

Використання Суперпікселів в Ефективному Алгоритмі Сегментації Зображень на Основі Графу

Дмитро Луп'як
кафедра автоматки та інформаційно-вимірювальної
техніки
Вінницький національний технічний університет
Вінниця, Україна
dima.lupyak@gmail.com

Роман Кветний
кафедра автоматки та інформаційно-вимірювальної
техніки
Вінницький національний технічний університет
Вінниця, Україна
rkvetny@sprava.net

Superpixels Using for Efficient Graph Based Image Segmentation Algorithm

Dmytro Lupaik
dept. of Automation and Information-Measuring
Technologies
Vinnytsia National Technical University
Vinnytsia, Ukraine
dima.lupyak@gmail.com

Roman Kvyetnyy
dept. of Automation and Information-Measuring
Technologies
Vinnytsia National Technical University
Vinnytsia, Ukraine
rkvetny@sprava.net

Анотація—Досліджено аспекти ефективної реалізації ефективного алгоритму сегментації зображень на основі графу. Запропоновано модифікацію алгоритму з використанням суперпікселізації, яка дозволила уникнути пересегментації зображення.

Abstract—The aspects of the efficient graph based image segmentation algorithm implementation are investigated. Modification of algorithm with superpixels using is proposed, that allowed to avoid image oversegmentation.

Ключові слова—компонент; сегментація зображень, кластеризація, теорія графів, відмінність кольорів, мінімальне кістякове дерево, комп'ютерне бачення.

Keywords—image segmentation, clustering, graph theory, color difference, minimum spanning tree, computer vision.

I. ВСТУП

В даний час простежується тенденція розширення сфери застосування систем комп'ютерного зору для вирішення прикладних задач в різних областях людської діяльності. Наприклад, задача середнього рівня, така як визначення траєкторії руху об'єкта, вимагає певну область підтримки для виконання відповідних розрахунків. Дослідження шляхів підвищення швидкодії та покращення якості сегментації зображення дозволить зменшити час прийняття рішення інтелектуальною системою.

Сегментація зображення представляє собою розбиття кадру на регіони за подібністю характеристик їх пікселів. В більшості випадків поділ на регіони проводять по яскравості для чорнобілого кадру та по кольорним характеристикам для зображення в кольорі. В деяких методах реалізована сегментація заснована на контурах. В таких методах за ознаку подібності виступає текстура, або форма.

На сьогодні існує велика кількість різних способів поділу зображення на частини з використанням різних математичних моделей. Одними напрямів в сегментації зображень, що активно розвивається є методи теорії графів.

Ідея методів даного класу є наступною. Зображення представляється у вигляді зваженого графа, пікселі зображення виступають в якості вершин. Вага ребра графа показує подібність точок по деякій властивості (відстань між точками за деякою метрикою). Поділ зображення відбувається розрізами графа.

В багатьох методах теорії графів вводиться функціонал вартості розрізу, що представляє якість вихідної сегментації. Таким чином задача поділу зображення на однорідні регіони зводиться до задачі оптимізаційної, а саме пошуку розрізу мінімальної вартості на графі. Такий підхід дає змогу крім текстури та однорідності кольору

сегментів керувати також формою сегментів, складністю меж, їх розміром, тощо.

Серед сегментації на основі представлення зображення у вигляді графу варто виділити алгоритми на основі побудови мінімального кістякового дерева. Дані алгоритми мають значно меншу обчислювальну складність, ніж перші алгоритми на основі графу в основі яких лежить операція розрізу графу.

Ефективний графовий алгоритм сегментації зображення (ЕГАСЗ) [1] відносять до групи алгоритмів, які виконують поділ зображення за допомогою побудови мінімального каркаса графа. Для знаходження меж між сегментами використовується предикат порівняння пари регіонів, а для побудови мінімального каркаса графа – алгоритм Краскала.

У роботі [2] запропоновано спосіб, як реалізувати розпаралелення обчислень для цього алгоритму та запропоновано новий предикат для виявлення меж між регіонами.

У роботі [3] запропоновано використовувати структуру даних DSD з евристикami «unionbyrank» і «pathcompression».

У роботі [4] запропоновано модифікацію з наближеним сортуванням ребер графу та представленням графу у вигляді масиву однонаправлених списків, яка є в 4 рази швидше за базову реалізацію алгоритму.

II. ВИКОРИСТАННЯ СУПЕРПІКСЕЛІВ

Одною з проблем даного алгоритму є сегментація текстурованих зображень. В оригінальному алгоритмі кожен піксель зображення представляє собою окремий вузол графу та поєднується лише з сусідніми пікселями. Проте, якщо сегмент має певну текстуру і деякі перепади кольору, то вага між сусідніми пікселями одного сегменту буде велика і це приводить до пересегментації в текстурованих ділянках зображення.

Для усунення цієї проблеми пропонуться здійснювати побудову суперпікселів і вагу ребер, що входять до суперпікселя робити рівну нулю, тобто об'єднувати пікселі в межах суперпікселя в один сегмент.

Для побудови суперпікселів обрано алгоритм SLIC (Simple Linear Interactive Clustering) [5]. Основна ідея алгоритму сегментації SLIC полягає в кластеризації пікселів в обмежених областях, на які розбивається зображення. Кожна точка зображення характеризується п'ятимірним вектором (1).

$$p = (c_1, c_2, c_3, x, y)^T, \quad (1)$$

де c_1, c_2 і c_3 – координати точки зображення в вибраному кольоровому просторі;

x та y – просторові координати.

Алгоритм включає наступні кроки:

1) Зображення розбивається на K фрагментів розміру $a \times a$, котрі задають початкове приближення суперпікселів.

В якості початкових центрів суперпікселів обираються їх геометричні центри C_k .

2) Корегуються координати центрів фрагментів з умови мінімального значення колірної градієнту в 3×3 окрузі геометричного центру.

3) Формування локальних кластерів в $2a \times 2a$ окрузі центрів C_k аналогічно методу k -середніх.

4) Визначення нових центрів суперпікселів та розрахунок зміщення центрів.

5) Повторення кроків 3 та 4 до тих пір, поки зміщення центрів між ітераціями не стане менше вказаного значення.

На рис. 1 наведено результат алгоритму SLIC.



Рис. 1. Результуючі суперпікселі отримані з використанням алгоритму SLIC

На основі модифікації ефективного алгоритму сегментації на основі графів з наближеним сортуванням ребер графу та представленням графу у вигляді масиву однонаправлених списків [4] зробимо модифікацію з використанням суперпікселів.

В базовому алгоритмі кількість ребер можна знайти за формулою 2.

$$n = 2ab - (a+b), \quad (2)$$

де a, b – кількість пікселів по ширині та висоті.

В модифікації з використанням суперпікселів кількість ребер можна знайти за формулою 3

$$n = 2N - 2\sqrt{N}, \quad (3)$$

де N – кількість суперпікселів (виступає параметром алгоритму).

Нехай вхідне зображення є розміром 500×500 пікселів, кількість суперпікселів задамо 900. З використанням базового алгоритму, відповідно до рівняння (2) буде створено 499000 ребер. З використанням суперпікселів незалежно від розміру зображення, відповідно до рівняння (3) буде створено 1740 ребер. Така різниця в кількості ребер повинна значно збільшити швидкість алгоритму сегментації, але при цьому побудова суперпікселів може зайняти певний час.

III. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

Для дослідження впливу різних модифікацій на оригінальний алгоритм була розроблена комп'ютерна програма, що дозволяє завантажити вхідне зображення, вибрати модифікацію алгоритму, виконати сегментацію, відобразити результат та показати час роботи кожного з етапів алгоритму.

В програмі є можливість змінювати структуру даних, що використовується для представлення графу, спосіб побудови графу, спосіб сортування ребер графу, предикат визначення меж сегменту, матрику різниці між пікселями, параметри алгоритму, кількість суперпікселів.

Інтерфейс розробленої програми для дослідження наведений на рис. 2

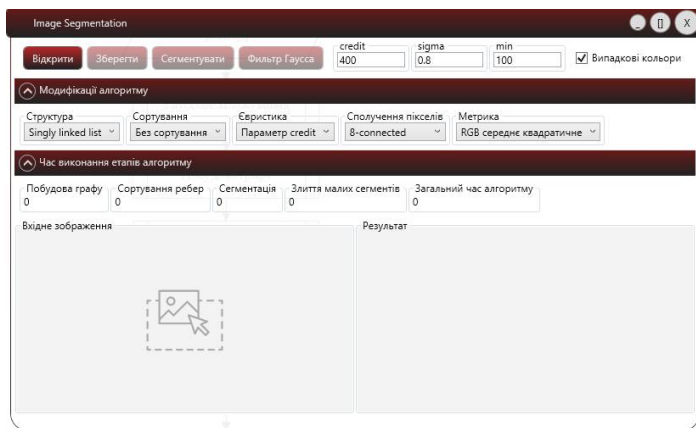


Рис. 2. Розроблена програма для дослідження впливу модифікацій на роботу алгоритму сегментації

В роботі [4] завдяки даної розробленої програми було досліджено вплив структури даних та методу сортування ребер на швидкодію алгоритму сегментації, заміряно час роботи алгоритму при різних модифікаціях для зображень різного розміру та побудовано графіки, що показують вплив модифікації на час виконання алгоритму (рис.3)

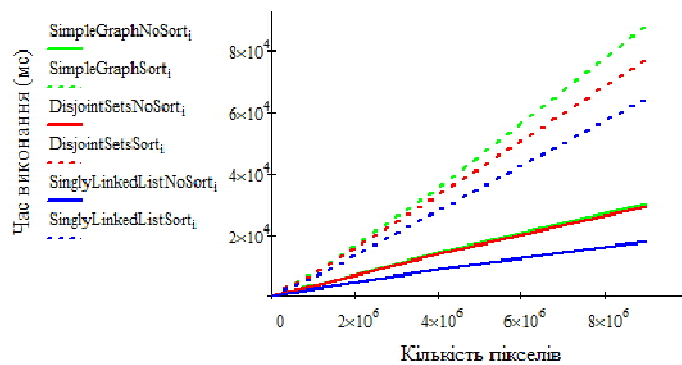


Рис. 3. Залежність часу виконання модифікації алгоритму від кількості пікселів у вхідному зображенні

З графіку видно, що використання динамічних списків найменш ефективне, використання DSD (системи неперетних множин) дає краще значення швидкодії, проте

найкращий результат був отриманий при використанні однонаправлених зв'язаних списків.

На швидкодію алгоритму сильно впливає алгоритм сортування, що є складовою частиною алгоритму Краскала. Навіть використання швидкого сортування призводить до великих затрат часу через кількість ребер в графі. Проте якщо значення ваг заокруглити до цілочисельного значення, то можна використати сортування методом підрахунку, що дозволить провести наближене сортування ребер за лінійний час.

Окрім цього на швидкодію впливає метод побудови множини ребер графу, при чотирьохпіксельному зв'язуванні швидкодія приблизно в 1.4 рази більша ніж при восьмипіксельному зв'язуванні.

Модифікація алгоритму з використанням масиву однонаправлених зв'язаних списків для представлення сегментів та наближеним сортуванням ребер графу за лінійний час збільшує швидкодію алгоритму приблизно в 4 рази при невеликій зміні якості сегментації. Для подальших досліджень, а саме впливу використання суперпікселів на роботу алгоритму, за основу була взята данна модифікація.

A. Дослідження впливу використання суперпікселів на швидкодію

Побудуємо графіки залежності швидкодії від кількості пікселів для оригінального алгоритму, модифікації без використання суперпікселів та модифікації з використанням суперпікселів (рис.4).

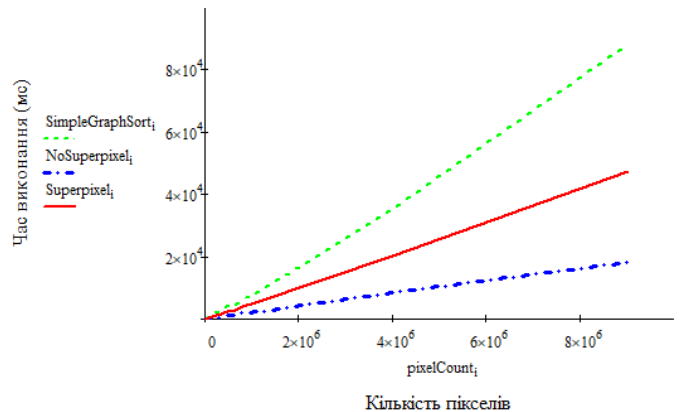


Рис. 4. Залежність часу виконання модифікації алгоритму від кількості пікселів у вхідному зображенні

З графіку видно, що побудова суперпікселів займає досить багато часу і швидкодія алгоритму менша в два рази порівняно з модифікованим алгоритмом[4] без використання суперпікселів, проте, якщо порівнювати з оригінальним алгоритмом така модифікація з використанням суперпікселів показує вдвічі кращу швидкодію.

B. Дослідження впливу використання суперпікселів на якість сегментації

Результати сегментації в залежності від використання суперпікселів наведені на рисунку 5.

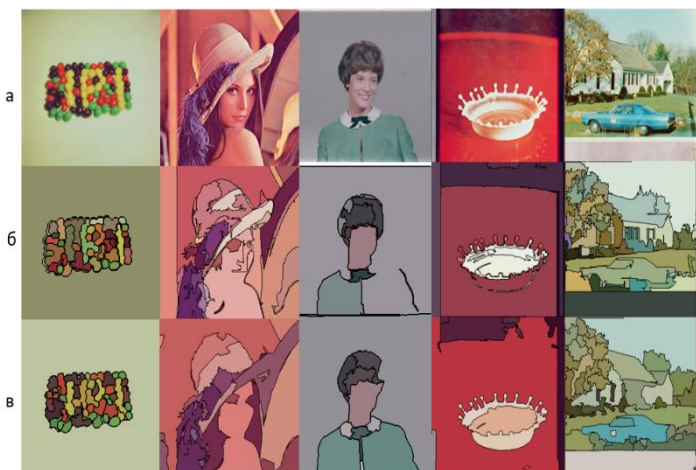


Рис. 5. Результат сегментації зображення в залежності від використання суперпікселів (а – вхідне зображення; б – сегментоване зображення без використання суперпікселів; в – сегментоване зображення з використанням суперпікселів)

Базуючись на результатах наведених на рис. 5 можна стверджувати, що використання суперпікселів покращує якість сегментації: сегменти точніше відповідають границям реальних об'єктів, краї сегментів більш чіткі, менша кількість надлишкових дрібних сегментів.

С. Дослідження впливу використання суперпікселів на якість сегментації

Для дослідження було використано стандартні тестові зображення з бази USC-SIP1 від Інституту обробки сигналів і зображень Університету Південної Каліфорнії [6]. Для декількох зображень з бази виконувалась сегментація з різним значенням параметру кількості сегментів. Результати сегментації при кількості суперпікселів 10, 100, 500, 1000 та 5000 наведені на рис.6 та рис. 7.

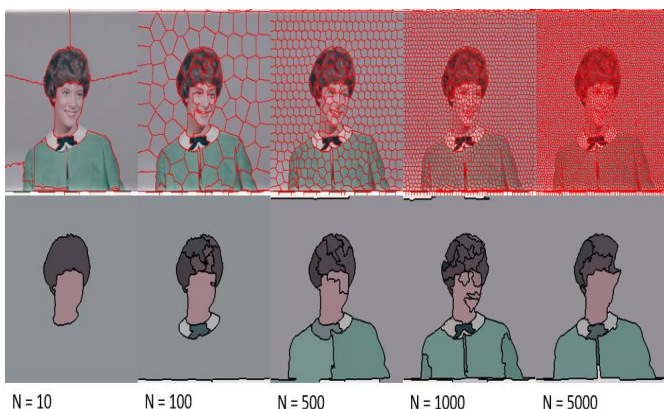


Рис. 6. Результат сегментації зображення в залежності від величини і кількості суперпікселів

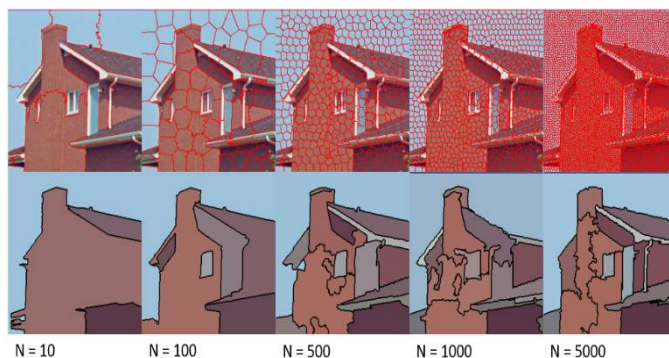


Рис. 7. Результат сегментації зображення в залежності від величини і кількості суперпікселів

Можна зробити висновок, що завдяки зміні розміру сегментів можна контролювати виділення дрібних об'єктів. При великому розмірі сегментів можна отримати межі певного об'єкту. При використанні дрібних сегментів, окрім меж об'єкту, можна отримати межі складових об'єкту.

ВИСНОВКИ

Запропонована модифікація ефективного алгоритму сегментації зображення на основі графу дозволяє уникнути пересегментації на текстурованих ділянках зображення, що досягається завдяки побудові суперпікселів і їхньому використанню в якості вузлів графу.

При комбінуванні даної модифікації з модифікацією розробленою в роботі [4] швидкість алгоритму вдвічі перевищує оригінальний алгоритм.

Оскільки кількість суперпікселів є сталою, інші параметри алгоритму є менш залежними від розміру вхідного зображення, що робить даний алгоритм більш універсальним.

ЛІТЕРАТУРА REFERENCES

- [1] Felzenszwalb P. F. Efficient graph-based image segmentation / P. F. Felzenszwalb, D. P. Huttenlocher // International Journal of Computer Vision, vol. 59, issue 2, 2004. – P. 167-181.
- [2] Wassenberg J. An Efficient Parallel Algorithm for Graph-Based Image Segmentation / J. Wassenberg, W. Middelmann, P. Sanders. – Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns (CAIP'09). – Münster, Germany: September 2-4, 2009.
- [3] Ленько В. С. Застосування методів штучного інтелекту до сегментації графічного образу / В. С. Ленько, Ю. М. Щербина // Вісник Національного університету «Львівська політехніка». Інформаційні системи та мережі. – 2004. – с. 194-203.
- [4] Луп'як Д. Д. Модифікації методу сегментації зображення на основі теорії графів // Фундаментальні та прикладні дослідження у сучасній науці. Збірка наукових праць. / Д. Д. Луп'як, А. В. Олесенко. – Харків, Х.: Технологічний Центр, 2017. – 100 с.
- [5] Achanta R., SLIC superpixels / R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk // Lausanne: TEPFL, 2010. Technical Report
- [6] The USC-SIP1 Image Database [Електронний ресурс]: S.I.P.I. of the University of Southern California. – Режим доступу: <http://sipi.usc.edu/database/>.