

Шифр: Дифракція

**КОМП'ЮТЕРНИЙ АНАЛІЗ ДИФРАКЦІЙНИХ КАРТИН,
ОТРИМАНИХ МЕТОДОМ ПРОСВІЧУЮЧОЇ
ЕЛЕКТРОННОЇ МІКРОСКОПІЇ**

Зміст

Вступ.....	4
1. Використання графічного програмного інструменту LabVIEW для автоматизації аналізу дифракційних картин, отриманих методом просвічуючої електронної мікроскопії.....	5
1.1 Послідовність розшифровки електронограми.....	6
1.2 Функціональні блоки лицьової панелі.....	8
1.3 Блок-діаграма.....	9
1.4 Вірогіднісний алгоритм пошуку центру.....	12
1.4.1 Сканування зображення.....	12
1.4.2 Обробка та фільтрація зрізу дифракційної картини.....	14
1.4.3 Визначення центру зрізу.....	16
1.4.4 Визначення дійсного центру дифракційної картини.....	18
1.5 Пошук дифракційних кілець.....	19
1.5.1 Особливості аналізу зображення за секторами.....	19
1.5.2 Вибір дифракційних кілець.....	22
2. Обговорення результатів досліджень.....	24
Висновки.....	26
Список літератури.....	27

ВСТУП

Сучасний фізичний експеримент потребує від вчених та інженерів проектування і впровадження складних програмно – апаратних систем автоматизації експериментальних досліджень. Особливу увагу в цьому напрямі слід приділяти системам аналізу отриманих в результаті експерименту даних. Актуальною і сучасною, на сьогоднішній день, залишається проблема побудови надійних високоточних систем розпізнавання та аналізу зображень з подальшим отриманням результатів нанорозмірної точності та низькою імовірністю виникнення похибки.

Мета роботи – розробка програмного забезпечення для розпізнавання та аналізу дифракційних картин з нанорозмірних плівкових систем, отриманих методом просвічуючої електронної мікроскопії.

Основними задачами наукової роботи є програмування оригінальних індивідуальних алгоритмів з розпізнавання дифракційних картин, власних алгоритмів фільтрації та математичної обробки зображень, автоматичного розрахунку типу та параметрів кристалічних ґраток.

Результати роботи розробленого програмного забезпечення порівнювалися з табличними та розрахованими вручну значеннями параметрів кристалічних ґраток для дифракційних картин різноманітних матеріалів.

1. Використання графічного програмного інструменту LabVIEW для автоматизації аналізу дифракційних картин, отриманих методом просвічуючої електронної мікроскопії

Компанія National Instruments та розроблений нею програмний продукт LabVIEW являється ідеальним інструментом для проектування і створення сучасного надійного програмного забезпечення складних апаратно – експериментальних систем, що дозволяють вченим та інженерам з усього світу швидко та надійно проводити вимірювання фізичних величин різного ступеня складності з подальшим отриманням результатів високої імовірності і точності[2].

Суттєвою перевагою запропонованого програмного інструменту є модульний підхід до створення програмно-апаратних систем, систем реального часу, систем розпізнавання та аналізу зображень, наявність великої кількості написаного коду, розробленого програмного забезпечення, апаратних систем високого ступеня інтеграції і швидкодії для вирішення задач, що ставлять перед вченими та інженерами проблеми сучасної науки і техніки. Обумовлена на даному етапі розвитку науки потреба в швидкій розробці апаратно-експериментальних, розрахункових систем високої надійності та точності, створення програмного забезпечення для керування, спостереження та аналізу сучасного фізичного експерименту практично виключає можливість використання інших програмних інструментів.

Широкий спектр потенціальних можливостей програмного інструменту LabVIEW забезпечується різноманітними комплектаціями, а також набором модулів та інструментів LabVIEW[3].

Один із цих модулів було вдало використано при розробці програмного забезпечення, а саме модуль технічного зору LabVIEW IMAQ Vision Development Module.

Програмний код, що створюється в LabVIEW має вигляд окремих блоків з'єднаних між собою провідниками. Тому область програмного коду прийнято

називати блок-діаграмою. В LabVIEW використовується метод обробки потоку даних (data flow). Згідно цього методу виконання окремого блоку розпочинається лише в тому випадку, коли на його входи прийшли всі необхідні вхідні дані. Потік даних рухається від блока до блока за причинно-наслідковим зв'язком в ланцюзі обробки сигналу. Таким чином, на відміну від текстових мов програмування де порядок виконання програми визначається послідовністю інструкцій, в LabVIEW обробка потоку даних керується самим потоком даних і залежить від самих даних (data dependent). Принцип потоку даних LabVIEW спрощує розробку багатопоточних і багатозадачних програм[4].

1.1 Послідовність розшифровки електронограми

Здійснюючи будь – які розрахунки потрібно чітко усвідомлювати послідовність дій які необхідно виконати для отримання бажаного правильного результату. У випадку з дифракцією, результатом аналізу є інформація про параметри кристалічної решітки досліджуваного матеріалу, а саме його міжатомну відстань.

Для отримання дифракційної картини в лабораторній практиці використовуються електронографи або просвічуючі електронні мікроскопи (ПЕМ) в режимі дифракції. Методика розшифровки отриманих електронограм від матеріалів кубічної сингонії базується на використанні приведених співвідношень.

Перше співвідношення визначає сталу приладу C . В розробленому програмному забезпеченні для визначення сталої приладу використовується аналіз дифракційної картини від еталона у якого є відомими міжплощинні відстані d_{hkl} :

$$C = d_{hkl} \cdot D_{hkl},$$

де D_{hkl} – діаметри відповідних кілець еталона.

Для розшифровки електронограм від невідомого зразка та визначення типу кристалічної ґратки, необхідно скористатись співвідношеннями [1]:

$$\frac{1}{d_{hkl}^2} = \frac{h^2 + k^2 + l^2}{a_{hkl}^2},$$

$$\frac{1}{d_{h_1k_1l_1}^2} : \frac{1}{d_{h_2k_2l_2}^2} : \frac{1}{d_{h_3k_3l_3}^2} \dots = (h_1^2 + k_1^2 + l_1^2) : (h_2^2 + k_2^2 + l_2^2) : (h_3^2 + k_3^2 + l_3^2) \dots$$

Також при визначенні типу кристалічної ґратки враховуються співвідношення міжплощинних відстаней, що представляють собою послідовності чисел:

ОЦК, ПК: 1; 2; 3; 4; 5; 6; 8; 9;...

ГЦК: 1; 1.33; 2.66; 3.67; 4; 5.33;...

Визначення міжатомної відстані відбувається за наступною формулою [1]:

$$a_{hkl} = d_{hkl} \cdot \sqrt{h^2 + k^2 + l^2}.$$

Використовуючи розроблене програмне забезпечення, для отримання значення параметра кристалічної ґратки потрібно пройти через 2 етапи, кожен із яких в свою чергу складається з, майже, однакової послідовності операцій (рис. 1.1).

Такий підхід значно спрощує розуміння процесу аналізу дифракційної картини, робить програмне забезпечення зрозумілим та простим у використанні і дозволяє швидко отримувати розраховані значення параметрів кристалічних ґраток у діапазоні високої нанорозмірної точності.

Для отримання коректного результату необхідно мати дифракційні картини від еталону і від досліджуваного зразка. На першому етапі необхідно відкрити в програмі зображення дифракційної картини еталону та виконати послідовність дій приведену на рисунку 1.1.

Результатом виконання першого етапу аналізу дифракційної картини є інформація про сталу приладу, що є необхідним параметром для початку

виконання наступного етапу аналізу. Слід зазначити, що переходячи до другого етапу, потрібно змінити зображення еталона на дифракційну картину досліджуваного матеріалу. Після виконання такої ж послідовності операцій, як і у випадку з аналізом еталона, програма визначить шуканий параметр кристалічної ґратки.



Рис.1.1. – Послідовність аналізу дифракційної картини

1.2 Функціональні блоки лицьової панелі

Управління функціональними можливостями розробленого програмного забезпечення здійснюється за допомогою елементів управління розміщених на лицьовій панелі. Основною метою при розробці лицьової панелі було створення зручного і зрозумілого для користувача інтерфейсу. Умовно його можна поділити на три функціональних блоки.

Перший блок складається з елементів управління, що надають користувачеві можливість гнучко задавати параметри аналізу області зображення, отримувати результати у вигляді графіків гістограм і таблиць. Отримані результати відображаються в окремих вікнах, що за потребою викликаються користувачем. Також в цьому блоці розміщені кнопки, кожна з яких відповідає за певну функцію по роботі з областю відображення дифракційної картини. Усі створені програмні функції поділяються на основні та

допоміжні. Основними є функції які реалізують виконання операцій в головній послідовності аналізу дифракційної картини. В свою чергу призначенням допоміжних функцій є підготовка параметрів для належного виконання програмних алгоритмів, покладених в основу операцій аналізу дифракційного зображення. В програмі забезпечена можливість роботи як в ручному, так і в автоматичному режимах. (рис. 1.2, поз. 1).

Призначення другого блоку полягає у контролі та відображенні поточних даних, отриманих у результаті виконання певних операцій аналізу дифракційної картини, а саме інформація про сталу приладу, координати розміщення центру, параметри типу кристалічної ґратки і дані про вибрані кільця дифракційної картини. В згаданому блоці також міститься функція задання типу кристалічної ґратки з можливістю редагування індексів Міллера для кожної міжплощинної відстані. (рис. 1.2, поз. 2).

Третій блок містить область відображення дифракційної картини. Даний блок являється джерелом даних для роботи програмних алгоритмів та подальшого розрахунку параметрів кристалічних ґраток (рис. 1.2, поз. 3).

1.3 Блок діаграма

На сьогоднішній день єдиним ефективним способом для швидкого програмування складного програмного забезпечення з великою кількістю функціональних можливостей є використання стейт машин. Перевагами запропонованого підходу являються покладені в їх основу принципи паралельного виконання частин коду та гнучкого задання ходу виконання програмних алгоритмів. Розроблене програмне забезпечення базується саме на використанні одного із варіантів існуючих видів стейт машин, а саме стейт машини на базі черги Queue State Machine (рис.1.3). Відмінною рисою використаного патерну є те, що відслідковування стану елементів управління, відповідальних за певні функції програмного забезпечення, та обробка даних отриманих в ході виконання цих функцій відбувається паралельно. Таким чином, створюється незалежність інтерфейсу користувача від програмного коду[5].

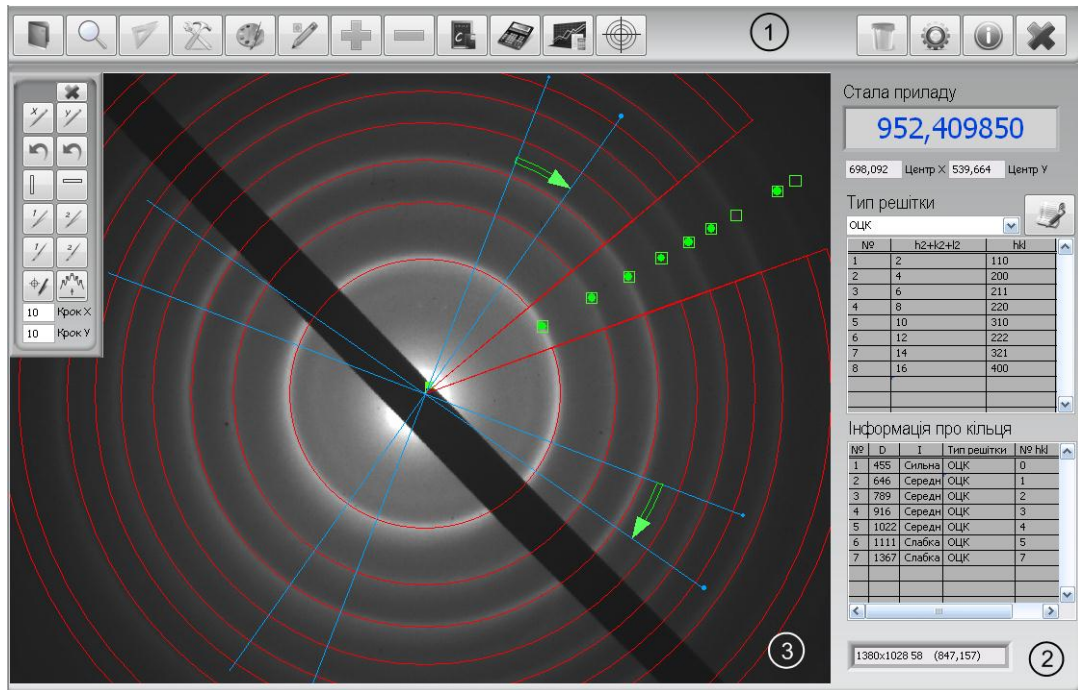


Рис.1.2. – Зовнішній вигляд лицьової панелі розробленого програмного забезпечення

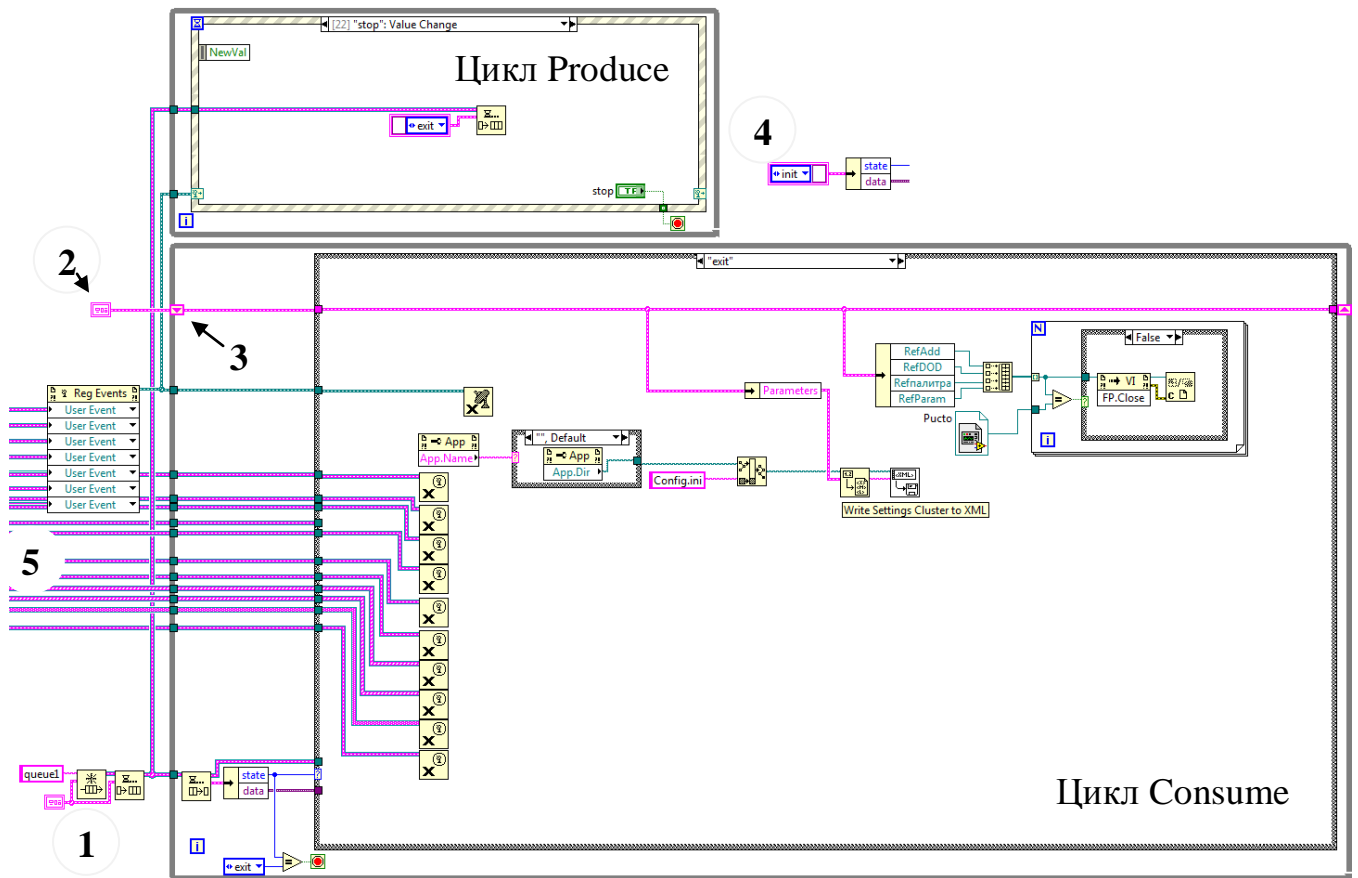


Рис.1.3. – Блок-діаграма розробленого програмного забезпечення

Перш за все, після запуску програмного забезпечення відбувається автоматичне створення черги (рис.1.3, поз.1), що представляє собою посилення на певну область оперативної пам'яті. Черга працює за принципом FIFO (First In First Out) та призначена для обміну інформацією між двома паралельними циклами. Перший цикл produce містить у собі структуру подій (Event Structure), призначення якої полягає у безперервному відслідковуванні подій, пов'язаних з елементами управління лицьової панелі та областю відображення дифракційної картини. Структура подій (Event Structure) – найбільш ефективний спосіб обробки подій інтерфейсу користувача[6]. Другий цикл consume забезпечує обробку даних, що передаються від циклу produce в результаті реагування останнього на виникнення певної події. Для забезпечення швидкої обробки отриманих даних і надання можливості створення гнучких програмних алгоритмів, цикл consume містить вкладену структуру варіантів (Case Structure). Весь програмний код розділено на частини, кожна з яких розміщена на окремій сторінці структури варіантів та відповідає за певну функцію або складову дію одного із алгоритмів. Усі типи даних, що використовуються в програмних алгоритмах визначені в головному кластері (рис.1.3, поз.2) у вигляді констант та передаються між станами Case Structure за допомогою Shift Register (рис.1.3, поз.3). Для переведення циклу consume у потрібний стан необхідно записати в чергу пакет-кластер (рис.1.3, поз.4), що містить назву стану з типом Enum type definition та даних приведених до загального типу variant.

В програмному забезпеченні також вдало реалізовано функціонування віконного інтерфейсу. Кожне окреме вікно представляє собою підпрограму, що здійснює двосторонній обмін даними з головною програмою. Для реалізації передачі інформації використовуються динамічні події User Events (рис.1.3, поз.5). Як і у випадку з чергою, всі динамічні події створюються та реєструються на початку роботи програми. Тип даних створеної динамічної події вказується під час написання коду в залежності від типу даних, яким повинні обмінюватися підпрограми з головною програмою. Початкова ініціалізація та управління

кожним вікном відбувається за допомогою функцій та методів VIsServer LabVIEW.

Використана модель побудови надає можливість швидко вносити зміни на етапах перевірки та пошуку помилок, що по суті є одним із найважливіших і довготривалих ступенів розробки програмного забезпечення.

1.4 Вірогіднісний алгоритм пошуку центру

Згідно послідовності аналізу дифракційної картини приведеної на рисунку 1.1, першою і водночас необхідною умовою подальшого аналізу є операція визначення координат положення центру дифракції. Для вирішення цієї задачі було запрограмовано індивідуальний алгоритм автоматичного пошуку центру дифракційної картини (рис.1.4). Даний алгоритм базується на виборі найбільш вірогідного значення із розрахованих статистичних даних.

Розроблений алгоритм приведений на рисунку 1.4 і складається з 3 етапів:

- паралельного покрокового сканування зображення за двома осями X та Y;
- обробки та фільтрації даних, отриманих на кожному кроці сканування;
- вибору найбільш вірогідного значення центру дифракційної картини.

Наступні розділи присвячені обговоренню принципів роботи кожного із етапів вірогіднісного алгоритму автоматичного пошуку центру дифракційної картини.

1.4.1 Сканування зображення

Метою сканування зображення дифракційної картини є збір та підготовка даних для подальшої фільтрації і визначення центру. Сканування виконується паралельно в ітераційному процесі за двома осями X та Y. На кожному кроці сканування із зображення вирізається й одразу ж аналізується прямокутник, ширина якого задається перед початком сканування.

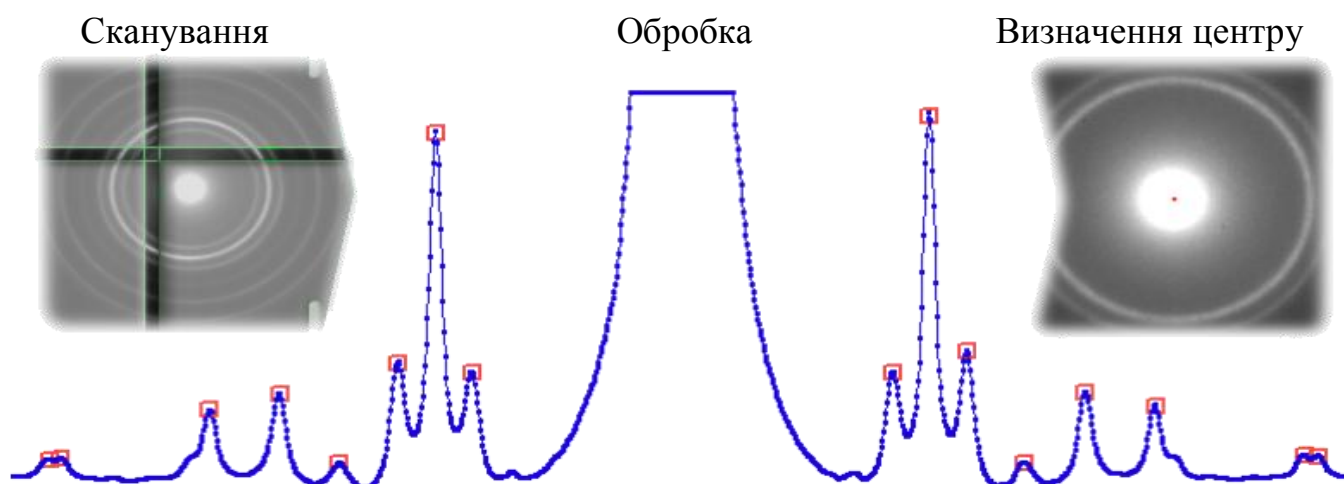


Рис.1.4. – Вірогідніший алгоритм автоматичного пошуку центру дифракційної картини

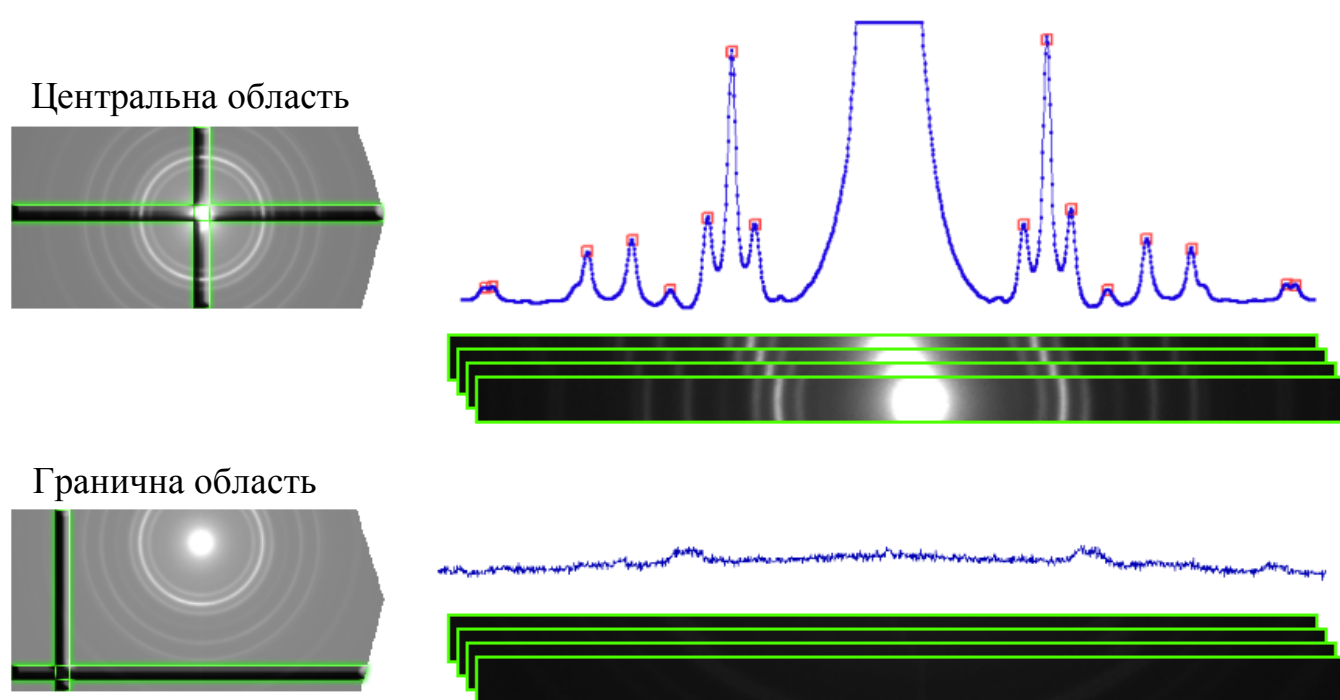


Рис.1.5. – Особливості сканування зображення дифракційної картини

Для вирізання прямокутника із зображення застосовується функція IMAQ Vision – Image to Array. Результатом виконання цієї функції є вирізана частина зображення представлена у вигляді двомірного масиву пікселей, що є рівносильним одномірному масиву ліній. Надалі розраховується середнє арифметичне значення отриманих ліній, яке є придатним для подальшої фільтрації і обробки. Слід також зауважити, що за замовчуванням, лінією в двомірному масиві вважається горизонтальна послідовність точок. Тому при скануванні зображення за віссю Y необхідно використовувати функцію транспонування отриманого масиву Transpose 2D Array.

На рисунку 1.5 приведені дві області зображення та їх зрізи у вигляді графіків, на яких кожному пікселю відповідає певне значення інтенсивності. Аналізуючи ці графіки можна зробити висновок, що найбільш корисна інформація про координати розташування центру зрізу дифракційної картини та положення дифракційних кілець, міститься в центральній області зображення. Гранична область містить в більшій мірі помилкову інформацію.

1.4.2 Обробка та фільтрація зрізу дифракційної картини

Особливістю даного етапу аналізу дифракційної картини є те, що обробка та фільтрація відбувається на кожному кроці сканування. Тобто кожен окремий зріз дифракційної картини проходить обробку. Зрізом дифракційної картини було прийнято називати одномірний масив пікселей, вирізаний із зображення у вигляді лінії вздовж осі X або Y.

Слід зазначити, що цифрове зображення дифракційної картини не є ідеальним, і містить у собі певну кількість шумів та має деякий рівень фону (рис.1.6, поз.1). Беручи до уваги цей факт можна впевнено говорити про наявність шумів та фону на графіках окремих зрізів, що отримуються під час сканування дифракційної картини. Звичайно, ці фактори є небажаними, оскільки вносять суттєву похибку у подальший аналіз дифракційної картини та розрахунок параметрів кристалічної ґратки. Тому основною метою фільтрації являється якомога краще згладжування шумів та ефективне відсічення фону.

Кінцевим результатом роботи такого фільтру повинна бути гістограма з піками, основи яких розміщуються на одній прямій лінії. Для створення такого фільтру було запропоновано та розроблено алгоритм на основі математичних функцій LabVIEW, який дозволяє зі стабільною надійністю обробляти зрізи дифракційних картин різноманітної якості. Блок-схема процесу фільтрації приведена на рисунку 1.6(поз. 2).

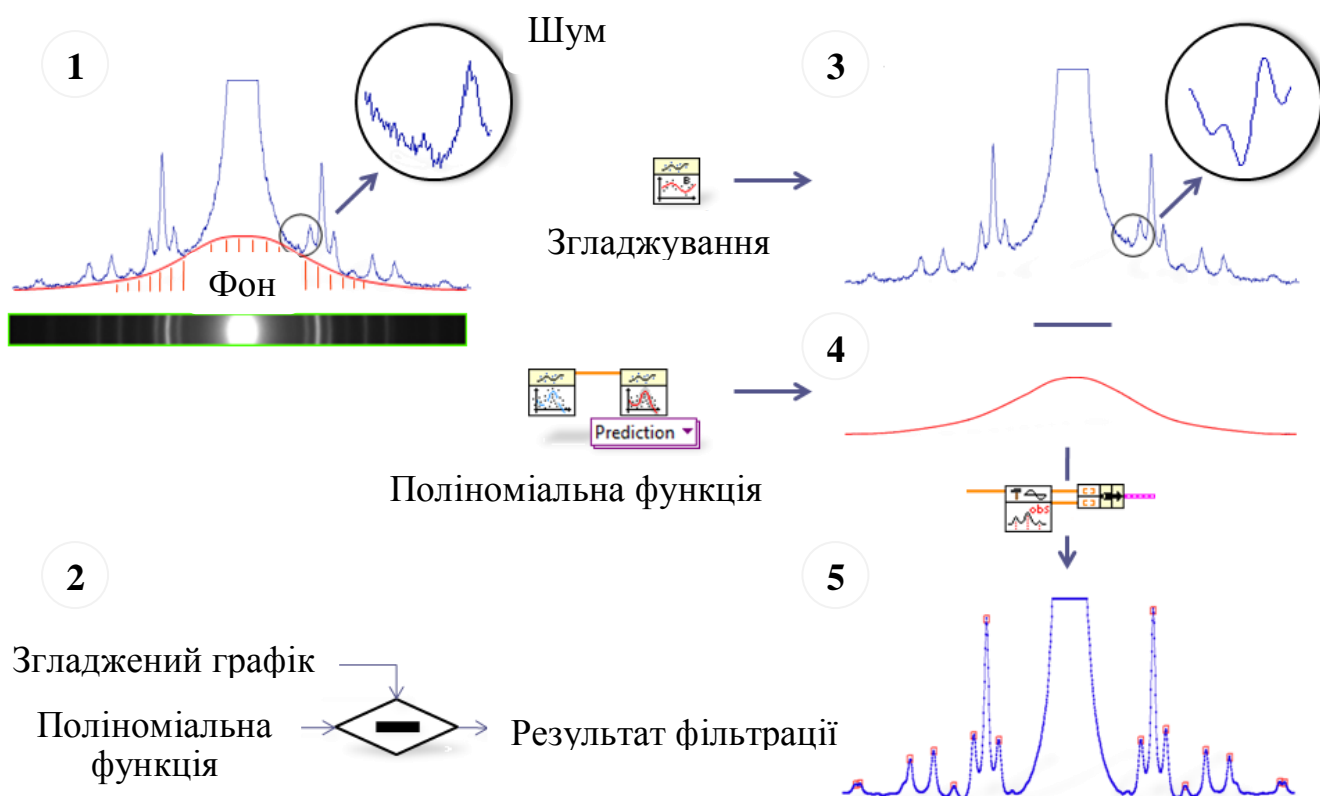


Рис.1.6. – Процес обробки і фільтрації зрізу дифракційної картини

Із блок-схеми слідує, що кінцева оброблена гістограма зрізу є результатом віднімання графіку поліноміальної функції (рис.1.6. поз.4) від первинно згладженого графіка (рис.1.6. поз.3). Поліноміальна функція була вибрана за основу, оскільки її вигляд повторює форму наявного у зрізі фону. Основними функціями описаного фільтру є Polynomial Fit та B-Spline Fit із математичного набору функцій LabVIEW. Параметри фільтрації зрізу дифракційної картини доступні для налагодження в окремому вікні, та дозволяють гнучко підбирати

необхідні значення для отримання бажаного результату. Оброблений графік дозволяє застосувати до себе функцію пошуку піків. За пошук піків відповідає математична функція LabVIEW Peak Detector, яка дозволяє отримати координати розміщення піків та їх інтенсивності. Результат обробки зрізу із знайденими піками, зображено на рисунку 1.6 (поз. 5). Отримані піки та відповідні їм інтенсивності є вихідними даними розробленого фільтру та беруть участь у визначенні вірогідного центру конкретного зрізу.

1.4.3 Визначення центру зрізу

Як вже було зазначено у попередньому параграфі, отримані в результаті обробки піки та відповідні їм інтенсивності є вихідними даними створеного фільтру та беруть участь у визначенні вірогідного центру конкретного зрізу. За визначення центру кожного зрізу дифракційної картини відповідає одна підпрограма, вхідними даними якої є вихідні дані підпрограми фільтрації. А саме інтенсивності піків і їх координати. Виходячи з цих даних, покладений в основу роботи підпрограми алгоритм розраховує найбільш вірогідний центр зрізу дифракційної картини. Для отримання бажаного результату необхідно виконати ряд послідовних операцій алгоритму. По-перше визначаються центри кожного піку з кожним. Іншими словами розраховується середнє арифметичне значення кожного піку з кожним (рис.1.7). Отриманий після проведення даної операції масив, окрім корисної інформації містить також повторювані значення, що утворились в результаті знаходження центрів від одних і тих же піків переставлених місцями. Наприклад із послідовності піків 1, 2 і 3 будуть повторюватись значення центрів для пар піків 1-2 і 2-1, 1-3 і 3-1, 2-3 і 3-2. Тому наступною операцією алгоритму є сортування масиву отриманих значень і видалення повторюваних значень. Слід зазначити, що очевидно значення правдивого вірогідного центру зрізу, повинно знаходитись у вузькому діапазоні середньоарифметичних значень від дзеркальних відносно центру піків. Для знаходження значення правдивого вірогідного центру необхідно в масиві відсортованих значень виділити середньоарифметичні значення дзеркальних

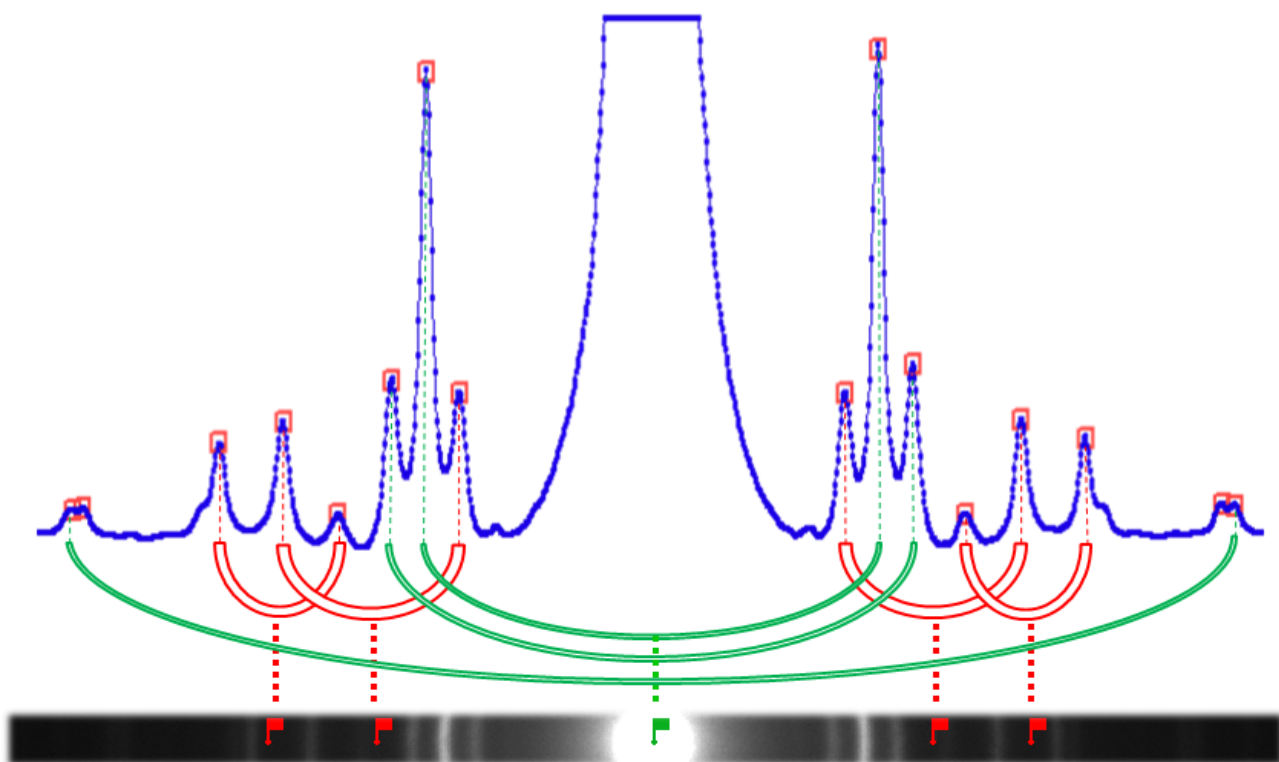


Рис.1.7. – Визначення вірогідного центру зрізу дифракційної картини

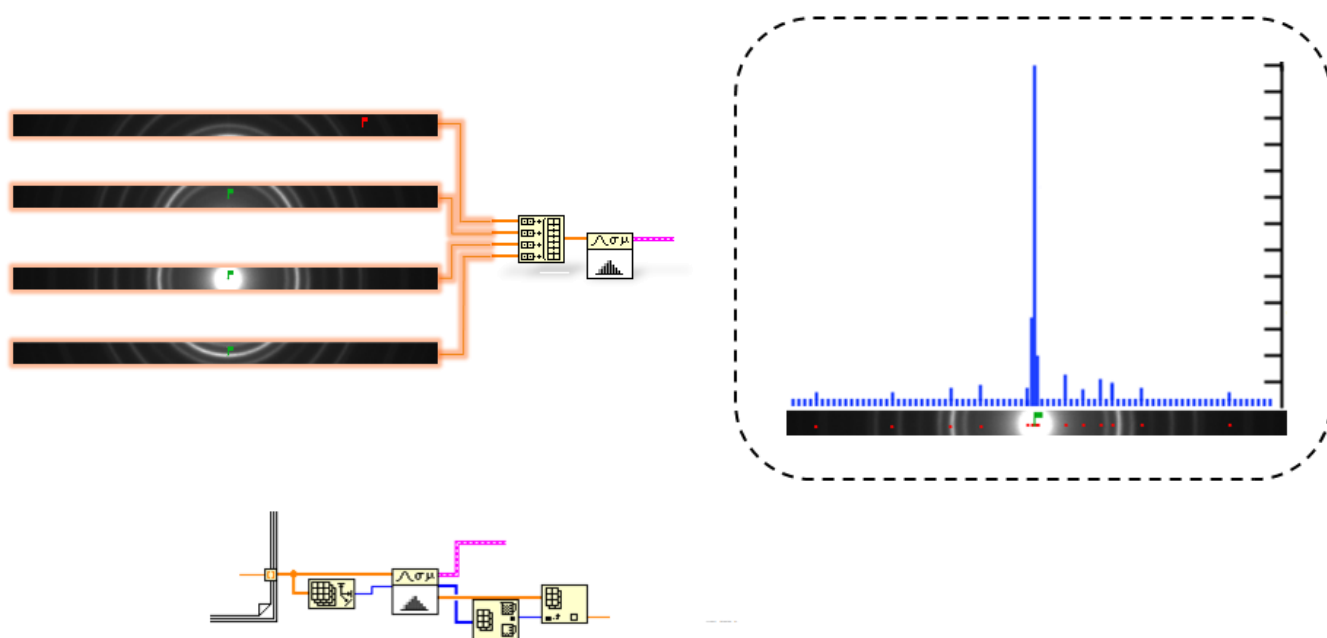


Рис.1.8. – Вибір дійсного центру дифракційної картини

піків. Для цього в алгоритмі використовується операція віднімання кожного елемента масиву від кожного елемента. Тобто для кожного розрахованого середньоарифметичного значення буде побудовано одновимірний масив кожний елемент якого містить результат різниці $(i+1)$ -го та i -го середньоарифметичних значень.

Це дає змогу зробити вибір із отриманого масиву лише тих середньоарифметичних значень, що були утворені існуючими дзеркальними піками. Але як виявилось в ході розробки даного алгоритму, недостатньо лише вибору значень центрів дзеркальних піків. Для створення більш надійного і ефективного алгоритму було створено додаткову операцію пов'язану з аналізом інтенсивностей відповідних дзеркальних піків. Її введення пояснюється тим, що найбільш вірогідні значення центру зрізу отримуються від дзеркальних піків з явно вираженими інтенсивностями. Щоб врахувати дану особливість, кожне значення центру від дзеркальних піків записується до масиву рівно стільки разів, скільки одиниць складає інтенсивність поточного піку, тобто інтенсивність дифракційного кільця. Після завершення виконання алгоритму, із масиву отриманих вірогідних центрів зрізу дифракційної картини вибирається найбільш повторюване значення. Це значення зберігається в головному статистичному масиві вірогідних центрів однієї з двох осей X або Y , як значення вірогідного центру поточного зрізу дифракційної картини.

1.4.4 Визначення дійсного центру дифракційної картини

Попередні етапи алгоритму автоматичного пошуку центру дифракційної картини були спрямовані на отримання статистичного масиву вірогідних значень центрів за двома осями X та Y . Даний етап пов'язаний з визначенням дійсного центру дифракційної картини і передбачає аналіз отриманих статистичних масивів. Нагадаю, що на кожній ітерації сканування до статистичних масивів записуються певні значення вірогідних центрів зрізів дифракційної картини (рис.1.8).

Після завершення сканування всього зображення із двох отриманих масивів вірогідних центрів за осями X та Y , вибирається найбільш повторюване значення. Дане значення зберігається в пам'яті програми і надалі використовується як координати положення дійсного центру. Про кількість повторів кожного з вірогідних центрів можна говорити за побудованою вірогідною гістограмою, яку користувач може переглянути в окремому вікні.

1.5 Пошук дифракційних кілець

Окрім визначення центру дифракційної картини, для здійснення розрахунку параметра кристалічної ґратки необхідно також мати інформацію про місцеположення дифракційних кілець відносно центральної точки. Для точного визначення діаметрів та врахування можливості наявності у дифракційних кілець деякої еліптичності, що може виникати внаслідок апаратного впливу на процес формування дифракційної картини, було розроблено методику аналізу області дифракційної картини за секторами.

1.5.1 Особливості аналізу зображення за секторами

Суть зазначеної методики полягає в аналізі області зображення дифракційної картини у вигляді секторів. Даний аналіз передбачає сканування сектору вибраної області дифракційної картини. Перед початком сканування користувачу надається можливість вибору тієї області, аналіз якої необхідно провести. За допомогою цього методу створюється можливість вибору найбільш придатної для аналізу області зображення дифракційної картини. Вибір сектору для сканування проходить у два етапи і передбачає зазначення початкової і кінцевої меж аналізу. Для вибору меж сканування на панелі управління передбачено два функціональних елемента, вибір одного з яких починає роботу певної потрібної функції. Межі сканування представляють собою лінії, довжина яких динамічно вираховується в залежності від координат положення центру дифракційної картини та поточних координат курсору на області зображення. В програмному забезпеченні реалізовано варіанти задання радіус-сектора та

діаметр-сектора. У другому випадку передбачено динамічне віддзеркалення радіус-сектора відносно центру дифракційної картини, в залежності від положення курсору на зображенні. За допомогою початкової межі користувач має можливість вказати радіус, а використовуючи кінцеву межу сканування з'являється можливість задати кут сектору для аналізу φ . Після задання області сканування програма розпочне роботу алгоритму аналізу за секторами. Особливою ознакою цього алгоритму є те, що при однаковій довжині ліній початкової і кінцевої меж сектору, розмір вирізаних масивів цих ліній суттєво відрізняється (рис.1.9. поз.1). Причина такої відміни, по-перше, полягає у апаратних властивостях екранів ЕОМ, а саме визначається формою пікселя, що представляє собою квадрат. По-друге, лінія яка проведена на області зображення включає лише ті значення пікселів, які повністю їй належать. Для усунення описаного ефекту було використано інтерполяцію отримуваних на кожному кроці даних. Блок-схема сканування області зображення за секторами зображена на рисунку 1.9 (поз. 2). Із схеми бачимо, що сканування відбувається шляхом покрокового руху лінії 2 до лінії 1. Рух лінії представляє собою розрахунок та переміщення координат вершин лінії 2 при зменшенні кута сектору на 1° . В даному алгоритмі реалізовано рух за наступними формулами:

$$X=R\cos\varphi \text{ та } Y=R\sin\varphi,$$

де X та Y – координати наступного положення вершин лінії;

R – радіус сектору.

На кожній ітерації сканування отриманий масив поточної лінії інтерполюється і додається до загального масиву вирізаних ліній. Після завершення сканування області сектору розраховується середньоарифметичне значення загального масиву вирізаних ліній, яке надалі приймає участь в аналізі з метою визначення піків. Аналіз отриманого після сканування графіка відбувається по аналогії з алгоритмом автоматичного пошуку центру.

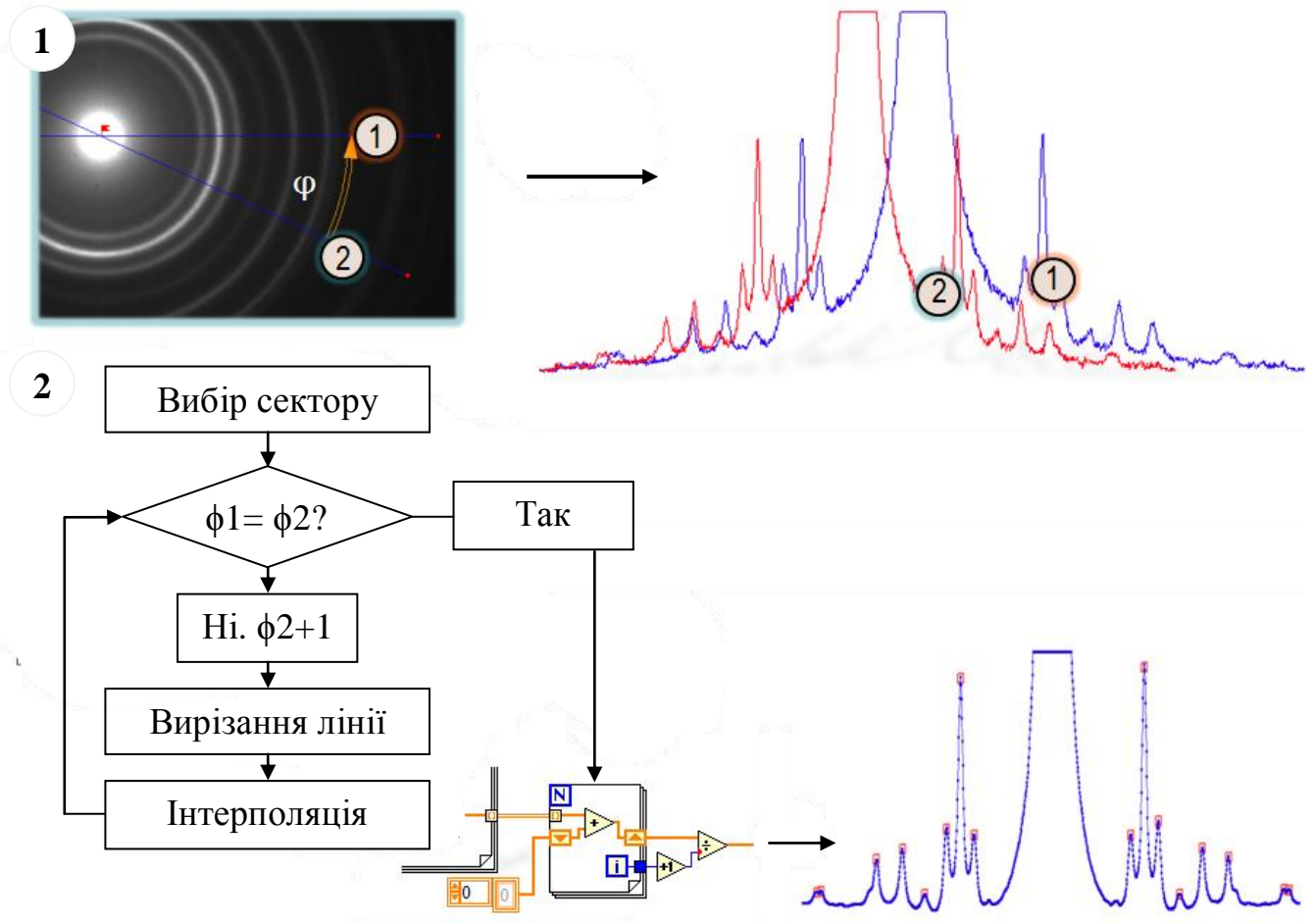


Рис.1. 9. – Особливості аналізу області зображення у вигляді сектору

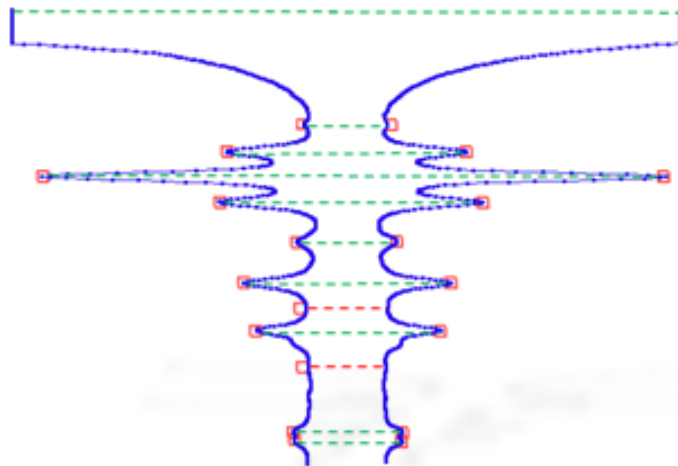


Рис.1.10. – Формування дифракційних кілець

1.5.2 Вибір дифракційних кілець

Після завершення сканування та аналізу середнього значення лінії сектору, необхідно провести перевірку отриманих в результаті аналізу піків, на можливість формування парою піків одного із кілець дифракційної картини.

Як і у випадку з визначенням центру дифракційної картини, розраховується середньоарифметичне значення кожного піку з кожним та визначаються координати розміщення центру зрізу сектора дифракційної картини. Подальший аналіз полягає у порівнянні середньоарифметичних значень кожного піку з кожним із значенням координати центру графіка. Та пара піків, значення центру якої знаходиться у допустимому діапазоні, формує одне із кілець дифракційної картини (рис.1.10). Використовуючи описаний підхід вдалося на цьому етапі реалізувати фільтрацію помилкових піків, що не мають свого дзеркального відображення. Зафіксовані кільця візуалізуються за допомогою функцій IMAQ Vision – Overlay.

Після візуалізації дифракційних кілець користувачу надається можливість вибору тих кілець, які на його погляд повинні брати участь в подальшому розрахунку параметра кристалічної ґратки. Вибір здійснюється за допомогою сектору управління, що представляє собою вільний від контурів кілець сектор з розміщеними всередині квадратами (рис.1.11).

Кожен квадратик відповідає за окреме кільце дифракційної картини і дозволяє відмітити його для участі у розрахунку. Алгоритм вибору кілець полягає в аналізі координат натискання курсору і перевірки потрапляння останніх в площину того чи іншого квадрата. Дані про діаметри, інтенсивності та номери вибраних кілець автоматично відображаються в таблиці розміщеній на лицьовій панелі.

Останнім завершальним етапом роботи програмного забезпечення є саме його ціль, тобто розрахунок параметрів кристалічної ґратки. Для проведення розрахунку необхідно мати інформацію про сталу приладу, діаметри та інтенсивності вибраних кілець і суму квадратів індексів Міллера для відповідних кілець.

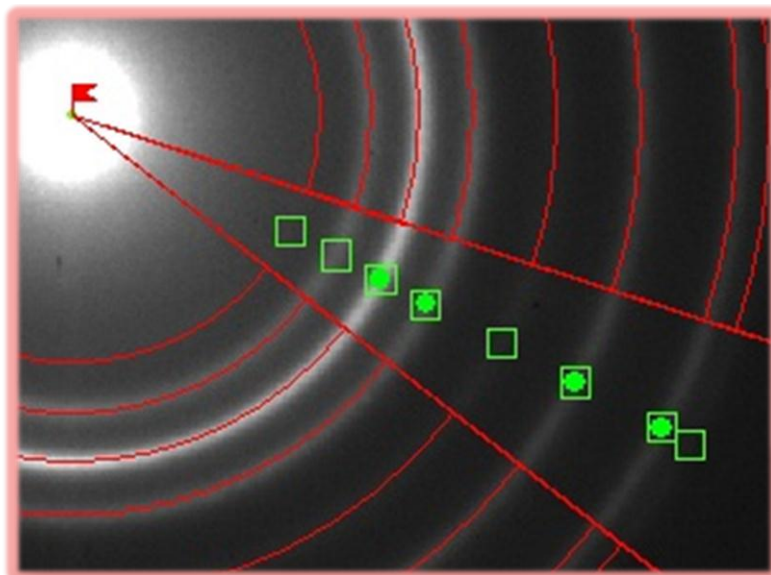


Рис.1.11. – Вибір кільця за допомогою сектору управління

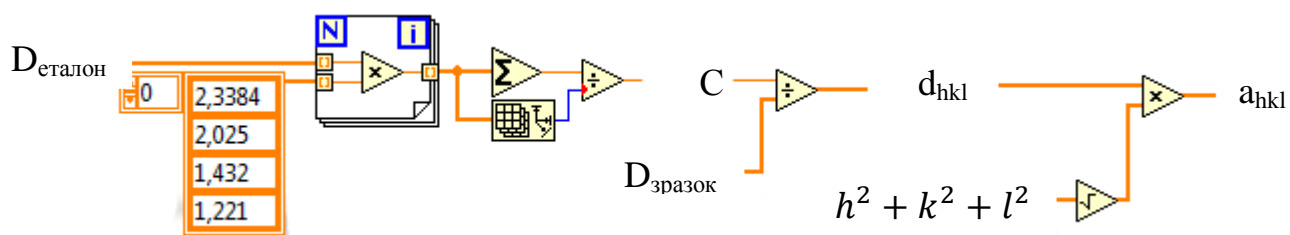


Рис.1.12. – Послідовність розрахунку параметра кристалічної ґратки

Тобто для розрахунку залишається визначити лише тип кристалічної ґратки. В програмному забезпеченні реалізовано визначення типу кристалічної ґратки як в ручному, так і в автоматичному режимах. Методика визначення типу кристалічної ґратки в автоматичному режимі полягає у визначенні типу безпосередньо на етапі вибору дифракційних кільць. Дана методика основана на розрахунках описаних у розділі 1. Слід зазначити, що для коректного визначення типу, необхідно обов'язково вказувати перше кільце, що пов'язано з особливостями методики розрахунку. В програмі реалізована можливість редагування індексів Міллера в ручному режимі. Після задання всіх необхідних параметрів програма виконає розрахунок параметра кристалічної ґратки. Послідовність розрахунку зображена на рисунку 1.12.

2. Обговорення результатів досліджень

Оскільки метою створення програмного забезпечення було проведення розрахунків з нанорозмірною точністю, обов'язково необхідно переконатись у правильності визначення параметрів. Для перевірки достовірності даних отриманих при роботі з програмним забезпеченням було проведено експериментально-порівняльне тестування. Метою тестування був розрахунок різних дифракційних картин в ручну та з використанням розробленої програми і подальше порівняння отриманих результатів з відомими табличними даними.

Результат експериментально-порівняльного тестування приведені у таблиці 2.1.

Табл.2.1. – Результати експериментально-порівняльного тестування

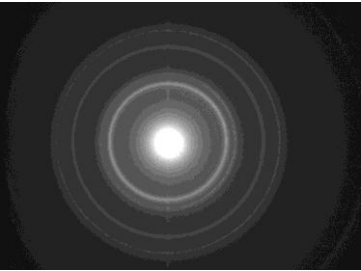
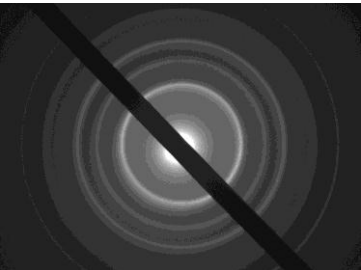
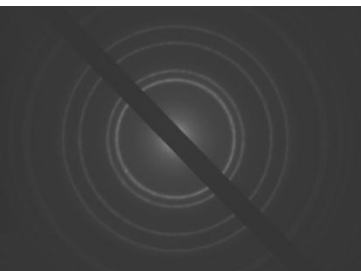
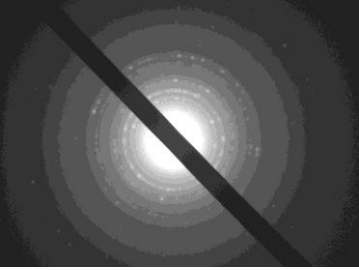
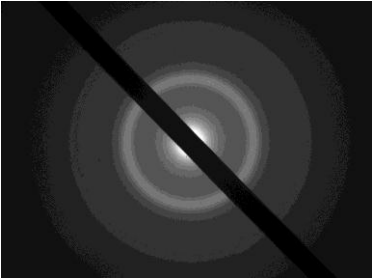
Зображення дифракційної картини	Розрахунок		
	Ручний	Програмний	Табличний
<p style="text-align: center;">Cu</p> 	a=0,3620 нм	a=0,3603 нм	a=0,3608 нм
<p style="text-align: center;">α-Fe</p> 	a=0,2851 нм	a=0,2855 нм	a=0,2861 нм
<p style="text-align: center;">Ni</p> 	a=0,3501 нм	a=0,3520 нм	a=0,3517 нм

Табл.2.1. - Продовження

Зображення дифракційної картини	Розрахунок		
	Ручний	Програмний	Табличний
<p style="text-align: center;">FeO</p> 	a=0,8203 нм	a=0,8222 нм	a=0,8231 нм
<p style="text-align: center;">Co</p> 	a=0,3535 нм	a=0,3547 нм	a=0,3554 нм

Із результатів тестування добре видно, що похибка ручного розрахунку значно більше ніж похибка програмного забезпечення.

Висновки

1. Розроблене сучасне програмне забезпечення для розпізнавання, аналізу та розрахунку дифракційних картин з нанорозмірних плівкових систем отриманих методом просвічуючої електронної мікроскопії. Розроблений власний оригінальний вірогіднісний алгоритм для розпізнавання і аналізу образів, алгоритм математичної обробки дифракційних картин, алгоритми встановлення меж аналізу та відображення дифракційних кілець. Проведено експериментально-порівняльне тестування, що дало змогу переконатись у правильності і високій точності розрахункових результатів;

2. Інтерфейс розробленого програмного забезпечення є інтуїтивно зрозумілим і дозволяє швидко здійснювати необхідні розрахунки дифракційної картини.

3. Архітектурний підхід, що базується на використанні Queue State Machine дозволив створити гнучку структуру програмного коду з широкими можливостями подальшої розробки та виправлення помилок. Використання паралельних потоків дозволяє прискорити швидкість аналізу при використанні багатоядрових процесорів;

4. Використання LabView 2010, NI Vision Development Module дозволяє швидко і ефективно створювати надійні системи для автоматизації сучасних фізичних експериментів, навчання, розпізнавання образів з нанорозмірних систем, систем контролю та керування процесами;

5. Під час тестування програмного забезпечення було розраховано параметри кристалічних ґраток різних матеріалів. Результати тестування показали значно кращу точність у порівнянні із ручним розрахунком.

Список літератури

1. Проценко І. Ю. Методичні вказівки до лабораторних робіт з курсу «Технологія та фізика тонких плівок». – Суми: СумДУ, 1995.- 38с.
2. Дж. Тревис, Дж. Кринг. LabVIEW для всех, третье издание. – М.: ДМК Пресс, 2008.- 880с.
3. Визильтер Ю. В., Желтов С. Ю., Князь В. А., Ходарев А. Н., Моржин А. В. Обработка и анализ цифровых изображений с примерами на LabVIEW и IMAQ Vision. – М.: ДМК Пресс, 2007.- 464с.
4. Евдокимов Ю.К., Линдваль В.Р., Щербаков Г.И. LabVIEW для радиоинженера: от виртуальной модели до реального прибора. Практическое руководство для работы в программной среде LabVIEW. – М.: ДМК Пресс, 2007.- 400с.
5. A. Lukindo – LabVIEW Queued State Machine Architecture,
<http://expressionflaw.com>
6. Питер Блум. LabVIEW: Стиль программирования. – М.: ДМК Пресс, 2009.- 400с.
7. Бутырин П.А. Автоматизация физических исследований и эксперимента: компьютерные измерения и виртуальные приборы на основе LabVIEW 7 – М.: ДМК Пресс, 2005.- 264 с.
8. R. Bitter, T. Mohiuddin. LabVIEW. Advanced programming techniques. - Illinois: CRS Press, 2007. – 497 p.
9. Суранов А.Я. LabVIEW 8.20: Справочник по функциям. – М.: ДМК Пресс, 2007.- 536с.
10. Davies E. R. Machine Vision: Theory, Algorithms, Practicalities. //Academic Press. 2-nd Edition, San Diego.- 1997.- P.750.