

# Про Проблеми Відновлення Якості Зображень

Андрій Фельтов

кафедра інформаційних технологій  
Національний лісотехнічний університет України  
Львів, Україна

Оксана Чмир

кафедра прикладної математики і механіки  
ЛДУ безпеки життєдіяльності  
Львів, Україна  
o\_chmyr@yahoo.com

Наталія Процах

кафедра інформаційних технологій  
Національний лісотехнічний університет України  
Львів, Україна  
protsakh@ukr.net

Андрій Мельниченко

кафедра інформаційних технологій  
Національний лісотехнічний університет України  
Львів, Україна

## On Images Quality Restoration

Andrii Fel'tov

dept. of Information Technology  
National Forestry University of Ukraine  
Lviv, Ukraine

Oxana Chmyr

dept. of Applied Mathematics and Mechanics  
Lviv State University of Vital Activity Safety  
Lviv, Ukraine  
o\_chmyr@yahoo.com

Nataliia Protsakh

dept. of Information Technology  
National Forestry University of Ukraine  
Lviv, Ukraine  
protsakh@ukr.net

Andrii Mel'nychenko

dept. of Information Technology  
National Forestry University of Ukraine  
Lviv, Ukraine

**Анотація**—У роботі засобами C# з використанням методу аналізу ієрархій, розроблено та реалізовано алгоритм відновлення якості зображень, а також здійснено короткий огляд існуючих програм, які дозволяють редагувати зображення.

**Abstract**—In this work with the use of C# and the method of hierarchy analysis the algorithm of restoring and improving of image quality is developed and implemented. A brief overview of existing program applications for image manipulation is also provided.

**Ключові слова**—графічні формати; цифрова обробка зображень; покращення якості зображення

**Keywords**—grafic formats; digital image processing; image quality improvement

### I. ВСТУП

Із появою засобів отримання візуальної інформації стали актуальними задачі покращення якості зображення. Однією з таких задач є задача фільтрування шумів та

відновлення втрачених фрагментів зображень. Шум в цифровому зображенні може з'являтися внаслідок: недосконалості пристроїв формування цифрового зображення, забруднення середовища навколо фотографованої ділянки, або оптичних каналів передачі світла системи формування та захоплення зображень, підчас роботи з динамічними сценами, коли потрібно формувати вихідне зображення достатньо швидко і, відповідно, на процес формування яскравості кожного пікселя зображення починає впливати обмеженість та дискретність кількості фотонів, що потрапляють на світлочутливі комірки матриці, та ін.

У будь-якому випадку, завдання полягає в пошуку і усуненні шумових фрагментів зображення з метою збереження якості зображення, мінімального спотворення корисної початкової інформації.

### II. ПРОГРАМИ РОБОТИ З ГРАФІЧНИМИ ФАЙЛАМИ

На сьогоднішній час відомо достатньо багато програм, які дозволяють переглядати та редагувати зображення

користувачам з різним рівнем комп'ютерних навичок. Наведемо короткий опис можливостей редагування зображень найбільш відомими з них.

- Програма *Adobe Photoshop* має різні засоби для покращення якості фотографічних зображень і усунення дефектів на них. За її допомогою можна змінити насиченість зображення, посилити контрастність і різкість, відкоригувати кольори, пом'якшити тіні і відблиски, які виникають при використанні спалаху, згладити зморшки на обличчі і усунути ефект «червоних очей», приховати подряпини і сторонні предмети, що потрапили в кадр, і виконати безліч інших операцій корекції і ретушування фотознімків. Для пом'якшення різких переходів зображень програма використовує, зокрема фільтр *Gaussian Blur* (Гаусове розмиття).
- *GIMP* може використовуватися як редактор для малювання (завдяки підтримці графічних планшетів) і як інструмент для обробки фотографій (в тому числі пакетної обробки). Має можливості розширення, малювання, анімації, виділення, перетворення, пакетної обробки зображень. Зниження рівня шумів здійснюється за допомогою фільтра "Вибіркове Гаусове розмиття".
- *Picasa* – програма від компанії Google, яка дозволяє упорядковувати колекції фотографій, автоматично змінює їх розмір; надає можливості змінювати яскравість і контрастність зображення, усувати ефект червоних очей та ін..
- *Photo! Editor* – дозволяє змінювати формат фотографій, підтримує формати .jpeg, .bmp, .png, .gif, .tiff; надає можливість корекції колірного балансу, контрастності, яскравості і насиченості, видаляти світлові та кольорові шуми на фотографіях, має новітню технологію налаштування різкості фотографій (навіть для дуже розпливчастих), деформування зображень для створення карикатур, ретушування портретних зображень, баланс освітлення для створення художніх світлових ефектів і 3D рельєфу і т.д.
- *LightBox Free Image Editor* – дозволяє редагувати зображення, має функцію видалення ефекту червоних очей з технологією виявлення обличчя, накладання тіней зі збереженням реальності зображень і т.д.

### III. ПАКЕТИ В C# ДЛЯ РОБОТИ З ЗОБРАЖЕННЯМИ

До складу *.NET Framework* входить API *GDI +* для роботи з графікою. За допомогою *GDI +* можна створювати малюнки, малювати текст і управляти графічними зображеннями як об'єктами, виводити графічні зображення в формах *Windows Forms* і елементах управління. Хоча *GDI +* можна використовувати безпосередньо в веб-формах, можна виводити графічні зображення з використанням серверного веб-елемента управління *Image*.

При створенні елемента управління *Windows Forms* можна використовувати інтерфейс *GDI +* для доступу до зображення і його відновлення, для створення власних зображень незалежно від призначеного для користувача інтерфейсу додатку.

Для малювання зображення в *.NET Framework* використовують об'єкт *Graphics*. Клас *Graphics* має безліч методів для малювання і роботи з зображеннями. Наприклад, методи для малювання ліній: *DrawArc*, *DrawBezier*, *DrawEllipse*, *DrawImage*, *DrawLine*, *DrawPolygon*, *DrawRectangle* і *DrawString*; методи для заповнення фігур: *FillClosedCurve*, *FillEllipse*, *FillPath*, *FillPolygon* і *FillRectangle*; метод для очищення поверхні малювання: *Clear*; метод для створення нового об'єкта *Graphics* з зображення *FromImage* та ін.

Клас *Bitmap* інкапсулює *GDI +* точкового малюнка, який складається з даних пікселів графічного зображення, і його атрибути. А *Bitmap* - об'єкт, який використовується для роботи з зображеннями, які визначаються даними пікселів.

### IV. ФОРМАТИ ТА СТРУКТУРА ГРАФІЧНИХ ФАЙЛІВ RGB

Формати та структура графічних файлів визначають спосіб зберігання інформації у файлі (растровий або векторний), а також форму зберігання інформації, тому методи їх аналізу також є різними.

Для зберігання зображень в комп'ютерній графіці використовують декілька десятків форматів файлів. За типами графічні формати можна розділити на: *растрові*, *векторні*, *метафайлові*, *сцени*, *анімації*, *мультимедійні*, *тривимірні*.

- *Формат GIF (розширення імені файлу .GIF)* є растровим форматом, який розроблявся для мереж з низькими швидкостями передачі даних, підтримується *Web*. *GIF* здатен ефективно стискати графічні дані, використовуючи алгоритм *LZW*, який полягає в стисканні ряду однакових символів в один символ, помножений на кількість повторень. Анімаційні файли *GIF* дозволяють в одному файлі зберігати декілька зображень, які відтворюються послідовно.
- *Формат JPEG (розширення імені файлу .JPG)* є растровим; дозволяє регулювати співвідношення між мірою стискання файлу і якістю зображення, стискує зображення, зберігаючи його повну чорно-білу версію і більшу частину колірної інформації; є форматом зі втратами, що проявляється у вигляді розмитого або випадкового розподілення пікселів.
- *Формат PNG (розширення імені файлу .PNG)* є растровим, призначений для публікації зображень в Інтернеті; підтримує три типи зображень – кольорові з глибиною 8 або 24 біти і чорно-білі з градацією 256 відтінків сірого. Стискання інформації здійснюється без втрат.
- *Формат TIFF (розширення імені файлу .TIF)* є растровим і призначений для збереження

зображень високої якості та великого розміру. Забезпечує зберігання чорно-білих зображень та зображень з глибиною кольору 8, 16, 24 і 32 біт. Для зменшення розміру файлу використовується вмонтований алгоритм LZW.

- *Формат Windows Bitmap (розширення імені файлу .BMP або .DIB)* є растровим, дозволяє використовувати палітри в 2, 16, 256 кольорів або повну палітру в 16 млн. кольорів.
- *Формат PCX (розширення імені файлу .PCX)* растровий, використовується редактором Paintbrush підтримує палітри в 2, 16, 256 кольорів або повну палітру в 16 млн. кольорів.
- *Формат WMF (розширення імені файлу .WMF)* підтримує векторну і растрову графіку у середовищі Windows, використовуючи палітри в 65 тис. і 16 млн. кольорів.
- *Формат CGM (розширення імені файлу .CGM)* підтримує векторну і растрову графіку з використанням повної палітри в 16 млн. кольорів та палітри зі змінною кількістю кольорів, орієнтований на складні та високохудожні зображення, створює компактні файли.
- *Формат EPS (розширення імені файлу .EPS)* описує векторні і растрові зображення на мові PostScript, одночасно може зберігати як векторну, так і растрову графіку, шрифти, параметри калібрування обладнання, профілі кольору.
- *Формат PDF (розширення імені файлу .PDF)* є апаратно незалежним і призначений для зберігання документів, однак його можливості забезпечують ефективне представлення зображень.
- *Формат PSD (розширення імені файлу .PSD)* має потужні можливості зберігання растрової графічної інформації; дозволяє запам'ятовувати параметри пластів, каналів, міри прозорості, множини масок і підтримує 48-бітове кодування кольору, розділення кольорів і різноманітні моделі кольору.
- *Растрові файли формату BMP* пов'язані з файлами зображень, такими як фотографії, кліпарті та інші графічні елементи. Кожен файл BMP складається з заголовка файлу, інформації заголовка, визначення таблиці кольорів (за винятком BMP файлів в режимі 24 бітного кольору) і набору растрових пікселів, що створюють зображення файлу. Файли формату BMP складаються з кольорових пікселів з відтінками червоного, зеленого і синього (RGB), які в сукупності створюють образ. Оскільки окремі пікселі об'єднуються, щоб сформувати файл формату BMP, користувачі можуть вносити зміни в один піксель без зміни іншої частини зображення. Крім того, формат BMP дає нестислий файл.

## V. АЛГОРИТМ ВІДНОВЛЕННЯ ЗАШУМЛЕНИХ ЗОБРАЖЕНЬ НА ПІДСТАВІ МЕТОДУ АНАЛІЗУ ІСРАРХІЙ

Редагуючи зображення важливо, в першу чергу, виявити і усунути шум із зображення, оскільки зашумлене зображення має небажану візуальну якість, знижує видимість об'єктів. Існують два основні підходи до зниження зашумленості зображення: точкові методи фільтрування (нелінійні та лінійні фільтри), методи фільтрування, які передбачають перетворення зашумленої області (адаптивне і неадаптивне перетворення даних) [1-6].

Розглянемо один з алгоритмів відновлення зображення, згідно з яким аналізуються пікселі, що оточують пошкоджений, обирається колір пошкодженого пікселя серед середнього кольору восьми найближчих сусідів з урахуванням того, що і вони можуть бути пошкоджені. Так, для пікселя, який знаходиться біля пошкодженого, потрібно виявляти різкі перепади кольору. Також до уваги береться набір сусідів, які мають такий самий колір, що дозволяє виявити області суцільної заливки, а також відновлювати межі колірних областей.

Отже, розглядаємо пікселі, що оточують пошкоджений. Причому будемо аналізувати як найближчі пікселі, так і наступні за ним, як показано на таблиці 1.

TABLE I. АНАЛІЗОВАНА КАРТА ПІКСЕЛІВ

d1	d2	d3	d4	d5
d6	c1	c2	c3	d7
d8	c4	X	c5	d9
d10	c6	c7	c8	d11
d12	d13	d14	d15	d16

Колір пошкодженого пікселя X обираємо з восьми кольорів найближчих сусідів. Введемо три критерії, що характеризують найближчих сусідів c1, ..., c8, які дозволять вибрати тільки один з них:

K1 (виявляє різкі перепади кольорів) – відміну кольору даного пікселя від середнього кольору найближчих сусідів,

K2 (виявляє області суцільної заливки) – кількість сусідніх пікселів, що мають такий самий колір,

K3 (виявляє випадки, коли пошкоджений піксель знаходиться на межі двох колірних областей) – відміну кольору даного пікселя від пікселя, розташованого з іншого боку від пошкодженого пікселя.

Вважаємо, що всі критерії рівнозначні і мають однакові вагові коефіцієнти. Визначимо вагові коефіцієнти цих критеріїв.

Для критерію K1 будемо характеризувати кожен піксель  $c_i$  величиною  $x_i = 1 - \frac{\Delta c_i}{m}$ , де  $\Delta c_i$  – середнє значення кольору пікселів, що оточують  $c_i$ ,  $m$  – максимальне значення кольору в палітрі. Елементи матриці парних порівнянь будуть знаходитися як  $A_{ij} = \frac{x_i}{x_j}$ .

З матриці парних порівнянь знаходимо вагові коефіцієнти різних рішень в рамках критерію K1:

$$u_i = \frac{x_i}{x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8} \quad (i = 1, \dots, 8).$$

В критерії K2 розглядаємо сім пікселів, які є сусідами з даним. Кожному пікселю  $c$  поставимо у відповідність величину  $y_i$ , яка дорівнює кількості сусідів, що мають той самий колір. Елементи матриці парних порівнянь матимуть вигляд:  $B_{ij} = \frac{y_i}{y_j} \quad (i = 1, \dots, 8)$ . Вагові коефіцієнти будуть мати вигляд:

$$v_i = \frac{y_i}{y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8} \quad (i = 1, \dots, 8).$$

В критерії K3 будемо розглядати для кожного пікселя  $c_i$ , його відмінність від протилежного пікселя  $dc_i$ . Тоді кожен піксель буде характеризуватися величиною  $z_i = 1 - \frac{dc_i}{m} \quad (i = 1, \dots, 8)$ . Вагові коефіцієнти:

$$w_i = \frac{z_i}{z_1 + z_2 + z_3 + z_4 + z_5 + z_6 + z_7 + z_8} \quad (i = 1, \dots, 8).$$

Для кожного рішення розраховуємо величини

$$P(c_i) = v_i + w_i \quad (i = 1, \dots, 8).$$

Далі можливий один з двох підходів:

1. Пропущений піксель заповнюється кольором  $c = c_i$ , для якого  $P(c)$  максимальна.
2. Для заповнення пропущеного пікселя замість кольору вибирається зважена сума  $c = \sum_{i=1}^8 c_i P(c_i)$ .

Перший підхід дозволяє чіткіше відновлювати межі колірних областей, а другий – робити плавніші переходи.

Реалізацію алгоритму здійснено засобами C#. Результатом виконаної програми є відновлене зображення на рис.3 із зашумленого зображення на рис.1.

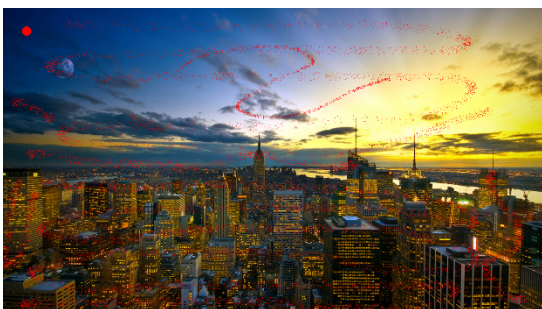


Рис. 1. Зображення з імпульсними шумами



Рис. 2. Результат функції “Прибрати імпульсні шуми”



Рис. 3. Результат функції “Відновити втрачені фрагменти”

#### ЛІТЕРАТУРА REFERENCES

- [1] J. Harikiran, B. Saichandana, B. Divakar “Impulse noise removal in digital images,” International Journal of Computer Applications (0975-8887) Vol.10, No 8, 2010.
- [2] M. Sezgin, B. Sankur “Survey on Image thresholding technique and qualitative performance evaluation,” Journal of Electronic Imaging vol. 13(1); pp 146-165, 2004.
- [3] И.Б. Ларионов “Алгоритм автоматизированного восстановления поврежденных графических файлов” Вестник Омского университета, 2011, с. 176-177.
- [4] С.В. Белим, Н.Ф. Богаченко “Применение метода анализа иерархий для оценки рисков утечки полномочий с ролевым разграничением доступа”, Информационно-управляющие системы. Москва 2013, с. 67-27.
- [5] J.N.Ahamed, V.Rajamani “Design of hybrid filter for denoising images” American Journal of Scientific Research, p. 5-14, 2009.
- [6] T.L.Saaty “Relative Measurement and its Generalization in Decision Making” Review of the Royal Spanish Academy of Sciences. p. 251-318, 2008.