Київський національний університет імені Тараса Шевченка

Кононов М.В.

Візуалізація інформації

Методичні вказівки до проведення лабораторних робіт

для студентів спеціальності
172 «Телекомунікації та радіотехніка»
факультету радіофізики, електроніки
та комп'ютерних систем

Київ – 2022

УДК 004

Візуалізація інформації. Методичні вказівки до проведення лабораторних робіт для студентів спеціальності 172 «Телекомунікації та радіотехніка» факультету радіофізики електроніки та комп'ютерних систем / Кононов М.В. – Київ: ФРЕКС Київського національного університету імені Тараса Шевченка, 2022. – 36 с.

Рецензенти:

доцент кафедри радіотехніки та радіоелектронних систем Київського національного університету імені Тараса Шевченка, кандидат фізико-математичних наук Кельник О.І.

доцент кафедри медичної радіофізики Київського національного університету імені Тараса Шевченка, кандидат фізико-математичних наук Веремій Ю.П.

Ухвалено Вченою радою
факультету радіофізики електроніки та комп'ютерних систем
Київського національного університету
імені Тараса Шевченка
(протокол № 3 від 16.11.2022)

© Кононов М.В., 2022

Зміст

Вступ 4

1. Використання в системі візуалізації цифрових методів обробки інформації 6

1.1. Особливості сприйняття зображення 6

1.2. Математична модель зображення 8

1.3. Цифрова обробка зображення 9

1.4. Фільтрація зображень 12

1.5. Стиснення даних 14

2. Описи лабораторних робіт 21

Лабораторна робота №1. Дослідження ефективності стиснення на прикладі растрових графічних файлів 21

Лабораторна робота №2. Дослідження точкової обробки зображень на прикладі перетворення координат 27

Лабораторна робота №3. Дослідження локальної обробки зображень на прикладі згорткової фільтрації 31

Література 36

Вступ

Людина постійно спирається на інформацію, яку отримує через зір як «візуальні образи». Саме зір забезпечує найбільшу швидкість надходження інформації до мозку людини. Завдяки цьому можна отримати і зрозуміти велику кількість інформації одночасно. Через це представлення інформації у візуальній формі отримало надзвичайно широке розповсюдження практично в усіх видах людської діяльності.

Під візуалізацією інформації розуміють інженерну галузь, яка забезпечує інтерактивне представлення даних у вигляді зорових образів для забезпечення високої ефективності людського пізнання. З врахуванням сучасних тенденцій, які спираються на цифрову обробку інформації методи і засоби візуалізації тісно пов’язані з комп’ютерними технологіями та алгоритмізацією комп'ютерної графіки.

Загалом візуалізація інформації спирається на досить різні галузі. Основою реалізації засобів відтворення візуальної інформації забезпечує електроніка, яка свого часу спирається на здобутки фізики та хімії. Але з іншого боку значний всклад в технології візуалізації має цифрова обробка інформації. Оскільки електроніка, яка використовується для таких потреб, наразі є цифровою, методи підготовки даних для виведення на екран для подальшого спостереження людиною вимагають застосування складних алгоритмів та їх ефективної реалізації. Наприклад, цифрові технології дозволили суттєво змінити таку галузі, як телебачення, яке до їх впровадження було вінцем реалізації складної телекомунікаційної системи на основі аналогових технологій.

З врахуванням того, що телебачення поєднує візуалізацією інформації з телекомунікаційними технологіями, в рамках спеціальності «телекомунікації та радіотехніка» слушно при розгляді візуалізації максимум уваги приділити саме реалізації такої системи. Аналогові технології до певного моменту вдовольняли потреби телекомунікацій, в тому числі системи телебачення, для якого було навіть було реалізовано досить якісне відтворення кольорового зображення. Але спроба переступити через певні обмеження якості зображення, пов’язані в першу чергу з обмеженістю методів аналогового кодування для потреб передачі на велику відстань через канал обмеженої пропускної здатності, призвели до того, що аналогове телебачення практично зайшло в тупик. Так

наприклад, ідея телебачення високої чіткості без цифрових технологій не змогла бути доведена до широкого використання. Поява достатньо швидких цифрових засобів обробки інформації та цифрових інформаційних мереж суттєво змінили підходи до роботи з зображенням, в тому числі на всіх етапах телебачення як системи, дозволило повністю відмовитись від старих технологій запису відео та кодування для широкомовної передачі.

Цифрові технології роботи з зображенням вимагають зовсім інших методів роботи з інформацією і невідривні від програмної обробки даних. Наразі телевізор є спеціалізованим комп'ютером, а засоби доставки значною мірою відійшли від тих, що використовувались в аналоговому телебаченні. Серед основних тенденцій розвитку треба відмітити зближення технологій передачі даних для різних потреб, в тому числі телебачення, а також уніфікацію методів формування контенту для телебачення, кінематографу, соціальних мереж, тощо.

З врахуванням зазначеного вище, одним з найважливіших практичних питань візуалізації інформації є програмна обробка даних з метою

- узгодження з апаратними засобами,
- представлення даних у вигляді, який забезпечує комфортний перегляд зображення людиною
- забезпечення високої ефективності використання інформаційної ємності засобів збереження інформації та каналів передачі.

Відповідно, для ознайомлення з зазначеними методами обробки даних для потреб візуалізації особливу увагу буде приділено дослідженню стиснення зображення (в тому числі втратними методами), та використанню фільтрації, без якої не можна собі уявити ані фотокамеру, ані систему підготовки або відтворення телевізійного контенту.

Лабораторні роботи запропонованого практикуму виконуються на комп’ютері з використанням безкоштовного графічного редактора PixBuilder Studio та спеціалізованого програмного забезпечення, реалізованого на кафедрі медичної радіофізики Київського національного університету імені Тараса Шевченка.

1. Використання в системі візуалізації цифрових методів обробки інформації

1.1. Особливості сприйняття зображення

Мозок аналізує зорову інформацію на основі сукупності даних, які постачають очі через зоровий нерв – пучок нервових волокон, що передають сигнали від клітин-фоторецепторів. Ці фоточутливі клітини утворюють сітківку – внутрішню оболонку ока.

Клітини сітківки діляться на ковбочки та палички. Палички завдяки специфічному пігменту родопсину забезпечують сприйняття слабкого світлового сигналу, але не забезпечують колірний зір, за який відповідають менш чутливі ковбочки. Фоточутливість ковбочок забезпечується іншим пігментом групи йодопсину. Сприйняття кольору забезпечується тим, що є три типи ковбочок. Ковбочки кожного з них мають окремий вид йодопсину, максимум чутливості якого лежить відповідно в діапазонах 420–440 нм, 534–545 нм та 564–580 нм. Нормалізовані криві чутливості наведені на рис.1.1.

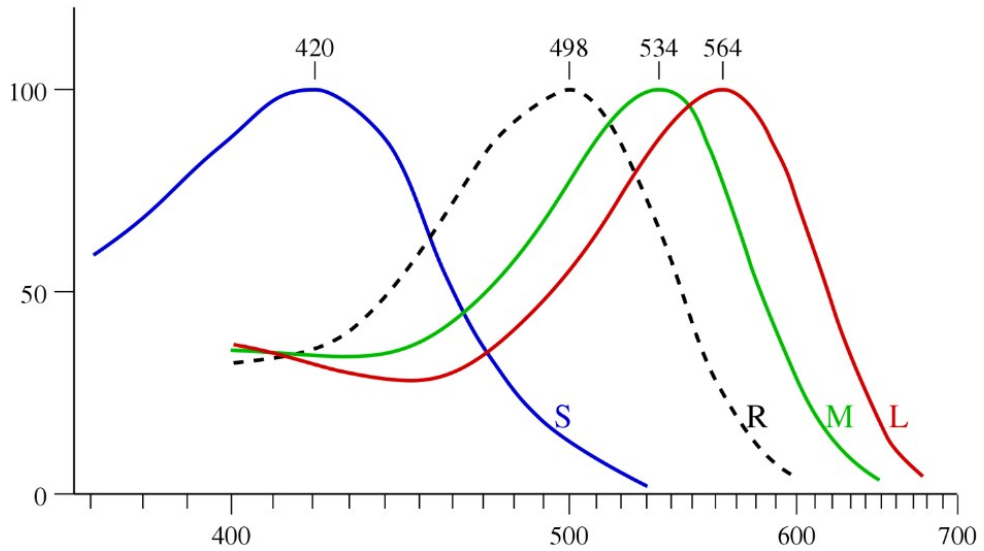

Рис. 1.1. Нормалізовані криві чутливості йодопсину трьох видів (S,M,L) та родопсину R).

Таким чином очі не забезпечують ту повноту інформації, яку можна отримати від спектрометра. Але навіть таке, досить обмежене, сприйняття спектральних складових окремих фрагментів зображення значно збільшує кількість інформації,

яку сприймає мозок. З врахуванням зазначеної особливості зорового апарату для роботи з кольором як в апаратних засобах, так і в методах обробки даних (як цифрових, так і аналогових) обмежуються трьома спектральними складовими, які можна вважати базисними кольорами.

Для забезпечення високої якості системи або пристрою візуалізації слід враховувати ще деякі особливості зорового апарату людини. Найбільш важливими параметрами зображення, що відтворюється, є яскравість та контрастність. Якщо яскравість зазвичай забезпечується апаратно, то контрастність спирається також на підготовку даних для пристрою відтворення.

Контрастність визначається відношенням яскравості найсвітлішої та найтемнішої частин зображення. З врахуванням того, що в даному випадку мова йде про дані, які будуть візуалізовані, яскравістю точки зображення є деяке числове значення. Але слід пам'ятати про те, що динамічний діапазон цих значень повинен враховувати динамічний діапазон сприйнятті світла сітківкою. Загалом око може працювати з світловими потоками в дуже широкому діапазоні – приблизно 10–12 порядків. Але значна частина цього діапазону забезпечується механізмом повільної адаптації (за рахунок зміни кількості родопсину та йодопсину у фоторецепторах протягом певного часу) та швидкої адаптації (звуження або розширення зіниці). Тому для засобів візуалізації найбільш важливим є миттєвий динамічний діапазон ока, який відповідає фіксованим налаштуванням систем адаптації. Він охоплює приблизно два порядки. Ця особливість ока, як і врахування особливостей сприйняття кольору, суттєво послаблює вимоги до реалізації засобів відтворення зображень і, що особливо важливо, дозволяє суттєво зменшити кількість даних для збереження та передачі зображення. Зазвичай для запису даних одного колірного каналу обмежуються одним байтом (8 бінарних розрядів).

При розробці системи візуалізації або її пристроїв необхідно забезпечити людині комфортність роботи безпосередньо з зображенням. Однією з проблем засобів відтворення зображення є високий рівень шумів та поява артефактів – штучних елементів на зображенні, які виникають через неідеальність методів обробки даних та технологій отримання та відтворення візуальної інформації. Проблема шумів в більшості випадків пов'язана з аналоговою обробкою

сигналів, оскільки будь-який аналоговий обробник погіршує співвідношення сигнал/шум, а передача аналогового сигналу на велику відстань супроводжується затуханням та зовнішніми завадами. Ці проблеми можна вирішити переходом до цифрових технологій, але й вони можуть формувати характерні дефекти, наприклад через використання втратних алгоритмів обробки даних.

1.2. Математична модель зображення

Загалом зображення є розподілом яскравості або кольору на деякій поверхні, зазвичай площині. З метою обмеження кількості даних розглядають обмежену, зазвичай прямокутну її ділянку.

Для певних задач величиною, що характеризує точку зображення, достатньо вважати тільки яскравість, нехтуючи кольором. Таке зображення називають монохромним, або «в градаціях сірого». Воно може бути описане неперервною скалярною функцією

$$I = F(x, y)$$

на обмеженій області визначення, наприклад

$$0 \le x \le W;\ 0 \le y \le H\ .$$

Область значень зручно також обмежити нормуванням до одиниці

$$0 \le I \le 1;$$

Для кольорового зображення кожна його точка може бути описана трьома значеннями – складовими кольору, тобто

$$I_n = F_n(x, y)\ ,$$

Цю векторну функцію можна представити трьома скалярними, аналогічними до монохромного випадку – колірними шарами. В цьому випадку значення кольорів для точки зображення з деякими координатами відповідають значенням на окремих шарах з цими ж координатами.

Оскільки систему запису або передачі даних вважаємо цифровою, необхідно нескінчену кількість даних неперервної функції звести до обмеженої (зліченої). Таку обмеженість

кількості даних забезпечує операція дискретизації по координатах, за рахунок чого неперервна функція від двох координат замінюється двовимірним масивом пікселів:

$$I_n = F_n(x_i, y_j) .$$

1.3. Цифрова обробка зображення

Безпосередньо в процесі введення зображення в систему візуалізації наявність його дефектів зменшує кількість отриманої інформації. Тому вже при формування оптичного зображення, яке в подальшому оцифровується, прагнуть забезпечити прийнятну роздільну здатність, широкий динамічний діапазон, достатньо низький рівень аберацій. При цьому слід пам'ятати, що кількість інформації не можна збільшити її обробкою – навпаки неідеальність обробника її зменшує. Особливо проблемними в цьому плані є аналогові методи але часткова втрата інформації може мати місце і при цифровій обробці. Перевагою цифрових методів є те, що втратність є контрольованою і накопичення втрат при довгих ланцюжках послідовної обробки є значно повільнішим. Крім того є можливість використання складніших та неспрощених (більш правильних математично) методів обробки даних.

Оскільки система візуалізації орієнтується на сприйняття інформації людиною через зоровий образ, вже введена до системи графічна інформація обробляється з врахуванням фізіологічних та психологічних особливостей. Покращенню сприйняття зображення людиною сприяє:

- ресемплінг (зміна кроку дискретизації) при потребі геометричного масштабування;
- корекція яскравості та контрастності;
- виправлення кольоропередачі;
- збільшення візуальної чіткості;
- зменшення рівня шумів;
- виправлення геометрії зображення.

Ресемплінг необхідний через те, що для дискретизованого зображення, що є обов'язковим для цифрових пристроїв, збільшення кутового розміру спостереження при незмінній кількості пікселів збільшує сам піксель. За рахунок цього зображення сприймається мозаїчним. Перерахунок зображення зі зменшенням кроку дискретизації дозволяє зменшити таку мозаїчність, тобто покращує візуальну якість.

Щодо корекції яскравості та контрастності, повнота використання динамічного діапазону ока дозволяє легше виділяти окремі деталі на зображенні. А більш правильна кольоропередача при аналізі інформації мозком забезпечує кращу відповідність до раніше отриманого досвіду.

Більш цікавим питанням є збільшення чіткості без збільшення роздільної здатності, як міри розділення дрібних фрагментів зображення. Така технологія є широковживаною в сучасній цифровій фотографії. Розмита границя об'єкту на зображенні, тобто малий градієнт яскравості впоперек границі, не заважає зоровому апарату відокремити його від іншої частини зображення. Але деяке підвищення цього градієнту хоч і не привносить додаткової інформації, більше того є втратною операцією, але імітує збільшення роздільної здатності, чим психологічно покращує сприйняття.

Іншим важливим питанням обробки зображення є вирішення проблеми його зашумленості. Необхідність відокремити корисну складову сигналу від шуму «відволікає» обробник, тобто мозок, від основної задачі – інтерпретації отриманої інформації. Тому прагнуть співвідношення сигнал/шум зробити якомога більшим, навіть за рахунок часткової втрати інформації. Наприклад при придушенні шумів на зображенні можуть губитись дрібні деталі.

Викривлення пропорцій (перспективні спотворення), або границь об'єкту (дисторсія) не настільки сильно заважають сприйняттю зображення, як раніше розглянуті фактори, але і ці дефекти за можливістю стараються виправити, для чого крім традиційного ускладнення конструкції оптики та камери може використовуватись розрахункова постобробка даних, тобто цифрові методи роботи з зображенням.

Таким чином для вже введеного в систему зображення можна підготовити його для більш якісної візуалізації використанням обробки за певними методами, зазвичай з цифровою реалізацією.

Під цифровою (комп'ютерною) обробкою сигналу (зображення) можна розуміти будь який розрахунок вихідного сигналу (зображення), характеристики кожного відліку якого є деякою функцією відповідних характеристик відліків вхідного сигналу (зображення) а також одержання деяких числових величин, які є такою функцією точок вхідного сигналу (зображення).

Точкова обробка є найпростішим варіантом, коли для розрахунку даних точки вихідного (результуючого) зображення $D(x_2, y_{2)})$ необхідні тільки дані однієї точки вхідного $D(x_1, y_{1)})$:

$$D(x_2, y_{2)}) = F\big(D(x_1, y_1)\big)$$

Використання терміну «точка», замість «піксель» пов'язано з тим, що алгоритми перетворення та формули, які використовуються при їх реалізації відповідають недискретизованому зображенню.

Найпростішим є випадок, коли координати точки вихідного зображення співпадають з координатами точки вхідного, тобто для дискретного зображення індекси вихідного пікселя співпадають з індексами вхідного. Отже при переході до дискретного випадку формули, розроблені для неперервного зображення не змінюються. Дещо складнішим є випадок, коли точка вхідного зображення однозначно відповідає точці вихідного, але не співпадає за координатами. Таке перетворення також описується формулами для координат однієї точки і таку обробку зображення можна вважати точковою. Але для дискретизованого зображення положення вхідного пікселя не має точного співпадіння з положенням вихідного. Відповідно в цьому випадку для використання «точкових» методів доводиться залучати інтерполяцію, яка все ж таки вимагає врахування значень сусідніх пікселів. Але сам математичний апарат для виконання певної задачі, наприклад повороту, збігається з використовуваним для неперервних функцій, отже і таку обробку можна вважати точковою.

Локальною обробкою є така, коли для розрахунку даних точки вихідного зображення необхідно взяти до уваги дані усіх точок з деякого околу заданої точки вхідного зображення. Таким чином на відміну від точкової обробки треба виконати інтегрування (для дискретного випадку – підсумовування) по околу.

Глобальна (нелокальна) обробка зображення відповідає використанню таких методів, коли для розрахунку однієї точки вихідного зображення обробляються дані усього вхідного зображення. Окіл при використанні локальної обробки зазвичай значно менший за усе зображення. Тобто глобальні, методи є

суттєво повільнішими і замість них за можливості повинні бути використані локальні.

1.4. Фільтрація зображень

Недоліки отриманого сигналу (в нашому випадку зображення, яке є просторовим сигналом) вимагають використання обробки, яка змінює певні характеристики сигналу, що отримало назву «фільтрація». Частіше за все використовується лінійна фільтрація, яка математично описується застосуванням деякого лінійного оператора. При використанні лінійної системи сигнал може бути представлений сумою інших сигналів, кожен з яких в подальшому обробляється окремо, а вихідний сигнал є сумою результатів обробки цих складових.

Періодичний часовий сигнал може бути представлений сумою синусоїд кратних частот (гармонік), рис.1.2. Така сума отримала назву ряду Фур'є. Для неперіодичного сигналу розклад виконується в інтеграл Фур'є – по усіх можливих частотах.

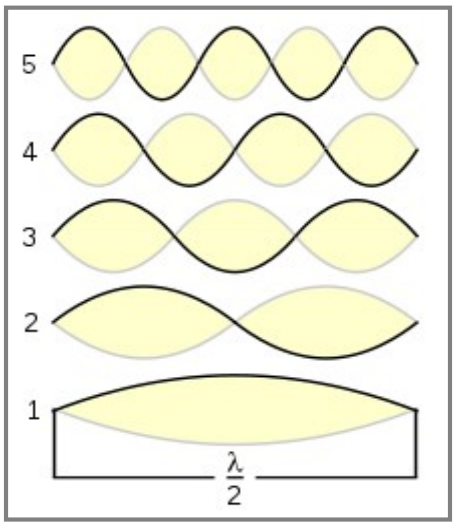

Рис. 1.2. Гармоніки, по яких розкладається періодичний сигнал

У радіотехніці фільтрація використовується для виділення корисного сигналу на фоні шуму. Така можливість пов'язана з тим, що зазвичай корисні сигнали мають деяку нерівномірність розподілу за частотами. Тобто можна придушити ті частотні складові (верхні, нижні або деяку смугу), які в більшій мірі є

шумовими. Для цього може бути використаний зазначений вище розклад по частотах.

Зображення може бути також розкладене по гармоніках, але в даному випадку гармоніки, які мають назву просторових, визначаються функцією не однієї координати (часу) а двох просторових координат, тобто є своєрідними зображеннями, рис. 1.3. На відміну від звичного для радіотехнічного застосування розкладу по гармоніках, коли гармоніка характеризується одним числом – частотою, просторова гармоніка визначається парою чисел – просторовою частотою.

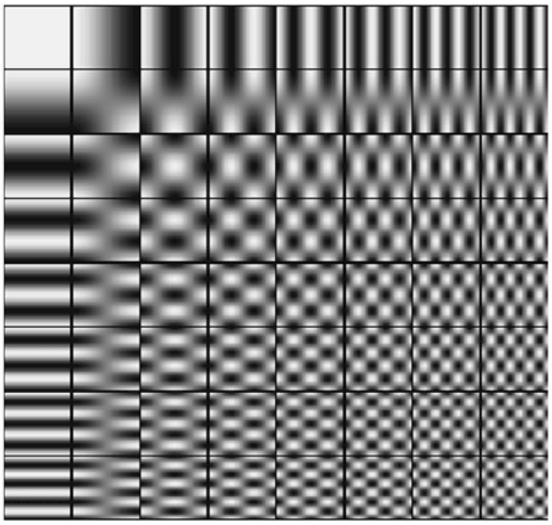

Рис. 1.3. Просторові гармоніки, по яких розкладається зображення

Потреби фільтрації зображень охоплюють:

- придушення шумів (фільтр низьких частот);
- виділення країв (фільтр високих частот);
- підкреслення країв (комбінація оригіналу з дією фільтра високих частот).

Фільтрація просторових частот є методом глобальної обробки, отже є повільною і використовується доволі рідко. Замість цього зазвичай використовують локальний метод – згорткову фільтрацію, яка за результатами близька до частотної. Така заміна методу обробки спирається на теорему про згортку – фур'є-образ згортки функцій дорівнює добутку їх фур'є-образів. Тобто для реалізації загорткової фільтрації є достатнім розрахувати обернене перетворення Фур'є для

функції частотного фільтра. Але при цьому ресурсоємність фільтрації не зменшиться. Ситуацію виправляє те, що для більшості задач фільтрації, апертуру ядра згортки, еквівалентного застосуванню частотної фільтрації, можна суттєво обмежити, а значення функції фільтру (формально – ненульові) поза апертурою замінити нулями. Зрозуміло, що цей наближений фільтр дещо погіршить результат його застосування, але такий підхід є розумним компромісом між якістю обробки зображення та швидкістю розрахунків.

1.5. Стиснення даних

Довільна інформація представляється для збереження, накопичення та обробки у формі кодування деяким потоком даних, тобто числових значень. При кодування певного обсягу інформації може бути отримане повідомлення різної довжини (в байтах), оскільки можуть різнитись самі способи кодування. Для певних потреб, а саме для накопичення та передачі інформації є критичним, наскільки компактним буде результат кодування. З врахуванням великого обсягу інформації, що міститься в зображенні, особливо динамічному (відео), оптимізація методів кодування з метою зменшення кількості даних є однією з базових задач для реалізації системи візуалізації, наприклад сучасного телебачення. Відповідно застосовується стиснення – процедура перекодування даних, яка дозволяє зменшити їх обсяг при умові збереження можливості подальшого відновлення (зворотного перетворення) даних.

Методи стиснення можна поділити на безвтратні та втратні. Використання безвтратного методу дозволяє після відновлення стиснених даних отримати блок даних, який точно відповідає первинним даним, що стискались. Втратні методи, як слідує з назви, призводять до деякої відмінності відновлених даних від початкових. З одного боку це є певним недоліком, але такі методи дозволяють досягти значно більшої ефективності стиснення, а це є критичним для візуальної інформації. Зрозуміло, що як самі методи втратного стиснення, так і їх налаштування для конкретного випадку вимагає реалізації прийнятного рівня втратності. Для збереження та передачі візуальної інформації зазвичай використовуються певні формати даних, які в більшості випадків спираються на використання одного або декількох методів стиснення, безвтратних або втратних.

Групове стиснення. Найбільш простим для розуміння принципів реалізації безвтратного стиснення є алгоритм групового стиснення. Цей метод є прийнятний тільки для стиснення досить вузького класу зображень, наприклад креслень або у випадку заповнення певних ділянок однаковими значеннями пікселів. Тобто для випадків, коли у потоці даних є послідовності однакових байтів. Групове стиснення крім цього використовується як один з етапів високоефективних втратних методів. Найчастіше серед можливих реалізацій групового стиснення використовуються дві:

- PackBits (розробка Apple);
- RLE (використовується у стандартизованих графічних форматах для стиснення індексованих зображень).

Індексованим є зображення, для якого колір пікселя кодується номером рядка деякої таблиці кольорів, яка має назву «палітра». Кожен рядок палітри вміщує значення RGB тріади. При цьому палітра формується за найбільш часто застосованими у зображенні значеннями RGB-тріад, а кольори, які не потрапили у палітру, замінюються на найближчі з наявних в ній.

В методах групового стиснення кодувальник повинен закодувати послідовність N однакових байтів так, щоб забезпечити максимально короткий запис цієї послідовності. Розглянемо алгоритм RLE.

Нехай вхідна послідовність має вигляд

aaaabcccccccaaaccccccccddddaaaaa.

Послідовність однакових байтів запишемо як кількість повторень та значення байта після нього. Таким чином, уся послідовність прийме вигляд:

4a1b7c3a7c4d5a.

Декодувальник розглядає пару байтів також як кількість повторень та значення, що дозволяє виконати коректне відновлення.

Але одночасно з цим необхідно передбачити можливість запису без перекодування послідовності байтів, у якій немає повторень (має назву «літеральна група»). Інакше буде суттєве «розбухання» неповторюваних байтів, бо для кожного такого байту буде додаватись байт кількості зі значенням 1. Для

кодування літеральної групи (в наступному зразку показана жирним шрифтом)

aaaabccc**abdcadad**ccccaaaccccccddddaaaaa,

на початку коду додаються два спеціальних байти. Перший є нульовим, він є ознакою початку літеральної групи, другий – її довжина. Відповідно, наведений приклад потоку, що стискається, кодується таким чином:

4a1b7c**08abdcadad**3a7c4d5a.

Статистичне стиснення. Значно більш універсальними є статистичні алгоритми, назва яких говорить про те, що використовується деяка статистика значень байтів потоку, що стискається. Як приклад розглянемо метод Хаффмана. В даному алгоритмі використовується нерівномірність статистичного розподілу значень байтів, тобто те, що одні байти зустрічаються частіше, інші рідше.

Для запису стиснених даних використовуються коди, не вирівняні на довжину байту, частина з яких є коротшою за байт. Оскільки більш коротких кодів буде менше, ніж 256 можливих значень байту вхідного потоку, для певної частини значень доведеться використовувати коди, довші за 8 бінарних розрядів. Таким чином частина потоку запишеться більш короткими фрагментами бітового потоку (стиснення даних), а частина більш довгими (розбухання даних). При умові, що короткі коди будуть відповідати значенням, які зустрічаються частіше, стиснення перебільшує розбухання, що ї є умовою досягнення загального стиснення.

Для реалізації алгоритму треба на основі частотного аналізу вхідних даних побудувати бітову таблицю кодування. Ця таблиця приєднується до стисненого потоку і використовується при відновленні даних. Таблиця кодування створюється на основі побудови асиметричного бінарного дерева (рис. 1.4). Гілки вліво та вправо відповідають двом значенням біта. Бітовим кодом є значення, яке відповідає завершенню одного з шляхів від кореня дерева до тупикового положення, наприклад на рис. 1.4. код 001 відповідає значенню байта з нестисненого потоку «c».

Тобто декодер починає шлях від кореня відповідно бітів стисненого потоку і, таким чином, виходить на одне з потрібних для формування відновленого потоку байтів значення, після чого повертається до кореня дерева.

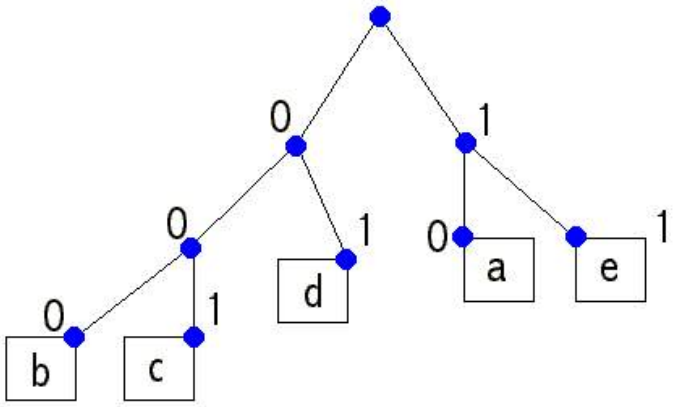

Рис. 1.4. Приклад дерева для кодування за алгоритмом Хаффмана

Для побудови цього дерева кодер реалізує ітераційний процес. У рамках кожної ітерації вибираються два значення, ймовірність яких є найменшою при поточному стані процесу. Цим значенням приписуються відповідно «0» та «1» як бінарний розряд їх коду (один крок шляху на дереві Хаффмана). Для переходу до наступної ітерації ця пара об'єднується в єдиний елемент, а ймовірність пари розраховується як сума двох ймовірностей. Оскільки елементи з найбільшою ймовірністю включаються у розгляд на останніх ітераціях, їм відповідають коротші коди.

Словникові алгоритми спираються на кодування на рівні слова (послідовності байтів) замість окремого байта. Зауважимо, що в даному випадку слово є просто деякою послідовністю байтів, границю якої визначає сам алгоритм. Тобто словниковий алгоритм може бути застосований не тільки для стиснення тексту, а для довільного байтового потоку, в тому числі зображення.

Для пояснення ідеї такого методу стиснення розглянемо кодування потоку, що складається зі слів деякої мови. Можна створити словник цієї мови, тобто побудувати відповідність усіх можливих слів до їх номерів. Ці номери і будуть кодами. У мовах спілкування досить обмежена кількість слів (близько 10^5). Навіть з врахуванням специфічної форми узгодження закінченнями в деяких мовах, наприклад українській, отримаємо 10^6, тобто для такого кодування вистачить трьох байтів на одне

слово. А середня довжина слова (залежить від мови) сягає 5-7 літер. Таким чином отримаємо стиснення приблизно вдвічі. Зрозуміло, що наведений приклад є виключно ілюстративним, оскільки використовує словник певної мови. Але є алгоритми, які дозволяють будувати словник безпосередньо під час кодування та декодування на основі оброблюваного потоку. Це і є словникові методи стиснення, наприклад доволі популярний LZW (Lempel–Ziv–Welch), який був опублікований Велчем у 1984 році на основі розвитку алгоритму LZ78 (Лемпел і Зів, 1978 рік).

Алгоритми втратного стиснення. Для мультимедійної інформації безвтратні методи стиснення є недостатньо ефективними. Відповідно, заради компактності даних доводиться жертвувати частковою втратою інформації. Але якщо врахувати особливості її сприйняття людиною, такі втрати можна зробити майже непомітними.

Наразі найбільш популярним алгоритмом втратного стиснення статичних зображень фотографічної якості, для яких безвтратне стиснення є найменш ефективним, є JPEG. Він підтримує повноцінні кольорові (RGB) та монохромні зображення (у градаціях сірого).

Для RGB зображення першим кроком алгоритму є перехід до інших колірних координат $RGB \rightarrow YC_BC_R$ на основі лінійної комбінації колірних компонент вхідного зображення з попередньо заданими ваговими коефіцієнтами перетворення:

$$\mathrm{Y} = 0.299\,\mathrm{R} + 0.587\,\mathrm{G} + 0.114\,\mathrm{B}$$

$$\mathrm{C_B} = 128 - 0.1687\,\mathrm{R} - 0.3313\,\mathrm{G} + 0.5\,\mathrm{B}$$

$$\mathrm{C_R} = 128 + 0.5\,\mathrm{R} - 0.4187\,\mathrm{G} + 0.0813\,\mathrm{B}$$

За вимоги високої ефективності стиснення на даному етапі може бути використане проріджения даних, тобто кількість пікселів у рядку та кількість самих рядків різницевих шарів C_B та C_R, які відповідають саме за кольоровий контраст, зменшуються вдвічі. Далі кожен з отриманих в результаті зазначеного перетворення шарів даних обробляється незалежно. Для монохромних зображень алгоритм починається з цього моменту, оскільки шар Y фактично відповідає такому зображенню.

Наступний етап обробки спирається на те, що дрібні елементи зображення є менш критичними для сприйняття. Для цього виконується представлення окремих фрагментів усього

зображення (квадратних блоків 8x8 пікселів) розкладом по просторових гармоніках. В результаті блок субпікселів (одноколірних складових цілого пікселя) з 64 скалярних значень утворює нові 64 скалярних значення, які є амплітудами гармонік у вказаному розкладі. Це перетворення виконується зі збільшенням розрядності з 8 бітів до 12. Кількість даних при цьому навіть збільшується, але цей крок перерозподіляє інформацію на більш корисну, яку необхідно записувати максимально докладно, і таку, яку можна частково відкинути – амплітуди гармонік з меншими значеннями індексів зберігаються з більшої кількості бітів, ніж ті, що мають більші індекси. Операцію відкидання частини розрядів називають квантуванням. При виконанні квантування береться до уваги основний параметр настроювання алгоритму, який визначає якість стисненого зображення. При меншому значенні цього параметра зі зростанням індексів просторової гармоніки швидше, ніж при більшому значенні, зменшується кількість бітів, що зберігаються при квантуванні. Саме квантування є обов'язковим джерелом втратності даного алгоритму. А розглянуте вище проріджування, є лише опцією і, зазвичай, вмикається при умові, що параметр якості є меншим за деяке порогове значення.

Квантування визначає специфічні артефакти стисненого за допомогою даного алгоритму зображення, а саме появу так званого "ефекту Гіббса", коли навколо елементів зображення з різким переходом кольору або контрастності утворюється ореол. Крім того, при дуже малому значенні параметра якості на зображенні може проявитись характерна мозаїка з клітинок-блоків (8x8).

Далі виконуються зміна порядку проходження растра таким чином, що поруч розміщуються амплітуди гармонік приблизно однакової значимості, і застосовуються безвтратні методи стиснення Усе це разом забезпечує дуже високу ефективність стиснення для більшості зображень навіть при непомітних на око втратах. Алгоритм реалізує найкраще співвідношення якість/розмір саме для фотографій, але на синтезованих зображеннях, особливо таких, що мають чіткі контрастні лінії, можуть з'являтись небажані артефакти, особливо коли зменшення параметра якості включає механізм прорідження даних.

Відеопотік є послідовністю кадрів. Найпростішим способом стиснення є кодування кожного кадру незалежним чином, тобто

з врахуванням тільки просторової корельованості в межах кадру. Але ефективність стиснення при цьому буде далекою від максимально досяжної, оскільки не буде врахована часова корельованість, тобто схожість сусідніх кадрів між собою. Найпростішим способом врахування такої схожості є віднімання даних попереднього кадру. Такий метод стиснення є досить простим, але й він не є достатньо ефективним. Основою врахування часової корельованості в сучасних алгоритмах є розбиття потоку на декілька видів кадрів:

- I-кадр (Intraframe);
- P-кадр (Predictive);
- B-кадр (Bi-directional).

I-кадр є повністю незалежним від інших зображенням, кодується схожим до JPEG чином. Завдяки цьому він може бути точкою входу у відтворюваний відеопотік.

P-кадр включає у себе кодування змін, що відбулися з часу деякого попереднього (не обов'язково сусіднього) I-кадру, або навіть P-кадру. Зрозуміло, що цей кадр буде компактнішим (приблизно вдвічі) за Intraframe. Гіпотетично можна було б реалізувати весь потік зі стартового I-кадру і подальшої послідовності P-кадрів. Але така структура потоку не використовується, в основному через накопичення похибки при ланцюжковому обчисленні. До того ж деяким додатковим ускладненням алгоритму можна досягти ще більшої ефективності стиснення.

Таке додаткове збільшення ефективності забезпечує B-кадр. Для його розрахунку треба мати як попередній, так і наступний кадри. Отже використовується не одне, а два посилання, що і дало назву «Bi-directional». У кадрі цього типу кодуються відмінності не від попереднього, а від інтерпольованого за двома кадрами, що і забезпечує більшу компактність в порівнянні з P-кадром (приблизно у два рази). Відповідно для стиснення потоку кадрів, є сенс використовувати послідовність типу:

$$I_0 - P_1 - B_2 - B_3 - P_4 - B_5 - B_6 - P_7 ...$$

Таким чином, P_1 посилається на I_0 ; P_4 на P_1 ; B_2 та B_3 на пару (P_1, P_4) і далі за ланцюжком аналогічним чином. Через декілька кадрів знову в потік включається наступний I-кадр.

2. Описи лабораторних робіт

Лабораторна робота №1. Дослідження ефективності стиснення на прикладі растрових графічних файлів

Мета роботи: Порівняти основні характеристики базових методів стиснення, які застосовуються у форматах найбільш популярних растрових графічних файлів.

Загальні відомості

В сучасній комп'ютерній графіці використовуються два основних типи – растрова і векторна графіка. Векторна графіка полягає в записі основних характеристик окремих елементів зображення (точок, відрізків, прямокутників, кіл та іншого). Цей метод найбільш підходить для так званих штрихових малюнків і застосовується у системах розробки схем і креслень. Растрова або матрична графіка полягає у відображенні кожного пікселя (точки) зображення при його скануванні. Для напівтонових зображень, які використовуються значно частіше (фотографія, телебачення, наукові та інженерні потреби, медична діагностика, тощо), більш коректним є застосування растрової графіки. Відповідно, більшість форматів зображень, які не є результатом комп'ютерного синтезу, є растровими.

Вибір типу графіки в основному стосується способу запису зображення у файл, в той час як для відображення майже завжди виконується растеризация, тобто перехід до растру. Це обумовлено тим, що найбільш популярні пристрої відображення (монітор, принтер) мають растровий принцип формування зображення.

Головній недолік растрової графіки полягає в дуже великому розмірі файлу. При цьому спостерігається наявність великої кількості пікселів, що мають однакове значення. Таким чином, для збереження на зовнішньому носії або для передачі через інформаційну мережу, особливо при застосуванні не дуже швидких каналів, потрібно застосовувати стискання графічної інформації (п.1.5).

Загалом для стиснення довільної інформації використовуються архіватори – спеціальні програми, які

обробляють потік даних за допомогою певного безвтратного алгоритму, зазвичай на основі використання словника.

Оскільки для графічної інформації стиснення є надзвичайно актуальним, розроблена велика кількість графічних форматів, які відрізняються алгоритмом стискання, в тому числі з втратами або без втрат. Кожен формат має свої переваги і недоліки, і в різних випадках найбільш зручними є ті чи інші з цих форматів.

В графічному файлі можна виділити заголовок, в якому зібрано всіляку службову інформацію, що стосується характеристик зображення, і інформаційну частину, яка по суті є записом тіла зображення. Найбільш простою є інформаційна частина формату BMP без стиснення – це фактично дамп пам'яті, тобто запис у файл копії ділянки пам'яті комп'ютера, у якій зберігається підготовлене до відтворення растрове зображення. Саме такий формат відображає реальній розмір зображення і його зручно використовувати для порівняння ефективності стиснення з іншими форматами.

В форматі PCX використовується групове стиснення, тобто запис кількості пікселів, що повторюються, з характеристикою самого пікселя. Цей метод добре працює при зображеннях, що мають великі рівнозабарвлені області (наприклад такі, як креслення, кадри традиційної мальованої мультиплікації), але погано стискає напівтонові зображення (наприклад, фотографії).

Формат TIF як основний метод використовує стиснення з словником, схоже до дії розповсюджених архіваторів.

Усі перераховані формати використовують безвтратне стиснення. Ще один цікавий формат GIF використовує також безвтратне стиснення, але підтримує тільки індексовані зображення з палітрою до 256 кольорів, а сама операція переходу до індексованого зображення має доволі великі втрати.

Найбільш ефективним для запису зображень є використання втратного стиснення, яке ми розглянемо на прикладі формату JPEG. При цьому без особливої втрати якості можна відкинути найменш суттєві елементи зображення. У цьому методі фактично вноситься додаткова похибка у менш суттєві просторові частоти при виконанні процедури перенормування. Слід відзначити, що при стисканні фотографічних зображень за таким алгоритмом при великій або середній якості, що відповідає не дуже великому загрубленню перенормування, іноді спостерігається навіть покращення

візуальної якості. Це обумовлено тим, що відкинуті частоти саме відповідають шумовій компоненті зображення. Зазвичай, шум рівномірно розподілений за просторовими частотами, чого не можна сказати про інформативну складову зображення.

Завдання

1. На прикладі декількох різних зображень кожного типу перевірити, який з методів стиснення (групове, словникове, JPEG) є більш ефективним. Типи зображень, які використовуються:
   - Напівтонове кольорове RGB зображення.
   - Напівтонове зображення у градаціях сірого.
   - “Штрихове” (“stroke”) зображення (на кшталт креслення).

2. Дослідити залежність втратності від рівня стиснення при використанні алгоритму JPEG.

3. Дослідити вплив рівня шумів на ефективність роботи алгоритму JPEG.

При виконанні роботи використовуються програмні засоби:

1. ***PixBuilder Studio*** (безкоштовний графічний редактор, надається),
2. ***PicCalc*** (спеціалізована програма, надається для виконання роботи***)***
3. ***AddNoise*** (спеціалізована програма, надається для виконання роботи),
4. ***MS Excel*** (або аналог) – для виконання розрахунків.

Звіт виконується за допомогою MS Word (або аналогу). У звіт вноситься назва роботи, виконавець, дата виконання, результати роботи у вигляді таблиць і графіків (імпортуються з MS Excel). У звіт також вноситься опис інтерпретації отриманого.

Хід виконання

1. Порівняння ефективності різних методів стиснення, реалізованих у певних форматах файлів.

1.1 Завантажити у графічний редактор ***PixBuilder*** тестове кольорове зображення (з наданих до роботи) у форматі TIFF (цей формат використовує словниковий алгоритм стиснення LZW).

1.2 Виконати його запис (Файл → Зберегти як) у форматах BMP (цей формат без стиснення), GIF (групове стиснення одночасно зі зменшенням кількості кольорів до 256), JPEG (для цього формату задати у відповідному спливаючому вікні, рис.2.1, значення параметру якості 100%).

1.3 Для кожного з форматів розрахувати (використовуючи ***Excel***) **коефіцієнт стиснення – співвідношення розміру в байтах нестисненого зображення** (BMP) **до стисненого**.

1.4 Повторити п.1.1 – 1.3 для трьох різних зображень, виконати усереднення для кожного з форматів (способів стиснення).

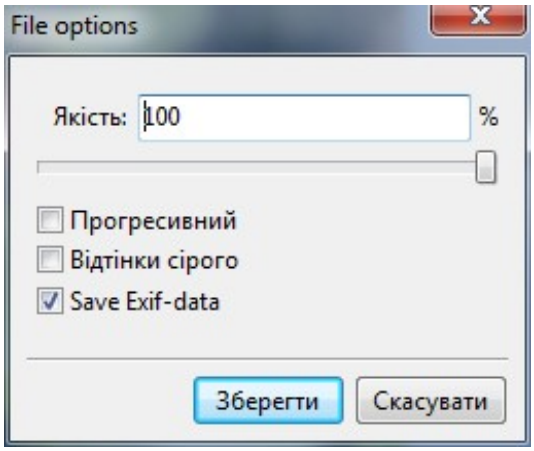

Рис. 2.1

1.5 Виконати п.1.1 – 1.4 з "сірими" зображеннями.

1.6 Виконати п.1.1 – 1.4 з "штриховими" (stroke) зображеннями.

1.7 Визначити, який з методів стиснення є найкращим для кожного типу зображень (кольорових, "сірих", "штрихових").

2. Дослідження залежності втратності від стиснення для формату JPG.

2.1 Ознайомитись з програмою ***PicCalc*** (рис.2.2), яка виконує порівняння двох зображень. Для такого порівняння необхідно завантажити ці зображення, числовий результат їх відмінності (значення середнього квадрату різниці усіх пікселів цих двох зображень) виводиться як **mid**. Увага, зображення, що порівнюються, повинні мати однаковий розмір (у пікселях)!

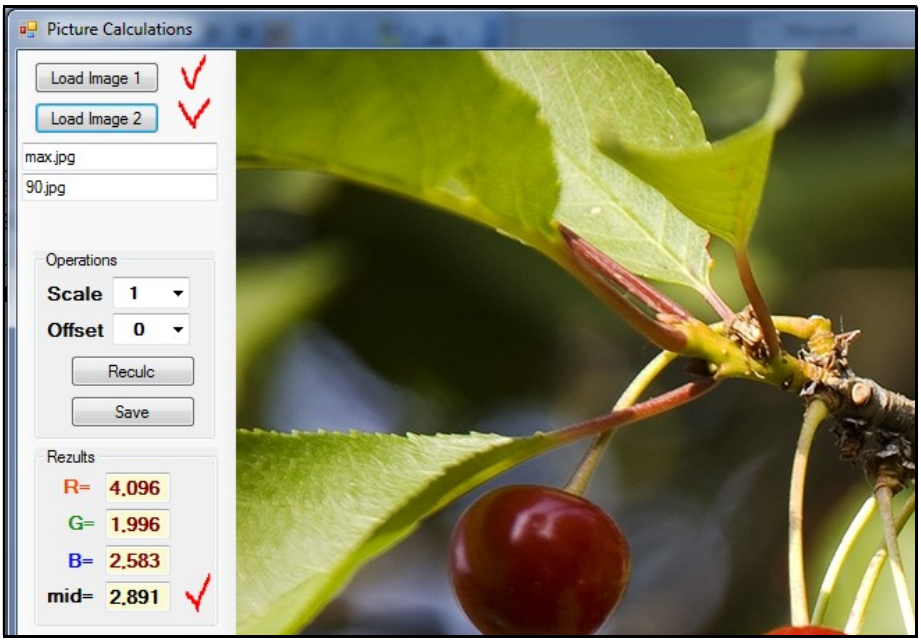

Рис. 2.2

2.2 Завантажити у графічний редактор досліджуване кольорове зображення у форматі TIFF (це буде оригінал, з яким будемо порівнювати) та послідовно виконати його запис у форматі JPG з різними значеннями (не менше 10) параметра якості – від 100 до 10).

2.3 За допомогою програми ***PicCalc***, завантажуючи як зразок оригінал зображення (BMP) та один з варіантів стиснення алгоритмом JPEG, отриманих у

попередньому пункті, розрахувати дані і побудувати графік залежності втратності (відмінності від оригіналу) від коефіцієнта стиснення.

2.4 Повторити п.2.2–2.3 для "сірого" зображення.

3. Дослідження впливу шумів.

3.1 Формування зашумленого зображення виконується за допомогою програми ***AddNoise*** (рис.2.3). Для цього треба завантажити у цю програму "чисте" зображення, задати певний рівень шуму (рекомендуємо обрати Level=64) та одне зі значень інтенсивності та натиснути кнопку Add. Після цього слід записати результат додавання шуму як нове зображення. Треба створити три таких зображення з різним рівнем шуму.

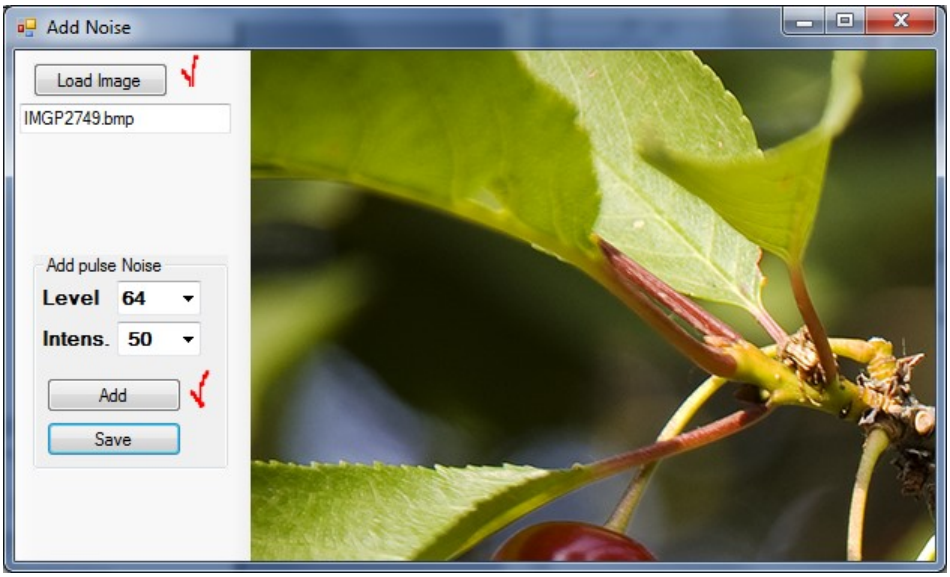

Рис. 2.3

3.2 Записати графічним редактором кожне з зроблених у п.3.1. зашумлених зображень у форматі JPG для чотирьох рівнів якості алгоритму (100, 70, 30, 10).

3.3 Побудувати графіки залежності коефіцієнта стиснення від параметру якості для обраних значень інтенсивності шуму.

3.4 Побудувати графіки залежності якості (відмінності від оригіналу) від параметру якості для обраних значень інтенсивності шуму.

Лабораторна робота №2. Дослідження точкової обробки зображень на прикладі перетворення координат

Мета роботи: Дослідити вплив інтерполяції на якість ресемплінгу дискретного зображення.

Загальні відомості

Комп'ютерна обробка зображень часто використовує перетворення простору таким чином, що значення точки результуючого зображення є функцією єдиної точки вхідного (п.1.3). В даному випадку не випадково використано термін точка, а не піксель, оскільки не завжди коректно говорити про відповідність єдиного пікселя дискретизованого зображення через те, що піксель вихідного зображення не відповідає точно геометричному положенню пікселя вхідного зображення. При цьому виникає потреба в інтерполяції. Тобто в обробку включається декілька сусідніх пікселів, але математична формалізація самого перетворення все одно відповідає отриманню «точки з точки»

Найбільш частою задачею зазначеного перетворення простору є ресемплінг кадру, тобто зміна кількості його пікселів. Іноді ресемплінг ототожнюють з терміном «масштабування», але під зміною масштабу більш правильно розуміти зміну співвідношення між метричними і піксельними розмірами зображення (без ресемплінгу).

Ця операція використовується для:

- підготовки для друку,
- зменшення об'єму даних (розміру файлу) при умові допустимості часткової втрати інформації (погіршення відтворення),
- уніфікації розмірів об'єктів на двох кадрах при потребі їх сумісної обробки (наприклад їх взаємному накладанні).

Іншим випадком точкової обробки (в даній роботі не розглядається) є поворот і зсув, які використовуються

- для правильного відтворення інформації,
- для узгодження положення об'єкту на двох кадрах при їх сумісній обробці.

Ще одним (більш складним) варіантом точкової обробки є деформація кадру, яка частіше за все використовується для

компенсації деформації кадру при його записі. До таких деформацій можна віднести перспективні (проективні) при неможливості оптимального положення камери, що зустрічається як при художньому, так і при технічному або науковому застосуванні фотографії. Така обробка також не розглядається в даній роботі.

Якість інтерполяції суттєво впливає на остаточний результат не тільки у художній фотографії, а й у технічній або науковій. Найбільш розповсюдженим (і надзвичайно швидким) методом є лінійна інтерполяція даних, яка забезпечує на відміну від найпростішої трансляції пікселя нерозривність значення функції. У випадку двовимірного сигналу (зображення) така інтерполяція отримала модифіковану назву білінійної, оскільки виконується лінійна інтерполяція по двох координатах. Рідше використовується бікубічний алгоритм, який забезпечує неперервність першої та другої похідних, що дає покращення в порівнянні з білінійною, але значно менше помітне, ніж різниця білінійної і трансляції.

Завдання

1. Ознайомитись з інструментом зміни піксельного та метричного розміру графічного редактора Image → Image Size та параметрами його настроювання.

2. Порівняти втратність ресемплінгу на основі використання кубічної інтерполяції з методом "найближчого сусіда".

При виконання роботи використовуються програмні засоби:

1. ***PixBuilder Studio*** (безкоштовний графічний редактор, надається)
2. ***PicCalc*** (спеціалізована програма, надається для виконання роботи***)***
3. ***MS Excel*** (або аналог)

Звіт виконується за допомогою MS Word (або аналогу). У звіт вноситься назва роботи, виконавець, дата виконання, результати роботи у вигляді таблиць і графіків (імпортуються з MS Excel). У звіт також вноситься опис інтерпретації отриманого.

Хід виконання

1. Завантажити у графічний редактор ***PixBuilder Studio*** деяке зображення, на основі його обробки ознайомитись з дією налаштувань вікна зміни розміру зображення (Редагування → Розмір зображення), рис. 2.4.

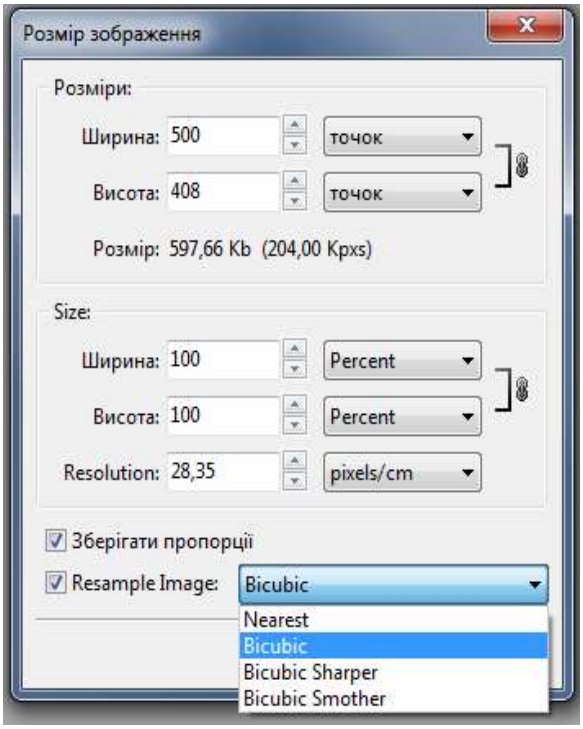

Рис. 2.4

2. На прикладі кольорового напівтонового зображення виконати набір операцій для декількох значень ***нових піксельних розмірів*** (не менше 10 значень в бік зменшення та 4 в бік збільшення). Для цього:

   2.1 Відкрити графічним редактором файл-оригінал зображення.

   2.2 Змінити розмір зображення (Редагування → Розмір зображення) з опцією бікубічної інтерполяцією (Bicubic), рис. 2.4, до нового значення у пікселях, яке задається у відповідному вікні форми (знайдіть його).

2.3 Змінити розмір зображення до оригінального (з такою ж інтерполяцією). Перевірити збіг розмірів (у пікселях) обробленого зображення з розмірами оригіналу за обома координатами, оскільки може з'явитись похибка округлення! Якщо один з розмірів не співпадає, треба зняти прапорець "зберігати пропорції" та виправити розміри явним завданням чисел.

2.4 Записати отриманий файл під новим ім'ям, у яке для зручності включити змінений розмір у пікселях.

2.5 Використовуючи програму ***PicCalc***, виконати розрахунок відхилення кожного «двічі обробленого» зображення від оригіналу. Увага, при виконанні п.2.1 – 2.4 кожного разу зображення-оригінал треба відкривати заново!

3. Повторити п.2 з використанням опції Nearest ("найближчий сусід"), тобто без інтерполяції.
4. Побудувати графік залежності відхилення "двічи обробленного" зображення від оригіналу від коефіцієнта ресемплінгу для варіантів бікубічної інтерполяції та без інтерполяції, порівняти результати.
5. Виконати аналогічне дослідження (п.2–4) для зображення міри.
6. Виконати візуальне порівняння якості ресемплінгу на прикладі міри (комбіноване зображення, надається з іншими зразками). При цьому зверніть увагу на "читабільність" текстової вставки на цьому зображенні.

Лабораторна робота №3. Дослідження локальної обробки зображень на прикладі згорткової фільтрації

Мета роботи: Дослідити особливості згорткової фільтрації зображень.

Загальні відомості

Для зменшення впливу шумів, покращення візуальної якості зображення і виділення на зображенні якихось елементів зазвичай використовують фільтрацію зображень. З математичної точки зору така фільтрація є аналогом більш звичної фільтрації часових сигналів з тією різницею, що зображення фактично є двовимірним просторовим сигналом. Тобто просторові координати (x; y) займають місце часу у розгляді сигналів. Таким чином, у найбільш загальному випадку фільтрація зображень зводиться до фільтрації просторового спектра зображення. Але такий метод потребує великих обчислювальних ресурсів. Тому часто замість частотної фільтрації використовують згорткову (п.1.4).

Згорткова фільтрація формально є повністю еквівалентною просторово-частотній (згадайте теорему про згортку у двовимірному випадку). При цьому має використовуватись ядро згортки, яке є результатом зворотного фур'є-перетворення просторово-частотного фільтра, тобто за координатними обмеженнями відповідає кадру зображення, що фільтрується. Але в багатьох випадках можна використати таке ядро згортки, у якому, крім відносно невеликого околу центральної точки, воно буде нульовим. Таким чином, згортку можна розраховувати за відносно малою апертурою, що суттєво зменшує ресурсоємність.

При такому способі значення кожного пікселя вихідного зображення формується на основі обробки деякої наперед заданої області (апертури) навколо оброблюваного пікселя вхідного зображення.

Зазвичай ядро згортки задається на квадратній апертурі непарного лінійного розміру, 3*3 або 5*5 (більші розміри використовуються досить рідко). Зразки таких функцій наведені на рис. 2.5.

1	1	1
1	1	1
1	1	1

/9

1	1	1
1	2	1
1	1	1

/10

0	-1	0
-1	5	-1
0	-1	0

-1	-1	-1
-1	9	-1
-1	-1	-1

Рис. 2.5. Зразки ядер згортки різного призначення.

Перші два випадки є прикладами так званих згладжувальних фільтрів, які використовуються для боротьби з шумами і для послаблення дрібних деталей зображення. Фактично такий спосіб фільтрації з точки зору обробки просторового спектра зводиться до фільтра нижніх частот, тобто придушує високі просторові частоти. Дві інші маски аналогічно до фільтру верхніх частот придушують низькочастотну частину спектру, що призводить до підкреслення країв. Слід відзначити, що останні дві матриці мають суму елементів 1, що їх відрізняє від відповідних матриць для виділення країв (останні як оператор диференціювання мають відповідно нульову суму).

Завдання

На кольоровому зображенні дослідити залежність ефективності придушення шуму при використанні згорткового фільтра.

Кількісно результат фільтрації будемо оцінювати як відмінність "зашумленого" зображення від "чистого" після виконання згорткової фільтрації цих зображень з гаусівським ядром згортки однакової апертури (однакового радіуса).

При виконання роботи використовуються програмні засоби:

1. ***PixBuilder Studio*** (безкоштовний графічний редактор, надається)
2. ***PicCalc*** (спеціалізована програма, надається для виконання роботи***)***
3. ***AddNoise*** (спеціалізована програма, надається для виконання роботи)
4. ***MS Excel*** (або аналог)

Звіт виконується за допомогою MS Word (або аналогу). У звіт вноситься назва роботи, виконавець, дата виконання, результати роботи у вигляді таблиць і графіків (імпортуються з MS Excel). У звіт також вноситься опис інтерпретації отриманого.

Хід виконання

1. Для дослідження придушення рівномірного (uniform) шуму за допомогою гаусової фільтрації вибрати з запропонованого набору зображень таке, що має дрібні деталі.
2. За допомогою програми ***AddNoise*** (рис. 2.6) задати (на ваш розсуд) деякий рівень (level) та інтенсивність (intensity) шуму. Натиснути кнопку Add та записати результат додавання шуму як окремий файл.
3. Виконати п.2 з іншими значеннями інтенсивності шуму.

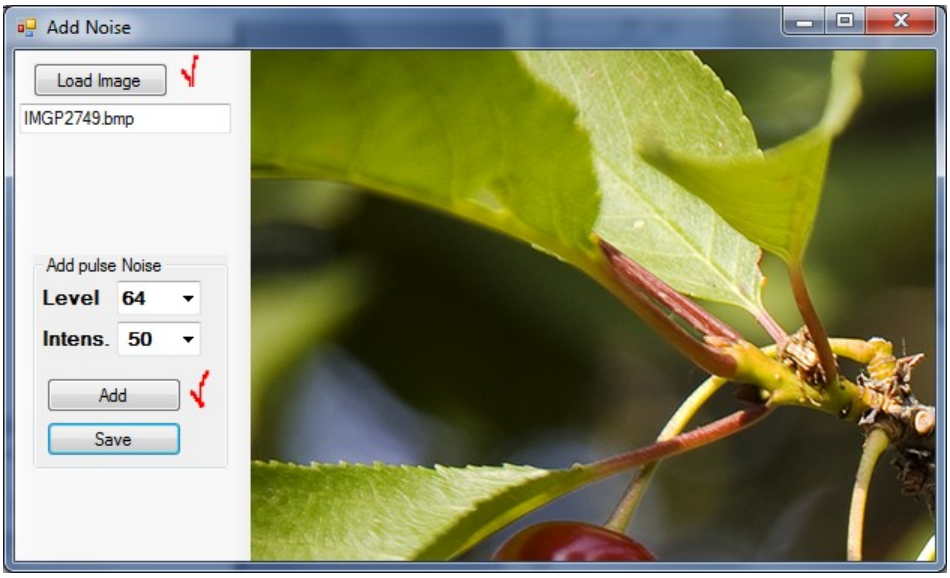

Рис. 2.6.

4. Для фільтрації використовуться графічний редактор ***PixBuilder Studio***, команда Ефекти → Розмиття Гауса. Налаштування апертури робимо у спливаючому вікні, яке при цьому з'явиться (рис.2.7). Для вибору мінімальної апертури найпростіше посунути слайдер вліво (показано

на рис.). Кожна наступна (більша) апертура фільтра найпростіше задається курсорними клавішами комп'ютера (більше на крок – "вгору", менше на крок – "вниз", потренуйтесь), див. табл. 2.1.

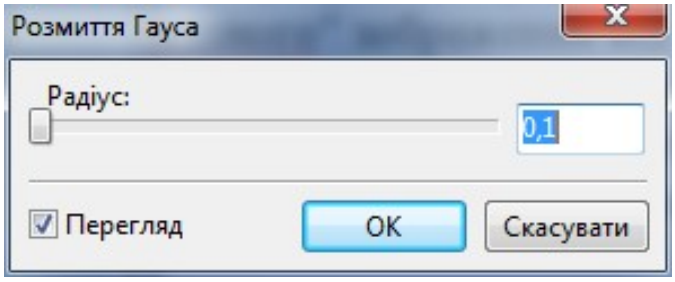

Рис. 2.7.

Таблиця 2.1 – співвідношення кроків та апертур:

Номер кроку	Радіус апертури
1	0,1
2	0,37
3	0,59
4	0,76
5	0,89
5	1

5. Для побудови необхідного графіку для різних значень апертури фільтра треба розрахувати значення відмінності "зашумленого" зображення від "чистого" після виконання згорткової фільтрації цих зображень з гаусівським ядром однакової апертури (однакового радіуса). Кожна точка шуканої залежності може бути отримана таким чином:
   5.1 Завантажити в ***PixBuilder Studio*** "чисте" зображення, виконати фільтрацію з необхідною апертурою, записати його як окремий файл. Зручно для подальшої обробки в назві файлу вказувати номер кроку або апертуру фільтра, який було застосовано (табл.1).
   5.2 Виконати п.5.1 для "зашумленого" зображення з тою самою апертурою.
   5.3 За допомогою програми ***PicCalc*** розрахувати числове значення відмінності фільтрованого

"зашумленого" зображення від фільтрованого "чистого".

5.4 Повторити п.5.1 – 5.3 для декількох (вказаних у табл.1) апертур фільтра.

5.5 Побудувати графік.

6. Повторити п.5 для зображення з іншим налаштуванням доданого до чистого зображення шуму (див. виконання п.2 – 3).
7. Розрахувати та побудувати графік залежності відмінності незашумленого оригіналу від апертури фільтра (результатів фільтрації з вказаними вище апертурами). Для цього знову використовується програма ***PicCalc***.
8. Дослідити, до результату фільтрації я яким значенням апертури згортки (раніше виконане завдання) є найближчим результат згорткової фільтрації використанням команди Ефекти → Довільна матриця (рис. 2.8.) у таких варіантах:
   - Настроювання – Blur;
   - Настроювання – Blur Gaussian;

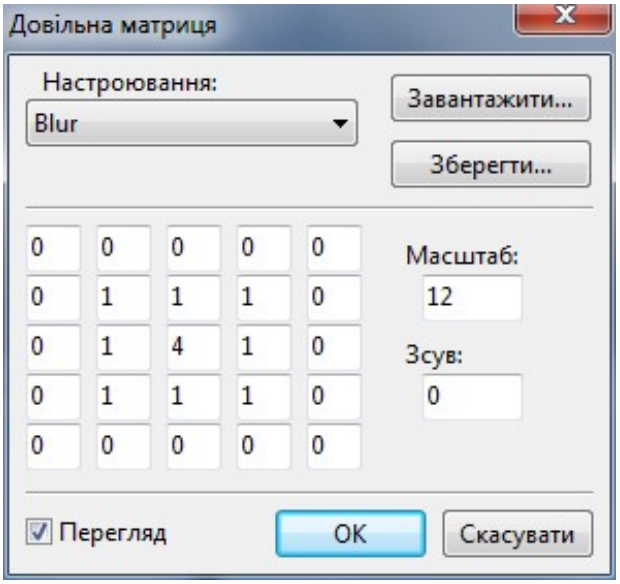

Рис. 2.8.

Література

1. *William K Pratt. Digital image processing. Third edition. John Wiley & Sons, Inc. 2001 – 738 p.*

2. *Rafael C. Gonzalez., Richard E. Woods. Digital image processing. Second edition. – Pearson Education Inc. 2002. – 793 p.*

3. *Кононов М.В. Комп'ютерні технології. – К.: Київський університет, 2019. – 281 с. (http://rex.knu.ua/wp/wp-content/uploads/2019/11/Kononov_Computer_techologies.pdf)*

4. *Лавер В.О., Левчук О.М. Обробка зображень та мультимедіа. – Ужгород : ПП «АУТДОР-ШАРК», 2021. – 51 с.*