УДК 621.37

ОПТИМІЗОВАНИЙ ВАРІАНТ ФІЛЬТРАЦІЇ ЗОБРАЖЕНЬ

Король М. В., магістр, Кондрат'єва Н. О., к. ф-м. н., доцент, Мухін В. В., к. т. н., доцент, Леонтьєва В. В., к. ф-м. н., доцент

Запорізький національний університет,
вул. Жуковського, 66, м. Запоріжжя, 69600, Україна

n-kondr@mail.ru, vleonteva@mail.ru

У статті наведені способи реалізації фільтрації зображень, оптимізовані за швидкістю виконання. Розроблено методику реалізації алгоритму швидкої обробки значень пікселів зображення. Проведено дослідження ефективності прискореного способу обробки для експериментального набору вхідних даних.

Ключові слова: оптимізація, зображення, фільтрація, об'єктно-орієнтоване програмування.

ОПТИМИЗИРОВАННЫЙ ВАРИАНТ ФИЛЬТРАЦИИ ИЗОБРАЖЕНИЙ

Король М. В., магистр, Кондратьева Н. А., к. ф-м. н., доцент, Мухин В. В., к. т. н., доцент, Леонтьева В. В., к. ф-м. н., доцент

Запорожский национальный университет,
ул. Жуковского, 66, г. Запорожье, 69600, Украина

n-kondr@mail.ru, vleonteva@mail.ru

В статье приведены способы реализации фильтрации изображений, оптимизированные по скорости использования. Разработана методика реализации алгоритма быстрой обработки значений пикселей изображения. Проведено исследование эффективности ускоренного способа обаботки для экспериментального набора входных данных.

Ключові слова: оптимизация, изображение, фильтрация, объектно-ориентированное программирование.

OPTIMIZED OPTION OF IMAGE FILTERING

Korol M. V., graduate student,
Kondratieva N. O., PhD in Physics and Mathematics, Associate Prodessor,
Mukhin V. V., PhD in Enginering, Associate Professor,
Leontieva V. V., PhD in Physics and Mathematics, Associate Prodessor

Zaporizhzhue National University,
Zhukovsky str., 66, Zaporizhzhya, 69600, Ukraine

n-kondr@mail.ru, vleonteva@mail.ru

The paper presents ways of implementation filtering of images that are optimized for speed of execution. The method of the algorithm fast processing of pixel values of the image is developed. A study of the effectiveness of accelerated processing method for the experimental set of input data.

Filters for enhance picture quality is among the most common methods of processing images in computer graphics. Improving possible by reducing the noise in the image, increasing contrast or color images, image edge detection and so on.

Nowadays, the number of images from different digital media is very large, and their quality can not always satisfy. Conventional methods of improving the image quality can sometimes be a little behind. For example, when a lot of images or volume images are very large, so their treatment can be spent quite some time. In this situation it is necessary to improve conventional filters, making them faster. Sometimes speed can pay for quality, though not very much.

Acceleration of procedures performed by filtering unsafe method that exists in the programming language C#. Usually, working with an image, we need to read the value of pixels for further processing. To read and write pixel values using methods familiar with C# getPixel / setPixel, but these methods do not include a special rate by using these methods for each new method challenged released new buffer memory for recording pixel value. Therefore unsafe to use the method in the case of image processing to be rational. Its essence lies in the fact that all the images placed in the buffer where we

have an opportunity to do with the values of pixels using pointers. This method is faster because they do not need to create new variables at each pixel to appeal and no need to allocate new volumes of memory, since the image is always in buffer.
Developed subsystem which filters (removing noise from images, image edge detection filter to enhance image color range) united and could be used to image one by one using a method of optimizing and without it, thus making it possible to check in which cases the program handles image faster and slower. Comparing filters implemented using optimization unsafe and no optimization has been proven that this method really speeds up filters (filter used to size a floating window size 3×3).
For implementation used development environment C# 2010.
Key words: optimization, image filtering, object-oriented programming.

ВСТУП

Зазвичай зображення, сформовані різними інформаційними системами, спотворюються під впливом перешкод (шумів), що у свою чергу ускладнює як їх візуальний аналіз людиною-оператором, так і автоматичну обробку в ЕОМ. Так, при розв'язанні деяких задач обробки зображень в ролі перешкод (шуму) можуть виступати ті чи інші компоненти самого зображення. Наприклад, при аналізі космічного знімку земної поверхні може виникнути задача визначення меж між її окремими ділянками – лісом і полем, водою і сушею тощо. Отже, з точки зору цієї проблеми, окремі деталі зображення в середині певних підобластей є перешкодою (шумом).

Як відомо [1-4], ослаблення дії перешкод (шуму) досягається фільтрацією, завдяки якій яскравість (сигнал) кожної точки вихідного зображення, спотвореного шумом, замінюється деяким іншим значенням яскравості, яке визначається в найменшій мірі рівнем зашумленості. При цьому часто зображення – це двовимірна функція просторових координат, яка змінюється за цими координатами повільніше (іноді значно повільніше), ніж шум, який також є двовимірної функцією. Це дозволяє при оцінці корисного сигналу в кожній точці зображення взяти до уваги деяку кількість сусідніх точок, скориставшись певною схожістю сигналу в цих точках. В інших випадках, навпаки, ознакою корисного сигналу є різкі перепади яскравості. Однак, як правило, частота цих перепадів є відносно невеликою, так що на значних проміжках між ними сигнал є або постійним, або таким, що повільно змінюється. В цьому випадку властивості сигналу проявляються при спостереженні його не тільки в локальній точці, але й при аналізі її околиці. При цьому потрібним є зауважити, що поняття околиці є досить умовним. Вона може бути утворена лише найближчими сусідами, а також може містити досить віддалені точки зображення. Звичайно, в останньому випадку буде абсолютно різною ступінь впливу віддалених і близьких точок на прийняття рішення фільтром у даній точці зображення.

Для зменьшення шуму на зображенні досить часто використовується так звана Медіанна фільтрація, запропонована Дж. Тьюки в 1971 році для аналізу економічних процесів, яка є одним з найбільш поширених методів (алгоритмів) обробки зображень. Сутність цього методу полягає в наступному. Одновимірний медіанний фільтр використовує вікно, яке охоплює непарне число елементів зображення. Далі, у зазначеному вікні центральний елемент замінюється медіаною усіх елементів зображення. При цьому для непарного випадку медіаною дискретної послідовності обирається той її елемент, для якого існують елементи, які є меншими або рівними (більшими або рівними) йому за величиною. Крім того, потрібним є дотримання правил:

– медіана добутку сталої K і послідовності $f(j)$ дорівнює

$$med\{Kf(j)\} = Kmed\{f(j)\};$$

– медіана суми сталої K і послідовності $f(j)$ дорівнює

$$med\left\{K+f\left(j\right)\right\}=K+med\left\{f\left(j\right)\right\};$$

– медіана суми двох довільних послідовностей $f\left(j\right)$ і $g\left(j\right)$ не дорівнює сумі їх медіан

$$med\left\{f\left(j\right)+g\left(j\right)\right\}\neq med\left\{f\left(j\right)\right\}+med\left\{g\left(j\right)\right\}.$$

Однак, при всій привабливості зазначеного методу потрібно зауважити, що можливості аналізу дії медіанного одновимірного фільтра є досить обмеженими. Концепцію медіанного фільтра можливо узагальнити на два виміри, застосовуючи двовимірне вікно бажаної форми, наприклад, прямокутне або близьке до кругового. У результаті двовимірний медіанний фільтр з вікном розміру $L\times L$ забезпечить ефективніше зменшення шуму порівняно з послідовно застосованим горизонтальним або вертикальним одновимірним медіанним фільтром з вікном розміру $L\times 1$; двовимірна обробка, проте, може призвести до істотнішого послаблення сигналу.

У цій статті наведено спосіб реалізації медіанного фільтра, який дозволяє оптимізувати роботу фільтра за швидкістю виконання. Крім того, у роботі проводиться порівняння існуючих методик із запропонованим підходом.

ЗВИЧАЙНА РЕАЛІЗАЦІЯ ФІЛЬТРА

Існуючі комерційні системи обробки зображення не дають можливості зробити порівняльну характеристику роботи фільтрів, оскільки більшість використовують багатопоточність і роблять фільтрацію на попередження ще під час діалогу з користувачем, показуючи частковий результат роботи. Для вирішення цієї проблеми реалізуємо однопоточний медіанний фільтр.

Після ознайомлення з методами фільтрації зображень, стає зрозумілим, що пікселі слід обробляти зліва направо і зверху вниз. При цьому для кожного відгуку фільтра потрібно зчитувати, чи потрапили вони в додатковий масив для тимчасового зберігання значень пікселів, потім проводити маніпуляції з цим набором пікселів та повертати відгук фільтра у середній елемент масиву. Все це реалізується за допомогою циклу, використовуючи покажчик на центральний піксель і покажчик на піксель-результат.

Представимо на блок-схемі, як працює фільтр видалення шуму (рис. 1).

Суть цього фільтра [7] полягає в тому, що за допомогою рухомого вікна здійснюється проходження по кожному пікселю зображення, сортування значення пікселів рухомого вікна та заміна значення поточного пікселя на значення середнього пікселя відсортованого масиву, що, таким чином, дозволяє позбутися на зображенні різких стрибків значень пікселя.

ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ФІЛЬТРА

Цей розділ статті присвячений висвітленню основних положень методики оптимізації роботи медіанного фільтра за показником швидкості обробки зображення.

Прискорювати роботу процедури фільтрації будемо за рахунок unsafe методу, який існує в мові програмування C# [5, 6, 8]. Зазвичай, працюючи з зображенням, нам потрібно зчитувати значення пікселів зображення для подальшої їх обробки. Для зчитування та записування значень пікселів звично використовувати методи C# getPixel/setPixel, але ці методи не вирізняються особливою швидкістю, за допомогою цих методів для кожного нового виклику методу виділяється новий буфер пам’яті для запису значення пікселю. Тому використовувати метод unsafe у випадку з обробкою зображень буде більш раціональним. Сутність зазначеного методу полягає в тому, що все зображення розміщується в буфері, де існує змога беспосередньо використовувати значення пікселів, застосовуючи вказівники. Цей метод працює швидше, оскільки не потрібно створювати нові змінні при кожному звертанні

до значень пікселів, а також не потрібно виділяти нові обсяги пам'яті, оскільки зображення постійно знаходиться в буфері пам'яті.

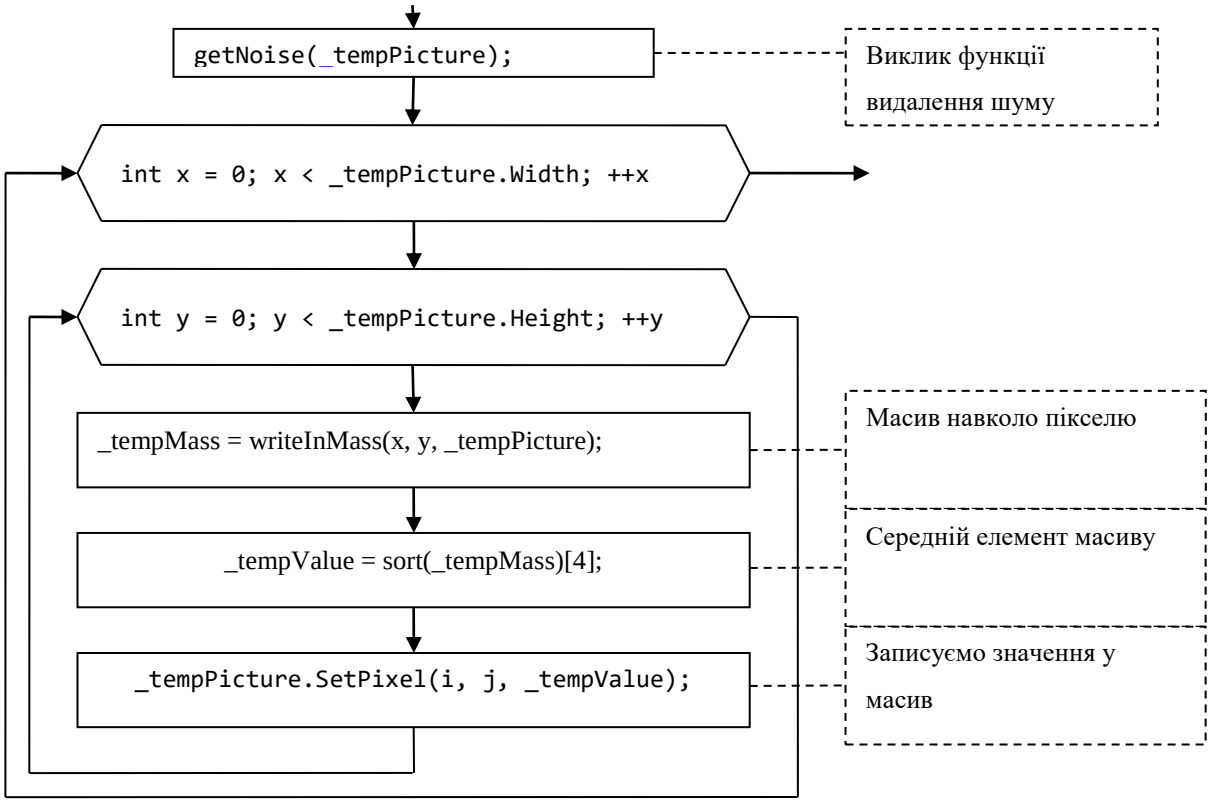

Рис. 1. Блок-схема реалізації фільтра видалення шуму

За допомогою методу LockBits переносимо зображення в буфер і дозволяємо проводити з зображенням певні маніпуляції, а саме зчитувати та змінювати значення пікселів, отримувати довжину та ширину зображення.

C# підтримує метод LockBits, який дозволяє управляти масивом точок растрового рисунку в некерованому буфері пам'яті та замінювати точки на рисункові точками з буферу.

Проведемо аналіз застосовуваних підходів до обробки зображення. У першому випадку ми працювали з зображенням за допомогою команд getPixel/setPixel, а в другому випадку – за допомогою команди LockBits перенесли зображення до буфера та працювали зі значеннями беспосередньо, що виявилося значно швидшим.

Результати роботи реалізованих медіанних фільтрів зведені в таблицю 1. Для реалізації використовувалося середовище розробки C# 2010.

Таблиця 1 – Час роботи в секундах при застосуванні звичайного методу та методу unsafe

Фільтр	Час роботи без використання unsafe методу	Час роботи, використовуючи unsafe метод
Покращення контрасту	1.457	0.411
Видалення шуму	2.231	0.673
Підвищення чіткості	0.77	0.217

Порівнюючи наведені в табл. 1 результати, бачимо, що методи фільтрації зображень можуть бути оптимізовані за швидкістю більш ніж у 3 рази (для фільтрів використовувалось рухоме вікно розміром 3×3).

ВИСНОВКИ

Отже, в роботі запропонований спосіб реалізації медіанного фільтра, який дозволяє оптимізувати роботу фільтра за швидкістю виконання обробки зображення. Для зазначеного підходу проведено аналіз з існуючим підходом, у результаті якого отримано висновок, що методи оптимізованої фільтрації із рухомим вікном розміром 3×3 зображень можуть обробляти зображення швидше, ніж у 3 рази. Цей підхід може бути узагальнений на випадок застосування фільтра з вікном розміром $L \times L$.

ЛІТЕРАТУРА

1. Абламейко С. В. Обработка изображений: технология, методы, применение / С. В. Абламейко, Д. М. Лагуновский. – М. : Амадфея, 2000. – 304 с.
2. Быков Р. Цифровое преобразование изображений / Р. Быков. – М. : Горячая линия, 2003. – 228 с.
3. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес. – М. : Техносфера, 2005. – 1007 с.
4. Гонсалес Р. Цифровая обработка изображений в среде MATLAB / Р. Гонсалес. – М. : Техносфера, 2006. – 616 с.
5. Гуриков С. Введение в программирование на языке Visual C#/ С. Гуриков. – М. : Инфра, 2013. – 448 с.
6. Мартин М. Принципы работы на C# / М. Мартин. – М. : Символ-Плюс, 2011. – 768 с.
7. Стругайло В. В. Обзор методов фильтрации и сегментации цифровых изображений / В. В. Стругайло // Наука и образование. Электронное научно-технические издание. – Московский автомобильно-дорожный государственный технический университет, 2012. – С. 270-281.
8. Фленов М. Библия C# / М. Фленов. – М. : БХВ, 2011. – 560 с.

REFERENCES

1. Ablameyko, S.V. and Lagunovsriy, D.M. (2000), *Obrabotka izobrazheniy: tekhnologiya, metody, primenenie* [Image Processing: technology, methods, application], Amadfeya, Moskow, Russia.
2. Bykov, R. (2003), *Tsifrovoe preobrazovanie izobrazheniy* [Digital image conversion], Goryachaya liniya, Moskow, Russia.
3. Gonsales, R. (2005), *Tcifrovaya obrabotka izobrageniy* [Digital image processing], Tekhnosfera, Moskow, Russia.
4. Gonsales, R. (2006), *Tsifrovaya obrabotka izobrazheniy v srede MATLAB* [Digital image processing in MATLAB environment], Tekhnosfera, Moskow, Russia.
5. Gurikov, S. (2013), *Vvedenie v programmirovanie na yazyke C#* [Introduction to programming in Visual C #], Infra, Moskow, Russia.
6. Martin, M. (2011), *Printsipy raboty C#* [How it works in C #], Simvol-Plus, Moskow, Russia.
7. Strugaylo, V.V. (2012), "Overview filtering techniques and segmentation of digital images", *Nauka i obrazovanie*, *Elektronnoe nauchno-tekhnicheskie izdanie*, Moskovskiy avtomobil'no-dorozhnyy gosudarstvennyy tekhnicheskiy universitet, Russia, pp. 270-281.
8. Flenov, M. (2011), *Bibliya C#* [Bible C #], BKHV, Moskow, Russia.