

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається

Завідувач кафедри

_____ Скарга-Бандурова І.С.

«____» _____ 20__ р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Програмно-алгоритмічні засоби комп'ютерного моделювання дифракції та обробки
експериментальних зображень

Освітньо-кваліфікаційний рівень “Магістр”

Спеціальність 123 – «Комп'ютерна інженерія»

Науковий керівник роботи:

(підпис)

Кривуля Г.Ф

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Критська Я.О.

(ініціали, прізвище)

Студент:

(підпис)

Рязанцев А.О.

(ініціали, прізвище)

Група:

КІ-17зм

Северодонецьк 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень Магістр
Напрямок підготовки _____
Спеціальність 123 – «Комп'ютерна інженерія»
(шифр і назва)
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____
І.С. Скарга-Бандурова
« _____ » _____ 20__ р.

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

Рязанцеву Андрію Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Програмно-алгоритмічні засоби комп'ютерного моделювання
комп'ютерного моделювання дифракції та обробки експериментальних зображень.

керівник проекту (роботи) Кривуля Геннадій Федорович, д.т.н., проф.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затвержені наказом вищого навчального закладу від _____ " 18 " 10 _____ 2019 р. № 221/48

2. Строк подання студентом роботи 10.01.2018

3. Вихідні дані до роботи Матеріали переддипломної практики,
тестові зображення, література у галузі розробки методів обробки зображень,
середовище розробки MATLAB.

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

1. Аналіз задачі дослідження

2. Дослідження методів обробки експериментальних зображень

3. Розробка ПАЗ обробки експериментальних зображень

4. Охорона праці

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
4	Критська Я.О.		

7. Дата видачі завдання 18.10.2018

Керівник

Завдання прийняв до виконання

_____ (підпис)

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Літературний огляд та опрацювання по темі дипломної роботи	2.07.2018 – 17.08.2018	
2	Складання плану роботи	20.08.2018 – 31.08.2018	
4	Розробка та узгодження з керівником першого розділу диплома	3.09.2018-5.10.2018	
5	Розробка та узгодження з керівником другого розділу диплома	8.10.2018 -26.10.2018	
6	Розробка та узгодження з керівником третього розділу диплома	29.10.2018 – 14.12.2018	
7	Розробка розділу "Охорона праці та безпеки в надзвичайних ситуаціях"	17.12.2018 - 28.12.2018	
8	Подання дипломної роботи на кафедру	10.01.2019	
9	Підготовка доповіді та графічного матеріалу	11.01.2019 – 16.01.2019	
10	Захист випускної кваліфікаційної роботи	18.01.2019	

Студент

Рязанцев А.О.

_____ (підпис)

_____ (прізвище та ініціали)

Науковий керівник

Кривуля Г.Ф.

_____ (підпис)

_____ (прізвище та ініціали)

АНОТАЦІЯ

Рязанцев А.О. Програмно-алгоритмічні засоби комп'ютерного моделювання дифракції та обробки експериментальних зображень.

Тема магістерської роботи присвячена розробленню програмно-алгоритмічного засобу для дослідження процесів дифракції та обробки експериментальних зображень. Дослідження характеристик оптичних полів дозволяє презентувати повний опис пучка, який характеризує його метрологічну спроможність: похибку вимірювань та динамічний діапазон. Розроблений програмно-алгоритмічний засіб дозволить виконувати теоретичні та експериментальні дослідження в оптичній лабораторії швидше, точніше та надійніше, що сприятиме розвитку прикладної оптики, зокрема метрології та мікроскопії.

Ключові слова: програмно-алгоритмічний засіб, обробка зображень, дифракція, метод Фур'є.

АННОТАЦИЯ

Рязанцев А.А. Програмно-алгоритмические средства компьютерного моделирования дифракции и обработки экспериментальных изображений.

Тема магистерской работы посвящена разработке программно-алгоритмического средства для исследования процессов дифракции и обработки экспериментальных изображений. Исследования характеристик оптических полей позволяет представить полное описание пучка, характеризующий его метрологическую состоятельность: погрешность измерений и динамический диапазон. Разработанное программно-алгоритмическое средство позволит выполнять теоретические и экспериментальные исследования в оптической лаборатории быстрее, точнее и надежнее, что будет способствовать развитию прикладной оптики, в частности метрологии и микроскопии.

Ключевые слова: программно-алгоритмическое средство, обработка изображений, дифракция, метод Фурье

ANNOTATION

Ryazantsev A.O. Software and algorithmic tools for computer simulation of diffraction and processing of experimental images.

The theme of the master's degree work is devoted to the development of software and algorithmic tool for studying the processes of diffraction and processing of experimental images. The study of the characteristics of optical fields allows us to present a complete description of the beam, which characterizes its metrological capacity: the measurement error and dynamic range. The developed software and algorithmic means will allow to perform theoretical and experimental research in the optical laboratory faster, more precisely and more reliable, which will promote the development of applied optics, in particular, metrology and microscopy.

Keywords: software algorithmic tool, image processing, diffraction, Fourier method.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	6
ВСТУП	7
1. АНАЛІЗ ЗАДАЧІ ДОСЛІДЖЕННЯ.....	9
1.1 Аналіз вимог для створення високоякісного комп'ютерного моделювання.....	9
1.2 Поняття моделі. Цілі моделювання.....	11
1.2.1 Види моделей.....	12
1.2.2 Комп'ютерне моделювання.....	13
1.3 Аналіз досліджуваного явища. Скалярна теорія дифракції.....	15
1.4 Постановка завдання.....	18
1.5 Технічне завдання.....	18
1.6 Висновки до першого розділу.....	19
2 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ МОДЕЛЮВАННЯ ЯВИЩА ДИФРАКЦІЇ ТА ОБРОБКИ ОПТИЧНИХ ЗОБРАЖЕНЬ	20
2.1 Чисельне інтегрування дифракції.....	20
2.1.1 Формули прямокутників.....	21
2.1.2 Формули Ньютона-Котеса.....	23
2.1.3 Оцінювання похибки при чисельному інтегруванні.....	25
2.2 Методи обробки зображень.....	26
2.2.1 Математичні моделі зображень.....	26
2.2.2 Статистичні методи аналізу зображень.....	29
2.2.3 Фільтрація зображень.....	31
2.2.3.1 Оптимальна лінійна фільтрація.....	33
2.2.3.2 Нелінійна фільтрація.....	35
2.3 Висновки до другого розділу.....	38
3 РОЗРОБКА ПРОГРАМНО-АЛГОРИТМІЧНОГО ЗАСОБУ ДЛЯ ОБРОБКИ ОПТИЧНИХ ЗОБРАЖЕНЬ	39
3.1 Призначення розробленого програмно-алгоритмічного засобу.....	39
3.2 Алгоритм реалізації моделі.....	39
3.3 Розробка ПАЗ в математичному пакеті MATLAB.....	40
3.4 Отримання експериментальних даних та реалізація ПАЗ.....	42
3.5 Результати реалізації алгоритму первинної обробки зображень.....	45
3.6 Результати реалізації алгоритму вторинної обробки зображень.....	47
3.7 Висновки до третього розділу.....	48
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	49

	5
4.1 Охорона навколишнього природного середовища.....	58
4.2 Висновки	59
ВИСНОВКИ	60
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61
Додаток А Код програми	66
Додаток Б. Комп'ютерна презентація.....	84

ПЕРЕЛІК СКОРОЧЕНЬ

ПАЗ – Програмно-алгоритмічний засіб;

ОВ – оптичний вихор;

ПМС – просторовий модулятор світла;

ШПФ – швидке перетворення Фур'є.

ВСТУП

Актуальність теми. У сучасній оптиці [1-3] активно досліджуються такі топологічні об'єкти, як максимуми, мінімуми, сідла і оптичні вихори в розподілах інтенсивності, фази і поляризації. Найбільш дивними властивостями володіє оптичний вихор, - лінія нульової амплітуди і невизначеної (сингулярної) фази, навколо якої циркулює енергія аналогічно руху повітряних мас в смерчі. Унікальні властивості топологічних об'єктів знайшли широке застосування в різних напрямках: від прецизійного метрологічних вимірювань менше довжини хвилі світла, а також оптичних пінцетів і багатоканальних маніпуляторів мікрочастинками до їх використання в квантових комп'ютерах.

Зупинимось на останньому застосуванні більш детально. Квантовий комп'ютер [4] концептуально схожий на цифровий класичний комп'ютер, в якому біти замінюються кубітами. Кубіти можуть перебувати в суперпозиції станів і можуть бути взаємно заплутаними - залежними (mutually entangled). Квантовий комп'ютер шукає рішення численних завдань, використовуючи послідовність логічних вентилів (logic gates) – базових елементів цифрової схеми, що виконують елементарні логічні операції, такі як додавання, множення, заперечення, виключення та інші, перетворюючи безліч вхідних логічних сигналів у вихідний логічний сигнал.

В результаті багаторічних досліджень виявлено, що в якості кубітів ефективніше використовувати нейтральні атоми внаслідок того, що:

- вони ідентичні і легко переводяться в потрібний стан за допомогою оптичного накачування;
- мають достатній час когерентності стану, що забезпечується можливістю ізолювати атоми від навколишнього середовища;
- безліч атомів може утримуватися разом без взаємодії один з одним, їх координати можуть бути точно виміряні за допомогою методів флуоресценції.

Нещодавно було показано, що метод захоплення атома світлом, краще, ніж магнітним полем. Це пояснюється тим, що внаслідок осциляції електричного поля світлової хвилі в атомі наводиться осцилюючий електричний дипольний момент із середньою затримкою в один період, що легко контролювати. Більш того, за допомогою певних розподілів світла можна управляти як кожним атомом окремо, так і групою атомів. Успішні експерименти в проведенні квантових обчислень з використанням атомів цезію, методи їх розвитку та перспективи наведені в [4].

Підкреслимо, що для успішного управління частками необхідно знати точні координати (x, y, z) топологічних об'єктів у збудженому світловому полі. Круглий отвір є базовим елементом в якості власників елементів будь-яких оптичних приладів і пристосувань, а також важливим дифракційним елементом, дифраговане поле якого становить значний інтерес для управління мікрочастинками. Тому, першочерговим завданням є обчислення точних характеристик

обуреного поля.

Проведені в роботі дослідження основані на принципах та методах комп'ютерного та математичного моделювання, які використовувались при розробленні програмно-алгоритмічного засобу.

Метою даної роботи є дослідження методів моделювання дифракції та інтерференції, методів обробки оптичних зображень, створення інформаційних моделей опису теоретичного та експериментального дослідження дифракції та інтерференції, розроблення алгоритмів реалізації моделей обробки зображення, реалізація та впровадження програмно-алгоритмічного засобу для первинної та додаткової обробки оптичних зображень.

Об'єкт дослідження – процес розробки програмно-алгоритмічного засобу для моделювання дифракції та обробки інтерференційних картин.

Предмет дослідження – методи чисельного інтегрування, Фур'є перетворення та математичні моделі розробки програмно-алгоритмічного засобу для моделювання дифракції та обробки інтерференційних картин.

Практичне значення роботи та отриманих результатів полягає в тому, що основні положення магістерської роботи можуть бути використані для подальшого розвитку інформаційної технології ідентифікації топологічних об'єктів в світловому полі та підтримки прийняття рішень в зворотних задачах дифракції.

Результатом роботи є розроблений програмно-алгоритмічний засіб.

Розроблені програми були використані при вивченні теоретичних даних, моделюванні технічних комплексів оптичної лабораторії і проектуванні оптичних схем.

Особистий внесок здобувача. Усі основні положення, результати, висновки та рекомендації магістерської роботи отримано автором самостійно.

Публікації. За темою магістерської роботи з викладенням її основних результатів опубліковано одну наукову статтю та опубліковано чотири тези доповіді в збірниках наукових праць всеукраїнських та міжнародних конференцій.

Структура та обсяг магістерської роботи. Магістерська робота складається із вступу, чотирьох розділів, висновків, списку використаних джерел і додатків. Загальний обсяг магістерської роботи складає 95 сторінок, з яких основний текст на 61 сторінках, список використаних джерел із 86 найменувань на 5 сторінках, додатки на 29 сторінках. Робота містить 2 таблиці, 2 додатки, 40 рисунків та 20 слайдів презентації.

1. АНАЛІЗ ЗАДАЧІ ДОСЛІДЖЕННЯ

У першому розділі розглянуто явище дифракції та методи математичного моделювання з комп'ютерним реалізуванням фізичного явища. У подальшому полягає робота із розроблення продукту за допомогою пакету MATLAB, що дозволить моделювати дифракцію за більш широким спектром параметрів та обробляти зображення отримані експериментально..

1.1 Аналіз вимог для створення високоякісного комп'ютерного моделювання

Комп'ютерне моделювання є одним з ефективних методів вивчення фізичних систем. Часто комп'ютерні моделі простіше і зручніше досліджувати, вони дозволяють проводити обчислювальні експерименти, реальна постановка яких утруднена або може дати непередбачуваний результат. Логічність і формалізованість комп'ютерних моделей дозволяє виявити основні фактори, що визначають властивості досліджуваних об'єктів, досліджувати відгук фізичної системи на зміни її параметрів і початкових умов.

Комп'ютерна анімація при сучасному рівні апаратного розвитку і програмного забезпечення є ідеальним засобом для візуального моделювання всіляких процесів, в тому числі відбуваються в хвильової оптики. [6]

Комп'ютерне моделювання вимагає абстрагування від конкретної природи явищ, побудови спочатку якісної, а потім і кількісної моделі. Після цього відбувається проведення серії обчислювальних експериментів на комп'ютері, інтерпретація результатів, зіставлення результатів моделювання з поведінкою досліджуваного об'єкта, подальше уточнення моделі і т.п.

До основних етапів комп'ютерного моделювання відносяться: постановка задачі, визначення об'єкта моделювання; розробка концептуальної моделі, виявлення основних елементів системи та елементарних актів взаємодії; формалізація, тобто перехід до математичної моделі; створення алгоритму та написання програми; планування та проведення комп'ютерних експериментів; аналіз і інтерпретація результатів.

Розрізняють аналітичне та імітаційне моделювання. Аналітичними називаються моделі реального об'єкта, що використовують алгебраїчні, диференціальні та інші рівняння, а також передбачають здійснення однозначної обчислювальної процедури, що призводить до їх точного розв'язання. Імітаційними називаються математичні моделі, які відтворюють алгоритм функціонування досліджуваної системи шляхом послідовного виконання великої кількості елементарних операцій.

Моделювання базується на декількох основоположних принципах, а саме:

- Принцип інформаційної достатності. При повній відсутності інформації про об'єкт побудувати модель неможливо. При наявності повної інформації моделювання позбавлене сенсу. Існує рівень інформаційної достатності, при досягненні якого може бути побудована модель системи.
- Принцип здійсненності. Створювана модель повинна забезпечувати досягнення поставленої мети дослідження за кінцевий час.
- Принцип множинності моделей. Будь-яка конкретна модель відображає лише деякі сторони реальної системи. Для повного дослідження необхідно побудувати ряд моделей досліджуваного процесу, причому кожна наступна модель повинна уточнювати попередню.
- Принцип системності. Досліджувана система подана в вигляді сукупності підсистем, що взаємодіють між собою, які моделюються стандартними математичними методами. При цьому властивості системи не є сумою властивостей її елементів.
- Принцип параметризації. Деякі підсистеми модельованої системи можуть бути охарактеризовані єдиним параметром: вектором, матрицею, графіком, формулою.

Також модель має відповідати певним вимогам, саме це адекватність, точність, простота, відкритість, сумісність, економічність, надійність.

Адекватність моделі можна визначити як відповідність моделі модельованого об'єкту або процесу. Модель вважається адекватною, якщо вона відображає задані властивості об'єкта з прийнятною точністю. Точність визначається як ступінь збігу вихідних параметрів реального об'єкта або результатів експериментальних вимірювань з результатами моделювання. Адекватність моделі можна визначити так само, як здатність відображати задані властивості об'єкта проектування з похибкою не вище заданої.

Наявність похибки комп'ютерного моделювання обумовлено рядом причин, основними джерелами є:

- Математична модель є лише наближеним описом реального процесу (похибка моделі).
- Вихідні дані, як правило, містять похибки, так як-небудь є результатами експериментів (вимірів), або рішеннями допоміжних завдань (похибка даних).
- Застосовувані для вирішення завдання методи в більшості випадків є наближеними (похибка методу).
- При введенні вихідних даних в ЕОМ, виконанні операцій виробляються округлення (обчислювальна похибка).

Важливими вимогами для комп'ютерних моделей є відкритість архітектури і сумісність з іншими моделями аналогічного призначення з точки зору забезпечення можливості обміну

даними і результатами моделювання. Відкрита архітектура комп'ютерної моделі передбачає можливість подальшого розвитку та адаптації моделі користувачем для своїх конкретних умов і вимог, підключення до основної моделі модулів власної розробки, які вирішують більш вузькі або нові завдання, що виникли в процесі подальшого розвитку технологічних систем.

Сумісність форматів вхідних та вихідних даних комп'ютерних моделей аналогічного або пов'язаного призначення дозволяє використовувати для моделювання результати робіт, виконаних за допомогою інших моделей, спростити структуру моделі і здешевити вартість розробки [13].

Економічність моделі зазвичай характеризують витратами обчислювальних ресурсів ЕОМ при їх реалізації, надійність - забезпеченням з плином часу необхідної точності і повторюваності результатів моделювання.

Виконати всі перераховані вимоги до моделі практично не представляється можливим, тому що вони найчастіше суперечливі, наприклад, висока точність моделі і її гранична простота. Тому розробка комп'ютерних моделей здійснюється на основі компромісних рішень, виходячи з поставлених пріоритетів.

1.2 Поняття моделі. Цілі моделювання

У процесі вивчення навколишнього світу суб'єкту пізнання протистоїть досліджувана частина об'єктивної реальності - об'єкт пізнання. Вчений, використовуючи емпіричні методи пізнання (спостереження і експеримент), встановлює факти, що характеризують об'єкт. Елементарні факти узагальнюються і формулюють емпіричні закони. Наступний крок полягає в розвитку теорії та побудові теоретичної моделі, що пояснює поведінку об'єкта і враховує найважливіші чинники, що впливають на досліджуване явище. Ця теоретична модель повинна бути логічною і відповідати встановленим фактам. Можна вважати, що будь-яка наука являє собою теоретичну модель певної частини навколишньої дійсності.

Часто в процесі пізнання реальний об'єкт замінюється деяким іншим ідеальним, уявним або матеріальним об'єктом, що несе досліджувані риси досліджуваного об'єкта, і званим моделлю. Ця модель піддається дослідженню: на неї чинять різний вплив, змінюють параметри і початкові умови, і з'ясовують, як змінюється її поведінка. Результати дослідження моделі переносять на об'єкт дослідження, зіставляють з наявними емпіричними даними і т.д.

Таким чином, модель - це матеріальний або ідеальний об'єкт, який замінює досліджувану систему і адекватним чином відображає її суттєві сторони. Модель повинна в чомусь повторювати досліджуваний процес або об'єкт зі ступенем відповідності, що дозволяє вивчити об'єкт-оригінал. Щоб результати моделювання можна було б перенести на досліджуваний об'єкт,

модель повинна мати властивість адекватності. Перевага підміни досліджуваного об'єкта його моделлю в тому, що часто моделі простіше, дешевше і безпечніше досліджувати. Дійсно, щоб створити літак, слід побудувати теоретичну модель, намалювати креслення, виконати відповідні розрахунки, виготовити його зменшену копію, досліджувати її в аеродинамічній трубі і т.д.

Модель об'єкта повинна відображати його найбільш важливі якості, нехтуючи другорядними [18].

1.2.1 Види моделей

Використовувані моделі надзвичайно різноманітні. Системний аналіз вимагає класифікації та систематизації, тобто структурування спочатку неупорядкованого безлічі об'єктів і перетворення його в систему. Відомі різні способи класифікації існуючого різноманіття моделей. Так, в [25] виділяють наступні види моделей:

- 1) детерміновані і стохастичні;
- 2) статичні і динамічні;
- 3) дискретні, безперервні і дискретно-безперервні;
- 4) уявні і реальні.

В інших роботах [15, 20], моделі класифікують за такими підставами:

- 1) за характером моделюється боку об'єкта;
- 2) по відношенню до часу;
- 3) за способом представлення стану системи;
- 4) за ступенем випадковості процесу, що моделюють;
- 5) за способом реалізації.

Основний інтерес представляють абстрактні моделі, які в свою чергу поділяються на вербальні, математичні і комп'ютерні. До вербальних або текстових моделей відносяться послідовності тверджень на природному або формалізованому мовою, що описують об'єкт пізнання. Математичні моделі утворюють широкий клас знакових моделей, в яких використовуються математичні дії і оператори. Часто вони являють собою систему алгебраїчних або диференціальних рівнянь. Комп'ютерні моделі являють собою алгоритм або комп'ютерну програму, вирішальну систему логічних, алгебраїчних або диференціальних рівнянь і імітує поведінку досліджуваної системи. Аналітичне моделювання передбачає написання системи з алгебри, диференціальних, інтегральних, звичайно-різницевих рівнянь і логічних умов. Для дослідження аналітичної моделі можуть бути використані аналітичний метод і чисельний метод. Останнім часом чисельні методи реалізуються на ЕОМ, тому комп'ютерні моделі можна розглядати як різновид математичних.

1.2.2 Комп'ютерне моделювання

Ефективним способом вивчення явищ навколишньої дійсності є науковий експеримент, що складається у відтворенні досліджуваного явища природи в керованих і контрольованих умовах. Однак часто проведення експерименту неможливо або вимагає занадто великих економічних витрат і може привести до небажаних наслідків. В цьому випадку досліджуваний об'єкт замінюють комп'ютерною моделлю і досліджують її поведінку при різних зовнішніх впливах. Повсюдне поширення персональних комп'ютерів, інформаційних технологій, створення потужних суперЕОМ зробило комп'ютерне моделювання одним з результативних методів вивчення фізичних, технічних, біологічних, економічних і інших систем. Часто комп'ютерні моделі простіше і зручніше досліджувати, вони дозволяють проводити обчислювальні експерименти, реальна постановка яких утруднена або може дати непередбачуваний результат. Логічність і формалізованість комп'ютерних моделей дозволяє виявити основні фактори, що визначають властивості досліджуваних об'єктів, досліджувати відгук фізичної системи на зміни її параметрів і початкових умов.

Комп'ютерне моделювання вимагає абстрагування від конкретної природи явищ, побудови спочатку якісної, а потім і кількісної моделі. Після цього відбувається проведення серії обчислювальних експериментів на комп'ютері, інтерпретація результатів, зіставлення результатів моделювання з поведінкою досліджуваного об'єкта, подальше уточнення моделі і т.д. Обчислювальний експеримент фактично є експериментом над математичною моделлю досліджуваного об'єкта, що проводиться за допомогою ЕОМ [24]. Часто він значно дешевше і доступніше натурного експерименту, його виконання вимагає меншого часу, він дає більш детальну інформацію про величинах, що характеризують стан системи.

Сутність комп'ютерного моделювання системи полягає в створенні комп'ютерної програми (пакета програм), яка описує поведінку елементів досліджуваної системи в процесі її функціонування, що враховує їх взаємодія між собою і зовнішнім середовищем, і проведенні на ЕОМ серії обчислювальних експериментів. Це робиться з метою вивчення природи і поведінки об'єкта, його оптимізації та структурного розвитку, прогнозування нових явищ. Модель виконана згідно вимог приведених у [23].

Розробка комп'ютерної моделі об'єкта є послідовність ітерацій: спочатку на основі наявної інформації про систему S будується модель M_1 , проводиться серія обчислювальних експериментів, результати аналізуються. При отриманні нової інформації про об'єкт S

враховуються додаткові чинники, виходить модель M_2 , поведінка якої теж досліджується на ЕОМ. Після цього створюються моделі M_3 , M_4 і т.д. до тих пір, поки не вийде модель, з необхідною точністю відповідає системі S (рисунок 1).

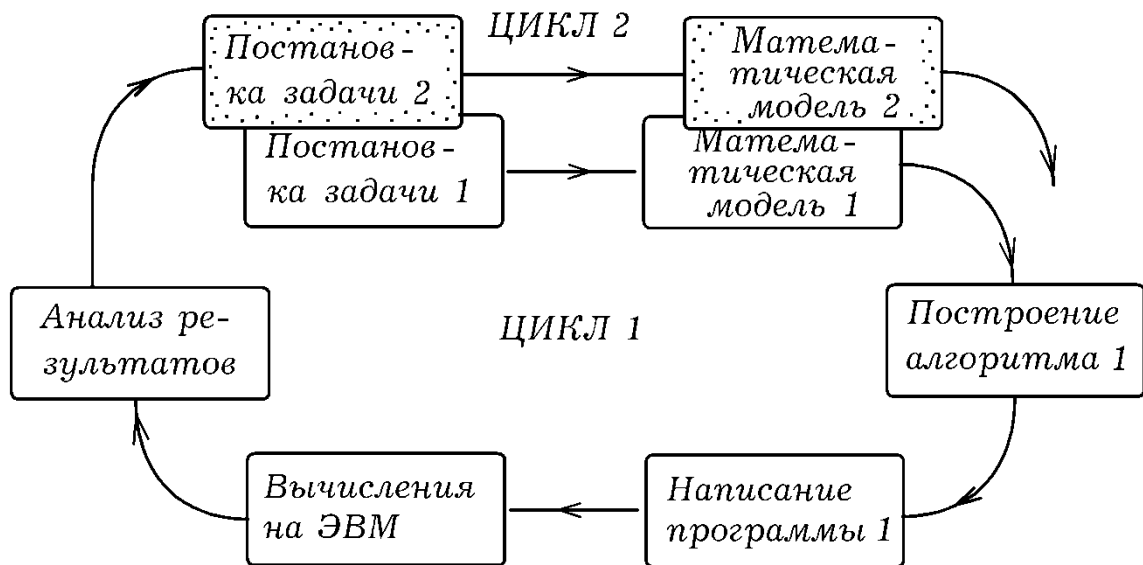


Рисунок 1.1 - Етапи комп'ютерного моделювання

У загальному випадку поведінка досліджуваної системи S описується законом функціонування $Y(t) = F(X, V, H, t)$, де $X = (x_1, x_2, \dots, x_n)$ - вектор вхідних впливів (стимулів), $Y = (y_1, y_2, \dots, y_m)$ - вектор вихідних сигналів (відгуків, реакцій), $V = (V_1, V_2, \dots, V_k)$ - вектор впливів зовнішнього середовища, $H = (h_1, h_2, \dots, h_l)$ - вектор власних параметрів системи [23]. Закон функціонування може мати вигляд словесного правила, таблиці, алгоритму, функції, сукупності логічних умов і т.д. У разі, коли закон функціонування містить час, говорять про динамічних моделях і системах. Наприклад, розгін і гальмування асинхронного двигуна, перехідний процес в ланцюзі, що містить конденсатор, функціонування обчислювальної мережі, системи масового обслуговування. У всіх цих випадках стан системи, а значить і її моделі, змінюється з плином часу.

Якщо поведінка системи описується законом $Y = F(X, V, H)$, що не містить час явно, то мова йде про статичних моделях і системах, вирішенні стаціонарних задач і т.д. Наведемо кілька прикладів: розрахунок нелінійного ланцюга постійного струму, знаходження стаціонарного розподілу температури в стержні при постійних температурах його кінців, форми пружною плівки, натягнутої на каркас, профілю швидкостей в сталому перебігу в'язкої рідини тощо.

Функціонування системи можна розглядати як послідовну зміну станів $q_1(t)$, $q_2(t)$, ...,

$q_r(t)$, яким відповідають деякі точки в багатовимірному фазовому просторі. Множина всіх точок $A = \{a_1, a_2, \dots, a_s\}$, що відповідають всіляким станів системи, називають простором станів об'єкта (або моделі). Кожній реалізації процесу відповідає одна фазова траєкторія, що проходить через деякі точки з множини. Якщо математична модель містить елемент випадковості, то виходить стохастична комп'ютерна модель. В окремому випадку, коли параметри системи і зовнішні впливи однозначно визначають вихідні сигнали, говорять про детерміновану модель.

1.3 Аналіз досліджуваного явища

Дифракція — явище, яке проявляє себе як відхилення від законів геометричної оптики при поширенні хвиль. Вона являє собою універсальне хвильове явище і характеризується одними і тими ж законами при спостереженні хвильових полів різної природи.

Дифракція нерозривно пов'язана з явищем інтерференції. Більш того, саме явище дифракції часто трактують як випадок інтерференції обмежених в просторі хвиль (інтерференція вторинних хвиль). Загальною властивістю всіх ефектів дифракції є залежність ступеня її прояви від співвідношення між довжиною хвилі λ і розміром ширини хвильового фронту d , або непрозорого екрану на шляху його поширення, або неоднорідностей структури самої хвилі.

Оскільки в більшості випадків, що мають практичне значення, це обмеження ширини хвильового фронту має місце завжди, явище дифракції супроводжує будь-який процес поширення хвиль. Так, саме явищем дифракції задається межа роздільної здатності будь-якого оптичного приладу, що створює зображення, який неможливо переступити принципово при заданій ширині спектра випромінювання, що використовується для побудови зображення [9, 14].

Наукові й технічні використання явища дифракції — різноманітні. Дифракційні ґратки служать для розкладу світла в спектр й для створення дзеркал (наприклад, для напівпровідникових лазерів). Дифракція рентгенівських променів, електронів та нейтронів використовується для дослідження структури кристалічних твердих тіл.

Теоретичний аналіз дифрагованих пучків, що містять дислокації хвильового фронту, в роботах попередників, як правило, базувався на розподілах фази та амплітуди поля в поперечному перетині пучка. Але питання щодо повного опису поля вздовж усього напрямку розповсюдження дифрагованого пучка є невивченим, що забезпечує актуальність досліджень топологічних об'єктів на всьому просторі як з наукової, так і практичної точки зору.

Скалярна теорія дифракції.

Уявімо світлове поле в площині джерела (рисунок 1.2) як безліч променевих трубок, інтенсивність випромінювання в кожній з яких пропорційна потоку енергії уздовж трубки. Дифракція в першому наближенні є ефект поперечної дифузії променевої амплітуди з одних трубок в суміжні, або, іншими словами, дифузію променевої амплітуди по фронтах

поширюються хвиль. Фронти хвиль можна вважати плоскими в параксіальній наближенні

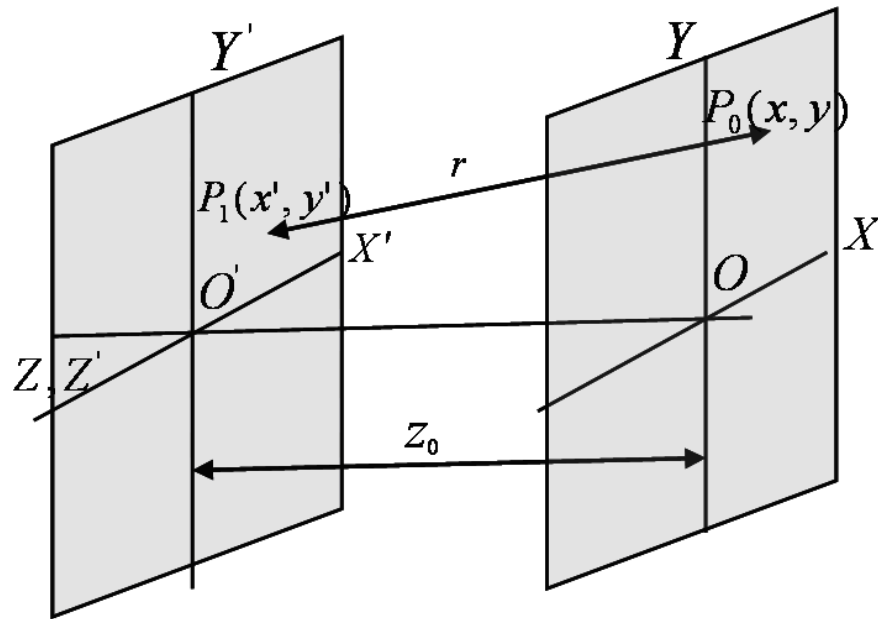


Рисунок 1.2 - Розташування систем координат в площинах джерела і спостереження дифракційної картини на відстані

Розглянемо інтегральну теорему Гемгольца-Кірхгофа і є основою скалярної теорії дифракції, яка розглядає пучок з лінійною поляризацією.

Для початкової плоскої хвилі $A = ae^{-ikr_1}$ розподіл поля в т.Р₀ має вигляд:

$$A_{P_0} = \frac{1}{4\pi} \int \left\{ \left(\frac{e^{ikr}}{r} \cos(n, r_1) \frac{\partial}{\partial r_1} [ae^{-ikr_1}] \right) - \left(ae^{-ikr_1} \cos(n, r) \frac{\partial}{\partial r} \left[\frac{e^{-ikr}}{r} \right] \right) \right\} dS \quad (1.1)$$

Для того, щоб інтегральне рівняння (1.1) використовувати як формулу для обчислення за відомими значеннями і в точках плоского екрану, Кірхгоф запропонував наступні правила:

1. На отворах амплітуда та її похідна мають такі ж значення, які вони мали б при відсутності непрозорих частин екрану.
2. На непрозорих частинах екрану амплітуда та її похідна дорівнюють нулю.

Оптичне наближення. При розгляді оптичних завдань для видимого діапазону виконується умова $1/r_0 \ll k$. Врахування якої при диференціюванні другого члена в (1.1) приводить до наступного наближення:

$$\frac{\partial}{\partial n} \left(\frac{e^{ikr}}{r} \right) = \cos(\vec{n}, \vec{r}) \frac{\partial}{\partial r} \left(\frac{e^{ikr}}{r} \right) = \cos(\vec{n}, \vec{r}) \left(ik - \frac{1}{r} \right) \frac{e^{ikr}}{r} \approx ik \cos(\vec{n}, \vec{r}) \frac{e^{ikr}}{r}.$$

Тоді остаточний вираз для дифракції плоскої хвилі набуде вигляду

$$A_p = \frac{-1}{4\pi} ik \int \left\{ \frac{a}{r} e^{-ikr} [\cos(n, r) + \cos(n, r_1)] \right\} dS. \quad (1.2)$$

Параксіальне наближення використовується при описі оптичних полів, для яких виконуються наступні умови:

1. Незначні зміни амплітуди вздовж пучка в порівнянні з градієнтом амплітуд в поперечному напрямку

$$\frac{\partial A}{\partial z} \ll \frac{\partial A}{\partial r} \quad (1.3)$$

2. Поперечна компонента нормалі до фазової поверхні набагато менше, ніж її поздовжня складова

$$n_{\perp} \ll n_{\parallel}. \quad (1.4)$$

Припустимо, що розглядаються світлові пучки які мають вищевказані властивості, і побудуємо математичну модель еволюції дифрагованого пучка в параксіальному наближенні. Якісний аналіз її рішень в різних умовах базується на рівнянні Шредінгера (параболічне рівняння Леонтовича).

1.4 Постановка завдання

В оптичній лабораторії проводяться два типи досліджень: теоретичні та експериментальні. Кожен тип має свою точність розрахунку та вимірювання. При виконанні теоретичного дослідження існує похибка при розробці математичної моделі, що описує явище дифракції та інтерференції; наступна похибка виникає при проведенні чисельного моделювання та визначається обраним методом. На проведення експериментального дослідження впливає багато факторів: схема експерименту, джерело світла, оптичні елементи, просторовий модулятор світла, камера для реєстрації зображення, кріплення для кожного елемента та пристрою, експериментатор, що юстує оптичну систему, тому величина похибки буде значно більшою. Вона складається з похибки кожного приладу, похибки обробки оптичного елемента, похибки що мають системи кріплення до кожного елемента, похибки юстування системи, стійкість оптичного зображення, що зумовлена стабільністю лазерного випромінювання, врахування впливу середовища (повітряні потоки, сторонні вібрації).

Наступним кроком в роботі оптичної лабораторії є проведення порівняльного аналізу отриманих експериментальних результатів з теоретичними розрахунками. Однією з основних проблем при проведенні кореляції даних є обмеженість експериментальних можливостей для будування аналогічної розрахункової системи. Обмеженість зумовлена певними геометричними розмірами кожного пристрою та кріпленням до нього, які неможливо передбачити в теоретичних розрахунках, тому існує потреба в розробці швидкого, точного та надійного ПАЗ, який дозволить одночасно з проведенням експерименту проводити перерахунок теоретичної моделі.

1.5 Технічне завдання

Метою даної роботи є розробка програмно-алгоритмічного засобу для дослідження процесів дифракції та обробки експериментальних зображень.

Основні задачі магістерської роботи:

- аналіз існуючих методів обробки експериментальних зображень;
- побудова оптимальної високоточної математичної моделі явища дифракції;
- реалізація програмно-алгоритмічного засобу, який дозволить визначити координати оптичних вихорів та їх морфологію

Об'єкт дослідження – процес розробки програмно-алгоритмічного засобу для моделювання дифракції та обробки інтерференційних картин.

Предмет дослідження – методи чисельного інтегрування, Фур'є перетворення та математичні моделі розробки програмно-алгоритмічного засобу для моделювання дифракції та

обробки інтерференційних картин.

Наукова новизна. Розроблено програмно-алгоритмічний засіб для обробки оптичних зображень, завдяки якому визначені координати оптичних вихорів та їх морфологія з великою точністю, що дозволяє підвищити метрологічну спроможність інтерференційних методів та мікроскопії.

1.6 Висновки до першого розділу

У першому розділі розглянуто моделі, види моделей та цілі моделювання, викладено аналіз вимог для створення високоякісного комп'ютерного моделювання, проведено аналіз явища дифракції та його математичної моделі. проблема створення високонадійного програмного забезпечення керуючих контролерів Основні задачі магістерської роботи:

- аналіз існуючих методів обробки експериментальних зображень;
- побудова оптимальної високоточної математичної моделі явища дифракції;
- реалізація програмно-алгоритмічного засобу, який дозволить визначити координати оптичних вихорів та їх морфологію

Проаналізовано існуючі математичні методи та методи комп'ютерного моделювання явища дифракції та сформульована постановка задачі.

2 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ МОДЕЛЮВАННЯ ЯВИЩА ДИФРАКЦІЇ ТА ОБРОБКИ ОПТИЧНИХ ЗОБРАЖЕНЬ

Даний розділ присвячується дослідженню засобів моделювання явища дифракції та методів обробки експериментальних оптичних зображень. В розділі розглядаються різні способи здійснення чисельного інтегрування дифракції, методи обробки зображень з подальшим вибором та обґрунтуванням оптимального рішення.

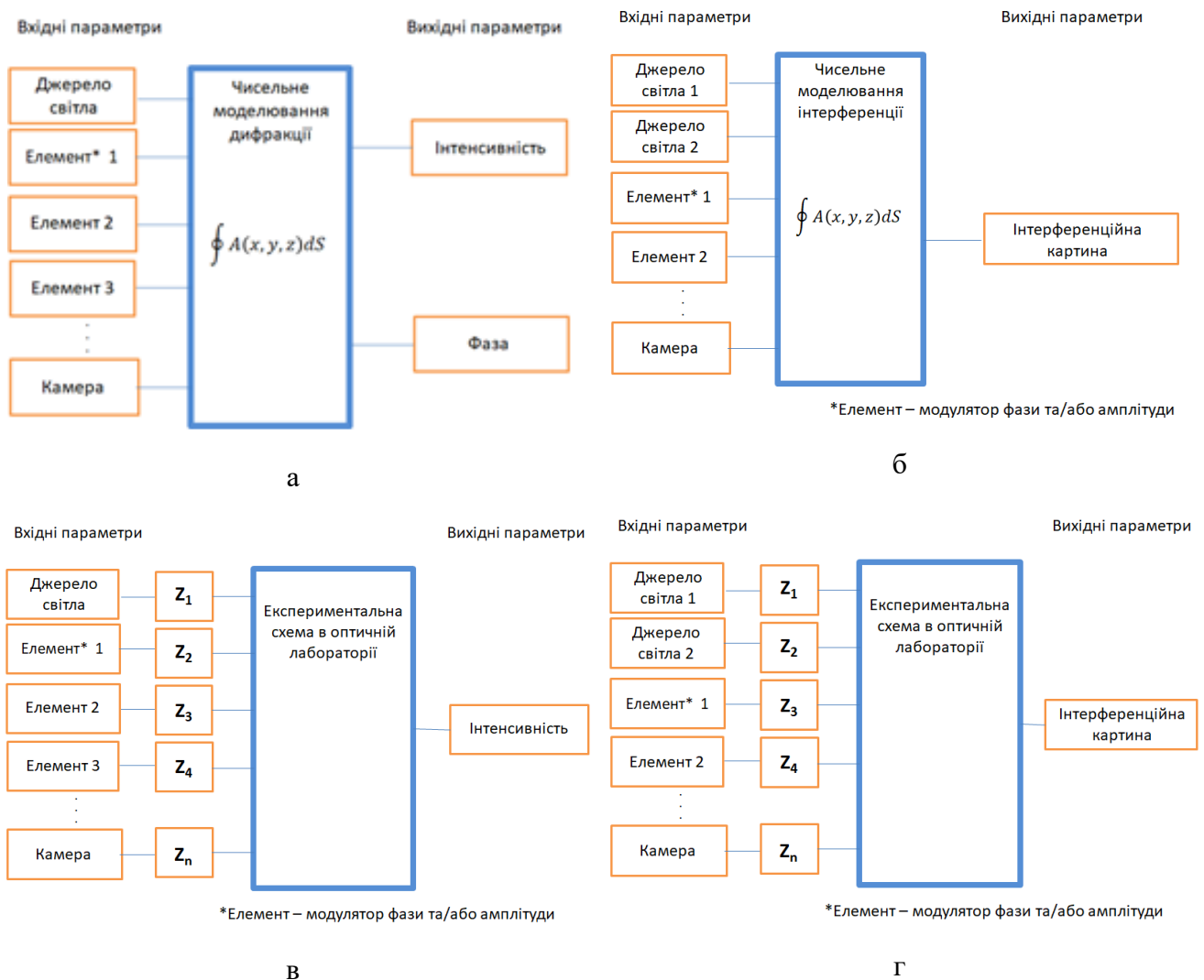


Рисунок 2.1 - Інформаційна модель опису теоретичного та експериментального дослідження дифракції (а,в) та інтерференції (б,г)

2.1 Чисельне інтегрування дифракції

В багатьох задачах, що пов'язані з аналізом, ідентифікацією, оцінювання якості різних засобів і систем автоматики та управління, виникає необхідність обчислення певних інтегралів.

Якщо функція $f(x)$ неперервна на відрізку $[a, b]$ й відома її первісна $F(x)$, то визначений інтеграл від a до b може бути обчислений за формулою Ньютона – Лейбніца

$$I = \int_a^b f(x) dx = F(b) - F(a),$$

де $F'(x) = f(x)$

Графічно інтеграл визначається площею, що обмежується графіком функції $y = f(x)$. Але часто точно обчислити інтеграл важко через велику складність аналітичних перетворень, а інколи це взагалі неможливо (в випадках невластних інтегралів), чи коли підінтегральна функція задана набором числових даних, наприклад, отриманих з експерименту. Задача чисельного інтегрування (numerical integration) функції полягає в обчисленні значення визначеного інтеграла на основі ряду значень підінтегральної функції. Формули чисельного інтегрування часто називають квадратурними.

Найбільш відомими методами знаходження визначених інтегралів є:

- формули прямокутників;
- методи Ньютона – Котеса, Гаусса, Чебишова, які основані на використанні так званих квадратурних формул, отриманих заміною $f(x)$ інтерполяційними багаточленами;
- методи Монте – Карло, основані на використанні статистичних моделей.

2.1.1 Формули прямокутників

Ідея методу полягає в розбитті відрізка інтегрування на дрібні частини $[x_{i-1}, x_i]$ і у побудові прямокутників, які спираються на відрізки $[x_{i-1}, x_i]$ – й мають висоту $f(\xi_i)$. Якщо розбиття відрізка рівномірне, то $x_i = a + i \cdot h$, де h – крок:

$$h = \frac{(a-b)}{n}, \quad i = \overline{0, n}.$$

Інтеграл вважається приблизно рівним сумі площ побудованих прямокутників. Узагальнена квадратурна формула прямокутників має вигляд:

$$I = \int_a^b f(x) dx \approx \sum_{i=1}^n f(\xi_i)(x_i - x_{i-1}),$$

де точка $\xi_i \in (x_i, x_{i-1})$.

В залежності від вибору ξ_i розрізняють формули лівих, правих й середніх прямокутників.

Нехай $\xi_i = x_{i-1}$ формула лівих прямокутників, має вигляд для нерівновіддалених та рівновіддалених вузлів відповідно:

$$I \approx \sum_{i=1}^n f(x_{i-1})(x_i - x_{i-1})$$

$$I \approx h \sum_{i=1}^n f(x_{i-1})$$

Порядок точності формули – перший, $O(h)$. Геометрична інтерпретація наведена на рисунку 2.2

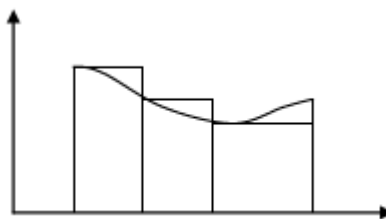


Рисунок 2.2 - Метод лівих прямокутників

Нехай $\xi_i = x_i$ формула правих прямокутників, має вигляд для нерівновіддалених та рівновіддалених вузлів відповідно.

$$I \approx \sum_{i=1}^n f(x_i)(x_i - x_{i-1})$$

$$I \approx h \sum_{i=1}^n f(x_i)$$

Порядок точності формули – перший, $O(h)$. Геометрична інтерпретація наведена на рисунку 2.3.

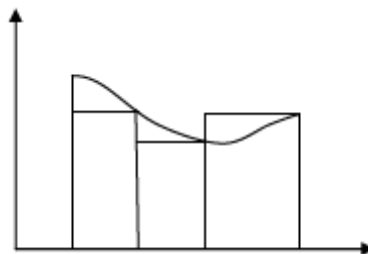


Рисунок 2.3 - Метод правих прямокутників

Нехай $\xi_i = (x_{i-1} + x_i)/2$ формула середніх прямокутників, має вигляд для нерівновіддалених та рівновіддалених вузлів:

$$I \approx \sum_{i=1}^n f\left(\frac{x_i + x_{i-1}}{2}\right)(x_i - x_{i-1})$$

$$I \approx h \sum_{i=1}^n f\left(x_{i-1} + \frac{h}{2}\right)$$

Порядок точності формул – другий, $O(h^2)$. Геометрична інтерпретація (рисунок 2.4):

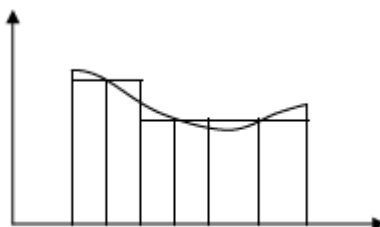


Рисунок 2.4 - Метод середніх прямокутників

Формули лівих та правих прямокутників можуть бути використані як для аналітично заданих функцій, так і для функцій, заданих таблично. Метод середніх прямокутників може використовуватись для пошуку інтегралів тільки від аналітично заданих функцій.

2.1.2 Формули Ньютона-Котеса

Для виведення формул Ньютона – Котеса інтеграл зображується у вигляді:

$$\int_a^b f(x) dx = \sum_{i=0}^n A_i f(x_i) + \Delta, \quad (2.1)$$

де x_i – вузли інтерполяції; A_i – коефіцієнти, які залежать від вигляду формули; Δ – похибка квадратурної формули.

Заміняючи в (2.1) підінтегральну функцію відповідним інтерполяційним поліномом Лагранжа для n рівновіддалених вузлів з кроком $h=(b-a)/n$, можна одержати таку формулу для розрахунку коефіцієнтів A_i при довільній кількості вузлів.

$$A_i = \frac{b-a}{n} \frac{(-1)^{n-1}}{i!(n-1)!} \int_0^n \frac{q(q-1)\dots(q-n)}{(q-i)} dq, \quad (2.2)$$

де $q = (x - a) / h$ – приведена змінна.

Звичайно коефіцієнти $H_i = A_i / (b - a)$ називають коефіцієнтами Котеса.

При цьому формула (2.1) набуде вигляду:

$$\int_a^b f(x)dx = (b-a) \sum_{i=0}^n H_i f(x_i), \quad (2.3)$$

і має такі властивості:

$$\sum_{i=0}^n H_i = 1 \quad \text{і} \quad H_i = H_{n-i}.$$

При $n=1$ і $n=2$ із (2.2) та (2.3) отримаємо формули трапецій і Сімпсона:

$$\int_a^b f(x)dx = h \left(\frac{y_0 + y_n}{2} + y_1 + y_2 + \dots + y_{n-1} \right),$$

$$\int_a^b f(x)dx = \frac{h}{3} (y_0 + y_n + 2(y_2 + y_4 + \dots + y_{n-2}) + 4(y_1 + y_3 + \dots + y_{n-1})).$$

В таблиці 1 наведені значення коефіцієнтів для $n = 1, 2, \dots, 8$. Похибки формул трапецій і Сімпсона визначаються, відповідно, із виразів:

$$\Delta = -\frac{h^3}{12} M_2 \quad \text{та} \quad \Delta = -\frac{h^5}{90} M_4,$$

де M_2 і M_4 - максимальні значення другої та четвертої похідних $f(x)$ при $x \in (a, b)$.

Таблиця 2.1 – Коефіцієнти Котеса

	H_0	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8
1	1	1							
2	1	4	1						
3	1	3	3	1					
4	7	32	12	32	7				
5	19	75	50	50	75	19			
6	41	216	27	272	27	216	41		
7	751	3577	1223	2989	2989	1223	3577	751	
8	989	5888	-928	10496	-4540	10496	-928	5888	989

Складові формули Ньютона – Котеса отримаємо шляхом комбінації простих формул. Наприклад, для формул трапецій і Сімпсона (для парних n):

$$I = \frac{h}{2}[f(x_0) + 2f(x_1) + 2f(x_2) + \dots + f(x_n)],$$

$$I = \frac{h}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \dots + f(x_n)].$$

Причому похибки складових формул будуть, відповідно:

$$\Delta = -n \frac{h^3}{12} M_2 \quad \text{та} \quad \Delta = -n \frac{h^5}{180} M_4.$$

Аналогічно можна отримати складові формули Ньютона – Котеса більш високих порядків. Для оцінювання похибки на практиці можна користуватись методом Рунге (екстраполяції Річардсона), аналогічно тому, як це робиться для однокрокових методів розв'язання задачі Коши.

2.1.3 Оцінювання похибки при чисельному інтегруванні

В випадку, коли підінтегральна функція задана аналітично, може бути поставлена задача про знаходження інтеграла с наперед заданою точністю. Оскільки точність розглянутих вище квадратурних формул залежить від кроку h , то точність результату можна підвищити, зменшуючи крок. Наприклад, можна ділити крок пополам. В цьому випадку для оцінювання наближення до точного значення інтеграла використовують вже згадуваний метод Рунге:

$$I - I_{h/2} \approx \frac{I_{h/2} - I_h}{2^p - 1},$$

де I – точне значення інтеграла; I_h – наближене значення, отримане з кроком h ;

$I_{h/2}$ – наближене значення, отримане з кроком $h/2$; p – порядок точності квадратурної формули

Тому для знаходження інтеграла з заданою точністю ε потрібно обрати квадратурну формулу й початковий крок $h=h_0$. Далі розрахувати інтеграл з кроком $h/2$. Зменшувати крок навпіл необхідно до тих пір, поки не виконається умова:

$$|I_{h/2} - I_h| < \varepsilon.$$

Коли ця умова виконана, інтеграл приблизно дорівнює $I_{h/2}$.

2.2 Методи обробки зображень

Методи обробки зображень (image processing) мають надзвичайно важливе значення в сучасній науці, вони є одними з таких, які постійно розвиваються та вдосконалюються. При цьому під обробкою зображень розуміють не лише поліпшення зорового сприйняття зображень, але й класифікацію об'єктів, що виконується при аналізі зображень.

Сфери застосування методів цифрової обробки в наш час значно розширюються, витісняючи аналогові методи обробки сигналів зображень. Методи цифрової обробки широко застосовуються в промисловості, мистецтві, медицині, космосі. Вони застосовуються при керуванні процесами, автоматизації виявлення об'єктів, розпізнаванні образів і в багатьох інших. Цифрова передача зображень із космічних апаратів, цифрові канали передачі сигналів зображень вимагають забезпечення передачі все більших потоків інформації. Формування зображень, поліпшення якості та автоматизація обробки медичних зображень, охоплюючи зображення, що створюються електронними мікроскопами, рентгенівськими апаратами, томографами тощо, є предметом сучасних досліджень та розробок. Автоматичний аналіз у системах дистанційного спостереження широко застосовується при аналізі місцевості, у лісовому господарстві, наприклад, для автоматичного підрахунку площі вирубок, у сільському господарстві для спостереження за дозріванням урожаю, у розвідці, у системах протипожежної безпеки. Контроль якості виробленої продукції виконується завдяки автоматичним методам аналізу сцен. Сьогодні важко уявити сферу діяльності, у якій можна обійтися без комп'ютерної обробки зображень. При комп'ютерній обробці зображень вирішується широке коло таких завдань, як поліпшення якості зображень; вимірювання параметрів зображення; спектральний аналіз багатовимірних сигналів; розпізнавання зображень; стиск зображень.

2.2.1 Математичні моделі зображень

Комп'ютерна обробка зображень можлива після перетворення сигналу зображення з неперервної форми в цифрову. Ефективність обробки залежить від адекватності моделі, що описує зображення, необхідної для розробки алгоритмів обробки. Модель зображення являє собою систему функцій, що описують істотні характеристики зображення: функцію яскравості, що відбиває зміну яскравості в площині зображення, просторові спектри й спектральні інтенсивності зображень, функції автокореляції. Канал зображення містить оптичну систему, оптико-електричний перетворювач, пристрій аналого-цифрового перетворення (АЦП) і цифрової обробки сигналів зображення. При обробці зображень широко використовується аналіз спектрів зображень. Спектр зображення (image spectrum) одержують прямим двовимірним перетворенням

Фур'є функції, що описує зображення:

$$F(\omega_x, \omega_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp(-i(\omega_x x + \omega_y y)) dx dy, \quad (2.4)$$

де ω_x, ω_y – просторові частоти; $i = \sqrt{-1}$ – уявна одиниця

Функція $\exp(-i(\omega_x x + \omega_y y)) dx dy$ при фіксованих значеннях просторових частот описує плоску хвилю в площині зображення (x, y) (відповідно до рисунку 2.5)

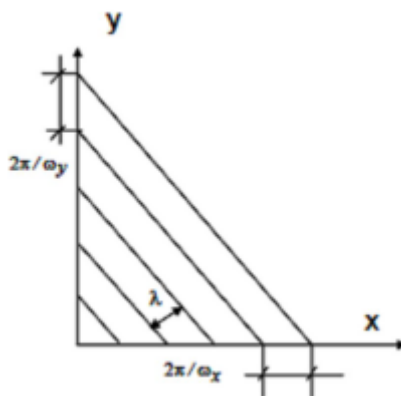


Рисунок 2.5 - Визначення просторових частот зображення

Формула (2.4) пов'язує дійсну функцію, що описує яскравість зображення $f(x, y)$, з комплексною функцією частоти – спектром зображення $F(\omega_x, \omega_y)$:

$$F(\omega_x, \omega_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \cos(\omega_x x + \omega_y y) dx dy + \\ + i \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (-f(x, y)) \sin(\omega_x x + \omega_y y) dx dy = \text{Re}(\omega_x, \omega_y) + i \text{Im}(\omega_x, \omega_y) \quad (2.5)$$

де $\text{Re}(\omega_x, \omega_y)$ – дійсна частина спектра, $\text{Im}(\omega_x, \omega_y)$ – реальна частина спектра.

Амплітуда і фаза спектра визначаються за формулами (2.6) і (2.7) відповідно:

$$F(\omega_x, \omega_y) = \sqrt{\text{Re}(\psi_x, \psi_y)^2 + \text{Im}(\psi_x, \psi_y)^2}, \\ \varphi(\psi_x, \psi_y) = \text{arctg}(\text{Im}(\psi_x, \psi_y) / \text{Re}(\psi_x, \psi_y)) \quad (2.6)$$

Із формули (2.6) виводиться наступне:

$$F(\omega_x, \omega_y) = F(\psi_x, \psi_y) \exp(i\varphi(\omega_x, \omega_y)) \quad (2.7)$$

Обернене перетворення Фур'є дозволяє відновити зображення за його спектром:

$$f(x, y) = (1/4\pi^2) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\psi_x, \psi_y) \exp(i\varphi(\omega_x, \omega_y)) \quad (2.8)$$

Спектральна інтенсивність (spectral intensity) зображення характеризує розподіл енергії по просторових частотах. Вона визначається як квадрат модуля спектра зображення:

$$S(\omega_x, \omega_y) = \text{Re}(\omega_x, \omega_y)^2 + i \text{Im}(\omega_x, \omega_y)^2 = F^2(\omega_x, \omega_y) \quad (2.9)$$

Для її назви використовуються терміни спектральна щільність (spectral density) і енергетичний спектр (energy spectrum). Енергія зображення визначається як інтеграл енергетичного спектра за просторовими частотами. Відповідно до теореми Парсеваля енергія зображення може бути обчислена відповідно до (2.10):

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f^2(x, y) dx dy = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |F(\omega_x, \omega_y)|^2 d\omega_x d\omega_y \quad (2.10)$$

Імовірнісні моделі зображень (probabilistic image models) широко використовуються для опису зображень. Зображення в цьому випадку розглядається як випадкова функція просторових координат (x, y) і часу t . Випадковий процес називається стаціонарним у широкому сенсі, якщо він має постійні значення математичного очікування й дисперсії, а його автокореляційна функція залежить не від координат, а від їх різниці (зсуву). Випадковий процес називається стаціонарним у вузькому сенсі, якщо його n -вимірна щільність розподілу ймовірностей інваріантна до зсуву. Випадковий процес описується щільністю розподілу ймовірності яскравості в зображенні по просторових координатах для деякого фіксованого моменту часу t $p(x, y)$.

У відповідності з визначенням математичне сподівання (середнє значення, mathematical expectation) стаціонарного процесу в широкому розумінні.

$$Mf = \xi = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) p(x, y) dx dy = const \quad (2.11)$$

Дисперсія (dispersion, variance)

$$Df = \sigma^2 = E(f(x, y) - \xi)^2 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (f(x, y) - \xi)^2 p(x, y) dx dy = const \quad (2.12)$$

Функція автокореляції (autocorrelation function) обчислюється таким чином:

$$R(\tau_x, \tau_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) f(x - \tau_x, y - \tau_y) dx dy, \quad (2.13)$$

де τ_x τ_y задають зсуви зображення по відповідних осях координат.

2.2.2 Статистичні методи аналізу зображень

Перші статистичні методи були основані на аналізі взаємного розташування відтінків сірого кольору зображення та частоти їх появи, що характеризувалась двовимірною функцією щільності ймовірності. Для класифікації та розпізнавання текстур за допомогою функції щільності ймовірності було визначено чотири характерних ознаки:

- кутовий момент;
- контрастність;
- кореляція;
- ентропія.

До статистичних методів можна віднести кореляційний аналіз зображень. На основі кореляційного підходу найбільш успішним є метод матриць взаємозв'язку (co-occurrence matrices – GLCM). Дані матриці характеризують частоту пар різних градацій сірого кольору, що присутні в зображенні, і визначаються шляхом кореляційного аналізу пікселів зображення, при цьому якщо піксель відповідає вибраній градації, то він враховується як одиничне значення, якщо ні, то як нульове. У випадку кольорових зображень даний підхід використовують до аналізу кожного з трьох базових кольорів. Як слідує з методу формування матриць взаємозв'язку, вони найкраще підходять до розв'язання задач класифікації текстур.

Наряду зі статистичним аналізом в просторовій області використовують аналогічний аналіз в спектральній області із застосуванням двовимірного дискретного перетворення Фур'є в базисі різних ортогональних функцій – дискретних експоненціальних функцій (ДЕФ),

косинусних, Уолша-Адамара, Хаара та інших. Для аналізу текстур використовують метод гістограм розподілу спектральних коефіцієнтів. Загальною характеристикою такого підходу є те, що гістограми спектральних коефіцієнтів, що отримані в результаті інтегральних перетворень, більш стійкі та надійні, ніж гістограми розподілу окремих пікселів чи груп пікселів. Даний метод також не чутливий до присутнього в зображенні шуму, але це є одночасно і недоліком в разі застосування методу до розв'язання задачі розпізнавання незначних за розміром об'єктів на текстурованому фоні. Також спектральні методи згладжують різкі границі між об'єктами зображення. Статистичний аналіз пікселів та спектральних коефіцієнтів на основі гістограм в основному використовують до класифікації статичних текстур, до аналізу динамічних текстур використовують методи на основі кореляційного аналізу. Найпростішою і найбільш часто вживаною є стохастична модель. Фон зображення характеризують гістограмою розподілу значень кольору за величиною в деякій базовій області, вільній від об'єктів. Гістограму апроксимують функцією щільності розподілу ймовірності, найбільш часто гаусовою. В цьому випадку параметрами моделі є середнє значення m та дисперсія σ . За максимальне відхилення сигналу моделі приймають 2σ . За мінімальне порогове значення величини відхилення можна прийняти величину $\varepsilon_{tr} = 3\sigma$.

Як приклад розглянемо алгоритм розпізнавання зображення об'єктів за допомогою стохастичної моделі:

$$\text{if } |u_{i,j} - m| > \varepsilon_r \text{ then } v_{i,j} = u_{i,j} \text{ else } v_{i,j} = 0 \quad (2.14)$$

де $u_{i,j}$ $v_{i,j}$ – елементи матриць заданого зображення та зображення об'єктів. На рисунок 2.6 наведено результат розпізнавання та відбору об'єктів згідно з алгоритмом (2.14) за статистичним методом.

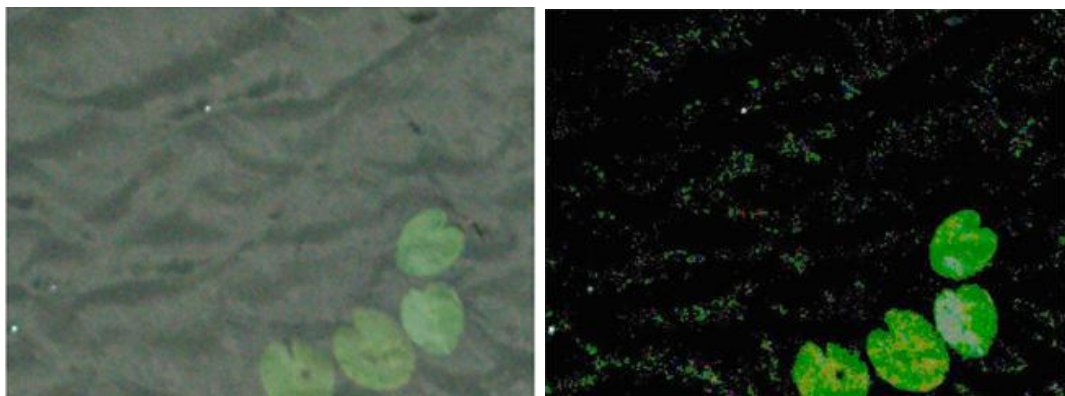


Рисунок 2.6 - Тестове зображення та результат розпізнавання об'єктів за статичним методом

Метод побудови моделі та її застосування викладено нижче:

1. Фрагмент зображення фону, вільний від сторонніх предметів, розміром $n_x \times n_y = 80 \times 80$ пікселів використано як базовий для того, щоб знайти статистичні параметри фону.

2. За допомогою виразу (2.14) створено зображення об'єктів. Як видно з рисунка, всі сторонні об'єкти, як великі (листки), так і малі (білого кольору) виділені, але присутній шум похибки. Зменшити похибку можна збільшивши порогове значення відбору, але, як показали чисельні експерименти, в цьому випадку втрачається якість зображення об'єктів. Таким чином, стохастична модель погано апроксимує фоновий сигнал та неспроможна повністю відділити шукані об'єкти від фону.

2.2.3 Фільтрація зображень

Зазвичай зображення, сформовані різними інформаційними системами, спотворюються дією завад. Це ускладнює як їхній візуальний аналіз, так і автоматичну обробку. При вирішенні деяких завдань обробки зображень у ролі завад можуть виступати ті або інші компоненти самого зображення. Наприклад, при аналізі космічного знімка земної поверхні може стояти завдання визначення границь між її окремими ділянками – лісом і полем, водою й сушею тощо. З погляду цього завдання окремі деталі зображення всередині розділених ділянок є завадою.

Ослаблення дії завад досягається фільтрацією. При фільтрації яскравість (сигнал) кожної точки вихідного зображення, спотвореного завадою, замінюється деяким іншим значенням яскравості, яке меншою мірою було спотворене завадою. Фільтрація зображень здійснюється в просторовій і частотній областях. При просторовій фільтрації зображень перетворення виконується безпосередньо над значеннями відліків зображення. Результатом фільтрації є оцінка корисного сигналу зображення. Це досягається завдяки тому, що зображення часто являє собою двовимірну функцію просторових координат, що змінюється по цих координатах повільніше, ніж завада, що також є двовимірною функцією. Це дозволяє при оцінюванні корисного сигналу в кожній точці зображення взяти до уваги сусідні точки, скориставшись певною подібністю сигналу. В інших випадках, навпаки, ознакою корисного сигналу є різкі перепади яскравості. Однак, як правило, частота цих перепадів відносно невелика, так що на значних проміжках сигнал або постійний, або змінюється повільно. І в цьому випадку властивості сигналу проявляються при спостереженні не тільки його окремої точки, але й при аналізі її околу. Поняття околу є досить умовним. На рисунку 2.7 наведена ієрархія околів відліку, позначеного "0". "1" позначений окіл першого порядку, для якого відстань між елементами дорівнює 1. "2" позначений окіл другого порядку, до якого відносяться діагональні елементи, відстань від яких до центрального відліку "0" дорівнює $\sqrt{2}$. Окіл третього порядку представлений елементами, що

знаходяться від центрального елемента на відстані 2, і так далі.

9	8	7	6	7	8	9
8	5	4	3	4	5	8
7	4	2	1	2	4	7
6	3	1	0	1	3	6
7	4	2	1	2	4	7
8	5	4	3	4	5	8
9	8	7	6	7	8	9

Рисунок 2.7 - Конфігурації околу елемента "0" у кадрі зображення в ієрархічній послідовності.

Відповідно до рисунка 2.7 формується ієрархія конфігурацій околу центрального відліку розглянутого фрагмента кадру за зростанням відстаней від нього до відліку околу. Окіл може бути утворений лише найближчими сусідами, але може містити й досить багато елементів кадру. При розгляді околу великого розміру іноді встановлюється різний ступінь впливу далеких і близьких від центра околу точок на сигнал, формований на виході фільтра в даній точці кадру. Таким чином, ідеологія фільтрації ґрунтується на використанні як даних поточної точки, так і її околу.

Традиційна фільтрація в частотній області вимагає виконання такої послідовності перетворень:

- двовимірне дискретне перетворення зображення із просторової області в частотну (наприклад, за допомогою дискретного перетворення Фур'є),
- перетворення дискретного спектра сигналу зображення,
- обернене двовимірне дискретне перетворення, що дозволяє відновити корисний сигнал зображення в просторовій області.

Завдання полягає в тому, щоб знайти таку обчислювальну процедуру, що забезпечила б одержання найкращих результатів. Використання фільтрів для аналізу текстурованих зображень аналогічно спектральному аналізу але має ряд переваг в тих випадках, коли характерні ознаки спектра зображення відомі. Фільтрація дозволяє більш точно розділити сигнал на складові, що відповідають різним частотним смугам.

2.2.3.1 Оптимальна лінійна фільтрація

Нехай x_{ij} – значення яскравості зображення – корисного сигналу на перетині i -го рядка та j -го стовпця, а зображення, що знаходиться на вході фільтра описується моделлю:

$$y_{i,j} = f(x_{i,j}, n_{i,j}), \quad i = \overline{0, I-1}, \quad j = \overline{0, J-1}, \quad (2.15)$$

де n_{ij} – значення завади в точці з координатами (i, j) , $f(\cdot)$ – функція, що описує взаємодію сигналу і завади, а I та J – відповідно число рядків і стовпців у кадрі.

Надалі будемо дотримуватися прийнятої при цифровій обробці зображень декартової системи координат з початком у лівому верхньому кутку кадру та з додатними напрямками із цієї точки вниз та вправо.

При лінійній фільтрації вихідний ефект визначається лінійною комбінацією вхідних даних:

$$x(i, j) = \sum_{(i_1, j_1) \in S} a(i_1, j_1) \cdot y(i - i_1, j - j_1). \quad (2.16)$$

В цьому рівнянні $x(i, j) = x_{i,j}$ – результат фільтрації корисного сигналу в точці кадру з координатами (i, j) ; S – множина координат точок, що утворюють околицю; $a(i_1, j_1)$ – вагові коефіцієнти, сукупність яких являє собою двовимірну імпульсну характеристику (ІХ). Якщо область кінцева, то імпульсна характеристика має кінцеву довжину й фільтр називається КІХ-фільтром. В іншому випадку імпульсна характеристика має нескінченну довжину, а фільтр назву НІХ-фільтра. У виразі (2.16) прийнято, що ІХ не залежить від координат точки, у якій визначається вихідний ефект. Процедури обробки зображень, що мають властивість незалежності від координат, називаються *однорідними*.

Найбільш поширеним критерієм оптимальності, який використовується для оцінювання якості обробки зображень, є критерій мінімуму середньоквадратичної похибки. Для фільтрації запишемо його так:

$$E \left\{ \left[x(i, j) - \sum_{(i_1, j_1) \in S} a(i_1, j_1) \cdot y(i - i_1, j - j_1) \right]^2 \right\} = \min_{a(\cdot)}, \quad (2.17)$$

де $E\{\circ\}$ – символ математичного сподівання. Згідно з (2.17) пошук оптимального фільтра полягає у визначенні його КХ таким чином, щоб середній квадрат похибки $\varepsilon(i, j) = x(i, j) - x^*(i, j)$, який виражає різницю між сигналом $x(i, j)$ і його оцінкою $x^*(i, j)$, яка формується фільтром, був мінімальним. Математичне сподівання обчислюється за всіма випадковими величинами, які є в (2.17), що свідчить про орієнтацію критерію на врахування середніх похибок.

Розглянемо реалізацію лінійної фільтрації в середовищі Matlab.

Вбудована функція $D=\text{filter2}(h,S)$ виконує фільтрацію даних, заданих у двовимірному масиві S , двовимірним КХ фільтром, коефіцієнти якого зведені в матрицю h , яку також називають маскою фільтра. Як правило, D і S є напівтоновими зображеннями. Результат фільтрації, що повертається в матриці D , обчислюється як двовимірна згортка, яка виконується за допомогою функції `conv2`.

Результат роботи функції $D=\text{filter2}(h,S, \text{shape})$ залежить від значення параметра `shape`. Якщо параметру присвоєно значення «same», то в D повертається центральна частина згортки, розмір якої дорівнює розміру вихідних даних – S . Цей параметр використовується по замовчуванню. Якщо параметру присвоєно значення «full» то повертається весь результат згортки. При цьому $\text{size}(D) > \text{size}(S)$. Якщо параметру присвоєно значення `valid`, повертається лише та частина згортки, при обчисленні якої не використовувались крайові частини вихідних даних, доповнені нулями. При цьому $\text{size}(D) < \text{size}(S)$.

В залежності від складності зображення використовують два або більше двовимірних фільтрів з коротко-імпульсною характеристикою (КХ) або банки вузькосмугових фільтрів виду:

$$s_{i,j} = - \sum_{m,n=0}^{N-1,M-1} h_{m,n} u_{i-m,j-n}, \quad (2.18)$$

де $s_{i,j}$ та $u_{i-m,j-n}$ – відповідно фільтрований сигнал та сигнал зображення, $h_{m,n}$ – імпульсна характеристика розміром $M \times N$. Як критерій оптимальності використовують співвідношення середніх значень сигналів на виході фільтра, що отримані для текстури, яка відповідає характеристиці фільтра, та для текстур, які не відповідають характеристиці фільтра.

Для фільтрації зображень на основі їх спектральних ознак використовують фільтри Габора. Імпульсна характеристика фільтра має вигляд:

$$h_{i,j} = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{i^2}{2\sigma_x^2} - \frac{j^2}{2\sigma_y^2}\right) \exp(2\pi i(f_x i + f_y j)). \quad (2.19)$$

Вона поєднує гаусову функцію та гармоніку, що характеризує частотну характеристику зображення. За допомогою параметрів дисперсії σ^2 в (2.19) регулюють смугу частотного фільтра з парою центральних частот (f_x, f_y) за двома просторовими координатами. Як правило, для характеристики текстурованого зображення використовують набір фільтрів з фіксованими або визначеними для кожної текстури парами частот.

Алгоритм класифікації та розпізнавання зображень охоплює такі етапи:

- 1) фільтрація зображення за допомогою набору фільтрів;
- 2) обчислення потужності на виході кожного з фільтрів;
- 3) згладжування значень потужності;
- 4) нормалізація значень потужності;
- 5) класифікація (розпізнавання) за допомогою шаблонів або співвідношень значень потужності.

Недоліком даного методу є те, що він потребує значного обсягу обчислень в тому разі, коли розмір фільтра $M \times N$ значний і число фільтрів велике.

У практиці цифрової обробки зображень широко використовується маскова фільтрація. Її лінійний різновид є одним з варіантів двовимірної фільтрації з кінцевою імпульсною характеристикою (КІХ) фільтра. Як маска використовується безліч вагових коефіцієнтів, заданих у всіх точках околу S , що зазвичай симетрично оточують поточну точку кадру.

2.2.3.2 Нелінійна фільтрація

У результаті застосування лінійних згладжувальних фільтрів відбувається заглушення шумів, але одночасно розмиваються границі між областями з різною амплітудою сигналу. Для зменшення кількості розмитих границь розроблені різні нелінійні фільтри. Як і лінійні КІХ-фільтри, нелінійні фільтри працюють у ковзному вікні. Різниця між лінійною та нелінійною фільтрацією полягає в тому, що при лінійній фільтрації обчислюється лінійна комбінація відліків сигналу, а при нелінійній фільтрації виконуються нелінійні перетворення відліків сигналу в околі елементів, які зумовлюються маскою фільтра.

Сігма-фільтр

Сігма-фільтр призначений для заглушення шумів у зображенні зі збереженням контурів (різких границь областей). Центральний елемент маски заміщується зваженим середнім значенням, обчисленим лише за тими амплітудами відліків, значення яких попадають в $\pm k\sigma$ – область вибрана згідно з яскравістю центрального елемента. σ вибирається або як

середньоквадратичне відхилення (СКВ) заглушеного шуму, або як СКВ в масці, або встановлюється рівною СКВ, отриманому на всьому зображенні:

$$g(x, y) = \sum_{s, t \in S} h(s, t) f(x - s, y - t), \quad (2.20)$$

де S – окіл становлять ті значення координат маски, у яких виконується накладена умова:

$$S = \{(s, t) : |f(x - s, y - t) - f(x, y)| \leq k\sigma\}, \quad (2.21)$$

$h(s, t)$ - КІХ лінійного згладжуючого фільтра.

При $k = 2$ діапазон значень, які замінюються, становить $\pm 2\sigma$, у випадку нормального розподілу шуму ймовірність потрапляння амплітуди за межі діапазону дорівнює 4,55%.

Медіанний фільтр

Медіанний фільтр (МФ) замінює центральний елемент маски медіаною впорядкованої вибірки, сформованої зі всіх амплітуд відліків, що покриваються маскою фільтра. При застосуванні МФ відбувається послідовна обробка кожної точки кадру, у результаті чого утворюється послідовність оцінок. При медіанній фільтрації використовується ковзне двовимірне вікно. У принципі, для кожного відліку виконується незалежне оцінювання медіани у вікні. З метою прискорення оцінювання доцільно алгоритмічно на кожному кроці використовувати раніше виконані обчислення. Розмір вікна встановлюється непарним і рівним $m \times n$. Відліки зображення, що знаходяться в межах вікна, утворюють робочу вибірку поточного відліку. Якщо впорядкувати послідовність $\{f_i [mn]\}$ за зростанням, то її медіаною буде той елемент вибірки, що займає центральне місце в цій упорядкованій послідовності. Цей елемент є $(mn + 1) / 2$ найбільшим і $(mn + 1) / 2$ найменшим значенням у вибірці й визначає результат медіанної фільтрації для поточної точки кадру. Введемо позначення описаної процедури у вигляді:

$$g_{med} = med(f_1, f_2, \dots, f_n) \quad (2.22)$$

Розглянемо приклад. Припустимо, що впорядкована послідовність Y у вікні розміром 3×3 має вигляд: $Y = \{76, 100, 69, 120, 210, 143, 87, 130, 155\}$, де елемент 210 відповідає центру вікна (x, y) . Велике значення яскравості в цій точці кадру є результатом впливу імпульсної завади. Упорядкована за зростанням вибірка має вигляд: $\{69, 76, 87, 100, 120, 130, 143, 155, 210\}$, отже,

відповідно до розглянутої вище процедури (2.22), на виході медіанного фільтра одержуємо $g_{med} = 120$. Бачимо, що врахування яскравостей елементів околу при фільтрації в поточній точці призвело до заглушення імпульсної завади. Якщо імпульсна завада не є точковою, а займає деяку область, то вона також може бути подавлена, якщо розмір цієї локальної області буде менше, ніж половина розміру апертури МФ. Тому для заглушення імпульсних завад, що вражають локальні ділянки зображення, варто збільшувати розміри апертури МФ.

З (2.12) слідує, що дія МФ полягає в «ігноруванні» як позитивних, так і негативних викидів значень вхідної вибірки. Такий принцип заглушення завад може бути застосований і для ослаблення шуму на зображенні. Однак дослідження заглушення шуму за допомогою медіанної фільтрації показує, що її ефективність при вирішенні цього завдання нижча, ніж лінійної фільтрації. Медіанна фільтрація краще зберігає границі зображення, ніж будь-яка лінійна фільтрація.

Медіанні фільтри заглушають імпульсні шуми. До таких шумів належить шум типу «сіль і перець», відліки якого мають значення, що відповідають максимальному («сіль») і мінімальному («перець») рівням квантування в сигналі зображення. Різкі зміни амплітуди зберігаються медіанним фільтром, а імпульсна завада, розмір якої $\leq mn/2$ таким фільтром заглушається. Однак при збільшенні маски фільтра можна втратити інформацію про малорозмірні області зображення та отримати спотворення границь областей, особливо в кутових положеннях.

Оскільки застосування МФ призводить до заглушення високих частот зображення, викликаючи розмивання країв і текстур, все більший розвиток отримують схеми адаптивної фільтрації, які дозволяють змінити імпульсну характеристику фільтра залежно від локального значення сигналу зображення.

Один з алгоритмів адаптивної медіанної фільтрації (АМФ) виконується таким способом. У вікні фільтрації оцінюються мінімальне значення сигналу f_{min} , максимальне значення f_{max} і медіана f_{med} . Фільтрації піддається тільки той центральний елемент вікна $f(x, y)$, для якого виконується умова 2.23 а): значення медіани більше мінімального й менше максимального значень у вікні й не виконується умова 2.23 б): значення сигналу в центрі вікна більше мінімального й менше максимального значень у вікні.

$$\begin{aligned} \text{а) } A1 &= f_{med} - f_{min}; A2 = f_{med} - f_{max}; A1 > 0, A2 < 0, \\ \text{б) } B1 &= f(x, y) - f_{min}; B2 = f(x, y) - f_{max}; B1 > 0, B2 < 0. \end{aligned} \quad (2.23)$$

Застосування такого фільтра дозволяє видалити біполярну імпульсну заваду, забезпечити згладжування шумів і зменшити заглушення високих частот у зображенні.

На рисунку 2.8 наведено приклад усунення шуму за допомогою медіанного фільтра.

У системі Matlab (Image Processing Toolbox) існує можливість формування й накладання на зображення трьох типів шумів. Для цього використовується вбудована функція `imnoise`, що призначена, в основному, для створення тестових зображень, які використовуються при виборі й дослідженні методів фільтрації шуму. В даному випадку на зображення був накладений імпульсний шум за допомогою команди: `J = imnoise(I,'salt & pepper',0.02);`

Для наочного порівняння приведемо три зображення разом: вихідне, зашумлене та відновлене. Як видно з рисунка 2.7 вихідне та відновлене зображення майже не відрізняються один від одного.



Вихідне, зашумоване та відновлене зображення відповідно

Рисунок 2.8 - Приклад застосування медіанної фільтрації

2.7 Висновки до другого розділу

В цьому розділі розглянуто математичне забезпечення системи моделювання явища дифракції та обробки оптичних зображень, а саме: методи прямокутників та Ньютона-Котеса чисельного моделювання дифракції, та оцінка похибки; методи обробки зображень: лінійна та нелінійна інтерполяції, апроксимація, регресія, згладжування та фільтрація. Проаналізовані різні методи обробки зображень та обраний метод перетворення Фур'є, перевагою якого є не тільки можливість відновити фазу пучку, але й визначити координати оптичних вихорів з високою точністю.

3 РОЗРОБКА ПРОГРАМНО-АЛГОРИТМІЧНОГО ЗАСОБУ ДЛЯ ОБРОБКИ ОПТИЧНИХ ЗОБРАЖЕНЬ

Згідно з технічним завданням даний розділ присвячується розробці програмно-алгоритмічного засобу для первинної та вторинної обробки оптичних зображень. У цьому розділі розроблено та реалізовано алгоритм первинної та вторинної обробки зображення. Наведено розробку програмно-алгоритмічного засобу в математичному пакеті MATLAB. На основі отриманих результатів обробки зроблені висновки про досліджуване явище та сформульовані подальші пропозиції впровадження та удосконалення програмно-аналітичного засобу.

3.1 Призначення розробленого програмно-алгоритмічного засобу

На сьогодні у світі існує тисячі оптичних лабораторій, які перевіряють якість оптичних елементів, розробляють нові елементи та проводять експериментальні дослідження з використанням явищ дифракції, інтерференції та поляризації. Інтерферометричними методами приділяють особливу увагу, завдяки їхнім високим метрологічним параметрам. Отримані інтерференційні картини описують як статичні дефекти оптичних елементів так і динамічні параметри системи. Для збільшення швидкості отримання результатів та підвищення їхньої якості необхідне розроблення програмно-алгоритмічного засобу для обробки зображень.

3.2 Алгоритм реалізації моделі

Розроблено алгоритм реалізації моделі обробки оптичних зображень, який складається з серії вхідних зображень, що підлягають усередненню з наступною фільтрацією від шумів. Задля забезпечення швидкодії процесу усереднене зображення конвертується у монохромне. Для зручності математичної обробки дані нормуються. На цьому етапі початкову обробку зображення закінчено та його можна використовувати для визначення параметрів морфології оптичного вихору, а саме: еліптичності та кута нахилу еліпсу.

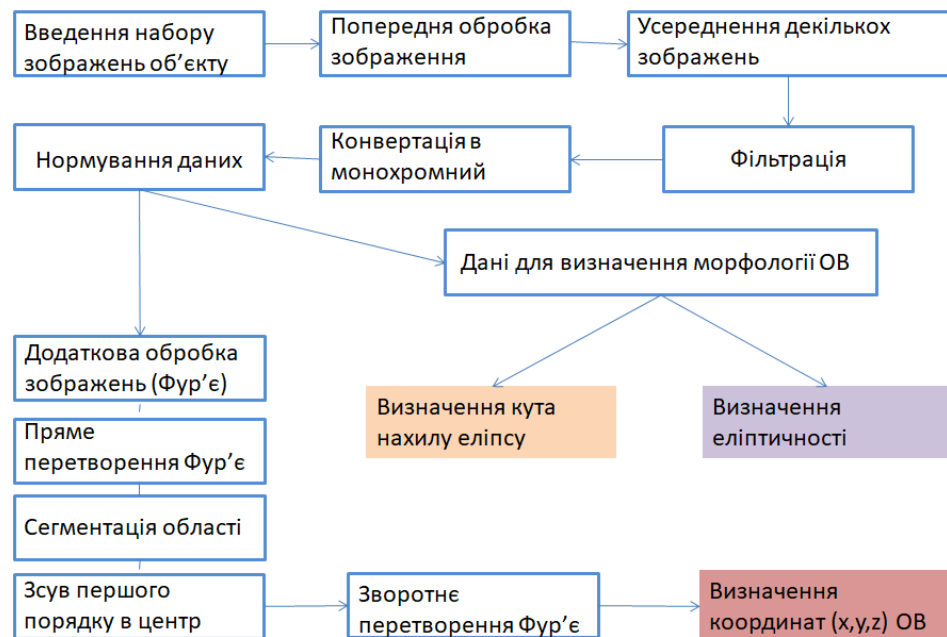
Алгоритм виконання циклу ПАЗ первинної обробки та визначення морфологічних параметрів оптичних вихорів наступний (рисунок 3.1):

- Введення набору зображень об'єкту.
- Усереднення вхідних зображень.
- Фільтрація отриманого зображення.
- Конвертація в монохромне зображення.
- Нормування даних.

- Отримання даних для подальшого визначення морфологічних параметрів.
- Розрахунок величини еліптичності.
- Розрахунок кута нахилу еліпсу.

Алгоритм виконання циклу ПАЗ вторинної обробки зображень та визначення координат оптичних вихорів (рисунок 3.1):

- Введення даних після первинної обробки.
- Пряме перетворення Фур'є.
- Сегментація області.
- Зсув першого порядку в центр картини.
- Зворотнє перетворення Фур'є.
- Визначення координат оптичних вихорів.
- Будування скелетону.



– Рисунок 3.1 - Алгоритм реалізації моделі обробки оптичних зображень.

3.3 Розробка ПАЗ в математичному пакеті MATLAB

В останні роки наукове програмування зазнало серйозної трансформації: розвиваються інтегровані середовища, що ґрунтуються на спеціалізованих алгоритмічних мовах, зростає застосування універсальних математичних систем типу MATLAB, MAPLE, MATHEMATICA, MATCAD та ін. Ці системи мають дружній інтерфейс, потужні графічні засоби, власні мови

програмування, реалізують ряд стандартних і спеціальних математичних операцій. Все це надає широкі можливості для ефективної роботи спеціалістів різних галузей, про що говорить активне застосування математичних пакетів у наукових дослідженнях.

В MATLAB реалізовані класичні чисельні алгоритми розв'язання рівнянь, задач лінійної алгебри, знаходження значень визначених інтегралів, апроксимації, розв'язання окремих рівнянь і систем диференціальних рівнянь. Для застосування більшості можливостей пакета достатньо базових знань з чисельних методів. Розв'язання спеціальних задач, зрозуміло, вимагає відповідної теоретичної підготовки.

Пакет MATLAB є інтерактивною системою з розвинутою довідковою системою, містить велику кількість операторів та функцій, застосування яких дає змогу чисельно розв'язувати складні задачі математики, фізики, хімії, інженерії та фінансів. Крім вбудованих засобів, пакет можна доповнювати власними розробками, використовуючи як власну мову (М-програми), так і зовнішні алгоритмічні мови, наприклад, C#, JAVA. Підсумовуючи, охарактеризуємо спектр задач, які розв'язуються за допомогою пакета MATLAB:

- проведення математичних досліджень, що вимагають обчислень і аналітичних розрахунків;
- розробка і аналіз алгоритмів;
- математичне моделювання, комп'ютерний експеримент;
- аналіз і обробка даних;
- візуалізація, наукова та інженерна графіка;
- розробка графічних і розрахункових додатків.

Розробка ПАЗ проводилась з використанням інтерфейсу GUI для більш зручного керування процесом обробки зображень, що дозволить спростити використання ПАЗ і потребуватиме лиш базового ознайомлення з математичним пакетом MATLAB (рисунок 3.2)

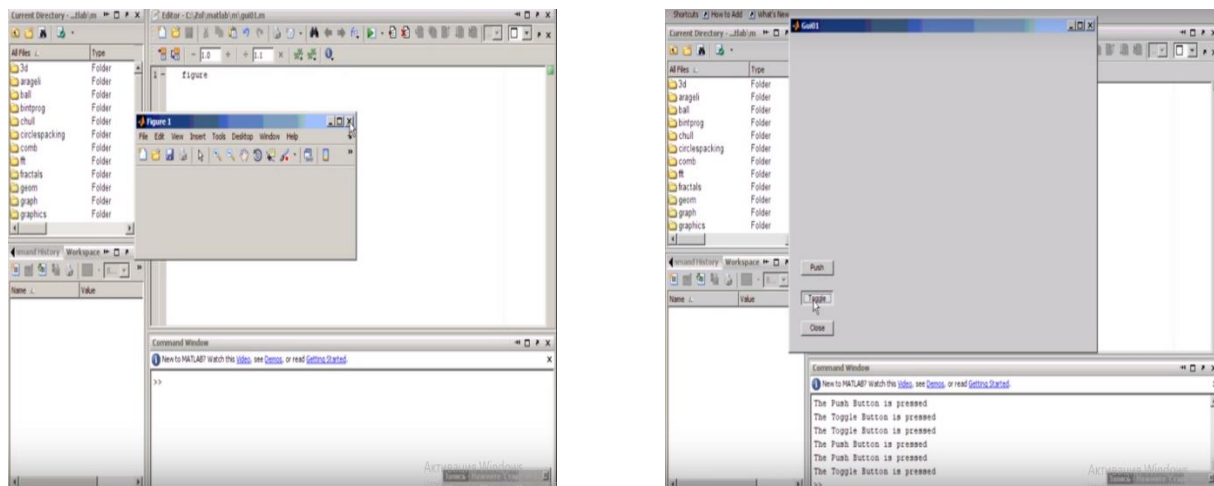


Рисунок 3.2 - Тестові зображення інтерфейсу

3.4 Отримання експериментальних даних та реалізація ПАЗ

Інтерферограма та розподіл інтенсивності реєструються ПЗЗ-камерою (розмір пікселя 3,45 мікрона, роздільна здатність 2200x3000), яка розміщується на різних відстанях від ПМС, $\in [20, 290]$ см. Зображення інтенсивності показує темні плями, розподілені вздовж лінії (рисунок 3.3в). На інтерферограмі (рисунок 3.3г) можна побачити вилки, характерні для вихрової структури, вбудованої в пучок.

Після отримання декількох експериментальних зображень проводиться усереднення з подальшим фільтруванням. Для пришвидшення програмного розрахунку та більш точного знаходження координат кольоровою складовою зображень нехтують та конвертують у монохромне зображення.

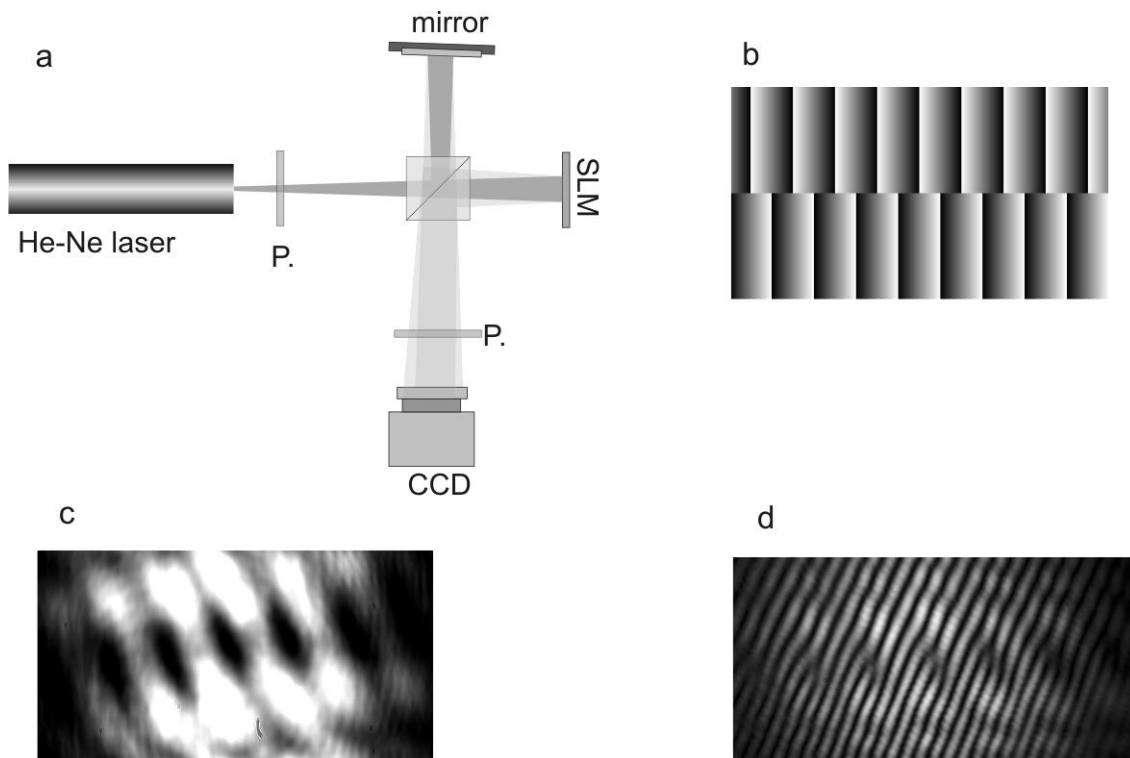


Рисунок 3.3 - Експериментальна установка показана схематично (а). Вигляд конвертера ПСФК відображається на ПМС (б). Паттерн інтенсивності містить п'ять оптичних вихорів з гарною роздільною здатністю і два ОВ в області малої інтенсивності з топологічним зарядом $m = + 1$ (в). Інтерферограма поля містить сім смугових ділянок розщеплення, отриманих через інтерференцію з гауссовим променем.

Застосовано метод Фур'є до виявлених інтерферограм для відновлення фазової карти пучка, відбитого від ПМС. Основні моменти цієї техніки показані на подальших рисунках. На першому кроці ШПФ застосовується до виявленої інтерферограми (рисунок 3.4)

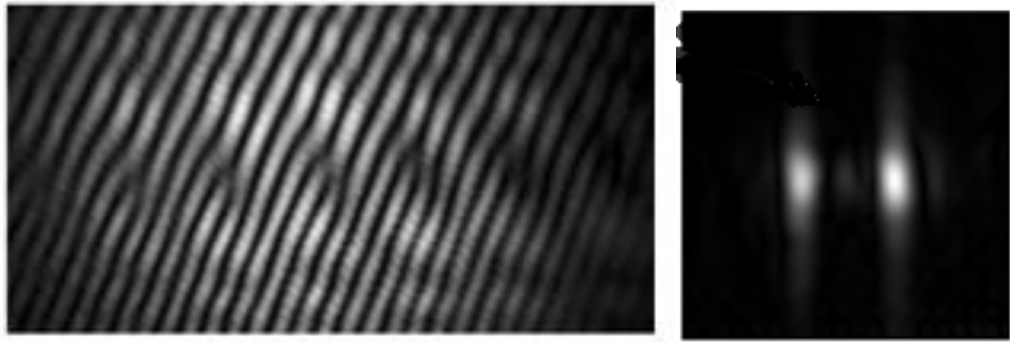


Рисунок 3.4 - Усереднене експериментальна інтерферограма та результат швидкого перетворення Фур'є

Далі перший порядок з спектра Фур'є витягується і зміщується до позиції нульового порядку (рисунок 3.5).

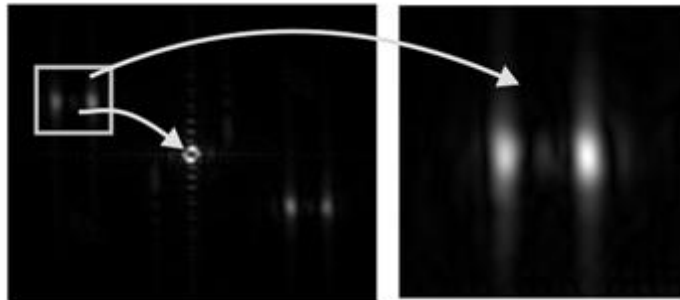


Рисунок 3.5 - Зміщення першого порядку спектра Фур'є

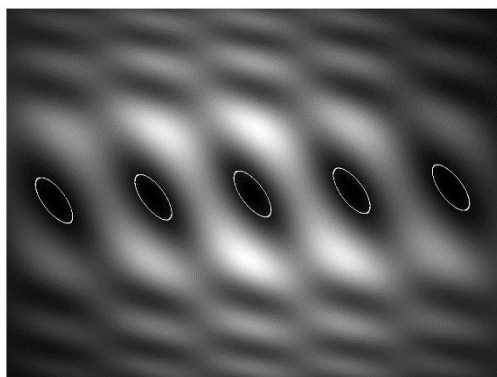
Потім зворотнє перетворення Фур'є та відновлюється фазова карта (рисунок 3.6). На відновленій карті фази ми можемо легко знайти розташування вихорів у масиві. Ця процедура застосовується до інтерферограм, виявлених вздовж осі z . За допомогою цих даних був побудований сингулярний скелет.



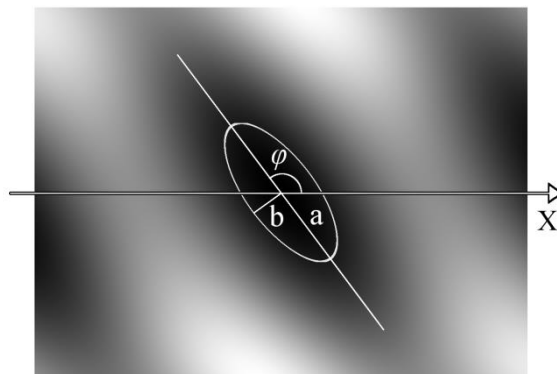
Рисунок 3.6 Відновлена фазова карта

Щоб обчислити форму області ядра, співвідношення осі еліпса та його орієнтації, використано зображення інтенсивності, виявлені на ПЗЗ-камері. Порядок обробки зображень застосовується для виявлення темних областей зображення, як на рисунках 3.4 – 3.6, а потім був використаний алгоритм установки еліпса. Проведено серію експериментів для визначення параметрів еліпсу на відстані z .

3.4.1 Результати реалізації алгоритму первинної обробки зображень



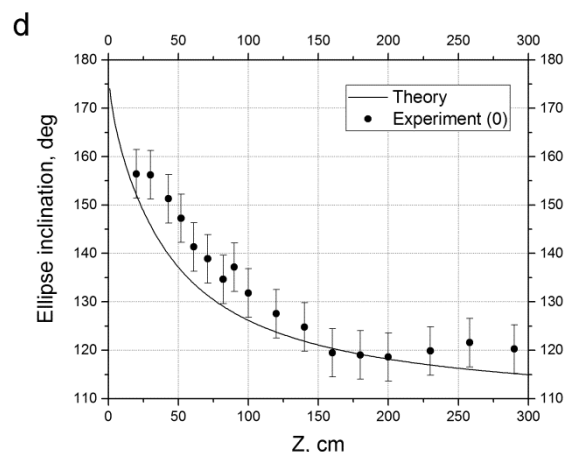
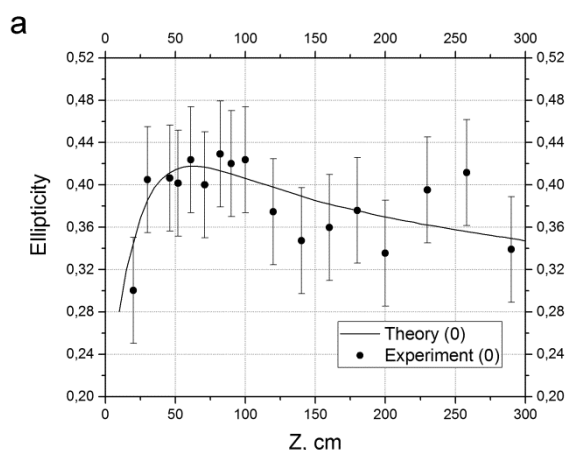
а



б

Рисунок 3.7 - Розраховані розподіли інтенсивності, які містять п'ять оптичних вихорів (а) Еліптична структура ядра навколо оптичного вихору по шару інтенсивності на поперечному перерізі пучка в площині ХУ. Точка невизначеної фази розташована в центрі еліпса. Позначення: велика піввісь a , мала піввісь b і кут φ між позитивним напрямком осі X і великою піввіссю (б).

Перший виникає завдяки збільшенню природного пучка з зростанням відстані. З'являється кутовий компонент завдяки поперечній енергії циркуляції в області, що індукується гелікоїдальними хвильовими частинами згенерованих ОВ. Одиночний скелет демонструє зовнішні особливості ОВ, що існували в цій галузі, включаючи топологічні реакції. Для нашого випадку топологічні реакції не зустрічаються в полі, а ОВ залишаються окремими і не реагують на всі відстані пропускання пучка.



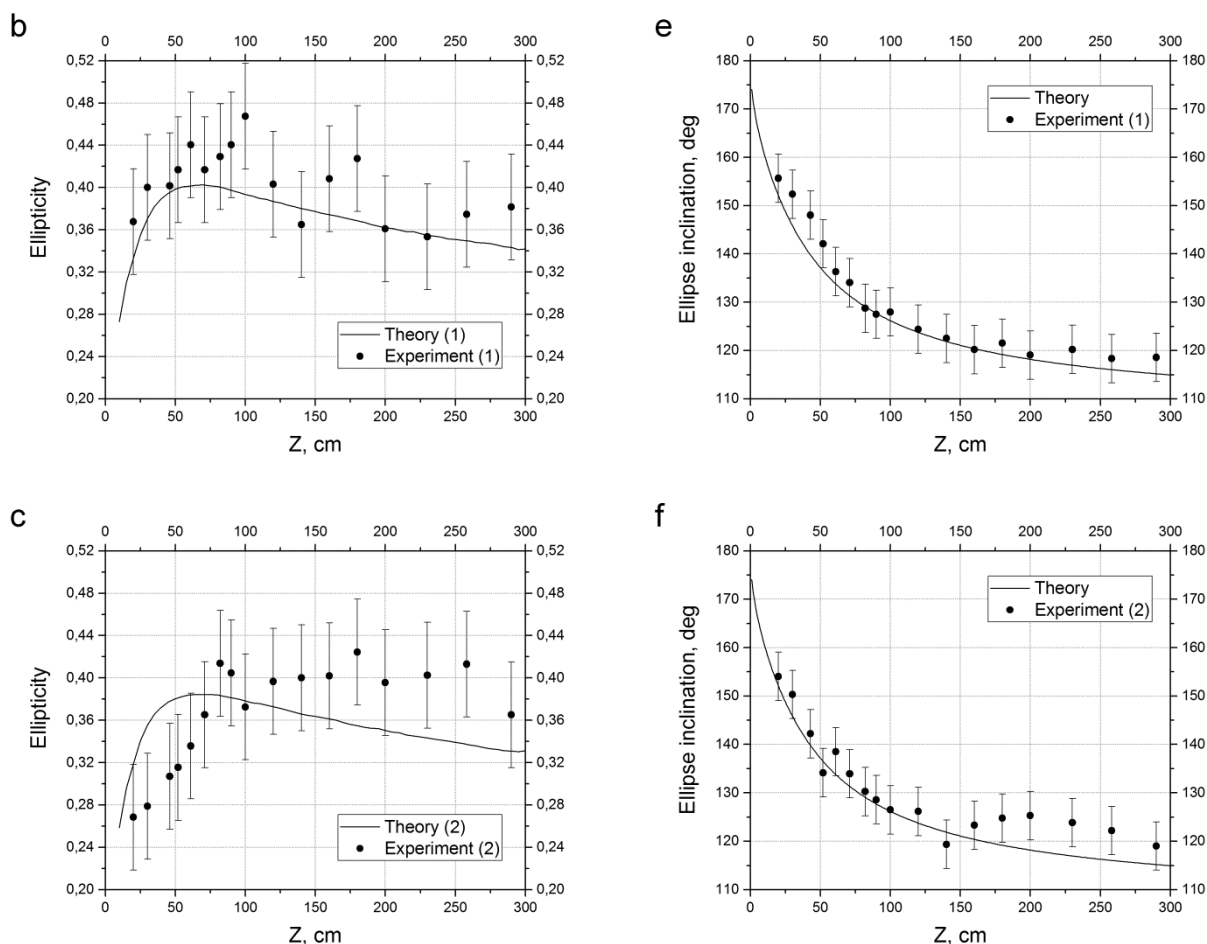


Рисунок 3.8 - Порівняння теоретичних та експериментальних параметрів досліджуваного явища. Теоретичні криві показані твердими чорними лініями, а експериментальні дані представлені точками з баром абсолютної похибки вимірювання.

Внутрішні особливості ОВ представлені основними параметрами. Залежно від співвідношення осі еліпса рисунок 3.8 (а, b, c) та кута нахилу рисунок 3.8(d, e, f) для порядку 0, 1, 2 ОВ відповідно до відстані z відповідно.

Внутрішні особливості генерованих фазових сингулярностей можна визначити на основі еліптичності форми ядра ОВ. Ми розрахували та виміряли такі параметри, як топологічний заряд, еліптичність та кут орієнтації еліпсу. Топологічний заряд - це одиничне значення з позитивним знаком для кожного з п'яти оптичних вихорів. Залежність еліптичності γ від відстані z (рисунок 5.6a, b, c) є схожою для різних порядків ОВ та її максимальне значення, яке відповідає найвищій циркуляції пучка, що протікає на відстані близько 60 см. Кут нахилу φ основної осі еліпса з позитивним напрямком осі X не залежить від порядку ОВ (рисунок 5.6d, e, f). Функція має переріз з віссю кута нахилу на 180 градусів на нульовій відстані та горизонтальної асимптоти з межею: $\lim_{z \rightarrow \infty} \varphi(z) = b$, де $b = 107.231$ градусів.

3.4.2 Результати реалізації алгоритму вторинної обробки зображень

У цьому розділі представлені дослідження поля, отримані з падаючого гаусового пучка, що проходить через перетворювач ПСФК у формі. Інтенсивність візерунків та фазових структур показують, що синтезовані окремі вигнуті оптичні вихори подібні до теоретичних структур для фіксованого випадку K значення. Кількість згенерованих оптичних вихорів обмежується роздільною здатністю ПМС. Докладно продемонстровано випадок створення п'яти окремих вирівняних оптичних вихорів у полі з використанням перетворювача ПСФК з градієнтом фаз $K = 10$. З цієї причини використано відповідні аналітичні формули та експериментальну установку, представлені на рисунку 3.3. Теоретично вивчаються характеристики вихідного поля для всього діапазону z і експериментально для розрахункових відстаней від 18 до 290 см конвертора ПСФК.

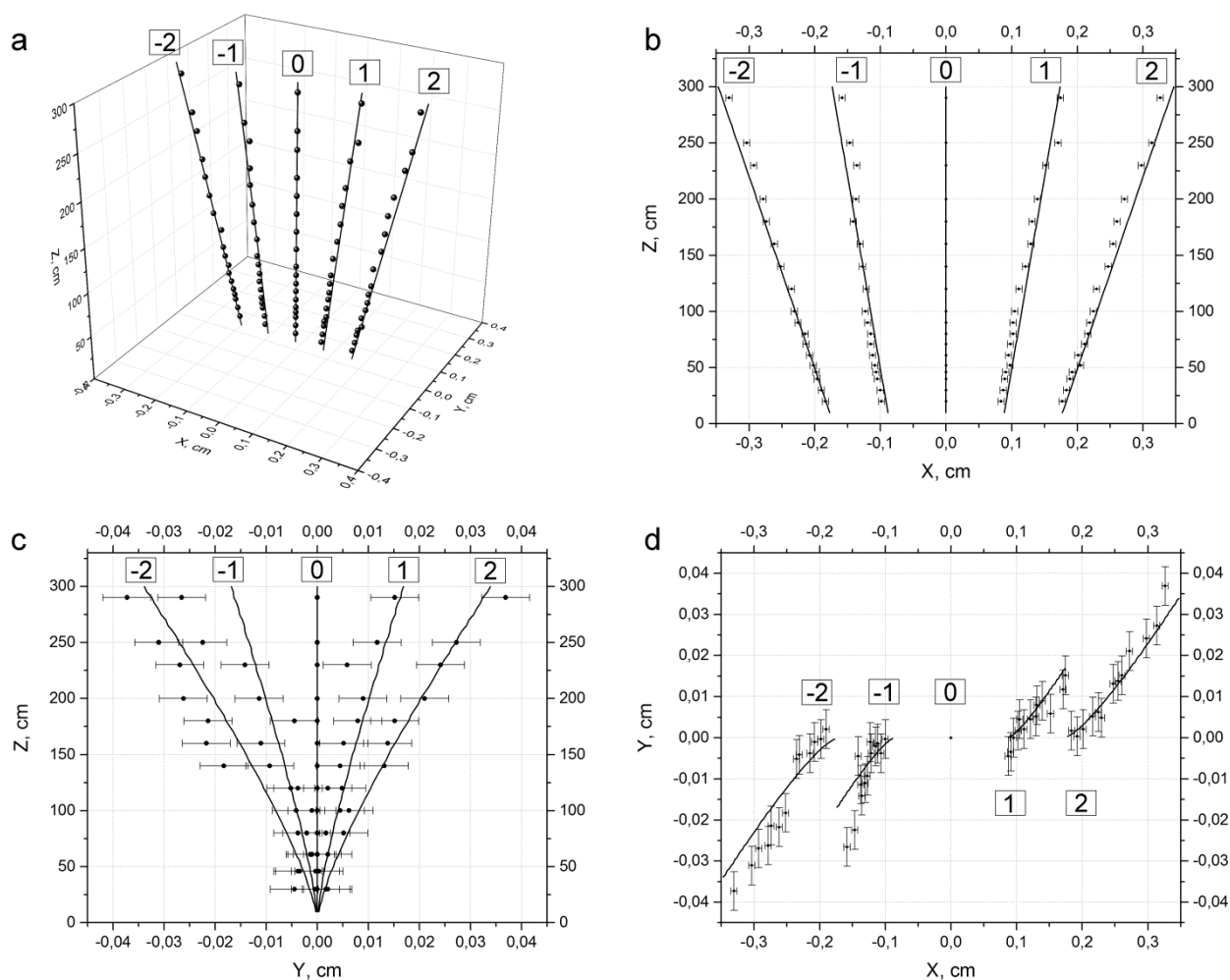


Рисунок 3.9 - Зовнішні особливості поширеного поля з оптичними вихорами. Одиночний скелет поля отриманий з вхідного гаусового пучка, перетворений перетворювачем ПСФК з

градієнтом фази $K = 10$ (a). Зображення (b), (c) та (d) являють собою однозначні проєкції каркасів на поперечні площини XZ, YZ та XY, відповідно. Напівжирні лінії вказують на теоретичні розрахунки.

Зовнішні та внутрішні особливості фазових особливостей у галузі є головним інтересом нашого дослідження. Вичерпна інформація про зовнішні особливості ОВ включає траєкторії ОВ для всього діапазону z , яка виконується сингулярним скелетом (СС). Теоретичні траєкторії ОВ та експериментальні точки їх вимірювання координат представлені на рисунок 3.9а. У тривимірних просторах сингулярний скелет створює п'ять ліній, локалізованих уздовж осі X. Кращу візуалізацію сингулярного скелету можна спостерігати, беручи до уваги сингулярні проєкції скелета на координатні площини. Лінійні траєкторії ОВ можуть бути видимі на площині XZ, які добре узгоджуються з експериментальними даними (рисунок 3.9b). Проєкція на площину YZ являє собою траєкторії, які виявляють кривину, яка зменшується з збільшенням порядку порядку ОВ (рисунок 3.9c). Цінна інформація про поведінку ОВ представлена на проєкції XY (рисунок 3.9d). Це показує зміщення ОВ на відстані розповсюдження в двох напрямках: радіальному та кутовому.

3.5 Висновки до третього розділу

У третьому розділі проведено розроблення програмно-алгоритмічного засобу для первинної та вторинної обробки оптичних зображень. У цьому розділі розроблено та реалізовано алгоритм первинної та вторинної обробки зображення. Наведено розробку програмно-алгоритмічного засобу в математичному пакеті MATLAB. На основі отриманих результатів обробки зроблені висновки про досліджуване явище та сформульовані подальші пропозиції впровадження та удосконалення програмно-алгоритмічного засобу.

Створено програмно-алгоритмічний засіб для роботи з експериментальними зображеннями. Проведено порівняльний аналіз теоретичних даних та експерименту. Отримані після обробки за допомогою ПАЗ інтерференційні картини описують як статичні дефекти оптичних елементів так і динамічні параметри системи. ПАЗ збільшує швидкість отримання результатів та підвищує їхню якість та точність.

Розроблено алгоритм реалізації моделі обробки оптичних зображень та реалізовано за допомогою математичного пакету MATLAB. Наведено порівняльний аналіз теоретичної моделі та експериментальних зображень явищ дифракції і інтерференції з визначенням параметрів оптичного вихору: морфології (еліптичності та кута нахилу еліпсу), та будованням скелетону.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкту, що мають вплив на персонал

У даному дипломному проєкті розробляється програмне забезпечення.

Розроблене програмне забезпечення орієнтоване на роботу з персональним комп'ютером. Експлуатовані для вирішення внутрішньовиробничих завдань ПЕОМ типу IBM PC мають наступні характеристики:

споживана потужність	220 Вт;
робоча напруга	220 В;
напруга джерел живлення	+12 В; - 12 В; +5 В;
робоча частота	50 Гц.

Виходячи з приведених характеристик, вочевидь, що для людини існує небезпека поразки електричним струмом, унаслідок недбалого поводження з комп'ютером і порушення правил експлуатації, залишення частин ПЕОМ, що знаходяться під напругою, відкритими або знятих для ремонту вузлів.

Відповідно до [66] до легкої фізичної роботи відносяться всі види діяльності, виконувані сидячи і ті, що не потребують фізичної напруги. Робота користувача ПК відноситься до категорії 1а.

При роботі на ПЕОМ користувач піддається ряду потенційних небезпек. Унаслідок недотримання правил техніки безпеки при роботі з машиною (невиконання огляду відкритих частин ПЕОМ, що знаходяться під напругою або знятих для ремонту вузлів) для користувача існує небезпека поразки електричним струмом.

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;
- джерела живлення;
- блоки ПЕОМ і друку, що знаходяться в ремонті.

Ще одна проблема полягає у тому, що спектр випромінювання комп'ютерного монітора включає рентгенівську, ультрафіолетову і інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння мала, оскільки цей вид випромінювання поглинається речовиною екрану. Проте велику увагу слід приділяти біологічним ефектам низькочастотних електромагнітних полів (аж до порушення ДНК).

Відповідно до [67], при обслуговуванні ПЕОМ мають місце фізичні і психофізичні

небезпечні, а також шкідливі виробничі чинники:

- підвищене значення напруги в електричному ланцюзі, замикання якої може відбутися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітних випромінювань;
- підвищена або знижена температура повітря робочої зони;
- підвищений або знижений рух повітря;
- підвищена або знижена вологість повітря;
- відсутність або недостатність природного світла;
- підвищена пульсація світлового потоку;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- розумове перенапруження;
- емоційні навантаження;
- монотонність праці.

4.2 Заходи щодо техніки безпеки

Основним небезпечним чинником при роботі з ЕОМ є безпека поразки людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм чинить на нього складну дію, що є сукупністю термічної(нагрів тканин і біологічних середовищ), електролітичної(розкладання крові і плазми) і біологічної(роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Тяжкість поразки людини електричним струмом залежить від цілого ряду чинників:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;
- роду і частоти струму;
- індивідуальних властивостей людини і навколишнього середовища.

Розроблений дипломний проект передбачає наступні технічні способи і засоби, що застерігають людину від ураження електричним струмом [68]:

- заземлення електроустановок;
- занулення;

- захисне відключення;
- електричне розділення ятерів;
- використання малої напруги;
- ізоляція частин, що проводять струм;
- огорожа електроустановок.

Занулення зменшує напругу дотику і обмежує година, протягом якого людина, ткнувшись до корпусу, може потрапити під дію напруги.

Струм однофазного короткого замикання визначається по наближеній формулі:

$$I_K = U_{\phi} / (Z_{\Pi} + Z_T / 3) \quad (4.1)$$

де U_{ϕ} - номінальна фазна напруга мережі, В;

Z_{Π} - повний опір петлі, створене фазними і нульовими дротами, Ом;

Z_T - повний опір струму короткого замикання на корпус, Ом.

Згідно таблиці 4 [69]: $Z_T / 3 = 0,1$ Ом.

Для провідників і жил кабелю для розрахунку повного опору петлі використовуємо формулу(4.2.) :

$$Z_{\Pi} = \sqrt{R_{\Pi}^2 + X_{\Pi}^2} \quad (4.2)$$

де $R_{\Pi} = R_{\phi} + R_0$ - сумарний активний опір фазного R_{ϕ} і нульового R_0 дротів, Ом;

X_{Π} - індуктивний опір паяння дротів, Ом.

Перетин 1 км мідного дроту $S = 2.5$ мм, тоді згідно таблицям 5 і 6 [69], має такий опір:

$X_{\Pi} = 0,11$ Ом;

$R_{\phi} = 7,55$ Ом;

$R_0 = 7,55$ Ом.

Отже, $R_{\Pi} = 7,55 + 7,55 = 15,1$ Ом.

Тоді по формулі (4.2) знаходимо повний опір петлі :

$$Z_{\Pi} = \sqrt{15,1^2 + 0,11^2} \approx 15,1 \text{ (Ом)}$$

Струм однофазного короткого замикання рівний:

$$I_K = 220 / (15,1 + 0,1) = 14,47 \text{ (A)}.$$

Дія плавкої вставки на ПЕОМ забезпечується, якщо виконується співвідношення:

$$I_K \geq k * I_n, \quad (4.3)$$

де I_n - номінальний струм спрацьовування плавкої вставки, А;

k - коефіцієнт кратності нелінійного струму I_n , А.

Коефіцієнт кратності нелінійного струму I_n розраховується по формулі (4.4.) :

$$I_n = P / U, \quad (4.4)$$

де $P = 220$ Вт - споживана потужність;

$U = 220$ В - робоча напруга;

$k = 3$ А - для плавких вставок.

Отже, $I_n = 220 / 220 = 1$ А.

Підставивши значення у вираз (4.3), одержимо:

$$14,47 > 3 * 1.$$

Таким чином, доведено, що апарат забезпечить спрацьовування (і захист) при підвищенні номінального струму.

4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Вимоги до виробничих приміщень встановлюються [70], СНіП, відповідними ГОСТами і ОСТАми з урахуванням небезпечних і шкідливих чинників, що утворюються в процесі експлуатації електроустаткування.

Підвищення працездатності людини і збереження її здоров'я забезпечується стабільними метеорологічними умовами. Мікроклімат виробничих приміщень [71] визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря, а також

температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводить до порушення систем кровообігу, нервової і потовидільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

Відповідно до [72] встановлюють оптимальну і допустиму температуру, відносну вологість і швидкість руху повітря в робочій зоні. За відсутності надмірного тепла, вологи, шкідливих речовин в приміщенні досить природної вентиляції.

У приміщенні для виконання робіт операторського типу (категорія 1а), пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (табл. 4.1).

Таблиця 4.1 - Санітарні норми мікроклімату робочої зони приміщень для робіт категорії 1а.

Пора року	Температура, С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодна	22...24	40...60	0,1
Тепло	23...25	40...60	0,1

У приміщенні, де знаходиться ПЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (з пристроєм вентиляційних каналів в перекриттях будівлі і вертикальних шахт) й устанавленого промислового кондиціонера фірми Mitsubishi, який дозволяє вирішити переважну більшість завдань по створінню та підтримці необхідних параметрів повітряного середовища. Цей метод забезпечує приток потрібної кількості свіжого повітря, визначеного в СНіП (30 м³ в годину на одного працівника).

Шум на виробництві має шкідливу дію на організм людини. Стомлення операторів через шум збільшує число помилок при роботі, призводить до виникнення травм. Для оператора ПЕОМ джерелом шуму є робота принтера. Щоб усунути це джерело шуму, використовують наступні методи. При покупці принтера слід вибирати найбільш шумозахисні матричні принтери або з великою швидкістю роботи (струменеві, лазерні). Рекомендується принтер поміщати в найбільш віддалене місце від персоналу, або застосувати звукоізоляцію та звукопоглинання (під принтер підкладають демпфуючі підкладки з пористих звукопоглинальних матеріалів з листів тонкої повсті, поролону, пеноплону).

При роботі на ПЕОМ, проектом передбачені наступні методи захисту від електромагнітного випромінювання: обмеження часом, відстанню, властивостями екрану.

Обмеження годині роботи на ПЕОМ складає 3,5-4,5 години. Захист відстанню передбачає розміщення монітора на відстані 0,4-0,5 м від оператора. Передбачений монітор 20" TFT, Samsung 2043BW відповідає вимогам стандарту [73].

Стандарт [73] пред'являє жорсткі вимоги в таких областях: ергономіка (фізична, візуальна і зручність користування), енергія, випромінювання (електричних і магнітних полів), навколишнє

середовище і екологія, а також пожежна та електрична безпека, які відповідають всім вимогам [74].

Для зниження стомлюваності та підвищення продуктивності праці обслуговуючого персоналу в колірній композиції інтер'єру приміщень для ПЕОМ дипломним проектом пропонується використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

У проекті передбачається використання сумісного освітлення. У світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Як штучне освітлення необхідно використовувати штучне робоче загальне освітлення. Для загального освітлення необхідно використовувати люмінесцентні лампи. Вони володіють наступними перевагами: високою світловою віддачею, тривалим терміном служби, хоча мають і недоліки: високу пульсацію світлового потоку.

При експлуатації ПЕОМ виробляється зорова робота. Відповідно до [75] ця робота відноситься до розряду 5а. При цьому нормоване освітлення на робочому місці(Ен) при загальному освітленні рівна 200 лк.

Приміщення завдовжки 12 м, шириною 10 м, заввишки 4 м обладнується світильниками типу ЛПО2П, оснащеними лампами типу ЛБ зі світловим потоком 3120 лм кожна.

Виконаємо розрахунок кількості світильників в робочому приміщенні завдовжки $a=12$ м, шириною $b=10$ м, заввишки $z=4$ м, використовуючи формулу (4.5) розрахунку штучного освітлення при горизонтальній робочій поверхні методом світлового потоку:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.5)$$

де F - світловий потік = 3120 лм;

E - максимально допустима освітленість робочих поверхонь = 200 лк;

S - площа підлоги = 120 м²;

Z - поправочний коефіцієнт світильника = 1,2;

k - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників = 1,5;

n - кількість світильників;

U - коефіцієнт використання освітлювальної установки = 0,6;

M - кількість ламп у світильнику = 2.

Отже, $n = (200 \cdot 120 \cdot 1,2 \cdot 1,5) / (3120 \cdot 0,6 \cdot 2) = 12$.

Виходячи з цього, рекомендується використовувати 12 світильників. Світильники слід розміщувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників

зображена на рисунок 4.1.

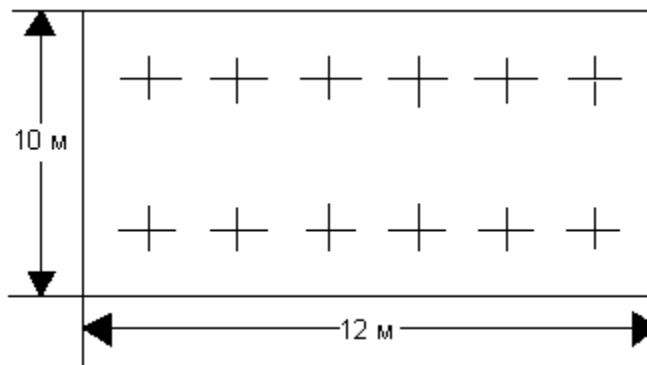


Рисунок 4.1 - Схема розташування світильників

4.4 Рекомендації по пожежній безпеці

Пожежі в приміщеннях, де встановлена обчислювальна техніка, представляють небезпеку для життя людини. Пожежі також пов'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що у свою чергу спричиняє за собою порушення ходу технологічного процесу.

Пожежа може виникнути при наявності горючої речовини та внесення джерела запалювання в горюче середовище. Пальними матеріалами в приміщеннях, де розташовані ПЕОМ, є:

- поліамід - матеріал корпусу мікросхеми, горюча речовина, температура самозаймання аерогелю 420 З ;
- полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 З, температура самозаймання 530 З, кількість енергії, що виділяється при згоранні - 18000 - 20700 кДж/кг;
- стеклотекстоліт ДЦ - матеріал друкарських плат, важкозаймистий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;
- пластика кабельний №489 - матеріал ізоляції кабелю, горючий матеріал, показник горючості більш 2.1;
- деревина - будівельний і обробний матеріал, матеріал з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, теплота згорання 18731 - 20853 кДж/кг, температура запалювання 399 З, схильна до самозаймання [66].

Згідно [77] приміщення відносяться до категорії В(пожежовибухонебезпечним) і згідно правилам побудови електроустановок простір усередині приміщення відноситься до

вогнебезпечної зони класу П - Па (зони, розташовані в приміщеннях, в яких зберігаються тверді горючі речовини).

Потенційними джерелами запалення при роботі ПЕОМ є:

- іскри при замиканні і розмиканні ланцюгів;
- іскри і дуги коротких замикань;
- перегріву від тривалого перевантаження і наявності перехідного опору.

Продуктами згорання, що виділяються при пожежі, є : оксид вуглецю, сірчистий газ, оксид азоту, синильна кислота, акролеїн, фосген, хлор та ін. При горінні пластмас, окрім звичайних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол та ін., що шкідливо впливають на організм людини.

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигаза з коробкою марки В(жовта).

Пожежна безпека об'єктів народного господарства регламентується [78] і забезпечується системами запобігання пожежам і протипожежному захисту. Для успішного гасіння пожеж вирішальне значення має швидке виявлення пожежі і своєчасний виклик пожежних підрозділів до місця пожежі.

Зменшити горюче навантаження не представляється можливим, тому проектом передбачається застосувати наступні способи і їх комбінації для запобігання утворенню(внесення) джерел запалення :

- застосування устаткування, що задовольняє вимогам електростатичної безпеки;
- застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалення;
- виключення можливості появи іскрового заряду статичної електрики в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення;
- підтримка температури нагріву поверхні машин, механізмів, устаткування, пристроїв, речовин і матеріалів, які можуть увійти до контакту з палим середовищем, нижче гранично допустимої, становить 80% якнайменшої температури самозаймання пального.
- заміна небезпечних технологічних операцій більш безпечними;
- ізолюване розташування небезпечних технологічних установок і устаткування;
- зменшення кількості палих і вибухонебезпечних речовин, що знаходяться у виробничих приміщеннях;
- запобігання можливості утворення палих сумішей на лінії, вентиляційних системах і ін.;

- механізація, автоматизація та справність(потокова) виробництва;
- суворе дотримання стандартів і точне виконання встановленого технологічного режиму;
- запобігання можливості появи в небезпечних місцях джерел запалення;
- запобігання розповсюдженню пожеж і вибухів;
- використання устаткування і пристроїв, при роботі яких не виникає джерел запалення;
- виконання вимог сумісного зберігання речовин і матеріалів;
- наявність громовідводу;
- організація автоматичного контролю параметрів, що визначають джерела запалення;
- ліквідація можливості самозаймання речовин і матеріалів .
- Для запобігання пожежі в обчислювальних центрах проектом пропонується виконання наступних вимог :
- електроживлення ЕОМ повинно мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолодження і кондиціонування;
- система вентиляції обчислювальних центрів повинна бути обладнана блокуючими пристроями, що забезпечують її відключення на випадок пожежі;
- робочі місця повинні бути оснащені пожежними щитами, сигналізацією, засобами для сповіщення про пожежну небезпеку (телефонами), медичними аптечками для надання першої медичної допомоги, розробленим планом евакуації.

Для зниження пожежної небезпеки в приміщеннях використовуються первинні засоби гасіння пожеж, а також система автоматичної пожежної сигналізації, яка дозволяє знайти початкову стадію загоряння, швидко і точно оповістити службу пожежної охорони про час і місце виникнення пожежі.

Відповідно до [79] приміщення категорії В підлягають устаткуванню системами автоматичної пожежної сигналізації. Проектом передбачається застосування датчика типу ІДФ - І(димовий фотоелектричний датчик), оскільки специфікою пожеж обчислювальної техніки і радіоапаратури є, в першу чергу, виділення диму, а потім - підвищення температури.

При виникненні пожежі в робочому приміщенні обслуговуючий персонал зобов'язаний негайно вжити заходи по ліквідації пожежі. Для ліквідації пожежі використовують вогнегасники (хімічно-пінні, пінні для повітря ОП-5, ОП-6, ОП-9, вуглекислотні ОУ-5), пісок, пожежний інвентар(сокири, ломи, багри, шерстяну або азбестову ковдри) [80]. Як засіб індивідуального захисту проектом передбачається використання промислового протигаза з маскою, фільтруючої коробки В.

В якості організаційно-технічних заходів рекомендується проводити навчання робочого персоналу правилам пожежної безпеки.

4.5 Охорона навколишнього природного середовища

Діяльність за темою магістерської роботи, а саме розробці автоматизованої системи моделювання рівноважного складу впливає на навколишнє природне середовище і регламентується нормами діючого законодавства [81 - 86].

Основним екологічним аспектом в процесі діяльності за даними спеціальностями є процеси впливу на атмосферне повітря та процеси поводження з відходами, які утворюються, збираються, розміщуються, передаються на видалення (знешкодження), утилізацію, тощо в ІТ галузі.

Вплив на атмосферне повітря при нормальних умовах праці не оказує, бо не має в приміщенні сканерів, принтерів та інших джерел викиду забруднюючих речовин в повітря робочої зони.

В процесі створення/розробки програми на робочому місці виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- Відпрацьовані люмінесцентні лампи - I клас небезпеки
- Змінні носії інформації - IV клас небезпеки
- Відпрацьовані вогнегасники - IV клас небезпеки
- Макулатура - IV клас небезпеки
- Відпрацьовані фільтрувальні засоби індивід. захисту (респіратори, протигази) - IV клас небезпеки
- Побутові відходи - IV клас небезпеки

4.6 Висновки

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в кваліфікаційній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника.

Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Була наведена схема, розміри приміщення та наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

ВИСНОВКИ

Під час написання дипломної роботи виконано ряд поставлених задач:

- Проаналізовано існуючі методи моделювання та математичні моделі дифракції, інтерференції.
- Створено інформаційні моделі опису теоретичного та експериментального дослідження дифракції та інтерференції
- Проаналізовано відомі методи обробки зображень
- Розроблено алгоритм реалізації моделі первинної обробки зображення
- Розроблено алгоритм реалізації моделі додаткової обробки зображення за допомогою перетворення Фур'є.
- Спроектовано та реалізовано програмно-алгоритмічний засіб для первинної та додаткової обробки оптичних зображень
- Впроваджено програмно-алгоритмічний засіб для використання в оптичній лабораторії Вроцлавського технологічного університету.

В результаті виконання дипломної роботи досягнуто поставлену мету: проведено дослідження методів моделювання дифракції та інтерференції, методів обробки оптичних зображень, створення інформаційних моделей опису теоретичного та експериментального дослідження дифракції та інтерференції, розроблення алгоритмів реалізації моделей обробки зображення, реалізація та впровадження програмно-алгоритмічного засобу для первинної та вторинної обробки оптичних зображень.

Рекомендації щодо подальшого розвитку досліджень. Отримані результати можуть бути використані для подальшого розвитку інформаційної технології ідентифікації топологічних об'єктів в світловому полі та підтримки прийняття рішень в зворотних задачах дифракції. Розроблені моделі, алгоритм та програмно-алгоритмічний засіб можуть бути використані при вивченні теоретичних даних, моделюванні технічних комплексів оптичної лабораторії і проектуванні оптичних схем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. M. S. Soskin and M.V. Vasnetsov, "Singular optics," *Prog. Opt.* 42, 219–276 (2001).
2. M. R. Dennis, Y. S. Kivshar, M. S. Soskin, G. A. Swartzlander Jr., "Singular optics: More ado about nothing," *J. Opt. A: Pure Appl. Opt.* 11, 090201–090203 (2009).
3. P. Senthilkumaran, Sh. Sato, J. Masajada, "Singular Optics," *International Journal of Optics*, 1-106 (2012).
4. Quantum computing with neutral atoms David S. Weiss and Mark Saffman, *Physics Today* 70, 7, 44 (2017); doi: 10.1063/PT.3.3626ДСТУ 2938-94 Основні поняття. Терміни та визначення.
5. Хорошун Г.М., Рязанцев А.О. Топологія дифракційного поля для керування траєкторіями руху мікро- та наночастинок. *Вісник Східноукраїнського національного університету імені Володимира Даля* No 9 (239) 2017, с. 72-77.
6. Бутиков, Е.И. Интерактивные компьютерные модели в преподавании физики / Е.И. Бутиков. – Санкт-Петербургский государственный университет, 2000. – 80 с.
7. С. А. Ахманов, А. П. Сухоруков, П.В. Хохлов Самофокусировка и дифракция света в нелинейной среде, 1967.
8. Jönsson C. Elektroneninterferenzen an mehr ernen kunstlich hergestellten Feinspalten // *Zeitschrift für Physik.* – 1961. – V. 161. – P. 454–474. Reprinted in English as «Electron diffraction at multiple slits» // *Am. J. Phys.* – 1974. – V. 42. – P.4–11.
9. Дифракція [Електронний ресурс]: <https://uk.wikipedia.org/wiki/Дифракція> (Дата звернення: 22.10.2018)
10. Дифракційна ґратка [Електронний ресурс]: https://uk.wikipedia.org/wiki/Дифракційна_ґратка (Дата звернення: 22.10.2018)
11. MATLAB [Електронний ресурс]: <https://ru.wikipedia.org/wiki/MATLAB> (Дата звернення: 22.10.2018)
12. Palmer C. *Diffraction Grating Handbook*, 7th edition. — New York: Newport Corporation, 2014.
13. Майер Р. В. ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ: Учебное пособие. -- Глазов: ГГПИ, 2005. -- 25 с.
14. Оптична дифракція на періодичних структурах : монографія / В. М. Фітьо, Я. В. Бобицький ; М-во освіти і науки України, Нац. ун-т "Львів. політехніка". – Львів : Вид-во Львів. політехніки, 2013. – 298, [3] с. : іл. – Бібліогр.: с. 278-298 (307 назв). – ISBN 978-617-607-489-2
15. Боев В.Д., Сыпченко Р.П., Компьютерное моделирование. — ИНТУИТ.РУ, 2010. — – 349 с.

16. Булавин Л.А., Выгорницкий Н.В., Лебовка Н.И. Компьютерное моделирование физических систем. — Долгопрудный: Издательский Дом “Интеллект”, 2011. — 352 с.
17. Бусленко Н.П. Моделирование сложных систем. — М.: Наука, 1968. — 356 с.
18. Дворецкий С.И., Муромцев Ю.Л., Погонин В.А. Моделирование систем. — М.: Изд. центр “Академия”, 2009. — 320 с.
19. Кунин С. Вычислительная физика. — М.: Мир, 1992. — 518 с.
20. Паничев В.В., Соловьев Н.А. Компьютерное моделирование: учебное пособие. — Оренбург: ГОУ ОГУ, 2008. -- 130 с.
21. Рубанов В.Г., Филатов А.Г. Моделирование систем учебное пособие. — Белгород: Изд-во БГТУ, 2006. — 349 с.
22. Самарский А.А., Михайлов А.П. Математическое моделирование: Идеи. Методы. Примеры. — М.: Физматлит, 2001. — 320 с.
23. Советов Б.Я., Яковлев С.А. Моделирование систем: Учеб для вузов — М.: Высш. Шк., 2001. — 343 с.
24. Федоренко Р.П. Введение в вычислительную физику: Учеб. пособие: Для вузов. — М.: Изд-во Моск. физ.-техн. ин-та, 1994. — 528 с.
25. Шеннон Р. Имитационное моделирование систем: искусство и наука. — М.: Мир, 1978. — 302 с.
26. Самарский А.А., Михайлов А.П. Математическое моделирование: Идеи. Методы. Примеры. — М.: Физматлит, 2001. — 320 с.
27. Советов Б.Я., Яковлев С.А. Моделирование систем: Учеб для вузов — М.: Высш. Шк., 2001. — 343 с.
28. Федоренко Р.П. Введение в вычислительную физику: Учеб. пособие: Для вузов. — М.: Изд-во Моск. физ.-техн. ин-та, 1994. — 528 с.
29. Шеннон Р. Имитационное моделирование систем: искусство и наука. — М.: Мир, 1978. — 302 с.
30. Y. Izdebskaya, “Optical necklaces generated by the diffraction on a stack of dielectric wedges,” *Phys. Lett. A* 372, 3909-3913 (2008).
31. A. N. Khoroshun, A. V. Chernykh, A. N. Tsimbaluk, J. A. Kirichenko, P. V. Yezhov, A. V. Kuzmenko, J. T. Kim, “Properties of an Axial Optical Vortex Generated with the use of a Gaussian Beam and Two Ramp,” *Journal of Nanoscience and Nanotechnology* 16 (2), 2105-2107 (2016).
32. V.G. Shvedov, Y.V. Izdebskaya, A.N. Alekseev, A.V. Volyar, “The formation of optical vortices in the course of light diffraction on a dielectric wedge,” *Technical Physics Letters* 28 (3), 256-259 (2002).
33. H. M. Takeda, H. Ina, S. Kobayashi, "Fourier-transform method of fringe pattern analysis for computer based topography and interferometry," *JOSA*, 72, 156-160 (1982).

34. G. H. Kim, J. H. Jeon, K. H. Ko, H. J. Moon, J. H. Lee, and J. S. Chang, "Optical vortices produced with a nonspiral phase plate," *Appl. Opt.* 36, 8614–8621 (1997).
35. Bekshaev, A. Ya. & Karamoch, A. I. (2008). "Spatial characteristics of vortex light beams produced by diffraction gratings with embedded phase singularity". *Opt. Commun.*, Vol. 281, No 6, 1366–1374.
36. Bekshaev, A. & Karamoch, A. (2008). "Displacements and deformations of a vortex light beam produced by the diffraction grating with embedded phase singularity". *Opt. Commun.*, Vol. 281, No 14, 3597–3610.
37. Gonzales R. C. *Digital Image Processing Using MATLAB* R. C. Gonzales, R. E. Woods, S. Eddins. – Prentice Hall, Upper Saddle River, NJ, 2004 – 492 p.
38. Greenspan D. *Introduction to Numerical Analysis and Applications* /D. Greenspan. – Markham, Chicago, 1971. – 176 p.
39. *Image Processing Toolbox For Use with Matlab, User's Guide. Version 3.* – The Math Works Inc., 2004. – 775 p
40. Квуетный R. *Basics of Modelling and Computational Methods* /R. Квуетный. – Вінниця : ВДТУ, 2007. – 147 с
41. *Методы компьютерной обработки изображений / Под ред. Сойфера В. А. – 2 изд. испр. – М. : ФИЗМАТЛИТ, 2003. – 784 с.*
42. Абрамовиц М. *Справочник по специальным функциям / М. Абрамовиц, И. Стиган. – М. : Наука, 1979. – 486 с*
43. Амосов А. А. *Вычислительные методы для инженеров: учеб. пособ. /Амосов А. А. , Дубинский Ю. А., Копченова Н. В. – М. : Высшая школа, 1994. – 554 с.*
44. Анисимов Б. В. *Распознавание и цифровая обработка изображений /Анисимов Б. В., Курганов В. Д., Злобин В. К. – М. : Высшая школа, 1983. – 468 с.*
45. Аттетков А. В. *Методы оптимизации: учеб. для вузов / Аттетков А. В. , Галкин С. В., Зарубин В. С. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2003. – 440 с*
46. Батищев Д. И. *Методы оптимального проектирования / Батищев Д. И. – М. : Радио и связь, 1984. – 248 с.*
47. Бахвалов Н. С. *Численные методы / Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. – М. : Наука, 1987. – 630 с.*
48. Васильков Ю. В. *Компьютерные технологии вычислений в математическом моделировании : учеб. пособие / Ю. В. Васильков, Н. Н. Василькова. – М. : Финансы и статистика, 2002. – 256 с.*
49. Верлань А. Ф. *Интегральные уравнения: методы, алгоритмы, программы: справочное пособие / Верлань А. Ф., Сизиков В.С. – К. : Наукова думка, 1986. – 544 с*

50. Вержбицкий В. М. Численные методы (математический анализ и обыкновенные дифференциальные уравнения): учеб. пособие для вузов / Вержбицкий В. М. – М. : Высш.шк., 2001. – 382 с.
51. Воеводин В. В. Матрицы и вычисления / Воеводин В. В. – М. : Наука, 1984. – 320 с.
52. Глинченко А. С. Цифровая обработка сигналов / Глинченко А. С. –Красноярск : Изд-во КГТУ, 2001. – 199с.
53. Глушков В. М. Основы безбумажной информатики / Глушков В. М.– М. : Наука, 1987. – 552 с
54. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.
55. Горелик А. Л. Методы распознавания / А. Л. Горелик, В. А. Скрипкин. – М. : Высшая школа, 1984. – 208 с.
56. Грузман И. С. Цифровая обработка изображений в информационных системах / Грузман И. С., Киричук В. С.. – Новосибирск : Изд-во НГТУ, 2002. – 352 с.
57. Дахнович А. А. Дискретные системы и цифровая обработка сигналов: учебное пособие / Дахнович А. А. – Тамбов : Изд-во Тамб. гос. техн. ун-та, 2007. – 100с.
58. Демидович Б. П. Основы вычислительной математики / Демидович Б. П., Марон И. А. – М. : Наука, 1970. – 664 с.
59. Дубовой В. М. Основи застосування ЕОМ у інженерній діяльності / Дубовой В. М. , Кветний Р. Н. – К. : ІСДО України, 1994. – 285 с.
60. Дубовой В. М. Програмування комп'ютеризованих систем управління та автоматики / Дубовой В. М. , Кветний Р. Н. – Вінниця : ВДТУ, 1997. – 208 с.
61. Дубовой В. М. Програмування персональних комп'ютерів систем управління / В. М. Дубовой, Р. Н. Кветний.. – Вінниця : ВДТУ, 1999.
62. Каллан Р. Основные концепции нейронных сетей / Р. Каллан. – М.: Изд. дом «Вильямс», 2001. –288 с.
63. Канторович Л. В. Оптимальные решения в экономике / КанторовичЛ. В., Горстков А. Б. – М. : Наука, 1972. – 335 с.
64. Кветный Р. Н. Математическое моделирование в задачахпроектирования средств автоматики и информационно-измерительной техники / Кветный Р. Н. – К. :УМК ВО,1989. – 112 с.
65. Кветный Р. Н. Информационная теория измерений: от модели к изделию / Кветный Р. Н., Маликов В. Т. – М. : Знание, 1988. – 213 с. ___
66. ГОСТ 12.1.005-88. Общие санитарно-гигиенические требования к воздуху рабочей зоны.
67. ГОСТ 12.0.003-74 Опасные и вредные производственные факторы. Классификация.
68. НПАОП 40.1-1.21-98. Правила безпечної експлуатації електроустановок споживачів
69. ГОСТ 12.1.009-76. ССБТ. Электробезопасность. Термины и определения.

70. ДСП 173-96. «Державні санітарні правила планування і забудови населених пунктів
71. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень.
72. ГОСТ 12.1.005-88. Система стандартів безпеки праці. Общие санитарно-гигиенические требования к воздуху рабочей зоны.
73. TCO' 07 Certified Displays. © 2007 Copyright TCO Development AB
74. ДСанПіН 3.3.2.007-98, Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.
75. ДБН В.2.5-28-2006. Природне і штучне освітлення
76. ГОСТ 12.1.044-89 Система стандартів безпеки праці. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения.
77. НАПБ Б.03.002-2007. Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою.
78. ГОСТ 12.1.004-91. "Система стандартів безпеки праці. Пожарная безопасность. Общие требования".
79. НАПБ А.01.001-2014 «Правила пожежної безпеки в Україні»
80. НАПБ Б.03.001-2004. Про затвердження Типових норм належності вогнегасників.
81. Закон України «Про охорону навколишнього природного середовища» [Електронний ресурс] / rada.gov.ua - Режим доступу: www.URL:https://zakon.rada.gov.ua/laws/show/1264-12
82. Закон України «Про забезпечення санітарного та епідемічного благополуччя населення» [Електронний ресурс] / rada.gov.ua - Режим доступу: www.URL:https://zakon.rada.gov.ua/laws/show/4004-12
83. Закон України «Про відходи» [Електронний ресурс] / rada.gov.ua - Режим доступу: www.URL:https://zakon.rada.gov.ua/laws/show/187/98-%D0%B2%D1%80
84. Закон України «Про охорону атмосферного повітря» [Електронний ресурс] / rada.gov.ua - Режим доступу: www.URL:https://zakon.rada.gov.ua/laws/show/2707-12
85. Закон України «Про захист населення і територій від надзвичайних ситуацій техногенного та природного характеру» [Електронний ресурс] / rada.gov.ua - Режим доступу: www.URL:https://zakon.rada.gov.ua/laws/show/1809-14
86. Водний кодекс України [Електронний ресурс] / rada.gov.ua - Режим доступу: www.URL:https://zakon.rada.gov.ua/laws/show/213/95-%D0%B2%D1%80

Додаток А. Код програми

```

1. function varargout = zoomFFT(varargin)
2. % ZOOMFFT MATLAB code for zoomFFT.fig
3. %
4.
5. % Begin initialization code - DO NOT EDIT
6. gui_Singleton = 1;
7. gui_State = struct('gui_Name',       mfilename, ...
8.                   'gui_Singleton',  gui_Singleton, ...
9.                   'gui_OpeningFcn', @zoomFFT_OpeningFcn, ...
10.                  'gui_OutputFcn',  @zoomFFT_OutputFcn, ...
11.                  'gui_LayoutFcn',  [], ...
12.                  'gui_Callback',    []);
13. if nargin && ischar(varargin{1})
14.     gui_State.gui_Callback = str2func(varargin{1});
15. end
16.
17. if nargout
18.     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
19. else
20.     gui_mainfcn(gui_State, varargin{:});
21. end
22. % End initialization code - DO NOT EDIT
23.
24.
25. % --- Executes just before zoomFFT is made visible.
26. function zoomFFT_OpeningFcn(hObject, eventdata, handles, varargin)
27.
28. % Choose default command line output for zoomFFT
29. handles.output = hObject;
30.
31. % Update handles structure
32. guidata(hObject, handles);
33.
34. set(handles.uitable1,'Data',[]); %czyszczenie tablicy maksimumow (очистка таблицы максимумов)
35.
36. clear all
37.
38. global p_obraz %poprzedni (предыдущий)
39. global p_wynik2D_1
40. global p_B;
41. global p_fi1;
42. global fi1;
43. global p_sciezka;
44.
45. global B;
46.
47. p_B=0;
48. B=0;
49.
50. % włączenie przetwarzania równoległego-(включение параллельной обработки)
51. try
52. %matlabpool
53. catch
54.     t='Nie wykryto Parallel Processing Toolbox, przetwarzanie równoległe wyłączone.'
55. % (Не найден Parallel Processing Toolbox, параллельная обработка отключена)
56. end
57.
58.
59. % UIWAIT makes zoomFFT wait for user response (see UIRESUME)
60. % uiwait(handles.figure1);
61.
62.
63. % --- Outputs from this function are returned to the command line.
64. function varargout = zoomFFT_OutputFcn(hObject, eventdata, handles)
65. varargout{1} = handles.output;
66.
67. global roboczy
68. global gdzie
69. global temp
70.

```

```

71. global sciezka;
72.
73.
74. gdzie=pwd; %zerowanie sciezek dostepu (обнуление путей доступа)
75. roboczy=pwd;
76. temp=pwd;
77.
78.
79. %--- ładowanie listy plikuw-(загрузка списка файлов)---
80. function wczytaj(handles)
81.
82. global roboczy;
83. global gdzie;
84. global plik;
85.
86. cd(gdzie); %wchodzimy do ostatniej dobrej lokalizacji plikuw (вход в последнее удачное
    расположение файла)
87. temp=gdzie;% zapamiętanie tej ostatniej sprawdzonej sciezki (запоминание последнего
    проверенного пути)
88. staralista=plik; %zapamiętanie starej listy plikuw (запоминание старого списка файлов)
89.
90. [co, gdzie]=uigetfile({'*.tif;*.bmp;*.jpg'}, 'Wybierz pliki','MultiSelect', 'on');
91.
92. %nie wybrano pliku - komunikat o bledzie-(при отсутствии выбора файла - оповещение об
    шибке)-----
93. if iscell(co)==0
94.     if (co==0)
95.         set(handles.text14,'String','Nie wczytano nowego pliku'); %pause(1);
96.         set(handles.text14,'String','Oczekiwanie');
97.         gdzie=temp; %ostania znana dobra sciezka, przywrocenie zmiennej (восстановление
    переменной)
98.         plik=staralista; %przywrocenie starej listy plikuw (восстановление старого списка
    файлов)
99.         cd(roboczy) %powrot do katalogu z programem
100.            return %wyjście z funkcji !!!(выход из функции)
101.        end
102.        resetplik(handles)
103.        plik{1,1}=co;
104.    else
105.        resetplik(handles)
106.        plik=sort(co); %posortuj, jeśli wiele plikuw (сортировать, если несколько файлов)
107.        plik(2,:)={0};
108.    end
109.
110.    set(handles.text1,'String',gdzie); %sciezka (путь)
111.    set(handles.listbox1,'String',plik(1,:));
112.
113.    cd (roboczy)
114.
115.    %-----
116.    function resetplik(handles)
117.        global plik;
118.        global pakiet;
119.
120.        %reset zmiennej plik (сброс файловой переменной)
121.        plik=[];
122.        %global plik;
123.        plik{1,1}=0; %pusty cell (пустая клетка)
124.        plik{2,1}=0;
125.
126.        pakiet=0; %flaga pakietu
127.        ostatni=100000; %trik
128.        set(handles.text1,'String',''); %aktywny plik
129.
130.        %zerowanie listy plikuw (обнуление списка файлов)
131.        set(handles.listbox1,'String','');
132.        set(handles.listbox1,'Value',1);
133.        %-----
134.
135.

```

```

136. % --- Wczytywanie pliku (загрузка файла)-----
137. function pushbutton1_Callback(hObject, eventdata, handles)
138.     wczytaj(handles)
139.     %     cd (roboczy)
140.     %-----457457457457547
141.     % policz wstknpie transformatk-przywryc wartosci domyslne okna FFT-
    (Предварительный расчет значений окна по умолчанию для преобразования-восстановления)
142.     %     resetuj(handles)
143.     %
144.     %     if (get(handles.checkbox2,'Value')==1) %czy usunac rzad zerowy?
145.     %         obraz2=obraz-mean(obraz(:));
146.     %     else
147.     %         obraz2=obraz;
148.     %     end
149.     %
150.     %     set(handles.text13,'BackgroundColor',[1 0.7 0.7]);
151.     %     set(handles.text13,'String','Obliczam...'); drawnow
152.     %     tic
153.     %     wynik2D_1=zoomFFT2(obraz2,0,0,30,1);
154.     %     a=toc;
155.     %
156.     %     wynik2D=wynik2D_1; %nie wiem, po co, ale dziaia
157.     %
158.     %     set(handles.text13,'BackgroundColor',[0.7 1 0.7]);
159.     %     set(handles.text13,'String',['Policzono w ' num2str(a) ' sekund. ']);drawnow
160.     %
161.     %     axes(handles.axes3)
162.     %     imshow(abs(wynik2D_1),[])
163.     %     mapa=colormap;
164.     %     kwadrat(handles)
165.
166. %end
167.
168.
169. % --- Przelicz----- (расчет)
170. function pushbutton2_Callback(hObject, eventdata, handles)
171.     transformata(handles)
172.
173.
174.
175. % --- wybor czkstoъci-auto---- (выбор частоты)
176. function pushbutton3_Callback(hObject, eventdata, handles)
177.
178.
179.     global wynik2D;
180.     global z;
181.     global fi1;
182.     global amplituda;
183.     global faza;
184.
185.     z=0;
186.
187.     r_fx=str2num(get(handles.edit1,'String')) %f_x
188.     r_fy=str2num(get(handles.edit2,'String')) %f_y
189.     r_df=str2num(get(handles.edit3,'String')) %df
190.     r_du=str2num(get(handles.edit4,'String')) %du
191.
192.     axes(handles.axes3)
193.     imshow(abs(wynik2D),[])
194.     kolory(handles);
195.
196. %szukanie czkstoъci poъniej (поиск несущей частоты)
197.     clear ww kk
198.
199.     p=max(max(abs(wynik2D)))
200.     [ww kk]=find(abs(wynik2D)==p)
201.     hold on;plot(kk,ww,'ob')
202.
203.     if (size(ww,1)>1)
204.         set(handles.text13,'String','Znaleziono wiêcej, nii jedno maksimum!')

```

```

205.         return
206.     end
207.
208.     amplituda=p;
209.     faza=angle(wynik2D(ww,kk))*180/pi;
210.
211.     [lw lk]=size(wynik2D);
212.
213.     set(handles.text13,'String',['Amplituda: ' num2str(amplituda) ' Faza: '
num2str(faza) '°']);
214.     set(handles.text15,'String',[num2str(lw) 'x' num2str(lw) ' maksimum: (' num2str(kk)
',' num2str(ww) ') ' ]);
215.
216.     %--obliczanie czkstoъci bezwzglkdnej (od ъrodka)- вычисление абсолютной частоты (от
середины)-----
217.         fx=r_fx-r_du*(r_df/r_du+1-kk)
218.         fy=r_fy-r_du*(r_df/r_du+1-ww)
219.     %-----
220.     set(handles.edit1,'String',num2str(fx));
221.     set(handles.edit2,'String',num2str(fy));
222.
223.     %
224.     %nowa macierz (новая матрица)
225.     try
226.         for xw=-r_df+1:r_df-1
227.             for yk=-r_df+1:r_df-1
228.                 z(xw+r_df,yk+r_df)=wynik2D(ww+(xw/r_du),kk+(yk/r_du));
229.                 plot(kk+(yk/r_du),ww+(xw/r_du),'*g')
230.             end
231.         end
232.         hold off
233.
234.     catch
235.         hold off %prawidiowe rysowanie gdy bin№d (правильный рисунок при ошибке)
236.     end
237.
238.     %wyni№cz szukanie maksimum ai do przeliczenia fft (отключить поиск максимумов до
конверсии)
239.     set(handles.pushbutton8,'Enable','off')
240.     set(handles.pushbutton3,'Enable','off')
241.
242.
243.     % --- odwrotna transformata---(обратное преобразование)---
244.     function pushbutton4_Callback(hObject, eventdata, handles)
245.
246.     global obraz;
247.     global r_fx;
248.     global r_fy;
249.     global r_df;
250.     global r_du;
251.     global wynik2D;
252.     global z;
253.     global fi1;
254.     global B;
255.     global p_B;
256.     global p_fi1;
257.
258.     r_df=str2num(get(handles.edit3,'String')) %df
259.     %----- odwrotna transformata---(обратное преобразование)---
260.     [ox oy]=size(obraz);
261.     % z=zeros(9,9);
262.     % z(5,5)=1
263.
264.     A=(fft2(z,ox,oy));
265.
266.     [x y]=meshgrid(1:oy,1:ox);
267.     m=r_df-1; %
268.     phi=2*pi*(x*(m-0)/oy+y*(m-0)/ox);
269.
270.     % size(phi);

```

```

271.     % size(A);
272.
273.     B=exp(i*phi).*A;  %%%<----- pochylenie (уклон)
274.
275.     % faza + ew. unwrapping
276.     fi1=angle(B);
277.     if (get(handles.checkbox1,'Value')==1)
278.         fi1=unwrap(fi1,[],1);
279.         fi1=unwrap(fi1,[],2);
280.     end
281.     fi1=180*fi1/pi;
282.
283.     set(handles.popupmenu2,'Value',1);
284.     wyswietlanie(handles)
285.
286.     set(handles.pushbutton8,'Enable','on') %uaktywnij z powrotem manualne szukanie
maksimum
287.     set(handles.pushbutton3,'Enable','on')
288.
289.     %save fi1
290.
291.     % --- Executes on button press in pushbutton5.
292.     function pushbutton5_Callback(hObject, eventdata, handles)
293.     %axes(handles.axes3)
294.     improfile
295.
296.     % ---Zapisywanie-----
297.     function pushbutton6_Callback(hObject, eventdata, handles)
298.
299.     global fi1;
300.     global plik;
301.     global gdzie;
302.     global roboczy;
303.     global B;
304.     global obraz
305.     global obraz2;
306.     global wynik2D
307.
308.     r_fx=str2num(get(handles.edit1,'String'))    %f_x
309.     r_fy=str2num(get(handles.edit2,'String'))    %f_y
310.     r_df=str2num(get(handles.edit3,'String'))    %df
311.     r_du=str2num(get(handles.edit4,'String'))    %du
312.
313.     global amplituda;
314.     global faza;
315.
316.
317.     % ktyry zaznaczony? (какой выбран?)
318.     ktory=get(handles.listbox1,'Value');
319.     nazwa=char(plik{1,ktory}) %nazwa wybranego
320.
321.     nazpliku=strrep(nazwa,'.bmp','');
322.         nazpliku=strrep(nazpliku,'.BMP','');
323.         nazpliku=strrep(nazpliku,'.tif','');
324.         nazpliku=strrep(nazpliku,'.tiff','');
325.         nazpliku=strrep(nazpliku,'.TIF','');
326.         nazpliku=strrep(nazpliku,'.jpg','');
327.         nazpliku=strrep(nazpliku,'.JPG','');
328.
329.     koniec=get(handles.edit5,'String');
330.     nazwa=strcat('amp_',nazpliku,'_',koniec,'.mat')
331.     zmienna=strcat('amp_',nazpliku,'_',koniec)
332.
333.     %nowa zmienna (новая переменная)
334.     eval([zmienna '=B;'])
335.
336.     cd (gdzie)
337.     save(nazwa,zmienna)
338.
339.     % zapis do pliku tekstowego (запись в текстовый файл)

```



```

340.     nazwa2=strcat(nazplik, '_PAR_', koniec, '.txt')
341.     fid = fopen(nazwa2, 'w');
342.     fprintf(fid, 'Plik\t%s\n', nazwa);
343.     fprintf(fid, 'Amplituda\t%f \n', amplituda);
344.     fprintf(fid, 'Faza\t%f\n', faza);
345.     fprintf(fid, 'fx\t%f\n', r_fx);
346.     fprintf(fid, 'fy\t%f\n', r_fy);
347.     fprintf(fid, 'du\t%f\n', r_du);
348.     fprintf(fid, 'df\t%f\n', r_df);
349.     fclose(fid);
350.
351.     cd (roboczy)
352.
353.     % --- wybyr palety-----
354.     function popumenu1_Callback(hObject, eventdata, handles)
355.     kolory(handles);
356.
357.     function kolory(handles)
358.     global mapa;
359.
360.     switch (get(handles.popupmenu1, 'Value'))
361.     case 1
362.         colormap(autumn)
363.     case 2
364.         colormap(bone)
365.     case 3
366.         colormap(colorcube)
367.     case 4
368.         colormap(cool)
369.     case 5
370.         colormap(copper)
371.     case 6
372.         colormap(flag)
373.     case 7
374.         colormap(gray)
375.     case 8
376.         colormap(hot)
377.     case 9
378.         colormap(hsv)
379.     case 10
380.         colormap(jet)
381.     case 11
382.         colormap(lines)
383.     case 12
384.         colormap(pink)
385.     case 13
386.         colormap(prism)
387.     case 14
388.         colormap(spring)
389.     case 15
390.         colormap(summer)
391.     case 16
392.         colormap(white)
393.     case 17
394.         colormap(winter)
395.     case 18
396.         colormap(mapa)
397.     end
398.
399.
400.     % --- Executes during object creation, after setting all properties.
401.     function popumenu1_CreateFcn(hObject, eventdata, handles)
402.
403.     if ispc && isequal(get(hObject, 'BackgroundColor'),
404.         get(0, 'defaultUiControlBackgroundColor'))
405.         set(hObject, 'BackgroundColor', 'white');
406.     end
407.
408.     % resetuj ustawienia okna FFT, pokaz wstkpny obszar i wybierz nowe okno-----

```

```

409. % сброс настроек окна FFT, показ предварительной области и выбор нового окна-----
410. function pushbutton7_Callback(hObject, eventdata, handles)
411. global wynik2D_1
412.
413. resetuj(handles)
414.     axes(handles.axes3)
415.     imshow(abs(wynik2D_1),[])
416.     mapa=colormap;
417.     %kwadrat(handles)
418.
419. function resetuj(handles)
420. set(handles.edit1,'String',num2str(0))
421. set(handles.edit2,'String',num2str(0))
422. set(handles.edit3,'String',num2str(30))
423. set(handles.edit4,'String',num2str(1))
424.
425. %--przepisz poprzednie do nowego (переназначение старого на новое)
426. function poprzednie(handles)
427.
428. r_fx=str2num(get(handles.text8,'String')) %f_x
429. r_fy=str2num(get(handles.text9,'String')) %f_y
430. r_df=str2num(get(handles.text10,'String')) %df
431. r_du=str2num(get(handles.text11,'String')) %du
432.
433. set(handles.edit1,'String',r_fx)
434. set(handles.edit2,'String',r_fy)
435. set(handles.edit3,'String',r_df)
436. set(handles.edit4,'String',r_du)
437.
438. %-zapamietaj parametry przed wczytaniem nowego pliku (запоминание параметров до ввода
    нового файла)
439. function zapamietaj(handles)
440.
441. r_fx=str2num(get(handles.edit1,'String')) %f_x
442. r_fy=str2num(get(handles.edit2,'String')) %f_y
443. r_df=str2num(get(handles.edit3,'String')) %df
444. r_du=str2num(get(handles.edit4,'String')) %du
445.
446. set(handles.text8,'String',r_fx)
447. set(handles.text9,'String',r_fy)
448. set(handles.text10,'String',r_df)
449. set(handles.text11,'String',r_du)
450.
451.
452. % --- zaznaczanie obszaru ----tylko na początku, du=1!---(определение области)---.
453. function kwadrat(handles)
454. global wynik2D_1;
455.
456. r_fx=str2num(get(handles.edit1,'String')) %f_x
457. r_fy=str2num(get(handles.edit2,'String')) %f_y
458. r_df=str2num(get(handles.edit3,'String')) %df
459. r_du=str2num(get(handles.edit4,'String')) %du
460.
461. axes(handles.axes3)
462. h = imrect(gca, [1 1 10 10]);
463. %addNewPositionCallback(h,@(p) title(mat2str(p,3)));
464. addNewPositionCallback(h,@(p) title(sprintf('(%1.0f,%1.0f,%1.0f,%1.0f)',p(1)-31,p(2)-
    31,p(3),p(4))));
465.
466. fcn = makeConstrainToRectFcn('imrect',get(gca,'XLim'),get(gca,'YLim'));
467. setPositionConstraintFcn(h,fcn);
468. setFixedAspectRatioMode(h,1)
469. position = wait(h)
470. x=position(1)+position(3)/2
471. y=position(2)+position(3)/2
472. imshow(abs(wynik2D_1),[]);hold on;plot(x,y,'xg');hold off;
473. kolory(handles)
474.
475. dx=round(-(r_df-x))
476. dy=round(-(r_df-y))

```

```

477.
478.     set(handles.edit1,'String',num2str(dx))
479.     set(handles.edit2,'String',num2str(dy))
480.     set(handles.edit3,'String',num2str(round(position(3)/2)))
481.
482.     %---wybór typu wyświetlania -(выбор типа отображения)-----
483.     function popupmenu2_Callback(hObject, eventdata, handles)
484.     wyswietlanie(handles)
485.
486.     function wyswietlanie(handles)
487.     global obraz;
488.     global obraz2;
489.     global mapa;
490.     global wynik2D_1;
491.     global wynik2D
492.     global fi1;
493.     global B
494.     global p_obraz;
495.     global p_wynik2D_1;
496.     global p_fi1;
497.     global p_B;
498.
499.     try
500.     delta_fi=p_fi1-fi1;
501.     % różnica różnicy faz:) (разность разности фаз)
502.     delta_fi_max=max(max(delta_fi));
503.     delta_fi_min=min(min(delta_fi));
504.     catch
505.         k='Macierze o różnych rozmiarach!' %«матрица разных размеров»
506.
507.     end
508.
509.     % różnica faz (разность фаз)
510.     fi_max=max(max(fi1));
511.     fi_min=min(min(fi1));
512.
513.     % różnica faz poprzednia (предыдущая разность фаз)
514.     p_fi_max=max(max(p_fi1));
515.     p_fi_min=min(min(p_fi1));
516.
517.     typ=get(handles.popupmenu2,'Value');
518.     axes(handles.axes1);
519.     cla
520.
521.     if typ==1
522.         imshow(fi1,[]); %faza
523.         title(['{\Phi}_{min}= ' num2str(fi_min) '\circ' '\Phi}_{max}= '
num2str(fi_max) '\circ' '\Delta\Phi]=' num2str(fi_max-fi_min) '\circ' ] )
524.
525.     elseif typ==2
526.         imshow(p_fi1,[]); %poprzednia faza
527.         title(['{\Phi}_{min}= ' num2str(p_fi_min) '\circ' '\Phi}_{max}= '
num2str(p_fi_max) '\circ' '\Delta\Phi]=' num2str(p_fi_max-p_fi_min) '\circ' ] )
528.
529.     elseif typ==3
530.         imshow(delta_fi,[]); % różnica
531.         title(['{\Phi}_{min}= ' num2str(delta_fi_min) '\circ' '\Phi}_{max}= '
num2str(delta_fi_max) '\circ' '\Delta\Phi]=' num2str(delta_fi_max-delta_fi_min) '\circ'
] )
532.
533.     elseif typ==4
534.         imshow(abs(B),[]); %amplituda
535.     elseif typ==5
536.         imshow(abs(p_B),[]); %poprzednia amplituda
537.     elseif typ==6
538.         imshow(obraz2,[]); %intensywność
539.     elseif typ==7
540.         imshow(p_obraz,[]); %poprzednia intensywność
541.     end
542.

```

```

543.     kolory(handles);
544.     colorbar
545.
546.     % --- Executes during object creation, after setting all properties.
547.     function popupmenu2_CreateFcn(hObject, eventdata, handles)
548.
549.         if ispc && isequal(get(hObject,'BackgroundColor'),
           get(0,'defaultUicontrolBackgroundColor'))
550.             set(hObject,'BackgroundColor','white');
551.         end
552.
553.     % --- manualne wyznaczenie ńrodka i wybieranie obszaru----(ручное определение центра
и выбор области)
554.     function pushbutton8_Callback(hObject, eventdata, handles)
555.
556.     global wynik2D;
557.     global z;
558.
559.     z=0;
560.
561.     r_fx=str2num(get(handles.edit1,'String'))    %f_x
562.     r_fy=str2num(get(handles.edit2,'String'))    %f_y
563.     r_df=str2num(get(handles.edit3,'String'))    %df
564.     r_du=str2num(get(handles.edit4,'String'))    %du
565.
566.     % wskazywanie częstości pońnej (индикация несущей частоты)
567.     axes(handles.axes3)
568.     imshow(abs(wynik2D),[])
569.     kolory(handles);
570.     drawnow
571.
572.     [kk,ww, P]=impixel
573.     [lw lk]=size(wynik2D);
574.
575.     p=abs(P(1));
576.     amplituda=p;
577.     faza=angle(wynik2D(ww,kk))*180/pi;
578.
579.     set(handles.text13,'String',['Amplituda: ' num2str(amplituda) ' Faza: '
           num2str(faza) '°'])
580.     set(handles.text15,'String',[num2str(lw) 'x' num2str(lw) ' maksimum: (' num2str(kk)
           ', ' num2str(ww) ')']]);
581.
582.     %--obliczanie częstości bezwzględnej (od ńrodka)---(расчет абсолютной частоты (от
середины))----
583.     fx=r_fx-r_du*(r_df/r_du+1-kk)
584.     fy=r_fy-r_du*(r_df/r_du+1-ww)
585.     %-----
586.
587.     set(handles.edit1,'String',num2str(fx));
588.     set(handles.edit2,'String',num2str(fy));
589.
590.     hold on;plot(kk,ww,'ob')
591.     %
592.     %nowa macierz (новая матрица)
593.     try
594.         for xw=-r_df+1:r_df-1
595.             for yk=-r_df+1:r_df-1
596.                 z(xw+r_df,yk+r_df)=wynik2D(ww+(xw/r_du),kk+(yk/r_du));
597.                 plot(kk+(yk/r_du),ww+(xw/r_du),'*g')
598.             end
599.         end
600.     hold off
601.
602.     catch
603.         hold off %prawidowe rysowanie gdy bin°d
604.     end
605.
606.     %zmiana na nieaktywny do momentu przeliczenia FFT! (смена на неактивный до момента
преобразования БПФ)

```

```

607.     set(handles.pushbutton8,'Enable','off')
608.     set(handles.pushbutton3,'Enable','off')
609.
610.     % --- Executes on button press in checkbox1.
611.     function checkbox1_Callback(hObject, eventdata, handles)
612.
613.     % --- Wczytaj wyniki -----
614.     function pushbutton9_Callback(hObject, eventdata, handles)
615.
616.     global plik;
617.     global gdzie;
618.     global roboczy;
619.     global sciezka;
620.
621.     global obraz;
622.     global obraz2;
623.     global mapa;
624.     global wynik2D_1
625.
626.     global p_obraz %poprzedni
627.     global p_wynik2D_1
628.     global p_B;
629.     global p_fi1;
630.     global fi1;
631.     global p_sciezka;
632.
633.     global B
634.
635.     cd (gdzie); %wchodzimy do ostatniej dobrej lokalizacji plikuw (вход в последнее
        удачное место хранения файлов)
636.     temp=gdzie;% zapamietanie tej ostatniej sprawdzonej sciezki (запоминание пути)
637.
638.
639.     [plik, gdzie]=uigetfile({'*.mat'}, 'Wybierz plik');
640.
641.     %nie wybrano pliku - komunikat o bledzie(не выбрал файл - ОШИБКА!)-----
642.
643.     if (plik==0)
644.         set(handles.text1,'String','Nie wczytano nowego pliku');
645.         gdzie=temp; %ostania znana dobra sciezka, przywrocenie zmiennej (восстановление
        переменной)
646.         cd (roboczy) %powrot do katalogu z programem (вернуться в каталог программ)
647.         return %wyjście z funkcji !!!
648.     else
649.
650.         cd (gdzie)
651.
652.         p_obraz=obraz2; %poprzedni
653.         p_wynik2D_1=wynik2D_1; %poprzedni
654.         p_B=B; %zapamietaj poprzedni amplituda
655.         p_fi1=fi1; %zapamietaj poprzedni faza
656.         p_sciezka=sciezka; %zapamietaj poprzednie sciezka
657.
658.         save 'handles.mat' handles
659.         load (plik); %wczytaj zestaw zmiennych (niepeiny!) (загрузить набор переменных)
660.         load handles.mat
661.
662.         sciezka=[gdzie plik ' <- wyniki'];
663.         set(handles.text1,'String',sciezka);
664.         % set(handles.text7,'String',p_sciezka);
665.
666.         %pokaz stare parametry (вывод старых параметров)
667.         set(handles.edit1,'String',num2str(r_fx)) %f_x
668.         set(handles.edit2,'String',num2str(r_fy)) %f_y
669.         set(handles.edit3,'String',num2str(r_df)) %f_df
670.         set(handles.edit4,'String',num2str(r_du)) %f_du
671.
672.         info=imfinfo(plik);
673.         metryka{1}=strcat('X: ',num2str(info.Width));
674.         metryka{2}=strcat('Y: ',num2str(info.Height));

```

```

675.         metryka{3}=strcat('Bityw/piksel: ',num2str(info.BitDepth));
676.         metryka{4}=strcat('Maksimum: ',num2str(max(max(obraz))));
677.         metryka{5}=strcat('Minimum: ',num2str(min(min(obraz))));
678.         metryka{7}=strcat('Data: ',info.FileModDate);
679.         metryka{6}=strcat('Rozmiar: ',num2str(info.FileSize));
680.
681.         set(handles.text6,'String',metryka);
682.
683.         cd (roboczy)
684.
685.         wyswietlanie(handles) %pokaï
686.         axes(handles.axes3) %pokaï widmo
687.         imshow(abs(wynik2D),[]) %bo juï jest
688.         end
689.
690.         % --- Executes on button press in checkbox2.
691.         function checkbox2_Callback(hObject, eventdata, handles)
692.
693.
694.         % --- uïyj poprzednich parametryw obszaru, przelicz-(используйте предыдущие параметры
        области, конвертируйте)---
695.         function pushbutton10_Callback(hObject, eventdata, handles)
696.         poprzednie(handles)
697.
698.         function edit1_Callback(hObject, eventdata, handles)
699.
700.         % --- Executes during object creation, after setting all properties.
701.         function edit1_CreateFcn(hObject, eventdata, handles)
702.         if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
703.             set(hObject,'BackgroundColor','white');
704.         end
705.
706.         function edit2_Callback(hObject, eventdata, handles)
707.         %
708.
709.         % --- Executes during object creation, after setting all properties.
710.         function edit2_CreateFcn(hObject, eventdata, handles)
711.         if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
712.             set(hObject,'BackgroundColor','white');
713.         end
714.
715.         function edit3_Callback(hObject, eventdata, handles)
716.
717.         % --- Executes during object creation, after setting all properties.
718.         function edit3_CreateFcn(hObject, eventdata, handles)
719.         if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
720.             set(hObject,'BackgroundColor','white');
721.         end
722.
723.         function edit4_Callback(hObject, eventdata, handles)
724.
725.         % --- Executes during object creation, after setting all properties.
726.         function edit4_CreateFcn(hObject, eventdata, handles)
727.
728.         if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
729.             set(hObject,'BackgroundColor','white');
730.         end
731.
732.         % --- Koniec pracy-----
733.         function pushbutton11_Callback(hObject, eventdata, handles)
734.
735.         try
736.         matlabpool close
737.         catch
738.             t='Nie wykryto Parallel Processing Toolbox, przetwarzanie rywnolegic wyinlczone.'
739.         end

```

```

740.
741.     clear all
742.     clear global all
743.     close(gcf)
744.
745.     function edit5_Callback(hObject, eventdata, handles)
746.
747.     % --- Executes during object creation, after setting all properties.
748.     function edit5_CreateFcn(hObject, eventdata, handles)
749.
750.     if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
751.         set(hObject,'BackgroundColor','white');
752.     end
753.
754.     % ---Dodawanie maksimuw do tabelki----(Добавление максимумов в таблицу)--
755.     function pushbutton12_Callback(hObject, eventdata, handles)
756.
757.     fx=str2num(get(handles.edit1,'String'))    %f_x
758.     fy=str2num(get(handles.edit2,'String'))    %f_y
759.
760.     tablica=get(handles.uitable1,'Data')
761.
762.     wiersze=size(tablica,1)
763.
764.     for x=1:wiersze
765.         tablica{x,3}='false';
766.     end
767.
768.     tablica=vertcat(tablica,{fx,fy,true}) %dodaj nowy aktywny
769.     set(handles.uitable1,'Data',tablica);
770.
771.     % --- skasuj tabelk-----
772.     function pushbutton13_Callback(hObject, eventdata, handles)
773.     set(handles.uitable1,'Data',[]); %czyszczenie tablicy maksimow (очистение таблицы)
774.
775.     % --- edyta tabelki----(правка таблицы)
776.     function uitable1_CellEditCallback(hObject, eventdata, handles)
777.     %zaznaczenie po kliknikciu (указание по нажатию)
778.     tablica=get(handles.uitable1,'Data');
779.     wiersze=size(tablica,1)
780.     selection = eventdata.Indices(:,1);
781.     for x=1:wiersze
782.         tablica{x,3}=0
783.     end
784.     tablica{selection,3}='true'
785.     set(handles.uitable1,'Data',tablica)
786.
787.     % --- wybrana pozycja (выбранная позиция)
788.     function uitable1_CellSelectionCallback(hObject, eventdata, handles)
789.
790.     % --- %--nowa siatka-czkstoъci---(новая сетка частот)-----
791.     function pushbutton14_Callback(hObject, eventdata, handles)
792.
793.     global wynik2D;
794.     global z;
795.     global amplituda;
796.     global faza;
797.
798.     z=0;
799.
800.     %dodac odczyt niezaleinie z tabeli i z okna fx i r_fx (нез. Читка из таблицы и из
    окна)
801.     r_fx=str2num(get(handles.edit1,'String'))    %f_x
802.     r_fy=str2num(get(handles.edit2,'String'))    %f_y
803.     r_df=str2num(get(handles.edit3,'String'))    %df
804.     r_du=str2num(get(handles.edit4,'String')) %du
805.
806.     %odczyt czkstoъci z tablicy
807.     tablica=get(handles.uitable1,'Data');

```

```

808.     wiersze=size(tablica,1);
809.
810.     for x=1:wiersze
811.         a=tablica{x,3};
812.         if(a~=0)
813.             wsp(1)=tablica{x,1};
814.             wsp(2)=tablica{x,2}
815.         end
816.     end
817.
818.     fx=wsp(1)
819.     fy=wsp(2)
820.
821.     axes(handles.axes3)
822.     imshow(abs(wynik2D),[])
823.     kolory(handles);
824.
825.     clear ww kk
826.
827.     %--obliczanie czkstoŃci bezwzglkdnej (od Ńrodka)
828.     kk=round((fx-r_fx)/r_du+r_df/r_du+1)
829.     ww=round((fy-r_fy)/r_du+r_df/r_du+1)
830.     %
831.
832.     %wylicz faze i amplitude, pokaz rozmiary macierzy (расчет фазы и амплитуды вывод
размеров матрицы)
833.     [lw lk]=size(wynik2D);
834.
835.     p=max(max(abs(wynik2D)))
836.     amplituda=p;
837.     faza=angle(wynik2D(ww,kk))*180/pi;
838.
839.     set(handles.text13,'String',['Amplituda: ' num2str(amplituda) ' Faza: '
num2str(faza) '°'])
840.     set(handles.text15,'String',[num2str(lw) 'x' num2str(lw) ' maksimum: (' num2str(kk)
',' num2str(ww) ')']]);
841.
842.     hold on;plot(kk,ww,'ob')
843.
844.     %nowa macierz
845.     try
846.         for xw=-r_df+1:r_df-1
847.             for yk=-r_df+1:r_df-1
848.                 z(xw+r_df,yk+r_df)=wynik2D(ww+(xw/r_du),kk+(yk/r_du));
849.                 plot(kk+(yk/r_du),ww+(xw/r_du),'*g')
850.             end
851.         end
852.     hold off
853.
854.     catch
855.         a='blad'
856.         hold off %prawidowe rysowanie gdy bin№d
857.     end
858.
859.     %wyi№cz szukanie maksimum aĩ do przeliczenia fft
860.     set(handles.pushbutton8,'Enable','off')
861.     set(handles.pushbutton3,'Enable','off')
862.
863.     % --- selektor du-----
864.     function popupmenu3_Callback(hObject, eventdata, handles)
865.
866.     ktory=get(handles.popupmenu3,'Value')
867.
868.     switch ktory
869.         case 1
870.             set(handles.edit4,'String','1')
871.         case 2
872.             set(handles.edit4,'String','0.5')
873.         case 3
874.             set(handles.edit4,'String','0.2')

```



```

875.         case 4
876.             set(handles.edit4,'String','0.1')
877.         case 5
878.             set(handles.edit4,'String','0.05')
879.         case 6
880.             set(handles.edit4,'String','0.02')
881.         case 7
882.             set(handles.edit4,'String','0.01')
883.     end
884.
885.     % --- Executes during object creation, after setting all properties.
886.     function popupmenu3_CreateFcn(hObject, eventdata, handles)
887.         if ispc && isequal(get(hObject,'BackgroundColor'),
888.             get(0,'defaultUicontrolBackgroundColor'))
889.             set(hObject,'BackgroundColor','white');
890.         end
891.
892.     % --- przenień z tablicy do okna-(переход от таблицы к окну)---.
893.     function pushbutton15_Callback(hObject, eventdata, handles)
894.
895.         tablica=get(handles.uitable1,'Data');
896.         wiersze=size(tablica,1)
897.
898.         for x=1:wiersze
899.             a=tablica{x,3}
900.             if(a~=0)
901.                 wsp(1)=tablica{x,1}
902.                 wsp(2)=tablica{x,2}
903.             end
904.         end
905.
906.         fx=wsp(1)
907.         fy=wsp(2)
908.
909.         %wstaw nowe wartości z tabelki (вставить новые значения из таблицы)
910.         set(handles.edit1,'String',num2str(fx))    %f_x
911.         set(handles.edit2,'String',num2str(fy))    %f_y
912.
913.
914.     % --- kliknięcie na listk -> FFT - (нажатие на список → FFT)-----
915.     function listbox1_Callback(hObject, eventdata, handles)
916.         global plik;
917.         global roboczy;
918.         global nazpliku;
919.
920.         global ostatni;
921.         global ktory;
922.         global pakiet;
923.         global interferogram;
924.         global gdzie;
925.
926.         cd (gdzie)
927.         gdzie
928.
929.         ktory=get(handles.listbox1,'Value');
930.         lista=get(handles.listbox1,'String');
931.         klik=get(gcf,'SelectionType');
932.
933.         if (strcmp(klik,'open')==0) %jeden klik, tylko zaiaduj i wujwiewietl (zawsze sie
934.             wykonaj!) (одним кликом, только загрузить и отобразить (это всегда будет сделано!))
935.
936.             nazwa=char(plik{1,ktory}) %nazwa wybranego
937.             policz(ktory,nazwa,handles); %zaiaduj i pokaż
938.
939.         else %dwa klik - policz!!!! (двойной клик - расчет!)
940.
941.             %autozapisywanie klatki (автосохранение клетки)
942.             sprawdzaj=get(handles.checkbox1,'Value');
943.             if (sprawdzaj==1)

```

```

943.         zapiszklatke(handles);
944.     end
945.
946.     nazwa=char(plik(1,ktory));
947.     set(handles.text33,'String',nazwa); %aktywny plik
948.
949.
950.     c=fix(clock);
951.     godzina=datestr(c,13);
952.     komunikat=[godzina ' Przeliczono plik ' nazwa];
953.     set(handles.text13,'String',komunikat);
954.
955.     %przygotuj nazwe bez rozszerzenia (имя без расширения)
956.     nazpliku=strrep(nazwa,'.bmp','');
957.     nazpliku=strrep(nazpliku,'.BMP','');
958.     nazpliku=strrep(nazpliku,'.tif','');
959.     nazpliku=strrep(nazpliku,'.tiff','');
960.     nazpliku=strrep(nazpliku,'.TIF','');
961.     nazpliku=strrep(nazpliku,'.jpg','');
962.     nazpliku=strrep(nazpliku,'.JPG','');
963.     ostatni=ktory; %ostatnio przeliczono plik numer (номер последнего
преобразованного файла)...
964.
965.     cd (roboczy) %wracamy do katalogu z programem
966. end
967.
968. %--wczytaj i przelicz-(вставить и пересчитать)---
969. function policz(ktory,nazwany,handles)
970.
971.     global interferogram;
972.     global mapa;
973.     global gdzie;
974.     global roboczy;
975.
976.     global plik;
977.     global gdzie;
978.     global roboczy;
979.     global sciezka;
980.     %
981.     global obraz;
982.     global obraz2;
983.     global mapa;
984.     global wynik2D_1;
985.     global wynik2D;
986.
987.     global p_obraz %poprzedni
988.     global p_wynik2D_1
989.     global p_B;
990.     global p_fi1;
991.     global fi1;
992.     global p_sciezka;
993.
994.     global B;
995.
996.     cd (gdzie) %idziemy do katalogu z plikiem
997.     pwd
998.
999.     set(handles.text13,'String','Wczytujk plik...'); %komunikat
1000.    set(handles.listbox1,'Value',ktory); %podswietl aktualny
1001.
1002.    naz=nazwany %peina nazwa wczytanego pliku
1003.
1004.    p_obraz=obraz2; %poprzedni
1005.    p_wynik2D_1=wynik2D_1; %poprzedni
1006.    p_B=B; %zapamiktaj poprzedni amplituda
1007.    p_fi1=fi1; %zapamiktaj poprzedni faza
1008.    p_sciezka=sciezka; %zapamietaj poprzednie sciezka
1009.    zapamietaj(handles) %zapisz parametry obszaru do pyl tekstowych
1010.
1011.    wczytany=imread(nazwany);

```

```

1012.     rr=size(wczytany);
1013.     r=size(rr);
1014.
1015.     if r(2)==3
1016.         obraz=double(rgb2gray(wczytany));
1017.         dohist=rgb2gray(wczytany); %tylko do histogramu
1018.     else
1019.         obraz=double(wczytany);
1020.         dohist=wczytany; %tylko do histogramu
1021.     end
1022.
1023.     axes(handles.axes1)
1024.     imshow(obraz,[])
1025.
1026.     %--histogram--wyswietlanie--konwersja
1027.     axes(handles.axes2)
1028.     [histogramek,x]=imhist(dohist);
1029.     stem(x,histogramek,'Marker','none')
1030.
1031.     set(handles.axes2,'xticklabel','')
1032.     set(handles.axes2,'yticklabel','')
1033.
1034.     axis auto
1035.     axis tight
1036.     %-----
1037.
1038.     info=imfinfo(nazwany);
1039.     metryka{1}=strcat('X: ',num2str(info.Width));
1040.     metryka{2}=strcat('Y: ',num2str(info.Height));
1041.     metryka{3}=strcat('Bityw/piksel: ',num2str(info.BitDepth));
1042.     metryka{4}=strcat('Maksimum: ',num2str(max(max(obraz))));
1043.     metryka{5}=strcat('Minimum: ',num2str(min(min(obraz))));
1044.     metryka{7}=strcat('Data: ',info.FileModDate);
1045.     metryka{6}=strcat('Rozmiar: ',num2str(info.FileSize));
1046.
1047.     set(handles.text6,'String',metryka);
1048.
1049.     obraz=double(obraz);
1050.
1051.     c=fix(clock);
1052.     godzina=datestr(c,13);
1053.     komunikat=[godzina ' Wczytano plik ' naz];
1054.     set(handles.text13,'String',komunikat);
1055.
1056.     set(handles.text13,'String','Oczekiwanie');
1057.
1058.     mapa=colormap;
1059.     cd (roboczy) %wracamy do katalogu z programem
1060.
1061.     %-----FFT-----
1062.
1063.     function transformata(handles)
1064.
1065.     global obraz;
1066.     global obraz2;
1067.     global wynik2D;
1068.     global r_fx;
1069.     global r_fy;
1070.     global r_df;
1071.     global r_du;
1072.
1073.     set(handles.pushbutton8,'Enable','on') %uaktywnij z powrotem manualne szukanie
maksimum
1074.     set(handles.pushbutton3,'Enable','on')
1075.
1076.     %--parametry robocze--
1077.     r_fx=str2num(get(handles.edit1,'String')) %f_x
1078.     r_fy=str2num(get(handles.edit2,'String')) %f_y
1079.     r_df=str2num(get(handles.edit3,'String')) %df
1080.     r_du=str2num(get(handles.edit4,'String')) %du

```

```
1081.
1082.     if (get(handles.checkbox2,'Value')==1) %czy usunl% rzlnd zerowy?
1083.         obraz2=obraz-mean(obraz(:));
1084.     else
1085.         obraz2=obraz;
1086.     end
1087.
1088.     set(handles.text13,'BackgroundColor',[1 0.7 0.7]);
1089.     set(handles.text13,'String','Obliczam...'); drawnow
1090.     tic
1091.         wynik2D=zoomFFT2(obraz2,r_fx,r_fy,r_df,r_du);
1092.     a=toc;
1093.     set(handles.text13,'BackgroundColor',[0.7 1 0.7]);
1094.     set(handles.text13,'String',['Policzono w ' num2str(a) ' sekund. ']);drawnow
1095.
1096.     axes(handles.axes3)
1097.     imshow(abs(wynik2D),[])
1098.     kolory(handles);
1099.     %drawnow
1100.
1101.     % --- Executes during object creation, after setting all properties.
1102.     function listBox1_CreateFcn(hObject, eventdata, handles)
1103.         if ispc && isequal(get(hObject,'BackgroundColor'),
1104.             get(0,'defaultUicontrolBackgroundColor'))
1104.             set(hObject,'BackgroundColor','white');
1105.         end
```

Додаток Б. Комп'ютерна презентація

Східноукраїнський національний університет імені
Володимира Даля
Факультет інформаційних технологій та електроніки
Кафедра комп'ютерних наук та інженерії

кваліфікаційна робота освітнього ступеня магістр
123 «Комп'ютерна інженерія»

**«Програмно-алгоритмічні засоби комп'ютерного
моделювання дифракції та обробки експериментальних
зображень»**

Магістрант групи *KI-173м*
Рязанцев Андрій Олександрович
Керівник доц. Кривуля Г.Ф.

1

Актуальність теми

Тема магістерської роботи присвячена дослідженню методів моделювання фізичних явищ та розробці програмно-алгоритмічного засобу комп'ютерного моделювання та обробки експериментальних зображень.

Цей засіб необхідний для визначення зовнішніх та внутрішніх параметрів світлового поля сукупність яких дозволяє презентувати повний опис пучка, що характеризує метрологічну спроможність сингулярного скелетону: похибку вимірювань та динамічний діапазон. Розроблення відповідного ПАЗ сприятиме розвитку прикладної оптики, зокрема метрології та мікроскопії.

2

Мета дослідження

Дослідити методи моделювання дифракції та інтерференції, обробки оптичних зображень, створити інформаційні моделі опису теоретичного та експериментального дослідження, реалізувати програмно-алгоритмічний засіб для первинної та додаткової обробки оптичних зображень.

Об'єкт дослідження

Процес розробки програмно-алгоритмічного засобу для моделювання дифракції та обробки інтерференційних картин.

Предмет дослідження

Методи чисельного інтегрування, Фур'є перетворення та математичні моделі розробки програмно-алгоритмічного засобу для моделювання дифракції та обробки інтерференційних картин.

Наукова новизна

В результаті виконання дипломної роботи були отримані наступні результати:

- Вперше розроблено в математичному пакеті MATLAB програмно-алгоритмічний засіб для обробки оптичних зображень, завдяки якому визначені координати оптичних вихорів та їх морфологія з великою точністю, що дозволяє підвищити метрологічну спроможність інтерференційних методів та мікроскопії.
- Вперше з використанням ПАЗ визначена траєкторія та внутрішня еліптична структура серцевини оптичного вихору..

Фазова сингулярність

Амплітуда

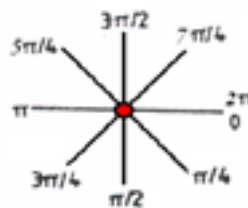
$$A = u + iv$$

$$A=0 \Rightarrow u=0, v=0$$

Фаза

$$\Phi = \arctg\left(\frac{v}{u}\right)$$

$$\Phi = \arctg\left(\frac{0}{0}\right)$$



$$\oint d\Phi = 2\pi m$$

$$m = \pm 1, \pm 2, \dots$$

m – топологічний заряд

Фаза не визначена – це є точкою сингулярності. Точки утворюють криву, за якою можна визначити природу еволюції пучка в залежності від відстані. Сукупність кривих з невизначеною фазою формують сингулярний скелетон пучку. Енергія рухається в пучку перпендикулярно до хвильового фронту - гелікоїду.



Диполь оптичного вихору

Подвійний симетричний фазовий клин (ПСФК)

$$\Phi(x, y, 0) = \begin{cases} 2\pi - K \cdot x, & y > 0 \\ \pi + K \cdot x, & y \leq 0. \end{cases} \quad (1)$$

Математична формула подвійного фазного клину з тангенсом нахилу клинів K . Значення K залежить від радіуса пучка на відстані положення маски та необхідної кількості оптичних вихорів у полі.

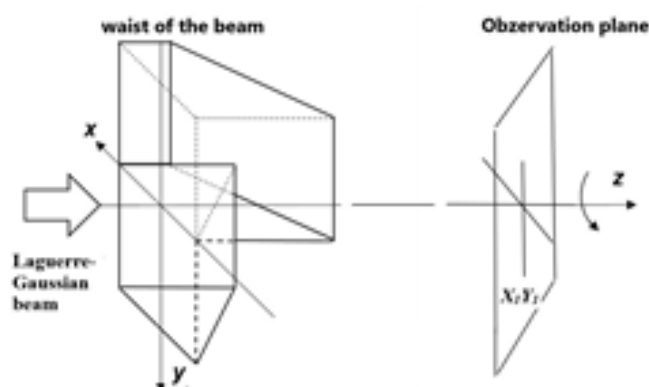
У кожній точці уздовж Y -осі, включаючи початок координат, різниця фази між півплощинами дорівнює π → деструктивна інтерференція та утворення ОВ.

Джерелом світла є гаусовий пучок, що поширюється вздовж осі Z .

$$E(\rho, z) = E_G \frac{w_0}{w} \exp\left(-\frac{\rho^2}{w^2}\right) \exp i\left(kz + \frac{k\rho^2}{2R} - \arctan\left(\frac{z}{z_R}\right)\right) \quad (2)$$

E_G – амплітудний параметр, k – хвильове число, w_0 – поперечний розмір пучка в перетяжці (параметр перетяжки), $z_R = kw_0^2/2$ – довжина Релея, $w = w_0(1+z^2/z_R^2)^{1/2}$ – поперечний розмір пучка на відстані z від перетяжки, $R(z) = z(1+z_R^2/z^2)$ – радіус кривини хвильового фронту на відстані z від перетяжки у перерізі пучка

Подвійний симетричний фазовий клин (ПСФК)



Інтеграл Кірхгофа-Френеля

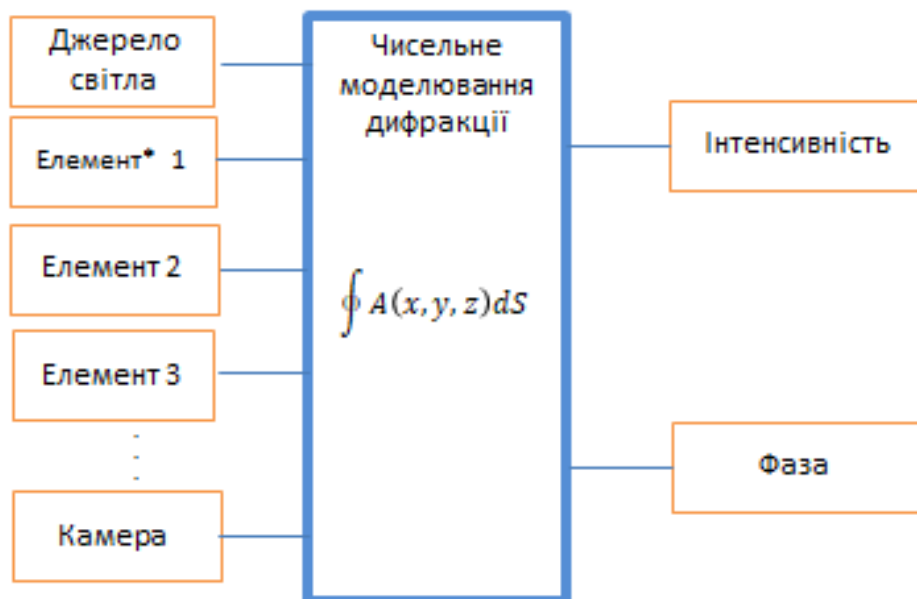
$$A(x, y, z) = C(z) \iint A_0 T(x, y) \exp \left[-ik \frac{(x-x')^2 + (y-y')^2}{2z} \right] dx' dy' \quad (3)$$

$T(x, y)$ – функція фазового клину $\exp(i\Phi(x, y, 0))$

Інформаційна модель опису теоретичного дослідження дифракції

Вхідні параметри

Вихідні параметри



*Елемент – модулятор фази та/або амплітуди

Алгоритм реалізації моделі



Експериментальні дослідження ланцюгу оптичних вихорів

Схема експериментальної установки (а) :

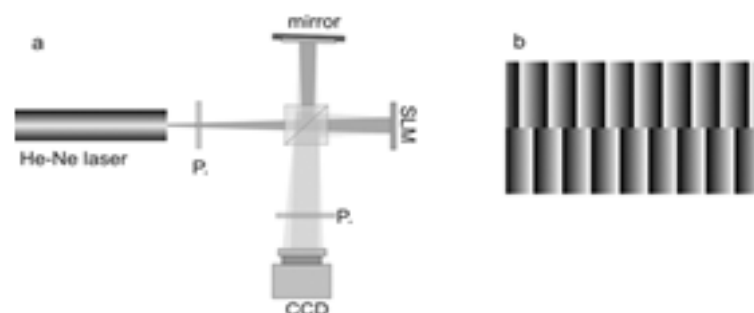


Схема має :

- Джерело світла He-Ne лазер (633 нм, $w_0 = 0.2$ мм)
- P1 , P2 – поляризатори для зміни інтенсивності пучку
- спліттер (дільник пучку)
- дзеркало
- просторовий модулятор світла (SLM)
- Камера (CCD) (розмір пікселя 3,45 мікрона, роздільна здатність 2200x3000)
- (b) – фазова маска на ПМС для генерації ланцюгу оптичних вихорів

Інформаційна модель опису експериментального дослідження дифракції

Вхідні параметри

Вихідні параметри



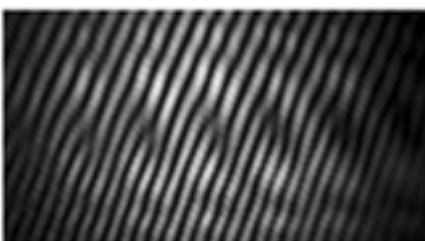
*Елемент – модулятор фази та/або амплітуди

Експериментальні дослідження ланцюгу оптичних вихорів

c



d

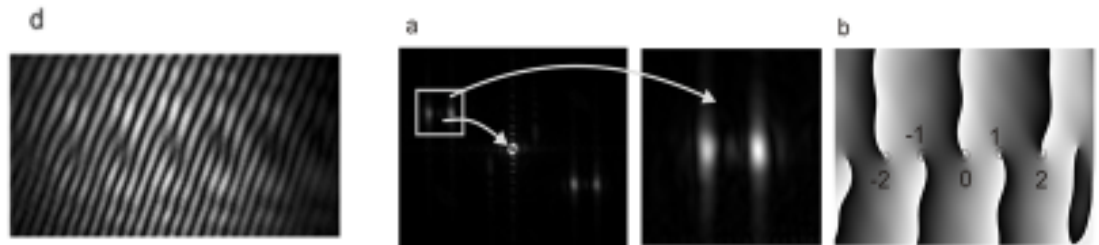


Зображення інтенсивності показує темні області, в яких знаходяться ОВ, розподілені вздовж лінії (c). На інтерферограмі (d) можна побачити вилки, характерні для вихрової структури.

В експерименті неможливо отримати фазу пучка. Щоб її отримати необхідне розробити програмно-алгоритмічний засіб для відновлення фази з інтерференційної картини. Спираючись на зроблений огляд методів обробки зображень мною був обраний метод перетворення Фур'є. Перевагою методу Фур'є є не тільки можливість відновити фазу пучка, але й визначити координати оптичних вихорів з високою точністю.

Експериментальні дослідження ланцюгу оптичних вихорів за допомогою просторового модулятора світла

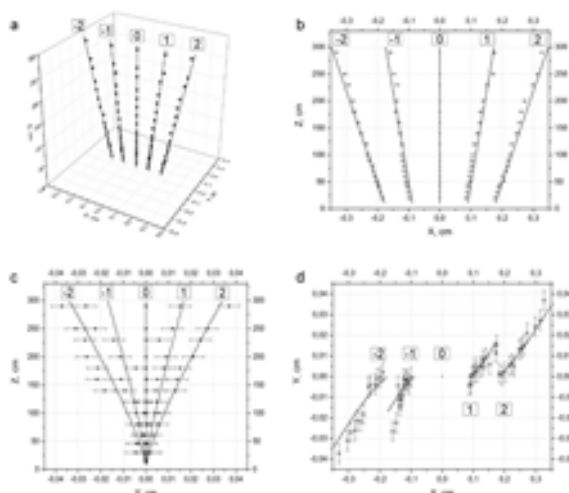
Визначення координат оптичних вихорів за допомогою перетворення Фур'є:



- Швидке перетворення Фур'є застосовується до виявленої інтерферограми (d)
- перший порядок спектра Фур'є позначено прямокутником, він містить необхідну інформацію і зміщується до позиції нульового порядку
- До нього застосовується зворотнє перетворення Фур'є та отримується відновлена фазова мапа (b)
- Координати ОВ з легкістю та високою точністю можна визначити на фазовій мапі (b)

Порівняльний аналіз теоретичних на експериментальних результатів дослідження ланцюгу оптичних вихорів

Зовнішні особливості ланцюгу оптичних вихорів - скелетон:



3D-скелет поля отриманий з вхідного гаусового пучка, перетворений ПСФК (2) з градієнтом фази $K = 10$ (a). Зображення (b), (c) та (d) являють собою проєкції скелетону на поперечні площини XZ, YZ та XY, відповідно. Напівжирні лінії вказують на теоретичні розрахунки, а маркери – на експериментальні точки.

Морфологія ОВ

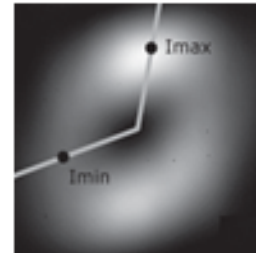
Еліптичність та кут нахилу еліпсу

$$\gamma = b/a \quad (3)$$



Ступінь радіальної симетрії інтенсивності

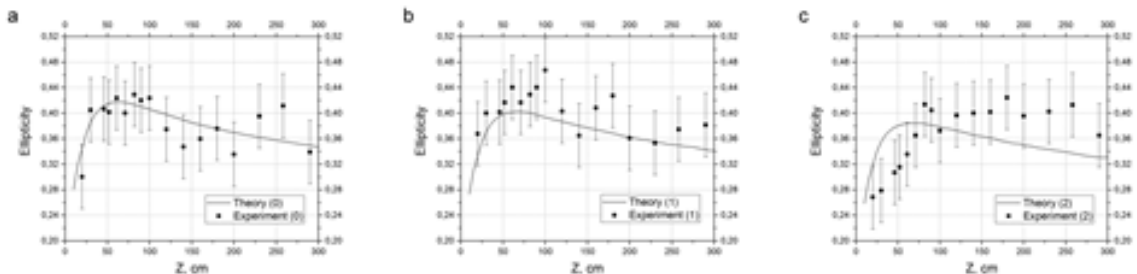
$$\eta = I_{min}/I_{max} \quad (4)$$



- Еліптичність γ зумовлена радіальною асиметрією фазової маски та визначається як відношення малої до великої піввісі еліпса, що проведені через загальний амплітудний контур та центр пучка.
- φ – кут нахилу між великою піввіссю еліпсу та позитивним напрямком осі X.
- Ступінь радіальної симетрії інтенсивності η синтезованого ОВ пучка визначимо, як співвідношення мінімальної інтенсивності в області максимальної інтенсивності пучка з осьовим ОВ на певній відстані z

Порівняльний аналіз теоретичних на експериментальних результатів дослідження ланцюгу оптичних вихорів

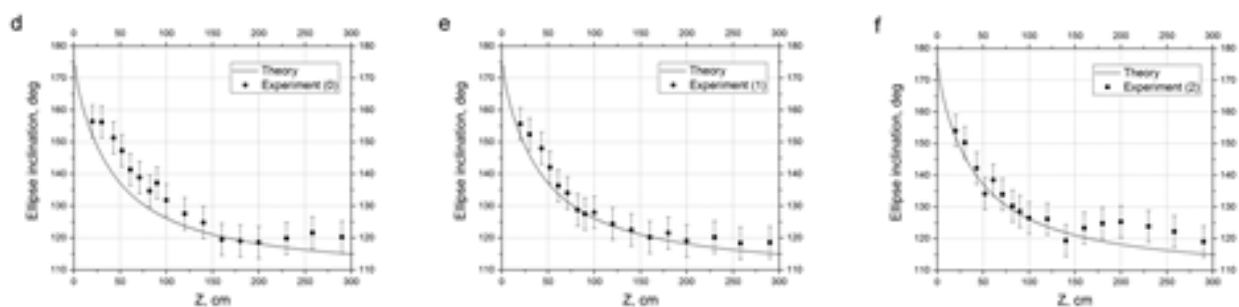
Внутрішні особливості (морфологія) ланцюгу оптичних вихорів :



Залежність еліптичності ОВ від відстані спостереження Z відповідно. Теоретичні криві показані суцільними чорними лініями, а експериментальні дані представлені точками з баром абсолютної похибки вимірювання.

Порівняльний аналіз теоретичних на експериментальних результатів дослідження ланцюгу оптичних вихорів

Внутрішні особливості (морфологія) ланцюгу оптичних вихорів :



Залежність куту нахилу великої осі еліпсу від відстані спостереження Z відповідно. Теоретичні криві показані суцільними чорними лініями, а експериментальні дані представлені точками з баром абсолютної похибки вимірювання.

Апробація

. Матеріали були представлені на міжнародних заходах:

1. Special features of a functional beam splitter: diffraction grating with groove bifurcation/ Aleksandr Y. Bekshaev, Aleksandr I. Karamoch, Ganna M. Khoroshun, Oleksandr I. Ryazantsev, Jan Masajada // Chapter in a book: *Advances in Engineering Research*, 2018, 70p.

2. Chain of optical vortices synthesized by a Gaussian beam and the double-phase-ramp converter/ Khoroshun G.M., Chernykh O., Tatarchenko H.O., S. Sato, Y. Kozawa, A. Popiołek-Masajada M. Szatkowski, W. Lamperska // *OSA Continuum*, Peer Review from 18 November 2018

3. Sensitivity of singular skeleton obtained by a gaussian beam and double phase ramp converter / Khoroshun G.M., Chernykh O., Ryazantsev A.O., S. Sato, Y. Kozawa, A. Popiołek-Masajada, M. Szatkowski, W. Lamperska // VIII International Conference on Optoelectronic Information Technologies "PHOTONICS-ODS 2018", Ukraine, Vinnytsia, VNTU October 2-4, 2018

4. Інформаційні технології лазерного керуванні мікрочастинками/ Рязанцев А., Черних О., Шатковські М., Ламперська В., Хорошун Г., Кривуля Г., Попіолек – Масаяда А, Масаяда Я., Сато С., Козава Ю. // Форум "ІТ-ІДЕЯ", Сєвєродонецьк, 7 грудня, 2018

5. Data object design of the intensity pattern for controlling micro- and nanoparticles/ A. Ryazantsev, T. Wrzeszcz, A. Khoroshun, J. Masajada // Всеукраїнська конференція «Майбутній науковець», Сєвєродонецьк, 14 грудня, 2018

• Висновки

1. Проаналізовано існуючі методи моделювання та математичні моделі дифракції, інтерференції.

2. Створено інформаційні моделі опису теоретичного та експериментального дослідження дифракції та інтерференції

3. Проаналізовано відомі методи обробки зображень

4. Розроблено алгоритм реалізації моделі первинної обробки зображення

5. Розроблено алгоритм реалізації моделі додаткової обробки зображення за допомогою перетворення Фур'є.

6. Спроектовано та реалізовано програмно-алгоритмічний засіб для первинної та додаткової обробки оптичних зображень

7. Впроваджено програмно-алгоритмічний засіб для використання в оптичній лабораторії Вроцлавського технологічного університету.