DOI: 10.15587/2312-8372.2018.134978

СТИСНЕННЯ ЗОБРАЖЕНЬ МЕТОДАМИ КУБІЧНИХ СПЛАЙН-ФУНКЦІЙ

Коцюбівська К. І., Чайковська О. А., Толмач М. С., Хрущ С. С.

1. Вступ

В умовах сьогодення, коли інформація набуває все більшого значення, а шляхи розповсюдження даних стають більш різноманітними питання стиснення даних посідає одне з найважливіших місць. Особливо така проблема стосується файлів зображень.

Однак в процесі перетворення аналогового сигналу в цифрову форму значно розширюється смуга частот, яку займають ці сигнали, що вимагає каналів передачі з високою пропускною здатністю та великих об’ємів пам’яті для їх зберігання. Особливо це відноситься до зображень, тому актуальним є виконання стиснення зображень перед їх збереженням і передачею по каналах зв’язку.

Особливістю стиснення зображень є те, що досягти високих показників стиснення можна тільки за рахунок втрат інформації, причому ці втрати не повинні бути помітні візуально.

При виборі алгоритмів важливо розуміти їхні позитивні і негативні сторони. Якщо обраний алгоритм із втратою даних, то варто зрозуміти його природу й умови, при яких зображення будуть псуватися. Використання нових оптимальних алгоритмів дозволить зберегти якість зображень, десятки і сотні мегабайт дискового простору.

Серед відомих методів стиснення (кодування) зображень найбільш оптимальні співвідношення між втратами і коефіцієнтом стиснення можна досягти при застосуванні інтерполяційних методів, а саме за рахунок використання кубічних сплайн-функцій.

2. Об'єкт дослідження та його технологічний аудит

Об’єктом дослідження є алгоритми стиснення зображень на основі математичних методів. Основною задачею нової розробки є підвищення показників стиснення при малих втратах точності зображення. Головною оцінкою точності зображення є візуальна якість, характеристикою методу є швидкодія.

Найбільш розповсюдженою комерційною технологією є методика *jpeg*, в якій стиснення досягається за рахунок косинус-розкладу Фур’є. Інший підхід може бути реалізований на основі сингулярного використання матриці. Модифікація існуючих технологій полягає в розробці нових математичних підходів та алгоритмів, а також створення нових програмних технологій та розв’язків [1].

Першими для архівації зображень стали застосовуватися алгоритми, які використовувалися і використовуються в системах резервного копіювання, при створенні дистрибутивів і т. п. Ці алгоритми архівували інформацію без змін. Однак основною тенденцією останнім часом стало використання нових класів зображень. Старі алгоритми перестали задовольняти вимогам, що пред'являються до архівації.

Багато зображень практично не стискалися. Це призвело до створення нового типу алгоритмів, які стискають з втратою інформації. Як правило, коефіцієнт архівації, а отже, ступінь втрат якості в них, можна задавати. При цьому досягається компроміс між розміром і якістю зображень.

Саме в оптимізації параметрів стиснення та наступного відновлення зображень полягає основна вимога до математичних моделей та створення алгоритмів кодування зображень.

Використання алгоритму, що розробляється, дозволить отримати хороші показники стиснення, причому не виконуючи складних обчислень, що збільшує швидкодію.

3. Мета та задачі дослідження

Метою роботи є створення та дослідження алгоритмів і програмного забезпечення стиснення зображень на основі кубічних сплайн-функцій.

Для досягнення поставленої мети необхідно виконати наступні задачі:

1. Обґрунтувати переваги стиснення на основі кубічних сплайн-функцій.
2. Проаналізувати якість стиснення зображень з використанням методу сплайнової інтерполяції.
3. Порівняти отримані результати з вже існуючими стандартами.

4. Дослідження існуючих рішень проблеми

Всі алгоритми стиснення оперують вхідним потоком інформації, мінімальною одиницею якої є біт, а максимальною – кілька біт, байт чи кілька байт. Метою процесу стиснення, як правило, є одержання більш компактного вихідного потоку інформаційних одиниць з деякого початкового некомпактного вхідного потоку за допомогою деякого їхнього перетворення.

Останнім часом зображення та ілюстрації стали використовуватися повсюди. Проблема, пов'язана з їхнім великим розміром, з'явилася при роботі і на робочих станціях, і на персональних комп'ютерах. Зважаючи на те, що звичайно використовується кілька ілюстрацій, і те, що вони часто бувають набагато більшого розміру (наприклад, при кольоровому друці), зберігати їх у неупакованому вигляді стає накладно. В останні роки рішенню цієї проблеми приділяється досить серйозна увага [1–5]. Розроблено велику кількість різних алгоритмів архівації графіки: використовувалися як видозмінені універсальні, так і абсолютно нові алгоритми, орієнтовані тільки на зображення. Більш того, були розроблені алгоритми, орієнтовані тільки на конкретний клас зображень.

Проведемо порівняння відомих методів кодування, вказавши їх переваги та недоліки.

Групове кодування – найбільш відомий простий підхід і алгоритм стиснення інформації оборотним шляхом – це кодування серій послідовностей (RLE). Проблема всіх аналогічних методів полягає у визначенні способу, за допомогою якого алгоритм декодування міг би відрізнити в результуючому потоці байтів кодовану серію від інших – некодованих послідовностей байтів, що вимагає додаткових витрат. Дані методи, як правило, досить ефективні для стиснення растрових графічних зображень (BMP, PCX, TIF, GIF), тому що останні містять досить багато довгих серій послідовностей байтів, що повторюються. Недоліком методу RLE є досить низький степінь стиснення файлів з малим числом серій і, що ще гірше – з малим числом повторюваних байтів у серіях. При використанні адаптивного *кодування Хаффмена* [1] ускладнення алгоритму полягає в необхідності постійного коректування дерева і кодів символів основного алфавіту відповідно до статистики вхідного потоку, що змінюється. Основним же недоліком даного методу є те, що при кодуванні потоку з двосимвольним алфавітом стиснення завжди відсутнє, тому що незважаючи на різні імовірності появи символів у вхідному потоці алгоритм фактично зводить їх до 1/2 [1]. При розробці методу *арифметичного кодування* виникають дві проблеми: по-перше, необхідна арифметика з плаваючою точкою, теоретично, необмеженої точності, і, по-друге, результат кодування стає відомий лише при закінченні вхідного потоку. *Алгоритм Лемпеля-Зіва-Велча (LZW)* відрізняють висока швидкість роботи як при упакуванні, так і при розпакуванні, досить скромні вимоги до пам'яті і проста апаратна реалізація, але недоліком є низький ступінь стиснення в порівнянні зі схемою двоступінчастого кодування [2]. Особливістю *фрактального стиснення* є потреба в колосальних обчислювальних потужностях при архівації. Найвідомішою програмою-архіваторм на сьогоднішній день є конвертор графічних файлів фірми Microsoft, що опирається на використання методу стиснення JPEG. Не дуже приємною властивістю JPEG є те, що нерідко горизонтальні і вертикальні смуги на дисплеї абсолютно не видимі, і можуть проявитися тільки при друці у вигляді муарового візерунка. У всіх наведених алгоритмів по сьогоднішніх мірках, занадто маленький коефіцієнт архівації.

Основними технічними характеристиками процесів стиснення і результатів їхньої роботи є:

− степінь стиснення (compress rating) чи відношення (ratio) обсягів вихідного і результуючого потоків;

− швидкість стиснення – час, що витрачається на стиснення деякого обсягу інформації вхідного потоку, до одержання з нього еквівалентного вихідного потоку;

− якість стиснення – величина, що показує наскільки сильно упакований вихідний потік визначається відношенням розміру упакованого файлу до розміру файлу, отриманого після застосування до нього повторного стиснення по цьому ж чи іншому алгоритмові.

Щоб коректно оцінювати напрямок зміни алгоритмів, недостатньо визначити тільки класи зображень. Необхідно задати і визначені критерії, одними з яких є гірший, середній і кращий коефіцієнти стиснення. Тобто частка, на яку зросте розмір зображення, якщо вихідні дані будуть найгіршими; деякий середньостатистичний коефіцієнт для того класу зображень, на який орієнтований алгоритм; і, нарешті, кращий коефіцієнт. Останній необхідний лише теоретично, оскільки показує степінь стиснення найкращого (як правило, абсолютно чорного) зображення, нерідко фіксованого розміру.

Далі будуть розглянуті статичні растрові зображення, що представляють собою двовимірний масив чисел – пікселів. Усі зображення можна підрозділити на дві групи: з палітрою і без неї. У зображень з палітрою в пікселі зберігається число – індекс у деякому одномірному векторі кольорів, що називається палітрою. Найчастіше зустрічаються палітри з 16 і 256 кольорів.

Першими для архівації зображень стали застосовуватися алгоритми стиснення без втрат. Ті, що використовувалися і використовуються в системах резервного копіювання, при створенні дистрибутивів і т. п. Однак час не стоїть на місці, і основною тенденцією сьогодні стало використання нових класів зображень. Старі алгоритми перестали задовольняти вимогам, висунутим до архівації. Це привело до створення нового типу алгоритмів – стиснення із втратою інформації. Як правило, у них можна задавати коефіцієнт архівації і, отже, степінь втрат якості. При цьому досягається компроміс між розміром і якістю зображень.

Базовим методом цифрового кодування зображень є імпульсно-кодова модуляція (ІКМ). Вона характеризується тим, що кожному закодованому в цифрову форму слову відповідає квантований у часі і по амплітуді відлік відеоінформації [3]. При цьому повинні виконуватися вимоги теореми дискретизації:

$$f_g = 2\omega, \tag{1}$$

де ω – максимальна частота сигналу.

Для запобігання появи фальшивих контурів для одної складової кольору необхідно не менш 50 рівнів квантування, що відповідає 6–8 розрядному слову на кожен елемент кольорової складової зображення. Через великі обсяги інформації ІКМ використовується лише при внутрістудійній передачі телевізійних сигналів паралельним кодом. ІКМ є базовим, канонічним представленням зображення в цифровій формі [3].

Серед статистичних методів найбільш широке застосування знайшли блокові методи кодування зображень. Дані методи працюють у такий спосіб. Блоки розміром $M \times N$ елементів кодуються відповідно до імовірності їхньої появи. Для найбільше ймовірносних блоків використовуються короткі кодові слова, а для менш ймовірностних – довгі кодові слова (алгоритм Хаффмана), у результаті чого досягається стиснення даних [4, 5]. Коефіцієнти стиснення при використанні цих методів можуть досягати 4–5.

Огляд літературних джерел показав, що основними напрямками вирішення проблеми стиснення зображень є використання wavelet-кодування [1–3] та використання формату jpeg для зменшення об'єму фалів зображень [4, 5]. Також запропоновано методику зменшення розміру фалів зображень на основі методів масштабування [6]. Велика кількість робіт присвячена використанню комбінованих методів стиснення зображень [7–10].

Зокрема, робота [1] присвячена огляду і порівнянню відомих методів стиснення зображень. Авторами [2, 3] при порівнянні різних методів кодування також відзначено, що при покращенні якості відновленого зображення зменшується коефіцієнт стиснення. Особливо така проблема зустрічається для зображень з насиченою кольоровою гамою.

При роботі з растровими зображеннями первинною задачею, перед будь-якою обробкою, є визначення інтенсивностей точок. Авторами роботи [4] також запропонований метод побудови поліноміальної функції відновлення, але вихідний файл уже буде містити векторне зображення. Такий метод дає досить хороше співвідношення між коефіцієнтом стиснення та якістю відновленного зображення.

Розглядаючи метод [5], автор зазначив, що отриманий растр менш надлишковий. Перетворення можна застосовувати для зображень з великими площами рівної зафарбованості. Дане кодування зворотне, тобто зображення відновлюється без втрат. Це перетворення знижує ентропію, отже підвищує ступінь стиснення.

Також цікавою є модель стиснення ICA [6], в основі якої лежить розбиття зображення на невеликі області для подальшої оброки з використанням комбінованого методу дельта функції та функції щільності розподілу Гауса.

Особливості wavelet-кодування, методів кодування без втрат полягають у тому, що вони стали основою для подальших досліджень [7–10], але поки що використовуються для двовимірних зображень.

5. Методи досліджень

Інтерполяційні кодуючі системи основані на чисельних методах апроксимації, через які послідовність або двохмірний масив відліків яскравості наближено представляються через неперервні функції. Процедура інтерполяції може бути застосована на етапі перетворення зображення в кодований сигнал (інтерполяція на передаючій стороні) або вона може бути частиною процесу відновлення зображення по кодованому сигналу (інтерполяція на приймальній стороні).

В кодуючих системах з інтерполяцією на передаючій стороні значення яскравості апроксимуються неперервними функціями з раніше встановленою точністю. Інтерполяція може проводитись вздовж стрічки розгортки або охвачувати деяку частину площини зображення.

Інтерполятор нульового порядку працює наступним чином: для всіх елементів зображення встановлений однаковий інтервал допустимих спотворень, в межах якого будується набір відрізків горизонтальних прямих максимальної довжини без додаткових обмежень на розміщення їх початкових та кінцевих точок. Всілякий елемент зображення перекривається яким-небудь із цих відрізків. По каналу зв’язку передаються вертикальна координата кожного відрізка та його довжина, виражена числом елементів. При відновлені зображення на приймальні стороні всі елементи в межах відрізку набувають рівень, відповідний його вертикальній координаті. Цей варіант інтерполяції надає більшу свободу в виборі можливих комбінацій відрізків горизонтальних прямих та в зв’язку з цим дозволяє отримати найбільш ефективне представлення вихідних даних за допомогою мінімальної кількості відрізків. Однак обсяг обчислювальних операцій, необхідних для побудови такого оптимального наближення, часто виявляється дуже великим. Спрощений варіант інтерполяції нульового порядку представляє собою кодування довжин серій з вказуванням яскравості першого елементу серії.

Дія різних інтерполяторів першого порядку полягає в наступному: всякий елемент зображення перекривається будь-яким із прямолінійних відрізків, розміщення яких в межах допустимого інтервалу похибок не пов'язано з додатковими обмеженнями на розміщення початкових та кінцевих точок. Обчислювальна процедура апроксимації може бути в деякій мірі спрощена з'єднанням початку чергового відрізка з кінцем попереднього відрізка. Подальше спрощення полягає в тому, щоб в якості початкових та кінцевих точок використовувати значення яскравості елементів, наближення такого типу часто називають веєрною інтерполяцією.

Поліноміальні функції більш високого порядку, наприклад кубічні сплайни, також можуть бути застосовані для кодування з інтерполяцією, однак підвищення порядку поліномів супроводжується швидким зростанням обсягу обчислень. Можна також ставити задачу двомірної інтерполяції нульового та першого порядків, однак відповідні інтерполятори важко реалізувати на практиці.

Вище були розглянуті різні методи стиснення зображень. Кожен з них спрямований на зменшення обсягу графічного зображення з метою економії дискового простору. Далі розглядатимуться алгоритми обробки зображень з використанням методу сплайнової інтерполяції.

Суть чи ядро програмного продукту, що розробляється – реалізація модуля відновлення пропущених відліків, використовуючи кубічні сплайн-функції. Тому в цьому підрозділі опишемо математичну сторону сплайнової інтерполяції.

Перед тим, як розглянути алгоритми стиснення зображень за допомогою згладжування, з'ясуємо суть самої задачі згладжування.

Вимірювальний прибор відповідає на вхідний сигнал x_n вихідним сигналом, при цьому числовий параметр n змінюється від 1 до N. Через загальну кількість N пар величин необхідно провести функцію $f(x)$ так, щоб можна було задати аналітичний зв'язок між вихідним і вхідним сигналами, тобто характеристику.

При інтерполяції шукана крива $f(x)$ проходить через опорні точки $f(x_n)=y_n$. Знайдена функція дозволяє обчислити вихідний сигнал $f(x)$ також і між опорними точками $x_n \leq x \leq x_{n+1}$. При акпроксимації шукану криву необхідно провести по визначеному закону між парами величин (розсіяних чи зашумлених), не добиваючись того, щоб на опорних точках значення функції співпадали з виміряними значеннями y_n.

Алгоритми інтерполяції функцій по точним даним, визначеним на дискретні множині точок, як правило, основані на використанні інтерполяційних многочленів Лагранжа. При цьому відносно функції $\varphi(x)$, яка інтерполюється, вводиться апріорне припущення про те, що вона має похідні до деякого порядку.

Інша, близька до проблеми інтерполяції, задача виникає в тому випадку, коли значення заданої функції $\varphi(x)$ відомі в вузлових точках не точно, а з деякою похибкою, максимальна величина якої для кожної точки задається в якості апріорної інформації. В цьому випадку задача полягає в побудові такої кривої, яка б найкращим чином апроксимувала функцію, задану з випадковими похибками в вузлових точках. Така задача зазвичай розв’язується на основі методу найменших квадратів.

Останнім часом теорія інтерполяції збагатилася новими методами, що отримали назву сплайнових інтерполяцій. Слід відзначити, що сплайном зазвичай є визначена в області D кусково-поліноміальна функція, тобто функція, для якої існує розбиття D на підобласті таке, що всередині кожного елементу розбиття функція являє собою многочлен деякої степені m. Крім того, ця функція, як правило, неперервна в області D разом з похідними $(m–1)$ порядку. Найбільш розповсюдженими в техніці стали сплайни – многочлени третього порядку.

Нехай на відрізку $[a, b]$ осі x задана сітка $a=x_0<x_1<\ldots<x_n=b$, в вузлах якої задані значення $\left\{ f_k \right\}_{k=0}^{n}$ функції $f(x)$, визначеної на $[a, b]$. Тоді задача кусково кубічної інтерполяції ставиться наступним чином. На відрізку $[a, b]$ необхідно знайти функцію $g(x)$, яка задовольняє умовам:

1) $g(x)$ належить класу $C^{(2)}(a, b)$, тобто неперервна разом зі своїми похідними до другого порядку включно;

2) на кожному з відрізків $[x_{k–1}, x_k]$ $g(x)$ є кубічним многочленом вигляду:

$$g(x)=g_k(x)=\sum_{l=0}^{3} a_l^k (x_k x)^l,\; k=1,\ldots,n; \qquad (2)$$

3) в вузлах сітки $\left\{ x_k \right\}_{k=0}^{n}$ виконуються рівності:

$$g(x_k)=f_k,\ (k=0,1,\ldots,n); \qquad (3)$$

4) $g^{//}(x)$ задовольняє граничним умовам:

$$g^{//}(a)=g^{//}(b)=0. \qquad (4)$$

Переваги обраної інтерполяції стануть зрозумілими пізніше, коли встановимо екстремальну властивість так визначеної функції $g(x)$.

Покажемо, що поставлена задача на знаходження інтерполюючої кусково-кубічної функції $g(x)$ має єдине рішення. Для цього скористаємося сформульованими вище умовами.

Оскільки друга похідна функції $g(x)$ наперервна та лінійна на кожному відрізку сітки $[x_{i–1}, x_i]$ $(i=1,\ldots,n)$, можемо записати при $x_{i–1}\le x\le x_i$:

$$g^{//}(x)=m_{i-1}\frac{x_i-x}{h_i}+m_i\frac{x-x_{i-1}}{h_i}, \qquad (5)$$

де $h_i=x_i-x_{i–1}$, $m_k=g^{//}(x_k)$. Проінтегруємо двічі обидві частини рівності (5). Отримаємо:

$$g(x)=m_{i-1}\frac{(x_i-x)^3}{6h_i}+m_i\frac{(x-x_{i-1})^3}{6h_i}+A_i\frac{x_i-x}{h_i}+B_i\frac{x-x_{i-1}}{h_i}, \qquad (6)$$

де A_i та B_i – деякі константи інтегрування, що обчислюються з умови $g(x_{i-1})=f_{i-1}$, $g(x_i)=f_i$. Підставляючи $x=x_i$ та $x=x_{i-1}$ в (6) отримаємо:

$$\begin{aligned} g(x)&=m_{i-1}\frac{(x_i-x)^3}{6h_i}+m_i\frac{(x-x_{i-1})^3}{6h_i}+\left(f_{i-1}-\frac{m_{i-1}h_i^2}{6}\right)\frac{x_i-x}{h_i}+ \\ &+\left(f_i-\frac{m_ih_i^2}{6}\right)\frac{x-x_{i-1}}{h_i}, \end{aligned} \qquad (7)$$

$$g^{/}(x)=-m_{i-1}\frac{(x_i-x)^2}{2h_i}+m_i\frac{(x-x_{i-1})^2}{2h_i}+\frac{f_i-f_{i-1}}{h_i}-\frac{m_i-m_{i-1}}{6}h_i. \qquad (8)$$

Із (8) знаходимо односторонні границі похідної в точках $x_1, x_2, \ldots, x_{n-1}$, і відповідно до умови неперервності функції $g^{//}(x)$ та $g^{/}(x)$ на $[a, b]$, та умови неперервності $g^{/}(x)$ в точках $x_1, x_2, \ldots, x_{n-1}$ отримаємо $n-1$ рівняння:

$$\frac{h_i}{6}m_{i-1}+\frac{h_i+h_{i+1}}{3}m_i+\frac{h_{i+1}}{6}m_{i+1}=\frac{f_{i+1}-f_i}{h_{i+1}}-\frac{f_i-f_{i-1}}{h_i}. \qquad (9)$$

Доповнюючи ці рівняння з умов (4) рівностями $m_0=m_n=0$, отримуємо лінійну алгебраїчну систему для знаходження невідомих $m_1, m_2, \ldots, m_{n-1}$:

$$Am=Hf. \qquad (10)$$

Квадратна матриця A має вигляд:

$$A=\left\|\begin{matrix} \frac{h_1+h_2}{3} & \frac{h_2}{6} & 0 & \ldots & 0 & 0 \\ \frac{h_2}{6} & \frac{h_2+h_3}{3} & \frac{h_3}{6} & \ldots & 0 & 0 \\ 0 & \frac{h_3}{6} & \frac{h_3+h_4}{3} & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & \frac{h_{n-1}}{6} & \frac{h_n+h_{n-1}}{3} \end{matrix}\right\|, \qquad (11)$$

вектори m та f, і прямокутна матриця H такі:

$$m = \begin{Vmatrix} m_1 \\ m_2 \\ \dots \\ m_{n-1} \end{Vmatrix}, \; f = \begin{Vmatrix} f_1 \\ f_2 \\ \dots \\ f_n \end{Vmatrix},$$

$$H = \begin{Vmatrix} \frac{1}{h_1} & \left(-\frac{1}{h_1}-\frac{1}{h_2}\right) & \frac{1}{h_2} & \dots & 0 & 0 \\ 0 & \frac{1}{h_2} & \left(-\frac{1}{h_2}-\frac{1}{h_3}\right) & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \left(-\frac{1}{h_{n-1}}-\frac{1}{h_n}\right) & \frac{1}{h_n} \end{Vmatrix}.$$

Матриця *A* симетрична зі строгою діагональною перевагою. Коефіцієнти $m_1, m_2, \dots, m_{n-1}$ визначаються з системи (10) однозначно. Отже, сплайн-функція *g(x)* також однозначно відновлюється по формулам (7) і задача про знаходження кусково-кубічної функції *g(x)* має єдине рішення.

Кубічні сплайни володіють дуже важливою властивістю, яка обумовлює високу ефективність сплайн-інтерполяції. А саме, розглянемо на відрізку [*a*, *b*] клас $W_2^2[a,\, b]$, який складається з функцій, які мають додаваємі з квадратом другі похідні. Поставимо задачу пошуку інтерполяційної функції:

$$u \in W_2^2[a,\, b],\; u\left(x_k\right) = f_k,\; k = 0, 1, \dots, n, \tag{12}$$

яка мінімізує функціонал:

$$\Phi(u) = \int_a^b [u^{//}(x)]^2 dx, \tag{13}$$

на класі $W_2^2[a,\, b]$. Стверджується, що мінімум такого функціонала досягається на кусково-кубічній сплайн-функції *g(x)*, яку щойно побудували. Насправді, розглянемо величину:

$$\Phi(u-g) = \int_a^b [u^{//} - g^{//}]^2 dx. \tag{14}$$

Інтегруючи по частинам і використовуючи властивості функцій *g* і $u \in W_2^2$, отримаємо:

$$\Phi(u-g)=\Phi(u)-\Phi(g)-2\sum_{k=1}^{n}\int_{x_{k-1}}^{x_k}[u^{/}-g^{/}]g^{///}dx, \qquad (15)$$

але $g^{///}=c_k=const$ на відрізку $[x_{k-1}, x_k]$, тому:

$$\Phi(u-g)=\Phi(u)-\Phi(g)-2\sum_{k=1}^{n}c_k(u-g)\Big|_{x=x_{k-1}}^{x=x_k}=\Phi(u)-\Phi(g).$$

Звідси та з (14) випливає, що:

$$\Phi(g)=\Phi(u)-\Phi(u-g)\le\Phi(u). \qquad (16)$$

Таким чином на кусково-кубічній функції $g(x)$ реалізується мінімум функціоналу (13). Неважко показати, що інших точок мінімуму у функціоналу немає [7].

Базуючись на (12), (13), можна дати інше, еквівалентне визначення кусково-кубічної сплайн-функції: це така функція із класу $W_2^2[a, b]$, яка приймає в вузлах сітки задане значення та мінімізує функціонал (13).

Кубічні сплайн-функції володіють хорошими апрксимуючими властивостями. Якщо функція f, яка інтерполюється належить класу $C^{(k)}[a, b]$ ($k=0,1,2,3,4$), то для функції похибки $\varphi(x)=f(x)-g(x)$ справедливі нерівності:

$$\max_{a\le x\le b}\left|\varphi^{(p)}(x)\right|\le Ch^{k\text{-}p},\ k\ge p, \qquad (17)$$

де C – невід'ємна константа, яка не залежить від сітки, $h=\max\limits_{1\le i\le n}\left|x_i-x_{i-1}\right|$.

Суть самого метода стиснення полягає в наступному: спочатку беруть перші дві точки з області згладжування. Обчислюються коефіцієнти полінома $P(x)$, і будується згладжуюча крива для обраних точок. Значення коефіцієнтів заноситься в якійсь інший файл. Використовуючи цей самий поліном, будується сгладжуюча крива для наступних двох точок, до тих пір, поки різниця між початковими значеннями функції в цих точках і значеннями, отриманими в результаті інтерполяції, не перевищуватимуть якогось заданого значення Δ:

$$\Delta=(P(x_n)-y_n), \qquad (18)$$

де $P(x_n)$ – інтерполяційні значення; y_n – початкові значення.

Для точок, де значення Δ перевищує задане, знову знаходять значення коефіцієнтів полінома $P(x)$, і повторюють алгоритм спочатку.

Даний алгоритм може бути ефективно застосований для зображень з малою кількістю кольорів (чорно-білих зображень). Оскільки при великій кількості кольорів, а відповідно і зміні інтенсивностей точок, виникає необхідність великої кількості розрахунків для обчислення коефіцієнтів. Це приводить до зниження швидкості обробки зображення, а в деяких випадках навіть до збільшення обсягу файлу, в якому зберігаються коефіцієнти. Для більшої степені стиснення файл коефіцієнтів можна обробити, використовуючи який-небудь метод кодування, або застосувавши будь-який з відомих архіваторів.

Зазвичай растрове зображення представляється масивом точок, значення яких відповідає інтенсивності. Ще один полягає в розбитті матриці на невеликі квадрати, наприклад, 4×4, для яких знаходять середнє значення інтенсивності, яке і передається в інший файл. Таким чином замість 16 точок отримуємо 1. Для відновлення зображення між отриманими точками проводять апроксимуючий або інтерполюючий поліном.

6. Результати досліджень

Основними етапами процесу кодування зображень є:

1. Перетворення зображення в оптимальний кольоровий простір (тільки для кольорових зображень).
2. Субдискретизація компонент кольоровості усередненням груп пікселів (тільки для кольорових зображень), що дозволяє без втрат якості зменшити обсяг інформації в два рази.
3. Задається різницевий коефіцієнт *Δ*, який показує наскільки відрізняється значення поліному в даній точці від початкового значення інтенсивності.
4. Для двох точок зображення будується поліном третьої степені, яким далі згладжуються наступні дві точки, до тих пір, поки різниця між значенням поліному в точці та початковим значенням в ній не відрізнятиметься на задану величину *Δ*.
5. Відновлення пропущених відліків інтерполяційним многочленом (формуються відліки низькочастотної компоненти зображення – $\overline{X(m_1,m_2)}$).
6. Формування відліків різницевої (високочастотної) компоненти зображення та їх квантування:

$$\begin{aligned} &\delta(m_1,m_2) = x(m_1,m_2) - \overline{X(m_1,m_2)}, \\ &\delta_{KB}(m_1,m_2) = \sum_{1}^{n} A_j 1\left\{\delta(m_1,m_2) - \Delta_j\right\}, \end{aligned} \tag{19}$$

де $x(m_1,m_2), \overline{X(m_1,m_2)}$ – відповідно значення відліків початкового зображення та низькочастотної компоненти; A_j – кроки квантування початкового сигналу; Δ_j – пороги квантування вхідного сигналу; $1\{\bullet\}$ – одинична функція.

7. Кодування відліків високочастотної та низькочастотної компоненти (тільки опорні відліки) методами статистичного кодування.

Відновлення – це зворотній процес кодуванню, з тією різницею, що немає необхідності обчислення коефіцієнтів сплайн-функції та відсутнє квантування. Зрозуміло, що схема відновлення простіша і, відповідно, процес відновлення швидший від процесу кодування. Експериментальні дослідження показали, що коефіцієнт прискорення операції відновлення знаходиться в діапазоні 8–15 % від часу кодування даних.

6.1. Перетворення зображення в оптимальний кольоровий простір

Цей алгоритм призначений для перетворення *RGB* сигналу в сигнал *YUV*, де *Y* – яркісний сигнал, *U* і *V* – кольорорізницеві сигнали. Це перетворення дозволяє зменшити надмірність зображення.

Перетворення виконується від точки до точки за наступними формулами:

$$
\begin{aligned}
&Y = 0.3 \cdot R + 0.59 \cdot G + 0.11 \cdot B; \\
&U = -0.15 \cdot R - 0.29 \cdot G + 0.44 \cdot B; \\
&V = 0.62 \cdot R - 0.52 \cdot G - 0.1 \cdot B,
\end{aligned}
\qquad (20)
$$

де *R, G, B* – кольорові сигнали зображення (червоний, зелений і синій відповідно).

6.2. Розробка алгоритму кодування зображень методом кубічних сплайн-функцій

Першим етапом виконання алгоритму кодування є розрахунок коефіцієнтів кубічної сплайн-функції. Для цього беруться дві точки з області даних і для них будується поліном.

На другому етапі наступні дві точки згладжуються вже побудованим поліномом, що повторюється до тих пір, поки різниця між значенням поліному в даній точці та її початковим значенням не відрізнятиметься на задану величину Δ. Якщо в якійсь точці *n* значення різниці ($P(x_n)–f(x_n)$) перевищує значення Δ, то для цієї точки, та наступної сусідньої з нею будуються новий поліном. Значення першої точки та $(n–1)^i$ точки заносяться в допоміжний файл. Побудова поліному проводиться спочатку по строкам, а потім по стовбцям.

Заключним етапом кодування зображення методом сплайнової інтерполяції є квантування, яке виконується по рівномірній шкалі.

6.3. Розробка алгоритму декодування зображень

Вхідним потоком даних для алгоритму декодування будуть значення точок, занесені до файлу, та значення коефіцієнтів кубічної сплайн-функції на відповідних відрізках. Таким чином, алгоритм декодування спрощується, оскільки немає необхідності розраховувати коефіцієнти сплайну.

Перший етап роботи алгоритму декодування – створення відновленого образу, шляхом згладжування заданих точок.

Другий і заключний етап роботи алгоритму декодування – об’єднання відновленого зображення з квантованими відліками та нормалізація вихідного потоку (перевірка на переповнення).

Порівняно з аналогами розробка алгоритму стиснення зображень на основі методів кубічних сплайн-функцій надає можливість ефективного стиснення зображень без комплексного використання запропонованих методів, оскільки він є універсальним і дозволяє виконувати стиснення для зображень будь-якого типу. В запропонованому методі згладжування проводиться по двом точкам, що надає можливість більш точно передавати інтенсивності в точках, також задається коефіцієнт точності декодованого зображення, який визначається як різниця між значенням інтенсивності точки в вихідному файлі та декодованому.

Для порівняння стандарт JPEG обраний не випадково. Це один із найбільш розповсюджених стандартів стиснення графічних зображень. Тому цей стандарт є аналогом, з яким можемо порівнювати отримані результати.

При кодуванні за стандартом JPEG застосовувалася програма ACDSee 32. Отримані результати по шести різним файлам зведемо в таблицю (табл. 1).

Таблиця 1

Зведена таблиця результатів експериментальних досліджень

Ім'я фала	Розмір файлу, байт		
	BMP	JPEG	ICP
128018.bmp	1 087 074	115 140	57 794
Cat.bmp	958 878	75 759	68 470
Cindy04.bmp	387 654	42 764	35 017
Girl.bmp	192 054	15 408	14 387
Lake.bmp	921 654	284 755	101 996
Bull.bmp	663 542	101 281	60 511

За часом роботи конвертер у формат JPEG працював у 2–2,5 рази швидше ніж дана розробка. Однак, у наявності перевага в коефіцієнті стиснення. Причому, на зображеннях, які не мають великої кількості контурів, кодувальник ICP кодував зображення в 1,8–2,1 рази ефективніше, ніж JPEG.

7. SWOT-аналіз результатів досліджень

Strengths. Розроблений метод стиснення зображень дозволяє не тільки скоротити час на обробку файлів растрових зображень, але й отримати кращі співвідношення степені стиснення та якості відновленого зображення. Крім того розробка не потребує особливих економічних затрат, достатньо персонального комп'ютера. Важливою особливістю розробленого алгоритму є можливість користувача самому обирати, якою буде якість відновленого зображення.

Weaknesses. Використання запропонованого методу стиснення буде найбільш ефективним для зображень, в яких інтенсивності сусідніх точок відрізняються не дуже сильно, або послідовності точок однакової інтенсивності досить великі. В іншому ж випадку коефіцієнт стиснення буде не великим. Також такий метод не можна застосувати для 3D-зображень.

Opportunities. В подальшому розвиток запропонованого методу можливий для досліджень стиснення 3D-зображень та відео файлів.

Threats. Для застосування запропонованого методу для 3D-зображень повинна використовуватись апроксимація в просторі.

8. Висновки

1. Аналіз відомих методів кодування зображень показав, що використання кількох підходів одночасно дає кращі результати при стисненні файлів зображень. Перевагами багатьох методів є отримання високої степені

стиснення, але при цьому спостерігається втрата якості відновленого зображення. Комбіноване використання методів не тільки зберігає високу степінь стиснення, але значно покращує і якість відновленого зображення. Запропонований підхід стиснення зображень кубічними сплайн-функціями не тільки не потребує використання додаткових підходів для покращення алгоритму, але й досить простий з математичної точки зору, та з точки зору програмної реалізації.

Розробка математичної та алгоритмічної моделі розв'язку поставленої задачі стиснення зображень на основі методу кубічних сплайн-функцій дала в результаті не складну, з точки зору обчислень, методику. Такий підхід не тільки покращить якість зображення після стиснення, але й не займатиме багато часу на обробку. Важливою перевагою отриманого алгоритму є те, що користувач сам може задавати точність відновленого зображення, залежно від подальшого його використання.

2. Аналіз результатів роботи показав, що відношення об'ємів початкового та стисненого файлів для запропонованої методики складає 0,25–0,5, в той час коли існуючі методи показують результат 0,5–0,7.

Література

1. Lezhnev V. G. Matematicheskie algoritmy szhatiya izobrazheniy. Krasnodar: Kuban. gos. un-t, 2009. 55 p.
2. Obzor algoritmov szhatiya s poteryami. URL: http://mf.grsu.by/UchProc/livak/po/comprsite/theory_fractal.html (Last accessed: 02.12.2017).
3. Metody szhatiya dannykh: Szhatie izobrazheniy. URL: http://www.compression.ru/book/part2/part2__3.htm (Last accessed: 04.12.2017).
4. Jiao L. C., Tan S., Liu F. Ridgelet theory: from ridgelet transform to curvelet // Chinese Journal of Engineering Mathematics. 2005. Vol. 22, No. 5. P. 761–773.
5. Chiang T.-H., Dung L.-R. A VLSI Progressive Coding for Wavelet-based Image Compression // IEEE Transactions on Consumer Electronics. 2007. Vol. 53, No. 2. P. 569–577. doi: http://doi.org/10.1109/tce.2007.381731
6. Velisavljevic V., Beferull-Lozano B., Vetterli M. Space-Frequency Quantization for Image Compression With Directionlets // IEEE Transactions on Image Processing. 2007. Vol. 16, No. 7. P. 1761–1773. doi: http://doi.org/10.1109/tip.2007.899183
7. Iano Y., da Silva F. S., Cruz A. L. M. A fast and efficient hybrid fractal-wavelet image coder // IEEE Transactions on Image Processing. 2006. Vol. 15, No. 1. P. 98–105. doi: http://doi.org/10.1109/tip.2005.860317
8. Utsugi A. Independent components of natural images under variable compression rate // Neurocomputing. 2002. Vol. 49, No. 1–4. P. 175–185. doi: http://doi.org/10.1016/s0925-2312(02)00530-1
9. Remya S., Dilshad Rasheed V. A. Resolution Progressive Compression of Encrypted Images // International Journal of Signal Processing Systems. 2013. Vol. 1, No. 1. P. 7–10. doi: http://doi.org/10.12720/ijsps.1.1.7-10
10. A Hybrid Compression Method for Integral Images Using Discrete Wavelet Transform and Discrete Cosine Transform / Elharar E. et al. // Journal of Display Technology. 2007. Vol. 3, No. 3. P. 321–325. doi: http://doi.org/10.1109/jdt.2007.900915