

### ДОСЛІДЖЕННЯ ВПЛИВУ ВИБОРУ МЕТОДА ПОПЕРЕДНЬОЇ ОБРОБКИ НА РЕЗУЛЬТАТ АНАЛІЗУ ЗОБРАЖЕННЯ

**Т. А. Григорова, С. С. Жук**

Кременчуцький національний університет імені Михайла Остроградського  
вул. Першотравнева, 20, м. Кременчук, 39600, Україна. E-mail: gasta1@yandex.ua

Розглянуто проблему, що пов'язана з аналізом кольорового зображення методом кластеризації для ідентифікації кольорів і побудови їх розподілу на зображенні, згідно якого проводиться аналіз якості і особливостей структури волокно-утворюючого полімеру. Для дослідження впливу методів фільтрації на результати кластерної обробки зображення було розроблено аналітичний програмний продукт для проведення експерименту, який дозволяє виділяти необхідну область на зображенні, видаляти задній фон, який не потрібен для аналізу, використовувати один з визначених методів фільтрації (білатеральна, гаусівська, медіанна та розмиття), проводити кластеризацію з різними варіантами та показувати знайдені кольори на зображенні. Під час дослідження, що проводилось із зображенням інтерференційної картини волокна було виявлено, що кластеризація сильно залежить від попередньої обробки зображення, від самого зображення та кількості кольорів, що його складають. Було виявлено, що попередню обробку зображення для аналізу волокна методом кластеризації краще виконувати за допомогою медіанної фільтрації, яка показала найкращі результати у порівнянні з іншими методами.

**Ключові слова:** фільтрація зображення, кластерна обробка, комп'ютерний експеримент.

### ИССЛЕДОВАНИЕ ВЛИЯНИЯ ВЫБОРА МЕТОДА ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ НА РЕЗУЛЬТАТ АНАЛИЗА ИЗОБРАЖЕНИЯ

**Т. А. Григорова, С. С. Жук**

Кременчугский национальный университет имени Михаила Остроградского  
ул. Первомайская, 20, г. Кременчуг, 39600, Украина. E-mail: gasta1@yandex.ua

Рассмотрена проблема, связанная с анализом цветного изображения методом кластеризации для идентификации цветов и построения их распределения на диаграмме, согласно которого проводится анализ качества и особенностей структуры волоконно-образующего полимера. Для исследования влияния методов фильтрации на результат кластерной обработки изображения был разработан аналитический программный продукт для проведения эксперимента, который позволяет выделять необходимую область на картинке, удалять задний фон, не нужный для анализа, использовать один из определенных методов фильтрации (билатеральная, гауссова, медианная и размытие), проводить кластеризацию с различными вариантами и показывать найденные на изображении цвета. В ходе исследования, проведенного с изображением интерференционной картины волокна, было обнаружено, что кластеризация сильно зависит от предварительной обработки изображения, от самого изображения и количества цветов из которых оно состоит. Было обнаружено, что предварительную обработку изображения для анализа волокна методом кластеризации лучше выполнять с помощью медианной фильтрации, которая показала лучшие результаты по сравнению с другими методами.

**Ключевые слова:** фильтрация изображения, кластерная обработка, компьютерный эксперимент.

**АКТУАЛЬНІСТЬ РОБОТИ.** При вирішенні задач обробки зображень волокон і ниток, отриманих за допомогою модуля цифрової камери «Експерт» та представлених в форматі BMP, перешкодами виступають компоненти самого зображення. Для того, щоб послабити дії перешкод перед проведенням аналізу зображення застосовується фільтрація. Під час фільтрації яскравість кожної точки вихідного зображення, спотвореного перешкодою, замінюється іншим значенням яскравості. Основа фільтрації полягає в раціональному використанні даних, як фільтрованої точки, так і її околиці.

Вибір фільтрації зводиться до пошуку алгоритму, який в залежності від поставленого завдання дасть найкращий результат. Для опису методів і алгоритмів, використовуються різні математичні моделі, аналіз яких не завжди дозволяє зробити правильний вибір. В цьому випадку вибір методу виконується експериментальним шляхом. Щоб уникнути зайвої роботи, яка пов'язана з експериментом, для його проведення необхідно використовувати програмний продукт за допомогою якого, з'являється можливість підбирати методи фільтрації та аналізу для різного типу зображень. В програ-

мі для аналізу зображення використовується кластеризація - один з методів сегментації зображень, яка є найбільш ефективною коли йде мова про отримання кольорових спектрів зображення. Вона також застосовується в пошуковому інтелектуальному аналізі даних, в загальній методиці аналізу статистичних даних, використовується в багатьох областях, в тому числі машинного навчання, розпізнавання образів, аналізу зображень, пошуку інформації, біоінформації, стисненні даних і комп'ютерній графіці.

Метою роботи було встановлення причини розбіжності візуальної та комп'ютерної оцінки зображення і пошук рішення щодо усунення знайденого дефекту. Для проведення експерименту щодо впливу методів фільтрації на результати кластерної обробки зображення потрібно розробити програмний продукт для проведення аналізу.

**МАТЕРІАЛ І РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ.** Відомо, що більшість текстильних волокон мають анізотропну (неоднорідну) структуру [1], тому при опроміненні поляризаційним світлом вони інтерференційно фарбуються. За кольором на малюнку інтерференційної картини здійснюється ідентифікація волокон [2]. Малюнок інтерференційного фарбу-

вання волокна залежить від якості і особливостей структури волокно-утворюючого полімеру [3]. Волокна з низьким ступенем кристалічності та з рівномірною структурою мають однорідне забарвлення. Волокна, що з високим ступенем кристалічності і орієнтованістю структурних елементів мають малюнок у вигляді симетричних смуг різної ширини і кольору. Волокна, структура яких раз-орієнтована, не мають вираженого малюнка і пофарбовані не рівномірно у вигляді плям. Візуальна оцінка інтерференційної картини дозволяє проводити міжвидову ідентифікацію волокон. Комп'ютерна ідентифікація волокон включає в себе наступні етапи:

- виділення області зображення для проведення аналізу;
- процес локалізації волокна за допомогою видалення фону;
- попередню обробку виділеної області фільтром, що згладжує для усунення зашумлення і підвищення якості зображення;
- аналіз колірної маски зображення за допомогою виділення колірних регіонів;
- виконання розрахунків площ для кожного колірного регіону;
- збереження отриманих результатів в базу даних;

виконання статистичної обробки шляхом побудови гістограми процентного співвідношення площ колірних регіонів до загальної площі виділеної області зображення волокна.

Для реалізації перерахованих етапів було розроблено спеціальну програму «Fiber Analize», в якій отримання аналізу зображення у вигляді гістограми спрощує процес ідентифікації волокон і виключає людський фактор при формуванні висновку.

При аналізі гістограм було виявлено, що на різних волокнах при однаковій яскравості освітлення поля зору мікроскопа, інтенсивність кольорів поляризаційного фарбування може відрізнятися. На це вказує поява на деяких гістограмах категорій чорного і сірого кольору. В результаті в програмі було передбачено видалення небажаного колірного регіону і відповідно при виконання розрахунків площа цього регіону не враховується.

В ході експлуатації програми було з'ясовано, що програма не зовсім коректно оцінює колірні характеристики волокна і відповідно не коректно виконується міжвидова ідентифікація волокон. Всі волокна в залежності від якості структури розрізняють за кольоровими характеристиками. Якісні волокна з високим ступенем кристалічності і орієнтованістю структурних елементів мають малюнок у вигляді широкої коричневої смуги по центру волокна і різної ширини та кольору (червоний, синій, жовтий, зелений) смуг по краях. Менш якісне волокно має більш широку смугу зеленого кольору. При цьому волокно може мати вкраплення коричневого кольору. Раз-орієнтовані волокна мають забарвлення у вигляді плям різних кольорів, це говорить про те, що структура волокна починає руйнуватися. При цьому в залежності від площі колірних регіонів встановлюється ступінь руйнування волокна. В

результаті порівняння комп'ютерного і візуального аналізу зображення було помічено, що оцінка якості, яка була отримана у вигляді гістограми для менш якісного волокна давала переважання коричневого кольору перед зеленим, що не відповідало візуальній оцінці зображення. Щоб з'ясувати причину отримання розбіжностей результатів роботи програми і візуальної оцінки було розроблено аналітичну програму, що дозволила визначити вплив методу фільтрації на подальшу обробку і аналіз кольору регіонів. Для цього було проаналізовано і виявлено найбільш вживані методи фільтрації для згладжування зображень. До них відносяться: лінійне розмиття, Гауса, медіанний та білатеральний. Всі вони відрізняються один від одного своїми параметрами і вихідними результатами.

При згладжуванні зображень використовують апертуру – це прямокутна або квадратна ділянка зображення, на якій визначається функція. Задана функція в апертурі називається ваговою, або функцією вікна, а сама апертура разом з цією функцією називається маскою зображення.

Лінійні фільтри усереднюють значення пікселів, які знаходяться в апертурі шляхом підбору розміру апертури, що впливає саме на шум. Розмиття [4] представляє простий фільтр, у якому значення кожного вихідного пікселя є середнім значенням пікселів його сусідів (всі вони вносять свій внесок з рівною вагою). Маска для розмиття представлена формулою:

$$K = \frac{1}{K_{width}K_{height}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \quad (1)$$

Фільтр Гауса – це метод фільтрації зображення за допомогою функції Гауса [4]. Функція Гауса використовується для розрахунку трансформації кожного пікселя у зображенні. Одновимірна функція Гауса має вигляд:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}, \quad (2)$$

де  $x$  – це відстань від початку координат на осі абсцис,  $y$  – це відстань від початку координат на осі ординат, а  $\sigma$  є параметр розподілу Гауса. Коли метод застосовується у двох вимірах, отримується поверхня, контури якої є концентричні кола розподілу Гауса з центральною точкою. Значення з цього розподілу використовуються для створення матриці згортки. Для кожного нового значення пікселя визначається середнє зважене в околиці пікселя. Значення поточного оригінального пікселя має більшу вагу (найвище значення розподілу Гауса), а сусідні пікселі отримують меншу вагу в залежності від відстані до поточного оригінального пікселя.

Це надає ефект розмитості, яка зберігає межі краще, ніж інші аналогічні фільтри розмиття.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \quad (3)$$

Рівняння (3) дозволяє обчислювати коефіцієнти для шаблону Гауса, який потім згортається із зображенням. Функція Гауса істотно усуває вплив точок, які більше, ніж  $3\sigma$  віддалені від центру шаблону. Цей фільтр було використано у програмі «Fiber Analize».

Медіанний фільтр [4] – один з видів цифрових фільтрів, що широко використовується в цифровій обробці сигналів та зображень для зменшення рівня шуму. Він використовується для видалення імпульсного шуму на зображеннях. Сам імпульсний шум представляє собою заміну частини пікселів на зображенні значеннями фіксованої або випадкової величини, які ізольовані контрастними точками. Фільтр побудований на основі рангової статистики. Алгоритм обробки зображення полягає у визначенні позиції кожного імпульсу і заміні його значенням оцінки, а під час збереження інших пікселів зображення вони залишаються незмінними. Але значення пікселів, які відповідають масці, записуються в ряд. В результаті сортування цього ряду результатом фільтру буде центральний елемент в відсортованій послідовності, що і є медіаною ряду. Значення рангу усередині вікна фільтра сортується в порядку зростання (спадання); і значення, що знаходиться в середині упорядкованого списку, надходить на вихід фільтра. У разі парного числа рангу у вікні вихідне значення фільтра дорівнює середньому значенню двох рангів в середині упорядкованого списку. Вікно переміщується уздовж сигналу, що фільтрується, і обчислення повторюються.

Білатеральний фільтр [0] є нелінійним і шумозахисним згладжувальним фільтром для зображень. Він відносно простий. Основна відмінність білатеральної фільтрації від інших фільтрів згладжування шуму полягає в тому, що вона враховує не лише область спостереження зображення, тобто сусідство пікселів, але і діапазон значень пікселів.

Фільтр замінює значення пікселя середнім значенням подібних величин і значень прилеглих пікселів. У гладких областях він в основному працює як фільтр стандартної області. На краях множник нормування виключає пікселі з іншого боку краю.

Фільтр має наступний вигляд [5]:

$$\tilde{f}(x) = g^{-1}(x) \iint f(\delta) q(\delta, x) s(f(\delta), f(x)) d\delta \quad (4)$$

з множником нормування  $g(x)$  вигляду:

$$g(x) = \iint q(\delta, x) s(f(\delta), f(x)) d\delta,$$

де  $\tilde{f}(x)$  – значення пікселя  $x$  у вихідному зображенні,  $f(x)$  – значення пікселя  $x$  у вхідному зображенні,  $q(\delta, x)$  – геометрична близькість, відстань між центром околиці  $x$  і сусідньою точкою  $\delta$ ,  $s(f(\delta), f(x))$  – близькість значень пікселів, діапазон значень пікселів між центром околиці  $x$  і сусідньою точкою  $\delta$ .

Для видалення шуму також використовується піраміда зображення, яка являє собою набір зображень з одного вихідного зображення, що послідовно зменшують дискретизацію поки яка-небудь потрібна точка зупинки не буде досягнута [6].

Після застосування піраміди до зображення в градієнтах сірого необхідно використати фільтр Кенні [7], який призначений для знаходження границь об'єктів.

Застосування кластерного аналізу в загальному вигляді зводиться до наступних етапів:

1. Відбір вибірки об'єктів для кластеризації.
2. Визначення безлічі змінних, за якими будуть оцінюватися об'єкти у вибірці. При необхідності – нормалізація значень змінних.
3. Обчислення значень міри схожості між об'єктами.
4. Застосування методу кластерного аналізу для створення груп схожих об'єктів (кластерів).
5. Представлення результатів аналізу.

Можна виділити дві основні класифікації алгоритмів кластеризації: ієрархічні і плоскі.

Ієрархічні алгоритми (також звані алгоритмами таксономії) будують не одне розбиття вибірки на непересічні кластери, а систему вкладених розбиттів. Тобто на виході ми отримуємо дерево кластерів, коренем якого є вся вибірка, а листям – найбільш дрібні кластери. У програмі «Fiber Analize» використовується статистичний ієрархічний агломеративний алгоритм [8].

Плоскі алгоритми будують одне розбиття об'єктів на кластери і поділяються на чіткі і нечіткі.

Чіткі (або непересічні) алгоритми кожному об'єкту вибірки ставлять у відповідність номер кластера, тобто кожен об'єкт належить тільки одному кластеру. Нечіткі (або пересічні) алгоритми кожному об'єкту ставлять у відповідність набір дійсних значень, що показують ступінь відношення об'єкта до кластерів. Тобто кожен об'єкт відноситься до кожного кластера з певною ймовірністю.

Ієрархічний алгоритм, що використовується в програмі дає систему повного розбиття, щоб можна було обрати необхідні кольори для аналізу, але це не потрібно для нашого дослідження, що стосується виявлення впливу методу фільтрації на результат. Було прийнято рішення використати плоский алгоритм кластеризації.

До типу плоских алгоритмів відносяться алгоритми квадратичної помилки. Завдання кластеризації можна розглядати як побудову оптимального розбиття об'єктів на групи. При цьому оптимальність може бути визначена як вимога мінімізації середньоквадратичної помилки розбиття [9]:

$$e^2(X, L) = \sum_{j=1}^K \sum_{i=1}^{n_j} \|x_i^{(j)} - c_j\|^2, \quad (5)$$

де  $c_j$  – «центр мас» кластера  $j$  (точка з середніми значеннями характеристик для даного кластера).

Найпоширенішим алгоритмом цієї категорії є метод  $k$ -середніх [9].

Мета методу – розділити  $n$  спостережень на  $k$  кластерів, так щоб кожне спостереження належало до кластера з найближчим до нього середнім значенням. Метод базується на мінімізації суми квадратів відстаней між кожним спостереженням та центром його кластера, тобто функції (5):

$$\sum_{i=1}^N d(x_i, m_j(x_i))^2, \quad (6)$$

де  $d$  – метрика,  $x_i$  –  $i$ -й об'єкт даних, а  $m_j(x_i)$  – центр кластера, якому на  $j$ -й ітерації приписаний елемент  $x_i$ .

1. Вибрати  $K$  центрів кластерів, випадково або на основі деякої евристики.

2. Помістити кожен піксель зображення в кластер, центр якого найближче до цього пікселя.

3. Знову визначити центри кластерів, усереднюючи всі пікселі в кластері.

4. Повторювати кроки 2 і 3 до збіжності (наприклад, коли пікселі будуть залишатися в тому ж кластері).

Тут за відстань зазвичай береться сума квадратів або абсолютних значень різниць між пікселем і центром кластера. Різниця зазвичай базується на кольорі, яскравості, текстурі і місці знаходження пікселя, або на зваженій сумі цих чинників.  $K$  може бути вибране вручну, випадково чи евристично.

Для даного алгоритму необхідно задавати кількість кластерів для розбиття.

Для створення повнофункціональної програми необхідно було обрати бібліотеку з широким набором можливостей. Вибір пав на OpenCV (англ. Open Source Computer Vision Library, бібліотека комп'ютерного зору з відкритим кодом) – бібліотеку функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. Бібліотека надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів на фотографіях (наприклад, осіб і фігур людей, тексту тощо), відстежування руху об'єктів, перетворення зображень, застосування методів машинного навчання і виявлення загальних елементів на різних зображеннях [Ошибки! Источник ссылки не найден.].

У якості мови розробки використана мова C#. Для взаємодії коду, написаному на C++ та C#, застосовується крос-платформова .NET обгортка Emgu CV [11] для бібліотеки OpenCV.

У якості API (англ. Application Programming Interface, інтерфейс програмування додатків) було обрано WPF (Windows Presentation Foundation), у якому використовується векторна система візуалізації, яка не залежить від роздільної здатності монітору і враховуються можливості сучасного графічного обладнання.

Для зчитування зображення використано клас, який стандартно використовується для зчитування зображень у Emgu CV та його конструктор, який приймає у якості параметру строкове ім'я файлу.

```
[SerializableAttribute]
public class Image<TColor, TDepth> :
    CvArray<TDepth>,
    IImage, IDisposable, ICloneable,
    IInputOutputArray,
    IInputArrayOfArrays,
    IOutputArrayOfArrays,
    IInputArray, IEquatable<Image<TColor,
    TDepth>>
where TColor : struct, new(), IColor
where TDepth : new()
...
```

```
public Image(string fileName)
```

Для відображення зображення на екрані за допомогою WPF використано фрагмент коду для перетворення його із зображення Image у BitmapSource [0].

Коли зображення зчитано, необхідно виділити область, що цікавить. Для зручності обрано використання виділення за допомогою клавіатурної миші. Для цього призначений вбудований клас Rectangle та події натискання, руху та відпускання миші.

Для того, щоб можна було видалити фон, необхідно отримати маску з розділенням переднього та заднього плану.

Спочатку зображення перетворюємо у градієнти сірого за допомогою методу Convert:

```
public Image<TOtherColor,
TOtherDepth> Convert<TOtherColor,
TOtherDepth>()
where TOtherColor : struct, new(),
IColor
where TOtherDepth : new()
```

Потім необхідно застосувати піраміду зображення для зменшення шуму. Ця процедура (зменшення та збільшення дискретизації) реалізована у функціях OpenCV: PyrUp та PyrDown.

Після їх застосування до зображення в градієнтах сірого необхідно використати фільтр Кенні [10], який призначений для знаходження границь об'єктів.

Рекомендоване значення відношення верхнього та нижнього порогу знаходиться між 2:1 та 3:1. У Emgu CV метод приймає два значення порогів:

```
public Image<Gray, byte> Canny(
    double thresh,
    double threshLinking
)
```

У Emgu CV є метод для виконання морфологічних операцій

```
public static void cvMorphologyEx(
    IntPtr src,
    IntPtr dst,
    IntPtr temp,
    IntPtr element,
    CV_MORPH_OP operation,
    int iterations
)
```

де src, dst – вхідне та вихідне зображення відповідно; temp – тимчасове зображення, у даному випадку непотрібно; element – примітив на вибір; operation – у даному випадку використано закриття, iterations – кількість ітерацій виконання методу, обирається у результаті експерименту.

Отримане зображення треба інвертувати та додати до початкового для отримання зображення без фону.

Для фільтрації зображення у бібліотеці OpenCV є декілька фільтрів на вибір [0]. У програмі використовуються на вибір білатеральний фільтр, розмиття, фільтр Гауса та медіанний фільтр:

```
public Image<TColor, TDepth>
SmoothBilateral(int kernelSize,
               int colorSigma, int
               spaceSigma);
public Image<TColor, TDepth>
SmoothBlur(int width, int height);
public Image<TColor, TDepth>
SmoothGaussian(int kernelSize);
public Image<TColor, TDepth>
SmoothMedian(int size);
```

де kernelSize – розмір ядра для фільтра; colorSigma – стандартне відхилення у кольоровому просторі; spaceSigma – стандартне відхилення в просторі координат (у пікселях); width, height – ширина та висота; size – розмір вікна.

Для кластеризації у бібліотеці OpenCV передбачена спеціальна функція [Ошибк! Источник ссылки не найден.]. Її реалізація у Emgu CV представлена у наступному фрагменті коду:

```
public static int cvKMeans2(
    IntPtr samples,
    int clusterCount,
    IntPtr labels,
    MCvTermCriteria termcrit,
    int attempts,
    IntPtr rng,
    KMeansInitType flags,
    IntPtr centers,
    IntPtr compactness
)
```

де samples – вхідна матриця пікселів; clusterCount – кількість кластерів для розділення; labels – вихідний вектор зберігання індексів кластера; termcrit – задає максимальну кількість ітерацій і/або точності; attempts – кількість спроб, використовується дві; rng – покажчик на CvRNG, не використовується; flags – тип ініціалізації k-means; centers – покажчик на масив центрів кластерів; compactness – покажчик на масив double, не використовується.

У програмі створено повзунок для задання значення clusterCount та реалізований вибір flags:

1. k-means++ – покращена версія алгоритму кластеризації k-середніх. Суть покращення полягає в знаходженні більш «гарних» початкових значень центроїдів кластерів. Оригінальний k-means не регламентує те, як виконується цей етап алгоритму, і тому є нестабільним. Алгоритм запропонований в 2007 році Девідом Артуром і Сергієм Вассільвітським [15];

2. випадкові центри – на відміну від k-means++ задаються випадковим чином;

3. початкові мітки, які задаються за допомогою параметру labels.

Для наочності результату виникла необхідність виділяти обраний колір на відкластеризованому зображенні. За допомогою Windows Forms це неможливо реалізувати простим шляхом, зате на WPF за допомогою елементу Canvas можна малювати поверх будь-якого елементу інтерфейсу.

Для аналізу було використане зображення інтерфренційного зафарбованого волокна рис. 1.

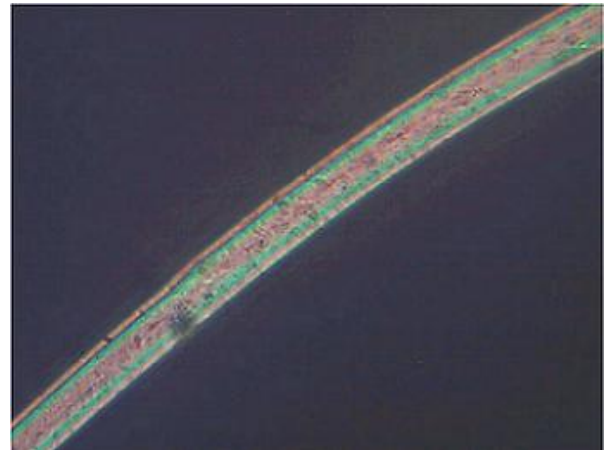


Рисунок 1 – Зображення для аналізу

Для створення програми було використано сучасний шаблон Modern UI [16]. Інтерфейс програми представлений на рис. 2.

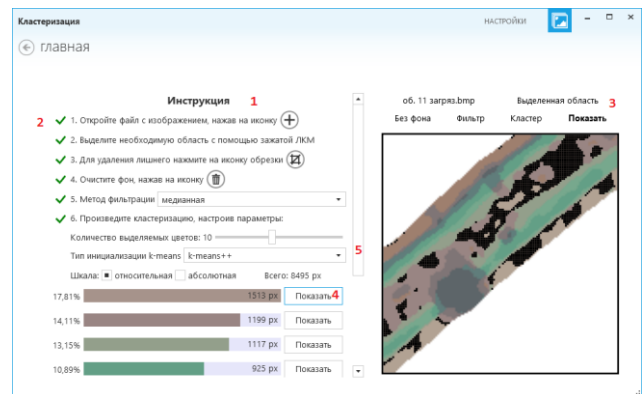


Рисунок 2 – Інтерфейс програми

1. У лівій частині екрану наведена інструкція для користувача;

2. Успішне виконання кроків відмічається зеленою галкою;

3. Етапи виконання можна перемикаати;

4. Для виділення визначеного кольору є кнопки, при наведенні на колір у лівій частині екрану, буде показана підказка з кольором у моделі RGB.

5. Стрічку можна переміщувати для доступу до потрібної частини, або розвернути програму на весь екран для вміщення всіх елементів у полі зору.

Представимо результати аналізу для білатерального фільтру:



Рисунок 3 – k-means++ (білатеральний)



Рисунок 4 – Випадкові центри (білатеральний)

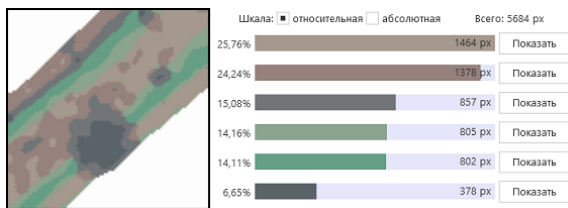


Рисунок 5 – Початкові мітки (білатеральний)

k-means++ та випадкові центри показали приблизно однакові результати, початкові мітки по іншому відреагували на відтінки сірого. Всі методи ініціалізації k-середніх виявили 5684 пікселя.

Представимо результати аналізу для розмиття:

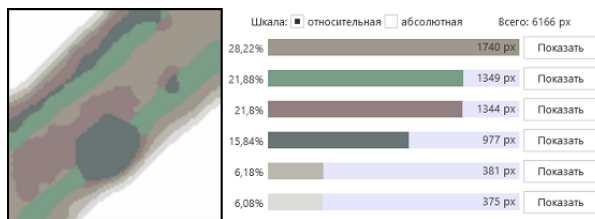


Рисунок 6 – k-means++ (розмиття)



Рисунок 7 – Випадкові центри (розмиття)

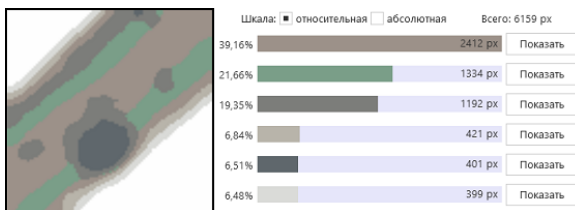


Рисунок 8 – Початкові мітки (розмиття)

Відтінки зеленого злилися в один колір, внаслідок розмиття на краях зображення з'явилися неpravильні додаткові кольори.

Всі методи ініціалізації k-середніх показали різні результати та виявили більшу кількість зайвих пікселів. Представимо результати аналізу для фільтру Гауса:

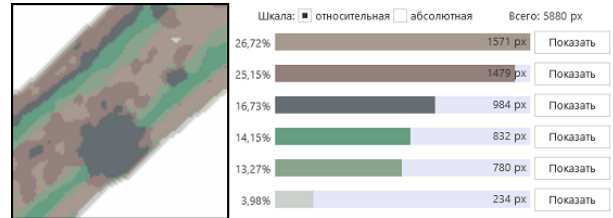


Рисунок 9 – k-means++ (Гаус)

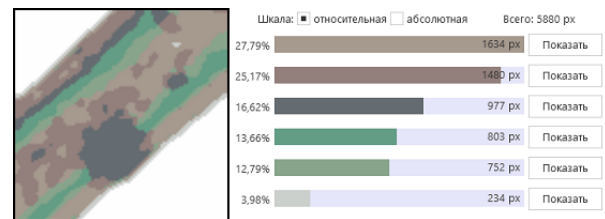


Рисунок 10 – Випадкові центри (Гаус)

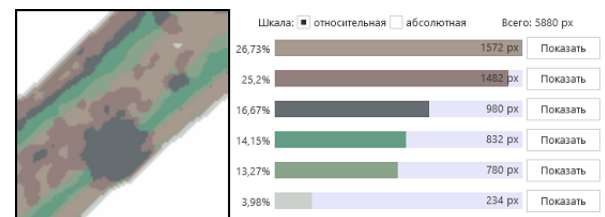


Рисунок 11 – Початкові мітки (Гаус)

Фільтр Гауса як і розмиття додав до зображення зайвих пікселів, але у меншій кількості. У порівнянні з білатеральним фільтром два основні кольори помінялись місцями.

Всі методи ініціалізації k-середніх показали приблизно однакові результати та виявили 5880 пікселів. Представимо результати аналізу для медіанного фільтру:

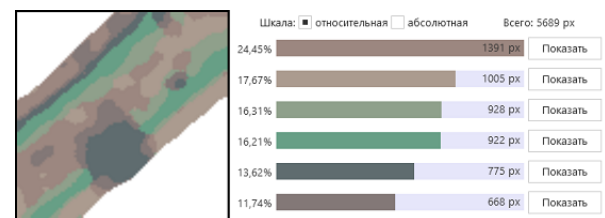


Рисунок 12 – k-means++ (медіанний)

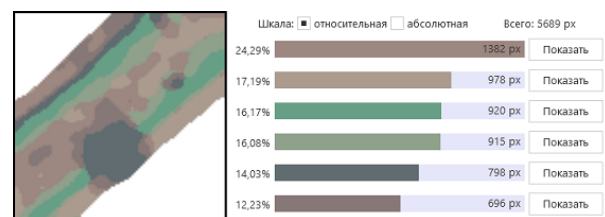


Рисунок 13 – Випадкові центри (медіанний)

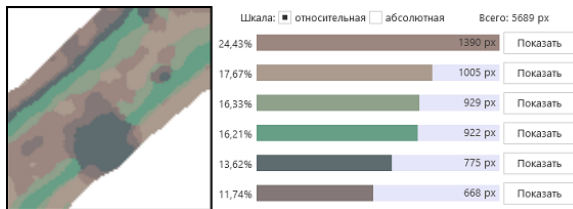


Рисунок 14 – Початкові мітки (медіанний)

Медіанний фільтр виявив майже однакову кількість пікселів у порівнянні з білатеральним фільтром та показав стабільний результат на всіх трьох дослідженнях, серед яких k-means++ та початкові мітки відрізняються у одному кольорі на соті долі відсотка.

Всі методи ініціалізації k-середніх показали приблизно однакові результати та виявили 5689 пікселів.

**ВИСНОВКИ.** В ході роботи було розроблено програму для проведення аналізу по вибору методів підготовки та кластеризації зображення. Отриманий продукт дозволяє виділяти необхідний регіон на зображенні, видаляти задній фон, який не потрібен для аналізу, використовувати один з визначених методів фільтрації (білатеральна, гаусівська, медіанна та розмиття), проводити кластеризацію з різними варіантами та показувати знайдені кольори на зображенні. Створена програма має адаптивний дизайн та дружній інтерфейс користувача.

За допомогою розробленої програми було проведено аналіз вхідного зображення. Отримані результати показали, що кластеризація сильно залежить від попередньої обробки зображення, від самого зображення та кількості кольорів, що його складають. Було виявлено, що для програми «Fiber Analyze» необхідно застосовувати медіанну фільтрацію, яка показала найкращі результати у порівнянні з іншими методами.

#### ЛІТЕРАТУРА

1. Калиновски Е., Урбанчик Г.В. Химические волокна (исследования и свойства). – М.: Легкая индустрия – 1966. – 319 с.
2. Скрипко Г.О. Дослідження мікрооб'єктів волокнистої природи за допомогою поляризаційної мікроскопії. – Київ: ДНДЕКЦ МВС України. – 2004. – 42 с.
3. Криминалистическое исследование волокнистых материалов и изделий из них. Выпуск 2. Исследование текстильных волокон / В.А. Пучков,

А.А. Сёдов, Г.А. Сергаева, В.П. Чернов: – М.: МЮ СССР ВНИИСС, 1983. – 311 с.

4. Nixon M., Aguado A. Feature Extraction and Image Processing. – Academic Press, 2008 – 350 p.

5. Tomasi C., Manduchi R. Bilateral filtering for gray and color images // Proceedings of the IEEE International Conference on Computer Vision, January 1998. – pp. 839–846.

6. Image Pyramids [Електронний ресурс] – Режим доступу до ресурсу: <http://docs.opencv.org/2.4/doc/tutorials/imgproc/pyramids/pyramids.html>.

7. Clustering [Електронний ресурс] – Режим доступу до ресурсу: <http://docs.opencv.org/2.4/modules/core/doc/clustering.html#kmeans>.

8. Григорова Т. Анализ изображения интерференционной картины волокна методом выделения цветовых регионов // Вісник Харківського національного університету імені В.Н. Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління». – Харків: ХНУ, 2013. – Вип. 22 (№ 1063). – С. 57–62.

9. k-means clustering [Електронний ресурс] – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering).

10. Pulli K. Realtime Computer Vision with OpenCV [Електронний ресурс] / K. Pulli, A. Vaksheev, K. Korniyakov, V. Eruhimov – 2012. – Режим доступу до ресурсу: <http://queue.acm.org/detail.cfm?id=2206309>.

11. Emgu CV [Електронний ресурс] – Режим доступу до ресурсу: <https://sourceforge.net/projects/emgucv/>.

12. WPF in CSharp [Електронний ресурс] – Режим доступу до ресурсу: [http://www.emgu.com/wiki/index.php/WPF\\_in\\_CSharp](http://www.emgu.com/wiki/index.php/WPF_in_CSharp).

13. Canny Edge Detector [Електронний ресурс] – Режим доступу до ресурсу: [http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny\\_detector/canny\\_detector.html](http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html).

14. Smoothing Images [Електронний ресурс] – Режим доступу до ресурсу: [http://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian\\_median\\_blur\\_bilateral\\_filter/gaussian\\_median\\_blur\\_bilateral\\_filter.html](http://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian_median_blur_bilateral_filter/gaussian_median_blur_bilateral_filter.html).

15. Arthur D. k-means++: The Advantages of Careful Seeding / D. Arthur, S. Vassilvitskii // Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. – 2007. – 11 p.

16. GitHub – firstfloorsoftware/mui: Modern UI for WPF [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/firstfloorsoftware/mui/>.

## RESEARCH OF INFLUENCE THE SELECTION OF METHOD IMAGE PREPROCESSING ON THE ANALYSIS IMAGES

T. Hryhorova, S. Guk

Kremenchug Mykhailo Ostrohradskyi National University

vul. Pershotravneva, 20, Kremenchuk, 39600, Ukraine E-mail: gasta1@yandex.ua

**Purpose.** A problem that is associated with the analysis of color image clustering method to identify colors and build their distribution in the image discussed in the paper. According to this distribution analyzes the quality and structure features fiber-forming polymer. **Methodology.** To investigate the impact of filtering methods for image processing results by clustering has developed analytical software studied the effect of image filtering techniques to result of the cluster image processing. In the process, it was developed analytical software for the experiment, which allows you to select the necessary part on the image, to remove the background, not needed for analysis, use one of the specific filtration methods (bilateral, Gaussian, median, and blur), to carry out clustering with various options and show the results in

the color image. **Results.** During the study, which was conducted with the image of the interference pattern fibers has been found that clustering depends on the pretreatment images from the images and the number of colors that make up the image. **Practical value.** It was found that image preprocessing for analysis by clustering fibers perform better using median filtering, which showed the best results compared to other methods.

**Key words:** image filtering, cluster processing, computer experiment.

#### REFERENCES

1. Kalinovskiy, E., Urbanchik, G. V. (1966) *Khimicheskie volokna (issledovaniya I svoystva)* [Chemical fibers (research and properties)], Legkaya industriya, Moscow, Russia.
2. Skripko, G .O. (2004) *Doslidgenya mikroobyektiv voloknistoyi pryrody za dopomogoyu polarizacionnoyi mikroskopiyi* [Research microscopic fibrous nature through polarization microscopy], DNDEKC MVS, Kuiv, Ukraine.
3. Puchkov, V. A., Sedov, G. A., Sergeeva, G. A., Chernov, V.P. *Kriminalisticheskoe issledovanie voloknistyh materialov I izdeliy iz nih. Vypusk 2. Issledovanie tekstilnyh volokon* [Forensic investigation of fibrous materials and products from them. Issue 2. Study of textile fibers], MU SSSR VNIISE, Moscow, Russia.
4. Nixon, M., Aguado A. (2008) *Feature Extraction and Image Processing*, Academic Press, New York, USA.
5. Tomasi, C., Manduchi R. "Bilateral filtering for gray and color images" *Proceedings of the IEEE International Conference on Computer Vision*, January, 1998, pp. 839-846.
6. Image Pyramids [E-resource], mode access to the resource: <http://docs.opencv.org/2.4/doc/tutorials/imgproc/pyramids/pyramids.html>.
7. Clustering [E-resource], mode access to the resource: <http://docs.opencv.org/2.4/modules/core/doc/clustering.html#kmeans>.
8. Hryhorova, T. (2013) "Image analysis of the interference pattern fiber by selection of color regions" *V. N. Karazin Kharkiv National University Bulletin. Series Mathematical Modeling. Information Technology. Automated Control Systems* vol. 22, no. 1063, pp. 57-62.
9. k-means clustering [E-resource], mode access to the resource: [https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering).
10. Pulli, K., Baksheev, A., Korniyakov, K., Eruhimov, V. (2012) *Realtime Computer Vision with OpenCV* [E-resource], mode access to the resource: <http://queue.acm.org/detail.cfm?id=2206309>.
11. Emgu CV [E-resource], mode access to the resource <https://sourceforge.net/projects/emgucv/>.
12. WPF in CSharp [E-resource], mode access to the resource: [http://www.emgu.com/wiki/index.php/WPF\\_in\\_CSharp](http://www.emgu.com/wiki/index.php/WPF_in_CSharp).
13. Canny Edge Detector [E-resource], mode access to the resource: [http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny\\_detector/canny\\_detector.html](http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html).
14. Smoothing Images [Електронний ресурс] – Режим доступу до ресурсу: [http://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian\\_median\\_blur\\_bilateral\\_filter/gaussian\\_median\\_blur\\_bilateral\\_filter.html](http://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian_median_blur_bilateral_filter/gaussian_median_blur_bilateral_filter.html).
15. Arthur, D., Vassilvitskii, S. "k-means++: The Advantages of Careful Seeding" *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007.
16. GitHub – firstfloorsoftware/mui: Modern UI for WPF [E-resource], mode access to the resource: <https://github.com/firstfloorsoftware/mui/>.

Стаття надійшла 11.10.2016.