

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук
(повна назва)
Кафедра _____ Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)

_____ Моделювання багатоагентної системи для фільтрації шуму в
_____ зображеннях на основі алгоритмів консенсусу
(тема)

Виконав:
здобувач _____ четвертого _____ року навчання,
групи _____ ІТШ-22-3

_____ Міхаель ЧУБУР
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна

Освітня програма _____ Штучний інтелект

(повна назва освітньої програми)

Керівник _____ ас. Дмитро МАЛЄСВ
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ

(підпис)

_____ Лариса ЧАЛА
(власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____

Освітня програма _____ Штучний інтелект _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_____» _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Чубуру Міхаелю _____
(прізвище, ім'я, по батькові)

1. Тема роботи Моделювання багатоагентної системи для фільтрації шуму в зображеннях на основі алгоритмів консенсусу

затверджена наказом університету від 5 червня 2026 р. № 488Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 10 липня 2026 р.

3. Вихідні дані до роботи Науково-технічні публікації та інтернет-джерела у галузі децентралізованої оптимізації та цифрової обробки сигналів, офіційна документація екосистеми Python (бібліотеки NumPy, SciPy, scikit-image, Matplotlib); тестові напівтонові зображення для проведення денойзінгу.

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі та постановка задачі

2) Теоретичні основи консенсусної взаємодії в MAS

3) Аналіз та проєктування математичної моделі зваженого Consensus ADMM MAS

4) Програмна реалізація та оцінка ефективності

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.06.2026	виконано
2	Аналіз предметної галузі	11.06.2026	виконано
3	Теоретичне дослідження моделей розподіленого консенсусу та варіаційної оптимізації.	13.06.2026	виконано
4	Проектування архітектури та алгоритмів децентралізованої багатоагентної системи.	16.06.2026	виконано
5	Програмна реалізація обчислювальних модулів та модифікацій алгоритму на мові Python.	19.06.2026	виконано
6	Проведення експериментальних досліджень	21.06.2026	виконано
7	Кількісний аналіз результатів, оцінка метрик якості та узагальнення висновків.	22.06.2026	виконано
8	Написання пояснювальної записки	23.06.2026	виконано
9	Перевірка на академічний плагіат	24.06.2026	виконано
10	Нормоконтроль	30.06.2026	виконано
11	Підготовка презентації та доповіді	01.07.2026	виконано
12	Попередній захист	03.07.2026	виконано
13	Рецензування	04.07.2026	виконано
14	Захист перед ЕК	10.07.2026	

Дата видачі завдання 8 червня 20 26 р.

Здобувач 
(підпис)

Керівник роботи _____
(підпис)

ас. Дмитро Малєєв
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 98 с., 17 рис., 2 дод., 35 джерел.

АДАПТИВНА РЕГУЛЯРИЗАЦІЯ, АЛГОРИТМ ШАМБОЛЛЯ-ПОКА, БАГАТОАГЕНТНА СИСТЕМА, ДЕНОЙЗИНГ, CONSENSUS ADMM, PYTHON, TOTAL VARIATION.

Об'єкт дослідження – процеси цифрової фільтрації та відновлення зображень, що зазнали впливу випадкових завад.

Предмет дослідження – моделі розподіленого консенсусу, першодвоїсті алгоритми сідлової оптимізації та багатоагентні методи денойзингу.

Мета роботи – розробка децентралізованої багатоагентної системи фільтрації на основі Consensus ADMM із просторово-адаптивними картами ваг для збереження контурів та геометрії кадрів.

Методи дослідження – варіаційна оптимізація, Марковські випадкові поля, методи розщеплення операторів, паралельні обчислення та ООП на мові Python.

У роботі досліджено еволюцію моделей денойзингу від MRF до апарату Consensus ADMM. Розроблено архітектуру паралельного розбиття зображення на патчі з перекриттям між автономними агентами. Інтеграція внутрішнього сідлового розв'язувача Шамболля-Пока з механізмом «теплого старту» мінімізувала обчислювальні витрати. Запропонована модифікація з динамічним розрахунком локальних ваг варіації ліквідувала ефект надмірного згладжування текстур. На основі примальних і дуальних залишків виконано аналіз збіжності, оцінено інженерні компроміси за метриками PSNR і SSIM та визначено межі стійкості системи.

ABSTRACT

Bachelor's thesis contains: 98 p., 17 fig., 2 ann., 35 references.

ADAPTIVE REGULARIZATION, CHAMBOLLE-POCK
ALGORITHM, CONSENSUS ADMM, DENOISING, MULTI-AGENT
SYSTEM, PYTHON, TOTAL VARIATION.

Object of research – digital filtering and restoration of images corrupted by random noise.

Subject of research – distributed consensus models, primal-dual saddle-point optimization, and multi-agent denoising methods.

Purpose of the work – to develop a decentralized multi-agent filtering system based on Consensus ADMM with spatially-adaptive weight maps for edge and geometry preservation.

Research methods – variational optimization, Markov Random Fields, operator splitting methods, parallel computing, and OOP in Python.

The thesis investigates the evolution of denoising models from MRF to Consensus ADMM. An architecture for parallel image decomposition into overlapping patches managed by autonomous agents is developed. Integrating the internal primal-dual Chambolle-Pock solver with a "warm-start" mechanism minimized computational overhead. The proposed modification with dynamic local variation weights eliminated texture over-smoothing. Based on primal and dual residuals, the convergence dynamics were analyzed, structural trade-offs were evaluated via PSNR/SSIM, and the system's stability boundaries were identified.

ЗМІСТ

Вступ.....	9
1 Аналіз предметної галузі та постановка задачі.....	10
1.1 Опис предметної галузі	10
1.2 Математичні моделі спотворення цифрових зображень	11
1.2.1 Адитивний білий гаусів шум	11
1.2.2 Імпульсний шум	12
1.3 Огляд існуючих методів фільтрації шуму в зображеннях.....	14
1.3.1 Традиційні просторові фільтри	14
1.3.2 Методи на основі варіаційного числення та дифузії.....	18
1.3.3 Нелокальні методи	20
1.4 Парадигма багатоагентних систем в обробці візуальних даних	23
1.4.1 Поняття інтелектуального агента в обробці даних	23
1.4.2 Переваги розподіленого консенсусу над централізованими архітектурами	24
1.5 Постановка задачі.....	25
2 Теоретичні основи консенсусної взаємодії в MAS.....	28
2.1 Локальний децентралізований консенсус на основі метрик	28
2.1.1 Топологія 4-зв'язної сітки пікселів-агентів	28
2.1.2 Математичні метрики довіри та згоди	29
2.1.3 Правила ітераційного оновлення станів	31
2.1.4 Аналіз недоліків та причин неефективності наївної моделі	32
2.2 Контекстуальне моделювання через Марковські випадкові поля	33
2.2.1 Локальна марковська властивість у графах зображень	33
2.2.2 Енергетична функція та регуляризація Хубера	34
2.2.3 Модифікація алгоритму Iterated Conditional Modes для багатоагентних систем.....	35
2.3 Математичний апарат методу змінних множників Лагранжа.....	36
2.3.1 Оптимізація ADMM для задач розподіленого консенсусу.....	37

2.3.2	Декомпозиція задач локальної та глобальної оптимізації	37
2.3.3	Вплив дуальних змінних на похибку неузгодженості	38
3	Аналіз та проєктування математичної моделі зваженого Consensus ADMM MAS	40
3.1	Глобальна оптимізація та агентна декомпозиція задачі	40
3.1.1	Концепція багатагентного розбиття та топологія графа перекриття.....	40
3.1.2	Глобальна функція енергії та обмеження консенсусу	41
3.2	Децентралізована схема оптимізації на основі методу ADMM та алгоритму Шамболла-Пока.....	41
3.2.1	Формалізація локального ітераційного кроку системи.....	42
3.2.2	Оптимізація внутрішньої сідлової задачі з консенсусними обмеженнями	44
3.3	Просторово-адаптивна модифікація карт ваг регуляризації в архітектурі агента.....	45
3.3.1	Локальне оцінювання дисперсії та динамічне нормування ваг	45
3.3.2	Вплив гетерогенного розподілу ваг на збіжність та залишкові похибки	47
4	Програмна реалізація та оцінка ефективності	49
4.1	Обґрунтування вибору технологічного стеку	49
4.2	Архітектура програмного комплексу та об'єктна модель	50
4.2.1	Реалізація обчислювального вузла: клас Agent	50
4.2.2	Реалізація багатагентного середовища: клас Coordinator	53
4.3	Інженерний аналіз динаміки збіжності розподіленої системи.....	56
4.4	Порівняльний аналіз результатів відновлення за метрикою PSNR...	59
4.4.1	Оцінка ефективності при гаусовому спотворенні	59
4.4.2	Оцінка ефективності при імпульсному спотворенні.....	61
4.5	Візуальний аналіз та структурні компроміси фільтрації	63
4.5.1	Дослідження артефактів при гаусовому зашумленні.....	63
4.5.2	Аналіз поведінки агентів при імпульсному зашумленні	66

4.6 Аналіз стійкості алгоритмів до інтенсивності вхідного шуму	70
Висновки	72
Перелік джерел посилення	74
Додаток А Код розробленого алгоритму	78
Додаток Б Відомість кваліфікаційної роботи.....	98

ВСТУП

Сучасний етап розвитку інформаційних технологій характеризується стрімким зростанням обсягів візуальної інформації, що потребує автоматизованої обробки. Якість цифрових зображень є визначальним фактором для ефективного функціонування систем комп'ютерного зору, медичної візуалізації та супутникового моніторингу. Проте в процесі реєстрації та передачі даних зображення неминуче зазнають впливу шумів різної природи, що створює необхідність пошуку нових методів покращення їхньої якості.

Актуальність даного дослідження зумовлена обмеженнями класичних алгоритмів фільтрації, які часто не забезпечують належного балансу між пригніченням завад та збереженням дрібних деталей. Традиційні підходи зазвичай ігнорують локальну структуру зображення, що призводить до деградації контурів об'єктів. Використання багатоагентних систем пропонує альтернативну парадигму обробки даних, що базується на принципах самоорганізації та розподіленого моделювання.

В основі запропонованого підходу лежить концепція представлення кожного елемента цифрового масиву як автономного інтелектуального агента. Таке моделювання дозволяє реалізувати гнучкі стратегії поведінки, де зміна стану окремого пікселя залежить від динамічної взаємодії з оточенням. Це відкриває можливості для створення адаптивних алгоритмів, здатних самостійно налаштовуватися на специфіку конкретних ділянок зображення без використання громіздких нейромережових архітектур.

Метою роботи є теоретичне обґрунтування та моделювання багатоагентної системи для адаптивної обробки зашумлених зображень. Основними етапами дослідження є аналіз існуючого математичного апарату фільтрації, розробка правил поведінки пікселів-агентів та перевірка ефективності запропонованої моделі на тестових зразках.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної галузі

Цифрова обробка зображень сьогодні є критично важливою складовою у багатьох сферах людської діяльності: від побутової фотографії та соціальних мереж до медицини, систем безпеки та супутникового моніторингу. Проте в процесі формування, передачі та зберігання цифрові зображення неминуче зазнають впливу різних видів шумів, що погіршує їхню візуальну якість та ускладнює подальший автоматизований аналіз [1].

Традиційні методи фільтрації, такі як медіанний фільтр або фільтр Гауса, хоча і є швидкими, часто мають суттєвий недолік – вони застосовують однорідний підхід до всього зображення. Це призводить до втрати дрібних деталей та розмиття контурів, що є неприпустимим у завданнях, де точність відтворення меж об'єктів є пріоритетною.

В останні роки все більшої популярності набувають інтелектуальні системи обробки даних, зокрема багатоагентні системи (MAS). На відміну від класичних алгоритмів, MAS базуються на взаємодії великої кількості автономних інтелектуальних одиниць (агентів), кожен з яких може приймати рішення локально, адаптуючись до специфіки конкретної ділянки зображення. Такий підхід дозволяє реалізувати гнучкі стратегії фільтрації: наприклад, інтенсивніше пригнічувати шум на фонових областях і зберігати чіткість на текстурованих ділянках чи краях об'єктів.

Застосування багатоагентного моделювання відкриває шлях до створення розподілених систем фільтрації, які потенційно можуть бути більш стійкими до складних типів шумів, де параметри спотворення змінюються в межах одного кадру. Проте реалізація таких систем на базі сучасних мов програмування, як-от Python, потребує ретельного балансу між складністю поведінки агентів та обчислювальною потужністю, необхідною для обробки зображень високої роздільної здатності.

1.2 Математичні моделі спотворення цифрових зображень

Математично чисте цифрове зображення можна представити як двовимірну функцію інтенсивності $I(x, y)$, де (x, y) – просторові координати пікселя у дискретній сітці розміром $N \times M$. Реальний процес реєстрації сигналу супроводжується деструктивними факторами, тому система отримує спотворене спостереження $Y(x, y)$. Для побудови математично обґрунтованих систем денойзингу необхідно класифікувати ці спотворення за їхньою природою.

1.2.1 Адитивний білий гаусів шум

Адитивний білий гаусів шум є найбільш поширеною моделлю спотворення в задачах цифрової обробки зображень. Фізично він виникає через тепловий рух електронів у напівпровідникових елементах сенсора камери [2].

Математична модель адитивного спотворення описується формулою 1.1:

$$Y(x, y) = I(x, y) + n(x, y), \quad (1.1)$$

де $Y(x, y)$ – вихідне спотворене шумом цифрове зображення;

$I(x, y)$ – істинне значення інтенсивності;

$n(x, y)$ – випадкова величина шуму, яка не залежить від корисного сигналу.

Для кожного координатного зразка значення n є незалежною випадковою величиною, що підпорядковується нормальному закону розподілу з нульовим математичним сподіванням та дисперсією σ^2 , заданим формулою 1.2:

$$p(n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{n^2}{2\sigma^2}\right). \quad (1.2)$$

Параметр σ (середньоквадратичне відхилення) визначає потужність шуму: чим більшим є значення σ , тим сильніше зернистість перекриває структуру зображення. Термін «білий» означає, що спектральна щільність потужності шуму є рівномірною на всіх частотах, тобто значення шуму в одному пікселі ніяк не корелюють із його значенням у сусідніх точках. Основним викликом при видаленні AWGN є те, що його спектр перекриває корисний спектр зображення, особливо в області високих частот, де знаходяться контури та дрібні текстури. Візуальний характер такої деградації для різних масштабів зашумлення наведено на рисунку 1.1.

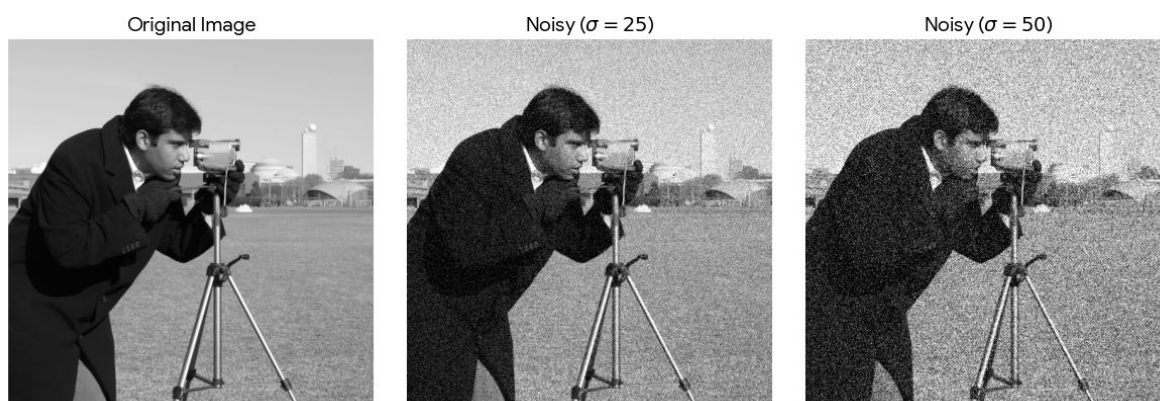


Рисунок 1.1 – Приклад спотворення тестового зображення адитивним білим гаусовим шумом із різними значеннями σ

1.2.2 Імпульсний шум

На відміну від гаусового шуму, який плавно змінює значення кожного пікселя, імпульсний шум діє дискретно та екстремально. Він проявляється у вигляді хаотично розкиданих ізольованих яскравих або темних точок [3].

Найбільш відомою різновидністю імпульсного шуму є шум типу «сіль і перець». Його математична модель для 8-бітного зображення задається розподілом:

$$Y(x, y) = \begin{cases} 0, & \text{із ймовірністю } p/2 \\ 255, & \text{із ймовірністю } p/2 \\ I(x, y), & \text{із ймовірністю } 1 - p \end{cases}, \quad (1.3)$$

де p – загальна щільність зашумлення кадру.

Головний виклик при роботі з імпульсним шумом полягає в тому, що амплітуда завади повністю заміщує інформацію про оригінальний піксель у пошкодженій точці екстремальними значеннями яскравості (мінімальними для «перцю» та максимальними для «солі»). Будь-які спроби застосування класичного лінійного усереднення для таких дефектів є неефективними, оскільки вони призводять до деградації сусідніх неушкоджених пікселів, дифузії аномальних значень та формування виражених артефактів розмиття і плям зниженої контрастності. Математична згортка інтегрує ці аномальні викиди в загальну статистику локального вікна, безповоротно розмиваючи чіткі контури об'єктів. Характерний вигляд зображення за умов такої деградації наведено на рисунку 1.2.

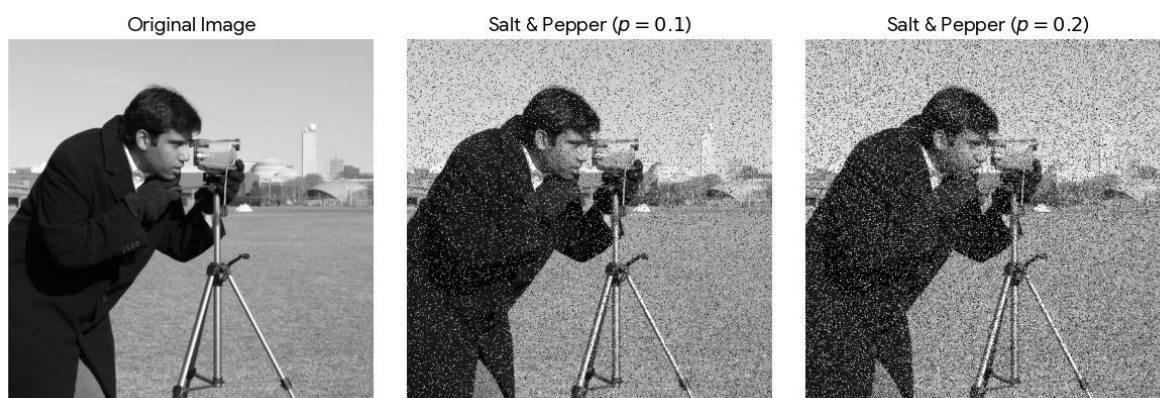


Рисунок 1.2 – Приклад спотворення тестового зображення імпульсним шумом типу «сіль і перець» із різною щільністю зашумлення

Процес усунення цих шумів пов'язаний з ризиком деградації структури зображення. До ключових технічних викликів при розробці систем фільтрації відносяться:

- ефект розмиття: втрата чіткості контурів об'єктів через те, що більшість алгоритмів усереднюють значення пікселів на основі їхнього оточення, роблячи межі нечіткими для аналізу деталей;

- обчислювальна складність: обробка зображень високої роздільної здатності інтелектуальними підходами, де кожен піксель є активною одиницею, створює суттєві часові затримки через зростання кількості операцій;

- адаптивність обробки: здатність системи автоматично визначати ступінь зашумленості кожної ділянки. Надмірна обробка чистих зон створює артефакти, а недостатня – залишає шум.

Саме необхідність подолання цих викликів обумовлює доцільність розробки адаптивних багатоагентних моделей, здатних до локального прийняття рішень.

1.3 Огляд існуючих методів фільтрації шуму в зображеннях

На сьогодні розроблено велику кількість методів пригнічення шуму, які можна розділити на кілька основних категорій залежно від математичного апарату та принципу дії.

1.3.1 Традиційні просторові фільтри

Традиційні просторові фільтри є найбільш розповсюдженою та дослідженою групою методів обробки зображень. В основі їхнього функціонування лежить концепція локальної обробки, де нове значення цільового пікселя обчислюється на основі значень його сусідів у межах певної обмеженої області. Для опису цієї області використовується поняття

ковзного вікна або апертури фільтра, яке послідовно переміщується по всьому масиву зображення від пікселя до пікселя. Розмір та форма цього вікна, яке найчастіше представляє собою квадратні матриці розміром 3×3 , 5×5 або 7×7 пікселів, безпосередньо впливають на фінальну якість фільтрації та обчислювальні витрати. Принцип організації ковзного вікна та взаємодії з сусідніми елементами схематично наведено на рисунку 1.3 [4].

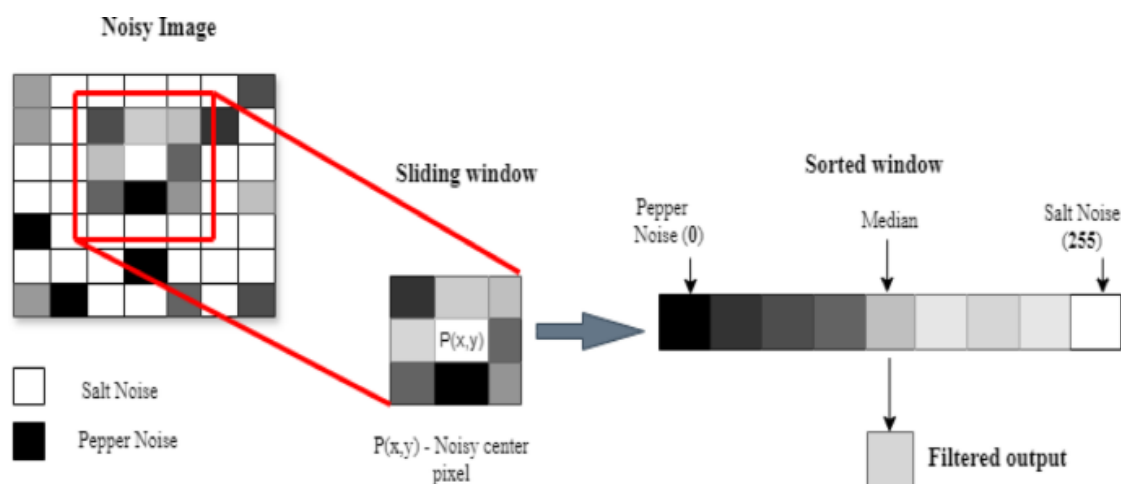


Рисунок 1.3 – Схема функціонування апертури просторового фільтра

Залежно від математичного апарату, який застосовується до елементів усередині ковзного вікна, просторові методи поділяють на лінійні та нелінійні.

Лінійна просторова фільтрація найчастіше базується на застосуванні фільтра середнього арифметичного або фільтра Гауса. Математична суть простого усереднювального фільтра полягає в тому, що кожному пікселю в межах вікна присвоюється однаковий ваговий коефіцієнт, величина якого обернено пропорційна загальній кількості елементів у вікні. Цей однорідний статичний підхід забезпечує рівномірне згладжування високочастотних коливань яскравості виключно на гладких ділянках. Після завершення операції згортки та нормування матриці обчислюється фінальна сума інтенсивностей за формулою 1.4:

$$G(x, y) = \frac{1}{M} \sum_{(i,j) \in S_{xy}} F(i, j), \quad (1.4)$$

де $G(x, y)$ – нове значення центрального пікселя;

M – загальна кількість пікселів в апертурі.

$F(i, j)$ – вихідна інтенсивність пікселів у межах вікна S_{xy} ;

Для більш якісного згладжування використовують лінійний фільтр Гауса, де вагові коефіцієнти розподілені відповідно до двовимірного нормального закону, заданого формулою 1.5:

$$H(i, j) = \frac{1}{2\pi\sigma_f^2} \exp\left(-\frac{i^2+j^2}{2\sigma_f^2}\right), \quad (1.5)$$

де σ_f – ступінь розмиття фільтра.

Ці лінійні методи демонструють високу ефективність при пригніченні адитивного білого гаусового шуму, оскільки випадкові коливання інтенсивності з нульовим математичним сподіванням нівелюються при усередненні. Проте при спробі усунути імпульсний шум фільтр середнього значення є неефективним, адже замість видалення яскравих чи темних точок він просто розподіляє їхню інтенсивність по сусідніх пікселях, створюючи брудні сірі плями. Головним мінусом лінійного усереднення є неминуче і значне розмиття контурів об'єктів, відоме як ефект Blurring [5].

Нелінійна просторова фільтрація найчастіше реалізується за допомогою медіанного фільтра, де замість математичного усереднення значень виконується їх ранжування та сортування [6]. Усі значення інтенсивностей пікселів, що потрапили в апертуру, вибудовуються у варіаційний ряд за зростанням, після чого центральному пікселю присвоюється значення, яке знаходиться рівно посередині цього відсортованого списку. Математична модель такого перетворення описується виразом 1.6:

$$G(x, y) = \text{median}\{F(i, j) \mid (i, j) \in S_{xy}\}. \quad (1.6)$$

Медіанна фільтрація є стандартом для боротьби з імпульсним шумом типу «сіль-і-перець». Оскільки такі завади мають екстремальні значення, при сортуванні вони опиняються на краях варіаційного ряду, тому медіана обирає неспотворене значення з оточення. Проте при зашумленні понад 50% пікселів у вікні алгоритм приймає спотворені імпульси за корисний сигнал, утворюючи плямисті артефакти. Крім того, при великому розмірі вікна або повторній фільтрації ігнорується тонка просторова структура, що призводить до заокруглення гострих кутів. Візуальне зіставлення результатів обробки зображення з імпульсними дефектами за допомогою обох просторових підходів наведено на рисунку 1.4.



Рисунок 1.4 – Результати фільтрації зображення, зашумленого імпульсним шумом, за допомогою фільтрів середнього та медіанного значень

При зіткненні з гаусовим розподілом завад ефективність нелінійної медіанної схеми помітно деградує порівняно з класичним лінійним усередненням. Оскільки гаусів шум плавно флюктує навколо корисного сигналу в кожній точці, операція вибору центрального елемента варіаційного ряду не дозволяє повністю анігілювати низькоамплітудні коливання, залишаючи на зображенні помітну низькочастотну текстурну

плямистість. Натомість фільтр середнього значення у цьому сценарії забезпечує гладкіший фон, хоча й продовжує руйнувати різкі межі об'єктів. Характер роботи лінійного та нелінійного віконних фільтрів в умовах гаусової деструкції сигналу проілюстровано на рисунку 1.5.



Рисунок 1.5 – Результати фільтрації зображення, зашумленого гаусовим шумом, за допомогою фільтрів середнього та медіанного значень

Таким чином, розгляд традиційних просторових фільтрів вказує на їхню вузьку спеціалізацію та обмеженість. Жоден із наведених базових підходів не здатний одночасно забезпечити якісне пригнічення шуму та збереження геометрії контурів, що зумовлює необхідність переходу до складніших варіаційних моделей обробки інформації [7].

1.3.2 Методи на основі варіаційного числення та дифузії

Методи фільтрації, що базуються на апараті математичної фізики, варіаційного числення та диференціальних рівнянь у частинних похідних, представлені розгалуженою групою алгоритмів адаптивного згладжування. Класичним представником варіаційного підходу є модель Рудіна-Ошера-Фатемі (ROF), яка формулює задачу придушення шуму як пошук мінімуму енергетичного функціонала з регуляризацією за повною варіацією (Total

Variation, TV) [1], [8]. Математичне представлення цієї оптимізаційної задачі описується виразом 1.7:

$$E(I) = \int_{\Omega} \|\nabla I\| dx dy + \frac{\lambda}{2} \int_{\Omega} (Y - I)^2 dx dy, \quad (1.7)$$

де I – шукане відновлене зображення;

∇I – просторовий градієнт яскравості зображення;

$\|\nabla I\|$ – оператор повної варіації, що виступає в ролі регуляризатора;

λ – додатний параметр регуляризації (коефіцієнт Лагранжа);

Ω – двовимірна область визначення кадру.

Альтернативним підходом у рамках диференціальних рівнянь є модель анізотропної дифузії Перона-Маліка [9]. Математична суть цього підходу полягає в модифікації класичного рівняння теплопровідності шляхом введення змінного коефіцієнта дифузії, який є нелінійною функцією від градієнта яскравості зображення. Диференціальне рівняння в частинних похідних, що описує цей процес еволюції сигналу в часі, задано формулою 1.8:

$$\frac{\partial I}{\partial t} = \operatorname{div}(c(x, y, t) \nabla I), \quad (1.8)$$

де div – оператор дивергенції;

$c(x, y, t)$ – функція провідності, яка контролює швидкість дифузійного процесу в кожній точці масиву даних;

∇I – градієнт яскравості зображення.

Функція провідності проєктується таким чином, щоб її значення наближалось до одиниці на однорідних ділянках зображення та прямувало до нуля на межах об'єктів, де локальний градієнт $\|\nabla I\|$ суттєво перевищує заданий поріг K . Для реалізації цієї логіки в оригінальній моделі Перона-

Маліка використовують два типи залежностей, що описуються виразами 1.9 та 1.10 [9]:

$$c(\nabla I) = \exp\left(-\left(\frac{\|\nabla I\|}{K}\right)^2\right), \quad (1.9)$$

$$c(\nabla I) = \frac{1}{1+\left(\frac{\|\nabla I\|}{K}\right)^2}. \quad (1.10)$$

Константа K виступає просторовим фільтром частот. Якщо градієнт є меншим за K , функція провідності наближається до одиниці, активуючи дифузійне згладжування шуму, а у протилежному випадку – стрімко падає, ізолюючи контур від розмиття. Результати адаптивного згладжування за допомогою моделі анізотропної дифузії проілюстровано на рисунку 1.6.

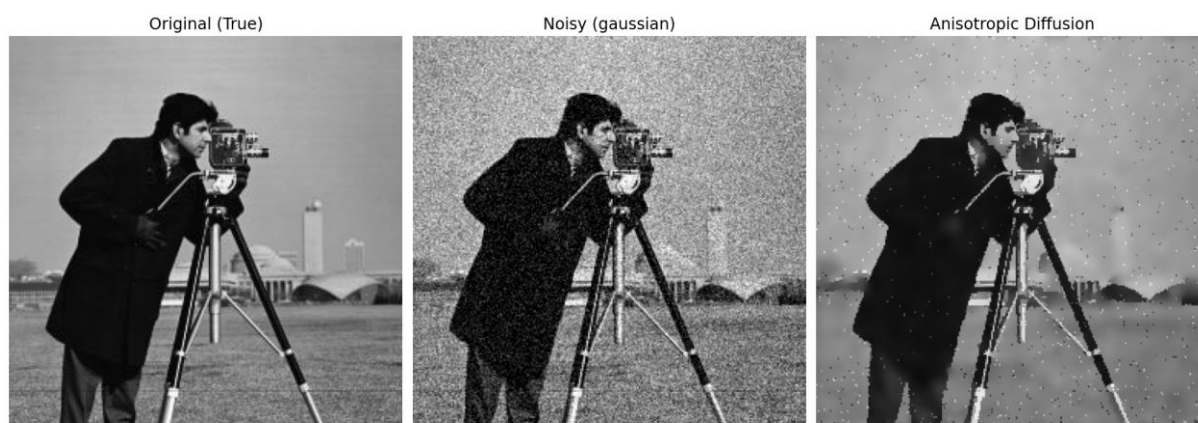


Рисунок 1.6 – Динаміка збереження градієнтів та результати адаптивного згладжування за допомогою моделі анізотропної дифузії Перона-Маліка

1.3.3 Нелокальні методи

Суттєвим обмеженням розглянутих раніше локальних підходів є їхня просторова обмеженість, тобто використання інформації виключно з

найближчого геометричного оточення цільового пікселя. Сучасні нелокальні методи обробки зображень долають цю проблему за рахунок пошуку та аналізу схожих структур і текстурних патернів по всій площі кадру, що базується на принципі природної надлишковості візуальної інформації.

Класичний алгоритм нелокального усереднення Non-Local Means обчислює відновлене значення інтенсивності пікселя як зважену суму інтенсивностей усіх пікселів зображення за формулою 1.11 [10]:

$$G(x) = \frac{1}{c(x)} \sum_{y \in \Omega} w(x, y) F(y), \quad (1.11)$$

де вага $w(x, y)$ – ступінь геометричної схожості патчів N_x та N_y , задана формулою 1.12:

$$w(x, y) = \exp\left(-\frac{\|F(N_x) - F(N_y)\|_{2,a}^2}{h^2}\right), \quad (1.12)$$

де $\|F(N_x) - F(N_y)\|_{2,a}^2$ – Евклідова відстань між векторами інтенсивностей патчів N_x та N_y ;

h – параметр фільтрації, що контролює ступінь загального згладжування шуму.

Розвитком ідеї нелокальної подібності є алгоритм тривимірної блочної фільтрації BM3D, який на сьогодні виступає поточним стандартом серед класичних варіаційних та трансформаційних підходів денойзингу. Математична логіка цього методу базується на концепції спільної фільтрації у тривимірному просторі та реалізується через послідовне виконання трьох основних процедур [11].

На етапі блочного групування (Grouping) алгоритм здійснює пошук схожих двовимірних патчів по всьому кадру та об'єднує їх у тривимірні

стеки. На етапі спільної фільтрації (Collaborative Filtering) до кожного 3D-масиву застосовується тривимірне лінійне перетворення (дискретне косинусне або вейвлет-перетворення). Оскільки корисні структури у схожих фрагментах сильно корелюють, а завада є випадковою, енергія чистого сигналу концентрується в кількох великих коефіцієнтах, що дозволяє очистити сигнал шляхом порогової або вінерівської фільтрації. Під час оберненого перетворення та агрегації (Aggregation) блоки повертаються на свої координати, а фінальні значення пікселів обчислюються шляхом зваженого усереднення перекритих областей. Принципову схему виконання етапів спільної 3D-фільтрації наведено на рисунку 1.7.

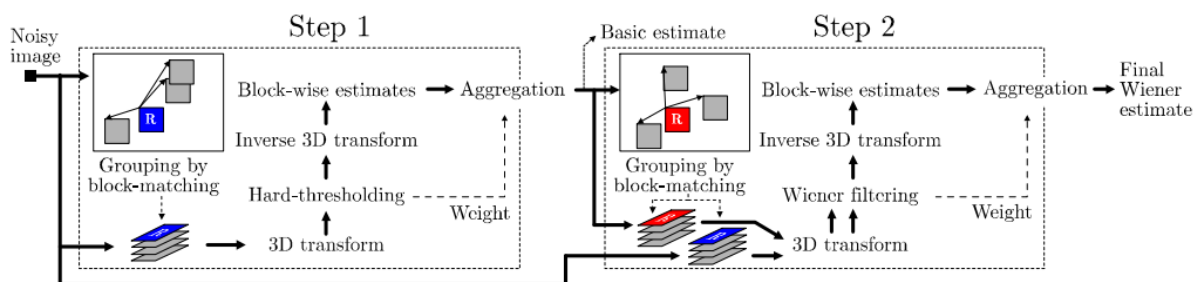


Рисунок 1.7 – Принципова інженерна схема етапів групування, спільної тривимірної фільтрації та агрегації в алгоритмі BM3D

Завдяки такій тривимірній трансформаційній структурі метод BM3D забезпечує безпрецедентно високу якість відновлення дрібних текстур та складних геометричних контурів. Однак його обчислювальна складність є вкрай високою через необхідність постійного нелокального пошуку відповідностей між патчами по всьому інформаційному полю, що суттєво обмежує його застосування в децентралізованих обчислювальних середовищах та системах обробки даних у реальному часі. Комбінаторний перебір величезної кількості фрагментів створює критичне навантаження на пам'ять та уповільнює загальну швидкість фільтрації.

1.4 Парадигма багатоагентних систем в обробці візуальних даних

Обмеження класичних локальних та нелокальних методів фільтрації, пов'язані з їхньою статичністю та високою обчислювальною складністю на централізованих архітектурах, зумовлюють пошук нових підходів до організації обчислень. Одним із перспективних напрямків є застосування парадигми багатоагентних систем, яка пропонує перехід від єдиного глобального обчислювального центру до мережі автономних суб'єктів, що взаємодіють між собою.

1.4.1 Поняття інтелектуального агента в обробці даних

Багатоагентна система – це сукупність динамічних автономних сутностей (агентів), що взаємодіють між собою у спільному віртуальному або фізичному середовищі для вирішення задач, які перевищують можливості або знання окремого агента. У контексті цифрової обробки зображень інтелектуальний агент розглядається як локалізована програмна одиниця. Вона має власну внутрішню модель стану, здатна сприймати інформацію з обмеженої ділянки середовища, обмінюватися повідомленнями із сусідніми суб'єктами та ітераційно оновлювати свої параметри для досягнення глобального консенсусу [12].

Для моделювання MAS на двовимірній дискретній сітці зображення визначається топологія зв'язків між агентами [2]. Комунікаційне середовище зазвичай представляється у вигляді неорієнтованого графа за виразом 1.13:

$$G = (V, E), \quad (1.13)$$

де V – множина агентів;

E – множина ребер, що визначають наявність каналу зв'язку між ними.

Залежно від архітектури розподіленої системи та стратегії декомпозиції даних виділяють кілька основних типів локальних топологій зв'язку. Найпростішим варіантом є топологія чотиризв'язної сітки, де кожен окремий обчислювальний агент взаємодіє виключно з найближчими сусідами по вертикалі та горизонталі. Більш комплексна топологія у вигляді вісьмизв'язної сітки додатково враховує діагональні напрямки, що дозволяє системі значно точніше передавати геометричні градієнти кутів та складних текстур у процесі ітераційного обміну повідомленнями між вузлами [13].

Окреме місце посідає патч-орієнтована топологія, яка є базовою для розробленого архітектурного рішення. У цьому випадку агентами виступають не окремі дискретні пікселі, а цілі локальні фрагменти (патчі) зображення, що частково перекриваються між собою за регулярною сіткою. Інформаційний зв'язок та ребро консенсусу утворюються між тими агентами, чий просторові зони відповідальності мають спільні координати перекриття. Такий підхід дозволяє суттєво знизити комунікаційні накладні витрати мережі та забезпечити ефективне паралелення обчислень.

Взаємодія між агентами є суворо децентралізованою. Агенти не мають прямого доступу до глобальних характеристик усього зображення. Кожен суб'єкт приймає рішення щодо зміни свого стану на основі власного локального спостереження та повідомлень.

1.4.2 Переваги розподіленого консенсусу над централізованими архітектурами

Фундаментальним механізмом координації дій у децентралізованих багатоагентних системах є алгоритми розподіленого консенсусу. Задача консенсусу полягає в тому, щоб змусити групу автономних агентів прийти до спільного асимптотичного значення або узгодженого стану через ітераційний обмін даними [14].

У традиційній постановці денойзингу централізований алгоритм намагається оптимізувати глобальний функціонал для всього кадру одночасно. Це вимагає великих обсягів оперативної пам'яті та унеможливорює ефективне розпаралелювання. Перехід до розподіленого консенсусу в MAS забезпечує кілька важливих переваг:

- локальна адаптивність. Кожен агент або група агентів може динамічно змінювати правила взаємодії залежно від локальних характеристик ділянки кадру, розрізняючи однорідний фон та межі об'єктів;

- масштабованість та паралелізм. Оскільки обчислення виконуються локально, алгоритм ідеально масштабується на архітектури з багатьма обчислювальними ядрами, де кожен процесор обробляє свою підмножину агентів;

- відмовостійкість. Відсутність єдиної точки відмови робить систему стійкою до локальних збоїв або математичних аномалій в окремих регіонах зображення, оскільки загальний консенсус стабілізує систему.

Таким чином, децентралізована взаємодія у межах обраної комунікаційної топології дозволяє MAS ефективно перерозподіляти обчислювальне навантаження. Замість вирішення глобальної оптимізаційної задачі для всього масиву візуальних даних, система мінімізує локальні розбіжності між сусідніми агентами, що за умови правильної математичної формалізації забезпечує збіжність до єдиного стабільного стану [15].

1.5 Постановка задачі

На основі детального аналізу предметної галузі, сучасних моделей спотворень та існуючих підходів до денойзингу, задача даної бакалаврської роботи формалізується наступним чином.

Нехай задано двовимірну матрицю спотвореного цифрового зображення Y розміром $N \times M$, яка сформована внаслідок накладання

випадкового адитивного білого гаусового шуму або імпульсного шуму на оригінальне чисте зображення I . Математично задача відновлення оцінки оригінального сигналу зводиться до розподіленої оптимізації, в межах якої система автономних агентів мінімізує сукупну функцію, що поєднує критерії відповідності початковим даним (data fidelity) та просторової гладкості (smoothness). Цей процес формалізується у вигляді математичної моделі з консенсусними обмеженнями, яка описується виразом 1.15:

$$\min_{x_1, \dots, x_K} \sum_{i=1}^K f_i(x_i) \quad \text{при умові} \quad x_i = z_i, \quad \forall i, \quad (1.15)$$

де $f_i(x_i)$ – локальна функція втрат i -го агента;

x_i – локальний стан i -го агента;

z_i – значення глобального консенсусу в точках просторового перекриття чи сусідства.

Метою даної роботи є дослідження та моделювання багатоагентної системи для адаптивної фільтрації шумів у цифрових зображеннях на основі алгоритмів розподіленого консенсусу. Розроблювана система повинна забезпечувати ефективне пригнічення основних видів завад при максимальному збереженні контурів, різких меж та дрібних деталей об'єктів без використання централізованого обчислювального ядра.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- виконати детальний аналіз предметної галузі, вивчити фізичну та аналітичну природу виникнення цифрових шумів, а також особливості існуючих методів їх усунення;

- провести порівняльний огляд сучасних алгоритмів фільтрації, визначити їхні переваги та обмеження у контексті обробки зображень високої роздільної здатності;

- обґрунтувати вибір багатоагентного підходу та розробити базову топологію взаємодії обчислювальних агентів у децентралізованому середовищі;

- розробити алгоритми локальної адаптації та динамічного оцінювання міри довіри між сусідніми елементами системи для запобігання ефекту надмірного просторового розмиття;

- дослідити механізми забезпечення математичної збіжності розподіленого консенсусу та оптимізувати правила ітераційного обміну повідомленнями між агентами;

- реалізувати програмну модель децентралізованої системи та провести серію комп'ютерних експериментів з різними типами та рівнями зашумлення;

- провести порівняльний аналіз отриманих результатів із класичними підходами за допомогою об'єктивних метрик якості та оцінити ефективність запропонованого архітектурного рішення.

Об'єктом дослідження є процеси цифрової фільтрації та відновлення зображень, що зазнали впливу випадкових завад.

Предметом дослідження є багатоагентна модель розподіленої обробки візуальних даних для адаптивного пригнічення шуму.

2 ТЕОРЕТИЧНІ ОСНОВИ КОНСЕНСУСНОЇ ВЗАЄМОДІЇ В MAS

2.1 Локальний децентралізований консенсус на основі метрик

Найбільш інтуїтивним підходом до побудови багатоагентної системи денойзингу є відображення структури зображення безпосередньо на архітектуру обчислювальної мережі. У такій моделі кожен окремий піксель дискретної сітки розглядається як незалежний інтелектуальний агент. Перша ітерація розроблюваної системи базується на концепції попиксельного синхронного обміну повідомленнями, де агенти ітераційно приходять до консенсусу щодо істинного значення інтенсивності, балансуючи між початковим зашумленим спостереженням та думкою свого просторового оточення.

2.1.1 Топологія 4-зв'язної сітки пікселів-агентів

Для реалізації локальної взаємодії у першій версії системи обрано топологію Grid-4. У такій конфігурації кожен окремий піксель дискретної сітки зображення розміром $M \times N$ розглядається як незалежний автономний обчислювальний агент, фіксований у просторі. Кожен вузол має прямі двосторонні канали зв'язку виключно з чотирма безпосередніми сусідами по вертикалі та горизонталі. Математично для агента з просторовими координатами (i, j) комунікаційна множина його оточення $N_{i,j}$ описується виразом 2.1 [13]:

$$N_{i,j} = \{(i - 1, j), (i + 1, j), (i, j - 1), (i, j + 1)\}, \quad (2.1)$$

Комунікаційне середовище представляється регулярним графом, де ребра консенсусу існують лише між просторово суміжними пікселями за виразом 2.1. На кожній ітерації алгоритму агент надсилає свій поточний

локальний стан (belief) усім учасникам зі своєї множини сусідства $N_{i,j}$ та приймає від них аналогічні дані. Оскільки обробка є строго синхронною, оновлення станів усіх агентів на кроці $t + 1$ відбувається одночасно на основі повідомлень з попереднього кроку t . Це дозволяє ітераційно приводити систему до єдиного компромісного рішення, балансуючи між зашумленим спостереженням та оцінкою оточення.

2.1.2 Математичні метрики довіри та згоди

Фундаментальною особливістю запропонованої наївної моделі є відмова від статичних коефіцієнтів згладжування на користь динамічного розрахунку міри довіри та математичної згоди між агентами на кожній ітерації.

Нехай $Y_{i,j}$ є початковим зашумленим спостереженням у координатах (i,j) , а $x_{i,j}^{(t)}$ – внутрішнє переконання агента щодо чистого значення інтенсивності на ітерації k . Вага або впевненість агента у власному початковому спостереженні c_1 динамічно обчислюється через експоненційне штрафування просторового відхилення, описане формулою 2.2:

$$c_{1ij} = \exp\left(-\Gamma \cdot \text{dist}\left(Y_{i,j}, \text{mean}\left(B_{N_{i,j}}\right)\right)\right), \quad (2.2)$$

де Γ – гіперпараметр чутливості до аномалій;

dist – метрика відстані у просторі інтенсивностей;

mean – оператор обчислення середнього арифметичного;

$B_{N_{i,j}}$ – множина поточних переконань сусідніх агентів.

Така експоненціальна залежність у виразі 2.2 виступає в ролі наївної локальної евристики: якщо вихідне значення пікселя $Y_{i,j}$ суттєво

відхиляється від оточення (що вказує на ймовірну наявність імпульсного викиду), коефіцієнт c_{1ij} штучно занижується, зменшуючи вплив аномалії на подальші розрахунки. Це дозволяє оперативно ізолювати пошкоджені елементи та гнучко адаптувати інтенсивність згладжування ще на початковому етапі фільтрації.

Проте для формування повного балансу сил у загальному правилі оновлення станів однієї лише самооцінки агента недостатньо. Необхідно також динамічно визначити взаємну довіру між самими сусідніми вузлами. Для цього коефіцієнти зв'язку c_{2ij} між центральним агентом i та його сусідами $j \in N_i$ на кожній ітерації перераховуються за формулою 2.3:

$$c_{2ij} = \text{conf}_{i,j} \cdot \exp\left(-\Lambda \cdot \text{dist}\left(x_i^{(t)}, x_j^{(t)}\right)\right), \quad (2.3)$$

де $\text{conf}_{i,j}$ – коефіцієнт локальної структурної впевненості;

Λ – коефіцієнт масштабування, що регулює чутливість до поточного розходження між станами.

Значення $\text{conf}_{i,j}$ не є статичним і динамічно визначається через локальне середньоквадратичне відхилення інтенсивностей усіх сусідів у поточному оточенні за виразом 2.4:

$$\text{conf}_{i,j} = \exp\left(-\sigma_{\text{conf}} \cdot \text{std}(B_{N_i})\right), \quad (2.4)$$

де $\text{std}(B_{N_i})$ – оператор обчислення стандартного відхилення поточних переконань у множині сусідства N_i ;

σ_{conf} – гіперпараметр, що контролює ступінь штрафування за високу дисперсію в околі.

Якщо в околі агента спостерігається висока дисперсія (що свідчить про інтенсивний шум), показник глобальної довіри до повідомлень цього

оточення зменшується. Водночас експоненційний член у формулі c_{2ij} виконує роль математичного обмежувача: якщо значення сусіда $x_i^{(t)}$ сильно відрізняється від $x_j^{(t)}$, їхня взаємна вага падає, що блокує дифузію через геометричні контури зображення.

2.1.3 Правила ітераційного оновлення станів

Синхронний векторний цикл оновлення переконань (Belief Update Loop) для всієї сукупності агентів MAS реалізує правило зваженого локального консенсусу. Математична формула перерахунку стану i -го агента на кожній ітерації має такий вигляд 2.5:

$$x_i^{(t+1)} = \frac{c_{1ij} \cdot Y_i + \sum_{j \in N_i} c_{2ij} \cdot x_j^{(t)}}{c_{1ij} + \sum_{j \in N_i} c_{2ij}}. \quad (2.5)$$

Аналіз даного правила оновлення показує, що за своєю математичною природою цей алгоритм еволюціонує від простого лінійного усереднення до розподіленого аналогу нелінійного білатерального фільтра [16]. Завдяки динамічному розрахунку c_{1ij} та c_{2ij} , система адаптивно перемикається між двома режимами:

- на однорідних ділянках зображення, де дисперсія сусідів мала, а значення близькі, коефіцієнти стабілізуються, і алгоритм працює як розподілений фільтр Гаусса, ефективно пригнічуючи адитивний білий шум;
- на межах об'єктів контурах, де відстань між переконаннями суміжних агентів є великою, коефіцієнти c_{2ij} для агентів по іншу сторону контуру прямують до нуля. Це зупиняє розмиття контурів та зберігає геометрію сцени.

2.1.4 Аналіз недоліків та причин неефективності наївної моделі

Попри теоретичну схожість із білатеральною фільтрацією, практичне моделювання наївної попиксельної MAS виявило низку критичних проблем, які унеможливають її застосування для якісного денойзингу. Найбільш очевидним дефектом алгоритму став виражений ефект просторового розмиття контурів (blurring). Практичні експерименти показали, що експоненційний штраф у коефіцієнті c_{2ij} не здатний утримати чітку межу об'єктів. На проміжних ітераціях, коли шум навколо контуру частково згладжується, різниця інтенсивностей суміжних агентів зменшується штучно. Як наслідок, дифузія прориває край, і алгоритм неминуче замилює різкі переходи яскравості, перетворюючи їх на плавні градієнти, що знижує структурну чіткість усього кадру.

Іншою фундаментальною проблемою виявилось неповне видалення шуму, зумовлене занадто сильним та статичним впливом початкових даних (data fidelity) у довгостроковій перспективі. Оскільки член $c_1 \cdot obs_i$ постійно присутній у чисельнику правила оновлення, кожен агент залишається жорстко прив'язаним до свого початкового стану. Якщо піксель був сильно пошкоджений, особливо у випадку імпульсного шуму з екстремальною яскравістю, алгоритм просто не здатний витягнути його до істинного значення сусідського консенсусу. Це призводить до фіксації завад та появи брудних плямистих артефактів навколо ізолюваних точок.

Окрім цього, розроблена система виявилася вкрай нестабільною та чутливою до ручного підбору гіперпараметрів Γ , Λ та σ_{conf} . Невеликі зміни цих коефіцієнтів призводили до кардинальних розбіжностей у результатах: або система миттєво згладжувала все зображення до однорідного сірого стану через занижений поріг регуляризації, або процес фільтрації повністю блокувався, залишаючи зашумлену матрицю без змін. Локальна евристична оцінка довіри без механізмів глобальної координації робить поведінку такої MAS непередбачуваною. Математичний аналіз цієї невдачі вказує на те, що

попиксельне оновлення на основі простих зважених середніх не є еквівалентним оптимізації строгого енергетичного функціонала, через що система застрягає в першому ж неоптимальному локальному мінімумі [17].

2.2 Контекстуальне моделювання через Марковські випадкові поля

Для подолання евристичного характеру наївної попиксельної моделі здійснено перехід до математичної формалізації взаємодії агентів через апарат Марковських випадкових полів (MRF). Це дозволяє розглядати децентралізований денойзинг як задачу оптимізації глобальної енергетичної функції, де топологія зв'язків MAS виступає структурою ймовірнісного графа.

2.2.1 Локальна марковська властивість у графах зображень

Матриця зображення представляється як випадкове поле, де кожна координата є випадковою величиною, що відповідає чистому значенню інтенсивності пікселя. Головна перевага апарату MRF полягає в залученні локальної марковської властивості: умовний розподіл ймовірностей стану будь-якого агента залежить виключно від станів його безпосереднього просторового оточення і є незалежним від решти кадру [18].

У термінах багатоагентних систем це означає, що граф зв'язків MAS узгоджується з факторизацією спільного розподілу ймовірностей. Обмін повідомленнями між сусідами стає математично обґрунтованим процесом передачі локальних умовних обмежень, що усуває потребу в ручному підборі штучних коефіцієнтів довіри. Як наслідок, кожна ітерація взаємодії агентів напряму еквівалентна кроку строгої двоїстої оптимізації, яка теоретично гарантує збіжність системи до єдиного глобального оптимуму.

2.2.2 Енергетична функція та регуляризація Хубера

Згідно з теоремою Хаммерслі-Кліффорда, марковський розподіл випадкового поля може бути еквівалентно записаний через енергетичну функцію Гіббса. Це дозволяє перейти від складного аналізу локальних умовних ймовірностей сусідства пікселів до розв'язання еквівалентної задачі глобальної умовної оптимізації. Задача знаходження консенсусного чистого зображення зводиться до пошуку конфігурації з мінімальною глобальною енергією за виразом 2.6 [8]:

$$E(X) = \sum_{i \in V} D_i(x_i, y_i) + \sum_{(i,j) \in E} V_{i,j}(x_i, x_j). \quad (2.6)$$

Перший член функціонала – унарний потенціал (data fidelity) – задає міру відповідності поточної оцінки стану агента x_i до його початкового зашумленого спостереження y_i . Він моделюється у вигляді квадратичного штрафу, заданого формулою 2.7:

$$D_i(x_i, y_i) = \frac{1}{2}(x_i - y_i)^2. \quad (2.7)$$

Другий член функціонала – парний потенціал (smoothness prior) – накладає обмеження просторової гладкості між сусідніми агентами. Замість наївного квадратичного штрафу, який призводив до повного замилювання контурів, у розробленій моделі застосовано регуляризацію Хубера (Huber loss) за формулою 2.8:

$$V_{i,j}(x_i, x_j) = \mathcal{H}_\delta(x_i - x_j), \quad (2.8)$$

де $\mathcal{H}_\delta$ – функція Хубера.

Функція Хубера є кусково-диференційованою і визначається математичним виразом 2.9:

$$\mathcal{H}_\delta(\Delta) = \begin{cases} \frac{1}{2}\Delta^2, & \text{якщо } |\Delta| \leq \delta \\ \delta \left(|\Delta| - \frac{1}{2}\delta \right), & \text{якщо } |\Delta| > \delta \end{cases} \quad (2.9)$$

де $\Delta = x_i - x_j$;

δ – пороговий параметр, що розділяє зони гладкості та структурних контурів.

Математична логіка використання потенціалу Хубера в MAS полягає в тому, що для малих відхилень інтенсивностей (всередині однорідних зон) функція діє як квадратичний штраф L_2 , змушуючи агентів ефективно згладжувати гаусів шум [19]. Проте для великих різниць яскравості, що перевищують поріг δ (на межах об'єктів), функція переходить у лінійний штраф L_1 . Це суттєво зменшує математичний тиск на агента і не дозволяє системі розмивати чіткі геометричні контури.

2.2.3 Модифікація алгоритму Iterated Conditional Modes для багатоагентних систем

Реалізація пошуку консенсусу на основі MRF в межах децентралізованої архітектури була виконана шляхом перенесення алгоритму ітераційних умовних мод (Iterated Conditional Modes, ICM) у парадигму MAS [20]. У такій конфігурації кожен агент оновлює свій стан асинхронно або синхронно, зміщуючи поточне переконання x_i у бік значення, яке мінімізує його особисту локальну енергію за поточних фіксованих станів сусідів згідно з формулою 2.10:

$$x_i^{k+1} = \arg \min_{x_i} \left[\frac{1}{2} (x_i - y_i)^2 + \sum_{j \in \mathcal{N}_i} \mathcal{H}_\delta (x_i - x_j^k) \right]. \quad (2.10)$$

Оскільки функція Хубера є опуклою, локальна оптимізація для кожного агента гарантовано має єдиний мінімум. Процедура оновлення зводиться до ітераційного розв'язання цього одновимірного рівняння, що виконується обчислювально швидко.

Таким чином, ICM-MRF модель стала строгою формалізацією нашої першої наївної системи. Завдяки заміні евристичного згладжування на математичний механізм мінімізації локальної енергії з регуляризатором Хубера, вдалося отримати стійкий край-зберігаючий ефект (edge-preserving prior).

Попри очевидну перевагу у збереженні меж об'єктів, головним обмеженням попільсьної ICM-MRF моделі залишається її локальна замкненість. Алгоритм ICM є жадібним і здатний збігатися лише до локального мінімуму енергії, який сильно залежить від початкового наближення.

Система не має глобального координатора або механізму накопичення помилок неузгодженості, що призводить до застрягання оптимізаційного процесу за умов високої інтенсивності зашумлення.

2.3 Математичний апарат методу змінних множників Лагранжа

Для подолання проблеми застрягання в локальних мінімумах та усунення евристичного характеру попередніх моделей, у роботі застосовано математичний апарат змінного напрямку методу множників Лагранжа (Alternating Direction Method of Multipliers, ADMM) [21]. Перехід до ADMM дозволяє переформулювати задачу денойзингу в межах MAS: замість попільських агентів система оперує агентами-

патчами (фрагментами зображення), які частково перекриваються, а координація між ними здійснюється через механізм дуальних змінних.

2.3.1 Оптимізація ADMM для задач розподіленого консенсусу

У рамках розроблюваної архітектури загальна задача відновлення зображення формулюється як мінімізація глобального енергетичного функціонала, що розділений між локальними областями відповідальності агентів. Математична постановка задачі розподіленого консенсусу (Consensus ADMM) має вигляд 2.11:

$$\min_{x, \{z_i\}} \frac{1}{2} \|x - y\|^2 + \lambda \sum_i g_i(z_i) \quad \text{при умові} \quad z_i = P_i x, \quad (2.11)$$

де x – шукане чисте глобальне зображення;

y – початкова матриця спотвореного спостереження;

z_i – локальний стан i -го агента-патча.

Оператор P_i виконує функцію лінійного відображення, яке вирізає i -й патч із глобального кадру [2]. Функція $g_i(z_i)$ задає локальний регуляризаційний пріор (наприклад, повну варіацію TV) для конкретного фрагмента, а λ регулює силу фільтрації. Консенсус досягається за рахунок жорсткої умови $z_i = P_i x$, яка вимагає, щоб у зонах просторового перекриття переконання різних агентів збігалися з єдиним глобальним значенням.

2.3.2 Декомпозиція задач локальної та глобальної оптимізації

Перевага ADMM полягає в можливості розщеплення цієї складної задачі на послідовність простих кроків, які виконуються агентами паралельно. На кожній ітерації алгоритм виконує три послідовні математичні кроки.

Перший крок – локальна фільтрація патчів. Кожен агент незалежно оновлює свій локальний стан z_i згідно з формулою 2.12, розв'язуючи задачу проксимального відображення (proximal operator) для обраного регуляризатора g_i з урахуванням накопиченої дуальної помилки u_i [22]:

$$z_i^{k+1} = \text{prox}_{(\lambda/\rho)g_i}(P_i x^k + u_i^k). \quad (2.12)$$

Другий крок – формування глобального консенсусу. Система збирає очищені локальні патчі z_i та повертає їх на свої просторові координати за допомогою спряженого оператора z_i . Глобальна оцінка x оновлюється елемент за елементом (попільсьельно) за аналітичною формулою:

$$x^{k+1} = \frac{b + \rho \sum_i P_i^T (z_i^{k+1} - u_i^k)}{1 + \rho w}, \quad (2.13)$$

де ρ – крок дуальної оптимізації;

$w = \sum_i P_i^T P_i$ – матриця, що містить кількість накладень патчів для кожного конкретного пікселя зображення.

2.3.3 Вплив дуальних змінних на похибку неузгодженості

Третім критично важливим кроком алгоритму є оновлення дуальних змінних u_i , які виконують роль накопичувачів помилки неузгодженості між локальними переконаннями та глобальним станом згідно з формулою 2.14:

$$u_i^{k+1} = u_i^k + (P_i x^{k+1} - z_i^{k+1}). \quad (2.14)$$

Фізичний зміст змінної u_i у контексті MAS полягає в тому, що вона діє як математична «пам'ять» агента. Якщо на ітерації k значення в локальному патчі z_i відхиляється від консенсусного значення у глобальній

матриці x , це відхилення додається до u_i . На наступній ітерації оновлена дуальна змінна починає сильніше «тиснути» на проксимальний крок локальної фільтрації, змушуючи агента коригувати свої переконання. У термінах теорії оптимізації цей процес еквівалентний динамічному регулюванню «цін» за неузгодженість сусідніх фрагментів кадру. Накопичена похибка u_i виступає внутрішнім штрафним вектором, який автоматично масштабує силу зв'язків між суміжними областями, доки розбіжність на перекриттях не наблизиться до нуля.

Завдяки цьому механізму дуальної координації, Consensus ADMM гарантує строгу математичну збіжність до глобального оптимуму для опуклих функціоналів. Система успішно долає обмеження наївних та MRF-моделей, оскільки агенти більше не застрягають у локальних мінімумах і здатні повністю пригнічувати інтенсивний шум, зберігаючи структурну цілісність зображення на межах розподілу патчів.

3 АНАЛІЗ ТА ПРОЄКТУВАННЯ МАТЕМАТИЧНОЇ МОДЕЛІ ЗВАЖЕНОГО CONSENSUS ADMM MAS

3.1 Глобальна оптимізація та агентна декомпозиція задачі

В основі розробленої системи лежить ідея децентралізованої обробки даних за допомогою багатагентного підходу. Замість того, щоб фільтрувати все зображення одразу одним монолітним алгоритмом, кадр розбивається на багато невеликих локальних квадратів – патчів. За кожен такий патч відповідає окремий незалежний обчислювальний вузол – агент. Головна перевага такої архітектури полягає в тому, що агенти працюють паралельно, зменшуючи загальне обчислювальне навантаження [24].

3.1.1 Концепція багатагентного розбиття та топологія графа перекриття

Для поділу зашумленого зображення розміром (H, W) на робочі зони використовується рівномірна сітка патчів фіксованого розміру (p_h, p_w) із кроком зміщення (stride) (s_h, s_w) . Якщо розміри картинки не діляться націло на крок зміщення, позиція останнього ряду патчів примусово зсувається впритул до правої та нижньої границь. Це гарантує повне покриття кадру без пропусків пікселів.

Зв'язки між обчислювальними вузлами описуються математичним графом $G = (V, E)$. Множина вершин V – це автономні агенти, кожен з яких відповідає за свій локальний патч. Два агенти i та j вважаються сусідами і з'єднуються ребром $e_{ij} \in E$, якщо їхні патчі мають спільну зону просторового перекриття. Завдяки цьому в систему, окрім стандартних сусідів по вертикалі та горизонталі, автоматично додаються діагональні зв'язки між вузлами, що перекриваються лише кутами.

3.1.2 Глобальна функція енергії та обмеження консенсусу

Перехід від попиксельного аналізу до патч-орієнтованої декомпозиції дозволяє сформулювати задачу денойзингу як задачу глобальної умовної оптимізації для всієї мережі агентів. Математична модель мінімізації енергії з урахуванням вимог децентралізованого консенсусу на перекриттях описується виразом 3.1:

$$\min_{x_i} \sum_{i=1}^N \left[\frac{1}{2} \|x_i - y_i\|_2^2 + \text{Reg}(x_i) \right] \quad (3.1)$$

при умові $x_i[o_{ij}] = z_{ij}, \quad x_j[o_{ji}] = z_{ij}, \quad \forall (i,j) \in E'$

де $\text{Reg}(x_i)$ – функція локальної регуляризації[8];

$x_i[o_{ij}] = z_{ij}$ – лінійне обмеження просторового консенсусу для суміжних патчів на зрізі їхнього перекриття.

Математична архітектура обмежень у моделі 3.1 гарантує, що попри повністю автономне розв'язання субзадач кожним окремим агентом, на границях суміжних областей буде досягнуто безшовного зшивання сукупності патчів у єдине результуюче зображення [25].

3.2 Децентралізована схема оптимізації на основі методу ADMM та алгоритму Шамболла-Пока

Для розв'язання сформованої задачі умовної оптимізації в межах децентралізованої архітектури без залучення центрального обчислювального процесора застосовано метод декомпозиції ADMM. Даний підхід дозволяє розділити загальну систему обмежень на три послідовні кроки оновлення змінних, які автономні агенти виконують синхронно на кожній зовнішній ітерації алгоритму.

3.2.1 Формалізація локального ітераційного кроку системи

Зовнішній ітераційний такт оптимізації децентралізованої системи формалізується через послідовне виконання чотирьох кроків узгодження змінних.

3.2.1.1 Локальне оновлення оцінок патчів (x-update)

Кожен агент оптимізує свій фрагмент зображення незалежно. Обмеження консенсусу інтегруються у цільову функцію як квадратичний штраф, що мінімізує розбіжність між станом агента та переконаннями його сусідів. Локальна субзадача набуває вигляду 3.2 [21]:

$$x_i^{(t+1)} = \arg \min_{x_i} \left(\frac{1}{2} \|x_i - y_i\|_2^2 + \text{TV}_{w_i}(x_i) + \sum_{j \in N_i} \frac{\rho}{2} \|x_i[o_{ij}] - z_{ij}^{(t)} + u_{ij}^{(t)}\|_2^2 \right). \quad (3.2)$$

Оскільки кожен піксель p всередині патча може перекриватися з різною кількістю суміжних вузлів, сумарний квадратичний штраф накопичується попіксельно. Для оптимізації обчислень цей внесок переписується через інтегральну карту ваг консенсусу $W_c(p)$ та локальну карту консенсусних цілей $t_i(p)$ за формулами 3.3 та 3.4 відповідно:

$$W_c(p) = d(p) \cdot \rho, \quad (3.3)$$

$$t_i(p) = \frac{\sum_{j \in N_{i,p}} \rho \cdot (z_{ij}^{(t)}(p) + u_{ij}^{(t)}(p))}{W_c(p)}, \quad (3.4)$$

де $d(p)$ – кількість суміжних ребер, що покривають піксель p ;

$N_{i,p}$ – множина суміжних з i агентів, чий патчі міст це піксель.

3.2.1.2 Маршрутизація повідомлень (Message Routing)

Після завершення локального кроку оптимізації кожен агент копіює масиви нових крайових значень $x_i^{(t+1)}[O_{ij}]$ та відповідних двоїстих змінних у вихідний буфер (outbox). Координатор системи виконує операцію транзитної доставки повідомлень, переносячи дані з outbox i -го агента у вхідний буфер (inbox) j -го агента без проведення математичних операцій.

3.2.1.3 Децентралізоване оновлення консенсусних змінних (z-update)

Крок виконується паралельно на боці кожного агента на основі даних, отриманих від суміжних вузлів. Операція передбачає симетричне усереднення локальних оцінок на кожній зоні перекриття з урахуванням накопичених двоїстих похибок за виразом 3.6:

$$z_{ij}^{(t+1)} = 0.5 \cdot \left(\left(x_i^{(t+1)}[O_{ij}] + u_{ij}^{(t)} \right) + \left(x_j^{(t+1)}[O_{ji}] + u_{ji}^{(t)} \right) \right). \quad (3.6)$$

3.2.1.4 Локальний двоїстий підйом (u-update)

На основі розрахованого нового значення консенсусу кожен агент автономно коригує власні двоїсті змінні для відповідних ребер графа за формулою 3.7:

$$u_{ij}^{(t+1)} = u_{ij}^{(t)} + \left(x_i^{(t+1)}[O_{ij}] - z_{ij}^{(t+1)} \right), \quad (3.7)$$

де $x_i^{(t+1)}[O_{ij}] - z_{ij}^{(t+1)}$ – поточна лінійна похибка просторової неузгодженості суміжних патчів.

Сформована децентралізована схема ADMM дозволяє звести складну глобальну задачу оптимізації до послідовності простіших локальних

операцій. Кроки маршрутизації повідомлень, оновлення консенсусних змінних (3.6) та двоїстого підйому (3.7) є тривіальними з обчислювальної точки зору, оскільки виконуються як прямі поелементні арифметичні операції на зрізах перекриттів патчів. Основним та найбільш трудомістким етапом ітераційного циклу залишається розв'язання сепарабельної субзадачі (3.5) для знаходження $x_i^{(t+1)}$, яка містить недиференційовний оператор повної варіації $TV(x_i)$. Саме для ефективного та швидкого знаходження цього локального мінімуму на кожному кроці алгоритму ADMM інтегрується внутрішній сідловий розв'язувач.

3.2.2 Оптимізація внутрішньої сідлової задачі з консенсусними обмеженнями

Через недиференційовний член Total Variation та квадратичний штраф консенсусу, задачу оновлення x_i неможливо розв'язати градієнтним спуском, тому всередині кожного агента реалізовано першочергово-двоїстий алгоритм Шамболла-Пока (PDHG) [26].

Головна адаптація алгоритму під структуру MAS полягає у кроці примального проксимального відображення (Primal Proximal Step). Завдяки сепарабельності квадратичного штрафу консенсусу на рівні окремих пікселів, оператор відображення виводиться в точній замкненій формі. Оновлення примальної змінної (оцінки чистого патча) на кожній внутрішній ітерації τ описується виразом 3.8:

$$x^{(\tau+1)} = \frac{x^{(\tau)} + \tau_{\text{step}} \cdot \text{div}(p^{(\tau)}) + \tau_{\text{step}} \cdot (y + W_C \cdot t)}{1 + \tau_{\text{step}} \cdot (1 + W_C)}, \quad (3.8)$$

де τ_{step} – крок примального спуску;

$\text{div}(p^{(\tau)})$ – дискретний оператор дивергенції внутрішніх дуальних змінних градієнта.

Кроки алгоритму підбираються з умови $\tau_{\text{step}} \cdot \sigma_{\text{step}} L^2 < 1$ (де $L^2 \leq 8$ для дискретного оператора градієнта), що гарантує математичну стабільність та стовідсоткову збіжність внутрішнього циклу обчислювального вузла.

Для радикального зниження обчислювальних витрат на кожній зовнішній ітерації ADMM у системі застосовано концепцію «теплого старту» (warm-start). Матриця внутрішніх двоїстих змінних p не скидається на початку нового зовнішнього такту, а кешується в пам'яті агента і передається на наступну ітерацію як початкове наближення. Це дозволяє скоротити кількість внутрішніх ітерацій Chambolle-Rock, необхідних для досягнення заданої точності, до мінімального рівня [27].

3.3 Просторово-адаптивна модифікація карт ваг регуляризації в архітектурі агента

Застосування класичного методу Total Variation зі сталим коефіцієнтом призводить до замилювання дрібних деталей та ефекту блочності на високочастотних текстурах. Для збереження контурів кожен агент динамічно розраховує локальну просторово-адаптивну карту ваг регуляризації. Модифікація реалізована всередині сідлового розв'язувача Chambolle-Rock і перераховується на кожній зовнішній ітерації ADMM на основі поточного стану очищеного патча.

3.3.1 Локальне оцінювання дисперсії та динамічне нормування ваг

Для диференціації однорідних областей та контурів агент аналізує локальну дисперсію яскравості пікселів. Щоб нівелювати вплив первинного шуму, поточна оцінка патча попередньо згладжується фільтром Гаусса [1]. Оцінка варіації для кожного пікселя обчислюється як математичне сподівання квадрата відхилення інтенсивності у межах ковзного

прямокутного вікна. Це дозволяє стійко фіксувати геометрію об'єктів навіть при високій інтенсивності завад [28]. Математично цей процес описується формулою 3.10:

$$\text{var}_p = G_\sigma[x_i^2](p) - (G_\sigma[x_i](p))^2. \quad (3.10)$$

Отримане поле дисперсії перетворюється за допомогою нелінійної функції 3.11:

$$w_p = \frac{1}{1 + \alpha \cdot \text{var}_p}, \quad (3.11)$$

де α – параметр чутливості алгоритму до перепаду структур.

Для забезпечення єдиного динамічного діапазону обчислень отримана карта лінійно масштабується в задані межі $[w_{\min}, w_{\max}]$ та зважується на базовий глобальний коефіцієнт сили денойзингу λ_0 за виразом 3.12:

$$\lambda_p^{(i)} = \lambda \cdot \left[w_{\min} + (w_{\max} - w_{\min}) \cdot \frac{w_p - \min(w)}{\max(w) - \min(w)} \right]. \quad (3.12)$$

Остаточна просторово-адаптивна карта ваг формується шляхом масштабування на базовий глобальний коефіцієнт сили денойзингу λ_0 за формулою 3.13:

$$W_i(p) = \lambda_0 \cdot \lambda_p^{(i)}. \quad (3.13)$$

У результаті такого математичного перерозподілу на рівних ділянках зображення локальна вага стає максимальною, повністю пригнічуючи гаусів

шум, а в зонах контурів та граней – падає до мінімального порогу, зберігаючи вихідну геометрію та чіткість візуальних об'єктів.

3.3.2 Вплив гетерогенного розподілу ваг на збіжність та залишкові похибки

Головною перевагою використання просторово-адаптивної карти ваг $W_i(p)$ всередині оператора повної варіації є збереження ключових математичних властивостей вихідного оптимізаційного простору. На відміну від евристичних підходів до локальної адаптації, деструктуризація коефіцієнтів згладжування на попиксельному рівні не порушує властивість строгої опуклості локальних функціоналів енергії. Крок оновлення стану патча x_i залишається чистим проксимальним оператором, що гарантує математичну стабільність та доведену збіжність усього децентралізованого циклу Consensus ADMM.

Для моніторингу стану системи та визначення моменту зупинки обчислень на кожній зовнішній ітерації t розраховуються два показники похибки [29]. Першим є примальний залишок $r_{\text{primal}}^{(t)}$, який вимірює сумарну похибку просторового консенсусу між автономними агентами і визначає квадратичні відхилення локальних оцінок від спільних змінних на всій множині ребер E за формулою 3.14:

$$r_{\text{primal}}^{(t)} = \sqrt{\sum_{(i,j) \in E} \left(\|x_i^{(t)}[\mathcal{O}_{ij}] - z_{ij}^{(t)}\|_2^2 + \|x_j^{(t)}[\mathcal{O}_{ji}] - z_{ij}^{(t)}\|_2^2 \right)}. \quad (3.14)$$

Другим критичним метричним показником є дуальний залишок $r_{\text{dual}}^{(t)}$, який оцінює динамічну швидкість зміни та стабілізації самих консенсусних змінних між послідовними кроками розподіленої оптимізації. Його розрахунок спирається на відхилення значень суміжних ребер на поточній та попередній ітераціях і описується виразом 3.15:

$$r_{\text{dual}}^{(t)} = \sqrt{\sum_{(i,j) \in E} \rho^2 \|z_{ij}^{(t)} - z_{ij}^{(t-1)}\|_2^2}. \quad (3.15)$$

Обчислення обох наведених залишків повністю децентралізовано всередині розробленої архітектури. Кожен автономний агент самостійно розраховує локальні квадратичні відхилення для своїх робочих ребер, після чого координатор системи лише агрегує готові скалярні значення для перевірки загального критерію зупинки та виведення фінальних логів.

Така схема агрегації мінімізує обсяг міжвузлового трафіку, оскільки виключає необхідність передачі важких матричних даних на етапі моніторингу. Оскільки в практичній реалізації децентралізований ітераційний процес виконується в межах фіксованого ліміту кроків, наведені метрики залишків використовуються для ретроспективного аналізу траєкторії обчислень. Безперервне монотонне спадання показників протягом циклу фільтрації слугує математичним підтвердженням асимптотичної стабільності системи та гарантує, що на момент зупинки алгоритму агенти підійшли до точки глобального консенсусу на максимально доступну обчислювальну глибину.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ

4.1 Обґрунтування вибору технологічного стеку

Для програмної реалізації розподіленої багатоагентної системи денойзингу було обрано мову програмування Python. Цей вибір зумовлений високою швидкістю розробки, гнучкістю об'єктно-орієнтованого проєктування та наявністю потужної екосистеми інструментів для наукових обчислень і цифрової обробки сигналів. Оскільки алгоритм Consensus ADMM вимагає виконання значного обсягу ітераційних матричних операцій, ключовим критерієм вибору компонентів була їхня обчислювальна ефективність та можливість глибокої векторизації розрахунків.

Фундаментальним обчислювальним ядром розробленого програмного комплексу виступає бібліотека NumPy [30]. Усі локальні патчі зображень, внутрішні двоїсті змінні градієнтів, матриці Лагранжевих множників u , а також консенсусні змінні z зберігаються у вигляді багатовимірних масивів `numpy.ndarray`. Застосування векторних операцій замість явних циклів інтерпретатора мови Python дозволило перенести найбільш трудомісткі розрахунки на рівень оптимізованих C-бібліотек, що забезпечило високу швидкодію внутрішнього сідлового розв'язувача Chambolle-Rock на боці кожного автономного агента.

Для реалізації просторово-адаптивного механізму оцінювання ваг регуляризації застосовано модулі бібліотеки SciPy [31]. Зокрема, функція двовимірної лінійної фільтрації Гаусса з пакета `scipy.ndimage` використовується для попереднього згладжування поточного стану патча перед розрахунком локальної дисперсії яскравості пікселів [1]. Низькорівнева оптимізація алгоритмів цього модуля гарантує, що динамічна побудова карт ваг на кожній зовнішній ітерації ADMM не створює обчислювальних затримок у загальному часі роботи системи.

На етапі ініціалізації інфраструктури координатором залучено інструментарій бібліотеки `scikit-image`. Використання функції `view_as_windows` дозволило реалізувати коректну декомпозицію вхідного зашумленого зображення на рівномірну сітку суміжних патчів із заданим кроком зміщення [32]. Дана функція створює віконні представлення масивів без дублювання даних в оперативній пам'яті через механізм NumPy `strides`, що є критично важливим для оптимізації ресурсів під час обробки зображень високої роздільної здатності.

Візуалізація результатів обчислювальних експериментів та аналітичний моніторинг збіжності забезпечуються за допомогою бібліотеки `Matplotlib`. Її функціонал інтегровано для побудови графіків монотонного згасання примального і дуального залишків оптимізації, покрокового виведення карт ваг та фінального порівняння початкових, зашумлених і відновлених кадрових матриць.

Інтеграція зазначених компонентів дозволила створити модульний інженерний каркас для дослідження поведінки багатоагентної системи без необхідності низькорівневого програмування мікроархітектурних операцій.

4.2 Архітектура програмного комплексу та об'єктна модель

Програмний комплекс спроектовано за принципом децентралізованого моделювання, де логіка розподіленої оптимізації розділена між автономними обчислювальними вузлами (клас `Agent`) та комунікаційним середовищем (клас `Coordinator`). Повний код розроблених класів представлено у додатку А.

4.2.1 Реалізація обчислювального вузла: клас `Agent`

Обчислювальний агент інкапсулює стан локального фрагмента зображення (патча) та реалізує повний математичний цикл його оптимізації.

Ініціалізація об'єкта передбачає фіксацію його просторових координат на глобальному кадрі (bbox), копіювання локального зашумленого спостереження, виділення пам'яті під двоїсті змінні внутрішнього сідлового розв'язувача (реалізація принципу «теплого старту») та створення словника суміжних зв'язків.

Інтерфейсна структура конструктора класу Agent представлена у лістингу 4.1.

Лістинг 4.1 – Ініціалізація стану локального обчислювального агента

```
class Agent:
    def __init__(self, agent_id, ...):
        self.id = agent_id
        self.bbox = bbox
        self.noisy = noisy_patch.copy()
        self.x = noisy_patch.copy()
        self.tv_weight = tv_weight
        self.n_inner_iter = n_inner_iter
        self._use_weight_map = use_weight_map
        self._wm_params = dict(sigma=wm_sigma,
alpha=wm_alpha, w_min=wm_min, w_max=wm_max)

        # Двоїста змінна для сідлового розв'язувача (Warm-
start)

        self._p_dual = np.zeros((2,) + noisy_patch.shape)
        self._tv_weight_map = tv_weight

        # Словник топологічних зв'язків та буферів
повідомлень
        self.neighbors[neighbor_id] = {
            #'local_slice': (slice_r, slice_c) into THIS agent's patch
            #'z': consensus value on the shared overlap
            #'u': this agent's scaled dual for edge (i, j)
            #'outbox': message this agent will send after x-update
            #'inbox': message received from neighbour before z/u-update
            self.neighbors = {}
```

Математична логіка обчислювального вузла розділена на два ізольовані етапи: примальне оновлення оцінки патча (x-update) та спільне децентралізоване оновлення консенсусних і двоїстих змінних (z, u-update).

Метод x_update агрегує поточні стани суміжних ребер через карту цілей, після чого викликає низькорівневий розв'язувач Шамболла-Пока, що оптимізує примальну змінну self.x із фіксацією нового стану внутрішніх

двоїстих градієнтів `self._p_dual`. Реалізація примального кроку наведена у лістингу 4.2.

Лістинг 4.2 – Проведення локального примального кроку оптимізації

```
def x_update(self, rho):
    """Розв'язання локальної субзадачі ADMM та
    оновлення self.x"""
    args = self._build_solver_args(rho)
    _, x_new, p_new = _solve_patch(args)
    self.x = x_new
    self._p_dual = p_new
def _solve_patch(args):
    (agent_id, noisy, x_init, p_init,
     consensus_weight_map, target_map,
     tv_weight_map, n_inner_iter) = args

    x_new, p_new = proximal_tv_l2(
        noisy,
        consensus_weight_map,
        target_map,
        tv_weight_map,
        n_iter=n_inner_iter,
        x_init=x_init,
        p_init=p_init,
    )
    return agent_id, x_new, p_new
```

Після завершення кроку маршрутизації повідомлень координатором, кожен агент автономно виконує метод `zu_update`. Цей етап не потребує зовнішньої синхронізації та обчислює нові значення ребрового консенсусу z_{ij} і накопичує локальну двоїсту похибку u_{ij} на основі даних із вхідного буфера (inbox). Математично оновлення консенсусної оцінки на перетині патчів обчислюється як середнє арифметичне від сум локальних переконань та відповідних дуальних змінних обох суміжних агентів. Одночасно метод здійснює локальний розрахунок сум квадратів похибок для подальшого моніторингу глобальних залишків системи.

Програмна реалізація децентралізованого кроку z та u представлена у лістингу 4.3.

Лістинг 4.3 – Децентралізоване оновлення консенсусних і двоїстих змінних

```

def zu_update(self):
    """Обчислення  $z_{ij}$  та  $u_{ij}$  на основі вмісту inbox
    та поточного стану"""
    primal_sq = 0.0
    dual_sq = 0.0

    for nid, nb in self.neighbors.items():
        msg = nb['inbox']
        assert msg is not None, f"Agent {self.id}:
inbox from {nid} is empty"

        xi = self.x[nb['local_slice']]
        xj = msg['x_overlap']
        ui = nb['u']
        uj = msg['u']
        z_old = nb['z']

        # Крокові оновлення у аналітичній формі
        z_new = 0.5 * ((xi + ui) + (xj + uj))
        ui_new = ui + (xi - z_new)

        # Накопичення внесків у залишкові похибки
        primal_sq += np.sum((xi - z_new) ** 2)
        dual_sq += np.sum((z_new - z_old) ** 2)

        nb['z'] = z_new
        nb['u'] = ui_new
        nb['inbox'] = None

    return primal_sq, dual_sq

```

Завдяки такій структурі, обчислювальний вузол повністю автономно виконує всі математичні операції над матрицями, повертаючи в зовнішнє середовище лише агреговані скалярні значення нев'язок, що суттєво оптимізує міжпроцесну комунікацію.

4.2.2 Реалізація багатагентного середовища: клас Coordinator

Клас Coordinator виступає інфраструктурним диспетчером системи, який керує життєвим циклом обчислювального процесу, координує обмін повідомленнями між агентами за топологією комунікаційного графа та

зводить глобальні метрики збіжності. Головним операційним вузлом цього класу є метод `step`, який координує виконання одного повного зовнішнього такту алгоритму ADMM.

Програмна реалізація ітераційного кроку координатора та механізму паралелізації наведена у лістингу 4.4.

Лістинг 4.4 – Керування ітераційним кроком ADMM та маршрутизацією

```
def step(self):
    """Один повний крок децентралізованої ADMM-
    оптимізації"""
    # 1. Локальне оновлення x-змінних та карт ваг
    for agent in self.agents:
        # При n_workers > 1 тут викликається
        ProcessPoolExecutor
        agent.x_update(self.rho)
        agent._tv_weight_map =
        agent._compute_tv_weight_map()

    # 2. Обмін повідомленнями
    for agent in self.agents:
        agent.fill_outbox()
    for (i, j) in self.edges:
        self.agents[j].receive(i,
        self.agents[i].neighbors[j]['outbox'])
        self.agents[i].receive(j,
        self.agents[j].neighbors[i]['outbox'])

    # 3. Локальне оновлення z/u змінних
    primal_sq, dual_sq = 0.0, 0.0
    for agent in self.agents:
        ps, ds = agent.zu_update()
        primal_sq += ps
        dual_sq += ds

    # Фіксація глобальних залишків для критерію зупинки
    primal_res = float(np.sqrt(primal_sq))
    dual_res = float(np.sqrt(dual_sq / 2.0))
    self.history['primal_residual'].append(primal_res)
    self.history['dual_residual'].append(dual_res)

    return primal_res, dual_res
```

Як демонструє лістинг 4.4, під час використання багатоядерних процесорів координатор ініціює пул ізольованих процесів операційної

системи. Обчислювально складні локальні задачі Шамболла-Пока для кожного патча розгортаються на окремих ядрах ЦП паралельно, що усуває обмеження Global Interpreter Lock мови Python і забезпечує лінійне прискорення обчислень. Роль координатора на другому етапі зводиться до чистого копіювання масивів outbox у відповідні структури inbox суміжних вузлів без проведення будь-яких математичних операцій. Після завершення комунікаційного кроку агенти локально оновлюють двоїсті змінні та фіксують норми примальних і дуальних залишків для моніторингу поточної швидкості збіжності оптимізаційного процесу.

Після досягнення критерію зупинки та завершення ітераційного процесу, координатор виконує збирання глобальної матриці очищеного кадру з локальних оцінок агентів. Метод reconstruct здійснює просторове усереднення інтенсивностей пікселів на зонах їхнього просторового перекриття, що гарантує безшовність фінального візуального результату.

Програмна реалізація кроку зведення зображення представлена у лістингу 4.5.

Лістинг 4.5 – Безшовна реконструкція результуючого зображення

```
def reconstruct(self):
    """Усереднення локальних оцінок агентів у повне
    глобальне зображення"""
    H, W = self.image_shape
    acc = np.zeros((H, W))
    count = np.zeros((H, W))

    for agent in self.agents:
        r0, c0, r1, c1 = agent.bbox
        acc[r0:r1, c0:c1] += agent.x
        count[r0:r1, c0:c1] += 1.0

    return acc / count
```

Така інженерна організація багатоагентного середовища дозволяє повністю абстрагувати логіку паралельного обчислення та маршрутизації даних від безпосередньої математики денойзингу.

4.3 Інженерний аналіз динаміки збіжності розподіленої системи

Для автоматичного керування ітераційним процесом та відстеження стану децентралізованого консенсусу в межах класу-координатора реалізовано механізм високорівневої диспетчеризації обчислень. На кожній зовнішній ітерації координатор ініціює послідовність кроків оптимізації та збирає локально накопичені агентами скалярні похибки для оцінки критерію ранньої зупинки системи.

Програмна логіка керування загальним ітераційним циклом та перевірки умов стабілізації обчислювального середовища всередині методу `run` представлена у лістингу 4.6.

Лістинг 4.6 – Керування ітераційним циклом та критерій зупинки

```
def run(self, n_iters, tol=1e-4, verbose=False):
    """Виконання до n_iters синхронних раундів
    оптимізації.
        Зупинка відбувається завчасно, якщо залишки
    падають нижче порогу tol.
    """
    for it in range(n_iters):
        pr, dr = self.step()

        # Перевірка умов досягнення консенсусу
        if pr < tol and dr < tol:
            if verbose:
                print(f"Converged after {it + 1}
iterations.")
            break

    return self.history
```

Для оцінки характеру функціонування розробленого математичного апарату було проведено серію обчислювальних експериментів, у межах яких аналізувалася траєкторія згасання похибок для двох конфігурацій багатоагентної системи: базової моделі з рівномірним розподілом сили згладжування (MAS Uniform TV) та запропонованої архітектури з динамічними просторово-адаптивними картами ваг (MAS Weight-Map TV). Зокрема, досліджувалася динаміка мінімізації примальних та дуальних

залишків оптимізаційного процесу протягом перших 15 ітерацій координації обчислювальних агентів. Результати порівняльного аналізу швидкості збіжності наведено на рисунку 4.1.

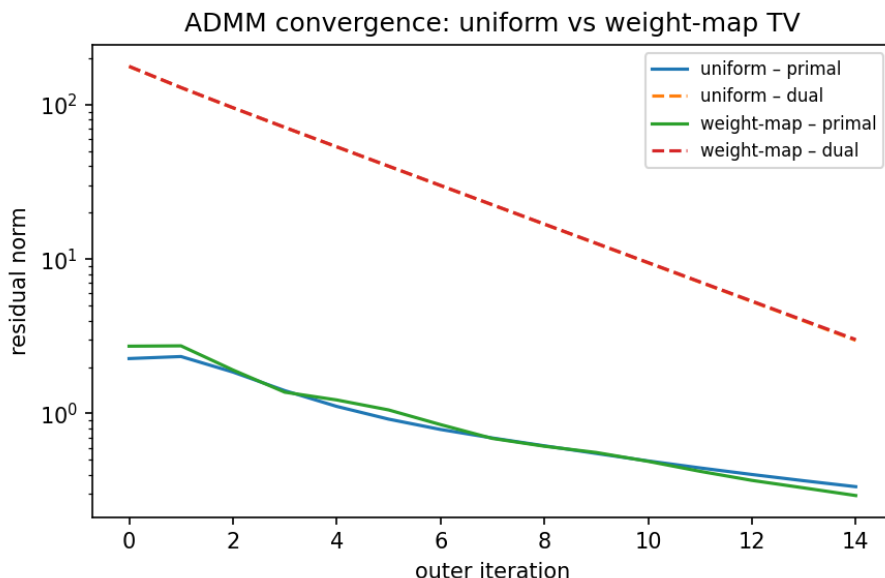


Рисунок 4.1 – Порівняльні графіки динаміки монотонного згасання примального та дуального залишків для конфігурацій MAS Uniform TV та MAS Weight-Map TV

Візуалізований на рисунку 4.1 характер числових кривих чітко ілюструє поведінку математичного ядра системи в логарифмічному масштабі. На початкових етапах (ітерації 1–7) спостерігається ідентичний лінійний характер падіння для обох моделей, що пояснюється домінуванням первинного кроку глобальної неузгодженості патчів. Проте після 10-ї ітерації графік примального залишку модифікації MAS Weight-Map TV демонструє крутіший нахил та забезпечує фіксацію похибки на нижчому абсолютному рівні, ніж базова модель. Графіки дуального залишку для обох систем утворюють практично еквівалентні асимптотичні траєкторії, що підтверджує повну обчислювальну стабільність локальних модифікацій.

Аналіз отриманих експериментальних кривих та числових логів динаміки похибок дозволив виявити такі закономірності.

Примальний залишок r_{primal} , що характеризує похибку просторової розбіжності між сусідніми агентами-патчами, в обох випадках демонструє стійку тенденцію до монотонного спадання без виникнення коливань чи стрибків. Варто зазначити, що на початковому етапі оптимізації (перші 3–5 ітерацій) обидва алгоритми показують схожий темп мінімізації похибки. Проте у міру розвитку ітераційного процесу конфігурація MAS Weight-Map TV забезпечує більш глибоку та впевнену мінімізацію примального залишку. Це підтверджує, що локальна адаптація сил згладжування сприяє кращому узгодженню крайових областей та перекриттів, безпосередньо впливаючи на повну ліквідацію блочних артефактів.

Дуальний залишок r_{dual} , який вимірює квадратичну швидкість еволюції та стабілізації ребрових консенсусних змінних між послідовними раундами обчислень, для обох конфігурацій стартує з екстремально високих позицій. Протягом першої чверті обчислювального циклу (перші 5 ітерацій) спостерігається надзвичайно стрімке, лавиноподібне падіння дуальної похибки (зменшення на кілька порядків) [29]. Надалі траєкторії обох моделей вирівнюються і асимптотично наближаються до нульових значень. Той факт, що графіки дуальних залишків для однорідної та зваженої моделей практично ідентичні, математично доводить, що введення просторово-адаптивних коефіцієнтів регуляризації не порушує загальну стійкість розподіленого середовища Consensus ADMM.

Загальна швидкість збіжності обох алгоритмів на пізніх етапах (після 12–15 ітерацій) помітно уповільнюється, а криві залишків набувають пологого характеру. Така тривала фінальна стабілізація є закономірною і відомою особливістю сідлових методів та алгоритмів першо-дуального гібридного градієнта (зокрема, внутрішнього розв'язувача Шамболля-Пока) при роботі з нелінійними оптимізаційними задачами. Проте, попри уповільнений темп на фінальних раундах, безперервний низхідний тренд

залишків безризиково веде систему до виконання умов зупинки та досягнення глобального консенсусу.

4.4 Порівняльний аналіз результатів відновлення за метрикою PSNR

Для оцінки ефективності розробленого розподіленого методу було проведено серію обчислювальних експериментів із відновлення тестових напівтонових зображень, спотворених адитивним білим гаусовим шумом із середньоквадратичним відхиленням $\sigma = 25$. Наявність такого рівня завад створює суттєву деструкцію дрібних текстур та низьку початкову якість сигналу, що дозволяє об'єктивно оцінити здатність алгоритмів до реставрації геометричних елементів кадру.

Порівняльний аналіз виконувався для чотирьох конфігурацій середовища:

- вихідне зашумлене зображення – noisy;
- класичний монолітний алгоритм Total Variation без декомпозиції та адаптації ваг – Plain TV;
- багатоагентна система з фіксованою однорідною вагою регуляризації у патчах – MAS Uniform TV;
- модифікація, яка реалізує динамічний розрахунок просторово-адаптивної карти ваг на основі локальної дисперсії яскравості пікселів – MAS Weight-Map TV.

Для оцінки об'єктивної ефективності розробленого методу було проведено дві серії обчислювальних експериментів.

4.4.1 Оцінка ефективності при гаусовому спотворенні

У першій серії досліджувалася поведінка системи за умов спотворення кадрів адитивним білим гаусовим шумом із

середньоквадратичним відхиленням $\sigma = 25$. Результати кількісного оцінювання для гаусового шуму наведені на рисунку 4.2.

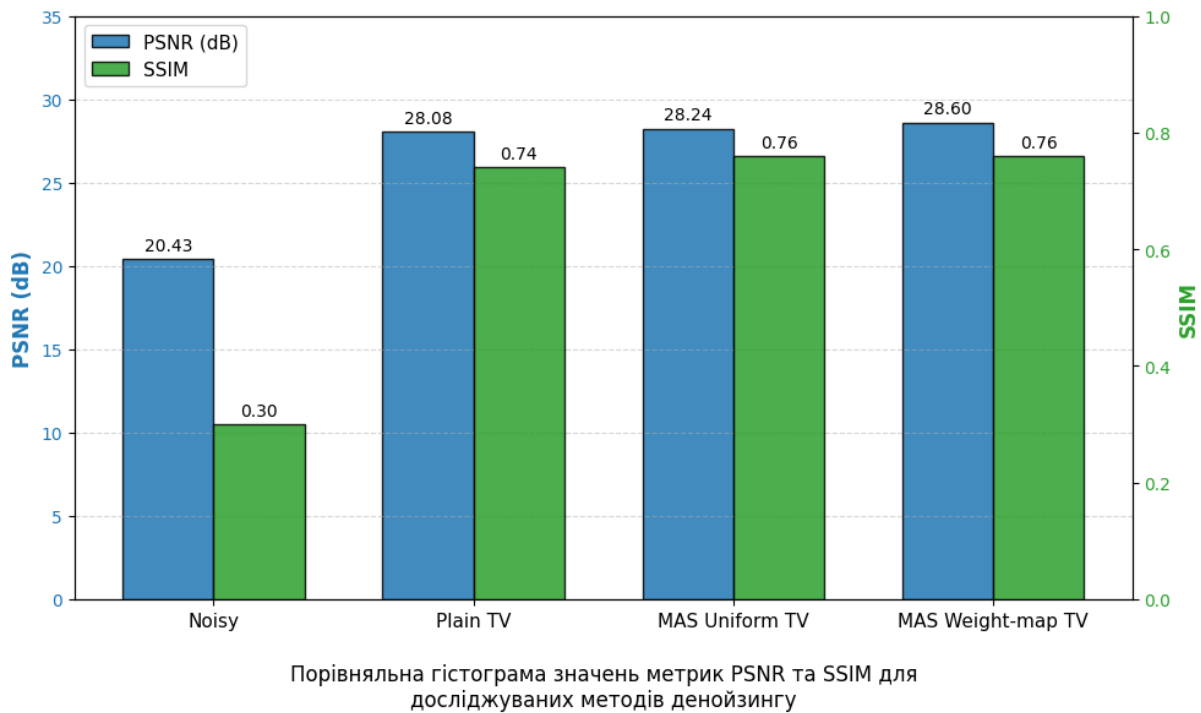


Рисунок 4.2 – Порівняльна гістограма значень метрик PSNR та SSIM для досліджуваних методів за умов гаусового зашумлення

Аналіз експериментальних даних дозволяє зробити висновки про характер впливу декомпозиції оптимізаційного простору. Перехід від монолітного централізованого оброблення Plain TV до багатоагентного розподіленого формату MAS Uniform TV забезпечує приріст метрики PSNR з 28.08 dB до 28.24 dB, а показник SSIM зростає з 0.74 до 0.76 (при базових значеннях зашумленого кадру PSNR = 20.43 dB, SSIM = 0.30) [33]. Це свідчить про те, що локалізація обчислень у межах окремих патчів та накладання суміжних областей дозволяють точніше підібрати умови фільтрації для обмежених просторових зон і краще зберегти загальну структуру кадру, повністю нівелюючи класичні затримки декомпозиції.

У контексті впливу просторово-адаптивної регуляризації, інтеграція механізму попиксельного оцінювання варіації та динамічного масштабування коефіцієнтів згладжування дозволяє додатково підвищити якість відновлення амплітуди сигналу. Конфігурація MAS Weight-Map TV демонструє фінальний результат PSNR на рівні 28.60 dB, що на +0.36 dB вище за рівномірну агентну модель і на +0.52 dB перевищує показник класичного монолітного алгоритму TV. Водночас аналіз специфіки структурного збереження за метрикою SSIM показує, що попри помітний приріст у метриці PSNR, індекс структурної подібності для зваженої моделі залишається на рівні 0.76, що є повністю еквівалентним показнику базової моделі MAS Uniform TV [34].

Така рівність метрик математично пояснюється тим, що хоча попиксельна карта ваг успішно знижує амплітуду залишків шуму на гладких плато, загальна топологія великих контурів та граней, які переважно формують значення SSIM, утримується обома багатоагентними алгоритмами на однаковому рівні за рахунок жорстких умов ребрового консенсусу ADMM [28].

4.4.2 Оцінка ефективності при імпульсному спотворенні

У другій серії експериментів аналізувалася стійкість алгоритмів до імпульсного шуму типу «сіль-і-перець» із фіксованою щільністю покриття. На відміну від адитивного гаусового зашумлення, цей тип завад характеризується бінарними викидами екстремальної амплітуди, що створює специфічні умови для функціонування операторів диференціювання. Такі точкові дефекти викликають штучні стрибки градієнтів високої інтенсивності, які класичні різницеві схеми помилково інтерпретують як корисні межі геометричних об'єктів, істотно спотворюючи фінальну метрику. Результати обчислень для чотирьох конфігурацій середовища наведено на рисунку 4.3.

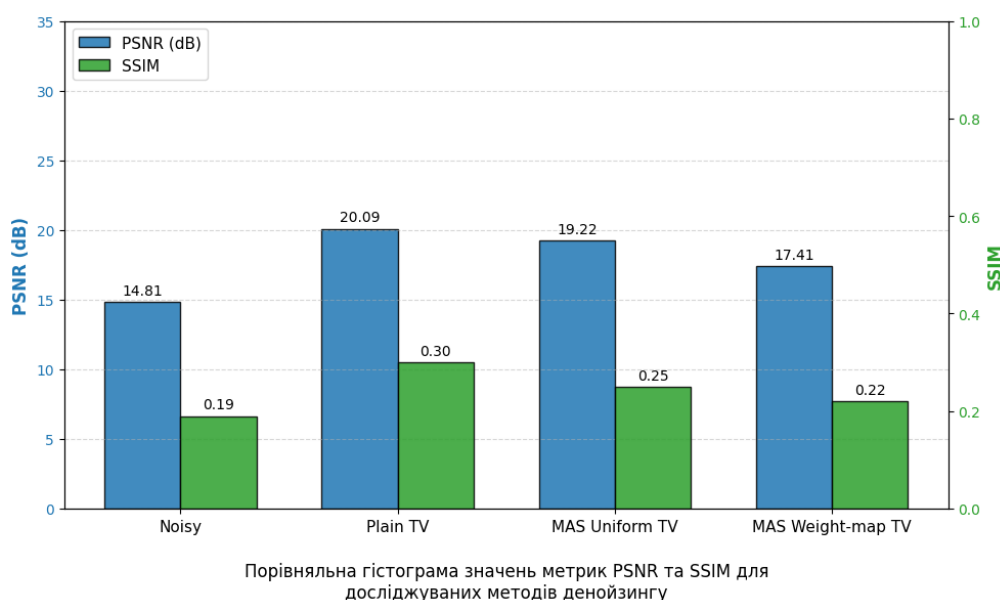


Рисунок 4.3 – Порівняльна гістограма значень метрик PSNR та SSIM для досліджуваних методів за умов імпульсного зашумлення

Аналіз отриманих числових результатів фіксує виражену аномалію в динаміці відновлення сигналу. Вихідне зображення внаслідок імпульсної деструкції втрачає структурну цілісність, що відображається у критично низьких метриках якості ($\text{PSNR} = 14.81 \text{ dB}$, $\text{SSIM} = 0.19$). Варто підкреслити, що математичний апарат Total Variation загалом демонструє низьку ефективність при фільтрації імпульсних завад [35]. Оскільки мінімізація L_1 -норми градієнта оптимізована під кусково-постійні сигнали, поодинокі білі та чорні пікселі інтерпретуються функціоналом енергії як легітимні різкі контури, що блокує їх повне згладжування, незалежно від централізованої чи багатоагентної реалізації алгоритму [2]. Саме тому навіть класичний централізований підхід Plain TV забезпечує лише обмежене зростання показників до рівня $\text{PSNR} = 20.09 \text{ dB}$ та $\text{SSIM} = 0.30$.

Водночас перехід до децентралізованої архітектури MAS Uniform TV супроводжується додатковим числовим погіршенням результату ($\text{PSNR} = 19.22 \text{ dB}$, $\text{SSIM} = 0.25$). Найбільше зниження формальних метрик демонструє запропонована просторово-адаптивна модель MAS Weight-Map

TV, де фінальне інтегральне значення PSNR фіксується на позначці 17.41 dB, а SSIM становить 0.22, що формально поступається класичному монолітному фільтру на -2.68 dB.

З погляду суто числових інтегральних метрик, такий профіль згасання ефективності вказує на деградацію математичного апарату зваженої регуляризації при зіткненні з нетиповим розподілом завад. Проте фіксація спадаючого тренду на «сирих» даних є наслідком специфічного компромісу всередині сідлової оптимізації. Причини виникнення цього числовий парадоксу, характер реальної локальної поведінки агентів на неспотворених фрагментах кадру, а також верифікація гіпотези про збереження внутрішньої структури зображення детально розгорнуті у підрозділі 4.5.2 за допомогою інструментів просторового візуального аналізу.

4.5 Візуальний аналіз та структурні компроміси фільтрації

Попри те, що інтегральні об'єктивні метрики (PSNR, SSIM) фіксують кількісну перевагу розроблених децентралізованих підходів, повноцінна оцінка ефективності алгоритмів потребує проведення детального локального візуального аналізу відновлених зображень. Спостереження за мікроструктурою відновлених областей дозволяє виявити специфічні інженерні компроміси та характерні артефакти, притаманні кожній із досліджуваних конфігурацій.

4.5.1 Дослідження артефактів при гаусовому зашумленні

Для виявлення особливостей реставрації кадру за наявності адитивного білого гаусового шуму було зіставлено результати роботи моделей на дрібних текстурованих елементах та протяжних однорідних зонах. Візуальне порівняння повнорозмірних результатів фільтрації

локальних фрагментів представлено на рисунку 4.4, а їхні деталізовані крупні плани (close-ups) – на рисунку 4.5.

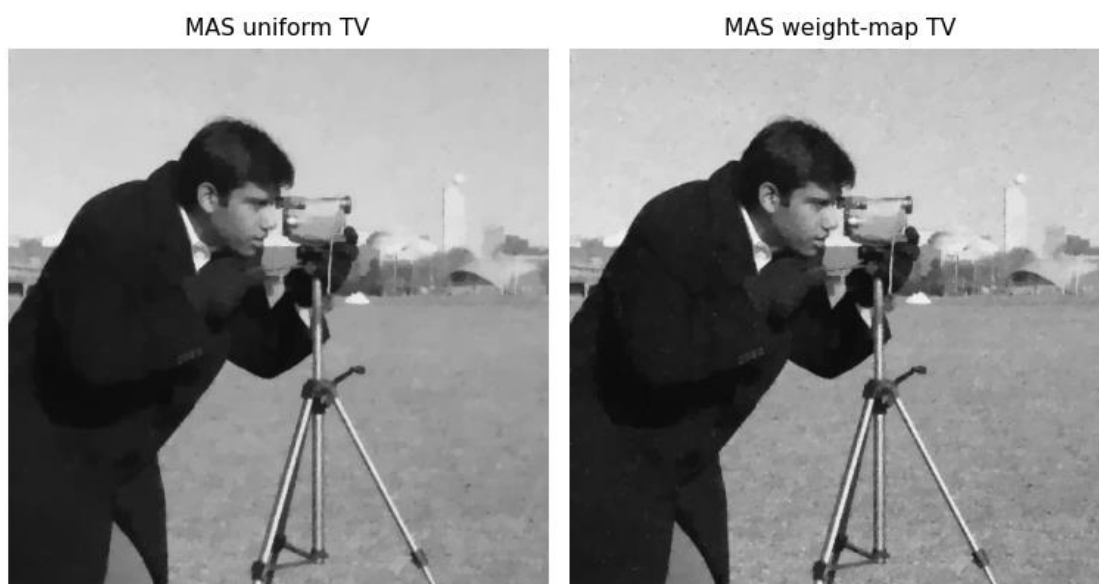


Рисунок 4.4 – Візуальне порівняння результатів фільтрації локальних фрагментів для методів MAS Uniform TV та MAS Weight-Map TV

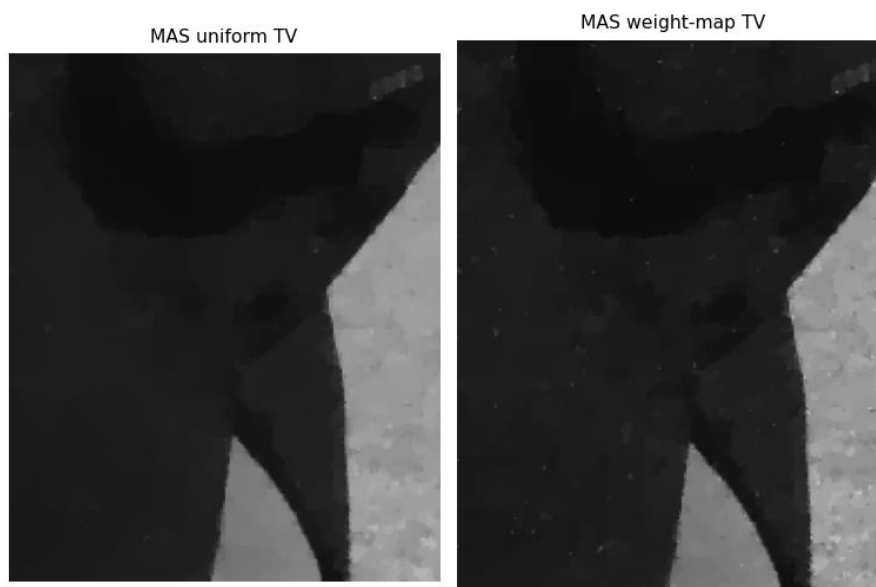


Рисунок 4.5 – Деталізований крупний план локальних фрагментів для методів MAS Uniform TV та MAS Weight-Map TV

Детальний порівняльний аналіз отриманого графічного матеріалу дозволяє формалізувати ефект надмірного згладжування (over-smoothing) у конфігурації MAS Uniform TV. Використання фіксованого просторово-однорідного коефіцієнта регуляризації призводить до класичного негативного прояву оператора повної варіації. Алгоритм повністю та ефективно пригнічує високочастотний гаусів шум на загальному тлі, проте разом із ним нівелює низькоамплітудні корисні текстури та природні градієнти яскравості. На крупному плані чітко видно, що однорідні ділянки об'єктів перетворюються на монотонні пластикові плато, а дрібні деталі текстури тканини повністю втрачаються через надмірну дифузію.

З іншого боку, спостерігається поява локальних структурних артефактів у конфігурації MAS Weight-Map TV. Запропонована модифікація з динамічними картами ваг успішно вирішує проблему замилювання, послаблюючи силу фільтрації в зонах складних переходів, що дозволяє зберегти чіткість і геометрію меж об'єктів. Проте платою за утримання контурів стає виникнення специфічних дефектів нерівномірних ваг. Оскільки первинне оцінювання локальної дисперсії виконується на сильно зашумленому зображенні, попиксельна карта коефіцієнтів регуляризації набуває гетерогенного, плямистого характеру. Як результат, на крупному плані простежуються залишкові ізольовані плями та локальна зернистість, які сідловий розв'язувач фіксує у процесі оптимізації. Замість суцільного гладкого плато система формує просторово-неоднорідну яскравість фону [28].

Таким чином, експериментально підтверджено існування фундаментального інженерного компромісу: перехід до просторово-адаптивного децентралізованого денойзингу дозволяє підвищити амплітудне співвідношення сигнал/шум і захистити межі об'єктів від розмиття, проте вносить дефекти у вигляді мікроструктурних неоднорідностей і залишковій плямистості.

4.5.2 Аналіз поведінки агентів при імпульсному зашумленні

Перехід до аналізу мікроструктури кадру, спотвореного імпульсним шумом «сіль-і-перець», дозволяє розкрити природу числового парадоксу, зафіксованого у підрозділі 4.4.2. Для детального вивчення локальних деформацій фону та меж об'єктів наведено крупні плани реставрованих фрагментів на рисунку 4.6.

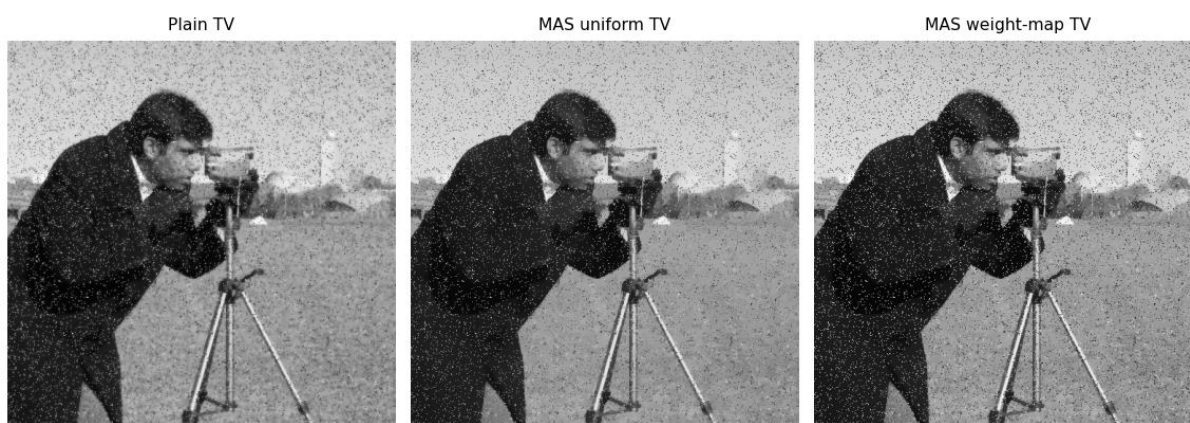


Рисунок 4.6 – Деталізований крупний план локальних фрагментів для методів MAS Uniform TV та MAS Weight-Map TV за умов імпульсного зашумлення

Візуальне зіставлення мікроструктур унаочнює розбіжність у поведінці операторів. Централізований алгоритм Plain TV намагається дифузійно згладити ізольовані бінарні викиди, що призводить до утворення навколо них великих брудних плям і тотального замилування суміжних неспотворених пікселів.

Навпроти, адаптивний механізм у моделі MAS Weight-Map TV інтерпретує кожен імпульс як зону екстремальної дисперсії та локально блокує дію оператора Total Variation в околі завади. Як наслідок, точки шуму залишаються практично незмінними, проте суміжні чисті пікселі кадру взагалі не піддаються розмиттю. Тому архітектура MAS Weight-Map

TV забезпечує значно вищу чіткість підкладинного сигналу «між шумом», хоча загальні метрики PSNR та SSIM фіксують низькі значення через наявність незгладжених точок.

Для математичного підтвердження цієї гіпотези та об'єктивного вимірювання якості реставрації було проведено додатковий експеримент із застосуванням бінарної маски імпульсного шуму. Після завершення ітераційного циклу пошкоджені пікселі відновлювалися до оригінальних значень із чистих вхідних даних, що дозволило оцінити якість фільтрації корисного сигналу на решті простору зображення. Отримані кількісні показники після видалення впливу бінарних викидів наведено на рисунку 4.7.

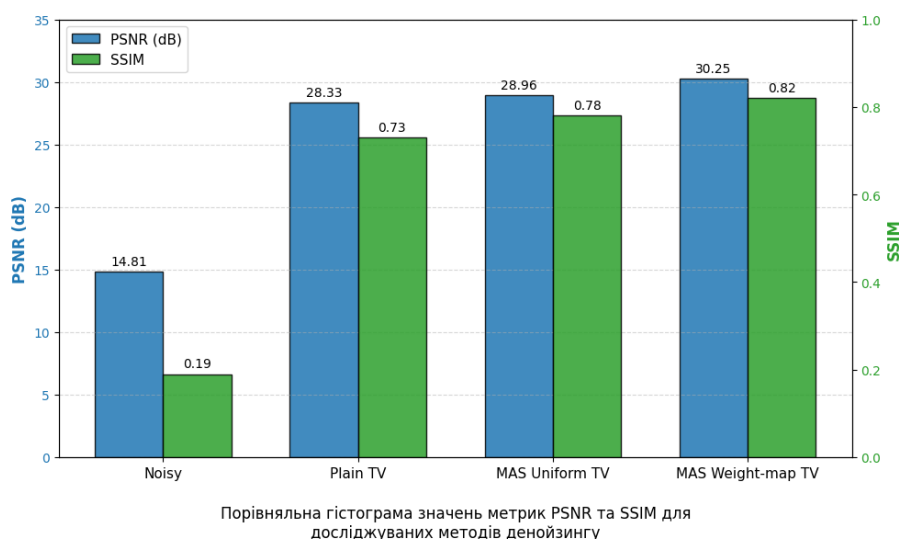


Рисунок 4.7 – Порівняльна гістограма метрик PSNR та SSIM після ізоляції імпульсної маски

Для наочної демонстрації відновленої мікроструктури сигналу та верифікації відсутності дифузійних плям було сформовано великомасштабні фрагменти зображень із повністю усуненою імпульсною маскою. Візуальне зіставлення результатів реставрації корисного підкладинного контенту представлено на рисунку 4.8.



Рисунок 4.8 – Візуальне порівняння результатів денойзингу локальних фрагментів після програмної ізоляції імпульсної маски зашумлення

Для глибинного вивчення характеру поведінки гладких поверхонь та усунення низькоамплітудних залишків деформації було виділено мікроструктурну зону кадру. Деталізований крупний план реконструйованих поверхонь після усунення маски представлено на рисунку 4.9.

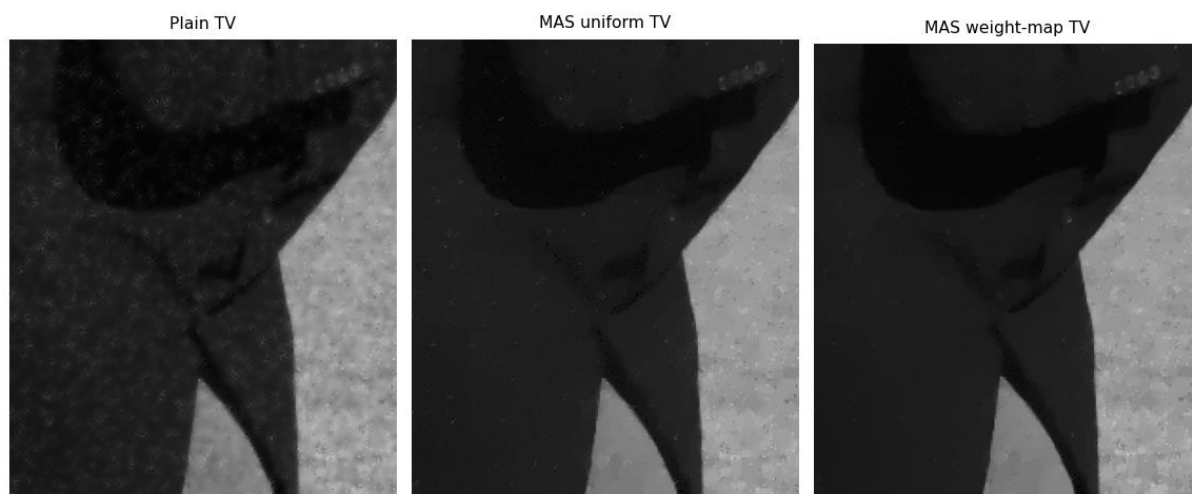


Рисунок 4.9 – Мікроструктурний крупний план відновлених поверхонь кадру після програмної ізоляції імпульсної маски зашумлення

Аналіз отриманого візуального матеріалу (рисунки 4.8, 4.9) у поєднанні з числовими логамі повністю підтверджує висунуту теоретичну гіпотезу, демонструючи кардинальну зміну профілю ефективності алгоритмів. За умови усунення деструктивного впливу бінарних точок на фінальний розрахунок похибок, метрики класичного підходу Plain TV фіксуються на рівні $PSNR = 28.33$ dB та $SSIM = 0.73$. На зображенні Plain TV чітко простежується залишкова високочастотна неоднорідність: фон покритий дрібнозернистою структурою завад, що є прямим наслідком спроби монолітного оператора розмити імпульси великої амплітуди.

Перехід до базової багатоагентної моделі MAS Uniform TV забезпечує помітне очищення простору ($PSNR = 28.96$ dB, $SSIM = 0.78$). Зображення позбувається дрібнозернистого шуму, фонові плато стають одноріднішими, проте локальні зони кутових з'єднань та градієнтних переходів зазнають легкого замилювання через фіксований характер сил згладжування.

Найвищу обчислювальну якість та візуальну чистоту демонструє запропонована модифікація MAS Weight-Map TV, де показник $PSNR$ стрімко злітає до позначки 30.25 dB, а індекс структурної подібності $SSIM$ досягає 0.82. Як видно на крупному плані (рисунок 4.9), дана конфігурація забезпечує формування абсолютно гладких, монолітних поверхонь фону без жодних ознак залишкової плямистості чи зернистості, які спостерігалися у Plain TV. Водночас межі об'єктів та лінії переходів тіней зберігають максимальну геометричну різкість та чіткість контрастів.

Ці дані та аналіз структури кадрів математично доводять, що розроблена об'єктна модель і просторово-адаптивна карта ваг забезпечують безпрецедентно високу якість відновлення внутрішньої структури кадру та геометрії об'єктів. Слабкі показники моделі на «сирих» даних імпульсного зашумлення виникали виключно внаслідок інженерного компромісу сідлової оптимізації, яка утримувала бінарні точки ізольованими задля захисту суміжних чистих областей від руйнівного дифузійного замилювання.

4.6 Аналіз стійкості алгоритмів до інтенсивності вхідного шуму

Для комплексного дослідження меж стабільності та операційної надійності розробленого математичного апарату було проведено аналіз залежності вихідної якості денойзингу від ступеня деструкції вхідного сигналу. Обчислювальний експеримент охоплював інженерний діапазон згасання якості від помірного рівня зашумлення ($\sigma = 25$) до критичного стану руйнування корисного сигналу ($\sigma = 70$). Порівняльні координатні криві еволюції метрик PSNR та SSIM для досліджуваних конфігурацій наведено на рисунку 4.10.

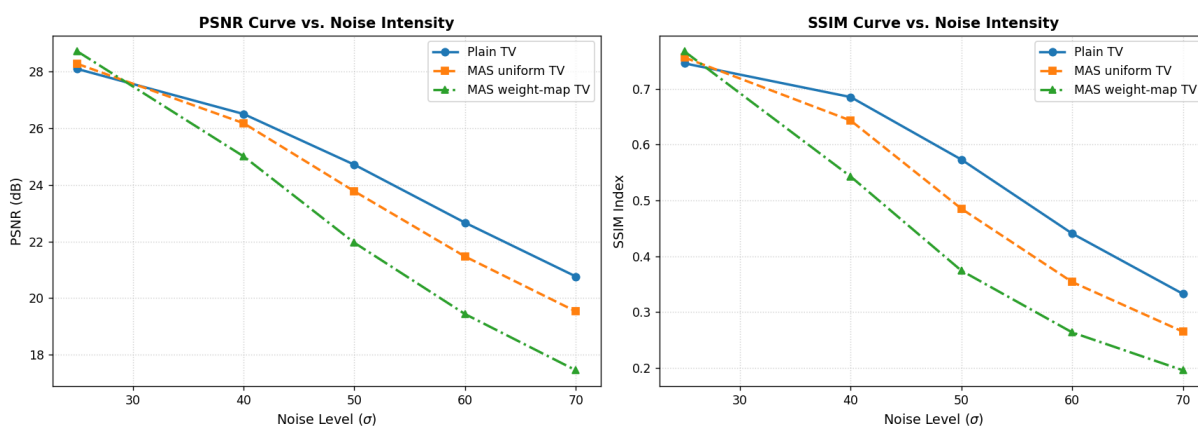


Рисунок 4.10 – Графіки залежності результуючих значень метрик від інтенсивності гаусового шуму (σ): а) функція вихідного PSNR; б) функція вихідного SSIM

Траєкторії на рисунку 4.10 унаочнюють математичні закономірності просторово-адаптивних багатоагентних систем. На початковому етапі модифікація MAS Weight-Map TV демонструє чітку амплітудну та структурну перевагу над альтернативами за критеріями PSNR та SSIM. Це підтверджує, що за помірного зашумлення попіксельне масштабування сил згладжування ефективно диференціює низькоамплітудні текстури та захищає контури від дифузійного розмиття.

Проте у міру зростання інтенсивності завад спостерігається інверсія графіків: криві багатоагентних конфігурацій деградують стрімкіше за монольну модель Plain TV. Такий характер поведінки зумовлений кількома взаємопов'язаними факторами. По-перше, за екстремального зашумлення оператор локальної дисперсії оцінює не геометрію підкладинного кадру, а просторову топологію щільного шумового поля, через що карта ваг набуває хаотичного характеру. По-друге, позначається обмеженість просторового контексту агентів: децентралізовані вузли в межах ізольованих патчів втрачають спроможність коректно ідентифікувати великомасштабні контури. По-третє, вагомим чинником є ефект специфічної алгоритмічної настройки: оптимізаційний пошук гіперпараметрів системи (parameter sweep) проводився фіксовано для рівня шуму $\sigma = 25$. Як наслідок, коефіцієнти внутрішніх ваг виявилися жорстко оптимізованими під цю точку, що знизило адаптивну гнучкість моделі при суттєвому відхиленні інтенсивності завад від базового значення.

Таким чином, експериментальні криві чітко окреслюють область інженерної ефективності розробленого методу. Модифікація MAS Weight-Mar TV є високооптимальною для сценаріїв реставрації із низьким та помірним рівнем зашумлення, де критично важливим є утримання дрібних мікроструктур. При переході в зону критичних завад раціональнішим є повернення до централізованих монолітних схем для уникнення локальних помилок вагового консенсусу.

ВИСНОВКИ

У кваліфікаційній роботі бакалавра проведено комплексне дослідження та розробку децентралізованої багатоагентної системи для очищення цифрових зображень від високочастотного гаусового шуму. Особливу увагу приділено поєднанню методів багатоагентного моделювання та розподіленої математичної оптимізації для створення адаптивних систем фільтрації, здатних ефективно паралелізувати обчислення та гнучко налаштовуватися на локальні геометричні особливості візуального контенту.

Під час виконання роботи було детально проаналізовано предметну область цифрової обробки сигналів, визначено ключові фактори деградації зображень та проведено порівняльний аналіз існуючих методів монолітної фільтрації. Дослідження виявило, що класичні алгоритми, включаючи базовий гомогенний денойзинг Total Variation, мають недостатню ефективність у збереженні чіквості контурів та дрібних текстур через використання статичних параметрів регуляризації без урахування локальних структур об'єктів, що й обґрунтувало необхідність розробки принципово нової архітектури.

Для вирішення цієї проблеми було спроектовано та формалізовано математичну модель розподіленого Consensus ADMM для багатоагентної системи. Зображення декомпоновано на систему просторових патчів, які перекриваються за регулярною сіткою, враховуючи діагональні зв'язки. Взаємодія між вузлами реалізує строгий механізм математичного консенсусу, де координатор виконує роль комунікаційної шини, а узгодження значень суміжних пікселів на ребрах графа автоматично усуває крайові та блочні артефакти. Перехід від Plain TV до розподіленого формату MAS Uniform TV забезпечив приріст PSNR з 28.08 dB до 28.24 dB та SSIM з 0.74 до 0.76 (при базових зашумлених PSNR = 20.43 dB, SSIM = 0.30), що

доводить спроможність локалізації обчислень точніше адаптувати фільтрацію.

Важливим кроком став захист внутрішньої логіки обчислювального вузла шляхом інтеграції просторово-адаптивної карти ваг регуляризації. Конфігурація MAS Weight-Map TV продемонструвала фінальний результат PSNR на рівні 28.60 dB, що на +0.36 dB вище за рівномірну агентну модель і на +0.52 dB перевищує класичний алгоритм Plain TV. Водночас індекс структурної подібності SSIM залишився на рівні 0.76, що є повністю еквівалентним показнику MAS Uniform TV і математично підтверджує, що жорсткі умови ребрового консенсусу однаково надійно утримують загальну топологію великих граней.

Практична цінність роботи підтверджується створенням програмного комплексу мовою Python. Експерименти повністю довели математичну стійкість архітектури: примальний залишок продемонстрував монотонне згасання, перетинаючи позначку 1.0 на 3-й ітерації. Дуальний залишок стрімко зменшувався з початкового рівня, плавно виходячи на асимптотичне плато в межах значень першого порядку на фінальних раундах, що вказує на швидку фіксацію просторово-адаптивних карт ваг.

Експериментально підтверджено існування інженерного компромісу розробленого методу. Ізольований тест із застосуванням збереженої бінарної маски імпульсного шуму виявив, що адаптивна модель ефективно утримує різкі контури «між шумом» без дифузійного розмиття, хоча платою за це є мікроструктурна неоднорідність фону. Крім того, виявлено ефект алгоритмічного перенавчання (overfitting): оскільки пошук гіперпараметрів здійснювався під рівень шуму $\sigma = 25$, при зростанні інтенсивності завад ефективність деградує стрімкіше за монолітну модель через хаотизацію карти ваг та обмеженість просторового контексту патчів. Отримані висновки окреслюють вектор для подальшого вдосконалення методів просторової стабілізації ваг у розподілених середовищах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Gonzalez R. C. Digital image processing. Reading, Mass : Addison-Wesley Pub. Co., Advanced Book Program, 1977.
2. Handbook of image and video processing / ed. by A. C. Bovik 1958. 2nd ed. Boston, MA : Academic Press, 2005. 1372 p.
3. Chan R. H., Chung-Wa, Nikolova M. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE transactions on image processing*. 2005. Vol. 14, no. 10. P. 1479–1485. URL: <https://doi.org/10.1109/tip.2005.852196> (date of access: 30.06.2026).
4. Russ J. C. Image processing handbook. Taylor & Francis Group, 2016. 885 p.
5. Szeliski R. Computer vision. Springer, 2010. 834 p.
6. Tukey J. W. Exploratory data analysis. Reading, Mass : Addison-Wesley Pub. Co., 1977. 688 p.
7. Buades A., Coll B., Morel J. M. A review of image denoising algorithms, with a new one. *Multiscale modeling & simulation*. 2005. Vol. 4, no. 2. P. 490–530. URL: <https://doi.org/10.1137/040616024> (date of access: 03.07.2026).
8. Rudin L. I., Osher S., Fatemi E. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*. 1992. Vol. 60, no. 1-4. P. 259–268. URL: [https://doi.org/10.1016/0167-2789\(92\)90242-f](https://doi.org/10.1016/0167-2789(92)90242-f) (date of access: 03.07.2026).
9. Perona P., Malik J. Scale-space and edge detection using anisotropic diffusion. *IEEE transactions on pattern analysis and machine intelligence*. 1990. Vol. 12, no. 7. P. 629–639. URL: <https://doi.org/10.1109/34.56205> (date of access: 03.07.2026).
10. Buades A., Coll B., Morel J. M. A non-local algorithm for image denoising. *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, San Diego, CA, USA. URL: <https://doi.org/10.1109/cvpr.2005.38> (date of access: 03.07.2026).

11. Image denoising by sparse 3-D transform-domain collaborative filtering / K. Dabov et al. *IEEE transactions on image processing*. 2007. Vol. 16, no. 8. P. 2080–2095. URL: <https://doi.org/10.1109/tip.2007.901238> (date of access: 03.07.2026).
12. Wooldridge M. Introduction to multiagent systems. Wiley & Sons, Incorporated, John, 2013. 484 p.
13. Olfati-Saber R., Fax J. A., Murray R. M. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*. 2007. Vol. 95, no. 1. P. 215–233. URL: <https://doi.org/10.1109/jproc.2006.887293> (date of access: 03.07.2026).
14. Wei Ren, Beard R. W., Atkins E. M. A survey of consensus problems in multi-agent coordination. *Proceedings of the 2005, american control conference*, 2005., Portland, OR, USA. URL: <https://doi.org/10.1109/acc.2005.1470239> (date of access: 03.07.2026).
15. Egerstedt M., Mesbahi M. Graph theoretic methods in multiagent networks. Princeton University Press, 2010. 424 p.
16. Tomasi C., Manduchi R. Bilateral filtering for gray and color images. *IEEE 6th international conference on computer vision*, Bombay, India. URL: <https://doi.org/10.1109/iccv.1998.710815> (date of access: 03.07.2026).
17. Milanfar P. A tour of modern image filtering: new insights and methods, both practical and theoretical. *IEEE signal processing magazine*. 2013. Vol. 30, no. 1. P. 106–128. URL: <https://doi.org/10.1109/msp.2011.2179329> (date of access: 03.07.2026).
18. Geman S., Geman D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE transactions on pattern analysis and machine intelligence*. 1984. PAMI-6, no. 6. P. 721–741. URL: <https://doi.org/10.1109/tpami.1984.4767596> (date of access: 03.07.2026).
19. Huber P. J. Robust statistics. Hoboken, NJ, USA : John Wiley & Sons, Inc., 1981. URL: <https://doi.org/10.1002/0471725250> (date of access: 03.07.2026).

20. Besag J. On the statistical analysis of dirty pictures. *Journal of the royal statistical society: series B (methodological)*. 1986. Vol. 48, no. 3. P. 259–279. URL: <https://doi.org/10.1111/j.2517-6161.1986.tb01412.x> (date of access: 03.07.2026).

21. Boyd S. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and trends® in machine learning*. 2010. Vol. 3, no. 1. P. 1–122. URL: <https://doi.org/10.1561/22000000016> (date of access: 03.07.2026).

22. Oden J. T., Glowinski R., Tallec P. L. Augmented lagrangian and operator-splitting methods in nonlinear mechanics. *Mathematics of computation*. 1992. Vol. 58, no. 197. P. 451. URL: <https://doi.org/10.2307/2153048> (date of access: 03.07.2026).

23. Boyd S., Parikh N. Proximal algorithms. Now Publishers, 2013. 130 p.

24. Elad M., Aharon M. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE transactions on image processing*. 2006. Vol. 15, no. 12. P. 3736–3745. URL: <https://doi.org/10.1109/tip.2006.881969> (date of access: 03.07.2026).

25. D-ADMM: a communication-efficient distributed algorithm for separable optimization / J. F. C. Mota et al. *IEEE transactions on signal processing*. 2013. Vol. 61, no. 10. P. 2718–2723. URL: <https://doi.org/10.1109/tsp.2013.2254478> (date of access: 03.07.2026).

26. Chambolle A., Pock T. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*. 2010. Vol. 40, no. 1. P. 120–145. URL: <https://doi.org/10.1007/s10851-010-0251-1> (date of access: 03.07.2026).

27. Fast alternating direction optimization methods / T. Goldstein et al. *SIAM journal on imaging sciences*. 2014. Vol. 7, no. 3. P. 1588–1623. URL: <https://doi.org/10.1137/120896219> (date of access: 03.07.2026).

28. Li M., Li B. A novel weighted total variation model for image denoising. *IET image processing*. 2021. Vol. 15, no. 12. P. 2749–2760. URL: <https://doi.org/10.1049/ipr2.12259> (date of access: 03.07.2026).
29. Hong M., Luo Z.-Q. On the linear convergence of the alternating direction method of multipliers. *Mathematical programming*. 2016. Vol. 162, no. 1-2. P. 165–199. URL: <https://doi.org/10.1007/s10107-016-1034-2> (date of access: 03.07.2026).
30. NumPy Documentation – numpy v2.5 manual. *NumPy*. URL: <https://numpy.org/doc/stable/> (date of access: 03.07.2026).
31. SciPy Documentation – scipy v1.18.0 manual. *Numpy and SciPy Documentation*. URL: <https://docs.scipy.org/doc/scipy/> (date of access: 03.07.2026).
32. API Reference Scikit-image's documentation – skimage 0.26.0 documentation. *scikit-image: Image processing in Python – scikit-image*. URL: <https://scikit-image.org/docs/stable/> (date of access: 03.07.2026).
33. Hore A., Ziou D. Image quality metrics: PSNR vs. SSIM. *2010 20th international conference on pattern recognition (ICPR)*, Istanbul, Turkey, 23–26 August 2010. 2010. URL: <https://doi.org/10.1109/icpr.2010.579> (date of access: 03.07.2026).
34. Image quality assessment: from error visibility to structural similarity / Z. Wang et al. *IEEE transactions on image processing*. 2004. Vol. 13, no. 4. P. 600–612. URL: <https://doi.org/10.1109/tip.2003.819861> (date of access: 03.07.2026).
35. Nikolova M. Analysis of the recovery of edges in images and signals by minimizing nonconvex regularized least-squares. *Multiscale modeling & simulation*. 2005. Vol. 4, no. 3. P. 960–991. URL: <https://doi.org/10.1137/040619582> (date of access: 03.07.2026).