

УДК 004.921

**М.М. Ботвін,**  
(аспірант, Державний університет інфраструктури та технологій)  
**О.А. Герцій, к.т.н., доц.**  
(завідувач каф. «Автоматизація та комп'ютерно-інтегровані технології  
транспортів», Державний університет інфраструктури та технологій)

### АНАЛІЗ ГРАФІЧНИХ ФОРМАТІВ ТА АЛГОРИТМІВ КОДУВАННЯ ЦИФРОВИХ ЗОБРАЖЕНЬ

У статті проведено аналіз основних графічних форматів файлів, а також алгоритмів кодування цифрових зображень, які дозволяють зменшити їх інформаційний об'єм з метою збільшення швидкості передачі даних по цифрових каналах зв'язку, а також дозволять зберігати більшу кількість файлів на запам'ятовуючих пристроях.

**Ключові слова:** графічні формати, алгоритми кодування цифрових зображень, інформаційна ємність.

**Вступ.** Кожне цифрове зображення після кодування за допомогою певного алгоритму і запису на носій стає файлом. Залежно від алгоритмів кодування цифрових зображень розрізняють графічні формати файлів. Графічні формати файлів є стандартизованими засобами організації та зберігання цифрових зображень, таких як фотографії та малюнки. Графічні формати діляться на растрові та векторні. Більшість графічних форматів реалізують стиснення даних (одні – з втратами, інші – без).

**Постановка задачі.** Однією з характеристик цифрового зображення є його розмір. Загальна кількість бітів  $b$ , необхідна для зберігання цифрового зображення, визначається по формулі [1]:

$$b = M \times N \times k \quad (1)$$

де  $M$  – кількість стрічок зображення;

$N$  – кількість стовпчиків зображення;

$k$  – розрядність кодування кольору (біт).

У зв'язку з тим, що цифровим зображенням властива велика інформаційна ємність, наприклад, для зберігання 24-бітного зображення розмірами  $8192 \times 4320$  елементів необхідний суттєвий об'єм пам'яті, а саме: 106.16832 Мб; час на передачу такого зображення, наприклад, в корпоративній мережі, від одного користувача іншому, за

© Ботвін М. М., Герцій О. А. 2018

технологією Fast Ethernet на швидкості 100 Мбіт/с сягатиме майже 8,5 с, що у деяких випадках є занадто великим і неприйнятним, то постає задача стиснення зображень для зменшення їхнього об'єму на зберігання та передачу.

Над даним питанням в області цифрової обробки зображень працюють фахівці багатьох країн світу, зокрема, питання щодо скорочення розмірів графічних файлів з метою збільшення їх швидкості передачі в комп'ютерних мережах та економії місця на серверах центрів обробки даних залишаються актуальними, в зв'язку з щоденним ростом графічної інформації в інтернет-просторі [2, 3].

**Наукова новизна.** У даній статті проводиться аналіз алгоритмів кодування та основних відомих растрових графічних форматів файлів, таких як BMP, GIF, TIFF, PNG, JPEG і пропонується переведення цифрових зображень, збережених у дані формати, шляхом конвертації в один із векторних форматів, такий як SVGZ, що дасть можливість отримати більш високий відсоток стиснення зображень певних класів і збільшити швидкість їхньої передачі в комп'ютерних мережах.

**Мета статті.** Дослідити графічні формати файлів, а також алгоритми кодування цифрових зображень. Показати, які алгоритми у яких випадках найбільш ефективні та як можна досягти кращих результатів стиснення, комбінуючи методи кодування.

**Основний матеріал.** Із формули (1) можна зробити висновок, що для зменшення інформаційного розміру цифрового зображення можна зменшити його роздільність (рис.1), або глибину кольору, тобто розрядність кодування кольору (рис. 2). Але при цьому втрачаються візуальні характеристики зображення, а саме чіткість та кольорова гамма. В результаті складно встановити, що демонструє зображення.



Рис. 1. Ілюстрація зменшення чіткості цифрового зображення при зменшенні роздільної здатності

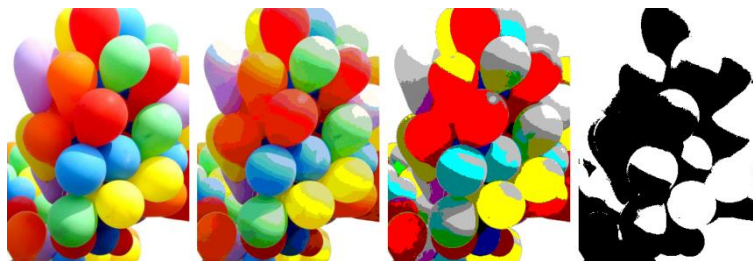


Рис. 2. Ілюстрація зменшення кольорової гамми при зменшенні розрядності кодування кольору

Але не тільки ці два способи дозволяють зменшити розмір зображення. Для зменшення інформаційного об'єму цифрових зображень застосовують певні алгоритми кодування, які дозволяють «стиснути» зображення. У результаті стиснення зменшується розмір зображення, через що зменшується час на його передачу по мережі та економиться місце для зберігання.

Алгоритми кодування поділяються на два класи: алгоритми кодування без втрат та алгоритми кодування з втратами. У першому випадку в процесі кодування інформація про зображення зберігається в повному об'ємі, а в іншому – частково втрачається. Перша група методів стиснення забезпечує відновлення вихідного зображення без втрат та спотворень. Для зберігання зображень, призначених для подальшої обробки, варто застосовувати формати, що використовують саме такі алгоритми кодування. При зберіганні зображень в формати, які застосовують другу групу методів стиснення – проявляються спотворення, так звані артефакти. Проте, якщо зображення призначене для візуального сприйняття, це не завжди критично. У ряді випадків вихідний сигнал вже містить такі спотворення та шуми, що невеликі втрати інформації при кодуванні (на користь високої ступені стиснення) не зіпсують якості зображення в цілому.

Розглянемо деякі алгоритми кодування та графічні формати, в яких застосовуються ці алгоритми.

#### *Алгоритми кодування без втрат*

1. Групове кодування – від англ. Run-length encoding (RLE) – це метод компресії даних, який використовується у більшості растрових форматах, таких як TIFF, BMP та PCX. Групове кодування підходить для компресії різних типів даних незалежно від їхнього інформаційного змісту, але контекст даних вплине на коефіцієнт компресії, досягнутий RLE. Хоча основна маса алгоритмів RLE не може досягти високих показників стиснення ніж більш сучасні методи стиснення, RLE є простим для реалізації та швидким у виконанні, що робить його хорошою альтернативою більш складному алгоритму компресії [4].

RLE зменшує фізичний розмір повторюваного рядка символів. Цей повторюваний рядок, який називається запуском, в основному кодується в два байти. 1-й байт являє собою кількість символів у запуску і називається підрахунком виконання. На практиці закодований пробіг може містити від 1 до 128 або 256 символів; графа бігу зазвичай містить як кількість символів мінус один (значення в діапазоні від 0 до 127 або 255). 2-й байт – це значення символу в запуску, який знаходиться в діапазоні від 0 до 255, і називається значенням виконання.

Відомо багато схем RLE кодування, одну з яких можна відобразити таким чином:

– вхідний потік даних: 18 9 55 0 0 0 0 98 6 17 0 0 46 24 0 0 0 0 0 4 68 0 0 0 9

– після кодування: 18 9 55 4 0 98 6 17 2 0 46 24 6 0 4 68 3 0 9

2. LZW (англ. Lempel–Ziv–Welch), алгоритм Лемпеля–Зіва–Велча – універсальний метод стиснення даних без втрат за допомогою алгоритму пошуку на основі таблиці, винайденими Абрахамом Лемпелем, Якобом Зівом та Террі Уелчем. Два загальноприйнятих формати файлів, в яких використовується стиснення LZV, – це формат зображень GIF, поданий з веб-сайтів та формат зображення TIFF. Стиснення LZW також підходить для стиснення текстових файлів [5].

## ІНФОРМАЦІЙНІ, ТЕЛЕКОМУНІКАЦІЙНІ ТА РЕСУРСОЗБЕРІГАЮЧІ ТЕХНОЛОГІЇ

Створюється таблиця кольорів, які присутні в зображенні, що піддається стисненню. Отже, замість показника кольору пікселя можна застосовувати індекс з таблиці. Кольори, які найчастіше будуть зустрічатися на зображенні, матимуть менші індекси, а кольори, які рідко зустрічатимуться, розміщуються в кінці таблиці. Палітра (таблиця кольорів) розміщується між заголовком і зображенням, табл. 1.

Потік даних на вході: 125 147 203 6 121 97 245 247 61 13 208 147 203 6 245 247

Дані, після кодування: 125 258 121 91 259 61 13 208 258 259

Таблиця 1

Таблиця кольорів при кодуванні методом LZW

Індекс	Значення
0000	0
0001	1
0256	256
0257	257
0258	147 203 6
0259	245 247
4095	xxx xxx xxx

Метод LZW краще підходить для зображень, які містять однорідні, вільні від шуму ділянки кольорів. При цьому даний метод діє набагато краще, ніж RLE, при компресії графічних даних, але процес стиснення та декодування відбувається повільніше.

Застосовується в таких форматах: GIF, TIFF.

3. Алгоритм Хаффмана – це алгоритм компресії даних без втрат. Ідея алгоритму полягає в тому, щоб присвоїти коди змінної довжини для введення символів; довжина призначених кодів залежить від частот відповідних символів. Найбільш частий символ отримує найменший код, а найменш частий символ отримує найбільший код [6].

Метод стиснення Хаффмана можна проілюструвати таким чином, табл. 2.

Вхідний потік даних: C E G A D F B E A

Потік даних після кодування: 0010 0001 000011 1 0011 000010 01 0001 1

Групування по байтах: (0010 0001) (000011 1 0) (011 00001) (0 01 00 1 0 1)

Отримані коди унікальні, тобто вони можуть бути записані в потік даних без роздільників і маркерів. По кількості нулів до і після 1 програма відновлення може однозначно визначити значення елемента.

Застосовується в наступних форматах: TIFF, GIF.

Таблиця 2

Таблиця реалізації кодування за алгоритмом Хаффмана

Значення	Частота появи	Код Хаффмана
A	0.154	1
B	0.110	01
C	0.072	0010
D	0.063	0011
E	0.059	0001
F	0.015	000010
G	0.011	000011

4. ССІТТ-3 – алгоритм стиснення даних без втрат, був запропонований третьою групою по стандартизації Міжнародного Консультативного Комітету по Телеграфії і Телефонії. Це один з форматів, які використовують факсимільні апарати. Він зазвичай вбудовується в інший формат файла зображення, наприклад TIFF [7].

Він поставляється в двох основних різновидах: так званих 1-мірних і 2-мірних (хоча обидва використовуються на 2-мірних зображеннях). 1-мірний використовує стиснення Modified Huffman (MH), а 2-мірний різновид використовує модифіковану READ (MR) систему стиснення.

5. JBIG (англ. Joint Bi-level Image Experts Group) – алгоритм кодування для дворівневих зображень (наприклад, чорно-білих друкованих матеріалів), що складається з однієї прямокутної площини біт, причому кожен піксель бере на одне з двох можливих кольорів. JBIG2 був стандартизований в 2001 році [8].

Стиснення такого типу зображення також розглядаються існуючими стандартами, наприклад, MH & MR, MMR і JBIG1. JBIG2 був підготовлений для стиснення зображень з втратами та з втратами в збітках, а також забезпечує кращу якість стиснення без втрат, ніж існуючі стандарти та забезпечує стиснення з втратами при значно вищих коефіцієнтах стиснення, ніж співвідношення втрат для існуючих стандартів, майже не спостерігаючи помітної деградації якості, використовуючи методи збору та заміни візерунків на додаток до технологій існуючих стандартів.

Окрім очевидного факсимільного додатка, JBIG2 буде корисним для зберігання та архівування документів, кодування зображень у World Wide Web, бездротової передачі даних, друкарського матеріалу та навіть телеконференцій. PDF-файли, що мають версії 1.4 та новіші, можуть містити дані зі стисненням JBIG2. Багато кольорів можна обробляти за допомогою відповідного стандарту вищого рівня, такого як Рекомендація T.44 МСЕ-Т.

6. Lossless JPEG – метод компресії без втрат, розроблений групою експертів в області фотографії (Joint Photographic Experts Group). Lossless JPEG був розроблений як пізніше доповнення до JPEG в 1993 році, використовуючи зовсім іншу техніку із стандарту JPEG з втратами. Вона використовує передбачувану схему на основі трьох найближчих (причинних) сусідів (верхня, ліва і верхня ліва), а кодування ентропії використовується на помилці прогнозування. Стандартні бібліотеки Independent JPEG Group не можуть кодувати та декодувати його, але Ken Murchison з Oceana Matrix Ltd. написав патч, який розширює бібліотеку IJG для обробки JPEG без втрат JPEG. Lossless JPEG має певну популярність у медичній візуалізації, і використовується в DNG та деяких цифрових камерах для стиснення сирих зображень, але інакше він ніколи не був широко поширений. Adobe DNG SDK надає бібліотеку програмного забезпечення для кодування та декодування без втрат JPEG з частотою до 16 біт на вибірку [9].

Також розглянемо *алгоритми стиснення з втратами*.

1. JPEG – один з найпопулярніших растрових графічних форматів, був створений у 1992 р. (остання версія, 1994 р.) як результат процесу, що розпочався у 1986 р., оснований на дискретному косинусному перетворенні (DCT). Ступінь стиснення можна регулювати, дозволяючи вибирати компроміс між розміром

зберігання та якістю зображення. JPEG, як правило, досягає стиснення 10:1 з невеликою відчутною втратою якості зображення [10].

Алгоритм JPEG найкраще підходить для компресії картин та фотографій, які містять реалістичні сцени з плавними переходами кольору та яскравості. Найбільшого застосування алгоритм JPEG дістав в цифрових фотографіях і для зберігання та передачі цифрових зображень по Інтернет мережі. Даний алгоритм не застосовується для кодування зображень при багатоетапному обробленні даних, бо погіршення будуть вноситися в зображення кожного разу при зберіганні періодичних результатів обробки.

Також алгоритм не повинен застосовуватися до зображень, які в подальшому будуть піддаватися аналізу, наприклад, астрономічні або медичні знімки.

Таблиця 3

**Характеристики основних алгоритмів кодування цифрових зображень**

Алгоритм	Розмірність	Коефіцієнти стиснення	Симетричність за часом	На що орієнтований	Втрати
RLE	1D	32, 2, 0.5	1	3, 4-х бітні	Є
LZW	1D	1000, 4, 5/7	1.2-3	1-8 бітні	Є
Хаффмана	1D	8, 1.5, 1	1-1.5	8 бітними	Є
CCITT-3	1D	213 (3), 5, 0.25	~ 1	1-бітні	Є
JBIG	2D	2-30 разів	~ 1	1-бітні	Є
Lossless JPEG	2D	20, 2, 1	~ 1	24-бітові, сірі	Є
JPEG	2D	2-20 разів	~ 1	24-бітові, сірі	Немає

Неформально можна виділити такі класи зображень.

1) Бінарні – зображення, що мають тільки два можливих значення кожного пікселя – 0 і 1. Як правило, два кольори, що використовуються у бінарному зображенні: білий та чорний. Їх ще називають однобітними, тому що для представлення одного пікселя необхідна одна двозначна цифра.

2) Сіро-масштабні – зображення у градаціях сірого, які ще називають одноколірними. Кількість бітів, яка використовуються для кожного пікселя, визначає кількість доступних рівнів сірого. Типове монохромне зображення містить 8 бітів/піксель, яке дозволяє нам бачити 256 різних рівнів сірого кольору.

В медицині та астрономії використовуються монохромні зображення з 12 або 16 бітами/піксель. Ці додаткові рівні сірого стають корисними, коли необхідно розпізнати невелику секцію зображення більш детально.

3) Кольорові – зображення, які можна моделювати як дані з триколірних монохромних зображень (червоний, синій і зелений), де кожен рядок даних відповідає іншому кольору. Використовуючи 8 біт монохромного стандарту як моделі, відповідне кольорове зображення становитиме 24 біт/піксель (8 біт на кожен з трьох кольорів).

4) Мультиспектральні – зображення, які містять інформацію поза межами норми людського сприйняття. Сюди можна віднести інфрачервоні, ультрафіолетові, рентгенівські, акустичні або радарні дані. Це не зображення у звичайному

## ІНФОРМАЦІЙНІ, ТЕЛЕКОМУНІКАЦІЙНІ ТА РЕСУРСОЗБЕРІГАЮЧІ ТЕХНОЛОГІЇ

розумінні, оскільки представлена інформація не є безпосередньо видимою людській системі. Проте інформація часто представлена у візуальній формі шляхом відображення різними спектральними смугами до компонентів RGB.

Проведемо порівняльний аналіз найбільш розповсюджених графічних форматів, які використовуються в процесі роботи з растровою графікою на персональних комп'ютерах при створенні Web-сайтів і публікаціях в інтернет. Як моделювання візьмемо чотири зображення, кожне з яких належить до одного із класів зображень.

Проаналізувавши візуально чотири класи зображень, збережених у різних графічних форматах, та їх гістограму інформаційної ємності можна зробити висновок, що найменший розмір займають файли, збережені в форматі BMP (монохромний рисунок), але при цьому найбільш візуально-прийнятним можна вважати бінарне зображення та, деякою мірою, мультиспектральне.

Таблиця 4

**Цифрові зображення, збережені у різні графічні формати**

Формат/ Клас	Бінарне	Сіро- масштабне	Кольорове	Мульти- спектральне
BMP, монохромне	4526 байт	4710 байт	6650 байт	6542 байт
BMP, 16- кольорове	17974 байт	19270 байт	25738 байт	25390 байт
BMP, 256- кольорове	36046 байт	38262 байт	51586 байт	51622 байт
BMP, 24-бітне	104958 байт	110278 байт	150846 байт	151686 байт
JPEG	22915 байт	8251 байт	13923 байт	40081 байт
GIF	11457 байт	9912 байт	26842 байт	38378 байт
TIFF	45560 байт	24360 байт	71930 байт	178040 байт
PNG	43126 байт	27356 байт	76384 байт	141945 байт

Зі збільшенням глибини кольору від монохромного до 24-бітного, в тому ж форматі, збільшився розмір файлів. При цьому набагато збільшились візуальні характеристики сіро-масштабного, кольорового та мультиспектрального зображень.

Цікавіші результати показали зображення, збережені в форматах JPEG, GIF, TIFF та PNG. Так, найменший розмір становить сіро-масштабне зображення в форматі JPEG, а найбільших розмірів зайняло мультиспектральне зображення в форматі TIFF та PNG. Вибираючи «золоту середину» між візуальною сприйнятливістю та розміром файла, найкращий результат показали останні три класи зображень, збережені в форматі JPEG.

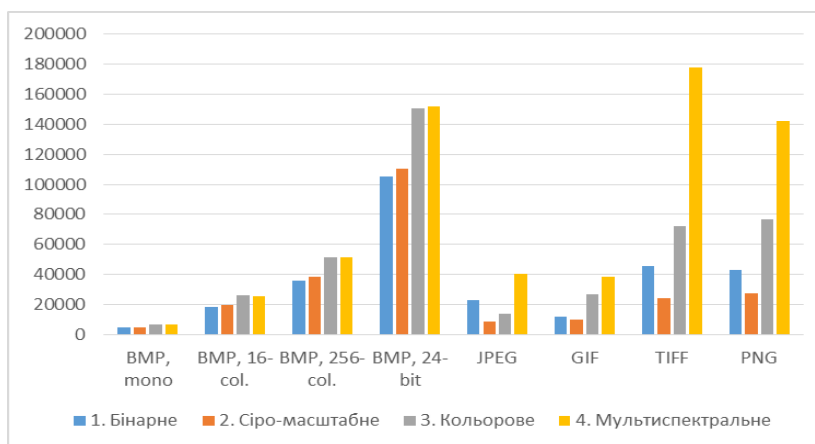
В даному моделюванні розглянуто властивості растрових форматів. Але що буде, якщо перевести дані зображення у векторну графіку, адже при відносно нескладній деталізації зображення розмір файла займає невеликі розміри.

Перетворимо зображення із формату BMP (24-бітне), які займають найбільші об'єми в один із форматів векторної графіки, наприклад, в SVGZ (Scalable Vector Graphics, Z – стиснутий).

Із проведеного моделювання можна зробити висновок, що зображення збережені в форматі SVGZ, трохи втратили свої візуальні характеристики, але,

## ІНФОРМАЦІЙНІ, ТЕЛЕКОМУНІКАЦІЙНІ ТА РЕСУРСОЗБЕРІГАЮЧІ ТЕХНОЛОГІЇ

при цьому, файли перших трьох класів займають в два рази менше місця для зберігання, окрім останнього, його розмір виріс в 3,55 разів. Це супроводжується тим, що розмір файла векторного зображення залежить від кількості векторів, і оскільки четверте зображення містить в собі велику кількість дрібних деталей, то після переведення у векторний формат виріс розмір файла.

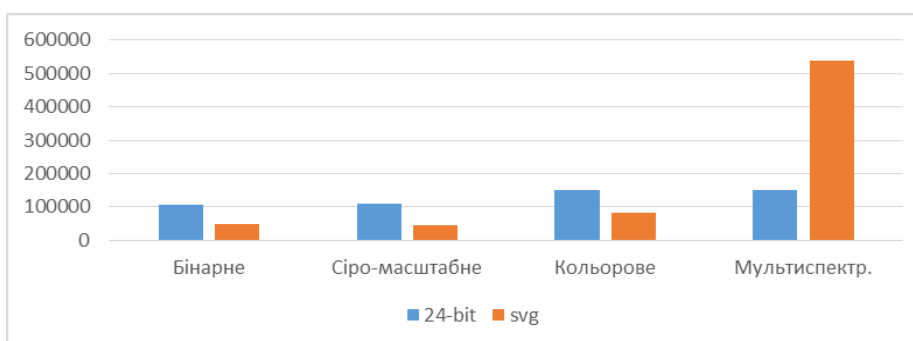


**Рис. 3. Гістограма інформаційної ємності 4-х класів зображень, збережених у різних графічних форматах**

*Таблиця 5*

### Порівняння графічних форматів BMP (24-бітне) та SVGZ

Формат/ Клас	Бінарне	Сіро-масштабне	Кольорове	Мульти-спектральне
BMP, 24-бітне	104958 байт	110278 байт	150846 байт	151686 байт
SVGZ	46812 байт	45230 байт	81770 байт	539055 байт



**Рис. 4. Гістограма порівняння розміру файлів чотирьох класів зображень, збережених в форматі BMP (24-бітне) та SVGZ (високої якості)**



Таблиця 6

Порівняння графічних форматів JPEG та SVGZ

Формат/ Клас	Бінарне	Сіро- масштабне	Кольорове	Мульти- спектральне
JPEG	22915 байт	8251 байт	13923 байт	40081 байт
SVGZ	5915 байт	13494 байт	19175 байт	73083 байт

Проте розмір файлів в формі SVGZ попереднього моделювання досить високий в порівнянні з зображеннями, збереженими в форматі JPEG. Якщо перевести зображення із JPEG в SVGZ отримаєм такі результати.

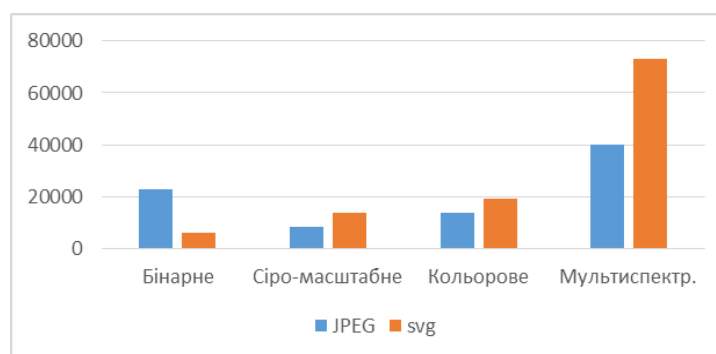


Рис. 5. Гістограма порівняння розміру файлів чотирьох класів зображень, збережених в форматі JPEG та SVGZ (низької якості)

Із рис. 5 видно, що найкращий результат показало бінарне зображення, переведене із растрового формату JPEG у векторний формат SVGZ.

**Висновки та пропозиції.** Залежно від класу зображень (бінарні, сіро-масштабні, кольорові, мультиспектральні), для якнайкращого їх стиснення, потрібно використовувати відповідні алгоритми кодування. При цьому цікавою і достатньо складною задачею є пошук найкращого алгоритму для конкретного класу зображень. Тому, судячи із проведених моделювань, можна зробити висновок, що використовувати векторний формат SVGZ для переведення зображень із растрової у векторну графіку доцільно для першого класу зображень, збережених в форматі JPEG, а також до зображень, які містять однорідні, вільні від шуму ділянки кольорів з великими областями повторюваного кольору, наприклад, ділова графіка, схеми, малюнки і т. д. Таке перетворення дозволить зменшити файл у розмірах, що дозволить економити місце на запам'ятовуючих пристроях та збільшити швидкість передачі файлів по мережі.

Даний аналіз може бути корисним для сфери транспорту, оскільки в інформаційних та комп'ютерних технологіях даної галузі широко використовуються цифрові графічні зображення, наприклад, при ревізії залізничних вагонів й локомотивів після ремонту чи планового технічного обслуговування на предмет виявлення тріщин, цифровими фото-відео засобами

фіксується наявність дефектів, які заносяться в бази даних. Розмір отриманої бази даних таких зображень є значним, то їхнє стиснення задля компактного зберігання й швидкої передачі в комп'ютерних мережах досить актуально.

#### ЛІТЕРАТУРА

1. Загальні значення параметрів цифрового зображення. [Електронний ресурс]. – Режим доступу: [https://nptel.ac.in/courses/117104069/chapter\\_1/1\\_1b.html](https://nptel.ac.in/courses/117104069/chapter_1/1_1b.html)
2. Kasban H., Mohsen A.M. Performance Improvement of Digital Image Transmission over Mobile WiMAX Networks // Wireless Personal Communications. – 2017., Volume 94, Issue 3, pp. 1087–1103.
3. Yvette E. Gelogo and Tai-hoon Kim. Compressed Images Transmission Issues and Solutions // International Journal of Computer Graphics. – 2014. – Vol.5, No.1 – pp.1-8.
4. Алгоритм групового кодування RLE. [Електронний ресурс]. – Режим доступу: [https://www.fileformat.info/mirror/egff/ch09\\_03.htm](https://www.fileformat.info/mirror/egff/ch09_03.htm)
5. Метод LZW. [Електронний ресурс]. – Режим доступу: <https://whatis.techtarget.com/definition/LZW-compression>.
6. Алгоритм Хаффмана. [Електронний ресурс]. – Режим доступу: <https://www.geeksforgeeks.org/huffman-coding-greedy-algo-3/>
7. Алгоритм Хаффмана з фіксованою таблицею ССІТТ-3. [Електронний ресурс]. – Режим доступу: [http://fileformats.archiveteam.org/wiki/CCITT\\_Group\\_3](http://fileformats.archiveteam.org/wiki/CCITT_Group_3).
8. Алгоритм JBIG. [Електронний ресурс]. – Режим доступу: <https://jpeg.org/jbig/>
9. Формат JPEG-LS. [Електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/JPEG-LS>.
10. Алгоритм JPEG. [Електронний ресурс]. – Режим доступу: <https://jpeg.org/jpeg/>

#### REFERENCES

1. Common values of digital image parameters. [Electronic resource]. – Mode of access: [https://nptel.ac.in/courses/117104069/chapter\\_1/1\\_1b.html](https://nptel.ac.in/courses/117104069/chapter_1/1_1b.html)
2. Kasban H., Mohsen A.M. (2017). Performance Improvement of Digital Image Transmission over Mobile WiMAX Networks. Wireless Personal Communications, Volume 94, Issue 3, 1087–1103.
3. Yvette E. Gelogo and Tai-hoon Kim (2014). Compressed Images Transmission Issues and Solutions. International Journal of Computer Graphics, Vol.5, No.1, 1-8.
4. Run-Length Encoding (RLE). [Electronic resource]. – Mode of access: [https://www.fileformat.info/mirror/egff/ch09\\_03.htm](https://www.fileformat.info/mirror/egff/ch09_03.htm)
5. LZW compression. [Electronic resource]. – Mode of access: <https://whatis.techtarget.com/definition/LZW-compression>.
6. Huffman Coding. [Electronic resource]. – Mode of access: <https://www.geeksforgeeks.org/huffman-coding-greedy-algo-3/>
7. Huffman algorithm with fixed table CCITT Group 3. [Electronic resource]. – Mode of access: [http://fileformats.archiveteam.org/wiki/CCITT\\_Group\\_3](http://fileformats.archiveteam.org/wiki/CCITT_Group_3).
8. JBIG algorithm. [Electronic resource]. – Mode of access: <https://jpeg.org/jbig/>
9. JPEG-LS format. [Electronic resource]. – Mode of access: <https://ru.wikipedia.org/wiki/JPEG-LS>.
10. JPEG algorithm. [Electronic resource]. – Mode of access: <https://jpeg.org/jpeg/>.

**М.Н. Ботвін**

*(аспірант, Государственный университет инфраструктуры и технологий);*

*А.А. Герций, к.т.н., доц.*

*(заведующий каф. «Автоматизация и компьютерно-интегрированные технологии транспорта», Государственный университет инфраструктуры и технологий)*

### **АНАЛИЗ ГРАФИЧЕСКИХ ФОРМАТОВ И АЛГОРИТМОВ КОДИРОВАНИЯ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ**

*В статье проведено анализ основных графических форматов файлов, а также алгоритмов кодирования цифровых изображений, которые позволяют уменьшить их информационный объем с целью увеличения скорости передачи данных по цифровым каналам связи, а также позволяют сохранять большее количество файлов на запоминающих устройствах.*

***Ключевые слова:** графический формат, алгоритмы кодирования цифровых изображений, информационная ёмкость.*

**Maxim Botvin,**

*(Postgraduate of State University of Infrastructure and Technologies)*

*Olexander A. Gertsy, PhD (Technical Sciences), Associate Professor*

*(Head of the Department of Automation and Computer-Integrated Transport Technologies, State University for Infrastructure and Technologies)*

### **RESEARCH OF GRAPHIC FORMATS AND CODING ALGORITHMS OF DIGITAL IMAGES**

*The object of this paper is to analyze the graphics file formats and coding algorithms of digital images. The article begins with a short discussion on the problem of large information sizes of digital images. Then there follows a discussion on the most common coding algorithms and graphical formats. The simulation of four classes of images stored in different graphic formats is presented, in particular, how the coding of the corresponding algorithms influenced the visual and information characteristics of the images. The next paragraph deals with comparison raster graphic format BMP (24-bit) and vector format SVGZ in high quality. Also a similar comparison of images stored in JPEG and SVGZ (low quality) formats was performed and presented. The final paragraph states that the best result showed a binary image translated from bitmap format JPEG into vector format SVGZ. Therefore, judging from the performed simulations, we can conclude that using the SVGZ vector format to convert images from bitmap to vector graphics is appropriate for the first class of images stored in JPEG format, and to the images that contain homogeneous, noise-free colored areas with large areas of repetitive color, such as business graphics, charts, drawings, etc.*

***Keywords:** graphic formats, coding algorithms of digital images, information capacity.*