

ВІДЛЕННЯ КОНТУРІВ ОБ'ЄКТІВ ПРИ ВИКОНАННІ ПОШУКУ ЗОБРАЖЕНЬ ЗА ЇХ ВМІСТОМ

У статті оцінюється місце виділення контурів у процесі пошуку зображень за їх вмістом, аналізуються методи виділення контурів. Описується модифікований алгоритм виділення контурів та його паралельна реалізація на платформі CUDA. Оцінюються характеристики роботи послідовної та паралельної версій модифікованого алгоритму

Вступ

Сучасні електронні інформаційні ресурси значною мірою подані у вигляді колекцій зображень та відео. Оскільки доступ до зображень та відеоматеріалів виконується переважно після попереднього пошуку, ефективне виконання пошуку таких даних є актуальною задачею. Під час пошуку має аналізуватися зміст зображення, наприклад, колір представлених на ньому об'єктів, їх форма, текстура, композиція сцени. Як узагальнення процесу вибірки зображень з колекції за будь-якими характеристиками об'єктів використовується пошук зображень за змістом (англ. Content-based image retrieval) - розділ комп'ютерного зору, в якому вирішується проблема пошуку зображень, що мають необхідний вміст, у великому наборі цифрових зображень.

Інтерес до пошуку зображень за змістовними критеріями обумовлений обмеженими можливостями методів, що використовують виключно текстові анотації вмісту зображень. Алгоритми пошуку за змістом можуть використовуватись, наприклад, під час пошуку зображень в мережі Internet, для каталогізації зображень творів мистецтва, організації роботи з архівами фотознімків, при створенні каталогів роздрібною продажу товарів, під час медичної діагностики захворювань, для запобігання злочинам та заворушенням, з метою контролю за поширенням об'єктів інтелектуальної власності, для контролю за вмістом масивів зображень. Існують компанії, що представляють програмні продукти, в яких алгоритми пошуку зображень за змістом застосовуються для фільтрації вмісту веб-сторінок і державного моніторингу мережного трафіку.

Алгоритми, методи і програмні інструменти пошуку зображень беруть початок в областях, пов'язаних з обробкою сигналів, комп'ютерним зором і статистикою. Найбільш популярні методи опису змісту зображень, що використовуються для подальшого порівняння їх між собою - колір, текстура і форма. Всі вони не є специфічними для будь-якого конкретного підкласу систем. Поширений різновид пошуку зображень часто виконується за колірними складовими, переважно шляхом порівняння точкових або гістограмних колірних ознак. Пошук за текстурними ознаками використовує структури даних, які містять не тільки інформацію, що описує текстуру, а й її місце розташування на описуваному зображенні. Саму текстуру зазвичай представляють у вигляді двомірного масиву зміни яскравості. Ще один різновид

пошуку передбачає опис геометричної форми окремих регіонів зображення. Для її визначення до регіону спочатку застосовують сегментацію зображення або виділення контурів об'єктів.

Пошук зображень, схожих на задане, полягає у послідовному виконанні кількох етапів: створення опису зображення, що виступає зразком пошуку, порівняння створеного опису з аналогічними описами вмісту усіх зображень, раніше включених до колекції і виділення зображень зі схожими описами [1]. Виділення контурів у зображенні виконується на етапі створення опису його вмісту, і, таким чином, час виділення контурів впливає на загальний час виконання пошуку. З іншого боку, на час пошуку впливають часові витрати на порівняння описів вмісту зображень. Отже, алгоритм виділення контурів має відрізнятися якомога вищою швидкістю і при цьому створювати компактний опис контуру, придатний для подальшого порівняння.

У статті розглядаються питання побудови модифікованого алгоритму виділення контурів з метою його подальшого застосування при пошуку зображень, та аналізу часових і просторових характеристик такого алгоритму.

Постановка задачі виділення контурів

Зазвичай задача виділення контурів формулюється як побудова зображення кордонів об'єктів та однорідних областей [2]. Якщо ж виділення контурів виконується при пошуку зображень, першочерговим питанням є створення для контуру опису, придатного для подальшого порівняння.

Традиційно контуром називається сукупність пікселів, навколо яких спостерігається зміна функції яскравості. Оскільки при цифровій обробці зображення представляється як функція цілочисельних аргументів, то контури утворюються лініями завширшки, як мінімум, в один піксель [2].

Надалі, з метою скорочення обсягу пам'яті для зберігання опису зображення, розглядатимемо контур як сукупність суміжних прямолінійних відрізків.

Нехай маємо вхідне зображення I , що представляє собою прямокутну матрицю $M \times N$, елементами якої є кольори відповідних пікселів зображення, M і N – ширина і висота зображення відповідно. Зображення I є сукупністю K фрагментів (вікон):

$$W_i, \bigcup_{i=1}^K W_i = I.$$

Далі, нехай існує F – алгоритм, за допомогою якого виділяються контури об'єктів, що містяться в зображенні I . Алгоритм працює з певним вікном W_i . Позначимо результат обробки зображення I алгоритмом F як A :

$$A = F(I).$$

A задає собою множину, кожен елемент якої описує прямолінійний відрізок, що є фрагментом контуру:

$$A = \{a_i\}, \quad a_i = (\alpha_i, d_i),$$

де α_i – кутовий коефіцієнт прямої, d_i – відстань до середини вікна W_i .

Таким чином, задача виділення контурів може розглядатися як задача отримання множини A для деякого зображення I шляхом виконання алгоритму F при фіксованому розмірі вікна W_i .

Аналіз алгоритмів, що використовуються для виділення контурів об'єктів у зображеннях

Алгоритми виділення контурів можна умовно розбити на три групи: алгоритми відслідковування, алгоритми сканування та їх комбінації.

Алгоритми відслідковування засновані на тому, що на зображенні відшукується об'єкт (перша точка об'єкта, що зустрілася) і контур об'єкта описується сукупністю векторів. Простежування контурів є послідовним процесом, у якому помилка, допущена на будь-якому кроці цього процесу, робить більш імовірними помилки й на наступних кроках. Тому алгоритми простежування контурів можуть застосовуватися тільки на зображеннях з низьким рівнем шуму [3]. Перевагою даних алгоритмів є простота, до недоліків можна віднести послідовну реалізацію й деяку складність при пошуку та обробці внутрішніх контурів [4]. Прикладами таких алгоритмів є алгоритм “жука” [4], метод активних контурів [5], метод Canny (оптимальний детектор краю) [6].

Алгоритми сканування засновані на перегляді усього зображення й виділенні контурних точок без прив'язки до попередніх точок контуру.

Комбінація методів відслідковування й сканування полягає у тому, що спочатку в результаті скануючого перегляду зображення здійснюється його попередня розмітка.

З точки зору розміру оброблюваного зображення можна розділити алгоритми на два великих класи за ознакою розміру оброблюваної області. Перша група розбиває зображення на окремі частини й обробляє їх окремо. До неї можна віднести локальні алгоритми виявлення перепадів яскравості, алгоритм «Snakes on the Watershed». Друга група передбачає обробку всього зображення. До неї належать алгоритми жука, активних контурів, Canny.

Другу групу так само можна розділити на дві підгрупи за типом обробки зображення – комплексні й послідовні алгоритми. Перші припускають обробку всього зображення відразу (це метод активних контурів, метод Canny), а другі – послідовну (попільсьельну) обробку зображення (алгоритм жука).

З точки зору кількості контурів, що виділяються алгоритмом, також можна вирізнити дві підгрупи алгоритмів. До алгоритмів, які можуть виявляти на зображенні кілька контурів об'єктів, що не перетинаються, належать алгоритми Canny, «Snakes on the Watershed», алгоритм Перова, нейромережевий алгоритм. Тільки один контур можуть виявити алгоритми жука, активних контурів і динамічного програмування.

У результаті аналізу вищеописаних алгоритмів виділення контурів була розроблена їх класифікація [7].

Модифікований алгоритм виділення контурів

При виборі базового алгоритму для модифікації враховувалась можливість озпаралелювання обчислень з використанням GPGPU (General-purpose graphics processing units, графічний процесор загального призначення), зокрема технології NVidia CUDA [8]. З цих міркувань було обрано нейромережевий алгоритм [9].

Для реалізації процедури виділення контуру за допомогою нейронної мережі була прийнята наступна модель. На вхід нейронної мережі подається зображення вікна спостереження розміром $M*N$. Це зображення задовольняє наступним вимогам:

- а) зображення завжди містить елемент контуру, тобто в будь-якому представленому зображенні міститься границя перепаду яскравостей об'єкта й фона;
- б) на зображенні є лише один контур;
- в) на зображенні може бути присутнім адитивний шум Гаусса, що перешкоджає правильній роботі алгоритму.

Щоб дотриматися вимог до вікна зображення, рекомендується виконати знаходження попереднього контуру більш швидким і грубим методом, після чого знайдене наближення контуру розбити на ділянки для уточнення на нейромережі [10].

Для попереднього визначення контурів обрана комбінація з декількох етапів алгоритму Canny. Так, спочатку виконується згладжування шуму у початковому зображенні (згортка, для якої використовується матриця з гауссовим розподілом елементів). Потім застосовується оператор Собеля, що виконує двомірне знаходження градієнта зображення. Використовуються дві матриці 3x3 для горизонтального (G_x) та для вертикального (G_y) градієнтів. Результати двох градієнтів комбінуються за формулою: $|G| = \sqrt{G_x^2 + G_y^2}$. Після цього виконується порогове претворення з метою усунення переривчастості контуру.

На виході алгоритму необхідно отримати основні параметри прямої, що апроксимує границю перепаду яскравості.

Опишемо структуру нейронної мережі, яка була розроблена для реалізації цієї моделі.

Вхідним сигналом нейронної мережі є вектор $X = \{x_1, x_2, \dots, x_n\}$, де $x_i = 0, 1$. Вектор представляє собою квадратний фрагмент бінаризованого зображення, що містить елемент контуру.

Вихідний сигнал: $k, b = [0, 1]$, де k – кут нахилу прямої, що апроксимує даний елемент коду, b – відстань прямої до центру фрагмента. При цьому $k = 0$ відповідає куту $(-\pi/2)$, а $k = 1$ відповідає куту $\pi/2$. Значення $b = 0$ відповідає відстані, рівній 0, а $b = 1$ відповідає стороні фрагмента, поділеній на $\sqrt{2}$.

Нейронна мережа являє собою двошаровий перцептрон, перший шар якого складається з 10 нейронів, а другий – з 2 нейронів. Кожен з нейронів першого шару пов'язаний з кожним елементом другого шару. Кожен елемент вхідної множини подається на кожен з нейронів першого шару.

Функція активації нейронів логістична, описується формулою:

$$f(x) = \frac{1}{1 + \exp(-0.1 \cdot x)}$$

Функція помилки визначає різницю між фактичним і бажаним значеннями вихідного сигналу.

Критерій якості системи визначає, що налаштування нейронної мережі триватиме до тих пір, поки обидві компоненти функції помилки одночасно не стануть менше заданої точності. Навчання (налаштування) нейронної мережі виконується за алгоритмом зворотного поширення помилки.

Крім того, окремою задачею є збір розрізаних ділянок контурів в один безперервний контур. Вирішити цю задачу можна, приміром, знаходячи середню точку між двома крайніми точками апроксимуючої прямої у двох сусідніх вікон [9].

Аналіз результатів виділення контурів за допомогою модифікованого алгоритму

Програмна реалізація алгоритму показала, що найкращі результати досягаються при обробці модифікованим алгоритмом зображень, що містять однотонні об'єкти, які не перетинаються, з чіткою, контрастною межею. Оскільки алгоритм розглядає кожен область зображення незалежно, то можливі помилки детектування в тих місцях, де межа недостатньо контрастна по відношенню до фону. Також виявлено, що алгоритм здатен знаходити контури декількох об'єктів, присутніх на зображенні, у випадку, якщо вони не перетинаються.

Також досліджувалась залежність якості знаходження контуру від контрастності об'єкту та фону на зображенні. Контрастність оцінювалась як середня різниця між кольором пікселя об'єкта та кольором пікселя фону. Наведений на рисунку 1 графік відображає відношення між точками знайденого контуру, які співпадають з реальним контуром, до всіх точок знайденого контуру.

Під час проведення дослідження, для виділення контурів використовувались зображення різних типів та різного розміру. Послідовна і паралельна (за технологією CUDA) реалізації дозволили порівняти витрати пам'яті та часу.

Проведені дослідження виявили, що на невеликих зображеннях більш ефективною є послідовна реалізація алгоритму, але зі збільшенням зображення значно збільшується ефективність розпаралелювання. Для реальних зображень, що мають велику роздільну здатність (а відповідно і розмір), слід віддавати перевагу паралельній реалізації. На рисунку 2 проілюстровано порівняння часових витрат послідовного та паралельного алгоритму для зображень різних розмірів.

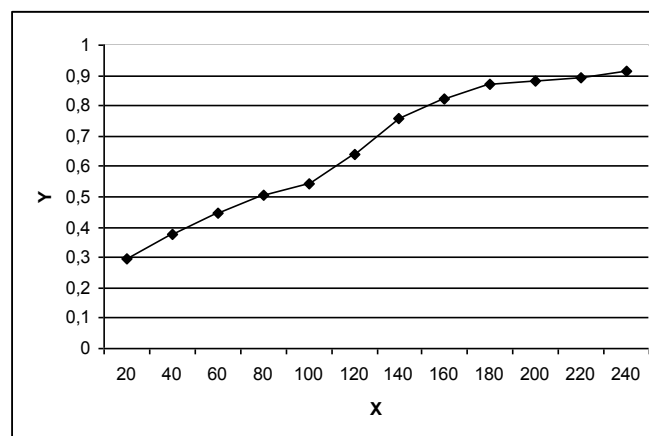


Рисунок 1– Залежність точності розпізнавання Y від контрасту об'єкту та фону зображення X

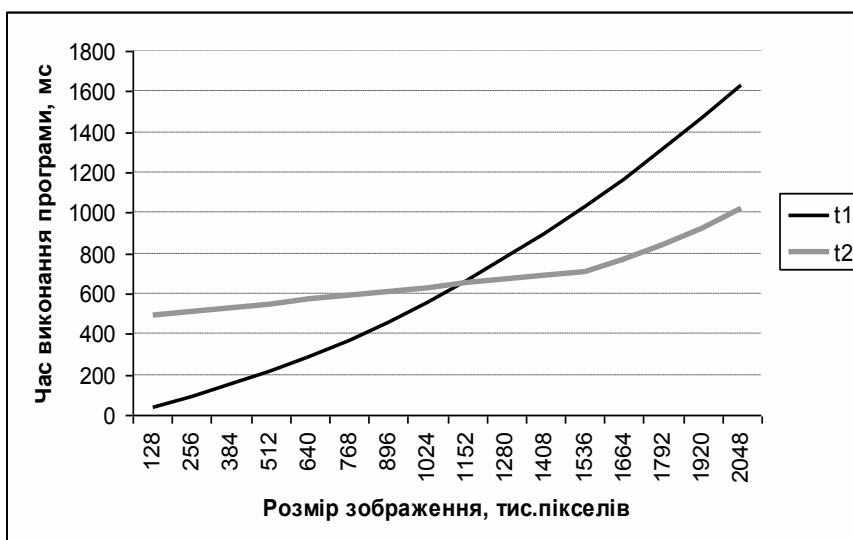


Рисунок 2 – Залежність час виконання програмного модуля від розміру зображення для паралельної (t1) та послідовної (t2) реалізації

Встановлено, що витрати пам'яті алгоритму збільшуються пропорційно зростанню розміру зображення.

Висновки

Аналіз існуючих послідовних алгоритмів виділення контурів на зображеннях показав, що часові характеристики не дають можливості використовувати їх в системах реального часу. Для вирішення цієї проблеми було запропоновано можливий спосіб модифікації алгоритму і оптимізація для розпаралелювання його за технологією CUDA.

Розроблений алгоритм реалізовано у вигляді програмного модулю, який можна долучити до іншої програми або використовувати незалежно. Роботу паралельного алгоритму було порівняно з послідовною версією та виявлено, що зі збільшенням розміру зображення збільшується різниця між часом виконання паралельної та послідовної версій (на великих зображеннях паралельна версія дає значну перевагу).

Також було проведено дослідження роботи алгоритму на різних класах зображень, яке показало, що краще обробляються синтетичні зображення, або підготовані зображення реального світу (наприклад, фотографія заводської деталі на білому фоні), де об'єкт має чіткі контури та має контраст з фоном.

Література

1. Е.А.Башков, Н.С.Костюкова. К оценке эффективности поиска изображений с использованием 2d – цветowych гистограмм. Проблемы управления и информатики, №6, 2006. с.84-89
2. Методы компьютерной обработки изображений / Под ред. В.А.Сойфера. – 2-е изд., испр. – М.: - Физматлит, 2003. – 784 с.
3. Дуда Р. Распознавание образов и анализ сцен / Р.Дуда, П.Харт – Москва: Мир, 1976.

4. Балахонов Е.В. Алгоритмы выделения контуров / Е.В.Балахонов // Научные работы белорусского государственного университета информатики и радиоэлектроники. – Минск, 2006.
5. Блонкин А.В. Преобразование растр-вектор изображений сосудов / А.В.Блонкин // Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта. CAD/CAM/PDM. – Москва, 2006. – Вып.№6.
6. William E. Green. Canny Edge Detection Tutorial / William E. Green // Information resources & Technology of Mechanical Engineering and Mechanics Department, Drexel University. – Philadelphia, 2002. – Vol.8.
7. Чудовська А.К. Порівняльний аналіз алгоритмів виділення контурів // Матеріали XIII Всеукраїнської (VIII Міжнародної) студентської наукової конференції з прикладної математики та інформатики — Львів, 22–23 квітня 2010.
8. Боресков А.В. Основы работы с технологией CUDA / А.В.Боресков, А.А.Харламов – Москва: ДМК-Пресс, 2010. – 232с.
9. Сирота А.А. Статистические алгоритмы обнаружения границ объектов на изображениях / А.А.Сирота, А.И.Соломатин // Вестник ВГУ – Воронеж, 2008. – № 1. – 154с.
10. Яне Б. Цифровая обработка изображений / Б.Яне – М.: Техносфера, 2007 – 584с.