

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії

(повне найменування інституту, назва факультету (відділення))

Кафедра обчислювальної техніки

(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до магістерської кваліфікаційної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему Засоби ущільнення зображень за допомогою методів паралельної
обробки даних

Виконав: студент 2 курсу, групи 2КІ-18М
напряму підготовки (спеціальності)

123 Комп'ютерна інженерія

(шифр і назва напряму підготовки, спеціальності)

Пуцал Д.О.

(прізвище та ініціали)

Керівник к.т.н. доц. Семеренко В.П.

(прізвище та ініціали)

Рецензент к.т.н. доц. Поплавський А.В.

(прізвище та ініціали)

ВНТУ - 2019 року

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Інститут, факультет, _____ інформаційних технологій та комп'ютерної інженерії

Кафедра, циклова комісія _____ обчислювальної техніки

Освітньо-кваліфікаційний рівень _____ магістр

Напрямок підготовки _____ 12 Інформаційні технології

(шифр і назва)

Спеціальність _____ 123 Комп'ютерна інженерія

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ,

Д.т.н., проф. Мартинюк Т. Б.

« 2 » вересня 2019 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Пуцалу Дмитру Олеговичу

(прізвище, ім'я, по батькові)

1. Тема роботи Засоби ущільнення зображень за допомогою методів паралельної обробки даних

керівник проекту _____ к.т.н., доцент Семеренко В.П. _____,

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

затверджені наказом вищого навчального закладу від “__” _____ 201__ року №__

2. Строк подання студентом проекту (роботи) 12.12.2019

3. Вихідні дані до проекту (роботи) тип зображень: растрові зображення, операції обробки зображень: фільтрація, сегментація

4. Зміст розрахунково-пояснювальної записки Вступ, дослідження існуючих типів зображень, їх форматів та засобів ущільнення, аналіз апаратних та програмних мет розпаралелення, розробка паралельних алгоритмів ущільнення зображень, розробка паралельних алгоритмів сегментації зображень, апаратна та програмна реалізація паралельних алгоритмів обробки зображень, висновки, список використаних джерел.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Семеренко Василь Петрович, к.т.н., доцент каф. ОТ		
4	Глуценко Л.Д. к.е.н., доцент каф.ЕПВМ		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз, вибір та обґрунтування актуальності розробки	26.09.2019 р.	
2	Аналіз типів графічних даних	08.10.2019 р.	
3	Аналіз паралельних апаратних архітектур	16.10.2019 р.	
4	Аналіз методів фільтрації та модифікація для розпаралелення обчислень	02.11.2019 р.	
5	Реалізація алгоритму паралельної обробки даних за допомогою технології CUDA	11.11.2019 р.	
6	Реалізація методу паралельної фільтрації зображень	23.11.2019 р.	
7	Економічна частина	09.12.2019 р.	
8	Оформлення пояснювальної записки, графічного матеріалу та презентації	11.12.2019 р.	

Студент _____ Пуцал Д.О.
(підпис) (прізвище та ініціали)

Керівник проекту _____ Семеренко В.П.

Засоби ущільнення зображень за допомогою методів паралельної обробки даних

Пуцал Дмитро Олегович

Семеренко Василь Петрович

2КІ-18м

РЕФЕРАТ

Магістерська кваліфікаційна робота присвячена дослідженню методів ущільнення зображень на основі фільтрації та сегментації зображень з використанням методів паралельної обробки даних. В роботі проаналізовано формати растрових зображень, методи ущільнення, фільтрації та сегментації. Проаналізовано математичні моделі різних типів фільтрації з позиції використання паралельної обробки даних. Обґрунтовано, що для DWT та DCT фільтрації доцільно використати модель матричної паралельної обробки даних запропоновано засоби ущільнення растрових зображень за допомогою методів паралельної обробки даних на основі OpenCV та OpenMP.

Модифіковано алгоритм ущільнення зображень та методи фільтрації зображень. Спрогнозовано прискорення роботи алгоритму у порівнянні з класичним алгоритмом послідовного виконання коду. В додатках наведена таблиця порівняння швидкості обробки зображень за допомогою модифікованого та класичного алгоритму, приклади фільтрації зображень.

Ключові слова: паралельні обчислення, фільтрація зображень, сегментація, OpenMP, ущільнення.

ABSTRACT

Image data compression using parallel methods of data processing

The master's degree work devoted to researching of image compression methods, image filtering and segmentation methods analysis of hardware and software methods of parallel data processing. During the master's work the types of images, raster image compression methods, filtering and segmentation of raster images were analyzed. Analogues are analyzed, their shortcomings are identified, and raster image compression methods are proposed using OpenCV and OpenMP parallel processing methods.

The image compression algorithm and image filtering methods have been modified. The algorithm is predicted to accelerate compared to the classic serial code algorithm. The appendices provide a table comparing image processing by using a modified and classical algorithm, examples of image filtering.

Keywords: parallel computing, image filtering, segmentation, OpenMP, compression.

ЗМІСТ

ВСТУП.....	8
1 ОГЛЯД ТИПІВ ЗОБРАЖЕНЬ, МЕТОДІВ ТА ЗАСОБІВ ПАРАЛЕЛЬНОЇ ОБРОБКИ ДАНИХ В ПК	10
1.1 Аналітичний огляд типів зображень використовуваних в ПК	10
1.2 Аналіз методів ущільнення зображень	14
1.3 Типовий алгоритм ущільнення растрових зображень	19
1.4 Типізація обчислювальних систем паралельної обробки даних	23
1.4.1 Аналіз паралельних архітектур.....	23
1.4.2 Проблеми розробки паралельного ПЗ	30
2 УЩІЛЬНЕННЯ ЗОБРАЖЕНЬ НА ОСНОВІ ПАРАЛЕЛЬНОЇ ОБРОБКИ ОПЕРАЦІЙ НАД ЗОБРАЖЕННЯМИ	31
2.1 Розпаралелення обчислень при фільтрації зображень	31
2.2 Ущільнення зображень на основі сегментації зображень.....	36
2.3 Градієнтні оператори та їх паралельна інтерпретація	39
2.4 Паралельна обробка при ущільненні зображень	42
2.5 Модифікації фільтрації дискретного-вейвлет перетворення	48
3 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ УЩІЛЬНЕННЯ ТА ФІЛЬТРАЦІЇ РАСТРОВИХ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ МЕТОДІВ ПАРАЛЕЛЬНОЇ ОБРОБКИ ДАНИХ	57
3.1 Засоби паралельного ущільнення на базі графічного процесора з використанням технології NVIDIA CUDA	57
3.2 Модифікація алгоритму фільтрації для виконання паралельних обчислень	66
4 ЕКОНОМІЧНА ЧАСТИНА.....	
4.1 Оцінювання комерційного потенціалу розробки засобів ущільнення ущільнення зображень за допомогою методів паралельної обробки даних.....	74
4.2 Прогнозування витрат на виконання науково-дослідної, (дослідно- конструкторської) та конструкторсько-технологічної роботи.....	77

					08-23.МКР.029.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розробив	Пуцал Д.О.				Засоби ущільнення зображень за допомогою методів паралельної обробки даних. Пояснювальна записка.	Літ.	Аркуш	Аркушів
Перевірив	Семеренко В.П.							6
Реценз.	Поплавський А.В.					ВНТУ, 2КІ-18м		
Н. контр.	Швець С.І.							
Затверд.	Мартинюк Т.Б.							

4.3	Прогнозування комерційних ефектів від реалізації результатів розробки засобів паралельної ущільнення за допомогою методів паралельної обробки даних	81
4.4	Розрахунок ефективності вкладених інвестицій та періоду їх окупності.....	82
4.5	Висновки до економічного розділу.....	85
	ВИСНОВКИ.....	86
	ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	87
	Додаток А.....	90
	Додаток Б.....	93
	Додаток В.....	94
	Додаток Г.....	95
	Додаток Д.....	96
	Додаток Е.....	98

					08-23.МКР.029.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність роботи.

Паралельне програмування – область науки з високим потенціалом завдяки розвитку сучасних архітектур комп'ютерів. Використання інструментів паралельного програмування може бути корисним у мультимедійних програмах завдяки скороченню часу, необхідного для виконання складних обчислень.

Метою роботи є підвищення швидкості обробки зображень за допомогою розпаралелювання етапів ущільнення зображення.

Об'єктом дослідження є процеси, що відбуваються під час ущільнення та фільтрації растрових зображень.

Предметом дослідження є теоретичні моделі, методи та програмно- апаратні засоби паралельного ущільнення, фільтрації та сегментації зображень за допомогою паралельного програмування та апаратних паралельних архітектур.

Методи дослідження, що були використані для вирішення поставленої задачі: теорія цифрових зображень, теорія паралельних архітектур, теорія алгоритмів, об'єктно-орієнтоване програмування.

Наукова новизна одержаних результатів полягає в модифікації алгоритмів ущільнення та фільтрації растрових зображень для роботи з сучасними багатопоточними обчислювальними системами за допомогою програмно-апаратних засобів з використанням бібліотек OpenCV та OpenMP для розпаралелення процесів ущільнення, фільтрації та сегментації.

Публікації.

1. Пуцал Д.О. Паралельна обробка алгоритмів ущільнення текстових та числових даних *Матеріали конференції «XLVII Науково-технічна конференція підрозділів Вінницького національного технічного університету (2018)»*, Вінниця, 2018

2. Пуцал Д.О. Обробка цифрових зображень з використанням методів паралельної обробки даних *Всеукраїнська науково-практична інтернет*

*конференція «Молодь в науці: дослідження, проблеми, перспективи (2020)»,
Вінниця, 2020*

1 ОГЛЯД ТИПІВ ЗОБРАЖЕНЬ, МЕТОДІВ ТА ЗАСОБІВ ПАРАЛЕЛЬНОЇ ОБРОБКИ ДАНИХ В ПК

1.1 Аналітичний огляд типів зображень використовуваних в ПК

В комп'ютері будь-яка інформація не залежно від її представлення в системі складається з закодованої послідовності бітів - нулів та одиниць. Цей принцип представлення інформації справедливий і для такого широкого класу інформації як цифрові зображення.

Існує два принципово різних підходи того, яким чином можна представити зображення у вигляді нулів і одиниць: растрова графіка і векторна графіка.

При використанні растрової графіки з допомогою певного числа біт кодується колір кожного найдрібнішого елемента зображення - пікселя. Зображення представляє собою матрицю із пікселів кожен з яких має свої параметри кольору, яскравості а іноді й прозорості. В комп'ютері під кожен піксел відводиться певна кількість біт, яку називають бітовою глибиною. Будь-який колір має свій унікальний двійковий код. Наприклад, якщо бітова глибина дорівнює 1, тобто під кожен піксел відводиться 1 біт, то 0 відповідає чорному кольору, 1 - білому, а зображення може бути тільки чорно-білим. Якщо бітова глибина дорівнює 2, тобто під кожен піксел відводиться 2 біти, 00 - відповідає чорному кольору, 01 - червоного, 10 - синього, 11 - чорному, тобто в малюнку може використовуватися чотири кольори. Далі, при бітової глибині 3 можна використовувати 8 кольорів, при 4 - 16 і т.д. Тому, графічні програми дозволяють створювати зображення з 2, 4, 8, 16, 32, 64, ... , 256, і т.д. кольорів. З кожним збільшенням кількості кольорів (палітри) вдвічі, збільшується обсяг пам'яті, необхідний для запам'ятовування зображення.

Основним недоліком растрової графіки є великий обсяг пам'яті, потрібний для зберігання зображення. Це пояснюється тим, що потрібно запам'ятати колір кожного пікселя, загальне число яких може бути дуже великим.

В векторній графіці в пам'яті комп'ютера зберігається математичний опис кожного графічного примітива - геометричного об'єкта (наприклад, відрізка, кола,

прямокутника і т.п.), з яких формується зображення. Зокрема, для відтворення кола досить запам'ятати становище її центру, радіус, товщину і колір лінії. За цими даними відповідні програми побудують потрібну фігуру на екрані дисплея. Зрозуміло, що такий опис зображення вимагає набагато менше пам'яті (у 10-1000 разів) ніж в растровій графіці, оскільки обходиться без запам'ятовування кольору кожної точки малюнка. Векторні графічні зображення не залежать від дозволу, описуються набором параметричних рівнянь прямих, дуг кіл і кривих Безьє. У всіх векторних форматах об'єкти можуть варіювати товщину і колір контуру, а замкнуті об'єкти ще і колір заливки. Об'єкти можуть накладатися, частково або повністю затуляючи один одного. В якості окремих об'єктів можуть включатися растрові зображення і текст. Головна перевага векторної графіки в тому, що всі об'єкти незалежні і неможливо здійснити незворотні дії, втратити частину вихідної інформації. Що стосується векторної графіки в інтернеті, то вона цінна ще й тим, що через невеликий розмір її можна швидко передати.

Основним недоліком векторної графіки є неможливість роботи з високоякісними художніми зображеннями, фотографіями і фільмами. Природа уникає прямих ліній, правильних кіл і дуг. Саме з їх допомогою (оскільки ці фігури можна описати засобами математики, точніше - аналітичної геометрії) і формується зображення при використанні векторної графіки. Тому основною сферою застосування векторної графіки є оцифровка креслень, схем, діаграм та створення логотипів.

Якщо на екрані малюнок з близьким до природного зображенням, з великим числом кольорів і відтінків, то, швидше за все це растровий малюнок. Якщо на малюнку креслення, діаграма, простий стилізований малюнок - векторна. Якщо програма дозволяє видаляти, копіювати або переміщувати цілі фрагменти зображення, то це растрова графіка. Якщо видалити, скопіювати, перемістити можна тільки якісь певні фігури або їх частини, то це векторний рисунок.

На рисунку 1.1.1 наведено приклад використання векторної графіки – технологічна схема роботи текстильного принтера.

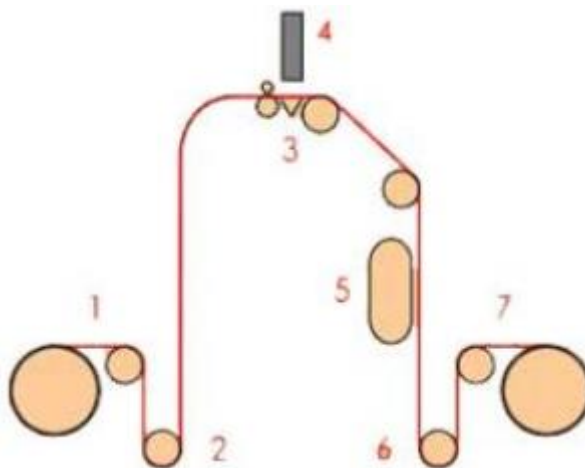


Рисунок 1.1.1 – Приклад використання векторної графіки

Файли **. bmp*, **. jpg*, **. gif*, **. msp*, **. img* та інші відповідають форматам растрового типу, **. dwg*, **. dxf*, **. pic* - векторного.

Для того, щоб наглядніше зрозуміти відмінності при практичному використанні різних видів графіки розглянемо наступний приклад. На рисунку 1.1.2 показано зображення складної геометричної форми. Рисунок був конвертований в растрову графіку і у векторну графіку. Після п'ятиразового збільшення копій фрагмента зображення видно їхні відмінності. На рисунку 1.1.3 зображена векторна копія зображення, тут при збільшенні розраховувалися більш точно графіки функцій кривих. На рисунку 1.1.4 растрова копія зображення, тут збільшення зображення відбувалося за рахунок збільшення розмірів самих пікселів.

У силу своїх особливостей (масштабування без втрати якості зображення, маленький розмір) векторна графіка використовується в поліграфії (шрифти, логотипи), в мережі (анімації, інтерфейс, банери), при конструюванні для зберігання креслень і схем.

Растрова графіка має якості, яких немає в векторній (незалежне редагування кожної деталі зображення до пікселя). Є незамінною в зображеннях, які дуже складні і їх неможливо описати математичними формулами (наприклад - фотографії). Саме тому растрова графіка застосовується практично у всіх областях, пов'язаних з комп'ютерною графікою(див. Додаток Б).

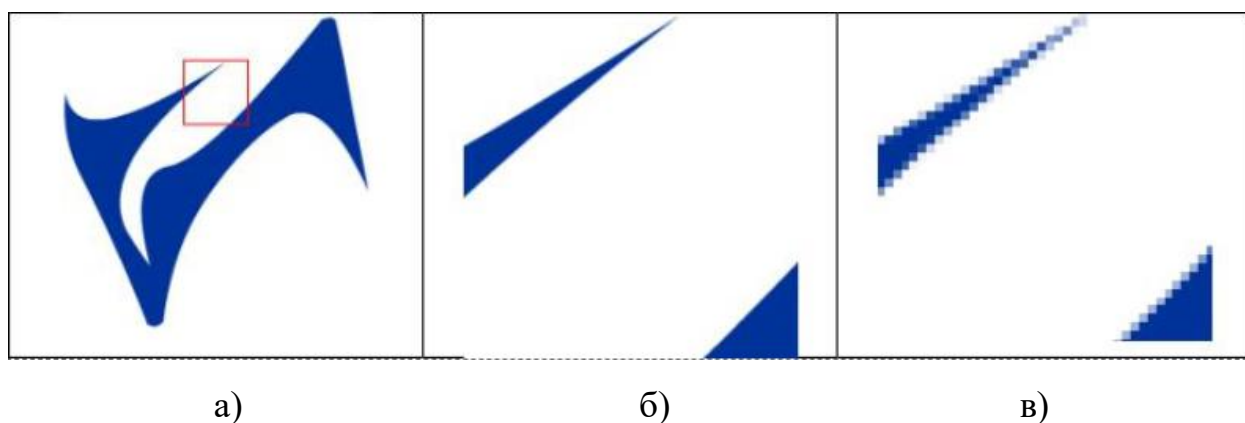


Рисунок 1.1.2: а – оригінал; б – векторне зображення в масштабі 5:1;
в – растрове зображення в масштабі 5:1

Роздільна здатність зображення показує, скільки пікселів міститься в одному дюймі (сантиметрі), і якщо відомі розміри зображення, можна точно сказати, скільки пікселів міститься в ньому. Це дробове значення часто виражають у пікселях на дюйм (pixels per inch - ppi). Наприклад якщо зображення має розмір, рівний одному квадратному дюйма, а дозвіл зображення дорівнює 8 ppi, то кількість пікселів, що містяться в ньому - 64 пікселі. Якщо дозвіл - 16 ppi, то зображення цього ж розміру міститимете 256 пікселів.

Ні 8, ні 16 пікселів на дюйм не можуть забезпечити створення якісного зображення. На рисунку 1.1.5 продемонстровано як саме виглядають зображення з різною щільністю пікселів.

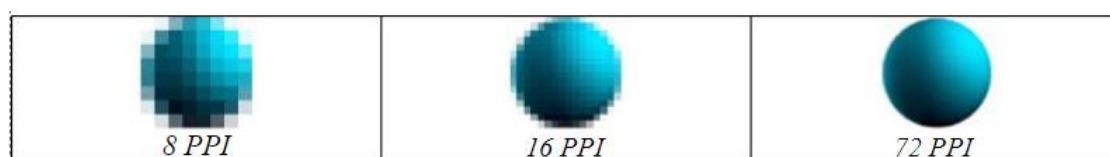


Рисунок 1.1.5 – Розширення растрового зображення.

Для збереження плавності кривих використовується метод згладжування контурів (анти-аліасінг). На рисунку 1.1.6 показана різниця між зображенням, де не використовувалося згладжування (ліворуч), і зображенням зі згладжуванням (праворуч).



Рисунок 1.1.6 – Ефект від використання згладжування.

Якщо збільшити растрове зображення, до якого застосовано ANTI-ALIASING, то спостерігається наступне явище: в міру віддалення від лінії, її колір ромивається фоном. Згладжування створює плавний перехід між внутрішньою областю форми та її фоном (Рисунок 1.1.7).



Рисунок 1.1.7 – Перехід кольору лінії в фон при згладжуванні

1.2 Аналіз методів ущільнення зображень

Методи ущільнення зображень ділять на дві умовні категорії - архівацію (ущільнення) та компресію (конвертацію). Різниця полягає в тому, що при конвертації ми можемо нехтувати якістю на користь ефективності ущільнення, але отримати точну копію зображення, що було на вході з конвертованого неможливо так як частина інформації в процесі обробки безповоротно видаляється. Яким б не був алгоритм компресії даних, для роботи з ним файл потрібно проаналізувати і розпакувати, тобто повернути дані в початковий стан для швидкої обробки.

Для зменшення обсягу даних, як один з можливих варіантів, проводиться аналіз на наявність однакових послідовностей даних, і при входженні вже раніше

вжитої послідовності бітів використовують адресу комірки з першим входженням такого ж фрагменту. Такими однаковими послідовностями можуть бути пікселі одного кольору, що повторюються або інша надлишкова інформація, яка в рамках даного масиву даних повторюється кілька разів.

Компресія (конвертація) даних - це спосіб збереження, при використанні якого не гарантується повне відновлення вихідних графічних даних. При такому способі зберігання даних зазвичай графічна інформація спотворюється в порівнянні з оригінальною, але цими спотвореннями можна керувати, і при їх невеликому значенні можна нехтувати. Файли, що збережені з використанням цього способу зберігання, займають значно менше дискового простору, ніж файли, збережені з використанням архівації. Як правило, при збереженні даних з використанням компресії, можна досягти компромісу між розміром вихідного файлу і його якістю. Зрозуміло, що можлива оптимізація тільки по одному параметру (чим гірша якість, тим менший обсяг вихідного файлу, і навпаки).

Головним недоліком компресії з частковою втратою якості є втрати, що виражаються у спотворенні колірному тону або появі характерної кубічної структури в контрастних ділянках зображення (проявляються так звані артефакти), що виникають кожного разу при збереженні зображення, і 'накладаються' один на одного при багаторазовому збереженні файлу в цьому форматі. Для вирішення цієї проблеми використовуються спеціальні маски – фільтри.

В якості допоміжних засобів, які можуть використовуватися для зниження обсягів файлів можна розглянути зміну колірної моделі графічного файлу, дозвіл растрового файлу і ресемплювання, тобто зміну глибини кольору пікселів.

Наприклад, файли в колірному просторі СМҮК більше аналогічних файлів в просторі RGB на 33% (так як в СМҮК є додатковий четвертий чорний канал).

Коли вперше постала проблема ущільнення зображень для них застосовували загальні, універсальні алгоритми, однак це виявилось неефективно для нових форматів зображень. Старі алгоритми не задовільняли потреби до архівації і більшість зображень практично не ущільнювалось, хоча на перший погляд мали надлишковість. Це привело до створення нового типу алгоритмів –

алгоритмів ущільнення з втратами. Як правило коефіцієнт архівації і відповідно ступінь втрат якості можна було задавати, при цьому зберігаючи баланс між якістю та розміром.

Одною з серйозних проблем графіки є те, що донині не знайдено адекватного критерію оцінки втрат якості зображень.

Існує 2 концептуально різних класи алгоритмів ущільнення зображень:

1) Алгоритм A називається алгоритмом ущільнення без втрат якщо існує алгоритм A^{-1} (зворотній до A) такий, щоб для будь-якого зображення I $A(I) = I_1$ та $A^{-1}(I_1) = I$. Зображення I задано як множину значень атрибутів пікселів. Після застосування до I алгоритму A отримуємо набір даних I_1 .

Ущільнення без втрат застосовується в таких форматах зображень як GIF, PCX, PNG, TGA, TIFF та деяких особливих форматах для цифрових фотокамер від їх виробників.

2) A називається алгоритмом ущільнення з втратами якщо він не забезпечує можливість точного відновлення джерела зображення. Парний до A алгоритм, що забезпечує приблизне відновлення, визначимо як A^* : для зображення I $A(I) = I_1$, $A^{-1}(I_1) = I_2$ і при цьому отримане відновлене зображення I_2 не обов'язково тотожне I . Пара A, A^* підбирається таким чином, щоб забезпечити високі коефіцієнти ущільнення і забезпечити мінімальну різницю при сприйнятті I та I_2 . Ущільнення з втратами використовують такі популярні формати як JPEG та JPEG2000 й маса інших, менш поширених.

Так як обсяг інформації, що необхідний для збереження зображень надзвичайно великий, а обчислювальні можливості сучасних комп'ютерів ростуть, стало очевидно що класичні алгоритми не забезпечують достатню ступінь ущільнення. Для зображень були розроблені алгоритми, що забезпечують ущільнення з втратами, що дозволило значно покращити ступінь компресії.

Ущільнення з втратами базується на особливостях сприйняття зображення людиною: найбільшою чутливістю в деякому діапазоні хвиль кольору, можливості сприймати зображення як одне ціле, не помічаючи спотворень. Також використовується той факт, що зображення – це двомірний об'єкт. Головний клас

зображень, на який орієнтовані алгоритми ущільнення з втратами є фотографії, зображення з плавними колірними переходами.

Щоб оцінити різницю між зображеннями I та I_2 при обробці зображень алгоритмом ущільнення з втратами, необхідно мати деякі міри, базовою з яких є візуальна схожість оригінального та ущільненого зображення, щоб розбіжності були непомітні оку людини.

Введемо спочатку норму для значень атрибутів пікселів. Для напівтонових зображень, де на кожне значення атрибута пікселя відводиться 8 біт:

$$|I(i,j)| = |I(I,j)|^2. \quad (1.1)$$

Причому $M^* = \max || \cdot ||$, тобто максимальне значення для такої норми буде дорівнювати $256^2=65025$. Для повнокольорових зображень з трьома 8-мібітними значеннями атрибута пікселів(трьома каналами):

$$||I(I,j)|| = |R(I,j)|^2 + |G(I,j)|^2 + |B(I,j)|^2. \quad (1.2)$$

$$\text{Відповідно } M^* = 3 \cdot 256^2 = 195075$$

Стандартною мірою для ущільнених зображень є відношення рівня сигналу до рівня шуму (PSNR - англ. Peak Signal-to-Noise Ratio), що визначається так:

$$PSNR(I_1, I_2) = 10 \log_{10}(), \quad (1.3)$$

де $MSE(I_1, I_2)$ – середньоквадратична помилка, що визначається так:

$$MSE(I_1, I_2) = . \quad (1.4)$$

Алгоритми ущільнення з втратами часто надають користувачеві можливість вибрати обсяг відкинутих даних. Таким чином, існує вибір між якістю і розміром зображення. Чим краща візуальна якість при більшому коефіцієнті ущільнення, тим алгоритм краще.

Розглянемо зображення як функцію двох змінних, визначену в точках кінцевого растра. $I(x, y)$ - значення атрибута пікселя (наприклад номер в палітрі, інтенсивність), що залежить від колірної моделі представлення зображення.

Множина таких функцій на точках фіксованого кінцевого растра утворює кінцевомірний евалідовий простір $R^{X, Y}$ розмірності $m \times n$ ($|X| = m$, $|Y| = n$) зі скалярним добутком $(I_1, I_2) =$

Ототожнимо з таким простором $L_2(X \times Y)$. У такому просторі існує базис, тобто така система елементів з $R^{X,Y}$ і такий набір не рівних одночасно нулю коефіцієнтів, що для будь-якої функції I з цього простору виконувалось $I =$.

Якщо додатково припустити ортонормальність базису, тобто

$$(e_p, e_q) = \delta_{pq},$$

то виконується наступне співвідношення:

$$C_k = (I, e_k). \quad (1.5)$$

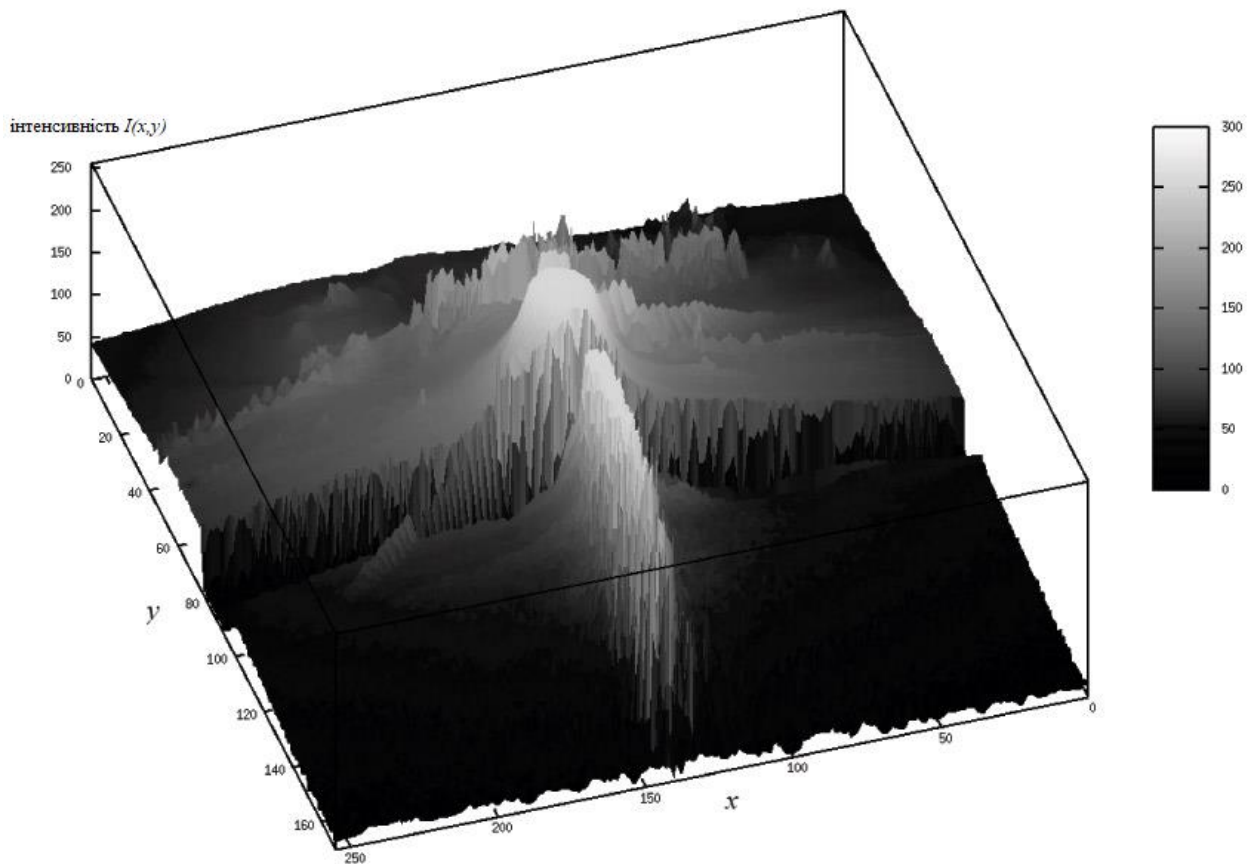


Рисунок 1.2 – Зображення як функція двох змінних

В той же час достатньо популярною моделлю визначення є дискретне перетворення Фур'є:

$$F(k,l) = \quad (1.6)$$

Для всіх $k = 1 \dots m, l = 1 \dots n$. Зворотне перетворення визначається так:

$$I(p,q) = \quad (1.7)$$

Система функцій визначає базис в просторі функцій-зображень.

Розглянемо визначення дискретного косинусного перетворення (ДКП). Нехай зображення має розміри $N \times N$. Пряме перетворення записується так:

$$t(u, v) = c(u)c(v) \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} I(k, l) \cos \frac{(2k+1)u\pi}{2N} \cos \frac{(2l+1)v\pi}{2N}, \quad (1.8)$$

$$i, j = 0, \dots, N-1, c(i) = \begin{cases} \sqrt{\frac{1}{N}}, & i = 0 \\ \sqrt{\frac{2}{N}}, & i \neq 0 \end{cases}.$$

Зворотнє перетворення має такий вигляд:

$$I(k, l) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} c(u)c(v)t(u, v) \cos \frac{(2k+1)u\pi}{2N} \cos \frac{(2l+1)v\pi}{2N}, \quad (1.9)$$

$$i, j = 0, \dots, N-1, c(i) = \begin{cases} \sqrt{\frac{1}{N}}, & i = 0 \\ \sqrt{\frac{2}{N}}, & i \neq 0 \end{cases}.$$

Дискретне перетворення має наступні властивості:

- Коефіцієнти незалежні один від одного, тобто точність представлення одного коефіцієнта не залежить від будь-якого іншого.
- Перетворення зберігає основну інформацію в малій кількості коефіцієнтів. Дана властивість найсильніше проявляється на фотореалістичних зображеннях.

Коефіцієнти $t(u, v)$ - це амплітуди просторових частот зображення. У разі зображень з плавними переходами більша частина інформації міститься в низькочастотному спектрі.

1.3 Типовий алгоритм ущільнення растрових зображень

Алгоритм ущільнення, який використовується в форматі зберігання зображень JPEG1, побудований на використанні дискретного косинусного перетворення (DWT). Схема ущільнення в алгоритмі являє собою конвеєр, де дане перетворення - лише одна зі стадій. Наведемо алгоритм, припускаючи, що на вхід дано 24-бітні зображення, значення атрибутів пікселів якого є елементами колірному простору RGB.

- 1) Переведення в колірний простір YCbCr, де Y - компонента яскравості, Cb і Cr - компоненти кольору. Людське око більш чутливе до яскравості, ніж до

кольору. Тому важливіше зберегти більшу точність при передачі яскравості, ніж при передачі кольору. Переведення здійснюється за такою формулою:

Зворотне перетворення:

2) Субдискретизація компонент кольору. Після переведення в колірний простір YCbCr здійснюється субдискретизація (кодування зображення зі зменшенням колірної здатності на противагу яскравості) по наступних співвідношеннях: 4: 4: 4 (відсутність передискретизації), 4: 2: 2 (компоненти кольоровості міняються через одну по горизонталі), 4: 2: 0 (компоненти кольоровості міняються через одну по горизонталі ; при цьому по вертикалі вони змінюються через рядок).

Далі компоненти обробляються і зберігаються окремо один від одного. Таким чином, в останніх двох випадках «відкинуто» та інформації відповідно.

Очевидно, що за відсутності субдискретизації ступінь ущільнення погіршиться, а при схемі 4: 2: 0 буде найбільшим. Якщо зображення не ділиться без остачі на блоки 4 x 4 піксела - воно доповнюється, тобто в разі якщо розмір по вертикалі не ділиться на 4, то додається ще від одного до трьох рядків. Аналогічно і для випадків коли розмір по горизонталі не ділиться на 4 - додаються стовпці, що збігаються з самим правим.

3) Застосування дискретного косинусного перетворення (ДКП). Зображення розбивається на блоки 8 x 8; до кожного блоку застосовується дискретне косинусне перетворення (окремо для компонент Y, Cb і Cr).

4) Квантування. Людське око практично не помічає зміни в високочастотних складових, отже, коефіцієнти, що відповідають за високі частоти, можна зберігати з меншою точністю. Квантування здійснюється за допомогою множення матриці коефіцієнтів ДКП на так звану матрицю квантування:

$$\begin{pmatrix} t_{11} & \dots & t_{18} \\ \vdots & \ddots & \vdots \\ t_{81} & \dots & t_{88} \end{pmatrix} \otimes \begin{pmatrix} q_{11} & \dots & q_{18} \\ \vdots & \ddots & \vdots \\ q_{81} & \dots & q_{88} \end{pmatrix} = \begin{pmatrix} T_{11} & \dots & T_{18} \\ \vdots & \ddots & \vdots \\ T_{81} & \dots & T_{88} \end{pmatrix},$$



Рисунок 1.3.2 – Артефакти ущільненого в 30 разів JPEG зображення.

5) До кожної квантованої матриці застосовується так зване зігзаг-упорядкування. Це особливий прохід матриці для отримання послідовності (Рисунок 1.3.1).

Спочатку проходить елемент T_{00} , потім T_{01} , T_{10} , T_{11} ... При чому для типових фотореалістичних зображень спочатку йтимуть ненульові коефіцієнти, відповідні низькочастотним компонентам, а потім - безліч нулів.

6) Ущільнення методом RLE. Отримана послідовність кодується за допомогою модифікованого алгоритму групового кодування. Виводяться пари чисел: перше - число нулів, друге - значення підпослідовності нулів.

7) Ущільнення методом Хаффмена. Проводиться кодуванням методом Хаффмена зі спеціальною фіксованою таблицею.

Алгоритм Хаффмена використовує префіксний код - код змінної довжини, що володіє властивістю префікса: коротші коди не співпадають з початковою частиною довших. Такий код дозволяє здійснювати взаємно-однозначне кодування.

Процес ущільнення полягає в заміні кожного елемента вхідної послідовності його кодом. Декодування здійснюється прямою заміною коду на відповідний елемент.

Алгоритм відновлення зображення інвертує кроки кодування. Ступінь ущільнення при цьому сягатиме від 5 до 100 і більше разів (варіюється за допомогою матриць квантування і заданні методу субдискретизації). При цьому візуальна якість для більшості фотореалістичних зображень залишається на гарному рівні при агресивності ущільнення до 15 разів.

Даний алгоритм і формат є найпоширенішими для передачі і зберігання повнокольорових зображень. Настільки широке поширення пояснюється декількома причинами: відносно невисокою обчислювальною складністю, достатнім ступенем стандартизованого формату і алгоритму а також відсутністю необхідності платити будь-які ліцензійні відрахування. До недоліків алгоритму відносять виникнення артефактів, які занадто помітні оку.

Описана схема є типовою і для інших алгоритмів ущільнення зображень з втратами (за винятком фрактального). Відмінність в основному складається в перетворенні на кроці 3.

1.4 Типізація обчислювальних систем паралельної обробки даних

1.4.1 Аналіз паралельних архітектур

Загальна класифікація ЕОМ по признакам наявності паралелізму в потоках команд та даних була запропонована в 70-ті роки Майклом Фліном. За Фліном є 4 класи обчислюваних систем:

1) SISD (single instruction stream / single data stream) - одиночний потік команд і одиночний потік даних. До цього класу належать класичні послідовні машини фон-нейманівського типу. У таких машинах є тільки один потік команд, всі команди обробляються послідовно один за одним і кожна команда ініціює одну операцію з одним потоком даних.[5]

2) SIMD (single instruction stream / multiple data stream) - одиночний потік команд і множинний потік даних. В архітектурі подібного виду зберігається один

потік команд, що включає векторні команди. Це дозволяє виконувати одну арифметичну операцію відразу над багатьма даними - елементами вектора.

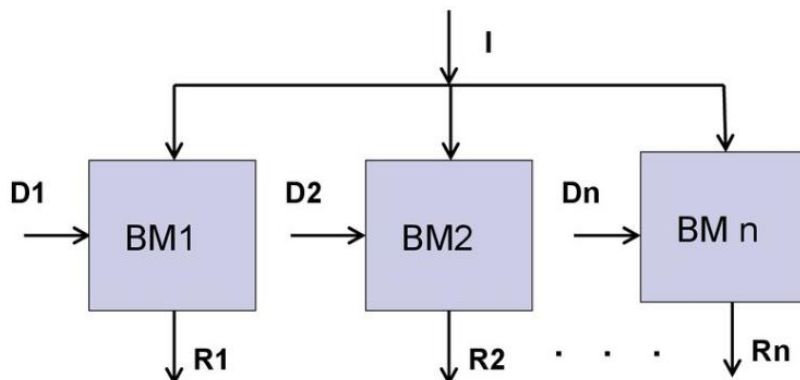


Рисунок 1.4.1 – Процесорна матриця, що реалізує SIMD архітектуру

В такій архітектурі використовується процесорна матриця, що складається з групи однакових процесорних елементів об'єднаних в комутаційну мережу, яка керується єдиним пристроєм керування, виконуючи одну програму(Рисунок 1.5.2).

3) MISD (multiple instruction stream / single data stream) - множинний потік команд і одиночний потік даних. Мають структуру у вигляді ланцюга послідовно з'єднаних процесорів, що утворюють процесорний конвеєр. У таких системах

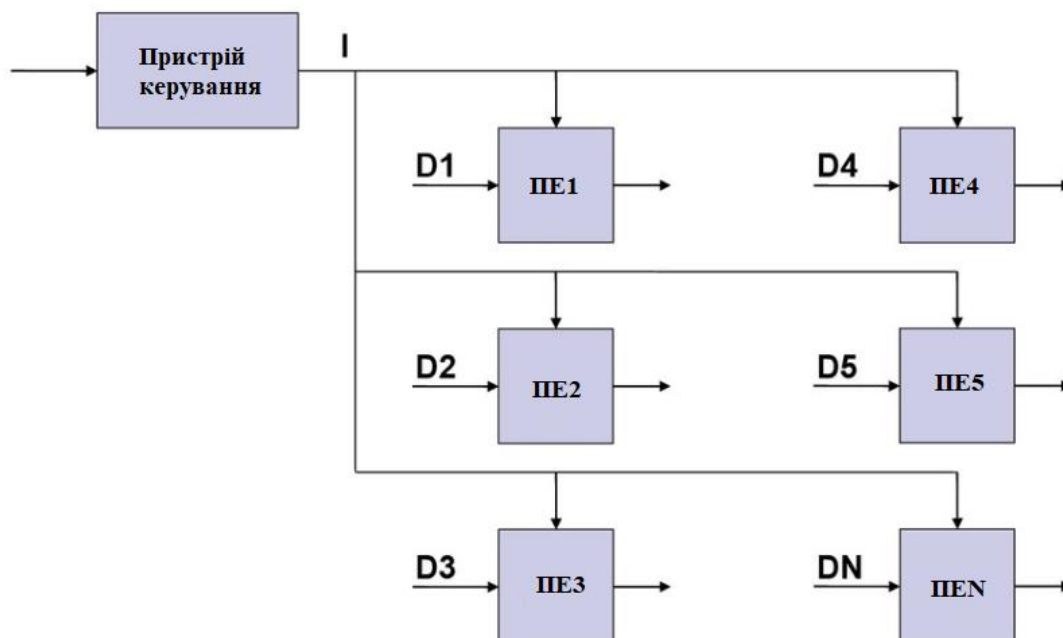


Рисунок 1.4.2 – Структура обчислювальної системи класу SIMD

реалізований принцип магістральної обробки даних. Процес ділиться на етапи, кожен з яких виконується на окремому процесорі. Кожен процесор вирішує свою

частину завдання і результати вирішення передаються на вхід наступного процесора. До кожного процесору підводиться свій набір команд. Реалізацією можна представити розподілену мультипроцесорну систему з загальними даними у якості локальної мережі персональних комп'ютерів що працює з однією БД.

4) MIMD (multiple instruction stream / multiple data stream) - множинний потік команд і множинний потік даних. Цей клас припускає, що в обчислювальній системі є кілька пристроїв обробки команд, об'єднаних в єдиний комплекс і працюючих кожен зі своїм потоком команд і даних.

MIMD в свою чергу розширюються на симетричні обчислювальні системи(SMP), кластерні обчислювальні системи з спеціалізованими кластерами або системи з кластерами загального призначення та масово паралельними системами(MPP).

SMP складаються з множини процесорів, що мають рівний доступ до пам'яті та зовнішніх пристроїв, що функціонують під керуванням однієї ОС.

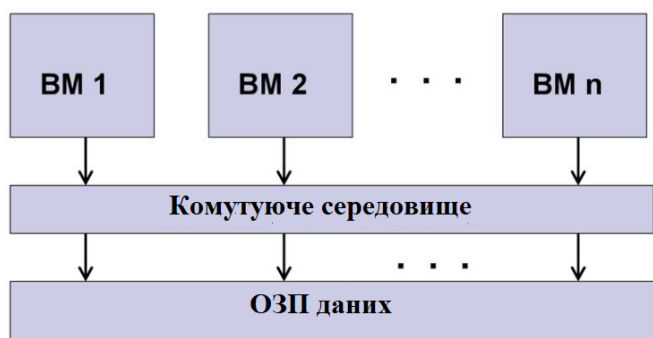


Рисунок 1.4.3 – Структура симетричних мультипроцесорних систем

За способом обробки даних виділяють конвеєрні, матричні, мультипроцесорні, розподілені та обчислювальні системи з програмованою структурою.

Конвеєризація – метод проектування, в результаті виконання якого в обчислювальній системі забезпечується сумісне виконання різних дій над обчисленням базових функцій за рахунок їх розбиття на підфункції.

Умовами конвеєризації є:

- Обчислення базових функцій еквівалентне обчисленню деякої послідовності підфункцій;
- Величини, що є вхідними для даної підфункції є вихідними попередньої в процесі обчислення;
- Між підфункціями відсутні будь-які інші взаємозв'язки окрім обміну даних між ними;
- Кожна підфункція може бути виконана апаратними блоками;
- Час, необхідний для реалізації апаратними блоками своїх дій мають однаковий порядок величини.

В основі конвеєрів лежить послідовно-паралельний спосіб обробки інформації, що представлений векторами даних, а їх функціональні структури представляються у вигляді послідовності зв'язаних елементарних блоків обробки інформації. Всі блоки конвеєра працюють паралельно, але кожен з них реалізує лише свою операцію над даними(компонентами) одного і того ж потоку(вектора).

Конвеєрний підхід можна віднести до MISD систем. Реальні промислові високопродуктивні обчислювані системи наразі є мультиконвеєрними[1]. В них єдиний управляючий пристрій формує один потік даних на підсистеми-конвеєри.

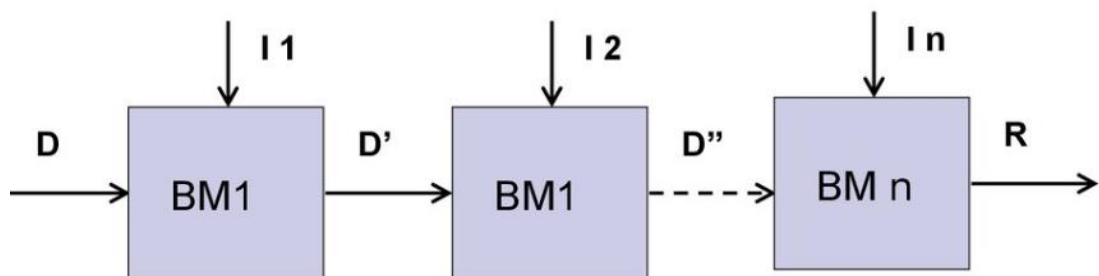


Рисунок 1.4.4 – Конвеєрна паралельна обчислювальна система

Недоліками конвеєрних обчислювальних систем є проблеми керування системою, до яких відносять різний період часу обробки даних на різних етапах, відсутність зв'язку між попередніми стадіями, нагромадженість шляхів, залежностей в порядку виконання.

Матричні обчислювальні системи – системи з масовим паралелізмом(Massively Parallel Processing Systems).

Оснащуються «матричним» процесором, який реалізує можливість одночасної обробки фіксованого числа операцій на однорідних обчислювачах – елементарних процесорах, що об'єднані в «матрицю». Матриця формується за принципом роботи обчислюваних елементів: кожен процесор в двомірній матриці пов'язаний з чотирьома сусідніми.

З огляду на особливості роботи матричного процесору підхід, що реалізується такою системою можна віднести до SIMD класу.

Векторно-конвеєрна архітектура містить конвеєр операцій, на якому обробляються паралельно елементи векторів та отримані результати послідовно записуються в єдину пам'ять. При цьому відпадає необхідність в комутаторі процесорних елементів, який є недоліком матричних комп'ютерів.

Систолічна матриця реалізує однорідне обчислювальне середовище, що представляє собою регулярний масив з процесорних елементів, виконуючих на протязі кожного такту однакові обчислювальні операції з пересиланням результатів обчислень своїм найближчим сусідам.

Досить продуктивною видалась асоціативна обчислювальна система (АОС), яка є представником систем SIMD. Її архітектура представлена на Рисунку 1.4.5. В таких системах звернення до даних проводиться за відмінностями в даних або додаткових розрядах даних.

Ця відмінність відрізняє АОС від матричних систем, в яких доступ до даних проводився за допомогою адрес. Можливість звертатися саме за відмінностями в даних забезпечується спеціальними асоціативними процесорами на базі АЗП – асоціативного запам'ятовуючих пристроїв, що керуються асоціативним процесором, на базі якого будується система[2].

Асоціативні процесори відрізняються від АЗП за двома особливостями: наявністю засобів обробки даних та можливістю паралельного запису у всі комірки, для яких було зафіксовано співпадіння по асоціативній ознаці. Остання відмінність ще дістала назву мультизапис[2].

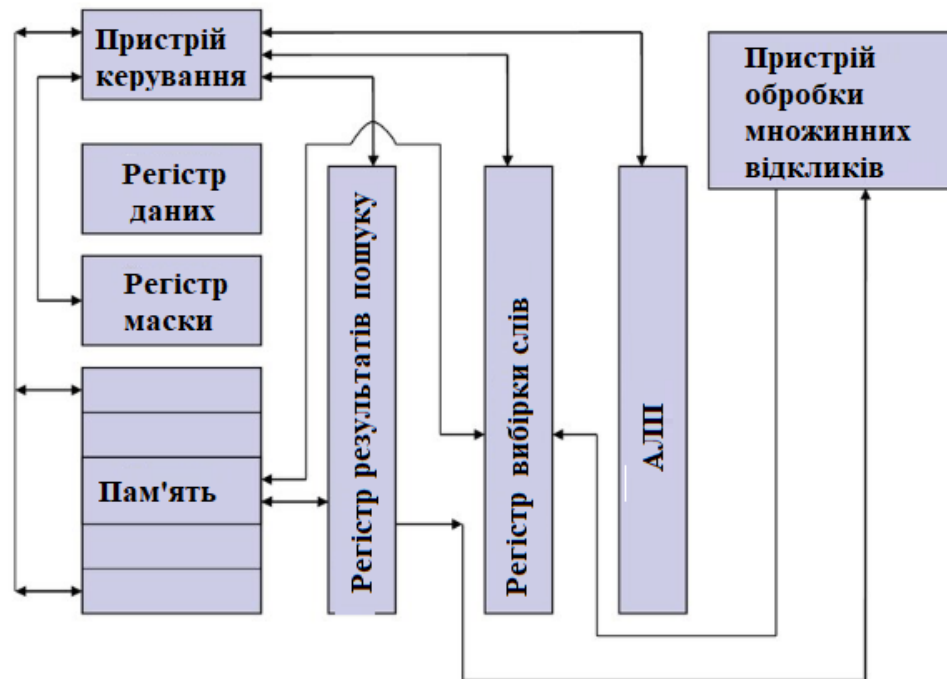


Рисунок 1.4.5 – Архітектура асоціативної обчислювальної системи

Спосіб виконання операцій над машинними словами дозволяє виділити 4 класи асоціативних процесорів:

- Паралельні;
- Порозрядно-послідовні;
- Послівно-послідовні;
- Блочно-орієнтовані.

Сучасні продуктивні обчислювальні системи беруть за основу MIMD. Недоліком класу є нагромаджений опис у систематиці Фліна (Рисунок 1.4.6)

Особливістю SMP (Symmetric Multiprocessing) є наявність загальної фізичної пам'яті для всіх процесорів (Рисунок 1.4.7). Це спрощує взаємодію процесорів між собою, однак накладає обмеження на їх число – не більше 32 в реальних системах, що все ж розширює масштабність. Прикладом таких систем є Hewlett-Packard 9000 V-Class, N-Class, сервери та робочі станції на базі процесорів Intel.

Працюють такі системи під єдиною ОС (UNIX або Windows NT), в яких ОС автоматично розподіляє процеси та вузли між процесорами, іноді можлива явна прив'язка.

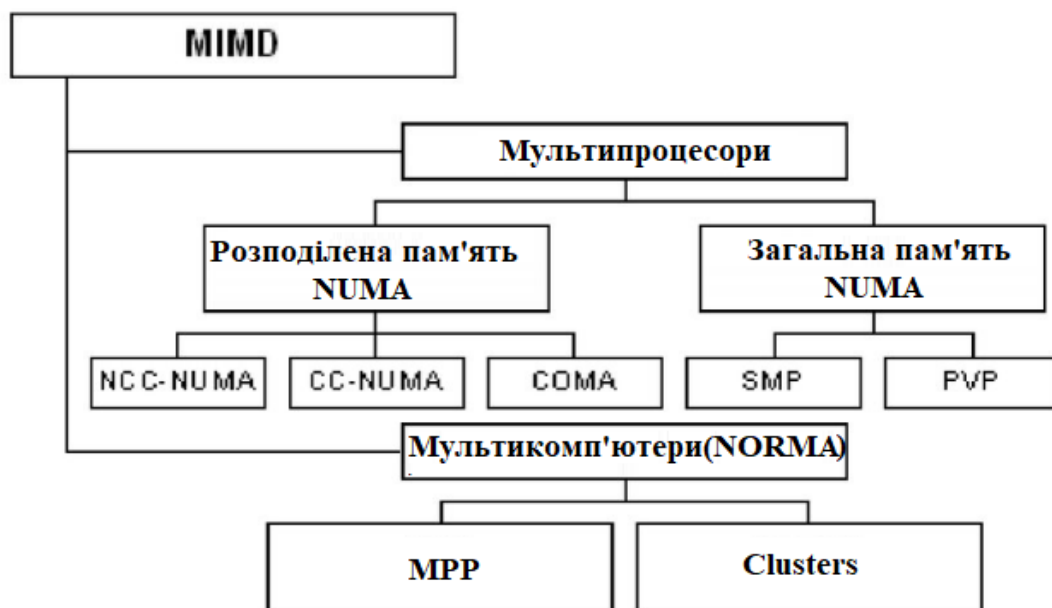


Рисунок 1.4.6 – Класифікація паралельних обч. систем класу MIMD

Модель програмування – обмін даними через загальну пам'ять за допомогою POSIX threads, OpenMP.

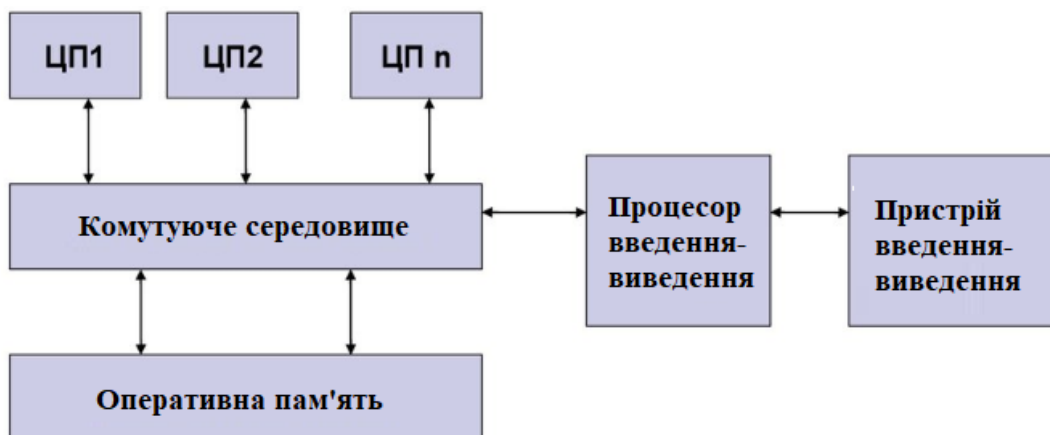


Рисунок 1.4.7 – Архітектура SMP

Мультипроцесори з розподіленою пам'яттю (NUMA) поділяють на СОМА, СС та NCC (Рисунок 1.4.6).

СОМА для представлення даних використовує тільки кеш пам'ять використовуваних процесорів, СС (Cache-Coherent NUMA) забезпечує однозначність локальних кеш-пам'ятей різних процесорів, а NNC (Non-CC NUMA) не підтримує підтримку когерентності кеша на апаратному рівні забезпечуючи загальний доступ до локальної пам'яті різних процесорів.

Елементарною базою паралельних обчислювальних систем є мікропроцесори, основними вимогами до яких є висока продуктивність, розвинуті засоби обміну при низькій розсіювальній потужності.

1.4.2 Проблеми розробки паралельного ПЗ

До проблем розробки паралельного ПЗ відносять проблему розпаралелювання, проблему відладки та верифікації, проблему нарощування та проблему переносності.

Ефективність та обґрунтовність використання паралельного ПЗ можна визначити за законом Амдала, в якому, де S – прискорення програми у порівнянні з послідовним виконанням, p – кількість процесорів, а f – доля послідовного коду у програмі ($0 < f < 1$)

Для створення паралельного прикладного ПЗ виділяють три підходи:

- 1) Написання паралельної програми з нуля;
- 2) Розпаралелювання (автоматичне) існуючої програми з послідовним кодом;
- 3) Комбінований підхід – автоматичне розпаралелювання з подальшою оптимізацією.

Кожен підхід володіє своїми перевагами і недоліками.

Написання програми з нуля дає змогу отримати ефективний код, що в свою чергу веде до таких недоліків як висока трудомісткість, високі вимоги до кваліфікації програміста та висока ймовірність помилок в коді й складність відладки.

Другий підхід своїм недоліком має низьку ефективність розпаралелювання, та в свою чергу володіє такими перевагами як висока надійність коду та використання напрацьованої основи, що зменшує навантаження та скорочує терміни написання коду.

Комбінований підхід відрізняється перевагами і недоліками попередніх двох: його застосування потребує обширного набору інструментальних програмних засобів.

2 УЩІЛЬНЕННЯ ЗОБРАЖЕНЬ НА ОСНОВІ ПАРАЛЕЛЬНОЇ ОБРОБКИ ОПЕРАЦІЙ НАД ЗОБРАЖЕННЯМИ

2.1 Розпаралелення обчислень при фільтрації зображень

Як відомо виділяють частотну і просторову фільтрацію.

Частотні методи ґрунтуються на ідеї перетворення Фур'є, що полягає в поданні вихідної функції у вигляді суми тригонометричних функцій різних частот, помножених на коефіцієнти. Функцію, представлену Фур'є-перетворенням, після здійснення над нею операцій можна повернути до початкового стану. У практичному застосуванні реалізація частотних підходів може бути аналогічна просторовим методам фільтрації [10].

Просторові методи застосуються до зображень, представлених у вигляді двовимірних матриць. Принцип просторових алгоритмів полягає в застосуванні спеціальних операторів до кожної точки вихідного зображення. В якості операторів виступають прямокутні або квадратні матриці (маски, ядра або вікна) [10-12]. Маска являє невеликий двовимірний масив, а методи, що базуються на такому підході часто називають обробкою по масці або фільтрацією по масці.

Фільтрація зображення – це комплексний процес, який виконується за величезну кількість операцій зсуву маски, розрахунків значень всередині маски згортки, серед яких можуть бути прості арифметичні операції, такі як сума та різниця значень яскравості пікселів, так і складні операції знаходження квадратного кореня, квадратів, похідних та т.і.

Відомі підходи до фільтрації зображень існують вже десятки років, і з часу винайдення, коли переважна більшість фільтрів були освоєні в часи, коли переважна більшість обчислювальних машин були однопоточні, оснащені одноядерними центральними процесорами, не виникало потреби і можливостей оптимізувати обчислення за допомогою розпаралелювання виконання обчислень.

Сьогодні ж ситуація змінилася, і сучасні ПК оснащені багатоядерними процесорами, які можуть опрацьовувати декілька потоків даних одночасно, що значно підвищує продуктивність виконання обчислень.

Розглянемо варіанти розпаралелення обчислень за допомогою матричного та конвеєрного підходів на найрозповсюдженіших методах фільтрації цифрових зображень.

Розпочнемо з розпаралелення обчислень найпримітивнішого з методів фільтрації зображень основна ідея якого усереднювати значення пікселів в межах певних зон. Принцип його роботи - для кожного пікселя аналізувати сусідні для нього пікселі, які розташовуються в деякому прямокутному вікні навколо цього пікселя (Рисунок 2.1), і в якості нового значення центрального пікселя брати середнє арифметичне тих його сусідів, значення яких відрізняється від значення центрального не більше ніж на деякий поріг. Чим більше величина цього порога, тим прогресивніше відбувається усереднення.

Припустимо, що ми маємо деяке зображення розміром $m \times n$ пікселів. Оскільки фільтр повинен мати центральний елемент, розмір вікна буде набувати значень 3×3 , 5×5 і т.д.

Оскільки в зображеннях ми маємо справу з однорідними даними, можна розбити його на прямокутні блоки. Розмір блоків залежатиме від кількості потоків, які будуть приймати участь в перетвореннях, тобто $S_T =$ де m – висота зображення, n – ширина, t – кількість потоків.

Після того, як області роботи кожного потоку визначені визначаємо розмір вікна. Чим більший розмір вікна тим агресивніше буде згладжено зображення і тим більше деталей буде втрачено.

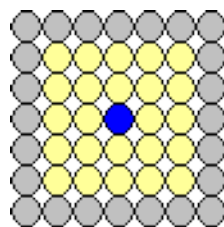


Рисунок 2.1 – Вікно маски фільтрації

Якщо яскравість даного елемента перевищує середню яскравість групи найближчих елементів на деяку порогову величину, яскравість елемента замінюється на середню яскравість:

$$[x - \frac{1}{8} \sum_{i=1}^8 O_i] > \varepsilon, \text{ то } x = \frac{1}{8} \sum_{i=1}^8 O_i \quad (2.1)$$

Низькочастотна просторова фільтрація може служити ефективним засобом згладжування шумів. Масив Q розміру $M \times M$ вихідного зображення формується шляхом дискретної згортки масиву F розміру $N \times N$ вихідного зображення з згладжуючим масивом H розміру $L \times L$ згідно з формулою

$$Q(m_1, m_2) = \sum_{n_1} \sum_{n_2} F(n_1, n_2) H(m_1 - n_1 + 1, m_2 - n_2 + 1) \quad (2.2)$$

Згладжування шуму забезпечується низькочастотної фільтрацією за допомогою масиву H з позитивними елементами. Нижче наведені згладжують масиви трьох різновидів, часто звані шумоподавляючими масками:

$$H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}; \quad H = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}; \quad H = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix};$$

Обчислення значень в масках згладження шуму можна розпаралелити шляхом паралельного виконання арифметичних дій, що може дещо скоротити час необхідний для обчислення операцій множення.

Якщо придушення шуму пов'язане з використанням масивів великого розміру доцільно виконувати згортку непрямым чином перетворенням Фур'є, так як зазвичай це дає вигоду в швидкості виконання обчислень.

Так як при виділенні границь зображення перекриваються двома апертурами сканування, одна з яких відповідає нормальному розширенню, а інша – зниженого, процес перекриття можна розпаралелити по функціям: на вхід поступає фрагмент зображення, потік з функцією перекриття оригіналу читає значення пікселів та опрацьовує своєю функцією перекриття $F(j, k)$, а другий потік – функцією зниженої розмірності $F_L(j, k)$. В результаті отримаємо відповідно масив нормального зображення $F(j, k)$ масив розмитих знімків $F_L(j, k)$.

Потім формуємо масив маскованого зображення:

$$F_M(j, k) = cF(j, k) - (1-c) F_L(j, k), \quad (2.3)$$

де C - коефіцієнт пропорційності.

Медіанний фільтр являє собою ковзаюче вікно, яке охоплює непарне число елементів зображення. Центральний елемент замінюється медіаною всіх елементів зображення у вікні. Медіаною дискретної послідовності a_1, a_2, \dots, a_N для непарного N є той елемент, для якого існують елементів, менших або рівних йому за величиною, більших або рівних йому за величиною.

Можливості аналізу дій медіанного фільтра обмежені. Медіана похідної K і послідовності $f(j)$ дорівнює:

$$\text{Med}\{K f(j)\} = K \text{med}\{f(j)\}. \quad (2.4)$$

Окрім того, $\text{med}\{K + f(j)\} = K + \text{med}\{f(j)\}$.

Двовимірний медіанний фільтр з вікном розміру $L \times L$ забезпечує ефективне придушення шуму. Канонічний двовимірний медіанний фільтр проходить зображення по горизонталі, потім по вертикалі або ж навпаки.

Для оптимізації фільтрації можна піти двома шляхами:

- 1) Паралельно виконувати перекриття зображення фільтром по горизонталі та вертикалі масками згортки розміру $L \times 1$;
- 2) Виконати наступні дії:
 - а. Перекрити зображення масками різних розмірів (3x3, 5x5, 7x7...);
 - б. Зберігти в буфер зображення після кожного фільтру;
 - в. Порівняти значення по масці 3x3 за формулою:

$\text{Pixel}_r =$

де $F_1 \dots F_N$ – значення одного і того ж пікселя в результуючих зображеннях, покритих матрицями згортки різного діапазону, N – кількість зображень, що приймали участь у вибірці

У результаті ми отримаємо усереднене значення, яке ми отримаємо для пікселя результуючого зображення.

Пряме дискретне вейвлет-перетворення (ДВП) - це ітераційне застосування низькочастотного і високочастотного фільтрів з подальшим видаленням кожного другого елементу (проріджуванням) до низькочастотного сигналу, що отримується на виході. В результаті низькочастотної фільтрації виходить наближення вихідного сигналу, в результаті високочастотної - деталізує інформація про

вихідний сигнал, а отримані значення високочастотного сигналу називаються вейвлет-коефіцієнтами. Зворотнє ДВП складається з ітераційного застосування зворотних фільтрів до високочастотних і низькочастотних коефіцієнтами з відновленими елементами (їх значення приймаються за 0) та їх складання. Пряме перетворення називається аналізом, а зворотнє - синтезом. Пара фільтрів, що беруть участь в перетворенні - вейвлетного базисом. Схематично все вищесказане зображено на рисунку 2.3.

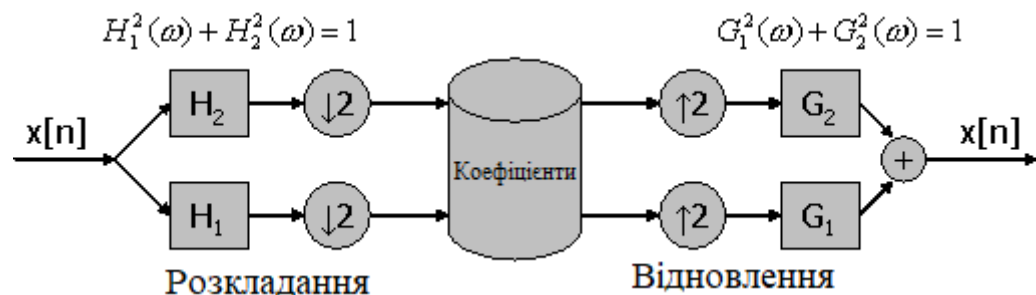


Рисунок 2.3 – Схема одномірного вейвлет-перетворення

Тут $H_1(G_1)$ – високочастотний фільтр аналізу (синтезу), $H_2(G_2)$ – низькочастотний фільтр синтезу.

Двовимірне дискретне вейвлет-перетворення виходить в результаті застосування одновимірного перетворення послідовно до рядків і стовпців зображення. В результаті ми отримуємо одну низькочастотну смугу і три високочастотних, причому ВЧ-коефіцієнти кожної смуги відповідають за конкретний напрямок - вертикальний, горизонтальний, або діагональний (Рис.2.4).

Після такого перетворення дрібні деталі зображення поряд з шумом будуть міститися в високочастотних коефіцієнтах, тому шумозаглушення проводиться шляхом зменшення значень вейвлет-коефіцієнтів невеликої амплітуди. Коефіцієнти зменшуються на величину, що залежить від загального рівня шуму і від імовірності того, що даний коефіцієнт являє собою шум.

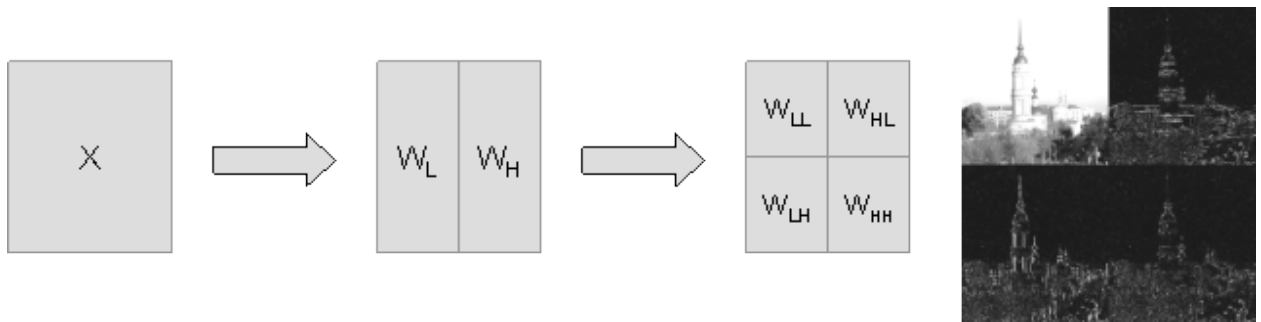


Рисунок 2.4 – Схема двовірного вейвлет-перетворення

Цю ймовірність можна оцінити різними способами. Головне при цьому - як можна менше впливати на коефіцієнти, відповідні деталям зображення, щоб уникнути їх розмиття.

Схему видалення шуму за допомогою ДВП можна описати таким чином:

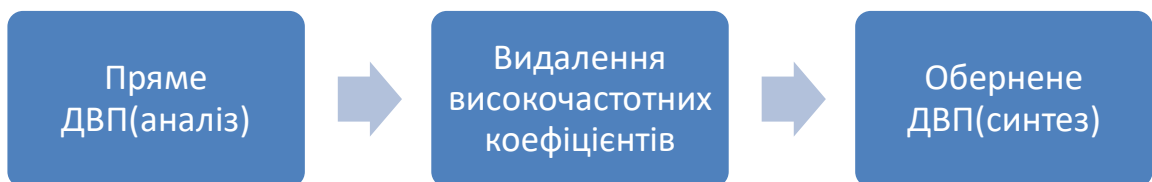


Рисунок 2.5 – Схема заглушення шуму на основі вейвлет-перетворення

Вейвлет-коефіцієнти на першому рівні розкладання відповідають високочастотному шуму і придушуються сильніше, ніж вейвлет-коефіцієнти на наступних рівнях.

2.2 Ущільнення зображень на основі сегментації зображень

Кінцевий результат аналізу зображень багато в чому визначається якістю сегментації [10]. У більшості випадків вибирається один або кілька алгоритмів, і модифікується під специфічні умови задачі.

Сегментація вирішує дві основні задачі [13]:

- поділ зображення на частини для подальшого аналізу;
- зміна форми опису елементів зображення, що дозволяє представити точки як високорівневі структури

Виділяються різні класифікації методів, але більшість з них ґрунтуються на двох властивостях сигналу яскравості - розривності і однорідності.

Поділ зображення на частини базується на ідеях, заснованих на різких перепадах яскравості. Зміна форми опису елементів зображення ґрунтується на поділі зображення на однорідні області з урахуванням заздалегідь обраних критеріїв [10].

До методів на основі розривності яскравості відноситься виявлення точок ліній і перепадів [10]. При виявленні точок і ліній за допомогою спеціальних масок організовується відповідний пошук. В якості методів виявлення перепадів використовуються похідні і градієнти від функцій яскравості, такі методи засновані на більш загальних ідеях [10, 13].

Методи порогової обробки ґрунтуються на ідеях поділу значущих характеристик по деякому порогу. Поріг може бути глобальним, тобто конкретним на всьому зображенні або локальним, який визначений на деякій області [10, 13,1].

Метод нарощування областей ґрунтується на виявленні розривів яскравості. У методах виділяються кілька центральних точок або груп точок, після чого, до них приєднуються пікселі, що задовольняють заданим умовам [10, 13].

Гістограмні методи засновані на виборі мінімальних і максимальних значень або інтервалів між екстремумами.

Методи теорії графів, ґрунтуються на виявленні і зв'язуванні контурів за допомогою уявлення відрізків контурів у вигляді графа і здійсненні пошуку на цьому графі шляхів, відповідних шуканим контурам [10].

Для порівняння методів сегментації цифрових зображень були змодельовані методи, що подаються у вигляді масок. Принцип роботи таких методів заснований на різниці яскравості елементів і фону зображення [10]. Математичне обґрунтування методів полягає в обчисленні похідних, представлених для цифрових зображень у вигляді дискретних наближень градієнта.

Цікавим розвитком комплексного методу обробки зображень, ідеєю якого є видалення шуму є фільтра Гаусса.

Розмивання Гауса – це метод фільтрації за допомогою функції, що розмиває деталі зображень. На зображення особливим чином накладається формула Гауса:

$$g(x, y) = Ae^{-\frac{x^2+y^2}{\sigma^2}} \quad (2.5)$$

де s задає ступінь розмиття, а параметр A забезпечує нормування. Фільтр схожий на лінійний, де піксель змішується з оточуючими за певним законом, заданому функцією Гаусса. Матричний фільтр, що оперує вказаною формулою, називається гауссіаном; чим більше його розмір, тим сильніше розмиття (при фіксованому s). Оскільки даний фільтр сепарабельний, представимо його у вигляді $g(x, y) = f_1(x) \cdot f_2(y)$. Згортку можна проводити послідовно по рядках і по стовпцях, що призводить до значного прискорення роботи методу при великих розмірах фільтра.

Поблизу границь (контурів зображення) такий фільтр застосовувати не можна, аби не змазати деталі зображення. Як наслідок уздовж границь залишається зашумлений контур.

Можна модифікувати цей метод для кращої адаптації до границь: шукати в кожному вікні напрямок розмиття (наявність границі), обчислюючи похідні за напрямками, і застосовуючи в даному вікні спрямований Гауссіан уздовж знайденої границі. В результаті розмиття буде проводитися вздовж границь зображення, і зашумленого контуру не буде.

Нехай $f(x, y)$ - функція яскравості аналізованого зображення; X - кінцева підмножина на якій визначено $f(x, y)$.

$S = \{S_1, S_2, \dots, S_k\}$ - розбиття X на k непустих зв'язаних підмножин $S_i, i=1, 2 \dots k$, LP – предикат, визначений на множині S і приймає дійсні значення тоді і тільки тоді, коли будь-яка пара точок з кожного підмножини S_i відповідає критерію однорідності.

Сегментацією зображення $f(x, y)$ по предикату LP називається розбиття $S^* = \{S_1^*, S_2^*, \dots, S_k^*\}$ що задовольняє вимоги:

- а) ;
- б) ;
- в) ;

г) суміжні області

Умови а) і б) означають, що кожна точка зображення повинна бути певним чином віднесена до деякої області, в) визначає тип однорідності одержуваних областей, г) виражає властивість "максимальності" областей розбиття.

LP називається предикатом однорідності і може бути записаний у вигляді:

Де \sim – відношення еквівалентності, x, y – довільні точки з таким чином, сегментацію можна розглядати як оператор виду

Де f – функції що визначають початкове і сегментоване зображення відповідно i – мітка i -ї області.

Існують два підходи до вирішення завдання сегментації які базуються на альтернативних методологічних концепціях. Перший підхід заснований на ідеї "розривності" властивостей точок зображення при переході від однієї області до іншої. Цей підхід зводить задачу сегментації до задачі виділення границь областей. Другий підхід дозволяє виділити точки зображення, однорідні за своїми локальними властивостями, і об'єднати їх в область, якій пізніше буде присвоєно ім'я або мітка. Перший підхід називають сегментацією шляхом виділення меж областей, а другий - сегментацією шляхом розмітки точок області. Дане вище математичне визначення завдання дозволяє характеризувати ці підходи в термінах предиката однорідності LP . У першому випадку в якості LP повинен виступати предикат, який приймає справжні значення на граничних точках областей і помилкові значення на внутрішніх точках. Однак можна відзначити суттєве обмеження цього підходу, що полягає в тому, що розбиття є тут двоелементною множиною. У практичному плані це означає, що алгоритми виділення границь не дозволяють ідентифікувати різними мітками різні області.

2.3 Градієнтні оператори та їх паралельна інтерпретація

Як градієнтні методи виділяють оператори, що представляють собою матричні маски:

- перехресний оператор Робертса;
- оператор Превіта;
- оператор Собеля.

Немаловажливим при фільтрації зображень є виділення границь об'єктів на ньому. Як було зазначено в розділі 1 можна розпаралелювати задачі як по даним, так і по функціям. Оскільки фільтри криють під собою математичні формули що виконуються процесором за деяку кількість тактів цей процес можна розпаралелити.

Оператор Робертса простий в реалізації і володіє високою швидкодією, однак сильно чутливий до перешкод. На практиці для обчислення дискретних градієнтів найчастіше використовуються оператори Превіта і Собеля. Маски оператора Превіта простіше реалізувати, ніж маски оператора Собеля, однак у останнього оператора вплив шуму кутових елементів маски дещо менше, що істотно при роботі з похідними [14]. Слід зазначити, що у кожної з масок сума коефіцієнтів дорівнює нулю [10].

Оператор Робертса обчислює суму квадратів різниць між діагонально суміжними пікселями, це може бути виконано згорткою з двома вікнами:

використовує чотири значення яскравості на зображенні [12] і має такий вигляд:

$$G_{i,j} = \sqrt{(E_{(i+1),(j+1)} - E_{i,j})^2 + (E_{(i+1),j} - E_{i,(j+1)})^2}, \quad (2.6)$$

або

$$G_{i,j} = |E_{(i+1),(j+1)} - E_{i,j}| + |E_{(i+1),j} - E_{i,(j+1)}|, \quad (2.7)$$

де $[E_{i,j}]$ – елемент матриці вхідного зображення.

З огляду на структуру математичної формули кількість тактів для обробки даних над вікном можна скоротити з шести послідовних операцій до чотирьох.

Оператор Собеля використовує вісім відліків яскравості в області аналізованого елемента:

$$G_{i,j} = \sqrt{G_x^2 + G_y^2} . \quad (2.8)$$

Матриці оператора Собеля мають вид:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \cdot E \text{ та } G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \cdot E$$

де E - матриця вихідного зображення.

У програмному представленні:

$$G_{i,j} = \sqrt{G_{i,j(x)}^2 + G_{i,j(y)}^2} \text{ або } G_{i,j} = |G_{i,j(x)} + G_{i,j(y)}| , \text{ де} \quad (2.9)$$

$$G_{i,j(y)} = [E_{(i-1),(j-1)} + E_{i,(j-1)} + E_{(i+1),(j-1)}] - [E_{(i-1),(j+1)} + E_{i,(j+1)} + E_{(i+1),(j+1)}] .$$

Розпаралелення виконання формули надасть змогу скоротити кількість тактів для виконання дій згідно формули з п'яти тактів до трьох.

Оператор Превіта подібний оператору Собеля і відрізняється від нього маскою. Матриці оператора Превіта мають вид:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \cdot E \quad G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} \cdot E .$$

В якості методів заснованих на похідній другого порядку виділяють оператор Лапласіан [10,13]. Даний оператор виявляє границі в місцях зміни знаку похідної функції яскравості. Оператор Лапласіан дуже чутливий до шуму. Крім того, використання модуля приводить до подвоєння контурів, що дає небажаний ефект і ускладнює сегментацію. З метою зменшення впливу шуму часто використовують лапласіан в поєднанні з згладжуванням, наприклад, за методом Гаусса. Таке поєднання називають оператором лапласіан гауссіана (Laplacian of Gaussian - LoG) [10].

Маска оператора Лапласіан гауссіана створюється за формулою:

$$LoG(x, y) = -\frac{1}{\pi \cdot \sigma^4} \cdot \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) \cdot e^{-\frac{x^2 + y^2}{2\sigma^2}}, \quad (2.10)$$

Ще одним часто реалізованим програмно методом є детектор Кані(Canny). Фактично це набір послідовно застосовуваних алгоритмів. Даний підхід стійкий до шуму і дає, як правило, кращі результати в порівнянні з градієнтними методами. Але, так як це в принципі набір алгоритмів, то і швидкодія даного методу поступається більш простим операторам.

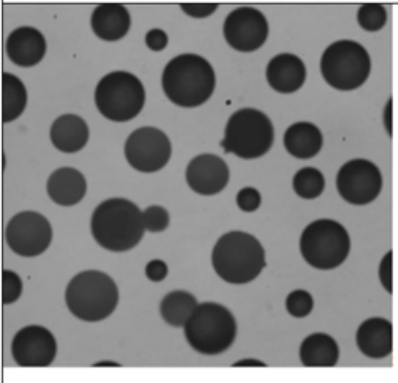
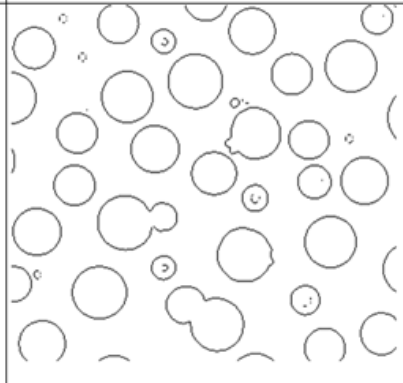
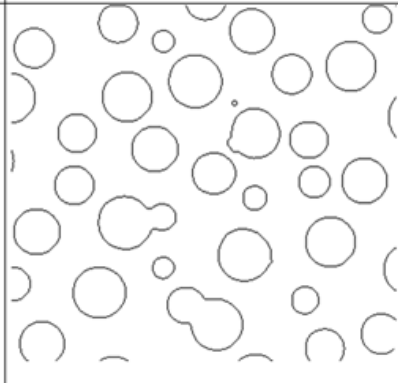
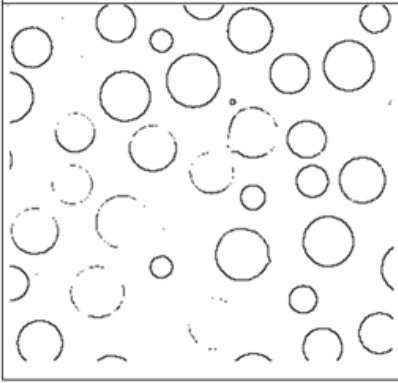
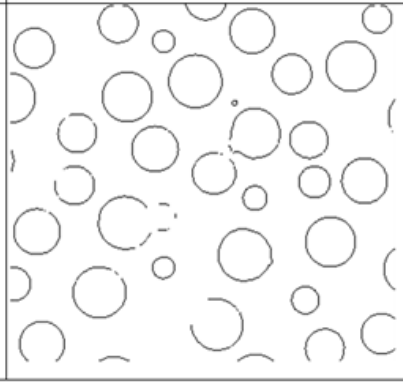
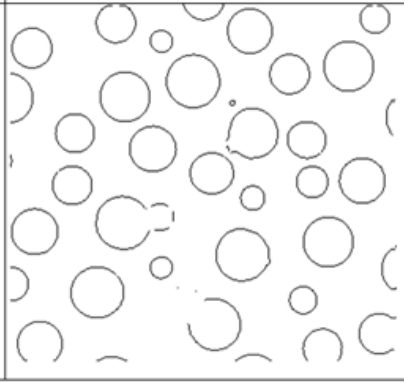
Зображення до сегментації	LoG	Метод Canny
		
Оператор Робертса	Оператор Превігга	Оператор Собела
		

Рисунок 2.6 – Результати сегментації зображень

2.4 Паралельна обробка при ущільненні зображень

Ступінь ущільнення є важливим критерієм ефективності методів ущільнення на ряду з часом на ущільнення та відновлення, складність програмної та апаратної реалізації, стійкості до помилок та звісно ж можливості розпаралелення обчислень.

Перспективним напрямком ущільнення є використання мінімізації булевих функцій. Кожен піксель приймається як вхідний параметр булевої функції та проводиться мінімізація відомими методами. Отримана диз'юнктивна нормальна форма функції представляє собою ущільнене представлення блоку даних.

Переваги ущільнення зображень на основі мінімізації булевих функцій залежать від методу мінімізації. Головною проблемою в мінімізації є великий обсяг обчислень на різних етапах перетворень, що зводить задачу до числа NP-складних задач. Актуальною є задача розробки пономінальних приближених алгоритмів мінімізації булевих функцій.

Паралельна обробка дозволить суттєво підняти якість ущільнення і отримати продуктивну програмно-апаратну реалізацію.

Як відомо, мінімізація складається з 2 етапів: на першому ДДНФ перетворюється в скорочену ДНФ шляхом знаходження всіх мінтермів максимальних рангів, а на другому – в тупикову ДНФ. Для функції F є деяка множина ТДНФ, серед яких знаходиться мінімальна ДНФ.

Перший етап має поліміальну складність і належить до класу P задач. Другий етап виконується перебором за допомогою таблиць покриття, і для великих n число ТДНФ перевищує . Цей етап зводить задачу до класу NP.

Причиною по якій відомі методи мінімізації залишаються в класі NP – ланцюг перетворень СДНФ \rightarrow скорочена ДНФ \rightarrow ТДНФ \rightarrow мінімальна ДНФ.

Для більшості логічних функцій асимптотна складність СДНФ складає n тоді як число мінтермів в скороченій ДНФ перевищує .

Метод мінімізації полягатиме в ітеративному зменшенні складності мінімізованої функції за схемою СДНФ \rightarrow «майже» мінімальна ДНФ, тобто ми обмежимося найближчими по складності до мінімальної ДНФ, що дозволить скоротити час перетворення і заощадить пам'ять.

Розглянемо напівтонове зображення кожен піксель якого представлений у якості k біт. Зображення може мати 2^k градацій яскравості і його можна розбити на k бітових площин, кожен піксель площини матиме одне з двох значень – «1» або

«0». Метод розглядається для бінарних зображень, хоча може використовуватись і для всіх зображень, які можна звести в сукупність бітових площин.

Всі пікселі можна представити як мінтерми однієї білевої функції. Це займає багато часу, і для прискорення процесу складно здійснити паралельну обробку, тому бітова площина ділиться на блоки розміром 4x4, 8x8 або 16x16 пікселів.

Мінімізуються пікселі, де кольори змішані, а для одноколірних визначається лише яскравість.

Серед методів мінімізації краще обрати той, що має зручне графічне представлення кожного етапу перетворень. З огляду на специфічність блоків їх зручно обробляти за допомогою карт Карно з одиниць і нулів чорних та білих пікселів. Для блоків великого розміру блоки нумеруються за допомогою коду Грея.

Перевагою карт Карно є те, що можна виконувати мінімізацію як за одиничними значеннями функції, так і по нулевим значенням, отже можна зробити вибір між ними в залежності від кількості значень «0» та «1».

Оскільки після мінімізації дані зберігатимуться в масиві трійкового алфавіту $\{0,1,x\}$, де $x \in [0,1]$, то для зворотного переходу в двійковий формат необхідне кодування, наприклад, один біт для символу x і по два для символів «1» і «0»:

$$x \rightarrow 0, 0 \rightarrow 10, 1 \rightarrow 11.$$

Таким чином ущільнення складатиметься з таких етапів:

- 1) розбиття зображення на блоки фіксованого розміру;
- 2) представлення кожного блоку картою Карно і нумерація пікселів блоку кодом Грея;
- 3) для змішаних блоків виконати мінімізацію булевої функції по одиничним або нульовим значенням;
- 4) Кодування блоків у форматі ущільнених даних.

Для мінімізації булевих функцій теоретичним базисом є булева алгебра. Для операцій з двозначними елементами визначені операції кон'юнкції, дез'юнкції і віднімання. Під логічною операцією від m змінних буде функція $F\{x_1, x_2, \dots, x_m\}$ з областю лог. значень $\{0,1\}$. ДНФ – диз'юнкція мінтермів $h_1 \vee h_2 \vee \dots \vee h_n$, на яких булева функція F прийматиме значення лог. 1. Мінтерм містить добуток або

різницю змінних, а їх кількість визначає ранг. Кількість мінтермів визначає довжину ДНФ. ДНФ з мінтермів тільки нульового рангу буде досконалою ДНФ.

Для програмно-апаратної реалізації в задачах ущільнення на базі сучасної мікросхемотехніки введемо булеву алгебру кубічних функцій з трьома елементами $\{0,1,x\}$ де $x \in [0,1]$, що формують p -куби. P -куб представляє собою m -мірний набір елементів $e_1 \dots e_m$, де $e_i \in [0,1,x]$, $i=$, а незалежна координата x може приймати значення 1 або 0.

Кубічна функція $\Psi = \Psi(e_1 \dots e_m)$ дає однозначне відображення множини кубів в множині $\{0,1\}$ і використовує три основні операції (перенасичення, об'єднання, доповнення кубів), а також ряд додаткових операцій.

Розрізняють одиничний і нульовий кубічний комплекс $K^{(1)}$ та $K^{(0)}$ на яких функція Ψ приймала значення 1 і 0 відповідно. Кожен з кубічних комплексів однозначно визначає кубічну функцію Ψ , тому одночасно використовуватиметься той з них, що матиме меншу кількість кубів. В ході мінімізації 0-куби об'єднуються в 1-куби, 2-куби і т.д.

В результаті кубічні комплекси перетворюються в кубічні комплекси кубічного покриття. Мінімальне одиничне D -покриття кубічної функції Ψ має мінімальну кількість кубів m_D максимальної розмірності, що вміщують кубічний комплекс $K^{(1)}$ чи $K^{(0)}$.

Якщо провести аналогію з булевою алгеброю логічних функцій, тоді мінімальному D -покриттю кубічної функції Ψ відповідає мінімальна ДНФ логічної функції F , один 0-куб комплексу $K^{(1)}$ відповідає одному мінтерму СДНФ, а s -куб $d_j (j=)$ мінімального D -покриття відповідає імпліканту рангу p мінімальної ДНФ функції F . Наприклад, D -покриття відповідає наступній ДНФ:

$$D = \begin{vmatrix} 0 & 1 & 0 & x \\ 1 & x & 0 & 1 \\ x & 1 & 1 & x \end{vmatrix}$$

Відповідає наступній ДНФ:

$$f(a,b,c,d) = \bar{a} \& b \& \bar{c} \vee a \& \bar{c} \& d \vee b \& c.$$

Під мінімізацією логічної функції F будемо приймати знаходження ДНФ мінімальної складності по критерію кількості v її мінтермів і суми рангів окремих мінтермів:

$$L_{\min}(F) = v + , \quad (2.11)$$

Для оцінки складності використаємо величину вартості покриття. Для покриття, що складається з v кубів, вартість C_D дорівнює:

$$(2.12)$$

Запропонований алгоритм мінімізації складається з ітерацій, в ході яких з покриття $D^{(z)}$ -покриття вартості $C_D^{(z)}$ отриманого на z -й ітерації, формується $D^{(z+1)}$ на $(z+1)$ -й ітерації: $C_D^{(z+1)} < C_D^{(z)}$. Формування нового покриття $D^{(z+1)}$ відбувається в результаті операцій над кубами d_i і d_j . $D^{(z)}$ – покриття на основі наступних правил:

$$(d_i D^{(z)}, d_j D^{(z)}, d_k D^{(z+1)}). \quad (2.13)$$

На основі обґрунтувань можна скласти наступний алгоритм:

Нехай задано блок зображення розміром $(n \times n)$ пікселів у вигляді комірок карти Карно в алфавіті $\{0,1\}$.

- 1) сформуванню в кодї Грея адреси комірок карти Карно розміром $(\log_2 n \times \log_2 n)$, що відповідає заданому блоку зображення;
- 2) розбити комірки карт Карно на дві групи:
 - а. Групу з комірками, які відповідають одиничним пікселям зображення, тобто 0-кубам кубічного комплексу $K^{(1)}$;
 - б. Групу з комірками, які відповідають нульовим пікселям зображення, тобто 0-кубам кубічного комплексу $K^{(0)}$;
- 3) обрати меншу за розміром групу комірок, нехай це буде $K^{(1)}$, і з відповідних 0-кубів сформуванню поточне кубічне покриття D_z . Задати номер ітерації по рядкам $z=1$.
- 4) розбити покриття D_z на t пар покриття $D_z^{(i)}$ та $D_z^{(i+1)}$, що містить куби з t сусідніх пар по $2 \times w$ рядків в парі ($i=1,2,\dots,w, w=2^{z+1}$);
- 5) виконати операції *-добутку між кубами покриття $D_z^{(i)}$ та $D_z^{(i+1)}$ і сформуванню покриття :

$$(2.14)$$

Виконуємо суму сусідніх пар t сусідніх рядків:

$$(2.15)$$

6) якщо $t > 1$ збільшити номер ітерації ($z++$) і повернутися до п.4;

7) сформуувати мінімальне покриття по рядкам $D_{\min, \text{row}}$:

$$D_{\min, \text{row}} = \quad (2.16)$$

8) задати номер ітерації по стовбцям $z=1$;

9) розбити покриття D_z на t пар покриття $D_z^{(i)}$ та $D_z^{(i+1)}$, що включають куби з t сусідніх пар по $2 \times w$ стовбців в парі ($I = 1, 2, \dots, w, w = 2^{z-1}$);

10) виконати операції *-добутку між кубами покриття $D_z^{(i)}$ та $D_z^{(i+1)}$ і сформуувати покриття за правилом додавання та видалення:

$$; \quad (2.17)$$

11) об'єднати отримані покриття всіх t сусідніх пар стовбців:

$$(2.18)$$

12) Якщо $t > 1$, то збільшити номер ітерації ($z++$) і повернутися до п. 4;

13) Сформуувати мінімальне покриття по стовбцям $D_{\min, \text{col}}$ що є шуканим покриттям D_{\min} :

$$D_{\min, \text{col}} =, \quad D_{\min} = D_{\min, \text{col}} \quad (2.19)$$

Відновлення ущільненої інформації відповідає отриманню значення кожного пікселя зображення. Ця задача еквівалентна отриманню комплексу $K^{(1)}$ по відомому D_{\min} покриттю. Для переходу від D_{\min} покриття до комплексу $K^{(1)}$ кожен куб d_i розмірності s_i замінити 0-кубами комплексу $K^{(1)}$. Ми отримаємо деяку кількість однакових 0-кубів, які при відновленні необхідно видалити.

Ефективніше використати для цілей відновлення два m -розрядних лічильника, які вестимуть рахунок в кодї Грея від 0 до 2^m . Перший лічильник(рядків) послідовно видаватиме адресу рядку карти Карно, а інший(стовбців) – адресу. Двійковий $2m$ -розрядний вектор отриманий конкантинацією m -розрядних значень обох лічильників представлятиме собою 0-куб комплексу $K^{(1)}$. На кожному такті роботи в процесі відновлення ущільненої інформації виконується операція перетину кожного куба d_i D_{\min} покриття з вектором v_i .

За 2^n тактів буде відновлено один блок зображення. Виведення результатів можливе одночасно з відновленням.

При відновленні зображення можна отримати дещо спрощений його варіант у випадку якщо лічильники формуватимуть не кожен k -ту адресу 0-куба ($k = 2, 4, 8, \dots$) ми отримаємо таким чином частину пікселів, що відзначиться на якості, але дозволить отримати результат в k разів швидше.

Проведемо аналіз ступеню ущільнення тестових зображень (lena, tulips, baboon) в форматі bmp:

Таблиця 2.1 – Ступінь ущільнення бінарних зображень

Тестове зображення	Розмір блоку 16x16 пікселів			Розмір блоку 8x8 пікселів			Розмір блоку 4x4 пікселя		
	η	N	N_{n0t}	η	N	N_{n0t}	η	N	N_{n0t}
lena	1,727	1024	71	2,156	4096	371	1,908	16384	2427
tulips	1,452	1333	251	1,956	5334	539	1,837	21338	3916
baboon	0,560	156	144	0,801	625	404	1,021	2500	1185

Як видно з таблиці 2.1, для зображень з великими компактними областями одного кольору найбільша ступінь ущільнення з блоками 8x8 та 16x16 пікселів. Для дрібнозернистих зображень необхідно використовувати блоки мінімального розміру, оскільки для великих блоків ефект ущільнення відсутній.

Для більш точного аналізу можна вирахувати відношення кількості N_{n0t} блоків що не ущільнюються до загальної кількості N блоків. Для зображень, що можуть ущільнюватись в два і більше разів доля блоків, що не підлягають ущільненню складає не більш ніж 10%.

2.5 Модифікації фільтрації дискретного-вейвлет перетворення

Методи шумозаглушення з використанням ДВП можна класифікувати за двома параметрами: за типом перетворення і способом придушення коефіцієнтів.

Розглянемо різні типи ДВП і аналогічних їм перетворень:

1) Звичайне ДВП - як правило, використовується з базисом Хаара або Добеші. Не є надлишковим і на кожному наступному кроці загальне число коефіцієнтів зберігається, тобто якщо вихідне зображення було розміру $N \times N$ то сумарне число коефіцієнтів на кожному кроці буде дорівнювати N^2 .

Не є інваріантним до зсуву (якщо зсунути вихідне зображення на 1 піксель, величини вейвлет-коефіцієнтів будуть зовсім інші), і зміна високочастотних вейвлет-коефіцієнтів призводить до артефактів у відновленому зображенні. Тому для шумозаглушення зазвичай використовується надлишкове ДВП, в якому загальне число коефіцієнтів на кожному кроці більше N^2 .

Найпростіший спосіб отримання надмірності - не проріджувати, тобто використовувати наступну схему (Рисунок 2.7):

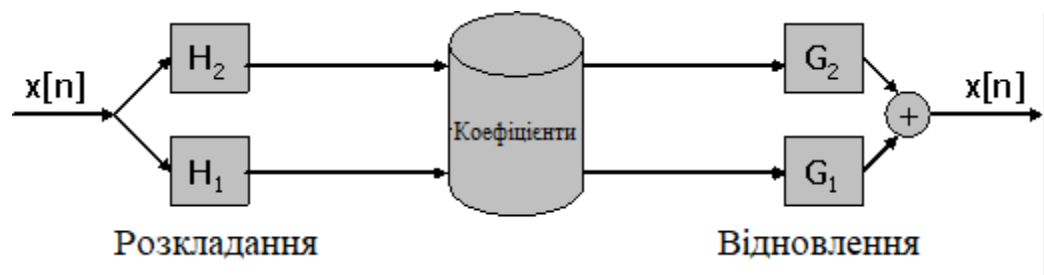


Рисунок 2.7 – Схема одномірного надлишкового вейвлет-перетворення

Однак така надмірність вимагає багато додаткової пам'яті для зберігання коефіцієнтів ($+3N^2$ на кожному рівні розкладання). Далі будуть розглянуті перетворення з меншою надмірністю, які допомагають запобігти появі артефактів, і вимагають при цьому на порядок менше ресурсів.

При використанні надлишкового ДВП слід застосовувати фільтри з кроком, що збільшується на кожному наступному рівні розкладання в два рази. Тобто якщо на першому рівні згортка проводилася з фільтром (0.5, 1, 0.5), то на другому вона повинна проводитися з фільтром (0.5, 0, 1, 0, 0.5) і т. д. Якщо проріджування не робити, шум буде придушуватися тільки в високочастотній області.

2)Комплексне ДВП - представляє собою паралельне застосування до вихідного сигналу двох ДВП (з проріджуванням) з різними НЧ/ВЧ-фільтрами, підібраними спеціальним чином. Потім кожна смуга, отримана в результаті першого ДВП, інтерпретується як дійсна частина комплексного НЧ (ВЧ) сигналу, а кожна смуга другого ДВП - як уявна.

У двовимірному випадку вводиться поняття дійсного ДВП, що укладена в паралельному застосуванні двох звичайних двовимірних ДВП з проріджуванням і обчисленні для кожної пари смуг їх суми та різниці (всього виходить шість вейвлетов, спрямованих в шести різних напрямках). Двовимірне комплексне ДВП складається з двох паралельно застосованих дійсних ДВП, одне з яких аналогічно одновимірному випадку інтерпретується як уявна частина комплексного вейвлета, а інше - як дійсна. У комплексних вейвлет придушуються не самі коефіцієнти, а їх модулі (в інтерпретації коефіцієнтів як комплексних чисел).

Перевагами такого комплексного ДВП для одновимірному випадку є паралельне застосування до вихідного сигналу двох ДВП (з проріджуванням) з різними НЧ / ВЧ-фільтрами. Потім кожна смуга, отримана в результаті першого ДВП, інтерпретується як дійсна частина комплексного НЧ (ВЧ) сигналу, а кожна смуга другого ДВП - як уявна. Надмірність стала набагато менше - тепер вона становить $3N^2$ при будь-якій кількості рівнів. По-друге, такі вейвлети мають кращу спрямованість, і на відміну від звичайних вейвлетов розрізняють напрямки в 45° і 135° , що сприяє якіснішій обробці діагональних границь деталей на зображенні.

Звичайне ДВП є сепарабельном. Воно може обчислюватися окремо по рядках і стовпцях, причому і для рядків, і для стовпців застосовуються однакові фільтри. Несепарабельне ДВП може краще адаптуватися до локальних властивостей зображення. Наприклад, аналізуючи для кожного пікселя його сусідів в деякому вікні можна мінімізувати корисну енергію зображення в ВЧ-смугі і краще зберегти деталі. Алгоритм придушення шуму з використанням несепарабельного ДВП описаний в [8].

Для придушення шуму часто використовують направлені вейвлети (directional wavelets), які забезпечують кращу направленість. Приклад алгоритму придушення шуму з використанням направлених вейвлетів можна знайти в [9].

Для виділення ВЧ-складової можна замість ДВП застосовувати прості перетворення. Наприклад, провести тільки НЧ-фільтрацію, а ВЧ-складову отримати як різницю вихідного і низькочастотного сигналу. Тоді для відновлення треба буде зробити додавання двох сигналів. Таким чином, чотириразове застосування різних фільтрів може бути зведене до однократного. Таке перетворення можна застосовувати, як і звичайне ДВП, послідовно до рядків і стовпців для розкладання сигналу на чотири смуги або використовувати один двовимірний НЧ-фільтр для розкладання сигналу на дві смуги. Обчислюється воно набагато простіше і швидше, ніж ДВП, а якість придушення шуму при такій заміні погіршується незначно.

За способом придушення коефіцієнтів всі алгоритми можна розділити на два класи. Алгоритми, що належать першого класу, при придушенні коефіцієнтів ґрунтуються тільки на значенні порога, порахованого окремо для кожної смуги кожного рівня розкладання. Серед них виділяються два базових методи: різке зменшення по порозу полягає в обнуленні всіх вейвлет-коефіцієнтів, менших заданого порогу. Дуже добре працює при низькому рівні шуму, так як весь шум потрапляє в область нижче порога і забирається повністю, а всі деталі потрапляють в область вище порога і залишаються незмінними. При високому рівні шуму з'являються артефакти - окремі не прибрані сплески шуму. Зі збільшенням порогу вони пропадають, але при цьому під значення порогу починають потрапляти деталі, і зображення виходить нерізде.

Плавне зменшення по порозу полягає в обнуленні всіх вейвлет-коефіцієнтів, менших заданого порогу, і зменшенні інших вейвлет-коефіцієнтів. Перетворення еквівалентно формулі (ϵ використовується щоб уникнути діленню на 0):

$$w_{new} = w * gain, \quad gain = \left(1 - \frac{T}{|w| + \epsilon}\right) \quad (2.20)$$

де T – деякий поріг.

При завданні одного і того ж значення порога для всіх коефіцієнтів всередині смуги цей метод або сильно розмиває зображення, або залишає багато шуму, однак саме він є основою для інших методів.

Алгоритми з другого класу забезпечують локальну просторову адаптацію алгоритму придушення коефіцієнтів всередині кожної смуги. Ось деякі з них:

1) Адаптація по сусіднім коефіцієнтам - підхід заснований на тому, що ступінь придушення кожного вейвлет-коефіцієнта залежить не тільки від величини цього коефіцієнта, а й від зваженої суми модулів (при аналізі вейвлет-коефіцієнтів має значення саме їх амплітуда, а не знак) сусідніх коефіцієнтів, розташованих в деякій просторовій області навколо цього коефіцієнта. Можна, з огляду на взаємозв'язок коефіцієнтів між рівнями, включити в цю суму коефіцієнти з наступного рівня розкладання. Ваги можна задавати різними способами. Як варіант, можна задати більші ваги уздовж основного напрямку границь в смузі: або взяти одне з трьох основних напрямів (вертикальний, горизонтальний або діагональний) в залежності від смуги [8], або для більшої точності самостійно оцінити домінуючий напрямок і шукати ваги, наприклад, за формулою:

$$w = \operatorname{argmin}_w \sum_{i,j} (|Y_{i,j}| - w^T u_{i,j})^2 \quad (2.21)$$

де $Y_{i,j}$ - окремий вейвлет-коефіцієнт даної смуги, w - вектор, що містить шукані ваги, а $u_{i,j}$ - вектор, що складається з сусідніх по відношенню до $Y_{i,j}$ вейвлет-коефіцієнтів [15]. Тобто шукаються такі ваги, щоб зважена сума сусідів максимально наближала кожен коефіцієнт в смузі.

2) Метод, заснований на контексті - шумозаглушення здійснюється за формулою плавного зменшення по порогу, де поріг задається наступним чином:

$$T = \frac{\sigma^2}{\sigma_w} \quad (2.22)$$

σ_w - дисперсія зашумленого сигналу, яка обчислюється таким чином, щоб забезпечити локальну адаптивність величини порога. Для кожного коефіцієнта з контекстом Z визначається підмножина коефіцієнтів, що мають найближчі до Z значення контекстів (як менші Z , так і більші). Після цього значення σ_w

обчислюється для кожного вейвлет-коефіцієнта окремо як дисперсія для цього коефіцієнта підмножини. Потужність підмножини важливо вибрати оптимально, оскільки недостатня кількість коефіцієнтів не забезпечить об'єктивної оцінки. Таким чином виділяються схожі області на зображенні, які просторово можуть розташовуватися далеко один від одного, і по ним рахується дисперсія.

3) Байєсівський підхід - заснований на аналізі статистичних властивостей зображення. Він ґрунтується на припущенні, що розподіл всіх вейвлет-коефіцієнтів зашумленого зображення являє собою комбінацію двох нормальних розподілів з різними відхиленнями. Типовий вид гістограм розподілу вейвлет-коефіцієнтів для чистих і зашумлених зображень представлений на рис. 2.8 (а, б).

Було встановлено, що в загальному випадку щільність розподілу вейвлет-коефіцієнтів зображення досить точно описується наступною функцією званою узагальненим Лапласіаном:

$$p(y) = \frac{\nu}{2s\Gamma(\frac{1}{\nu})} e^{-\frac{|y|^\nu}{s}} \quad (\nu, s > 0) \quad (2.23)$$

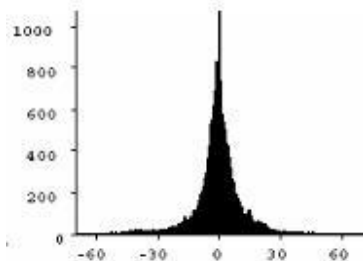


Рисунок 2.8а. - Гістограма розподілу вейвлет-коефіцієнтів для зображення без шуму

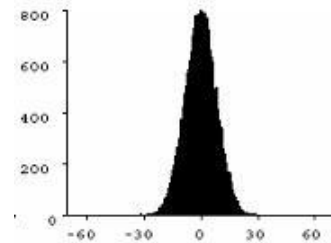


Рисунок 2.8б - Гістограма розподілу вейвлет-коефіцієнтів для зображення з шумами

Де $\Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1} e^{-t} dt$ - гама функція.

Дисперсія такого розподілення визначається за формулою:

$$\sigma_w^2 = \frac{s^2 \Gamma(3/\nu)}{\Gamma(1/\nu)} \quad (2.24)$$

Шум описується функцією нормального гауссовського розподілу:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{z^2}{2\sigma^2}} \quad (2.25)$$

Тоді для зашумлених коефіцієнтів w маємо:

$$\sigma_w^2 = \sigma^2 + \frac{s^2 \Gamma(3/\nu)}{\Gamma(1/\nu)} \quad (2.26)$$

Оскільки можна обчислити, виразимо параметр s з формули:

$$s = \sqrt{0.5(\sigma_w^2 - \sigma^2)} \quad (2.27)$$

Таким чином, модель незашумленого вихідного сигналу повністю відновлена.

Знаючи розподіл вихідного сигналу, можна оцінити співвідношення між шумом і деталями для даної смуги вейвлет-перетворення. Припустимо, що вейвлет-коефіцієнти, менші по модулю деякого порогу T , відповідають шуму, а коефіцієнти, що більші по модулю за T відповідають деталям зображення. Тоді відношення сигналу до шуму на всій смузі дорівнюватиме:

$$\rho = \frac{\int_{-\infty}^{-T} p(y) dy + \int_T^{\infty} p(y) dy}{\int_{-T}^T p(y) dy} = \frac{1 - \Gamma_{\text{inc}}((T/s)^\nu, 1/\nu)}{\Gamma_{\text{inc}}((T/s)^\nu, 1/\nu)} = \frac{e^{-T/s}}{1 - e^{-T/s}} \quad (2.28)$$

Тут $\Gamma_{\text{inc}}(x, a) = \frac{1}{\Gamma(a)} \int_0^x t^{a-1} e^{-t} dt$ - неповна гама-функція.

Метод головних компонент (МГК) дозволяє виділити структуру в багатовимірному масиві даних і застосовується для розпізнавання або ущільнення зображень.

Нехай задано N -мірний простір і M векторів в ньому. Під набором даних будемо приймати вектор, складений з i -ї координати всіх векторів. Отже, нехай у нас є N наборів даних відповідних N вимірам. Спочатку будемо матрицю коваріацій для цих наборів і шукаємо її власні вектора і значення. Ортонормований базис з N власних векторів і буде шуканим базисом, що максимально наближає вихідні дані, тобто відображає їх структуру в N -вимірному просторі. Причому, чим

більше власне значення - тим сильніше відображає характер розташування даних відповідний власний вектор.

Власний вектор, відповідний максимальному власному значенню, визначає основний напрямок, уздовж якого розташовані дані. З іншого боку, власні вектора фактично не відображають характер розташування даних. Виключивши з нового базису власні вектора з найменшими значеннями, можна зменшити розмірність вихідних даних без особливих втрат. Таким чином, МГК забезпечує найкраще наближення вихідних даних мінімальним числом базисних векторів. На зменшенні розмірності ґрунтуються алгоритми ущільнення.

Спочатку зображення розбивається на блоки. Блоки обробляються незалежно, тому вони повинні розташовуватися з невеликим перекриттям, щоб уникнути артефактів блочності при їх стикуванні. Пікселі блоку $N \times N$ утворюють вектор довжини N^2 .

Навколо кожного блоку розглядається деяке вікно, з якого вибираються блоки того ж розміру, що і центральний. Вектора, відповідні цим блокам з центральним, утворюють вихідний набір даних, для якого застосовується алгоритм МГК. Будується матриця коваріацій, шукаються власні вектора і власні значення, потім проводиться сортування векторів по спадаючій власних значень. Шумозаглушення здійснюється шляхом зменшення довжин власних векторів і/або їх обнулення в залежності від величини власних значень.[16]

Відновлення вихідного зображення з урахуванням змін проводиться в два етапи: спочатку вихідні дані приводяться до нового базису з власних векторів (якщо частина власних векторів була обнульована, то в результаті вийдуть дані меншої розмірності), потім назад до попереднього базису. При цьому якщо власні вектора не змінювалися, відновлені дані будуть точно збігатися з вихідними.

Оскільки шум розподілений по зображенню хаотично, без будь-якої системи, йому будуть відповідати власні вектора з маленькими власними значеннями. Нехтуючи ними будемо зменшувати амплітуду значень пікселів в цьому напрямку, а отже, придушувати і шум. Тому метод головних компонент добре справляється

лише з білим шумом: у шуму не повинно бути ніякої структури, інакше МГК прийме його за деталь зображення і не придушить.

Найкраще метод головних компонент працює на текстурованих областях, навіть якщо текстура дуже дрібна. Після придушення шуму методом головних компонент текстура залишиться як і раніше чіткою, чого не можна сказати про всі перераховані вище методи.

3) Анізотропна дифузія - основна ідея підходу - яскравість кожного пікселя інтерпретується як значення температури в даній точці зображення, таким чином, всі зображення представляється у вигляді карти температур. Шумозаглушення проводиться шляхом вирівнювання температур (фактично, інтенсивностей пікселів) за допомогою моделювання процесу теплопереносу:

$$I_t = c \cdot \Delta I$$

де c – коефіцієнт теплопровідності, а ΔI - різниця яскравостей(температур) двох пікселів.

Коефіцієнт теплопровідності c залежить від градієнта інтенсивності зображення:

$$c(\nabla I) = \exp\left(-\left(\frac{|\nabla I|}{k}\right)^2\right) \quad (2.29)$$

Так як на границях деталей зображення, як правило, відбувається значна зміна яскравості, коефіцієнт теплопровідності, який визначається за вказаною вище формулою, буде невеликим і, як наслідок, перенесення <тепла> через границю буде мінімальним. Іншими словами, розмиття границь не відбудеться. На рівних областях, навпаки, зміна яскравості пікселів незначна, тому такі області будуть добре згладжені. За своєю суттю даний метод дуже нагадує гауссівське розмиття.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ УЩІЛЬНЕННЯ ТА ФІЛЬТРАЦІЇ РАСТРОВИХ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ МЕТОДІВ ПАРАЛЕЛЬНОЇ ОБРОБКИ ДАНИХ

3.1 Засоби паралельного ущільнення на базі графічного процесора з використанням технології NVIDIA CUDA

Різні методи ущільнення зображення при паралельному використанні графічного процесора з втрат і без втратами удосконалюються в роботі.

1) DCT (дискретне косинусне перетворення) з використанням Леффлера на основі Корди [1] - чотиристадійний процес. Залежність даних є у кожному процесі, тому кожен етап потрібно виконувати послідовно, та частину розрахунків можна виконати паралельно. Алгоритм можна розділити на дві частини: одна частина - коефіцієнт передбачений, а друга - непарні коефіцієнти. Рівна частина - це 4-точковий DCT. Ця частина далі розділена на наступному кроці. Для різної щільності зображення це дає різні результати параметрів продуктивності.

DCT є перетворенням на основі вікна, тому його спочатку застосовують до стовпця матриці 8×8 , а потім DCT застосовують до рядків результатів. Щоб уникнути обчислення значень косинусів під час виконання, косинуси різних фаз попередньо розраховуються і зберігаються в постійну матрицю 8×8 C. Для виконання DCT на стовпчику макроблоку M постійна матриця 8×8 перетинається множиною з макроблоком 8×8 як показано в рівнянні:

$$C \times M = DCT_R$$

Коли виконується другий прохід DCT до рядка стовпця результату DCT DCT_R, як показано в нерівності:

$$DCT_R \times C^T = DCT_2D$$

Існує два типи підходу до реалізації DCT. Спочатку по 64 потокам для кожного блоку в матриці. Кожен потік завантажуватиме один піксель. На основі потоку X та Y індекс зчитування присвоюється кожному потоку. Ці дані зберігаються у загальній пам'яті. Кожен потік обчислює один елемент вихідної матриці.

Результати записуються у загальну пам'ять. Другий підхід використовує 512 потоків для кожного блоку. Блок потоків призначений для обробки 8 макроблоків. Квантування - це простий набір обчислень. Зображення ділиться на кроки, що рівні найближчому дільнику на 16. Тоді матриця квантування множиться на ширину зображення. Матриця квантування нижче використовується як множник / дільник.

Тут відбуваються три дії. Спочатку потоки множуться та діляться для квантування зображення, але зображення все ще є частиною глобальної пам'яті. Друге зображення розділено на блоки 8×8 , і ця матрична сітка квантована за допомогою квантованої матриці, після чого записана назад у глобальну пам'ять. На третьому етапі використовується загальна пам'ять завантажується поділом зображення за квантованими значеннями, а потім копія використовується для множення квантованої матриці в зворотному порядку. Для всіх трьох результатів час виконання лінійно з розміром зображення не збільшується. Накладні витрати на великі зображення можуть бути спричинені конфліктом значень з плаваючою точкою, оскільки обчислення DCT працюють з даними цього типу значень. Навряд чи нелінійність обумовлена обмеженням пропускної здатності глобальної пам'яті, оскільки розмір зображень порівняно невеликий порівняно з доступною глобальною пропускною здатністю пам'яті.

2) Паралельна інтерпретація JPEG2000 з DWT та EBCOT 1 рівня

DWT - це піддіапазонне перетворення, яке перетворює зображення з просторової області в частотну. DWT може нехтувати просторовим співвідношенням між пікселями на зображенні компонент перед виконанням перетворення, тому його паралельна реалізація досить проста. Коефіцієнти DWT розподіляються на окремі кодові блоки та кодуються theEBCOT незалежно.

Є два рівні в EBCOT: Рівень-1 призначений для КБП (кодування бітової площини) та адаптації до контексту АК (Арифметичне кодування), рівень 2 - для оптимізації спотворення швидкості та спотворення бітового потокового шару. Для зняття залежності між зразками використовується паралельний КБП.



Рисунок 3.1 – Алгоритм кодування потоку

Кодер бітової площини обробляє код блоку бітової площини за бітовою площиною, від найбільш значущого біта (MSB) до найменш значущого біта (LSB). У кожній бітній площині бітові лінії групуються у смуги з чотирьох ліній та сканує смуги в стовпцях, зліва направо, використовуючи кодування проходів, що перекриваються екраном.

Масиви після завершення кодування називаються пост-масивами. Це дає змогу усунути залежність між зразком за допомогою комбінування масивів до та після. Необхідна інформація про стан для 3x3 вибірки, але змінні стану кожного зразка не визначені, оскільки вони оновлюються на льоту. Цю ситуацію можна розглядати як предикат перед оновленням умов проходів. Тому потрібен метод прогнозування, щоб попередньо визначити інформацію про стан до початку кодування, щоб він міг обробляти всі вибірки одночасно. Отже, паралельна обробка виконується на цьому етапі для економії часу при розпаралелюванні процесу прогнозування. Блоки коду обробляються незалежно, і кожен зразок блоку коду повинен оброблятися паралельно з використанням методу прогнозування. У GPGPU будь-яка інструкція з керування потоком може суттєво вплинути на введення інструкції, викликаючи провали між потоками одного й того ж паралельного блоку потоків. Правила прийняття рішення про контекст визначені заздалегідь, тому таблиці пошуку для формування контексту можуть бути побудовані завчасно, щоб уникнути контрольного розгалуження. Слід зауважити, що є шанс появи критичних для оптимізації використання пам'яті конфліктів пам'яті. Проблему конфлікту доступу до пам'яті, як і затримки пам'яті може бути знижена за допомогою встановлення швидкої, спільної пам'яті, яка постійно знаходиться на чіпі. Після набору дані

ефективно розподіляються у вибрані регіони пам'яті, використання пам'яті, особливо спільної пам'яті, слід мінімізувати до збільшення багатопроцесорної обробки блоками коду GPGPU. Великий кодовий блок значно зменшує навантаження багатопроцесорних процесів, що приводить до значного прискорення затримок невеликих розмірів, що не доцільно для ефективності ущільнення. За допомогою зовнішніх режимів роботи паралельний кодер 1 рівня (Рисунок 3.1) може обробляти кодовий блок 64 x 64, використовуючи невеликий буфер спільної пам'яті.

3) Vzip2 – це комбінація трьох різних методів трансформації Берроуса-Уілера (BWT), move-to-front (MTF) та кодування Хаффмана.

Vzip2 використовується для ущільнення без втрат і слугує в якості прикладу неефективного методу ущільнення для паралельної обробки.

BWT сам по собі не ущільнює. Натомість він спрощує ущільнення шляхом оборотного переупорядкування рядків. Новий рядок, створений BWT, як правило, має багато повторень символів, що поліпшить ущільнення.

У GPU застосовується BWT алгоритм сортування рядків на основі злиття. Алгоритм використовує ієрархічний підхід для максимізації паралелізму під час кожного етапу. Кожен потік сортує 8 елементів з бітонічним сортуванням. Далі алгоритм об'єднує впорядковані послідовності в кожному блоці. На останньому етапі здійснюється глобальне міжблокове злиття.

Алгоритм Хаффмана замінює кожен символ бітовим кодом змінної довжини. Довжина коду обчислюється за частотою виникнення символу в коді[18].

Дані кодування в вузлах Хаффмана додаються та з'єднуються, гістограма змінюється через те, що не можна створювати вузол паралельно. Встановимо пошук двох найменш частих записів гістограми паралельно, використовуючи схему скорочення. vzip2 має середню швидкість ущільнення 7.72 Мб/с, тоді як реалізація GPU ущільнює в 2,78 повільніше – 2.77 Мб/с.

Таблиця 3.1 – Ефективність алгоритмів ущільнення зображень адаптованих для розпаралелення за допомогою технології CUDA

Назва методу паралельної обробки	Продуктивність
Дискретне косинусне перетворення	~20 разів швидше
Вейвлет-перетворення	Від 10 до 20 разів швидше
Вейвлет перетворення та EBCOT 1 рівня	~16 разів швидше

Можна підсумувати, що за допомогою паралельної платформи CUDA можна скоротити час ущільнення. CUDA управляє паралельним кодом над графічним процесором, який містить велику кількість обчислювальних ядер, що використовуються для роботи з обчисленням графіки.

DWT вимагає перетворення кольорів для усунення залежностей, які роблять паралельне виконання більш доступним. EBCOT обробляє дані у двох рівнях. Паралельна реалізація використовує спільну пам'ять для обробки даних. Доступ до спільної пам'яті підвищить ефективність роботи алгоритму. DWT з паралельним виконанням першого рівня на GPU дає 16-кратний приріст швидкості роботи порівняно з реалізацією програмного забезпечення Jasper. 2D вейвлет перетворення, інтерпретоване паралельно використовуючи пристрої з підтримкою CUDA, прискорює в 21,7 час необхідний для ущільнення в порівнянні з послідовною реалізацією, а варіант на основі OpenMP дає 20,8-кратний приріст швидкості[19]. bzip2 – метод ущільнення без втрат, який при паралельній інтерпретації виконується навіть повільніше за послідовний варіант, і тому його та подібні йому методи кодування розпаралелювати на графічних процесорах неефективно. Алгоритм уповільнює використаний на третьому кроці метод Хаффмена, тому навіть bzip2 при інтерпретації на комбінуванні обчислень на центральному процесорі та графічному процесорі може в теорії дати пришвидшення роботи.

Відтворимо процес ущільнення зображення.

- 1) Прочитати зображення.
- 2) Розбити зображення на блоки 8x8 пікселів.

- 3) Визначення кольору та яскравості пікселів в блоках.
- 4) Паралельна фільтрація блоків за допомогою ДКТ.
- 5) Паралельне квантування блоків пікселів.
- 6) Паралельне ентропійне кодування.

На першому кроці необхідно отримати зображення і прочитати його.

Дані пікселів зображення розбиваємо на матриці 8×8 . Так як оперувати даними пікселів кожного з блоків матриць можна незалежно, етап розбиття можна виконати паралельно використовуючи принцип розпаралелювання SIMD.

На другому кроці для методу ущільнення з втратами доцільно буде виконати наступні дії:

- 1) якщо зображення не ділиться на блоки 8×8 націло, доповнюємо недостаючі блоки по краям зображення та заповнюємо значеннями 0. Цю операцію в англійських джерелах називають Zero-Padding;
- 2) визначаємо компоненти кольору пікселів моделі RGB в блоках матриць 8×8 . Для кольорових зображень дані зчитуються і записуються в буферні матриці;
- 3) паралельне дискретне косинусне перетворення відбувається за наступні етапи:
 - а. Згортка матриць значеннями з функції ДКТ;
 - б. Складність алгоритму складає $O(n^2 \times m^2)$;
 - в. Паралельне застосування алгоритму ДКТ до розділених матриць 8×8 .

якщо $x = 0$, інакше 1 якщо $x > 0$

- 4) паралельне квантування – процес зменшення кількості біт необхідних для збереження значення цілочисельного типу зменшенням точності та множення матриці ДКТ з окремою матрицею квантування, що видаляє ірелевантні(невідповідні) частити;

Квантоване значення(i, j) = , округлене до найближчого цілого.

- 5) паралельне ентропійне кодування
 - а. Конвертація ДК коефіцієнтів у відповідні значення;

б. Переупорядкування ДКТ блоків зігзаг скануванням;

в. Ентропійне кодування(кодування довжин серій або методом Хаффмана).

Декодування відбуватиметься в 3 етапи:

- 1) інверсне квантування;
- 2) інверсне дискретне косинусне перетворення;
- 3) відновлення зображення з матриць груп пікселів.

Для виконання перетворення на різних етапах знадобляться такі бібліотеки:

- 1) OpenCV (Open Source Computer Vision Library) – розробка Intel написана на C, яка включає алгоритми для інтерпритації зображень, калібровки камер, усунення оптичних спотворень, визначення схожих фрагментів, аналіз переміщення об'єктів, визначення форми та слідкування за об'єктом, сегментація, розпізнавання жестів та ін. Передбачується використання бібліотеки для читання і запису даних зображення.
- 2) OpenMP (Open Multi-Processing) – бібліотека програмування для систем з розподіленою пам'яттю, задачею якої є ефективно автоматичне розпаралелювання, що виконується згідно моделі обміну через загальну пам'ять. Альтернатива інструментального засобу POSIX threads. Структуру OpenMP програми відображено на рисунку 3.2.

Особливості OpenMP структури:

- Основна гілка, і лише вона виконує всі послідовні області програми;
- При вході в паралельну область породжуються додаткові гілки;
- Після проходження кожна гілка отримує свій унікальний номер, при чому мастер-гілка завжди має номер 0;
- Всі гілки оперують одним і тим же кодом, що відповідає паралельній області;

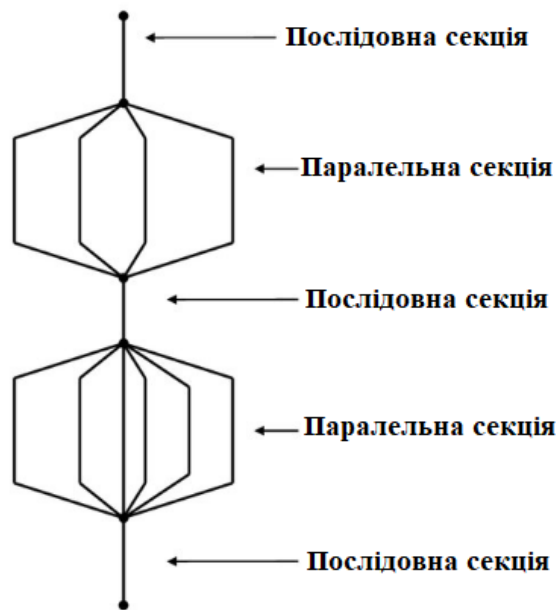


Рисунок 3.2 – Структура OpenMP програми

— При виході з паралельної області основна гілка очікує завершення виконання інших гілок для подальшого виконання коду програми, входити і виходити з ключових областей надано можливість тільки мастер-гілці.

В паралельній області всі змінні програми поділяють на публічні(SHARED) та локальні(PRIVATE).

SHARED змінні існують в одному екземплярі і доступні всім гілкам під однаковим ім'ям. Створення локальної змінної викликає породження свого екземпляру даної змінної для кожної гілки, а зміни в локальній змінній в одній із гілок не впливає на значення в інших локальних гілках.

Окрім конструкцій керування роботи над даними PRIVATE та SHARED оперує такими ключовими конструкціями як директива `parallel` для створення потоків, директиви `DO / for` та `section` для розподілення роботи між потоками, `critical` `atomic` і `barrier` для синхронізації потоків, процедури підтримки часу виконання, такі як `omp_get_thread_num` та ін. та змінними оточення.

Сценарій технік розпаралелювання використовуватиметься для розпаралелювання роботи з різними блоками зображення. Використання

директиви `#pragma omp parallel for` з різними техніками запуску, такими як `Static`, `Dynamic`, `Runtime` та `Guided`.

Розпаралелювання алгоритмів ущільнення дозволить скоротити час обробки в рази, і чим більша вибірка зображень – тим очевидніше буде перевага паралельного виконання у порівнянні з класичним послідовним алгоритмом.

На рисунку 3.3 наведена діаграма де за метрики взято час виконання в секундах по вертикалі та кількість зображень по горизонталі для методу DWT.

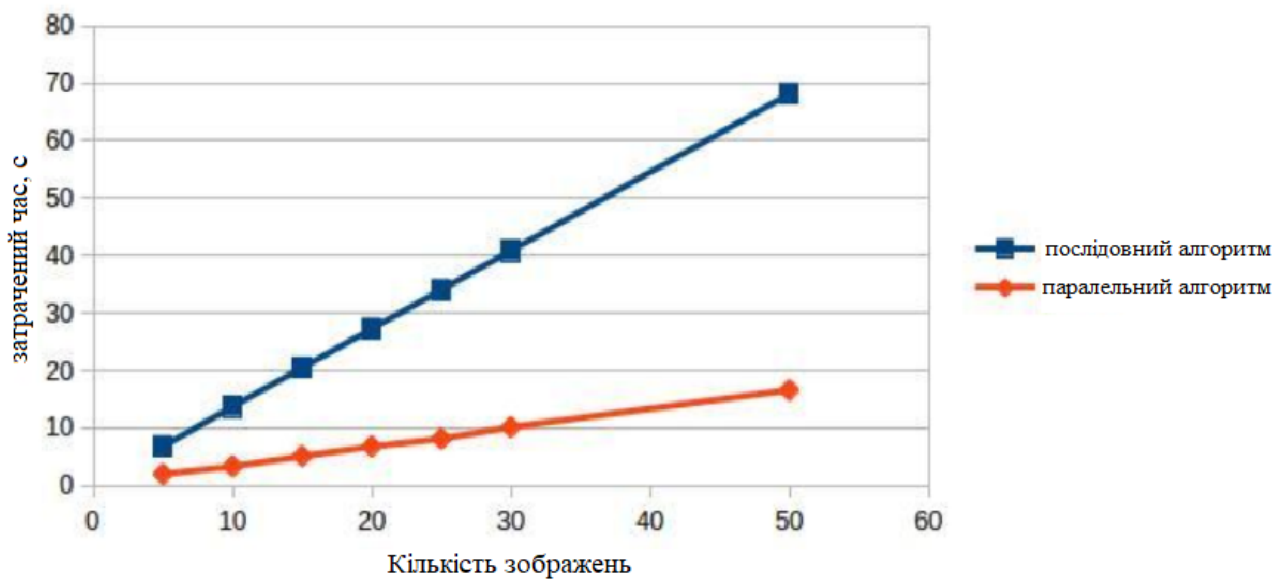


Рисунок 3.3 – Графік порівняння ефективності послідовного алгоритму та модифікованого паралельного алгоритму ущільнення зображень з використанням дискретного вейвлет перетворення

Таблицю порівняння затраченого часу на обробку групи зображень представлено у додатку В.

Можемо обчислити середнє значення прискорення виконання операцій над групою зображень за допомогою формули

$$\text{Прискорення}_{\text{сер}} =$$

Можна зробити висновок що із збільшенням кількості даних для обробки ефективність паралельного алгоритму зростає у порівнянні з класичною послідовною реалізацією. В реальних сценаріях роботи це може стати у нагоді в задачах, де необхідно опрацьовувати великі об'єми графічних даних.

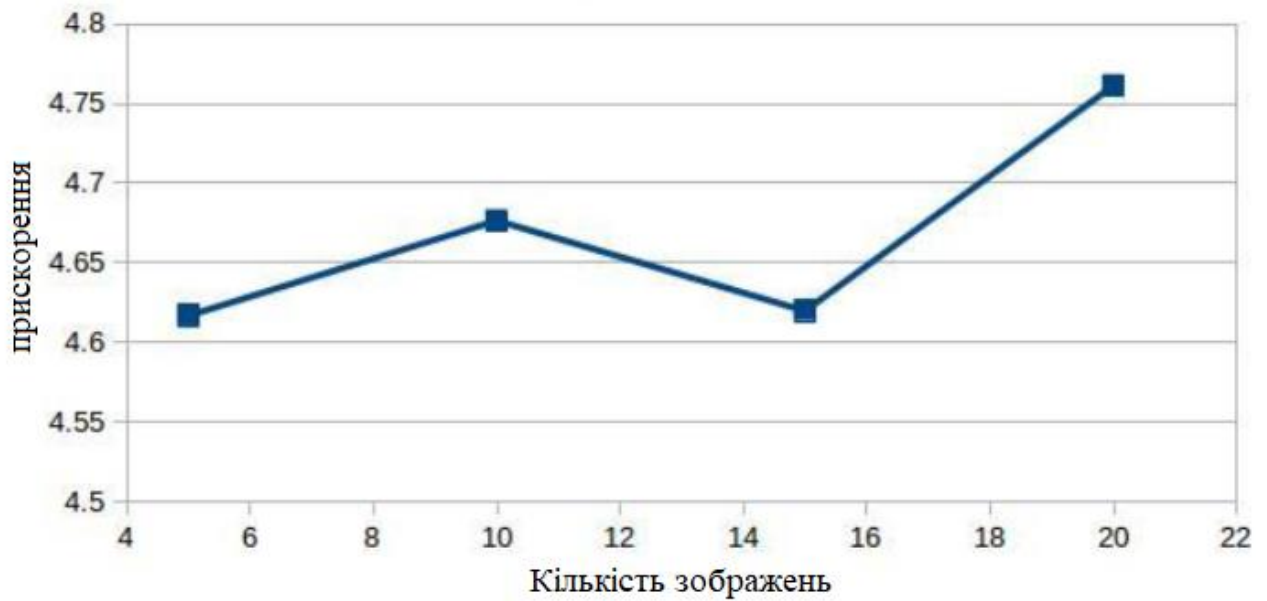


Рисунок 3.4 – Графік прискорення роботи алгоритму

3.2 Модифікація алгоритму фільтрації для виконання паралельних обчислень

Першочерговою задачею методів фільтрації зображень є видалення шумів, та окрім того в розділі 2 ми визначили, що фільтрація є одним з етапів ущільнення зображень. Також при матричній обробці систолічній обробці даних ми можемо виконувати однорідні операції над різними групами пікселів, тому можна вважати що фільтрацію зображення в деяких випадках можна виконувати з іншими етапами процесу ущільнення цифрового зображення.

В розділі 1 ми визначили, що однією з найпопулярніших кольорових моделей є модель RGB [4]. Кожен піксель складається з трьох компонентів: R (червоний), G (зелений) і B (синій) в діапазоні $\{0, \dots, 255\}$.

Припустимо, що розрахунки будуть проводитися за допомогою моделі RGB. Нехай $R_{n \times m}$ - двовимірний масив з значеннями пікселів зображення, де n – кількість пікселів по ширині, а m – по висоті зображення.

$R[i, j]$ представляє собою позицію (i, j) на зображенні, $i \in \{1, \dots, n\}$, $j \in \{1, \dots, m\}$. Ідея базується на змінах зображення при переміщенні маски згортки на не оброблені пікселі. Нові значення трьох компонентів R, G, B залежать від пікселів у найближчій зоні до вже обчислених. Цей процес показаний на рисунку 3.5 на прикладі 3×3 маски. Нове значення позиції (x, y) залежить від 9 пікселів. Можливі

й інші розміри маски згортки. Чим більше маска, тим більша кількість пікселів враховується під час обробки. Вплив кожного пікселя визначається таблицею ваг, що називається фільтром.

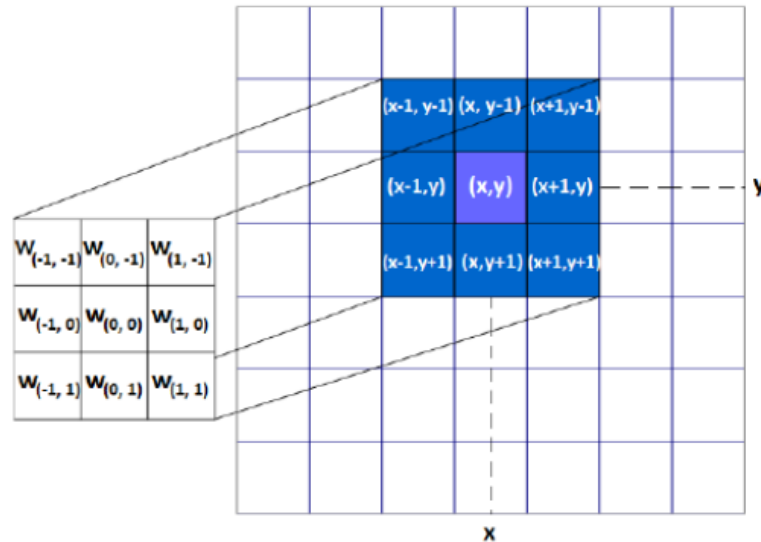


Рисунок 3.5 – Маска згортки 3x3 застосована в методі фільтрації

У роботі було перевірено три типи масок згортки: 3×3 , 5×5 та 9×9 . Чим більший фільтр, тим більше деталей розмиваються маскою [20]. Під час обчислень на великих масках необхідно створити допоміжне зображення з рамками, закресленими чорними пікселями (маски згортки перевищують оригінальні). Ця операція не має значного впливу на фінальне зображення, але це дає можливість використовувати фільтр для зображень, що менші за розмір фільтру. Шаблон для нового значення кожного компонента пікселя, що знаходиться в положенні (i, j) для маски 5×5 , є таким:

$$R'(i, j) = \frac{1}{M} \sum_{k=-2}^2 \sum_{l=-2}^2 w(k, l) \cdot R(i + k, j + l),$$

де M - сума всіх значень масиву (маски згортки), $w(k, l)$ - вага пікселя, розташованого в положенні $(i + k, j + l)$, а $R(i + k, j + l)$ дорівнює попередньому значенню пікселя в заданій позиції. Можливий випадок коли $M = 0$. У цій ситуації коефіцієнт пропущений. Шаблиони для масок згортки іншої розмірності створюються аналогічно.

За таких умов можливо використати фільтр Лапласа(рисунок 3.6), можлива реалізація якого запропонована в розділі 2. Головним завданням фільтру є виділяти границі. Те, як саме фільтр працює з вікном зображено на рисунку 3.7(збільшена копія представлена у додатку Г)

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 2 & 2 & 2 & 1 & 1 & 0 \\ 1 & 2 & 4 & 5 & 5 & 5 & 4 & 2 & 1 \\ 1 & 4 & 5 & 3 & 0 & 3 & 5 & 4 & 1 \\ 2 & 5 & 3 & -12 & -24 & -12 & 3 & 5 & 2 \\ 2 & 5 & 0 & -24 & -40 & -24 & 0 & 5 & 2 \\ 2 & 5 & 3 & -12 & -24 & -12 & 3 & 5 & 2 \\ 1 & 4 & 5 & 3 & 0 & 3 & 5 & 4 & 1 \\ 1 & 2 & 4 & 5 & 5 & 5 & 4 & 2 & 1 \\ 0 & 1 & 1 & 2 & 2 & 2 & 1 & 1 & 0 \end{pmatrix}$$

Рисунок 3.6 – Маска фільтру Лапласа 3x3, 5x5, 9x9 пікселів

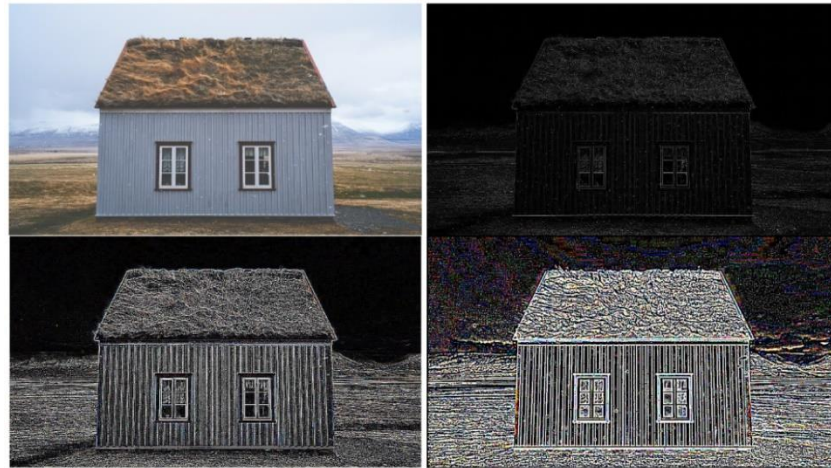


Рисунок 3.7 – робота фільтру Лапласа з масками 3x3, 5x5, 9x9 пікселів

Фільтрація є складною операцією з великою кількістю складних обчислень, і чим більше пікселів в зображенні – тим більше розрахунків необхідно буде провести, відповідно час виконання розрахунків може істотно збільшитись. При аналізі великої кількості знімків вирішальне значення має мінімізація часу на обчислення. Ідея, паралельної обробки спирається на те, щоб паралельно зсувати матриці згортки зображення. Фото можна розділити на прямокутники, що відповідають кількості потоків (приклад для 4 потоків зображено на рисунку 3.8). Тоді кожен потік застосовує фільтр до виділеної області і не впливає на інші частини. В результаті всі фільтровані фрагменти об'єднані в одне зображення.

Постає лише одне питання що полягає у визначенні способу призначення відповідних областей для потоків. Кожен потік після запуску отримує власний index (нумерація починається з нуля). Нехай w - ширина оброблюваного зображення (у пікселях). Тоді кількість піксельних потоків для кожного дорівнюватиме $\lceil \frac{w}{t} \rceil$. Функція обмеження необхідна, коли розділити групи пікселів між потоками не можна так, щоб їх кількість виразити цілим числом. У такій ситуації останній потік буде мати трохи більше пікселів для обчислення. У алгоритмі 1 представлено псевдокод методу паралельної фільтрації.

Алгоритм 1 – Псевдокод методу паралельної фільтрації зображень:

Input: зображення для фільтрації, розмір: ширина w , висота h , маска згортки, кількість потоків t .

Обрахувати кількість пікселів для обробки на кожен потік: .

Створити t потоків.

For $i = 0$ to $t - 1$ *do*

 Встановити межі для потоків від $i \cdot \lceil \frac{w}{t} \rceil + 1$ to $i \cdot \lceil \frac{w}{t} \rceil + \lceil \frac{w}{t} \rceil$.

If $i = t - 1$ *then*

 Встановити діапазон для $(t-1)$ -ого потоку з $(t - 1) \cdot \lceil \frac{w}{t} \rceil + 1$ до w

End if

Фільтр визначив зони згідно маски покриття

End for

З'єднати всі частини в одне зображення

Протестовано застосування трьох фільтрів Лапласа. У всіх випадках було проведено 100 вимірювань, а результати були усереднені. Досліджуване зображення має ширину 1200 пікселів та висоту 676 пікселів. Для кожної маски згортання зображення фільтрували, використовуючи 1, 2, 4, 8, 16 і 32 потоків. Алгоритм реалізований мовою C. Детальні результати експериментів представлені в табл. 3.2 і показано на рис. 3.8-3.10. На всіх графіках горизонтальна вісь представляє кількість потоків, а вертикальна вісь - час.

Можна помітити, що навіть поділ зображення на 2 прямокутники пришвидшує складні обчислення (53% -57% часу обчислення для одного потоку). У разі масці

згортки 9×9 час було скорочено з 141 до 75 секунд. На рис. 3.8-3.10 показано гіперболічне скорочення обчислювального часу. Різниці між послідовними випадками стають меншими, але все-таки найбільш вигідними програмами при використанні 32 потоків. Чим більша маска згортки, тим більше часу можна заощадити завдяки багатопотоковому (7-10 секунд у випадку маски 3×3 , 22-42 секунди у випадку 5×5 маски та 66-131 секунд з маскою 9×9). Отже, багатопоточний метод може бути корисним інструментом для прискорення обчислень.

Дослідження показали, що запропонована паралелізація суттєво скорочує час обчислень. Програма видає найкращі результати, коли використовуються всі потоки процесора. Чим більше зображення та обчислювальна маска, тим важливіше скорочення часу, витраченого на розрахунки. Варто згадати, що час, необхідний для фіксації зображення розміром 1200×676 пікселів, використовуючи лише один потік, а маска згортки 9×9 обробляється дві хвилини. Можна зробити висновок, що зображення більшого (наприклад, розміром 6000×6000 пікселів або вище) вимагає декількох хвилин, особливо у випадку великих масок згортки, також більших за 9×9 . Прагнення до значного скорочення часу обчислення є надзвичайно важливим. Метод можна додатково модифікувати, вивчаючи інші паралельні алгоритми або намагаючись обробити багато зображень за той самий час.

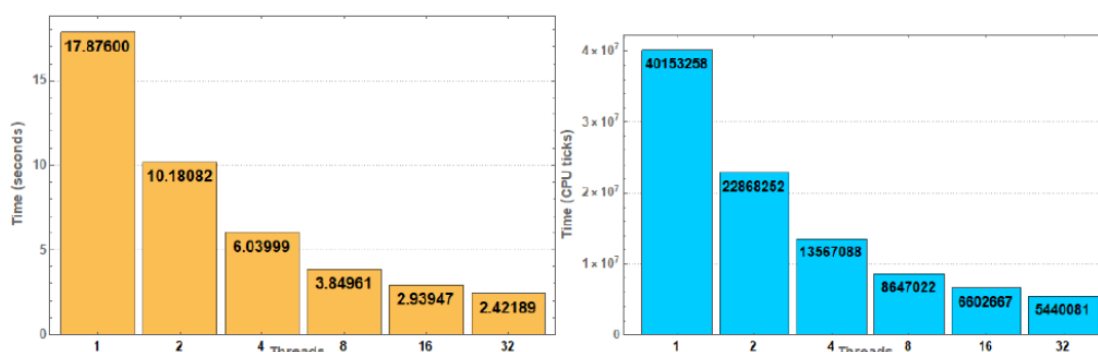


Рисунок 3.8 – Діаграма виконання обчислень маскою згортки 3×3 .

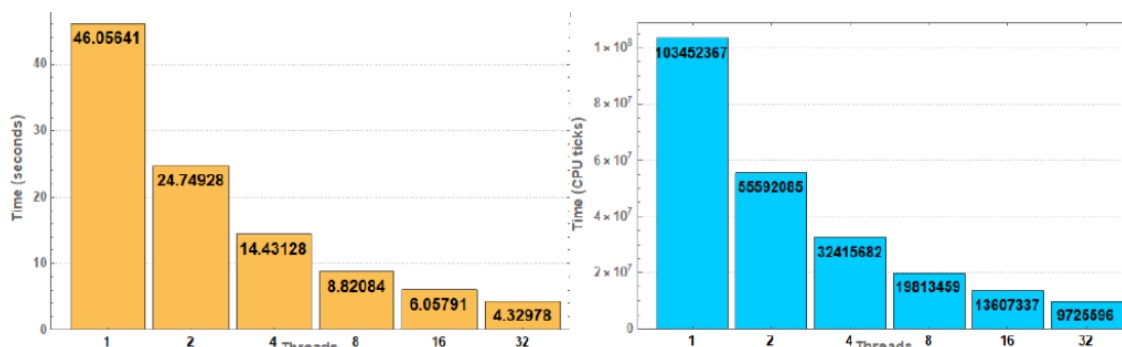


Рисунок 3.9 – Діаграма виконання обчислень маскою згортки 5x5.

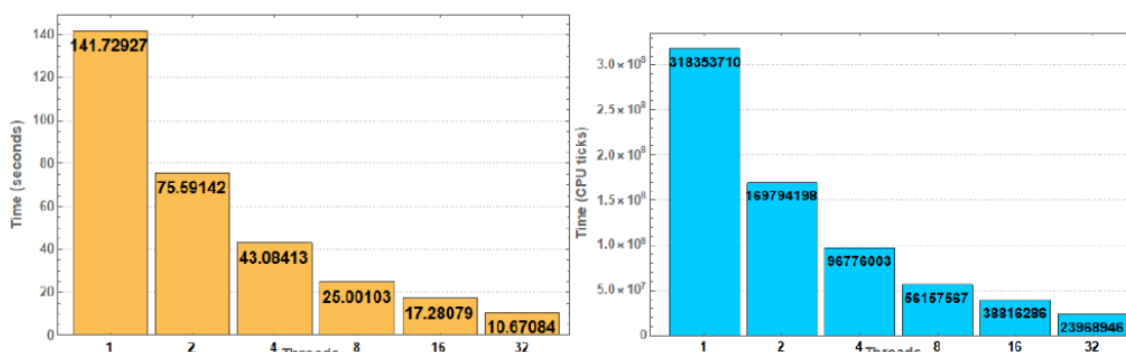


Рисунок 3.10 – Діаграма виконання обчислень маскою згортки 9x9.

У роботі запропоновані модифікації методів фільтрації, сегментації та ущільнення зображень, які дозволяють заощадити значний час під час ущільнення великих масивів графічних даних.

Паралельна фільтрація Лапласа, виконана на багатопотоковій обчислювальній системі з кількістю діючих потоків наближеними до кількості фізичних потоків дає значний приріст в швидкості виконання обчислень. Доведено, що чим більший розмір маски згортки, що проходить по зображенню, чим більшого розміру зображення і чим більше їх кількість тим складніші обчислення, і тим більший вигащ при розпаралелюванні процесу паралельної фільтрації.

Сучасні технології розпаралелювання та засоби мов високого рівня програмування мають в своєму інструментарії бібліотеки, що дозволяють експериментально перевірити доцільність використання паралельного коду на багатоядерних багатопоточних обчислювальних системах.

Таблиця 3.2 – результати фільтрації зображень за допомогою засобів паралельних обчислень, вибірка 100 зображень.

Маска згортки 3x3					
К-кість потоків	Середній час, с	%	відхилення	Середнє число тактів	відхилення
1	17.87600	100%	0.23354	40153258	524569
2	10.18082	58,95%	0.30065	22868252	675319
4	6.039999	33,79%	0.28318	13567088	636091
8	3.94961	21,54%	0.25843	8647022	580477
16	2.93947	16.44%	0.21808	6602667	489848
32	2.42189	13.55%	0.08085	5440081	181596
Маска згортки 5x5					
1	46.056041	100%	1.08065	103452367	2427362
2	24.74928	53,74%	0.26110	55592085	586472
4	14.43128	31,33%	0.26109	32415682	1912613
8	8.82084	19,15%	0.97053	19813459	2180018
16	6.05791	13,15%	0.49399	13607337	1109609
32	4.32978	9,40%	0.15660	9725596	351746
Маска згортки 9x9					
1	141.72927	100%	1.55923	318353710	3502354
2	75.59142	53,34%	1.27017	169794198	2853071
4	43.08413	30,40%	4.14568	96776003	9312067
8	25.00103	17,64%	2.58287	56157567	5801661
16	17.28079	12,19%	1.61762	38816286	3633518
32	10.67084	7,53%	0.51039	23968946	1146444

В цьому в нагоді може стати апаратно-програмна архітектура Nvidia CUDA, що дозволяє виконувати обчислювання, пов'язані з графічними масивами швидше ніж на центральному процесорі при правильній оптимізації коду для виконання, а високі мови програмування в свою чергу володіють такими бібліотеками, як Threads в Java та OpenMP, OpenCV в C.

Результати роботи показують, що розвиток в цьому напрямку дозволить в майбутньому значно скоротити час виконання обчислень над масивами графічних

даних на складних багатопотокових обчислювальних системах, які в сучасному світі зустрічаються на будь-якому підприємстві, організації, в будинках громадян.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки засобів ущільнення ущільнення зображень за допомогою методів паралельної обробки даних

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання комерційного потенціалу буде здійснене за 12-ма критеріями, що наведені в додатку Д.[17]

Результати оцінювання комерційного потенціалу розробки зведено в таблицю 4.1 за зразком.

Таблиця 4.1 – Результати оцінювання комерційного потенціалу розробки

Критерії	Експерти		
	1	2	3
	Бали, виставлені експертами:		
1	1	2	2
Ринкові переваги (недоліки):			
2	4	4	4
3	3	2	3
4	2	2	3
5	2	2	2
Ринкові перспективи			
6	3	2	3
7	4	4	4
Практична здійсненність			
8	1	1	2
9	3	2	3
10	3	4	4
11	3	3	3
12	3	3	3
Сума балів	СБ ₁ =32	СБ ₂ =31	СБ ₃ =36
Середньоарифметична сума балів $\overline{СБ}$	33		

За даними таблиці 4.1 можна зробити висновок, щодо рівня комерційного потенціалу розробки. Зважимо на результат й порівняємо його з рівнями комерційного потенціалу розробки, що представлено в таблиці 4.2.

Таблиця 4.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Отже, зробимо висновок що на основі таблиці 4.3 середньоарифметична сума балів на основі висновків експертів в 33 бали відповідає рівню «вище середнього», знаходячись в межах від 31 до 40 балів.

Проаналізуємо суть технічної проблеми та розглянемо аналоги.

Актуальність розробки алгоритму полягає у відсутності аналогів на ринку програмних та апаратних засобів.

Найближчим за принципом роботи алгоритмом ущільнення з втратами є алгоритм ущільнення запропонований Joint Photographic Expert Group на базі якого був реалізований алгоритм запропонований в даній роботі. Вибір бази ґрунтується на JPEG оскільки алгоритм поширюється вільною ліцензією Independent JPEG Group License. Оскільки JPEG не є комерційним продуктом кошти розробникам не відраховуються. Особливості, переваги і недоліки JPEG були детально проаналізовані в розділі 1.

Запропонований алгоритм вирішує проблему аналогу за допомогою розпаралелювання етапів обробки зображення – а саме фільтрації та сегментації, що допоможе зберегти час при обробці великих масивів графічних даних.

Для коректної роботи кінцевий користувач програмного продукту, що реалізовуватиме роботу засобу ущільнення зображень повинен володіти мультипроцесорний ПК. Важливо зазначити що програмний продукт, що реалізує роботу алгоритму за мету має обробку зображення за допомогою використання апаратних особливостей архітектури ПК, а саме процесорних блоків, що виконують обчислення над масивами даних, що поступають на входи ЛЕ

паралельно на відміну від існуючих методів що обробляють дані одним потоком послідовно. Алгоритм побудований таким чином, щоб забезпечити безвідмовну роботу на процесорах x86 архітектури. Пропонується простий сценарій для взаємодії кінцевого користувача з програмним продуктом, що реалізовуватиме алгоритм в практичних задачах, який описаний в розділі 3. Головною перевагою розробки в порівнянні з найближчим аналогом є економія часу та апаратних затрат на виконання коду при дотриманні вимог до програмного продукту. Продукт просуватиметься на ринок по вільній ліцензії.

Цінність розробки полягає в економії часу на одному з етапів кодування цифрових зображень – фільтрації. Ця перевага перед послідовним алгоритмом фільтрації особливо помітна при роботі з великими масивами графічних даних.

Ідея реалізовуватиметься в якості модифікованого алгоритму ущільнення JPEG і може використовуватися програмним забезпеченням для обробки цифрових зображень та у веб сервісах, що дозволяють обробляти зображення.

Продукт може використовуватися як в некомерційних(для домашнього користування) так і в комерційних(в дата центрах, картографічних сервісах, при обробці зображень в спеціалізованих центрах) цілях для ущільнення та фільтрації зображень.

Продукт має за мету прискорити процес ущільнення зображень та усунути артефакти зображень за допомогою фільтру, тому вимоги до продукту обумовлені метою роботи.

Розповсюдження продукту відбуватиметься за комерційною ліцензією.

Для того, щоб здійснити розробку продукту орієнтовно знадобиться близько 8 місяців за наявності технічних спеціалістів, а саме інженерів програмного забезпечення та експерта консультанта в області моделей паралельного програмування

Продукт передбачається розповсюджувати за допомогою інтернет магазинів спеціалізованого програмного забезпечення. Поширювати інформацію планується за допомогою таргетингової реклами в мережі інтернет.

4.2 Прогнозування витрат на виконання науково-дослідної, (дослідно-конструкторської) та конструкторсько-технологічної роботи

Прогнозування витрат на виконання даних робіт може складатись з таких етапів:

- 1) розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи
- 2) розрахунок загальних витрат на виконання даної роботи.
- 3) прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

1. Основна заробітна плата кожного із розробників (дослідників) Z_o :

$$Z_o = \frac{M}{T_p} \cdot t \text{ грн.}, \quad (4.1)$$

де M – місячний посадовий оклад конкретного розробника, грн.

T_p – число робочих днів в місяці; $T_p = 21$ день;

t – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки бажано звести до таблиці:

Таблиця 4.3 - Прогнозування витрат

Найменування посади виконавця	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на оплату праці, грн.	Примітка
1. Науковий керівник	6100	291	3	873	
2. Інженер ПЗ	7300	347	63	21900	
3. Експерт в області паралельних обчислень	6000	286	21	6000	
Всього				28773	

Додаткова заробітна плата Z_d всіх розробників та робітників, які брали

участь у виконанні даного етапу роботи, розраховується як 10% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$З_д = 0.1 \cdot 28773 = 2877,3 \text{ (грн)}.$$

Нарахування на заробітну плату $H_{зп}$ розробників та робітників, які брали участь у виконанні даного етапу роботи, становить 22%:

$$H_{зп} = (З_о + З_д) \cdot \frac{\beta}{100}, \quad (4.2)$$

де $З_о$ – основна заробітна плата розробників, грн.;

$З_д$ – додаткова заробітна плата всіх розробників та робітників, грн.;

β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, 22 %.

$$H_{зп} = 28773 \cdot 0.22 = 6330 \text{ (грн)}.$$

Амортизація обладнання, комп'ютерів та приміщень А, які використовувались під час (чи для) виконання даного етапу роботи. Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

У спрощеному вигляді амортизаційні відрахування А в цілому бути розраховані за формулою:

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12} \text{ грн} \quad (4.3)$$

де $Ц$ – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн;

H_a – річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що $H_a = (10...25)\%$;

T – термін, використання обладнання, приміщень тощо, місяці.

Зроблені розрахунки зведемо до таблиці (за 3 місяці розробки):

Таблиця 4.4 - Амортизаційні відрахування

Найменування	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн.
Комп'ютери	42000	10	3	1500
Приміщення	112000	10	3	100
Всього				1600

5. Витрати на силову електроенергію V_e :

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi} \text{ (грн)}, \quad (4.4)$$

Де V – вартість 1 кВт-год. електроенергії, в 2019 р. $V \approx 1,68$ грн./кВт;

Π – установлена потужність обладнання, кВт(1.16кВт);

Φ – фактична кількість годин роботи обладнання, годин,

K_{Π} – коефіцієнт використання потужності; ($K_{\Pi} < 1$, $K_{\Pi}=0,85$).

$$V_e = 1.68 \cdot 1.16 \cdot 528 \cdot 0.85 = 874,62 \text{ (грн)}.$$

Також потрібно врахувати витрати на доступ до мережі Інтернет, що використовувався під час виконання роботи.

Витрати за доступ до Інтернет можна розрахувати за формулою:

(4.6)

де $C_{ді}$ – це ціна доступу за місяць;

T – кількість місяців використання доступу до мережі.

Отже, витрати на доступ до мережі Інтернет становлять:

$$V_{ді} = 150 \cdot 4 = 600 \text{ (грн)}.$$

Величина інших витрат складе:

$$V_{ін} = 30 \cdot 100\% \text{ [грн]}. \quad (4.7)$$

$$B_{ін} = 28773 \cdot 1 = 28773 \text{ (грн)}.$$

Обчислимо витрати на виконання даної роботи, що являтиме собою суму всіх попередніх витрат:

$$B = Z_o + Z_d + H_{зп} + A + H_A + B_e + B_{ін} \text{ [грн]},$$

$$B = 26673 + 2667,3 + 6330 + 1600 + 874,72 + 600 + 28773 = 67518,02 \text{ (грн)}$$

2-й етап: розрахунок загальних витрат на виконання даної роботи

Загальна вартість всієї наукової роботи визначається за $B_{заг}$ формулою:

$$B_{заг} = \frac{B}{\alpha}, \quad (4.8)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях, $\alpha = 0,4$.

$$B_{заг} = 75020 \text{ (грн)}.$$

3-й етап: прогнозування загальних витрат на виконання та впровадження результатів виконаної наукової роботи. Прогнозування загальних витрат $ЗВ$ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{B_{заг}}{\beta}, \quad (4.9)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Так, якщо розробка знаходиться:

на стадії науково-дослідних робіт, то $\beta \approx 0,1$;

на стадії технічного проектування, то $\beta \approx 0,2$;

на стадії розробки конструкторської документації, то $\beta \approx 0,3$;

на стадії розробки технологій, то $\beta \approx 0,4$;

на стадії розробки дослідного зразка, то $\beta \approx 0,5$;

на стадії розробки промислового зразка, $\beta \approx 0,7$;

на стадії впровадження, то $\beta \approx 0,9$.

Оскільки даний передавач знаходиться на стадії розробки технологій, то $\beta = 0,4$

(грн).

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки засобів паралельної ущільнення за допомогою методів паралельної обробки даних

Прогнозування комерційних ефектів здійснюється у двох основних випадках: коли можна прямо оцінити зростання чистого прибутку підприємства від впровадження результатів наукової розробки або коли не можливо прямо оцінити зростання чистого прибутку підприємства від впровадження результатів наукової розробки.

Дана розробка підпадає під другий випадок, оскільки вона знаходиться на стадії науково-дослідних робіт, тому для розрахунків будемо використовувати дані підприємства з аналогічною системою передачі. Розрахунки здійснюватимемо за формулою 3.14:

$$\Delta\Pi_i = \sum_1^n (\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{U}{100}\right), \quad (4.10)$$

Де ΔC_o – покращення основного оціночного показника від впровадження результатів розробки у даному році, $\Delta C_o = 80$ грн ;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки, $N=1$ шт.;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки, $\Delta N=300$ шт. ;

C_o – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $C_o=170$ грн;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки, $n=3$;

λ – коефіцієнт, який враховує сплату податку на додану вартість $\lambda = 0,8547$;

ρ – коефіцієнт, який враховує рентабельність продукту, $\rho = 0,25$;

U – ставка податку на прибуток, $U = 18\%$.

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Розрахуємо абсолютну ефективність вкладених інвестицій $E_{абс}$.

Для цього використаємо формулою:

$$E_{абс} = (ПП - PV), \quad (4.10)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.; PV – теперішня вартість інвестицій $PV = ЗВ$, грн.

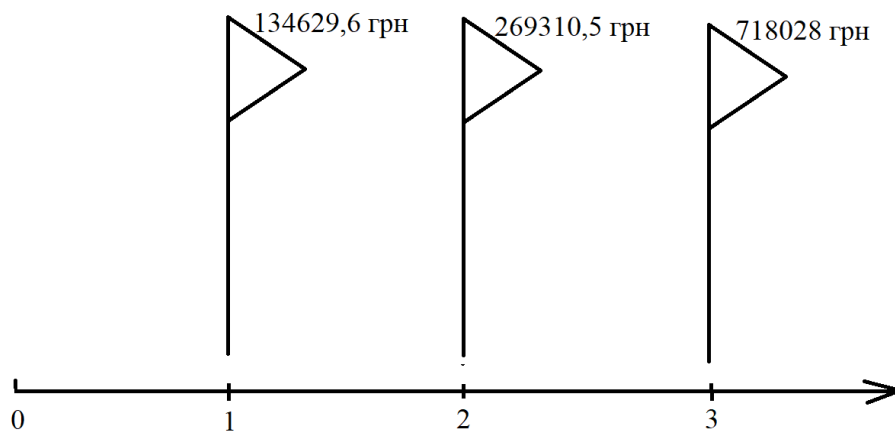


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_1^m \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, $t=3$;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; $\tau=0,1$;

t – період часу від моменту отримання чистого прибутку до точки „0”.

5-й крок. Розраховують відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_v . Для цього користуються формулою:

$$E_v = T_{ж} \sqrt{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.12)$$

де $E_{абс}$ – абсолютна ефективність вкладених інвестицій, грн; PV – теперішня вартість інвестицій $PV = 3B$, грн; $T_{ж}$ – життєвий цикл наукової розробки, роки.

У загальному вигляді мінімальна (бар'єрна) ставка дисконтування τ мін визначається за формулою:

$$\tau = d + f, \quad (4.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках;

$d = 0,15$;

f – показник, що характеризує ризикованість вкладень; зазвичай, $f = 0,09$.

Оскільки $E_v > \tau$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ можна розрахувати за формулою:

Оскільки $T_{ок} < 3$ років, то це свідчить про доцільність фінансування такої розробки.

Основна заробітна плата розробників(дослідників) склала 28773 грн.

Додаткова заробітна плата всіх розробників складає 2877,3 грн. Нарахування на заробітну плату розробників - 6330 грн. На третьому етапі було спрогнозовано загальні витрати на виконання та впровадження результатів наукової роботи, що склали 187550 грн.

У четвертому розділі було проведено розрахунок ефективності вкладених інвестицій та періоду їх окупності. Приведена вартість всіх чистих прибутків склала 843591,75 грн. Абсолютна ефективність вкладених інвестицій – 656041,75 грн. Було обраховано відносну ефективність вкладених коштів в наукову розробку інвестицій. Вона складає 1,165. Оскільки вона менша за мінімальну ставку дисконтування, що складає 0,24, інвестор буде більше чим зацікавлений у фінансуванні даної наукової розробки. Також, оскільки термін окупності складає 0.9 років, що є меншим за три роки, фінансування такої розробки є доцільним.

4.5 Висновки до економічного розділу

В даному розділі було виконано оцінювання комерційного потенціалу розробки засобу обробки зображень за допомогою методів паралельної обробки даних.

Проведено технологічний аудит з залученням трьох незалежних експертів. Визначено, що рівень комерційного потенціалу розробки вище середнього. Аналіз комерційного потенціалу розробки показав, що розробка не має аналогів на ринку, а найближчого за принципом конкурента розробка випереджає в швидкості роботи з даними. Продукт має кращі функціональні показники, а тому є конкурентоспроможним товаром на ринку. Існуючі переваги нової розробки дозволять швидко її популяризувати.

Згідно із розрахунками всіх статей витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи загальні витрати на розробку складають грн.

Термін окупності вкладених у реалізацію проекту інвестицій становить ~11 місяців, що також свідчить про доцільність фінансування нової розробки.

ВИСНОВКИ

В магістерській кваліфікаційній роботі проведено аналіз зображень, їх структури, типів, форматів. Проаналізовано методи обробки зображень, а саме методи виділення границь, фільтрації та сегментації. Також було проведено аналіз апаратних апаратних архітектур паралельної обробки даних та програмних засобів, що дозволяють розпаралелювати виконання коду

В роботі розглянуті традиційні методи фільтрації та сегментації зображень та запропоновано методи розпаралелювання, що дозволять виконати операції над зображеннями швидше на сучасних мікропроцесорних системах з кількістю обчислювальних ядер більше 1. Була покращена швидкість роботи алгоритмів на мультипроцесорних системах за допомогою методів паралельної обробки даних.

Розглянуто варіанти розпаралелення виконання коду оптимізованого як під центральний процесор, так і для виконання на відеопроцесорі з використанням технології Nvidia CUDA.

Проведено порівняльний аналіз існуючих класичних методів з потенційною розробкою як на основі центрального процесора так і за допомогою технології CUDA. Підтверджено потенційність розвитку алгоритму в майбутньому.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Грузман И.С., Киричук В.С., Косых В.П., Перетягин Г.И., Спектор А.А. Цифровая обработка изображений в информационных системах. — Новосибирск: Изд-во НГТУ, 2002. — 352 с.
2. Вычислительные методы, алгоритмы аппаратурно-програмный инструментарий параллельного моделирования природных процессов/ М.Г. Куриносков[и др]; отв. Ред. В.Г. Хорошевский; Рос. Акад. Наук, Сиб. Отд-ние, Ин-т физики полупроводников им А.В. Ржанова [и др.] – Новосибирск: Изд-во СО РАН, 2012. – 355 с.
3. Цилькер Б.Я. Организация ЭВМ и систем: учебник для вузов / С.А. Орлов, Б.Я. Цилькер. - СПб.: Питер, 2011. - 688 с.
4. Черемков Ю.В. Вычислительные системы, сети и телекоммуникации. Издание второе, исправленное и дополненное – М.: ДМК Пресс, 2009. – 184с.
5. Новожилов О.П. Архитектура ЭВМ и систем: учеб. Пособие для бакалавров – М.: Издательство Юрайт, 2015. – 527 с.
6. Szymon Graboskwi, Wojciech Bienieck "A two-pass median-like filter for impulse noise removal in multi-channel images". KOSYR 2003, str. 195-200
7. S. Mallat "A wavelet tour of signal processing", Academic Press, London, 1998.
8. Miroslav Vrankic, Damir Sersic "Image Denoising Based on Adaptive Quincunx Wavelets". In Proc. of the 2004 IEEE 6th Workshop on Multimedia Signal Processing, (MMSP 2004), Siena, Italy, September 29 - October 01, 2004, pp. 251-254
9. Vladan Velisavljevic, Baltasar Beferull-Lozano1, Martin Vetterli, Pier Luigi Dragotti "Discrete Multi-Directional Wavelet Bases". Proc. IEEE International Conference of Image Processing (ICIP2003)

10. Гонсалес Р., Вудс Р. Цифровая обработка изображений. — М.: Техносфера, 2006. — 1072 с.
11. Фисенко В.Т., Фисенко Т.Ю., Компьютерная обработка и распознавание изображений: учеб. пособие. – СПб: СПбГУ ИТМО, 2008. – 192 с
12. Яне Б. Цифровая обработка изображений. – М.: Техносфера, 2007. – 584 с.
13. Шапиро Л., Стокман Дж. Компьютерное зрение. — М.: БИНОМ. Лаборатория знаний, 2006. — 752 с.
14. Лайонс Ричард. Цифровая обработка сигналов: 2 изд. — М.: ООО Бином-Пресс, 2006. — 656 с.
15. S.G. Chang, B. Yu, and M. Vetterli, "Spatially adaptive wavelet thresholding with context modeling for image denoising". *IEEE Trans. Image Proc.*, Sept. 2000.
16. D. Darian Muresan, Thomas W. Parks "Adaptive principal components and image denoising". *IEEE ICIP 2003*
17. Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт / Уклад. В. О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.
18. Mohammad H. Asghari, "Discrete Anamorphic Transform for Image Compression", *IEEE Signal Processing Letters*, VOL. 21, NO. 7, JULY 2014
19. Joaquín Franco, Gregorio Bernabé, Juan Fernández and Manuel E. Acacio, "A Parallel Implementation of the 2D Wavelet Transform Using CUDA", 1066-6192/09 *IEEE 2009*

20. D. D. Burdescu, M. Brezovan, L. Stănescu, C. S. Spahiu, and D. C. Ebăncă, “Graph-based semantic segmentation for 3d digital images,” in 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2017, pp. 114–119, DOI:10.1109/WAINA.