Аналіз та реалізація класичного алгоритму маркування зв'язних компонент на бінарному зображенні

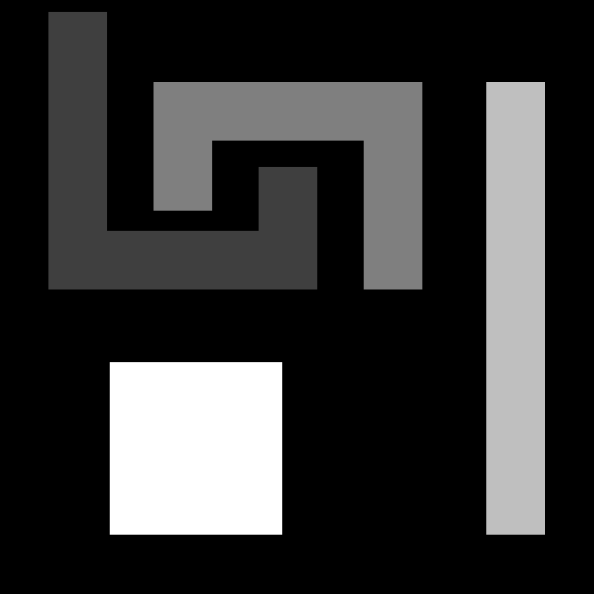
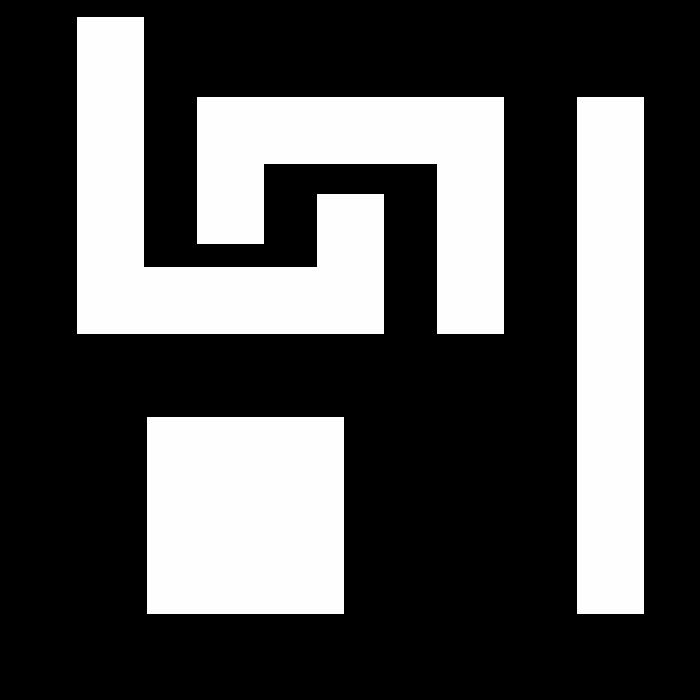
В статті розглянуто двопрохідний порядковий алгоритм маркування звязних компонент, розглянуто спосіб реалізаціі алгоритму та використаня динамічного масиву для збереження класів еквівалентності. Алгоритм маркування звязних компонент використовується для виділення звязної компоненти, що використовується для подальшого морфологічного аналізу компоненти.

В статье рассмотрено двухпроходной порядковый алгоритм маркировки связных компонент, рассмотрено способ реализации алгоритма и использование динамичного массива для сохранения классов эквивалентности. Алгоритм маркировки связных компонент используется для выделения связной компоненты, что используется для дальшейшего морфологического анализа компоненты.

In the article considered the two-pass consistent connected components labeling algorithm, considered a way to implement the algorithm and the use of a dynamic array to store the equivalence classes. Connected components labeling algorithm is used to select a connected component, which is used for further morphological component analysis.

Вступ

Двопрохідний порядковий алгоритм маркування зв'язних компонент або класичниий алгоритм маркування зв'язних компонент на бінарному зображенні [2] виявляє зв'язні компоненти(області) переднього плану та надає виявленій компоненті чисельне значення або маркер. В результаті роботи алгоритму, з початкового зображення, в якому всі піксели компонент мають значення 1, формується маркіроване зображення, в якому пісксели компоненти мають значення маркера компоненти. На рис. 1а наведено початкове бінарне зображення, в якому всі чотири компоненти мають однакові значення пікселів. На рис. 2б зображено результат алгоритму маркування, кожна компонента отримала свій маркер, всі піксели компоненти мають значення маркера , внаслідок чого комопненти зображені різними значеннями інтенсивності.



а. Початкове зображення б. Маркіроване зображення.

Рис. 1. Приклад роботи алгоритму маркування звязних областей.

Маркування зв'язних компонент необхідне для подальшого морфологічного аналізу виявлених компонент. Наприклад визначаються ознаки форми сформованої звя'зної компоненти[1].

Алгоритм двопрохідний адже пребирає всі піксели зображення два рази. В порівнянні з рекурсивним, в класичному алгоритмі виконується менше операцій та не використовується рекурсивний аналіз всіх сусідніх пікселів. Слово порядковий означає, що піксели вектора рядка аналізуються поступово в діапазоні [0, C – 1], рядки матриці зображення аналізуються в діапазоні [0, R -1], де C - це кількість колонок, R - це кількість рядків матриці зображення. На відміну від рекурсивного алгоритму, який потребує завантаження всього зображення в пам'ять для обробки, класичний алгоритм потребує завантаження мінімум 2 рядків, що зменщує необхідний розмір памяті та дозволяє працювати з великими зображеннями.

Опис алгоритму

Класичний алгоритм виконується за два проходи. На першому проході формуються класи еквівалентності та надаються тимчасові маркери зв'язним компонентам або їх частинам. На другому проході тимчасові маркери заміняються маркермами відповідно до кореня структури об'єднання-пошуку класа еквівалентності.

Клас еквівалентності являє собою множину тимчасових маркерів окремої зв'язної компонети. Клас еквівалентності в класичному алгоритмі маркування представлено в структурі об'єднання-пошуку, яка зберігає тимчасові мітки в деревоподібній структурі, з'єднує дерева між собою та дозволяє швидко знаходити корінь дерева [1]. На рис. 2 наведено приклад структури об'єднання-пошуку у вигляді таблиці та графів дерев.

A description...

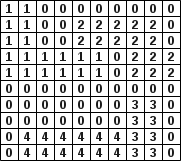
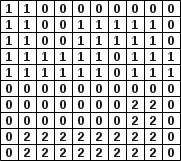
A description...

Рис. 2. Структура об'єднання-пошуку у вигляді таблиці та графу.

Таблиця на рис. 2 в першому рядку містить значення маркерів від 1 до кількості маркерів, значення 0 не враховуєтсья адже відповідає значенню фона зображення. Другий рядок містить значення маркера-батька для маркера з першого рядка, значення 0 означає, що маркер являється коренем дерева. На рис. 2 зображено два графи дерев, що відповідають двум структурам об'єднання-пошуку або класам еквівалентності. Два дерева об'єднують дві множини маркерів - {6, 5, 3, 2, 1} та {8, 7, 4}. Корені дерев - маркери 6 та 8, тому вони мають значення 0 в другому рядку таблиці. Маркер 1 має маркер-батька 2, тому в дгугому рядку має значеня 2, маркери 2, 3, 5 мають маркера-батька 6, тому значення в другому рядку в усіх 6, маркери 4, 7 мають маркера-батька 8, тому в другому рядку міститься 8. Оскільки в першому рядку значення являють нумерацію, то дану тструктуру можливо представити у вигляді одномірного масиву, в якому індекси - це дані першого рядка, а значення масиву - це дані другого рядка.

При необхідності об'єднати два маркери в структуру, або об'єднати дві структури, ящо їхні корені не співпадають, вbконується операція об'єднання. Для виконання операції об'єднання двох дерев досить в таблиці в другий рядок кореня дерева, що додається замінити 0 на значення кореня дерева, в яке виконується додавання. Якщо необхідно знайти батька для маркера, щоб визначити до якого класу еквівалентності він належить, виконується операція пошуку. Для пошуку корення в даній структурі виконується перебирання значень батьків маркерів до того моменту доки не буде знайдено 0.

На першому проході алгоритм надає тимчасовий маркер ще не поміченій області та намагається росповсюджити маркер правим та нижнім сусідам компоненти. Якщо виникає ситуація? що два не однакових маркера намагаються розповсюдитись на один і той же піксел то обирається маркер з меншим значенням. Кожен маркер заноситься в структуру об'єднання-пошуку та при наявності двох різних маркерів в одній компоненті маркери об'єднуються за допомогою описаної вище операції об'єднання. Піля першого проходу класи еквівалентності сформовані у вигляді структур об'єдання-пошуку та мають унікальний маркер або ідентификатор, що відповідає кореню дерева структури. На другому проході тимчасові маркери замінюються ідентифікаторами класів еквівалентності, застосовуючи операцію пошуку кореня дерева, яка описана вище.



а. Результат першого проходу б. Результат другого проходу

A description...

в. Структура об'єдання-пошуку після першого проходу

Рис. 3. Результати роботи двух проходів та структура об'єднання-пошуку.

На рис. 3 показано результат роботи першого та другого проходів алгоритму та наведено струкруру об'єднаня-пошуку, яка містить два класи еквівалентності – {2, 1} та {4, 3}.

Реалізаця алгоритму

Для з,ереження класів еквівалентності або структур об'єднання-пошуку використано динамічний масив, що дозволяє створювати масив відповідно до кількості тимчасових маркерів.

Перший прохід реалізовано двома послідовними циклами, в яких оброблюються піксели зі значенням 1 вхідного зображення. Відносно поточного піксела перевіряться наявність сусідів зі значенням 1 зліва та зверху, з задачею ініціювати маркер MC. Якщо є сусід зліва з маркером ML то МC := ML, якщо сусід зверху також існує і його маркер MT менше МC, то MC:=MT. Якщо сусідів не виявлено то в MC записується значення наступного маркера по порядку. Маркер MC записується в пксел результуючої матриці.

Якщо маркер МС не дорівнює значенню зліва або зверху то виконується об'єднання маркерів в струртурі об'єднання-пошуку.

На другому проході також виконується прохід двома циклами, в яких оброблюються піксели зі значеням більше 0. Піксел зберігає значення тимчасового маркера, по якому за допомогою операції пошуку знаходиться корінь структури об'єднання-пошуку. Для уникнення збою в порядку нумерації маркерів створюється окремий масив, в якому значенню кореня структури присвоюється порядковий номер. Порядковий номер, що відповідає кореню структури записується в пікселі зв'язної компоненти та є її остаточним маркером.

Порівняння класичного та рекурсивного алгоритмів.

Порівняння класичного та рекурсивного алгоритмів проводиться на основі намальованих вручну зображеннях в форматі PGM. В програму зчитується матриця пікселів зображення та проводиться бінаризація значень з порогом в половину максимальної інтенсивності. Для кожного зображення проводиться 10 експериментів, сумується час роботи алгоритму та знаходиться середнє арифметичне. При зміні розміру малюється нове зображення, тому зображення відрізняються складністю. В таб. 1 прведено результати досліджень. Значення StackOverflow означає, що при виконанні рекурсивного алгоритму переповнено стек задач. По результатам видно, що класичний алгоритм працює швидше рекурсивного.

Таблиця 1. Результати роботи класичного та рекурсивного алгоритмів.A description...

Висновок

В статті детально розглянуто двопрохідний порядковий алгоритм маркування звєязних компонент, приведено опис реалізації алгориму з використанням динамічних масивів для зберігання структрур об'єдняння-пошуку. Порівняння часу роботи класичного та рекурсивного алгоритму показало що класичний алгоритм працює в два рази швидше рекурсивного, також при розмірі зображення 600 і більше пікселів рекурсивний алгоритм призводить до переповення стеку задач. Враховуючи також те, що класичний алгоритм потребує для роботи мінімум два рядки, класичний алгоритм ефективно працює з великими зображенями.

Список використаних джерел.

1. Л. Шапиро, Дж. Стокман, Компьютерноезрение, М.: Бином. Лабораториязнаний, 2006.
2. A. Rosenfeld, P. Pfaltz, “Sequential Operations in Digital Picture Processing”, Journal of the Association for Computing Machinery, Vol. 12., 1966.