

РОЗПІЗНАВАННЯ ГЕОМЕТРИЧНИХ ФІГУР ТА ВИЗНАЧЕННЯ ЇХ ХАРАКТЕРИСТИК ЗАСОБАМИ КОМП'ЮТЕРНОГО ЗОРУ

Вступ. В багатьох застосуваннях комп'ютерного зору існує потреба у виконанні певних процедур з геометричними фігурами на зображеннях, таких як розпізнавання типу фігури (круг, трикутник, чотирикутник, п'ятикутник та інше), визначення центроїду фігури тощо [1]. Розв'язання зазначених задач досить ефективно можна досягти через використання процедури визначення контурів фігур довільної форми. Для цього, потрібно визначити границі об'єктів і локалізувати їх на зображенні. Часто це є першим етапом для багатьох застосунків комп'ютерного зору, таких як відокремлення об'єктів від фону, сегментація зображень, виявлення та розпізнавання різноманітних об'єктів тощо.

1. Особливості конвеєру обробки зображень з фігурами довільної форми

Одним із варіантів успішного розв'язання задач визначення центроїду, як і параметрів геометричних фігур досягається шляхом використання бібліотек OpenCV, орієнтованих на обробку зображень. У загальному випадку весь процес як визначення певних характеристик двовимірних фігур, так і їх розпізнавання потребують виконання ланцюжка таких операцій, як перетворення кольорового зображення в зображення у градаціях сірого, його покращення та бінарна порогова класифікація. Кожен з визначених етапів є важливим і безпосередньо впливає на якість та коректність отриманого результату. Так при уведенні кольорового зображення (рис. 1, а) слід звертати увагу на той факт, що OpenCV виконує введення кольорових зображень у BGR форматі на відміну від звичного для нас RGB формату! Зазвичай у реальних умовах зображення має не завжди відмінну якість. Вважати, що методи обробки зображень завжди будуть обробляти зображення високої якості, є великою помилкою. На якість зображення впливає багато факторів, які приносять певні зміни у зображення. Вони можуть бути викликані якістю апаратури для фіксації зображень, умовами освітлення та інше. Для зниження впливу зазначених

Розглянуто питання певних процедур з геометричними фігурами на зображеннях, таких як розпізнавання типу фігури (круг, трикутник, чотирикутник, п'ятикутник та інше), визначення центроїду фігури тощо, які використовуються у багатьох застосуваннях комп'ютерного зору. В проведеному дослідженні особливу увагу надано аналізу застосування методів вибору порогової класифікації для задач розпізнавання фігур у зображенні та запропоновано підхід до його вибору, розроблено програмну модель визначення оптимального порогу класифікації зображень, розроблено програмні моделі визначення центроїду фігур довільної форми і підрахунку фігур у зображенні та їх розпізнавання.

Ключові слова: контур, сегментація, бінаризація зображень, комп'ютерний зір, гістограма.

факторів існує багато методів покращення зображень [2]. Завдяки ним досягається покращення зображення та підвищення його різкості, видаляються небажані завади у зображенні. Якщо не позбавитися від подібних недоліків, то це може призвести до того, що результати сегментації будуть хибні. На рис. 1, б окремі геометричні фігури зливаються з сусідніми, що призводить у подальшому до некоректного визначення центру фігури і, таким чином, недоліки або недоробки у виконанні певного етапу вищевказаного ланцюжка операцій впливають на кінцевий результат.

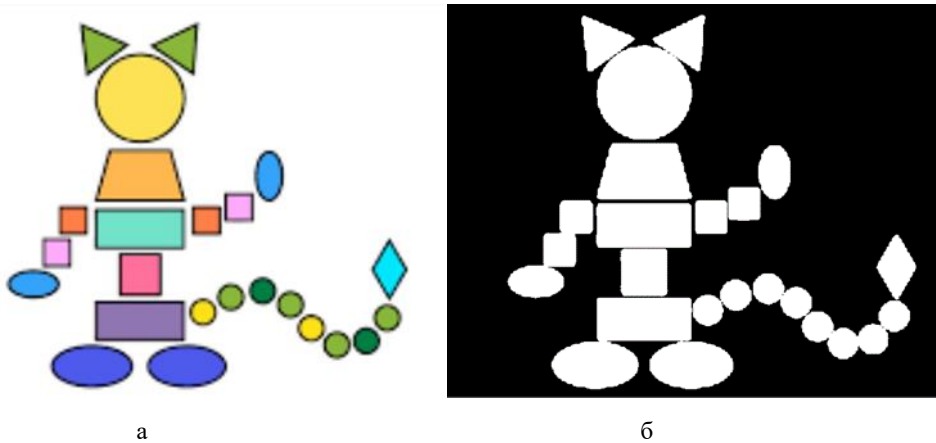


РИС. 1, а – зразок зображення; б – бінаризоване зображення без усунення завад в зображенні

Наступний етап (порогова класифікація) обробки зображення дає кращі результати якщо зображення має більш високий контраст. Тому перед виконанням процедури сегментації важливо підвищити контраст зображення. Одним із варіантів покращення контрасту зображення є застосування методів на основі гамма корекції [2].

Порогова класифікація зображення – надто відповідальний етап обробки зображення через те, що його результатом є розподіл первинного зображення на передній і задній фон (у випадку двох класів пікселів) у вигляді бінаризованого зображення. Важливість цього етапу обробки зображення обумовлена не тільки виконанням очищення зображення у градаціях сірого від шуму і виділенням певних об'єктів уваги на задньому фоні, а й тим щоб на цьому етапі разом з завадами не були втрачені потрібні об'єкти уваги. Саме це і привертає особливу увагу до процесу порогової класифікації зображень. Серед великої кількості наявних методів визначення порогу класифікації зображень [3, 4] особливе місце займає найбільш відомий і найефективніший метод Otsu [5].

1.1. Метод Otsu

Методи автоматичного визначення порогів класифікації можна приблизно поділити на дві великі групи: глобальні і локальні (рис. 2). До першої групи методів належать ті, що визначають одне порогове значення з гістограми всього зображення. До другої – відносяться методи, що визначають декілька порогових значень для певних областей зображення. Вони є локальними і оптимізованими для кожної визначеної області зображення. Для їх визначення використовується інформація про розподіл рівня сірого тільки цієї області. Методи вибору глобального порогу класифікації простіші й легші для реалізації, але їх результат залежить від гарного (рівномірного) освітлення. Зазвичай вони використовуються завдяки їх простоті та швидкості. Методи вибору локальних порогів класифікації можуть мати справу з нерівномірним освітленням, але вони складні й повільні.

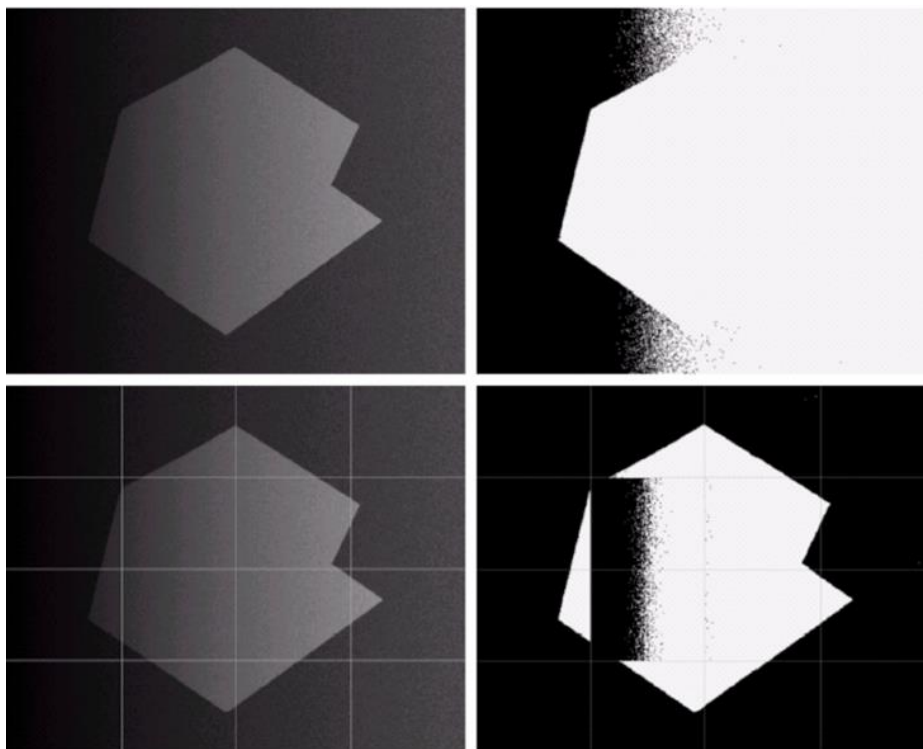


РИС. 2. Приклад глобальної та локальної бінаризації зображень

Серед методів вибору глобального порогового значення найбільшого розповсюдження набув метод Otsu [5]. Він часто застосовується у бінаризації документів [6], комп'ютерному зорі [7] та розпізнаванні шаблонів [8]. Метод Otsu автоматично вибирає оптимальний поріг, максимізуючи дисперсію сегментованого зображення між класами. Це просто, ефективно та без додаткових параметрів. Завдяки цим перевагам була запропонована велика кількість методів для вдосконалення оригінального методу Otsu, зокрема в роботах [9–12].

Оскільки рівні сірого характеризують об'єкти на сірому зображенні, багато методів встановлення порогів витягують об'єкти з їхнього фону на основі статистики одновимірної (1D) та двовимірної (2D) гістограми сірих рівнів. Найчастіше використовується метод Otsu, який вибирає глобальний оптимальний поріг, максимізуючи дисперсію між класами. При дворівневому пороговому рівні піксель, рівень сірого якого менше порогового значення, буде присвоєний фоновому режиму, інакше передньому плану. На жаль, не завжди добре знайти єдиний поріг, який є «найкращим» для довільного рівня сірого. Отже, існує багато методів визначення багаторівневого порогу класифікації [13–17], які працюють краще, ніж методи визначення дворівневого порогу, коли зображення має складні об'єкти або фон. Одновимірні (1D) гістограми рівнів сірого враховують лише рівні сірого. Тому різні зображення з однаковою гістограмою будуть призводити до того самого порогового значення.

Якщо припустити, що зображення представлене в L рівнях сірого $[0, 1, \dots, L-1]$, а кількість пікселів на рівні i позначити n_i , то загальна кількість пікселів у зображенні дорівнює $N = n_1 + n_2 + \dots + n_L$, а ймовірність рівня сірого i становить

$$p_i = n_i/N, \quad p_i \geq 0, \quad \sum_0^{L-1} p_i = 1. \quad (1)$$

У випадку використання дворівневого методу порогової класифікації значення пікселів зображення поділяються на два класи з рівнем розподілу t : клас C_1 з рівнями сірого $[0, 1, \dots, t]$ і C_2 з рівнями сірого $[t+1, \dots, L-1]$ з відповідним розподілом ймовірностей рівня сірого

$$w_1 = \text{PR}(C_1) = \sum_{i=0}^t p_i, \quad (2)$$

$$w_2 = \text{PR}(C_2) = \sum_{i=t+1}^{L-1} p_i. \quad (3)$$

Відповідно середнє значення класів C_1 і C_2 обчислюються наступним чином:

$$u_1 = \sum_{i=0}^t ip_i / w_1, \quad (4)$$

$$u_2 = \sum_{i=t+1}^{L-1} ip_i / w_2. \quad (5)$$

Загальне по всьому зображенню середнє значення рівня сірого u_T визначається так:

$$u_T = w_1 u_1 + w_2 u_2. \quad (6)$$

Дисперсії класів визначаються так:

$$\sigma_1^2 = \sum_{i=0}^t (i - u_1)^2 p_i / w_1, \quad (7)$$

$$\sigma_2^2 = \sum_{i=t+1}^{L-1} (i - u_2)^2 p_i / w_2. \quad (8)$$

Дисперсія всередині класу становить

$$\sigma_w^2 = \sum_{k=1}^M w_k \sigma_k^2 \quad (9)$$

між класами

$$\sigma_B^2 = w_1 (u_1 - u_T)^2 + w_2 (u_2 - u_T)^2. \quad (10)$$

Загальна дисперсія рівня сірого дорівнює сумі дисперсії всередині класу і дисперсії між класами

$$\sigma_T^2 = \sigma_w^2 + \sigma_B^2. \quad (11)$$

Метод Otsu вибирає оптимальний поріг t шляхом вибору максимального значення дисперсії рівня сірого між класами, що є еквівалентним вибору дисперсії всередині класу тому, що загальна дисперсія є величиною постійною для будь-якого розподілу на класи.

$$t = \arg \left\{ \underset{0 < t < L-1}{MAX} \left\{ \sigma_B^2(t) \right\} \right\} = \arg \left\{ \underset{0 < t < L-1}{MIN} \left\{ \sigma_W^2(t) \right\} \right\}. \quad (12)$$

Без особливих труднощів метод Otsu може бути розширено до декількох рівнів класифікації.

На рис. 3, а – е показані приклади реалізації перетворення вхідного зображення у бінаризоване із застосуванням методу вибору порогу класифікації Otsu.

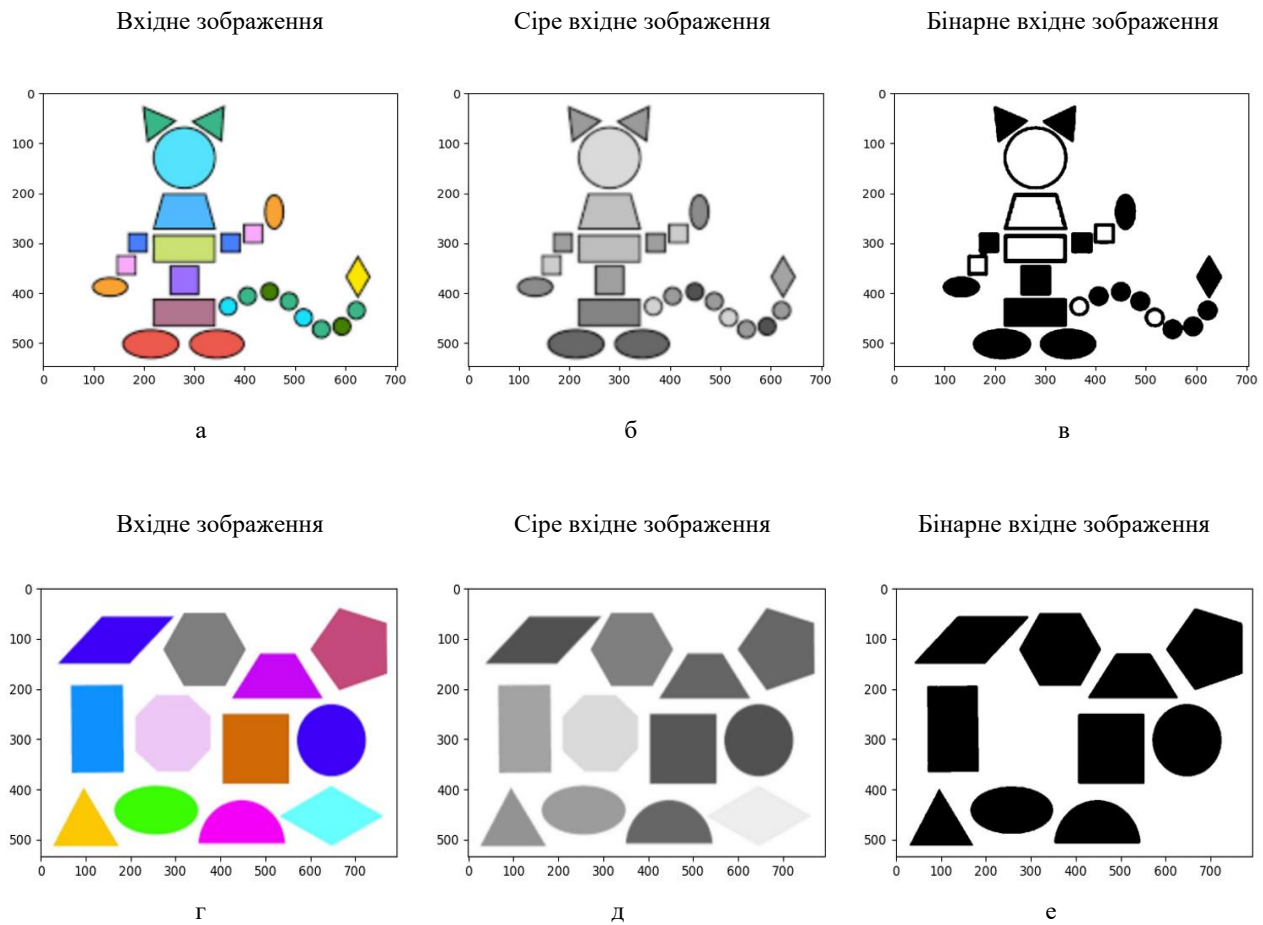


РИС. 3. Приклади реалізації перетворення вхідного зображення в бінаризоване із застосуванням методу вибору порогу класифікації Otsu

Якщо уважно придивитися до результатів бінаризації, то можна побачити втрату двох фігур на рис. 3, е та а – всі фігурки мають контур незалежно від кольору, а на рис. 3, г фігури рівномірні без видимих контурів. Застосувавши метод Otsu визначається поріг класифікації зображень. Саме мінімальне значення варіації у межах класу і максимальне значення варіації між класами відповідають оптимальному значенню порога класифікації (рис. 4, а і б).

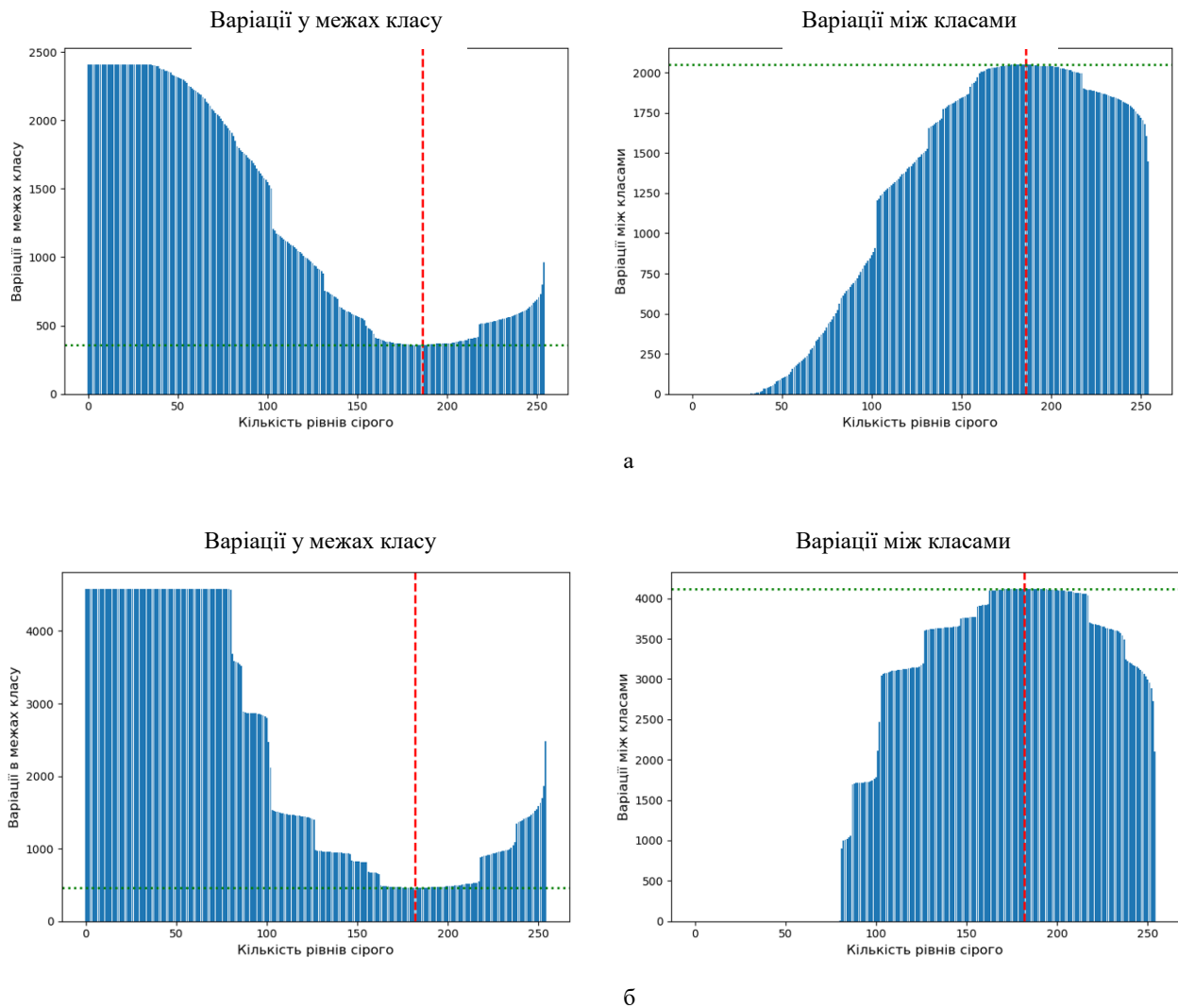


РИС. 4, а і б – приклади розподілу варіацій яскравості у межах класу та між класами (червона пунктирна лінія відображає поріг класифікації)

Результати обчислень зображення з рис. 3, а:

"C:\Python\Python 3.7.6\python.exe" "E:\Python My Scripts\Project_Segmentation\my_segmentation_otsu_0_v_0.py"

Розміри зображення - (546, 704)

Кількість пікселів в зображенні = 384384

Середнє значення сірого всього зображення = 231.01757617382617

Мінімальне значення варіації у межах класу = 357.8682913205712

Максимальне значення варіації між класами = 2051.9302155251107

Поріг класифікації по Otsu = 186

Результати обчислень зображення з рис. 3, г:

"C:\Python\Python 3.7.6\python.exe" "E:\Python My Scripts\Project_Segmentation\my_segmentation_otsu_0_v_0.py"

Розміри зображення - (533, 793)

Кількість пікселів в зображенні = 422669

Середнє значення сірого всього зображення = 198.78594834255648

Мінімальне значення варіації у межах класу = 463.0962640484265

Максимальне значення варіації між класами = 4119.264645089448

Поріг класифікації по Otsu = 182

Після отримання оптимального значення порогу класифікації зображень по методу Otsu можна повернутися до визначення причин втрати певних фігур у бінаризованому зображенні на рис. 3, е, побудувавши гістограми вхідного зображення (рис. 5).

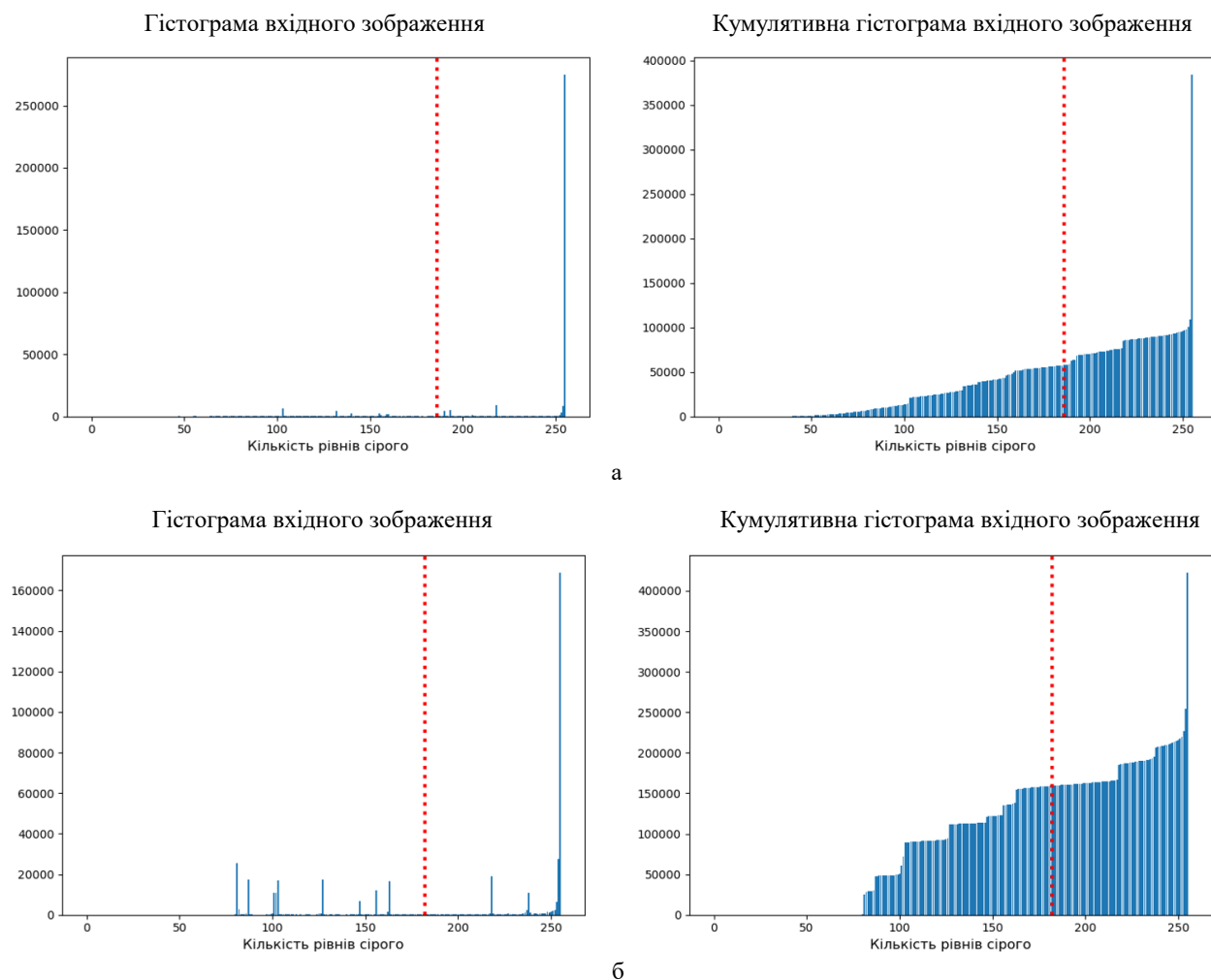


РИС. 5, а і б – приклади гістограм розподілу рівнів яскравості зображень з рис. 3, а та г (червона пунктирна лінія відображає поріг класифікації)

З наведених прикладів витікає, що зображення фігур певних кольорів на етапі перетворення кольорового зображення в сіре зображення з градаціями сірого при гомогенній структурі заливки фігур, втрачаються на етапі бінаризації у тому випадку, коли поріг класифікації перевищує значення яскравості сірого цієї фігури (рис. 5, б). Слід приймати до уваги, що після перетворення зображення у градації сірого діапазон рівнів яскравості усіх наявних геометричних фігур простягається від 0 до 255. Якщо задача виявлення геометричних фігур не накладає певних обмежень на колір або яскравість фігур, то не можна застосовувати автоматичні методи на кшталт методу Otsu для виявлення порогу класифікації. Цей метод визначає оптимальне значення порогу класифікації на основі визначення максимального значення міжкласової варіації (дисперсії) рівнів яскравості і в такому разі фігури з низьким рівнем яскравості можуть бути втрачені (рис. 3, е). Таким чином, остаточний вибір порогу класифікації визначається згідно умов застосування.

В застосуваннях комп'ютерних методів етапи конвеєру від отримання зображення до повного використання даних з цих зображень не мають чітких меж, проте у ньому можна розрізняти комп'ютеризовані процеси низького, середнього і високого рівня. Процеси низького рівня стосуються тільки примітивних операцій типу попередньої обробки з метою зменшення шуму, підвищення контрасту або поліпшення різкості зображень. Для низькорівневих процесів характерний той факт, що на вході й на виході присутні зображення. Обробка зображень на середньому рівні охоплює такі завдання, як сегментація (розподіл зображення на області або виділення на ньому об'єктів), опис об'єктів і стиснення їх у зручну для комп'ютерної обробки форму, а також класифікація (розпізнавання) окремих об'єктів. Для процесів середнього рівня характерна наявність зображень тільки на вході, на вихід поступають вже тільки ознаки і атрибути, які добувають із цих зображень (наприклад, границі областей, лінії контурів, відмінні ознаки конкретних об'єктів). Високорівнева обробка включає у себе «осмислення» набору розпізнаних об'єктів та розпізнавання їх взаємодії.

Виходячи з вищенаведеного зауваження видно, що на середньому рівні обробки відбувається перетворення просторових даних зображень у метадані, стискання обсягу інформації, в результаті чого важливість цих даних суттєво зростає. Це свідчить про те, що на вході середнього рівня зображення повинно бути максимально інформативним (з високою контрастністю, без завад, артефактів тощо). І після перетворення просторових даних зображення у метадані вже ніякі наступні процедури не в змозі коригувати у бік покращення чи підвищення інформативності отримані відеосенсорами дані.

Саме з сегментації починається процес поділу зображення на пікселі об'єкта інтересу або передній фон та пікселі, що належать до заднього фону зображення. Отже сегментація зображень є відповідальним і важливим етапом у ланці всього конвеєру обробки зображень. Вищерозглянутий метод Otsu визначення порогу класифікації належить до групи методів порогової класифікації, сегментація зображень на основі яких дає кращі результати, якщо зображення має більший контраст.

2. Визначення центру довільної геометричної форми

Кінцевою метою вищенаведених маніпуляцій з зображеннями є виявлення крайових точок фігур. А маючи набір крайових точок фігур у зображенні після застосування певних методів апроксимації можна визначити контури фігур. Контури представляють суттєвий механізм для аналізу зображень об'єктів і широко застосовуються в задачах комп'ютерного зору.

У подальшому при використанні бібліотек OpenCV маніпуляції з зображеннями фігури, які підлягають аналізу представлені у білому кольорі (передній план) на задньому чорному фоні.

Це спрощує пошук положення цих білих плям на чорному фоні шляхом визначення контурів за допомогою функції `cv2.findContours`. В OpenCV пошук контурів схожий на пошук білого об'єкта на чорному тлі. В такому разі слід пам'ятати, що об'єкт, який потрібно знайти, має бути білим, а фон – чорним. Після визначення контурів застосовується функція `cv.drawContours()` для нанесення визначеного контуру на первісне зображення в RGB форматі.

Подальше визначення центру довільної фігури визначається через моменти зображення [18, 19]. У комп'ютерному зорі та обробці зображень моменти зображення часто використовують для характеристики форми предмета на зображенні. Ці моменти фіксують основні статистичні властивості фігури, включаючи площу об'єкта, центроїд (тобто центр (x, y) – координати об'єкта), орієнтацію, поряд з іншими бажаними властивостями. Координати центроїду фігури визначаються через моменти зображення

$$C_x = m_{1,0}/m_{0,0}, \quad (13)$$

$$C_y = m_{0,1}/m_{0,0},$$

де геометричні моменти низьких порядків мають інтуїтивне значення $m_{0,0}$, яке є "масою" зображення (для двійкових зображень це площа об'єкта), $m_{1,0}/m_{0,0}$ та $m_{0,1}/m_{0,0}$ визначають центр ваги або центроїд зображення (рис. 6).

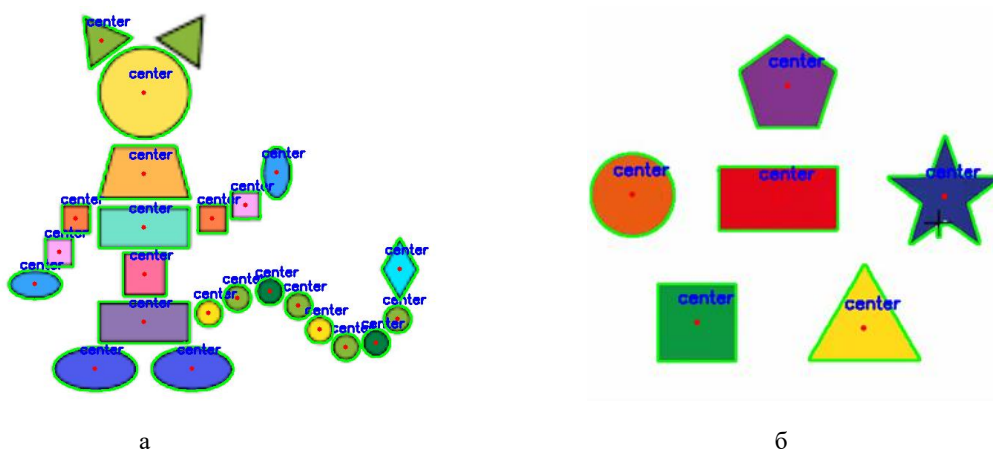


РИС. 6. Визначення центру контурів фігур довільної форми

Для визначення моментів фігур у зображенні потрібно провести всі зазначені вище маніпуляції з зображенням, а саме перетворення кольорового зображення у зображення у градаціях сірого та порогову класифікацію для отримання бінаризованого зображення. З метою уникнення втрати певних фігур, як показано на рис. 7, слід приймати до уваги вищевказане попередження стосовно коректного вибору порогу класифікації та умов застосування.

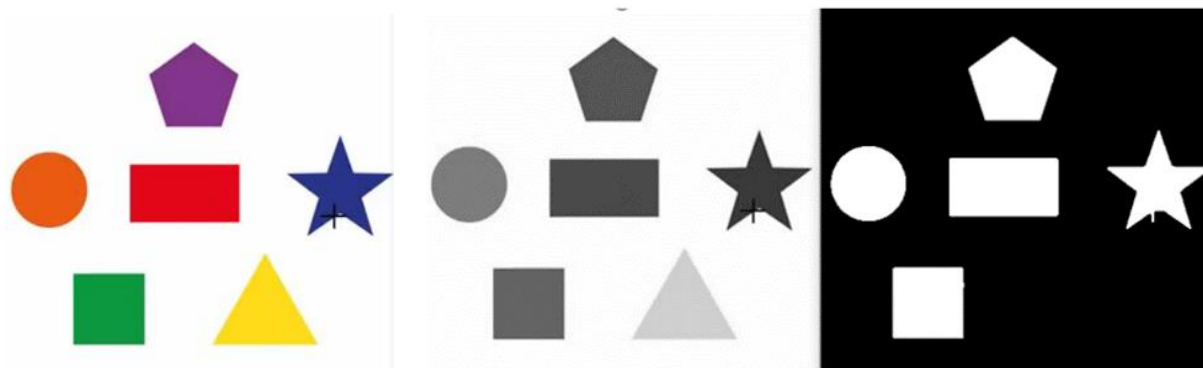


РИС. 7. Зразок зображення та втрата фігури на зображенні через невдалий вибір порогу класифікації зображення

3. Розпізнавання типу фігур довільної геометричної форми

Задача визначення типу фігур довільної геометричної форми досить часто виникає на практиці, наприклад, контроль розмірів окремих деталей у автоматизованих системах контролю якості, розпізнавання об'єктів певної форми (дорожні знаки) та інше. Всі ці задачі поєднані однією спільною особливістю, а саме визначенням контуру об'єкта і зменшенням кількості точок, які в найбільш придатній для застосування формі їх представляють.

Отриманню точок контуру фігури довільної форми передують такі операції (перетворення кольорового зображення в формат відтінків сірого, покращення та бінарна порогова класифікація), як і для визначення центроїду фігури.

Як правило, отриманий за допомогою OpenCV контур заданої фігури складається з великої кількості точок. Деякі з них точно відповідають контуру первинної фігури, а деякі зміщені від фактичного контуру фігури через наявні завади в зображенні. Опис кривої у вигляді серії двовимірних розташувань $x_i = x(s_i)$ забезпечує загальне подання, придатне для найбільшого наближення до первинної кривої і подальшої обробки. Однак, у багатьох застосуваннях бажано наблизити таку криву за допомогою простого подання, наприклад, кусково-лінійної полілінії або В-сплайн кривої [20], як показано на рис. 8.

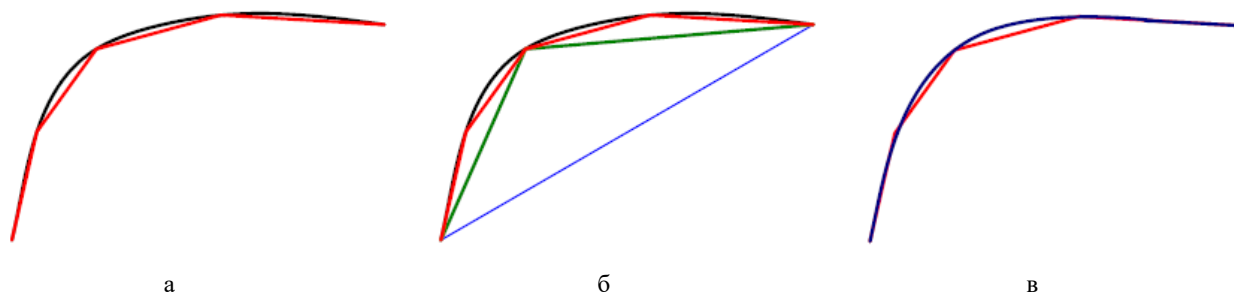


РИС. 8. Апроксимація кривої (показана чорним кольором) як полілінії або В-сплайн: а – оригінальна крива та наближення полілінії показано червоним кольором; б – послідовне наближення шляхом рекурсивного знаходження точок, найбільш віддалених від поточного наближення; в – плавний інтерполюючий сплайн, показаний темно-синім кольором, підходить до вершин полілінії

Серед великої кількості методів для виконання апроксимації одним з найдавніших і найпростіших є Ramer-Douglas-Peucker алгоритм [21, 22], мета якого полягає у створенні спрощеної ламаної лінії, яка має менше точок, ніж первинна, але при цьому зберігає характеристики та форму оригіналу.

Алгоритм рекурсивно ділить лінію (рис. 9). Входом алгоритму служать координати всіх точок між першою і останньою. Перша і остання точка зберігаються незмінними. Після чого алгоритм знаходить точку, найбільш віддалену від відрізка, що з'єднує першу і останню. Якщо точка знаходиться на відстані, меншій ϵ , то всі точки, які ще не були відзначені до збереження, можуть бути вилучені з набору і отримана пряма згладжує криву з точністю не нижче ϵ .



РИС. 9. Згладжування кусково-лінійної кривої алгоритмом Ramer-Douglas-Peucker

Якщо ж відстань більше ϵ , то алгоритм рекурсивно виконується на наборі від початкової до даної і від даної до кінцевої точках (що означає, що дана точка буде відзначена до збереження). По закінченню всіх рекурсивних викликів вихідна ламана будується тільки з тих точок, що були відзначені до збереження.

Для реалізації процедури розпізнавання типу геометричних фігур застосовуємо контури об'єктів у зображенні та такі функції OpenCV:

- **cv.findContours()** функція, що визначає всі контури зображення, які може знайти, з двійкових зображень. Функція **cv.findContours()** може приймати зображення, створені після застосування детектора контурів Canny, або зображень отриманих у результаті простої (**cv.threshold()**) або адаптивної (**cv.adaptiveThreshold()**) порогової класифікації, у яких краї є безпосередньо границями між позитивними і негативними областями.

Аргументи функції:

- **image**: вхідне бінарне зображення;
- **mode**: аргумент режиму пошуку контурів повідомляє OpenCV, яким чином отримувати контури. Є чотири можливих значення для режиму:

cv.RETR_EXTERNAL – отримуються лише крайні зовнішні контури;

cv.RETR_LIST – отримуються всі контури і поміщуються у список, у цьому випадку всі контури знаходяться і з'єднуються між собою ієрархією;

cv.RETR_CCOMP – отримуються всі контури та організуються у дворівневу ієрархію, де межі верхнього рівня – це зовнішні межі компонентів та межі другого рівня – це межі отворів.

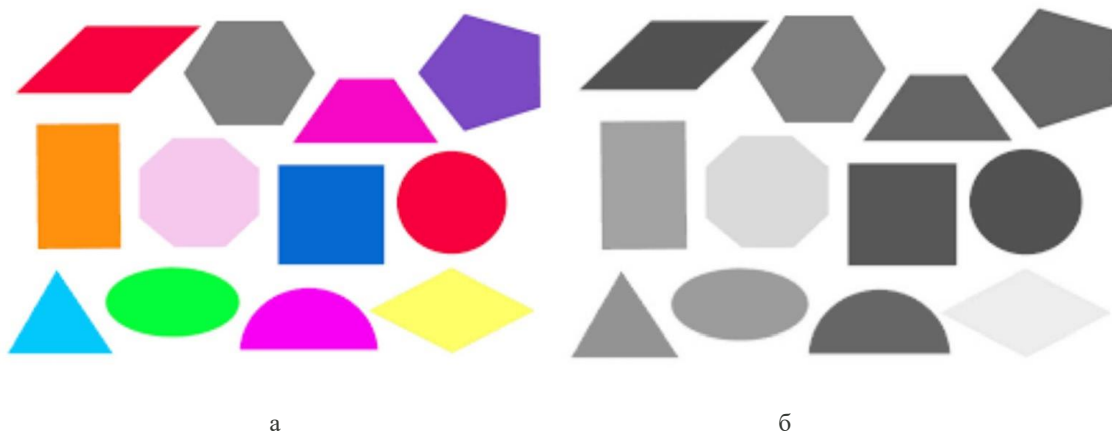
cv.RETR_TREE – отримуються всі контури і відновлюється повна ієрархія вкладених контурів.

- **method:** визначає метод апроксимації контуру (тобто яким чином представлено контур):
 - cv.CHAIN_APPROX_NONE – перетворює всі точки з коду контурів у точки. Ця операція буде продукувати велику кількість точок, оскільки кожна точка буде одним із восьми сусідів попередньої точки. Не робиться ніяких спроб зменшити кількість вершин.
 - cv.CHAIN_APPROX_SIMPLE – стискає горизонтальні, вертикальні та діагональні відрізки, залишаючи лише їх кінцеві точки. Для багатьох особливих випадків це може призвести до значного зменшення кількості точок.
 - cv.CHAIN_APPROX_TC89_L1 або cv.CHAIN_APPROX_TC89_KCOS – застосовує один із варіантів алгоритму апроксимації Teh-Chin, що застосовується для зменшення кількості точок.
- **cv.approxPolyDP** (curve, approxCurve, epsilon, closed) – дана функція апроксимує криву або багатокутник з іншою кривою / багатокутником з меншою кількістю вершин, так що відстань між ними менша або дорівнює заданій точності. Для реалізації цієї функції використовується алгоритм Ramer-Douglas-Peucker [23].

Аргументи функції:

- **curve** – вхідний вектор 2D точки;
- **approxCurve** – результат апроксимації, тип результату повинен співпадати з типом вхідної кривої;
- **epsilon** – параметр для визначення точності апроксимації, яка вимірюється максимальною відстанню між оригінальною кривою та її апроксимацією;
- **closed** – параметр, значення якого є «Істинно» у випадку, коли апроксимована крива замкнута (перша і остання вершини кривої об'єднані), і «Хибно», коли апроксимована крива не замкнута.

Проведене моделювання розпізнавання фігур у зображенні на прикладі (рис. 10) дало наступні результати:



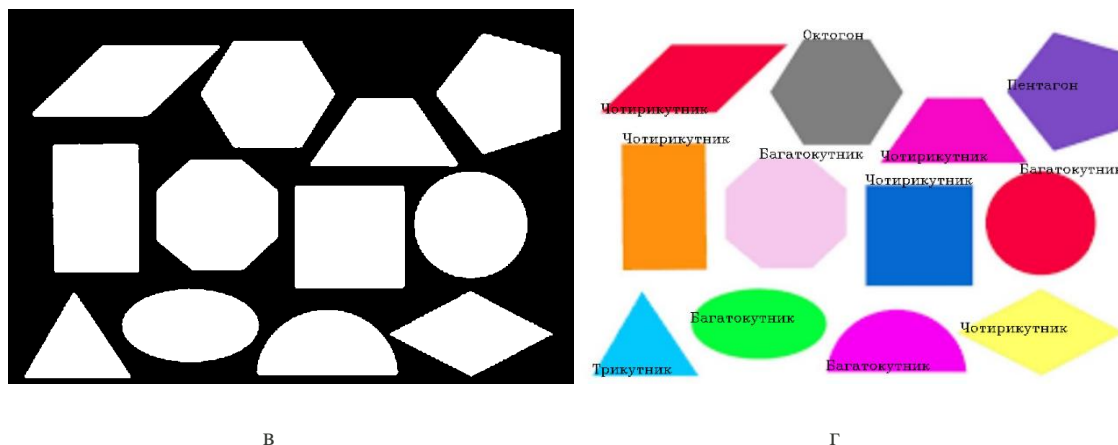


РИС. 10, а – вхідне кольорове зображення; б – зображення в градаціях сірого; в – бінаризоване зображення; г – зображення з розпізнаними геометричними фігурами

Програмна модель розпізнає геометричні фігури у зображенні та підраховує їх кількість. За результатами видно, що в зображенні трикутників – 1 шт., чотирикутників – 5 шт., шестикутників – 1 шт. і решта багатокутників. Загальна кількість фігур – 12 шт. При потребі більш детальної класифікації (квадрат, прямокутник, трапеція тощо) застосовуються додаткові умовні оператори.

```
"C:\Python\Python 3.7.6\python.exe" "E:\Python My Scripts\Project_Object shape/shape_detection.py"
```

```
Number of contours = 12
```

```
contour approx = 8
```

```
contour approx = 3
```

```
contour approx = 4
```

```
contour approx = 12
```

```
contour approx = 4
```

```
contour approx = 14
```

```
contour approx = 8
```

```
contour approx = 4
```

```
contour approx = 4
```

```
contour approx = 4
```

```
contour approx = 6
```

```
contour approx = 5
```

В процесі досліджень було:

- проведено аналіз існуючих методів сегментації зображень;
- розроблено програмну модель визначення оптимального порогу класифікації зображень;
- проведено аналіз застосування методів вибору порогової класифікації до задач розпізнавання фігур у зображенні та запропоновано підхід до його вибору;
- розроблено програмну модель визначення центроїду фігур довільної форми в зображеннях;
- розроблено програмну модель підрахунку фігур в зображенні та їх розпізнавання.

Отримані результати можуть бути використані у багатьох застосуваннях комп'ютерного зору, наприклад, для підрахунку об'єктів у сцені, визначення їх параметрів, визначення дистанції між об'єктами у сцені та інше.

Список літератури

1. Zhang D.S. Review of shape representation and description techniques. *J. Pattern Recognition*. 2004. P. 1–19. <https://doi.org/10.1016/j.patcog.2003.07.008>
2. Golovin O. Image Enhancement in Video Analytics Systems. *Control Systems and Computers*. 2020. 6. P. 3–17. <https://doi.org/10.15407/csc.2020.06.003>
3. Glasbey C.A. An analysis of histogram-based thresholding algorithms. *CVGIP: Graphical Models and Image Processing*. 1993. 55. P. 532–537. <https://doi.org/10.1006/cgip.1993.1040>
4. Sezgin M., Sankur B. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*. 2004. 13 (1). P. 146–156. <https://doi.org/10.1117/1.1631315>
5. Otsu N. A threshold selection method from gray-level histogram, *IEEE Transactions on System Man Cybernetics*. 1979. Vol. SMC-9, No. 1. P. 62–66. <https://doi.org/10.1109/TSMC.1979.4310076>
6. Moghaddam R., Cheriet M. Adotsu: an adaptive and parameterless generalization of Otsu's method for document image binarization, *Pattern Recogn.* 2012. 45. P. 2419–2431. <https://doi.org/10.1016/j.patcog.2011.12.013>
7. Xu X., Xu S., Jin L., Song E. Characteristic analysis of Otsu threshold and its applications, *Pattern Recogn. Lett.* 2011. 32. P. 956–961. <https://doi.org/10.1016/j.patrec.2011.01.021>
8. Sirisha P., Raju C., Reddy R., An efficient fuzzy technique for detection of brain tumor. *J. Softw. Eng.* 2013. 7. <https://doi.org/10.26634/jse.7.4.2316>
9. Alsaeed D.H., Bouridane A., Elzaart A., Sammouda R., Two modified Otsu image segmentation methods based on lognormal and gamma distribution models, in: 2012 International Conference on Information Technology and e-Services (ICITeS), IEEE. 2012. P. 1–5. <https://doi.org/10.1109/ICITeS.2012.6216680>
10. Cai H., Yang Z., Cao X., Xia W., Xu X. A new iterative triclass thresholding technique in image segmentation, *IEEE Trans. Image Process.* 2014. 23. P. 1038–1046. <https://doi.org/10.1109/TIP.2014.2298981>
11. Lai Y.K., Rosin P.L. Efficient circular thresholding, *IEEE Trans. Image Process.* 2014. 23. P. 992–1001. <https://doi.org/10.1109/TIP.2013.2297014>
12. Xue J., Titterington D. t-tests, f-tests and Otsu's methods for image thresholding, *IEEE Trans. Image Process.* 2011. 20. <https://doi.org/10.1109/TIP.2011.2114358>
13. Arora S., Acharya J., Verma A., Panigrahi P.K. Multilevel thresholding for image segmentation through a fast statistical recursive algorithm. *Pattern Recognition Letters*. 2008. 29. P. 119–125. <https://doi.org/10.1016/j.patrec.2007.09.005>
14. Dong L.J., Yu G., Ogunbona P., Li W.Q. An efficient iterative algorithm for image thresholding. *Pattern Recognition Letters*. 2008. 29. P. 1311–1316. <https://doi.org/10.1016/j.patrec.2008.02.001>
15. Huang D.Y., Wang C.H. Optimal multi-level thresholding using a two-stage Otsu optimization approach. *Pattern Recognition Letters*. 2009. 30. P. 275–284. <https://doi.org/10.1016/j.patrec.2008.10.003>
16. Liao P.S., Chen T.S., Chung P.C. A fast algorithm for multilevel thresholding. *Journal of Information Science and Engineering*. 2001. 17. P. 713–727.
17. Papamarkos N., Gatos B. A new approach for multilevel threshold selection. *Graphics Models Image Process.* 1994. 56. P. 357–370. <https://doi.org/10.1006/cgip.1994.1033>
18. Flusser J., Suk T., Zitová B. What are moments? Moments and Moment Invariants in Pattern Recognition. John Wiley & Sons Ltd, 2009. P. 6. ISBN 978-0-470-69987-4 <https://doi.org/10.1002/9780470684757>
19. Pouli T., Reinhard E., Cunningham D. *Image Statistics in Visual Computing*. CRC Press, 2014. P. 35. ISBN 978-1-4665-3982-2
20. Farin G. E. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press, Boston, Massachusetts, 4th edition. 1996.
21. Douglas D.H., Peucker T.K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 1973. 10 (2). P. 112–122. <https://doi.org/10.1002/9780470669488.ch2>
22. Ramer U. An Iterative Procedure for the Polygonal Approximation of Plane Curves. *Computer Graphics and Image Processing*. 1972. 1 (3). P. 244–256. [https://doi.org/10.1016/S0146-664X\(72\)80017-0](https://doi.org/10.1016/S0146-664X(72)80017-0)
23. Ramer–Douglas–Peucker algorithm. http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm (звернення: 28.12.2021)

Одержано 28.12.2021

Головін Олександр Миколайович,
кандидат технічних наук, старший науковий співробітник
Інституту кібернетики імені В.М. Глушкова НАН України, Київ.
<https://orcid.org/0000-0002-0279-812X>
o.m.golovin.1@gmail.com

UDC 364.2:331

Oleksandr Golovin

Recognition of Geometric Figures and Determination of Their Characteristics by Means of Computer Vision

V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv
Correspondence: o.m.golovin.1@gmail.com

Introduction. Many computer vision applications often use procedures for recognizing various shapes and estimating their dimensional characteristics. The entire pipeline of such processing consists of several stages, each of which has no clearly defined boundaries. However, it can be divided into low, medium, and high-level processes. Low-level processes only deal with primitive operations such as preprocessing to reduce noise, enhance contrast, or sharpen images. The processes of this level are characterized by the fact that there are images at the input and output. Image processing at the middle level covers tasks such as segmentation, description of objects, and their compression into a form convenient for computer processing. Middle-level processes are characterized by the presence of images only at the input, and only signs and attributes extracted from images are received at the output. High-level processing involves “understanding” a set of recognized objects and recognizing their interactions.

Using the example of the developed software models for recognizing figures and estimating their characteristics, it is shown that the image processing process is reduced to transforming spatial image data into metadata, compressing the amount of information, which leads to a significant increase in the importance of data. This indicates that at the input of the middle level, the image should be as informative as possible (with high contrast, no noise, artifacts, etc.) because after the transformation of the spatial image data into metadata, no further the procedures are not able to correct the data obtained by the video sensors in the direction of improving or increasing the information content.

Recognition of figures in an image can be realized quite efficiently through the use of the procedure for determining the contours of figures. To do this, you need to determine the boundaries of objects and localize them in the image, often the first step for procedures such as separating objects from the background, image segmentation, detection and recognition of various objects, etc.

The purpose of the article is to study the image processing pipeline from the moment of image fixation to the recognition of a certain set of figures (for example, geometric shapes, such as a triangle, quadrilateral, etc.) in an image, the development of software models for recognizing figures in an image, determining the center of mass figures by means of computer vision.

Results. We proposed and tested some variants of nonlinear estimating problem. The properties of such problems depend on value of regulating parameter. The dependence of estimation on value of parameter was studied. It was defined a range for parameter's value for which estimating problem gives adequate result for initial task.

Numerical examples show how much volume of calculations reduces when using a dynamic branching tree.

Conclusions. The results obtained can be used in many applications of computer vision, for example, counting objects in a scene, estimating their parameters, estimating the distance between objects in a scene, etc.

Keywords: contour, segmentation, image binarization, computer vision, histogram.