

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ РАДІОФІЗИКИ, ЕЛЕКТРОНІКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ

Кафедра радіотехніки та радіоелектронних систем

До захисту допущено:

«На правах рукопису»

Завідувач кафедри _____ Ігор АНІСІМОВ

« __ » червня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему:

**«МЕТОД ВИЗНАЧЕННЯ ПРОСТОРОВИХ РОЗПОДІЛІВ КОМПОНЕНТНОГО
СКЛАДУ ОБ'ЄКТІВ ЗА КОЛЬОРОВИМ ЗОБРАЖЕННЯМ»**

Виконав:

студентка 4-го курсу

денної форми навчання

спеціальності 172 - Телекомунікації та радіотехніка

ОП «Інформаційна безпека телекомунікаційних систем і мереж»

Микитенко Ольга Сергіївна _____

Науковий керівник:

д.т.н., доц. Ольшевський Сергій Валентинович _____

Рецензент:

к.ф.-м.н., ас. Недибалюк Олег Анатолійович _____

Засвідчую, що у цій бакалаврській роботі

немає запозичень з праць інших авторів без

відповідних посилань

Студент _____

Робота допущена до захисту в ЕК рішенням кафедри радіотехніки та радіоелектронних систем від «23» червня 2023 р., протокол № 22.

Завідувач кафедри радіотехніки та радіоелектронних систем,

доктор фіз.-мат. наук, професор

Анісімов Ігор Олексійович _____

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ.....	5
1.1. Загальний опис.....	5
1.2. Змістовна постановка задачі.....	6
РОЗДІЛ 2. АНАЛІЗ СТАНУ ПРОБЛЕМУ.....	8
2.1. Загальні відомості.	8
2.1.1. Кластерний аналіз.....	9
2.1.2. Аналіз головних компонентів.....	9
2.1.3. Вейвлет-аналіз.....	11
2.1.4. Сегментація зображення.....	11
2.2. Висновки до розділу.....	12
РОЗДІЛ 3. ЗАПРОПОНОВАНИЙ ПІДХІД.....	15
3.1. Використання емісійного спектру.....	15
3.2. Розрахунок координат кольору (x, y, z).....	16
3.3. Перерахунок в RGB-координати.....	18
3.4. Застосування OLAM методу декомпозиції.....	19
РОЗДІЛ 4. РЕЗУЛЬТАТИ ЕСПЕРИМЕНТУ ТА ОБГОВОРЕННЯ.....	22
4.1. Достовірність розрахунків.....	22
4.2. Вплив рівня шуму на достовірність алгоритмів.....	25
4.3. Шум квантування для RGB форматів.....	26
ВИСНОВКИ.....	29
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	30
ДОДАТОК А. КОД ОПИСАНИЙ У ПРОГРАМНОМУ ЗАБЕЗПЕЧЕННІ R ДЛЯ ПРОМІЖНИХ ТА ОСНОВНИХ ОБЧИСЛЕНЬ.....	31
ДОДАТОК Б. ЗОБРАЖЕННЯ РОЗПОДІЛУ ЕЛЕМЕНТІВ.....	41

ВСТУП

У сучасному світі методи аналізу та визначення характеристик об'єктів займають важливе місце в багатьох наукових та технологічних галузях. Особливо це стосується області де вивчення просторових розподілів компонентного складу об'єктів є ключовим для розуміння та вдосконалення певних технологічних процесів. Прикладом такого об'єкту може бути плазмовий факел.

Методи аналізу кольорових зображень факела прямогочного плазмотрона вимагають врахування неоднорідностей, шумів та інших небажаних факторів, які можуть вплинути на точність результатів. Проте, розробка ефективних методів визначення просторових розподілів компонентного складу має великий потенціал для вдосконалення процесів, пов'язаних з плазмовими технологіями. Один з поширених методів аналізу об'єктів полягає у використанні кольорових зображень, які можуть надати цінну інформацію про компонентний склад та структуру об'єктів.

Метою даної дипломної роботи є розробка нового методу визначення просторових розподілів компонентного складу об'єктів за кольоровим зображенням факела прямогочного плазмотрона. Цей метод буде базуватися на аналізі колірної інформації, що міститься у зображеннях, та використанні OLAM – методу декомпозиції.

Дослідження в області визначення просторових розподілів компонентного складу об'єктів за кольоровим зображенням факела прямогочного плазмотрона має великий практичний потенціал. Отже, дана дипломна робота є актуальним кроком у напрямку розробки нових методів аналізу об'ємних світних об'єктів та визначення просторових розподілів їх компонентного складу. Її результати можуть мати велике значення для наукових досліджень та промислових застосувань, сприяючи подальшому

розвитку технологій плазмової обробки і забезпечуючи покращення якості та ефективності процесів, що ґрунтуються на використанні плазмотронів.

РОЗДІЛ 1

ПОСТАНОВКА ЗАДАЧІ

1.1. Загальний опис.

При дослідженні протяжних джерел світіння часто важливо отримати просторовий розподіл центрів випромінювання. Особливо важливо отримати такий розподіл для нестационарних об'єктів, що швидко змінюються, з об'ємними неоднорідностями. Такими об'єктами можуть бути полум'я пальників або факели реактивних двигунів. Отримання таких розподілів з необхідною точністю апаратними методами потребує досить дорогого обладнання. Крім того, у випадку високошвидкісних процесів, коли час зміни просторових характеристик властивостей співмірний з часом накопичення сигналу спектрометра, це, як правило, неможливо. Однак отримання усереднених за часом і площею спектрів випромінювання об'єкта сучасними апаратними засобами є досить простим і не потребує дорогого обладнання.

Таким чином, розробка цифрових методів обробки зображень протяжних нестационарних об'єктів з використанням їх усереднених спектрів випромінювання є актуальною, особливо такі методи можуть бути перспективними при дослідженні різних режимів горіння котлів або вихлопів реактивних двигунів для контролю хімічного складу палива.

В 1970-х роках дослідники розпочали вивчення аналізу кольорових зображень. Вони почали експериментувати зі способами вимірювання, витягування та моделювання кольорових даних, щоб отримати інформацію про розподіл компонентів об'єктів. У 1980-1990-х роках були розроблені традиційні методи обробки зображень для визначення кольорових характеристик об'єктів. Ці методи включали фільтрацію, сегментацію, аналіз градієнтів та статистичні методи.

У 2000-х роках з'явилися нові методи, що базуються на використанні машинного та глибинного навчання. Ці підходи дозволили автоматично витягувати ознаки з кольорових зображень та аналізувати просторові розподіли компонентів об'єктів з високою точністю.

Дані методи знайшли широке застосування в багатьох галузях. Наприклад, вони використовуються в медичній діагностиці для аналізу зображень, в комп'ютерній графіці для створення реалістичних образів, в робототехніці для розпізнавання об'єктів та в багатьох інших сферах.

Завдяки прогресу у обчислювальній потужності та алгоритмічних підходах, методи визначення просторових розподілів компонентного складу об'єктів можуть бути застосовані в реальному часі, що робить їх корисними для широкого спектру додатків.

Сучасні методи включають у себе розширені можливості обробки зображень, такі як машинне навчання, глибинне навчання та штучний інтелект. Це дозволяє отримати більш точні результати та автоматизувати процес аналізу кольорових зображень.

1.2. Змістовна постановка задачі.

У цій дипломній роботі розроблений метод для визначення просторових розподілів компонентного складу об'єктів зображення факела прямогочного плазмотрона. Робота включає в себе розробку алгоритмів, експериментальні дослідження та аналіз різних методів для досягнення мети.

Предметом дослідження є кольорове зображення, отримане з цифрової фотографії. Кольорові зображення складаються з пікселів, які мають тривимірні кольорові значення (часто відображені у форматі RGB — червоний, зелений, синій). Кожен піксель представляє окремий об'єкт або його частину на зображенні.

Об'єктом дослідження є просторовий розподіл компонентного складу об'єктів на кольоровому зображенні. Компонентний склад об'єктів включає різні параметри, такі як колір, текстуру, яскравість тощо.

Метод, що розглядається, використовується для отримання кількісних даних про просторовий розподіл компонентного складу об'єктів на кольоровому зображенні.

Таким чином, можна сформулювати список задач, які треба вирішити у процесі розробки методу:

1. Розв'язати задачу по декомпозиції зображення на окремі компоненти, які формуються окремими центрами висвічування по всьому полю зору.
2. Застосувати метод OLAM-декомпозиції для відшукування вагових коефіцієнтів.
3. Оцінити вплив шуму квантування для стандартних RGB форматів на достовірність OLAM-декомпозиції.

Результати дослідження можуть бути корисними у таких галузях, як обробка зображень, медична діагностика, розпізнавання образів та інші.

РОЗДІЛ 2

АНАЛІЗ СТАНУ ПРОБЛЕМИ

2.1. Основні методи визначення просторових розподілів.

Існує кілька методів для визначення просторових розподілів компонентного складу об'єктів за кольоровим зображенням. Основні методи:

Кластерний аналіз: метод використовується для групування пікселів на зображенні в окремі кластери на підставі їх кольору. Кластерний аналіз дозволяє виявити різні області зображення, що мають схожі кольорові характеристики.

Аналіз головних компонентів (PCA): метод здійснює лінійну проекцію вхідних даних у новий простір зменшеної розмірності. Він може бути застосований до кольорового зображення для виділення основних характеристик розподілу кольорів в зображенні.

Вейвлет-аналіз: застосовується для розкладання зображення на різні масштаби та орієнтації. Вейвлет-аналіз дозволяє виявити текстурні особливості та розподіл кольорів в зображенні на різних масштабах.

Сегментація зображення шляхом злиття гістограм на основі K-середніх кластерів: ідея полягає в тому, щоб спочатку виконати сегментацію зображення за допомогою алгоритму кластеризації K-середніх. Після виконання кластеризації K-середніх для отримання початкової сегментації зображення, виконується вибіркоче злиття гістограм.

Ці методи можуть застосовуватись окремо або в поєднанні один з одним для аналізу просторових розподілів компонентного складу об'єктів за кольоровим зображенням.

2.1.1. Кластерний аналіз.

Кластерний аналіз - це метод машинного навчання та аналізу даних, який використовується для групування об'єктів на основі схожості їх характеристик. У контексті визначення просторових розподілів компонентного складу об'єктів за кольоровим зображенням, кластерний аналіз допомагає виявити різні області зображення зі схожими кольоровими характеристиками.

Спочатку вибирається набір даних, які представляють кольорове зображення. Кожен піксель на зображенні може бути розглянутий як окремий об'єкт з колірними характеристиками.

Слід визначити міру схожості, яка визначає, наскільки близькі один до одного два об'єкти. Наступним кроком є вибір алгоритму кластеризації. Існує багато алгоритмів кластеризації, таких як k-середніх (k-means), ієрархічна кластеризація, спектральна кластеризація тощо.

Об'єкти (пікселі) групуються в кластери залежно від їх схожості за обраною мірою.

Після завершення кластеризації необхідно оцінити і проаналізувати отримані результати. Це може включати візуалізацію кластерів на зображенні або обчислення статистичних показників, таких як середні значення кольорів у кожному кластері. Кластерний аналіз може допомогти виділити різні області зображення зі схожими кольоровими характеристиками, що визначають просторові розподіли компонентного складу об'єктів. Він є потужним інструментом для аналізу кольорових зображень та виділення різних областей чи об'єктів .

2.1.2. Аналіз головних компонентів

Аналіз головних компонентів (Principal Component Analysis, PCA) є

статистичним методом, який використовується для зменшення розмірності даних та виявлення основних шаблонів або головних компонентів у наборі даних. У контексті визначення просторових розподілів компонентного складу об'єктів за кольоровим зображенням, PCA може застосовуватися для виділення основних характеристик розподілу кольорів у зображенні.

Кольорові значення пікселів на зображенні конвертуються у вектори даних. Кожен піксель розглядається як окремий об'єкт з характеристиками, які складаються з кольорових компонентів.

Вектори даних стандартизуються шляхом віднімання середнього значення кольору та поділу на стандартне відхилення. Це забезпечує одиничну дисперсію для кожної компоненти та уникнення впливу на розподіл через різний діапазон значень кольорів. За допомогою стандартизованих даних обчислюється матриця коваріації. Ця матриця відображає ступінь взаємозв'язку між різними компонентами даних.

Далі застосовується метод власних значень або сингулярного розкладу для отримання головних компонентів. Головні компоненти є лінійними комбінаціями початкових компонентів, які представляють основні шаблони або напрямки варіації в даних. Вибирається підмножина головних компонентів з найбільшою варіацією. Зазвичай, відбираються ті компоненти, які пояснюють більшу частину загальної дисперсії даних.

Головні компоненти використовуються для відновлення даних. Це дозволяє зменшити розмірність даних та зберегти основні характеристики розподілу кольорів у зображенні.

Аналіз головних компонентів дозволяє знайти основні шаблони або напрямки варіації в кольорових даних зображення. Це може бути корисним для розпізнавання об'єктів, виділення текстур або зменшення розмірності даних для подальшого аналізу.

2.1.3. Вейвлет-аналіз.

Вейвлет-аналіз є методом аналізу сигналів, що дозволяє розкласти сигнал або зображення на різні масштаби та орієнтації. У контексті визначення просторових розподілів компонентного складу об'єктів за кольоровим зображенням, вейвлет-аналіз може застосовуватися для виявлення текстурних особливостей та розподілу кольорів у зображенні на різних масштабах.

Перши кроком є вибір вейвлет-функції, яка буде використовуватися для розкладання зображення. Вейвлет-функції мають різні форми та властивості, що впливають на результати аналізу. Зображення розкладається на різні масштаби та орієнтації за допомогою вейвлет-функцій. Результатом розкладання є ряд підсигналів, які представляють різні деталі та апроксимації зображення на різних рівнях деталізації.

Отримані коефіцієнти розкладання аналізуються для визначення текстурних особливостей та розподілу кольорів. За необхідності, після аналізу коефіцієнтів може проводитися синтез сигналу. Це зворотне перетворення вейвлет-функцій, що дозволяє відновити оригінальне зображення з розкладених компонент.

Використання вейвлет-аналізу у визначенні просторових розподілів компонентного складу об'єктів за кольоровим зображенням допомагає виділити текстурні особливості, виявити зміни в колірних характеристиках та аналізувати їх розподіл у просторі.

2.1.4. Сегментація зображення

Процес сегментації зображень полягає в розкладанні зображення на компоненти на основі схожості атрибутів, таких як колір, текстура тощо.

Серед різних методів сегментації, кластерні підходи, такі як кластеризація K_{means} , є найпоширенішими через їхню ефективність та простоту. Кластеризація K-середніх - це ітераційний алгоритм, який розділяє n спостережень на k кластерів таким чином, що кожне спостереження відноситься до того кластеру, центроїд якого має мінімальну відстань до цього спостереження, ніж будь-який інший центроїд. Ідея сегментації кольорових зображень шляхом злиття гістограм на основі K-середніх кластерів була вперше запропонована в ефективний і простий у реалізації підхід до сегментації, який дозволяє отримати більш якісне сегментоване зображення одночасно вирішуючи проблеми, пов'язані з часом обробки та апаратних ресурсів, необхідних для реалізації в реальному часі.

Вхідне кольорове зображення перетворюється у шість різних кольірних просторів, щоб скористатися властивостями, пов'язаними з кожним з них. Початкові карти сегментації створюються шляхом взяття локальної колірної гістограми кожного пікселя, заснованої на колірному вмісту пікселя та його найближчих сусідніх пікселів, а потім віднесення кожного пікселя до біну в переквантованому колірному просторі. Переквантована кольорова гістограма зображення, кластеризується за допомогою K-середніх у кожному з кольірних просторів. Отримані таким чином початкові карти сегментації об'єднуються разом, щоб отримати остаточне сегментоване зображення.

Запропонований підхід, полягає в об'єднанні карт сегментації, відібраних за найвищими значеннями ентропії. Сегментація зображення дозволяє розпізнавати об'єкти, визначати їх межі, виконувати вимірювання та аналізувати характеристики зображення для подальшого розуміння та обробки даних.

2.2. Висновок до розділу.

Усі перераховані методи активно застосовуються для обробки зображень з ціллю вирішення конкретних задач, але кожен з них має чимало недоліків, які є завадою для досягнення встановленої нами мети.

Так кластерний аналіз має проблему з встановленням кількості кластерів неправильний вибір яких може призвести до некоректних або неінтерпретованих результатів.

Аналіз головних компонентів базується на припущенні про лінійну залежність між змінними. Він може бути неефективним, якщо вихідні дані мають складну нелінійну структуру. Також вибір кількості головних компонентів для збереження є суб'єктивним процесом і може призвести до втрати інформації або надмірного шуму у зменшених даних. Після застосування PCA важко інтерпретувати головні компоненти, оскільки вони є лінійними комбінаціями вихідних змінних. Головні компоненти не завжди мають прямий зв'язок з конкретними змінними, що ускладнює їх фізичне або семантичне пояснення.

Вейвлет-аналіз вимагає вибору вейвлет-функції, яка визначає форму та властивості вейвлету. Немає універсального вейвлету, який підходить для всіх типів сигналів або зображень.

Також вейвлет-аналіз є залежним від розмірності та шкали сигналу або зображення. Вибір оптимальної шкали для аналізу може бути складним завданням. Велика шкала може приводити до втрати дрібних деталей, тоді як мала шкала може приводити до втрати загальної структури. У даному методі також відімчається здатність до локального аналізу зображення в часовій та частотній областях, але він не зберігає повну локалізацію в часі. Це означає, що точна інформація про те, коли виникають певні події в сигналі, може бути втрачена.

Результати методу сегментації зображення шляхом злиття гістограм на

основі K-середніх кластерів можуть залежати від випадкового вибору початкових центроїдів у кластеризації K-середніх. Якщо початкові центроїди обрані неоптимально, це може призвести до поганої сегментації зображення.

Наявність шумів в зображенні може вплинути на гістограму кластерів і, відповідно, на процес злиття. Це може привести до неправильного злиття регіонів або до зміни меж сегментів.

Відсутність універсального алгоритму, який підходить для всіх типів зображень, може створювати виклики при виборі оптимального методу сегментації.

Також серйозними недоліками, які притаманні кожному з вище перерахованих методів, є велика обчислювальна складність, висока чутливість до освітлення, контрасту, шуму та аномалій, складність в обробці великого розміру даних та інтерпретації результатів. Дані фактори стануть серйозними перешкодами для досягнення поставленої нами мети.

РОЗДІЛ 3

ЗАПРОПОНОВАНИЙ ПІДХІД

3.1. Використання інтегрального емісійного спектра.

Кольори, живого об'єкта світіння та його зображенням мають різну історію формування, не зважаючи на те, що сприймаємо ми їх однаково. У зображенні інформацію про те у який спосіб воно було сформоване втрачена, оскільки колір пальника плазмотрона формується суцільним спектром довжин хвиль, який є неперервним.

Сумарна дія всіх довжин хвиль спричиняє той факт, що ми оком бачимо той чи інший відтінок. Довжини хвилі які випромінює плазмовий факел обумовлені тим з яких спектральних характеристик центрів висвічування він складається. До складу будь-якого полум'я входять атоми і молекули, при високій температурі вони починають світитись. Нам відомий інтегральний спектр полум'я, знятий спеціальним приладом.

Емісійні спектри є унікальними для кожного хімічного елементу. Такі спектральні характеристики дозволяють ідентифікувати наявність конкретних елементів або сполук у випромінюючому джерелі.

Полум'я розподілене по простору, різні точки мають різні яскравість та відтінки, тобто у нас є розподіл концентрації центрів висвічування по полю зору зображення полум'я.

Розглядаючи піксель у його виключно один колір, та володіючи спектральними характеристиками кожного з елементів, ми групуємо спектри між собою з різними ваговими коефіцієнтами, в результаті їх сумарної дії в певному місці утворюється колір, який ми спостерігаємо.

Нами було синтезоване зображення з наперед відомими ваговими коефіцієнтами. У форматі RGB, воно було оброблене за допомогою методу

OLAM, з метою розрахунку коефіцієнтів та порівняння їх із заданими на початку. Тож, вагові коефіцієнти підбираємо таким чином щоб в конкретній точці отримати колір, ідентичний тому, що зображений на фото. Якщо підібраний за рахунок вагових коефіцієнтів колір буде збігатись з тим, що на зображенні то можна зробити висновок про концентрацію елементів у даній області. На Рис. 3.1.1 зображена блок-схема застосованого алгоритму.



Рис. 3.1.1 – Блок-схема алгоритму.

3.2. Розрахунок координат кольору (x, y, z).

Як протяжне джерело неоднорідного випромінювання використовували плазмовий пальник прямоочного плазмотрона. Довжини хвиль, які він випромінює, залежать від атомів, з яких складається полум'я, а саме від спектральних характеристик центрів висвічування, яких у даному випадку чимало.

Усереднений спектр випромінювання $S(\lambda)$ розширеного джерела в діапазоні довжин хвиль від 360 до 830 нм було використано як джерело первинних даних для побудови бази спектрів центрів випромінювання. Для цього були розшифровані окремі фрагменти спектрів з ідентифікацією центрів випромінювання.

Повні спектри центрів випромінювання для побудови базису були відновлені за допомогою відповідних баз даних. Це NIST - для молекулярних спектрів або таблиці Зайделя для атомних спектрів. Також використовувався симулятор молекулярних спектрів повітряного компонента Specair 2.2. Група неідентифікованих центрів випромінювання протяжного джерела розглядалася як окремий елемент базису центрів випромінювання. Вона отримала назву "залишок". Для цього з усередненого спектра виділялися спектри встановлених центрів випромінювання для ідентифікованих фрагментів.

Полум'я розподілене по простору, різні точки мають різні яскравість та відтінки, тобто в нас є розподіл концентрації цих центрів висвічування по полю зору зображення полум'я.

В оцифрованому вигляді середній спектр був одновимірним масивом даних $S[n]$ за розмірністю N . Основу центрів випромінювання було сформовано у вигляді двовимірного масиву $F[i, n]$, розміром $M \times N$. Розмір кроку довжини хвилі $\Delta\lambda$ дискретних спектрів був обраний рівним 1 нм.

На основі отриманих спектрів були розраховані координати кольору (x , y , z) в колірному просторі CIE 1931 для кожного компонента основи:

$$x[i] = k_c \sum_{n=1}^N F[i, n] \bar{x}[n] \Delta\lambda;$$

$$y[i] = k_c \sum_{n=1}^N F[i, n] \bar{y}[n] \Delta\lambda;$$

$$z[i] = k_c \sum_{n=1}^N F[i, n] \bar{z}[n] \Delta\lambda$$

Тут k_c - константа, яку можна взяти за 1 для розрахунку просторового розподілу центрів випромінювання. Масиви $\bar{x}[n]$, $\bar{y}[n]$, $\bar{z}[n]$ є табличними кривими додавання довжин хвиль $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, $\bar{z}(\lambda)$ у просторі CIE 1931 року.

Тривимірність кольору дає підставу виразити його у вигляді вектора в

просторі. Використовуючи координатні осі, можна знайти вектори будь-якого кольору, одержуваного за допомогою обраних основних. Сукупність кольорів, виражена в даній системі основних, називається колірним простором системи.

3.3. Перерахунок в RGB-координати.

Для всіх компонентів базису координати були перераховані в RGB-координати.

Координати кольору в лінійному просторі RGB $(\bar{r}, \bar{g}, \bar{b})$ можна розрахувати за таким співвідношенням:

$$\begin{bmatrix} \bar{r} \\ \bar{g} \\ \bar{b} \end{bmatrix} = [M]^{-1} \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix},$$

де $\bar{x}, \bar{y}, \bar{z}$ - координати кольору в колорному просторі МКО 1931 р. СІЕ XYZ, а $[M]^{-1}$ – обернена матриця перерахунку М.

Пряма матриця перерахунку визначається за формулою:

$$[M] = \begin{bmatrix} S_r \bar{x}_r & S_g \bar{x}_g & S_b \bar{x}_b \\ S_r \bar{y}_r & S_g \bar{y}_g & S_b \bar{y}_b \\ S_r \bar{z}_r & S_g \bar{z}_g & S_b \bar{z}_b \end{bmatrix}$$

де $(\bar{x}_r, \bar{y}_r, \bar{z}_r)$, $(\bar{x}_g, \bar{y}_g, \bar{z}_g)$ та $(\bar{x}_b, \bar{y}_b, \bar{z}_b)$ – координати кольору (в колорному просторі МКО 1931 р. СІЕ XYZ) основних кольорів у використовуваному типі колорного простору RGB, які можуть бути розраховані по відомим координатам кольоровостей (x_r, x_r) , (x_g, x_g) , (x_b, x_b) ; S_r, S_g, S_b – коефіцієнти перерахунку для червоного, зеленого та синього каналів.

Значення координат кольору основних кольорів розраховуються наступним чином:

$$\bar{x}_r = \frac{x_r}{y_r} \quad \bar{x}_g = \frac{x_g}{y_g} \quad \bar{x}_b = \frac{x_b}{y_b}$$

$$\bar{y}_r = 1 \quad \bar{y}_g = 1 \quad \bar{y}_b = 1$$

$$\bar{z}_r = \frac{(1-x_r-y_r)}{y_r} \quad \bar{z}_g = \frac{(1-x_g-y_g)}{y_g} \quad \bar{z}_b = \frac{(1-x_b-y_b)}{y_b}$$

Значення коефіцієнтів перерахунку для червоного, зеленого та синього каналів можна визначити за наступним виразом:

$$\begin{bmatrix} S_r \\ S_g \\ S_b \end{bmatrix} = \begin{bmatrix} \bar{x}_r & \bar{x}_g & \bar{x}_b \\ \bar{y}_r & \bar{y}_g & \bar{y}_b \\ \bar{z}_r & \bar{z}_g & \bar{z}_b \end{bmatrix}^{-1} \begin{bmatrix} \bar{x}_w \\ \bar{y}_w \\ \bar{z}_w \end{bmatrix}$$

де $(\bar{x}_r, \bar{y}_r, \bar{z}_r)$, $(\bar{x}_g, \bar{y}_g, \bar{z}_g)$, $(\bar{x}_b, \bar{y}_b, \bar{z}_b)$ - розраховані раніше за формулами координати кольору основних кольорів у використуваному типі колірному простору, а $(\bar{x}_w, \bar{y}_w, \bar{z}_w)$ - координати кольору точки білого у використуваному типі колірному простору.

В результаті отримуємо F_{rgb} - матрицю кольоровості центрів випромінювання, яка при множенні на вектор вагових коефіцієнтів \vec{w} пропорційних кількості центрів випромінювання в розширеному елементі джерела, визначає колір пікселя його зображення. Таким чином, колір кожного пікселя визначається системою лінійних рівнянь, матрична форма яких має вигляд:

$$F_{rgb} \times \vec{w} = \vec{P}_{rgb}$$

Тут F_{rgb} матриця кольорів центрів випромінювання, \vec{w} вектор шуканих вагових коефіцієнтів, \vec{P}_{rgb} колір пікселя у вигляді RGB-вектора.

3.4. Застосування методу OLAM-декомпозиції.

RGB-компонент $F[n]$ є дискретними груповими спектрами, для яких проводилась подальша декомпозиція.

Оскільки спектри різних молекул є незалежними, можна стверджувати, що спектр суміші молекул є суперпозицією спектрів окремих компонентів. Таким чином, є справедливим ствердження, що інтегральний спектр суміші молекул також є суперпозицією групових спектрів компонентів:

$$F[n] = \sum_{i=0}^M a_i f_i[n]$$

де a_i – невідомий ваговий коефіцієнт кожної складової суміші, який в подальшому будемо називати ваговим коефіцієнтом декомпозиції.

Як правило, для інтегральних спектрів загальна кількість відліків сигналу N значно перевищує кількість компонентів M . Оскільки невідомі коефіцієнти a_i , які є предметом розв'язку задачі декомпозиції, однакові для $\forall n$, має місце кінцева множина M систем неоднорідних лінійних алгебраїчних рівнянь з M невідомими, яка пов'язує між собою числові значення відліків спектрів $f_i[n]$, спектра суміші $F[n]$ і вагові коефіцієнти декомпозиції a_i . Матрична форма таких систем рівнянь має вигляд:

$$S \times \vec{a} = \vec{F}$$

Тут $S = \begin{pmatrix} f_1[n+1] & \dots & f_M[n+1] \\ \dots & \dots & \dots \\ f_1[n+M] & \dots & f_M[n+M] \end{pmatrix}$ – матриця відліків спектрів розмірністю

$M \times M$, $\vec{a} = (a_1, a_2, \dots, a_i, \dots, a_M)^T$ – вектор вагових коефіцієнтів декомпозиції, $\vec{F} = (F[n+1], F[n+2], \dots, F[n+M])^T$ – вектор відліків змішаного сигналу

Для кожної з цих систем рівнянь вектор розв'язків \vec{a} є однаковим. Назвемо цей вектор точним розв'язком задачі декомпозиції. Таким чином, загальна задача відшукування вагових коефіцієнтів суміші речовин або вагових коефіцієнтів декомпозиції, виходячи з відомих спектрів кожної окремої компоненти та експериментального спектра суміші, має точний розв'язок.

Процес побудови множини коректних систем неоднорідних лінійних рівнянь, що дають ансамбль реалізацій точного розв'язку задачі декомпозиції, вимагає попередньої ідентифікації якісного елементного складу досліджуваної суміші. Коли кількість компонент базису > 3 , задача декомпозиції стає некоректною. У цьому випадку ми використовували декілька зображень протяжного об'єкта, отриманих з різними фільтрами

одночасно.

Для практичного використання більш доцільно застосовувати метод OLAM-декомпозиції відшукування вагових коефіцієнтів. Він оснований на матричній формі принципу суперпозиції.

Точний розв'язок задачі декомпозиції в матричній формі:

$$\vec{a} = W \times \vec{F}$$

$$\text{Де } W = S^{-1}.$$

Виходячи з того, що один і той же точний розв'язок може бути отримано для всіх фрагментів довжиною M групових спектрів, матриця S не обов'язково повинна бути квадратною. Тобто кількість відліків реального і синтезованого спектрів може перевищувати кількість компонент суміші, для якої отримано сумарний спектр. Важливо, щоб кількість стовпців матриці S відповідала кількості відліків фрагмента експериментального групового спектра \vec{F} . Метод OLAM-декомпозиції полягає в тому, що дає точну формулу для відшукування вагової матриці W :

$$W = I \times (S^T \times S)^{-1} \times S^T$$

тут I – одинична матриця розмірністю M .

РОЗДІЛ 4

РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТУ ТА ОБГОВОРЕННЯ

4.1. Достовірність розрахунків.

Інтегральні емісійні спектри різних ділянок прямогочного плазмотрона, зображені на Рис. 4.1.1

Виходячи з цих спектрів сформовано перелік ідентифікованих центрів

висвічування випромінюючого джерела.

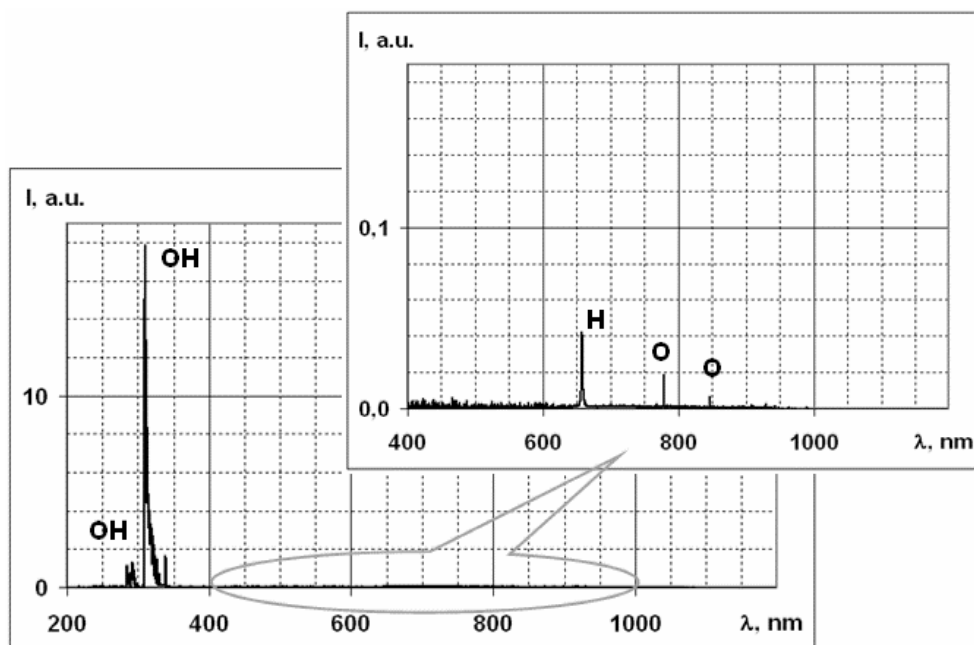


Рис. 4.1.1 – Інтегральний емісійний спектр на периферії факелу.

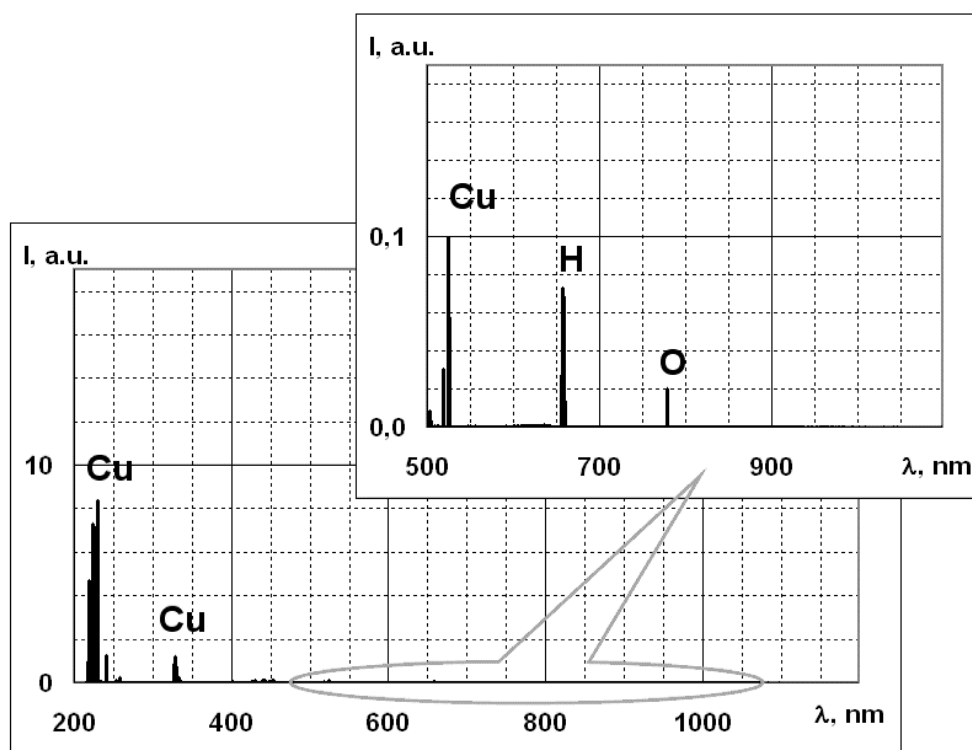


Рис. 4.1.2 – Інтегральний емісійний спектр з приелектродної ділянки факелу.

На Рис. 4.1.3 – 4.1.5 зображені унікальні емісійні спектри елементів Al, CO та CU, які, в свою чергу, входять до складу полум'я, взяті з відповідних баз даних (NIST, Atom Trace, Kurucz).

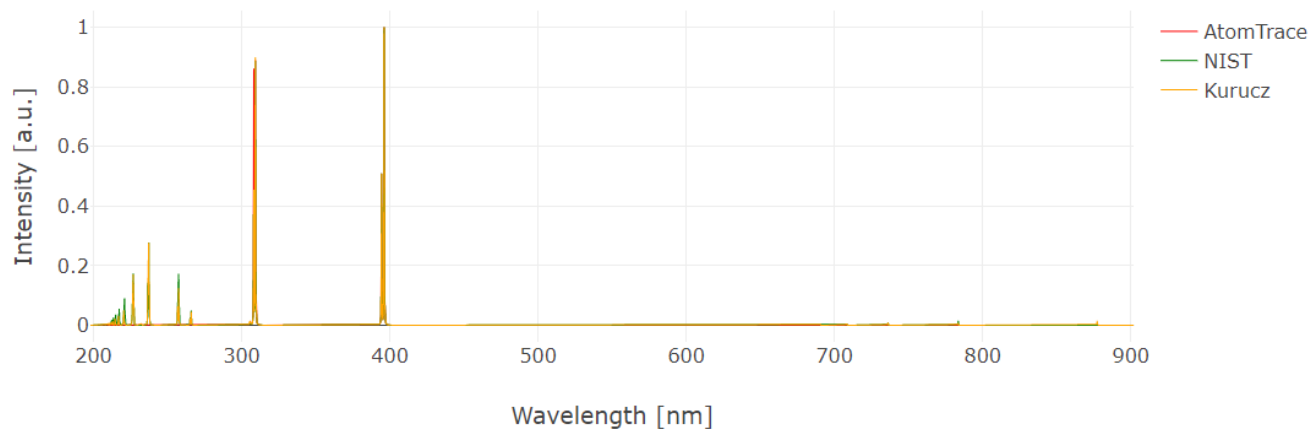


Рис. 4.1.3 – Емісійний спектр елементу Al

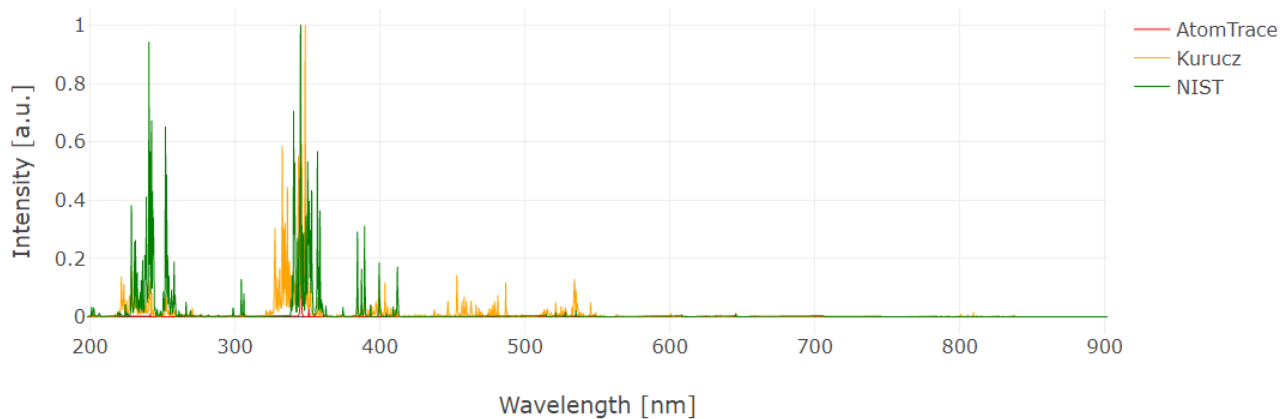


Рис. 4.1.4 – Емісійний спектр елементу CO

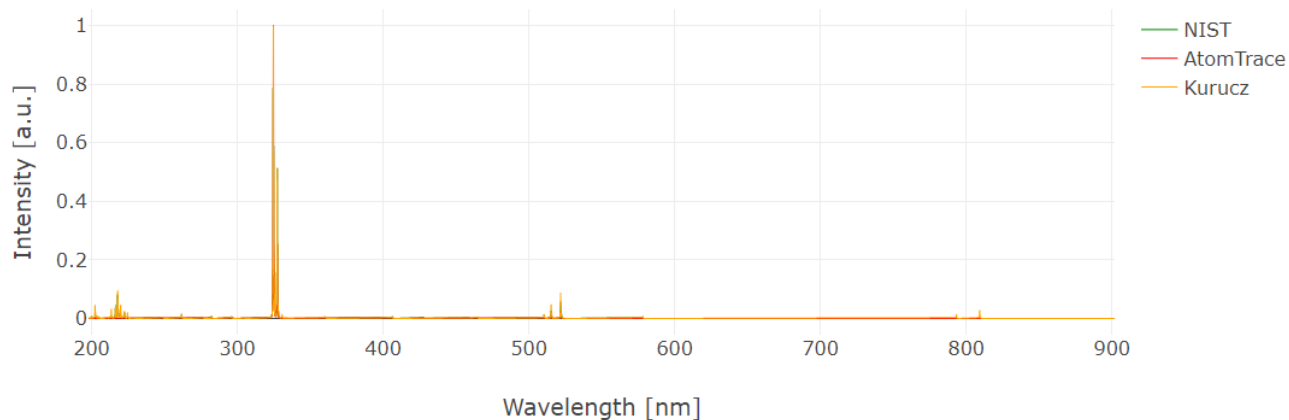


Рис. 4.1.5 – Емісійний спектр елемента CU

Полум'я розподілене по простору, різні точки мають різні яскравість та відтінки, тобто у нас є розподіл концентрації центрів висвічування по полю зору зображення полум'я.

Задля отримання коректних кінцевих результатів було синтезовано спеціальні таблички кольорності із попередньо заданими ваговими коефіцієнтами. Для відшукання вагових коефіцієнтів елементів зображення RGB формату був використаний метод OLAM-декомпозиції, в основі якого лежить принцип суперпозиції. У процесі пошуку коректного розв'язку задачі декомпозиції було встановлено, що одного трикомпонентного зображення нам не достатньо, оскільки в базисі 12 компонентів. Тому ми використали декілька зображень об'єкта, отриманих з різними світлофільтрами, наведені на Рис. 4.1.6.

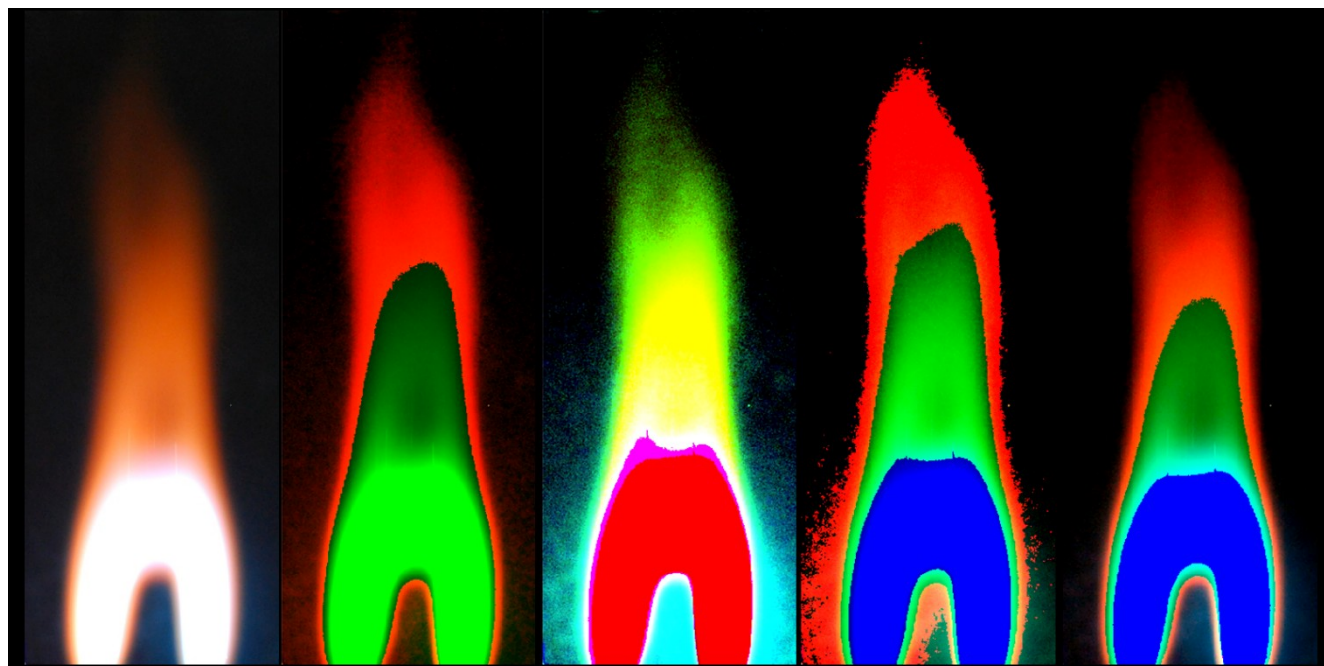


Рис. 4.1.6 – Зображення отримані за допомогою фільтрів.

Згаданий метод дозволяє за допомогою прямих формул та наявних компонентів, отримати розподіл їх концентрацій по полю зору зображення. На Рис. 4.1.7 зображені розподіли концентрацій елементів Al, CN, CO. Решта розподілів елементів Cu, Mg, Mn, N₂, N₂i, NH, NO, O₂, OH, наведені у Додатку Б.

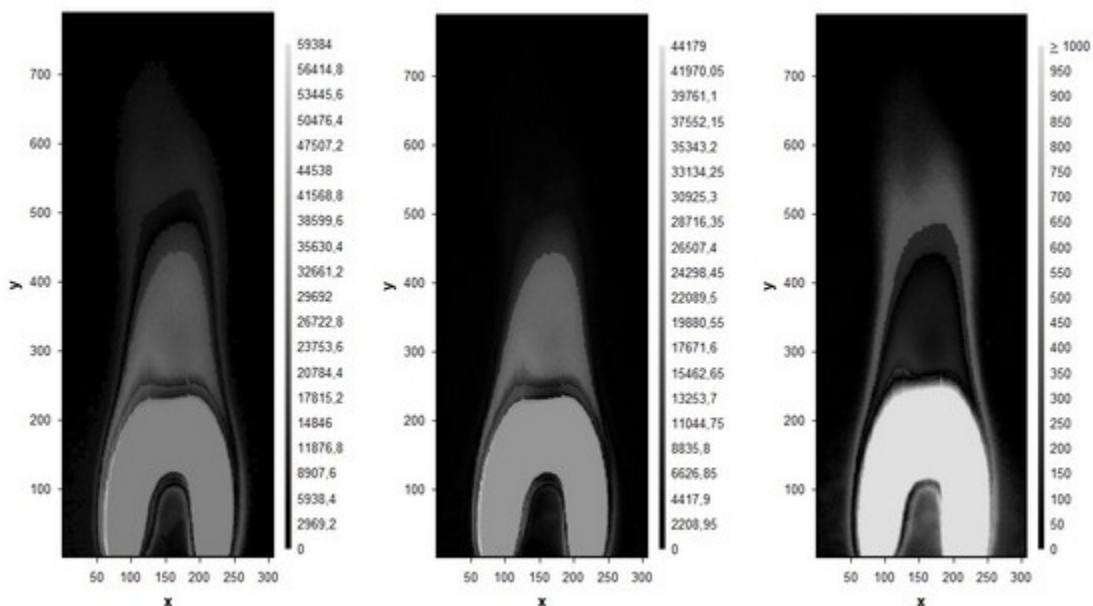


Рис. 4.1.7 – Розподіл концентрацій по полю зору зображення елементів Al, CN, CO.

4.2. Вплив рівня шуму на достовірність алгоритмів.

Особливість точного розв'язку задачі декомпозиції сигналу з шумом полягає в сильній чутливості розв'язку до шумової складової. Тобто вектор розв'язку систем неоднорідних лінійних рівнянь для сигналу з шумом перетворюється у випадкову величину, дисперсія якої більше ніж на порядок перевищує дисперсію шуму. При цьому структура точного рішення не дозволяє встановити явний вид закону розподілу ймовірності для вектора вагових коефіцієнтів \vec{a} . Таким чином для сигналу з шумом має місце похибка встановлення значень вагових коефіцієнтів компонентів суміші шляхом декомпозиції. Отже є важливим провести оцінки цієї похибки в залежності від характеристик адитивного шуму в вихідному сигналі.

Дослідження, проведені в роботі *Using Sensors as Analyzers for Decomposition and Classifying the Combined Spectral Signals of Various Nature*, показали наявність двох діапазонів відносної потужності шуму, в межах яких відносна поведінка похибки методу суттєво відрізняється. Так, при значенні відносної потужності шуму $\leq 75\%$ для MVSS-спектрометрії відносна похибка визначення вагових коефіцієнтів розкладання змінюється мало і не перевищує 4%, що є прийнятним для більшості випадків практичного використання. При збільшенні відносної потужності шуму від 75 до 99% спостерігається різке зростання відносної похибки від 4 до 30% з перевищенням 10% порогу допустимості при відносній потужності шуму 85%.

4.3. Шум квантування для RGB форматів.

Для кроку квантування інтенсивності RGB-стандарту використовується

8-бітове кодування, що надає 255 рівнів інтенсивності для кожного каналу (R, G, B). Для розрахунку потужності шуму квантування використана формула для квантування шуму:

$$P_{noise} = \frac{Q^2}{12}$$

де P_{noise} - потужність шуму квантування, Q - крок квантування. У даному випадку $Q = 1$.

$$P_{noise} = \frac{1^2}{12} = \frac{1}{12}$$

Таким чином, потужність шуму квантування становить $1/12$.

Максимальний рівень потужності для RGB сигналу можна визначити, таким чином:

$$P_{max} = (2^8 - 1)^2 = 255^2 = 65025$$

Значення SNR (відношення сигнал-шум) визначаємо, використовуючи формулу:

$$SNR = 10 * \ln \left(\frac{P_{signal}}{P_{noise}} \right)$$

де P_{signal} - потужність сигналу, P_{noise} - потужність шуму. За умовами, потужність сигналу буде максимальна, тобто $P_{signal} = P_{max}$.

$$SNR = 10 * \ln \left(\frac{65025}{\left(\frac{1}{12} \right)} \right) = 10 * \ln (65025 * 12) \approx 10 * \ln (780300) \approx 69,96 \text{ дБ}$$

Таким чином, значення SNR для RGB сигналу становить близько 69.93 дБ.

Отже, опираючись на графік 4.2.1, взятий зі статті [1], робимо висновок, що достовірність OLAM-декомпозиції виходить за межі стандартної похибки експерименту.

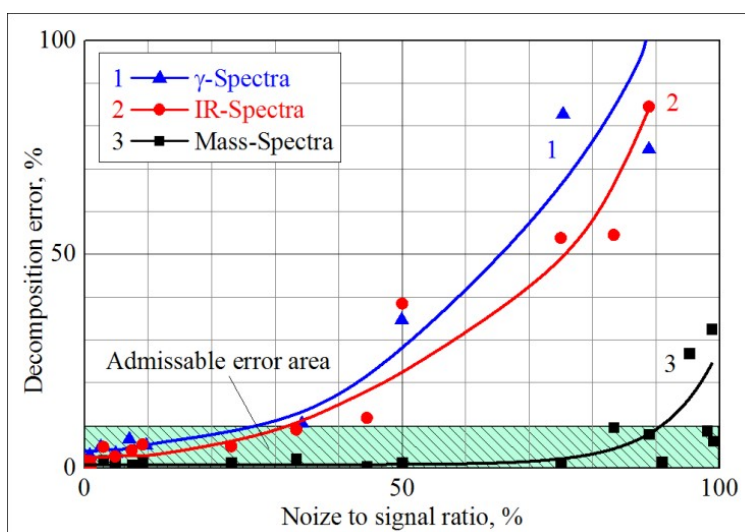


Рис. 4.2.1 - Залежність відносної похибки визначення коефіцієнтів розкладу від відношення потужності шуму до потужності вихідного сигналу.

Виходячи з результатів розрахунку шуму квантування для форматів RGB зображень, можна зробити висновок, що для застосованого нами методу OLAM-декомпозиції зображення в RGB-форматі не підходять, бо рівень шуму значно перевищує максимально допустимий. На відміну від pdf, jpeg, bmp, png і т.д. RAW формат зображення є некомпресованим форматом, який зберігає повний набір даних, знятих зі сенсора фотокамери без будь-якої обробки. RAW формат з досить високою точністю зберігає інтенсивність сигналу CCD матриці фотокамери, шум квантування у ньому відсутній.

Основна перевага RAW полягає в тому, що він зберігає більшу кількість інформації про зображення, порівняно з іншими форматами.

Дослідження придатності зображень в RAW-форматі для коректної OLAM-декомпозиції виходять за межі даної роботи і планується проводитись в майбутньому.

ВИСНОВКИ

В результаті проведеної роботи розроблено метод визначення просторового розподілу концентрації центрів випромінювання на основі усередненого спектру випромінювання та зображення протяжного об'єкта, названий «OLAM-декомпозиція».

Розроблено і реалізовано на платформі R 3.5.3 оригінальний алгоритм, що втілює розроблений метод OLAM-декомпозиції. Створений алгоритм успішно апробований на експериментальних емісійних спектрах і реальних

зображеннях факелу прямогочного плазмотрона.

Відношення корисного сигналу до шуму квантування в зображеннях стандартних форматів унеможлиблює коректне використання OLAM-декомпозиції. Значення вагових коефіцієнтів, розрахованих на основі зображення в RGB форматі, виходять за межі допустимої похибки методу. Для усунення цієї проблеми можна рекомендувати застосовувати алгоритми OLAM-декомпозиції для зображень в RAW форматі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Serge Olszewski. Using Sensors as Analyzers for Decomposition and Classifying the Combined Spectral Signals of Various Nature (Digital Object Identifier 10.1109/ACCESS.2017.Doi Number).
2. Yi Yang, Dong Xu. Image Clustering Using Local Discriminant Models and Global Integration// IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 19, NO. 10, OCTOBER 2010, 2761.
3. Sommerfeld, Arnold (1919). Atombau und Spektrallinien [Atomic Structure and Spectral Lines]. Braunschweig: Friedrich Vieweg und Sohn. ISBN 3-87144-484-7.
4. NIST (National Institute of Standards and Technology). URL: <https://www.nist.gov/> (дата звернення: 28.04.2023).
5. AtomTrace. URL: <https://www.atomtrace.com/> (дата звернення: 28.04.2023).
6. Fatima Nabeel, Syed Nabeel Asghar. Optimizing Image Segmentation by Selective Fusion of Histogram based K-Means Clustering: Proceedings of 2015 12th International Bhurban Conference on Applied Sciences & Technology (IBCAST) Islamabad, Pakistan, 13th – 17th January, 2015.
7. Messiah, Albert (1999). Quantum Mechanics. New York: Dover. p. 1136

ДОДАТОК А

ПРОГРАМНИЙ КОД

```

library(pracma) # пакет для математичного аналізу
library(magick); # пакет для оброблення та перетворення зображення
library(schemr); # пакет для сегментації та аналізу зображень

# Визначає функцію Flg_Avr, яка виконує обчислення ковзного середнього

Flg_Avr <- function(Wnd, Strim)
{
  N <- length(Strim)-Wnd;
  Strt_Q <- data.frame(s1=1:Wnd);
  Ret_strim <- data.frame(s1=1:N);
  for (i in 1:Wnd){ Strt_Q$s1[i] <- Strim[i]; };
  Ret_strim$s1[1] <- mean(Strt_Q$s1);
  for (i in 2:N)
  {
    Ret_strim$s1[i]<-Ret_strim$s1[i-1]+
    (Strim[i-1+Wnd]-Strim[i-1])/Wnd;
  };
  return(Ret_strim);
}

# Визначає функцію Vawe2CIE_F_, яка виконує обчислення компонентного
складу кольорів на основі вхідних даних

Vawe2CIE_F_ <- function(Vdg, Sp_bss, Flt, W) # Slave
{
  m <- ncol(F);
  p <- vector("double",3*(m+1));
  q <- as.matrix(Sp_bss)%*%W;
  clr <- Vdg[,2:4]*q;
  for(i in 1:3){p[i] <- sum(clr[,i])};};
  for(j in 1:m){for(i in 1:3){p[3*j+i]<- sum(clr[,i]*F[,j])};};};
  return(p);
}

# Визначає функцію image_xyz, яка перетворює вхідне зображення Img у
просторові координати XYZ

image_xyz <- function(Img)
{
  Img_I <- image_info(Img);N<-Img_I$width; M <- Img_I$height;
  m_bmp <- Img[[1]];
  w <- array(NA,c(3,N,M));
  t_clr <- data.frame(
  r = as.integer(m_bmp[1,,]),
  g = as.integer(m_bmp[2,,]),
  b = as.integer(m_bmp[3,,])
  );
  q <- as.matrix(rgb_to_xyz(t_clr,transformation=im_i$colorspace));
  J1<-1; J2<-N;
  for(i in 1:M){w[,1:N,i]<-p[,J1:J2]; J1<-J1+N; J2<-J2+N;};
  return(q);
}

```

Визначає функцію Ifcf, яка виконує обробку зображення Img за допомогою фільтра Flt

```
Ifcd <- function(Img,Flt)
{
im_i <- image_info(Img); N<-im_i$width; M <- im_i$height;
w <- q <- p <- array(NA,c(3,N,M));
w_bmp <- Img[[1]];
for(i in 1:N)
{
p[,i,] <- t(as.matrix(xyz_image(Img,i)));
for(j in 1:M){q[,i,j] <- dccd(p[,i,j],Flt);}
w[,i,]<-t(
as.matrix(
xyz_to_rgb(data.frame(x=q[1,i,],y=q[2,i,],z=q[3,i,]))
)
);
};
w<-255*w/max(w);
w_bmp[,,]<-as.raw(w[,,]);
Ret_W <- image_read(w_bmp);
return(Ret_W);
}
```

Визначає функцію cramer_inv, яка обчислює обернену матрицю методом Крамера

```
cramer_inv <- function(A)
{
n <- nrow(A)
detA <- det(A)
b <- matrix(NA, nrow = n, ncol = n)
for (i in 1:n)
{
for (j in 1:n) {b[i, j] <- (-1)^(i+j) * det(A[-j, -i])/detA}
}
return(b);
}
```

Визначає функцію Vawe2CIE_n, яка виконує перетворення вектора Vdg у просторові координати CIE XYZ за допомогою вектора Sp_bss та матриці W

```
Vawe2CIE_n <- function(Vdg, Sp_bss, W)
{
g <- Vdg[,2:4];
q <- as.matrix(Sp_bss)%*%W;
clr <- g*q;
clr[,1]<-(095.047*clr[,1])/(length(clr[,1])*max(clr[,1]));
clr[,2]<-(100.000*clr[,2])/(length(clr[,2])*max(clr[,2]));
clr[,3]<-(108.883*clr[,3])/(length(clr[,3])*max(clr[,3]));
Ret_XYZ <- as.vector(c(sum(clr[,1]),sum(clr[,2]),sum(clr[,3])));
return(Ret_XYZ);
}
```

```

# Визначає функцію Vawe2CIE, яка виконує обчислення
компонентного складу кольорів на основі вхідних даних Vdg,
Sp_bss та W

Vawe2CIE <- function(Vdg, Sp_bss, W)
{
  g <- Vdg[,2:4];
  q <- as.matrix(Sp_bss)%*%W;
  clr <- g*q;
  Ret_XYZ <- as.vector(c(sum(clr[,1]),sum(clr[,2]),sum(clr[,3])));
  return(Ret_XYZ);
}

# Визначає функцію Vawe2CIE_F, яка виконує обчислення компонентного
складу кольорів на основі вхідних даних Vdg, Sp_bss, Flt та W

Vawe2CIE_F <- function(Vdg, Sp_bss, Flt, W)
{
  m <- ncol(Flt);
  p <- vector("double",3*(m+1));
  p[1:3] <- Vawe2CIE(Vdg, Sp_bss, W);
  for(j in 1:m)
  {
    k<-3*j+1; l<-3*j+3;
    f<-Vdg; f[,2:4]<-f[,2:4]*Flt[,j];
    p[k:l]<- Vawe2CIE(f, Sp_bss, W);
  };
  return(p);
}

# Визначає функцію xyz_image, яка виконує перетворення кольорового
зображення в просторові координати XYZ на основі вхідних даних
Img(зображення) та rastr_n (номер растра).

xyz_image <- function(Img, rastr_n)
{
  Img_I <- image_info(Img); m_bmp <- Img[[1]];
  t_clr <- data.frame(
    r = as.integer(m_bmp[1,rastr_n,]),
    g = as.integer(m_bmp[2,rastr_n,]),
    b = as.integer(m_bmp[3,rastr_n,])
  );
  Ret_XYZ <- as.matrix(
    rgb_to_xyz(t_clr, transformation = Img_I$colorspace)
  );
  return(Ret_XYZ);
}

# Визначає функцію xyz_image_F, яка виконує перетворення кольорових
зображень у просторові координати XYZ на основі вхідних даних
Img(масив зображень) та rastr_n (номер растра)

xyz_image_F <- function(Img, rastr_n)
{
  N=length(Img);
  im_i <- image_info(Img[1]); M <- im_i$height;
  p <- matrix(NA, nrow=M, ncol=3*N);
  for(j in 1:N)
  {
    k <- 3*(j-1)+1; l <- 3*(j-1)+3;
    p[,k:l] <- xyz_image(Img[j],rastr_n);
  };
  return(p);
}

```

Код визначає функцію `Weight_factors_cr`, яка обчислює вагові коефіцієнти за допомогою методу найменших квадратів.

```
Weight_factors_cr <- function(equ, res)
{
  n <- nrow(equ);
  Ret_W <- eye(n)%*%cramer_inv(equ%*%t(equ))%*%equ%*%res
  return(Ret_W);
}
```

Код визначає функцію `Weight_factors`, яка обчислює вагові коефіцієнти за допомогою методу найменших квадратів.

```
Weight_factors <- function(equ, res)
{
  n <- nrow(equ);
  Ret_W <- eye(n)%*%inv(equ%*%t(equ))%*%equ%*%res
  return(Ret_W);
}
```

Код визначає функцію `dccf`, яка обчислює кросс-кореляцію між двома сигналами x та y за допомогою швидкого перетворення Фур'є (FFT)

```
dccf <- function(x,y)
{
  z <- fft(x)*fft(y);
  Ret_W <- fft(z, inverse = TRUE)/length(z);
  return(Ret_W);
}
```

Код використовує метод зсуву та множення матриць для обчислення кросс-кореляції між двома сигналами

```
dccd <- function(x,y)
{
  N<-min(c(length(x),length(y)));
  z<-vector('double',N);
  p<-matrix(NA,N,N); p[1,]<-y[N:1];
  q<-t(as.matrix(x));
  for(i in 2:N){w<-p[i-1,1];p[i,1:N-1]<-p[i-1,2:N];p[i,N]<-w;};
  z<-(q%*%p)[N:1];
  return(z);
}
```

#####

Отримуємо з нормалізованого зображення набір побітових даних
`normal_img_data <- image_data(normalized_img)`

```
Ifc <- function(Img,Flt)
{
  im_i <- image_info(Img); N<-im_i$width; M <- im_i$height;
  w <- q <- p <- array(NA,c(3,N,M));
  w_bmp <- Img[[1]];
  for(i in 1:N)
  {
    p[,i,] <- t(as.matrix(xyz_image(Img,i)));
    for(j in 1:M)
    {
      z <- fft(p[,i,j])*fft(Flt);
      q[,i,j] <- as.double(fft(z, inverse = TRUE)/length(z));
    };
    w[,i,]<-t(
```

```

as.matrix(
xyz_to_rgb(data.frame(x=q[1,i,],y=q[2,i,],z=q[3,i,]))
)
);
};
w<-255*w/max(w);
w_bmp[,,]<-as.raw(w[,,]);
Ret_W <- image_read(w_bmp);
return(Ret_W);
}

# Код виконує фільтрацію кольорового зображення за допомогою
функції dccd та обробляється піксель за пікселем у циклі for

Ifcd <- function(Img,Flt)
{
  im_i <- image_info(Img); N<-im_i$width; M <-
im_i$height;
  w <- q <- p <- array(NA,c(3,N,M));
  w_bmp <- Img;
  for(i in 1:N)
  {
    p[,i,] <- t(as.matrix(xyz_image(Img,i)));
    for(j in 1:M){q[,i,j] <- dccd(p[,i,j],Flt)};
    w[,i,]<-t(
      as.matrix(
xyz_to_rgb(data.frame(x=q[1,i,],y=q[2,i,],z=q[3,i,]))
      )
    );
  };
  #w<-255*w/max(w);
  Ret_W <- w;
  return(Ret_W);
}

# Отримуємо з сегментованого зображення набір побітових даних
thresholded_img_data <- image_data(thresholded_img)

Ifcf <- function(Img,Flt)
{
  im_i <- image_info(Img); N<-im_i$width; M <- im_i$height;
  w <- q <- p <- array(NA,c(3,N,M));
  w_bmp <- Img[[1]];
  for(i in 1:N)
  {
    p[,i,] <- t(as.matrix(xyz_image(Img,i)));
    for(j in 1:M){q[,i,j] <- as.double(dccf(p[,i,j],Flt))};
    w[,i,]<-t(
      as.matrix(
xyz_to_rgb(data.frame(x=q[1,i,],y=q[2,i,],z=q[3,i,]))
      )
    );
  };
  #w<-255*w/max(w);
  w_bmp[,,]<-as.raw(w[,,]);
  Ret_W <- image_read(w_bmp);
  return(Ret_W);
}

# Код виконує підготовку до обробки зображень пламенів

flame_path<-"d:/Work/FLAME_Treatment/BMP/WRK/";
flame_name <- c(

```

```

"TEST_FLAME_F_0.bmp",
"TEST_FLAME_F_1.bmp",
"TEST_FLAME_F_2.bmp",
"TEST_FLAME_F_3.bmp",
"TEST_FLAME_F_4.bmp"
);
vd_func_src <- "d:/Work/FLAME_Treatment/R_SCRIPT/Vidget_function.txt";
spb_src <- "d:/Work/FLAME_Treatment/R_SCRIPT/spectra_basis.txt";
flt_src <- "d:/Work/FLAME_Treatment/R_SCRIPT/Filters.txt";
flame_src <-gsub(" ", "", paste(flame_path, flame_name));
#-----#
vd_func <- read.table(vd_func_src, header = TRUE);
spectra_set <- read.table(spb_src, header = TRUE);
filter_set <- read.table(flt_src, header = TRUE);
flame_image <- image_read(flame_src);
fl_i <- image_info(flame_image[1]);
N <- fl_i$height; M <- fl_i$width;

# Перетворення отриманих даних у вектор чисел типу double
thresholded_img_data_num <- as.numeric(thresholded_img_data)

st <- matrix(NA, nrow=16, ncol=13);st[,1]<-0; st[1:12,2:13] <-
eye(12); # Спектральний колір кожного окремого елемента
st[13,] <- c(0,03,07,14,09,02,01,06,04,03,01,08,09);
st[14,] <- c(0,25,27,08,24,14,01,00,03,17,20,26,02);
# Тестові приклади кольорів #
st[15,] <- c(0,26,02,19,27,01,08,24,12,16,15,17,06);
# сумішевих спектрів #
st[16,] <- c(0,15,14,22,27,21,02,09,04,19,11,20,01);
sp_f<-spectra_set;
# Завантаження базиса спектрів центрів висвічування
L<-length(spectra_set[,1]);
sp_fs<-array(NA,c(5,L,13));
for(i in 1:13){sp_fs[1,,i]<-spectra_set[,i];};
for(i in 2:5)
{
sp_fs[i,,1]<-spectra_set[,1];
for(j in 2:13)
{
sp_fs[i,,j]<-spectra_set[,j]*
(filter_set[,i]/max(filter_set[,i]));
# Фільтрація спектрів
};
};
#-----#
q <- matrix(NA, nrow=16, ncol=3);
f <- matrix(NA, nrow=16, ncol=15);
for (j in 1:5)
{
F1<-(j-1)*3+1;F2<-3*j;
for (i in 1:16)
{
f[i,F1:F2]<- Vawe2CIE(vd_func, sp_fs[j,,], st[i,]);
p <- Vawe2CIE(vd_func, sp_fs[j,,], st[i,]);
q[i,]<-t(
# Формування rgb-вектора
as.matrix(
# одного пікселя гіперкуба
xyz_to_rgb(
#фільтрованих зображень.
data.frame(x=p[1],y=p[2],z=p[3])
)
)

```

```

        )
      );
      f[i,F1:F2]<-q[i,];
    };
  };
#-----#
W<-image_blank(200, 200, "black")[[1]];
u <- array(NA, c(15,200,200));
G_rgb<-array(NA,c(3,200,200));
K<-c(95,100,109);
h<-50;
#-----#
c<-a<-1;d<-b<-h;
for(i in 1:4)
{
  for(j in 1:4)
  {
    k<-4*(i-1)+j;
    u[,a:b,c:d]<-f[k,];
    a<-a+h;b<-b+h;
  };
  a<-1;b<-h;c<-c+h;d<-d+h;
};
#-----#
G_rgb[,,]<-u[10:12,,];
for(i in 1:3)
{
  if(max(G_rgb[i,,])!=0)
  {
    W[i,,]<-as.raw(K[i]*G_rgb[i,,]/max(G_rgb[i,,]));
  };
};
image_ggplot(image_read(W));
#-----#
P<-image_blank(200, 200, "white")[[1]];
K<-c(95,100,109);
for(i in 1:3)
{
  P_xyz[i,,] <- K[i]*(G[i,,]/max(G_xyz[i,,]));
  P[i,,] <- as.raw(P_xyz[i,,]);
};
image_ggplot(image_read(P));

# Змінити шлях до файлів зображень, що містяться в векторі test_img.

test_img<-c(
  "Test_TBL_0.png",
  "Test_TBL_1.png",
  "Test_TBL_2.png",
  "Test_TBL_3.png",
  "Test_TBL_4.png"
);
for(i in 1:5)
{
  test_img[i]<-gsub(" ", "", paste(flame_path,test_img[i]));
};
#-----#
# image_write(image_read(W),test_img[5],"png");

# Код встановлює шляхи до файлів зображень, що містяться в векторі
test_tbl_nm, і завантажує ці зображення за допомогою функції

```

```
image_read()
```

```

test_tbl_path<-"d:/Work/FLAME_Treatment/BMP/WRK/";
test_tbl_nm <- c(
  "Test_TBL_0.png",
  "Test_TBL_1.png",
  "Test_TBL_2.png",
  "Test_TBL_3.png",
  "Test_TBL_4.png"
);
vd_func_src <- "d:/Work/FLAME_Treatment/R_SCRIPT/Vidget_function.txt";
spb_src <- "d:/Work/FLAME_Treatment/R_SCRIPT/spectra_basis.txt";
flt_src <- "d:/Work/FLAME_Treatment/R_SCRIPT/Filters.txt";
test_tbl_src <-gsub(" ", "", paste(test_tbl_path,test_tbl_nm));
#-----#
vd_func <- read.table(vd_func_src, header = TRUE);
spectra_set <- read.table(spb_src, header = TRUE);
filter_set <- read.table(flt_src, header = TRUE);
test_tbl_image <- image_read(test_tbl_src);
fl_i <- image_info(test_tbl_image[1]);
N <- fl_i$height; M <- fl_i$width;

# У цьому коді відбувається обробка зображень з test_tbl_image і
отримання певних характеристик для кожного зображення.

# У цьому коді проводиться обробка та створення зображень з використанням
різних операцій

test_tbl_path<-"d:/Work/FLAME_Treatment/BMP/WRK/";
distr_nm <- c(
  "Al_distr_test.txt",
  "Cu_distr_test.txt",
  "Mg_distr_test.txt",
  "Mn_distr_test.txt",
  "N2_distr_test.txt",
  "N2+_distr_test.txt",
  "O2_distr_test.txt",
  "NO_distr_test.txt",
  "NH_distr_test.txt",
  "OH_distr_test.txt",
  "CO_distr_test.txt",
  "CN_distr_test.txt"
);
#-----#
-----#
for(i in 1:12)
{
  distr_fl <- data.frame(x = Dst[,1], y = Dst[,2], z = Dst[,i+2]);
  distr_dst <-gsub(" ", "", paste(test_tbl_path, distr_nm[i]));
  write.table(distr_fl,distr_dst,row.names = FALSE);
};
#=====
=====#
#-----#
-----#
# xyz_to_rgb(data.frame(x = 95, y = 100, z = 109));
# rgb_to_xyz(data.frame(r = 255, g = 255, b = 255 ));
# x y z
# A 109.850 100.000 035.585
# B 099.072 100.000 085.223
# C 098.074 100.000 118.232
# D50 096.422 100.000 082.521
# D55 095.682 100.000 092.149
# D65 095.047 100.000 108.883 +

```

```

# D75 094.972 100.000 122.638
# E 100.000 100.000 100.000
# F2 099.186 100.000 067.393
# F7 095.041 100.000 108.747 +
# F11 100.962 100.000 064.350
# max(vd_func[,2])
# flt_4<-Vawe2CIE(vd_func, filter_set,c(0,0,0,0,1));
# x <- 1:3;
# y <- 3:5;
# z <- fft(x)*fft(y);
# fft(z, inverse = TRUE)/length(z);
#-----#
-----#
  Img<-flame_image[1];
  Flt<-flt_1;
  Flt<-flt_2;
  Flt<-flt_3;
  Flt<-flt_4;
  Flt<-c(10,100,0)
  im_i <- image_info(Img); N<-im_i$width; M <- im_i$height;
# N = 307; M=789;
#-----#
-----#
  w <- q <- p <- array(NA,c(3,N,M));
  w_bmp <- Img[[1]];
  for(i in 1:N)
  {
    p[,i,] <- t(as.matrix(xyz_image(Img,i)));
    for(j in 1:M)
    {
      z <- fft(p[,i,j])*fft(Flt);
      q[,i,j] <- as.double(fft(z, inverse = TRUE)/length(z));
    };
    w[,i,]<-
t(as.matrix(xyz_to_rgb(data.frame(x=q[1,i,],y=q[2,i,],z=q[3,i,]))));
  };
  w<-255*w/max(w);
  w_bmp[,,]<-as.raw(w[,,]);
#-----#
-----#
  image_ggplot(Ifc(flame_image[1],flt_2));
  image_ggplot(flame_image[1]);
#=====
=====#
xyz_to_rgb(data.frame(x=q[1,150,],y=q[2,150,],z=q[3,150,]));
z <- fft(p[,150,300])*fft(c(0,0,1));
# plot(f[1,150,], type="l", lwd=2, col="blue");
#-----#
-----#
# N = 307; M=789;
  Img<-flame_image[1];
  m_bmp<-Img[[1]];
  im_i <- image_info(Img); N<-im_i$width; M <- im_i$height;
  w <- array(NA,c(3,N,M));
  q <- image_xyz(flame_image[1]);
  p <- t(as.matrix(xyz_to_rgb(data.frame(x=q[1,,],y=q[2,,],z=q[3,,]))));
  J1=1;J2=M;for(i in 1:N){w[,i,1:M]<-p[,J1:J2]; J1 <- J1+M; J2 <-
J2+M;};
  m_bmp[,,]<-as.raw(w[,,]);
  image_ggplot(image_read(m_bmp));

```

У цьому кодi виконуються деякі операції з векторами x та y.

Основні дії, які відбуваються в кодi:

Задаються значення для векторів x і y.

Викликається функція `dccf(x, y)`, яка обчислює зведену кореляцію між векторами x і y. Результат обчислення функції перетворюється на тип `double` за допомогою функції `as.double()`.

Викликається функція `dccd(x, y)`, яка обчислює зведену відстань між векторами x і y. Результат обчислення функції повертається.

Код # !mP\$Wi-Fi є коментарем і не впливає на виконання програми.

```
x<-c(22,118,20);y<-c(111,11,1);
```

```
as.double(dccf(x,y));
```

```
dccd(x,y);
```

```
#=====
=====#
```

ДОДАТОК Б

ЗОБРАЖЕННЯ РОЗПОДІЛУ ЕЛЕМЕНТІВ

