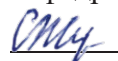


**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені
ІГОРЯ СІКОРСЬКОГО»**

**Радіотехнічний факультет
Кафедра радіотехнічних систем**

До захисту допущено: Завідувач
кафедри

 Сергій ЖУК

«16» 06 2025 р.

Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Радіотехнічні комп'ютеризовані системи»
спеціальності 172 «Телекомунікації та радіотехніка»
на тему: «Алгоритм придушення завад на текстурних
зображеннях»

Виконав :
студент IV курсу, групи РС-11

Бабій Владислав Вікторович



Керівник:
СТ.ВИКЛ., К.Т.Н.
Вишневий Сергій Валерійович



Рецензент:
СТ.ВИКЛ.
Сокольський Сергій Олегович



Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів без
відповідних посилань.

Студент 

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Радіотехнічний факультет
Кафедра радіотехнічних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – **172 «Телекомунікації та радіотехніка»**

Освітня програма «**Радіотехнічні комп'ютеризовані системи**»

ЗАТВЕРДЖУЮ

Завідувач кафедри

 Сергій ЖУК

« 14 » 04 _____ 2025 р.

ЗАВДАННЯ

на дипломну роботу студенту

Бабію Владиславу Вікторовичу

1. Тема роботи «Алгоритм придушення завад на текстурних зображеннях», керівник роботи Вишневий Сергій Валерійович, ст.викл., к.т.н. затверджені наказом по університету від «29» 05. 2025р. №1841-с
2. Термін подання студентом роботи 16.06.2025р.
3. Вихідні дані до роботи Виконати дослідження фільтрації обраними методами при обробці синтезованих та реальних зображень.
4. Зміст роботи 1. Огляд практичних задач, в яких виникає необхідність виконувати обробку цифрових зображень. 2. Технології отримання зображення в різних технічних системах. Поява завад на зображеннях. Огляд видів завад.

3.Огляд методів придушення завад. Вибір фільтрів, які будуть досліджуватися в роботі 4. Виконання моделювання в MATLAB шляхом програмної реалізації алгоритму фільтрації та аналіз отриманих результатів при обробці штучних синтезованих зображень та реальних зображень. 5. Висновки.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) Презентація у формі слайдів.

6. Дата видачі завдання 14 квітня 2025р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Огляд практичних задач, в яких виникає необхідність виконувати обробку цифрових зображень.	14.04-25.04	Виконано
2	Технології отримання зображення в різних технічних системах. Поява завад на зображеннях. Огляд видів завад.	25.04-06.05	Виконано
3	Огляд методів придушення завад. Вибір фільтрів, які будуть досліджуватися в роботі	06.05-18.05	Виконано
4	Виконання моделювання в MATLAB шляхом програмної реалізації алгоритму фільтрації та аналіз отриманих результатів при обробці штучних синтезованих зображень та реальних зображень	18.05-01.06	Виконано
5	Оформлення пояснювальної записки	01.06-12.06	Виконано

Студент


(підпис)

Керівник роботи


(підпис)

В.В. Бабій

(ініціали, прізвище)

С.В.Вишневий

(ініціали, прізвище)

АНОТАЦІЯ

Дипломна робота включає в себе пояснювальну записку обсягом 81 сторінок, 52 рисунки, 34 посилання, 1 додаток.

Актуальність дослідження полягає в тому, що в умовах широкого використання цифрових зображень у різних сферах, зокрема в медицині та супутникових знімках, якість зображень має критичне значення. Придушення завад на текстурних зображеннях є важливим для забезпечення ефективної обробки й аналізу візуальної інформації.

Мета дослідження: аналіз алгоритму фільтрації цифрових текстурних зображень, які спотворені дією завади, шляхом проведення комп'ютерного моделювання в програмному середовищі MATLAB

Поставлено такі завдання: проаналізувати практичні задачі обробки зображень, дослідити типи завад, оглянути існуючі методи їх усунення та розробити і реалізувати вибрані підходи в середовищі MATLAB, а також провести їхній аналіз і оцінку ефективності.

Об'єкт дослідження: фільтрація текстурних зображень, що спотворені дією завади.

Предмет дослідження: алгоритм придушення завади на текстурних зображеннях.

Практичне значення роботи полягає у можливості застосування реалізованих підходів для покращення якості зображень у реальних умовах. Це дозволить підвищити точність і ефективність обробки візуальної інформації в таких сферах, як медицина, дистанційне зондування, промисловість та інших галузях, де висока якість зображень є важливим чинником.

Ключові слова: обробка цифрових зображень, придушення завад, спекл-шум, методи придушення завад, фільтрація зображень, MATLAB, види завад на зображеннях, алгоритми обробки зображень.

ANNOTATION

The thesis includes an explanatory note of 81 pages, 52 figures, 34 references, and 1 appendix.

The relevance of the study is that with the widespread use of digital images in various fields, including medicine and satellite photography, image quality is critical. Suppression of interference in texture images is important to ensure efficient processing and analysis of visual information.

The purpose of the study is to develop methods to reduce the impact of noise on texture images in order to improve their quality.

The following tasks have been set: to analyze practical tasks of image processing, to investigate the types of noise, to review existing methods of their elimination, and to develop and implement the selected approaches in MATLAB, as well as to analyze and evaluate their effectiveness.

Object of study: filtering texture images distorted by interference.

The subject of research: an algorithm for suppressing noise in texture images.

The practical significance of the work lies in the possibility of applying the implemented approaches to improve the quality of images in real conditions. This will increase the accuracy and efficiency of visual information processing in such areas as medicine, remote sensing, industry, and other areas where high image quality is an important factor.

Keywords: digital image processing, noise reduction, speckle noise, noise reduction methods, image filtering, MATLAB, types of noise in images, image processing algorithms.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	8
ВСТУП.....	9
1 ОГЛЯД ПРАКТИЧНИХ ЗАДАЧ, В ЯКИХ ВИНИКАЄ НЕОБХІДНІСТЬ ВИКОНУВАТИ ОБРОБКУ ЦИФРОВИХ ЗОБРАЖЕНЬ.....	10
1.1 Медицина	10
1.2 Сільське господарство	12
1.3 Система розпізнавання обличчя	14
1.4 Відеоаналітика для розпізнавання номерних знаків	15
1.5 Відновлення зображень	16
1.6 Зображення отримані за допомогою радарів із синтезованою апертурою (RSA)	17
1.7 Системи допомоги водію	18
1.8 Комп'ютерне моделювання	19
1.9 Висновки до першого розділу.....	20
2 ТЕХНОЛОГІЇ ОТРИМАННЯ ЗОБРАЖЕНЬ У ТЕХНІЧНИХ СИСТЕМАХ ТА ВИДИ ЗАВАД, ЩО ВИНИКАЮТЬ ПІД ЧАС ЇХ ФОРМУВАННЯ	21
2.1 Технології отримання зображення в різних технічних системах	21
2.2 Огляд видів завад, які з'являють на зображеннях	27
2.3 Висновки до другого розділу	36
3 МАТЕМАТИЧНІ МОДЕЛІ ТЕКСТУРНИХ ЗОБРАЖЕНЬ	37
3.1 Класифікація моделей текстур.....	37
3.2 Математичні основи моделей	37
3.3 Висновки до третього розділу	41
4 ОГЛЯД МЕТОДІВ ПРИДУШЕННЯ ЗАВАД.....	42
4.1 Просторове фільтрування.....	42
4.1.1 Згладжувальні просторові фільтри.....	44
4.1.2 Порядково-статистичні фільтри	48
4.2 Частотна фільтрація	49
4.2.1 Ідеальний низькочастотний фільтр	50
4.2.2 Гаусовий низькочастотний фільтр	51
4.2.3 Низькочастотний фільтр Баттерворта.....	51
4.2.4 Ідеальні високочастотні фільтри отримані з низькочастотних фільтрів ...	52
4.2.5 Фільтр Вінера	52
4.2.6 Фільтр Калмана	54

4.2.7 Фільтр Лі	55
4.2.8 Фільтр Фроста	56
4.2.9 Сигма-фільтр	56
4.2.10 Висновки до четвертого розділу.....	57
5 МОДЕЛЮВАННЯ ТА АНАЛІЗ АЛГОРИТМУ ДЛЯ ПРИДУШЕННЯ СПЕКЛ-ШУМУ	59
5.1 Генерація синтезованих зображень.....	59
5.2 Реалізація алгоритму фільтрації спекл-шуму	62
5.3 Аналіз результатів фільтрації	66
5.4 Висновки до п'ятого розділу	73
ВИСНОВКИ.....	75
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	77
ДОДАТОК А. КОД ОСНОВНОЇ ПРОГРАМИ	

ПЕРЕЛІК СКОРОЧЕНЬ

КТ – комп’ютерна томографія

MPT – магнітно-резонансна томографія

NDVI – нормалізований відносний індекс рослинності

БПЛА – безпілотний літальний апарат

DSLR – цифрова однооб’єктивна дзеркальна фотокамера

ПЕТ – позитронно-емісійна томографія

РСА – радари із синтезованою апертурою

ДПФ – дискретне перетворення Фур’є

MRF – марковське випадкове поле

AR – авторегресивна модель

PSNR (Peak Signal-to-Noise Ratio) – пікове співвідношення сигнал/шум

SSIM (Structural Similarity Index) – індекс структурної подібності

ВСТУП

Цифрові технології все більше стають частиною нашого повсякденного життя. Їх використовують усюди: в медицині, промисловості, наукових дослідженнях, штучному інтелекті та навіть у сфері автоматизації. Однак важливу роль у багатьох із цих напрямків відіграє цифрова обробка зображень. Це одна з ключових технологій, яка дозволяє аналізувати, сегментувати чи розпізнавати об'єкти на зображеннях. Але виникає серйозна проблема — якість зображення. Шум, що з'являється під час створення, передачі чи зберігання даних, може значно ускладнити процеси обробки.

Особливий інтерес викликають текстурні зображення. Їх активно застосовують у найрізноманітніших галузях — від аналізу зображень у медичних дослідженнях до створення промислової графіки чи візуальних ефектів. Такі зображення мають свої особливості: вони складаються зі складних і часто неповторних візерунків. Саме тому їхнє оброблення є непростим завданням. Усунення шуму може пошкодити унікальну текстуру, яка є основною характеристикою таких зображень.

Мета цього дослідження — дослідити методи, які дозволять ефективно зменшувати шум на текстурних зображеннях, не пошкоджуючи їхньої структури. Для цього планується використовувати MATLAB — універсальне програмне середовище, яке надає широкі можливості для аналізу, моделювання та оцінки результатів.

Актуальність цієї теми важко переоцінити. Завдання збереження якості зображення є критично важливим у багатьох сферах. Наприклад, у медицині це може бути діагностика захворювань за допомогою рентгенівських або томографічних знімків. У промисловості якісні текстурні зображення необхідні для точного аналізу поверхонь чи матеріалів. Тому розвиток технологій обробки зображень і вдосконалення методів роботи з текстурами сприятиме прогресу в багатьох напрямках.

1 ОГЛЯД ПРАКТИЧНИХ ЗАДАЧ, В ЯКИХ ВИНИКАЄ НЕОБХІДНІСТЬ ВИКОНУВАТИ ОБРОБКУ ЦИФРОВИХ ЗОБРАЖЕНЬ

Обробка цифрових зображень є важливим етапом у багатьох практичних задачах, де зображення відіграють ключову роль. Застосування методів обробки зображень охоплює широке коло галузей, від медицини до промисловості та розваг. Розглянемо кілька основних сфер, де знаходить своє застосування цифрова обробка зображень.

1.1 Медицина

Ультразвукова діагностика: це популярний метод у сучасній медицині, який використовує високочастотні звукові хвилі для створення зображень внутрішніх органів і тканин. Завдяки можливості бачити процеси в реальному часі, цей метод дозволяє, наприклад, досліджувати кровотік під час доплерографії [1]. На рис. 1.1 показано приклад ультразвукового сканування плоду, яке може містити шуми чи артефакти, що потребують обробки для підвищення якості зображення та точності діагностики [2].



Рисунок 1.1 – Ультразвукове зображення плоду людини

Комп'ютерна томографія (КТ): це неінвазивний діагностичний метод, який забезпечує детальну візуалізацію внутрішніх структур організму шляхом обробки рентгенівських знімків, отриманих під різними кутами. Розроблена в 1970-х роках, ця технологія базується на різному поглинанні рентгенівських променів тканинами. Під час сканування пацієнт розміщується на рухомій платформі, а КТ-сканер обертається на 360 градусів, створюючи серію зображень, які комп'ютер перетворює на двовимірні «зрізи» [3]. На рис. 1.2 показано приклад КТ голови, що демонструє високу роздільну здатність, але такі зображення можуть містити шуми чи артефакти, що потребують обробки для придушення завад і покращення діагностичної цінності [4].

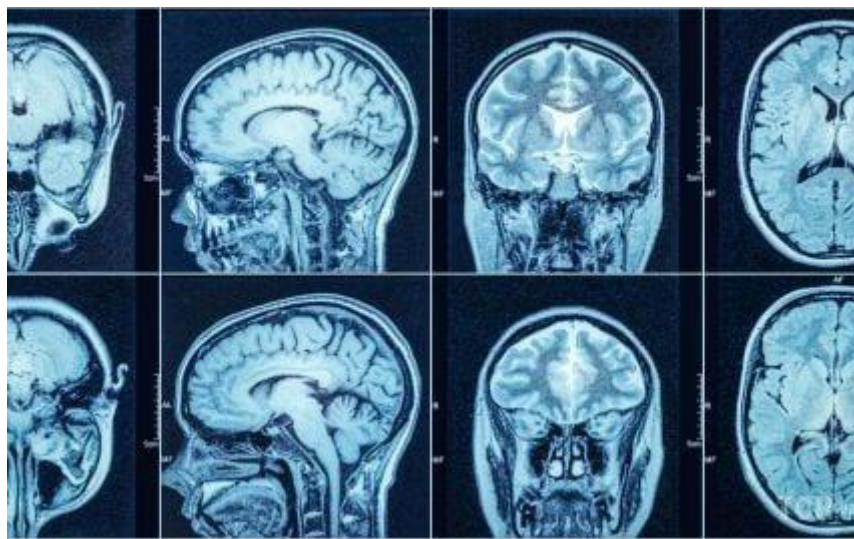


Рисунок 1.2 – Комп'ютерна томографія голови

Магнітно-резонансна томографія (МРТ): це сучасний неінвазивний метод діагностики, який активно використовується для оцінки стану здоров'я та контролю ефективності лікування. На відміну від рентгенографії чи комп'ютерної томографії, МРТ не застосовує іонізуюче випромінювання, а формує зображення за допомогою потужного магнітного поля та електромагнітних хвиль. Комп'ютерна обробка даних забезпечує високу деталізацію м'яких тканин, кісток і внутрішніх органів [5]. На рис. 1.3 представлено приклади МРТ-зображень головного мозку та колінного суглоба, які можуть містити шуми чи артефакти, що

потребують обробки для придушення завад, щоб покращити якість візуалізації та точність діагностичного аналізу [5].

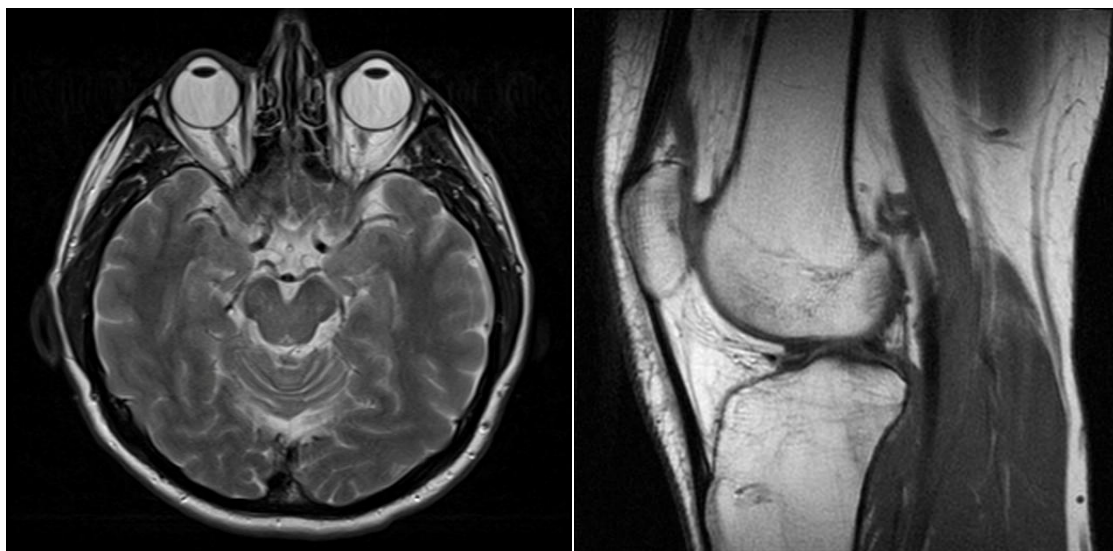


Рисунок 1.3 – Магнітно-резонансна картина головного мозку в нормі та магнітно-резонансне зображення коліна

1.2 Сільське господарство

Супутниковий моніторинг: дозволяє отримувати актуальні дані про стан сільськогосподарських посівів завдяки знімкам високої роздільної здатності та спектральним камерам. На основі цих даних обчислюють нормалізований вегетаційний індекс NDVI (Normalized difference vegetation index), який відображає густоту рослинності та допомагає визначити ділянки, що потребують додаткового удобрення чи пересіву [6]. На рис. 1.4 зображено супутниковий знімок, отриманий із Sentinel-2, який показує поля сільськогосподарського масиву. Такі зображення можуть містити завади, наприклад, атмосферні спотворення чи шуми, що потребують обробки для підвищення точності аналізу та якості отриманих результатів [6].



Рисунок 1.4 – Супутниковий знімок Sentinel-2. Масив полів господарства

Моніторинг за допомогою безпілотних літальних апаратів (БПЛА): значно полегшує збір даних про стан сільськогосподарських посівів. На відміну від супутникових систем, дрони є більш гнучким інструментом, що забезпечує високу деталізацію зображень. Завдяки польотам на висоті від 100 до 300 метрів над землею, БПЛА можуть створювати знімки з роздільною здатністю в сантиметрах на піксель [6]. На рис. 1.5 зліва зображено фотографію поля, отриману з дрона, а праворуч — карту NDVI, що відображає стан сходів ярого ячменю. Зелені ділянки вказують на здорову рослинність, тоді як червоні позначають зони з пригніченою вегетацією. Такі зображення можуть містити завади, наприклад, шуми, спотворення через погодні умови чи технічні артефакти, які потребують обробки для підвищення точності аналізу та якості результатів [6].

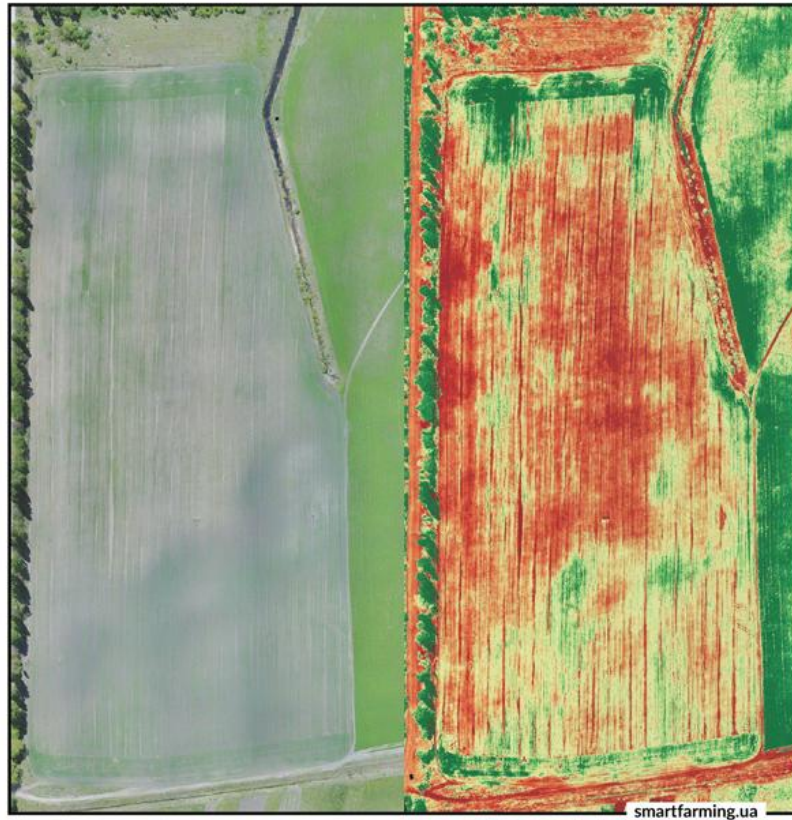


Рисунок 1.5 – Знімок поля, отриманий з дрона (зліва), та побудований NDVI індекс стану сходів ярого ячменю (праворуч).

1.3 Система розпізнавання обличчя

Технологія розпізнавання облич дозволяє ідентифікувати особу на цифрових зображеннях шляхом порівняння рис обличчя з даними, що зберігаються в базі. Ефективність цієї технології значною мірою залежить від якості вихідних зображень, які можуть містити завади, такі як шуми чи артефакти. Обробка зображень для придушення цих завад відіграє ключову роль у забезпеченні високої точності та надійності систем розпізнавання [7]. На рис. 1.6 представлено приклад роботи системи розпізнавання обличчя, який підкреслює важливість чіткості та чистоти зображень для коректної ідентифікації особи [8].



Рисунок 1.6 – Система розпізнавання обличчя

1.4 Відеоаналітика для розпізнавання номерних знаків

Технологія автоматичного розпізнавання номерних знаків, заснована на штучних нейронних мережах, дає змогу виявляти та ідентифікувати номери транспортних засобів на відеозаписах у реальному часі. Ця система є ключовим елементом відеонагляду, полегшуючи моніторинг транспорту, пошук автомобілів у розшуку та виявлення порушень правил дорожнього руху [9]. Проте точність розпізнавання залежить від якості зображень, які можуть містити шуми, артефакти чи спотворення. Обробка зображень для придушення таких завад є необхідною для забезпечення високої точності, адже навіть незначні дефекти можуть спричинити помилки в ідентифікації, особливо під час аналізу рухомих об'єктів. На рис. 1.7 зображено приклад роботи системи розпізнавання номерних знаків, що підкреслює важливість чіткості зображень для успішної ідентифікації [9].



Рисунок 1.7 – Система розпізнавання автомобільних номерів

1.5 Відновлення зображень

Відновлення зображень — це процес оцінювання ступеня пошкодження цифрового зображення, його аналізу та відтворення втрачених деталей. Зображення можуть зазнавати різних дефектів, таких як розмиття через рух, шуми або неточне фокусування камери. Для відновлення застосовують методи, які компенсують ефекти, що спричинили погіршення якості. Зокрема, використовують функцію поширення точки, отриману з аналізу зображення точкового джерела, щоб відновити втрачену інформацію [10]. Основна мета таких методів — зменшити рівень шуму та підвищити роздільну здатність зображення. Обробка може виконуватися в просторовій або частотній області. Одним із традиційних підходів є деконволюція, яка застосовується в частотній області після обчислення перетворення Фур'є та усуває втрату чіткості, спричинену розмиттям. Сьогодні для реставрації зображень використовують цифрові інструменти та спеціалізоване програмне забезпечення, які дозволяють виправляти різноманітні дефекти та покращувати чіткість і якість деталей [10]. Приклад результатів відновлення зображення показано на рис. 1.8 [11].

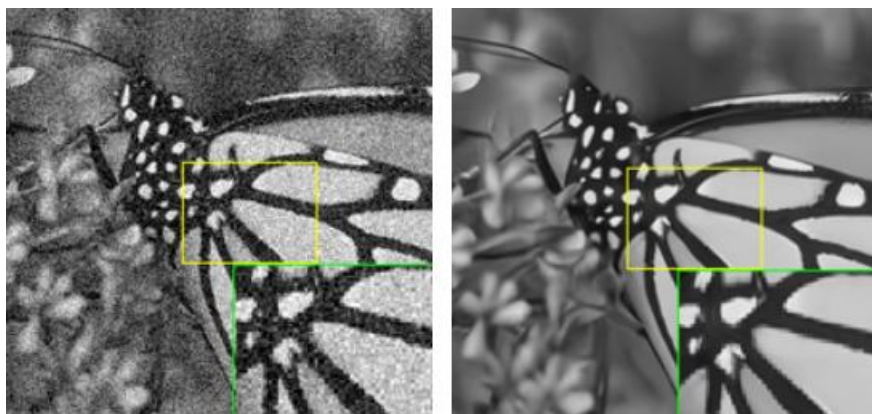


Рисунок 1.8 – Крило метелика: зашумлена текстура (зліва) та частково відновлена версія (справа)

1.6 Зображення отримані за допомогою радарів із синтезованою апертурою (РСА)

Радіолокація із синтезованою апертурою (РСА) застосовується для отримання зображень земної поверхні з високою роздільною здатністю за допомогою супутників або літаків. Технологія ґрунтується на русі платформи з антеною, яка випромінює радіохвилі, що відбиваються від поверхні та обробляються для формування синтетичної апертури. Проте якість таких зображень погіршується через спекловий шум, викликаний інтерференцією відбитих сигналів, що ускладнює текстурний аналіз і виявлення об'єктів. Для усунення цієї проблеми розроблено метод, який включає обчислення індексів сингулярності на основі градієнта сигналу для кожного пікселя, виділення ключових характеристик і реконструкцію зображення з пониженим рівнем шуму. Цей підхід ефективно придушує завади, зберігаючи текстурні деталі, і демонструє переваги порівняно з традиційними методами, такими як нелінійна фільтрація. Приклад результатів фільтрації зображено на рис. 1.9 [12].

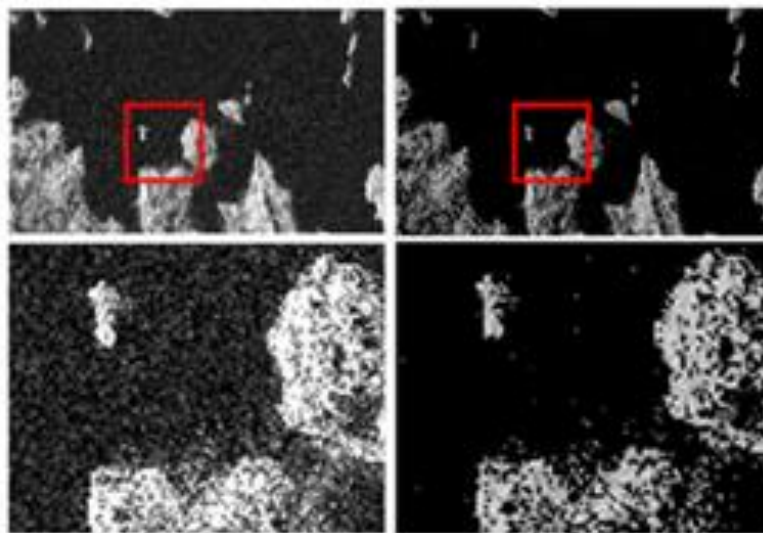


Рисунок 1.9 – PCA-зображення: вихідне з шумом та оброблене

1.7 Системи допомоги водію

Системи допомоги водієві, такі як адаптивний круїз-контроль, автоматичне екстрене гальмування, утримання смуги руху та моніторинг сліпих зон, є ключовими елементами технологій автономного водіння. Ці функції потребують здатності автомобіля «сприймати» навколишнє середовище, що забезпечується різноманітними датчиками, які варіюються залежно від автовиробника. Відеокамери відіграють центральну роль у реалізації таких функцій, як розпізнавання смуг руху, дорожніх знаків, пішоходів і велосипедистів, а також у підтримці адаптивного круїз-контролю та екстреного гальмування. Одна камера формує двовимірне зображення, яке обробляється комп'ютером для виявлення об'єктів. Крім того, камери, встановлені в салоні, разом із інфрачервоними випромінювачами стежать за увагою водія, а зовнішні інфрачервоні камери забезпечують нічне бачення для виявлення людей чи тварин у темряві[13]. Оскільки чіткість зображень є критично важливою для роботи цих систем, обробка зображень для придушення завад, таких як шуми чи спотворення, спричинені поганим освітленням або динамічними умовами, є необхідною для стабільної роботи алгоритмів розпізнавання. Приклад такої системи зображено на рис. 1.10 [13].



Рисунок 1.10 – Нічне бачення з інфрачервоними камерами для розпізнавання об'єктів на дорозі

1.8 Комп'ютерне моделювання

У задачах комп'ютерного моделювання синтез штучних текстурних зображень застосовується для дослідження алгоритмів обробки зображень, зокрема для фільтрації та придушення завад. Наприклад, у комп'ютерній графіці створюються синтетичні текстури, що імітують реальні поверхні, такі як дерево чи камінь, із контрольованим рівнем шуму для тестування ефективності алгоритмів обробки зображень [16]. Такі зображення генеруються шляхом зміни параметрів текстурних моделей, що дозволяє оцінювати алгоритми в різних умовах, наприклад, за наявності гаусівського чи імпульсного шуму. Цей підхід забезпечує контрольоване середовище для вдосконалення методів обробки зображень, які застосовуються в задачах візуалізації та аналізу даних. На рис. 1.11 показано приклад синтетичної текстури, що імітує поверхню дерева, із доданим шумом і результатом його придушення.

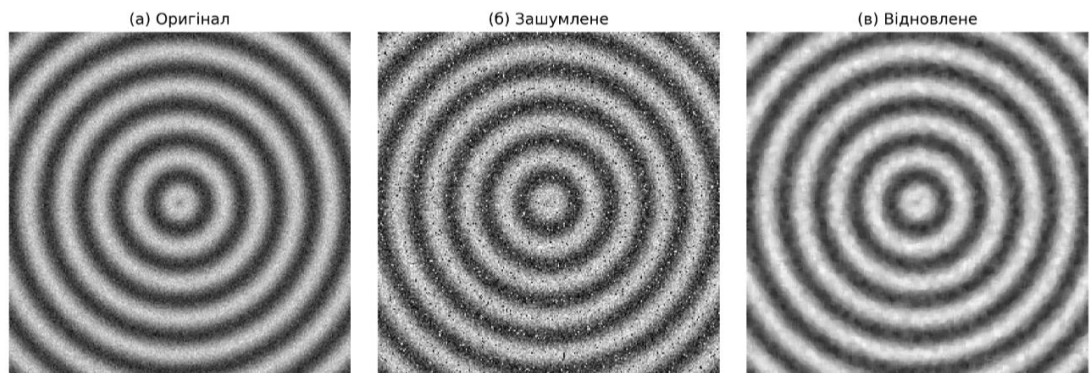


Рисунок 1.11 – Приклад обробки синтетичної текстури, що імітує поверхню дерева

1.9 Висновки до першого розділу

Різноманітність розглянутих практичних задач підкреслює універсальність проблеми придушення завад у цифрових зображеннях та необхідність розробки ефективних методів її вирішення, що враховують специфіку різних типів зображень та характер присутніх на них завад. Особливий інтерес становлять завдання придушення завад на текстурних зображеннях, які зустрічаються в багатьох розглянутих прикладах і вимагають специфічних підходів через складну структуру текстур.

2 ТЕХНОЛОГІЇ ОТРИМАННЯ ЗОБРАЖЕНЬ У ТЕХНІЧНИХ СИСТЕМАХ ТА ВИДИ ЗАВАД, ЩО ВИНИКАЮТЬ ПІД ЧАС ЇХ ФОРМУВАННЯ

2.1 Технології отримання зображення в різних технічних системах

Оптичні системи

Основою фотографії є взаємодія світла з об'єктами та його подальша реєстрація. На рис. 2.1 зображено схему роботи фотокамери. Світло, що походить від природного (наприклад, сонця) чи штучного джерела (1), відбивається від об'єктів перед об'єктивом, проходить через об'єктив фотокамери (2) і, за наявності, затвор (7), після чого потрапляє на матрицю (сенсор) (8), розташовану на задній стінці корпусу. У дзеркальних фотокамерах (DSLR) до моменту натискання спускової кнопки світло спрямовується дзеркалом (3) через призму (4) до видошукача (5). Під час зйомки дзеркало піднімається, дозволяючи світлу досягти матриці, як у компактних камерах. Цей процес подібний до проходження світла через кришталик ока до сітківки. На матриці світло перетворюється чутливим елементом (датчиком зображення) в електричні сигнали, формуючи цифрове зображення. Проте якість зображень може погіршуватися через завади, такі як шуми матриці, оптичні спотворення чи артефакти, спричинені низьким освітленням, що вимагає обробки для підвищення чіткості та точності [15].

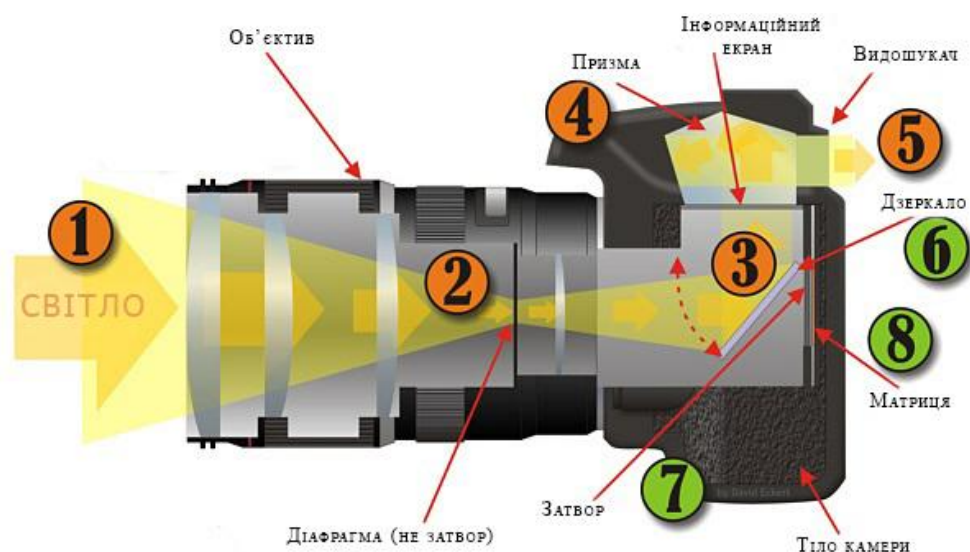


Рисунок 2.1 – Будова фотокамери

Медичні системи візуалізації

Ультразвукова діагностика (УЗД): використовує високочастотні звукові хвилі (зазвичай у діапазоні 1–20 МГц) для створення зображень м'яких тканин і внутрішніх органів. Датчик (трансд'юсер) випромінює звукові хвилі, які відбиваються від меж тканин із різною акустичною щільністю, а відбиті сигнали фіксуються та обробляються для формування двовимірних або тривимірних зображень у реальному часі. УЗД широко застосовується в медицині для діагностики захворювань серця, печінки, нирок, щитоподібної залози, а також для моніторингу вагітності завдяки своїй неінвазивності та відсутності іонізуючого випромінювання. Однак якість УЗД-зображень погіршується через завади, зокрема спекловий шум, який виникає внаслідок інтерференції відбитих хвиль від дрібних структур тканин. Цей шум створює зернисту текстуру, що ускладнює аналіз деталей і діагностику. Крім того, артефакти, спричинені рухом пацієнта, неоднорідністю тканин або обмеженнями роздільної здатності датчика, також знижують чіткість зображень. Для придушення цих завад застосовуються сучасні методи обробки, такі як адаптивна фільтрація, алгоритми відстеження спеклу та глибоке навчання, які підвищують контрастність і зберігають текстурні деталі [16]. Приклад впливу спекл-шуму на УЗД-зображення показано на рис. 2.2 [17].



Рисунок 2.2 – Тривимірна сонограма

Позитронно-емісійна томографія (ПЕТ): має схожий принцип роботи з рентгенівською комп'ютерною томографією, але замість зовнішнього джерела рентгенівських променів використовує радіоактивний ізотоп, уведений у тіло пацієнта. Цей ізотоп під час розпаду випромінює позитрони, які при зіткненні з електронами анігілюють, генеруючи два гамма-промені, що рухаються в протилежних напрямках. Ці промені фіксуються детекторами, а комп'ютерна обробка даних формує томографічне зображення на основі принципів реконструкції. На рис. 2.3 зображено приклад ПЕТ-зображення, яке є частиною тривимірної моделі тіла пацієнта, де пухлини в мозку та легенях видно як невеликі білі ділянки. Проте якість ПЕТ-зображень може погіршуватися через завади, такі як статистичний шум, спричинений обмеженою кількістю виявлених гамма-променів, або артефакти від руху пацієнта чи неоднорідного розподілу ізотопу. Ці завади ускладнюють текстурний аналіз і діагностику, що вимагає обробки зображень для придушення шумів і підвищення контрастності [14].

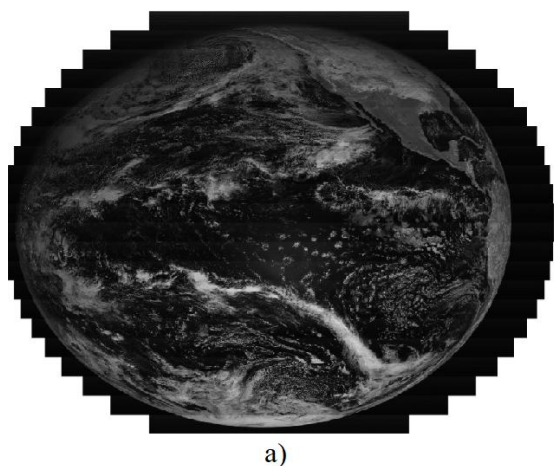


Рисунок 2.3 – ПЕТ-зображення.

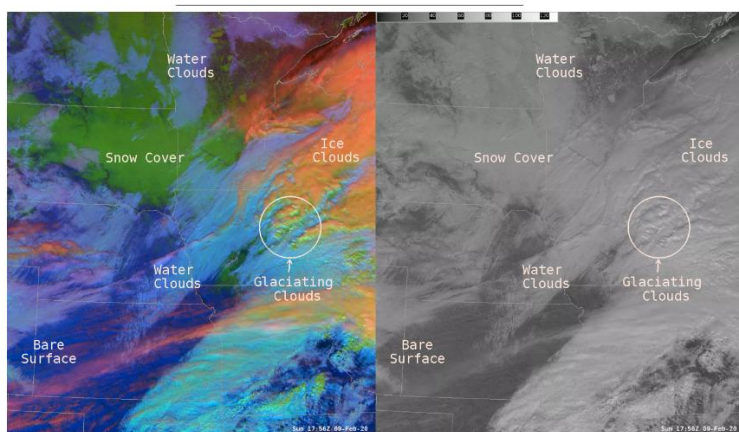
Системи дистанційного зондування

Отримання супутникових зображень здійснюється за допомогою приладу Advanced Baseline Imager (ABI), встановленого на геостаціонарних супутниках. Цей пристрій вимірює енергію електромагнітного випромінювання у видимому та інфрачервоному діапазонах, використовуючи 16 спектральних каналів. ABI сканує земну поверхню послідовно за допомогою двох рухомих дзеркал: одне рухається в напрямку північ-південь, інше — схід-захід, що забезпечує формування деталізованого зображення планети. Фотони, захоплені детекторами, перетворюються в електричні сигнали, пропорційні інтенсивності випромінювання. Дані передаються на наземні станції у вигляді радіосигналів, де декодуються в цифровий формат і обробляються програмним забезпеченням, яке трансформує сигнали в пікселі, кожен із яких відповідає окремій ділянці поверхні. Кольорове RGB-зображення формується шляхом комбінації монохромних даних із урахуванням спектральних характеристик [18].

Якість супутникових зображень може погіршуватися через завади, такі як атмосферні спотворення, хмарність, шуми детектора чи артефакти від сонячного блиску, які впливають на текстурну чіткість і точність аналізу. Для придушення цих завад застосовуються методи обробки, зокрема корекція атмосферного впливу та адаптивна фільтрація, що підвищують контрастність і якість зображень. На рис. 2.4 зображено приклади необроблених смуг даних і остаточне кольорове зображення, отримане з супутника [18].



а)



б)

Рисунок 2.4 – Необроблені некалібровані смуги зображення повного диска з «червоної» частини електромагнітного спектру а); RGB-зображення з розрізненням фази денної хмари (ліворуч) і зображення видимого каналу .0,64 мкм (праворуч) за той самий час б)

Термографічні системи

Тепловізійні прилади та монокуляри використовують технологію інфрачервоного зображення, що дозволяє виявляти теплову енергію об'єктів і створювати їхні зображення навіть у повній темряві. Теплове випромінювання від об'єктів потрапляє на чутливу матрицю, змінюючи електричний опір у кожному пікселі, після чого сигнали обробляються процесором, який формує термальне зображення. У ньому сині відтінки позначають низькі температури, зелені — середні, а червоні та білі — високі; темніші кольори, як пурпуровий чи темно-синій, вказують на холодніші об'єкти з меншим випромінюванням [19]. Якість зображень може погіршуватися через завади, такі як шуми матриці, артефакти теплового фону, атмосферні спотворення чи низька роздільна здатність датчика, що впливають на текстурну чіткість і точність інтерпретації. Для придушення цих завад застосовуються методи обробки, зокрема адаптивна фільтрація та алгоритми підвищення контрасту, які покращують розрізнення об'єктів і деталей. На рис. 2.6 зображено приклади термальних зображень, отриманих тепловізором, які ілюструють розподіл температур об'єктів у сцені [19].



Рисунок 2.6 – Приклад зображень зроблених тепловізором.

Системи технічного зору

Машинний зір — це застосування комп'ютерного зору в промисловості та виробництві. Датчик синхронізації визначає, коли деталь, яка швидко рухається по конвеєру, знаходиться в положенні, що підлягає інспекції. Датчик вмикає камеру, щоб зробити знімок деталі, коли вона проходить під камерою і часто синхронізується з імпульсом освітлення, щоб зробити чітке зображення [20]. На рис. 2.7 показано принцип роботи системи машинного зору з використанням камери, датчика та обробки зображення [21]. Зображення з камери потрапляє в захоплювач кадрів або до пам'яті комп'ютера в системах, де захоплювач кадрів не використовується. Програмне забезпечення, як правило, здійснює кілька кроків для обробки зображень. Часто зображення для початку обробляється з метою зменшення шумів або конвертації багатьох відтінків сірого в просте поєднання чорного та білого (бінаризації). Після початкової обробки програма буде

обчислювати, проводити вимірювання і/або визначати об'єкти, розміри, дефекти та інші характеристики зображення. Як останній крок, програма пропускає або забраковує деталь відповідно до заданого критерію [20].

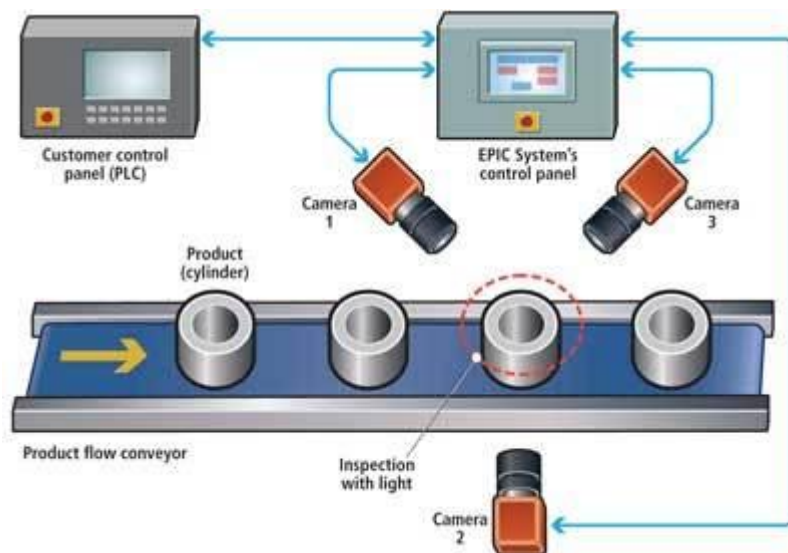


Рисунок 2.7 – Схема роботи системи машинного зору для автоматизованої інспекції на виробничій лінії

2.2 Огляд видів завад, які з'являють на зображеннях

У процесі отримання, зберігання та обробки цифрових зображень неминуче виникають різноманітні завади, які суттєво впливають на їх якість. Ці спотворення можуть бути викликані як фізичними чинниками, так і технічними недоліками обладнання. Аналіз видів завад є важливим етапом у розробці ефективних методів їх придушення, оскільки кожен тип шуму має свої характерні особливості та природу виникнення. У цьому розділі розглянуто основні види завад, які спотворюють зображення.

Артефакти стиснення — це спотворення, що виникають у цифрових зображеннях, відео чи аудіо внаслідок застосування алгоритмів стиснення з втратами. Під час такого стиснення частина даних видаляється, щоб зменшити обсяг файлу для зберігання або передачі через цифрові канали. Чим вищий

ступінь стиснення (тобто нижчий бітрейт або якість), тим більш помітними стають дефекти, такі як блокові артефакти, розмиття чи втрата текстурних деталей, що погіршують візуальну якість зображення. Ці завади ускладнюють аналіз текстурних структур, особливо в задачах, де важлива чіткість деталей, наприклад, у медичній діагностиці чи супутниковому моніторингу. Для їхнього придушення застосовуються методи обробки, такі як декомпресійна фільтрація чи алгоритми відновлення текстур, що підвищують якість зображень [22].

Графічні артефакти стиснення:

Блочні спотворення: виникають у зображеннях, стиснених за допомогою дискретного косинусного перетворення, через незалежну обробку окремих блоків без урахування кореляції між сусідніми блоками. Це спричиняє появу видимих меж між блоками, відомих як блочні артефакти, які погіршують текстурну цілісність зображення. Такі спотворення є особливо помітними порівняно з іншими видами артефактів і можуть ускладнювати аналіз деталей у задачах, як-от медична діагностика чи розпізнавання об'єктів. Для їхнього придушення застосовуються методи деблокінгу та адаптивної фільтрації. Приклад блочних артефактів зображено на рис. 2.8 [22].

Кільцеві артефакти (ringing): з'являються в зображеннях через втрату даних у ділянках із різкими перепадами яскравості, наприклад, на контурах об'єктів. Вони найчастіше виникають під час використання вейвлет-стиснення, створюючи коливальні спотворення навколо контрастних меж. Ці артефакти погіршують чіткість текстурних деталей, що може впливати на точність аналізу зображень. Для їхнього зменшення застосовуються методи постобробки, такі як вейвлет-фільтрація чи алгоритми відновлення контурів. Приклад кільцевих артефактів також показано на рис. 2.8 [22].

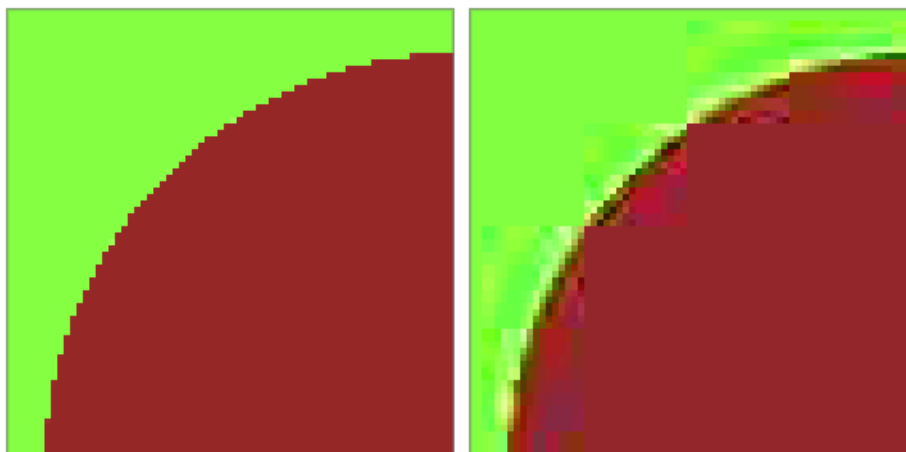


Рисунок 2.8 – Демонстрація блочних і кільцевих артефактів стиснення.

Розмивчасті спотворення, або замилювання, виникають через втрату високочастотних складових зображення, що призводить до згладжування дрібних деталей і текстур. Такі артефакти з'являються в областях із різкими змінами яскравості, наприклад, на контурах об'єктів чи текстурних ділянках, і стають менш помітними при перегляді зображення здалеку. Ці завади погіршують чіткість текстурних структур, ускладнюючи аналіз у задачах, таких як медична візуалізація чи обробка супутникових зображень. Для їхнього придушення застосовуються методи підвищення різкості та відновлення високих частот, які покращують деталізацію зображень [22].

Бандинг проявляється у вигляді небажаних одноколірних смуг у зображеннях, де потрібна плавна градація кольору, через недостатню глибину кольору для точного відтворення тонів. У комп'ютерній графіці цей ефект відомий як постеризація, але в контексті обробки зображень бандинг є небажаним спотворенням, що знижує якість текстурного аналізу. Такі завади особливо помітні в областях із поступовими колірними переходами, наприклад, на зображеннях неба чи шкіри. Для зменшення бандингу використовуються методи збільшення глибини кольору та алгоритми згладжування градієнтів. Приклад бандингу зображено на рис. 2.9 [22].

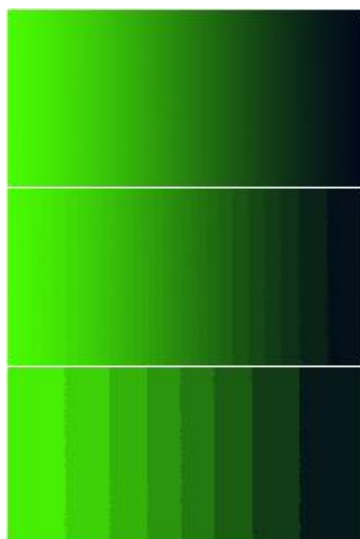


Рисунок 2.9 – Ефект бандингу

Розмиття руху (motion blur) виникає під час зйомки швидко рухомих об'єктів або при різкому зміщенні камери в момент експозиції. Цей артефакт проявляється у вигляді змазування об'єктів, що втрачають різкість і виділяються на тлі сцени, погіршуючи чіткість текстурних деталей. Ефект може бути небажаним у задачах, де важлива деталізація, наприклад, у системах комп'ютерного зору чи медичній діагностиці [23].

Для придушення розмиття руху застосовуються методи обробки, такі як деконволюція та алгоритми відновлення зображень, які підвищують різкість і відновлюють текстурні структури. У деяких випадках фотографи навмисно використовують цей ефект для створення динамічності кадру. Приклад розмиття руху зображено на рис. 2.10 [23].



Рисунок 2.10 – Розмиття зображення під час руху.

Шуми

Гаусівський шум є одним із найпоширеніших видів завад у цифрових зображеннях і моделюється як адитивний процес, коли до значення кожного пікселя додається випадкова величина з нормального розподілу. Функція щільності ймовірності цього розподілу має вигляд:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}}, \quad (2.1)$$

де z – відображає сірий рівень, μ - середнє значення випадкової величини і σ – стандартне відхилення.

Причинами гаусівського шуму є термічні коливання електронів, недостатнє освітлення, слабкий сигнал приймача чи інші зовнішні впливи. Відповідно до центральної граничної теореми, сума великої кількості незалежних випадкових величин із незначним індивідуальним внеском формує гаусівський розподіл, що пояснює його поширеність [24].

Цей шум погіршує текстурну чіткість зображень, ускладнюючи аналіз деталей у задачах, таких як медична діагностика чи розпізнавання об'єктів.

Наприклад, термічний шум, спричинений вібрацією численних електронів, є незалежним і має однаковий вплив на кожен піксель, що відповідає моделі гаусівського розподілу. Для придушення гаусівського шуму застосовуються методи обробки, зокрема згорткові фільтри, медіанна фільтрація чи алгоритми на основі машинного навчання, які відновлюють текстурні деталі. Приклад гаусівського шуму зображено на рис. 2.11 [24].

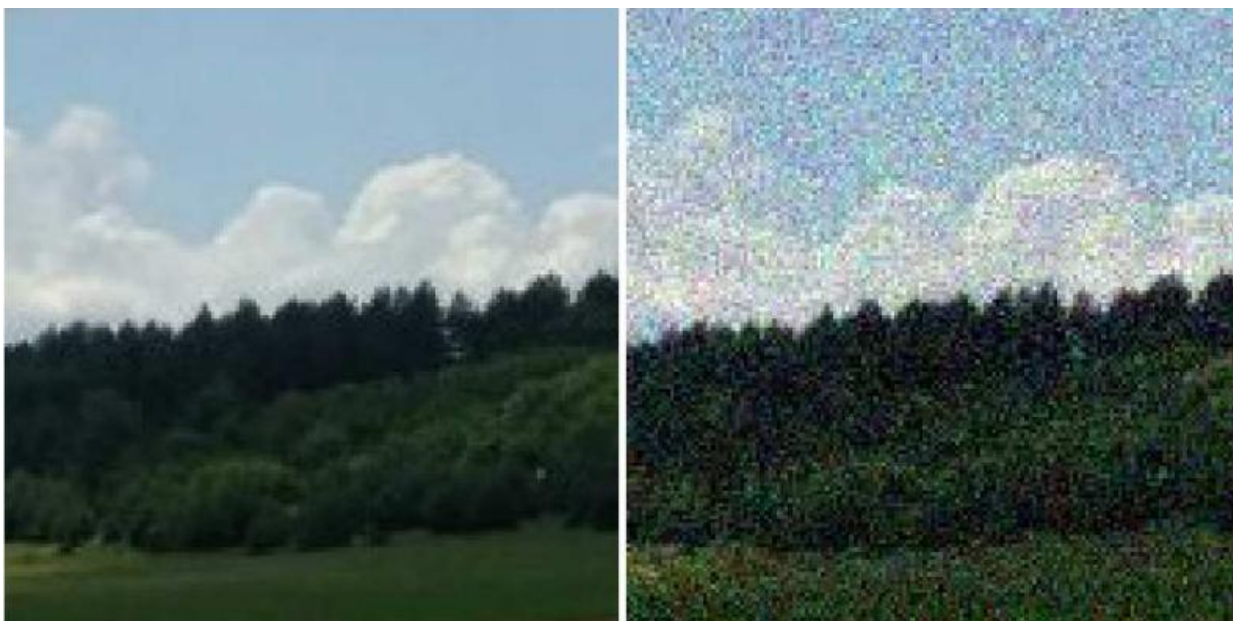


Рисунок 2.11 – Приклад накладання гаусівського шуму на зображення

Дробовий шум, відомий також як шум Пуассона, є поширеним видом завад в електроніці та оптиці, що описується розподілом Пуассона. У цифрових системах він виникає через дискретну природу електричного заряду, а в оптичних приладах — через корпускулярну природу світла під час підрахунку фотонів. Цей шум погіршує якість зображень, зокрема текстурних деталей, у задачах, таких як астрономічна зйомка чи медична візуалізація [24].

У процесі випромінювання світла лазером фотони генеруються випадково. При високій яскравості лазера кількість фотонів достатньо велика, щоб флуктуації їхньої кількості були незначними. Однак за низької яскравості відносні коливання числа фотонів стають помітними, що призводить до флуктуацій яскравості та появи дробового шуму. Ці спотворення ускладнюють аналіз текстурних структур зображень. Для їхнього придушення застосовуються методи обробки, такі як

статистична фільтрація чи алгоритми відновлення сигналу, які підвищують чіткість деталей. Приклад дробового шуму зображено на рис. 2.12 [24].

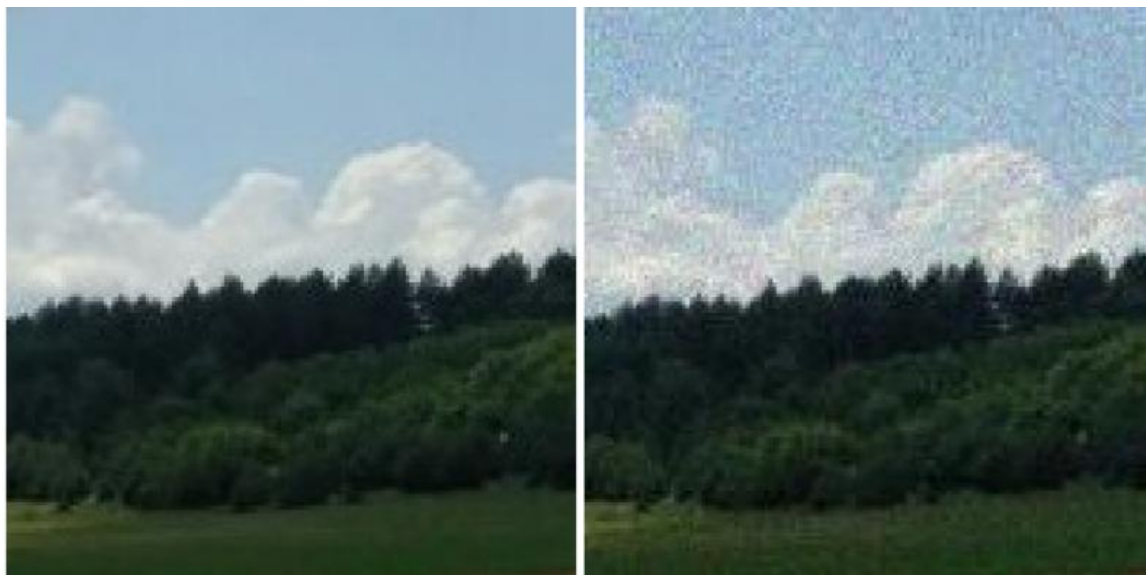


Рисунок 2.12 – Приклад накладання на зображення шуму Пуассона

Імпульсний шум, або шум типу "сіль і перець", проявляється у вигляді темних пікселів у світлих областях зображення та світлих пікселів у темних. Він виникає через помилки під час аналого-цифрового перетворення або побітові збої при передачі даних. Функція щільності розподілу цього шуму має вигляд: $p(z) = P_a$ (для $z = a$) та $p(z) = P_b$ (для $z = b$). При цьому a відповідає темній точці на зображенні, b – світлій. Якщо P_a бо $P_b = 0$, то такий шум є уніполярним; якщо обидва значення ненульові, на зображенні з'являються випадкові чорні та білі точки [24].

Цей шум погіршує текстурну чіткість зображень, ускладнюючи аналіз у задачах, таких як медична візуалізація чи розпізнавання об'єктів. Для його придушення використовуються методи віднімання темного фрейму та інтерполяції, які дозволяють відновити пошкоджені пікселі та покращити якість текстурних деталей. Приклад імпульсного шуму зображено на рис. 2.13 [24].



Рисунок 2.13 – Приклад накладання на зображення шуму типу "сіль і перець"

Шум зерен фотоплівки виникає через особливості формування зображень на фотоплівці, де зображення складається з численних зерен, що накладаються на поверхню. Під час експозиції світло впливає на зерна: одні поглинають фотони і набувають металево-сріблястого відтінку, інші, що не отримали достатньої дози світла, залишаються незмінними і змиваються під час обробки. Співвідношення опромінених і неопромінених зерен визначає прозорість окремих ділянок зображення: більша кількість опромінених зерен робить ділянку менш прозорою, і навпаки. Цей шум створює зернисту структуру, яка погіршує текстурну чіткість зображень, особливо в задачах, що потребують високої деталізації, наприклад, у фотографії чи архівній обробці [24].

Цифровий аналог цього шуму, відомий як **яскравий шум**, проявляється у вигляді дрібнозернистих сторонніх точок різної яскравості на цифрових зображеннях, що нагадують зерна фотоплівки. Він виникає через флуктуації сигналу в сенсорах камер і ускладнює аналіз текстурних деталей. Для придушення такого шуму застосовуються методи обробки, такі як медіанна фільтрація чи алгоритми зменшення зернистості, які підвищують якість зображень. [24].

Спекл-шум є складною завадою через залежність від просторових і сигнальних характеристик. Він виникає при використанні когерентних джерел, таких як лазери, ультразвук чи радіохвилі, коли відбиті хвилі інтерферують через неоднорідність поверхні в межах пікселя. Це створює зернистий візерунок, який погіршує чіткість текстурних зображень, ускладнюючи аналіз у задачах комп'ютерного зору чи синтезу текстур. На відміну від адитивних шумів, як гаусівський, спекл-шум має мультиплікативну природу, що ускладнює його придушення, адже шум залежить від інтенсивності сигналу [24]. Наприклад, на ультразвукових зображеннях (УЗД) спекл-шум з'являється через когерентність ультразвукових хвиль, які взаємодіють із розсіювачами в тканинах, як клітини печінки, формуючи зернисту текстуру, що знижує контрастність і ускладнює діагностику [25]. У радіолокаційних зображеннях із синтезованою апертурою (РСА) спекл-шум виникає від когерентних радіохвиль, відбитих від нерівних поверхонь, як ґрунт чи рослинність, погіршуючи аналіз текстур у дистанційному зондуванні, наприклад, при моніторингу лісових масивів. Приклад такого шуму можна побачити на рис. 2.14 [26].

Спекл-шум моделюється як мультиплікативна завада:

$$I(x, y) = S(x, y) * N(x, y), \quad (2.2)$$

де $I(x, y)$ — інтенсивність спостережуваного зображення в точці (x, y) , $S(x, y)$ — інтенсивність ідеального зображення без шуму, $N(x, y)$ — шумова компонента, що має експоненціальний розподіл [26]:

$$p_I(I) = \left(\frac{1}{\sigma^2}\right) * \exp\left(-\frac{I}{\sigma^2}\right), \quad (2.3)$$

де σ — стандартне відхилення, яке характеризує розкид інтенсивності. Амплітуда сигналу підкоряється розподілу Релея, що відображає когерентну природу шуму. Статистично спекл-шум залежить від кількості розсіювачів, що імітує зернистість, наприклад, на УЗД. Це ускладнює придушення шуму стандартними фільтрами, які можуть розмити текстурні деталі [26].



Рисунок 2.14 – Приклад зображення з спекл-шумом

2.3 Висновки до другого розділу

У даному розділі було розглянуто технології отримання зображень у технічних системах, таких як оптичні камери, медичні системи (УЗД, ПЕТ), супутникове зондування, термографія та системи машинного зору, що демонструють їх різноманітність і застосування в медицині, промисловості та моніторингу. Визначено основні види завад, зокрема гаусівський шум, імпульсний шум та спекл-шум, а також артефакти стиснення, розмиття руху та бандинг, які виникають під час формування зображень. Ці спотворення погіршують текстурну чіткість, ускладнюючи аналіз деталей у задачах медичної діагностики, дистанційного зондування та промислової інспекції. Виявлено, що завади мають різну природу та походження, що підкреслює необхідність їх детального вивчення для подальшого аналізу. Таким чином, розуміння технологій і завад є важливим етапом для вирішення проблеми підвищення якості текстурних зображень у сучасних системах.

3 МАТЕМАТИЧНІ МОДЕЛІ ТЕКСТУРНИХ ЗОБРАЖЕНЬ

Текстурні зображення є ключовим елементом у цифровій обробці зображень. Текстура — це візуальна характеристика поверхні, що проявляється у вигляді повторюваних або псевдовипадкових візерунків, таких як зернистість деревини, хвилястість тканини чи шорсткість піску. Текстури допомагають розрізняти об'єкти на зображеннях і є важливими для задач сегментації, класифікації та придушення завад. Математичні моделі текстур дозволяють формалізувати їхні характеристики, що спрощує аналіз і обробку зображень, особливо в умовах шуму [28].

3.1 Класифікація моделей текстур

Математичні моделі текстур поділяються на три основні категорії: статистичні, стохастичні та структурні.

Статистичні моделі аналізують розподіл яскравості пікселів та їхні взаємозв'язки, використовуючи, наприклад, гістограми, кореляційні функції або матриці співвідношення сірого.

Стохастичні моделі розглядають текстуру як випадковий процес. Прикладом є випадкові поля Маркова (Markov Random Fields, MRF), які моделюють залежності між сусідніми пікселями.

Структурні моделі описують текстуру як набір повторюваних елементів (текстонів), що формують регулярний візерунок, наприклад, цегляну кладку чи сітку [29].

28

3.2 Математичні основи моделей

Марковські випадкові поля

Марковські випадкові поля (MRF- Markov random field) моделюють текстуру через просторові залежності пікселів, використовуючи параметри θ для опису локальних взаємодій. Ймовірність конфігурації задається як [30]:

$$P(X = x) = \frac{1}{Z} \exp \left(\sum_k \omega_k f_k(x_{\{k\}}) \right), \quad (3.1)$$

де Z — нормувальна стала, ω_k — ваги, $f_k(x_{\{k\}})$ — функції ознак. Ця модель враховує контекстуальні властивості текстур, що робить її придатною для синтезу. Наприклад, MRF застосовуються для синтезу текстур і сегментації, зокрема в медичних зображеннях, де аналізують тканини. Також вони інтегруються з кореляційними матрицями для аналізу просторових розподілів. У PCA-зображеннях MRF допомагають класифікувати поверхні, досягаючи високої точності. Однак обчислювальна складність зростає через оцінку параметрів, що обмежує їхнє використання в реальному часі. Ці моделі ефективні для природних і синтетичних текстур, але потребують значних ресурсів. Приклади згенерованих текстур показано на рис. 3.1 [29].



Рисунок 4.1 – Текстури згенеровані марковськими моделями випадкових полів

Авторегресивна модель

Авторегресивна (AR) модель описує значення пікселя через лінійну комбінацію сусідніх пікселів [29]:

$$I(x, y) = \sum_{i,j} a_{i,j} I(x - i, y - j) + \varepsilon(x, y), \quad (3.2)$$

де $a_{i,j}$ — коефіцієнти, $\varepsilon(x, y)$ — шум. Як частина Gaussian MRF, AR-моделі моделюють статистичні залежності між пікселями, що робить їх корисними для класифікації текстур. У віддаленому зондуванні вони використовуються для розпізнавання типів поверхонь у PCA-зображеннях, наприклад, для класифікації водойм, полів чи лісів, аналізуючи просторові зв'язки. У медичних додатках AR-моделі допомагають сегментувати тканини, наприклад, виділяти контури органів

у рентгенівських зображеннях, що важливо для діагностики. AR-моделі оцінюють параметри текстур для класифікації поверхонь із заданими категоріями, такими як грубі чи гладкі текстур. Вони також можуть застосовуватися для синтезу текстур, хоча поступаються MRF за якістю генерації візерунків через меншу здатність моделювати складні візерунки. Порівняно з кореляційними матрицями, AR-моделі менш точні в класифікації, оскільки їхня продуктивність залежить від правильного вибору сусідства. Основним недоліком є чутливість до шуму: якщо дисперсія $\varepsilon(x, y)$ велика, точність аналізу знижується, що потребує додаткової фільтрації. Складність підбору параметрів $a_{i,j}$ також ускладнює обробку великих зображень, особливо в реальному часі, наприклад, у системах автоматичного аналізу [29].

Фрактальні моделі

Фрактальні моделі моделюють шорсткість і самоподібність природних поверхонь. Детермінований фрактал — це множина A що складається з N зменшених копій себе з масштабом r , із фрактальною розмірністю [29]:

$$D = \frac{\log N}{\log(1/r)}, \quad (3.3)$$

яка вимірює грубість текстури. Для оцінки розмірності використовується метод коробки, що аналізує, як кількість коробок, потрібних для покриття множини, змінюється зі зміною масштабу. Лакуність, яка обчислюється як [29]:

$$L(L) = \frac{M^2(L) - (M(L))^2}{(M(L))^2}, \quad (3.4)$$

дозволяє розрізнити текстур з однаковою розмірністю, вимірюючи нерівномірність маси фрактального набору та допомагає класифікувати текстур: низька для дрібних, висока для грубих. У віддаленому зондуванні фрактальні моделі класифікують ландшафти, наприклад, ліси чи пустелі. У медичних зображеннях вони сегментують тканини, наприклад, для аналізу рентгенівських знімків. Пентланд стверджує, що природні поверхні, як-от шкіра чи ґрунт, моделюються як ізотропні фрактали. Для синтезу текстур застосовується алгоритм зміщення середньої точки. Недоліком є складність оцінки розмірності

для недетермінованих текстур. Приклад згенерованої текстури можна побачити на рис. 3.2 [29].

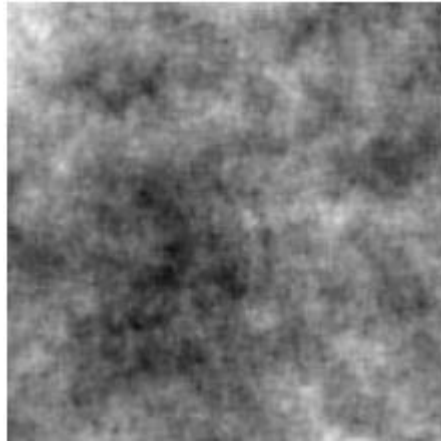


Рисунок 3.2 –Текстура згенерована за допомогою фрактальної моделі

Статистичні моделі

Кореляційні матриці аналізують просторовий розподіл сірих рівнів через функцію [29]:

$$R(\Delta x, \Delta y) = E[I(x, y)I(x + \Delta x, y + \Delta y)]. \quad (3.5)$$

Ця кореляційна функція пов'язана з автокореляційними ознаками, де вона нормалізується для оцінки періодичності текстур. Вони є основою для класифікації та сегментації текстур, використовуючи ознаки, такі як енергія, контраст чи ентропія. У віддаленому зондуванні вони класифікують поверхні, наприклад, водойми чи поля. У медичних додатках метод сегментує тканини в рентгенівських зображеннях, допомагаючи виділити контури органів. У РСА-зображеннях кореляційні матриці усувають шум шляхом відновлення зображень, але потребують великих обсягів даних для точного аналізу. У синтезі текстур вони менш ефективні, ніж MRF, через обмежену здатність генерувати візерунки [29].

Основним недоліком є чутливість до шуму: перешкоди можуть спотворити просторові розподіли, що знижує точність класифікації. Крім того, метод потребує значних обчислень, особливо для великих зображень, що ускладнює його використання в реальному часі. [29].

Моделі Габора

Фільтри Габора моделюють текстури через спектральний аналіз. Ці фільтри представляють зображення як сукупність гармонійних компонентів із різними частотами та орієнтаціями, використовуючи функцію Габора [29]:

$$f(x, y) = \exp\left(-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right) \cos(2\pi u_0 + \varphi), \quad (3.5)$$

де u_0 і φ - частота і фаза синусоїдальної хвилі, σ_x та σ_y - розміри гауссової огибаючої в напрямках x та y відповідно. Цей підхід часто використовується для виділення текстурних елементів, такі як смужки чи крапки, аналізуючи енергію в різних напрямках. У віддаленому зондуванні їх застосовують для класифікації поверхонь, наприклад, виявлення доріг чи полів. У медичних зображеннях вони сегментують тканини, наприклад, у МРТ для розпізнавання пухлин. У синтезі текстур Gabor-фільтри менш ефективні, ніж MRF, але корисні для створення періодичних візерунків. Їхня перевага — здатність адаптуватися до різних масштабів і орієнтацій, що робить їх придатними для аналізу складних текстур. Однак основним недоліком є висока обчислювальна складність, особливо при використанні багатьох фільтрів для покриття частотного діапазону. Це обмежує їхнє застосування в реальному часі, наприклад, у відеопотоках. [29].

3.3 Висновки до третього розділу

Розглянуто різні моделі, які використовуються для синтезу текстурних зображень. Загалом серед підходів виділяються: структурні моделі, стохастичні моделі, статистичні моделі, марковські випадкові поля, моделі Габора, фрактальні моделі, авторегресивні моделі. Дані моделі дозволяють синтезувати текстурні зображення, які описують властивості текстур, що зустрічаються на реальних зображеннях. Тому дані моделі можуть бути використані для створення штучних зображень на основі яких проводиться попереднє дослідження алгоритмів фільтрації, сегментації, класифікації, розпізнавання з метою їх дослідження та подальшого впровадження для реальних систем обробки зображень, які вирішують відповідні практичні задачі.

4 ОГЛЯД МЕТОДІВ ПРИДУШЕННЯ ЗАВАД

Метою придушення завад є обробка зображень для отримання результату, який найкраще відповідає вимогам конкретного застосування. Поняття "конкретне" підкреслює проблемно-орієнтовану природу методів, адже підхід, ефективний для обробки медичних рентгенівських зображень, може бути менш придатним для астрономічних зображень, отриманих із космічних апаратів. Придушення завад відіграє ключову роль у підвищенні якості текстурних зображень, забезпечуючи чіткість деталей для візуального аналізу та автоматизованих систем, таких як розпізнавання об'єктів чи діагностика [14].

Методи придушення завад поділяються на дві основні категорії: просторові та частотні. Просторові методи базуються на безпосередній модифікації значень пікселів, що дозволяє ефективно усувати завади, такі як шум чи артефакти, у текстурних структурах зображення. Частотні методи використовують перетворення Фур'є для аналізу та коригування частотних складових сигналу, що особливо корисно для придушення періодичних завад. Обидва підходи є важливими для відновлення текстурної чіткості залежно від типу завад і специфіки задачі [14].

4.1 Просторове фільтрування

Методи обробки у просторовій області ґрунтуються на застосуванні оператора до пікселів зображення, що описується виразом [14]:

$$g(x, y) = T[f(x, y)], \quad (4.1)$$

де $f(x, y)$ — вхідне зображення, $g(x, y)$ — вихідне зображення, а T — оператор, застосований до f у межах околу точки (x, y) . Цей оператор може обробляти пікселі одного зображення або набір зображень, наприклад, шляхом покомпонентного усереднення послідовності кадрів для зменшення шуму. Такі методи ефективно усувають завади, як-от імпульсний чи гаусівський шум, у текстурних зображеннях, зберігаючи їхні деталі [14].

Процес, зображений на рис. 4.1, полягає в послідовному переміщенні околу по пікселях зображення та застосуванні оператора T до пікселів у цьому околі для формування значення вихідного зображення в точці (x, y) . Таким чином, значення $g(x, y)$ у певній точці є результатом дії оператора T на окіл із центром у точці (x_0, y_0) , як показано на рис. 4.1. Цей підхід дозволяє локально коригувати текстурні структури [14].

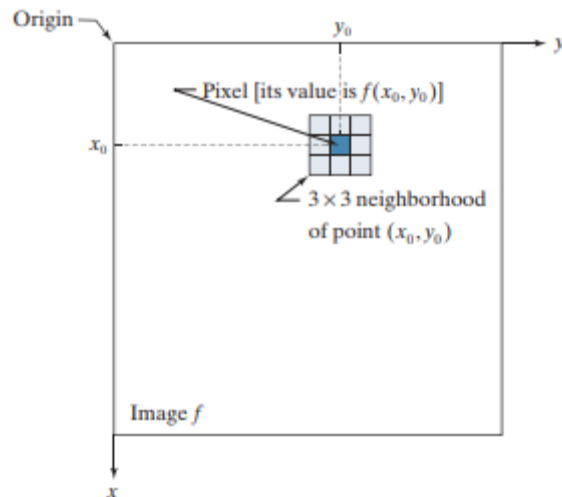


Рисунок 4.1 – Окіл розміром 3×3 навколо точки (x_0, y_0) в зображенні

Наприклад, припустимо, що окіл має розмір 3×3 , а оператор T визначений як "обчислити середнє значення інтенсивності пікселів в околі". Розглянемо довільну точку в зображенні, наприклад, $(100, 150)$. Результат у цій точці у вихідному зображенні, $g(100, 150)$, дорівнює сумі значень $f(100, 150)$ та її 8 сусідів, поділений на 9 [14].

Найменший можливий окіл має розмір 1×1 . У цьому випадку значення g залежить лише від значення f в одній точці (x, y) , і оператор T у рівнянні (4.1) стає функцією перетворення інтенсивності, яка має вигляд [14]:

$$s = T(r). \quad (4.2)$$

Для простоти у позначеннях ми використовуємо s та r , щоб позначити, відповідно, інтенсивність у g та f у будь-якій точці (x, y) .

Наприклад, якщо $T(r)$ має вигляд, показаний на рис. 4.2(a), це приводить до зображення з вищою контрастністю шляхом затемнення рівнів нижче k та

освітлення рівнів вище k . У граничному випадку, показаному на рис. 4.2(б), $T(r)$ формує двоярусне (бінарне) зображення. Таке перетворення називається функцією порогового значення [14].

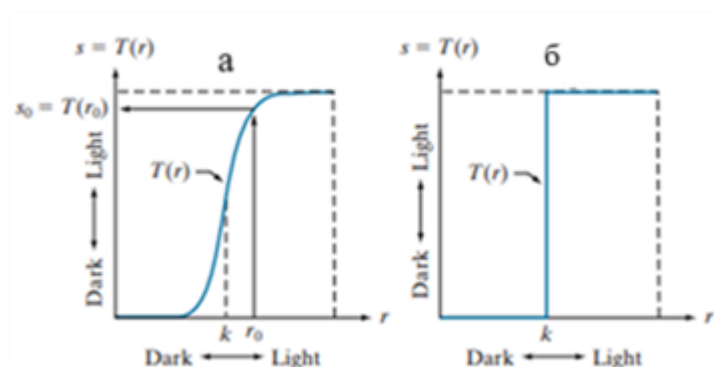


Рисунок 4.2 – (а) Функція розтягування контрасту. (б) Функція порогового значення

4.1.1 Згладжувальні просторові фільтри

Згладжувальні (або усереднювальні) фільтри в просторовій області застосовуються для зменшення різких змін інтенсивності пікселів, що є характерною ознакою випадкового шуму, такого як гаусівський чи імпульсний. Основне завдання цих фільтрів — зниження шуму, що покращує чіткість текстурних деталей зображень. Вони також використовуються перед повторним семплюванням для запобігання аліасінгу — ефекту накладання сигналів під час дискретизації, що спричиняє нечіткість. Крім того, згладжування допомагає усунути дрібні деталі, розмір яких менший за ядро фільтра, та хибні контури, що виникають через обмежену кількість рівнів інтенсивності. Згладжувальні фільтри часто комбінуються з іншими методами обробки, такими як модифікація гістограми чи підвищення різкості, для відновлення текстурних структур у задачах, як-от синтез текстур у комп'ютерній графіці [14].

Лінійна просторова фільтрація базується на згортці зображення з ядром фільтра. У разі згладжувального фільтра згортка розмиває зображення, а ступінь розмиття залежить від розміру ядра та його коефіцієнтів. Фільтри низьких частот, до яких належать згладжувальні, є основою для багатьох задач обробки зображень, оскільки вони слугують базою для створення інших фільтрів, зокрема

високочастотних, що підвищують різкість. Такі методи ефективно придушують завади, зберігаючи текстурні особливості зображень, що є критично важливим для застосувань, таких як комп'ютерний зір чи обробка фотографій [14].

Усереднювальний фільтр

Найпростішим прикладом фільтра низьких частот є усереднювальний фільтр, у якому всі коефіцієнти мають однакову вагу, зазвичай одиницю. На рис. 4.3 зображено приклад усереднювального фільтра розміром 3×3 , який ефективно згладжує завади, такі як гаусівський чи імпульсний шум, у текстурних зображеннях, зберігаючи їхні основні структури [14].

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Рисунок 4.3 – Усереднювальний фільтр.

Усереднювальний фільтр розміром $m \times n$ складається з масиву $m \times n$ одиничних коефіцієнтів, помножених на нормалізуючу константу $1/(m * n)$. Нормалізація виконує дві функції: по-перше, забезпечує збереження середньої інтенсивності в областях із постійною яскравістю після фільтрації; по-друге, запобігає зміщенню значень пікселів, гарантуючи, що сума пікселів у вхідному та вихідному зображеннях залишається однаковою. Оскільки всі рядки та стовпці ядра ідентичні, його ранг дорівнює 1. Цей метод широко застосовується для придушення шуму в текстурних зображеннях у задачах, таких як обробка медичних зображень чи аналіз аерофотознімків [14].

Ядра гауссових фільтрів низьких частот

Усереднювальні фільтри є зручними для швидкої обробки зображень, але мають суттєві недоліки. Вони погано відтворюють ефект розмиття оптичних систем і спричиняють небажане згладжування в перпендикулярних напрямках, що може погіршити якість зображень із високою деталізацією або чіткими геометричними структурами. Для текстурних зображень, де важливі дрібні деталі, це може призводити до втрати текстурної чіткості [14].

На відміну від усереднювальних, гаусівські фільтри є ізотропними (круглосиметричними) і роздільними, що забезпечує кращу якість згладжування. Вони поєднують обчислювальну ефективність із додатковими властивостями, що роблять їх оптимальними для обробки зображень. Гаусівські фільтри ефективно придушують шум, зберігаючи текстурні деталі, що є важливим для таких застосувань, як синтез текстур у комп'ютерній графіці чи аналіз поверхонь у комп'ютерному зорі [14].

$$w(s, t) = G(s, t) = Ke^{-\frac{s^2 + t^2}{2\sigma^2}}. \quad (4.3)$$

Змінні s та t у рівнянні (3.3) є дійсними (зазвичай дискретними) числами. Тому це рівняння можна спростити до такого вигляду [14]:

$$G(r) = Ke^{-\frac{r^2}{2\sigma^2}}, \quad (4.4)$$

де r — це відстань від центру до будь-якої точки на функції G . На рис. 4.4 показані значення r для кількох розмірів ядер, використовуючи цілі числа для s і t [14].

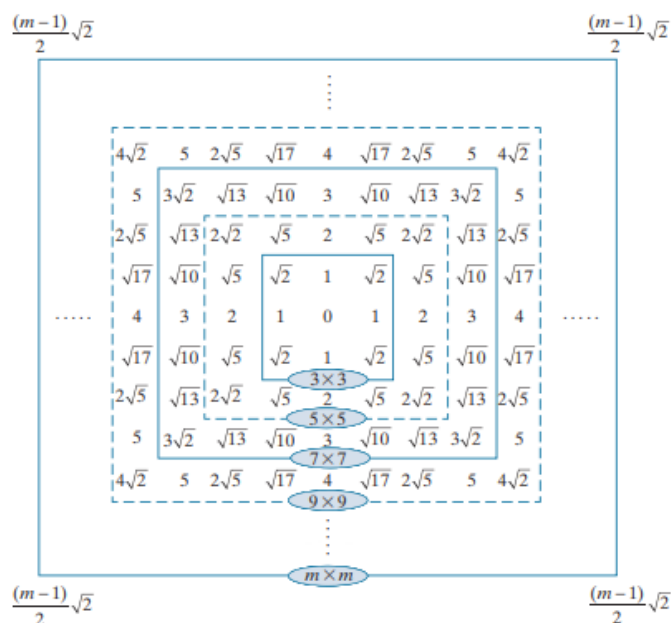


Рисунок 4.4 – Відстані від центру для різних розмірів квадратних фільтрів

Відстань, піднесена до квадрату, до кута для ядра розміру $t \times t \in [14]$:

$$r_{max}^2 = \left[\frac{(m-1)}{2} \sqrt{2} \right]^2 = \frac{(m-1)^2}{2}. \quad (4.5)$$

Фільтри Гауса є роздільними, що дозволяє отримати 2D-ядро, вибираючи вибірки вздовж поперечного розрізу через центр. Це дозволяє зменшити обчислювальну складність. Також важливо зазначити, що значення функції Гауса на відстані більше ніж 3σ від середнього значення є малими, тому можна використовувати ядро розміру $[6\sigma] \times [6\sigma]$ для досягнення практично однакового результату. На рис. 4.5 зображена дискретизація функції Гауса [14].

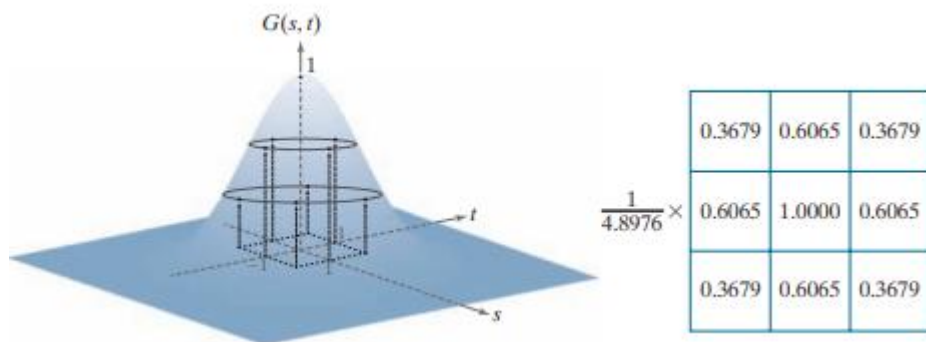


Рисунок 4.5 – Дискретизація функції Гауса для отримання дискретного ядра Гауса. Результат 3×3 ядра

4.1.2 Порядково-статистичні фільтри

Фільтри порядкової статистики належать до нелінійних просторових методів, які оперують пікселями в околі фільтра шляхом їхнього впорядкування. Згладжування досягається заміною значення центрального пікселя на величину, отриману за результатами ранжування. Найпоширенішим у цій категорії є **медіанний фільтр**, який замінює значення центрального пікселя медіаною інтенсивностей пікселів у його околі, враховуючи й сам центральний піксель. Медіанні фільтри ефективно знижують певні типи шуму, зокрема спекл-шум та імпульсний (відомий як шум "сіль і перець", що проявляється у вигляді чорних і білих точок, як показано на рис. 3.7), спричиняючи мінімальне розмиття порівняно з лінійними згладжувальними фільтрами того ж розміру. Це робить їх ідеальними для збереження текстурних деталей у задачах, таких як обробка відеостримів чи комп'ютерний зір [14].

Медіанний фільтр, що відповідає 50-му перцентилю відсортованих значень, є найпоширенішим, але не єдиним у цій категорії. Використання 100-го перцентиля дає max-filter, який визначає найяскравіше значення в околі, наприклад, $R = \max\{z_k | k = 1, 2, 3, \dots, 9\}$ для околу 3×3 . Цей фільтр застосовується для виділення яскравих точок або ерозії темних ділянок, прилеглих до світлих. Натомість min-filter, що відповідає 0-му перцентилю, використовується для протилежних задач, наприклад, для виділення темних областей. Такі методи ефективні для обробки текстурних зображень у задачах синтезу текстур у комп'ютерній графіці чи аналізу поверхонь [14].

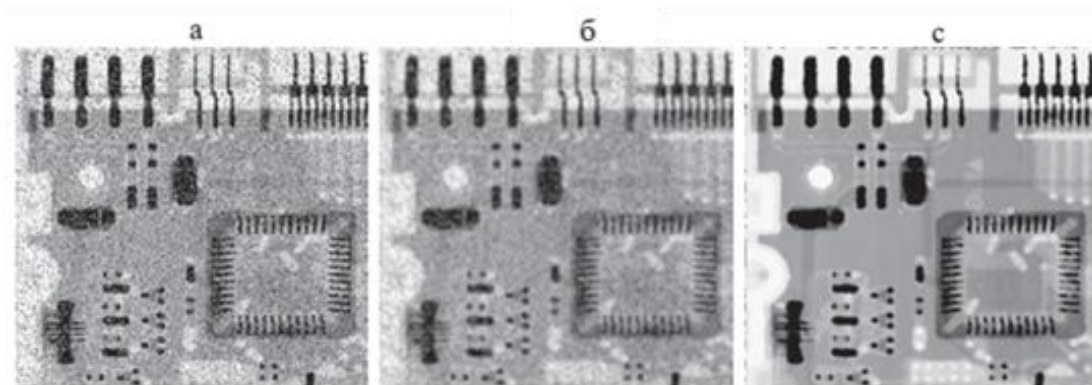


Рисунок 4.7 – (а) Рентгенівське зображення плати, пошкоджене шумом "сіль і перець". (б) Зменшення шуму за допомогою гаусівського низькочастотного фільтра. (с) Зменшення шуму за допомогою медіанного фільтра

4.2 Частотна фільтрація

Частотна область характеризує швидкість зміни значень пікселів у просторовій структурі зображення. Частота відображає темп варіацій кольорових чи інтенсивних компонентів: високі частоти відповідають швидким змінам, що спостерігаються в деталізованих або контрастних ділянках, таких як краї текстур, тоді як низькі частоти пов'язані з повільними змінами в однорідних і плавних областях. Для текстурних зображень аналіз частотної області дозволяє ефективно виявляти та придушувати завади, зберігаючи ключові деталі [31].

На відміну від просторової області, де обробка здійснюється безпосередньо зі значеннями пікселів, у частотній області зображення спочатку перетворюється на набір частотних компонентів. Ці компоненти поділяються на високочастотні, що відповідають краям і деталям, і низькочастотні, що відображають гладкі ділянки. Результатом перетворення є не зображення, а його частотна трансформація. Для відновлення зображення після обробки, наприклад, придушення шуму чи артефактів, застосовується зворотне перетворення. Такий підхід є ефективним для аналізу текстурних структур у задачах синтезу текстур чи обробки зображень у комп'ютерній графіці [31].

Процедура фільтрації в частотній області є простою і включає в себе наступні кроки:

1. Вихідне зображення множиться на $(-1)^{x+y}$, щоб його перетворення Фур'є було центрованим.
2. Обчислюється пряме перетворення Фур'є $F(u, v)$ зображення, що отримано після кроку 1.
3. Функція $F(u, v)$ множиться на функцію фільтра $H(u, v)$.
4. Обчислюється обернене перетворення Фур'є від результату кроку 3.
5. Виділяється дійсна частина результату кроку 4.
6. Результат кроку 5 множиться на $(-1)^{x+y}$.

4.2.1 Ідеальний низькочастотний фільтр

Двовимірний низькочастотний фільтр, який пропускає без затухання всі частоти всередині кола радіуса D_0 від початку координат і повністю «обрізає» всі частоти за межами цього кола, називається **ідеальним низькочастотним фільтром**. Його передавальна функція задається як [14]:

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases} \quad (4.7)$$

D_0 — додатна константа (радіус відсікання), $D(u, v)$ — відстань між точкою (u, v) у частотній області та центром частотного прямокутника розміру $P \times Q$

Характеристики:

1. Фільтр ILPF є радіально симетричним щодо початку координат, тобто його поведінку можна повністю описати перерізом у радіальному напрямку.
2. Фільтр повністю передає частоти всередині кола радіуса D_0 і повністю пригнічує частоти за його межами.
3. Функція передачі фільтра отримується шляхом обертання радіального перерізу на 360° .
4. Частота відсікання D_0 визначає точку переходу між зонами $H(u, v) = 1$ та $H(u, v) = 0$.

4.2.2 Гаусовий низькочастотний фільтр

Гаусовий низькочастотний фільтр описується передавальною функцією [14]:

$$H(u, v) = e^{-\frac{D^2(u, v)}{2\sigma^2}}, \quad (4.8)$$

де $D(u, v)$ — відстань до центру частотної області, а D_0 — частота відсікання.

Особливості:

1. На відміну від ідеального фільтра, гаусовий не має різкого відсікання, а поступово зменшує частоти.
2. При $D = D_0$ передавальна функція спадає до 0.607 від максимуму.
3. Не створює ряди (ringing effect) у просторі, оскільки його зворотне перетворення Фур'є також є Гаусовим.
4. Зворотна залежність між шириною фільтра в частотній і просторовій областях: вузький фільтр у частотній області дає широку згортку в просторі і навпаки.

4.2.3 Низькочастотний фільтр Баттерворта

Передавальна функція низькочастотного фільтра порядку n , з частотою відсікання на відстані D_0 від центру частотної області, має вигляд [14]:

$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)}{D_0}\right]^{2n}}, \quad (4.9)$$

де $D(u, v)$ — відстань до центру частотної області.

Особливості:

1. Фільтр дозволяє плавно переходити від низьких частот до високих, підлаштовуючи його характеристики за допомогою параметра n .
2. Для великих значень n функція фільтра наближається до ідеального низькочастотного фільтра, а для менших значень — до гаусового фільтра.

3. Фільтри порядку 1 не мають ряди (ringing), а фільтри порядку 2 або 3 мають незначні ряди, які стають більш виразними для фільтрів вищих порядків.

4.2.4 Ідеальні високочастотні фільтри отримані з низькочастотних фільтрів

Як і у випадку з ядрами в просторі віднімання передавальної функції низькочастотного фільтра від 1 дає відповідну передавальну функцію високочастотного фільтра в частотній області [14]:

$$H_{HP}(u, v) = 1 - H_{LP}(u, v), \quad (4.10)$$

де $H_{LP}(u, v)$ — передавальна функція низькочастотного фільтра.

Таким чином, згідно з рівнянням (3.7), передавальна функція ідеального високочастотного фільтра має вигляд [16]:

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}, \quad (4.11)$$

де $D(u, v)$ — відстань від центру прямокутника частот.

Аналогічно, згідно з рівнянням (4.8), передавальна функція гаусового високочастотного фільтра матиме форму [14]:

$$H(u, v) = 1 - e^{-\frac{D^2(u, v)}{2D_0^2}}. \quad (4.12)$$

З рівняння (4.9) передавальна функція Баттерворта високочастотного фільтра виглядає так [14]:

$$H(u, v) = \frac{1}{1 + [\frac{D_0}{D(u, v)}]^{2n}}. \quad (4.13)$$

4.2.5 Фільтр Вінера

Вінерівська фільтрація є класичним методом для відновлення зображень, уражених адитивним шумом і використовується для мінімізації середньоквадратичного відхилення між оригінальним зображенням і відфільтрованим [27]:

$$e^2 = E(f - \hat{f})^2, \quad (4.14)$$

де e – оператор математичного очікування, а f – неспотворене зображення.

Вона базується на моделі деградації [14]:

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y), \quad (4.15)$$

де $h(x, y)$ — функція розмиття, що описує деградацію через згортку, а $\eta(x, y)$ — адитивний шум із нульовим середнім, некорельований із $f(x, y)$. У частотній області після дискретного перетворення Фур'є (ДПФ) це стає [16]:

$$G(u, v) = H(u, v)F(u, v) + N(u, v), \quad (4.16)$$

де $G(u, v), F(u, v), N(u, v)$ — спектри зашумленого зображення, оригінального зображення та шуму, а $H(u, v)$ — спектр функції деградації.

Класичний підхід вінерівської фільтрації у частотній області полягає в оцінці спектру відфільтрованого зображення через вираз [27]:

$$\hat{F}(u, v) = \left[\frac{1}{H(u, v)} \times \frac{|H(u, v)|^2}{H(u, v)|H(u, v)|^2 + \frac{S_\eta(u, v)}{S_f(u, v)}} \right] G(u, v), \quad (4.17)$$

де $H(u, v)$ – функція, що спотворює; $|H(u, v)|^2 = H^*(u, v)H(u, v)$; $H^*(u, v)$ – комплексно-спряжена функція $H(u, v)$; $S_\eta(u, v) = |N(u, v)|^2$ – енергетичний спектр шуму; $S_f(u, v) = |F(u, v)|^2$ – спектр неспотвореного зображення; частка $S_\eta(u, v)/S_f(u, v)$ називається енергетичним співвідношенням шум/сигнал. Цей вираз забезпечує баланс між збереженням сигналу та приглушенням шуму: у частотах із високим співвідношенням сигнал/шум фільтр пропускає більше сигналу, а у частотах із низьким співвідношенням — зменшує шумові компоненти. Для оцінки спектральних характеристик використовують середні значення: $\eta_A = \frac{1}{MN} \sum_u \sum_v S_\eta(u, v)$ для сигналу та $f_A = \frac{1}{MN} \sum_u \sum_v S_f(u, v)$ для шуму, що дозволяє адаптувати фільтр до реальних даних [27].

Вінерівська фільтрація ефективна для зображень із відомими характеристиками шуму, але потребує точного визначення $H(u, v), S_f(u, v)$ і $S_\eta(u, v)$, що часто викликає труднощі. Вона також припускає

стаціонарність шуму, що може не відповідати реальним умовам. Метод широко застосовується в обробці зображень для зменшення шуму.

4.2.6 Фільтр Калмана

Фільтр Калмана є рекурсивним алгоритмом для придушення адитивного гаусівського шуму, зокрема на текстурних зображеннях, адаптуючись до змінних умов у реальному часі. Він працює в два етапи: прогнозування та оновлення, об'єднуючи оцінку стану з новими вимірами. Для текстурних зображень, де шум маскує деталі, Калман використовує лінійну модель динаміки пікселів [32].

Алгоритм базується на моделі стану [32]:

$$x_k = F_k x_{k-1} + B_k u_k + w_k, \quad (4.18)$$

де F_k є моделлю переходу стану, що застосовується до попереднього стану x_{k-1} ; B_k є моделлю впливів керування, що застосовується до вектора керування u_k ; w_k є шумом з багатовимірний нормальний розподіл, нульовим середнім значенням і коваріацією Q_k .

Вимірювання [32]:

$$z_k = H_k x_k + v_k, \quad (4.19)$$

де z_k — спостереження, H_k — матриця спостереження, v_k — шум із коваріацією R_k .

Прогнозування оцінки стану [32]:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k|k-1} + B_k u_k. \quad (4.20)$$

Коваріація передбаченої оцінки [32]:

$$P_{k|k-1} = F_k P_{k|k-1} F_k^T + Q_k. \quad (4.21)$$

Оновлена оцінка стану [32]:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k. \quad (4.22)$$

Для текстурних зображень Калман адаптується до локальних областей, моделюючи шум як гаусівський, що дозволяє приглушувати шум, зберігаючи текстури, якщо F_k і H_k точно відображають динаміку. Перевага — адаптивність до послідовних кадрів, що корисно для відео чи динамічних текстур, без потреби

в повному спектрі шуму. Обмеження включають вимогу лінійності моделі, нормальності шуму та складність налаштування Q_k і R_k . Неправильна модель може призвести до втрати деталей [32].

4.2.7 Фільтр Лі

Фільтр Лі (Lee filter) є адаптивним методом, призначеним для придушення спекл-шуму, мультиплікативного шуму, який часто впливає на текстурні зображення, наприклад, синтетичні текстури в комп'ютерному зорі чи РСА-зображення. Цей фільтр базується на мінімізації середньоквадратичної помилки, замінюючи значення центрального пікселя зваженою комбінацією його оригінального значення та локального середнього в околі. Основна формула фільтра Лі [26]:

$$I_{out} = I_{in}W + \bar{I}(1 - W), \quad (4.23)$$

де I_{out} — відфільтроване значення пікселя, I_{in} — вхідне значення, $\bar{I}$ — локальне середнє в околі (наприклад, у вікні 3×3), а $W = 1 - \frac{\sigma_n^2}{\sigma_x^2}$ — вага, яка залежить від співвідношення дисперсії шуму σ_n^2 до дисперсії зображення σ_x^2 . Якщо дисперсія мала, W наближається до 0, і фільтр діє як середнє; при великій дисперсії W наближається до 1, зберігаючи оригінальне значення [26].

Такий підхід дозволяє ефективно зменшувати спекл-шум, зберігаючи текстурні деталі, такі як візерунки чи градієнти. Фільтр аналізує статистичні характеристики у вікні (наприклад, 3×3 чи 5×5), адаптуючись до локальних умов. У гомогенних областях він усуває шум, а в текстурних — захищає деталі. Це робить його ідеальним для текстурних зображень, де важливе збереження візерунків [26].

Перевагою фільтра Лі є його адаптивність, що забезпечує кращий баланс між придушенням шуму і збереженням текстур порівняно з лінійними фільтрами. Обмеження включають можливе розмивання дрібних деталей при великих вікнах або неточну оцінку дисперсії, що може погіршити результат. У задачах обробки

текстурних зображень, фільтр Лі часто використовується як базовий метод, іноді комбінуючись із іншими алгоритмами для підвищення якості [26].

4.2.8 Фільтр Фроста

Фільтр Фроста (Frost filter) є адаптивним методом, який добре придушує спекл-шуму на текстурних зображеннях. На відміну від статичних фільтрів, він використовує експоненційно затухаючу згортку, адаптуючись до локальної неоднорідності зображення. Його дія базується на моделі [26]:

$$\hat{x}(t) = z(t) * m(t), \quad (4.24)$$

де $\hat{x}(t)$ — відфільтроване значення, $z(t)$ — вхідне значення, а $m(t) = ae^{-\alpha|t|}$ — вага, що залежить від відстані $|t|$ та параметра затухання α , який регулюється локальними статистичними даними. У гомогенних областях фільтр сильніше згладжує шум, тоді як у текстурних ділянках зберігає деталі, такі як візерунки.

Особливістю фільтра є його підхід до країв: він менш усереднює в областях із високою неоднорідністю, що дозволяє зберегти структуру текстур, але може призводити до легкого розмиття країв порівняно з фільтром Лі. Параметр α обчислюється на основі локального середнього та дисперсії у вікні 5×5 , що робить його чутливим до змін у текстурних особливостях. Цей метод ідеально підходить для аналізу складних поверхонь чи синтетичних текстур, де важливе збереження візуальних деталей [26].

Перевагою фільтра Фроста є його здатність адаптивно обробляти неоднорідні області, забезпечуючи кращий баланс між придушенням спекл-шуму і збереженням текстур, ніж прості лінійні методи. Обмеження включають високу обчислювальну складність, а також залежність від точного налаштування α , що може вплинути на якість у сильно зашумлених областях. [26].

4.2.9 Сигма-фільтр

Sigma-фільтр фільтрує шум зображення, усереднюючи лише ті пікселі, які потрапляють у двосигмовий діапазон від центрального пікселя у скануючому вікні (наприклад, 5×5). Відомо, що двосигмова ймовірність гауссівського

розподілу становить 0,955. Пікселі поза цим діапазоном ігноруються, оскільки вважаються аномаліями, що дозволяє зберігати висококонтрастні деталі текстурних зображень, таких як РСА-зображення. Обчислення відфільтрованого значення $I'(x, y)$ у вікні $N \times N$ визначається як:

$$I'(x, y) = \frac{\sum_{i,j \in W} I(x+i, y+j) * \omega_{i,j}}{\sum_{i,j \in W} \omega_{i,j}}, \quad (4.25)$$

де $\omega_{i,j}$ – ваговий коефіцієнт. Верхні та нижні межі двосигмового діапазону для всіх рівнів сірого (наприклад, 256 відтінків) попередньо обчислюються та зберігаються в окремому масиві. Для центрального пікселя у вікні двосигмовий діапазон витягується з масиву, і пікселі у вікні порівнюються з цими межами. Пікселі, що потрапляють у межі, усереднюються, і це середнє замінює центральний піксель, а інші ігноруються. Метод є обчислювально ефективним, оскільки використовує лише арифметику з фіксованою точкою.

Результати фільтрації з вікном 5×5 показують, що більшість висококонтрастних деталей підсилюються, але зображення все ще містять дрібні темні плями. Для зменшення цього ефекту сигма-фільтр слід застосовувати повторно зі зменшенням значень σ . Як варіант, якщо в двосигмовому діапазоні менше двох-трьох пікселів, центральний піксель замінюється середнім значенням у вікні 3×3 . Варіація цього алгоритму застосовується для збереження текстур [26].

4.2.10 Висновки до четвертого розділу

Розділ продемонстрував, що обробка зображень для зниження шуму є проблемно-орієнтованою задачею, де вибір методу залежить від специфіки застосування, зокрема для текстурних зображень. Просторові методи, такі як згладжувальні, ефективно усувають гаусівський і імпульсний шум, зберігаючи основні текстурні структури, хоча можуть призводити до розмиття дрібних деталей. Порядково-статистичні фільтри, зокрема медіанний, max- і min-фільтри, вирізняються здатністю придушувати спекл-шум із мінімальним спотворенням текстур, що робить їх придатними для комп'ютерного зору.

Частотні методи, включаючи ідеальні, гаусові та Баттерворта низько- і високочастотні фільтри, дозволяють аналізувати та коригувати частотні компоненти, ефективно усуваючи періодичні завади та зберігаючи деталі текстур. Спеціалізовані фільтри, такі як Лі, Фроста і сигма-фільтр, адаптивно справляються зі спекл-шумом, адаптуючись до локальних умов, хоча мають обмеження, пов'язані з налаштуванням параметрів і обчислювальною складністю. Фільтр Вінера та Калмана забезпечують високоточне відновлення сигналу, особливо для адитивного шуму, але потребують точних моделей шуму й лінійності даних.

Отже, для текстурних зображень комбінація просторових і частотних методів, а також адаптивних фільтрів дозволяє оптимізувати якість обробки, зберігаючи візуальні деталі. Вибір методу залежить від типу шуму, обчислювальних ресурсів і цілей аналізу, що підкреслює необхідність подальшого дослідження для інтеграції цих підходів у практичні системи.

5 МОДЕЛЮВАННЯ ТА АНАЛІЗ АЛГОРИТМУ ДЛЯ ПРИДУШЕННЯ СПЕКЛ-ШУМУ

5.1 Генерація синтезованих зображень

Для оцінки ефективності алгоритму видалення спекл-шуму було використано три текстурні зображення розміром 256×256 пікселів рис. 5.1, що імітують різні природні та штучні візерунки: "хвилі", "цегляна кладка" і "трава". Вибір розміру 256×256 пікселів обґрунтовано компромісом між обчислювальною складністю та достатньою деталізацією текстур для аналізу. Кожна текстура створена з урахуванням специфічних особливостей, які дозволяють оцінити вплив спекл-шуму на різні типи візерунків: плавні переходи, чіткі межі та нерівномірні структури.



Рисунок 5.1 – Тестові текстурні зображення

Крім того, для розширення аналізу використано три реальних текстурних зображення рис 5.2, отриманих із відкритих джерел. Ці зображення приведено до розміру 256×256 пікселів для узгодженості з синтетичними текстурами, конвертовано в чорно-білий формат за допомогою функції MATLAB *rgb2gray*, а також нормалізовано до діапазону $[0,1]$ для подальшої обробки.



Рисунок 5.2 – Реальні текстурні зображення

Для моделювання спекл-шуму використано функцію MATLAB *imnoise* із типом шуму '*speckle*'. Спекл-шум додано з параметром *variance*, який варіювався від 0.05 до 0.2 для оцінки впливу різного рівня шуму. Візуальний аналіз зашумлених зображень показує характерну зернистість спекл-шуму, яка найбільш помітна в областях з однорідною текстурою. Зі збільшенням рівня шуму деталі зображення стають менш розрізненими, що ускладнює подальший аналіз текстури. Зашумлені зображення показано на рис. 5.3-рис. 5.5.

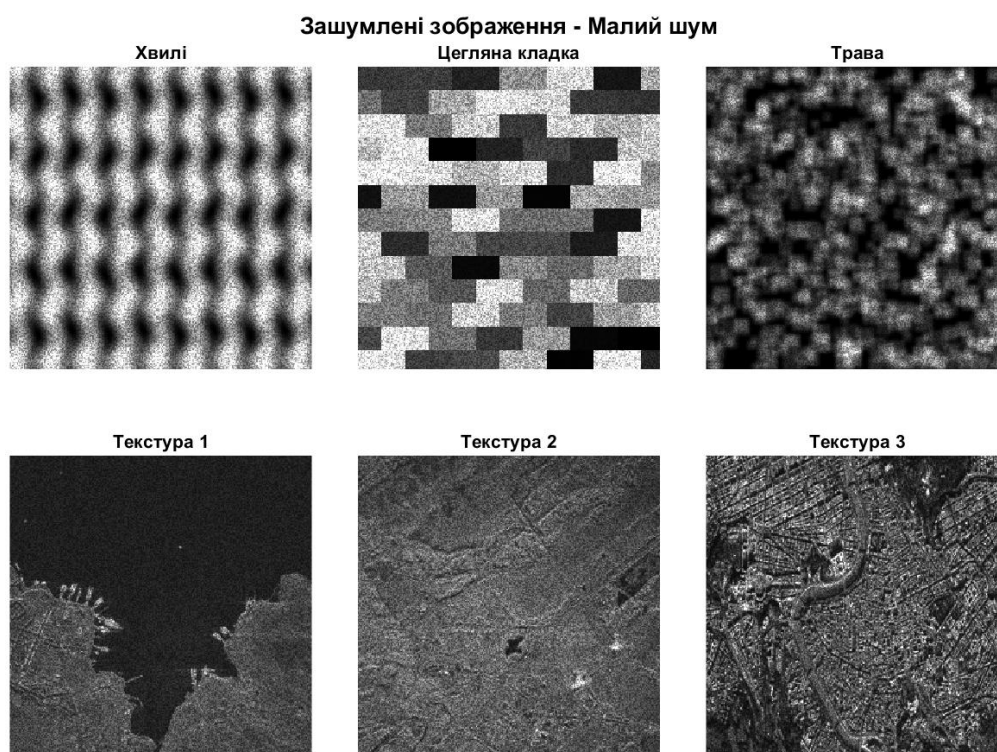


Рисунок 5.3 – Зображення з малим рівнем шуму

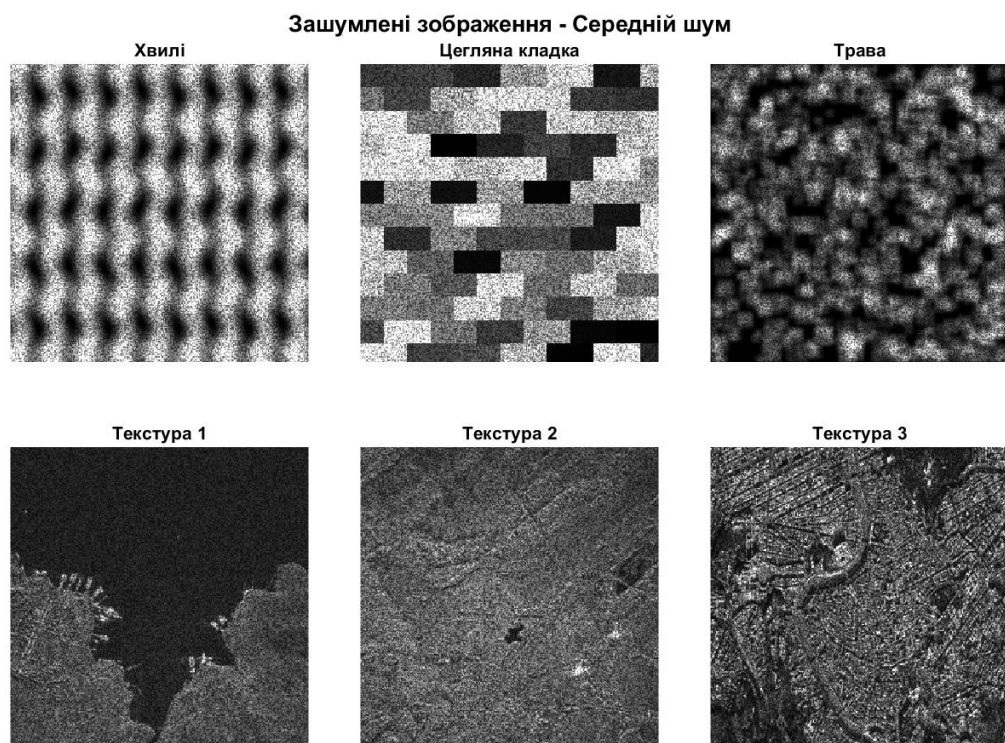


Рисунок 5.4 – Зображення з середнім рівнем шуму

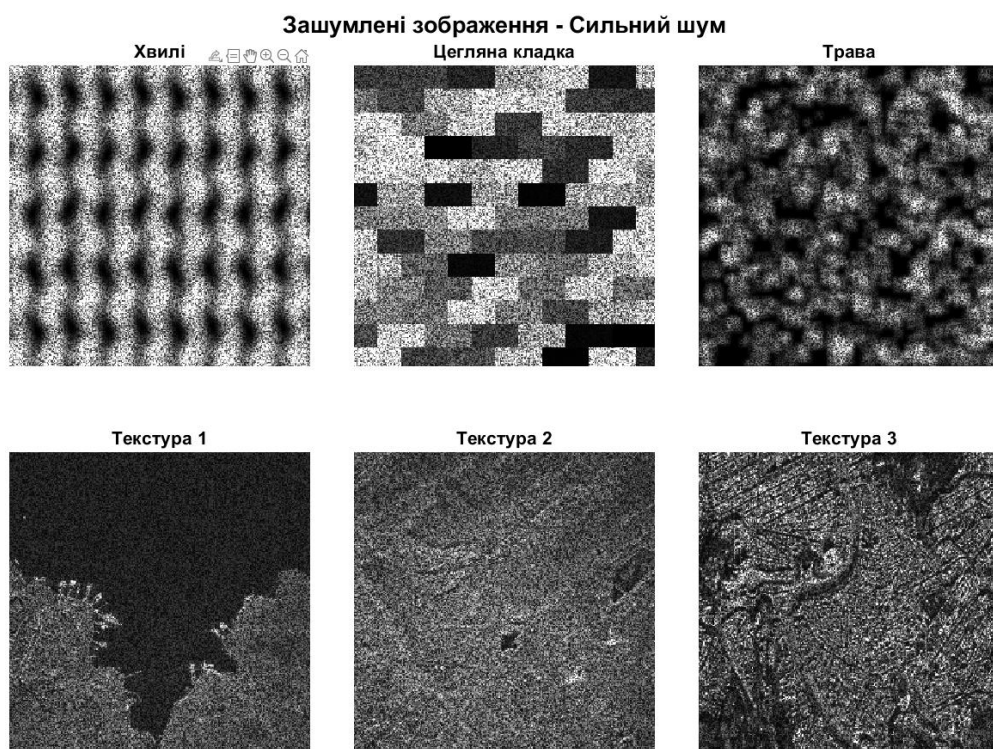


Рисунок 5.5 – Зображення з сильним рівнем шуму

5.2 Реалізація алгоритму фільтрації спекл-шуму

Для видалення спекл-шуму реалізовано алгоритм в MATLAB, який передбачає використання трьох різних фільтрів на вибір: Sigma-фільтр, Frost-фільтр і Lee-фільтр. Кожен фільтр адаптовано для роботи з текстурними зображеннями, враховуючи їхні особливості, такі як чіткість меж або нерівномірність деталей. Усі фільтри використовують квадратне вікно розміром 5×5 пікселів, що є компромісом між точністю фільтрації та обчислювальною ефективністю. Вибір розміру вікна обґрунтовано: менші вікна (наприклад, 3×3) недостатньо згладжують шум, а більші (наприклад, 7×7) надмірно розмивають деталі текстур.

Процес моделювання включає завантаження зображень, додавання спекл-шуму з дисперсією 0.05, 0.1 і 0.2, вибір фільтру, обчислення метрик PSNR і SSIM, збереження результатів та їх візуалізацію. Послідовність дій відображена на блок-схемі алгоритму рис. 5.6, яка деталізує ключові етапи.

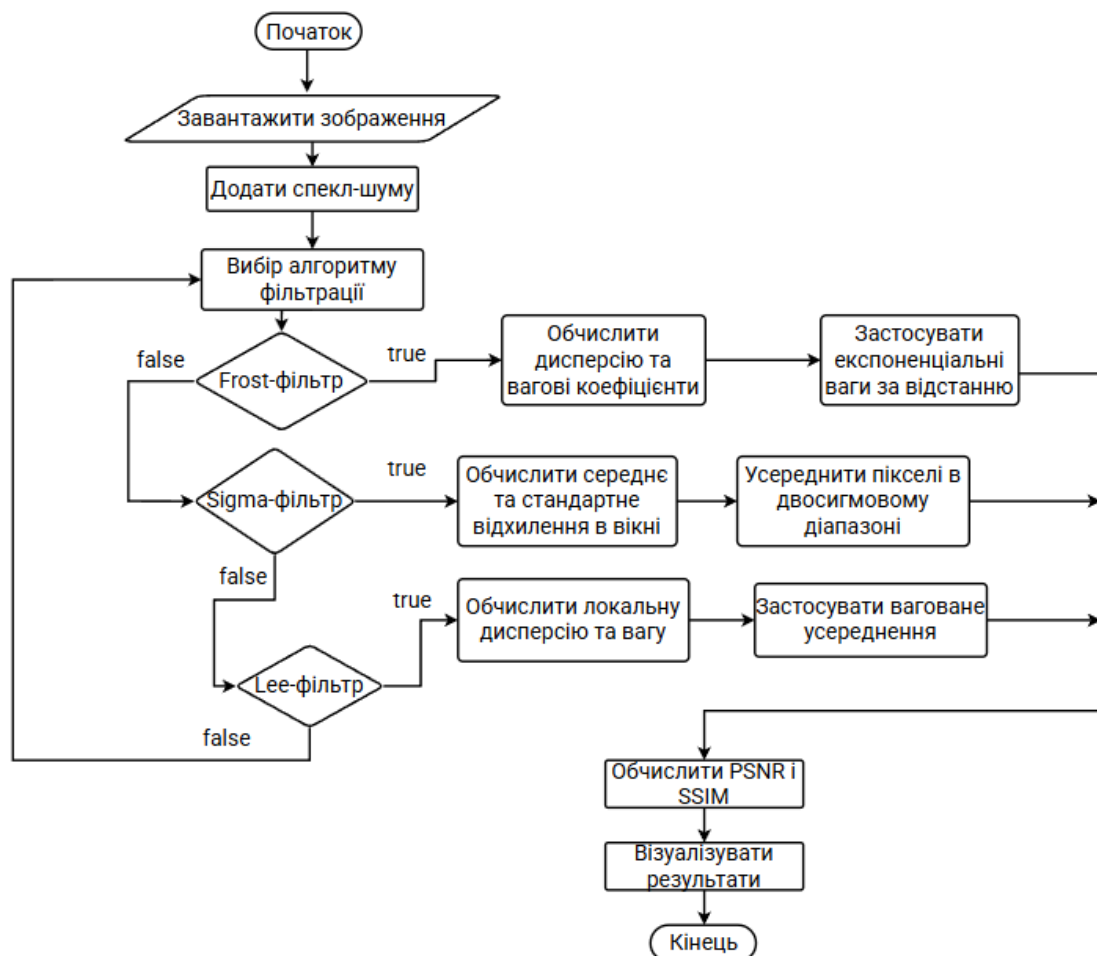


Рисунок 5.6 – Блок-схема алгоритму

Sigma-фільтр: Використовує статистичний підхід, відкидаючи пікселі, що значно відхиляються від середнього значення у вікні. Границі обробляються реплікацією. Його реалізація зображена на рис. 5.7.

```
% Sigma Filter
function J = sigmaFilter(I, windowSize, sigma)
    [rows, cols] = size(I);
    J = zeros(rows, cols);
    half = floor(windowSize/2);
    I_padded = padarray(I, [half half], 'replicate');
    for i = 1+half:rows+half
        for j = 1+half:cols+half
            window = I_padded(i-half:i+half, j-half:j+half);
            meanVal = mean(window(:));
            stdVal = std(window(:), 1);
            if stdVal == 0
                J(i-half, j-half) = meanVal;
            else
                validPixels = abs(window - meanVal) <= sigma * stdVal;
                if sum(validPixels(:)) == 0
                    J(i-half, j-half) = meanVal;
                else
                    J(i-half, j-half) = mean(window(validPixels));
                end
            end
        end
    end
end
```

Рисунок 5.7 – Реалізація Sigma-фільтра

Основні функції Sigma-фільтра:

`I_padded = padarray(I, [half half], 'replicate');`

Додає граничні пікселі методом реплікації для обробки країв.

`window = I_padded(i-half:i+half, j-half:j+half);` та `meanVal = mean(window(:));`

Вибирає вікно 5×5 і обчислює середнє значення пікселів.

`J(i-half, j-half) = mean(window(validPixels));`

Фільтрує пікселі в межах двосигмового діапазону та записує результат.

Frost-фільтр: Адаптивний метод із врахуванням просторової кореляції, де ваги обчислюються на основі відстані та локальної статистики. Реалізація зображена на рис. 5.8.

```
% Frost Filter
function ima-fi = Frost_filter(ima_nse, Damp_fact, sz)
    ima_nse = double(ima_nse);
    ima-fi = zeros(size(ima_nse));
    if mod(sz,2) ~= 1
        sz = sz + ~mod(sz,2);
    end
    mn = round((sz-1)/2);
    EImg = padarray(ima_nse, mn);

    % Обчислення матриці відстаней |
    [x,y] = meshgrid(-mn(1,1):mn(1,1), -mn(1,2):mn(1,2));
    S = sqrt(x.^2 + y.^2);

    for i = 1:size(ima_nse,1)
        for j = 1:size(ima_nse,2)
            K = EImg(i:i+sz(1,1)-1, j:j+sz(1,2)-1);
            meanV = mean(K(:));
            varV = var(K(:), 1);

            if meanV == 0
                meanV = eps;
            end
            B = Damp_fact * (varV / (meanV * meanV));
            Weigh = exp(-S .* B);

            if any(isnan(Weigh(:)))
                ima-fi(i,j) = mean(K(:));
            else
                sumWeigh = sum(Weigh(:));
                if sumWeigh == 0
                    ima-fi(i,j) = mean(K(:));
                else
                    ima-fi(i,j) = sum(K(:) .* Weigh(:)) / sumWeigh;
                end
            end
        end
    end
end
```

Рисунок 5.8 – Реалізація Frost-фільтра

Основні функції Frost-фільтра:

$EImg = \text{padarray}(ima_nse, mn);$ та $[x,y] = \text{meshgrid}(\dots); S = \text{sqrt}(x.^2 + y.^2);$

Додає граничні пікселі та обчислює матрицю відстаней для ваг.

```
Weigh = exp(-S .* B);
```

Застосовує експоненціальні ваги на основі відстані та дисперсії.

```
ima_fi(i,j) = sum(K(:) .* Weigh(:)) / sumWeigh;
```

Обчислює зважене середнє значення для фільтрації.

Lee-фільтр: Адаптивний метод із дисперсією шуму, що регулює вагу згладжування залежно від локальної неоднорідності. Границі обробляються реплікацією. Реалізація зображена на рис. 5.7.

```
% Lee Filter (з параметром варіації шуму)
function J = leeFilter(I, windowSize, noiseVariance)
    [rows, cols] = size(I);
    J = zeros(rows, cols);
    half = floor(windowSize/2);
    I_padded = padarray(I, [half half], 'replicate');
    for i = 1+half:rows+half
        for j = 1+half:cols+half
            window = I_padded(i-half:i+half, j-half:j+half);
            meanVal = mean(window(:));
            varLocal = var(window(:), 1);

            if (noiseVariance + varLocal) == 0
                w = 0;
            else
                w = 1 - noiseVariance / (noiseVariance + varLocal);
            end
            J(i-half, j-half) = meanVal + w * (I_padded(i,j) - meanVal);
        end
    end
end
```

Рисунок 5.9 – Реалізація Lee-фільтра

Основні функції Lee-фільтра:

```
I_padded = padarray(I, [half half], 'replicate');
```

Додає граничні пікселі методом реплікації.

```
w = 1 - noiseVariance / (noiseVariance + varLocal);
```

Обчислює вагу на основі шумової та локальної дисперсії.

```
J(i-half, j-half) = meanVal + w * (I_padded(i,j) - meanVal);
```

Застосовує ваговане усереднення для фільтрації пікселя.

5.3 Аналіз результатів фільтрації

Для оцінки якості видалення спекл-шуму використано дві ключові метрики: PSNR (Peak Signal-to-Noise Ratio) і SSIM (Structural Similarity Index). PSNR вимірює співвідношення сигнал/шум у децибелах, виходячи з середньоквадратичної помилки між оригінальним і відфільтрованим зображенням, і є показником загальної якості відновлення. Ця метрика обчислюється за формулою [33]:

$$PSNR = 10 * \log_{10}(\frac{R^2}{MSE}), \quad (4.26)$$

де MSE – середньоквадратична помилка, R – максимальне можливе значення пікселя (для нормалізованих зображень [0, 1]). Вищі значення PSNR вказують на меншу присутність шуму.

SSIM, навпаки, оцінює структурну схожість зображень, враховуючи яскравість, контраст і деталі, і варіюється від 0 (повна відмінність) до 1 (ідентичність). Її формула виглядає так [34]:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (4.27)$$

де μ_x, μ_y — середні значення пікселів зображень x (оригінал) і y (відфільтроване), σ_x^2, σ_y^2 — дисперсії пікселів, σ_{xy} — коваріація між x і y , а C_1, C_2 — константи для стабільності при нормалізованих зображеннях [0, 1]. Ці метрики доповнюють одне одного, дозволяючи аналізувати як кількісну, так і якісну сторони обробки.

Результати фільтрації проаналізовано для всіх шести текстур із використанням Sigma-, Frost- і Lee-фільтрів. Для візуального аналізу відфільтровані зображення від спекл-шуму різної дисперсії згруповано за типами та показано на рис. 5.10 – рис. 5.18.

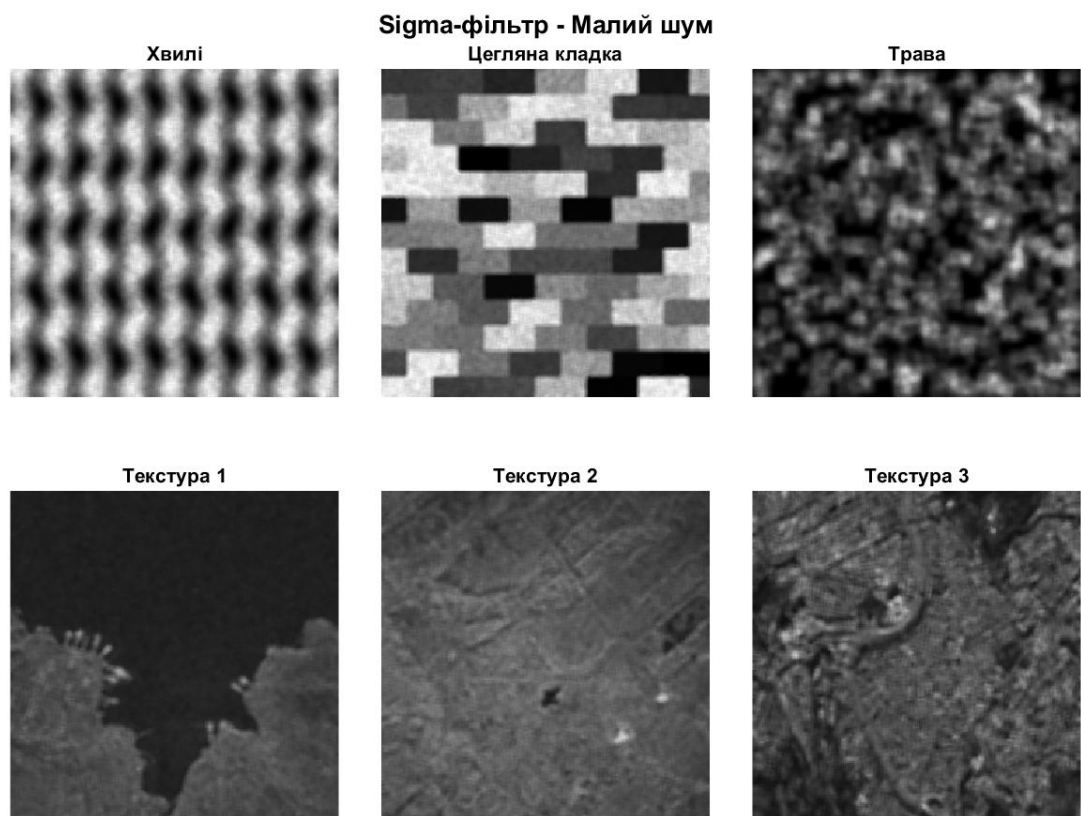


Рисунок 5.10 – Результати фільтрації Sigma-фільтром спекл-шуму дисперсією 0.05

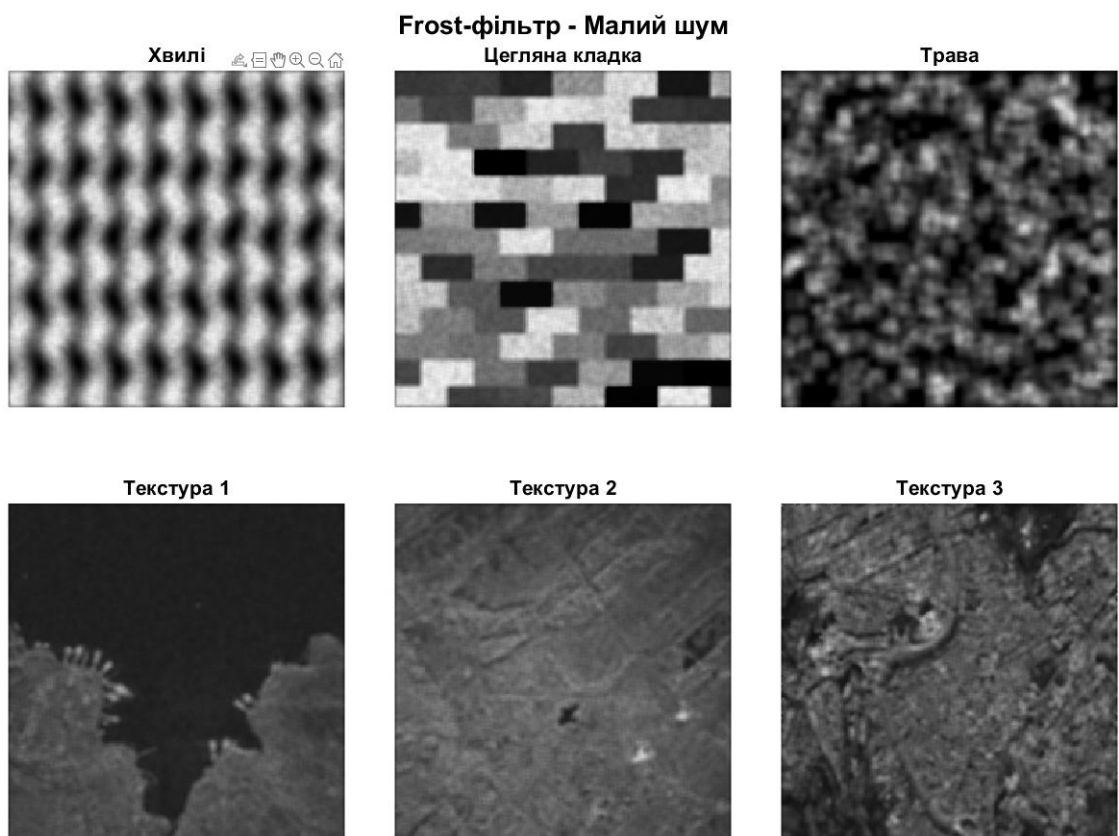


Рисунок 5.11 – Результати фільтрації Frost-фільтром спекл-шуму дисперсією 0.05

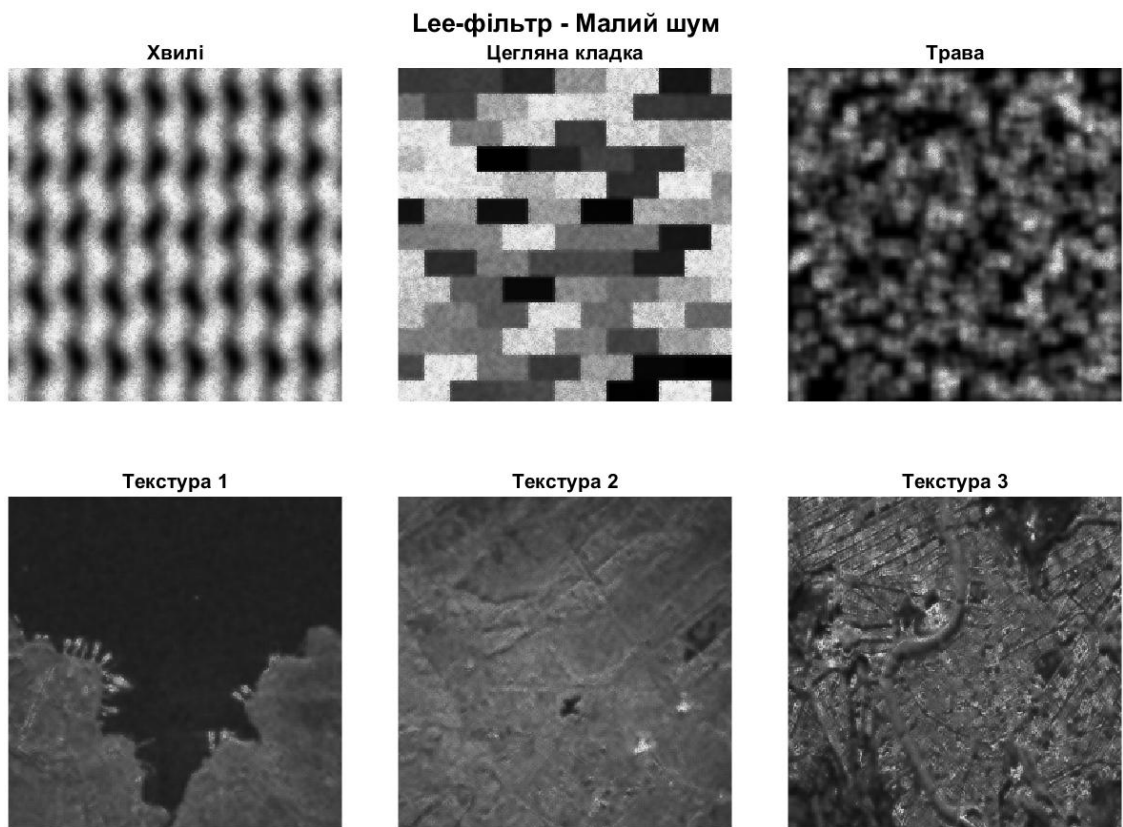


Рисунок 5.12 – Результати фільтрації Lee-фільтром спекл-шуму дисперсією 0.05

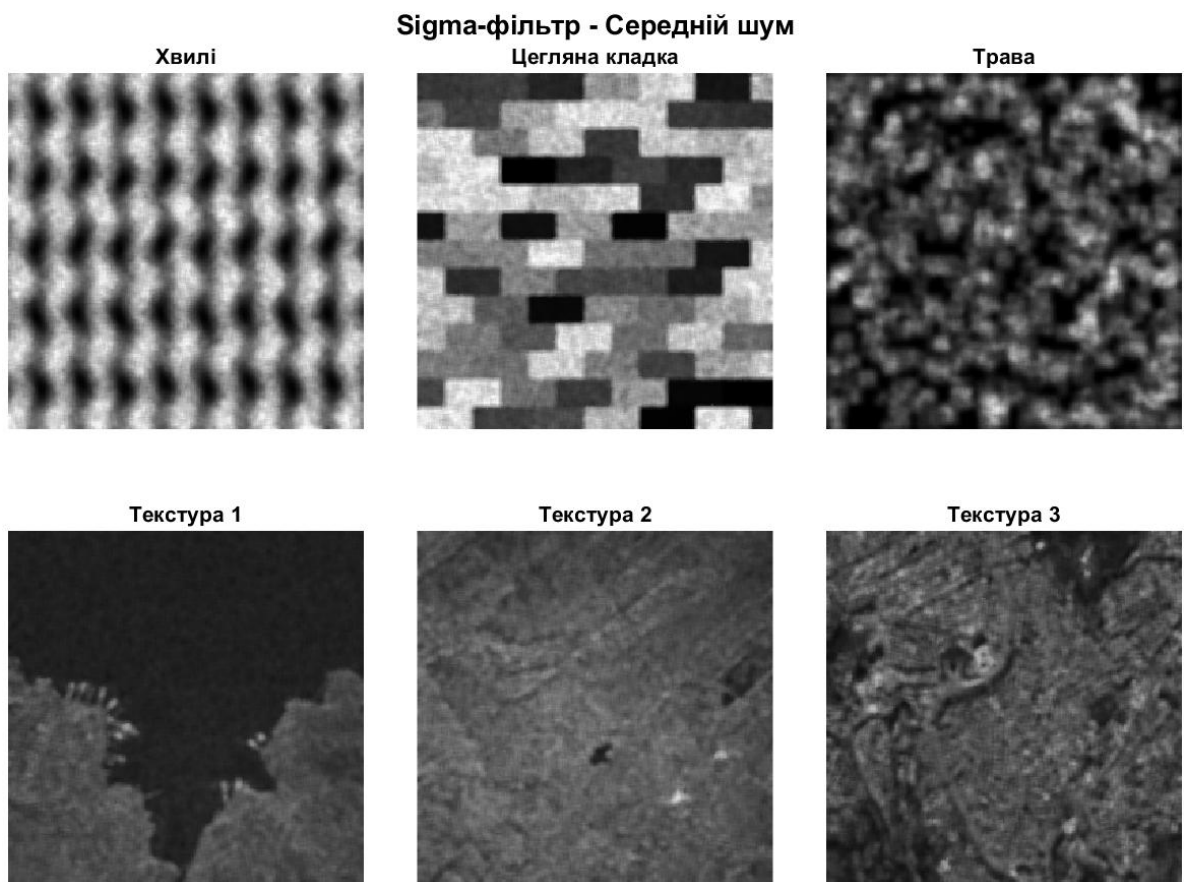


Рисунок 5.13 – Результати фільтрації Sigma-фільтром спекл-шуму дисперсією 0.1

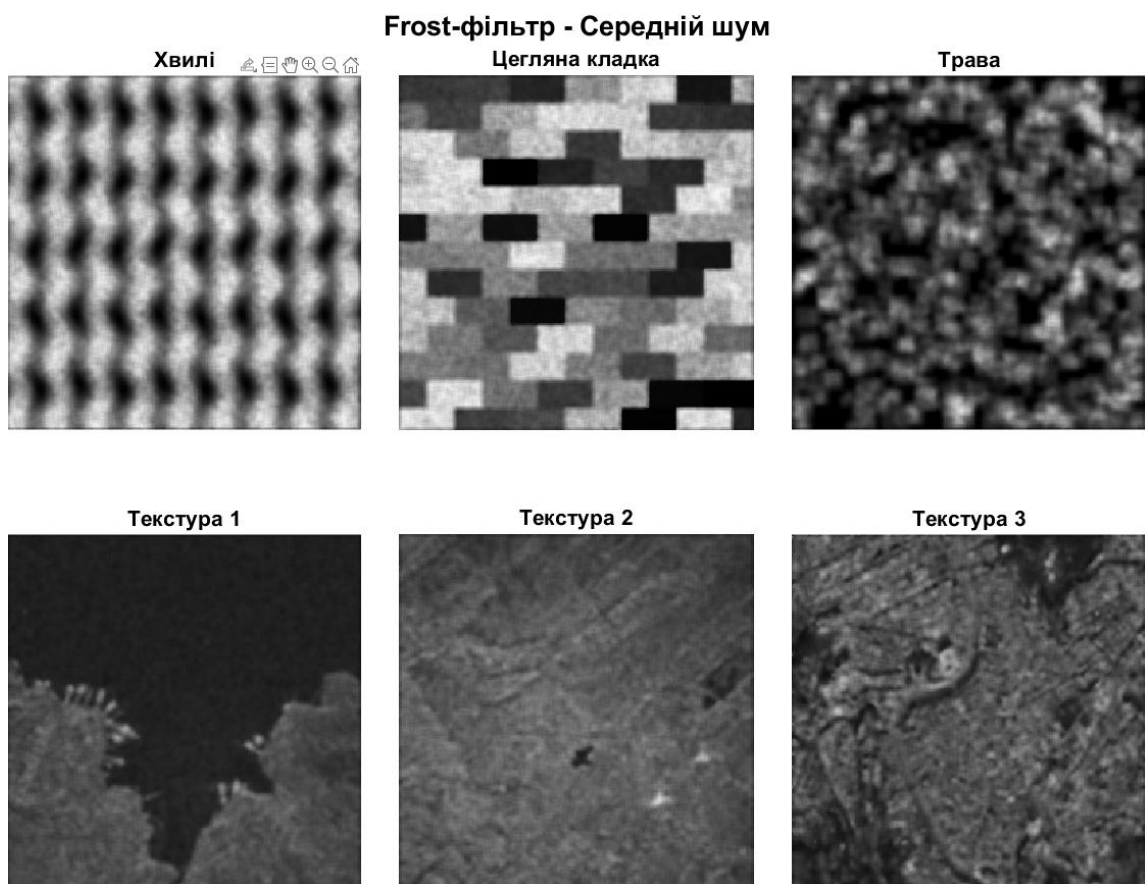


Рисунок 5.14 – Результати фільтрації Frost-фільтром спекл-шуму дисперсією 0.1

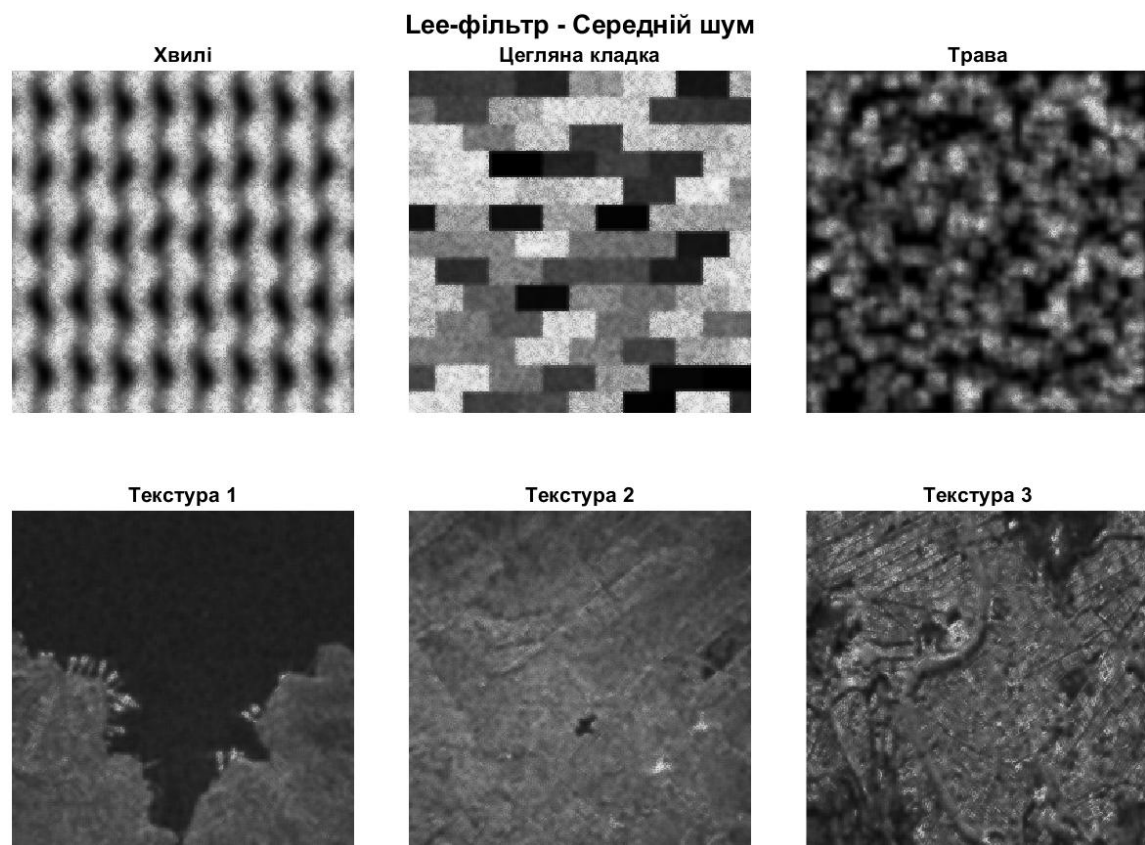


Рисунок 5.15 – Результати фільтрації Lee-фільтром спекл-шуму дисперсією 0.1

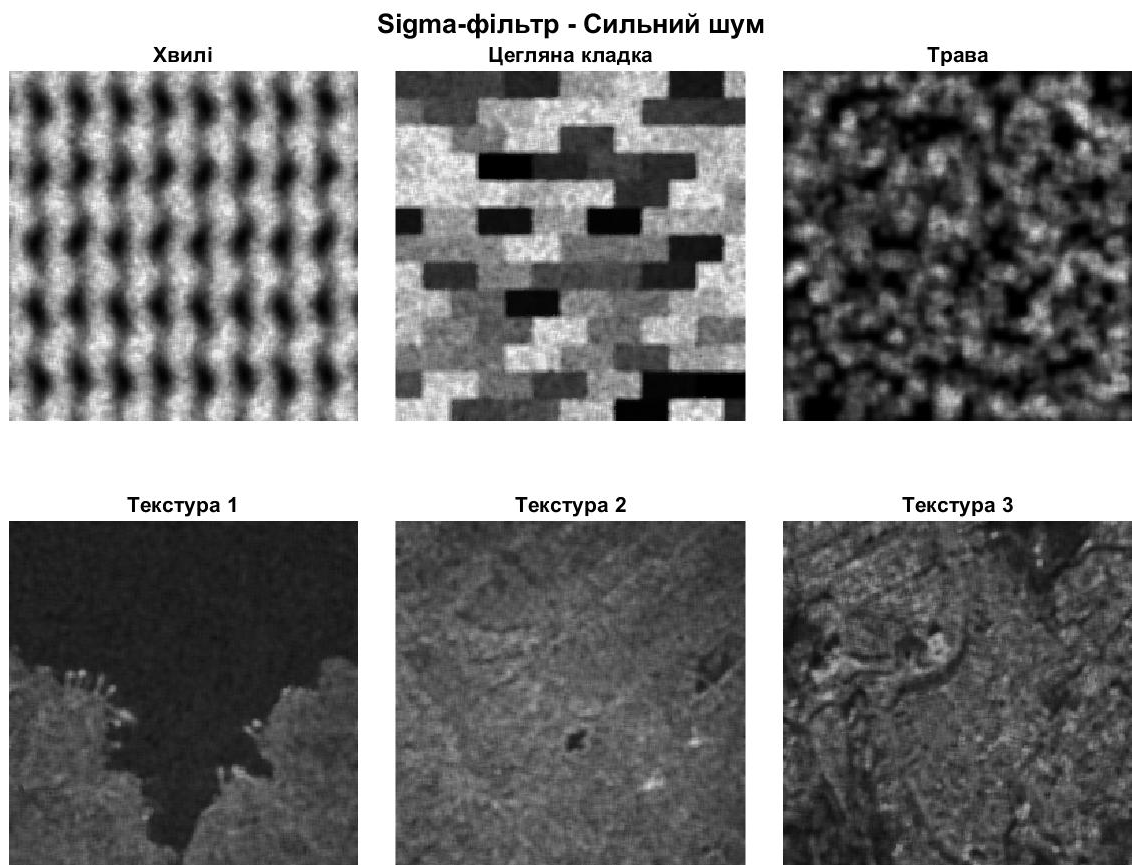


Рисунок 5.16 – Результати фільтрації Sigma-фільтром спекл-шуму дисперсією 0.2

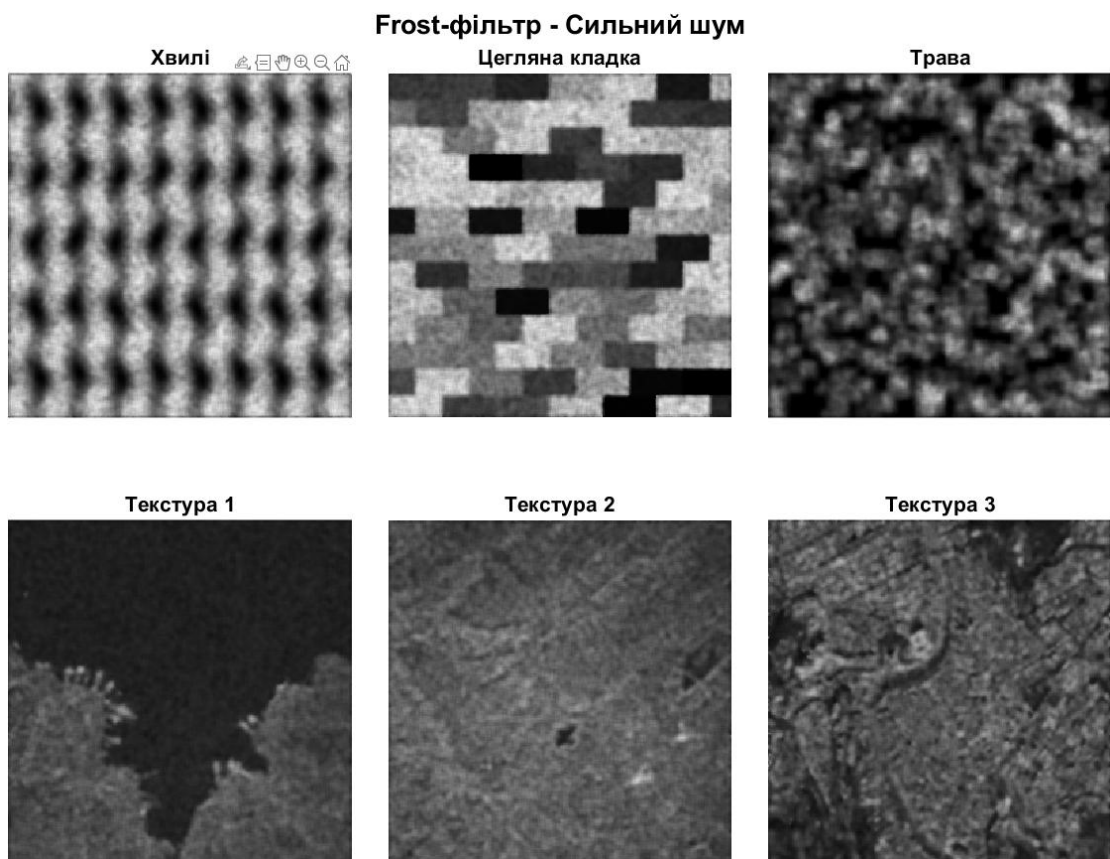


Рисунок 5.17 – Результати фільтрації Frost-фільтром спекл-шуму дисперсією 0.2

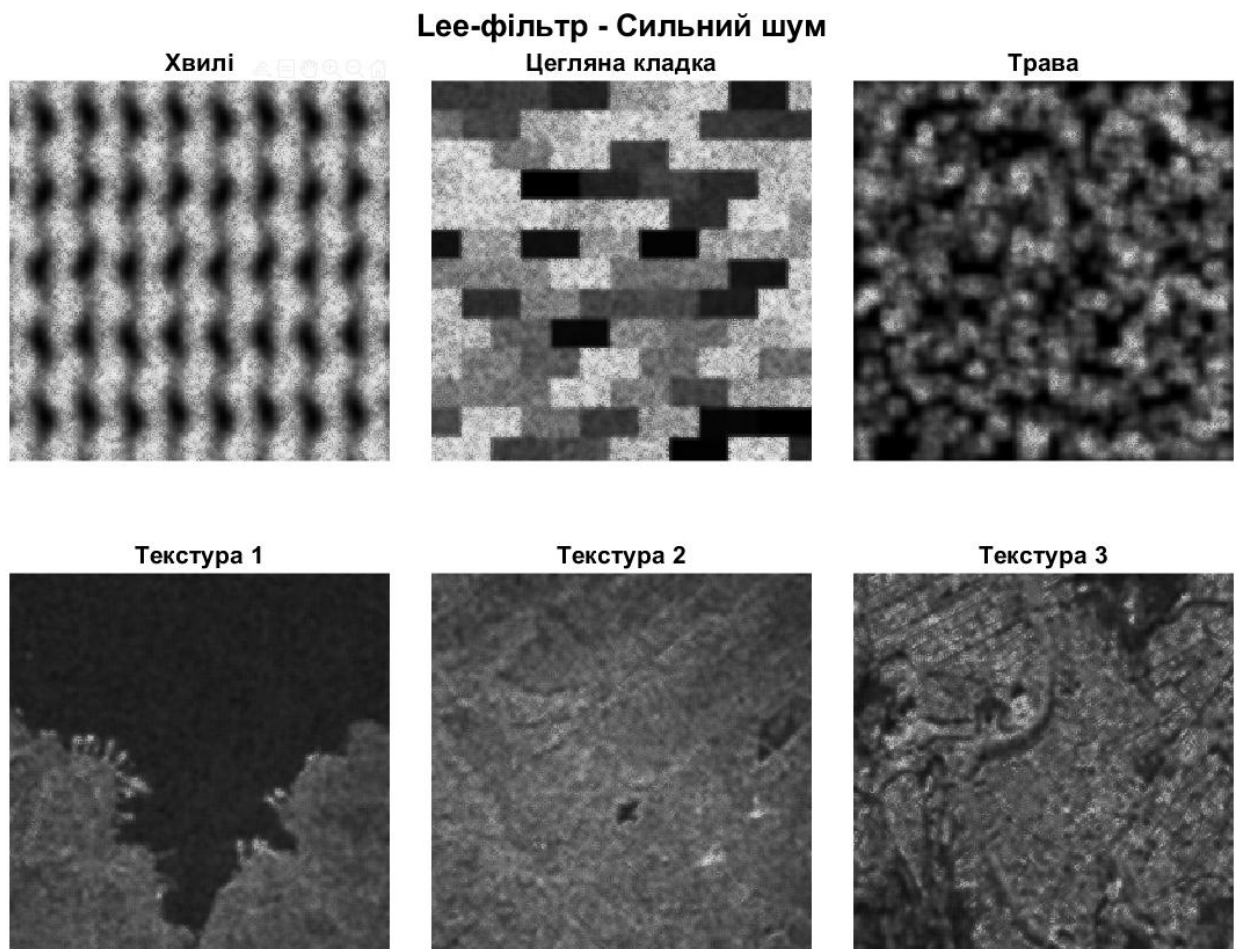


Рисунок 5.18 – Результати фільтрації Lee-фільтром спекл-шуму дисперсією 0.2

Візуально видно, що Frost-фільтр краще зберігає деталі для синтетичних текстур із гладкими або випадковими структурами, тоді як Lee-фільтр виявляє переваги для реальних текстур із природними неоднорідностями.

Кількісні результати наведено в табл. 5.1. Для текстури "Хвилі" за умов малого шуму найвищий PSNR досяг Sigma-фільтр (28.88 дБ), однак SSIM у нього лише на 0.001 вищий за Frost (0.929 проти 0.928), що свідчить про приблизно однакову ефективність з перевагою у згладжуванні. За середнього та сильного шуму Frost стабільно демонструє вищу SSIM, що вказує на кращу здатність зберігати структуру зображення.

Для "Цегляної кладки" Lee-фільтр продемонстрував найвищий PSNR при малому шумі (24.26 дБ), але Frost мав стабільно вищі SSIM для всіх рівнів шуму, що підтверджує його ефективність у збереженні геометрично чітких меж. У Sigma-фільтра зниження показників особливо помітне при сильному шумі.

У випадку текстури "Трава" Frost-фільтр домінує як за PSNR, так і за SSIM на всіх рівнях шуму, з максимумами 31.77 дБ і 0.936 відповідно. Це підкреслює його перевагу при обробці природних, хаотичних текстур з високою варіативністю.

Для "Текстури 1", яка є реальним зображенням, Lee-фільтр показав найкращі результати: PSNR до 30.97 дБ і SSIM до 0.836. Інші фільтри поступаються, особливо за показником структурної схожості, що свідчить про ефективну адаптацію Lee до реальних неоднорідностей.

У випадку "Текстури 2" жоден фільтр не забезпечив високих значень SSIM, але Lee знову переважає, особливо при малому шумі (SSIM = 0.626). Це може вказувати на складність структури цієї текстури для адаптивних методів згладжування.

Нарешті, для "Текстури 3" найвищі результати спостерігаються у Lee-фільтра (наприклад, PSNR = 21.47 дБ, SSIM = 0.659 при малому шумі). У Sigma та Frost фіксується суттєве просідання SSIM (до 0.27–0.38 при сильному шумі), що вказує на низьку здатність зберігати структуру при значному зашумленні.

Таблиця 5.1 - Значення PSNR і SSIM для різних текстур і фільтрів

Текстура	Рівень шуму	Sigma PSNR, Дб	Sigma SSIM	Frost PSNR, Дб	Frost SSIM	Lee PSNR, Дб	Lee SSIM
Хвилі	Малий шум	28,88	0,929	26,72	0,928	26,16	0,837
	Середній шум	25,63	0,874	24,51	0,884	23,56	0,767
	Сильний шум	22,69	0,802	21,91	0,82	20,97	0,686
Цегляна кладка	Малий шум	21,91	0,703	21,91	0,719	24,26	0,686
	Середній шум	20,95	0,634	20,88	0,665	21,71	0,597
	Сильний шум	19,39	0,554	19,29	0,6	19,21	0,511

Таблиця 5.1 (продовження)

Текстура	Рівень шуму	Sigma PSNR, Дб	Sigma SSIM	Frost PSNR, Дб	Frost SSIM	Lee PSNR, Дб	Lee SSIM
Трава	Малий шум	31,03	0,923	31,77	0,936	30,95	0,915
	Середній шум	29,77	0,904	30,79	0,923	29,48	0,891
	Сильний шум	27,94	0,869	28,99	0,898	27,54	0,852
Текстура 1	Малий шум	29,07	0,784	29,13	0,783	30,97	0,836
	Середній шум	28,61	0,756	28,92	0,767	29,94	0,8
	Сильний шум	27,84	0,711	28,47	0,738	28,82	0,753
Текстура 2	Малий шум	25,54	0,492	25,55	0,5	26,99	0,626
	Середній шум	25	0,462	25,25	0,485	26	0,565
	Сильний шум	24,24	0,418	24,72	0,465	24,89	0,501
Текстура 3	Малий шум	18,07	0,284	18,59	0,35	21,47	0,659
	Середній шум	17,96	0,279	18,61	0,362	20,19	0,558
	Сильний шум	17,73	0,271	18,64	0,386	19,31	0,479

5.4 Висновки до п'ятого розділу

У п'ятому розділі було проведено комплексне моделювання процесу фільтрації спекл-шуму на текстурних зображеннях із використанням трьох різних фільтрів: Sigma, Frost та Lee. Було створено набір як синтетичних, так і реальних текстур розміром 256×256 пікселів, на які додавалися спекл-шум різної інтенсивності (дисперсія 0.05, 0.1 та 0.2).

Оцінка ефективності фільтрації здійснювалась за допомогою метрик PSNR та SSIM, що дозволяє комплексно характеризувати як загальний рівень пригнічення шуму, так і збереження структурної цілісності зображення.

Отримані результати показали, що:

Frost-фільтр є найефективнішим для текстур із природними або випадковими структурами, такими як "трава" або "хвилі". Він забезпечує найвищі або близькі до найвищих значення SSIM, особливо при середньому та сильному рівнях шуму.

Lee-фільтр демонструє перевагу при обробці реальних текстурних зображень, зокрема "текстура 1", "текстура 2" і "текстура 3", де він стабільно забезпечує найвищі значення як PSNR, так і SSIM. Це свідчить про його здатність адаптуватися до складної природної неоднорідності.

Sigma-фільтр, хоча і демонструє найвищі значення PSNR на деяких синтетичних текстурах (наприклад, "хвилі"), поступається за SSIM при високих рівнях шуму, що вказує на його обмежену ефективність у збереженні дрібних деталей.

Загалом, ефективність застосування відповідного фільтра в рамках реалізації алгоритму фільтрації залежить від типу текстури та рівня зашумлення. Найкращі результати досягаються при узгодженні характеристик фільтра з типом текстури: адаптивні методи (Frost і Lee) демонструють вищу універсальність, тоді як Sigma-фільтр більш придатний для задач із низьким рівнем шуму та простішими текстурами. Це дозволяє сформулювати рекомендації щодо вибору фільтра залежно від специфіки вхідного зображення.

ВИСНОВКИ

У рамках дипломної роботи виконано дослідження алгоритму придушення спекл-шуму на текстурних зображеннях, що є важливим для підвищення якості візуальної інформації у сферах медицини, дистанційного зондування, технічного зору та комп'ютерного моделювання. Проведено огляд прикладних задач, у яких виникає необхідність обробки цифрових зображень, розглянуто технології їх отримання та основні типи завад, включаючи гаусівський, дробовий, імпульсний та спекл-шум.

У теоретичній частині проаналізовано існуючі методи придушення шумів, серед яких для моделювання обрано адаптивні фільтри Sigma, Frost і Lee, що враховують локальні статистичні характеристики та є придатними для обробки текстурних структур. Математичний опис моделей текстур, таких як марковські випадкові поля, фрактальні та авторегресивні моделі, забезпечив теоретичну основу для розуміння структури зображень і оцінки результатів фільтрації.

Моделювання проведено в середовищі MATLAB для шести текстур — трьох синтетичних ("хвилі", "цегляна кладка", "трава") та трьох реальних ("текстура 1", "текстура 2", "текстура 3"), на які було накладено спекл-шум з різною дисперсією (0.05, 0.1, 0.2). Аналіз результатів за метриками PSNR і SSIM дозволив виявити закономірності між типом текстури, рівнем зашумлення та ефективністю фільтра. Frost-фільтр виявився найкращим для синтетичних текстур із випадковими або плавними візерунками ("трава", "хвилі"), тоді як Lee-фільтр забезпечив вищу якість для реальних, природно неоднорідних текстур, зокрема при сильному шумі. Sigma-фільтр показав себе ефективним на простих періодичних структурах, але менш стійким до високих рівнів завад.

Візуальний аналіз відфільтрованих зображень підтвердив кількісні висновки, продемонструвавши кращу деталізацію та збереження структурних особливостей зображень за умови відповідного підбору фільтра. Результати підкреслюють важливість адаптивного підходу до вибору алгоритму залежно від типу текстури й умов зашумлення.

Результати дослідження підкреслюють залежність вибору фільтру від типу текстури та створюють основу для подальшого вдосконалення методів обробки зображень шляхом комбінування алгоритмів і тестування різних параметрів шуму. Реалізований алгоритм може бути застосований для підвищення якості зображень у реальних задачах, в яких виникає необхідність виконувати обробку цифрових текстурних зображень, які спотворені дією спекл-шуму. З метою підвищення відношення сигнал/шум в цілях поліпшення як їх візуального аналізу так і їх подальшої обробки автоматичними алгоритмами, які можуть використовуватися на наступних стадіях обробки, наприклад, при реалізації задач класифікації, сегментації, розпізнавання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Сіненко О. О. Що таке УЗД? Лікар ультразвукової діагностики [Електронний ресурс]. – Режим доступу: <https://daily-med.com.ua/uk/blog-uk/shho-take-uzd-likar-ultrazvukovo%D1%97-diagnostiki-sinenka-o-o> – Дата звернення: 14.04.2025.
2. Медична візуалізація: лекційні матеріали [Електронний ресурс]. Кафедра біомедичної інженерії та оптико-електронних систем ВНТУ. Режим доступу: https://bmi.vntu.edu.ua/bioart/program/OZ_lect.pdf – Дата звернення: 14.04.2025.
3. LibreTexts. Медична візуалізація. Українська версія LibreTexts. URL: [https://ukrayinska.libretexts.org/Медицина/Анатомія_і_фізіологія/Книга:_Анатомія_та_фізіологія_\(OpenStax\)/Unit_1:_Рівні_організації/01:_Введення_в_людський_організм/1.07:_Медична_візуалізація](https://ukrayinska.libretexts.org/Медицина/Анатомія_і_фізіологія/Книга:_Анатомія_та_фізіологія_(OpenStax)/Unit_1:_Рівні_організації/01:_Введення_в_людський_організм/1.07:_Медична_візуалізація) – Дата звернення: 15.04.2025
4. Шевченко К. Комп'ютерна томографія голови: що може показати і як правильно підготуватися [Електронний ресурс]. ТСН.ua. Режим доступу: <https://tsn.ua/lady/zdorovye/aktualnaya-tema/komp-yuterna-tomografiya-golovi-scho-mozhe-pokazati-i-yak-pravilno-pidgotuvatisya-1712278.html> – Дата звернення: 15.04.2025
5. Магнітно-резонансна томографія [Електронний ресурс]. Вікіпедія. Режим доступу: https://uk.wikipedia.org/wiki/Магнітно-резонансна_томографія – Дата звернення: 16.04.2025
6. Дрони і супутники: моніторинг стану посівів впродовж сезону [Електронний ресурс]. SmartFarming. Режим доступу: <https://www.smartfarming.ua/drony-i-suputnyky-monitorynh-stanu-posiviv-vprodovzh-sezonu/> – Дата звернення: 16.04.2025
7. Система розпізнавання обличчя: правові аспекти використання в Україні та в ЄС [Електронний ресурс]. Українська Гельсінська спілка з прав людини.

- Режим доступу: <https://www.helsinki.org.ua/articles/systema-rozpiznavannia-oblychchia-pravovi-aspekty-vykorystannia-v-ukraini-ta-v-yes/> – Дата звернення: 17.04.2025
8. Налаштування розпізнавання осіб на обладнанні Dahua: керівництво користувача [Електронний ресурс]. – Режим доступу: <https://agent-007.com.ua/ua/articles/nastrojka-raspoznavaniya-lic-na-oborudovanii-dahua-rukovodstvo-polzovatelya> – Дата звернення: 17.04.2025
9. Автоматичне розпізнавання автономерів з відеоаналітики [Електронний ресурс]. Verna. Режим доступу: <https://www.verna.ua/videoanalytics/license-plate-recognition> – Дата звернення: 18.04.2025
- 10.Відновлення зображення штучним інтелектом [Електронний ресурс] // Вікіпедія. – Режим доступу: https://uk.wikipedia.org/wiki/Відновлення_зображення_штучним_інтелектом – Дата звернення: 18.04.2025
- 11.Denoising Results of the Monar Image [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/figure/Denoising-results-of-the-Monar-image-in-Set12-with-noise-level-25_fig3_369837488 – Дата звернення: 20.04.2025
- 12.Lei, B., Zeng, Y., Luo, M. et al. Hybrid Transformer for Low-Light Image Enhancement [Електронний ресурс] // IET Electronics Letters. – Режим доступу: <https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/ell2.13180> – Дата звернення: 20.04.2025
- 13.Як в сучасному авто працюють датчики, радары, лідары та камери [Електронний ресурс]. Auto24. Режим доступу: https://auto.24tv.ua/yak_v_suchasnomu_avto_pratsiuiut_datchyky_radary_lidary_ta_kamery_n57969 – Дата звернення: 22.04.2025
- 14.Gonzalez R. C., Woods R. E. *Digital Image Processing*. – 4th ed. – New York:

Pearson, 2018. – 1168 p.

15. Принцип дії цифрової фотокамери [Електронний ресурс]. Дніпропетровський центр професійно-технічної освіти державної служби зайнятості. Режим доступу: https://cpto.dp.ua/public_html/posibnyky/posibnyk_ooi/Digital_photo.html – Дата звернення: 24.04.2025
16. Murtaza, G., Javed, A. R., & Rizwan, M. (2017). Vision-Based Intelligent Wild Animal Detection and Alert System for Agricultural Fields. International Journal of Systems Assurance Engineering and Management [Електронний ресурс]. – Режим доступу: <https://link.springer.com/article/10.1007/s13534-017-0030-7> – Дата звернення: 24.04.2025
17. Ультразвукова візуалізація внутрішньоматкових засобів контрацепції [Електронний ресурс] // RH.ua. – Режим доступу: <https://rh.ua/statti/ultrazvukova-vizualizaciya-vnutrishnomatkovix-zasobiv-kontracserciyi/> – Дата звернення: 25.04.2025
18. Transforming Energy into Imagery: How Satellite Data Becomes Stunning Views of Earth [Електронний ресурс]. National Environmental Satellite, Data, and Information Service. Режим доступу: <https://www.nesdis.noaa.gov/news/transforming-energy-imagery-how-satellite-data-becomes-stunning-views-of-earth> – Дата звернення: 26.04.2025
19. Як працює тепловізор? [Електронний ресурс]. Hikvision. Режим доступу: <https://hikvision.co.ua/ua/kak-rabotaet-teplovizor/> – Дата звернення: 28.04.2025
20. Машинний зір [Електронний ресурс]. Вікіпедія. Режим доступу: https://uk.wikipedia.org/wiki/Машинний_зір – Дата звернення: 29.04.2025
21. Vision System [Електронний ресурс]. – Режим доступу: <https://www.indiamart.com/proddetail/vision-system-17873332933.html> – Дата звернення: 01.05.2025

- 22.Артефакти стиснення [Електронний ресурс]. Вікіпедія. Режим доступу: https://uk.wikipedia.org/wiki/Артефакти_стиснення – Дата звернення: 02.05.2025
- 23.Wildlife Motion Blur [Електронний ресурс]. – Режим доступу: <https://www.wildlife-pictures-online.com/wildlife-motion-blur.html> – Дата звернення: 04.05.2025
- 24.Scientific Bulletin of UNFU [Електронний ресурс]. Режим доступу: https://nv.nltu.edu.ua/Archive/2018/28_1/31.pdf – Дата звернення: 05.05.2025
- 25.Loupas T., McDicken W. N., Allan P. L. An adaptive weighted median filter for speckle suppression in medical ultrasonic images // *IEEE Transactions on Circuits and Systems*. – 1989. – Vol. 36, No. 1. – P. 129–135.
- 26.Lee J.-S., Jurkevich I., Dewaele P., Wambacq P., Oosterlinck A. Speckle filtering of synthetic aperture radar images: a review // *Remote Sensing Reviews*. – 1994. – Vol. 13, No. 3–4. – P. 245–276. – DOI: 10.1080/02757259409532206.
- 27.Лазебний В. С., Попович П. В. Цифрова обробка зображень: методичні рекомендації до виконання лабораторних робіт для студентів спеціальності 7.05080302, 8.05080302 «Аудіо-, відео- та кінотехніка» усіх форм навчання [Електронний ресурс]. – Київ: НТУУ «КПІ», 2016. – 73 с. – Режим доступу: <https://ela.kpi.ua/server/api/core/bitstreams/2301c9b9-f626-4c13-801c-5b700f77a8f3/content>
- 28.Image texture [Електронний ресурс] // Wikipedia – The Free Encyclopedia. – Режим доступу: https://en.wikipedia.org/wiki/Image_texture – Дата звернення: 10.05.2025
- 29.Tuceryan M., Jain A. K. Texture Analysis // *Handbook of Pattern Recognition and Computer Vision*. – 2nd ed. – Eds. C. H. Chen, L. F. Pau, P. S. P. Wang. – Singapore: World Scientific Publishing, 1998. – P. 207–248.
- 30.Markov random field [Електронний ресурс] // Wikipedia. – Режим доступу:

https://en.wikipedia.org/wiki/Markov_random_field – Дата звернення:
18.05.2025

31.Chandrasekaran K. 2D-Discrete Wavelet Transformation and its applications in Digital Image Processing using MATLAB [Електронний ресурс]. Medium. Режим доступу: <https://medium.com/@koushikc2000/2d-discrete-wavelet-transformation-and-its-applications-in-digital-image-processing-using-matlab-1f5c68672de3> – Дата звернення: 24.05.2025

32.Фільтр Калмана [Електронний ресурс] // Вікіпедія. – Вільна енциклопедія. – Режим доступу: https://uk.wikipedia.org/wiki/Фільтр_Калмана – Дата звернення: 01.06.2025

33.PSNR (Peak Signal-to-Noise Ratio) [Електронний ресурс] // MathWorks. – Режим доступу: <https://www.mathworks.com/help/vision/ref/psnr.html> – Дата звернення: 05.06.2025

34.SSIM (Structural Similarity Index) [Електронний ресурс] // MathWorks. – Режим доступу: <https://www.mathworks.com/help/images/ref/ssim.html> – Дата звернення: 07.06.2025

ДОДАТОК А. КОД ОСНОВНОЇ ПРОГРАМИ

```
% Розмір зображень
sizeImg = 256;

% Параметри шуму
noise_low = 0.05;      % Малий шум
noise_medium = 0.1;    % Середній шум
noise_high = 0.2;      % Сильний шум

% Параметри фільтрів
windowSize = [5 5];
sigma = 2;             % Для Sigma-фільтру
Damp_fact = 1;         % Для Frost-фільтру

% 1. Генерація синтезованих текстур
[X, Y] = meshgrid(1:sizeImg, 1:sizeImg);
I_wave = 0.5 + 0.3 * sin(2 * pi * X / 30 + 0.5 * sin(2 * pi * Y / 40)) + ...
        0.2 * cos(2 * pi * Y / 50 + 0.3 * cos(2 * pi * X / 60));
I_wave = mat2gray(I_wave);

I_brick = zeros(sizeImg, sizeImg);
brickHeight = 20; brickWidth = 40;
for i = 1:brickHeight:sizeImg
    offset = mod(floor(i/brickHeight), 2) * brickWidth/2;
    for j = 1:brickWidth:sizeImg
        xStart = max(1, j - offset);
        xEnd = min(sizeImg, j + brickWidth - 1 - offset);
        yStart = i;
        yEnd = min(sizeImg, i + brickHeight - 1);
        I_brick(yStart:yEnd, xStart:xEnd) = 0.7 + 0.2 * rand;
        if yEnd < sizeImg
            I_brick(yEnd+1:min(yEnd+2, sizeImg), xStart:xEnd) = 0.2;
        end
        if xEnd < sizeImg
            I_brick(yStart:yEnd, xEnd+1:min(xEnd+2, sizeImg)) = 0.2;
        end
    end
end
I_brick = mat2gray(I_brick);

I_grass = zeros(sizeImg, sizeImg);
numPatches = 1000;
for p = 1:numPatches
    x = randi([1 sizeImg]);
    y = randi([1 sizeImg]);
    intensity = 0.5 + 0.3 * rand;
    patchSize = randi([3 7]);
    xStart = max(1, x - patchSize);
    xEnd = min(sizeImg, x + patchSize);
    yStart = max(1, y - patchSize);
    yEnd = min(sizeImg, y + patchSize);
    I_grass(yStart:yEnd, xStart:xEnd) = I_grass(yStart:yEnd, xStart:xEnd) + intensity;
end
I_grass = imfilter(I_grass, fspecial('gaussian', [5 5], 1.5));
I_grass = mat2gray(I_grass);

% 2. Завантаження та обробка реальних текстур
% Завантаження реальних зображень
if exist('texture1.jpg', 'file')
    texture1 = imread('texture1.jpg');
```

```

else
    error('Файл texture1.jpg не знайдено');
end

if exist('texture2.jpg', 'file')
    texture2 = imread('texture2.jpg');
else
    error('Файл texture2.jpg не знайдено');
end

if exist('texture3.jpg', 'file')
    texture3 = imread('texture3.jpg');
else
    error('Файл texture3.jpg не знайдено');
end

% Перетворення у сіру шкалу, якщо зображення кольорове
if size(texture1, 3) == 3
    texture1 = rgb2gray(texture1);
end
if size(texture2, 3) == 3
    texture2 = rgb2gray(texture2);
end
if size(texture3, 3) == 3
    texture3 = rgb2gray(texture3);
end

% Нормалізація до діапазону [0,1] та зміна розміру до 256x256
texture1 = im2double(imresize(texture1, [sizeImg sizeImg]));
texture2 = im2double(imresize(texture2, [sizeImg sizeImg]));
texture3 = im2double(imresize(texture3, [sizeImg sizeImg]));

% Створення масивів для зберігання всіх текстур
all_textures = {I_wave, I_brick, I_grass, texture1, texture2, texture3};
texture_names = {'Хвилі', 'Цегляна кладка', 'Трава', 'Текстура 1', 'Текстура 2', 'Текстура 3'};
noise_levels = [noise_low, noise_medium, noise_high];
noise_names = {'Малий шум', 'Середній шум', 'Сильний шум'};

% Ініціалізація структур для зберігання результатів
results = struct();
metrics = struct();

% 3. Обробка кожної текстури з різними рівнями шуму
for tex_idx = 1:length(all_textures)
    original_img = all_textures{tex_idx};
    tex_name = texture_names{tex_idx};

    fprintf('Обробка текстури: %s\n', tex_name);

    for noise_idx = 1:length(noise_levels)
        noise_var = noise_levels(noise_idx);
        noise_name = noise_names{noise_idx};

        % Додавання шуму
        noisy_img = imnoise(original_img, 'speckle', noise_var);

        % Застосування фільтрів
        filtered_sigma = sigmaFilter(noisy_img, windowSize(1), sigma);
        filtered_frost = Frost_filter(noisy_img, Damp_fact, windowSize);
        filtered_lee = leeFilter(noisy_img, windowSize(1), noise_var);
    end
end

```

```

% Збереження результатів
results.(sprintf('tex%d_noise%d', tex_idx, noise_idx)) = struct(...
    'original', original_img, ...
    'noisy', noisy_img, ...
    'sigma', filtered_sigma, ...
    'frost', filtered_frost, ...
    'lee', filtered_lee ...
);

% Обчислення метрик
psnr_sigma = psnr(filtered_sigma, original_img);
ssim_sigma = ssim(filtered_sigma, original_img);
psnr_frost = psnr(filtered_frost, original_img);
ssim_frost = ssim(filtered_frost, original_img);
psnr_lee = psnr(filtered_lee, original_img);
ssim_lee = ssim(filtered_lee, original_img);

% Збереження метрик
metrics.(sprintf('tex%d_noise%d', tex_idx, noise_idx)) = struct(...
    'psnr_sigma', psnr_sigma, 'ssim_sigma', ssim_sigma, ...
    'psnr_frost', psnr_frost, 'ssim_frost', ssim_frost, ...
    'psnr_lee', psnr_lee, 'ssim_lee', ssim_lee ...
);
end
end

% 4. Візуалізація результатів з максимальним розміром
% Показати оригінальні текстури
figure('Position', [50, 50, 1800, 1000]);
set(gcf, 'Color', 'white');
for i = 1:length(all_textures)
    subplot(2, 3, i);
    imshow(all_textures{i});
    title(texture_names{i}, 'FontSize', 16, 'FontWeight', 'bold');
    set(gca, 'Position', get(gca, 'Position') .* [1 1 1.1 1.1]); % Збільшити розмір subplot
end
sgtitle('Оригінальні текстури', 'FontSize', 20, 'FontWeight', 'bold');
% Мінімізувати відступи
set(gcf, 'PaperPositionMode', 'auto');
tight_layout();

% Показати результати для кожного рівня шуму
for noise_idx = 1:length(noise_levels)
    noise_name = noise_names{noise_idx};
    noise_var = noise_levels(noise_idx);

    % Зашумлені зображення
    figure('Position', [50, 50, 1800, 1000]);
    set(gcf, 'Color', 'white');
    for tex_idx = 1:6
        field_name = sprintf('tex%d_noise%d', tex_idx, noise_idx);
        subplot(2, 3, tex_idx);
        imshow(results.(field_name).noisy);
        title(texture_names{tex_idx}, 'FontSize', 16, 'FontWeight', 'bold');
        set(gca, 'Position', get(gca, 'Position') .* [1 1 1.1 1.1]);
    end
    sgtitle(sprintf('Зашумлені зображення - %s', noise_name), 'FontSize', 20,
'FontWeight', 'bold');
    tight_layout();
end

```

```

% Sigma-фільтр
figure('Position', [50, 50, 1800, 1000]);
set(gcf, 'Color', 'white');
for tex_idx = 1:6
    field_name = sprintf('tex%d_noise%d', tex_idx, noise_idx);
    subplot(2, 3, tex_idx);
    imshow(results.(field_name).sigma);
    title(texture_names{tex_idx}, 'FontSize', 16, 'FontWeight', 'bold');
    set(gca, 'Position', get(gca, 'Position') .* [1 1 1.1 1.1]);
end
sgtitle(sprintf('Sigma-фільтр - %s', noise_name), 'FontSize', 20, 'FontWeight',
'bold');
tight_layout();

% Frost-фільтр
figure('Position', [50, 50, 1800, 1000]);
set(gcf, 'Color', 'white');
for tex_idx = 1:6
    field_name = sprintf('tex%d_noise%d', tex_idx, noise_idx);
    subplot(2, 3, tex_idx);
    imshow(results.(field_name).frost);
    title(texture_names{tex_idx}, 'FontSize', 16, 'FontWeight', 'bold');
    set(gca, 'Position', get(gca, 'Position') .* [1 1 1.1 1.1]);
end
sgtitle(sprintf('Frost-фільтр - %s', noise_name), 'FontSize', 20, 'FontWeight',
'bold');
tight_layout();

% Lee-фільтр
figure('Position', [50, 50, 1800, 1000]);
set(gcf, 'Color', 'white');
for tex_idx = 1:6
    field_name = sprintf('tex%d_noise%d', tex_idx, noise_idx);
    subplot(2, 3, tex_idx);
    imshow(results.(field_name).lee);
    title(texture_names{tex_idx}, 'FontSize', 16, 'FontWeight', 'bold');
    set(gca, 'Position', get(gca, 'Position') .* [1 1 1.1 1.1]);
end
sgtitle(sprintf('Lee-фільтр - %s', noise_name), 'FontSize', 20, 'FontWeight', 'bold');
tight_layout();
end

% 5. Виведення метрик
fprintf('\n=== РЕЗУЛЬТАТИ ФІЛЬТРАЦІЇ ===\n');
for tex_idx = 1:length(texture_names)
    fprintf('\n--- %s ---\n', texture_names{tex_idx});

    for noise_idx = 1:length(noise_levels)
        field_name = sprintf('tex%d_noise%d', tex_idx, noise_idx);
        m = metrics.(field_name);

        fprintf('\n%s ( $\sigma^2$ =%.3f):\n', noise_names{noise_idx}, noise_levels(noise_idx));
        fprintf('  Sigma Filter: PSNR = %.2f dB, SSIM = %.3f\n', m.psnr_sigma,
m.ssim_sigma);
        fprintf('  Frost Filter: PSNR = %.2f dB, SSIM = %.3f\n', m.psnr_frost,
m.ssim_frost);
        fprintf('  Lee Filter: PSNR = %.2f dB, SSIM = %.3f\n', m.psnr_lee, m.ssim_lee);
    end
end
end

```

```

% 6. Повна порівняльна таблиця результатів
fprintf('\n=== ПОВНА ПОРІВНЯЛЬНА ТАБЛИЦЯ РЕЗУЛЬТАТІВ ===\n');
fprintf('%-15s | %-12s | %-25s | %-25s | %-25s\n', 'Текстура', 'Рівень шуму', 'Sigma Filter', 'Frost Filter', 'Lee Filter');
fprintf('%-15s | %-12s | %-25s | %-25s | %-25s\n', '', '', 'PSNR / SSIM', 'PSNR / SSIM', 'PSNR / SSIM');
fprintf('%s\n', repmat('-', 1, 110));

for tex_idx = 1:length(texture_names)
    for noise_idx = 1:length(noise_levels)
        field_name = sprintf('tex%d_noise%d', tex_idx, noise_idx);
        m = metrics.(field_name);

        if noise_idx == 1
            tex_name_display = texture_names{tex_idx};
        else
            tex_name_display = '';
        end

        fprintf('%-15s | %-12s | %6.2f / %5.3f | %6.2f / %5.3f | %6.2f / %5.3f\n', ...
            tex_name_display, noise_names{noise_idx}, ...
            m.psnr_sigma, m.ssim_sigma, ...
            m.psnr_frost, m.ssim_frost, ...
            m.psnr_lee, m.ssim_lee);
    end
    fprintf('%s\n', repmat('-', 1, 110));
end

% Функція для мінімізації відступів між subplot'ами
function tight_layout()
    % Отримання всіх subplot'ів на поточній фігурі
    children = get(gcf, 'Children');
    axes_handles = findobj(children, 'Type', 'axes');

    if length(axes_handles) == 6 % 2x3 layout
        % Налаштування позицій для максимального використання простору
        positions = [
            0.05, 0.55, 0.28, 0.35; % subplot(2,3,1)
            0.36, 0.55, 0.28, 0.35; % subplot(2,3,2)
            0.67, 0.55, 0.28, 0.35; % subplot(2,3,3)
            0.05, 0.1, 0.28, 0.35; % subplot(2,3,4)
            0.36, 0.1, 0.28, 0.35; % subplot(2,3,5)
            0.67, 0.1, 0.28, 0.35 % subplot(2,3,6)
        ];

        % Застосування позицій
        for i = 1:length(axes_handles)
            set(axes_handles(i), 'Position', positions(length(axes_handles)-i+1, :));
        end
    end
end

% Функції

% Sigma Filter
function J = sigmaFilter(I, windowSize, sigma)
    [rows, cols] = size(I);
    J = zeros(rows, cols);
    half = floor(windowSize/2);
    I_padded = padarray(I, [half half], 'replicate');

```

```

for i = 1+half:rows+half
    for j = 1+half:cols+half
        window = I_padded(i-half:i+half, j-half:j+half);
        meanVal = mean(window(:));
        stdVal = std(window(:), 1);
        if stdVal == 0
            J(i-half, j-half) = meanVal;
        else
            validPixels = abs(window - meanVal) <= sigma * stdVal;
            if sum(validPixels(:)) == 0
                J(i-half, j-half) = meanVal;
            else
                J(i-half, j-half) = mean(window(validPixels));
            end
        end
    end
end
end

% Frost Filter
function ima_fi = Frost_filter(ima_nse, Damp_fact, sz)
    ima_nse = double(ima_nse);
    ima_fi = zeros(size(ima_nse));
    if mod(sz,2) ~= 1
        sz = sz + ~mod(sz,2);
    end
    mn = round((sz-1)/2);
    EImg = padarray(ima_nse, mn);

    % Обчислення матриці відстаней
    [x,y] = meshgrid(-mn(1,1):mn(1,1), -mn(1,2):mn(1,2));
    S = sqrt(x.^2 + y.^2);

    for i = 1:size(ima_nse,1)
        for j = 1:size(ima_nse,2)
            K = EImg(i:i+sz(1,1)-1, j:j+sz(1,2)-1);
            meanV = mean(K(:));
            varV = var(K(:), 1);

            if meanV == 0
                meanV = eps;
            end
            B = Damp_fact * (varV / (meanV * meanV));
            Weigh = exp(-S .* B);

            if any(isnan(Weigh(:)))
                ima_fi(i,j) = mean(K(:));
            else
                sumWeigh = sum(Weigh(:));
                if sumWeigh == 0
                    ima_fi(i,j) = mean(K(:));
                else
                    ima_fi(i,j) = sum(K(:) .* Weigh(:)) / sumWeigh;
                end
            end
        end
    end
end

% Lee Filter
function J = leeFilter(I, windowSize, noiseVariance)

```

```

[rows, cols] = size(I);
J = zeros(rows, cols);
half = floor(windowSize/2);
I_padded = padarray(I, [half half], 'replicate');
for i = 1+half:rows+half
    for j = 1+half:cols+half
        window = I_padded(i-half:i+half, j-half:j+half);
        meanVal = mean(window(:));
        varLocal = var(window(:), 1);

        if (noiseVariance + varLocal) == 0
            w = 0;
        else
            w = 1 - noiseVariance / (noiseVariance + varLocal);
        end
        J(i-half, j-half) = meanVal + w * (I_padded(i,j) - meanVal);
    end
end
end

```