.09.2020 р)

Методичні вказівки до виконання лабораторних робіт з дисципліни «Обробка біомедичних зображень та реконструкція об'єктів» для студентів спеціальності 163 - Біомедична інженерія освітня програма Інтелектуальні штучні імпланти та медичні апарати в біоінженерії / Уклад. Л.Г. Коваль. – Вінниця : ВНТУ, 2020.

Методичні вказівки до виконання лабораторних робіт призначений для студентів, які навчаються за II рівнем (магістерським) вищої освіти зі спеціальності 163 - Біомедична інженерія, освітня програма - Інтелектуальні штучні імпланти та медичні апарати в біоінженерії.

Лабораторна 1 Обробка медичних зображень у фотошоп

Програмний пакет Adobe Photoshop надає широкі можливості для обробки зображень. Розглянемо деякі з них, які зручно використовувати при обробці медичних зображень. Потрібно відзначити, що цей програмний пакет доцільно застосовувати при невеликій кількості зображень, наприклад на початкових стадіях експериментів, коли необхідно визначитися з процедурами обробки зображень.

1.1 Зміна гистограми

Корекція гистограм дозволяє підвищувати якість монохромних зображень.

Гистограмою зображення називають деяку дискретну функцію, визначену в діапазоні значень $[0; 2^{b_{pp}}]$, де b_{pp} - кількість біт, що відводиться для кодування яскравості одного пікселя.

Зміна закону розподілу яскравості пікселів шляхом зміни гистограми в деяких випадках призводить до поліпшення видимості окремих фрагментів зображення (рисунки 1.1, 1.2).

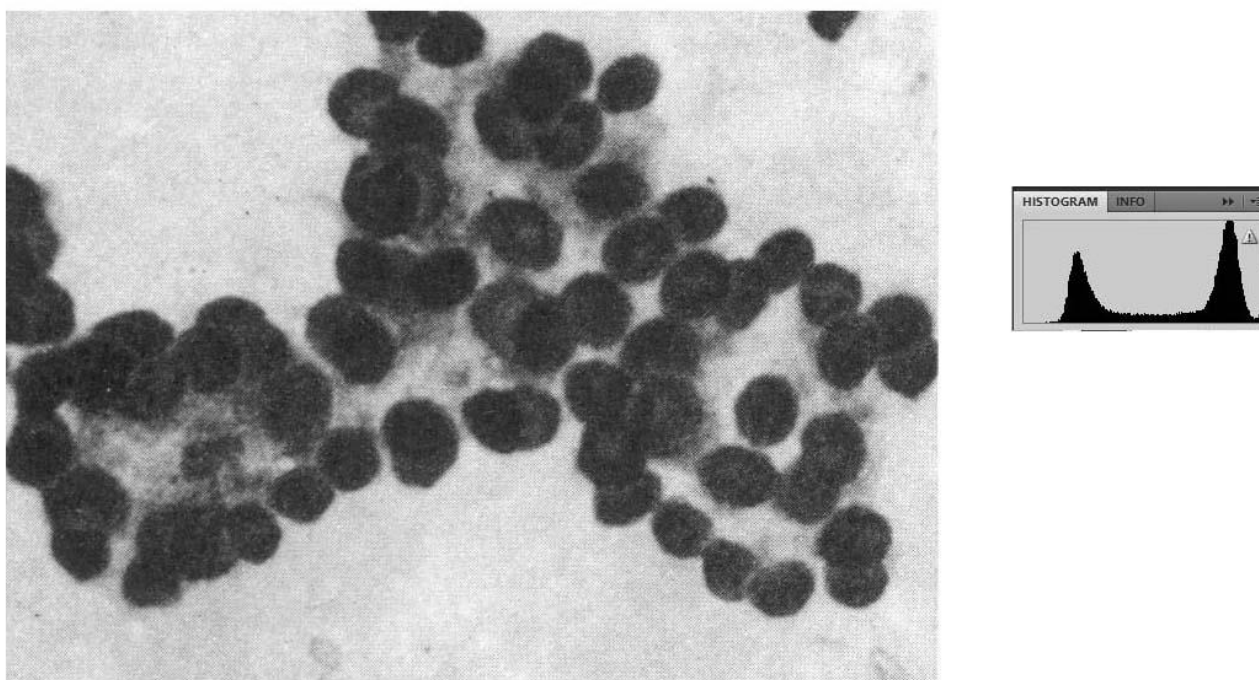


Рисунок 1.1 – Початкове зображення і його гистограма

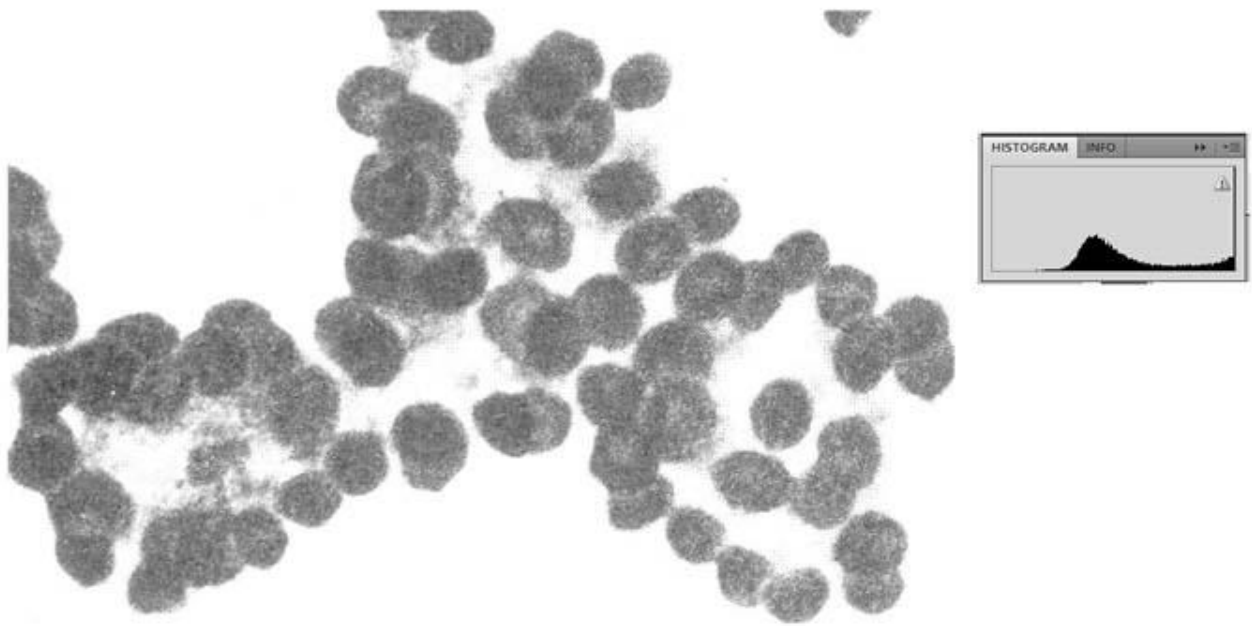


Рисунок 1.2 – Зображення після зміни гистограми

1.2 Перетворення зображення в негатив

Для виконання цієї процедури необхідно одночасне натискання клавіш Ctrl + I. Приклад такого перетворення показаний на рисунку 1.3.

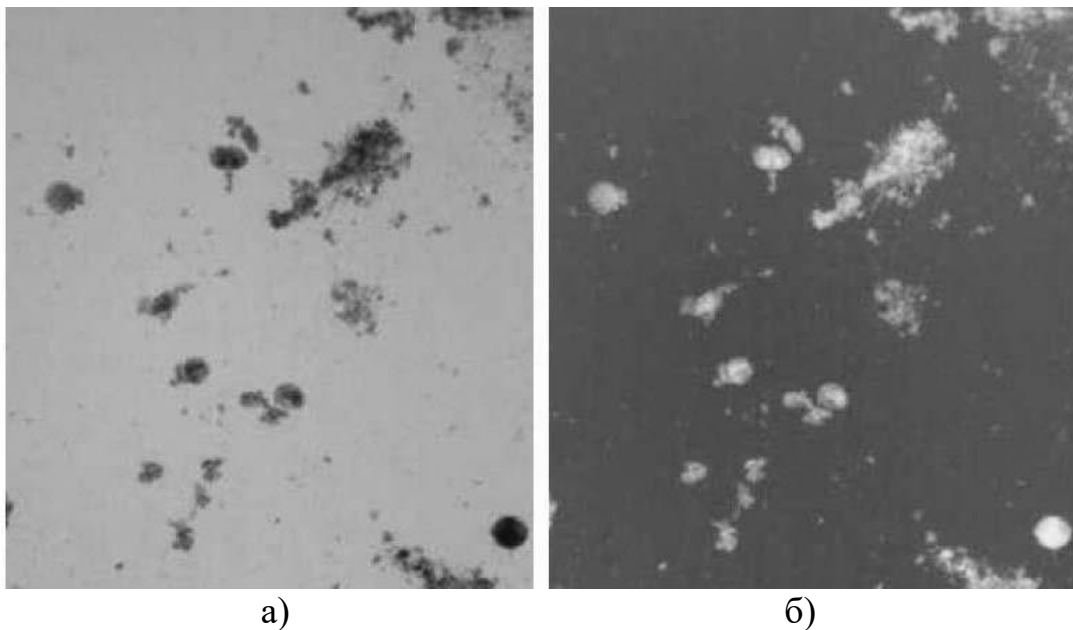


Рисунок 1.3 – Перетворення зображення в негатив: а) Початкове зображення; б) зображення після обробки

1.3 Зміна контрасту

Для виконання цієї процедури необхідно змінити властивості зображення:

Image – Adjustments-Brightness/Contrast. Приклад такого перетворення показаний на рис.1.4.



а)

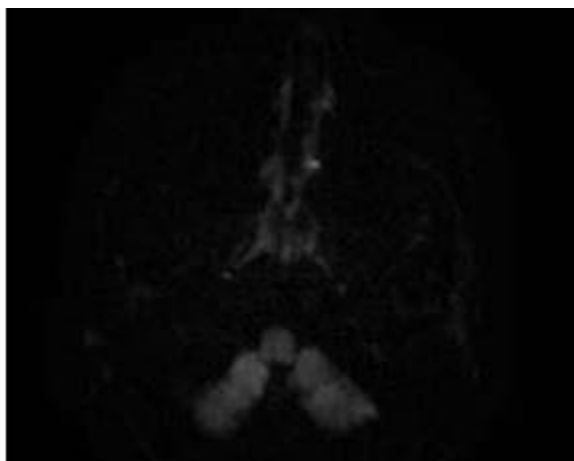


б)

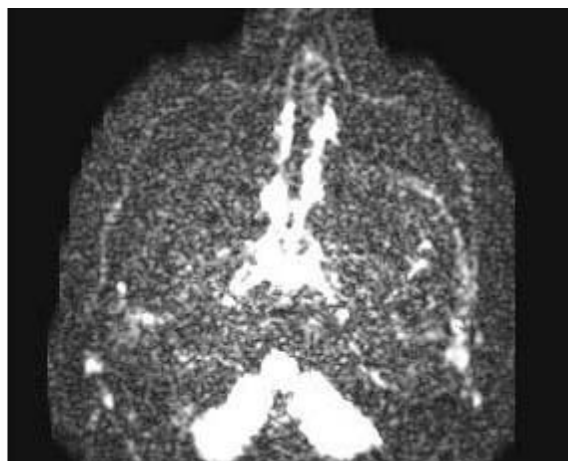
Рисунок 1.4 – Зміна контрасту: а) Початкове зображення; б) зображення після обробки

1.4 Зміна яскравості зображення

Зміна яскравості зображення виконується за тим же сценарієм, що і зміна контрастності. Приклади приведені на рисунках 1.5.,1.6.

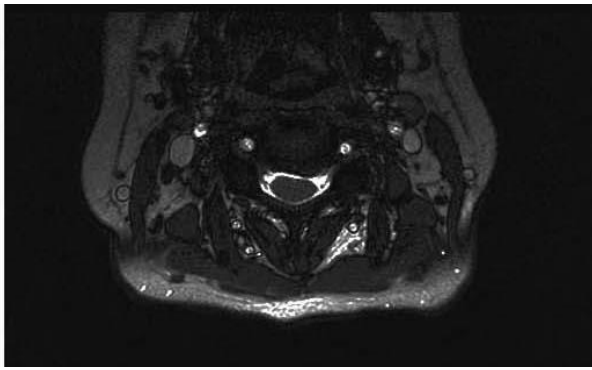


а)



б)

Рисунок 1.5 – Зміна яскравості: а) початкове зображення; б) зображення після обробки



а)

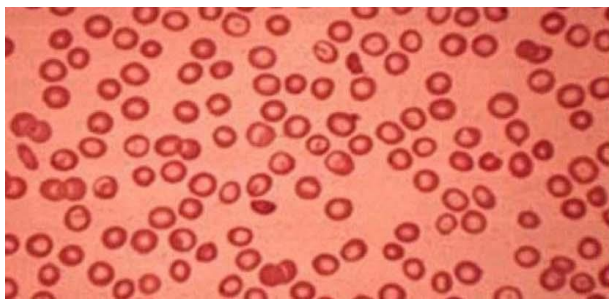


б)

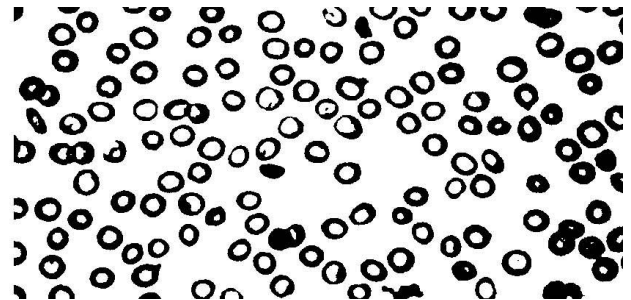
Рисунок 1.6 – Приклад зміни яскравості: а) початкове зображення; б) зображення після обробки

1.5 Бінаризація зображень

Бінаризація зображень в Adobe Photoshop проводиться одним методом - відсіканням за порогом яскравості. Користувач може задати поріг яскравості в діапазоні від 1 до 255 (рис.1.7, 1.8).

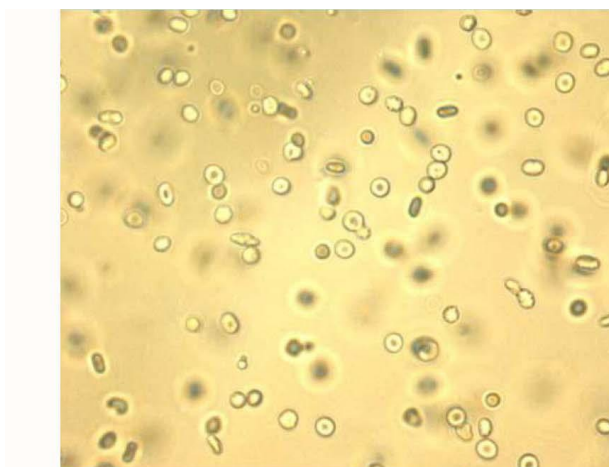


а)

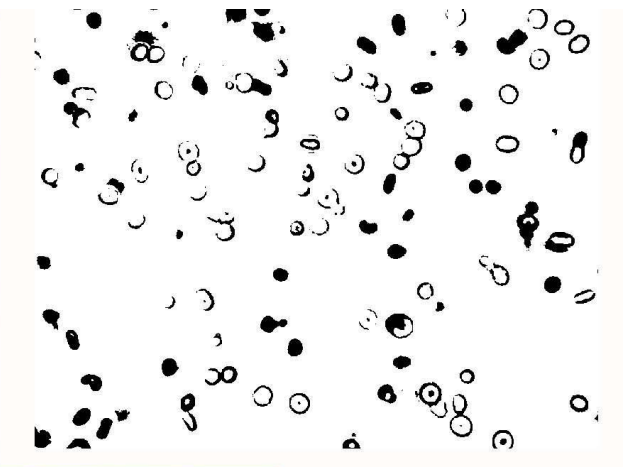


б)

Рисунок 1.7 – Бінаризація зображення: а) початкове зображення; б) зображення після бінаризації з порогом 112



а)



б)

Рисунок 1.8 – Приклад бінаризації зображення: а) початкове зображення; б) зображення після бінаризації з порогом 190

1.6 Виділення кольорових елементів на зображеннях

Дуже часто при проведенні різних медичних досліджень використовуються різні барвники, які дозволяють на зображеннях різних тканин виявити фрагменти з патологією. Пакет Adobe Photoshop дозволяє виконати виділення забарвлених елементів від решти зображення. При цьому дану процедуру можна автоматизувати шляхом створення відповідних скриптів.

Найбільш зручним способом є переклад зображення з системи RGB в багатоканальну систему кольорового зображення. Після цього розглядаючи окремо різні канали, можна вибрати той, який містить відповідні об'єкти інтересу (рисунок 1.9, 1.10).

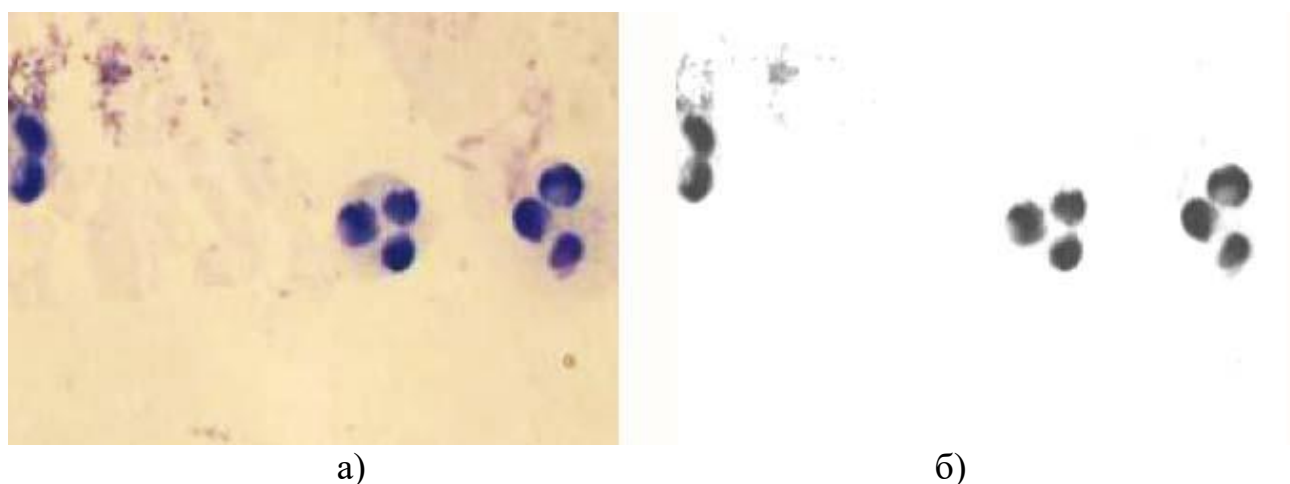


Рисунок 1.9 – Виділення кольорових елементів на зображенні: а) початкове зображення еозинофілів в осаді сечі; б) канал фрагментів синього кольору

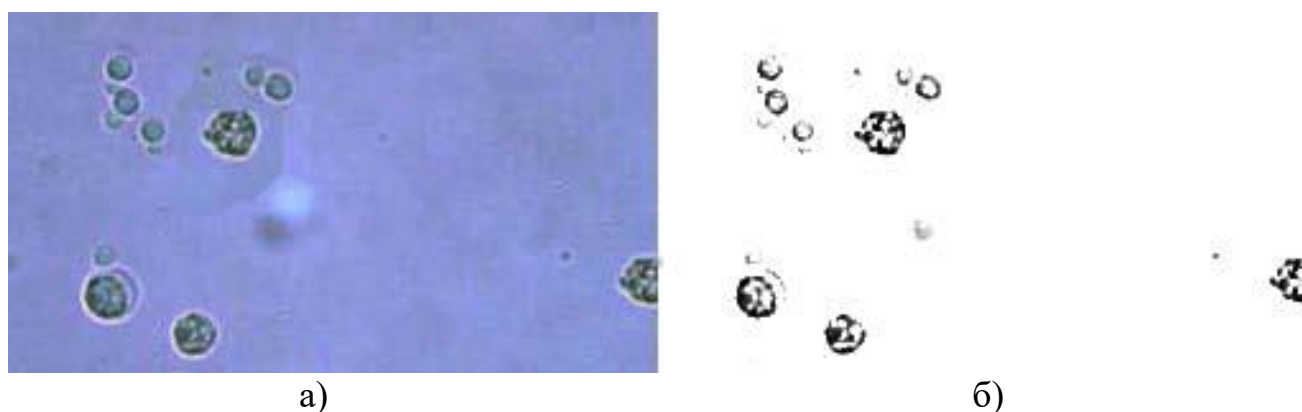


Рисунок 1.10 – Виділення кольорових елементів (еритроцитів) на зображенні: а) початкове зображення; в) канал Cyan

1.7 Визначення значення пікселя

Визначення такого параметра пікселя, як колір, дуже важлива процедура при аналізі зображень. У Adobe Photoshop для цього необхідно при перегляді

зображення натиснути клавішу F8. При цьому на екрані з'явиться вікно з координатами пікселя і параметрами його кольорових компонентів (рисунок 1.11).

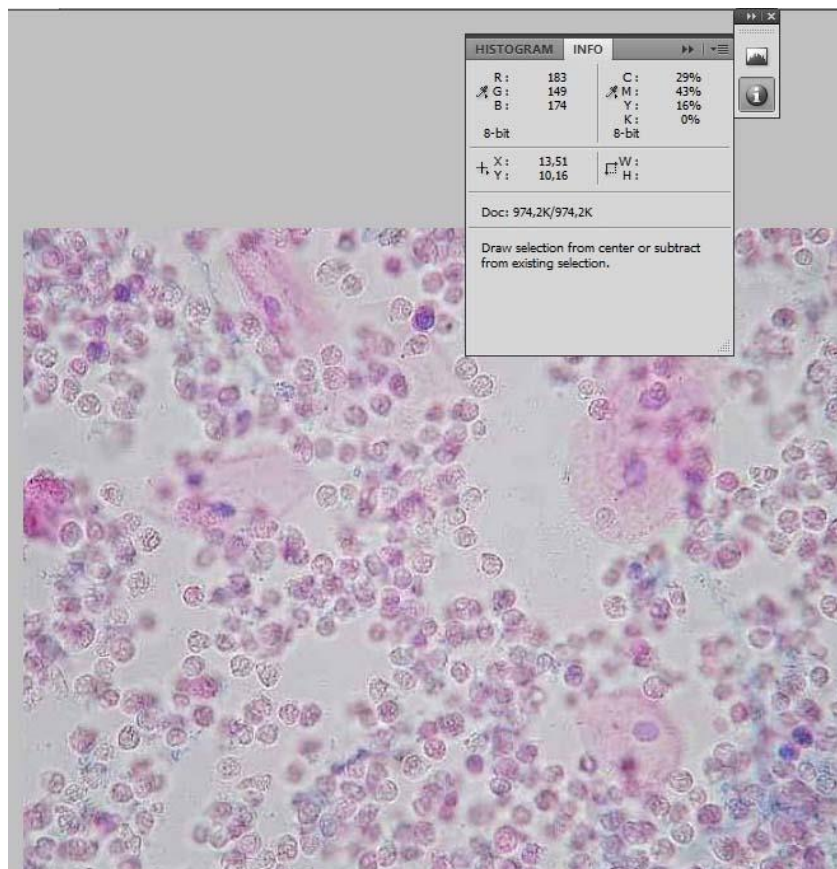


Рисунок 1.11 – Визначення параметрів пікселя у Adobe Photoshop

Недоліком Adobe Photoshop є те, що операції по обробці зображення проводяться вручну, що вимагає великих витрат за часом і підвищує ймовірність помилки через людський фактор.

ЛАБОРАТОРНА 2 DICOM VIEWER. ОСОБЛИВОСТІ ВІЗУАЛІЗАЦІЇ МЕДИЧНИХ ЗОБРАЖЕНЬ.

Для роботи з даними формату DICOM розроблена спеціальна мова програмування Interactive Data Language (IDL), яка містить додатковий модуль служби DICOM Network Services. Служба має ряд сервісів з графічним інтерфейсом користувача (рисунки 2.1, 2.2), що дозволяють конфігурувати мережеву інфраструктуру, запитувати і зберігати дані DICOM.

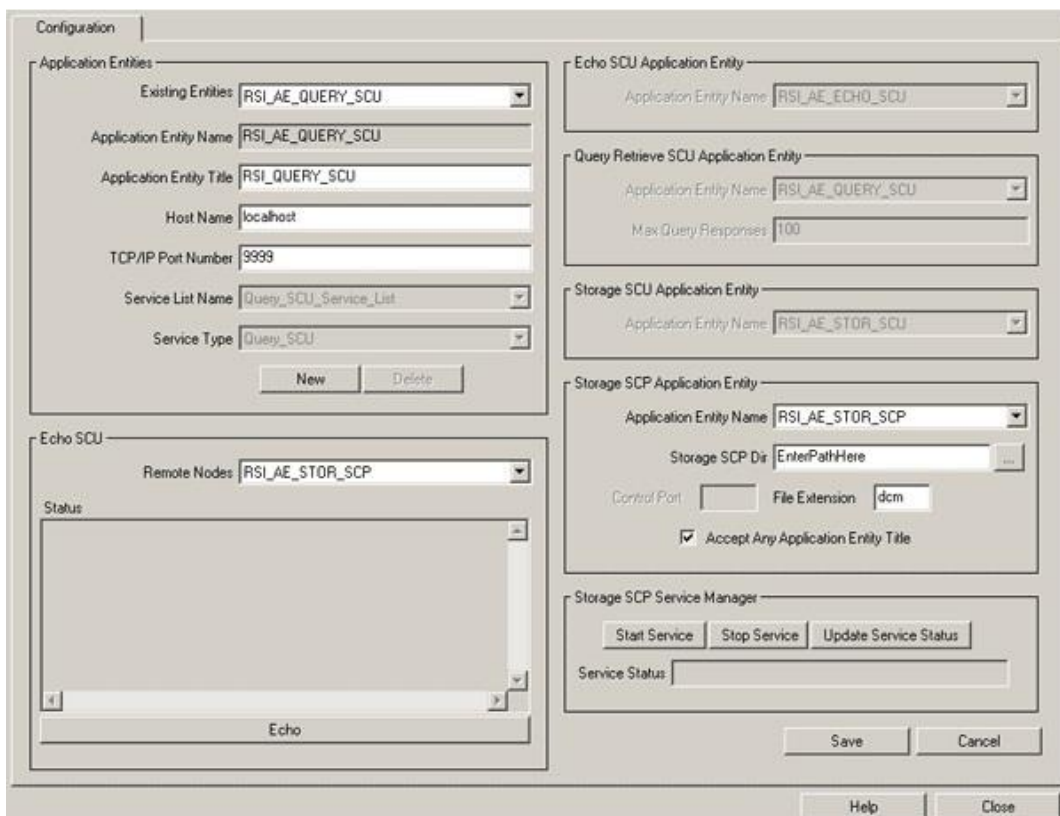


Рисунок 2.1 – Вид графічного інтерфейсу конфігурації DICOM Network Services

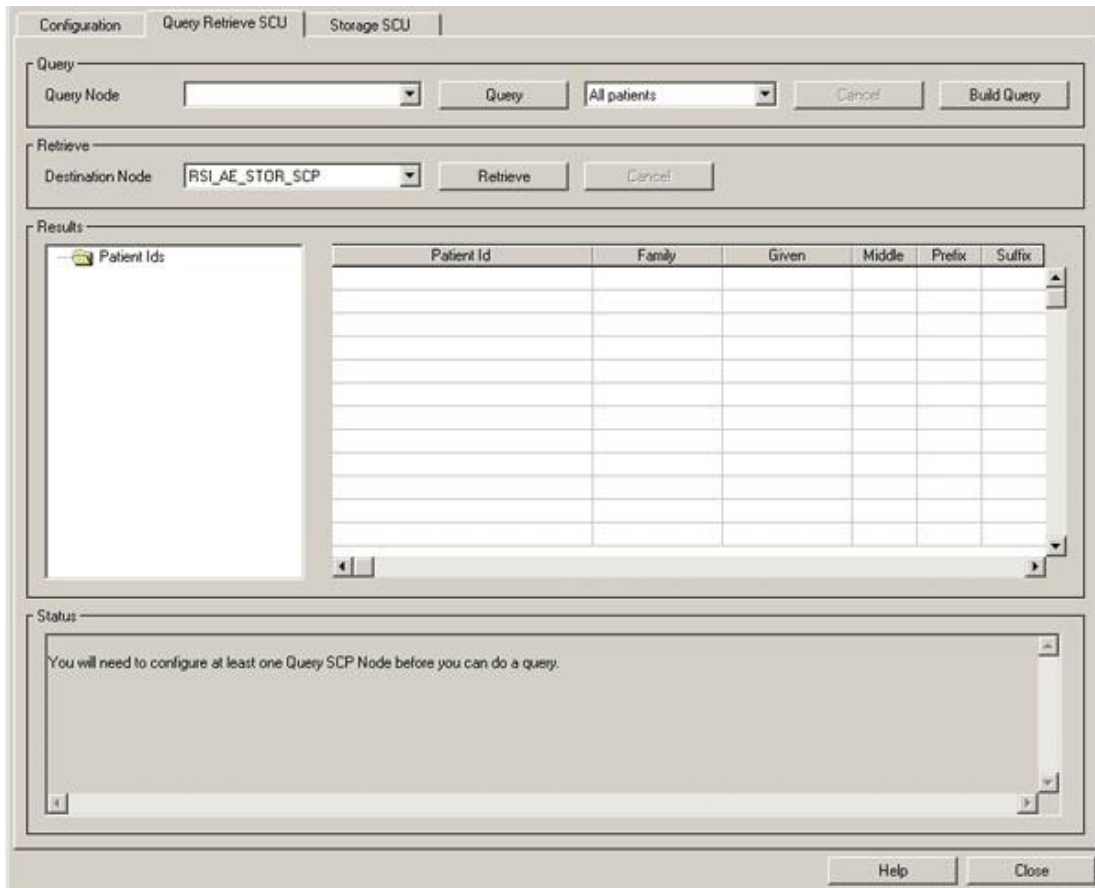


Рисунок 2.2 – Приклад запиту даних за допомогою DICOM Network Services

Крім цього є різні програми для роботи з файлами DICOM. Розглянемо деякі з них.

2.1 Rubo DICOM Viewer

Viewer є професійним засобом перегляду на комп'ютері файлів формату DICOM:

- Підтримка різних форматів стиснення зображень:
- Стиснення JPEG з втратами 8,12,16 біт
- Стиснення JPEG без втрат 8,12,16 до 32 біт
- jpeg 2000
- Можливість виконати сортування по пацієнтам, дослідженням.
- Можливість обробки послідовності серцевих скорочень (більше 1000 зображень).
- Підтримка показу ангиограмм.

Приклад робочого вікна програми показано на рисунках 2.3, 2.4.

Передбачено збереження зображень в різних форматах: (BMP), TIFF, JPEG, JPEG2000, RAW або DICOM, а також у вигляді фільмів у форматі DICOM, AVI, Windows Media Format (WMV) або анімованого GIF.

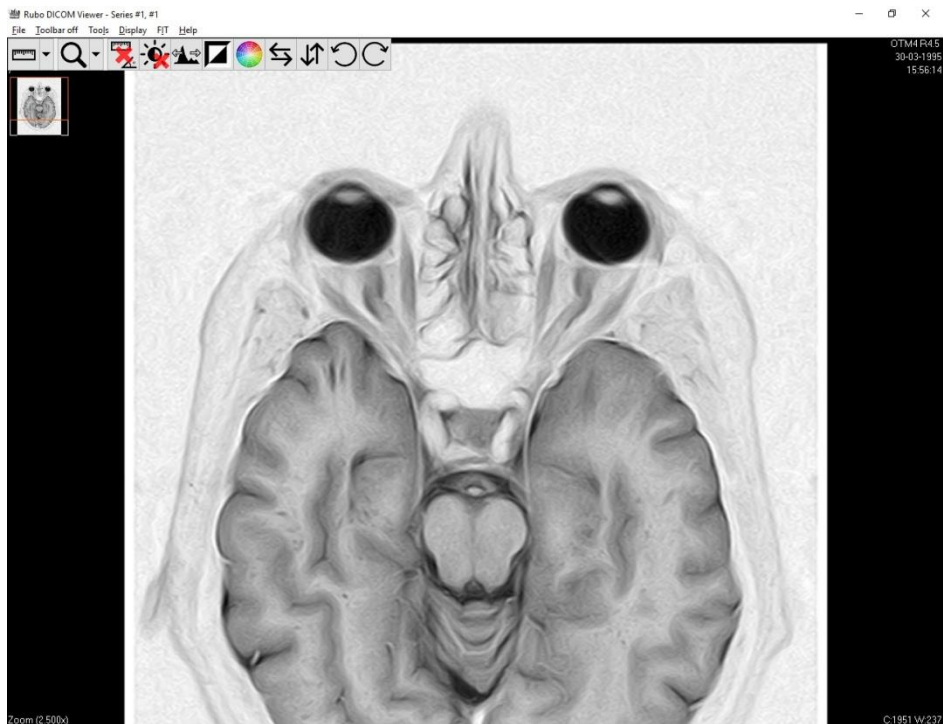


Рисунок 2.3 – Приклад файла МРТ в Rubo DICOM Viewer

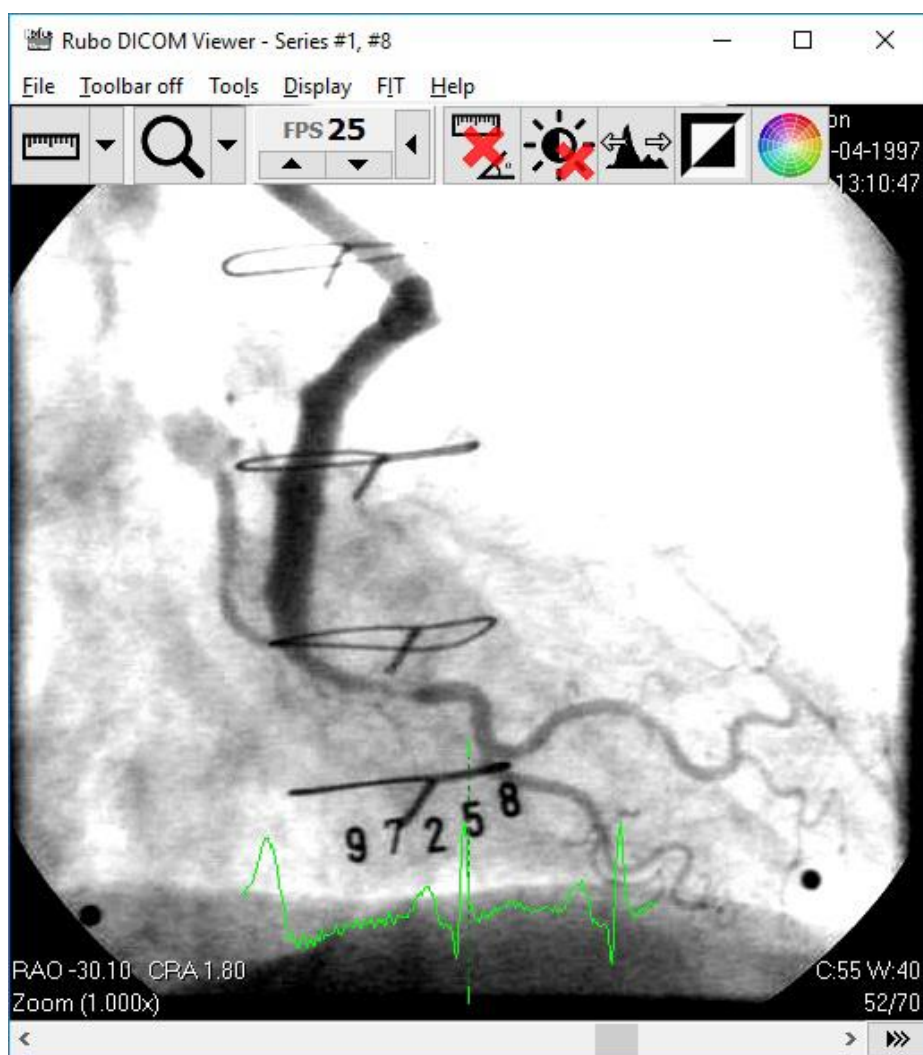


Рисунок 2.4 – Приклад ангиограми в Rubo DICOM Viewer

2.2 Sante DICOM Viewer Lite

Sante DICOM Viewer Lite - це програма перегляду DICOM, призначена для пацієнтів та інших користувачів, які не мають досвіду в медичній візуалізації (рисунок 2.5). Програма повністю безкоштовна, сумісна з усіма видами файлів DICOM.

Програма дозволяє виконувати поворот зображення, зберігати зображення в різних форматах.

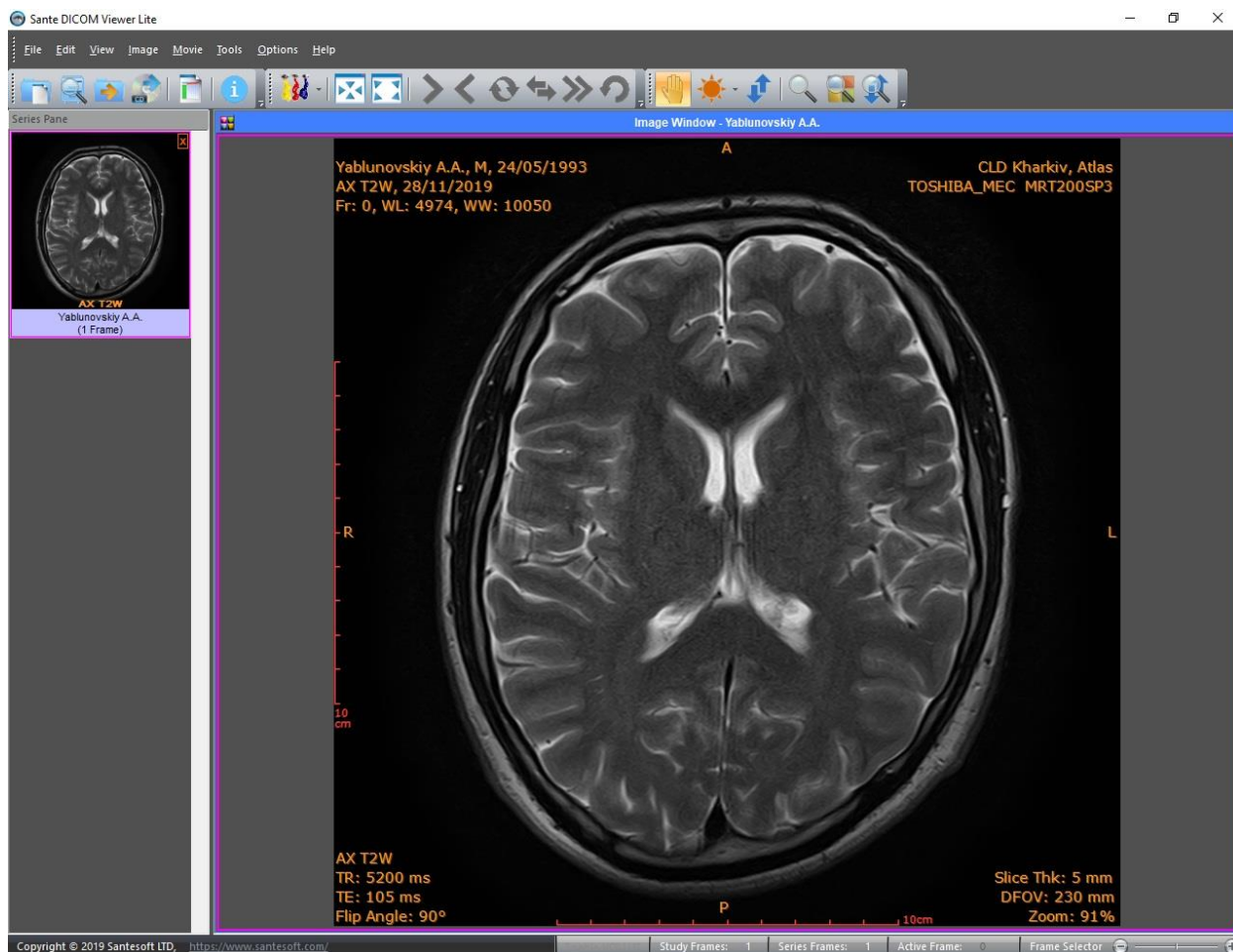


Рисунок 2.5 – Просмотр зображення МРТ програмою Sante DICOM Viewer Lite

2.3 Radiant DICOM VIEWER

RadiAnt DICOM Viewer надає наступні основні інструменти для маніпуляцій з зображеннями та їх вимірювання (рисунок 2.6): плавне масштабування і панорамування, коригування яскравості і контрастності, режим «негатив».

Ця програма здатна відкривати і відображати результати досліджень, які отримані з різних джерел медичних зображень:

- Цифрова рентгенографія (КР, ЦР);
- Мамографія (МГ);
- Комп'ютерна томографія (КТ);

- Магнітно-резонансна томографія (МРТ);
- Позитронно-емісійна томографія (ПЕТ-КТ);
- Ультразвукова діагностика (УЗД);
- Цифрова субтракційна ангіографія (ЦСА);
- Гамма-камера;
- Вторинні зображення і скановані зображення;
- Програма підтримує роботу з сенсорними екранами.



Рисунок 2.6 – Приклад перегляду зображення за допомогою RadiAnt DICOM Viewer

2.4 SonicDICOM Media Viewer

Багатофункціональна програма для роботи з файлами DICOM.

Крім перегляду зображення (рисунок 2.7) дозволяє виконувати над ним деякі операції: масштабування, поворот, інверсію зображення, додавання тексту і стрілок на зображенні, формування фальшивих кольорів, збереження зображення в різних форматах.

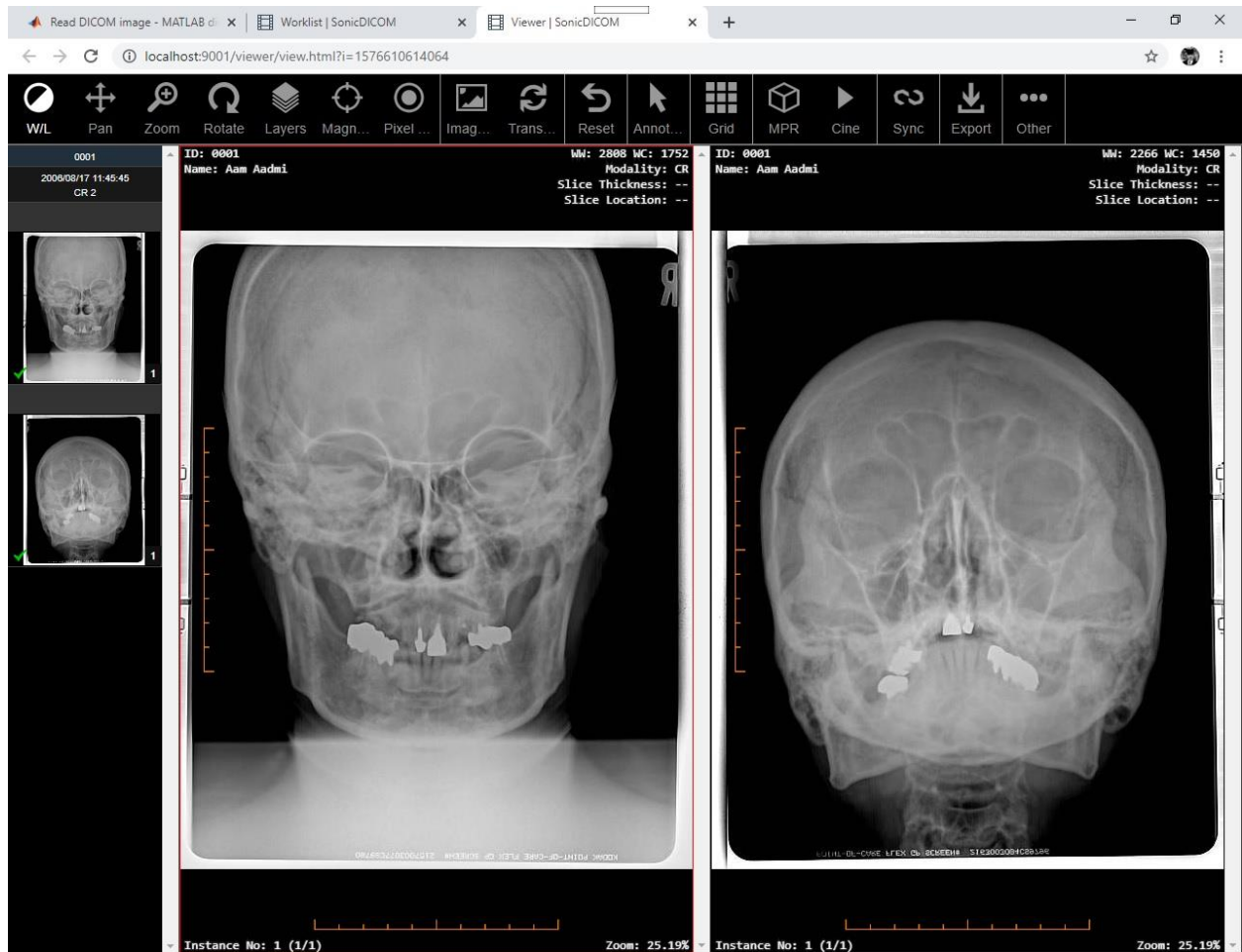


Рисунок 2.7 – Перегляд зображення із за допомогою SonicDICOM Media Viewer

2.5 Adobe Photoshop

Переглядати зображення формату DICOM можна і в Adobe Photoshop (рисунок 2.8). Можна відкрити як окремий кадр, так і кілька кадрів одночасно (рисунок 2.9).

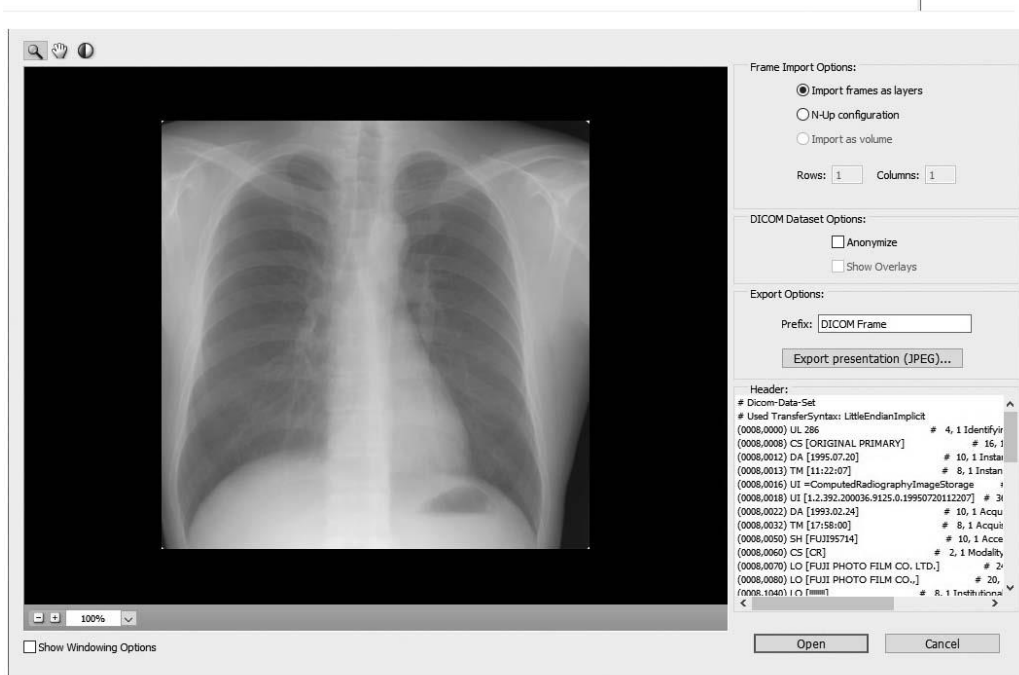


Рисунок 2.8 – Приклад відкриття файлу DICOM в Adobe Photoshop



Рисунок 2.9 – Приклад відкриття кадру з файлу DICOM

8-бітові файли DICOM можна зберігати в файл з будь-яким форматом, який підтримується Adobe Photoshop (16-бітові файли повинні зберігатися в файлах формату DICOM, Large Document Format, Photoshop PDF, Photoshop Raw, PNG або TIFF).

Кадри DICOM розміщуються на шарах. При виборі конфігурації N варіантів в сітці відображається кілька кадрів (для завдання висоти і ширини сітки вводяться значення в поля «Рядки» і «Стовпці»).

2.6 MATLAB

Файли DICOM зручно обробляти в програмному середовищі MATLAB. Додаток Image Processing Toolbox включає підтримку роботи з даними

зображень наступних медичних форматів файлів:

- DICOM-формат (Digital Imaging and Communications in Medicine);
- Формат Mayo Clinic Analyze 7.5;
- Міжфайловий формат.

2.6.1 Зчитування метаданих з DICOM-файла.

DICOM-файли містять метадані, в яких міститься інформація про такі параметри зображень, як розмір, розмірність, глибина кольору і т.п. В описі DICOM чітко позначено розміщення полів метаданих, додаткових полів і так званих персональних полів.

Для зчитування метаданих з DICOM-файла використовується функція `dicominfo`. Ця функція повертає інформацію у вигляді MATLAB-структури, поля якої містять опис частин метаданих. Далі структуру метаданих, яка отримана за допомогою функції `dicominfo`, можна використовувати при зчитуванні даних з використанням функції `dicomread`.

Розглянемо приклад зчитування метаданих (Приклад 2.1) з DICOM-файла (рисунок 2.10).

```
>> info = dicominfo ('brain.dcm')
```

```
info =
```

```
info = struct;
```

```
Filename = 'C:\Users\Александр\Documents\MATLAB\brain.DCM';
```

```
FileModDate = '19-январь-2019 14:30:56';
```

```
FileSize = 526336;
```

```
Format = 'DICOM';
```

```
FormatVersion = 3;
```

```
Width = uint16(512);
```

```
Height = uint16(512);
```

```
BitDepth = uint16(12);
```

```
ColorType = 'grayscale';
```

```
FileMetaInformationGroupLength = uint32(166);
```

```
FileMetaInformationVersion = uint8( ... [0; 1]);
```

```
MediaStorageSOPClassUID = '1.2.840.10008.5.1.4.1.1.4';
```

```
MediaStorageSOPInstanceUID = ['1.3.46.670589.11.0.4.19960823073800'06'];
```

```
TransferSyntaxUID = '1.2.840.10008.1.2.1';
```

```
ImplementationClassUID = '1.3.46.670589.17';
```

```
ImplementationVersionName = 'ARC_CONVERT';
```

```
SpecificCharacterSet = 'ISO_IR 100';
```

```
ImageType = 'ORIGINAL\PRIMARY\OTHER\R\IR';
```

```
InstanceCreationDate = '19960823';
```

```
InstanceCreationTime = '093801';
```

```
InstanceCreatorUID = '1.3.46.670589.11.0.5';
```

```
SOPClassUID = '1.2.840.10008.5.1.4.1.1.4';
```

```
SOPInstanceUID = '1.3.46.670589.11.0.4.1996082307380006';
```

```
StudyDate = '19950330';
```


SeriesDate = '19950330';
StudyTime = '150829.0000';
SeriesTime = '155614.6300';
AccessionNumber = "";
Modality = 'MR';
Manufacturer = 'Philips';
InstitutionName = 'OTM4 R4.5';
ReferringPhysicianName = struct;
ReferringPhysicianName.FamilyName = "";
ReferringPhysicianName.GivenName = "";
ReferringPhysicianName.MiddleName = "";
ReferringPhysicianName.NamePrefix = "";
ReferringPhysicianName.NameSuffix = "";
StudyDescription = 'BRAIN';
SeriesDescription = ['Brain STIR GR 512 TRA GraSE 4415 ' 64'];
ManufacturerModelName = 'T5';
PatientName = struct;
PatientName.FamilyName = 'MR/BRAIN/GRASE/1024';
PatientID = '7';
PatientBirthDate = '19930111';
PatientSex = 'M';
PatientWeight = 90;
ScanningSequence = 'IR';
SequenceVariant = 'OTHER';
ScanOptions = "";
MRAcquisitionType = '2D';
SliceThickness = 6;
RepetitionTime = 4414.826172;
EchoTime = 64;
InversionTime = 140;
NumberOfAverages = 2;
ImagingFrequency = 42.593601;
ImagedNucleus = '1H';
EchoNumber = 1;
MagneticFieldStrength = 1;
SpacingBetweenSlices = 7.5;
NumberOfPhaseEncodingSteps = 384;
EchoTrainLength = 0;
PercentSampling = 75;
PercentPhaseFieldOfView = 75;
DeviceSerialNumber = '00000';
SoftwareVersion = 'RTNC1';
.ProtocolName = 'STIR GR 512';
LowRRValue = 0;
HighRRValue = 0;
IntervalsAcquired = 102;
IntervalsRejected = 0;

```
HeartRate = 60;
ReceiveCoilName = 'H';
TransmitCoilName = 'B';
InPlanePhaseEncodingDirection = 'ROW';
FlipAngle = 90;
PatientPosition = 'HFS';
StudyInstanceUID = '1.3.46.670589.11.0.1.1996082307380006';
SeriesInstanceUID = '1.3.46.670589.11.0.2.1996082307380006';
StudyID = ' 191';
SeriesNumber = 1;
AcquisitionNumber = 13;
InstanceNumber = 1;
ImagePositionPatient = [-125.000992; -122.842384; 32.496708];
ImageOrientationPatient = [1; -5E-6; -0; 5E-6; 0.959915; -0.280292;];
FrameOfReferenceUID = '1.3.46.670589.11.0.3.1996082307380006';
PositionReferenceIndicator = "";
ImageComments = "";
SamplesPerPixel = uint16(1);
PhotometricInterpretation = 'MONOCHROME2';
Rows = uint16(512);
Columns = uint16(512);
PixelSpacing = [0.488281; 0.488281];
BitsAllocated = uint16(16);
BitsStored = uint16(12);
HighBit = uint16(11);
PixelRepresentation = uint16(0);
WindowCenter = 1951;
WindowWidth = 237;
```

Приклад 2.1 – Метадані з DICOM-файлу

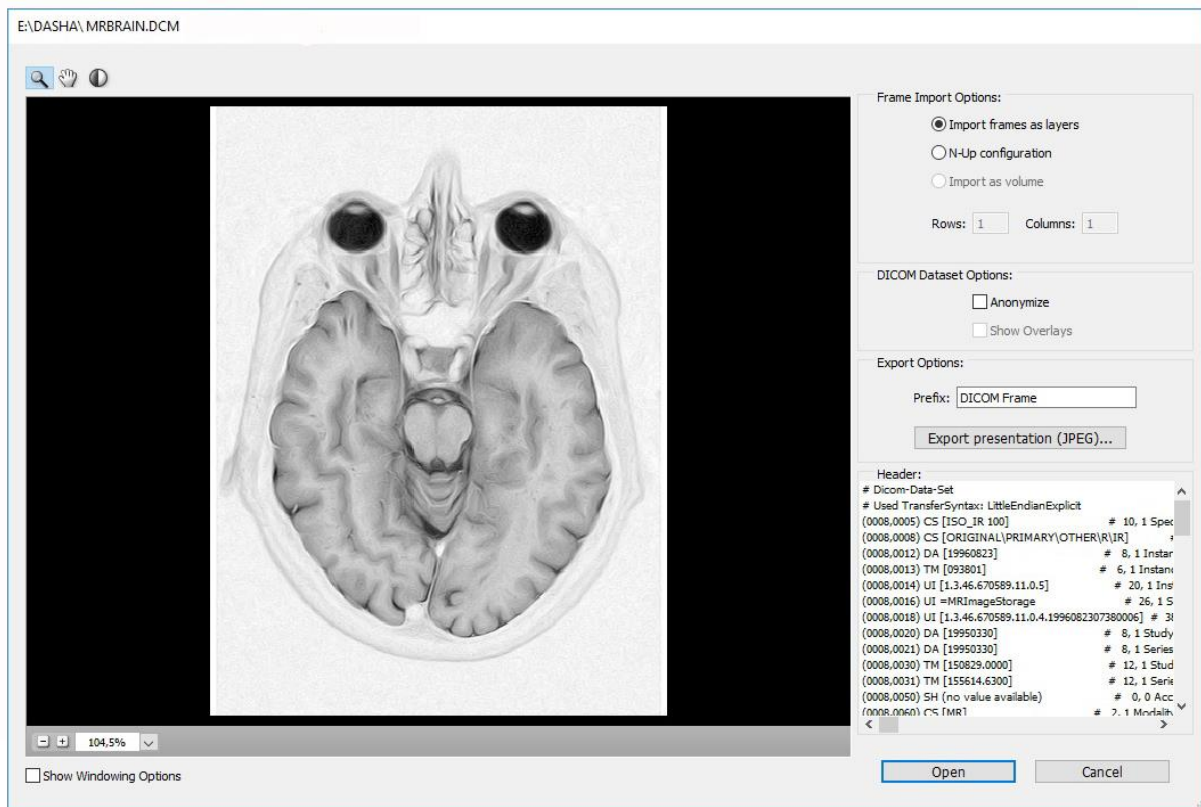


Рисунок 2.10 – DICOM-файл

2.6.2 Зчитування даних зображення з DICOM-файла.

Для зчитування даних зображення з DICOM-файла використовується функція `dicomread`. При використанні функції `dicomread` необхідно вказати назву файлу в якості аргументу:

```
I = dicomread ( 'bmode.dcm').
```

Для перегляду даних зображення, які отримані з DICOM-файла, використовуються відомі функції програми для візуалізації зображень `imshow` або `imshow`. Незважаючи на те, що дані зображення в DICOM-файлі представлені у вигляді 16-бітних даних без знака, існує необхідність проведення автомасштабування для поліпшення візуального сприйняття зображення:

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

Для запису даних зображення або метаданих в файл у форматі DICOM, використовується функція `dicomwrite`:

```
dicomwrite (I, 'h: \ MATLAB \ work \ brain.dcm ').
```

При запису метаданих та даних зображення в DICOM-файл функція `dicomwrite` автоматично включає мінімальні установки полів метаданих з

інформацією щодо створеного об'єкта інформації.

Функція `dicomwrite` підтримує такі типи зображень:

- вторинного захоплення (за замовчуванням);
- Магнітний резонанс;
- Комп'ютерна томографія.

Функція `dicomwrite` може також записувати і інші типи DICOM-даних (наприклад, рентгенівські, радіотерапії, ядерної медицини) в файл.

Витяг конфіденційної інформації з DICOM-файла

При використанні DICOM-файла в процесі тренінгу або презентації існує можливість видалення конфіденційної інформації про пацієнта, обробляючи так званий анонімний файл. Для цього використовують функцію `dicomanon`.

Функція `dicomanon` створює новий набір значень метаданих і записує їх у файл.

При записи модифікованого зображення в DICOM-файл існує можливість модифікувати зображення і запустити нову серію. У 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 у полі метаданих `SeriesInstanceUID`.

Видалення всієї текстової інформації з зображення.

Для цього необхідно знайти мінімальні і максимальні значення пікселів на зображенні. Пікселі, за допомогою яких представлена текстова інформація, приймають максимальне значення інтенсивності.

```
max (I (:))  
ans = 4080
```

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

```
ans = 32
```

Таким чином, для видалення текстової інформації, всім пікселям з максимальним значенням, необхідно призначити мінімальне значення:

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

Візуалізуємо оброблене зображення:

```
imshow (Imodified)
```

Крім цього, пакет `Image Processing Toolbox` дозволяє виконувати багато інших операцій по обробці зображень з DICOM-файлів.

ЛАБОРАТОРНА РОБОТА №3 Візуалізація медичних досліджень

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 представлене 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.

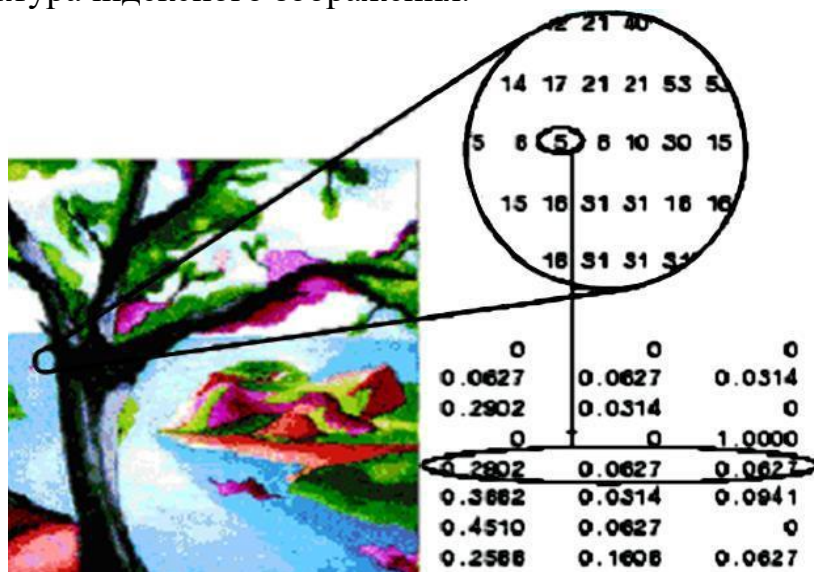


Рисунок 3.1 – Матриця пікселів індексного зображення та співвідношення значень її пікселів з палітрою

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**.

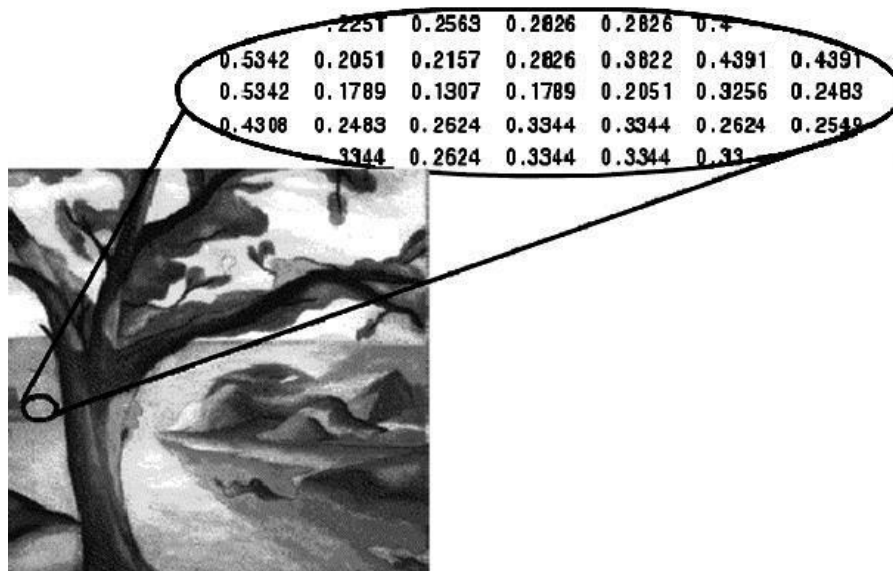


Рисунок 3.2 – Фрагмент матриці пік селів на півтонового зображення

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 .

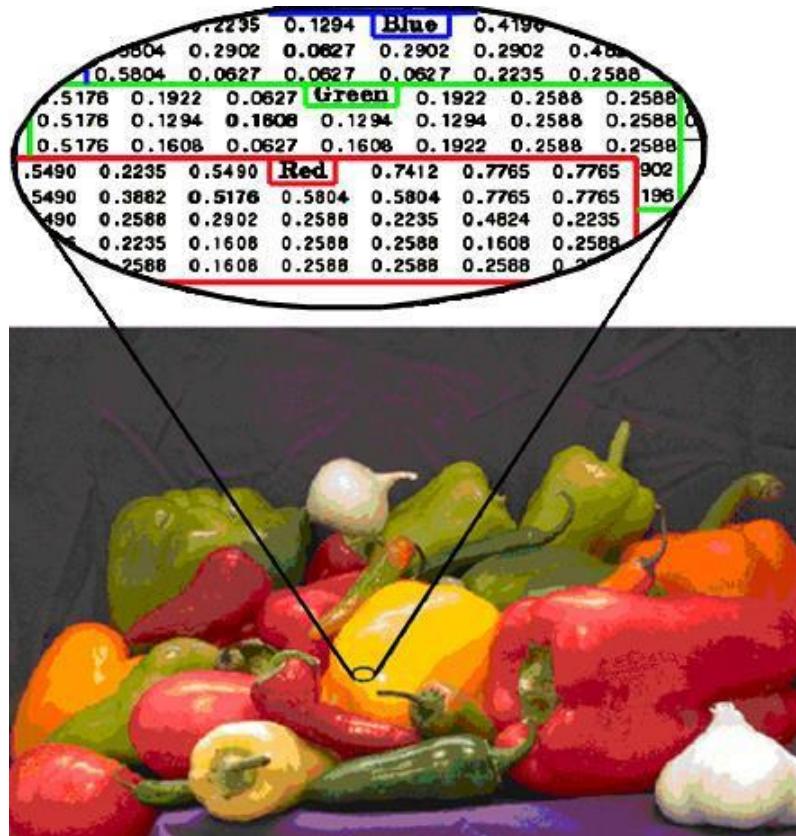


Рисунок 3.3 – Колірні складові повноколірного зображення

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 |
| grayscale | 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 **subplot**, 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=grayscale(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=grayscale(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 палітрового 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 сприймалися 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 файли |

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.

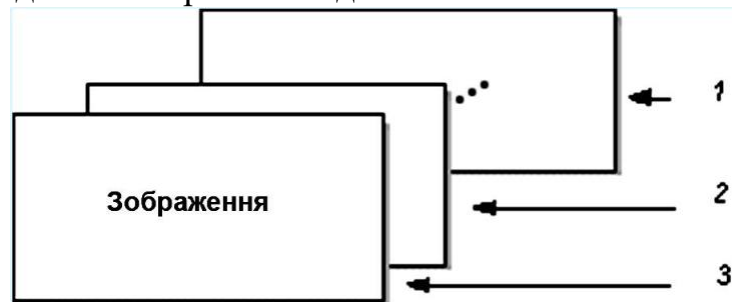


Рисунок 3.4 – Багатовимірний масив послідовності зображень

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 і y (на відміну від значень r та c , як в піксельній системі координат). Змінна y збільшується донизу (рис.3.5).

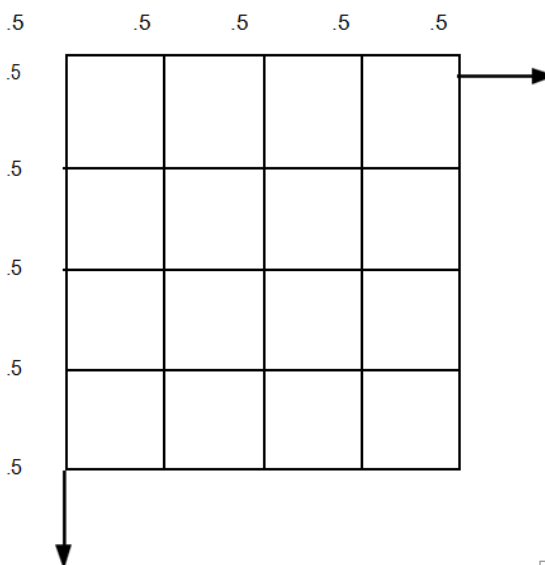


Рисунок 3.5 - Просторова система координат

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 відрізнятись 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 **ftm** 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.

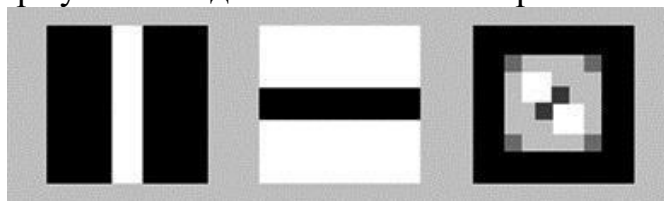


Рисунок 3.6 – Найпростіші тестові зображення

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

```

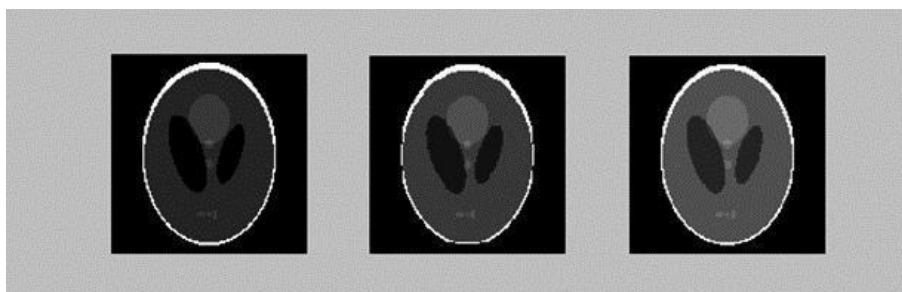


Рис.3.7 – Результати формування фантомів зрізу голови людини

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

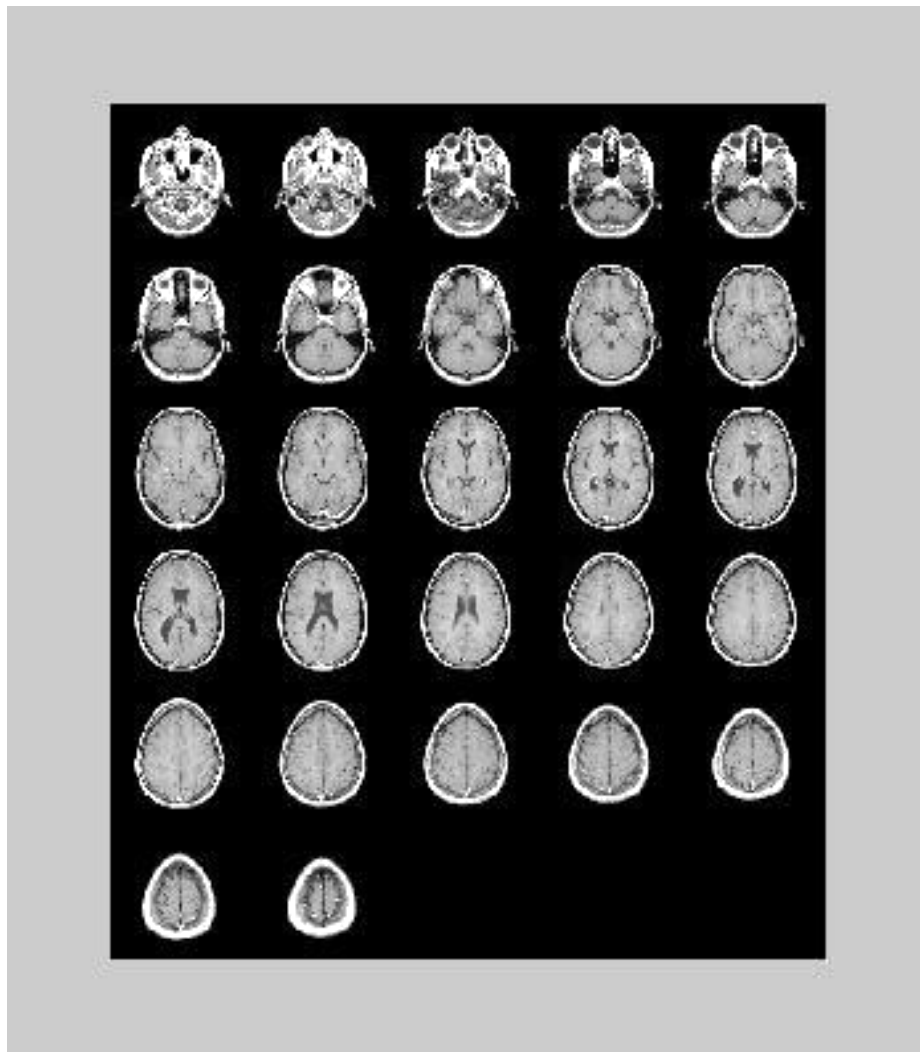


Рисунок 3.8 – Вивід 27 кадрів магнітно-резонансного зображення у вигляді послідовності зрізів

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; circles.tif; board.tif; cameraman.tif; canoe.tif forest.tif; kids.tif; lily.tif; logo.tif; m83.tif; cell.tif; moon.tif; circbw.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

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 радіоізоотопів 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 клавiші 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 клавiші 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 **impixel**. 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)
```

```
impixel % Визначення значення яскравості пікселя
```

```
Функції P=impixel(I)      P=impixel(X,map)      P=impixel(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

$$K = \sum_i \sum_j ((I_{ij} - I_{min}) / (I_{max} - I_{min}))^a$$

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.

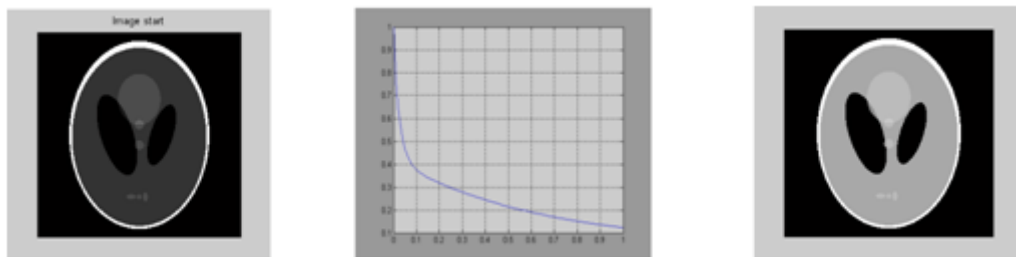


Рисунок 4.2 – Результати перетворень тестового зображення та графік $K = f(a)$

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.

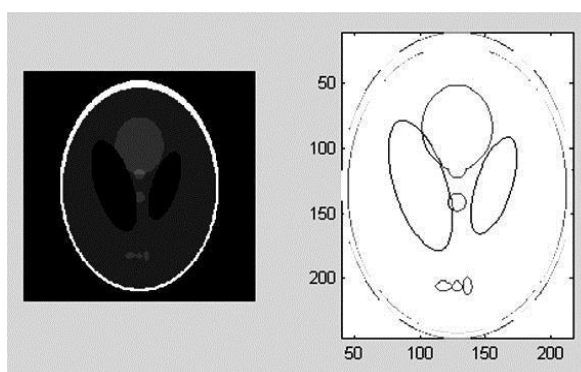


Рисунок 4.3 – Формування контурного зображення фантома голови

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(I,M,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);

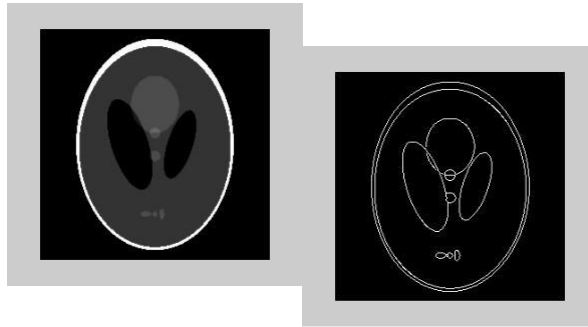


Рисунок 4.4 – Виділення меж структур на зображенні фантома голови людини функцією **edge**

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 квадро-деревах. 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 квадро-дерева 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)
```

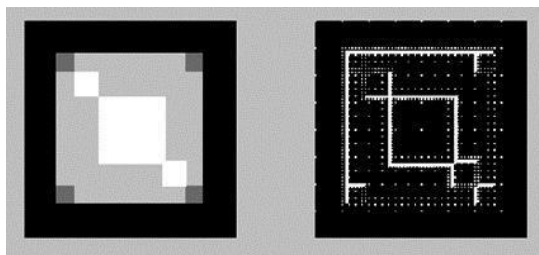


Рисунок 4.5 – Результат сегментації тестового зображення

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)

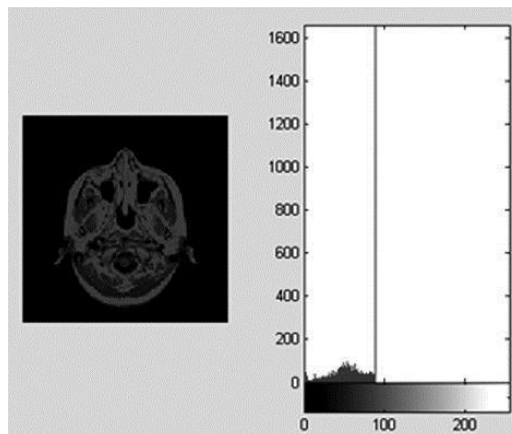


Рисунок 4.6 – Гістограма магніторезонансного зображення зрізу голови людини

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 званій 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');
```

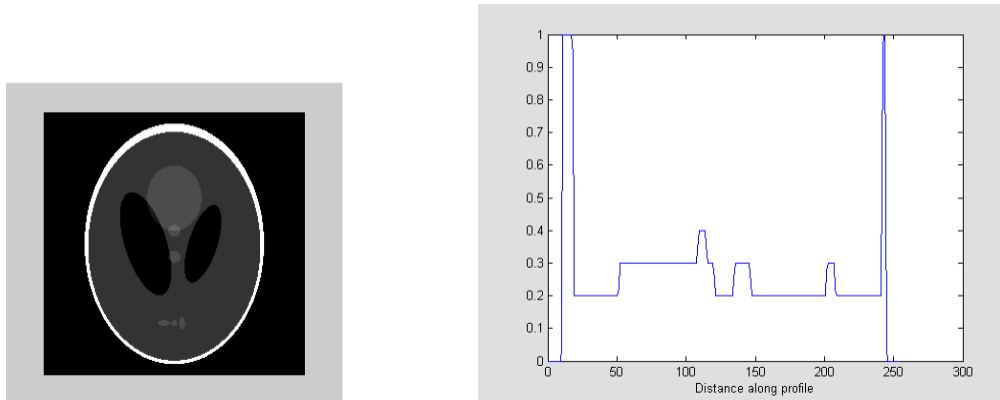


Рисунок 4.8 – Побудова яскравісного профілю перетину фантому голови

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?

This project has been funded with support from the European Commission. This publication / communication reflects the views only of the author, and the Commission cannot be held responsibility for any use which may be made of the information contained therein.