



ЗАСТОСУВАННЯ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ ВИЗНАЧЕННЯ РОЗМІРІВ КРАПЛІ РІДИНИ НА ЦИФРОВИХ ЗОБРАЖЕННЯХ

Біліщук В. Б. *

Кандидат технічних наук, доцент

Івано-Франківський національний технічний університет нафти і газу

76019, вул. Карпатська, 15, м. Івано-Франківськ, Україна

<https://orcid.org/0000-0003-0196-412X>

e-mail: viktor.bilishchuk@nung.edu.ua

Криницький О. С.

Кандидат технічних наук, доцент

Івано-Франківський національний технічний університет нафти і газу

76019, вул. Карпатська, 15, м. Івано-Франківськ, Україна

<https://orcid.org/0000-0003-2338-8661>

e-mail: hipnoz23@ukr.net

Витвицька Л. А.

Кандидат технічних наук, доцент

Івано-Франківський національний технічний університет нафти і газу

76019, вул. Карпатська, 15, м. Івано-Франківськ, Україна

<https://orcid.org/0000-0001-8491-1239>

e-mail: lidiia.vytvytska@nung.edu.ua

Анотація. В статті вирішується задача визначення розмірів краплі рідини на цифровому зображенні з використанням технологій комп'ютерного зору засобами бібліотеки OpenCV, що може бути використано для слідження поверхневих властивостей рідин, зокрема крайового кута змочування. Крайовий кут змочування розраховують на основі виміряного радіуса основи або меридіану і заданого об'єму краплі рідини. Вимірювати радіус краплі рідини пропонується шляхом отримання цифрового зображення краплі, а радіус визначати алгоритмами комп'ютерного зору. В статті показано, що алгоритми комп'ютерного зору передбачають виконання ряду операцій. Спочатку кольорове зображення перетворюють на сіре. Далі виконують розмиття зображення згортою гаусовим ядром для зменшення впливу шуму. Наступним етапом є виділення контурів об'єктів на зображенні алгоритмом Кенні, який обчислює згортку за допомогою оператора Собеля. Подальша обробка зображення виконується шляхом фільтрації для проріджування країв контурів і придушення контурів з використанням двох порогових рівнів. Це дозволяє отримати контури товщиною в один піксель і усунути ізольовані ділянки контурів, що не належать дійсному контуру краплі, а є результатом неусуненого шуму на зображенні. В роботі вказано, що правильний вибір цих порогів впливає на ефективність алгоритму детектування країв. Заключним етапом є застосування алгоритму сканування зображення для формування масивів координат пікселів зображення, які належать контуру краплі. На основі масиву координат пікселів визначають радіус кола, який їм відповідає. Реальні розміри краплі на зображенні перераховують з використанням калібрувального об'єкта відомих розмірів. Таким об'єктом

Запропоноване посилання: Біліщук, В., Криницький, О., & Витвицька, Л. (2025). Застосування комп'ютерного зору для визначення розмірів краплі рідини на цифрових зображеннях. *Методи та прилади контролю якості*, 2(55), 60-67. doi: 10.31471/1993-9981-2025-2(55)-60-67.

* Відповідальний автор



Copyright © The Author(s). This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (<https://creativecommons.org/licenses/by/4.0/>)

може бути тіло простої форми – квадрат, круг тощо. На основі виміряних і відомих розмірів калібрувального тіла розраховують коефіцієнт перетворення цифрового зображення. Запропонований алгоритм реалізовується функціями бібліотеки OpenCV: `cvtColor`, `GaussianBlur`, `findContours` і `minEnclosingCircle`, що дозволяє ефективно розробляти програмне забезпечення із використанням бібліотек OpenCV. Таке програмне забезпечення може бути впроваджене на мікрокомп'ютерах, що відкриває широкі можливості для розроблення приладів для дослідження поверхневих властивостей рідин.

Ключові слова: крайовий кут змочування, OpenCV, згортка зображення, контур.

Вступ

Обробка зображень і алгоритми ідентифікації об'єктів на зображенні з використанням технологій комп'ютерного зору відіграють важливу роль у розвитку автоматизованих систем та штучного інтелекту. З допомогою комп'ютерного зору вирішують актуальні питання знаходження, відстежування, ідентифікації та класифікації об'єктів, оброблення даних і аналізу отриманої інформації з зображень. Розроблення програмного забезпечення з використанням технологій комп'ютерного зору застосовують для вирішення різноманітних завдань, що відкриває багато можливостей у різних галузях. Окремо можна виділити завдання визначення розмірів різних об'єктів на зображеннях. Застосування технологій комп'ютерного зору для таких цілей дозволяє автоматизувати процеси визначення розмірів об'єктів на зображеннях, що автоматизує і прискорює процеси проведення вимірювань [1].

Мета роботи полягає у розробленні методики визначення розмірів краплі рідини на горизонтальній твердій поверхні, що може бути використано для дослідження поверхневих властивостей рідин, зокрема крайового кута змочування. У роботі розглядається методика розпізнавання країв краплі рідини і визначення її геометричних розмірів з використанням бібліотеки OpenCV.

Аналіз сучасних закордонних і вітчизняних досліджень та публікацій

Зараз відома велика кількість робіт, присвячених застосуванню різних алгоритмів і методів вилучення потрібної інформації із цифрових зображень. Однією із задач є пошук і вимірювання розмірів потрібного об'єкта на зображенні.

У роботі [2] розглядають задачу вимірювання розмірів об'єктів за допомогою функцій OpenCV. В публікації зазначено, що використовуючи функції OpenCV можна виявити об'єкти на зображенні, обчислити їх ширину та висоту.

Відома інша робота [3], в якій вирішують задачу вимірювання розмірів об'єкта за допомогою функцій бібліотеки OpenCV. Зазначають, що задача реалізується шляхом пошуку контурів, перетворенням зображення на градації сірого і застосовують пороговий рівень яскравості для відокремлення об'єкта від фону з подальшим пошуком контуру об'єкта. Обчислення площі об'єкта спочатку здійснюють в пікселях, і потім перераховують площу, виражену в реальних одиницях вимірювання за допомогою масштабного коефіцієнта.

В роботі [4] представлена розроблена методика аналізу цифрових зображень фіксованої обертової краплі для визначення координат точок контурів таких крапель. У ній пропонується грубий пошук контуру краплі здійснювати шляхом сканування зображення, а прецизійне встановлення координат точок контуру здійснювати з використанням методу Кенні із субпіксельним розрахунком координат точок контурів. В іншій праці [5] запропоновано використовувати кореляційний метод аналізу цифрових зображень. У ньому здійснюють порівняння зразкового образу шуканого об'єкта на цифровому зображенні, що дозволяє ідентифікувати розташування шуканого об'єкта на зображенні. У зазначених роботах програмне забезпечення розроблялося без використання спеціалізованих бібліотек, тому процес розроблення програмного забезпечення був досить трудомістким.

Основний текст статті

Використання спеціалізованих програмних бібліотек дозволяє спростити процес розробки програмного забезпечення. Одним із таких засобів є бібліотека OpenCV (Open Source Computer Vision Library), яка дозволяє виконувати обробку зображень, включаючи виявлення і розпізнавання об'єктів, відстеження руху і рухомих об'єктів, об'єднання зображень для отримання цілої сцени з високою роздільною здатністю і багато іншого. Бібліотека OpenCV налічує понад 2500 оптимізованих алгоритмів, які полегшують розроблення програмного забезпечення засобами мов програмування Python, Java, Matlab та інших, і може вільно використовуватися в академічних і комерційних цілях на умовах ліцензії BSD [6].

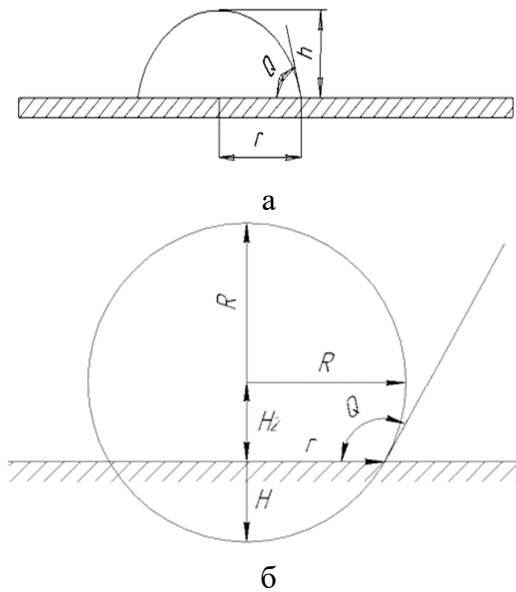
При використанні алгоритмів розпізнавання об'єктів на зображеннях виділяють типові послідовності дій:

- попередню обробку зображення: згладжування, фільтрація перешкод, підвищення контрасту;
- бінаризацію зображення і виділення контурів об'єктів;
- фільтрацію контурів за розмірами, площею, формою;
- пошук контуру, максимально схожого на шаблон.

Відомий метод визначення крайового кута змочування рідиною твердих тіл, в якому розміщують краплю із заданим об'ємом V на поверхні твердого тіла, вимірюють радіус основи краплі r або радіус меридіану R і розраховують крайовий кут змочування [7]. Якщо об'єм краплі V невеликий, то крапля приймає форму, близьку до сферичної. Розглянемо краплю на гідрофільній поверхні, тобто коли крайовий кут лежить в межах від 0° до 90° (рис. 1, а). У цьому випадку крайовий кут змочування розраховують згідно з такими залежностями:

$$\theta = \arccos \left(1 - \frac{2R - H}{\sqrt{\frac{2V}{\pi H} - \frac{H^2}{3}}} \right), \quad (1)$$

де



а – гідрофільна поверхня;
б – гідрофобна поверхня

Рисунок 1 – Форма краплі на поверхні

$$H = \frac{\sqrt[3]{\frac{2 \cdot (\gamma(R, V))^2}{\pi}} + R^2 \cdot \sqrt[3]{2 \cdot \pi} + R \cdot \sqrt[3]{\gamma(R, V)}}{\sqrt[3]{\gamma(R, V)}}.$$

На гідрофобній поверхні (рис. 1, б) значення кута змочування змінюються від 90° до 180° . Залежності для розрахунку кута змочування є дещо складнішими:

$$\theta = \arccos \left(1 - \frac{2R - H}{\sqrt{\frac{2V}{\pi H} - \frac{H^2}{3}}} \right), \quad (2)$$

де

$$H = \frac{\sqrt[3]{\frac{2 \cdot (\gamma(R, V))^2}{\pi}} + R^2 \cdot \sqrt[3]{2 \cdot \pi} + R \cdot \sqrt[3]{\gamma(R, V)}}{\sqrt[3]{\gamma(R, V)}}$$

$$\gamma(R, V) = -3V + 2R^3\pi + \sqrt{3 \cdot [-V \cdot (-3V + 4R^3\pi)]}$$

Вимірювання радіусів краплі рідини можна автоматизувати, скориставшись мікроскопом з відеокамерою, якою отримують зображення цієї краплі. Розміри краплі рідини на отриманому зображенні визначають відповідним програмним забезпеченням. Отже, існує задача розроблення програмного забезпечення для обробки зображення і визначення розмірів краплі рідини в розглянутому методі.

На початковому етапі розробки програмного забезпечення актуальним є вибір мови програмування. Зараз використовують безліч мов програмування. Однак основними лідерами можна вважати C++ та Python, оскільки саме в цих мовах можна використовувати бібліотеку комп'ютерного зору – OpenCV. Мова програмування Python приваблює простотою освоєння завдяки лаконічному синтаксису, що робить її доступною для початківців у програмуванні. Додатковою перевагою Python є активна спільнота, яка постійно розробляє безліч корисних бібліотек для використання у різних проєктах. Python – це мова програмування загального призначення, орієнтована на підвищення продуктивності розроблення різноманітного програмного забезпечення. Синтаксис ядра Python – мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій. Розроблене програмне забезпечення з використаною кросплатформовою бібліотекою OpenCV може працювати на різних операційних системах та платформах, таких як Windows (включаючи Windows 10, 8, та 7), Linux (Ubuntu, Fedora, Debian тощо), macOS, Android, iOS, Raspberry Pi, а також вбудованих системах.

Кожне зображення проходить певні етапи: попередньої обробки і аналізу. Попередня обробка зображення – це важливий етап в комп'ютерному зорі, який підготовляє зображення для подальшої обробки та аналізу. Метою попередньої обробки є підвищення якості зображення шляхом видалення шумів і покращення контрасту.

Спочатку відбувається перетворення зображення в градації сірого, щоб зменшити складність обробки, перетворивши кольорове зображення в шкалу сірого, що зменшує кількість каналів від трьох (червоний, зелений, блакитний) до одного. Інтенсивність кожного пікселя на сірому зображенні – це зважене середнє кольорових каналів оригінального пікселя, яке враховує людську чутливість до різних кольорів. Перетворення зображення з кольорового простору RGB (Red, Green,

Blue) – червоний, зелений, синій в градації сірого Y здійснюється функцією `cvtColor`, виражається як перерахунок яскравостей пікселів:

$$Y=0,299R+0,587G+0,114B, \quad (3)$$

де Y відповідає інтенсивності сірого кольору, а R , G , і B – інтенсивності червоного, зеленого і синього каналів відповідно. Коефіцієнти відображають важливість кожного кольору для сприйняття яскравості людським оком.

Наступним пунктом попередньої обробки є розмиття зображення, метою якого є зменшити вплив шуму та несуттєвих деталей на зображенні, що спрощує процес виявлення країв та контурів. Цього можна досягти з використанням гаусового ядра, що дає менше розмиття на краях і більше у центрі.

Розмиття зображення згорткою гаусовим ядром в OpenCV реалізовується функцією `cv2.GaussianBlur`. В цій функції задають розмір ядра згортки, який визначає ступінь розмиття. Більший розмір ядра дає сильніше розмиття і навпаки. Для кожного пікселя обчислюється зважена сума його сусідів, де ваги визначені гаусівським ядром:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (4)$$

де x та y – відстані від центру ядра по горизонталі та вертикалі відповідно, σ – стандартне відхилення розподілу, яке контролює рівень розмиття. Ядро згортки нормалізується так, щоб сума всіх елементів дорівнювала 1. Це забезпечує незмінність загальної яскравості зображення після розмиття.

Наступним етапом є виділення контурів об'єктів на зображенні. Для знаходження контурів об'єктів на зображенні використовують функцію `findContours`.

Функція `findContours` для виявлення країв об'єктів на зображенні використовує алгоритм Кенні. Він вважається одним з найефективніших алгоритмів виявлення країв і часто використовується в комп'ютерному зорі. У цьому алгоритмі обчислюється градієнт зображення для

визначення напрямку та інтенсивності зміни яскравості. Обчислення градієнтів здійснюється обчисленням згортки за допомогою оператора Собеля, який виявляє горизонтальні та вертикальні зміни:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I; \quad (5)$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I, \quad (6)$$

де I — це зображення у відтінках сірого.

Величина та напрямок градієнта:

$$G_{z,x} = \sqrt{G_x^2 + G_y^2}; \quad (7)$$

$$\theta = \arctg\left(\frac{G_y}{G_x}\right), \quad (8)$$

де G_x та G_y — градієнти в горизонтальному та вертикальному напрямках відповідно [8].

Після пошуку контурів відбувається їх фільтрування на основі різних критеріїв, таких як площа, форма або розмір, щоб ізолювати контур цільового об'єкта [9].

Для усунення розривів контурів і фіктивних контурів далі виконують ще ряд операцій. Немаксимальне придушення застосовується для “проріджування” країв. Для кожного пікселя перевіряється, чи є він максимумом у напрямку градієнта. Якщо ні, піксель придушується — його значення яскравості встановлюється в нульове. Таке перетворення створює краї границь об'єктів як тонкі лінії. В алгоритмі підпрограми `findContours` використовують два порогові значення σl і σh для визначення справжніх країв. Сильні краї з значенням градієнта, який вищий за високий поріг σh , визнаються як справжні краї. Краї зі значенням градієнта нижче низького порогу σl відкидаються. Інші краї з значенням градієнта, що знаходяться між двома порогамі, позначаються як слабкі краї. Такі області потребують додаткової перевірки.

Таким чином, нижній поріг для процесу гістерезису σl — це мінімальна інтенсивність, яка вважається досить сильною, щоб бути визначеною як край.

Верхній поріг для процесу гістерезису σh визначає мінімальну інтенсивність для пікселів, які можуть бути визнані як край, якщо вони з'єднуються з сильними краями. Правильний вибір цих порогів визначає ефективність алгоритму детектування країв. Низькі порогові значення залишають на зображенні багато шуму, а при виборі високих значень на зображенні можуть бути пропущені деякі краї. Для досягнення найкращих результатів відповідно до конкретного зображення та завдання значення порогів σl і σh встановлюють експериментально. Алгоритм придушення можна описати наступними виразами:

Сильні краї: $G(x,y) > \sigma h$;

Слабкі краї: $\sigma l < G(x,y) \leq \sigma h$,

Край відсутній: $G(x,y) \leq \sigma l$.

$$P(x, y) = \begin{cases} 255, & \text{якщо край слабкий} \\ & \text{або сильний;} \\ 0, & \text{якщо край відсутній} \\ & \text{або слабкий, що не з'єднаний} \\ & \text{з сильним.} \end{cases}$$

Інші параметри функції `findContours` `mode` і `method` визначають алгоритм сканування зображення і збереження контурів. Наприклад, виклик функції `findContours` з параметром `mode` рівним `cv2.RETR_EXTERNAL`, вказує функції `findContours` знаходити тільки зовнішні контури, тобто ті, які не містяться всередині інших контурів. У процесі сканування зображення згідно з алгоритмом визначається, чи слід з'єднувати знайдені слабкі краї з сильними. Якщо слабкий край з'єднаний з сильним краєм, він вважається справжнім краєм, інакше він відкидається. А застосування параметра `method` із значенням `cv2.CHAIN_APPROX_SIMPLE` вказує, що контури будуть апроксимовані лініями між сусідніми точками.

Функція `findContours` повертає кортеж: послідовність контурів $S = \{C_1, C_2, \dots, C_N\}$ та ієрархію H . Кожен контур S визначений як послідовність точок $\{p_1, p_2, \dots, p_n\}$. Ієрархія H може представлятись у вигляді матриці або деревовидної структури, що описує взаємозв'язки між контурами.

Кожен елемент p_{ij} контуру C_i містить координати (x, y) пікселя у вихідному зображенні. Ці елементи утворюють форму або границю об'єкта, яка може бути використана для аналізу та обробки. N – кількість контурів, що були знайдені на зображенні. Кожний контур C_i може містити різну кількість точок, що представляють границю об'єкта.

Крім цього, створюється ієрархія контурів, вона ж H – це матриця або вектор, що містить інформацію про ієрархічні зв'язки між контурами на зображенні.

Ієрархія визначає, як контури пов'язані між собою: які з них є “дитячими”, “батьківськими”, а також наступними та попередніми на даному рівні ієрархії H . Інформація про ієрархію може використовуватись для аналізу структури об'єктів на зображенні, визначення та інтерпретації взаємозв'язків між контурами. Кожний i -й елемент ієрархії H має таку структуру:

$$H_i = (next, prev, first_child, parent),$$

де $next, prev$ – це індекси наступного і попереднього контурів на цьому рівні ієрархії. Якщо немає таких контурів, $next$ і $prev$ дорівнюють -1. $first_child$ – це індекс першого «дитячого» контуру. $parent$ – це індекс «батьківського» контуру. Якщо немає таких контурів, ці значення рівні -1.

При якісно зробленій фотографії краплі на зображенні має бути один контур круглої форми, який відповідає розмірам основи або меридіану краплі рідини. Тому наступною задачею є знаходження розмірів, а точніше радіусу контуру у формі кола на зображенні.

Для визначення розмірів контуру об'єкта у вигляді еліпса або кола в бібліотеці OpenCV, використовують функції «minEnclosingCircle» або «cv2.fitEllipse». Оскільки контур краплі рідини на зображенні за формою найближчий до кола, то доцільніше використовувати функцію «minEnclosingCircle». Ця функція призначена для знаходження найменшого кола, яке може повністю охопити контур, заданий набором точок.

Завдання визначення найменшого охоплюючого кола (Minimum Enclosing Circle, МЕС) для набору точок є класичною

задачею геометричної оптимізації. OpenCV імплементує ефективний метод для знаходження МЕС, але деталі внутрішньої реалізації не розкриті. Для вирішення задачі визначення найменшого охоплюючого кола зазвичай використовують ітераційні методи, методи найменших квадратів, алгоритм Веллца тощо [10].

Функція minEnclosingCircle визначає розміри кола як кількість точок зображення (пікселів), яка йому відповідає. Наступним етапом є визначення реальних розмірів кола на зображенні. Для цього разом із об'єктом невідомих розмірів на зображенні розміщують об'єкт з відомими розмірами. В якості такого об'єкта розміщують тіло простої форми – квадрат чи круг, розміри якого також вимірює програма комп'ютерного зору. На основі вимірювань і відомих розмірів розраховують коефіцієнт перетворення цифрового зображення.

Висновки

В роботі зроблений аналіз алгоритмів ідентифікації і визначення розмірів об'єктів на цифрових зображеннях. На основі зробленого аналізу запропоновано використання бібліотеки OpenCV для розробки програмного забезпечення, яке використовують у задачах визначення крайового кута змочування на основі вимірювання розмірів краплі рідини. Застосування спеціалізованої бібліотеки дозволить пришвидшити і полегшити процес розроблення такого програмного забезпечення.

Запропонований алгоритм ідентифікації і визначення розмірів краплі рідини з використанням технологій комп'ютерного зору дозволить застосувати програмне забезпечення для мікрокомп'ютерів, що відкриває широкі можливості для проектування приладів для дослідження поверхневих властивостей рідин.

Подяки

Відсутні.

Конфлікт інтересів

Відсутній.

Список використаних джерел

1. Малкіна В. М., Мозговенко А. А. Визначення геометричних характеристик плодів черешні на основі технологій комп'ютерного зору. *Матеріали I Всеукраїнської науково-практичної інтернет-конференції «Сучасні комп'ютерні та інформаційні системи і технології»*. С. 49-52. URL: <http://www.tsatu.edu.ua/kn/wp-content/uploads/sites/16/malkyna-1.pdf>
2. Accurate Object Measurement with OpenCV Python. URL: <https://www.toolify.ai/ai-news/accurate-object-measurement-with-opencv-python-551694> (Дата звернення: 24.03.2025 p.)
3. Measure Size of an Object Using Python OpenCV. URL: <https://www.geeksforgeeks.org/measure-size-of-an-object-using-python-opencv/> (Дата звернення: 24.03.2025 p.)
4. Біліщук В. Б., Костів Б. В., Боднар Р. Т. Методика аналізу цифрових зображень для визначення координат точок контуру фіксованої оберткової краплі в міжфазній тензіометрії. *Методи та прилади контролю якості*. 2014. № 2 (33). С. 59-65. URL: <https://mpky.nung.edu.ua/index.php/mpky/article/view/231/235>
5. Біліщук В. Б. Застосування зразкових зображень для аналізу цифрових зображень фіксованих оберткових крапель. *Методи та прилади контролю якості*. 2015. № 2 (35). С. 48- 53. URL: <https://mpky.nung.edu.ua/index.php/mpky/article/view/273/277>
6. OpenCV. URL: <https://uk.wikipedia.org/wiki/OpenCV> (Дата звернення: 12.03.2025 p.)
7. Боднар Р. Т., Біліщук В. Б. Контроль змочування в промислових умовах оптичним методом. Збірник тез доповідей 8-ої міжнародної н/т конференції Сучасні прилади, матеріали і технології для неруйнівного контролю і технічної діагностики машинобудівного і нафтогазопромислового обладнання 14 - 16 листопада 2017. м. Івано-Франківськ: ФІНТУНГ. С. 111-112. URL: <http://194.44.112.14/bitstream/123456789/5588/1/6177p.pdf>
8. Satoshi Suzuki and others. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*. 1985. Vol. 30(1). P. 32-46. doi: 10.1016/0734-189X(85)90016-7
9. Canny J. A Computation Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell* 1986. Vol. 8. №6. P. 679-698. doi: 10.1109/TPAMI.1986.4767851
10. Задача про найменше коло. URL: https://uk.wikipedia.org/wiki/Задача_про_найменше_коло (Дата звернення: 24.03.2025 p.).

References

1. Malkina V. M., Mozghovenko A. A. Vyznachennia heometrychnykh kharakterystyk plodiv chereszni na osnovi tekhnolohii kompiuternoho zoru. *Materialy I Vseukrainskoi naukovo-praktychnoi internet-konferentsii «Suchasni kompiuterni ta informatsiini systemy i tekhnolohii»*. P. 49-52. URL: <http://www.tsatu.edu.ua/kn/wp-content/uploads/sites/16/malkyna-1.pdf> [in Ukrainian]
2. Accurate Object Measurement with OpenCV Python. URL: <https://www.toolify.ai/ai-news/accurate-object-measurement-with-opencv-python-551694> (Accessed: 24.03.2025 p.)
3. Measure Size of an Object Using Python OpenCV. URL: <https://www.geeksforgeeks.org/measure-size-of-an-object-using-python-opencv/> (Accessed: 24.03.2025 p.)
4. Bilishchuk V. B., Kostiv B. V., Bodnar R. T. Metodyka analizu tsyfrovyykh zobrazhen dla vyznachennia koordynat tochk konturu fiksovanoi obertovoi krapli v mizhfaznii tenziometrii. *Metody ta prylady kontroliu yakosti*. 2014. No 2 (33). P. 59-65. URL: <https://mpky.nung.edu.ua/index.php/mpky/article/view/231/235> [in Ukrainian]
5. Bilishchuk V. B. Zastosuvannia zrazkovyykh zobrazhen dla analizu tsyfrovyykh zobrazhen fiksovanykh obertovyykh krapel. *Metody ta prylady kontroliu yakosti*. 2015. No 2 (35). P. 48- 53. URL: <https://mpky.nung.edu.ua/index.php/mpky/article/view/273/277> [in Ukrainian]
6. OpenCV. URL: <https://uk.wikipedia.org/wiki/OpenCV> (Accessed: 12.03.2025 p.). [in Ukrainian]
7. Bodnar R. T., Bilishchuk V. B. Kontrol zmochuvannia v promyslovykh umovakh optychnym metodom. Zbirnyk tez dopovidei 8-oi mizhnarodnoi n/t konferentsii Suchasni prylady, materialy i tekhnolohii dla neruivnogo kontroliu i tekhnichnoi diahnostryky mashynobudivnogo i naftohazopromysloвого обладнання 14 - 16 lystopada 2017. Ivano-Frankivsk: IFINTUNH. P. 111-112. URL: <http://194.44.112.14/bitstream/123456789/5588/1/6177p.pdf> [in Ukrainian]
8. Satoshi Suzuki and others. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*. 1985. Vol. 30(1). P. 32-46. doi: 10.1016/0734-189X(85)90016-7

9. Canny J. A Computation Approach to Edge Detection. IEEE Trans. Pattern Anal. Mach. Intell 1986. Vol. 8. No 6. P. 679-698. doi: [10.1109/TPAMI.1986.4767851](https://doi.org/10.1109/TPAMI.1986.4767851)

10. Zadacha pro naimenshe kolo. URL: [https://uk.wikipedia.org/wiki/ Задача_про_найменше_коло](https://uk.wikipedia.org/wiki/Задача_про_найменше_коло) (Accessed: 24.03.2025 p.). [in Ukrainian]

APPLICATION OF COMPUTER VISION TO DETERMINE LIQUID DROP SIZE IN DIGITAL IMAGES

Bilishchuk V. B.

Candidate of Technical Sciences, Associate Professor
Ivano-Frankivsk National Technical University of Oil and Gas
76019, Karpatska St., 15, Ivano-Frankivsk, Ukraine
<https://orcid.org/0000-0003-0196-412X>
e-mail: viktor.bilishchuk@nung.edu.ua

Krynytskyi O. S.

Candidate of Technical Sciences, Associate Professor
Ivano-Frankivsk National Technical University of Oil and Gas
76019, Karpatska St., 15, Ivano-Frankivsk, Ukraine
<https://orcid.org/0000-0003-2338-8661>
e-mail: hipnoz23@ukr.net

Vytyvtska L. A.

Candidate of Technical Sciences, Associate Professor
Ivano-Frankivsk National Technical University of Oil and Gas
76019, Karpatska St., 15, Ivano-Frankivsk, Ukraine
<https://orcid.org/0000-0001-8491-1239>
e-mail: lidiia.vytyvtska@nung.edu.ua

Abstract. The article solves the problem of determining the size of a liquid drop in a digital image using computer vision technologies using the OpenCV library, which can be used to track the surface properties of liquids, in particular the marginal wetting angle. The marginal wetting angle is calculated based on the measured radius of the base or meridian and a given volume of the liquid drop. It is proposed to measure the radius of a liquid drop by obtaining a digital image of the drop, and determine the radius using computer vision algorithms. The article shows that computer vision algorithms involve performing a number of operations. First, the color image is converted to gray. Then the image is blurred by convolution with a Gaussian kernel to reduce the influence of noise. The next stage is to select the contours of objects in the image using the Kenny algorithm, which calculates the convolution using the Sobel operator. Further image processing is performed by filtering to thin out the edges of the contours and suppress the contours using two threshold levels. This allows us to obtain contours with a thickness of one pixel and eliminate isolated areas of contours that do not belong to the actual contour of the droplet, but are the result of unremoved noise in the image. The work indicates that the correct choice of these thresholds affects the efficiency of the edge detection algorithm. The final stage is the application of the image scanning algorithm to form arrays of image pixel coordinates that belong to the contour of the droplet. Based on the array of pixel coordinates, the radius of the circle that corresponds to them is determined. The real dimensions of the droplet in the image are recalculated using a calibration object of known dimensions. Such an object can be a body of a simple shape - a square, a circle, etc. Based on the measured and known dimensions of the calibration body, the conversion factor of the digital image is calculated. The proposed algorithm is implemented by the functions of the OpenCV library `cvtColor`, `GaussianBlur`, `findContours` and `minEnclosingCircle`, which allows us to effectively develop software using OpenCV libraries. Such software can be implemented on microcomputers, which opens up wide opportunities for the development of devices for studying the surface properties of liquids.

Keywords: contact angle; OpenCV; image convolution; contour.