УДК 519.7, 004.92, 004.93 PACS 07.05.Bx, 07.05.Kf, 07.05.Pj, 07.05.Rm, 07.05.Tp

О.М. Підлісний, О.А. Сердюк

МОДЕЛІ ПРЕДСТАВЛЕННЯ СТРУКТУРИ ЗОБРАЖЕННЯ У ЗАДАЧАХ КОМП'ЮТЕРНОГО ЗОРУ

У статті описано методику вибору та побудови структури даних для оптимального внутрішнього подання морфологічної структури зображення з метою застосування у додатках комп'ютерного зору. Наведено інформацію про використовувані графічні фільтри для виділення контурів об'єктів зображення. Проведено аналіз програмних структур даних для представлення морфологічної структури зображення та обґрунтовано вибір запропонованої гібридної структури внутрішнього подання об'єктів. Наводиться інформація про тестування реалізації отриманої структури у додатку, побудованому з використанням графічної бібліотеки OpenCV та бібліотеки для побудови графічних інтерфейсів Qt.

Ключові слова: *комп'ютерний зір, морфологічний аналіз зображення, фільтрація зображення, структури даних, бібліотека OpenCV.*

Вступ

Останніми роками все більшої популярності набирають задачі комп'ютерного зору, пов'язані з обробкою інформації, наявної у зображеннях [2, 3, 6, 7, 8, 9]. Графічні документи можуть містити інформацію суттєво різної структури, а тому для кожного виду інформації треба використовувати свій підхід, що досить часто суттєво відрізняється від підходів для інших видів.

Одним з початкових етапів практично кожної системи комп'ютерного зору є аналіз структури зображення (морфологічний аналіз [7, 8]), після чого виділяються потрібні фрагменти й подальша робота проводиться вже з ними. Саме з точки зору подальшої результативної роботи системи за рахунок коректного вичленування вихідної інформації величезну важливість становить використання оптимальних структур даних, що надають можливість ефективно відобразити структуру зображення та ефективно проводити її аналіз.

Наразі існують спеціалізовані програмні засоби, розроблені для розпізнавання зображень, наприклад, ABBYY FineReader, проте вони мають свої суттєві обмеження. Основними проблемами таких застосунків є або низька якість розпізнавання певних видів інформації, або їх надмірна закритість і вартість.

Тим не менше, будь-який програмний комплекс комп'ютерного зору потребує ефективних методів аналізу структури зображення. Особливо це стосується систем, що працюють з зображеннями в реальному часі, наприклад, з відеопотоком.

Мета статті

Метою статті є опис алгоритму, що надає можливість отримати морфологічну структуру зображення, якою можна ефективно керувати у подальшому, використовуючи лише відкриті технології та методи, та перевірка можливості використання описаного підходу у реальних задачах комп'ютерного зору.

Основний алгоритм

Нехай є деяке відскановане чи сфотографоване зображення.

1. Важливим етапом обробки є очищення зображення, оскільки вихідне зображення документу містить шуми, спричинені якістю фотокамери, алгоритмом стиснення, тощо. Якщо працювати з зображенням у початковому вигляді, то неможливо коректно проаналізувати його структуру, а отже необхідно зробити відповідні пікселі одного відтінку максимально схожими між собою та прибрати інші шуми. Відповідні дії виконуються з використанням набору фільтрів.
2. Наступним кроком є пошук усіх контурів на зображенні, оскільки саме вони обмежують об’єкти на зображенні і є предметом дослідження.
3. Після виділення контурів знаходиться їх ієрархія, тобто, для кожного контура відшукуються такі, що лежать безпосередньо у ньому.

У результаті роботи алгоритму повинна бути побудована така ієрархія контурів, щоб користувач, вибравши яку-небудь точку на зображенні, отримав виділену послідовність контурів на зображенні та на ієрархічній схемі, всередині яких знаходиться вказана точка.

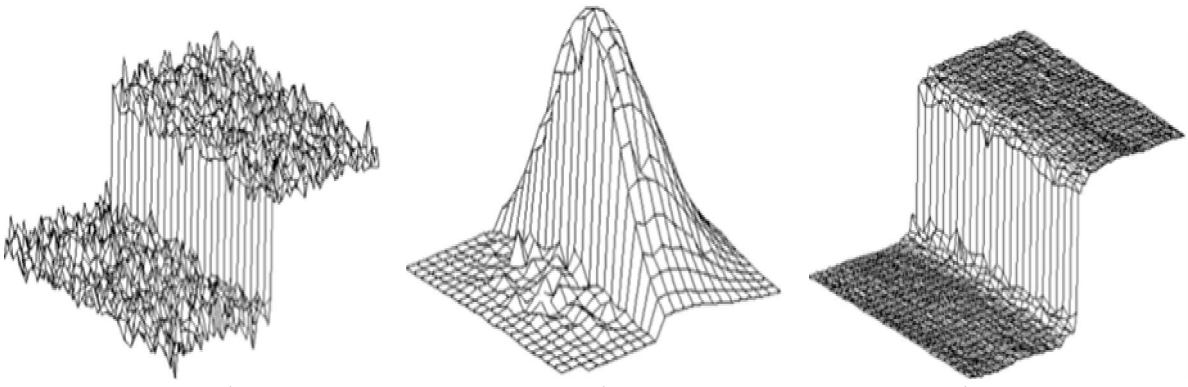
Очищення зображення

Найефективнішими для очищення зображення є білатеральний фільтр та Гаусове розмиття.

Білатеральний фільтр є нелінійним фільтром, що згладжує зображення, зберігаючи при цьому краї елементів зображення та зменшуючи шуми [5]. Значення інтенсивності кожного пікселя замінюється середнім зваженим інтенсивностей сусідніх пікселів. Вагові коефіцієнти можуть братись з Гаусового розподілу. Важливо зазначити, що вагові коефіцієнти залежать не лише від евклідової відстані між пікселями, а також і від радіометричних характеристик (інтенсивність кольору, глибина, тощо).

Білатеральний фільтр дуже схожий на Гаусове розмиття [2, 9]. Різниця полягає в тому, що білатеральний фільтр додатково враховує різницю значень сусідів, щоб зберегти краї при розмитті. Ключовою ідеєю цього фільтру є те, що кожен піксель для впливу на інший повинен не тільки бути його сусідом, а й мати близьке значення.

Приклад роботи фільтру наведено на рис. 1.



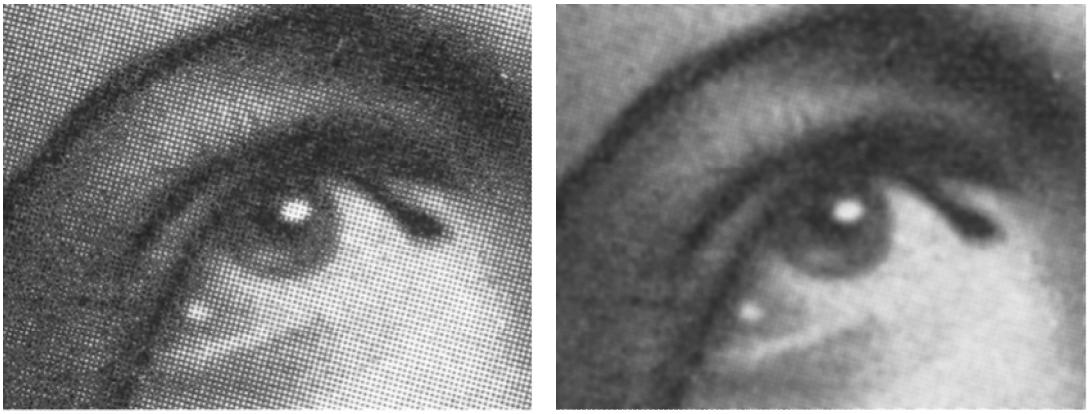
а) б) в)

Рис. 1. Робота білатерального фільтру: (а) початкове зображення; (б) функція Гауса з врахуванням інтенсивності; (в) результат роботи

Білатеральний фільтр реалізовано у бібліотеці OpenCV у функції bilateralFilter() [1, 4].

Розмиття Гауса – це метод розмивання зображення з використанням функції Гауса [2, 9]. Цей прийом широко використовується у графічних програмах – як правило, для зменшення шуму у зображенні та зниження деталізації. Візуальний ефект цієї техніки розмиття аналогічний погляду на зображення крізь напівпрозорий екран і суттєво відрізняється від ефекту Боке, який можна отримати за допомогою несфокусованого об'єктива або тіні об'єкта при звичайному освітленні. Приклад роботи фільтру наведено на рис. 2.

Розмиття Гауса у бібліотеці OpenCV реалізовано кількома способами, проте окремо його можна використати завдяки методу GaussianBlur() [1, 4].



а) б)

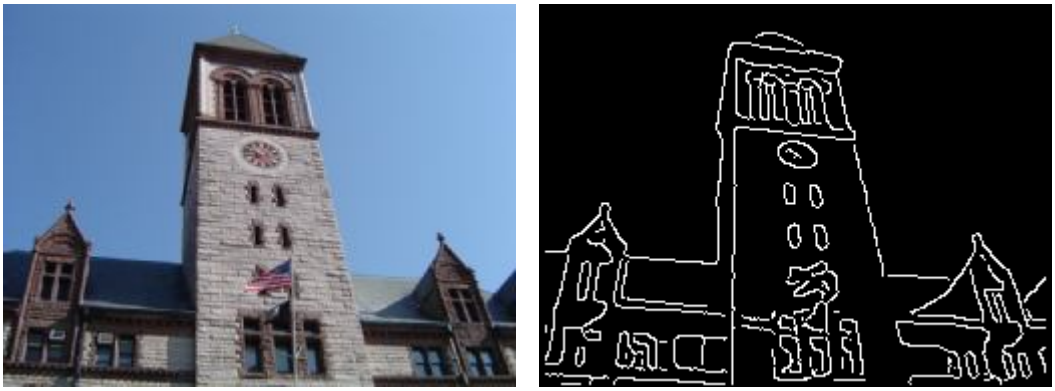
Рис. 2. Робота фільтру Гаусового розмиття: а) початкове зображення; б) розмите зображення

Виділення контурів

Одним із способів виділення контурів на зображенні є використання *оператора Кенні*, розробленого у 1986 році Джоном Кенні [2]. Оператор використовує багатоступеневий алгоритм для виявлення широкого спектру меж у зображеннях і особливо інтенсивно використовується у застосуваннях комп'ютерного зору.

Працюючи над задачею отримання фільтру, оптимального за критеріями виділення, локалізації та мінімізації кількох відгуків одного краю, Кенні показав, що шуканий фільтр являє собою суму чотирьох експонент. Одним з отриманих результатів став доказ того, що цей фільтр може бути добре наближений першою похідною Гаусової функції. Кенні ввів поняття подавлення немаксимумів (non-maximum suppression), що означає, що пікселями меж оголошуються пікселі, в котрих досягається локальний максимум градієнта у напрямку вектора градієнта. Хоча його робота була проведена на етапі становлення напрямку комп'ютерного зору, детектор Кенні досі залишається одним з найкращих детекторів.

Приклад роботи оператора Кенні наведено на рис. 3.



а) б)

Рис. 3. Робота оператора Кенні: а) початкове зображення; б) отримане зображення з межами об’єктів.

Оператор Кенні реалізовано у бібліотеці OpenCV у функції Canny() [1, 4].

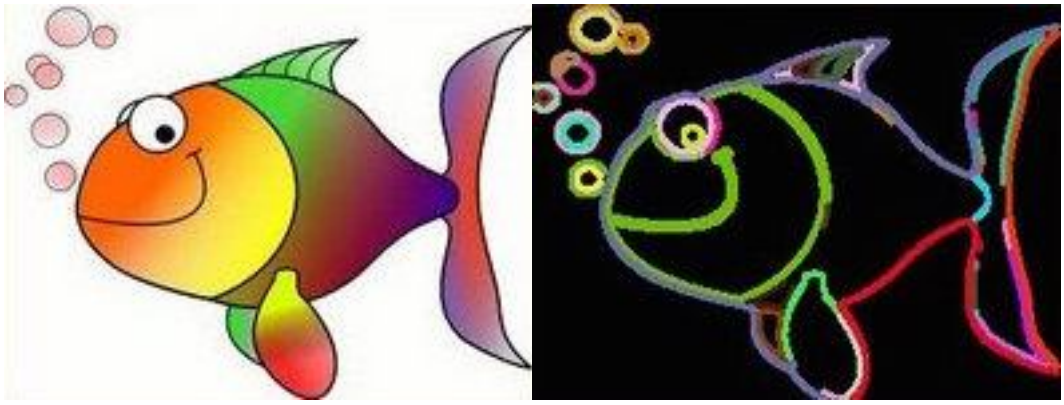
Для покращення якості отриманого після застосування оператора Кенні зображення використовується *морфологічний оператор* dilate() [4] з ядром

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Оптимізація контурів

Отримані на попередньому кроці контури необхідно перетворити у масиви точок, після чого максимально апроксимувати до набору ламаних.

Для пошуку контурів на отриманому монохромному зображенні доцільно використати алгоритм Сузукі (S. Suzuki) [7, 8]. У бібліотеці OpenCV цей алгоритм реалізовано у вигляді методу findContours() [1, 4]. Приклад роботи алгоритму, котрий і використовується при обробці зображення, наведено на рис. 4.



а) б)

Рис. 4. Пошук контурів: а) початкове зображення; б) отримане зображення з межами об’єктів.

За допомогою вказаного методу, окрім саме контурів, можна отримати їх

ієрархію. Метод може надати інформацію про ієрархію у кількох різних виглядах, але для розв'язання поставленої задачі потрібна деревовидна ієрархія, яку можна отримати при використанні параметру CV_RETR_TREE у функції findContours(). [4] Використання параметру CV_CHAIN_APPROX_SIMPLE, що відповідає за спрощення контурів з сукупності точок до сукупності відрізків, надає можливість суттєво економити оперативну пам'ять. Проте, для максимальної апроксимації отриманого результату до набору ламаних можна використати значно ефективніший метод approxPolyDP() [4].

Представлення структури зображення

Оскільки кожен контур у отриманому на попередньому кроці наборі характеризує певну структурну одиницю (об'єкт) на зображенні, то усі контури разом представляють повну структуру зображення, яку необхідно подати у вигляді ефективної структури даних. Для цього було розглянуто кілька варіантів: масив контурів, дерево контурів, гібридний варіант.

Масив контурів є найпростішою структурою даних, оскільки всі контури знаходяться разом у одному масиві даних. Незважаючи на те, що перевагами такої структури є зручність перебору усіх контурів одним циклом та можливість швидкого пошуку контура за номером, вона має суттєві недоліки:

- ієрархію контурів потрібно зберігати окремим масивом даних;
- пошук усіх рівнів вкладеності для контуру можливий лише за умови зберігання інформації про ієрархію;
- пошук контуру, що містить певну точку, вимагає $\theta(n)$ дій;
- повільний пошук сусідніх контурів.

Дерево контурів має певні переваги порівняно з масивом контурів: зручно перебирати контури, що лежать у якомусь певному контурі; пошук контуру, що містить певну точку, є значно швидшим, оскільки немає необхідності перебирати всі контури; не потрібно зберігати окремо ієрархію; наявність можливості швидкого та зручного пошуку сусідніх контурів. Однак, і ця структура даних не позбавлена недоліків:

- незручно перебирати усі контури підряд одним циклом, якщо, наприклад, необхідно зобразити всі контури на зображенні;
- незручний пошук контура за номером.

Для позбавлення вказаних вище недоліків використовується гібридний варіант (рис. 5), де наявні і масив контурів, і дерево, але дерево не містить інформації про точки контуру, а лише вказівники на контур у масиві та масив вказівників на контури, що лежать всередині даного контуру. Тобто, елемент дерева містить:

- вказівник на контур у масиві;
- масив вказівників на контури, що лежать у даному контурі.

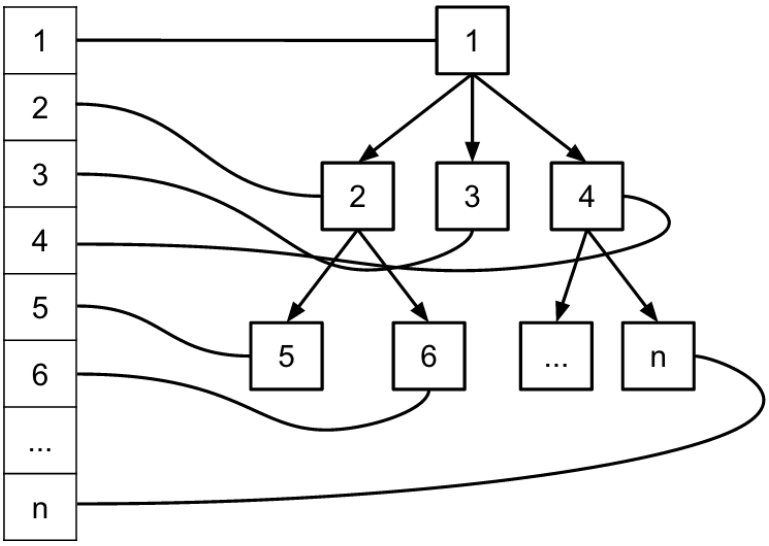


Рис. 5. Гібридна структура збереження контурів об’єктів зображення.

Гібридний варіант має досить багато переваг над іншими розглянутими раніше структурами. Проте, бувають випадки, коли компоненти зображення, що за логікою мали б бути на одному рівні, опиняються на різних рівнях. Наприклад, якісь шуми чи плями на фото, сонячні відблиски, тощо. Більше того, такими причинами можуть бути не лише деякі недоліки зображення, але й деякі об’єкти, як, наприклад, мокра пляма на асфальті чи тінь. За рахунок цього створюється додатковий компонентний рівень. А отже, робота з компонентами одного рівня у даному випадку не враховує компоненти, що насправді мали б бути на цьому рівні. Для коректної роботи з отриманими рівнями можна доповнити структуру так званими непрямими зв’язками, що є міжрівневими зв’язками, котрі показують, що у даної компоненти є, окрім прямої батьківської компоненти, ще компоненти, котрим вона належить (рис. 6).

Такі зв’язки можна зберігати, у залежності від потреб, як у батьківській компоненті (додається інформація про усі компоненти, що якимось чином лежать у даній), так і у компоненті-спадкоємці (додаткова інформація про усі батьківські компоненти). Звісно, доцільно зберігати й рівень зв’язку, і навіть зберігати прямі й непрямі зв’язки у різних властивостях компоненти для швидшого алгоритмічного обходу лише тих чи інших.

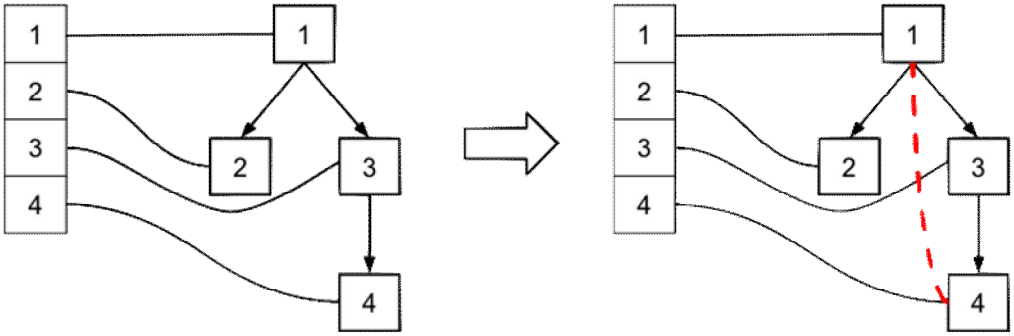


Рис. 6. Доповнена гібридна структура збереження контурів об’єктів зображення.

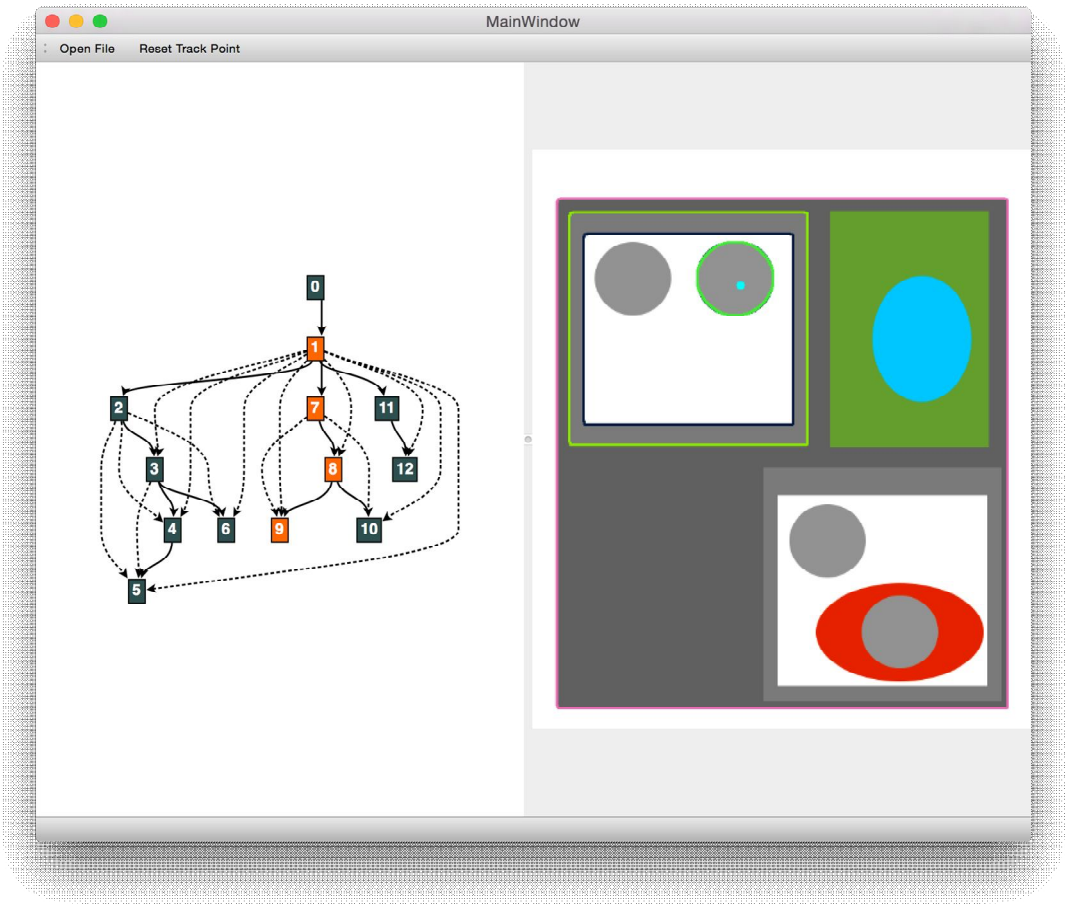


Рис. 7. Головне вікно програмного продукту, розробленого для перевірки працездатності описаної у роботі методики.

Така структура має досить суттєвий недолік: збільшення витрат пам'яті для зберігання структур під час роботи. Для уникнення такої ситуації доцільно доповнювати лише ті зв'язки, де це справді необхідно. Наприклад, тоді, коли у певних областях зображення звичайна гібридна структура не дала потрібного результату.

Однак, головною перевагою такого способу збереження контурів є обробка значно більшої кількості можливих комбінацій контурів на зображенні.

Тестування описаного підходу

Для тестування можливості збереження структури зображення описаним вище чином було розроблено програмний продукт, що має максимально простий інтерфейс (рис. 7).

У результаті роботи розроблено окрему бібліотеку аналізу зображення для отримання структури зображення, що реалізує вищезазначені структури даних. А отже, це надає `можливість` легко застосувати її в іншому програмному продукті чи при використанні іншого інструментарію розробки, наприклад, Microsoft Visual Studio.

Висновки

У ході виконання роботи було розроблено та реалізовано авторський алгоритм структурного аналізу та розпізнавання зображення. Алгоритм виконує пошук контурів на зображенні, після чого аналізує їх взаємне розміщення та будує відповідне

відображення у пам'яті ЕОМ у вигляді оптимальних структур даних. Алгоритм базується на відкритих алгоритмах та технологіях, що надає можливість використовувати його у будь-яких цілях незалежно від предметної області.

Розроблене програмне забезпечення, що використовує описаний підхід, демонструє близьке до оптимального використання оперативної пам'яті ЕОМ при збереженні структури аналізованого зображення та високу, порівняно з іншими методами, швидкість вибору ієрархії об'єктів, пов'язаних між собою. Використання лише відкритих технологій та інструментарію розробки надає можливість користуватись отриманою бібліотекою при розв'язанні багатьох задач морфологічного аналізу зображення, а реалізація програмного продукту мовою С++ з використанням інструментарію розробки Qt [10] дає можливість з легкістю перенести розроблене програмне забезпечення на будь-яку популярну комп'ютерну платформу.

Список використаної літератури

1. Bradski G. Learning OpenCV. Computer Vision with the OpenCV Library / G. Bradski, A. Kaehler – First Edition. – O'Reilly Media, September, 2008. – 577 p.
2. Burger W. Principles of Digital Image Processing. Advanced Methods / W. Burger, M.J. Burge – London: Springer-Verlag, 2013. – 374 p.
3. Davies E. Computer and Machine Vision. Theory, Algorithms, Practicalities / E.R. Davies. – Elsevier Inc., 2012. – 912 p.
4. OpenCV Documentation [Електронний ресурс]: Режим доступу: http://docs.opencv.org/
5. Paris S. Bilateral Filtering: Theory and Applications / S. Paris, P. Kornprobst, J. Tumblin, F. Durand. // Foundation and Trends in Computer Graphics and Vision – 2009. – Vol. 4, No. 1. –pp. 1-73.
6. Parker J. R., Algorithms for Image Processing and Computer Vision / J. R. Parker. – Wiley Publishing, 2010. – 249 p.
7. Soille P. Morphological Image Analysis. Principles and Applications / P. Soille – Springer-Verlag Berlin Heidelberg, 2004. – 399 p.
8. Soille P. Mathematical Morphology and Its Applications to Image and Signal Processing / P. Soille, M. Pesaresi, G.K. Ouzounis // 10th International Symposium, ISMM 2011. Proceedings. –Berlin, Heidelberg: Springer-Verlag, 2011. – 494 p.
9. Treiber M. An Introduction to Object Recognition. Selected Algorithms for a Wide Variety of Applications / M. Treiber. – London: Springer-Verlag, 2010. – 216 p.
10. Саммерфилд М. Qt. Профессиональное программирование. Разработка кроссплатформенных приложений на С++ / М. Саммерфилд – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 560 с., ил. ISBN 978-5-93286-207-0.

References

1. Bradski G., Kaehler A. (2008). Learning OpenCV. Computer Vision with the OpenCV Library, *First Edition*, O'Reilly Media.
2. Burger W., Burge M.J. (2013). Principles of Digital Image Processing. Advanced Methods, London: Springer-Verlag.
3. Davies E. (2012). Computer and Machine Vision. Theory, Algorithms, Practicalities, Elsevier Inc.
4. OpenCV Documentation: http://docs.opencv.org/.
5. Paris S., Kornprobst P., Tumblin J., Durand F. (2009). Bilateral Filtering: Theory and Applications, *Foundation and Trends in Computer Graphics and Vision*, *V.4*, *No.1*, 1-73.
6. Parker J.R. (2010). Algorithms for Image Processing and Computer Vision, Wiley

Publishing.
7. Soille P. (2004) Morphological Image Analysis. Principles and Applications, Berlin Heidelberg: Springer-Verlag.
8. Soille P., Pesaresi M., Ouzounis G.K. (2011). Mathematical Morphology and Its Applications to Image and Signal Processing, *10th International Symposium, ISMM 2011. Proceedings*, Berlin, Heidelberg: Springer-Verlag.
9. Treiber M. (2010). An Introduction to Object Recognition. Selected Algorithms for a Wide Variety of Applications, London: Springer-Verlag.
10. Sammerfild M. (2011). Professional'noe programmirovanije. Razrabotka krossplatformennih prilozhenij na C++, SPb: Simvol-Plus.

Summary

O.M. Pidlisnyi, O.A. Serdyuk

THE MODELS OF REPRESENTATION OF THE IMAGE STRUCTURE IN COMPUTER VISION TASCS

The article describes a method of selecting and constructing an optimum data structure for the internal representation of the morphological structure of the picture for use in computer vision applications. The information about the use of graphic filters to highlight the contours of image objects is provided. The analysis of program data structures to represent the morphological structure of the image and justified the choice of the proposed hybrid structure of the internal representation of the objects is described. The articles provides the information on the implementation of the resulting test structures in the application, which is built using OpenCV graphic library and libraries for building graphical interfaces Qt.

Keywords: *computer vision, morphological analysis of image, image filtering, data structures, OpenCV library.*

Стаття надійшла _03_11_2015
Прийнято до друку _ 27_11_2015