Компьютерная Математика

УДК 681.3.06

О.П. ДАВИДОВ

КОМПЛЕКСНА СИСТЕМА ОБРОБКИ ЗОБРАЖЕННЬ НА ОСНОВІ АЛГОРИТМІВ ТРІАНГУЛЯЦІЇ

Системный анализ

Запропоновано модифіковані алгоритми тріангуляції, проаналізовано характеристики запропонованих алгоритмів при аналізі просторової інформації. Розроблено комплексну система обробки зображень у якій використовуються зазначені алгоритми.

Задача поліпшення результатів тріангуляції – актуальна в наш час, тому, що інтелектуальна обробка зображення є нині галуззю, що розвивається, а алгоритмам тріангуляції не вистачає гнучкості та різноманітності для всіх задач по обробці зображень.

На даний момент існує багато алгоритмів тріангуляції: складний, теоретичний але дуже швидкий (лінійна складність) алгоритму, запропонованого Bernard Chazelle у [1]; базовий алгоритм, запропонований ще у 1978 [2], що має складність $O(n \log(n))$; відносно повільний (складність $O(n^2)$) але дуже простий і підтримуючий вкладені поліноми Ear Clipping алгоритм, описаний у [3]; наївний і найпростіший алгоритм має складність $O(n^3)$.

Ці алгоритми описують тріангуляцію поліномів, мають багато обмежень і потребують додаткової інформації. Усі ці обмеження не дають використовувати дані алгоритми на зображеннях, локальні особливості яких часто розподілені хаотично, а додаткова інформація недоступна.

Основна мета даної статті – це аналіз розробленої комплексної системи тріангуляції зображень, яка вирішує зазначені вище проблеми.

Далі представлено переваги і недоліки різноманітних алгоритмів тріангуляції. Зважаючи на переваги і недоліки кожного алгоритму вирішено побудувати навколо них комплексну систему обробки зображень, що дозволить використати той чи інший алгоритм або декілька алгоритмів у залежності від потреб. Аргументи алгоритмів можна змінювати і результати одразу відображаються на екрані. Точки тріангуляції – це локальні

особливості зображення. Аргументи алгоритму пошуку локальних особливостей теж можна змінювати. Інтуїтивний інтерфейс дозволяє додавати і видаляти локальні особливості а також лінії тріангуляції. Такий функціонал дає можливість пришвидшити розбиття зображення на сегменти і дозволяє записати дії користувача для подальшого відтворення на схожих зображеннях алгоритмами, що навчаються. Система також пристосована для порівняння ефективності різних алгоритмів тріангуляції.

Для тестування побудованої системи обрано декілька груп зображень різної тематики, розміру і складності. Перша група – чорно-білі контрастні зображення їх зручно використовувати для первинного налаштування програми, деякі з цих зображень були створені самостійно; друга група – різноманітні кольорові зображення використовуються для суб'єктивної оцінки якості роботи алгоритмів, а також для тестування під великим навантаженням, тому, що такі зображення найчастіше генерують багато локальних особливостей; третя група – кольорові контрастні зображення з різноманітних медичних пристроїв: томографів, УЗД, рентгенівських апаратів використовується для суб'єктивної оцінки зручності інтерфейсу користувача, шляхом перевірки чи зручно виділити ту чи іншу ваду на знімку.

Для порівняння алгоритмів тріангуляції візьмемо чорно-біле зображення першого типу (рис. 1), знайдемо на ньому локальні особливості і застосуємо різноманітні алгоритми тріангуляції за отриманими локальними особливостями.

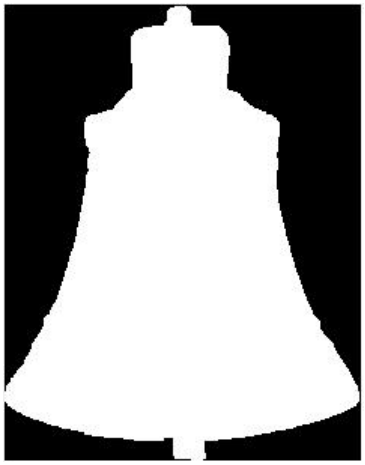

РИС 1. Тестове зображення

Convex Hull алгоритм – це перший розглянутий алгоритм, запропонований у [4]. Цей алгоритм призначений для виділення основного об'єкта зображення шляхом об'єднання деяких крайніх точок зображення безперервною лінією таким чином, що усі точки лежать всередині окресленої області. До плюсів такого алгоритму можна віднести те, що він швидкий, результат тріангуляції можна легко використати у подальших алгоритмах тріангуляції. До мінусів можна

віднести те, що цей алгоритм не дає задовільного результату при обробці зображення з декількома об’єктами, тому, що він окреслює обидва об’єкти в одну область. Ефективність роботи алгоритму також зменшується, якщо зображення має не опуклу форму. На рис. 2 показана робота алгоритму Convex Hull за локальними особливостями, що були знайдені на зображенні (рис. 1). На рисунку можна побачити, що локальна особливість V1 не потрапила до Convex Hull, тому, що вона лежить на не опуклій дузі зображення. Водночас локальна особливість V2 потрапила до Convex Hull, тому що вона лежить на опуклій дузі зображення. Ця особливість алгоритму призводить до того що деякі не опуклі ділянки зображення і внутрішня структура цілком не потрапляють до результатів тріангуляції, що неприпустимо для задач сегментації і класифікації зображення. Водночас всередині тріангуляції опинилися області, які не належать оригінальному об’єкту (чорні області усередині червоної лінії). Треба зазначити, що інформація про кількість об’єктів на зображенні не завжди є наявною і у випадку декількох об’єктів на одному зображенні Convex Hull об’єднає їх у один об’єкт, що є також незадовільним. З іншого боку, результатом роботи цього алгоритму – простий, опуклий поліном. Цей факт дозволяє використовувати Convex Hull для первинного аналізу зображення в цілому, наприклад, для визначення площі області, на якій присутній хоча б один об’єкт зображення.

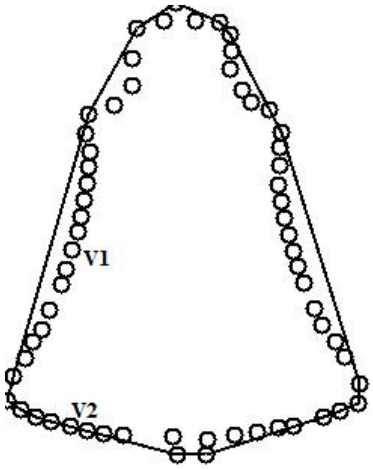

РИС. 2. Робота алгоритму Convex Hull

Delaunay тріангуляція. Ця тріангуляція максимізує мінімальний кут трикутників, на які розбивається зображення. Такий ефект досягається шляхом введення додаткових точок та проведення ліній тріангуляції таким чином, щоб коло, описане навколо довільного трикутника не мало всередині точок, окрім трьох точок довільного трикутника, навколо якого воно описане. Такий підхід запропонував ще Boris Delaunay в 1934 [5] і нині цей підхід розвивається як в сторону нових областей застосування, наприклад, існує модифікація алгоритму

для розмірності n [6], так і в сторону використання новітніх підходів у рамках цього алгоритму, наприклад, є модифікація цього алгоритму яка використовує sweep hull техніку [7]. Ця техніка виходить з sweep line техніки, що вперше представлена в [8]. На рис. 3 показано приклад Delaunay тріангуляції за тим же зображенням і над тими ж локальними особливостями, що і на рис. 1. На рис. 3 можна побачити, що зовнішній контур тріангуляції співпадає з Convex Hull (див. рис. 1). Також можна побачити додаткові точки, які не є локальними особливостями. Вони були додані з метою збільшення мінімального кута трикутника до 10^{o}. До мінусів цього алгоритму можна віднести занадто велику кількість зайвих ліній всередині зображення. Це у подальшому може завадити коректній сегментації і класифікації. Треба пам’ятати, що мета Delaunay тріангуляції – розбиття полінома на велику кількість маленьких і якісних (в плані мінімальних і максимальних кутів) трикутників і такий результат не завжди потрібен при обробці зображення. Цей алгоритм має і деякі мінуси властиві Convex Hull: він, наприклад, також обробляє області, що не належать об’єкту на зображенні. До плюсів алгоритму можна віднести гнучкість алгоритму в плані вибору мінімального та максимального кута трикутника.

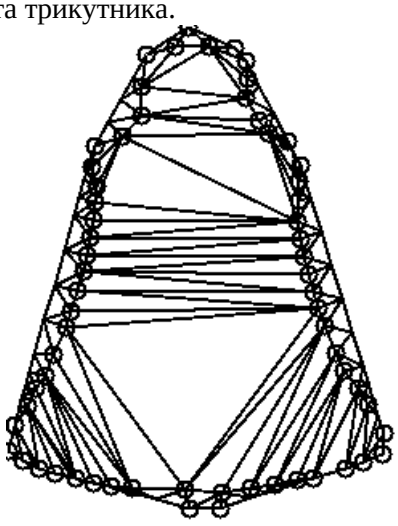

РИС. 3. Робота алгоритму Delaunay

Ear Cutting тріангуляція, запропонована у [3]. Цей алгоритм призначений для тріангуляції нормальних поліномів як опуклих, так і не опуклих з внутрішніми «отворами» у виді інших поліномів. Для такої тріангуляції необхідно щоб вершини полінома зберігалися у правильному порядку: всі сусідні вершини повинні мати відповідні лінії у поліномі. Також ми повинні знати до чого належить та чи інша вершина: до поліному чи до «отвору». У загальному випадку така інформація недоступна для зображення. Можна, звичайно, використати Convex Hull алгоритм, розташувати вершини у правильному порядку і над отриманим масивом точок запустити Ear Cutting тріангуляцію, але через те, що

Convex Hull не включає у себе всі локальні особливості зображення і завжди генерує опуклий поліном, результат буде незадовільним (рис. 4). Основою тріангуляції є такий алгоритм: обираємо довільну точку і дві наступні точки, якщо з першої до третьої точки можна провести лінію таким чином, що вона повністю належить поліному, то отриманий трикутник називається «вухом» і відтинається від основного поліному. Метод повторюється на модифікованому поліномі поки усі «вуха» не будуть знайдені. На рис. 4 можна побачити, що хоча тріангуляція і була коректною, вона ніяк не відображає структуру об’єкта. Такий алгоритм буде доцільно використовувати у разі якщо ми будемо мати інформацію про внутрішню структуру об’єкта і зможемо визначити які локальні особливості належать до отворів а які для поліномів. Результат роботи алгоритму також можна покращити шляхом застосування алгоритму окремо для різних частин зображення, представлених у формі різних поліномів.

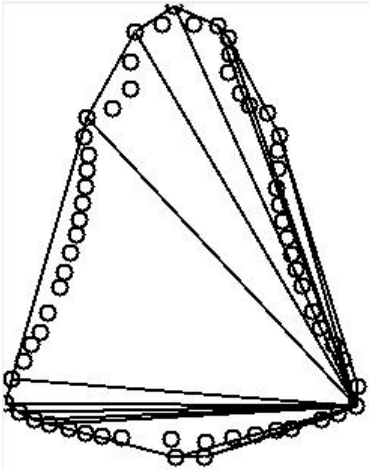

РИС. 4. Робота Ear Cutting тріангуляції

Модифікований Convex Hull алгоритм бере за основу звичайний Convex Hull, але покращує результати його роботи шляхом додаткової обробки точок, які не потрапили до фінальної тріангуляції. Це покращення відбувається наступним чином:

1) до зображення застосовується Convex Hull алгоритм;

2) береться точка, що не належить поліному, отриманому в п. 1);

3) знаходиться лінія, яка найближча до даної точки і належить поліному;

4) проводяться дві лінії – з точки, отриманої в п. 2) до початку та кінця лінії, отриманої у п. 3);

5) перевіряється, чи перетинаються нові лінії з вже існуючими. Якщо хоча б одна лінія перетинається з вже проведеними, то обидві лінії не будуть додані до фінальної тріангуляції. Точка, що обробляється видаляється з масиву точок для обробки і більше не буде оброблена;

6) перевіряється площа отриманої фігури. Площа нової фігури завжди буде менша за площу початкової фігури, тому, що усі точки, що не належать поліному знаходяться всередині полінома, і відбувається відтинання трикутника, площа якого не може бути меншою нуля. Якщо відношення між площами початкової (тут мається на увазі площа фігури, отриманої у п. 1), вона залишається сталою впродовж всього алгоритму) і отриманої фігури більша за певне значення (наприклад, більша за 20 %), то дві лінії не будуть додані до фінальної тріангуляції. Точка, що обробляється видаляється, так само як у п. 6);

7) якщо лінії пройшли обидві перевірки, вони потрапляють до фінальної тріангуляції, а лінія, отримана у п. 3) видаляється. Усі три точки (точка, що обробляється, кінець та початок лінії, отриманої у п. 3)) більше ніколи не будуть оброблені, тому що через них проходить лінія поліному;

8) алгоритм повторюється для отриманого поліному, починаючи з п. 2).

Алгоритм завжди завершується, тому, що не зважаючи на перевірки, точка, що обробляється ніколи не буде оброблена знову.

Якщо межа відношення між площами початковою і отриманою фігурою буде дуже малою (наприклад, 0 %), то результат роботи алгоритму буде дорівнювати результату роботи звичайного Convex Hull; якщо ж межа буде дуже великою (100 %) то алгоритм з’єднає усі точки, по яким проводиться тріангуляція однією безперервною лінією.

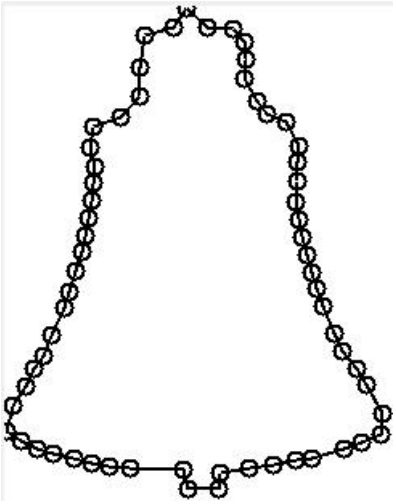

РИС. 5. Робота модифікованого Convex Hull алгоритму

До плюсів такого методу можна віднести хороший результат при обробці опуклих зображень, частини яких є не опуклими. У такому разі можна запустити алгоритм з дуже великою межею відношення площ. Якщо ж зображення є складним, то результат роботи Convex Hull можна покращити незначно, задавши межу відношення площ малою. Плюсом такого методу є також можливість об’єктивно обчислити його ефективність шляхом обчислення площі, отриманого

поліному і порівняння цієї площі з деяким очікуваним результатом. Для чорно-білих зображень очікувана площа може бути обчислена шляхом підрахунку чорних (або білих) пікселів зображення.

До мінусів можна віднести повільну роботу алгоритму, через додаткові перевірки на кожній ітерації алгоритму. Також, проблемою є підбір коректної межі відношення площ.

Розберемо покроковий приклад роботи алгоритму (рис. 6). У першій частині зображення показано чотири точки за якими треба провести тріангуляцію зображення. Далі показано результат роботи звичайного Convex Hull алгоритму; чорна точка – це точка, обрана для подальшої обробки; лінія L1 – це лінія, яка є найближча до обраної точки. Штрихами показані лінії, що є кандидатами на додавання до фінальної тріангуляції. Заштрихована область – це область, що буде віднята від оригінального поліному, саме це зменшення буде перевірено згідно з п. 6) алгоритму. У двох останніх частинах показано результат однієї ітерації алгоритму і наступна точка, що буде оброблена.

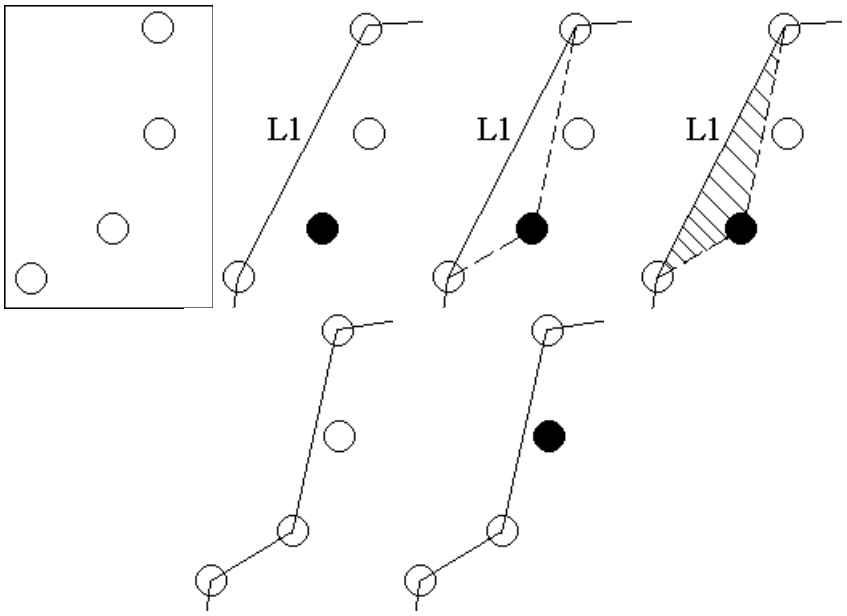

РИС. 6. Покроковий розбір роботи модифікованого Convex Hull алгоритму

Ефективність запропонованого алгоритму можна оцінити поділивши отриману площу фігури на очікувану площу фігури, що обробляється. Цю частку будемо називати помилкою алгоритму по площі, а різницю помилок двох алгоритмів – перевагою одного алгоритму над іншим по площі. Якщо на зображенні є тільки одна фігура, і вона біла на чорному фоні, то очікувану площу фігури можна підрахувати просто підрахувавши кількість білих пікселів. Для підрахунку отриманої площі фігур використовується формула Гаусса:

$$S_{CH}, S_{CHU} = \frac{1}{2}\left|\sum_{i=1}^{n-1} x_i y_{i+1} - x_{i+1} y_i\right|.$$

Тут i – порядковий номер вершини многокутника, n – кількість вершин многокутника, x_i, y_i – координати i-ї вершини многокутника.

Для підрахунку переваги по площі модифікованого алгоритму над немодифікованим, використовується наступна формула:

$$\left|\frac{S_{CH}}{S_E} - 1\right| - \left|\frac{S_{CHU}}{S_E} - 1\right|.$$

Тут S_{CH}, S_{CHU} – площі фігур, отриманих за допомогою немодифікованого та модифікованого convex hull алгоритму відповідно, S_E – очікувана площа фігури.

Для експериментів обрано 50 контрастних чорно-білих зображень 10-и типів: яблука, кажани, дзвони, коми, обличчя, риби, фонтани, черепахи, часи та схематичні зображення. Результати експерименту наведені у таблиці.

ТАБЛИЦЯ. Результати обчислення переваги по площі модифікованого алгоритму Convex Hull над немодифікованим, і середні помилки по площі обох алгоритмів

Тип зображення	Помилка алгоритму Convex Hull, у %	Помилка модифікованого CH алгоритму, у %	Перевага модифікованого алгоритму, у п. п.
Яблука	6,8	0,32	6,48
Кажани	55,4	0,27	55,13
Дзвони	11,3	0,77	10,53
Коми	24,7	0,15	24,55
Обличчя	3,3	0,38	2,92
Риби	23,1	1,45	21,65
Фонтани	25,9	1,03	24,87
Черепахи	26,0	1,02	24,98
Часи	23,8	0,49	23,31
Схематичні зображення	54,7	0,86	53,84

Помилка по площі модифікованого Convex Hull алгоритму у середньому поміж усіх типів зображення виявилась 0,674 %. Середня перевага по площі виявилась 24,83 процентних пункти.

Також, ефективність роботи алгоритму можна оцінити, взявши до уваги кути зображення. За середній очікуваний внутрішній кут зображення можна взяти кут 180°. Близькість середнього внутрішнього кута зображення до цього значення означає більш «гладке» а значить якісне розбиття. За аналогією з помилкою та перевагою по площі можна визначити помилку по куту як відношення середнього внутрішнього кута зображення до 180°, а перевагу по куту одного алгоритму над іншим, як різницю помилок двох алгоритмів.

Для підрахунку середнього кута усіх ліній многокутника використовується така формула:

$$A_{CH}, A_{CHU} = \sum_{i=1}^{n} \frac{180}{\pi} \left(\pi - \text{atan}(dx_i dy_{i+1} - dx_{i+1} dy_i, dx_i dx_{i+1} + dy_i dy_{i+1}) \right),$$

$$dx_i = B_i x - E_i x,$$

$$dy_i = B_i y - E_i y.$$

Тут i – порядковий номер сторони многокутника, n – кількість сторін многокутника. atan(x, y) – функція, що рахує арктангенс кута між лінією, що проведена на початку координат, до точки (x, y) та віссю x. $B_i x$, $B_i y$ – координати x та y початку лінії з порядковим номером i. $E_i x$, $E_i y$ – координати x та y кінця лінії з порядковим номером i. На останній ітерації замість $i+1$ береться 1.

Середня перевага за кутами виявилась 7,6 процентних пункти. Для тестування обрані ті ж самі вхідні дані.

Алгоритм тріангуляції, на основі близькості до інших точок тріангуляції. Запропонований алгоритм – це модифікація над інтуїтивним алгоритмом, який описано в [9]. Цей алгоритм полягає у тому, щоб проводити лінії з кожної точки до найближчої до неї точки, при цьому враховуючи відстань між лінією й іншими точками тріангуляції. Для пришвидшення роботи алгоритму, замість того щоб перевіряти кожну точку лінії, що є кандидатом на проведення, на близькість до інших точок, пропонується передобробка зображення, у ході якої кожному пікселю зображення присвоюється деяка «оцінка близькості», яка відображає кількість точок тріангуляції, що знаходяться на деякій заданій відстані від пікселя. Ця передобробка відбувається по інтуїтивному алгоритму – усім пікселям присвоюється оцінка близькості нуль, потім навколо кожної точки тріангуляції проводиться умовне коло, радіус якого дорівнює необхідній заданій відстані, і оцінка близькості всіх пікселей, що потрапили у це коло збільшується на один. Функція знаходження локальних особливостей зображення як один із аргументів має мінімальну відстань між локальними особливостями, враховуючи це, якщо задати цей аргумент рівним необхідній відстані, то умовні кола навколо локальних особливостей не будуть перетинатися, і оцінка близькості прийматиме значення нуль або один.

Після такої передобробки лінія, що є кандидатом на включення у фінальну тріангуляцію, перевіряється на близькість до інших точок тріангуляції. Для цього вибираються усі точки, що належать даній лінії і рахується сума їх оцінок близькості. Оцінка близькості приймає значення тільки один або нуль, тому сумарна оцінка близькості не може перевищувати довжини лінії, враховуючи це, вирішено не включати до фінальної тріангуляції лінії, для яких відношення довжини до сумарної оцінки близькості менше за деякий відсоток. Наприклад, якщо цей параметр дорівнює 50 %, то до фінальної тріангуляції потраплять тільки ті лінії, для яких половина, або більше їх точок знаходяться поруч з однією локальною особливістю.

До плюсів такого алгоритму можна віднести те, що він зменшує кількість ліній тріангуляції, що не знаходяться близько до локальних особливостей, що покращує результат тріангуляції, але він має суттєву ваду: він працює набагато довше ніж інтуїтивний алгоритм тріангуляції, через передобробку, обчислення масиву точок, що належать деякій лінії і підрахунку кількості точок, які належать одночасно і лінії і масиву точок, що знаходяться поблизу локальних особливостей. Також алгоритм у деяких ситуаціях може проводити лінії тріангуляції, частини яких лежать далеко від усіх локальних особливостей (рис. 8).

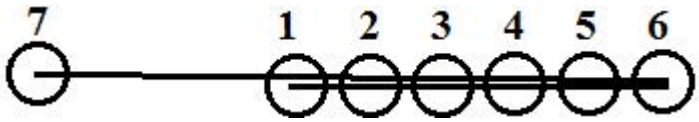

РИС. 8. Хибна робота покращеного алгоритму тріангуляції

Тут круги – зони в яких «оцінка близькості» дорівнює одиниці, а цифри – номера під якими були оброблені точки. Відсоток точок які повинні бути близько до інших точок більший за це значення для відрізка 1 – 7, але менший за це значення для відрізка 6 – 7, наприклад 30 %.

Для покращення роботи алгоритму пропонується:

1) прискорити передобробку зображення, шляхом використання відстані Чебишова замість евклідової метрики;

2) обробляти тільки певний відсоток локальних особливостей, найближчих до даної локальної особливості. Цей підхід частково вирішує проблему, показану на рис. 8. Після оптимізації лінія 6 – 7 не буде проведена через те, що точки 6 та 7 знаходяться далеко одна від одної.

Приклади роботи немодифікованого та модифікованого алгоритмів тріангуляції над текстовим зображенням (рис. 9) можна побачити на рис. 10. На ньому видно, що модифікований алгоритм позбавив тріангуляцію від зайвих ліній посеред зображення.

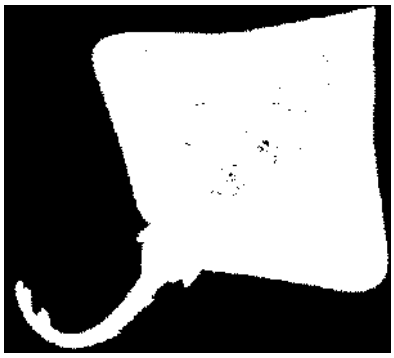

РИС. 9. Тестове зображення для інтуїтивного та модифікованого алгоритмів тріангуляції

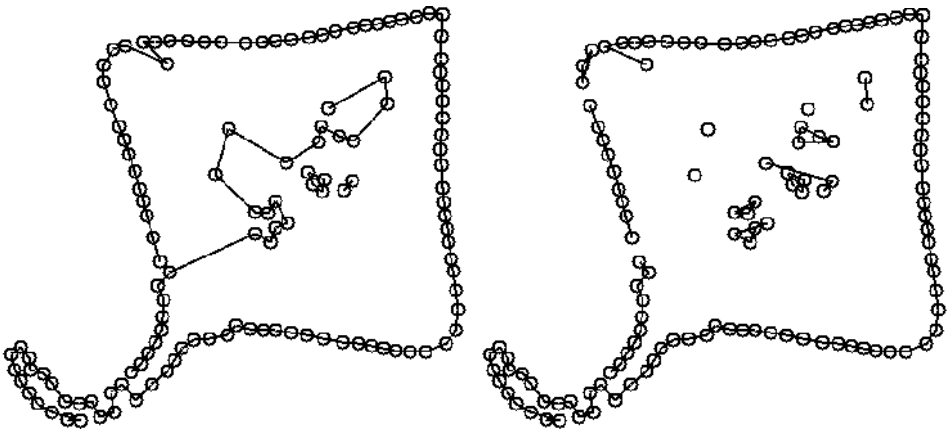

РИС. 10. Приклад роботи інтуїтивного та модифікованого алгоритмів тріангуляції

Висновки. Розроблено комплексну систему обробки зображень, ядро якої – функція пошуку локальних особливостей і декілька алгоритмів тріангуляції, які можна визивати за потребою з різними аргументами. Система підтримує додавання та видалення знайдених локальних особливостей та ліній тріангуляції. Такі можливості дозволяють швидко розбити зображення на необхідні сегменти, а також створює бази для розробки систем, що навчаються аналізувати просторову інформацію.

А.П. Давыдов

КОМПЛЕКСНАЯ СИСТЕМА ОБРАБОТКИ ИЗОБРАЖЕНИЙ НА ОСНОВЕ АЛГОРИТМОВ ТРИАНГУЛЯЦИИ

Проанализированы характеристики существующих и модифицированных алгоритмов триангуляции. Показано уменьшение ошибки модифицированного алгоритма Convex Hull в сравнении с обычным аналогом. Построена комплексная система, которая позволяет быстро разбить изображение на необходимые сегменты, используя указанные алгоритмы триангуляции с необходимыми аргументами, а также добавлять и удалять локальные особенности и линии триангуляции. Подготовлены основы для разработки систем, обучающихся анализировать пространственную информацию на похожих изображениях.

O.P. Davydov

COMPLEX SYSTEM OF IMAGE PROCESSING BASED ON TRIANGULATION ALGORITHMS

The advantages and disadvantages of existing and modified triangulation algorithms are analyzed. Error reduction of the modified Convex Hull algorithm is shown compared with its common analog. A comprehensive system is built which allows the user to split the image into the needed segments using the specified triangulation algorithms with required arguments, as well as to add and remove local features and triangulation lines. The background is prepared for developing learning systems, which analyze spatial information on similar images.

Список літератури

1. Chazelle B. Triangulating a Simple Polygon in Linear Time. Department of Computer Science, Princeton University, Princeton, NJ 08544, USA, 1991. 40 p.
2. Garey M.R., Johnson D.S., Preparata F.P., Tarjan R.E. Triangulating a Simple Polygon, Info. Proc. Letters. June 1978. Vol. 7(4). P. 175 – 179.
3. Eberly D. Triangulation by Ear Clipping Geometric Tools, Redmond WA 98052, 2015. 13 p.
4. Timothy M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete and Computational Geometry*. 1996. Vol. 16. P. 361 – 368.
5. Delaunay B. "Sur la sphère vide". Bulletin de l'Académie des Sciences de l'URSS, Classe des sciences mathématiques et naturelles. 1934. 6. P. 793 – 800.
6. De Loera, Jesús A. Rambau, Jörg Triangulations, Structures for Algorithms and Applications. *Algorithms and Computation in Mathematics*. 25. Springer. Santos, Francisco. 2010.
7. Sinclair D.A. S-hull: a fast radial sweep-hull routine for Delaunay triangulation. Cambridge, UK. 2010.
8. Fortune S. Stable maintenance of point-set triangulation in two-dimensions. 30th Annual Symposium on the Foundations of Computer Science. IEEE, New York., 1989.
9. Давидов О.П. Метод сегментації зображень на основі нечітких чисел. *Компьютерная математика.* 2016. Вип. 2. С. 66 – 74.

Одержано 16.12.2018

Про автора:

Давидов Олександр Павлович,
молодший науковий співробітник
Інституту кібернетики імені В.М. Глушкова НАН України.
E-mail: duff1994@gmail.com