

ОСОБЛИВОСТІ АЛГОРИТМІЧНИХ ТА ПРОГРАМНИХ ЗАСОБІВ ОПРАЦЮВАННЯ НЕЧІТКИХ ЗОБРАЖЕНЬ

Р. І. Яркун, Я. С. Парамуд

Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин
E-mail: ruslan.yarkun.ki.2020@lpnu.ua, yaroslav.s.paramud@lpnu.ua

© Яркун Р. І., Парамуд Я. С., 2023

Досліджено особливості алгоритмічних та програмних засобів опрацювання нечітких зображень. Розглянуто три фільтри: `CIgaussianBlur`, `CIUnsharpMask` та `CIBlendWithAlphaMask`, які дають змогу покращити якість зображень, зменшити шуми та відтворити деталі. Початкове завдання – опрацювання нечіткості зображень. Для цього використовують фільтр `CIgaussianBlur`, який застосовує гауссівське розмиття до зображення. Це розмиття зменшує високочастотні шуми і надає плавність контурам об'єктів. Фільтр `CIUnsharpMask` застосовують для відновлення деталей зображення. Цей фільтр віднімає розмиту версію від оригінального зображення, що дає змогу виділити важливі деталі й підвищити чіткість зображення. Крайній фільтр, `CIBlendWithAlphaMask`, використовується для змішування двох зображень з використанням альфа-маски. Цей фільтр дає змогу контролювати прозорість та налаштовувати спосіб змішування зображень. В результаті можна досягти реалістичнішого та естетичнішого зображення.

Розглянуто принципи роботи кожного з фільтрів, наведено приклади їх використання та описано отримані результати. Дослідження виявили, що застосування цих фільтрів дає змогу покращити якість нечітких зображень, знизити шуми і підвищити різкість деталей. Результати досліджень можуть бути корисними для використання з метою обробки зображень, для комп'ютерного зору та графічного дизайну. Використання описаних фільтрів може допомогти поліпшити візуальні характеристики зображень та забезпечити точнішу інтерпретацію нечітких зображень.

Ключові слова: нечіткі зображення; алгоритмічні засоби; програмні засоби; обробка зображень; фільтри.

Вступ

Неодноразово виникають ситуації, коли нечіткі зображення потребують опрацювання з метою поліпшення якості, аналізу або отримання корисної інформації. Такі зображення трапляються у багатьох галузях, серед яких медицина, транспорт, розвідка, промисловість тощо.

Методи та засоби опрацювання нечітких зображень мають важливе значення для відновлення, покращення якості та аналізу таких зображень. Нечіткі зображення характеризуються наявністю неоднорідностей, нечіткості та руху об'єктів, що призводить до втрати деталей та погіршення якості зображення.

Одним з найважливіших завдань у цій області є розроблення методів та алгоритмів для ефективного опрацювання нечітких зображень. Ці методи повинні забезпечувати якісне відновлення зображень, зменшення шуму, виявлення об'єктів та аналіз їх руху. Для досягнення цих цілей використовують різноманітні техніки, зокрема фільтрацію, сегментацію, відновлення зображень та методи компресії даних. Застосування нечіткої логіки, штучних нейронних мереж та інших методів машинного навчання також покращує результати опрацювання нечітких зображень.

Мета цього дослідження полягає у вивченні та аналізі методів та засобів опрацювання нечітких зображень з метою поліпшення якості, виявлення об'єктів та аналізу їх руху. Основні завдання – розроблення нових або вдосконалення наявних алгоритмів та технік, порівняння різних підходів та визначення їх ефективності за допомогою відповідних метрик.

Сподіваємося, що результати цього дослідження сприятимуть покращенню опрацювання нечітких зображень і широко застосовуватимуться у великому спектрі галузей, що потребують аналізу та обробки зображень. Дослідження у цій царині – актуальне наукове та технічне завдання.

Аналіз літературних джерел

У наведених літературних джерелах розглянуто використані методи, позитивні аспекти та недоліки у засобах опрацювання нечітких зображень.

У роботі [1] використано такі фільтри: баланс кольору, яскравість/контраст, розмиття, збільшення різкості, зміни положення. Ці фільтри не виконують роботи за оптимальний час. Автори не передбачили опрацювання усього зображення, а лише фрагмент, який виділяється. Відповідно такі засоби в багатьох випадках неефективні, а в окремих випадках неприйнятні. У літературному джерелі [2] автори проаналізували поведінку сингулярних чисел цифрових зображень під час застосування фільтра “Motion blur”. Використовуючи результати досліджень, розробили алгоритм для виявлення розмиття зображень за допомогою фільтра “Motion blur”. Доцільно висловити думку щодо деяких недоліків фільтра “Motion blur”. По-перше, його ефективність у відокремленні оригінальних зображень від фальсифікованих обмежена, особливо на складних зображеннях. По-друге, обмежений спектр досліджуваних кутів розмиття обмежує розуміння його властивостей [2]. У роботі [3] розглянуто метод нелінійної фільтрації. Недоліки цього методу: втрата деталей, виникнення артефактів, таких як гострі краї та неестетичні перехідні ефекти, висока обчислювальна складність [3]. Літературне джерело [4] висвітлює особливості застосування нейромережових технологій для опрацювання нечітких зображень. Однак недостатня кількість статистичних даних обмежує застосування таких технологій. У роботі [5] описано метод “двофазового виправлення розмиття”. Наше бачення недоліків цього методу таке: по-перше, цей метод може бути вимогливим щодо обчислювальних ресурсів, особливо під час оброблення великих зображень або високих рівнів шуму. По-друге, можливість впровадження цього методу для кольорових зображень ускладнюється потребою в розробленні адаптованих регуляризаторів та урахуванням різних моделей шуму для кожного каналу. Крім того, для подальшого вдосконалення методу може бути необхідне дослідження теоретичних аспектів, таких як швидкість збіжності та оптимізація параметрів методу [5]. Автори джерела [6] пропонують метод “Виправлення розмиття природного зображення з використанням супергауссових полів”. Під час аналізу звернуто увагу на деякі недоліки цього методу: по-перше, метод може бути обмежений у застосуванні до зображень з високим рівнем шуму, оскільки не враховує шумові компоненти в оцінці розмиття. По-друге, незважаючи на його ефективність у виправленні розмиття, метод може бути вимогливим до обчислювальних ресурсів через велику кількість параметрів, що потребують оптимізації. Складність реалізації методу “Виправлення розмиття природного зображення з використанням супергауссових полів” може бути високою через потребу в розробленні та оптимізації алгоритмів обробки зображень, обчислювальної логіки та параметрів методу. Для досягнення оптимальних результатів необхідні глибокі напрацювання в області оброблення зображень та математичних моделей, а також доступ до потужних обчислювальних ресурсів [6]. У роботі [7] описано “Глибокий генеративний фільтр для виправлення розмиття руху”. Серед недоліків цього можна відзначити таке: по-перше, цей метод може бути вимогливим щодо обчислювальних ресурсів і часу, оскільки використовує глибинні нейронні мережі, які потребують значних обчислень. По-друге, можливість застосування методу до зображень зі складним рухом може бути обмеженою, оскільки у моделі обмежена здатність відтворювати складні рухові шаблони. Крім

того, для досягнення кращих результатів, можливо, потрібно вдосконалити архітектуру нейронної мережі та оптимізувати параметри [7]. У літературному джерелі [8] автори запропонували використання обробки шуму для покращення якості зображень за допомогою напрямних фільтрів. Цей підхід може бути обмежений у вирішенні проблеми шуму, особливо в ситуаціях, коли шум є складним або іншого типу, що відрізняється від шуму, для якого були розроблені напрямні фільтри. Оброблення шуму може призводити до втрати деталей та розмиття зображення, що погіршує загальну якість відновленого зображення. Літературне джерело [9] висвітлює такі методи обробки зображень: виділення країв, кольорові простори, лінійна корекція та ретинекс. Доцільно висловити свою думку про недоліки одного з них, а саме недоліки методу “Виділення країв”. По-перше, цей метод часто має проблеми з точністю виділення країв на зображеннях з великою кількістю шуму або складними текстурами. По-друге, він може призводити до виникнення фальшивих країв або пропускання деяких дрібних деталей, що призводить до втрати важливої інформації. Також варто зазначити, що метод “Виділення країв” на великих зображеннях може бути ресурсозатратним та потребувати додаткового часу на підбір оптимальних параметрів для досягнення бажаного результату.

Аналіз літературних джерел дає підстави констатувати, що дослідження в царині опрацювання нечітких зображень є доцільними.

Постановка задачі

Як показує огляд літературних джерел, доцільно виконати дослідження щодо підвищення ефективності та швидкодії методів та засобів опрацювання нечітких зображень.

Результати дослідження

Є багато методів та алгоритмів обробки нечітких зображень. Ми розглянемо декілька з них і детальніше зупинимося на тих, що використані у дослідженнях. Доцільно почати з деконволюції. Деконволюція ґрунтується на математичній моделі розмиття зображення та використовує алгоритми оберненої фільтрації для відновлення різкості та деталей. Основна ідея полягає в тому, щоб знайти обернену функцію до функції розмиття, що дає змогу відновити оригінальні деталі. Один з популярних алгоритмів деконволюції – алгоритм Люсі – Річардсона (Lucy – Richardson). Він використовує ітеративний підхід, за яким обчислюють оцінку оригінального зображення, яка поступово покращується з кожною ітерацією. Цей алгоритм ґрунтується на моделі розмиття та використовує метод максимальної правдоподібності для оцінювання оригінального зображення. Деконволюцію можна використовувати для покращення якості зображень у відеообробці, де важливо зберегти деталі під час руху об’єктів або знизити рівень шуму. У фотографії цей метод може допомогти відновити розмиті або розфокусовані зображення. У медичній діагностиці деконволюція може допомогти поліпшити якість медичних зображень, потрібних для досліджень та діагностики. Відновлення різкості використовується для підвищення чіткості та деталізації розмитих або низької різкості зображень. Цей метод оснований на аналізі локальних структур зображення та застосуванні підвищення різкості до цих областей. Один з алгоритмів відновлення різкості – алгоритм Уншарп-маска (Unsharp Mask). Він передбачає фільтрацію з використанням маски, що збільшує різкість, для виділення деталей та контурів у зображенні. Цей алгоритм застосовує гауссівське розмиття для створення маски, яка виокремлює високочастотні компоненти зображення. Відновлення різкості може бути застосовано у відеообробленні для поліпшення якості зображень під час руху або в умовах обмеженої освітленості. У фотографії цей метод застосовують для виправлення розмитих зображень або збільшення деталей. У медичній діагностиці, відновлення різкості може допомогти виявляти точніші деталі та ознаки хвороби на медичних зображеннях. Фільтри видалення шуму застосовують для зменшення шумових компонентів на зображеннях, спричинених різними джерелами, такими як електронний шум камери або компресія даних. Існує багато алгоритмів видалення шуму, таких як фільтри низькочастотного відфільтрування, медіанні фільтри, фільтри усереднення та ста-

тистичні методи. Ці алгоритми аналізують структуру зображення та використовують різні математичні моделі, щоб виявити та видалити шумові компоненти, зберігаючи при цьому деталі та розмір зображення. Фільтри видалення шуму широко використовують у відеообробленні для покращення якості зображень та зниження шуму, що може бути спричинений різними джерелами, такими як електронний шум камери або компресія даних. У фотографії ці фільтри можуть допомогти знизити шум та покращити якість зображень. В медичній діагностиці видалення шуму може допомогти зберегти важливі деталі та структури на медичних зображеннях під час процедур діагностики та аналізу.

За результатами дослідження запропоновано трьохфазовий алгоритм опрацювання нечітких зображень. Цей алгоритм спрямований на зняття розмитості з вхідного зображення. Перша фаза передбачає застосування гаусівського розмиття для зменшення загальної розмитості. На другому етапі застосовується фільтр розшарування контурів для підвищення ребер і деталей на зображенні. Третя фаза передбачає змішування початкового зображення з результатом розшарування контурів для поєднання покращених деталей з підлягаючим розмитим зображенням.

Варто зауважити, що ефективність цього алгоритму може варіюватись залежно від характеристик вхідного зображення та вибраних значень параметрів. Налаштування параметрів, таких як радіус розмиття та інтенсивність розшарування контурів, допоможе уточнити ефект зняття розмитості для досягнення бажаного результату. А тепер розглянемо кожен фільтр окремо. Розпочнемо з гаусівського розмиття. Гаусівське розмиття є одним з найпоширеніших методів для зняття розмитості із зображень. Воно ґрунтується на гаусівській функції, що являє собою розподіл інтенсивності значень пікселів. Гаусівське розмиття використовує цей розподіл, щоб згладити грані та розрізніти деталі, що сприяє підвищенню різкості зображення та видаленню розмитості. Розшарування контурів має два основних параметри. Радіус розшарування (`kCInputRadiusKey`) визначає, як далеко залучені пікселі враховуються у розшаруванні контурів. Інтенсивність розшарування (`kCInputIntensityKey`) контролює ступінь поліпшення якості контурів та деталей на зображенні. Розшарування контурів використовує внутрішній оператор розмиття (наприклад, гаусівське розмиття) для створення розмитої версії зображення. Надалі від цієї розмитої версії віднімається оригінальне зображення, що призводить до виділення контурів і деталей. Гаусівське розмиття може використовуватись для зменшення шуму на зображенні. Воно розподіляє інтенсивність пікселів навколо заданого пікселя, що допомагає згладити неправильності та дрібні деталі, які можуть бути спричинені шумом. Це особливо корисно у разі фотографування в умовах низьких або занадто високих значень освітлення.

Гаусівське розмиття використовують для творчих цілей, наприклад, для створення ефекту м'якої фокусної зони або змивання фону. Цей ефект може бути особливо ефективним у портретній фотографії, де фокус зосереджений на об'єкті, а фон мерехтить, що створює приємну розмиту атмосферу.

Гаусівським розмиттям можна скористатись для видалення небажаних деталей або розмиття чутливої інформації на зображенні. Наприклад, на фотографіях архітектурних споруд можуть бути небажані елементи, такі як знаки або люди, які заважають сприйняттю зображень. Застосування гаусівського розмиття може допомогти згладити ці деталі та зробити їх менш помітними або навіть видалити їх.

Гаусівське розмиття може бути використане для підвищення роздільної здатності зображення. Це корисно в ситуаціях, коли зображення має низьку якість або розмитість, наприклад, в медичній діагностиці. Застосування гаусівського розмиття може допомогти зробити зображення чіткішим та зменшити розмитість, що сприяє точнішій інтерпретації та аналізу зображення. Наступний фільтр – це фільтр розшарування контурів (`CIUnsharpMask`). Розшарування контурів є методом, який посилює ребра і деталі на зображенні, надаючи йому чіткішого вигляду. Його застосовують після розмиття для покращення контурів та деталей, які можуть бути втрачені внаслідок розмиття.

Фільтром розшарування контурів можна скористатись для створення ефектів стилізації та художнього редагування зображень. Він підкреслює контури об'єктів і надає їм характерного гра-

фічного стилю. Це може бути корисно для створення художніх ефектів, графічного дизайну, ілюстрацій та інших творчих проєктів. Фільтр розшарування контурів застосовують також у медичній діагностиці. Він може допомогти виявити та підкреслити контури об'єктів на медичних зображеннях, таких як рентгенівські знімки, магнітно-резонансні томограми тощо. Це допомагає лікарям точніше аналізувати та інтерпретувати зображення для діагностики та планування лікування. Ну і останній фільтр – це фільтр змішування з альфа-маскою (CIBlendWithAlphaMask). Цей фільтр використовують для комбінування двох зображень із використанням альфа-маски. Альфа-маска визначає, які частини переднього плану мають бути видимими, а які прозорими. Це дає змогу додати деталі або текстуру з фонового зображення до переднього плану. Головні параметри цього фільтра охоплюють вхідне зображення (kCInputImageKey), фонове зображення (kCInputBackgroundImageKey) та альфа-маску (kCInputMaskImageKey). Комбінація цих зображень дає змогу створити змішане зображення з елементами переднього плану та деякими деталями фону. Для злиття зображень з використанням альфа-маски спочатку необхідно мати передній план, фонове зображення та альфа-маску.



Рис. 1. Загальна схема алгоритму

Під час злиття використовують альфа-маску, яка визначає, які частини переднього плану мають бути видимими, а які прозорими. Це дає змогу додати деталі або текстуру з фонового зображення до переднього плану. Фільтр злиття з альфа-маскою дозволяє комбінувати два зображення, забезпечуючи злиття їх пікселів за допомогою альфа-маски. Це дає змогу створювати складні композиції, додавати ефекти перекриття та нашарувань, а також комбінувати різні елементи зображень для досягнення бажаного візуального ефекту. Фільтр злиття з альфа-маскою можна використовувати для додавання текстур до зображень. Це особливо корисно в графічному дизайні, веброзробках та створенні спеціальних ефектів. Альфа-маску можна застосовувати для точного контролю над тим, як текстура накладається на основне зображення та взаємодіє з ним. Фільтр злиття із альфа-маскою може бути використаний для ретушування та коригування зображень. Ним можна скористатись для введення додаткових елементів, зміни фону, видалення небажаних об'єктів або ретушування недоліків. Альфа-маска дає змогу точно керувати областями змін і забезпечує плавний перехід між різними елементами зображення. Фільтр злиття з альфа-маскою застосовують також для оброблення відео. Його можна використовувати для створення ефектів перекриття, анімації, зміни фону, візуальних переходів тощо. Він дає змогу створювати складні візуальні композиції та досягати професійного вигляду відеоматеріалів. Загальну схему алгоритму наведено на рис. 1.

Вибираємо вхідне зображення. Перетворюємо зображення у формат CGImage, щоб отримати доступ до його пікселів та атрибутів. Створюємо CImage з CGImage для використання фільтрів з Core Image framework, контекст CIContext, який використовується для рендерингу зображень та взаємодії з Core Image. Застосовуємо фільтр “CIGaussianBlur” (гауссового розмиття) до вихідного зображення. Застосування фільтра “CIUnsharpMask” (непрозорого маскування) орієнтоване на посилення деталей на зображенні. Комбінування вихідного зображення із результатом непрозорого маскування відбувається за допомогою фільтра “CIBlendWith AlphaMask” (змішування з альфа-маскою). Це дає змогу поєднати оригінальне зображення з чіткішими деталями, отриманими після застосування непрозорого маскування. Наступний крок – рендеринг результуючого зображення за допо-

могою контексту CContext в CGImage із використанням вихідного розміру зображення. Надалі отримуємо на виході оброблене зображення.

Отже, алгоритм виконує послідовність операцій, ураховуючи розмиття, непрозоре маскування та змішування зображень для покращення вигляду та деталізації вхідного зображення.

Для дослідження розроблено мобільну програму, що є досить актуальним фактором. Нижче наведено фрагменти коду. Три базові фільтри об'єднано в один алгоритм для оброблення нечітких зображень.

```
let blurFilter = CIFilter(name: "CIGaussianBlur", parameters: [
    kCIInputImageKey: ciImage,
    kCIInputRadiusKey: 13 // Adjust this value to control the blur radius
])

guard let blurredImage = blurFilter?.outputImage else {
    return nil
}
```

Рис. 2. Гауссівське розмиття

```
// Apply unsharp mask to enhance details
let unsharpMaskFilter = CIFilter(name: "CIUnsharpMask", parameters: [
    kCIInputImageKey: ciImage,
    kCIInputRadiusKey: 4, // Adjust this value to control the radius of the unsharp mask
    kCIInputIntensityKey: 2 // Adjust this value to control the intensity of the unsharp mask
])

guard let unsharpMaskedImage = unsharpMaskFilter?.outputImage else {
    return nil
}
```

Рис. 3. Фільтр розширення контурів

```
// Combine the original image with the unsharp mask result using blend mode
let blendFilter = CIFilter(name: "CIBlendWithAlphaMask", parameters: [
    kCIInputImageKey: unsharpMaskedImage,
    kCIInputBackgroundImageKey: blurredImage,
    kCIInputMaskImageKey: ciImage
])

guard let blendedImage = blendFilter?.outputImage else {
    return nil
}

// Render the result
guard let resultCGImage = context.createCGImage(blendedImage, from: CGRect(x: 0, y: 0, width: width, height: height))
else {
    return nil
}
```

Рис. 4. Фільтр змішування з альфа-маскою

Алгоритм фільтра “CIGaussianBlur” (рис. 5) застосовує гауссівське розмиття до зображення. Гауссівське розмиття є методом згладжування зображення із поширенням яскравості пікселів за допомогою гауссової функції. Під час виконання алгоритму для кожного пікселя встановлюються параметри радіуса розмивання та інтенсивності. Потім обчислюють вагові коефіцієнти для кожного пікселя, використовуючи гауссову функцію. Ці коефіцієнти впливають на значення яскравості пікселя у результаті розмивання. Завдяки застосуванню фільтра “CIGaussianBlur” зображення стає розмитішим і м’якшим, сприяючи зменшенню шуму та покращенню загального вигляду.

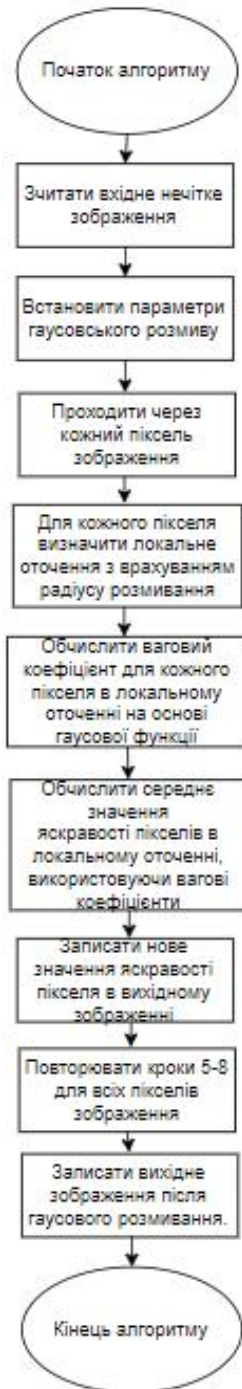


Рис. 5. Схема алгоритму фільтра "CIGaussianBlur"

Наступна схема алгоритму (рис. 6) містить послідовність операцій, які виконують, застосовуючи фільтр "CIUnsharpMask". Спочатку на вхідному зображенні здійснюють гауссівське розмивання, щоб створити розмиту версію. Потім цю версію віднімають від оригінального зображення, щоб отримати різницю. Наступний крок – підсилення контрасту в цій різниці, щоб підкреслити деталі. На виході отримують вихідне зображення з підсиленим контрастом.

Деталізована схема алгоритму для фільтра "CIBlendWithAlphaMask" (рис. 7) показує послідовність операцій, які виконують, застосовуючи фільтр. На вхід подається вхідне зображення 1, яке буде видимим у результатуючому зображенні, а також альфа-маска, що визначає прозорість різних ділянок зображення.

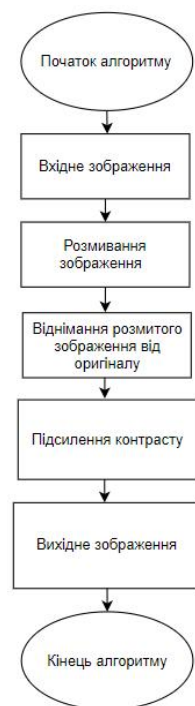


Рис. 6. Схема алгоритму фільтра "CIUnsharpMask"



Рис. 7. Схема алгоритму фільтра "CIBlendWithAlphaMask"

Вхідне зображення 2 також використовується для змішування.

Виконується змішування вхідного зображення 1 та вхідного зображення 2 на основі значень альфа-маски. Це означає, що в областях з альфа-значенням 1 буде використовуватись піксель із вхідного зображення 1, а в областях з альфа-значенням 0 – піксель з вхідного зображення 2. На виході отримуємо вихідне зображення, яке є результатом змішування двох вхідних зображень з урахуванням альфа-маски.

Цей фільтр дає змогу змішувати два зображення з урахуванням прозорості, створюючи ефекти переходів та змішування зображень у багатьох графічних програмах та редакторах.

Результати виконання тестової програми наведено на рис. 8–12.

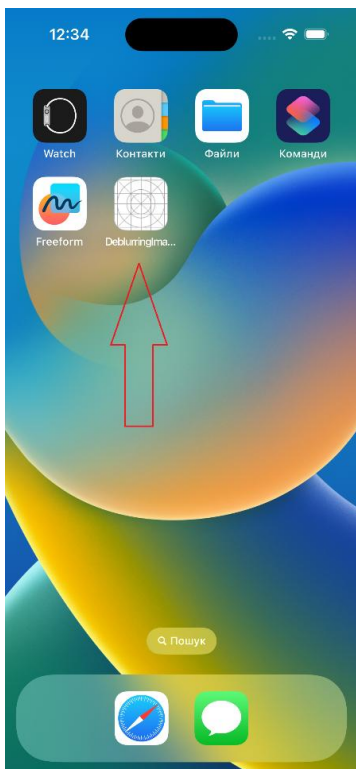


Рис. 8. Іконка програми

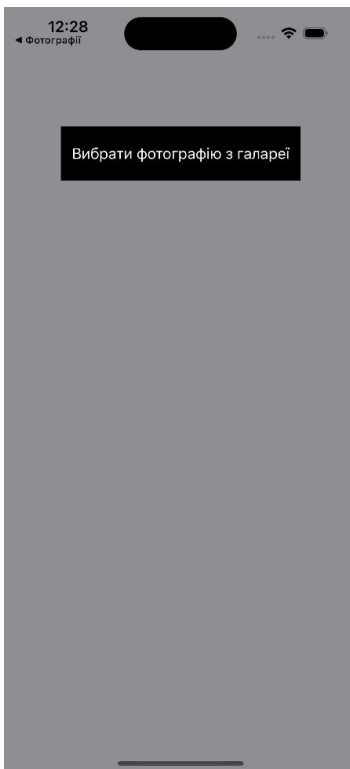


Рис. 9. Головне меню

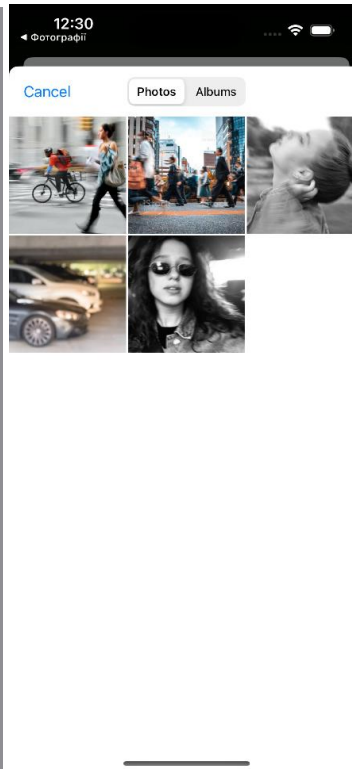


Рис. 10. Вибір зображення для подальшої обробки

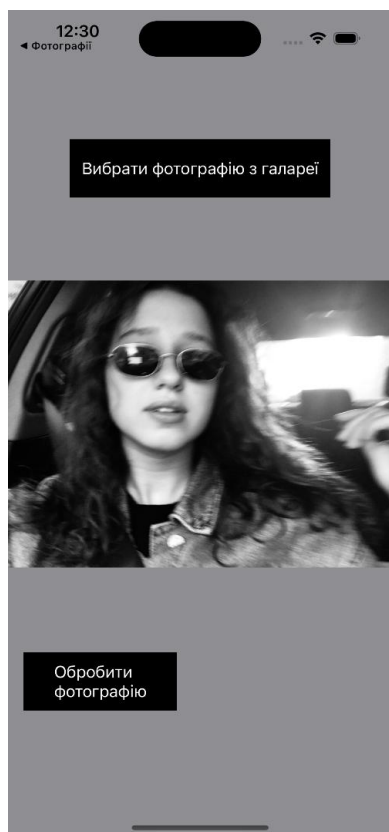


Рис. 11. Початкове зображення



Рис. 12. Зображення після опрацювання

Для тестування використано різні зображення. Продемонстровано роботу запропонованого алгоритму на реальних зображеннях. Одне з них подано на рис. 11.

В розробленій програмі використано дві метрики для оцінювання якості опрацювання нечітких зображень. Одна з них обчислює середньоквадратичну помилку між пікселями двох зображень RMSE. Ця метрика вимірює загальну різницю між інтенсивністю пікселів і дає змогу оцінити, наскільки вони відрізняються один від одного. Чим нижче значення RMSE, тим більш схожі зображення. Друга метрика обчислює відсоток різних пікселів між двома зображеннями. Вона порівнює значення кожного пікселя у двох зображеннях і підраховує, скільки пікселів відрізняються. Ця метрика вказує на загальний ступінь різниці між зображеннями. Чим більше значення піксельної різниці, тим більше відрізняються зображення. Тобто чим більша відмінність між оригінальним зображенням і кінцевим, тим краще оброблено зображення. Також зафіксували час на початку виконання алгоритму та під час завершення виконання, щоб дізнатися час виконання алгоритму для різних роздільних здатностей і розмитостей зображень. Нижче наведено таблицю результатів після опрацювання запропонованим алгоритмом різних зображень.

Результати опрацювання різних зображень

№ з/п	Роздільна здатність, пікселів	Час опрацювання, с	Різниця RMSE	Різниця пікселів
1	736×1104	0,12	19,6	2,3
2	1280×962	0,12–0,13	26,7	2,6
3	608×406	0,07	36,6	2,75

Наведені в таблиці результати підтверджують ефективність застосування запропонованого алгоритму.

Висновки

За результатами досліджень та тестування зроблено висновки, що запропонований алгоритм:

- опрацьовує зображення до 20 % швидше, ніж із застосування фільтра “Motion blur”;
- несуттєво відрізняється щодо часу виконання за різної роздільної здатності зображення;
- ефективний для різних типів розмитостей;
- простий в реалізації та мінімізований за ресурсами.

Подальші вдосконалення запропонованого алгоритму полягають у визначенні раціональних значень параметрів для фільтрів, щоб максимально його адаптувати до різних типів зображень та розмитостей.

Список літератури

[1] Hryshko B. O., Sharov S. B. Rozrobka prohramnoho zasobu dlya obrobky tsyfrovoyh zobragen. *Ukrainian Journal of Educational Studies and Information Technology*, Vol. 5. No. 2, June 2017, 46–49. Available at: <https://journals.indexcopernicus.com/api/file/viewByFileId/171586.pdf/> / Accessed: 10 October 2023]

[2] Zorilo V. V., Karpova O. A. *Algorytm vyjavlennya obrobky tsyfrovoho zobragennya filtrom Motion blur*”, *Informatyka ta matematychni metody v modeluvanni*. Odessa National Polytechnic University, 2019, T. 9, No. 1-2, 49–58. Available at: <http://dspace.opu.ua/jspui/handle/123456789/9125/> / Accessed: 10 October 2023.

[3] Reida O. M., Oliinyk U. V., Panchuk A. O., Synenkyi M. L. *Methody polipshennia tsyfrovoho zobragennia ta vidnovlennia ioho struktury*. *Naukovi pratsi VNTU*, 2010, No. 4. Available at: <https://praci.vntu.edu.ua/index.php/praci/article/view/232/230/> / Accessed: 10 October 2023]

[4] Paramud Y., Yarkun V. *Metod rozpoznavannya symboliv na zobragennyakh na osnovi zhortkovoii neiironnoi meregi*, *Transactions on Computer systems and networks*, Lviv Polytechnic National University Press, 2018, No. 905. 96–105 (in Ukrainian). DOI:10.23939/csn2018.905.096.

[5] Jian-Feng Cai, Raymond H. Chan, Mila Nikolova, “Two-phrase approach for deblurring images corrupted by impulse plus gaussian noise”, *CMLA, ENS Cachan, CNRS, PRES UniverSud 61 av. du Pr’esident Wilson, 94235*

Cachan Cedex, France, *Inverse Problems and Imaging*, Vol. 2, No. 2, 2008, 187–204. Available at: [c20ef5248c70599b6ebdbadb11756ec9311.pdf](https://arxiv.org/abs/20ef5248c70599b6ebdbadb11756ec9311) ([semanticscholar.org](https://www.semanticscholar.org))

[6] Yuhang Liu, Wenyong Dong, Dong Gong, Lei Zhang, Qinfeng Shi, “Deblurring Natural Image Using Super-Gaussian Fields”, Computer School, Wuhan University, Hubei, China, School of Computer Science, The University of Adelaide, Adelaide, Australia, 2018. Available at: *Deblurring Natural Image Using Super-Gaussian Fields* (thevcf.com)

[7] Sainandan Ramakrishnan, Shubham Pachori, Aalok Gangopadhyay, Shanmuganathan Raman, “Deep Generative Filter for Motion Deblurring”, Jijabai Technological Institute, Mumbai - 4000311 Indian Institute of Technology, Gandhinagar – 3823552, 2017. Available at: *Deep Generative Filter for Motion Deblurring* (thevcf.com)

[8] Paramud Y., Borovets D., Pavych T. Computer system for converting gestures to text and audio messages // *Advances in Cyber-Physical Systems*, 2021, Vol. 6, No. 2, 90–97. DOI: <https://doi.org/10.23939/acps2021.02.090>.

[9] Kataieva I. U., Breus B. V. Systemnyi analiz methodiv obrobky zobrazen. Cherkaskyi dergavnyi technologichnyi universitet, 2022. Available at: *Eurasian Scientific Discussions 13-15.02.22.pdf* (librarynmu.com)

FEATURES OF ALGORITHMIC AND SOFTWARE TOOLS FOR FRAMING FUZZY IMAGES

R. Yarkun, Y. Paramud

Lviv Polytechnic National University,
Computer Engineering Department

© Yarkun R., Paramud Y., 2023

This article examines the features of algorithmic and software tools for processing fuzzy images. The work uses three filters: CIGaussianBlur, CIUnsharpMask and CIBlendWithAlphaMask. The described filters allow you to improve image quality, reduce noise and reproduce details. The initial task is to process the blurring of images. For this, the CIGaussianBlur filter is used, which applies a Gaussian blur to the image. This blur reduces high-frequency noise and adds smoothness to the contours of objects. The second filter, CIUnsharpMask, is used to restore image details. This filter subtracts the blurred version from the original image, which allows you to highlight important details and increase the clarity of the image. The last filter, CIBlendWithAlphaMask, is used to blend two images using an alpha mask. This filter allows you to control the transparency and adjust how the images are blended. As a result, a more realistic and aesthetic image can be achieved.

The article considers the principles of operation of each of the filters, gives examples of their use and describes the results obtained. Research shows that using these filters can improve the quality of blurry images, reduce noise, and sharpen details.

The results of this work can be useful for use in the field of image processing, computer vision and graphic design. Using the described filters can help improve the visual characteristics of images and provide a more accurate interpretation of fuzzy data.

Key words: fuzzy images; algorithmic tools; software tools; image processing; filters.