

Міністерство освіти і науки України
Чернівецький національний університет
імені Юрія Федьковича

Навчально-науковий інститут фізико-технічних та комп'ютерних наук
(повна назва інституту/факультету)

Кафедра інформаційних технологій та комп'ютерної фізики
(повна назва кафедри)

Обробка медичних зображень (рентгенівських та
томографічних)

Кваліфікаційна робота

Рівень вищої освіти - перший (бакалаврський)

Виконав:

студент 4 курсу, групи 417 ск
спеціальності

126 Інформаційні системи та технології
(назва спеціальності)

Григоращук Андрій Андрійович
(прізвище та ініціали)

Керівник доктор фіз.-мат. наук, ст.наук.сп.,

Борча Мар'яна Драгошівна
(науковий ступінь, вчене звання, прізвище та ініціали)

До захисту допущено:

Протокол засідання кафедри №

20 від „15” серпня 2023 р.

зав. кафедри [підпис] доц. Борча М. Д.

Чернівці – 2023

РЕЦЕНЗІЯ НА БАКАЛАВРСЬКУ РОБОТУ

Обробка медичних зображень (рентгенівських та томографічних)

Студента Григоращука Андрія Андрійовича

Інститут фізико-технічних та комп'ютерних наук

Спеціальність інформаційні системи та технології

Чернівецького національного університету імені Юрія Федьковича

Актуальність теми. Діагностика змін, що відбуваються у біологічних шарах та тканинах за допомогою медичних X-променевих зображень, потребує коректної інтерпретації цих зображень та забезпечення високої точності визначення параметрів цих змін. Саме такі завдання вирішувались у дипломній роботі, що свідчить про її актуальність.

Практичне значення. Точна і своєчасна діагностика захворювання має вирішальне значення для правильного і успішного лікування хвороб. Коректна і точна інтерпретація зображень X-променевої комп'ютерної томографії необхідна при визначенні безлічі захворювань, зокрема: опорно-рухового апарату; м'яких тканин та тонких шарів - легень, трахей, бронхів; та ін.

Структура та зміст роботи. Робота складається з 3х розділів. У першому розділі описано особливості формату медичних X-променевих зображень, такі як протоколи для обміну зображеннями, стиснення зображень, 3-D візуалізація та презентація зображень, звітування про результати діагностики. У другому розділі описано методи фільтрації зображень та здійснено їх порівняльний аналіз. Для обробки медичних X-променевих зображень обрано метод білатеральної фільтрації, в якому параметри ядра фільтра обчислюються автоматично на основі рівня шуму зображення. У третьому розділі запропоновано алгоритм комплексної обробки X-променевих медичних зображень і його програмна реалізація, яка полягає у такій послідовності дій: білатеральна фільтрація → підвищення локального контрасту → Гамма-корекція та інвертування зображення → візуалізація у псевдокольорі..

Зауваження. До недоліків роботи слід віднести те, вибір параметрів для білатеральної фільтрації і гамма-корекції здійснюється емпірично

Оцінка. В результаті виконання бакалаврської роботи студент Григоращук А.А. освоїв та реалізував комплекс методів обробки медичних X-променевих зображень. Вважаю, що завдання бакалаврської роботи виконано повністю, а Григоращук А.А. заслуговує оцінки „добре” та присвоєння кваліфікації бакалавра зі спеціальності 126 – інформаційні системи та технології.


Рецензент

Завідувач кафедри комп'ютерних наук

Чернівецького національного університету

доктор фізико-математичних наук, професор

Ушенко Юрій Олександрович


“ 15 “ 06 2023 р.



Ім'я користувача:
Кафедра інформаційних технологій та комп фізики

ID перевірки:
1015667315

Дата перевірки:
21.06.2023 13:54:52 EEST

Тип перевірки:
Doc vs Internet + Library + DB

Дата звіту:
21.06.2023 13:55:36 EEST

ID користувача:
36471

Назва документа: Григоращук-1

Кількість сторінок: 7 Кількість слів: 3312 Кількість символів: 27526 Розмір файлу: 44.33 KB ID файлу: 101531263

5.62% Схожість

Найбільша схожість: 5.62% з джерелом з Бібліотеки (ID файлу: EF-100000099198)

Не знайдено джерел з Інтернету

0% Джерела з Бібліотеки 1

Сторінка 9

0% Цитат

Не знайдено жодних цитат

Вилучення списку бібліографічних посилань вимкнене

101263

2.17% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 12 слів та 2%)

0.39% Вилучення з Інтернету 3

Сторінка 10

2.17% Вилученого тексту з Бібліотеки 19

Сторінка 10

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 53

101263

Міністерство освіти і науки України
Чернівецький національний університет
імені Юрія Федьковича

Навчально-науковий інститут фізико-технічних та комп'ютерних наук

(повна назва інституту/факультету)

Кафедра інформаційних технологій та комп'ютерної фізики

(повна назва кафедри)

Обробка медичних зображень (рентгенівських та
томографічних)

Кваліфікаційна робота

Рівень вищої освіти - перший (бакалаврський)

Виконав:

студент 4 курсу, групи 417 ск

спеціальності

126 Інформаційні системи та технології

(назва спеціальності)

Григоращук Андрій Андрійович

(прізвище та ініціали)

Керівник доктор фіз.-мат. наук, ст.наук.сп.,

Борча Мар'яна Драгошівна .

(науковий ступінь, вчене звання, прізвище та ініціали)

До захисту допущено:

Протокол засідання кафедри №

_____ від „_____” _____ 2023 р.

зав. кафедри _____ доц. Борча М. Д.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА

Навчально-науковий інститут фізико-технічних та комп'ютерних наук
Кафедра інформаційних технологій та комп'ютерної фізики

ЗАТВЕРДЖУЮ

Завідувач кафедри
док. фіз.-мат.наук,доц.

_____ М. Д. Борча
” ____ ” ____ 2023 р.

Обробка медичних зображень (рентгенівських та
томографічних)

ЛИСТ ЗАТВЕРДЖЕННЯ

УЗГОДЖЕНО

Керівник роботи
док. фіз.-мат. наук,
доц.

_____ М. Д. Борча
” ____ ” ____ 2023 р.

Виконавець
студент 4-го курсу

_____ А.А. Григоращук
“ ____ ” _____ 2023 р.

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Григоращуку Андрію Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Обробка медичних зображень (рентгенівських та томографічних)

керівник роботи Борча Мар'яна Драгошівна док фіз-мат наук, доцент,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від “__”_____202_ року №

2. Строк подання студентом проекту (роботи) _____ 2023 р.

3. Вихідні дані до проекту (роботи). Мета роботи – створити програмне забезпечення для обробки медичних зображень (рентгенівських та томографічних) з використанням інформаційних ехнологій. Обробити медичні зображення за допомогою білатеральної фільтрації та підвищення локального контрасту. Початкові зображення зчитувати з графічних файлів. Дослідити ефективність розробленої інформаційної системи. Програмне забезпечення системи розробити на мові Matlab.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1) описати принципи запису та обробки формату рентгенівських зображень у медицині DICOM

2) описати технології і методи фільтрації зображень

3) розробити алгоритм комплексної обробки X-променевих медичних зображень

4) створити програмне забезпечення для обробки медичних зображень

5) дослідити ефективність розробленого програмного забезпечення

5. Перелік графічного матеріалу

1) схема алгоритму обробки медичних зображень

2) функціонал програмного продукту

Студент _____
(підпис)

Григоращук А.А.
(прізвище та ініціали)

Керівник проекту (роботи) _____ Борча М.Д.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота виконана студентом групи 417ск Григоращуком Андрієм Андрійовичем. Тема “ Обробка медичних зображень (рентгенівських та томографічних)”. Робота направлена на здобуття ступеня бакалавр за спеціальністю 126 “Інформаційні системи та технології”.

Метою кваліфікаційної роботи є обробка медичних зображень за допомогою інформаційних технологій та створення відповідного програмного забезпечення. Об’єктом дослідження є методи білатеральної фільтрації та підвищення локального контрасту. В результаті виконання кваліфікаційної роботи було оброблено медичні зображення методом білатеральної фільтрації та підвищенням локального контрасту.

Бакалаврська робота містить: кількість сторінок – 66, таблиць – 3, рисунків – 16, додатків – 1, використаних джерел – 12.

Ключові слова: білатеральна фільтрація, обробка медичних зображень, підвищення локального контрасту.

Робота містить результати власних досліджень. Використання чужих ідей, результатів і текстів мають посилання на відповідне джерело.

ABSTRACT

Qualification work was performed by a student of the 417 group Hryhorashchuk Andrii Andrievich. Topic “ Processing of medical images (x-ray and tomographic) ”. The work is aimed at obtaining a bachelor’s degree in speciality 126 “Information Systems and Technologies”.

The purpose of the qualification work is the processing of medical images by information technologies and development corresponding software. The object of research is the methods of bilateral filtration and increasing local contrast. As a result of qualification work, medical images were processed by bilateral filtration and local contrast enhancement.

The bachelor's thesis contains: number of pages – 66, tables – 3, figures – 16, appendices – 1, used sources – 12.

Keywords: bilateral filtration, processing of medical images, increasing of local contrast.

The work contains the results of their own research. The use of other people's, ideas, results and texts have links to the appropriatt source.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	8
1.1 Системний аналіз об'єкту дослідження та предметної області.....	8
1.1.1 Аналіз мети функціонування системи.....	8
1.1.2 Забезпечення якості.....	8
1.2. Очікувані ефекти від впровадження.....	9
1.3 Формат рентггенівських зображень у медицині DICOM.....	10
Висновки до розділу.....	11
РОЗДІЛ 2. ПОБУДОВА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	12
2 Білатеральна фільтрація шуму на зображеннях.....	12
2.1 Видалення шуму на зображенні методом білатеральної фільтрації.....	12
2.2 Визначення параметрів ядра білатерального фільтра	15
Висновки до розділу.....	19
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	20
3.1 Гамма-корекція зображень.....	20
3.2 Створення алгоритму обробки медичних зображень.....	21
3.3 Опис реалізації системи.....	22
Висновки до розділу.....	31
ВИСНОВКИ.....	32
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	33
ДОДАТКИ.....	34
Додаток А. Код програми.....	34

ВСТУП

Метою кваліфікаційної роботи є розробка програмного забезпечення для обробки медичних зображень (рентгенівських та томографічних).

На експериментальних цифрових X-променевих медичних зображеннях присутні значні рівні шумів, а контраст багатьох ділянок на зображеннях є недостатнім для безпомилкового виявлення деталей досліджуваних об'єктів. Вищевказані недоліки медичних зображень важко усунути при їх формуванні через особливості X-променевих сенсорів зображень та обмеження на дозу X-променевого опромінення. Тому для збільшення відношення сигнал/шум (ВСШ) та візуальної якості зображень застосовуються методи цифрової фільтрації та підвищення контрасту зображень.

Проте, існуючі методи фільтрації зображень, наприклад, з ядром фільтра Гауса та медіанної фільтрації, призводять до згладжування контурів зображень. Це впливає на зменшення чіткості зображень і втрату певної кількості деталей. Існуючі методи підвищення глобального контрасту зображень (наприклад, метод еквалізації гістограми) можуть забезпечити високу візуальну якість результату тільки в часткових випадках.

Тому у даній роботі запропоновано зменшувати рівень шумів на зображеннях за допомогою білатеральної фільтрації, яка забезпечує значне підвищення ВСШ при збереженні чіткості контурів. Контраст зображень запропоновано обчислювати за допомогою локальної обробки з використанням обвідних мінімальних та максимальних значень сигналу, що забезпечує вищу візуальну якість зображень, порівняно з глобальною обробкою. З ціллю кращої візуалізації окремих деталей зображень виконано гамма-корекцію та інвертування зображень. Оскільки медичні X-променеві зображення звичайно зберігаються у відтінках сірого (у форматі DICOM), то для підвищення їх інформативності запропоновано виконувати візуалізацію зображень з використанням псевдокольорів. У результаті цього значно підвищується точність наступних етапів обробки зображень: розпізнавання, визначення розмірів об'єктів на зображеннях.

РОЗДІЛ 1

ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Системний аналіз об'єкту дослідження та предметної області

1.1.1 Аналіз мети функціонування системи

Мета функціонування системи обробки медичних зображень, зокрема рентгенівських та томографічних, полягає в наданні точної, надійної та швидкої обробки та аналізу медичних зображень з метою діагностики, планування лікування та моніторингу пацієнтів. Основні аспекти, які слід враховувати при аналізі мети функціонування такої системи, включають:

- Точність діагностики: Система повинна забезпечувати високу точність обробки та аналізу зображень для діагностики хвороб, виявлення патологій та інших медичних відхилень.
- Швидкодія: Система повинна працювати ефективно та забезпечувати швидку обробку зображень для миттєвого отримання результатів та прийняття рішень медичним персоналом.
- Надійність: Система повинна бути стійкою та надійною, забезпечуючи безперебійну роботу та уникнення втрати даних.

Аналіз функціонування системи обробки медичних зображень важливий для визначення вимог до розробки та тестування системи, а також для забезпечення високої якості та задоволення потреб медичного співтовариства та пацієнтів.

1.1.2 Забезпечення якості

Перед кваліфікаційною роботою висуваються три основні критерії якості:

- Точність.
- Надійність.
- Розширюваність.

Забезпечення якості для обробки медичних зображень, таких як рентгенівські та томографічні зображення, має особливе значення, оскільки точність та надійність цих зображень впливають на діагностику та лікування пацієнтів. Ось деякі ключові аспекти забезпечення якості для обробки медичних зображень:

- Перевірка якості зображень: Необхідно розробити процедури та метрики для перевірки якості оброблених зображень. Це може включати оцінку роздільної здатності, контрастності, шуму та інших параметрів, які впливають на чіткість та точність зображень.
- Калібрування обладнання: Рентгенівські та томографічні пристрої мають бути правильно налаштовані та калібровані, щоб забезпечити надійні та точні результати. Регулярне калібрування та технічне обслуговування обладнання є важливими етапами забезпечення якості.
- Валідація даних: Перед використанням медичних зображень необхідно впевнитися в їх якості та правильності. Це включає перевірку маркування, інтегритету даних та відсутності артефактів, що можуть спотворювати результати обробки.

Ці аспекти забезпечення якості допомагають гарантувати, що обробка медичних зображень відбувається з високою точністю, надійністю та забезпечує потрібну якість для правильної діагностики та лікування пацієнтів.

Програмна реалізація системи обробки зображень виконана на мові Matlab з використанням стандартних бібліотек, що забезпечує надійність роботи програми.

1.2 Очікувані ефекти від впровадження

Впровадження системи обробки медичних зображень, рентгенівських та томографічних, може мати наступні очікувані ефекти: покращення точності та якості діагностики: Система обробки зображень дозволить отримувати більш точні та детальні результати, що допоможе лікарям

правильно визначити патології, виявити відхилення та поставити діагнози з вищою точністю. Збільшення ефективності та швидкості обробки: Автоматизована система обробки зображень дозволить зменшити час, необхідний для обробки та аналізу медичних зображень. Це дозволить медичному персоналу швидше отримувати результати та приймати рішення, що сприятиме зменшенню очікування пацієнтів та покращенню продуктивності медичного закладу. Зниження ризику помилок: Система обробки медичних зображень допомагає зменшити ризик помилок, що пов'язані зі штучними факторами, такими як неправильна інтерпретація або пропуск важливих деталей. Автоматичні алгоритми обробки можуть допомогти виявити аномалії та підказати лікареві наявність патологій, що сприятиме підвищенню безпеки та надійності діагностики. Підвищення задоволення пацієнтів: Швидша обробка та аналіз медичних зображень дозволить зменшити час очікування результатів та невизначеності для пацієнтів. Це може покращити їх задоволення та сприяти більш якісному та ефективному лікуванню.

1.3 Формат рентгенівських зображень у медицині DICOM

Цифрові зображення та комунікації в медицині (DICOM) - стандарт комунікації та управління медичною візуальною інформацією та суміжних даних. Частіше всього використовується для зберігання та передачі медичних зображень, що дозволяє інтегрувати медичні пристрої візуалізації, такі як сканери, сервери, робочі станції, принтери, мережеве обладнання та інші. Поширене використання у лікарнях та невеликих структурах, наприклад, стоматологічні кабінети. Такі файли можуть обмінюватися між двома сутностями, здатними приймати зображення та данні пацієнта у форматі DICOM. Різні пристрої поставляються з вимогами відповідності DICOM, які чітко визначають, які класи DICOM вони підтримують, а стандарт включає визначення формату файлів і протокол мережевого зв'язку, який використовує TCP / IP для зв'язку між системами.

Стандарт DICOM використовується у всьому світі для зберігання, обміну та передачі медичних зображень, а також займає центральне місце у розвитку сучасної рентгенологічної візуалізації: DICOM включає стандарти для способів візуалізації, такі як рентгенографія, УЗД, комп'ютерна томографія (КТ), магнітно-резонансна томографія (МРТ) та променева терапія. DICOM включає протоколи для обміну зображеннями (наприклад, за допомогою портативних носіїв, таких як DVD), стиснення зображень, 3-D візуалізації, презентації зображень і звітування про результати.

Висновки до розділу 1

Проаналізувавши потреби лікарів та медичних закладів, зроблено висновок про потребу розробки програми для обробки медичних зображень (рентгенівських та томографічних). Тому метою даної роботи є обробка медичних зображень за допомогою білатеральної фільтрації та підвищення локального контрасту.

Впровадження системи обробки медичних зображень, зокрема рентгенівських та томографічних, має багато переваг і може позитивно вплинути на якість та ефективність медичної діагностики. Деякі з основних висновків щодо системи обробки медичних зображень включають.

Прийнято рішення виконувати програмну реалізацію для обробки зображень на мові Matlab.

Застосування сучасних технологій обробки зображень може значно поліпшити процес діагностики, забезпечити більш точні результати та підвищити рівень медичного обслуговування.

РОЗДІЛ 2 ПОБУДОВА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2 Білатеральна фільтрація шуму на зображеннях

2.1 Видалення шуму на зображенні методом білатеральної фільтрації

Метод білатеральної фільтрації дозволяє зберігати контури і ефективно зменшувати шум на однорідних областях зображень, оскільки такий метод застосовує просторове зважене усереднення без згладжування країв (контурів). Це досягається за рахунок комбінування двох гаусівських фільтрів: один фільтр працює у просторовій області, а інший – в області інтенсивності (яскравості). Тому при формуванні ядра фільтра враховується не тільки просторова відстань від центру ядра фільтра, але й яскравість пікселів зображення.

Ядро білатерального фільтра $w_B = (w_B(m, n))$ описується формулою:

$$w_B(m, n) = \exp\left(\frac{-((m - m_C)^2 + (n - n_C)^2)}{2\sigma_{SB}^2}\right) \cdot \exp\left(\frac{-(f_w(m, n) - f_c)^2}{2\sigma_B^2}\right), \quad (2.1)$$

де $m = 1, \dots, M_w$; $n = 1, \dots, N_w$;

m – номер рядка, n – номер стовпця;

(M_w, N_w) – розміри ядра фільтра;

σ_{SB} – середнє квадратичне відхилення (СКВ) ядра білатерального фільтра у просторовій області;

σ_B – СКВ ядра білатерального фільтра в області яскравості;

m_c та n_c – координати центру ядра фільтра w_B за висотою і шириною відповідно.

$f_w(m, n)$ – яскравість пікселя зображення, який відповідає елементу ядра з номерами (m, n) ;

f_c – яскравість пікселя зображення, який відповідає центру ядра.

Значення ядра фільтра w_B нормуються до 1.

Згідно з формулою (2.1) ядро білатерального фільтра w_B (рис. 2.1в, рис. 2.2в) можна описати поелементним добутком двох ядер w_{SB} (рис. 2.1а,

рис. 2.2а) та w_{BB} (рис. 2.1б, рис. 2.2б), елементи яких обчислюються за формулами:

$$w_{SB}(m,n) = \exp\left(\frac{-((m-m_C)^2 + (n-n_C)^2)}{2\sigma_{SB}^2}\right), \quad (2.2)$$

$$w_{BB}(m,n) = \exp\left(\frac{-(f_w(m,n) - f_c)^2}{2\sigma_B^2}\right), \quad (2.3)$$

де $m = 1, \dots, M_w$; $n = 1, \dots, N_w$.

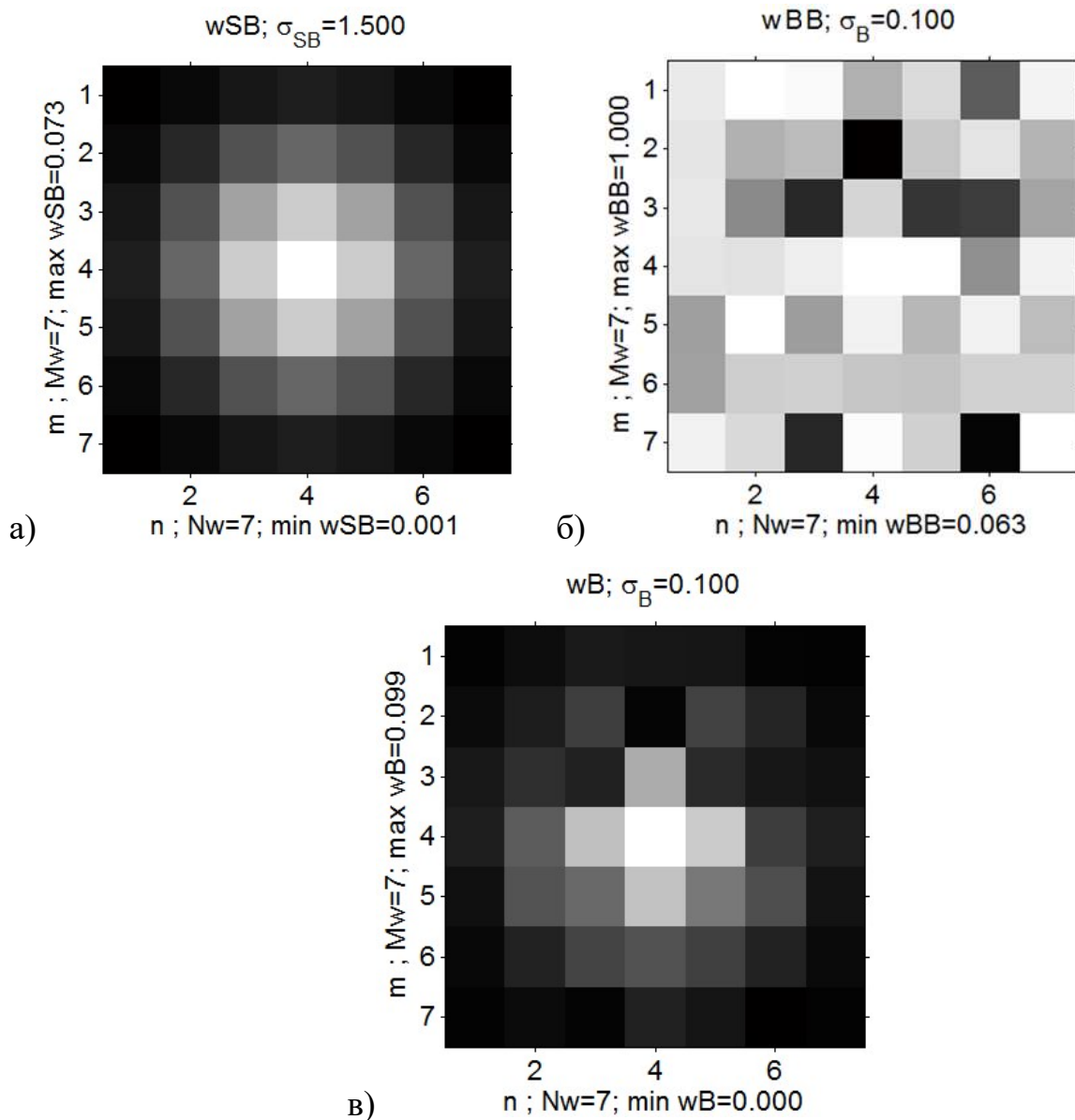


Рисунок 2.1. Ядра білатерального фільтра: а) ядро фільтрації в просторовій області w_{SB} ; б) ядро фільтрації в області яскравості w_{BB} ; в) сумарне ядро білатерального фільтра w_B

wSB=

0.0013	0.0041	0.0079	0.0099	0.0079	0.0041	0.0013
0.0041	0.0124	0.0241	0.0301	0.0241	0.0124	0.0041
0.0079	0.0241	0.0470	0.0587	0.0470	0.0241	0.0079
0.0099	0.0301	0.0587	0.0733	0.0587	0.0301	0.0099
0.0079	0.0241	0.0470	0.0587	0.0470	0.0241	0.0079
0.0041	0.0124	0.0241	0.0301	0.0241	0.0124	0.0041
0.0013	0.0041	0.0079	0.0099	0.0079	0.0041	0.0013

a)

wBB=

0.9199	0.9997	0.9796	0.7104	0.8680	0.4014	0.9567
0.9021	0.7115	0.7517	0.0635	0.7940	0.8980	0.7254
0.9149	0.5714	0.2128	0.8447	0.2627	0.2884	0.6689
0.9003	0.8880	0.9381	1.0000	0.9948	0.5917	0.9512
0.6487	1.0000	0.6450	0.9515	0.7372	0.9459	0.7561
0.6533	0.8184	0.8272	0.7894	0.7795	0.8294	0.8311
0.9456	0.8574	0.2091	0.9890	0.8248	0.0837	0.9979

б)

wB=

0.0017	0.0055	0.0105	0.0095	0.0093	0.0022	0.0017
0.0050	0.0119	0.0244	0.0026	0.0258	0.0150	0.0040
0.0098	0.0186	0.0135	0.0667	0.0166	0.0094	0.0072
0.0120	0.0360	0.0741	0.0986	0.0786	0.0240	0.0127
0.0069	0.0325	0.0408	0.0752	0.0466	0.0307	0.0081
0.0036	0.0136	0.0269	0.0320	0.0253	0.0138	0.0046
0.0017	0.0047	0.0022	0.0132	0.0088	0.0005	0.0018

в)

Рисунок 2.2 Ядра білатерального фільтра (рис. 2.1) у табличній формі: а) ядро фільтрації в просторовій області wSB; б) ядро фільтрації в області яскравості wBB; в) сумарне ядро білатерального фільтра wB; $\sigma_{SB} = 1.5$, $\sigma_B = 0.1$

З врахуванням правила 3σ для двовимірного нормального розподілу (розподілу Гауса) отримаємо такі розміри фільтра wB:

$$M_w = [6 \cdot \sigma_{SB}], N_w = [6 \cdot \sigma_B]. \quad (2.4)$$

З метою усунення крайових ефектів на межах зображення білатеральна фільтрація виконується над розширеним зображенням f_e .

2.2 Визначення параметрів ядра білатерального фільтра

Ядро w_{SB} білатерального фільтра описується розподілом Гауса з СКВ σ_{SB} . При цьому оптимальне СКВ σ_{wT} ядра фільтра Гауса залежить як від рівня шуму, так й від середнього просторового періоду зображення [10-11]. Таку залежність можна зобразити у вигляді поверхні (рис. 2.5), у вигляді графіків (рис. 2.6) або як таблицю (табл. 2.1).

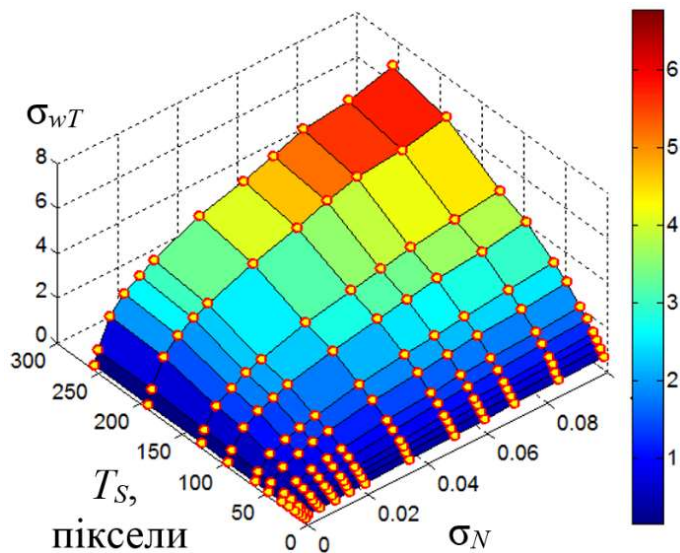


Рисунок 2.3 Тривимірна візуалізація залежності σ_{wT} (T_S , σ_N) для зображень, яскравість яких описано двома взаємно перпендикулярними синусоїдами з просторовим періодом T_S .

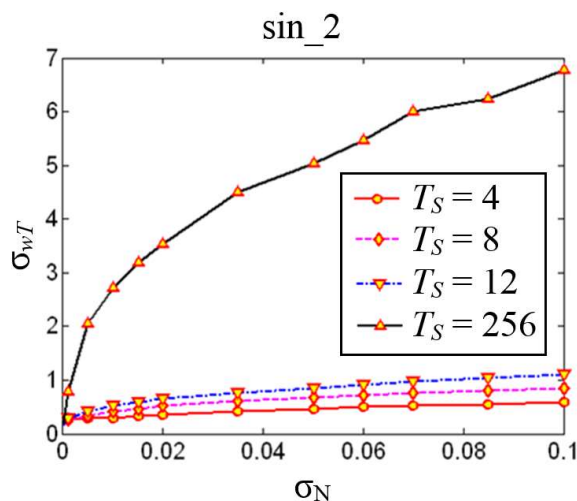


Рис. 2.4. Залежності σ_{wT} (T_S , σ_N) (рис. 2.5) для зображень, яскравість яких описано двома взаємно перпендикулярними синусоїдами з просторовим періодом T_S .

Таблиця 2.1

Залежність σ_{wT} (TS, σ_N) (рис. 2.5) для зображень, яскравість яких описано двома синусоїдами [10, 11]

σ_N T_S	0	0.001	0.005	0.01	0.015	0.02
4	0.16931	0.26428	0.28085	0.29189	0.32543	0.36094
8	0.16931	0.27533	0.34318	0.41421	0.47131	0.51572
12	0.16931	0.28637	0.42056	0.52931	0.59051	0.64637
16	0.16931	0.30294	0.50303	0.61771	0.69738	0.75672
24	0.17446	0.39054	0.63179	0.78015	0.88281	0.97488
32	0.17446	0.43959	0.72653	0.90792	1.06280	1.15316
48	0.17446	0.52932	0.93303	1.20124	1.39015	1.51752

Аналіз серії тестових зображень з шумом показав, що значення їх просторового середнього періоду TS приблизно дорівнює 16 пікселів. У такому випадку (при TS = 16) залежність σ_{wT} (σ_N) (рис. 2.5, табл. 2.2) наближено апроксимується формулою (за методом найменших квадратів):

$$\sigma_{wG} = k_1 + k_2 \sqrt{\sigma_N}, \quad (2.1)$$

де $k_1 = 0.16$; $k_2 = 3.9$.

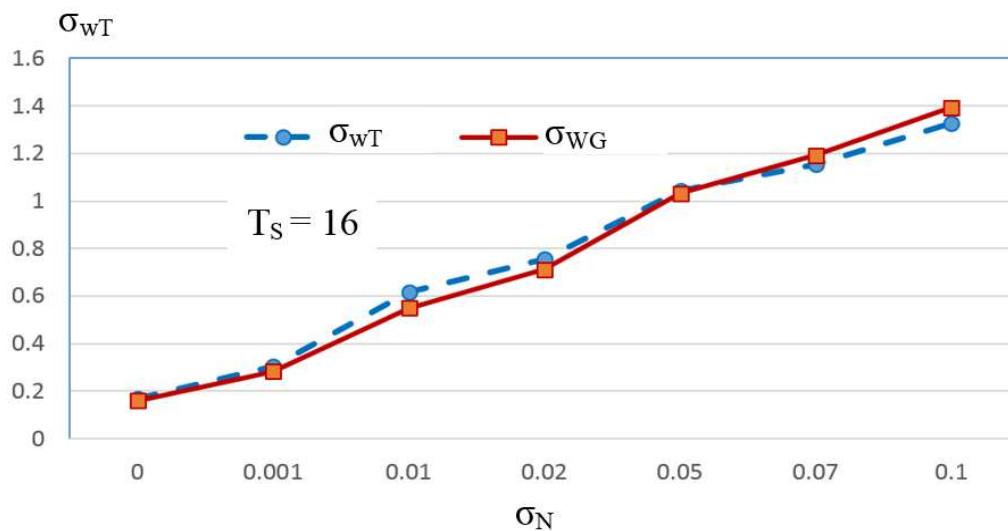


Рисунок 2.5 Залежність σ_{wT} (σ_N) для $TS=16$ пікселів; σ_{wG} – апроксимовані значення за формулою (2.1)

Таблиця 2.2

Залежність σ_{wT} (σ_N) для $TS=16$ пікселів (рис. 2.7)

σ_N	σ_{wT}	σ_{wG}
0	0.16931	0.16
0.001	0.30294	0.283329
0.01	0.617706	0.55
0.02	0.756722	0.711543
0.05	1.044858	1.032067
0.07	1.153163	1.191843
0.1	1.328314	1.393288

Отримана формула (2.1) справедлива для фільтра Гауса, проте в білатеральному фільтрі значення його елементів будуть обмежуватися з урахуванням яскравості зображення, що приводить до меншого згладжування зображення. Тому для визначення СКВ ядра білатерального фільтра використано

залежність $\sigma_{wT}(\sigma_N)$ для періоду яскравості зображення $TS = 32$ пікселів (рис. 2.8, табл.2.3), яка наближено апроксимується формулою:

$$\sigma_{SB} = c_1 + c_2\sqrt{\sigma_N}, \quad (2.2)$$

де $c_1 = 0.16$; $c_2 = 6.5$.

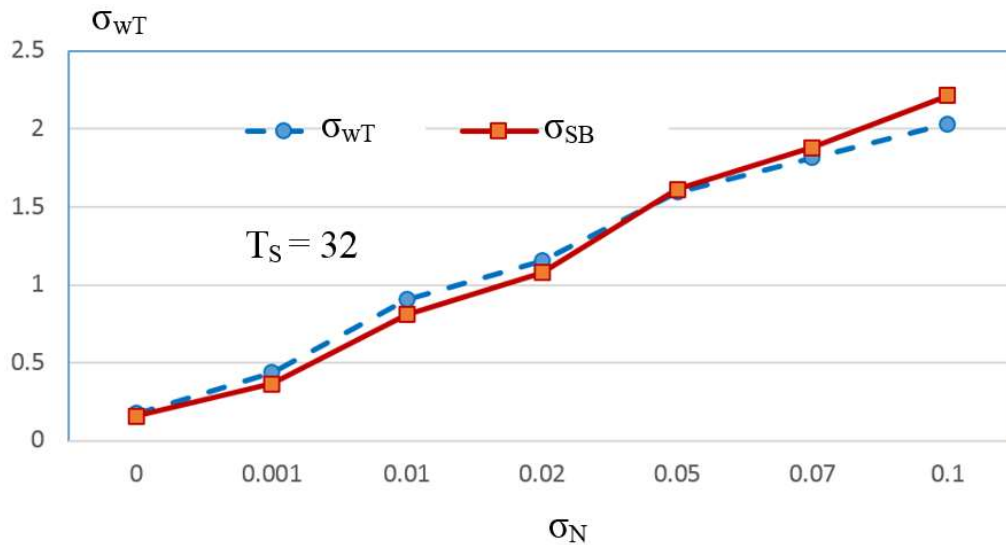


Рисунок. 2.6. Залежність $\sigma_{wT}(\sigma_N)$ для $TS=32$ пікселів; σ_{SB} – апроксимовані значення за формулою (2.2)

Таблиця 2.3

Залежність $\sigma_{wT}(\sigma_N)$ для $TS=32$ пікселів (рис. 2.8)

σ_N	σ_{wT}	σ_{SB}
0	0.174463	0.16
0.001	0.439589	0.365548
0.01	0.90792	0.81
0.02	1.153163	1.079239
0.05	1.594834	1.613444
0.07	1.816589	1.879738
0.1	2.031433	2.21548

Таким чином, за допомогою формул (2.1) та (2.2) можна обчислити СКВ ядра фільтра Гауса та білатерального фільтра відповідно.

Висновки до розділу 2

Метод білатеральної фільтрації дозволяє зберігати контури і зменшувати шум на зображеннях, що досягається за рахунок комбінування двох гаусівських фільтрів: один фільтр працює у просторовій області, а інший – в області яскравості.

Отримано емпіричні формули (2.1) та (2.2), які дозволяють обчислити СКВ ядра фільтра Гауса та білатерального фільтра через значення рівня шуму на зображенні.

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Гамма-корекція зображень

Функція контрастування зображення з гамма-корекцією $Imadjust ID = imadjust (IS,[low,high],[bottom top],gamma)$

Функція $ID = imajust(IS,[low,high],[bottom top],gamma)$ створює напівтонове зображення ID шляхом контрастування вихідного напівтонового зображення IS . Значення яскравості в діапазоні $[low,high]$ перетворюються на значення яскравості в діапазоні $[bottom top]$. Значення яскравості, менші low , приймають значення $bottom$, а значення яскравості, великі $high$, набувають значення top . Значення $top, bottom, low, high$ повинні лежати у діапазоні $[0,1]$. Якщо в якості другого або третього параметрів передати порожній вектор $[],$ то за умовчанням використовуватиметься вектор $[0,1]$. Параметр $gamma$ визначає форму кривої характеристики передачі рівнів яскравості.

Якщо $gamma$ менше 1, то характеристика передачі рівнів буде опуклою і результуюче зображення буде світлішим, ніж вихідне.

Якщо $gamma$ більше 1, то характеристика передачі рівнів буде увігнутою і результуюче зображення буде темнішим, ніж вихідне. За умовчанням $gamma$ дорівнює 1, що відповідає лінійній характеристиці передачі рівнів та відсутності гамма-корекції. Характеристика передачі рівнів різних значень $gamma$ наведені на рисунку 3.1

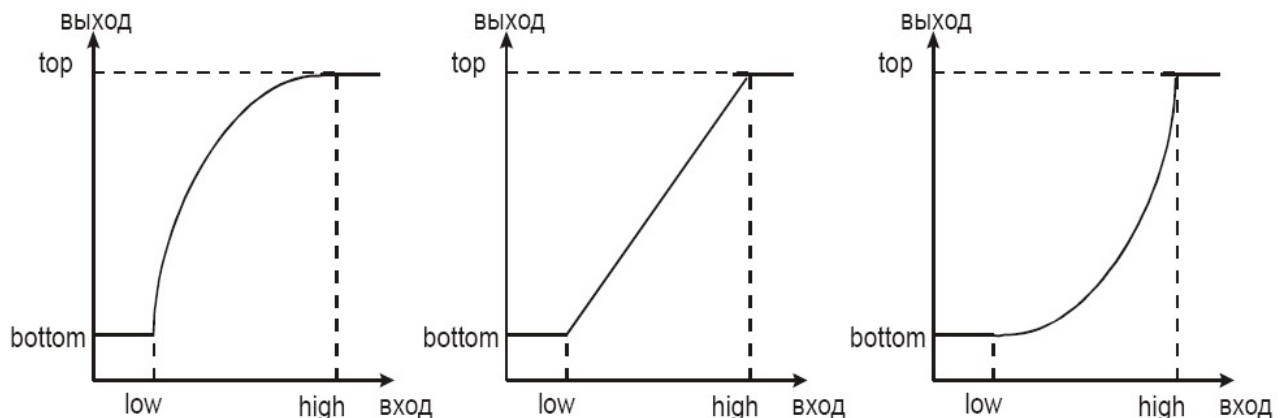


Рисунок 3.1 – Характеристика передачі рівнів

3.2 Створення алгоритму обробки медичних зображень

Обробка зображень полягає в наступному(рис.3.2).

Спочатку зчитується зображення f , наступний крок це білатеральна фільтрація, обчислення g_V ; далі йде підвищення контрасту, обчислення g_C ; наступний крок це гамма-корекція обчислення g_G ; потім інверсія зображень g_I та обчислення зображення у псевдокольорах g_P .



Рисунок 3.2 Схема алгоритму обробки медичних зображень

3.3 Опис реалізації системи

На першому етапі виконується зчитування початкового зображення f_A з файлу в форматі DICOM (Digital Imaging and Communications in Medicine). Зображення f формується як фрагмент зображення f_A , який містить найбільш інформативні ділянки. Наприклад, на рис.3.3 можна побачити посилений і деформований легеневий малюнок, на якому за рахунок фіброзних змін корені легень ущільнені. Таке зображення має відносно низький контраст і містить значний рівень шуму.

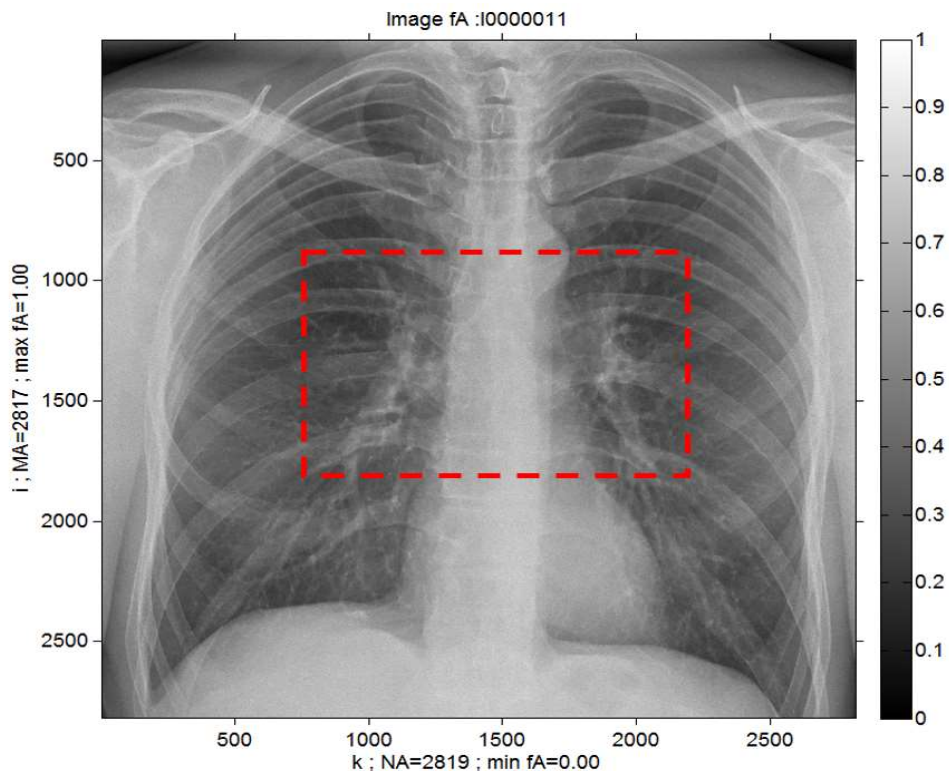


Рисунок 3.3 Початкове зображення f_A

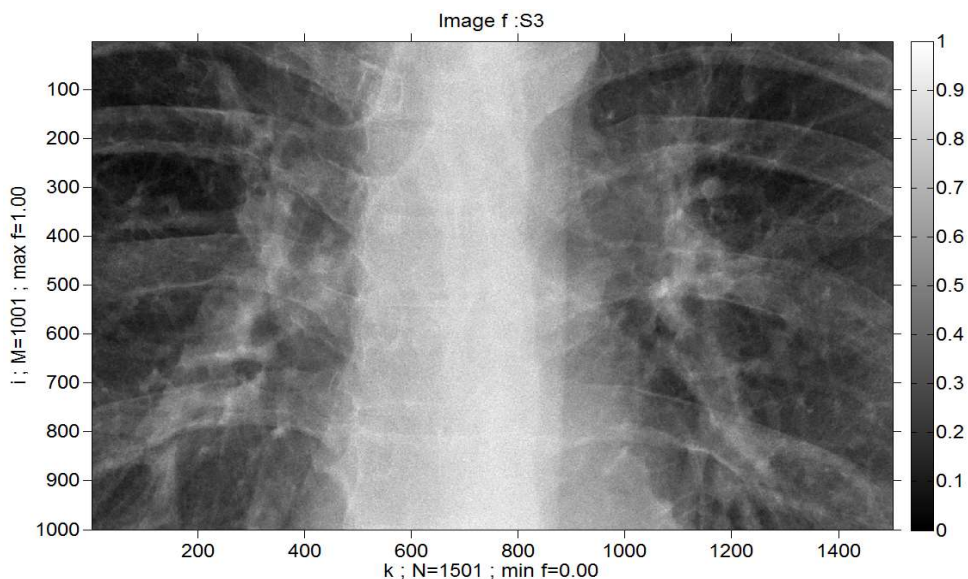


Рисунок 3.4 Фрагмент f початкового зображення f_A (рис.3.3)

Метод білатеральної фільтрації полягає у ефективній обробці зображення, дозволяючи зберігати чіткість контурів та значно зменшувати рівень шуму на однорідних ділянках зображення, оскільки такий метод застосовує просторове зважене усереднення без значного згладжування країв (тобто, контурів). Досягається такий результат за рахунок комбінування двох гаусівських фільтрів: один фільтр працює у просторовій області, а інший – в області яскравості. Саме тому при формуванні ядра фільтра необхідно враховувати не тільки просторову відстань від центру ядра фільтра, але й важливе значення яскравості пікселів зображення.

Сам по собі білатеральний фільтр - це нелінійний фільтр.

Вперше такий фільтр був представлений Томазі та Мандуччі в 1998 році.

Ядро білатерального фільтра $w_B = (w_B(m, n))$ описується формулою:

$$w_B(m, n) = \exp\left(\frac{-((m - m_C)^2 + (n - n_C)^2)}{2\sigma_{SB}^2}\right) \cdot \exp\left(\frac{-(f_w(m, n) - f_C)^2}{2\sigma_B^2}\right), \quad (1)$$

де $m = 1, \dots, M_w$; $n = 1, \dots, N_w$; m – номер рядка, n – номер стовпця;

M_w, N_w – розміри ядра фільтра;

σ_{SB} – середнє квадратичне відхилення (СКВ) ядра білатерального фільтра у просторовій області;

σ_B – СКВ ядра білатерального фільтра в області яскравості;

m_C та n_C – координати центру ядра фільтра w_B за висотою і шириною відповідно;

$f_w(m, n)$ – яскравість пікселя зображення, який відповідає елементу ядра з номерами (m, n) ;

f_C – яскравість пікселя зображення, який відповідає центру ядра.

Значення ядра фільтра w_B нормуються до 1.

З врахуванням правила 3σ для двовимірного нормального розподілу (розподілу Гауса) розміри фільтра w_B обчислюються за формулою:

$$M_w = [6 \cdot \sigma_{SB}], N_w = [6 \cdot \sigma_{SB}]. \quad (2)$$

Аналіз просторового розподілу яскравості X-променевих медичних зображень та їх Фур'є спектрів показує, що шум на таких зображеннях описується моделлю гаусового шуму з рівнем (СКВ) σ_{NE} [1, 3, 12-14]. Оптимальне СКВ σ_{SB} ядра фільтра залежить як від рівня шуму σ_{NE} , так й від середнього просторового періоду зображення. Тому що для досліджених X-променевих медичних зображень середній просторовий період є приблизно однаковим, тому σ_{SB} описується емпіричною формулою:

$$\sigma_{sb} = c_1 + c_2 \sqrt{\sigma_{NE}}$$

де $c_1 = 0.32$; $c_2 = 12$.

З врахуванням правила 3σ для нормального розподілу СКВ білатерального фільтра в області яскравості обчислюється за формулою:

$$\sigma_B = 3\sigma_{NE}. \quad (4)$$

У результаті вищеприписаної білатеральної фільтрації на основі зображення f обчислюється зображення g_B (рис. 4). При цьому рівень шуму зменшується більш ніж на порядок, що значно підвищує візуальну якість зображень і точність наступного їх аналізу. Внаслідок зменшення рівня шуму забезпечується чіткіша візуалізація дрібних елементів медичних зображень (наприклад, кальцинатів).

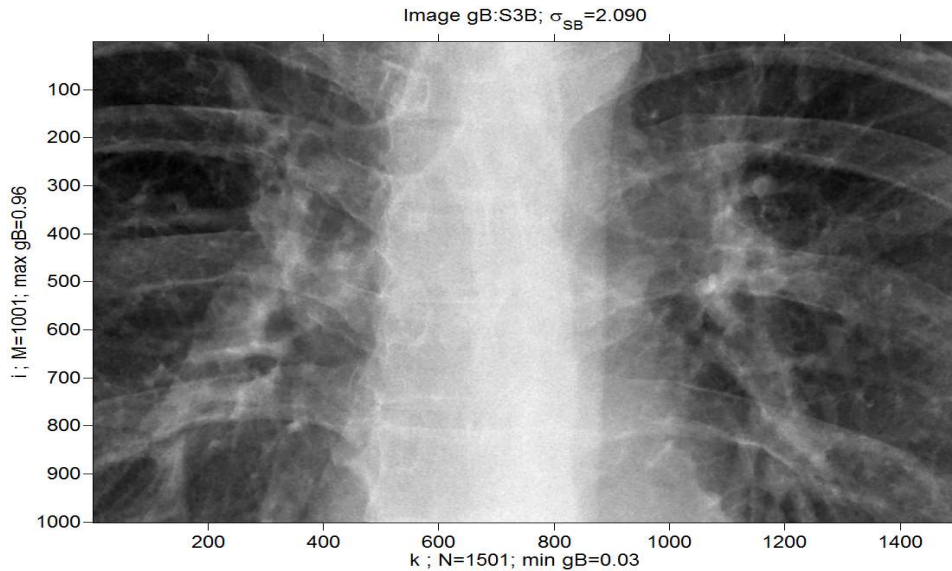


Рисунок 3.5 Зображення g_B після білатеральної фільтрації зображення f (рис. 3.4); рівень шуму до фільтрації $\sigma_{NE} = 0.0185$, після фільтрації $\sigma_{NC} = 0.0008$ (зменшено у 23 рази)

При зменшенні рівня шуму фільтром, параметри якого обчислені за формулами (1-4), можливе видалення також текстур та дрібних деталей зображення. Тому для збереження таких деталей зображення при фільтрації потрібно зменшити значення σ_{SB} (3) у кілька разів (наприклад, у 2 рази).

Зображення з підвищеним контрастом g_C обчислюється на основі обвідних (g_{Dn} і g_{Up}) та зображення g_B за формулою:

$$g_C(i, k) = (g_B(i, k) - g_{Dn}(i, k)) \cdot S_c, \quad (5)$$

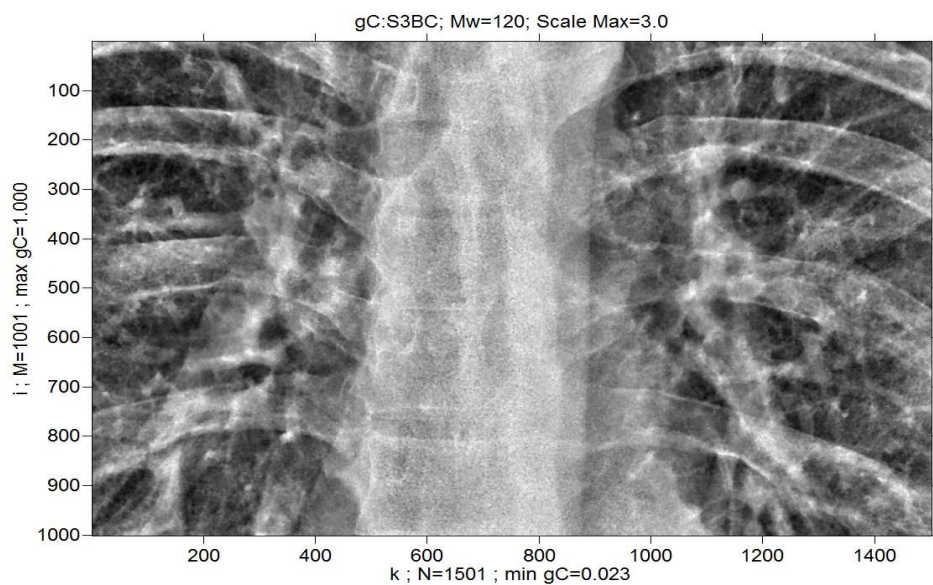
де $i = 1, \dots, M; k = 1, \dots, N;$

$S_c = 1/(g_{Up}(i, k) - g_{Dn}(i, k))$ – коефіцієнт контрасту.

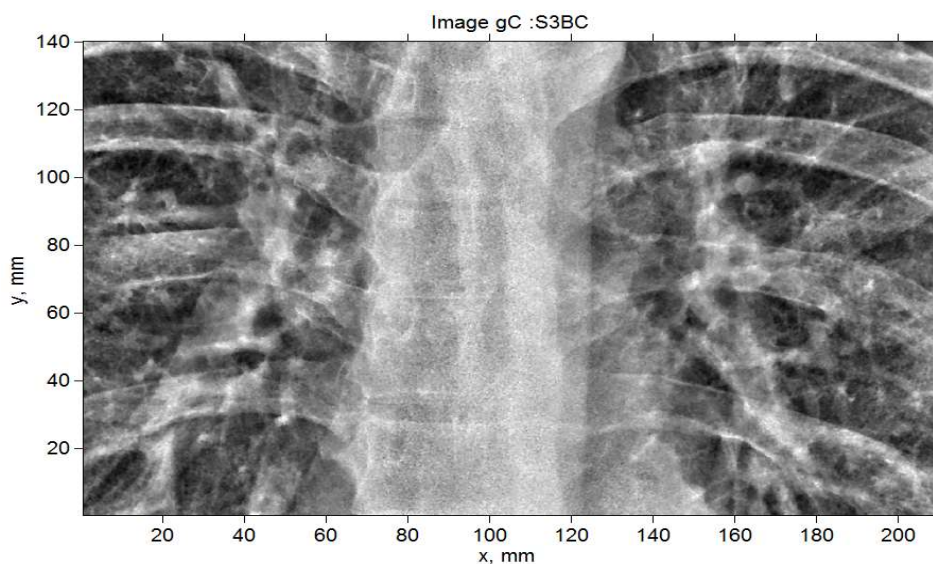
Обчислення зображення g_C за формулою (5) також дозволяє видаляти неоднорідний фон. Для запобігання появі артефактів (таких як, паразитні контури) на відновлених зображеннях g_C для найбільших значень коефіцієнту контрасту S_c встановлюється обмеження $Scale_Max$ (наприклад, $Scale_Max = 3$). Збільшення локального контрасту значно підвищує візуальну якість отриманого зображення g_C для ділянок м'яких тканин та особливо для ділянок кісток (рис. 3.4). Тим не менш, підвищення локального контрасту є корисним не для всіх задач цифрової обробки медичних зображень: за рахунок підвищення

контрасту кісткова структура стає чіткішою, але в той же час погіршується структура легеневого малюнка.

Ефективність використаних методів білатеральної фільтрації та підвищення локального контрасту сильно помітна при аналізі профілів оброблених зображень (рис. 3.6). При цьому для отримання сумарного профіля зображення з низьким рівнем шуму можливе усереднення кількох сусідніх профілів.



а)



б)

Рисунок 3.6 Зображення з підвищеним контрастом g_C , обчислене на основі зображення g_B (рис. 3.5): а) розмірності осей у пікселях; б) розмірності осей у мм

Профіль зображення g_B на рис. 3.7в демонструє, що у результаті білатеральної фільтрації значно зменшився рівень шуму (порівняно з профілем початкового зображення на рисунку 3.7б), але контури об'єктів (наприклад, кісток) при цьому не втратили своєї чіткості. Профіль зображення g_C на рис. 3.7г показує, що після підвищення локального контрасту значно зріс діапазон зміни яскравості зображення.

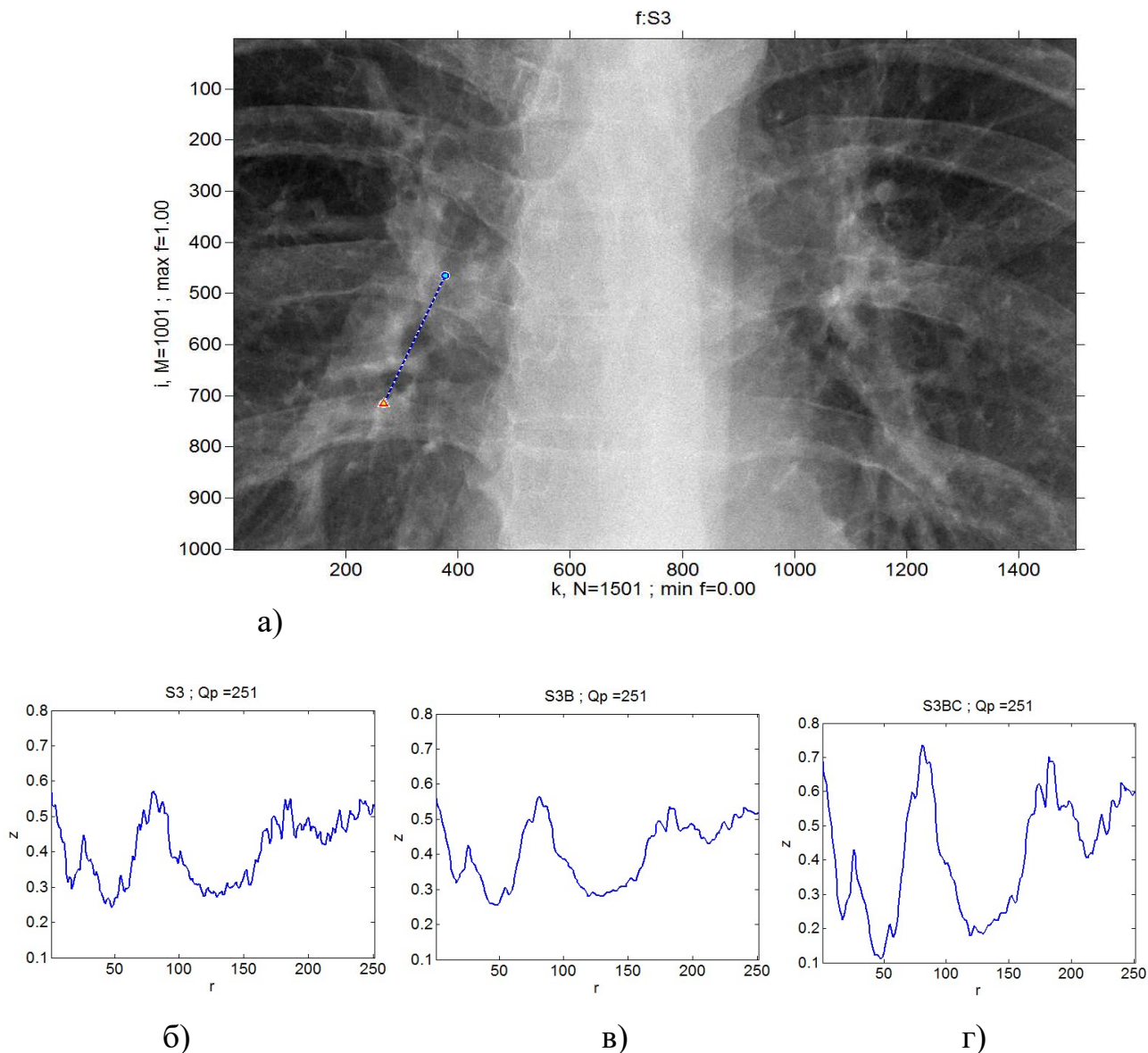


Рисунок 3.7. Профілі $z(r)$ зображень: а) початкове зображення f (рис. 3.3); б) профіль $z(r)$ для зображення f ; в) профіль зображення g_B після білатеральної фільтрації (рис. 3.5); г) профіль зображення g_C після підвищення локального контрасту (рис. 3.6);

z – яскравість зображення, r – довжина профілю (пікселі); Q_p – кількість точок профілю.

У даній роботі виконано гамма-корекцію за формулою

$$g_G = f^\gamma$$

де γ – параметр корекції.

У результаті гамма-корекції на основі фільтрованого зображення g_B обчислюються зображення g_G , на яких забезпечується краща візуалізація темних або світлих ділянок зображення, зокрема, чітко візуалізується стінка трахеї, бронхів, структура коренів легень, що добре видно на рис. 3.8.

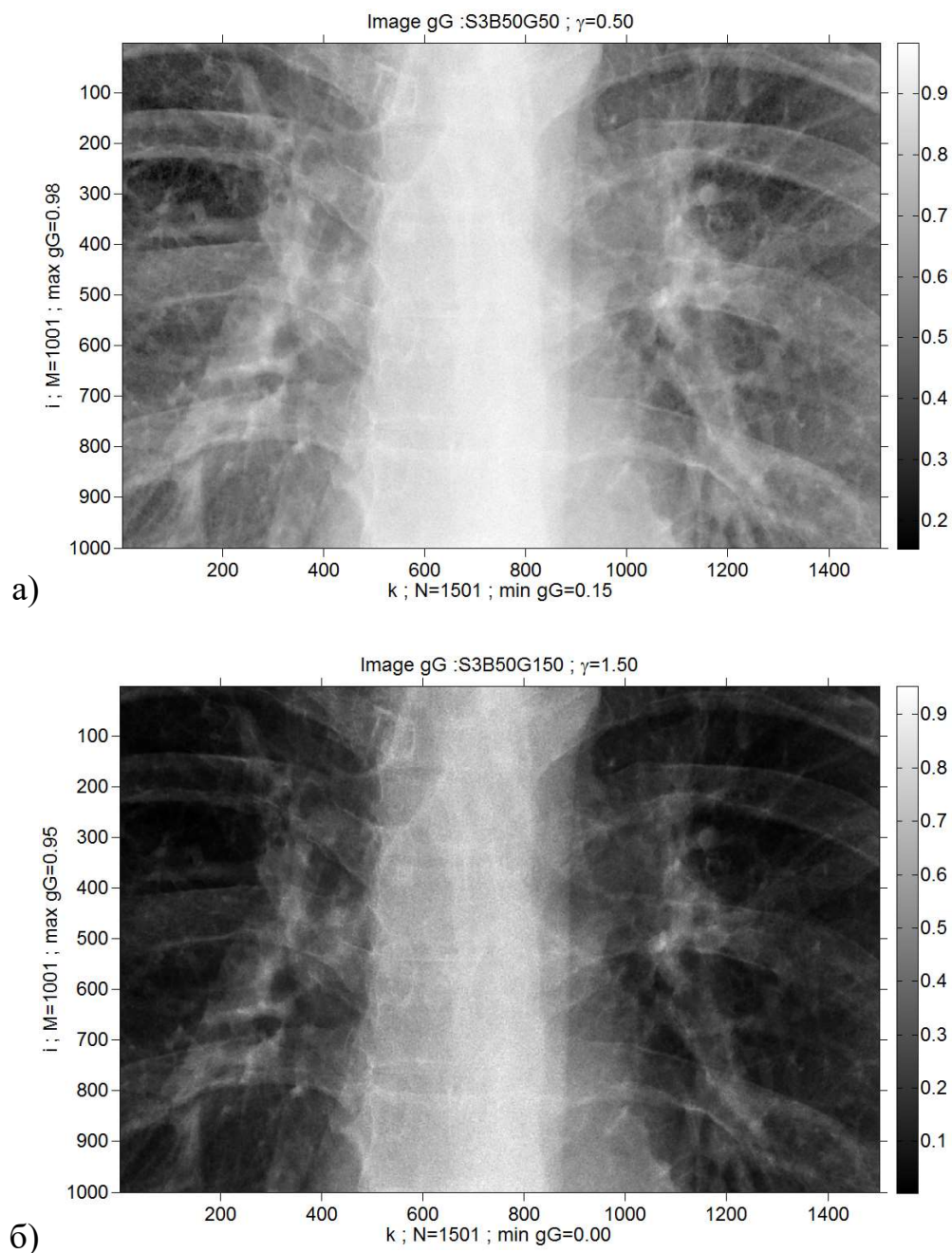


Рис. 3.8. Зображення g_G після гамма-корекції, обчислені на основі зображення g_B (рис. 3.5): а) $\gamma = 0.5$; б) $\gamma = 1.5$

Додатково для деяких зображень виконується інверсія їх яскравості, через те, що на інвертованих зображеннях g_I краще візуалізуються деталі, які знаходилися в темних ділянках (рис. 8). Особливо корисними з точки зору інформативності є інвертовані зображення, для яких попередньо виконувалась гамма-корекція (рис. 3.7б)

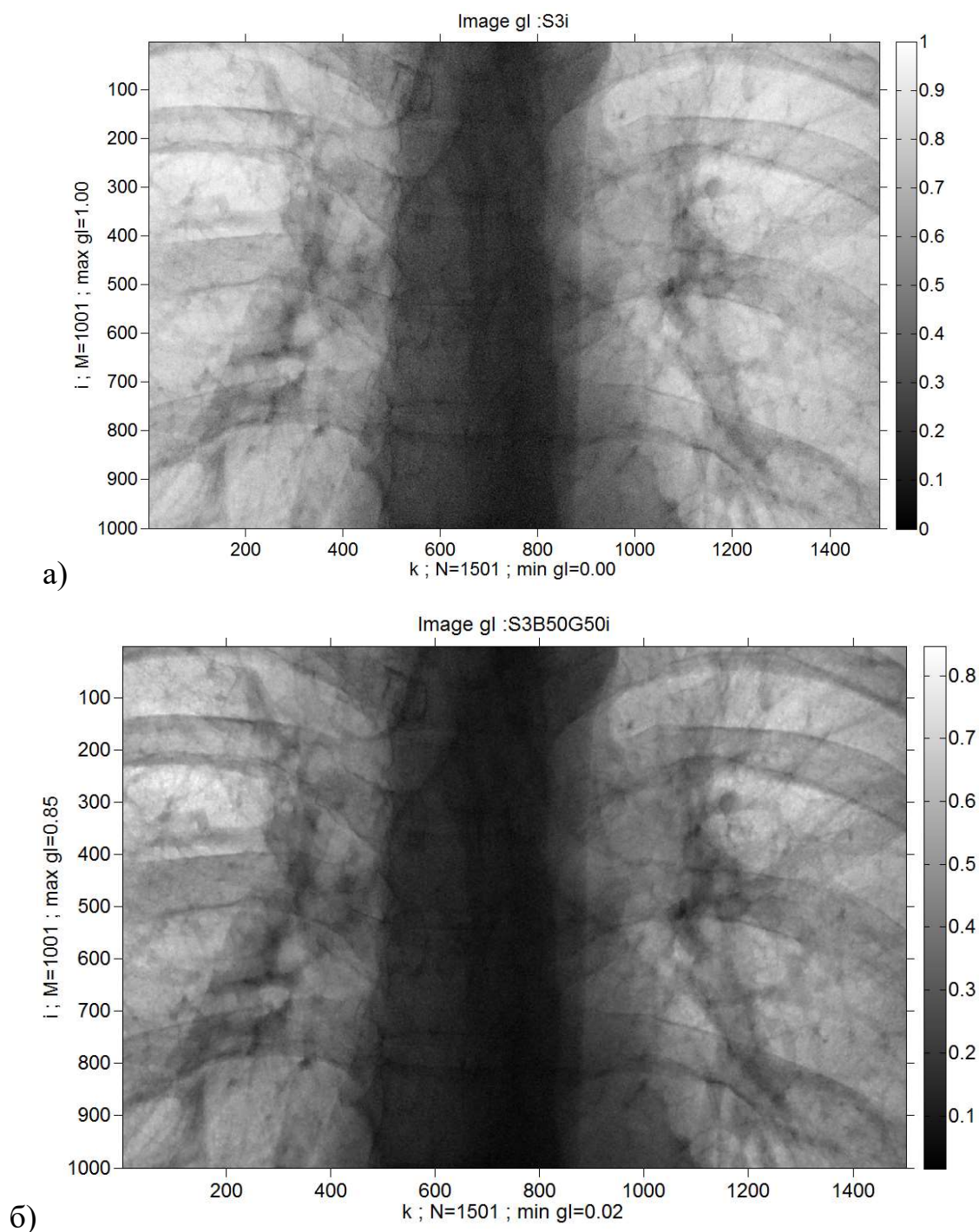


Рисунок 3.9. Інвертовані зображення g_I : а) для початкового зображення f (рис. 3.4); б) для зображення після білатеральної фільтрації та гамма-корекції g_G (рис. 3.8а)

Псевдокольори відносяться до групи методів візуалізації, які служать для відображення інформації в кольорі, которі попередньо були записані в різних частинах електромагнітного спектру. Псевдоколір дозволяє перетворити зображення у відтінках сірого, вибравши палітру кольорів, у зображення, на котрому чіткіше розрізняються деталі.

Саме тому в роботі було виконано перетворення зображення у псевдоколір, з метою кращої його візуалізації. (рис.3.10).

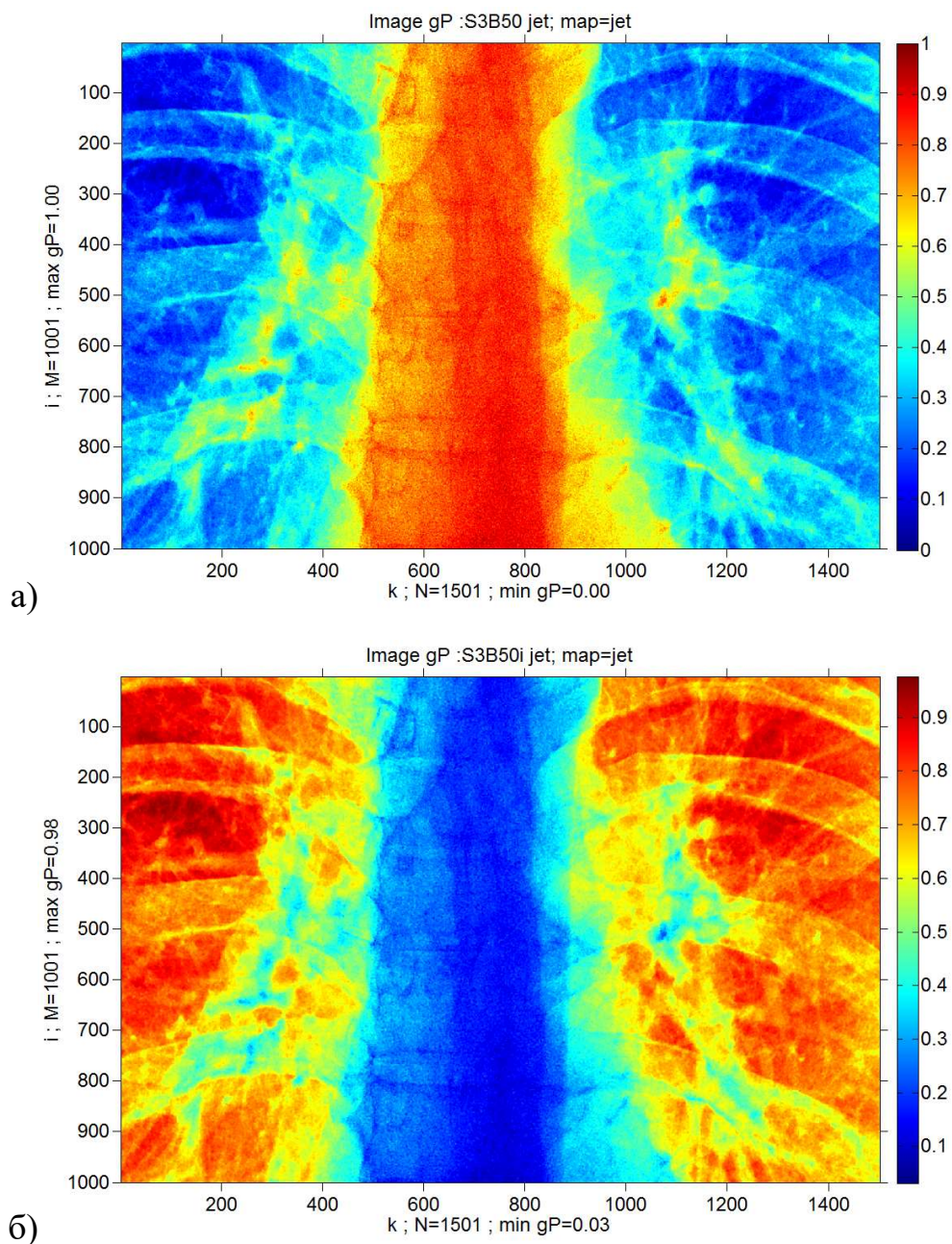


Рис. 3.10. Зображення у псевдокольорах g_P (з кольоровою палітрою «jet»), обчислені на основі: а) зображення після білатеральної фільтрації g_B (рис. 3.5); б) інвертованого зображення після білатеральної фільтрації g_B

На зображеннях у псевдокольорах щільність м'яких тканин відображається в одному кольорі, що покращує, зокрема, візуалізацію кальцинатів. Проте, ребра і хребет (які мають практично однакову густину) відображаються в різних кольорах, що пояснюється їх різною товщиною. Наприклад, товщина хребта в кілька разів перевищує товщину ребер, тому це призводить до більшого поглинання X-променів, які проходять через ділянки хребта. За рахунок вибору відповідної кольорової палітри можливо покращити візуалізацію елементів зображення, важливих для конкретної задачі.

Висновки до розділу 3

Розроблено програму для обробки медичних зображень (рентгенівських та томографічних) на мові Matlab.

Обробку медичних зображень було виконано за допомогою білатеральної фільтрації та підвищення локального контрасту. Зображення було оброблено за допомогою білатеральної фільтрації, підвищенням локального контрасту, було використано гамма-корекція, інвертування зображення та візуалізація у псевдокольорі.

ВИСНОВКИ

1. У даній кваліфікаційній роботі було розроблено інформаційну систему комплексної та поетапної обробки X-променевиx медичних зображень, яка включає білатеральну фільтрацію та підвищення локального контрасту зображень, гамма-корекцію та інверсію зображень, а також перетворення зображень з відтінків сірого у псевдокольорові.
2. Представлено метод білатеральної фільтрації, в якому параметри ядра фільтра обчислюються автоматично на основі рівня шуму зображення. Метод білатеральної фільтрації дозволяє зберігати контури і ефективно зменшувати шум на однорідних ділянках зображень. За рахунок такої фільтрації рівень шуму зменшується більш ніж на порядок, що значно підвищує візуальну якість зображень і точність наступного їх аналізу.
3. Реалізовано метод підвищення локального контрасту зображень з використанням обвідних мінімальних та максимальних значень яскравості зображення в локальних вікнах. Збільшення локального контрасту зображення дозволяє розрізнити навіть малопомітні, але важливі деталі досліджуваних об'єктів.
4. Завдяки реалізації методу гамма-корекції забезпечується краща візуалізація темних або світлих ділянок зображення. Для окремих зображень було виконано інверсію їх яскравості, тому що на інвертованих зображеннях краще видно деталі, які знаходилися до цього в темних ділянках. Показано, що найінформативнішими є інвертовані зображення, для яких попередньо виконано гамма-корекцію.
5. Для кращого візуального сприйняття зображень виконано перетворення зображень з відтінків сірого у псевдокольорові, що забезпечує краще розрізнення деталей на зображеннях за рахунок каналу кольору.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Jan J. *Medical Image Processing, Reconstruction and Restoration. Concepts and Methods* / J. Jan. – Taylor & Francis Group, 2006. – 710 p
2. C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images", *Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India*.
3. Гонсалес Р. *Цифровая обработка изображений* / Р. Гонсалес, Р. Вудс. – М.: Техносфера, 2005. – 1070 с.
4. Bankman I.H. *Handbook of Medical Image Processing and Analysis* / I.H. Bankman. – Elsevier Inc., 2009. – 984 p.
5. Gonzalez R.C. *Digital Image Processing Using MATLAB* / R. C. Gonzalez, R. E. Woods, S. L. Eddins. – Pearson Prentice-Hall, 2004. – 612 p.
6. Прэтт У. *Цифровая обработка изображений. Кн. 1* / У. Прэтт. – М.: Мир, 1982. – 312 с.
7. Порівняльний аналіз алгоритмів фільтрації медичних зображень / Бондіна Н.М., Калмичков О.С., Кривенцов В.Е. // *Вісник НТУ "ХПІ". Серія: Інформатика та моделювання*. – Харків: НТУ "ХПІ". – 2012. – No 38. – С. 14 – 25.
8. Smith, DV; Smith, S.; Бендери, Г.Н.; Картер, JR; Kim, Y.; Sawthon, MA; Leckie, RG; Weiser, JC; Romlein, J.; Goeringer, F. (1995). "Оцінка системи підтримки медичної діагностичної візуалізації на основі 2-х років клінічного досвіду". *Журнал цифрових зображень*. **8** (2): 75–87. doi : 10.1007 / BF03168130 . PMID 7612705 .
9. Баловсяк С. В. Многоуровневый метод повышения локального контраста и удаления неоднородного фона изображений / С. В. Баловсяк, И. М. Фодчук, Ю. Н. Соловей, Я. В. Луцки // *Кибернетика и вычислительная техника*. – 2015. – Вып. 182. – С. 15-26.
10. Balovsyak S.V. Hardware and Software Complex for Automatic Level Estimation and Removal of Gaussian Noise in Images / S.V. Balovsyak, Kh.S. Odaiska // *Advances in Intelligent Systems and Computing (AISC)*, Hu Z., Petoukhov S., Dychka I., He M. (Eds.). – 2019. – Vol. 754. – P.144-154. – DOI 10.1007/978-3-319-91008-6_15.
11. Balovsyak S.V. Method of calculation of averaged digital image profiles by envelopes as the conic sections / S.V. Balovsyak, O.V. Derevyanchuk, I.M. Fodchuk // *Advances in Intelligent Systems and Computing (AISC)*, Hu Z., Petoukhov S., Dychka I., He M. (Eds.). – 2019. – Vol. 754. – P. 204-212. – DOI 10.1007/978-3-319-91008-6_21.
12. Баловсяк С.В. Анализ рентгеновских муаровых изображений деформированных кристаллов по радиальным распределениям энергетического спектра Фурье / С.В. Баловсяк, С.Н. Новиков, И.М. Фодчук, И.В. Яремчук // *Металлофизика и новейшие технологии*. – 2019. – Т. 41, № 3. – С. 389-402. – DOI: doi.org/10.15407/mfint.41.03.0389.

ДОДАТКИ

Додаток А. Код програми

Білатеральна фільтрація

```
% p1_Bilat_19; % Bilateral Filter of Image
% Input Matrix (Image): f0, f
% Output (Image): g
% double - 64 bit, 15 digit after zero

clear all; close all; clc; % clc (Clear Command Window);
% This clears your workspace, closes all figures,
% and clears command window; quick way to "reset" Matlab

disp(' ----- p1_Bilat_19 ----- ');

MN_Max=10000; % Max size of image, max(M, N)

% Set Program Mode -----
Mode_Test=0; % 0/1, 0 - Original Image
% 1 - Add Noise to Image

Mode_Noise_Calc=1; % 0 - Not Noise Calculation, 1 - Noise Calculation
Mode_Filter_Calc=1; % 0 - Not Filter (Sigma_w) Calculation,
% 1 - Filter Calculation

Mode_Experim=2; % 0 - Image Simulated Noise, 1 - Image from File,
% 2 - experimental image

Mode_g_Color=0; % 0 - Grey; 1 - Image gRGB - Color g

% ----- Images f0, f - Read From File -----

p2_2_Image_FileName; % ; Dialog: Read Image FileName1

time_begin_All = clock; % Time begin All

st=FileName1;
FileNameP=strcat(pathname,st); % FileNameP = filename + path
disp('FileName1='); disp(FileName1);

p2_Image_Read; % Read Image (f0, f) from File

% Add theoretical Noise Level Sigma_N -----

Sigma_N=0.05; % SigmaN - Added Noise Level (theoretical)

fn=f;
if Mode_Test==1 % 0/1, 1 - Add Noise to Image, Open List File
p2_3_Create_Noise; % Add noise to Image fn, fn = fs + noise (Sigma_N)
end;

% p3_1_Profile_Manual; % Manual Read Profile from Image

Sigma_NE=Sigma_N;
if Mode_Noise_Calc==1 % 1 - Noise Calculation
p4_Noise_CalcL; % Sigma_NE - Gauss Noise Level Estimation
end;

p6_Filter; % Filter fn

% time All -----
```

```

% clock - [2015,5,23,23,8,34.703000000000000;]
% clock - [year,month,day,hour,minute,second]
time_end_All = clock; % Time end All
mdtime=time_end_All-time_begin_All;
dtime_All=((mdtime(3)*24+mdtime(4))*60+mdtime(5))*60+mdtime(6);
disp(' dtime_All,s = '); disp(dtime_All);
dtime_min=dtime_All/60;
% disp(' dtime,min = '); disp(dtime_min);

```

```

% p2_1_Image_Offset / Expand Image fne(Ms,Ns) to fe (Me,Ne)

```

```

% offset

```

```

Me=Ms+2*Mw2;
Ne=Ns+2*Nw2;

```

```

% disp('Me= '); disp(Me);
% disp('Ne= '); disp(Ne);
% disp('Mw2= '); disp(Mw2);
% disp('Nw2= '); disp(Nw2);

```

```

% fe=zeros(0);
fe=zeros(Me,Ne);
% for i=1:1:Me
% for k=1:1:Ne
% fe(i,k)=0;
% end; % k
% end; % i

```

```

iv=1:1:Ms; % vector
kv=1:1:Ns;
fe(iv+Mw2,kv+Nw2)=fne(iv,kv);
% for i=1:1:Ms
% for k=1:1:Ns
% fe(i+Mw2,k+Nw2)=fne(i,k);
% end; % k
% end; % i

```

```

% Mode_Offset -----

```

```

% Mode_Offset=0; % Zero Offset
Mode_Offset=1; % Symmetric Offset

```

```

% Zero Offset -----

```

```

if Mode_Offset==0; % Zero Offset
% fe - result
end; % Mode_Offset==0; % Zero Offset

```

```

% Symmetric Offset -----

```

```

if Mode_Offset==1; % Symmetric Offset

```

```

iv=1:1:Ms; % vector Left
kv=1:1:Nw2;
k1v=Nw2-kv+1;
for k=1:1:Nw2
if k1v(k)>Ns
k1v(k)=Ns;
end;
end; % k
fe(iv+Mw2,kv)=fne(iv,k1v);
% for k=1:1:Nw2 % Left
% k1=Nw2-k+1;
% if k1>Ns

```

```

% k1=Ns;
% end;
% for i=1:1:Ms
% fe(i+Mw2,k)=fne(i,k1);
% end; % i
% end; % k

% iv=1:1:Ms; % vector Right
% kv=1:1:Nw2;
k1v=Ns-kv+1;
for k=1:1:Nw2
if k1v(k)<1
k1v(k)=1;
end;
end; % k
fe(iv+Mw2,kv+Ns+Nw2)=fne(iv,k1v);
% for k=1:1:Nw2 % Right
% k1=Ns-k+1;
% if k1<1
% k1=1;
% end;
% for i=1:1:Ms
% fe(i+Mw2,k+Ns+Nw2)=fne(i,k1);
% end; % i
% end; % k

iv=1:1:Mw2; % vector Up
kv=1:1:Ne;
i1v=2*Mw2-iv+1;
for i=1:1:Mw2
if i1v(i)>(Ms+Mw2)
i1v(i)=(Ms+Mw2);
end;
end; % i
fe(iv,kv)=fe(i1v,kv);
% for i=1:1:Mw2 % Up
% i1=2*Mw2-i+1;
% if i1>(Ms+Mw2)
% i1=(Ms+Mw2);
% end;
% for k=1:1:Ne
% fe(i,k)=fe(i1,k);
% end; % k
% end; % i

% iv=1:1:Mw2; % vector Down
% kv=1:1:Ne;
i1v=Ms+Mw2-iv+1;
for i=1:1:Mw2
if i1v(i)<1
i1v(i)=1;
end;
end; % i
fe(iv+Ms+Mw2,kv)=fe(i1v,kv);
% for i=1:1:Mw2 % Down
% i1=Ms+Mw2-i+1;
% if i1<1
% i1=1;
% end;
% for k=1:1:Ne
% fe(i+Ms+Mw2,k)=fe(i1,k);
% end; % k
% end; % i

end; % Mode_Offset==1; % Symmetric Offset

```



```

% Sigma_N * 100% = Gauss Noise for Photoshop (Sigma_N = 0.1 = 10%)

mn=0; % average of distribution Noise
type_Name_Noise='gaussian';

fg=zeros(M,N);
fg=fg+0.5; % Image Grey for Add Noise

DN=Sigma_N.*Sigma_N; % DN - dispersion Noise
% fn=imnoise(f,'gaussian',mn,DN); % Add Noise to Image f

fn1=imnoise(fg,'gaussian',mn,DN)-0.5; % Add Noise to Image fg
fn=f+(fn1); % Add Noise to Image fs

% fn1=imnoise(fg,'gaussian',mn,DN)-0.5; % Add Noise to Image fg
% fn=fn+(fn1); % Add Noise to Image fs

% iiii
% ----- Image fn Show -----

Mode_Image_fn_Show=1; % 1 - show

if (Mode_Image_fn_Show==1) % scale_XY (i, k) - pixels
%set(0,'DefaultAxesFontSize',12,'DefaultAxesFontName','Arial Cyr'); % A4
%set(0,'DefaultTextFontSize',12,'DefaultTextFontName','Arial Cyr'); % A4
set(0,'DefaultAxesFontSize',13,'DefaultAxesFontName','Arial Cyr'); % A5
set(0,'DefaultTextFontSize',13,'DefaultTextFontName','Arial Cyr'); % A5
min_fn=min(min(fn));
max_fn=max(max(fn));
figure('Color','w');
% fig102=figure(102); % 102
% set(fig102, 'Color', [1 1 1]); % figure('Color','w');

imshow(fn,[]);
% impixelregion; % Show Pixel Region tool
Qmap=256;
map = gray(Qmap); cmap='grey'; % grey / black - grey - white
% map=jet(Qmap); cmap='jet'; % jet /dark blue - spectrum - dark red
% map = bone(Qmap); cmap='bone'; % bone / white-grey + blue-black
% map = copper(Qmap); cmap='copper'; % copper / black-brown-yellow
% map = pink(Qmap); cmap = 'pink'; % pink / white - brown - black
% map = hsv(Qmap); cmap = 'hsv'; % hsv / read - green - blue - red
% map = cool(Qmap); cmap = 'cool'; % cool / blue - purple
% map = hot(Qmap); cmap = 'hot'; % hot / black - red - white
% map = prism(Qmap); cmap='prism'; % prism / multiple All spectrum
% map = spring(Qmap); cmap = 'spring'; % spring / rose - yellow
% map = summer(Qmap); cmap = 'summer'; % summer / green - yellow
% map = autumn(Qmap); cmap = 'autumn'; % autumn / red - yellow
% map = winter(Qmap); cmap = 'winter'; % winter / blue - green
% map=1-map; % inverse color map
colormap(map);
not colorbar;
% colorbar;
axis on;
st='i ; M= ';
st=strcat(st,num2str(M));
st=strcat(st,' ; max fn= ');
st=strcat(st,num2str(max_fn,'%7.3f')); % '%5.1e'
ylabel(st);
st='k ; N= ';
st=strcat(st,num2str(N));
st=strcat(st,' ; min fn= ');
st=strcat(st,num2str(min_fn,'%7.3f'));
xlabel(st);

```



```

st='Image fn : ';
st=strcat(st,FileName1M);
st=strcat(st,'\sigma_N= ');
st=strcat(st,num2str(Sigma_N,'%5.2f'));
%st=strcat(st,'; cmap= ');
%st=strcat(st,cmap);
title(st);
end; % Mode_Image_fn_Show==1 / show

% p2_4_fn_g_Save; % Save Image fn, g to File

% ----- Image fn Write File -----

Mode_Image_fn_Write=0; % 1 - Write to File

if Mode_Image_fn_Write==1
% imwrite(fn,'fn1.bmp'); % *.bmp;*.gif;*.jpg;*.png;*.tif
% [filename, pathname] = uiputfile('.tif');
dTitle='Save the Image fn ';
[filename, pathname] = uiputfile('.bmp',dTitle);
FileNameP=strcat(pathname,filename);
disp(' FileNameP= '); disp(FileNameP);
imwrite(fn,FileNameP);
end; % if Image fn Write

Mode_Image_g_Write=0; % 1 - Write to File

if Mode_Image_g_Write==1
% imwrite(fn,'fn1.bmp'); % *.bmp;*.gif;*.jpg;*.png;*.tif
% [filename, pathname] = uiputfile('.tif');
dTitle='Save the Image g ';
[filename, pathname] = uiputfile('.bmp',dTitle);
FileNameP=strcat(pathname,filename);
disp(' FileNameP= '); disp(FileNameP);
imwrite(g,FileNameP);
end; % if Image g Write

Mode_Image_gB_Write=0; % 1 - Write to File

if Mode_Image_gB_Write==1
% imwrite(fn,'fn1.bmp'); % *.bmp;*.gif;*.jpg;*.png;*.tif
% [filename, pathname] = uiputfile('.tif');
dTitle='Save the Image g ';
[filename, pathname] = uiputfile('.bmp',dTitle);
FileNameP=strcat(pathname,filename);
disp(' FileNameP= '); disp(FileNameP);
imwrite(gB,FileNameP);
end; % if Image gB Write

Mode_Image_gBRGB_Write=1; % 1 - Write to File

if Mode_Image_gBRGB_Write==1
% imwrite(fn,'fn1.bmp'); % *.bmp;*.gif;*.jpg;*.png;*.tif
% [filename, pathname] = uiputfile('.tif');
dTitle='Save the Image g ';
[filename, pathname] = uiputfile('.bmp',dTitle);
FileNameP=strcat(pathname,filename);
disp(' FileNameP= '); disp(FileNameP);
imwrite(gBRGB,FileNameP);
end; % if Image gB Write

```

```

% p2_Image_Read; / Read Image (FileNameP) from File
% Input Image: f0, f

% disp('FileNameP='); disp(FileNameP);

% ----- Image info -----

info = imfinfo(FileNameP);
% disp('info = '); disp(info);

% disp('info.ColorType = '); disp(info.ColorType);
% disp('info.Format = '); disp(info.Format);

nColor_Type=0;

if (strcmp(info.ColorType,'grayscale'))
    nColor_Type=1;
    f0=imread(FileNameP);
end; %

if (strcmp(info.ColorType,'truecolor') | (strcmp(info.ColorType,'RGB')))
    nColor_Type=2;
    fRGB=imread(FileNameP);
    f0=rgb2gray(fRGB); % RGB to Gray
    fR=fRGB(:,:,1);
    fG=fRGB(:,:,2);
    fB=fRGB(:,:,3);
end; %

if (strcmp(info.ColorType,'indexed'))
    nColor_Type=3;
    [f0,map]=imread(FileNameP);
    f0=ind2gray(f0,map);
end; %

% ----- Image f0 Parameter -----

[M0,N0]=size(f0); % Image f0 Size

M=M0;
N=N0;
M02=floor(M0./2);
N02=floor(N0./2);
fq=f0; % initial image fq

M2=floor(M./2);
N2=floor(N./2);

% disp(' M= '); disp(M); disp(' N= '); disp(N);

% min_f0=min(min(f0)); max_f0=max(max(f0));
% disp(' min_f0= '); disp(min_f0);
% disp(' max_f0= '); disp(max_f0);

% Contrast -----

Mode_Contrast=0; % 0/1, 0 not Contrast, 1 - Contrast

if Mode_Contrast==0 % 0 - not Contrast
    f01=double(f0); % origin size (M0,N0);
    f01=double(f01./255.0); % f01-Input Image, Normalized Intensity: min-0, max-1
    f=double(fq);
    f=double(f./255.0); % f-Input Image, Normalized Intensity: min-0, max-1
end;

```

```

if Mode_Contrast==1 % 1 - Contrast
    f01=mat2gray(f0); % % origin size (M0,N0); f01 - scale [0..1]
    f=mat2gray(fq); % f - scale [0..1]
    % f=double(fq);
    % min_f0=min(min(f0)); max_f0=max(max(f0));
    % minr_f0=double(min_f0);
    % maxr_f0=double(max_f0);
    % if (maxr_f0-minr_f0)>0
    % f=double((f-minr_f0)*1.0/(maxr_f0-minr_f0)); % f - scale [0..1]
    % end;
end;

% min_f=min(min(f)); max_f=max(max(f));
% disp(' min_f= '); disp(min_f);
% disp(' max_f= '); disp(max_f);

x_min=1;
x_max=M;
y_min=1;
y_max=N;

% Disp Image Matrix -----

if (M<20) && (N<20)
% format short;
% format long g;
% format long e;
disp('f0= '); disp(f0);
disp('f= '); disp(f);
end; % disp(f0);

% ----- Image f0 Show -----

Mode_Image_f0_Show=1; % 1 - show

if (Mode_Image_f0_Show==1) % scale_XY (i, k) - pixels
% set(0,'DefaultAxesFontSize',11,'DefaultAxesFontName','Arial Cyr'); % A4
% set(0,'DefaultTextFontSize',11,'DefaultTextFontName','Arial Cyr'); % A4
set(0,'DefaultAxesFontSize',13,'DefaultAxesFontName','Arial Cyr'); % A5
set(0,'DefaultTextFontSize',13,'DefaultTextFontName','Arial Cyr'); % A5
min_f0=min(min(f0)); max_f0=max(max(f0));
figure('Color','w');
% fig101=figure(101); % 101
% set(fig101, 'Color', [1 1 1]); % figure('Color','w');

imshow(f0,[]);
% impixelregion; % Show Pixel Region tool
Qmap=256;
map = gray(Qmap); cmap='grey'; % grey / black - grey - white
% map=jet(Qmap); cmap='jet'; % jet / dark blue - spectrum - dark red
% map = bone(Qmap); cmap='bone'; % bone / white-grey + blue-black
% map = copper(Qmap); cmap='copper'; % copper / black-brown-yellow
% map = pink(Qmap); cmap = 'pink'; % pink / white - brown - black
% map = hsv(Qmap); cmap = 'hsv'; % hsv / red - green - blue - red
% map = cool(Qmap); cmap = 'cool'; % cool / blue - purple
% map = hot(Qmap); cmap = 'hot'; % hot / black - red - white
% map = prism(Qmap); cmap='prism'; % prism / multiple All spectrum
% map = spring(Qmap); cmap = 'spring'; % spring / rose - yellow
% map = summer(Qmap); cmap = 'summer'; % summer / green - yellow
% map = autumn(Qmap); cmap = 'autumn'; % autumn / red - yellow
% map = winter(Qmap); cmap = 'winter'; % winter / blue - green
% map=1-map; % inverse color map
% colormap(map);
not colorbar;
%colorbar;

```

```

axis on;
% axis square; % image - square
axis('image'); % pixel - square

st='i ; M= ';
st=strcat(st,num2str(M));
st=strcat(st,' ; max f0= ');
st=strcat(st,num2str(max_f0,'%4.0f')); % '%5.1e'
ylabel(st);
st='k ; N= ';
st=strcat(st,num2str(N));
st=strcat(st,' ; min f0= ');
st=strcat(st,num2str(min_f0,'%4.0f'));
xlabel(st);
st='Image f0 : ';
st=strcat(st,FileName1M);
% st=strcat(st,' ; map= ');
%st=strcat(st,cmap);
title(st);
end; % show f0

% ----- Image f Show -----

Mode_Image_f_Show=0; % 1 - show

if (Mode_Image_f_Show==1) % scale_XY (i, k) - pixels
% set(0,'DefaultAxesFontSize',11,'DefaultAxesFontName','Arial Cyr'); % A4
% set(0,'DefaultTextFontSize',11,'DefaultTextFontName','Arial Cyr'); % A4
set(0,'DefaultAxesFontSize',13,'DefaultAxesFontName','Arial Cyr'); % A5
set(0,'DefaultTextFontSize',13,'DefaultTextFontName','Arial Cyr'); % A5
min_f=min(min(f)); max_f=max(max(f));
figure('Color','w');

imshow(f,[]);
% impixelregion; % Show Pixel Region tool
Qmap=256;
map = gray(Qmap); cmap='grey'; % grey / black - grey - white
% map=jet(Qmap); cmap='jet'; % jet /dark blue - spectrum - dark red
% map = bone(Qmap); cmap='bone'; % bone / white-grey + blue-black
% map = copper(Qmap); cmap='copper'; % copper / black-brown-yellow
% map = pink(Qmap); cmap = 'pink'; % pink / white - brown - black
% map = hsv(Qmap); cmap = 'hsv'; % hsv / read - green - blue - red
% map = cool(Qmap); cmap = 'cool'; % cool / blue - purple
% map = hot(Qmap); cmap = 'hot'; % hot / black - red - white
% map = prism(Qmap); cmap='prism'; % prism / multiple All spectrum
% map = spring(Qmap); cmap = 'spring'; % spring / rose - yellow
% map = summer(Qmap); cmap = 'summer'; % summer / green - yellow
% map = autumn(Qmap); cmap = 'autumn'; % autumn / red - yellow
% map = winter(Qmap); cmap = 'winter'; % winter / blue - green
% map=1-map; % inverse color map
colormap(map);
not colorbar;
% colorbar;
axis on;
axis('image'); % pixel - square
st='i ; M= ';
st=strcat(st,num2str(M));
st=strcat(st,' ; max f= ');
st=strcat(st,num2str(max_f,'%7.3f')); % '%5.1e'
ylabel(st);
st='k ; N= ';
st=strcat(st,num2str(N));
st=strcat(st,' ; min f= ');
st=strcat(st,num2str(min_f,'%7.3f'));
xlabel(st);

```



```

    st=strcat(st,num2str(max_f,'%6.2f')); % '%5.1e'
    ylabel(st);
    st='k ; N= ';
    st=strcat(st,num2str(N));
    st=strcat(st,' ; min f= ');
    st=strcat(st,num2str(min_f,'%6.2f'));
    xlabel(st);
    st='Image f ';
    st=strcat(st,' ; map= ');
    st=strcat(st,cmap);
    title(st);
end; % if Image f0 + f Show; scale_XY - pixels

% ----- Image fRGB_ Show -----

Mode_Image_fRGB_Show=1; % 1 - show

if Mode_g_Color==0; % 0 - Grey; 1 - Image gRGB - Color g
    Mode_Image_fRGB_Show=0;
end;
if nColor_Type~=2 % 'truecolor', 'RGB'
    Mode_Image_fRGB_Show=0;
end;

if (Mode_Image_fRGB_Show==1) % scale_XY (i, k) - pixels
    min_f0=min(min(f0)); max_f0=max(max(f0));
    figure('Color','w');
    % fs=mat2gray(f0); % to scale [0..1]
    imshow(fRGB,[]);
    not colorbar;
    axis on;
    % axis square; % image - square
    axis('image'); % pixel - square
    st='i ; M= ';
    st=strcat(st,num2str(M));
    st=strcat(st,' ; max f0= ');
    st=strcat(st,num2str(max_f0,'%4.0f')); % '%5.1e'
    ylabel(st);
    st='k ; N= ';
    st=strcat(st,num2str(N));
    st=strcat(st,' ; min f0= ');
    st=strcat(st,num2str(min_f0,'%4.0f'));
    xlabel(st);
    st='Image fRGB : ';
    st=strcat(st,FileName1M);
    title(st);
end; % show fRGB

% p3_1_Profile_Manual; % Manual Read Profile from Image

p2_Image_Read; % Read Image (f0, f) from File

% Interactive Manual Profile -----

Mode_pr_Manual=1; % 0/1,1 - Mode Interactive Manual profile

if Mode_pr_Manual==1 % set Interactive Manual profile coordinate
    [mxP,myP,pr]=improfile(2);
    xP1=mxP(1); yP1=myP(1); % 1
    xP2=mxP(2); yP2=myP(2); % 2
    disp('mxP='); disp(mxP);
    disp('myP='); disp(myP);
end; % Mode_pr_Manual==1

```

```

p3_Read_Profile; % Read Profile from Image

% p3_2_Profile_Coord; % Read Profile from Image /Coordinate

p2_Image_Read; % Read Image (f0, f) from File

% Set Profile Coordinate -----

% Set Point : P1(xP1,yP1) - Begin of Profile and
               % P2(xP2,yP2) - End of Profile

%{
% Profile Strawberries
xP1=10;
yP1=100;
xP2=440;
yP2=140;
%}

% Profile: face_gr_G5.bmp
xP1=36; yP1=292;
xP2=87; yP2=225;

mxP=zeros(0);
myP=zeros(0);
mxP(1)=xP1; myP(1)=yP1;
mxP(2)=xP2; myP(2)=yP2;

% Interactive Manual Profile -----

Mode_pr_Manual=0; % 0/1,0 - profile coordinate

p3_Read_Profile; % Read Profile from Image

% p3_3_Save_Profile; / Save Profile zf(r) to File

disp(' ----- Save Profile zf (r) to Text File ----- ');
dTitle='Create the Text file /for Save Profile zf (r) /';
[filename, pathname] = uiputfile('*.txt',dTitle);
FileNamePS=strcat(pathname,filename);

[Ft,mes]=fopen(FileNamePS,'wt'); % Save zf(r) to File

for p=1:1:Qp
    fprintf(Ft,'%5.0f',p);
    fprintf(Ft,' ');
    fprintf(Ft,'%7.4f',zf(p));
    if (p<Qp)
        fprintf(Ft,'\n');
    end;
end; % p

fclose(Ft);

% p3_4_Save_Profile_FGB; / Save Profile zfR-G-B(r) to File

disp(' ----- Save Profile zf RGB(r) to Text File ----- ');
dTitle='Create the Text file /for Save Profile zf RGB(r) /';
[filename, pathname] = uiputfile('*.txt',dTitle);
FileNamePRGB=strcat(pathname,filename);

```

```

[Ft,mes]=fopen(FileNamePRGB,'wt'); % Save zfRGB(r) to File

for p=1:1:Qp
    fprintf(Ft,'%5.0f',p);
    fprintf(Ft,' ');
    fprintf(Ft,'%8.3f',zfR(p));
    fprintf(Ft,' ');
    fprintf(Ft,'%8.3f',zfG(p));
    fprintf(Ft,' ');
    fprintf(Ft,'%8.3f',zfB(p));
    if (p<Qp)
        fprintf(Ft,'\n');
    end;
end; % p

fclose(Ft);

% p3_Read_Profile; % Read Profile from Image f, fn
% Frofile data: from (p3_1_Profile_Manual) OR (p3_2_Profile_Coord)

% axis Profile Line y = k01*x + b01 -----

dx=xP2-xP1;
dy=yP2-yP1;

if abs(dx)<1E-6
    dx=1E-6;
end;
k01=dy/dx;
b01=yP1-k01*xP1; % y = k01*x + b01;

Alpha=atan2(dy,dx); % Alpha=atan2(k01); / arctan(k01)

% disp('Alpha, rad ='); disp(Alpha);
disp('Angle Profile : Alpha, degree ='); disp(Alpha*180/pi);

% Calculate profile / improfile -----

Mode_Calc_Prof=0; % calculate profile MATLAB function
% Mode_Calc_Prof=1; % calculate profile my

if Mode_Calc_Prof==0 % calculate profile MATLAB function
    pr_method='nearest';
    % pr_method='bilinear';
    % pr_method='bicubic';

    % zf=improfile(f,mxP,myP, 'bilinear');
    zf=improfile(f,mxP,myP, pr_method);
    zfn=improfile(fn,mxP,myP, pr_method);

    zfh=improfile(fh,mxP,myP, pr_method);
    zfd=improfile(fd,mxP,myP, pr_method);
    zfdc=improfile(fdc,mxP,myP, pr_method);

    zg=improfile(g,mxP,myP, pr_method);
    zgB=improfile(gB,mxP,myP, pr_method);

    if nColor_Type==2
        zfRGB=improfile(fRGB,mxP,myP, pr_method);
        zfR=zfRGB(:,1);
        zfG=zfRGB(:,2);
        zfB=zfRGB(:,3);
    end; % nColor_Type==2

```



```

Qp2=size(zf);
Qp=Qp2(1);
% disp('Qp='); disp(Qp);
end;

% Calculate profile / my method -----

if Mode_Calc_Prof==1 % calculate profile my
    zf=zeros(0);
    zfn=zeros(0);
    zfr=zeros(0); % R
    zfg=zeros(0); % G
    zfb=zeros(0); % B

    Qp=round(sqrt(dx^2+dy^2)); % Quantity Profile Point

    % disp('Qp='); disp(Qp);

% Mode Interpolation -----

% Mode_Interp=0; % Near
Mode_Interp=1; % BiLine

if Qp<=1
    disp('Qp<=1');
    break;
end;

% Near -----

if Mode_Interp==0 % near
    for p=1:1:Qp
        % xpr=xP1+(p-1)*cos(Alpha);
        % ypr=yP1+(p-1)*sin(Alpha);
        xpr=xP1+(p-1)*dx/(Qp-1);
        ypr=yP1+(p-1)*dy/(Qp-1);
        xp1=round(xpr);
        yp1=round(ypr);
        zf(p)=double(0);
        zfn(p)=double(0);
        if nColor_Type==2 % Color
            zfR(p)=double(0);
            zfG(p)=double(0);
            zfB(p)=double(0);
        end; %
        if (xp1>=1) && (xp1<=N) && (yp1>=1) && (yp1<=M)
            zf(p)=f(yp1,xp1);
            zfn(p)=fn(yp1,xp1);
            if nColor_Type==2
                zfr(p)=fR(yp1,xp1);
                zfg(p)=fG(yp1,xp1);
                zfb(p)=fB(yp1,xp1);
            end;
        end;
    end; % p
end; % Mode_Interp==0 % near

end; % Mode_Calc_Prof==1 % calculate profile my

% Create Profile -----

pr=1:1:Qp;

% ----- Matrix f Show. Real Dimension -----

```

```

Mode_Image_f_RD_Show=1; % 1 - show, Real Dimension

Mode_Grey_RGB=0; % 0 - Image f Grey; 1 - Image f RGB Color

if (Mode_Image_f_RD_Show==1) % x,y - Real Dimension
% set(0,'DefaultAxesFontSize',11,'DefaultAxesFontName','Arial Cyr'); % A4
% set(0,'DefaultTextFontSize',11,'DefaultTextFontName','Arial Cyr'); % A4
set(0,'DefaultAxesFontSize',13,'DefaultAxesFontName','Arial Cyr'); % A5
set(0,'DefaultTextFontSize',13,'DefaultTextFontName','Arial Cyr'); % A5
% set(0,'DefaultAxesFontSize',15,'DefaultAxesFontName','Arial Cyr'); % A5
% set(0,'DefaultTextFontSize',15,'DefaultTextFontName','Arial Cyr'); % A5

min_f=min(min(f)); max_f=max(max(f));
min_fn=min(min(fn)); max_fn=max(max(fn));

Mode_f_fn_Show=1; % 0- f, 1- fn

if Mode_f_fn_Show==0 % 0- f
fsh=f; % image for show
end;
if Mode_f_fn_Show==1 % 1- fn
fsh=fn; % image for show
end;

Mode_Reflect_y=0; % 1 - reflect, change Up / Down
if Mode_Reflect_y==1 % axis y - vertical
for i=1:1:M2
for k=1:1:N
i1=M-i+1;
r1=fsh(i1,k);
fsh(i1,k)=fsh(i,k);
fsh(i,k)=r1;
end; % k
end; % i
end; % Mode_Reflect_y==1

figure('Color','w');
% vX = x_min:Delta_x:x_max;
% vY = y_min:Delta_y:y_max;
x_minc=x_min-0.5; y_minc=y_min-0.5;
x_maxc=x_max-0.5; y_maxc=y_max-0.5;
vX = x_minc:1:x_maxc;
vY = y_minc:1:y_maxc;

if Mode_Grey_RGB==0; % 0 - Image f Grey
imshow(fsh,[]); % Grey Image
% imagesc(vX,vY,fsh); % imagesc(X,Y,W);
end;

if Mode_Grey_RGB==1; % 1 - Image f RGB Color
imshow(fRGB,[]); % RGB Color Image
end;

set(gca,'TickDir','out');

set(gca,'YDir','reverse'); % axes directions
% set(gca,'YDir','normal'); % axes directions

Qmap=256;
map = gray(Qmap); cmap='grey'; % grey / black - grey - white
% map=jet(Qmap); cmap='jet'; % jet /dark blue - spectrum - dark red
% map = bone(Qmap); cmap='bone'; % bone / white-grey + blue-black
% map = copper(Qmap); cmap='copper'; % copper / black-brown-yellow
% map = pink(Qmap); cmap = 'pink'; % pink / white - brown - black
% map = hsv(Qmap); cmap = 'hsv'; % hsv / read - green - blue - red

```

```

% map = cool(Qmap); cmap = 'cool'; % cool / blue - purple
% map = hot(Qmap); cmap = 'hot'; % hot / black - red - white
% map = prism(Qmap); cmap='prism'; % prism / multiple All spectrum
% map = spring(Qmap); cmap = 'spring'; % spring / rose - yellow
% map = summer(Qmap); cmap = 'summer'; % summer / green - yellow
% map = autumn(Qmap); cmap = 'autumn'; % autumn / red - yellow
% map = winter(Qmap); cmap = 'winter'; % winter / blue - green

% disp(' Qmap='); disp(Qmap);

% map=1-map; % inverse color map [0 ..1]

% disp('map(255,:'); disp(map(255,:));

colormap(map);
% not colorbar;

% colorbar;
axis on;
axis('image'); % pixel - square
if Mode_f_fn_Show==0 % 0- f
% st='y, pixels';
st='i, M=';
st=strcat(st,num2str(M));
st=strcat(st,' ; max f= ');
st=strcat(st,num2str(max_f,'%6.2f')); % '%5.1e'
ylabel(st);
st='k, N= ';
st=strcat(st,num2str(N));
st=strcat(st,' ; min f= ');
st=strcat(st,num2str(min_f,'%6.2f'));
xlabel(st);
st='f: ';
st=strcat(st,FileNameM);
end; % Mode_f_fn_Show==0 % 0- f

if Mode_f_fn_Show==1 % 1- f
% st='y, pixels';
st='i, M=';
st=strcat(st,num2str(M));
st=strcat(st,' ; max fn= ');
st=strcat(st,num2str(max_fn,'%6.2f')); % '%5.1e'
ylabel(st);
st='k, N= ';
st=strcat(st,num2str(N));
st=strcat(st,' ; min fn= ');
st=strcat(st,num2str(min_fn,'%6.2f'));
xlabel(st);
st='fn: ';
st=strcat(st,FileName1M);
end; % Mode_f_fn_Show==1 % 1- fn

% st=strcat(st,' ; map= ');
% st=strcat(st,cmap);
title(st);

% Show Profile Point -----

Mode_pr_Show=1; % 1- Show Profile Point

if Mode_pr_Show==1
hold on; % hold image

% Show Axis Line Profile P1 - P2 -----

```



```

%   k  black    d  diamond
%   w  white    v  triangle (down)
%                   ^  triangle (up)
%                   <  triangle (left)
%                   >  triangle (right)
%                   p  pentagram
%                   h  hexagram
figure('Color','w');

plot(pr,zf,'b-','LineWidth',2); % Profile

% plot(pr,zf,'b-','color',[0.500 0.500 0.000],'LineWidth',2); % Profile

% plot(pr,zf,'b-',...
%   'LineWidth',2,...
%   'MarkerEdgeColor','r',...
%   'MarkerFaceColor','y',...
%   'MarkerSize',4); % Profile

axis on;

% axis('equal');
vaxis = axis; % axis([xmin xmax ymin ymax])
if Qp>1
    vaxis(1)=1;
end;
vaxis(2)=Qp;
axis(vaxis);

xlabel('k');
ylabel('z');
st=' zf(r) ; ';
st=strcat(st,' f : ');
st=strcat(st,FileName1M);
st=strcat(st,' ; Qp = ');
st=strcat(st,num2str(Qp));
title(st);
% legend ('zf');
end; % Mode_pr_f_Show

% fn show -----

Mode_pr_fn_Show=1;

if Mode_pr_fn_Show==1
% set(0,'DefaultAxesFontSize',11,'DefaultAxesFontName','Arial Cyr'); % A4
% set(0,'DefaultTextFontSize',11,'DefaultTextFontName','Arial Cyr'); % A4
% set(0,'DefaultAxesFontSize',13,'DefaultAxesFontName','Arial Cyr'); % A5
% set(0,'DefaultTextFontSize',13,'DefaultTextFontName','Arial Cyr'); % A5
set(0,'DefaultAxesFontSize',15,'DefaultAxesFontName','Arial Cyr'); % A5
set(0,'DefaultTextFontSize',15,'DefaultTextFontName','Arial Cyr'); % A5
% Various line types, plot symbols and colors may be obtained with
% PLOT(X,Y,S) where S is a character string made from one element
% from any or all the following 3 columns:
%   b  blue    .  point    -  solid
figure('Color','w');
plot(pr,zfn,'b-','LineWidth',2); % Profile
axis on;
% axis('equal');
vaxis = axis; % axis([xmin xmax ymin ymax])
if Qp>1
    vaxis(1)=1;
end;
vaxis(2)=Qp;
axis(vaxis);

```

```

xlabel('r');
ylabel('z');
st=' zfn(r) ;';
st=strcat(st,' fn : ');
st=strcat(st,FileName1M);
st=strcat(st,' ; Qp = ');
st=strcat(st,num2str(Qp));
title(st);
% legend ('zfn');
end; % Mode_pr_fn_Show

% f_fn show -----

Mode_pr_f_fn_Show=0;

if Mode_pr_f_fn_Show==1
% set(0,'DefaultAxesFontSize',11,'DefaultAxesFontName','Arial Cyr'); % A4
% set(0,'DefaultTextFontSize',11,'DefaultTextFontName','Arial Cyr'); % A4
% set(0,'DefaultAxesFontSize',13,'DefaultAxesFontName','Arial Cyr'); % A5
% set(0,'DefaultTextFontSize',13,'DefaultTextFontName','Arial Cyr'); % A5
% set(0,'DefaultAxesFontSize',15,'DefaultAxesFontName','Arial Cyr'); % A5
% set(0,'DefaultTextFontSize',15,'DefaultTextFontName','Arial Cyr'); % A5
% Various line types, plot symbols and colors may be obtained with
% PLOT(X,Y,S) where S is a character string made from one element
% from any or all the following 3 columns:
% b blue . point - solid
figure('Color','w');
plot(pr,zf,'r--',pr,zfn,'b-', 'LineWidth',2); % Profile
axis on;
% axis('equal');
vaxis = axis; % axis([xmin xmax ymin ymax])
if Qp>1
vaxis(1)=1;
end;
vaxis(2)=Qp;
axis(vaxis);
xlabel('r');
ylabel('z');
st=' zf(r) ;';
st=strcat(st,' f : ');
st=strcat(st,FileName1M);
st=strcat(st,' ; Qp = ');
st=strcat(st,num2str(Qp));
title(st);
legend ('zf','zfn');
end; % Mode_pr_f_fn_Show

% fh show -----

Mode_pr_fh_Show=1;

if Mode_pr_fh_Show==1
% set(0,'DefaultAxesFontSize',11,'DefaultAxesFontName','Arial Cyr'); % A4
% set(0,'DefaultTextFontSize',11,'DefaultTextFontName','Arial Cyr'); % A4
% set(0,'DefaultAxesFontSize',13,'DefaultAxesFontName','Arial Cyr'); % A5
% set(0,'DefaultTextFontSize',13,'DefaultTextFontName','Arial Cyr'); % A5
% set(0,'DefaultAxesFontSize',15,'DefaultAxesFontName','Arial Cyr'); % A5
% set(0,'DefaultTextFontSize',15,'DefaultTextFontName','Arial Cyr'); % A5
% Various line types, plot symbols and colors may be obtained with
% PLOT(X,Y,S) where S is a character string made from one element
% from any or all the following 3 columns:
% b blue . point - solid
figure('Color','w');
plot(pr,zfh,'b-', 'LineWidth',2); % Profile
axis on;

```

```

% axis('equal');
vaxis = axis([xmin xmax ymin ymax])
if Qp>1
    vaxis(1)=1;
end;
vaxis(2)=Qp;
axis(vaxis);
xlabel('k');
ylabel('z');
st=' zfh(k) ;';
st=strcat(st,' f : ');
st=strcat(st,FileName1M);
st=strcat(st,' ; Qp = ');
st=strcat(st,num2str(Qp));
title(st);
legend ('fh');
end; % Mode_pr_fh_Show

% fd_fdc show -----

Mode_pr_fd_fdc_Show=1;

if Mode_pr_fd_fdc_Show==1
% set(0,'DefaultAxesFontSize',11,'DefaultAxesFontName','Arial Cyr'); % A4
% set(0,'DefaultTextFontSize',11,'DefaultTextFontName','Arial Cyr'); % A4
% set(0,'DefaultAxesFontSize',13,'DefaultAxesFontName','Arial Cyr'); % A5
% set(0,'DefaultTextFontSize',13,'DefaultTextFontName','Arial Cyr'); % A5
set(0,'DefaultAxesFontSize',15,'DefaultAxesFontName','Arial Cyr'); % A5
set(0,'DefaultTextFontSize',15,'DefaultTextFontName','Arial Cyr'); % A5
% Various line types, plot symbols and colors may be obtained with
% PLOT(X,Y,S) where S is a character string made from one element
% from any or all the following 3 columns:
%   b blue      . point      - solid
figure('Color','w');
plot(pr,zfd,'b-',pr,zfdc,'r--','LineWidth',2); % Profile
axis on;
% axis('equal');
vaxis = axis([xmin xmax ymin ymax])
if Qp>1
    vaxis(1)=1;
end;
vaxis(2)=Qp;
axis(vaxis);
xlabel('k');
ylabel('z');
st=' zf(k) ;';
st=strcat(st,' f : ');
st=strcat(st,FileName1M);
st=strcat(st,' ; Qp = ');
st=strcat(st,num2str(Qp));
title(st);
legend ('fd','fdc');
end; % Mode_pr_fd_fdc_Show

% fd show -----

Mode_pr_fd_show=1;

if Mode_pr_fd_fdc_Show==1
% set(0,'DefaultAxesFontSize',11,'DefaultAxesFontName','Arial Cyr'); % A4
% set(0,'DefaultTextFontSize',11,'DefaultTextFontName','Arial Cyr'); % A4
% set(0,'DefaultAxesFontSize',13,'DefaultAxesFontName','Arial Cyr'); % A5
% set(0,'DefaultTextFontSize',13,'DefaultTextFontName','Arial Cyr'); % A5
set(0,'DefaultAxesFontSize',15,'DefaultAxesFontName','Arial Cyr'); % A5
set(0,'DefaultTextFontSize',15,'DefaultTextFontName','Arial Cyr'); % A5

```

```

% Various line types, plot symbols and colors may be obtained with
% PLOT(X,Y,S) where S is a character string made from one element
% from any or all the following 3 columns:
%   b  blue      .  point      -  solid
figure('Color','w');
plot(pr,zfd,'b-', 'LineWidth',2); % Profile
axis on;
% axis('equal');
vaxis = axis; % axis([xmin xmax ymin ymax])
if Qp>1
    vaxis(1)=1;
end;
vaxis(2)=Qp;
axis(vaxis);
xlabel('k');
ylabel('z');
st=' zfd(k) ;';
st=strcat(st,' f : ');
st=strcat(st,FileName1M);
st=strcat(st,' ; Qp = ');
st=strcat(st,num2str(Qp));
title(st);
legend ('fd');
end; % Mode_pr_fd_Show

% fdc show -----

Mode_pr_fdc_Show=1;

if Mode_pr_fdc_Show==1
% set(0,'DefaultAxesFontSize',11,'DefaultAxesFontName','Arial Cyr'); % A4
% set(0,'DefaultTextFontSize',11,'DefaultTextFontName','Arial Cyr'); % A4
% set(0,'DefaultAxesFontSize',13,'DefaultAxesFontName','Arial Cyr'); % A5
% set(0,'DefaultTextFontSize',13,'DefaultTextFontName','Arial Cyr'); % A5
set(0,'DefaultAxesFontSize',15,'DefaultAxesFontName','Arial Cyr'); % A5
set(0,'DefaultTextFontSize',15,'DefaultTextFontName','Arial Cyr'); % A5
% Various line types, plot symbols and colors may be obtained with
% PLOT(X,Y,S) where S is a character string made from one element
% from any or all the following 3 columns:
%   b  blue      .  point      -  solid
figure('Color','w');
plot(pr,zfdc,'r--', 'LineWidth',2); % Profile
axis on;
% axis('equal');
vaxis = axis; % axis([xmin xmax ymin ymax])
if Qp>1
    vaxis(1)=1;
end;
vaxis(2)=Qp;
axis(vaxis);
xlabel('k');
ylabel('z');
st=' zfdc(k) ;';
st=strcat(st,' f : ');
st=strcat(st,FileName1M);
st=strcat(st,' ; Qp = ');
st=strcat(st,num2str(Qp));
title(st);
legend ('fdc');
end; % Mode_pr_fdc_Show

% fn_g show -----

Mode_pr_fn_g_Show=1;

```



```

if Mode_pr_fn_g_Show==1
% set(0,'DefaultAxesFontSize',11,'DefaultAxesFontName','Arial Cyr'); % A4
% set(0,'DefaultTextFontSize',11,'DefaultTextFontName','Arial Cyr'); % A4
% set(0,'DefaultAxesFontSize',13,'DefaultAxesFontName','Arial Cyr'); % A5
% set(0,'DefaultTextFontSize',13,'DefaultTextFontName','Arial Cyr'); % A5
set(0,'DefaultAxesFontSize',15,'DefaultAxesFontName','Arial Cyr'); % A5
set(0,'DefaultTextFontSize',15,'DefaultTextFontName','Arial Cyr'); % A5
% Various line types, plot symbols and colors may be obtained with
% PLOT(X,Y,S) where S is a character string made from one element
% from any or all the following 3 columns:
%   b blue      . point      - solid
figure('Color','w');
plot(pr,zfn,'r--',pr,zg,'b-', 'LineWidth',2); % Profile
axis on;
% axis('equal');
vaxis = axis; % axis([xmin xmax ymin ymax])
if Qp>1
    vaxis(1)=1;
end;
vaxis(2)=Qp;
axis(vaxis);
xlabel('r');
ylabel('z');
st=' zf(r) ';
st=strcat(st,' f : ');
st=strcat(st,FileName1M);
st=strcat(st,' ; Qp = ');
st=strcat(st,num2str(Qp));
title(st);
legend('zfn','zg');
end; % Mode_pr_fn_g_Show

```

% fn_gB show -----

```
Mode_pr_fn_gB_Show=1;
```

```

if Mode_pr_fn_gB_Show==1
% set(0,'DefaultAxesFontSize',11,'DefaultAxesFontName','Arial Cyr'); % A4
% set(0,'DefaultTextFontSize',11,'DefaultTextFontName','Arial Cyr'); % A4
% set(0,'DefaultAxesFontSize',13,'DefaultAxesFontName','Arial Cyr'); % A5
% set(0,'DefaultTextFontSize',13,'DefaultTextFontName','Arial Cyr'); % A5
set(0,'DefaultAxesFontSize',15,'DefaultAxesFontName','Arial Cyr'); % A5
set(0,'DefaultTextFontSize',15,'DefaultTextFontName','Arial Cyr'); % A5
% Various line types, plot symbols and colors may be obtained with
% PLOT(X,Y,S) where S is a character string made from one element
% from any or all the following 3 columns:
%   b blue      . point      - solid
figure('Color','w');
plot(pr,zfn,'r--',pr,zgB,'b-', 'LineWidth',2); % Profile
axis on;
% axis('equal');
vaxis = axis; % axis([xmin xmax ymin ymax])
if Qp>1
    vaxis(1)=1;
end;
vaxis(2)=Qp;
axis(vaxis);
xlabel('r');
ylabel('z');
st=' zf(r) ';
st=strcat(st,' f : ');
st=strcat(st,FileName1M);
st=strcat(st,' ; Qp = ');
st=strcat(st,num2str(Qp));
title(st);

```



```

imshow(gG,[]);
%  impixelregion; % Show Pixel Region tool
    Qmap=256;
    map = gray(Qmap); cmap='grey'; % grey / black - grey - white
% map=jet(Qmap); cmap='jet'; % jet /dark blue - spectrum - dark red
% map = bone(Qmap); cmap='bone'; % bone / white-grey + blue-black
% map = copper(Qmap); cmap='copper'; % copper / black-brown-yellow
% map = pink(Qmap); cmap = 'pink'; % pink / white - brown - black
% map = hsv(Qmap); cmap = 'hsv'; % hsv / read - green - blue - red
% map = cool(Qmap); cmap = 'cool'; % cool / blue - purple
% map = hot(Qmap); cmap = 'hot'; % hot / black - red - white
% map = prism(Qmap); cmap='prism'; % prism / multiple All spectrum
% map = spring(Qmap); cmap = 'spring'; % spring / rose - yellow
% map = summer(Qmap); cmap = 'summer'; % summer / green - yellow
% map = autumn(Qmap); cmap = 'autumn'; % autumn / red - yellow
% map = winter(Qmap); cmap = 'winter'; % winter / blue - green
% map=1-map; % inverse color map
colormap(map);
% not colorbar;
colorbar;
axis on;
% axis square; % image - square
axis('image'); % pixel - square

st='i ; M= ';
st=strcat(st,num2str(M));
st=strcat(st,' ; max gG= ');
st=strcat(st,num2str(max_gG,'%4.2f')); % '%5.1e'
ylabel(st);
st='k ; N= ';
st=strcat(st,num2str(N));
st=strcat(st,' ; min gG= ');
st=strcat(st,num2str(min_gG,'%4.2f'));
xlabel(st);
st='Image gG : ';
% st=strcat(st,FileName1);
st=strcat(st,'S7B');
st=strcat(st,' ; \gamma= ');
st=strcat(st,num2str(Gamma,'%4.2f'));
title(st);
end; % show gG

% p2_2_Image_FileName; Dialog: Read Image FileName
% Output: FileName1

disp(' ----- New Image FileName Read ----- ');

dTitle='Select the Image: Read f0 from Image File';
[filename, pathname] = uigetfile('*.bmp;*.gif;*.jpg;*.png;*.tif',dTitle);
FileName1=filename;

% disp(' filename= '); disp(filename);
% disp(' pathname= '); disp(pathname);
% disp(' filenameP= '); disp(filenameP);

FileName1M=FileName1;
for i=1:length(FileName1)
    if FileName1(i)=='_'
        FileName1M(i)=''; % filename without '_' for title
    end; % if
end; % i

```



```

% p2_Image_Read; / Read Image (FileNameP) from File
% Input Image: f0, f

% disp('FileNameP='); disp(FileNameP);

% ----- Image info -----

info = imfinfo(FileNameP);
% disp('info = '); disp(info);

% disp('info.ColorType = '); disp(info.ColorType);
% disp('info.Format = '); disp(info.Format);

nColor_Type=0;

if (strcmp(info.ColorType,'grayscale'))
    nColor_Type=1;
    f0=imread(FileNameP);
end; %

if (strcmp(info.ColorType,'truecolor') | (strcmp(info.ColorType,'RGB')))
    nColor_Type=2;
    fRGB=imread(FileNameP);
    f0=rgb2gray(fRGB); % RGB to Gray
    fR=fRGB(:,:,1);
    fG=fRGB(:,:,2);
    fB=fRGB(:,:,3);
end; %

if (strcmp(info.ColorType,'indexed'))
    nColor_Type=3;
    [f0,map]=imread(FileNameP);
    f0=ind2gray(f0,map);
end; %

% ----- Image f0 Parameter -----

[M0,N0]=size(f0); % Image f0 Size

M=M0;
N=N0;
M02=floor(M0./2);
N02=floor(N0./2);
fq=f0; % initial image fq

M2=floor(M./2);
N2=floor(N./2);

% disp(' M= '); disp(M); disp(' N= '); disp(N);

% min_f0=min(min(f0)); max_f0=max(max(f0));
% disp(' min_f0= '); disp(min_f0);
% disp(' max_f0= '); disp(max_f0);

% Contrast -----

Mode_Contrast=0; % 0/1, 0 not Contrast, 1 - Contrast

if Mode_Contrast==0 % 0 - not Contrast
    f01=double(f0); % origin size (M0,N0);
    f01=double(f01./255.0); % f01-Input Image, Normalized Intensity: min-0, max-1
    f=double(fq);
    f=double(f./255.0); % f-Input Image, Normalized Intensity: min-0, max-1
end;

```

```

if Mode_Contrast==1 % 1 - Contrast
    f01=mat2gray(f0); % % origin size (M0,N0); f01 - scale [0..1]
    f=mat2gray(fq); % f - scale [0..1]
    % f=double(fq);
    % min_f0=min(min(f0)); max_f0=max(max(f0));
    % minr_f0=double(min_f0);
    % maxr_f0=double(max_f0);
    % if (maxr_f0-minr_f0)>0
    % f=double((f-minr_f0)*1.0/(maxr_f0-minr_f0)); % f - scale [0..1]
    % end;
end;

% min_f=min(min(f)); max_f=max(max(f));
% disp(' min_f= '); disp(min_f);
% disp(' max_f= '); disp(max_f);

x_min=1;
x_max=M;
y_min=1;
y_max=N;

% Disp Image Matrix -----

if (M<20) && (N<20)
    % format short;
    % format long g;
    % format long e;
    disp('f0= '); disp(f0);
    disp('f= '); disp(f);
end; % disp(f0);

% ----- Image f0 Show -----

Mode_Image_f0_Show=0; % 1 - show

if (Mode_Image_f0_Show==1) % scale_XY (i, k) - pixels
% set(0,'DefaultAxesFontSize',11,'DefaultAxesFontName','Arial Cyr'); % A4
% set(0,'DefaultTextFontSize',11,'DefaultTextFontName','Arial Cyr'); % A4
set(0,'DefaultAxesFontSize',13,'DefaultAxesFontName','Arial Cyr'); % A5
set(0,'DefaultTextFontSize',13,'DefaultTextFontName','Arial Cyr'); % A5
    min_f0=min(min(f0)); max_f0=max(max(f0));
    figure('Color','w');
    % fig101=figure(101); % 101
    % set(fig101,'Color',[1 1 1]); % figure('Color','w');

imshow(f0,[]);
    % impixelregion; % Show Pixel Region tool
    Qmap=256;
    map = gray(Qmap); cmap='grey'; % grey / black - grey - white
    % map=jet(Qmap); cmap='jet'; % jet /dark blue - spectrum - dark red
    % map = bone(Qmap); cmap='bone'; % bone / white-grey + blue-black
    % map = copper(Qmap); cmap='copper'; % copper / black-brown-yellow
    % map = pink(Qmap); cmap = 'pink'; % pink / white - brown - black
    % map = hsv(Qmap); cmap = 'hsv'; % hsv / read - green - blue - red
    % map = cool(Qmap); cmap = 'cool'; % cool / blue - purple
    % map = hot(Qmap); cmap = 'hot'; % hot / black - red - white
    % map = prism(Qmap); cmap='prism'; % prism / multiple All spectrum
    % map = spring(Qmap); cmap = 'spring'; % spring / rose - yellow
    % map = summer(Qmap); cmap = 'summer'; % summer / green - yellow
    % map = autumn(Qmap); cmap = 'autumn'; % autumn / red - yellow
    % map = winter(Qmap); cmap = 'winter'; % winter / blue - green
    % map=1-map; % inverse color map
    % colormap(map);
    not colorbar;

```

```

%colorbar;
axis on;
% axis square; % image - square
axis('image'); % pixel - square

st='i ; M= ';
st=strcat(st,num2str(M));
st=strcat(st,' ; max f0= ');
st=strcat(st,num2str(max_f0,'%4.0f')); % '%5.1e'
ylabel(st);
st='k ; N= ';
st=strcat(st,num2str(N));
st=strcat(st,' ; min f0= ');
st=strcat(st,num2str(min_f0,'%4.0f'));
xlabel(st);
st='Image f0 : ';
st=strcat(st,FileName1M);
% st=strcat(st,' ; map= ');
%st=strcat(st,cmap);
title(st);
end; % show f0

% ----- Image f Show -----

Mode_Image_f_Show=1; % 1 - show

if (Mode_Image_f_Show==1) % scale_XY (i, k) - pixels
% set(0,'DefaultAxesFontSize',11,'DefaultAxesFontName','Arial Cyr'); % A4
% set(0,'DefaultTextFontSize',11,'DefaultTextFontName','Arial Cyr'); % A4
set(0,'DefaultAxesFontSize',13,'DefaultAxesFontName','Arial Cyr'); % A5
set(0,'DefaultTextFontSize',13,'DefaultTextFontName','Arial Cyr'); % A5
min_f=min(min(f)); max_f=max(max(f));
figure('Color','w');

imshow(f,[]);
% impixelregion; % Show Pixel Region tool
Qmap=256;
map = gray(Qmap); cmap='grey'; % grey / black - grey - white
% map=jet(Qmap); cmap='jet'; % jet /dark blue - spectrum - dark red
% map = bone(Qmap); cmap='bone'; % bone / white-grey + blue-black
% map = copper(Qmap); cmap='copper'; % copper / black-brown-yellow
% map = pink(Qmap); cmap = 'pink'; % pink / white - brown - black
% map = hsv(Qmap); cmap = 'hsv'; % hsv / read - green - blue - red
% map = cool(Qmap); cmap = 'cool'; % cool / blue - purple
% map = hot(Qmap); cmap = 'hot'; % hot / black - red - white
% map = prism(Qmap); cmap='prism'; % prism / multiple All spectrum
% map = spring(Qmap); cmap = 'spring'; % spring / rose - yellow
% map = summer(Qmap); cmap = 'summer'; % summer / green - yellow
% map = autumn(Qmap); cmap = 'autumn'; % autumn / red - yellow
% map = winter(Qmap); cmap = 'winter'; % winter / blue - green
% map=1-map; % inverse color map
colormap(map);
not colorbar;
% colorbar;
axis on;
axis('image'); % pixel - square
st='i ; M= ';
st=strcat(st,num2str(M));
st=strcat(st,' ; max f= ');
st=strcat(st,num2str(max_f,'%7.3f')); % '%5.1e'
ylabel(st);
st='k ; N= ';
st=strcat(st,num2str(N));
st=strcat(st,' ; min f= ');
st=strcat(st,num2str(min_f,'%7.3f'));

```



```

st=strcat(st,' ; max f= ');
st=strcat(st,num2str(max_f,'%6.2f')); % '%5.1e'
ylabel(st);
st='k ; N= ';
st=strcat(st,num2str(N));
st=strcat(st,' ; min f= ');
st=strcat(st,num2str(min_f,'%6.2f'));
xlabel(st);
st='Image f ';
st=strcat(st,' ; map= ');
st=strcat(st,cmap);
title(st);
end; % if Image f0 + f Show; scale_XY - pixels

% ----- Image fRGB_ Show -----

Mode_Image_fRGB_Show=0; % 1 - show

if nColor_Type~=2 % 'truecolor', 'RGB'
    Mode_Image_fRGB_Show=0;
end;

if (Mode_Image_fRGB_Show==1) % scale_XY (i, k) - pixels
    min_f0=min(min(f0)); max_f0=max(max(f0));
    figure('Color','w');
    % fs=mat2gray(f0); % to scale [0..1]
    imshow(fRGB,[]);
    not colorbar;
axis on;
% axis square; % image - square
axis('image'); % pixel - square
st='i ; M= ';
st=strcat(st,num2str(M));
st=strcat(st,' ; max f0= ');
st=strcat(st,num2str(max_f0,'%4.0f')); % '%5.1e'
ylabel(st);
st='k ; N= ';
st=strcat(st,num2str(N));
st=strcat(st,' ; min f0= ');
st=strcat(st,num2str(min_f0,'%4.0f'));
xlabel(st);
st='Image fRGB : ';
st=strcat(st,FileName1M);
title(st);
end; % show fRGB

```