

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Навчально-науковий  
інститут електроенергетики  
(інститут)

Факультет інформаційних технологій  
(факультет)  
Кафедра інформаційних технологій та комп'ютерної інженерії  
(повна назва)

**ПОЯСНЮВАЛЬНА ЗАПИСКА**  
**кваліфікаційної роботи ступеня бакалавра**

здобувача \_\_\_\_\_ Гатило Ірини Сергіївни \_\_\_\_\_  
(ПІБ)

академічної групи \_\_\_\_\_ 126-213-1 \_\_\_\_\_  
(шифр)

спеціальності \_\_\_\_\_ 126 Інформаційні системи та технології \_\_\_\_\_  
(код і назва спеціальності)

за освітньо-професійною програмою Інформаційні системи та технології \_\_\_\_\_  
(офіційна назва)

на тему “Розробка графічного редактору для зменшення кольорової палітри  
та контролю лініатури”  
(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц. Каштан В.Ю.			
Рецензент				
Нормоконтролер	проф. Коротенко Г.М.			

Дніпро  
2025

**ЗАТВЕРДЖЕНО:**  
завідувач кафедри  
інформаційних технологій  
та комп'ютерної інженерії  
(повна назва)  
Гнатусенко В.В.  
(підпис) (прізвище, ініціали)

" " \_\_\_\_\_ 2025 року

**ЗАВДАННЯ**  
**на кваліфікаційну роботу**  
**ступеня бакалавр**

здобувача Гатило І.С. академічної групи 126-21з-1  
(прізвище та ініціали) (шифр)

спеціальності 126 Інформаційні системи та технології

за освітньо-професійною програмою Інформаційні системи та технології  
(офіційна назва)

на тему " Розробка графічного редактору для зменшення кольорової палітри та контролю лініатури "

затверджену наказом ректора НТУ «Дніпровська політехніка» від 05.05.2025 № 337-с

Розділ	Зміст	Термін виконання
Розділ 1	Аналіз стану області рішення задач	07.04.2025 – 30.04.2025
Розділ 2	Теоретична частина	01.05.2025 – 25.05.2025
Розділ 3	Практична частина	26.05.2025 – 10.06.2025

Завдання видано \_\_\_\_\_  
(підпис керівника)

доц. Каштан В.Ю.  
(прізвище, ініціали)

Дата видачі 25.02.2025

Дата подання до екзаменаційної комісії \_\_\_\_\_

Прийнято до виконання \_\_\_\_\_

Гатило І.С.

## РЕФЕРАТ

Пояснювальна записка: 87 с., 48 рис., 3 табл., 1 додаток, 16 джерел.

ЛІНІАТУРА, ГЛИБИНА КОЛЬОРУ, РАСТРОВЕ ЗОБРАЖЕННЯ,  
ГІСТОГРАМА, ОБРОБКА ЗОБРАЖЕНЬ, RGB

*Об'єктом роботи* – є графічні растрові зображення мікроструктури металевих сплавів, зокрема їхні колірні характеристики та лініатура..

*Предметом роботи* – методи та алгоритми обробки графічних растрових зображень для зменшення колірної палітри, контролю лініатури та підвищення якості друку.

*Метою кваліфікаційної роботи* є розробка графічного редактора для зменшення колірної палітри та контролю лініатури растрових зображень з метою підвищення якості їхнього друку.

У рамках роботи було проведено аналітичну складову, що включає дослідження впливу глибини кольору на лініатуру та розробку методу підвищення якості друку з урахуванням властивостей металографічних растрових зображень. Також було розроблено структуру програмного додатку, визначено алгоритм роботи користувача з ним, сформовано алгоритми роботи окремих модулів програми та створено метод видалення кольорів із зображення для підвищення лініатури. В рамках реалізації програмного забезпечення була створена програма, яка дозволяє керувати лініатурою друку зображення шляхом зміни глибини кольору растрових зображень. Ця програма забезпечує можливість збільшення лініатури та якості друку за рахунок зменшення кількості кольорів у зображенні мікроструктури металевих сплавів.

## ABSTRACT

Explanatory Note: 87 pages, 25 figures, 3 tables, 1 appendix, 16 sources.

SCREEN RULING, COLOR DEPTH, RASTER IMAGE, HISTOGRAM, IMAGE PROCESSING, RGB

*The object* of the work is graphical raster images of the microstructure of metal alloys, specifically their color characteristics and screen ruling.

*The subject* of the work is methods and algorithms for processing graphical raster images to reduce the color palette, control screen ruling, and improve print quality.

*The aim* of the qualification work is the development of a graphic editor for reducing the color palette and controlling the screen ruling of raster images with the goal of improving their print quality.

Within the scope of the work, an analytical component was carried out, which includes research into the influence of color depth on screen ruling and the development of a method to improve print quality, taking into account the properties of metallographic raster images. The software application structure was also developed, the user interaction algorithm was defined, algorithms for the operation of individual program modules were formulated, and a method for removing colors from an image to increase screen ruling was created. As part of the software implementation, a program was developed that allows for the management of image print screen ruling by changing the color depth of raster images. This program provides the ability to increase screen ruling and print quality by reducing the number of colors in the image of metal alloy microstructure.

## ЗМІСТ

ВСТУП.....	7
1. АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ.....	8
1.1. Вступ.....	8
1.2. Глибина кольору та колірні моделі .....	9
1.2.1. Визначення кольору .....	9
1.2.2. Розрядність.....	10
1.2.3. Визначення глибини кольору.....	12
1.2.4. Глибина кольору на прикладі зображення .....	13
1.2.5. Вибір розрядності растрового зображення.....	18
1.2.6. Колірні моделі .....	19
1.2.7 Колірна палітра CIE Lab .....	23
1.3. Растр, растрування та лініатура.....	25
1.4. Особливості друку і роздільна здатність струйних принтерів .....	27
1.5. Узгодження пристроїв, CMS.....	28
1.6. Вибір середовища розробки .....	32
1.7 Постановка задачі роботи.....	33
2. ТЕОРЕТИЧНА ЧАСТИНА .....	35
2.1. Конкретизація задачі роботи.....	35
2.2 Лініатура та масштаб.....	37
2.3 Видалення кольорів.....	41
2.4 Додаткові методи підвищення якості друку.....	44
2.5 Розробка алгоритму роботи користувача.....	46
2.6 Розробка структури програмного додатку.....	47
2.7 Алгоритм роботи програми .....	50
2.7.1 Алгоритм роботи процедури FormCreate .....	50
2.7.2 Алгоритм роботи процедури ButtonOpenClick .....	51
2.7.3 Алгоритми роботи процедур Gistogram та Gistogram1 .....	53
2.7.4 Алгоритм роботи процедури LeftRight.....	54
2.7.5 Алгоритми роботи процедур CounSum та CounNull .....	55

	6
2.7.6 Алгоритми роботи процедур DiagramRG та DiagramNewRG .....	56
2.7.7 Алгоритм роботи процедури BitBtnRebuildClick .....	57
2.7.8 Алгоритм роботи процедури RebuildOriginRGB .....	58
2.7.9 Алгоритми роботи процедур ImageNewRGBMouseDown та Button1Click .....	61
3. ПРАКТИЧНА ЧАСТИНА .....	62
3.1 Опис структури програмного додатку .....	62
3.2 Опис інтерфейсу та функціоналу програмного додатку .....	63
3.2.1 Бічна панель .....	63
3.2.2 Вкладка «Оригінальне зображення» .....	67
3.2.3 Вкладка «Оброблене зображення» .....	68
3.2.4 Вкладка «Обробка RGB» .....	70
3.2.5 Вкладка «RG Діаграми» .....	75
3.3 Тестування програмного додатку .....	77
ВИСНОВКИ .....	80
СПИСОК ПОСИЛАНЬ .....	81
ДОДАТОК А. Фрагмент програмної реалізації .....	83

## ВСТУП

Друк зображення в такому ж вигляді, як бачить його людина або в високій якості, щоб можна було досліджувати структуру металевих сплавів по ньому є актуальним і на сьогодні. Дослідники для точного відображення фото під час друку застосовують різні інструменти, які по-своєму покращують фото.

Існують різні методи для покращення фото: обробка зображення (збільшення його розподільної здатності, змінення геометрії растру, поєднання різних методів растрування, корекція кольорів, конвертація в колірну модель принтера та інше), налаштування принтера, підбір правильного паперу та фарб для друку тощо. Однак всі ці методи мають як свої переваги, так і недоліки, отже питання якісного друку залишається завжди відкритим.

Дослідники мікроструктури металевих сплавів під друку отримують зображення з поганою якістю. Навіть якщо роздільна здатність оптичного мікроскопу була великою, під час друку мале значення лініатури може погіршити якість фото. Тому була поставлена задача розробити такий інструмент корекції кольору растрових зображень, який підвищить лініатуру, зменшуючи при цьому глибину кольору. В роботі також розроблений метод видалення кольорів зображення (зменшення глибини кольору), що збільшить лініатуру зображення для якісного відображення друкованого фото мікроструктури металевих сплавів.

# 1. АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ

## 1.1. Вступ

Проблема відображення на екрані та принтері кольорів растрових зображень пов'язана, по-перше, з взаємним перетворенням зображення колірних моделей RGB та CMYK, що приводить до втрати деяких кольорів; по-друге, з залежністю геометричної розподільної здатності та глибини кольору для друку.

Для коректного друку фото металографічних сплавів важлива лініатура, а точніше важливо її збільшити до роздільної здатності монітора або роздільної здатності оптичного мікроскопа, адже стандартна лініатура в струйних принтерах інколи буває маленькою, від чого помітна відстань між точками та зображення друкується з меншою точністю, ніж його можна побачити на екрані монітора чи оптичного мікроскопу.

Відношення роздільної здатності принтера до лініатури растра дає розмір сторони комірки растру, що вимірюється в точках принтера. Максимальна кількість точок принтера, що складають растрову точку рівна квадрату сторони комірки. Так, наприклад, якщо лініатура растру встановлена рівною 100 ліній/дюйм, а роздільна здатність принтера – 600 точок на дюйм, то сторона комірки растру буде рівна  $600/100=6$  точкам. При заданих умовах растрова точка формується з  $6 \times 6=36$  точок принтера. Ця величина має велике значення для адекватної передачі фото мікроструктури металевих сплавів, оскільки саме вона визначає кількість відтінків, що може передати растр з заданими параметрами. Чим вища лініатура – тим менш помітними будуть точки, що складають растр і тим ближче відбиток буде до фотографічного оригіналу. [1]

Лініатуру, що буде рівна роздільній здатності принтера можна здобути тільки в тому випадку, коли зображення монохромне, а точніше – чорно-біле (1 bit). Але в даному випадку зображення мікроструктури металевих сплавів мають близько 15 тисяч кольорів, а не 1.

Глибина кольору в зображенні являє собою кількість бітів, які використовуються для відображення кожного кольорового пікселя. Таким чином, чим вище бітова глибина, тим краще дозвіл і якість зображення. Глибина кольору - сумарна кількість градацій інтенсивності, що характеризує максимальну кількість кольорів, яку можна теоретично відтворити на екрані.

Управління глибиною кольору дозволяє побачити різницю, між зернистим і більш чітким зображенням. Саме з допомогою налаштування глибини кольору стає можливим змінити лініатуру друку зображення. Функція налаштування глибини кольору - це налаштування кількості колірних відтінків і тонів, які містить зображення. Якраз таку функцію і було реалізовано в дипломній роботі. Однак розглянемо, що саме є колір, глибина кольору та інші поняття, що стосуються даної теми.

## **1.2. Глибина кольору та колірні моделі**

### **1.2.1. Визначення кольору**

Колір - одна з властивостей об'єктів матеріального світу, сприймається як усвідомлене зорове відчуття. Той чи інший колір "присвоюється" людиною об'єктів в процесі їх зорового сприйняття.

Колірне відчуття виникає в результаті впливу на око потоків електромагнітного випромінювання з діапазону довжин хвиль, в якому це випромінювання сприймається оком (видимий діапазон довжини хвиль від 380 до 760 нм).

При описі кольору використовують три атрибути:

- 1) колірний тон;
- 2) насиченість;
- 3) світлоту.

Найбільш важливий атрибут кольору - колірний тон, який дозволяє розрізнити їх як червоний, жовтий, зелений, синій або як проміжний між двома сусідніми парами цих кольорів. Наприклад, зелений тон привласнюють

предметів з забарвленням, близькою до фарбування природної зелені. Насиченість характеризує рівень тону кольору. Цей атрибут в людській свідомості пов'язаний з кількістю фарби, барвника.

Безбарвними(ахроматичними) кольорами є чорний, білий і всі відтінки сірого, і вважають, що вони не мають насиченості і розрізняються лише по світлоті. Колір ахроматичності поверхні, яка відображає максимум світла, називається "білим".

Поверхні, пофарбовані в хроматичні кольору (всі інші кольори), по-різному відбивають світло з різною довжиною хвилі. [2]

### **1.2.2. Розрядність**

Яким би шляхом не було отримано комп'ютерне зображення, воно має такі основні параметри: дозвіл, формат, тип колірної моделі і палітру (колірна роздільність).

Перераховані фактори визначають розмір файлу растрового зображення (число бітів пам'яті, необхідне для зберігання файлу зображення) і якість зображення. Ці фактори тісно взаємопов'язані. Часто при досягненні високої якості зображення саме розмір його файлу виявляється головним фактором, що обмежує застосування зображення в створюваному документі.

Розмір і якість комп'ютерного зображення в основному визначаються таким поняттям, як роздільність. Зображення формується з елементів, званих пікселями (рис. 1.1). Кожен піксель растрового зображення має чотири характеристики - розмір, тоновий значення, глибину кольору та позицію. Ці чотири атрибута визначають роздільність, причому кожен це робить по-своєму.

Всі пікселі одного цифрового зображення мають однаковий розмір (рис. 1.1). Спочатку розмір пікселя визначено роздільністю, з якою було скановане або оцифроване зображення. Так, дозвіл в 1000 пікселів на дюйм вказує, що розмір кожного пікселя дорівнює приблизно 1/1000 дюйма.

Чим вище дозвіл, тим менше розмір пікселя і тим більше кількість інформації і деталей на одиницю виміру.

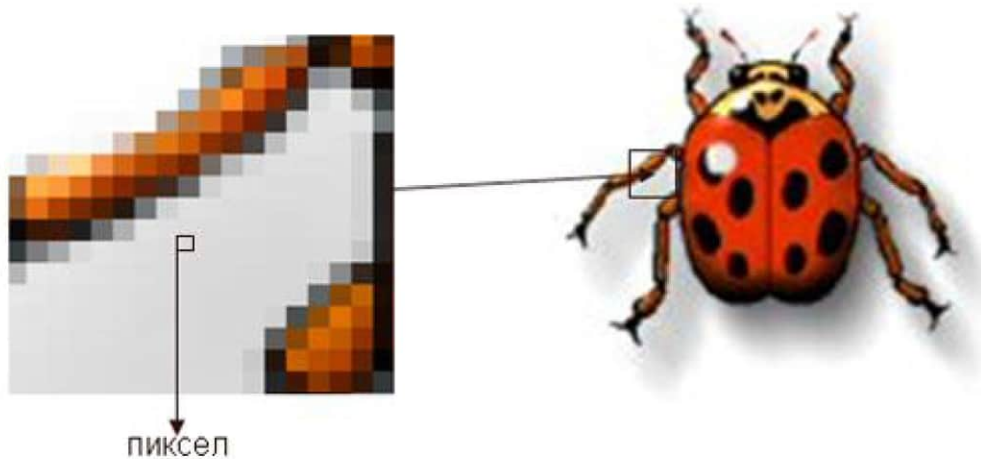


Рисунок 1.1 – Збільшений фрагмент (ліворуч) картинки (праворуч)

При низькій роздільній здатності розмір пікселя(точки) картинки збільшується, що призводить до мозаїчної структури зображення (рис. 1.2).



Рисунок 1.2 – Зліва цифрове зображення має більш високу роздільну здатність, ніж зображення праворуч

При оцифруванні зображення кожному пікселю присвоюється певне значення кольору. Ефект безперервності тонів виникає через те, що пікселі малі і сусідні пікселі тільки трохи відрізняються один від одного за кольором або тоном. Пристрої (сканери, цифрові фотоапарати) з високим відношенням сигнал / шум

і широким динамічним діапазоном найкращим чином передають безперервність тону.

Глибина кольору характеризує кількість рівнів яскравості, які може приймати окремий піксель. Збільшення кількості можливих кольорів або тонів у пікселя призводить до зростання розміру графічних файлів, але при цьому збільшується гладкість переходів між суміжними квітами і тонами.

У чорно-білих зображеннях рівні яскравості представляються у вигляді відтінків сірого. У кольорових зображеннях рівні проявляються у вигляді специфічних колірних тонів. Як правило, для чорно-білих зображень глибина кольору - 8 біт, що відповідає 256 градаціям яскравості; для кольорових - 24 біт (або 16,7 мільйонів різних кольорів).[3]

### **1.2.3. Визначення глибини кольору**

Для призначення кольору і яскравості точкам, що формують зображення на моніторі, потрібно задати значення інтенсивності для кожного каналу кольору. Сумарна кількість градацій інтенсивності визначає глибина кольору, що характеризує максимальну кількість кольорів, яку можна теоретично відтворити на екрані. Також глибина кольору характеризує число градацій яскравості, що відтворює піксел в чорно-білих зображеннях і кількість кольорів в кольоровому зображенні.

Розрядність цифрового зображення - це число розрядів (бітів), що використовуються для кодування кольору одного пікселя.

Слід розрізняти терміни біт на канал (bpc - bits per channel) і біт на піксель (bpp - bits per pixel). Розрядність по кожному з індивідуальних колірних каналів вимірюється в бітах на канал, сума ж розрядів всіх каналів виражається в бітах на піксель. Наприклад, зображення в палітрі Truecolor має розрядність 8 біт на канал, що еквівалентно 24 бітам на піксель, тому що колір кожного пікселя описується трьома колірними каналами: червоним, зеленим і синім (модель RGB).

Для зображення, закодованого в RAW-файлі, число біт на канал збігається з числом біт на піксель, оскільки до інтерполяції кожен піксель, отриманий за допомогою матриці з масивом кольорових фільтрів Байєра, містить інформацію лише про один із трьох первинних кольорів.

У цифровій фотографії описують розрядність переважно за допомогою біту на канал, і тому, говорячи про розрядності, мається на увазі виключно біти на канал, якщо прямо не зазначено інше. Розрядність визначає кількість відтінків, які можуть бути присутніми в колірній палітрі даного зображення. Наприклад, 8-бітове чорно-біле зображення може містити до  $2^8 = 256$  градацій сірого кольору. Кольорове 8-бітове зображення може містити по 256 градацій для кожного з трьох каналів (RGB), тобто всього  $2^8 \times 3 = 16777216$  унікальних комбінацій або колірних відтінків. [4]

Висока розрядність особливо важлива для коректного відображення плавних тональних або колірних переходів. Будь-який градієнт в цифровому зображенні не є безперервною зміною тону, а представляє собою ступінчасту послідовність дискретних значень кольору. Велика кількість градацій створює ілюзію плавного переходу. Якщо ж півтонів занадто мало, ступінчастість видно неозброєним оком і зображення втрачає реалістичність.

Ефект виникнення візуально помітних стрибків кольору в областях зображення, початково містять плавні градієнти, називається постеризацією (від англ. Poster - плакат), оскільки фотографія, в якій бракує напівтонів, стає схожою на плакат, надрукований з використанням обмеженого числа фарб.

#### **1.2.4. Глибина кольору на прикладі зображення**

Для наочності можна проілюструвати викладений вище матеріал, застосовуючи різну розрядність до одного і того ж зображення, де збільшення розрядності на 1 біт (рис.1.3) означає подвоєння кількості відтінків в палітрі зображення. Впродовж ілюстрування буде помітне відношення кількості бітів на канал зображення до його якості.



Рисунок 1.3 – 1 біт – 2 відтінка

1 біт дозволяє закодувати всього два кольори. В даному випадку це чорний та білий (рис.1.4).



Рисунок 1.4 – 2 біта – 4 відтінка

З появою напівтонів зображення перестає бути просто набором силуетів, але всерівно досі виглядає абстрактно (рис.1.5).



Рисунок 1.5 – 3 біта – 8 відтінків

Вже помітні деталі переднього плану. Смугосте небо - хороший приклад постеризації.



Рисунок 1.6 – 4 біта – 16 відтінків

Починають проявлятися деталі на схилах гір (рис.1.6). На передньому плані постеризація вже майже непомітна, але небо залишається смугастим.



Рисунок 1.7 – 5 біт – 32 відтінка

Очевидно, що області з низьким контрастом, відображення яких вимагає великої кількості близьких півтонів, найбільше страждають від постеризації (рис.1.7).



Рисунок 1.8 – 6 біт – 64 відтінка

Гори вже майже в ідеальній якості, але небо досі виглядає з ефектом сходинок, особливо ближче до кутів кадру (рис.1.8, рис.1.9).



Рисунок 1.9 – 7 біт – 128 відтінків

Немає до чого причіпитися, але зображення все ще не 8-бітне.



Рисунок 1.10 – 8 біт – 256 відтінків

Вихідна 8-бітна фотографія (рис.1.10). 8 біт цілком достатньо для реалістичної передачі будь-яких тональних переходів. На більшості моніторів не помітна різниця між 7 і 8 бітами, тому навіть 8 біт можуть здатися зайвими.

Але все ж стандартом для високоякісних цифрових зображень є саме 8 біт на канал, щоб з гарантованим запасом переkritи здатність людського ока розрізняти градації кольору. [5]

### **1.2.5. Вибір розрядності растрового зображення**

Якщо переглянути можливості стандартного графічного редактора MS Paint, то можна помітити, що там на виході пропонується зберегти зображення в форматі BMP у монохромному вигляді, 16 та 256-кольорному вигляді (4 та 8 розрядів) та у 24-розрядному. Але розрядність JPEG, GIF, PNG та TIFF обрати не можна.

Переглянемо розрядність та переваги вищеназваних форматів:

- GIF – підтримує прозорість, анімація, розрядність 8;
- JPEG – підтримує різні моделі кольорів, метадані, розрядність до 32;
- PNG – підтримує прозорість, метадані, розрядність до 48;
- TIFF - підтримує різні моделі кольорів, маски, розрядність 8 або 16;
- BMP – підтримує колірну модель RGB, розрядність до 32.

Для зберігання вже оброблених фотографій краще використовувати TIFF. Цей формат графіки підтримує зберігання растрової графічної інформації у всіляких колірних моделях (RGB, CMYK, Grayscale, LAB, HSB, Bitmap), зберігає інформацію про роздільну здатність і габарити зображення, може містити альфа канали та відсічні контури, а також копію зображення в зменшеному форматі (preview). Крім того, дуже важливою особливістю цього формату є можливість впровадження в нього профілів системи, в якій дане зображення створювалося. Всі ці можливості роблять використання цього формату широким і повсюдним. Зазвичай сканери, фотокамери та інші пристрої оцифровки зображень видають відеоінформацію саме в цьому форматі.

Враховуючи всі переваги формату TIFF, варто зауважити, що зображення мікроструктури металевих сплавів, отримані в форматі BMP. Формат BMP,

іноді званий точковим малюнком або DIB форматом (апаратно-незалежний растр). Цей формат використовується для зберігання растрових цифрових зображень в операційній системі Windows. Через відсутність стиснення BMP файли мають великий розмір. У форматі BMP підтримується друк в монохромному, 256 кольорів, відтінків сірого і кольоровому (24-бітному) варіанті. На відміну від формату JPEG, BMP не компресує відтінки зображення, що дає змогу керувати колірними даними зображення, не погіршуючи якість зображення. Тому саме він і буде використовуватись для обробки фото.

### 1.2.6. Колірні моделі

Файл цифрового зображення містить числові значення кольору для кожного пікселя. А колірна модель визначає колірну систему, яка використовується для опису значень кольору окремих пікселів і їх організації в зображенні. У напівтоновій моделі зображення можуть бути тільки чорно-білими, тоді як для кольорових зазвичай використовуються колірні моделі RGB (використовує монітор), CMYK (використовує принтер) або Lab. У цифровому фотоапараті застосовується тільки RGB, хоча він може бути змінений під час редагування та друку зображення.

Напівтонова модель (в градаціях сірого) призначена для некольорових зображень. Так, 8-розр. півтонування може містити 256 можливих градацій сірого, а 16-розрядне - теоретично 65536 градацій сірого. Чорно-біла фотографія на перший погляд містить тільки чорний і білий колір, але при більш ретельному вивченні виявляються ледь помітні тони, які посилюють забарвлення в області тіней і напівтонів. Якщо в цифровому фотоапараті доступна напівтонова модель, нею не варто користуватися, оскільки рішення про виключення конкретних кольорів і колірних тонів приймається не під час зйомки, а в процесі редагування зображення. На цьому етапі є також відмінна можливість отримати чорно-білі відбитки з файлу RGB, які не поступаються за якістю справжнім фотовідбиткам.

Колірна модель RGB (Red - червоний, Green - зелений, Blue - синій) є найпоширенішою для фотографічних зображень. У цій моделі інформація зберігається в трьох основних адитивних кольорах: червоному, зеленому і синьому. Це основні кольори не тільки для випромінюваного світла, але і для світла, що фіксується світлочутливими датчиками цифрової фотокамери. Крім того, модель RGB застосовується в моніторах комп'ютерів.

Основні кольори цієї моделі називаються адитивними, тому що довжини хвиль світла, що мають відповідне забарвлення, складаються, утворюючи остаточний колір (Рис. 1.11).

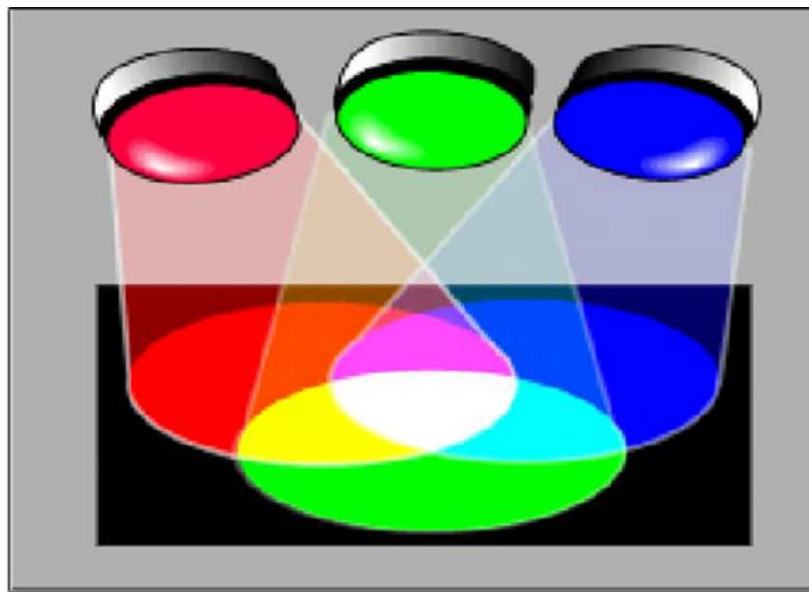


Рисунок 1.11 – Складання кольорів адитивною колірною моделлю RGB

Зображення, яке зберігається в файлі RGB, фактично містить три окремі канали, що описують значення червоного, зеленого і синього кольору кожного пікселя. Ці значення можуть змінюватися в межах від 0 (чорний колір) до 255 (білий колір) для 8-розрядних зображень. Так, якщо у всіх трьох каналах кольору встановлені значення 0 для окремого пікселя, він буде чорного кольору. А якщо ці значення рівні 255, то утворюється білий колір. Якщо ж у всіх трьох каналах кольору встановлене однакове проміжне значення, даний піксель буде пофарбований відтінком сірого.

В результаті змішування різних значень в кожному каналі кольору для забарвлення окремого пікселя в 8-розрядному зображенні може бути отримано понад 16,7 млн. кольорів. Як правило, всі кольорові зображення зберігаються в колірній моделі RGB, починаючи з фіксації цифрою фотокамерою і закінчуючи редагуванням і друкуванням.

В колірній моделі RGB є пару недоліків, один з яких – обмеженість колірного охоплення, яка пояснюється тим, що з допомогою адитивного синтезу неможливо отримати всі кольори видимого спектру, що доведено теоретично. В особливості такі кольори як чистий блакитний чи чистий жовтий не можуть бути точно відтворені на екрані. Інший недолік – апаратна залежність, адже монітори є різних виробників та різного пройденого часу експлуатації (з часом старіють люмінофори монітора та змінюються емісійні характеристики електронних прожекторів).

Колірна модель CMYK (Cyan-блакитний, Magenta - пурпурний, Yellow - жовтий, black - чорний) заснована на субтрактивних базових кольорах: блакитному, пурпуровому і жовтому. Ці кольори дають колір в результаті відображення світла, наприклад, від фарби на папері. А четвертий, чорний колір додається до трьох інших кольорів для отримання на папері справді чорного відтінку.

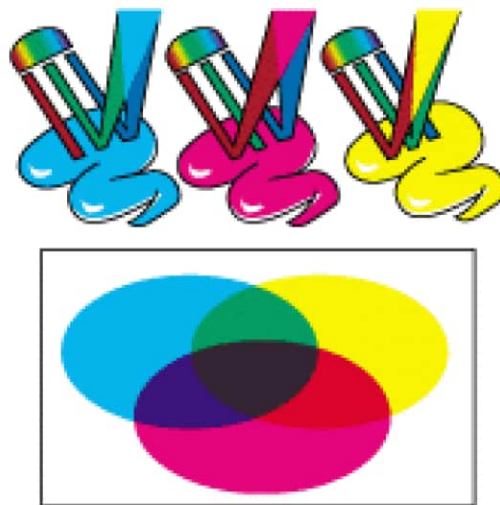


Рисунок 1.12 – Складання кольорів субтрактивною колірною моделлю CMYK

Колір, що виводиться під час друку на папір за допомогою моделі СМУК, в значній мірі визначається якістю самого паперу. Крейдований папір забезпечує відтворення більш широкого спектру кольорів в порівнянні з некрейдованим. На звичайному папері кольори виходять темнішими і приглушеними. Це обумовлено тим, що він має більш шорстку поверхню, що призводить до додаткового розсіювання світла.

Модель СМУК ще називають «виключаючою» через те, що при додаванні на поверхню паперу барвників інтенсивність світла зменшується, оскільки світло поглинається тим більше, чим більше барвників нанесено на поверхню. До речі, на відміну від RGB-пікселів, на роздрукованому зображенні можна помітити маленькі крапки жовтого, блакитного, пурпурного та чорного кольорів різного розміру. Для отримання світлих та темних тонів субтрактивних кольорів використовують відповідно точки маленьких та великих розмірів (Рис. 1.13). З чого можна зробити висновок, що в моделі СМУК пов'язані глибина кольору та геометричне розширення, що являє собою відстань між точками.

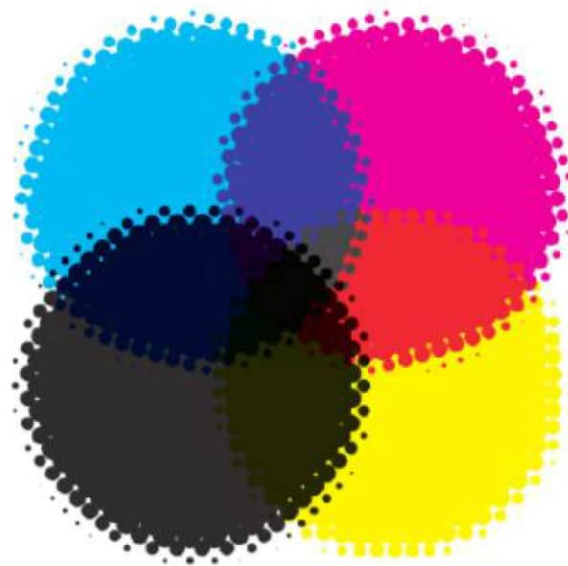


Рисунок 1.13 – Друк зображення з точками різної величини

Недоліки СМУК-моделі також в апаратній залежності та обмеженому колірному діапазоні. Але у випадку СМУК простежується більша апаратна

залежність, ніж в RGB-моделі, та модель CMYK має більш вузький колірний діапазон в порівнянні з RGB. В особливості, вона не може відтворити яскраві насичені кольори, а також ряд специфічних кольорів, як наприклад металевий. З приводу чого і виникла потреба вирішення цієї проблеми в даній роботі та розробка алгоритму корекції глибини кольору растрових зображень відповідно до вимог задач обробки металографічних зображень структури металевих сплавів, адже при потребі роздрукувати на папері зображення із золотим кольором, виглядати цей колір буде специфічно, не так, як на екрані монітора.[6]

### **1.2.7 Колірна палітра CIE Lab**

В якості відправної точки створення наочної системи специфікації кольору послужила теорія оппонентних процесів (Opponency) сприйняття кольору Е. Герінга. На відміну від трикомпонентної теорії, вона пояснює феномени зору, спираючись на уявлення про існування трьох оппонентних процесів і, відповідно, шести основних психологічних квітів: червоний-зелений, жовтий-синій, чорний-білий. Природною геометричною моделлю колірного простору за цією теорією служить система координат, осі якої проходять через перераховані «основні психологічні кольору». У цій геометричній моделі координатні осі, відповідні краснозеленому і синьо-жовтому оппонентном процесам, утворюють площину кольоровості. Ось чорно-білого оппонентного процесу утворює перпендикулярну цій площині вісь світлинності. Для встановлення зв'язку алгебраїчного простору колірних стимулів CIE RGB з геометричним простором кольорів об'єктів, побудованим на опонентних вимірах, використовувалася так звана процедура шкалювання, що представляє собою одну з технологій психофізики. Суть цієї процедури полягає у встановленні зв'язку між фізично вимірюваними колориметричною величинами (наприклад, координати CIE XYZ) з візуально сприймаються колірними атрибутами, як світлота, насиченість і колірний тон. Підсумком

досліджень щодо вдосконалення моделі СІЕ (МКО) стало створення в 1976 році її наступниці, що отримала назву СІЕ  $L^*a^*b^*$  (скорочено СІЕ Lab) (рис. 3.14). У цій моделі абстрактні параметри  $x$  і  $y$  були замінені на реальні параметри:

- $a$  – спектр кольорів діапазону від зеленого до червоного;
- $b$  - спектр кольорів діапазону від синього до жовтого;
- $L$  - світлота (Lightness), що представляє собою аналог яскравості.

Перевагою даної колірної моделі стала не тільки та обставина, що вона ефективно вирішила проблему розробки рівноконтрастного колірного простору, але також і те, що опис кольору в цій системі фактично моделює процес подання кольору апаратом людського зору.

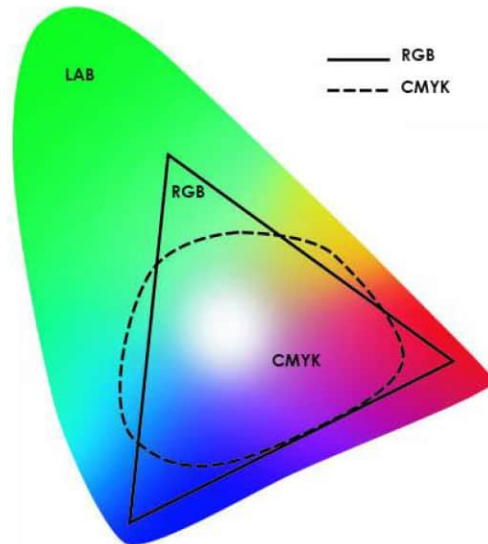


Рисунок 1.14 – Колірна модель Lab відносно RGB та CMYK

Також Lab-модель покликана вирішити проблему універсального підходу до репродукування зображень, пов'язану з використанням різних типів моніторів і пристроїв друку. В силу своєї незалежності від апаратних засобів вона дозволяє відтворювати одні і ті ж кольори незалежно від особливостей пристрою (монітора, принтера або комп'ютера), що використовується для створення або виведення зображень. Оскільки палітра кольорів цієї моделі перекриває колірні палітри RGB- і CMYK-моделей, то в ряді сучасних графічних програм ця модель використовується в якості внутрішньої моделі

для реалізації взаємного конвертування RGB- і CMYK-моделей. При роботі з графічними пакетами колірну модель Lab зазвичай використовують в зміні яскравості зображення без спотворення його колірних тонів (відтінків).

Сьогодні колірний простір CIE Lab (або, за іншою записи, CIE  $L^* a^* b^*$ ) є «рідним» для багатьох графічних пакетів та широко застосовується для всіх математичних розрахунків, вироблених комп'ютерами при роботі з кольором. Крім того, при корекції цифрових зображень криві в  $L^* a^* b^*$ -моделі дають користувачеві комплект можливостей, які доповнюють традиційний інструментарій растрових редакторів. Найбільш повно практичні аспекти застосування цієї системи для цифрової обробки зображень висвітлені в книзі Маргуліса.

### **1.3. Растр, растрування та лініатура**

Растр – набір точок (сітка з точок), що в сукупності формують зображення. В растровій графіці зображення представляється у вигляді масиву цифр. Саме тому при збільшенні подібних зображень починає виднітись структура типу мозаїки(сітки), яка складається з дрібних комірок.

Для того, аби відтворити готове зображення на папері були розроблені спеціальні 2 види технологічного підходу:

1. Використання півтонів. Для отримання півтонів використовується технологія растрування, вона з'явилась з процесу фотографування крізь сітку певної просторової частоти, в результаті чого отримується оригінал із сукупністю різного розміру точок.

2. Частотно-модульоване растрування. Ця технологія новіша за попередню та в ній відмовилися від регулярної структури растрів. Проте в цьому підході використовуються математичні методи розміщення точок в квазівипадкових позиціях фіксованого вже розміру.

Растрування має за мету створити ілюзії безперервного тону. В рамках амплітудного растрування цього можливо досягти створенням точок змінного

розміру, що в свій час розміщуються в регулярній матриці з рівновіддаленими центрами точок. Складені з великих точок області зображення вже сприймаються як більш темні тони, а невеликі стають світлішими.

Растрову форму можна описати такими параметрами:

- частота растру (лініатура);
- форма напівтонової точки;
- кут повороту.

Для отримання на роздрукованому зображенні ілюзії безперервних тонових переходів, зображення розбивається на безліч окремих елементів - растрових точок (рис. 1.5). Цей процес має назву – растування.

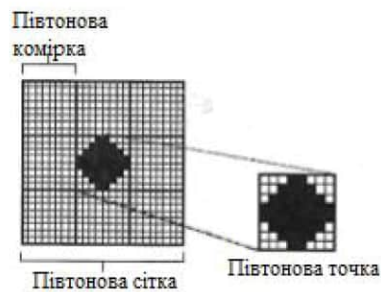


Рисунок 1.15 – Напівтонова (растрова) сітка

Під час растування зображення представляється у вигляді матриці півтонових осередків. Розмір напівтонового осередку матриці, що визначає лініатуру (lpi), може змінюватися, а ось відстань між друкованими точками фіксована і залежить від роздільної здатності принтера (dpi).

Частота (лініатура) растра характеризує растрову структуру кількістю її елементів (ліній з растрових точок) на одиницю довжини. Вимірюють її в лініях на дюйм (lines per inch, lpi), наприклад, 150 lpi. Типовий ряд лініатур: 150 lpi, 170 (175) lpi, 200 lpi (Рис. 1.16). [7]



Рисунок 1.16 – Вплив лініатури на якість зображення

Наприклад, лініатура растра 150 lpi означає, що принтер може формувати 150 півтонових осередків на дюйм.

#### **1.4. Особливості друку і роздільна здатність струйних принтерів**

Серед кольорового друку переважає струйна технологія. Для роздруківки знімків, отриманих цифровими камерами, в основному використовуються струменеві фотопринтери, що мають високий колірний обхват і широкий динамічний діапазон. Це досягається за рахунок використання підвищеної кількості кольорів (чорнила) - від шести до дванадцяти, а також впровадження додаткових технологічних рішень, включаючи друк краплями змінного розміру, нанесення декількох точок барвника на одне місце і використання витончених схем псевдозміщення. Перераховані фактори наближають струменеві принтери до категорії друкуючих пристроїв з безперервною передачею тону.

На відміну від лазерних принтерів, в них для відтворення зображення використовуються краплі чорнила. Друкуюча головка струйного принтера складається з набору сопел, що вистрілюють на папір маленькі крапельки рідкої фарби. Поряд з апаратними засобами (кількість сопел, розширений набір чорнила) для передачі кольорів і відтінків використовуються програмні засоби, в яких реалізовані алгоритми створення різних растрів (растрування), які опираються на обмежену роздільну здатність людського ока та особливості сприйняття зображень мозком. Якість передачі кольору і чіткість (колірна роздільна здатність) картинки сильно залежать від використовуваного алгоритму і вимагають для своєї реалізації високої роздільної здатності.

Однак, на відміну від лазерних принтерів і фотоскладальних автоматів, тут підхід до поняття роздільної здатності дещо інший. Частково це пов'язано з тим, що для струйних принтерів використовуються інші способи растрування (причому різні у різних виробників), в основі яких лежить комбінація стохастичного і напівтонового растрування. Тому для них не застосовується

поняття лініатури, а значить, і базуються на цьому понятті способи розрахунку роздільної здатності. Крім того, для струйного принтера в якості характеристик дозволу виробники наводять два параметри: роздільна здатність в dpi і розмір (обсяг) краплі. Сучасні струменеві принтери можуть похвалитися мікроскопічним розміром краплі - об'ємом 1-5 піколітрів - і надвисокою роздільною здатністю 4800 або навіть 5760 dpi. Але ці цифри зовсім не означають, що «чим менше (більше), тим обов'язково краще» - кінцеву якість струменевої фотографії залежить не тільки від того, які краплі і з якою точністю вистрілюються на папір, але і наскільки розумно вони вистрілюють.

В струйних принтерах розрядність також виражається в точках на дюйм, хоча правильніше було вимірювати його в краплях на дюйм. Крім того, на відміну від офсету або лазерного принтера, тут з-за сильного перекриття точок на папері класичне поняття роздільної здатності підмінюють іншим поняттям – адресована роздільна здатність. Саме відмінністю в зазначених поняттях можна пояснити декларовані виробниками надвисокі значення роздільних здатностей. Це вимагає додаткового пояснення. Справа в тому, що при здатності 4800 dpi розмір осередків сітки, що формується вихідною розрядністю, дуже малий - всього 0,005 мм. А крапля обсягом 2 пл має діаметр 0,0156 мм, тобто при всьому бажанні не зможе поміститися в клітинку і обов'язково захопить сусідні. В цьому випадку говорити про класичну роздільну здатність (коли ми можемо розрізнити краплі в сусідніх осередках) годі й говорити. І тому виробники придумали для розрядності нову назву - адресоване (або оптимізоване).

### **1.5. Узгодження пристроїв, CMS**

Для реалізації узгодження кольірних спектрів пристроїв потрібен надійний «перекладач» при обміні даними між пристроями системи. Такий перекладач був створений і названий Color Management System (система управління кольором, CMS). Подібні системи почали активно розроблятися різними

фірмами в зв'язку з широким розповсюдженням засобів обчислювальної техніки і їх впровадженням в поліграфічному виробництві та фотографії. Їх метою ставилося забезпечення автоматизованого контролю кольору на різних стадіях його відтворення і забезпечення узгодженої кольорами між різними відтворюють колір пристроями і матеріалами за рахунок компенсації колірних спотворень, властивих цим пристроям і матеріалами, а також колірних спотворень, викликаних розбіжністю їх колірних обхватів. Система управління кольором - це набір програмних засобів, призначених для узгодження колірних просторів різних компонентів настільної видавничої системи (сканерів, моніторів, принтерів, фотокамер, фотонабірних автоматів і друкуючих машин) з метою отримання узгодженого відтворення кольору на всіх етапах підготовки зображення для друку. Таким чином, призначення системи управління кольором полягає в компенсації різниці в способах відтворення кольору складовими частинами комп'ютерного апаратного забезпечення. В ідеалі це означає, що кольори, які ви бачите на екрані монітора, будуть без спотворення відтворені при друку. Це також має на увазі, що створений вами колір буде виглядати однаково скрізь, незалежно від використовуваних прикладних програм, моніторів або операційних систем. Однак фактично завданням систем управління кольором (CMS) є не досягнення повної відповідності кольору на екрані монітора кольором, отриманого при друку (що неможливо), а максимальне наближення цих квітів один до одного і досягнення їх передбачуваності. Використання CMS дає найбільший ефект при створенні публікацій, виведених на друк пристроями з низьким колірним охопленням: друкують машини, кольоропроби та настільні принтери. Це реалізується за рахунок того, що система управління кольором здійснює перетворення кольору пристроїв з високим колірним охопленням (монітори) в колірний простір пристроїв з більш низьким колірним охопленням. Тому видимі на екрані монітора кольори будуть адекватно відображені при друку. [8]

Прототипом сучасних систем управління кольором є система ColorSync, створена компанією Apple. Нею вперше була реалізована інтеграція системи управління кольором з операційною системою комп'ютера. Такий підхід забезпечує найбільшу ефективність, оскільки всі апаратні і програмні компоненти, що приєднується до системи, можуть безпосередньо включатися в систему управління кольором. Для забезпечення сумісності системи ColorSync з іншими платформами фірмою Apple було ініційовано створення Міжнародної комісії з кольором (International Color Consortium, ICC), засновниками якої стали провідні виробники операційних систем, програм і апаратури, включаючи Adobe, AGFA, Apple, Kodak, Silicon Graphics, Sun і ін. (пізніше в його склад увійшли нові члени, число яких на даний момент налічується більше 80). Підсумком роботи ICC є розробка архітектури систем управління кольором і специфікації формату колірної профілю, що лежить в основі функціонування систем управління кольором. У 2005 році була випущена остаточна версія специфікації формату колірної профілю, яка в тому ж році була прийнята в якості офіційного міжнародного стандарту Міжнародною організацією зі стандартизації (ISO) - ISO 15076-1. Фактично зусиллями міжнародної комісії був розроблений і впроваджений новий технологічний стандарт опису колірної простору пристроїв і способів можливої конверсії даних про зображення з одного колірної простору в інший. Запропонована ними система управління кольором включає чотири головні компоненти:

- апаратно-незалежний колірний простір
- колірні профілі, які визначають колірні характеристики окремих пристроїв системи відтворення кольору
- модуль управління кольором (СММ), який «розшифровує» знаходиться в профілі пристрою інформацію і виконує на її основі перетворення колірної інформації з одного колірної простору в інший.
- способи конвертації (Rendering Intents) кольорів, що не потрапляють в гаму того чи іншого пристрою.

Принцип дії системи управління кольором нагадує розглянуту вище ситуацію. В основі її функціонування лежить застосування апаратно-незалежної колірної моделі, використовуваної в якості спільної мови спілкування між окремими пристроями системи. У термінології ICC цей простір називається сполучним колірним простором, який часто називають внутрішнім колірним простором CMS. Функції такого простору виконує стандартне апаратно-незалежний колориметричний колірний простір CIE: XYZ або Lab. Перетворення колірного простору будь-якого пристрою (наприклад, сканера) в колірний простір моделі Lab або XYZ не представляє ніяких проблем. Складнощі виникають при виконанні перетворення у колірний простір вивідного пристрою, що пов'язано з його більш низьким колірним охопленням. Як монітор не може відобразити всі кольори, які сприймає око людини, так принтер не може відтворити всі кольори, які відображаються монітором. Модуль управління кольором здійснює коректний перерахунок невідтворених відтінків. Цей процес отримав назву процедури стиснення колірних просторів (gamut mapping). Він дозволяє скоординувати колірні охоплення вхідних і вихідних пристроїв таким чином, щоб невідтворювані тони були замінені близькими до них відтвореними тонами. Іншими словами, «більше» вхідний колірний простір як би «втягується» всередину «меншого» колірного простору. Наприклад, якщо сканований червоний колір не може бути надрукований вашим принтером, використання процедури gamut mapping призведе до того, що при друку буде відтворено відтінок, найбільш близький до оригінального червоному. Для реалізації даної технології розроблено ряд різних алгоритмів перенесення (або перерахунку) кольорів, що лежать поза колірним охопленням цільового пристрою. [9]

За такою технологією зображення в форматі RGB конвертується у колірну модель Lab, після чого під час перетворення Lab → CMYK будуть втрачені ще кольори, яких немає в спектрі кольорів CMYK. Дана технологія

використана програмно з регулюванням глибини кольору. Тому перейдемо до вибору середовища розробки програмного продукту.

### **1.6. Вибір середовища розробки**

В якості середовища розробки було обрано Python у зв'язку з його універсальністю, великою кількістю бібліотек та простотою освоєння.

Python – це високорівнева, інтерпретована мова програмування загального призначення, яка широко використовується для розробки широкого спектру додатків, від веб-сервісів до наукових обчислень та обробки даних. Програми на Python відомі своєю читабельністю та концептуальною простотою, що суттєво прискорює процес розробки та полегшує підтримку коду. Завдяки крос-платформній природі, програми, написані на Python, можуть працювати на різних операційних системах, включаючи Windows, macOS та Linux.

На відміну від Delphi, Python не має вбудованого візуального конструктора форм у тому ж розумінні. Розробка графічного інтерфейсу користувача (ГІК) у Python зазвичай здійснюється за допомогою спеціалізованих фреймворків, таких як Tkinter (стандартна бібліотека), PyQt, Kivy, або PySide. Ці фреймворки надають компоненти для побудови ГІК, а візуальний дизайн реалізується переважно програмно, хоча існують і візуальні конструктори для деяких із них (наприклад, Qt Designer для PyQt). Це дозволяє програмісту зосередитись на внутрішній логіці програми, ефективно інтегруючи її з інтерфейсом.

Перевагами Python як мови програмування є:

- завдяки чіткому та зрозумілому синтаксису, Python швидко освоюється новачками;
- наявність тисяч готових бібліотек (наприклад, Pillow для обробки зображень, NumPy та SciPy для наукових обчислень) значно прискорює розробку, дозволяючи не "винаходити велосипед";

- можливість розробки програм для різних операційних систем, а з деякими фреймворками – навіть для мобільних платформ;
- хоча це не завжди візуальне конструювання, як у Delphi, швидкість написання коду та прототипування в Python є дуже високою;
- величезна спільнота користувачів та розробників забезпечує постійну підтримку, велику кількість документації та прикладів.

Середовище розробки для Python (наприклад, PyCharm, VS Code) зазвичай складається з наступних елементів:

- редактор вихідного програмного коду;
- інтерпретатор Python (який виконує роль компілятора та відладчика);
- менеджер пакетів (наприклад, pip) для встановлення та управління бібліотеками;
- інтегровані інструменти відладки.

Python є оптимальним вибором для тих, хто бажає швидко навчитися розробляти ефективні програми, від простих сценаріїв до складних багатокомпонентних проектів. Незважаючи на свою простоту, середовище розробки має потужний потенціал та дозволяє реалізовувати навіть складні багаторівневі проекти.

### **1.7 Постановка задачі роботи**

Дослідивши аналітичну складову дипломної роботи можна зробити висновок, що поширеним методом зменшення глибини кольору є видалення кожного  $n$ -го елемента спектру, від чого втрачаються важливі відтінки та розмиваються краї в плямах на зображеннях мікроструктури металевих сплавів. Внаслідок чого під час дослідження мікроструктури сплаву дослідники отримують похибки, бо зображення змінилося після друку через мале значення лініатури під час друку фото. Лініатура в свою чергу залежить від глибини кольору, тому розробка методу зміни глибини кольору для підвищення

лініатури є актуальною для практичних досліджень зображень мікроструктури металевих сплавів.

Планований програмний додаток має містити в собі наступний набір можливостей:

- відкриття зображення та формування інформації про нього (параметри, кількість кольорів, глибина кольору), побудова гістограм для кожного каналу кольору;
- зменшення кількості кольорів (зменшення глибини кольору) у зображенні внаслідок керування гістограмами (обрізання, зсув);
- керування значенням лініатури друку зображення за рахунок зменшення глибини кольору;
- збереження результуючого зображення.

## 2. ТЕОРЕТИЧНА ЧАСТИНА

### 2.1. Конкретизація задачі роботи

Для коректної передачі кольорів використовується колірний профіль - докладний опис властивостей відтворення кольору конкретним пристроєм (файл з розширенням \*.icc або \*.icm). Це таблиця, де кожному значенню RGB або CMYK відповідає конкретне значення колірної моделі Lab. Для друку металографічних зображень не потрібні всі кольори колірних моделей RGB та CMYK, тому таку таблицю можна зменшити (кількість кольорів).

Під час друку є ті частини в колірній моделі, що не друкуються, тому можливо зменшити кількість кольорів (рис.2.1). Адже при растріванні зображення представляється у вигляді матриці півтонових комірок, розмір якої визначає лініатуру ( $l_{pi}$  - кількість растрових точок на дюйм): може змінюватися, а от відстань між друкованими точками фіксована і залежить від роздільної здатності принтера ( $dpi$ ).

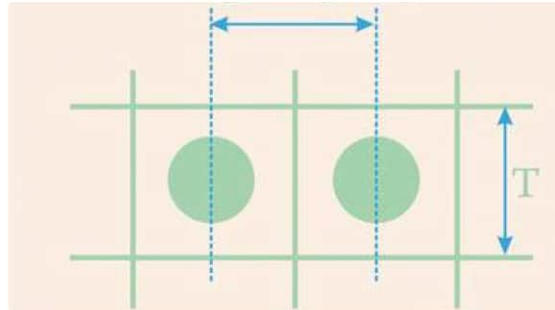


Рисунок 2.1 – Лініатура

Відношення геометричної роздільної здатності принтера до лініатури растра дає розмір сторони комірки растра, що вимірюється в точках принтера. При цьому визначення кількості відтінків  $N$  (глибина кольору), що передається растром, задається:

$$N = [dpi/l_{pi}]^2 + 1, \quad (2.1)$$

де 1 – білий колір (пуста растрова комірка)

Як вказувалось вище, поставлене завдання – зменшити кількість кольорів в зображенні для коректного (без втрат) його друку на папері. Фактично зображення мікроструктури металів зазвичай однотонні або в градаціях сірого. Тому замість того, щоб в ньому при зменшенні кількості кольорів видаляти кожен  $n$ -й колір зі спектру кольорів, краще звернутись до властивостей самого зображення.

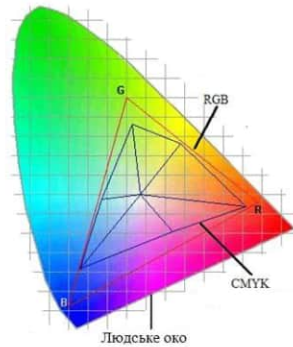


Рисунок 2.2 – Колірні моделі RGB та CMYK відносно спектру видимих кольорів

В колірній моделі RGB існує обмеженість колірного охоплення, яка пояснюється тим, що за допомогою адитивного синтезу неможливо отримати всі кольори видимого спектру. Особливо такі кольори як чистий блакитний чи чистий жовтий не можуть бути точно відтворені на екрані. В свою чергу модель CMYK має більш вузький колірний діапазон в порівнянні з RGB (Рис. 2.1.2), вона не може відтворити на екрані кольори RGB, а також ряд специфічних кольорів, як наприклад металевий. Прикладом цього обмеження є зображення срібла, яке в друкованому варіанті вже не так виглядає, як сприймає око людини. [3]

Тому, оскільки принтер не може надрукувати всі 16.7 млн кольорів, то при керуванні кольором зображення спочатку необхідно відмовитись від чистих синього, червоного та зеленого кольорів, після чого звернутись до властивостей самого зображення. Зображення мікроструктури металів зазвичай однотонні або в градаціях сірого, тому діапазон кольорів для друку потрібно обмежити. Якщо зображення у градаціях сірого та жовтого, тоді неможливо

відкинути інші кольори спектру, адже збільшення комплементарного (протилежного на колі кольорів) кольору призводить до зменшення яскравості головного кольору. Відповідно зменшення комплементарного кольору підвищує яскравість основного. Для зображення в градаціях жовтого потрібно зменшити його комплементарний колір, в даному випадку ним виступає синій.

Лініатура пропорційна до глибини кольору (1), тому, якщо відмовитись від тих кольорів, що взагалі не друкуються (звернутися до колірною профілю принтера) та зменшити кількість друкованих кольорів із врахуванням особливостей зображення, то внаслідок друковане зображення буде більш наближене до зображення на екрані монітора та матиме кращу якість у порівнянні з методом видалення кожного n-го кольору спектру. Таким чином можна наблизитись до мети – узгодження глибини кольору друку з геометричною роздільною здатністю фото-пристрою мікроскопа.

Можна виділити 2 основних пункти для коректного зображення мікроструктури металевих сплавів в друкованому варіанті:

1. Зменшити глибину кольору.
2. Використати властивості зображення для корекції кольорів.

Тому і потрібно розробити програмний продукт, що зможе після відкриття зображення формувати інформацію про нього (параметри, кількість кольорів, глибина кольору) на гістограмах для кожного каналу кольору, зменшувати кількість кольорів у зображенні, коректувати кольори в зображенні з використанням його властивостей та зберегти оброблене зображення.

## **2.2 Лініатура та масштаб**

Друковане зображення складається з окремих точок, тобто представляє собою растр. Якість друкованого фото напряму залежить від детальності растру. Вважається, що зображення в 300 точок на дюйм достатнє для якісного поліграфічного друку. Звідси легко порахувати, що зображення, що відповідає звичайному фото 10x15, має мати розміри приблизно 1200x1600 точок (2

мегапіксела) для якісного друку. Тому одним з факторів якісного друку фото можна вважати принтери з високою роздільною здатністю, адже такий принтер краще формує переходи між точками (рис. 2.3). [2]

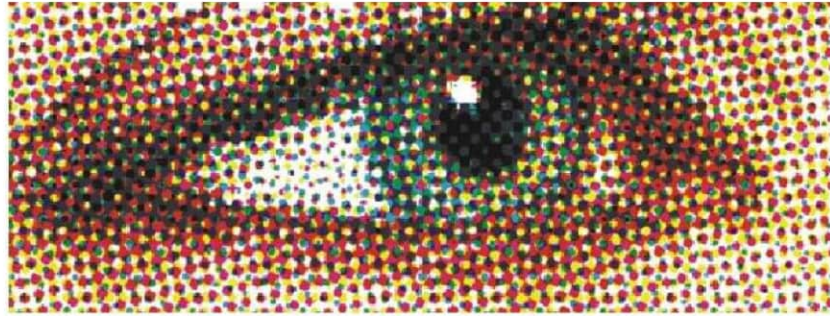


Рисунок 2.3 – Залежність якості друку зображення від лініатури

Другим фактором якісного друку є велике значення лініатури (lpi – кількість растрових точок на дюйм). Чим більше значення лініатури – тим якісніше виглядає надруковане зображення (при лініатурі малій крапки великі, видна відстань між ними, однак на зображенні з високим значенням лініатури переходи тонів плавні, точки менші, як і відстань між ними), на ньому видніються всі дрібні деталі, що і потрібно досягнути у випадку фото мікроструктури сплавів. Адже задача націлена не відтворення зображення під час друку практично в тій же якості, яким людина його бачить на фото-пристрої мікроскопа. Для досягнення цього є 2 методи:

1. Лініатура рівна геометричній роздільній здатності монітора комп'ютера (100dpi), або наближена до неї;
2. Вручну регулювати розмір зображення на комп'ютері для відображення його таким, як на оптичному мікроскопі, а потім вивести на друк в такому масштабі, де відстань між сусідніми точками буде однаковою з відстанню між ними на екрані оптичного мікроскопа з урахуванням масштабу.

Лініатуру растру для друку зображень також можна задати самому. При цьому верхньою межею її є роздільна здатність принтера (його dpi), але в такому випадку зображення має бути монохромним. Однак перший пункт, лініатуру 100ліній/дюйм цілком можна реалізувати, правильно застосовуючи

при цьому знання про глибину кольору та вміло керуючись нею. Зазвичай значення лініатури для друку визначається з урахуванням властивостей паперу, якості фарб друкарських та самого принтера.

Для друку реальними фарбами принтера має місце явище – розтиск точок (dot gain) – збільшення розміру друкованих точок через поглинання фарби папером. Для якісного друку точки раstra не мають накладатись один на одного. При надто сильному збільшенні лініатури розтиск приведе до саме цього ефекту, від чого не те що не зросте якість друку, а навпаки – погіршиться. Візуально це призведе до того, що з'явиться «бруд» на зображенні там, де є локальні зменшення щільності паперу, бо навіть найякісніший папір не є абсолютно однорідним. В такому випадку краще вибрати варіант, коли лініатура наближеною до геометричної роздільної здатності принтера, або принаймні такою, щоб зображення на оптичному мікроскопі було візуально однаковим із тим, що буде надруковане на папері. Тому варто зробити невеликий екскурс в теорію мікроскопів для визначення, як саме зображення має бути точно передане на друк.

Мікроскоп – оптична система, що має 2 ступені збільшення та призначена для спостереження дрібних об'єктів, що близько знаходяться між собою, з великим збільшенням і з більшою роздільною здатністю. Ступені збільшення мікроскопу – об'єктив та окуляр. Перший утворює дійсне збільшене зворотне зображення предмета в передній фокальній площині окуляра (4-10 лінз). Другий діє як лупа та утворює уявне зображення на відстані найкращого бачення (2-5 лінз). [11]

Основні характеристики мікроскопу:

– збільшення мікроскопу – відношення розміру остаточного зображення до розміру предмету.

– числова апертура – характеризує роздільну та світлозбиральну здатність мікроскопу.

– роздільна здатність – характеризує мінімальну відстань між двома точками предмету, що видимі на зображенні роздільно.

Роздільна здатність® обернено пропорційна межі дозволу, яка в свою чергу залежить від довжини хвилі світла ( $\lambda$ ) та числової апертури (A) мікроскопу. З чого формула роздільної здатності мікроскопу:

$$R = 2A / \lambda = (2 * n * \sin u) / \lambda, \quad (2.2)$$

де  $n$  – коефіцієнт заломлення,

$u$  – апертурний кут – той кут, під яким з точки, що знаходиться в головному корпусі об'єктиву, видно радіус передньої лінзи об'єктиву.

Визначення роздільної здатності – перший крок для якісного друку зображення, адже далі ця відстань між точками буде співставлятися до відстані між точками на екрані монітору, які між собою мають бути рівними або наближеними. Стандартна роздільна здатність оптичного мікроскопу – 0.3мкм.

Око спостерігача зможе сприймати дві точки як роздільні, якщо кутова відстань між ними буде не менше кутової межі роздільної здатності очей. Для того щоб очі спостерігача могли повністю використовувати роздільну здатність мікроскопа, необхідно мати відповідне видиме збільшення. Корисне збільшення - це видиме збільшення, при якому око спостерігача буде повністю використовувати роздільну здатність мікроскопа, тобто роздільна здатність мікроскопа буде така ж, як і роздільна здатність очей.

Зображення мікроструктури металів в більшості своїй мають масштаб, що зазначений на самому зображенні. Масштаб зображення - співвідношення лінійних розмірів зображеної ділянки до дійсної. В даному випадку ідеально, якщо зображення мікроструктури сплаву буде наближено ідентично виглядати що на фото, що в друкованому варіанті. Тобто відстань між точками на фото буде наближена відстані між точками в друкованому варіанті з врахуванням коефіцієнту масштабу.

Оскільки відстань між друкованими точками фіксована і залежить від роздільної здатності принтера ( $d_{pi}$ ), то, виходячи з формули глибини кольору,

для збільшення лінійності потрібно зменшити кількість кольорів, які саме будуть підлягати видаленню – розглянемо далі.

### **2.3 Видалення кольорів**

В колірній моделі RGB існує обмеженість колірною охоплення, а в CMYK вона ще більша. Модель RGB не може відтворити всі кольори, що бачить людина в реальному житті, а CMYK не може відтворити ще й чисті синій, червоний та зелений кольори. Тому для початку в зображенні потрібно видалити або замінити 255-й відтінок кожного каналу, що всерівно не буде прийнятно видано в друкованому варіанті.

Далі потрібно звернутись до властивостей самого зображення. Зображення мікроструктури металів зазвичай однотонні або в градаціях сірого, тому діапазон кольорів для друку потрібно обмежити.

Візьмемо для прикладу зображення мікроструктури сплаву, що забарвлене жовтими та оранжевими тонами (Рис.2.4). Для наглядності візьмемо його R-канал моделі RGB (Рис.2.5).

На гістограмі вісь X – відтінки червоного каналу від 0 до 255, а вісь Y – кількість пікселів у відсотках (відносно загальної кількості пікселів на зображенні), що використовують той чи інший відтінок каналу. З гістограми можна зробити висновок, що відтінки в зображенні використовуються приблизно від 100 до 250. Однак пологі краї гістограми та ті відтінки, що поруч них та не набирають навіть 1% присутності в зображенні – відтінки, що рідко застосовуються в зображенні. Їх можна онулити або замінити значення на ті відтінки, що найближче до них знаходяться і не підлягають онуленню. У випадку цієї фігури на гістограмі було прийнято рішення видалити ті відтінки,

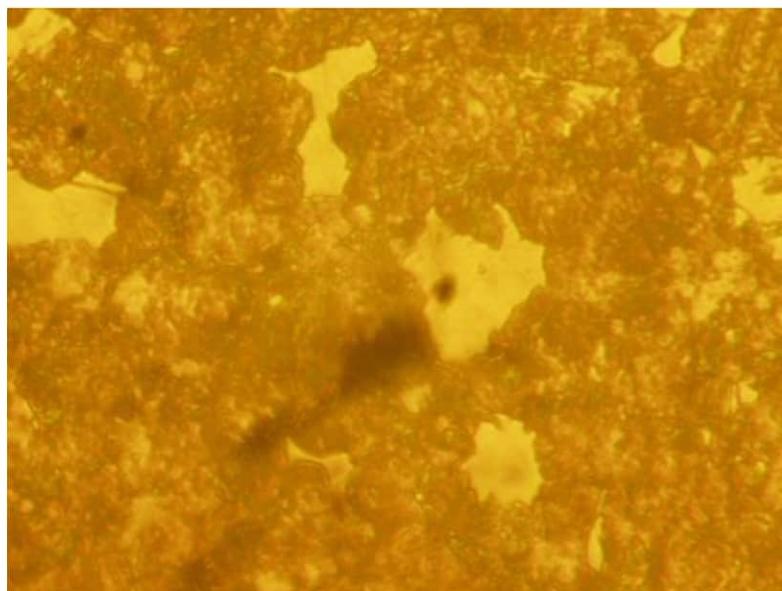


Рисунок 2.4 – Фото мікроструктури металевого сплаву

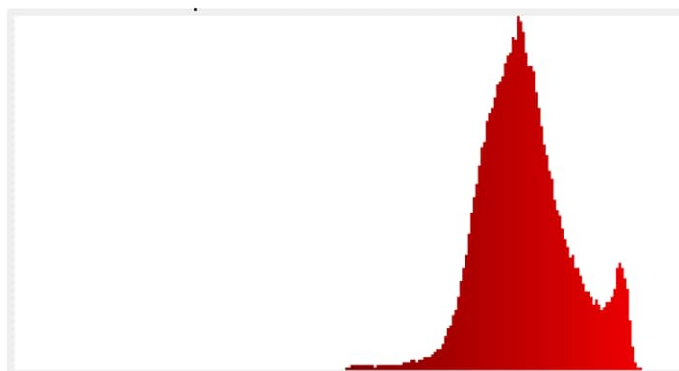


Рисунок 2.5 – Гістограма R-каналу мікроструктури металу що складають менше 10% на гістограмі, за рахунок цієї процедури кількість кольорів, що використовується в зображенні зменшиться приблизно на 11%, а кількість ненульових відтінків R стане майже вдвічі меншою (Рис. 2.6).

Відтінків R-компоненти у зображенні стало менше, але в таблиці кольорів вони знаходяться приблизно з 170 по 240 місце - 70 із 255 відтінків.

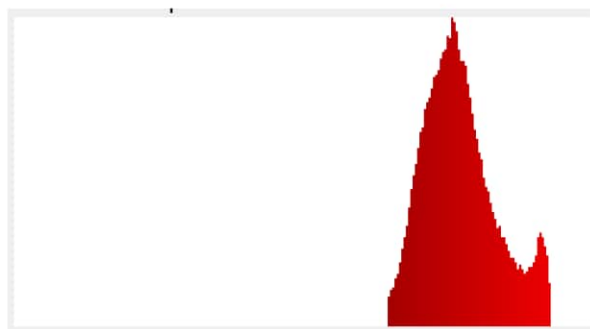


Рисунок 2.6 – Обрізання R-каналу мікроструктури металу

Під час друку принтер буде резервувати місця для всіх 255 відтінків, навіть якщо вони не використовуються в зображенні і не будуть йти на друк. В такому випадку потрібно розробити своєрідну матрицю переадресації, що перемістить всі відтінки до 0 (Табл.2.1), а фігура на гістограмі каналу в такому випадку зміститься вліво до 0 (Рис. 2.7). Після проведення таких дій зарезервовані для друку відтінки видаляються, принтер сприйматиме нову таблицю кольорів як ту, що в 3 рази менша за попередню. Внаслідок чого зменшиться кількість відтінків для друку і згідно з формулою збільшиться лініатура друку. Хоч і зображення матиме вигляд вже не такий, як був до цього, але межі плям на зображенні будуть більш чіткими за рахунок збільшеної лініатури, що дасть можливість коректніше оцінити структуру сплаву.



Рисунок 2.7– Гістограма R-каналу після зміщення

Таблиця 2.1 – Матриця переадресації

Old	New
0	
...	
170	0
...	...
240	70
...	
255	

## 2.4 Додаткові методи підвищення якості друку

Разом з вищеописаним методом підвищення якості друку за рахунок збільшення лініатури є ще ряд інших способів, які можна використовувати після обробки для покращення якості вихідного зображення.

Геометрія растру. Зображення під час друку підлягають регулярному або стохастичному раструванню (Рис. 2.8). Стохастичні растри дають репродукцію без муару і, отже, роблять можливим застосування необмеженого числа фарб з будь-якими значеннями тонового діапазону. Стохастичні зображення мають високу роздільну здатність, але можуть призводити до зернистості в однорідних тональних ділянках. Регулярні растри дають однорідні тони і при високих лініатур мають роздільну здатність, еквівалентну стохастичним растрам.[12] Накладення регулярних растрових сіток під різними кутами призводить до утворення обумовленого поворотом растрів характерного візерунка і муару (візерунок, що виникає при накладенні двох періодичних сітчастих малюнків) від взаємодії малюнка об'єкта з растровою сіткою.

Оскільки стохастичне та регулярне растрування мають свої як переваги, так і недоліки, то на сьогоднішній день існує ряд програм для друку, які поєднують ці методи растрування, взявши з кожного тільки переваги. Цей метод растрування ще має назву cross-modulated (ХМ) растр. [13]

У типових гібридних растрах регулярне растрування застосовується для формування середніх тонів (10-90%), а стохастичне - для областей світла та тіней, щоб домогтися максимально точної передачі деталей, але уникнути

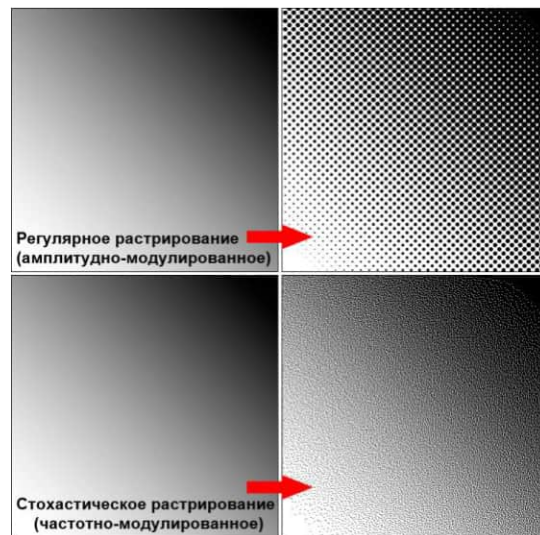


Рисунок 2.8 – Регулярне та стохастичне растрування зернистості в середніх тонах

Переваг у реалізації «гібридного» растра можна виділити багато, і ось лише деякі з них:

- можливість друку високими лініатурами там, де це було неможливо, з отриманням майже фотографічної якості навіть на неякісному папері або картоні;
- плавне, рівне і стабільне відтворення градієнтів без тонових «ступенів»;
- за рахунок застосування високих лініатур розетка класичного растра невидима неозброєним оком;
- відмінна деталізація в тінях і світлих областях за рахунок використання стабільно відтворюваних мінімальних растрових елементів;
- відсутність шумових ефектів, властивих стохастичному раструванню. [14]

Наступний метод підвищення якості друку – правильне визначення куту нахилу растра на зображенні, що підлягатиме друку. Кут нахилу растра - відносний геометричний кут нахилу растрових точок. Різні кути нахилу растра використовуються для можливості синтезу чистих кольорів в світлих областях зображення і усунення муару при друку з використанням регулярного методу растрування. Для тріадних фарб, використовуваних у пресі, використовуються

стандартизовані кути нахилу растра. Для блакитної (Cyan) фарби використовується поворот в 15 або 105 градусів, для пурпурової (Magenta) - 75 або 15 градусів, для жовтої (Yellow) - 0 або 90 градусів, для чорної (Black) - 45 або 135 градусів. [15] Неприпустимо для різних фарб використовувати одні і ті ж значення кута нахилу растра, оскільки при накладенні цих фарб виникне муар. При накладенні чотирьох тріадних фарб на папері з окремих растрових точок утворюється так звана розетка растру, геометрична форма якої дозволяє найбільш якісно розмістити растрові точки один щодо одного, що дозволяє найбільш чисто і якісно відтворювати кольори і відтінки. [16]

Далі розглянемо алгоритми роботи програмного додатку та алгоритм роботи користувача з додатком, розробленим в використанні методу, описаного в п.2.3.

## **2.5 Розробка алгоритму роботи користувача**

Після того, як користувач запустить програму для зміни глибини кольору зображень мікроструктури металевих сплавів, користувач матиме слідувати наступному алгоритму:

- відкрити нове зображення, натиснувши кнопку «відкрити» на боковій панелі та обравши фото мікроструктури сплаву зі своєї теки;
- дочекатися, доки індикатор оброблення зображення не дійде до кінця та доки не висвітиться над ним напис «Зображення опрацьовано»;
- перейти на вкладку «Обробка RGB» програми;
- переглянути гістограми складових R,G та B кольорів зображення, переглянути колірні дані про зображення на боковій панелі під написом «Оригінальне зображення» в групі «Результати обробки»;
- для видалення 255-го відтінку каналу поставити прапорець біля напису «і 255-й»;

- для видалення певного відсотку відтінків каналу (від найрідше вживаних до найчастіше вживаних у зображенні) в лічильнику біля напису «Обрізати» обрати бажане число, або власноруч його ввести (від 1 до 100);
- для зсуву фігури на гістограмі вліво чи вправо в лічильнику біля напису «Зсунути» обрати бажане додатне чи від’ємне значення (фігура може переміщуватись від 0 відтінку до 255-го);
- натиснути кнопку «ОК» на боковій панелі для застосування до зображення бажаних змін;
- переглянути зміни на нових гістограмах складових R,G та B кольорів зображення під оригінальними гістограмами, переглянути колірні дані про зображення на боковій панелі під написом «Оброблене зображення» в групі «Результати обробки», порівняти дані до та після, порівняти зображення до та після обробки;
- при необхідності повторити процедуру з п.5 або з п.1;
- перейти на вкладку «Оброблене зображення», клацнути на довільній точці зображення, щоб дізнатись складові R, G, B її відтінку;
- порівняти оригінальне та оброблене зображення на однойменних вкладках;
- перейти на вкладку «RG Діаграми», переглянути та порівняти однойменні діаграми до та після обробки зображення (чим темніше точка – тим більше такого відтінку в зображенні);
- для збереження обробленого зображення в форматі BMP натиснути на кнопку «Зберегти» на боковій панелі;
- вийти з програми.

## **2.6 Розробка структури програмного додатку**

Програма управління глибиною кольору растрових зображень розроблена на мові Delphi, в середовищі програмування Delphi 7. Як і більшість додатків,

розроблених на цій мові та в даному середовищі, мій проект складається з \*.exe(виконуваний файл) файлу, \*.dfm(проект), \*.pas(код додатку) та інших додаткових файлів для роботи в середовищі.

Програмний додаток складається з наступних процедур:

– DiagramRG - побудова RG діаграми для оригінального зображення. Виконується на 4 вкладці програми, де виводиться діаграма площини зі сторонами Red та Green, на якій можна побачити кількість пікселів (чим чорніша точка – тим їх більше), що використовують відтінок, що знаходиться на перетині цих координат. Наприклад, відтінок [10,15,0] не використовується в зображенні – точка біла, відтінок [222,210,0] використовується часто – відтінок пікселю темніший.

– DiagramNewRG - побудови RG діаграми для обробленого зображення. Дія цієї процедури аналогічна до попередньої, але ця діаграма оновлюється після кожного натискання на кнопку «ОК», тобто після кожної зміни в гістограмах користувачем.

– LeftRight - обчислення кількості нулів на гістограмі. Ця процедура допоміжна для процедур DiagramRG та DiagramNewRG, знаходить краї гістограм, щоб правильно перемістити фігуру.

– Gistogram - побудова R, G, B гістограм по оригінальному зображенню. В цій процедурі очищуються області для побудови та будуються гістограми трьох складових кольору піксела.

– Gistogram1 - побудова R, G, B гістограм по обробленому зображенню. Процедура аналогічна до попередньої, але ці гістограми будуються після обробки зображення. Гістограми очищуються та заново перебудовуються після кожного натискання кнопки «ОК», а саме – після щоразової зміни зображення користувачем.

– CounNull - занулення масиву кількості пікселів для всіх кольорів. Використовується ця процедура після відкриття нового зображення та під час кожної побудови обробленого зображення.

– CounSum - обчислення кількості кольорів у зображенні.

– RebuildOriginRGB - побудова обробленого зображення RGB. Ця процедура спрацьовує після натискання кнопки «ОК», а саме – після щоразової зміни зображення користувачем. Також в цій процедурі перераховуються кількості R,G, та B складових кольору та кількість відтінків, що використовує оброблене зображення, та виводяться на боковій панелі.

– ButtonOpenClick - відкриття нової картинки. Основна процедура, в якій завантажується нове зображення, обнулюються всі дані, що стосувались попередньо відкритого зображення, відкривається зображення, виводяться на боковій панелі дані про нього, спрацьовує індикатор обробки, підраховуються всі дані, що стосуються зображення, виводяться деякі на боковій панелі та запускаються процедури для нового зображення. Спрацьовує після натискання кнопки «Відкрити».

– FormCreate – процедура старту програми, в якій описуються параметри нової форми та її вкладок.

– BitBtnRebuildClick – процедура кліку на кнопці «ОК». Обчислення даних для зсуву гістограм та їх зсув, а також обтинання гістограм.

– ImageNewRGBMouseDown - при натисканні на зображенні отримаємо його R, G, B складові. Ця процедура працює на 2 вкладці програми, де відображається оброблене зображення, при кліку на нього біля курсору появляється коментар, в якому пишеться кількість R, G, та B складових піксела.

– Button1Click - збереження зображення. При натисканні на кнопку «Зберегти» оброблене зображення зберігається у форматі BMP.

## 2.7 Алгоритм роботи програми

Першою запускається процедура `FormCreate` (процедура старту програми). Після неї очікуються дії користувача, а саме – відкриття зображення мікроструктури металевого сплаву з допомогою кнопки «Відкрити» на бічній панелі. Після чого запускається процедура `ButtonOpenClick` (відкриття нового зображення), разом з якою запускаються процедури `DiagramRG`, `LeftRight`, `Gistogram`, `CounNull`, `CounSum` та стають доступними вкладки «Оригінальне зображення», «Оброблене зображення», «Обробка RGB» та «RG Діаграми». Далі користувач вводить дані для редагування гістограм у відповідні поля на вкладці «Обробка RGB» та натискає «ОК», від чого спрацьовують наступні процедури: `BitBtnRebuildClick` (процедура кліку на кнопці «ОК»), `DiagramNewRG`, `LeftRight`, `Gistogram1`, `CounNull`, `CounSum`, `RebuildOriginRGB`.

Отримавши оброблене зображення користувач має можливість на вкладці «Оброблене зображення» натиснути на довільну точку зображення та в підказці побачити складові RGB його відтінку, що здійснюється з допомогою процедури `ImageNewRGBMouseDown`. Після довільної обробки зображення користувач також має змогу зберегти оброблене зображення, натиснувши н кнопку «Зберегти» внизу бокової панелі, до якої прикріплена процедура `Button1Click`.

### 2.7.1 Алгоритм роботи процедури `FormCreate`

Процедура `FormCreate` відповідає за створення форми (інтерфейсу програми). В ній задаються параметри вікна та його розміщення на робочому столі користувача, колір фону вікна, видимість чи не видимій тої чи іншої вкладки на початку роботи з програмою та номер сторінки, яка буде відкрита автоматично після завантаження зображення. Блок-схема алгоритму роботи представлено на рис.2.9.

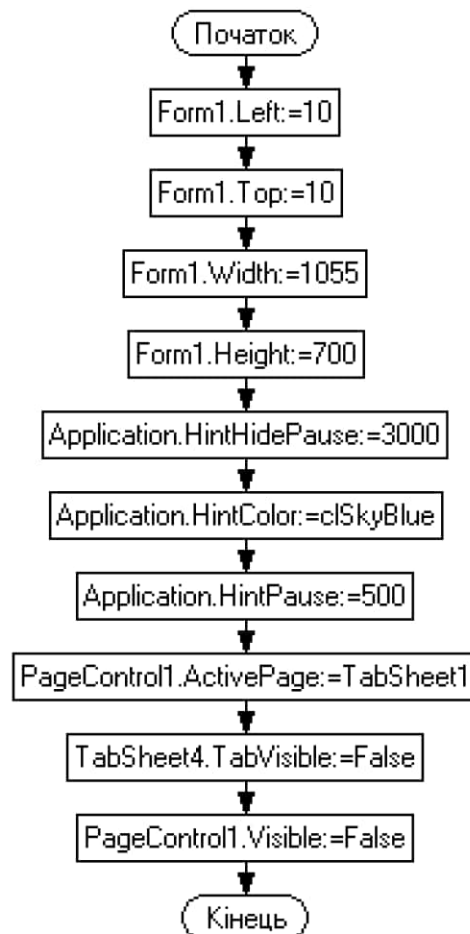


Рисунок 2.9 – Блок-схема процедури FormCreate

### 2.7.2 Алгоритм роботи процедури ButtonOpenClick

Процедура ButtonOpenClick відповідає за відкриття нової картинки після натискання кнопки «Відкрити», та обравши довільний файл для обробки зображення. Спочатку в ній формуються та очищуються всі поля та області, адже ця процедура може бути викликана і після обробки одного зображення. Далі формуються масиви кількості пікселів, кольорів цих пікселів, складових RGB кольорів цих пікселів та інше, що знадобиться для підрахунків та обробки. Після чого будується зображення на вкладці «Оригінальне зображення» та дві зменшені копії на вкладці «Обробка RGB», будуються гістограми каналів кольорів для оригінального зображення, виводиться інформація про кількість відтінків на боковій панелі, визначаються краї фігур на гістограмах та будуються діаграми RG.



Рисунок 2.10 – Блок-схема процедури ButtonOpenClick

### 2.7.3 Алгоритми роботи процедур **Gistogram** та **Gistogram1**

Процедура **Gistogram** відповідає за побудови R, G, B гістограм по оригінальному зображенню. На початку процедури йде очищення областей для гістограм, після чого для кожного каналу від 0 до 255 відтінку в цих областях будуються вертикальні відрізки, що означають відсоток кількості використання відтінку у вхідному зображенні. Процедура **Gistogram1** аналогічна до **Gistogram**, але якщо попередня запускала після завантаження зображення, то ця спрацьовує під час перебудови гістограм, а саме – після кліку на кнопку «ОК» на бічній панелі.

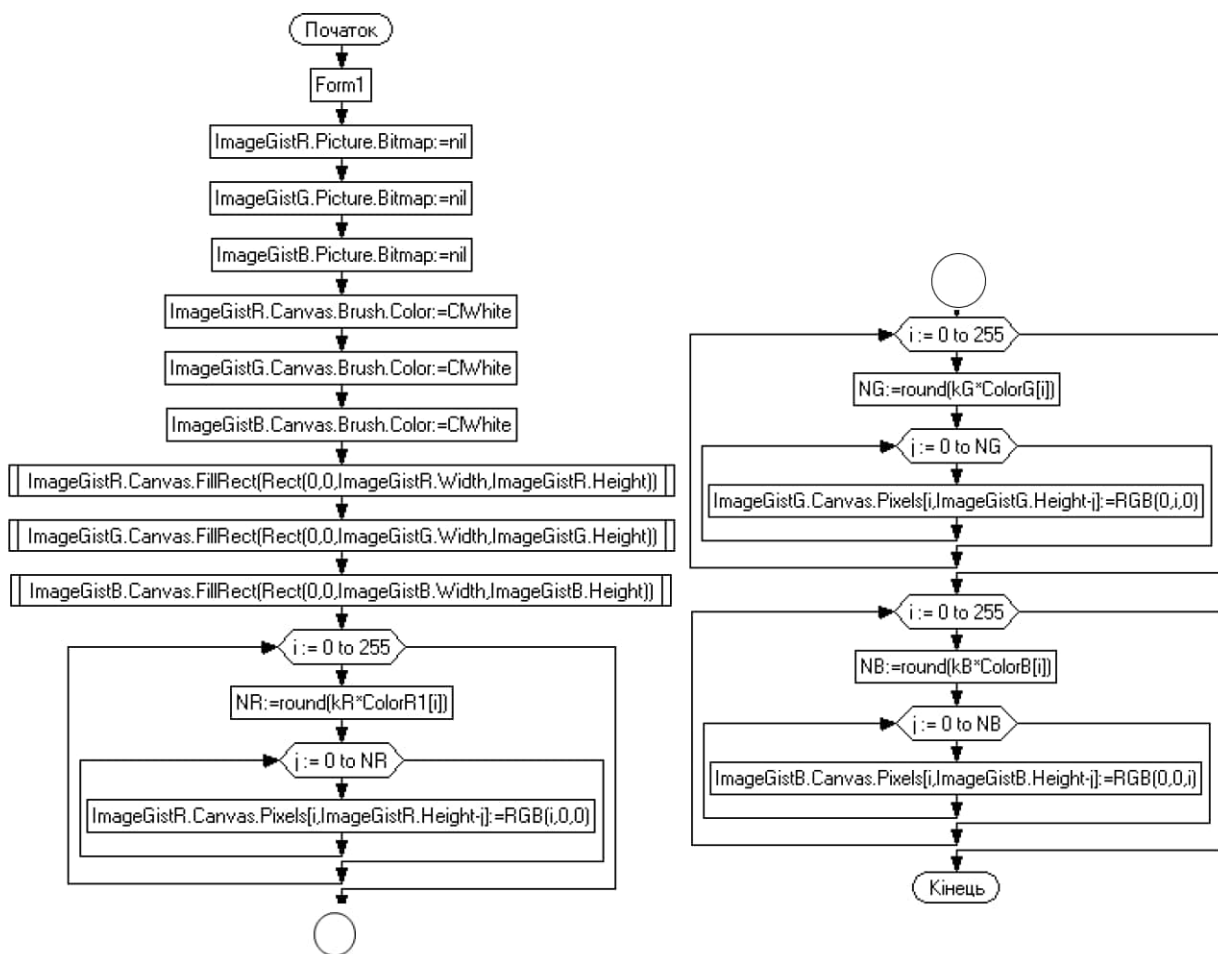


Рисунок 2.11 – Блок-схема процедури **Gistogram**

### 2.7.4 Алгоритм роботи процедури LeftRight

Процедура LeftRight (рис.2.12) розроблена для обчислення кількості нулів на гістограмі. Ця процедура допоміжна для процедур Gistogram, Gistogram1, DiagramRG, DiagramNewRG.

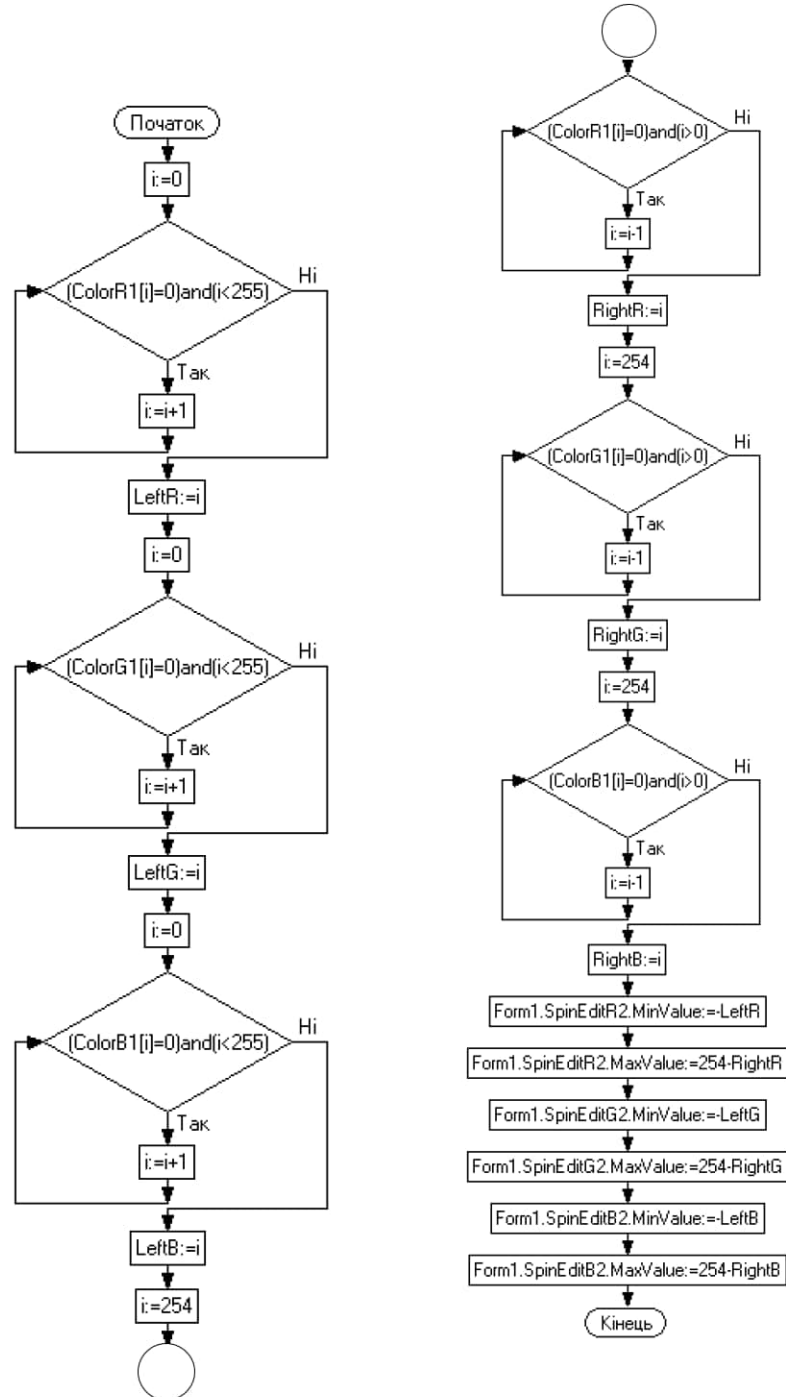


Рисунок 2.12 – Блок-схема процедури LeftRight

### 2.7.5 Алгоритми роботи процедур CounSum та CounNull

Процедура CounNull занулює масиви кількості пікселів для всіх кольорів, а CounSum обчислює кількість кольорів, що є в зображенні. Ці процедури є також допоміжними для процедур RebuildOriginRGB та ButtonOpenClick. Для коректного визначення кількості кольорів у зображенні було вирішено використати формулювання кольору як  $R+G*256+B*65536$  ( $R*256^0+G*256^1+B*256^2$ ), тобто в системі числення 256 (рядки з цим формулюванням (k) та формування масиву Coun[k] знаходяться в процедурах RebuildOriginRGB та ButtonOpenClick). Значення з CounSum виводиться в поле «всього кольорів» (рис.2.13 та рис.2.14).

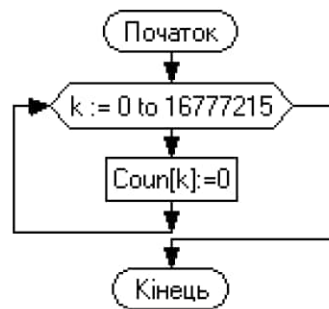


Рисунок 2.13 – Блок-схема процедури CounNull

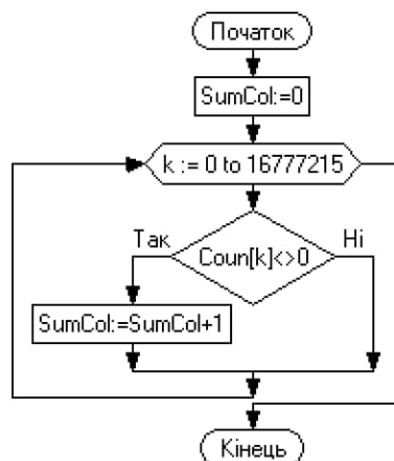


Рисунок 2.14 – Блок-схема процедури CounSum

## 2.7.6 Алгоритми роботи процедур DiagramRG та DiagramNewRG

Результат роботи процедур DiagramRG та DiagramNewRG знаходиться на 4 вкладці програми під назвою «RG Діаграми». Оскільки дані зображення мікроструктури металевих сплавів в градаціях жовтого та оранжевого, то для відтворення цих кольорів потрібні зелений та червоний канали колірної моделі RGB. З допомогою цих діаграм стає зрозуміло, наскільки мало кольорів застосовується в зображенні з можливих на перетину каналів R та G.

Принцип роботи процедур – побудова на області, де вісь  $x$  – R,  $y$  – G, точок, кожна з яких показує кількість у відсотках пікселів із тим чи іншим кольором. Якщо такого відтінку немає в зображенні, то точки білі, якщо є мало, то сірі, багато – чорні і подібним чином в градаціях сірого для наглядності. В першій процедурі – DiagramRG присутня побудова шкал (Рис.2.15). Друга процедура DiagramNewRG (Рис.2.16).

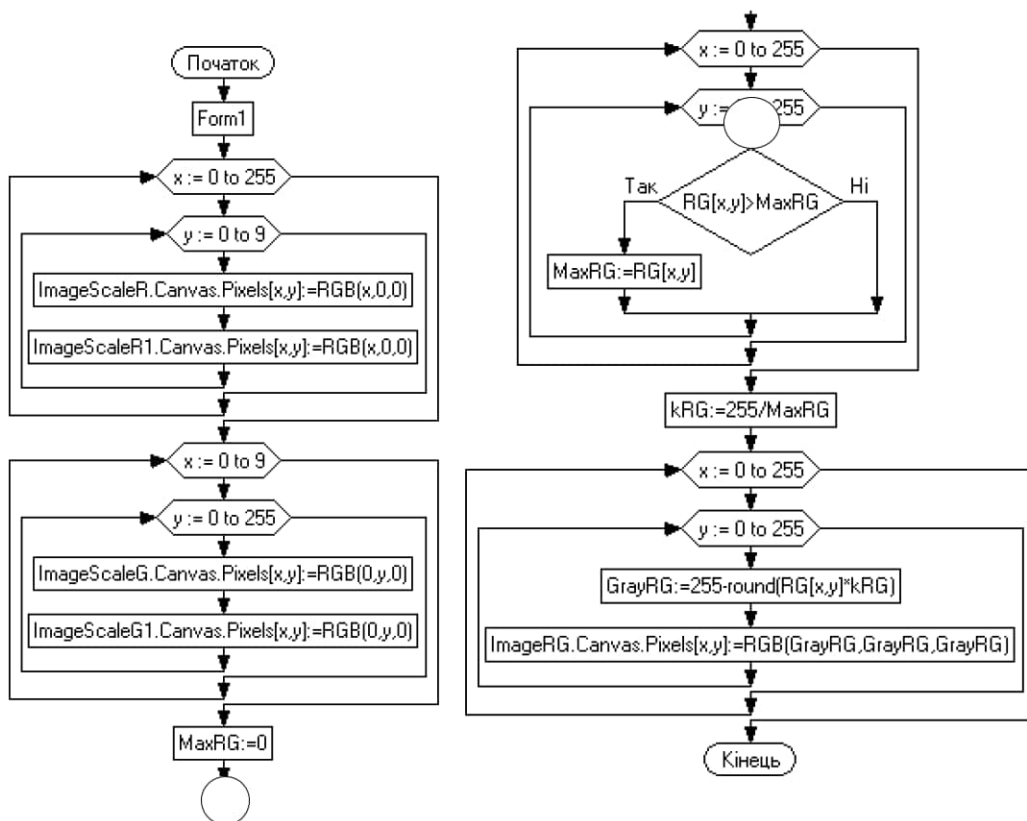


Рисунок 2.15 – Блок-схема процедури DiagramRG

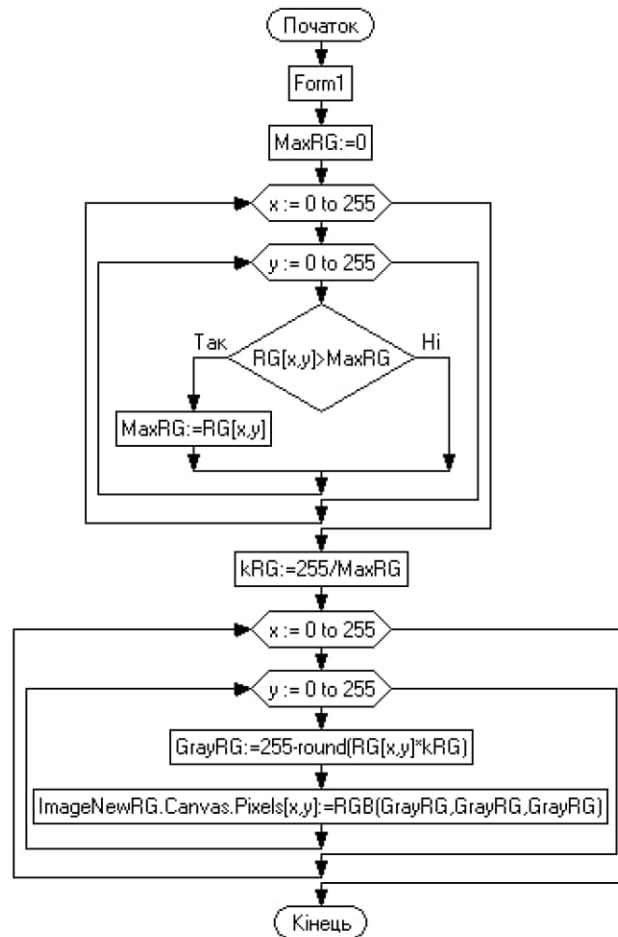


Рисунок 2.16 – Блок-схема процедури DiagramNewRG

### 2.7.7 Алгоритм роботи процедури BitBtnRebuildClick

Процедура BitBtnRebuildClick (Рис.2.17) – процедура кліку на кнопці «ОК» на боковій панелі програми. Користувач натискає цю кнопку, коли увів у всі дані для редагування поля значення та бажає застосувати зміни. В BitBtnRebuildClick спочатку зчитуються введені користувачем дані, потім перераховуються з використанням масиву ColorR1[i] – копії кольорів оригінального масиву кольорів ColorR[i]. Обробляються поля по порядку: найперші – обрізання гистограми, далі – видалення 255-го відтінку компоненти, завершують поля «Зсунути». Після чого викликаються на виконання процедури Gistogram1, RebuildOriginRGB, DiagramNewRG. Всі ці дії повторюються для кожного натискання «ОК», при чому оригінальні зображення, гистограми та діаграма залишаються незмінними впродовж всієї роботи з програмою.

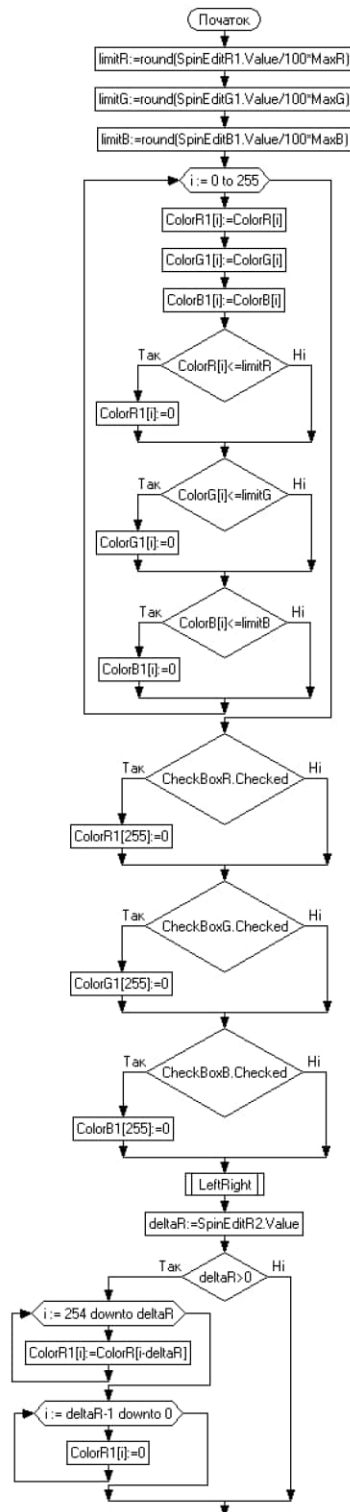
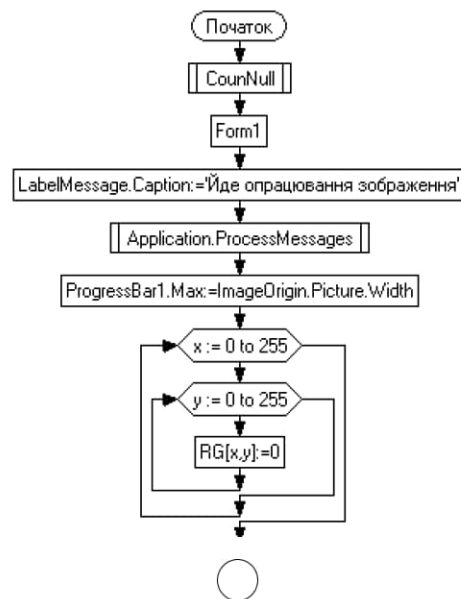


Рисунок 2.17 – Блок-схема процедури BitBtnRebuildClick

### 2.7.8 Алгоритм роботи процедури RebuildOriginRGB

RebuildOriginRGB (Рис.2.18) – процедура побудови обробленого зображення. Ця процедура спрацьовує після натискання кнопки «ОК». Починається процедура з занулення масиву кольорів, потім на боковій панелі вмикається індикатор та з'являється над ним повідомлення «Йде опрацювання

зображення». Далі зображення перераховується відносно нових даних, оброблених процедурою `BitBtnRebuildClick`, формується оброблене зображення на однойменній вкладці, формуються дві зменшені копії зображень (оригінального та обробленого) на вкладці «Обробка RGB» та перераховуються кількості R,G, та B складових кольору та кількість відтінків, що використовує оброблене зображення, після чого вони виводяться на боковій панелі. Також біля напису «Видалено» на бічній панелі з'являється число – кількість видалених відтінків та їх відсоткова складова відносно оригінально зображення. Перед формуванням зменшених копій зображень напис над індикатором «Йде опрацювання зображення» змінюється на «Зображення опрацьовано», від чого індикатор переходить в 0. В кінці процедури викликається процедура `DiagramNewRG` для виводу нової діаграми на 4 вкладці.



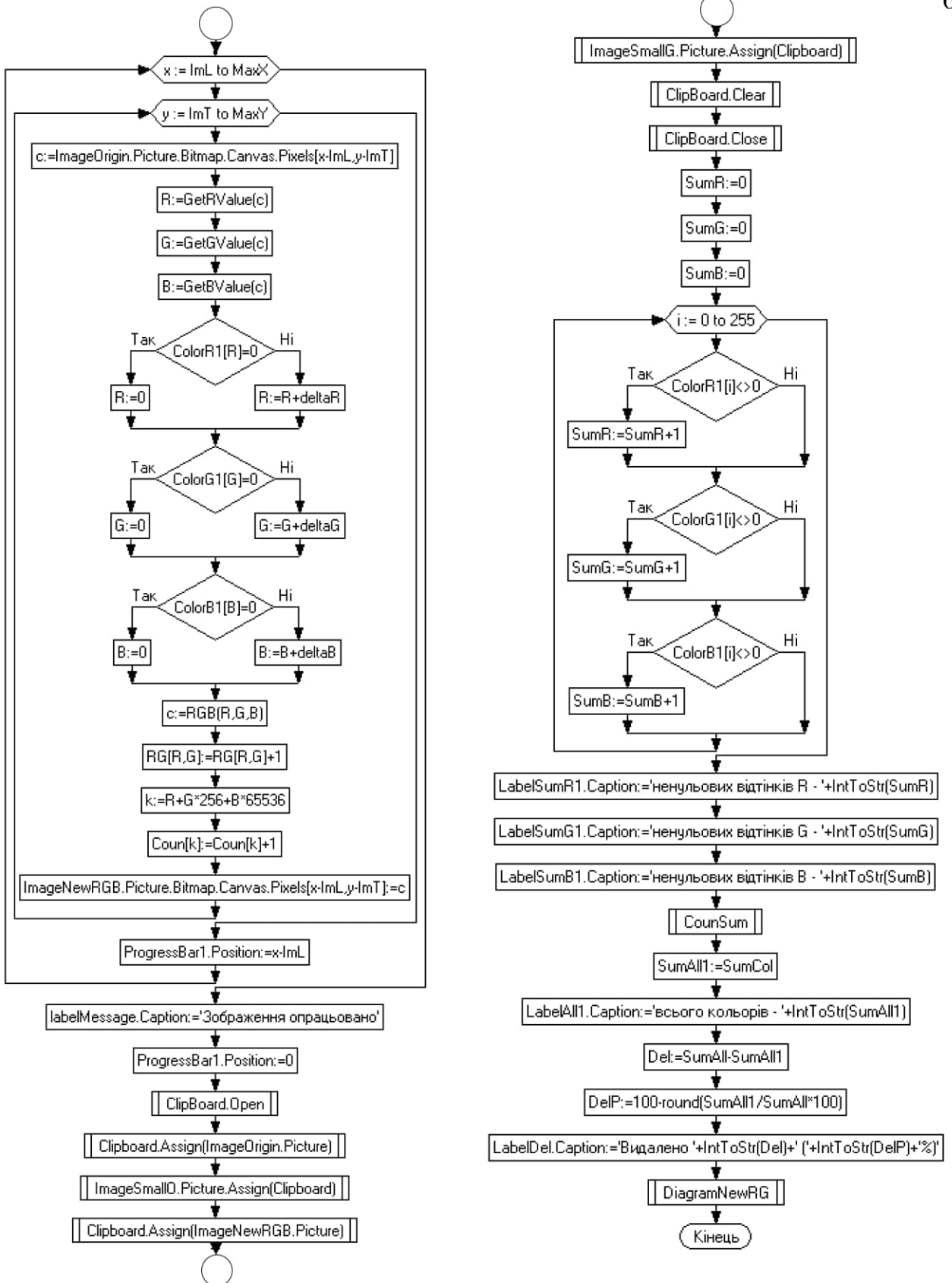


Рисунок 2.18 – Блок-схема процедури RebuildOriginRGB

### 2.7.9 Алгоритми роботи процедур ImageNewRGBMouseDown та Button1Click

ImageNewRGBMouseDown – процедура, що відповідає за функцію отримання R, G, B складових точки зображення. Вона працює на 2 вкладці програми «Оброблене зображення», де формується оброблене зображення. Після кліку на довільній точці в цьому зображенні біля курсору з'являється коментар, в якому виводиться кількість R, G, B складових піксела (Рис. 2.19).



Рисунок 2.19 – Блок-схема процедури ImageNewRGBMouseDown

Button1Click - збереження обробленого зображення (Рис. 2.20). При натисканні на кнопку «Зберегти» користувач має змогу зберегти оброблене зображення у форматі BMP в довільну теку, ним вказану. Після збереження зображення користувач може і далі продовжити роботу з програмою та зберігати інші версії зображення, або відкрити і почати обробку нового зображення.

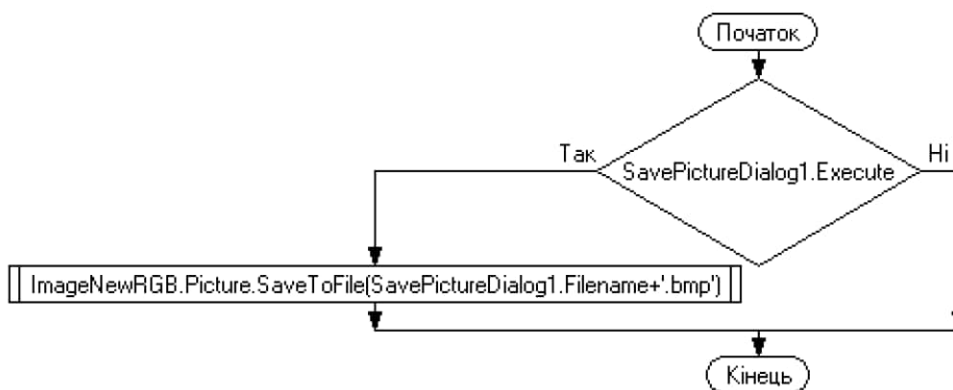


Рисунок 2.20 – Блок-схема процедури ImageNewRGBMouseDown

В наступному розділі розглянута детальніше робота програми та приклад її використання для зображення мікроструктури металевого сплаву.

### 3. ПРАКТИЧНА ЧАСТИНА

#### 3.1 Опис структури програмного додатку

Програма управління глибиною кольору растрових зображень дає користувачеві змогу:

- побачити гістограми R, G та B складових кольору зображення;
- видалити деякі відсотки відтінків або взагалі обнулити канал;
- змістити гістограми складових кольору (перекодувати зображення);
- побачити кількість відтінків загалом, що застосовуються в зображенні;
- побачити кількість ненульових відтінків компонент R, G, B, що використовуються зображенням;
- взнати складові кольору обраного пікселя обробленого зображення;
- побачити наочно кількість відповідних відтінків зображення мікроструктури сплавів на діаграмі RG (перетин червоної та зеленої компонент);
- зменшити кількість кольорів у зображенні;
- зберегти зображення у форматі BMP.

Для реалізації цих можливостей в програмному додатку є відповідні панелі та вкладки. Другі для зручності розділені по функціоналу роботи з ними.

Структура програмного додатку складається з наступних частин:

- бічна панель, яка розбита на окремі групи управляючих елементів: відкриття зображення, опрацювання зображення, результати обробки, збереження зображення;
- вкладка «Оригінальне зображення», на якій можна роздивитись завантажене зображення;

- вкладка «Оброблене зображення», на якій можна роздивитись оброблене зображення та при виборі довільної точки зображення дізнатись R, G та B складові його відтінку;
- вкладка «Обробка RGB» дає можливість користувачеві побачити гістограми червоної, зеленої та синьої складових зображення, видалити деякі відсотки відтінків (обрізати фігуру) або взагалі обнулити канал, видалити 255-й відтінок, що часто не впливає на якість зображення, зсунути фігуру гістограми вліво чи вправо та побачити виконані дії на обробленому зображенні після натискання кнопки «ОК» на боковій панелі;
- вкладка «RG Діаграми» дає можливість побачити наочно кількість відповідних відтінків зображення мікроструктури сплавів на діаграмах RG (перетин червоної та зеленої компонент) для оригінального та обробленого зображення.

## 3.2 Опис інтерфейсу та функціоналу програмного додатку

### 3.2.1 Бічна панель

Під час запуску програми користувач може бачити тільки бокову панель, адже інші вкладки сховані в цей час, щоб користувач зрозумів, з чого починати роботу з додатком (Рис. 3.1).



Рисунок 3.1 – Запуск програми

Бічна панель (рис.3.2) розбита на окремі групи управляючих елементів: «Відкрити зображення», «Опрацювання зображення», «Результати обробки», «Зберегти зображення».

The image shows a vertical sidebar with four distinct sections, each with a title bar and a light gray background. The sections are:

- Відкрити зображення:** Contains a 'Відкрити' button, a text input field for 'Ім'я відкритого файлу', and a label 'Розміри зображення:'.
- Опрацювання зображення:** Contains a button with a green checkmark and the text 'ОК', and a text input field.
- Результати обробки:** Contains two text blocks: 'Оригінальне зображення:' followed by 'ненульових відтінків R -', 'ненульових відтінків G -', 'ненульових відтінків B -', and 'всього відтінків -'; and 'Оброблене зображення:' followed by the same four lines. Below these is the label 'Видалено'.
- Зберегти зображення (\*.bmp):** Contains a 'Зберегти' button.

Рисунок 3.2 – Бічна панель

Група «Відкрити зображення». На ній є кнопка «Відкрити», з якої користувач і починає роботу. При натисканні на цю кнопку відкривається діалогове вікно, де користувач має знайти і обрати металографічне зображення в форматі BMP (Рис. 3.3). Такий формат краще підходить для обробки зображення, а ніж JPEG та подібні, що погіршують якість зображення. Після відкриття фото у полі «Ім'я відкритого файлу» з'явиться відповідно назва фото з його форматом, а нижче у полі «Розміри зображення» з'являться розміри зображення в пікселях (Рис. 3.4). При відкритті наступного файлу для обробки ці поля оновляться відповідно з новими параметрами.

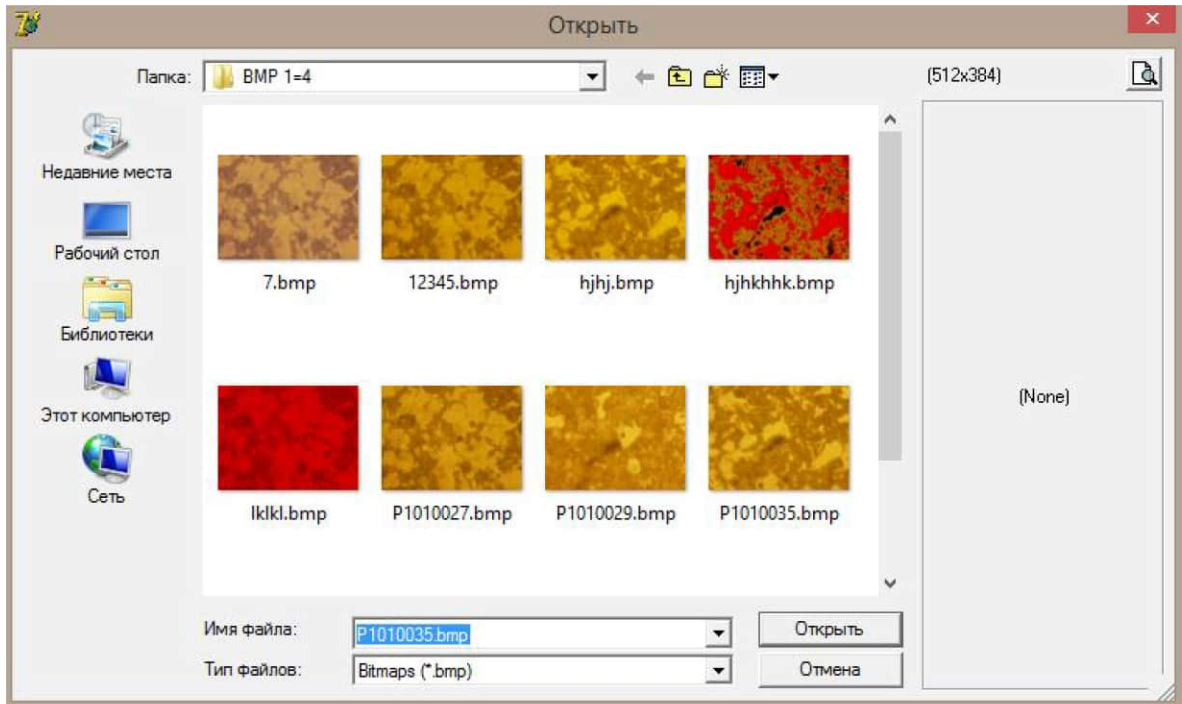


Рисунок 3.3 – Діалогове вікно

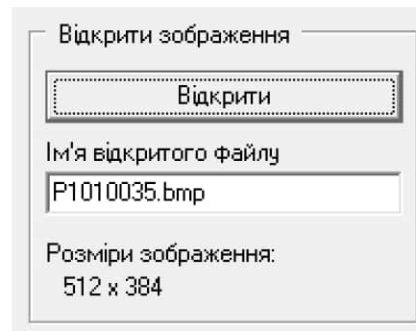


Рисунок 3.4 – Дані про відкрите зображення

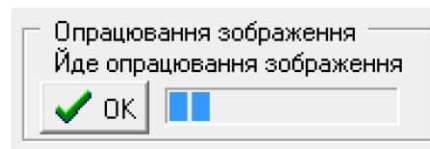


Рисунок 3.5 – Йде опрацювання зображення

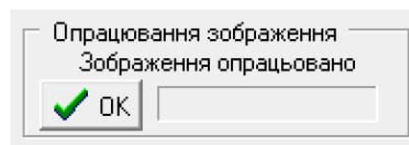


Рисунок 3.6 – Зображення опрацьовано

Група «Опрацювання зображення». Коли користувач відкриває зображення, то в цій групі вмикається індикатор, що показує, скільки залишилось до кінця опрацювання зображення (вираховування всіх масивів для даних для зображення, що знадобляться в подальшій роботі) та відповідно з'являється напис над ним «Йде опрацювання зображення» (Рис. 3.5). Якщо зображення розміру наприклад 600x500, то індикатор пройде скоро, а якщо розміри фото вимірюються в тисячах пікселів, то відповідно зображення буде довше опрацьовуватись. Після завершення опрацювання напис «Йде опрацювання зображення» змінюється на «Зображення опрацьовано», а індикатор стає в 0 (Рис. 3.6).

Кнопка «ОК» на області групи «Опрацювання зображення» знадобиться користувачеві під час обробки зображення на третій вкладці, коли він внесе дані у відповідні поля для обробки. Оскільки після натискання кнопки «ОК» йде обробка фото та оновлення масивів і гістограм, діаграм, то для наглядності також спрацьовує індикатор з тими ж написами, що і при завантаженні зображення.

Група «Результати обробки». Коли фото відкрилось та обрахувались всі дані, що його стосуються то в цій області виводяться дані кольору про оригінальне зображення, а саме кількості ненульових відтінків складових зображення R,G та B та кількість відтінків загалом, що є на вхідному зображенні (Рис. 3.7).

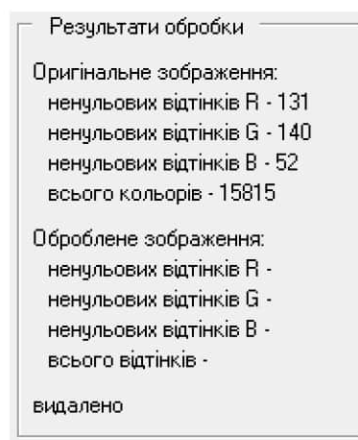


Рисунок 3.7 – Колірні дані вхідного зображення

Нижче, під результатами обробки оригінального зображення є пусті дані про оброблене зображення, вони заповняться автоматично вже після обробки зображення. Також під ними ще є напис «видалено», там після обробки зображення висвітиться кількість видалених відтінків та їх відсотковий склад відносно оригінального зображення.

Група «Зберегти зображення (\*.bmp)». Після завантаження зображення і його обробки користувач має змогу зберегти його, натиснувши на кнопку «Зберегти», в тому ж форматі BMP, в якому воно і було.

Під боковою панеллю в правому нижньому куті програми знаходиться інформація про авторство просто для особливо зацікавлених в цьому відвідувачів.

### **3.2.2 Вкладка «Оригінальне зображення»**

Після відкриття фото та його опрацювання відкривається доступ до інших вкладок програми для роботи із зображенням, а саме: «Оригінальне зображення», «Оброблене зображення», «Обробка RGB», «RG Діаграми» (Рис. 3.2.2.1). Автоматично відкривається після опрацювання зображення перша вкладка «Оригінальне зображення», на ній користувач може переглянути вхідне зображення та порівняти його в подальшому з обробленим на відповідній вкладці «Оброблене зображення».

При відкритті користувачем зображення з малими розмірами, що поміщаються в форму, зображення таким і залишається. Однак якщо зображення має великі розміри та не може поміститися в область, дану для його відображення, то відповідно для зручності огляду автоматично з'являються вертикальна та горизонтальна смуги прокрутки (Рис. 3.8). З ними користувачеві буде зручніше порівнювати змінені області з вхідними на зображенні.

Після відкриття нового фото мікроструктури металевих сплавів на цій вкладці з'явиться вже нове зображення, всі вкладки при цьому залишаються видимими.

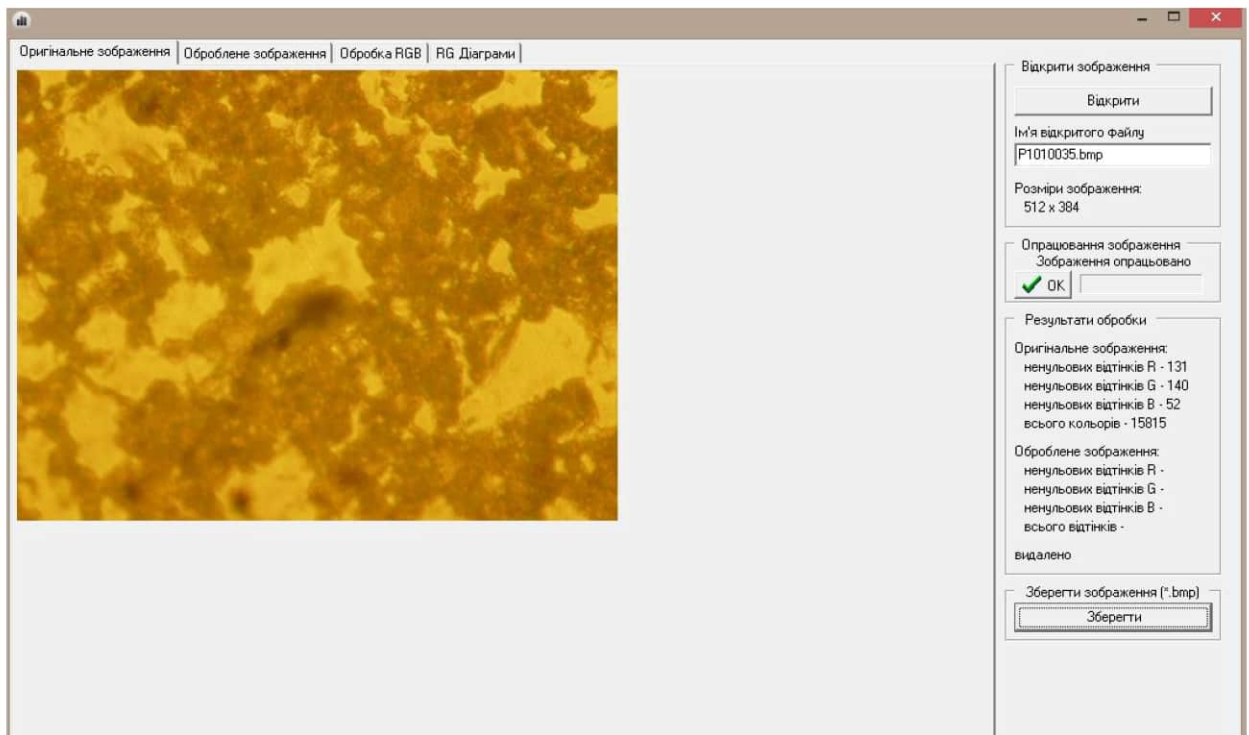


Рисунок 3.8 – Відкриття доступу вкладок для роботи із зображенням

### 3.2.3 Вкладка «Оброблене зображення»

Після обробки користувачем зображення на третій вкладці «Обробка RGB» та натискання кнопки «ОК» автоматично зображення буде перемальовуватись на тій же вкладці в зменшеному розмірі, а на цій в розмірах оригінального зображення (Рис. 3.9). Це зроблено для зручності порівняння вхідного зображення та його варіанту після обробки, щоб детально переглянути змінені області в зображенні.

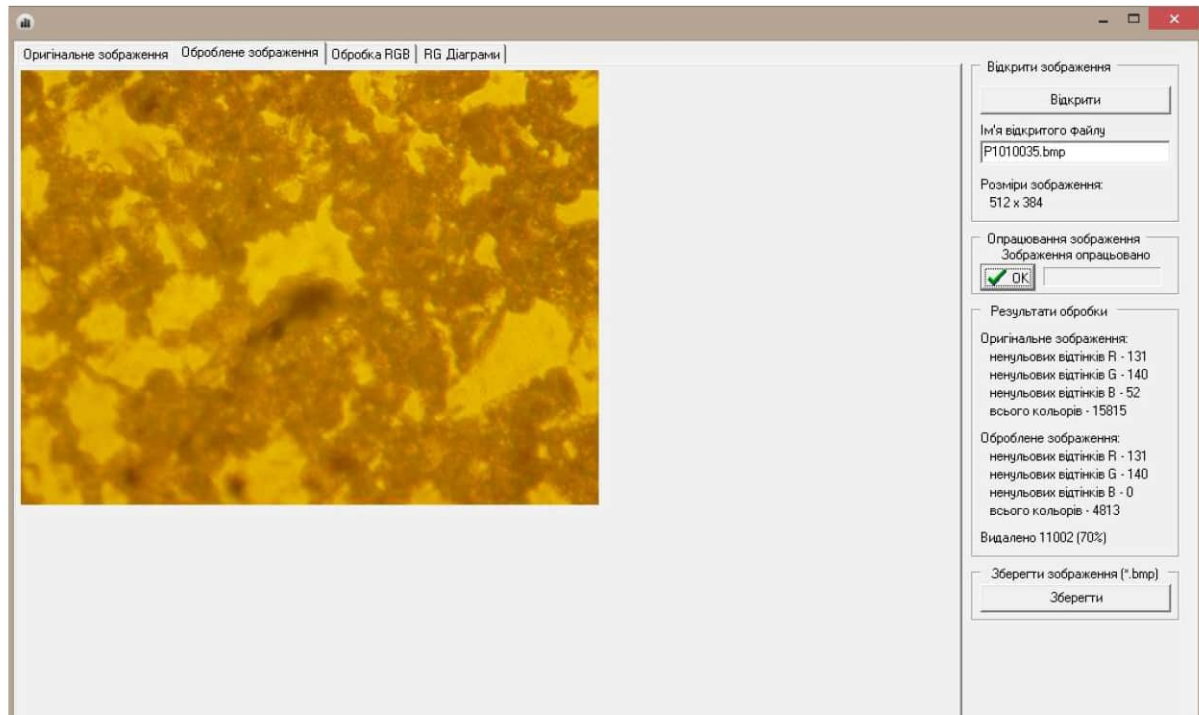


Рисунок 3.9 – Вкладка «Оброблене зображення» без обробки зображення

Під час обробки зображення, після отримання нових колірних даних про відтінки та після довільної операції із зображенням користувач програмного додатку може засумніватись у зміні колірних даних зображення або просто поцікавитись, чому на гістограмах фотографій мікроструктури металів такі фігури, які показуються у відповідних областях гістограм. Для вирішення цих потреб розроблена підказка.

При виборі довільної точки зображення на вкладці «Оброблене зображення» висвічується підказка, в якій вказані R, G та B складові її (точки, пікселя) відтінку. З допомогою цієї підказки користувач може зрозуміти, що два екстремуми на гістограмах R та G складових показують, що зображення складається зі світлих плям на темному фоні, контраст чого і є – ці два екстремуми. Також видаливши (обнуливши) повністю один з каналів, наприклад B, користувач може наглядно побачити результат – всі відтінки на обробленому зображенні міститимуть додатні тільки значення R та G, а складова B буде обнулена в будь-якій точці зображення (Рис. 3.10)

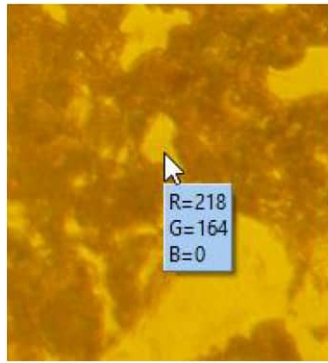


Рисунок 3.10 – Колірні дані обраної точки зображення

### 3.2.4 Вкладка «Обробка RGB»

Вкладка «Обробка RGB» (Рис. 3.11) - головна вкладка цієї програми. На ній можна побачити 6 областей для гістограм, 3 верхні з яких отримані після обробки вхідного зображення, а 3 нижні будуть отримані після зміни колірних даних зображення.

На 3-х верхній гістограмах можна побачити, які відтінки компонент R,G та B використовуються та наскільки їх багато. Чим вище лінія відтінку – тим більше цього відтінку компоненти в зображенні.

Під кожною гістограмою червоної, зеленої та синьої компонент є поля «Обрізати», «Зсунути» та прапорець «і 255-й».

В поле «Обрізати» можна ввести число від 1 до 100 – це відсотки обрізання фігури на гістограмі. Де 1% – відтінки, що мало використовуються, їх

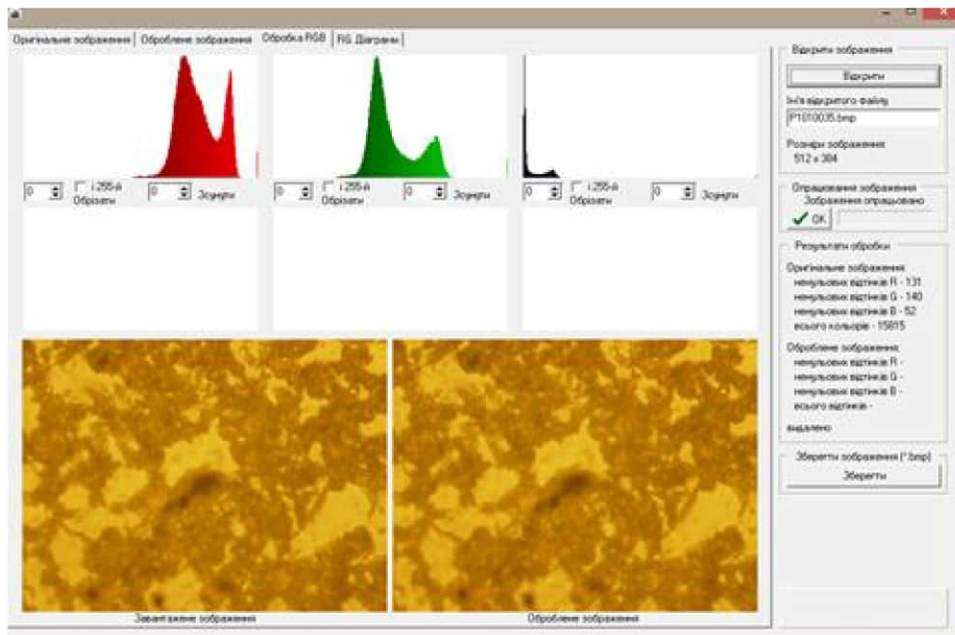


Рисунок 3.11 – Вкладка «Обработка RGB»

вертикальна лінія, що позначає кількість пікселів у відсотках, що мають такий відтінок компоненти, найменша. Відповідно лінії високі – ті відтінки, що часто застосовуються в зображенні. Після обрізання фігури на певний відсоток, наприклад, обрізати 2 відсотки в компоненті R, піксели, що мали рідковживані відтінки (зазвичай вони знаходяться зліва та справа фігури) компоненти, тепер матимуть обнулену компоненту. Тобто обрізання фігури – онулення складової відтінку (Рис. 3.12).

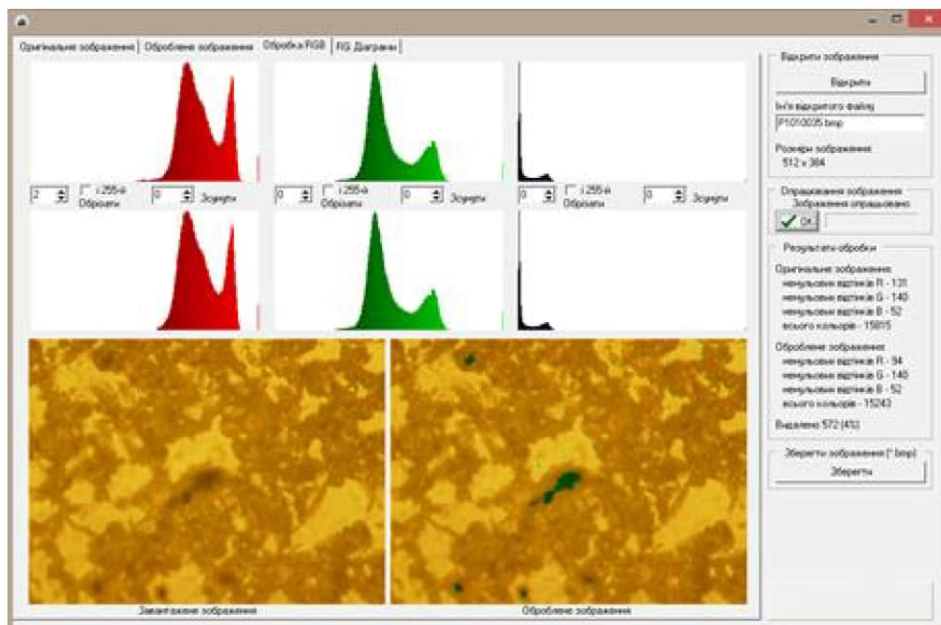


Рисунок 3.12 – Результат онулення 2% компоненти R

Після онулення компоненти на 2% фігура обрізалась з боків, що видно на нижніх гістограмах. На боковій панелі з'явилися нові колірні дані про зображення, а саме варто зауважити, що кількість відтінків червоної складової зображення зменшилась із 131 до 94. Навіть якщо обрізати 1% (Рис. 3.13), то кількість відтінків зменшиться, бо на гістограмі є ще ті відтінки, які так рідко застосовуються, що на гістограмі їх відсоток практично дорівнює 0.

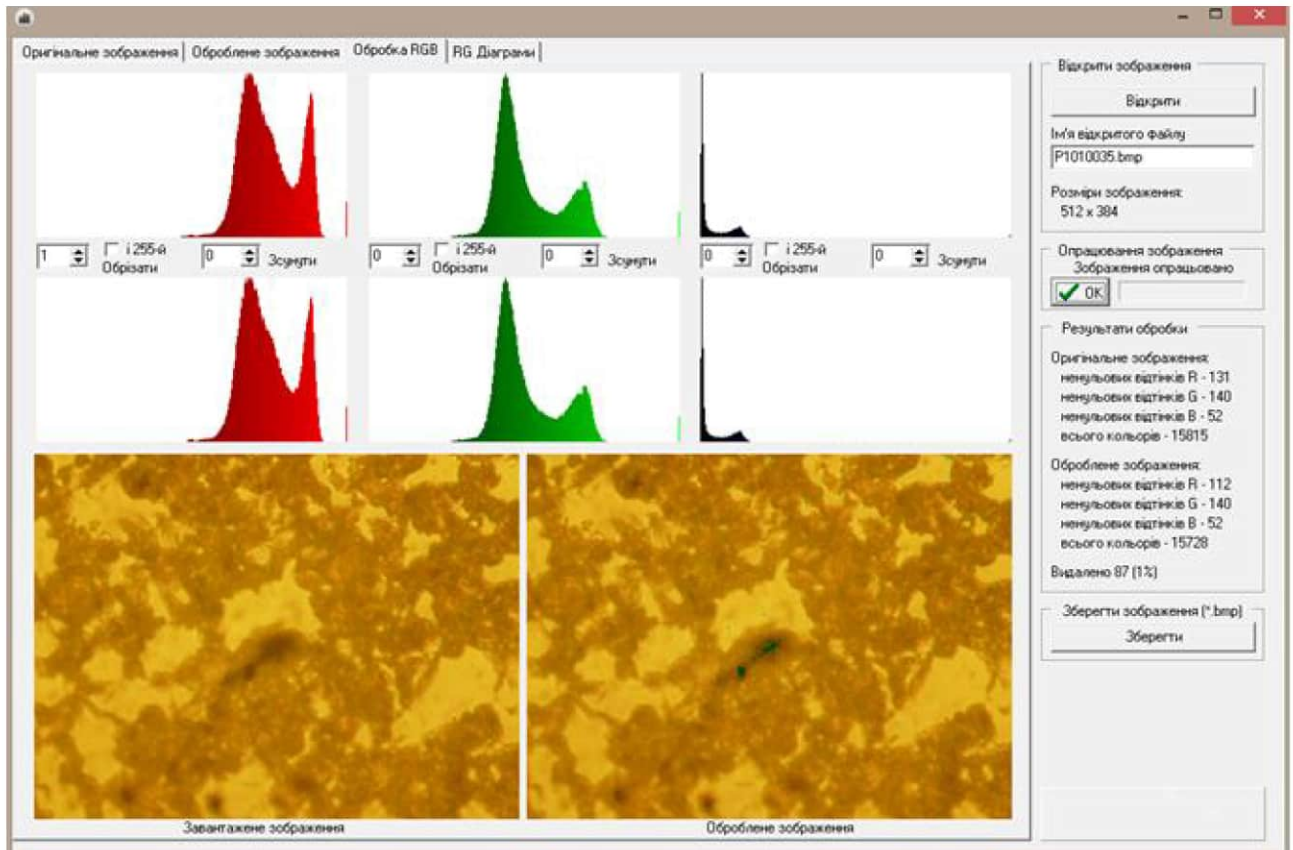


Рисунок 3.13 – Результат онулення 1% компоненти R

Такі відтінки після обрізання фігури онулюються та відповідно кількість кольорів, що є в зображенні стає меншою. На зображенні «Оброблене зображення» видно, що ці 2% відтінків знаходились на найтемніших місцях, яких мало на фото. А присутність в цих місцях зеленого кольору можна пояснити тим, що дані фото мікроструктури металевих сплавів, що застосовуються для тестування програми мають жовті кольори. Жовтий –

поєднання зеленого та червоного. Тому якщо видалити одну з цих складових в пікселі, то піксель замалюється його комплементарним кольором.

Разом з онуленням певного відсотку відтінків компоненти, можна обнулити і всю компоненту (канал кольору, Рис 3.14). Як і описувалось вище – якщо видалити в цих фото R або G складову кольору, то зображення замалюється в його комплементарний колір, адже ці 2 кольори важливі для даних фото. Однак якщо видалити всю сню складову зображення, а як видно по тону гістограми, це практично чорний колір з малим спектром відтінків, то на зображення це не так сильно вплине. Варто зауважити, що синій – комплементарний колір до жовтого, тому його відсутність тільки підвищить яскравість жовтого кольору зображення, при цьому кількість кольорів (глибина кольору – максимальна кількість кольорів, що використовуються в зображенні), що є в зображенні значно зменшиться, а якість фото залишиться такою ж, як і була до цього. Ця операція сприятиме зменшенню кількості фарби, що знадобиться для друку зображення, адже тепер синьої фарби для друку зображення практично не знадобиться. Результат онулення можна буде також перевірити на вкладці «Оброблене зображення», клацнувши на бажану точку зображення і перевіривши RGB складові її відтінку.

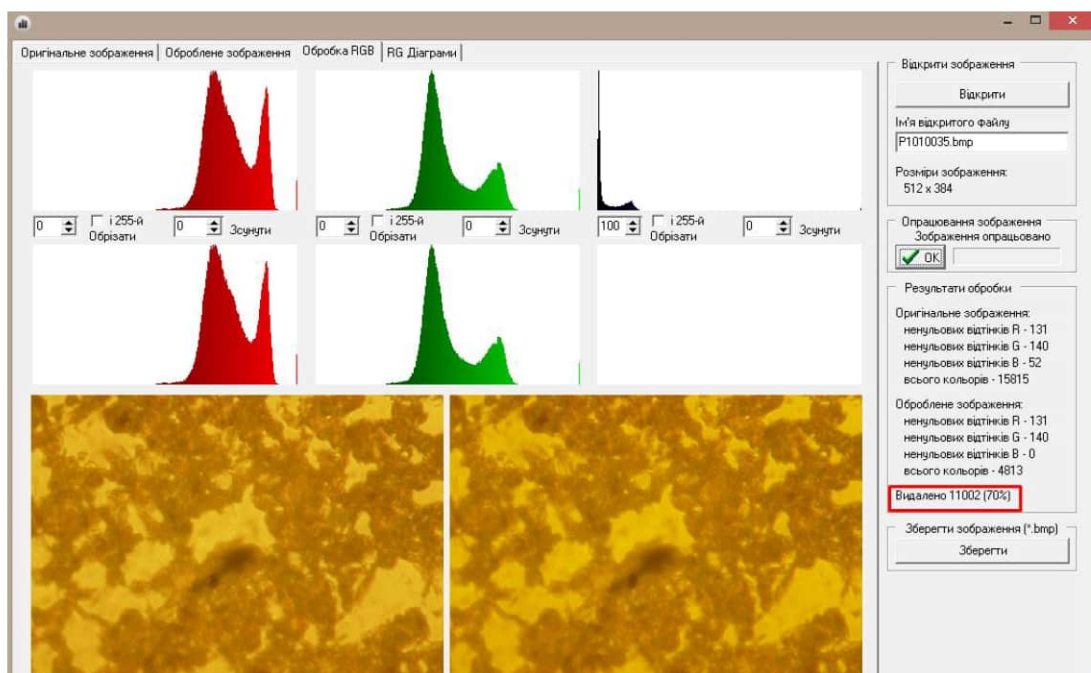


Рисунок 3.14 – Результат онулення каналу В

Поруч з полем для видалення відсотку компоненти є прапорець «і 255-й». Цей відтінок часто не впливає на якість зображення структури металографічних сплавів, являючи собою поодинокі точки на зображенні. Тому їх можна з легкістю видалити, абсолютно не змінивши візуально зображення. Для видалення їх потрібно поставити прапорці у відповідних полях та клацнути на «ОК» (Рис. 3.15).

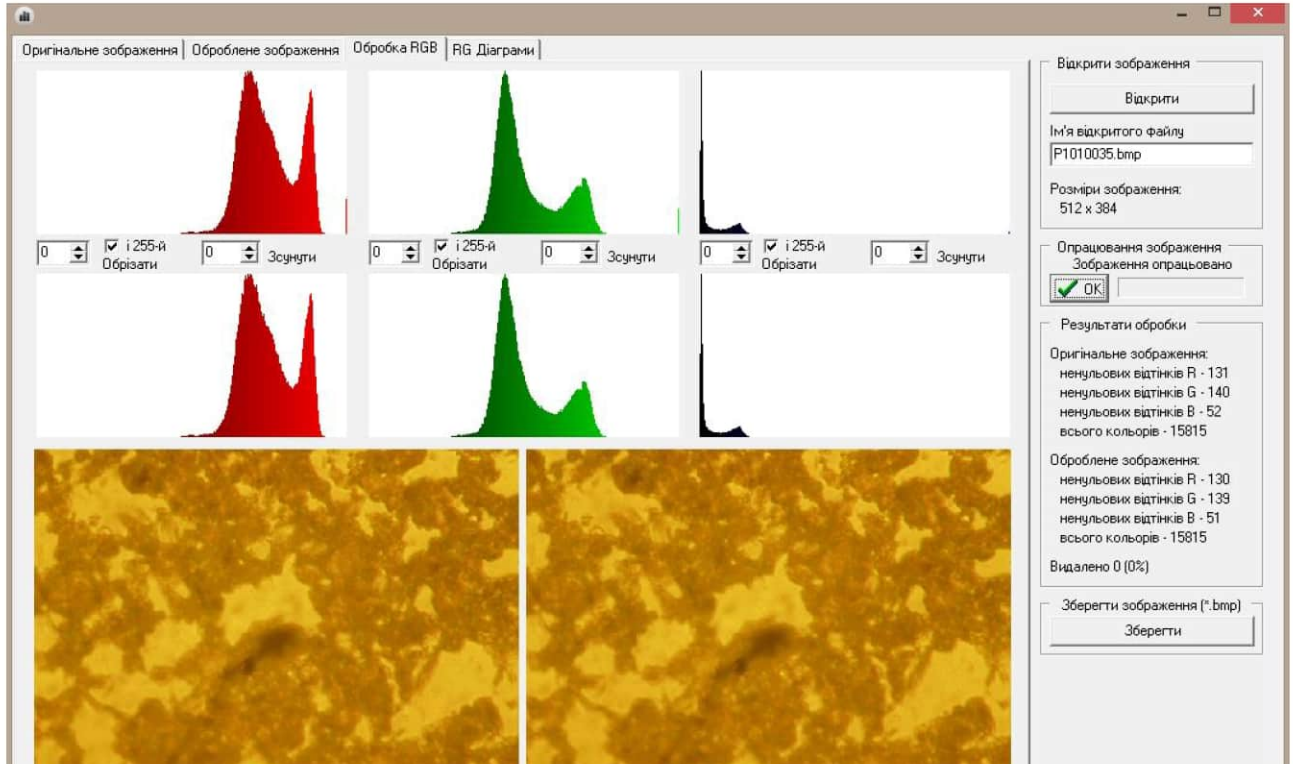


Рисунок 3.15 –Результат онулення 255-х відтінків

Справа від обрізання є поле «Зсунути». Дана процедура дозволяє зсунути фігуру на гістограмі на будь-яку відстань в межах 255 відтінків. Саме ця процедура дозволить зменшити глибину кольору в зображенні, якщо зсунути всі гістограми до 0. Для кращого результату бажано і попередньо обрізати гістограми хоч на 1-2 відсотки, щоб зменшити кількість кольорів в кожному каналі. Після зсуву відтінків до 0 програма автоматично перекодує зображення по новим гістограмам. За рахунок того, що область з відтінками тепер буде в першій частині кольорового спектру, а інші кольори автоматично

видаляються, то під час друку таке зображення матиме більшу лініатуру, адже кольорів стало менше аж на 93% у випадку з наступним фото (Рис. 3.16).

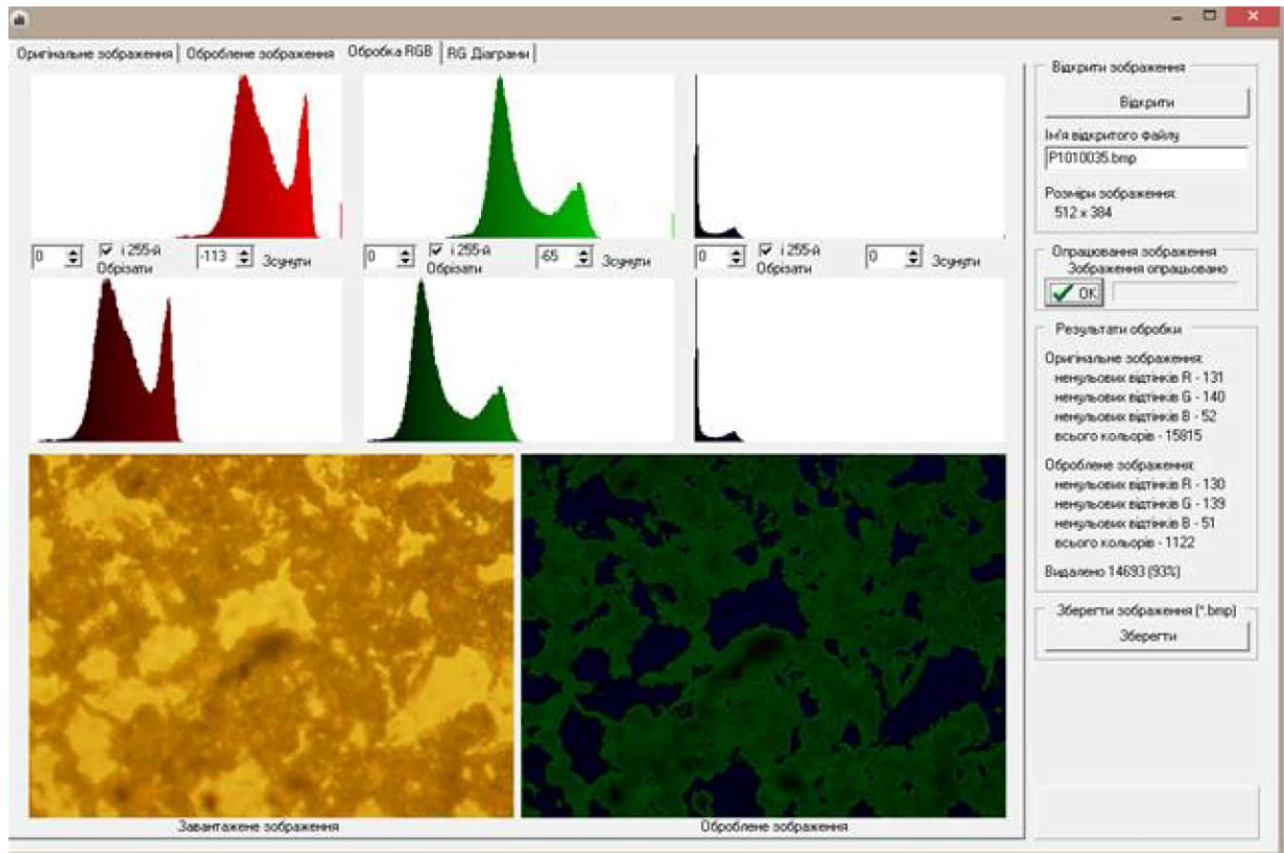


Рисунок 3.16 –Результат зсуву всіх відтінків до 0

### 3.2.5 Вкладка «RG Діаграми»

Зображення мікроструктури металевих сплавів в градаціях жовтого та оранжевого, то для відтворення цих кольорів потрібні зелений та червоний канали колірної моделі RGB. З допомогою цих діаграм стає зрозуміло, наскільки мало кольорів застосовується в зображенні з можливих на перетину каналів R та G (рис.3.17).

На вкладці «RG Діаграми» користувач може побачити дві діаграми «Для оригінального зображення» та «Для обробленого зображення», де в кожній вісь x – компонента R, вісь y – компонента G. Кожна точка на діаграмі показує кількість у відсотках пікселів із тим чи іншим кольором. Якщо відповідного відтінку немає в зображенні, то точка біла, якщо є мало - сіра, багато – чорна і подібним чином в градаціях сірого для наглядності.

Діаграма «Для оригінального зображення» формується одразу після завантаження фото з файлу, а діаграма «Для обробленого зображення» формується щоразу після натискання кнопки «ОК» на бічній панелі програми.

Нижче приклад RG Діаграми для обробленого зображення: зсув R на 10, G на 32:

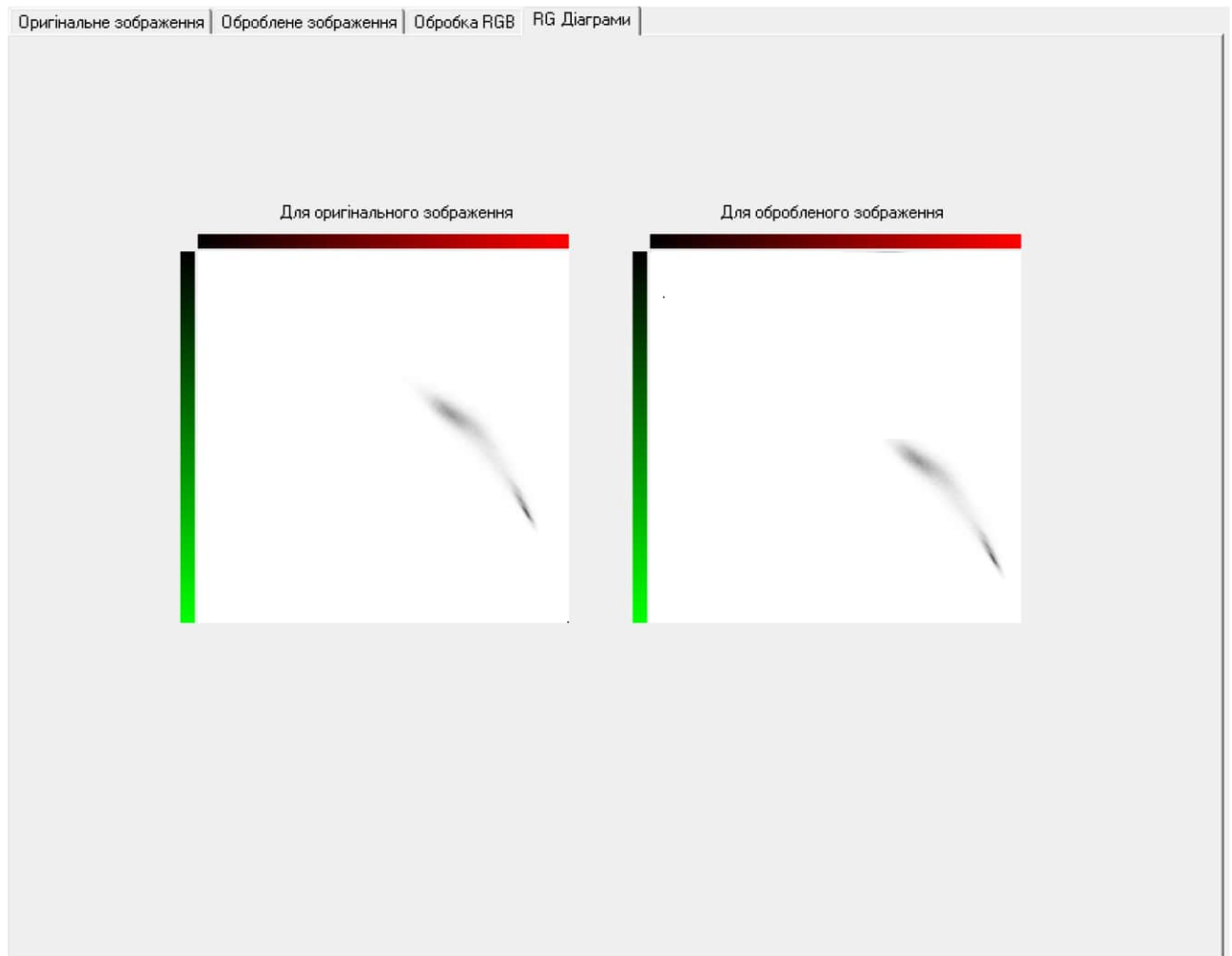


Рисунок 3.17– RG Діаграми до зсуву та після

Як видно з діаграм до та після зсуву, відтінки, що використовуються в зображенні змінилися з одних на інші, хоч і їх кількість поки зменшилась тільки на 10%, що і видно по зрізаній верхній частині фігури на діаграмі. Таким чином користувач може зсувати фігури в каналах або видаляти кольори чи канали, при цьому досліджуючи, який спектр кольорів залишається в зображенні. Очевидно, що якщо видалити, наприклад, червоний канал, то вся

фігура згрупується біля зеленої шкали діаграми. Тому далі детальніше будуть розглянуті реакції програми на різні вхідні дані.

### 3.3 Тестування програмного додатку

Вхідне зображення, отримане з оптичного мікроскопу «P1010035.bmp» (рис.3.18 та рис.3.19).

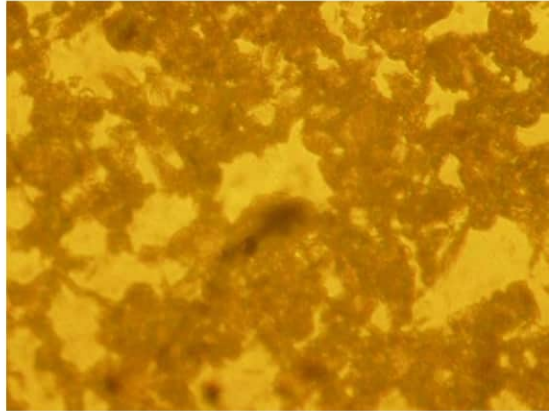


Рисунок 3.18 – Вхідне зображення P1010035.bmp

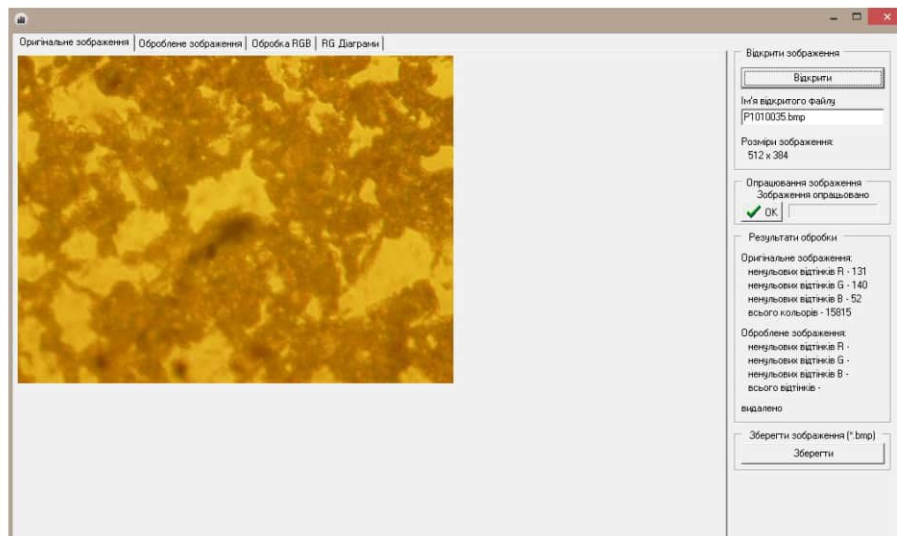


Рисунок 3.19 – Відкриття зображення програмою

Дані для обробки:

- Обрізання R на 3, G на 4, B на 0
- 255-й відтінок не видаляти для всіх каналів
- Зсув R на 5, G на 23, B на 0

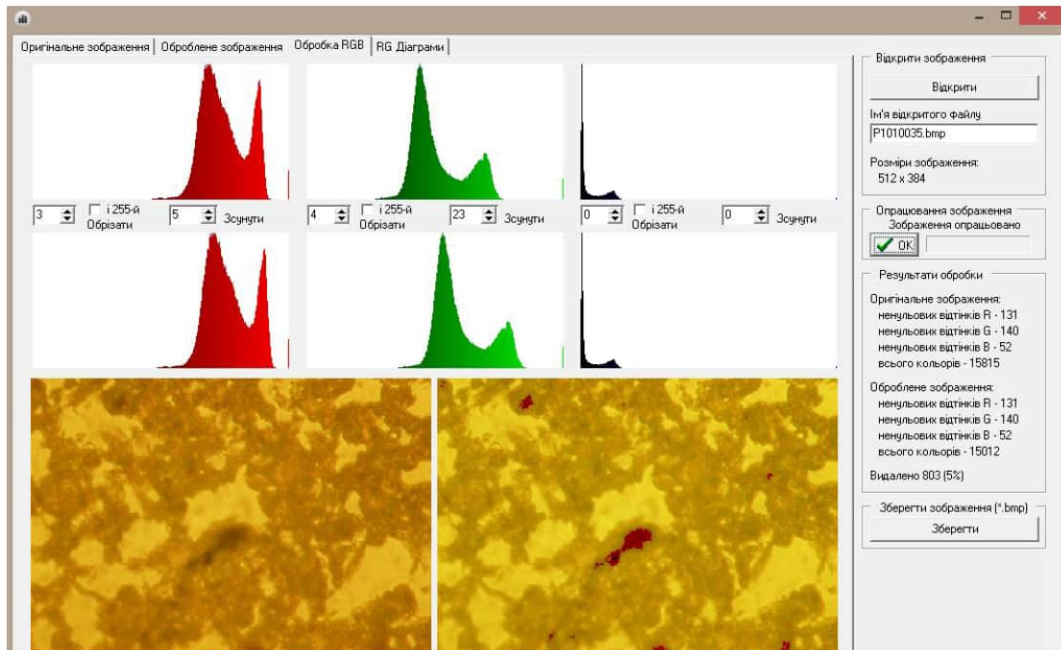


Рисунок 3.20 – Результат обробки зображення

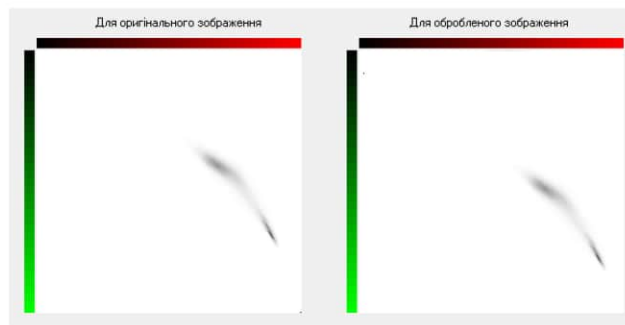


Рисунок 3.21 – RG діаграми до та після обробки зображення

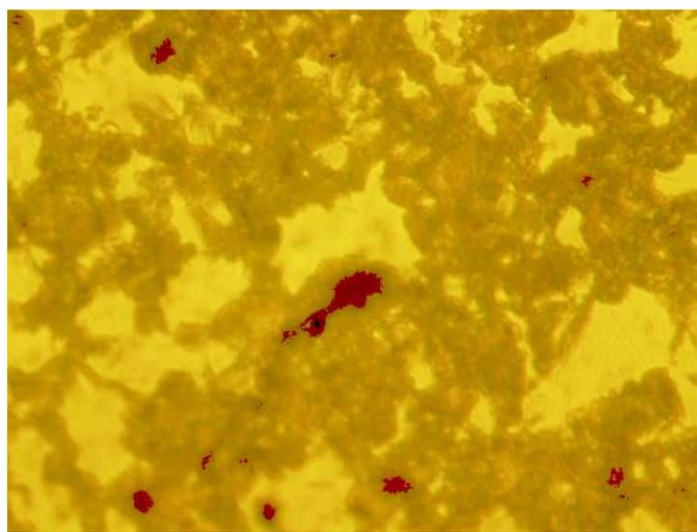


Рисунок 3.22 – Збережене оброблене зображення

У випадку останнього тестування зображення має найменші межі відтінків каналів R, G та B, тому під час друку матиме найвище значення лініатури серед трьох тестувань.

## ВИСНОВКИ

Під час кваліфікаційної роботи була розроблена аналітична складова дипломної роботи, досліджено вплив глибини кольору на лініатуру, розроблений метод підвищення якості друку, на основі врахування властивостей металографічних растрових зображень.

Було розроблено:

- структуру програмного додатку;
- алгоритм роботи користувача з програмним додатком;
- алгоритм роботи окремих модулів програми;
- метод видалення кольорів із зображення для підвищення лініатури;

Була створена програма управління лініатурою друку зображення на базі зміни глибини кольору растрових зображень. Дана програма дає можливість збільшити лініатуру і відповідно якість друку за рахунок зменшення кількості кольорів у зображенні мікроструктури металевих сплавів. Програма дає змогу також переглядати R, G, B складові кожної пікселу зображення, аналізувати гістограми, видаляти значення каналу кольору, обрізати та зміщувати гістограми R, G, B каналів зображення, переглядати кількість ненульових відтінків загалом та по каналам кольорів, переглядати на RG діаграмі наявність відтінків у зображенні. Внаслідок зробленої роботи вдалося досягнути мети – збільшити лініатуру друку для кращого відображення мікроструктури сплавів.

Отже, завдання кваліфікаційної роботи виконано і результат його виконання може використовуватись в якості інструмента дослідження колірних характеристик мікроструктури металевих сплавів.

**СПИСОК ПОСИЛАНЬ**

1. Потапчук, С. П., Булах, В. П. Основи теорії кольору та колірних просторів. — Київ: Видавництво НАУ, 2015. — 120 с.
2. Пресс, В. Т., Фленнери, Б. П., Тьюколскі, С. А., Веттерлінг, В. Т. Чисельні рецепти. Мистецтво наукових обчислень. 3-е вид. — Кембридж: Cambridge University Press, 2007. — 1230 с.
3. Хейт, Дж. Матеріалознавство. 8-е вид. — М.: Техносфера, 2008. — 800 с.
4. Макконнелл, С. Досконалий код. — Київ: Наш Формат, 2017. — 992 с.
5. ASTM E3-11. Standard Guide for Preparation of Metallographic Specimens. — ASTM International, 2011. — 21 p.
6. Image Processing Tutorials. GeeksforGeeks. — Режим доступу: <https://www.geeksforgeeks.org/tag/image-processing/> (Дата звернення: 13.06.2025).
7. Color quantization. Wikipedia, The Free Encyclopedia. — Режим доступу: [https://en.wikipedia.org/wiki/Color\\_quantization](https://en.wikipedia.org/wiki/Color_quantization) (Дата звернення: 13.06.2025).
8. Histogram Equalization. OpenCV-Python Tutorials. — Режим доступу: [https://docs.opencv.org/4.x/d4/d1b/tutorial\\_histogram\\_equalization.html](https://docs.opencv.org/4.x/d4/d1b/tutorial_histogram_equalization.html) (Дата звернення: 13.06.2025).
9. PyPI. The Python Package Index. — Режим доступу: <https://pypi.org/> (Дата звернення: 13.06.2025).
10. Halftone screens. Adobe Help Center. — Режим доступу: <https://helpx.adobe.com/illustrator/using/printing-documents.html> (Дата звернення: 13.06.2025).
11. Learn Python GUI programming with Tkinter. freeCodeCamp.org. — Режим доступу: <https://www.freecodecamp.org/news/learn-python-gui-programming-with-tkinter/> (Дата звернення: 13.06.2025).
12. VanderPlas, J. T. Python Data Science Handbook. — O'Reilly Media, 2016. — 530 p.

13. The Pillow Handbook. Official Pillow (PIL Fork) Documentation. — Режим доступу: <https://pillow.readthedocs.io/en/stable/> (Дата звернення: 13.06.2025).

14. Tkinter — Python interface to Tcl/Tk. Official Python Documentation. — Режим доступу: <https://docs.python.org/3/library/tkinter.html> (Дата звернення: 13.06.2025).

15. Sonka, M., Hlavac, V., Boyle, R. Image Processing, Analysis, and Machine Vision. 4th ed. — Cengage Learning, 2014. — 900 p.

16. OpenCV-Python Tutorials. Official OpenCV Documentation. — Режим доступу: <https://docs.opencv.org/4.x/> (Дата звернення: 13.06.2025)

## ДОДАТОК А. Фрагмент програмної реалізації

```

import tkinter as tk
from tkinter import filedialog, ttk
from PIL import Image, ImageTk
import os

# Глобальні змінні, що імітують змінні Delphi
# ColorR, ColorG, ColorB - масиви для 256 відтінків кольорів оригінального зображення
# ColorR1, ColorG1, ColorB1 - їх копії для змін (оброблених гістограм)
color_r = [0] * 256
color_g = [0] * 256
color_b = [0] * 256
color_r1 = [0] * 256
color_g1 = [0] * 256
color_b1 = [0] * 256

max_r, max_g, max_b, max_rg = 0, 0, 0, 0 # Макс. кількість пікселів для кожного кольору
kr, kg, kb, krg = 0.0, 0.0, 0.0, 0.0 # Коефіцієнти пропорційності для діаграм
num_max_r, num_max_g, num_max_b = 0, 0, 0 # Відтінок, що відповідає максимуму
pixels_count = 0 # Загальна кількість пікселів
x, y = 0, 0 # Координати пікселів
im_l, im_w, im_t, im_h, max_x, max_y = 0, 0, 0, 0, 0, 0
current_color = (0, 0, 0) # Колір пікселя в RGB
r_val, g_val, b_val = 0, 0, 0 # Складові RGB кольору пікселя
sum_r, sum_g, sum_b = 0, 0, 0 # Кількість ненульових відтінків
sum_all, sum_all1 = 0, 0 # Загальна кількість відтінків на зображенні
coun = {} # Словник для підрахунку унікальних кольорів (аналог `Coun` масиву)
sum_col = 0 # Сума унікальних кольорів
left_r, left_g, left_b, right_r, right_g, right_b = 0, 0, 0, 0, 0, 0 # Нулі зліва/справа на гістограмах
delta_r, delta_g, delta_b = 0, 0, 0 # Зміщення відтінків
rg_data = [[0 for _ in range(256)] for _ in range(256)] # Для RG діаграми
gray_rg = 0

# --- Допоміжні функції ---

def rgb_to_hex(r, g, b):
    """Перетворює значення RGB в шістнадцятковий формат для Tkinter."""
    return f'#{r:02x}{g:02x}{b:02x}'

def get_r_value(color_tuple):
    """Отримує значення червоного компонента з RGB кортежу."""
    return color_tuple[0]

def get_g_value(color_tuple):
    """Отримує значення зеленого компонента з RGB кортежу."""
    return color_tuple[1]

def get_b_value(color_tuple):
    """Отримує значення синього компонента з RGB кортежу."""
    return color_tuple[2]

def coun_null():
    """Занулює словник для підрахунку кольорів."""
    global coun
    coun = {}

def coun_sum():
    """Обчислює кількість ненульових кольорів."""
    global sum_col

```

```

sum_col = len(coun)

def left_right(app_instance):
    """Обчислення кількості нулів на гістограмі та встановлення діапазонів для SpinEdit."""
    global left_r, left_g, left_b, right_r, right_g, right_b

    i = 0
    while i < 255 and color_r1[i] == 0:
        i += 1
    left_r = i

    i = 0
    while i < 255 and color_g1[i] == 0:
        i += 1
    left_g = i

    i = 0
    while i < 255 and color_b1[i] == 0:
        i += 1
    left_b = i

    i = 255
    while i > 0 and color_r1[i] == 0:
        i -= 1
    right_r = i

    i = 255
    while i > 0 and color_g1[i] == 0:
        i -= 1
    right_g = i

    i = 255
    while i > 0 and color_b1[i] == 0:
        i -= 1
    right_b = i

    # Оновлення діапазонів Spinbox (MinValue та MaxValue)
    app_instance.spin_edit_r2.config(from_=-left_r, to=255 - right_r)
    app_instance.spin_edit_g2.config(from_=-left_g, to=255 - right_g)
    app_instance.spin_edit_b2.config(from_=-left_b, to=255 - right_b)

def diagram_rg(app_instance, image_widget, rg_data_to_draw):
    """Побудова RG діаграми."""
    global max_rg, krg, gray_rg

    # Створюємо нове зображення для діаграми
    diagram_image = Image.new('RGB', (256, 256), color='white')

    max_rg = 0
    for x in range(256):
        for y in range(256):
            if rg_data_to_draw[x][y] > max_rg:
                max_rg = rg_data_to_draw[x][y]

    if max_rg == 0:
        krg = 0
    else:
        krg = 255 / max_rg

    for x in range(256):
        for y in range(256):
            gray_rg = 255 - round(rg_data_to_draw[x][y] * krg)

```

```

    diagram_image.putpixel((x, y), (gray_rg, gray_rg, gray_rg))

    app_instance.rg_photo_image = ImageTk.PhotoImage(diagram_image)
    image_widget.config(image=app_instance.rg_photo_image)
    image_widget.image = app_instance.rg_photo_image # Зберегти посилання, щоб уникнути збирання
    сміття

def diagram_new_rg(app_instance, image_widget, rg_data_to_draw):
    """Побудова RG діаграми для обробленого зображення."""
    global max_rg, krg, gray_rg

    diagram_image = Image.new('RGB', (256, 256), color='white')

    max_rg = 0
    for x in range(256):
        for y in range(256):
            if rg_data_to_draw[x][y] > max_rg:
                max_rg = rg_data_to_draw[x][y]

    if max_rg == 0:
        krg = 0
    else:
        krg = 255 / max_rg

    for x in range(256):
        for y in range(256):
            gray_rg = 255 - round(rg_data_to_draw[x][y] * krg)
            diagram_image.putpixel((x, y), (gray_rg, gray_rg, gray_rg))

    app_instance.new_rg_photo_image = ImageTk.PhotoImage(diagram_image)
    image_widget.config(image=app_instance.new_rg_photo_image)
    image_widget.image = app_instance.new_rg_photo_image # Зберегти посилання

def histogram(app_instance, image_r_widget, image_g_widget, image_b_widget, colors_r, colors_g,
              colors_b):
    """Побудова RGB гістограм."""
    global kr, kg, kb, max_r, max_g, max_b

    hist_height = 200 # Висота гістограми (аналог ImageGistR.Height)

    # Обчислюємо коефіцієнти пропорційності
    max_r = max(colors_r) if colors_r else 1 # Запобігаємо діленню на нуль
    max_g = max(colors_g) if colors_g else 1
    max_b = max(colors_b) if colors_b else 1

    kr = hist_height / max_r
    kg = hist_height / max_g
    kb = hist_height / max_b

    # Гістограма R
    hist_r_img = Image.new('RGB', (256, hist_height), color='white')
    for i in range(256):
        nr = round(kr * colors_r[i])
        for j in range(min(nr, hist_height)): # Обмеження висоти, щоб не виходити за межі
            hist_r_img.putpixel((i, hist_height - 1 - j), (i, 0, 0)) # -1 бо координати з 0

    # Гістограма G
    hist_g_img = Image.new('RGB', (256, hist_height), color='white')
    for i in range(256):
        ng = round(kg * colors_g[i])
        for j in range(min(ng, hist_height)):

```

```

hist_g_img.putpixel((i, hist_height - 1 - j), (0, i, 0))

# Гістограма B
hist_b_img = Image.new('RGB', (256, hist_height), color='white')
for i in range(256):
    nb = round(kb * colors_b[i])
    for j in range(min(nb, hist_height)):
        hist_b_img.putpixel((i, hist_height - 1 - j), (0, 0, i))

app_instance.gist_r_photo = ImageTk.PhotoImage(hist_r_img)
image_r_widget.config(image=app_instance.gist_r_photo)
image_r_widget.image = app_instance.gist_r_photo

app_instance.gist_g_photo = ImageTk.PhotoImage(hist_g_img)
image_g_widget.config(image=app_instance.gist_g_photo)
image_g_widget.image = app_instance.gist_g_photo

app_instance.gist_b_photo = ImageTk.PhotoImage(hist_b_img)
image_b_widget.config(image=app_instance.gist_b_photo)
image_b_widget.image = app_instance.gist_b_photo

def histogram1(app_instance, image_r_widget, image_g_widget, image_b_widget, colors_r, colors_g,
colors_b):
    """Побудова RGB гістограм для оброблених даних."""
    global kr, kg, kb, max_r, max_g, max_b

    hist_height = 200 # Висота гістограми

    # Обчислюємо коефіцієнти пропорційності
    max_r = max(colors_r) if colors_r else 1
    max_g = max(colors_g) if colors_g else 1
    max_b = max(colors_b) if colors_b else 1

    kr = hist_height / max_r
    kg = hist_height / max_g
    kb = hist_height / max_b

    # Гістограма R
    hist_r_img = Image.new('RGB', (256, hist_height), color='white')
    for i in range(256):
        nr = round(kr * colors_r[i])
        for j in range(min(nr, hist_height)):
            hist_r_img.putpixel((i, hist_height - 1 - j), (i, 0, 0))

    # Гістограма G
    hist_g_img = Image.new('RGB', (256, hist_height), color='white')
    for i in range(256):
        ng = round(kg * colors_g[i])
        for j in range(min(ng, hist_height)):
            hist_g_img.putpixel((i, hist_height - 1 - j), (0, i, 0))

    # Гістограма B
    hist_b_img = Image.new('RGB', (256, hist_height), color='white')
    for i in range(256):
        nb = round(kb * colors_b[i])
        for j in range(min(nb, hist_height)):
            hist_b_img.putpixel((i, hist_height - 1 - j), (0, 0, i))

    app_instance.gist_r1_photo = ImageTk.PhotoImage(hist_r_img)
    image_r_widget.config(image=app_instance.gist_r1_photo)
    image_r_widget.image = app_instance.gist_r1_photo

```

```

def save_button_click(self):
    """Обробник події для кнопки 'Зберегти'."""
    if self.processed_image:
        file_path = filedialog.asksaveasfilename(
            defaultextension=".png",
            filetypes=[("PNG files", "*.png"), ("JPEG files", "*.jpg"), ("All files", "*.*")]
        )
        if file_path:
            try:
                self.processed_image.save(file_path)
                self.label_message.config(text=f"Зображення збережено як: {os.path.basename(file_path)}")
            except Exception as e:
                self.label_message.config(text=f"Помилка збереження зображення: {e}")
        else:
            self.label_message.config(text="Немає зображення для збереження.")

# Delphi FormCreate method is typically handled in __init__ in Python
# This example initializes global variables and UI state directly in __init__
# and `initialize_ui_state` method.

if __name__ == "__main__":
    root = tk.Tk()
    app = ImageProcessorApp(root)
    root.mainloop()

```