

79% relative to A^* . The advantage of JPS was most pronounced on map environments with large open areas and low obstacle density, where symmetry pruning and jump point identification most effectively reduce the number of explored nodes. In high-density obstacle environments, the performance gain of JPS remained substantial, with A^* also showing improved relative efficiency due to more focused heuristic guidance.

Conclusion. The experimental study confirms that heuristic and symmetry-based approaches substantially outperform classical graph traversal in grid pathfinding tasks. JPS is recommended for applications with high performance requirements and frequent re-planning between arbitrary node pairs. A^* provides an effective compromise when JPS applicability is constrained by environmental conditions. Dijkstra's algorithm remains appropriate for scenarios requiring full shortest-path trees from a fixed source node or in the absence of a valid heuristic function. Future research directions include analysis of suboptimal but faster algorithms, extension to dynamic environments, and adaptation to three-dimensional search spaces.

Keywords: pathfinding algorithms; Dijkstra's algorithm; A^* algorithm; Jump Point Search; heuristic search; two-dimensional grid; route optimization; maze generation; navigation systems; path planning.

Одержано редакцією 07.06.2024 р.
Прийнято до публікації 28.08.2024 р.

УДК 004.932

DOI 10.31651/2076-5886-2024-1-21-33

PACS 07.05.Pj, 42.30.Va

ГАВРЮШЕНКО Анна Миколаївна
учитель математики та інформатики
Черкаської загальноосвітньої школи І-ІІІ
ступенів № 32 Черкаської міської ради
e-mail: 91gavryushenko@gmail.com
ORCID 0009-0000-0188-8505

БОРОЗДИХ Катерина Сергіївна
учениця Черкаської загальноосвітньої
школи І-ІІІ ступенів № 32 Черкаської
міської ради

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ ВИЯВЛЕННЯ КЛЮЧОВИХ ТОЧОК НА ЗОБРАЖЕННЯХ

У статті розглянуто математичне підґрунтя методу SIFT як одного з перших та найбільш теоретично обґрунтованих підходів до розв'язання зазначеної задачі, а також проведено порівняльний аналіз шести сучасних методів: SIFT, SURF, BRISK, ORB, KAZE та AKAZE. Дослідження виконане на чотирьох синтетичних зображеннях із різними геометричними властивостями та на двох реальних зображеннях. За результатами обчислювальних експериментів здійснено порівняння методів за кількістю виявлених ключових точок, якістю їх конфігурації та часом пошуку однієї ключової точки. Встановлено, що жоден із розглянутих методів не є абсолютним переможцем за сукупністю показників, а вибір оптимального методу суттєво залежить від характеристик зображення та вимог конкретної задачі.

Ключові слова: ключові точки, дескриптор зображення, SIFT, SURF, BRISK, ORB, KAZE, AKAZE, комп'ютерний зір, масштабно-інваріантне перетворення.

Вступ

Задача пошуку одного зображення на іншому лежить в основі багатьох сучасних систем комп'ютерного зору. Вона є невід'ємною частиною розпізнавання об'єктів і сцен, реконструкції тривимірних структур за серією знімків, побудови стереоскопічних

зображень та відстеження руху в потоці відео [7, 8]. Особливо актуальним є її застосування у задачах автоматичного аналізу аерофотознімків, отриманих із безпілотних літальних апаратів, у системах розпізнавання номерних знаків транспортних засобів та ідентифікації осіб у місцях скупчення людей.

Вирішення задачі зіставлення зображень потребує виокремлення таких ознак, які залишаються стабільними при різних умовах спостереження – зміні масштабу, повороті, зсуві точки огляду або коливаннях освітлення. Такими ознаками є ключові точки – локальні особливості зображення, що добре локалізовані як у просторовій, так і у частотній областях, та характеризуються низькою ймовірністю порушення за рахунок часткового перекриття або шуму.

Розвиток методів виявлення ключових точок розпочався із робіт Г. Моравеца та К. Харріса [4, 12]. Їхні кутові детектори орієнтувались на пошук ділянок зображення з великими градієнтами в усіх напрямках, що добре відповідало потребам стереозіставлення та відстеження руху на коротких відстанях. Однак ці детектори виявились суттєво чутливими до зміни масштабу зображення, що обмежувало їх практичне застосування.

Масштабна інваріантність була досягнута в рамках методу SIFT, запропонованого Д. Лоувом [7, 8]. Метод забезпечує пошук ключових точок у повному просторі масштабів та формує стійкий дескриптор кожної з них на основі локальних градієнтів. Подальший розвиток на пряму пов'язаний із розробкою значно швидших методів – SURF [1], BRISK [5], ORB [15], KAZE та AKAZE [2] – що зберігають інваріантні властивості, але суттєво скорочують обчислювальні витрати.

Метою статті є проведення порівняльного аналізу шести сучасних методів виявлення ключових точок на зображеннях – SIFT, SURF, BRISK, ORB, KAZE та AKAZE – за двома ключовими характеристиками: якістю конфігурації знайдених точок та часом пошуку однієї ключової точки. Аналіз проводиться на зображеннях різного типу, що дає змогу оцінити ефективність кожного методу в залежності від змісту зображення.

Виклад основного матеріалу

1. Огляд методів виявлення ключових точок

1.1 Метод SIFT

Метод масштабно-інваріантного перетворення особливостей (Scale-Invariant Feature Transform, SIFT) [7, 8] є теоретично найбільш обґрунтованим серед розглянутих підходів. Його робота базується на чотирьох послідовних етапах.

Виявлення масштабно-просторових екстремумів. Ключові точки виявляються як локальні екстремуми у просторі масштабів, що формується за допомогою різниці Гаусових функцій (Difference of Gaussians, DoG):

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) \circ I(x, y) = \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned} \quad (1)$$

де $G(x, y, \sigma)$ – Гаусова функція з параметром масштабу σ , $I(x, y)$ – вхідне зображення, $L(x, y, \sigma)$ – його згортка з Гаусовою функцією. Ця функція є наближенням нормалізованого Лапласіана Гауса $\sigma^2 \nabla^2 G$, що забезпечує теоретично коректну масштабну інваріантність.

Кожна точка вибірки порівнюється з вісьмома сусідами на поточному рівні масштабу та з дев'ятьма сусідами на сусідніх рівнях – усього 26 сусідів. Точка

обирається як кандидат лише в тому разі, якщо вона є максимальною або мінімальною серед усіх сусідів.

Точна локалізація ключових точок. Для кожного кандидата виконується підбір тривимірної квадратичної функції до локальних точок вибірки шляхом розкладу в ряд Тейлора:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}, \quad (2)$$

де $\mathbf{x} = (x, y, \sigma^T)$ – зміщення від точки вибірки. Це дозволяє відкидати нестабільні екстремуми з низьким контрастом та точки, розташовані вздовж країв, де матриця Гессіана \mathbf{H} має велике значення відношення головних кривин $r = \alpha/\beta$ (де $\alpha \geq \beta$ – власні значення \mathbf{H}), перевіряючи умову:

$$\frac{Tr(\mathbf{H})^2}{|\mathbf{H}|} < \frac{(r+1)^2}{r}. \quad (3)$$

Визначення напрямку. Для кожної ключової точки будується гістограма напрямків локальних градієнтів із 36 інтервалами в діапазоні 0-360°. Домінуючий напрямок призначається точці, що забезпечує інваріантність до повороту зображення.

Формування дескриптора. Навколо кожної ключової точки у відповідному масштабі обчислюються гістограми градієнтів у 16 областях розміром 4×4. Дескриптор являє собою вектор із 128 елементів, нормалізований для досягнення інваріантності до змін освітлення.

1.2 Метод SURF

Метод SURF (Speeded-Up Robust Features) [1] розроблено з метою прискорення обчислень порівняно з SIFT. Він замінює Гаусові фільтри швидкими апроксимаціями у вигляді бокс-фільтрів, що дозволяє ефективно обчислювати наближення детермінанта матриці Гессіана за допомогою інтегральних зображень. Дескриптор SURF будується на основі відгуків вейвлетів Хаара. Метод демонструє значно вищу швидкість при порівнянні якості виявлення ключових точок, хоча може поступатися SIFT при суттєвих змінах освітлення та точки огляду сцени.

1.3 Метод BRISK

Метод BRISK (Binary Robust Invariant Scalable Keypoints) [5] використовує бінарний дескриптор, що формується на основі порівняння яскравостей пікселів у заздалегідь визначеному шаблоні навколо ключової точки. Бінарна природа дескриптора суттєво скорочує обчислювальні витрати на порівняння. Метод є масштабно-інваріантним завдяки застосуванню піраміди зображень. Серед переваг – висока стійкість до змін освітлення та малий час обчислення дескриптора.

1.4 Метод ORB

Метод ORB (Oriented FAST and Rotated BRIEF) [15] поєднує детектор FAST для виявлення ключових точок із модифікованим дескриптором BRIEF, доповненим механізмом врахування орієнтації точки. FAST відзначається надзвичайно низькими обчислювальними витратами на порівняння інтенсивностей пікселів у кільцевому

шаблоні. Дескриптор BRIEF є бінарним, що забезпечує ефективне порівняння. ORB є вільним від патентних обмежень та широко використовується у задачах реального часу.

1.5 Методи KAZE та AKAZE

Методи KAZE та AKAZE [2] використовують нелінійне дифузійне розмиття замість Гаусового, що дозволяє краще зберігати межі об'єктів при побудові масштабного простору. KAZE формує нелінійний дескриптор на основі нелінійно розмитих зображень, а AKAZE використовує бінарний дескриптор M-LDB (Modified Local Difference Binary). Обидва методи є масштабно-інваріантними; AKAZE відрізняється значно вищою швидкістю завдяки бінарному дескриптору.

2. Постановка задачі дослідження

Для порівняння ефективності розглянутих методів виділено дві ключові характеристики:

- кількість виявлених ключових точок як індикатор повноти охоплення деталей зображення;
- середній час пошуку однієї ключової точки (у мілісекундах) як міра обчислювальної ефективності методу.

Важливо наголосити, що поняття «надмірності» та «недостатності» кількості ключових точок є суб'єктивними і залежать від конкретної задачі. При аналізі результатів ураховується не лише кількість точок, а й їх просторова конфігурація відносно значущих деталей зображення – кутів, перетинів, меж об'єктів.

Усі методи реалізовані з використанням бібліотеки комп'ютерного зору OpenCV [20] та мови програмування Python.

3. Тестові зображення та план дослідження

Для дослідження використано чотири синтетичні та два реальні зображення.

Синтетичні зображення навмисно підібрані так, щоб охопити різні геометричні ситуації, характерні для практичних задач аналізу зображень (рис. 1).

Зображення 1.а складається з відрізків, деякі з яких перетинаються. На ньому лише 5 точок перетину, що дозволяє оцінити поведінку методів на зображеннях із мінімальною кількістю потенційних «точок прив'язки» і виключно прямолінійними структурами.

Зображення 1.б містить замкнені ламані (дві зірки, накладені одна на одну), що утворюють значно більшу кількість кутів і перетинів. Це зображення наближається до ситуації, у яких людина могла б впевнено ідентифікувати об'єкти.

Зображення 1.в побудоване із трьох еліпсів, що перетинаються. Характерна особливість – відсутність різких кутів: прямих «точок прив'язки» значно менше, ніж на зображенні б), а локальні деталі розподілені по плавним кривим.

Зображення 1.г є кольоровим і містить кілька логотипів сучасних мов програмування, що мають різноманітні кольори, деталі та текстові елементи. Воно найближче до реальних зображень, на яких застосовуються методи пошуку ключових точок.

Реальні тестові зображення є титульними сторінками двох книг з математики та програмування. Ці зображення містять складні кольорові візерунки, різноманітні шрифти та багату деталізацію, що наближує умови тестування до практичних задач комп'ютерного зору.

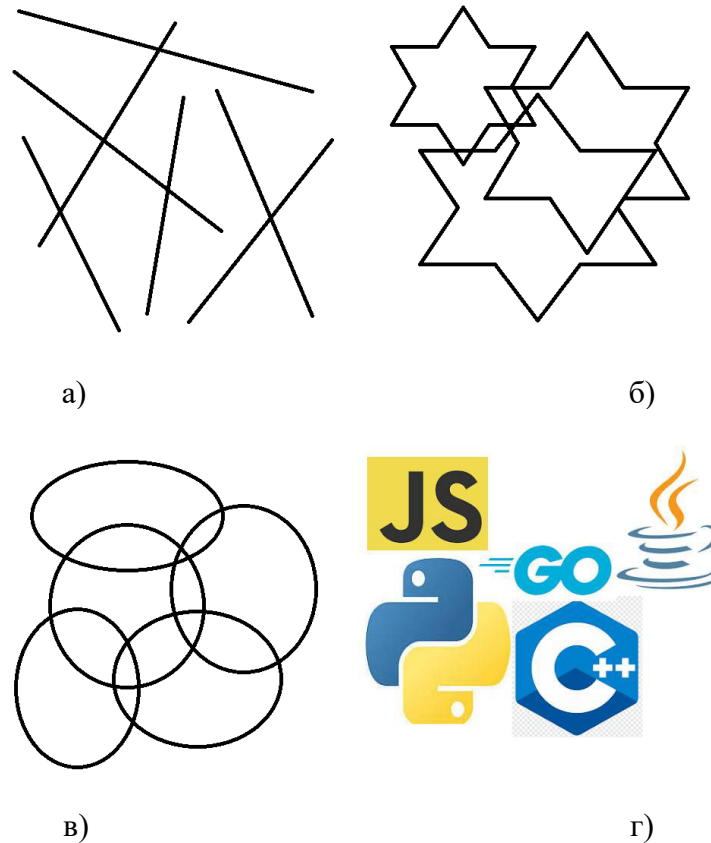


Рис. 1. Тестові синтетичні зображення: а) відрізки з перетинами; б) замкнені ламані (зірки); в) еліпси; г) кольорове зображення з логотипами мов програмування.

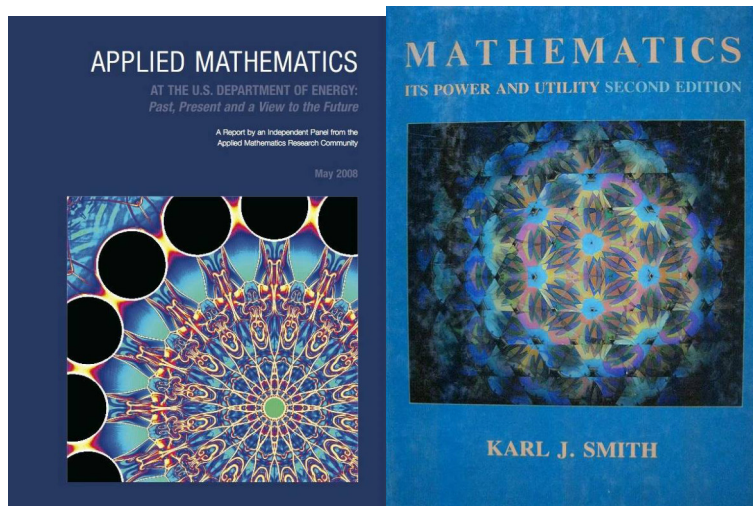


Рис. 2. Тестові реальні зображення (тительні сторінки книг), використані на другому етапі дослідження.

Дослідження проводилось у 2 етапи.

На першому етапі проводилась оцінка виявлення ключових точок на синтетичних зображеннях. Оцінювалась робота кожного методу для кожного зображення. Для цього виконувались наступні кроки:

- 1) відкривалось зображення та конвертувалось у відтінки сірого кольору;

- 2) ініціалізувалась реалізація відповідного методу у бібліотеці комп'ютерного зору OpenCV [20];
- 3) визначались ключові точки з оцінкою часу роботи;
- 4) дані про ключові точки зберігались у текстовому файлі. Для поточного дослідження ця інформація не потрібна, але вона може знадобитись у продовженні досліджень;
- 5) обраховувались дескриптори у відповідності до обраного методу та виводилась інформація про них у консоль;
- 6) отримані дані також зберігались у файл для використання результатів;
- 7) знайдені ключові точки відображались на зображенні для візуальної оцінки результату роботи методу.

На другому етапі проводилась оцінка роботи методів на реальних зображеннях. Ідеєю цього етапу було використання саме реальних зображень та застосування досліджуваних методів не один раз. Останнє досягалось за рахунок кількох застосувань одного й того ж методу на досліджуваних зображеннях. Для прикладу, ми знаходили на 2-х зображеннях ключові точки по 10 разів, що емулювало пошук ключових точок на 20 зображеннях.

Другий етап проводився з використанням наступних кроків:

- 1) попередньо усі тестові зображення конвертувались у відтінки сірого кольору та зберігались у окремому списку. Це робилось для того, щоб потім у кожному методі не витрачати час на конвертування;
- 2) перебирались методи по черзі і для кожного методу задану кількість разів (у програмі – 10) на кожному зображенні шукались ключові точки з визначенням часу роботи. Час додавався до сумарного, кількість знайдених ключових точок теж додавалась до сумарної. Знайдена кількість точок та сумарний час роботи повертались у основну програму;
- 3) отриманий час роботи, кількість знайдених ключових точок, середній час пошуку однієї точки та назва методу зберігались у результуючий файл для аналізу результатів.

4. Результати дослідження

4.1 Аналіз результатів на синтетичних зображеннях

Зображення з відрізками (рис. 3). На зображенні, що містить прямолінійні відрізки з п'ятьма точками перетину, методи продемонстрували суттєво різні результати.

Метод SIFT виявив найменшу кількість ключових точок, причому розташовані вони поза безпосередньою близькістю до точок перетину відрізків, що з практичної точки зору є небажаним для задачі зіставлення зображень. Методи SURF, BRISK та ORB виявили значно більшу кількість ключових точок; при цьому лише для ORB вони сконцентровані переважно у точках перетину – саме там, де «прив'язка» при зіставленні є найбільш надійною. KAZE та AKAZE розподілили ключові точки вздовж усієї довжини відрізків аналогічно до SURF та BRISK.

З точки зору обчислювальної ефективності методи розташовуються у такому порядку (від найменшого часу на одну точку): BRISK (0.04 мс), SURF (0.06 мс), AKAZE (0.14 мс), ORB (0.20 мс), KAZE (1.25 мс), SIFT (2.5 мс). Таким чином, SIFT є найповільнішим для цього типу зображень.

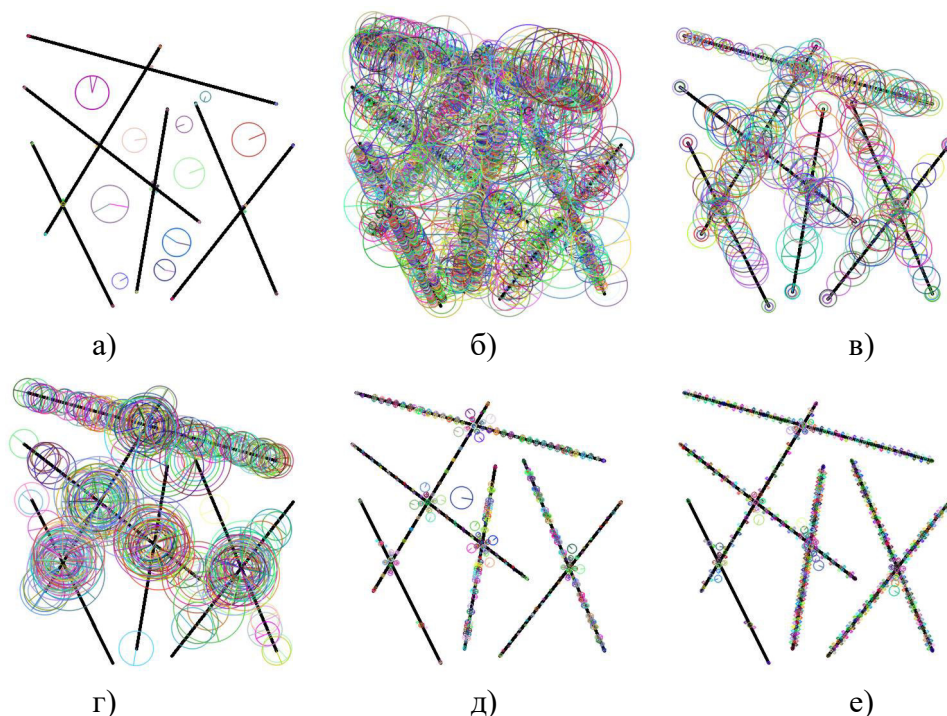


Рис. 3. Синтетичне зображення з відрізками із нанесеними ключовими точками: а) SIFT, б) SURF, в) BRISK, г) ORB, д) KAZE, е) AKAZE

Зображення з ламаними (рис. 4). При аналізі зображення з замкненими ламаними (зірками) ситуація дещо змінюється.

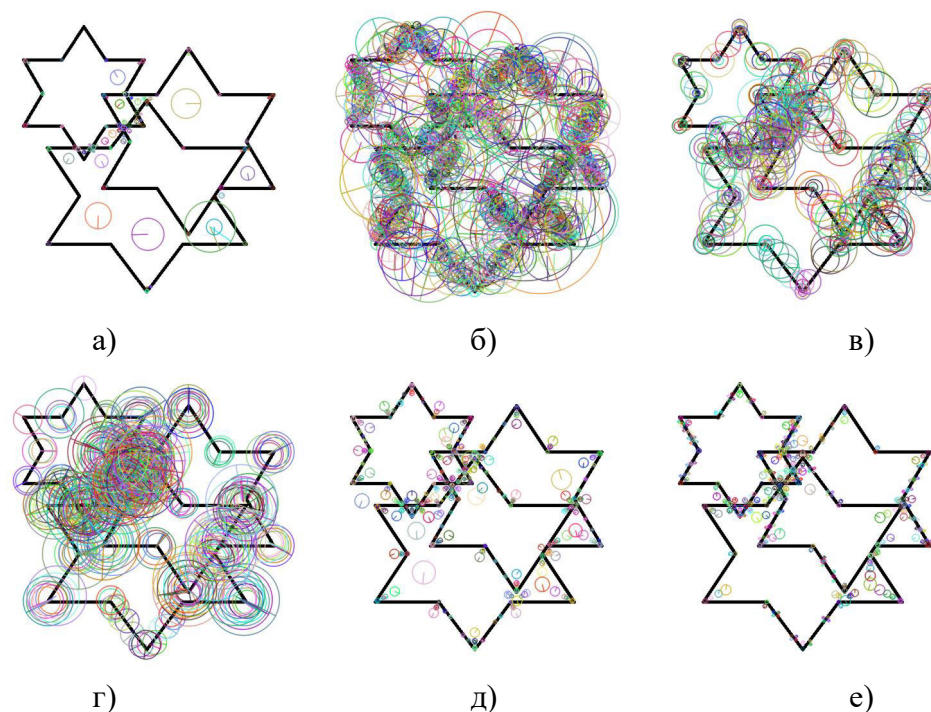


Рис. 4. Синтетичне зображення з ламаними із нанесеними ключовими точками: а) SIFT, б) SURF, в) BRISK, г) ORB, д) KAZE, е) AKAZE

SIFT та KAZE виявили найменшу кількість точок; більшість точок SIFT розташована поза вершинами та перетинами фігур, тоді як KAZE частково локалізує

точки у значущих місцях. SURF знайшов велику кількість точок по всій довжині ребер. Методи BRISK та ORB переважно концентрують точки у вершинах та перетинах ребер, хоча для обох методів характерна надмірна кількість точок з однаковими координатами та різними напрямками. З огляду на конфігурацію точок при достатньому їх фільтруванні перевагу можна надати методу ORB.

Порядок часової ефективності для цього зображення: BRISK (0.04 мс), SURF (0.09 мс), ORB (0.20 мс), AKAZE (0.30 мс), SIFT (1.06 мс), KAZE (1.70 мс).

Зображення з еліпсами (рис. 5).

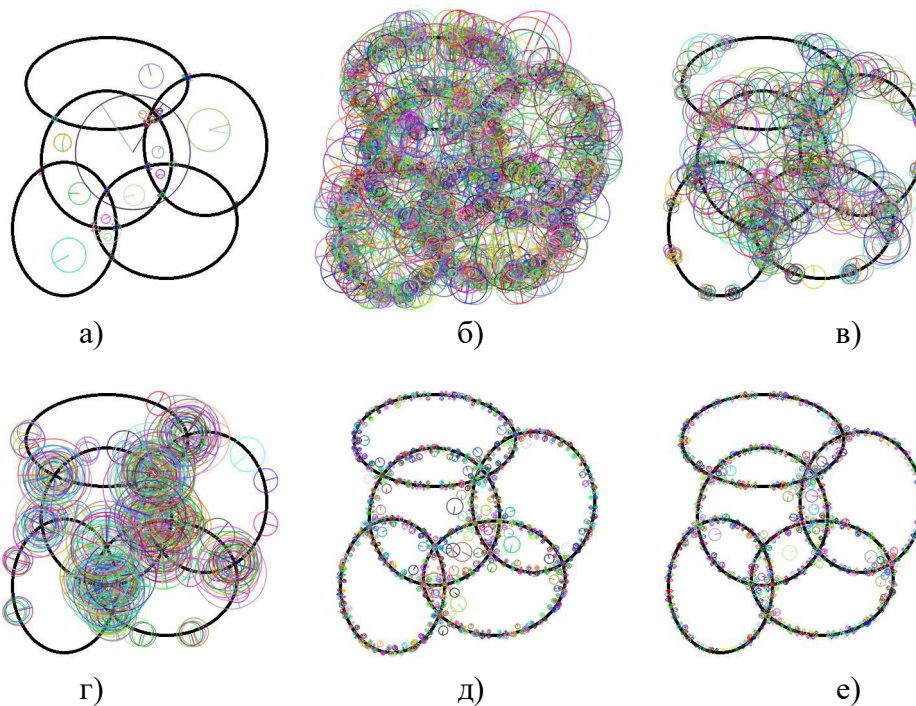


Рис. 5. Синтетичне зображення з еліпсами із нанесеними ключовими точками: а) SIFT, б) SURF, в) BRISK, г) ORB, д) KAZE, е) AKAZE

Загальна картина для зображення з еліпсами є подібною до попереднього випадку. SIFT виявив найменшу кількість точок, переважно у зонах з вираженою кривизною. ORB знову забезпечив раціональну конфігурацію точок у значущих локаціях при прийнятній швидкодії.

Порядок часової ефективності: BRISK (0.03 мс), SURF (0.06 мс), AKAZE (0.11 мс), ORB (0.20 мс), KAZE (0.60 мс), SIFT (1.70 мс).

Зображення з логотипами мов програмування (рис. 6).

Кольорове зображення з різноманітним вмістом показало більш збалансовані результати. SIFT виявив помірну кількість точок, розподілених по значущих деталях зображення – кутах літер, межах логотипів. ORB знов забезпечив другий найменший час обчислення (0.18 мс) після BRISK (0.03 мс) при задовільній конфігурації ключових точок.

Порядок часової ефективності: BRISK (0.03 мс), ORB (0.18 мс), AKAZE (0.20 мс), SURF (0.20 мс), SIFT (0.24 мс), KAZE (1.20 мс).

4.2 Результати на реальних зображеннях

На другому етапі дослідження кожен метод застосовувався по 10 разів до кожного з двох реальних зображень (титульних сторінок книг). Зведені результати наведено у таблиці 1.

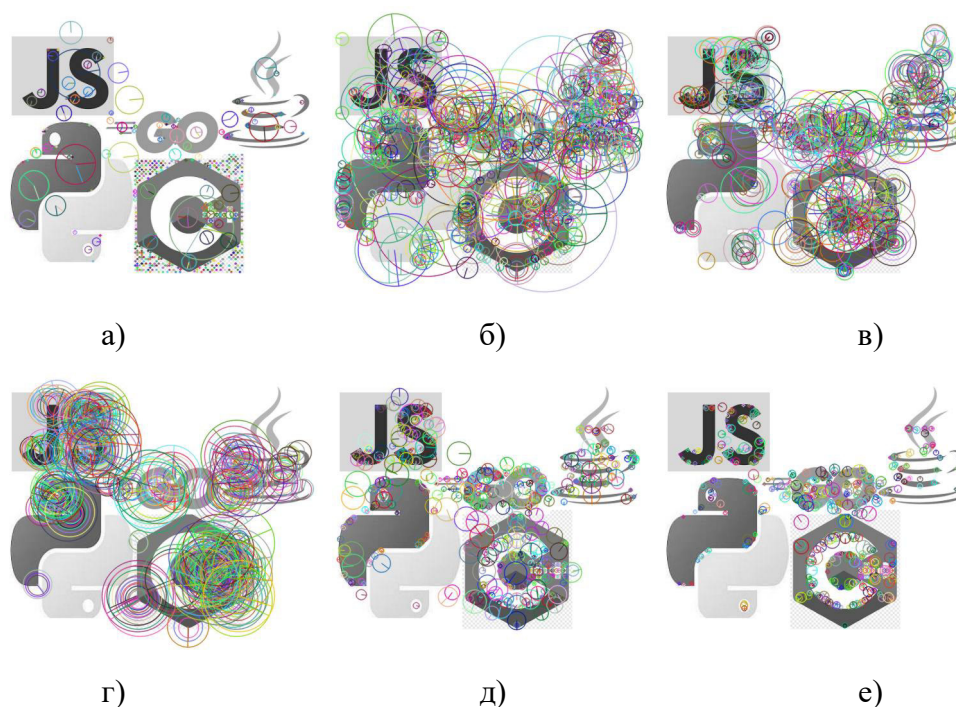


Рис. 6. Синтетичне зображення з логотипами мов програмування із нанесеними ключовими точками: а) SIFT, б) SURF, в) BRISK, г) ORB, д) KAZE, е) AKAZE

Таблиця 1

Результати застосування досліджуваних методів до пошуку ключових точок на реальних зображеннях

Метод	Ключових точок	Час пошуку усіх точок, мс	Час пошуку однієї точки, мс
BRISK	148300	4800	0.0324
ORB	10000	703	0.0703
SURF	86140	9202	0.1068
AKAZE	48650	7215	0.1483
SIFT	63730	14036	0.2202
KAZE	53230	45838	0.8611

Примітка: методи у таблиці впорядковані за зростанням усередненого часу пошуку однієї ключової точки.

Дані таблиці 1 свідчать про суттєву варіативність результатів між методами. BRISK демонструє найменший час пошуку однієї точки (0.0324 мс), однак виявляє надмірно велику їх кількість (148300 за 20 застосувань), що свідчить про низьку вибірковість методу та ймовірну присутність значної частки нестабільних точок. ORB забезпечує в 14.8 рази меншу кількість виявлених точок порівняно з BRISK та другий найменший час обчислення (0.0703 мс). SURF виявляє велику кількість точок (86140), але витрачає значний час на їх обробку. SIFT, попри найбільш теоретично обґрунтований підхід, демонструє час обчислення більш ніж у 6 разів більший, ніж ORB. KAZE є явним аутсайдером за швидкістю – час пошуку однієї точки перевищує аналогічний показник BRISK у 26.6 рази.

5. Аналіз результатів та наукова новизна

5.1 Зіставлення методів за сукупністю показників

Проведений аналіз демонструє принципове протиріччя між якістю виявлення ключових точок та швидкістю методів. Теоретично найбільш обґрунтований метод SIFT, що використовує точну апроксимацію масштабно-нормалізованого Лапласіана Гауса та детальну верифікацію кандидатів, виявляється найповільнішим у більшості тестованих випадків. Натомість методи, що використовують бінарні дескриптори (BRISK, ORB), досягають значного прискорення ціною деякого зниження вибіркості або точності локалізації.

Суттєвим спостереженням є те, що найшвидший метод – BRISK – виявляє надмірну кількість ключових точок, розподілених по зображенню, що ускладнює їх використання у практичних задачах без додаткової фільтрації. ORB демонструє більш збалансований результат: кількість виявлених точок є значно меншою та їх конфігурація краще відповідає значущим локальним деталям зображення.

Варто зазначити, що характер залежності між типом зображення та ефективністю методу є нетривіальним. На зображенні з відрізками (мінімальна кількість значущих деталей) жоден метод, крім ORB, не зосередив точки переважно у точках перетину. На зображеннях із замкненими ламаними та еліпсами методи BRISK та ORB виявились більш «прицільними», тоді як SURF та KAZE/AKAZE рівномірно покрили всі ребра.

5.2 Особливості методів щодо типу контенту

Детальний аналіз результатів дозволяє сформулювати наступні узагальнення.

Для зображень із переважно прямолінійними структурами та малою кількістю характерних деталей (тип а) метод SIFT забезпечує найбільш стриману реакцію, уникаючи масового виявлення нестабільних точок вздовж прямих ліній. Це, у поєднанні з надійністю дескриптора, може бути перевагою у задачах, де точність важливіша за швидкість.

Для зображень із кутовими структурами та великою кількістю перетинів (типи б та в) метод ORB забезпечує прийнятний компроміс між якістю конфігурації точок та швидкістю. BRISK, незважаючи на найменший час обчислення, потребує додаткового кроку фільтрації.

Для насичених кольорових зображень із текстурою та різноманітними деталями (типи г та реальні зображення) відмінності між методами з точки зору якості конфігурації стають менш вираженими, тоді як перевага ORB та BRISK у швидкодії зберігається.

5.3 Практична значущість дослідження

Практична значущість полягає у тому, що отримані результати можуть слугувати основою для обґрунтованого вибору методу виявлення ключових точок під конкретну прикладну задачу з урахуванням вимог до точності, швидкодії та типу оброблюваних зображень.

Висновки

У результаті проведеного дослідження виконано порівняльний аналіз шести методів виявлення ключових точок на зображеннях різного типу. Сформульовано такі висновки.

1. Методи виявлення ключових точок суттєво відрізняються за ефективністю залежно від вмісту зображення. Підхід, що є оптимальним для одного типу зображень, може виявитися неефективним для іншого, тому вибір методу повинен ґрунтуватись на аналізі характеристик задачі.

2. Найвища обчислювальна ефективність (найменший час на одну ключову точку) досягається методами BRISK та ORB, що використовують бінарні дескриптори. Однак BRISK демонструє надмірну кількість виявлених точок, що потребує додаткової фільтрації.

3. Метод ORB забезпечує найбільш збалансований результат: прийнятний час обчислення (0,0703 мс на точку для реальних зображень), поміркована кількість виявлених точок та задовільна їх конфігурація відносно значущих деталей зображення. Це робить ORB перспективним вибором для задач реального часу та систем із обмеженими обчислювальними ресурсами.

4. Метод SIFT, незважаючи на найбільші обчислювальні витрати, забезпечує стабільну та теоретично обґрунтовану поведінку. Для задач, де точність є пріоритетом над швидкістю, SIFT залишається конкурентоспроможним.

5. Метод KAZE є найповільнішим серед розглянутих (0,8611 мс на точку), що суттєво обмежує його застосування у задачах, де важлива швидкість.

6. Встановлено, що не існує методу, який є абсолютним переможцем за сукупністю характеристик. Ефективне застосування методів пошуку ключових точок потребує знання особливостей їх роботи на різних типах зображень, наявних обчислювальних ресурсів та конкретних вимог задачі.

Список використаної літератури

1. Bay H., Ess A., Tuytelaars T., Van Gool L. Speeded-Up Robust Features (SURF) // *Computer Vision and Image Understanding*. – 2008. – Vol. 110, № 3. – P. 346–359.
2. Brown M., Lowe D.G. Invariant features from interest point groups // *Proceedings of the British Machine Vision Conference*. – Cardiff, Wales, 2002. – P. 656–665.
3. Crowley J. L., Parker A. C. A representation for shape based on peaks and ridges in the difference of low-pass transform // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 1984. – № 6(2). – P. 156–170.
4. Harris C., Stephens M. A combined corner and edge detector // *Proceedings of the Fourth Alvey Vision Conference*. – Manchester, UK, 1988. – P. 147–151.
5. Leutenegger S., Chli M., Siegwart R. Y. BRISK: Binary Robust Invariant Scalable Keypoints // *Proceedings of the IEEE International Conference on Computer Vision*. – 2011. – P. 2548–2555.
6. Lindeberg T. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention // *International Journal of Computer Vision*. – 1993. – № 11(3). – P. 283–318.
7. Lowe D. G. Object recognition from local scale-invariant features // *Proceedings of the International Conference on Computer Vision*. – 1999. – P. 1150–1157.
8. Lowe D. G. Distinctive image features from scale-invariant keypoints // *International Journal of Computer Vision*. – 2004. – Vol. 60, № 2. – P. 91–110.
9. Matas J., Chum O., Urban M., Pajdla T. Robust wide baseline stereo from maximally stable extremal regions // *Proceedings of the British Machine Vision Conference*. – Cardiff, Wales, 2002. – P. 384–393.
10. Mikolajczyk K., Schmid C. An affine invariant interest point detector // *Proceedings of the European Conference on Computer Vision*. – Copenhagen, Denmark, 2002. – P. 128–142.
11. Mikolajczyk K., Zisserman A., Schmid C. Shape recognition with edge-based features // *Proceedings of the British Machine Vision Conference*. – Norwich, UK, 2003.
12. Moravec H. Rover visual obstacle avoidance // *Proceedings of the International Joint Conference on Artificial Intelligence*. – Vancouver, Canada, 1981. – P. 785–790.
13. Nelson R. C., Selinger A. Large-scale tests of a keyed, appearance-based 3-D object recognition system // *Vision Research*. – 1998. – № 38(15). – P. 2469–2488.
14. Pope A. R., Lowe D. G. Probabilistic models of appearance for 3-D object recognition // *International Journal of Computer Vision*. – 2000. – № 40(2). – P. 149–167.
15. Rublee E., Rabaud V., Konolige K., Bradski G. ORB: an efficient alternative to SIFT or SURF // *Proceedings of the IEEE International Conference on Computer Vision*. – 2011. – P. 2564–2571.

16. Schmid C., Mohr R. Local grayvalue invariants for image retrieval // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 1997. – № 19(5). – P. 530–534.
17. Shokoufandeh A., Marsic I., Dickinson S. J. View-based object recognition using saliency maps // *Image and Vision Computing*. – 1999. – № 17. – P. 445–460.
18. Torr P. Motion Segmentation and Outlier Detection: Ph.D. Thesis / University of Oxford. – Oxford, UK, 1995.
19. Zhang Z., Deriche R., Faugeras O., Luong Q. T. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry // *Artificial Intelligence*. – 1995. – Vol. 78. – P. 87–119.
20. Бібліотека комп'ютерного зору OpenCV [Електронний ресурс]. – Режим доступу: <https://opencv.org/>

References

1. Bay H., Ess A., Tuytelaars T., Van Gool L. (2008) Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3), 346–359.
2. Brown M., Lowe D.G. (2002) Invariant features from interest point groups. *Proceedings of the British Machine Vision Conference*, Cardiff, Wales, 656–665.
3. Crowley J.L., Parker A.C. (1984) A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2), 156–170.
4. Harris C., Stephens M. (1988) A combined corner and edge detector. *Proceedings of the Fourth Alvey Vision Conference*, Manchester, UK, 147–151.
5. Leutenegger S., Chli M., Siegwart R.Y. (2011) BRISK: Binary Robust Invariant Scalable Keypoints. *Proceedings of the IEEE International Conference on Computer Vision*, 2548–2555.
6. Lindeberg T. (1993) Detecting salient blob-like image structures and their scales with a scale-space primal sketch. *International Journal of Computer Vision*, 11(3), 283–318.
7. Lowe D.G. (1999) Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision*, 1150–1157.
8. Lowe D.G. (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110.
9. Matas J., Chum O., Urban M., Pajdla T. (2002) Robust wide baseline stereo from maximally stable extremal regions. *Proceedings of the British Machine Vision Conference*, Cardiff, Wales, 384–393.
10. Mikolajczyk K., Schmid C. (2002) An affine invariant interest point detector. *Proceedings of the European Conference on Computer Vision*, Copenhagen, Denmark, 128–142.
11. Mikolajczyk K., Zisserman A., Schmid C. (2003) Shape recognition with edge-based features. *Proceedings of the British Machine Vision Conference*, Norwich, UK.
12. Moravec H. (1981) Rover visual obstacle avoidance. *Proceedings of the International Joint Conference on Artificial Intelligence*, Vancouver, Canada, 785–790.
13. Nelson R.C., Selinger A. (1998) Large-scale tests of a keyed, appearance-based 3-D object recognition system. *Vision Research*, 38(15), 2469–2488.
14. Pope A.R., Lowe D.G. (2000) Probabilistic models of appearance for 3-D object recognition. *International Journal of Computer Vision*, 40(2), 149–167.
15. Rublee E., Rabaud V., Konolige K., Bradski G. (2011) ORB: an efficient alternative to SIFT or SURF. *Proceedings of the IEEE International Conference on Computer Vision*, 2564–2571.
16. Schmid C., Mohr R. (1997) Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5), 530–534.
17. Shokoufandeh A., Marsic I., Dickinson S.J. (1999) View-based object recognition using saliency maps. *Image and Vision Computing*, 17, 445–460.
18. Torr P. (1995) Motion Segmentation and Outlier Detection. Ph.D. Thesis, University of Oxford, UK.
19. Zhang Z., Deriche R., Faugeras O., Luong Q.T. (1995) A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78, 87–119.
20. OpenCV: Open Source Computer Vision Library. Available at: <https://opencv.org/>

HAVRYUSHENKO Anna Mykolaivna,

teacher of Mathematics and Computer Science at Cherkasy General Education School of Levels I-III No. 32 of Cherkasy City Council, Cherkasy, Ukraine

BOROZDYKH Kateryna Serhiivna,

student of Cherkasy General Education School of Levels I-III No. 32 of Cherkasy City Council, Cherkasy, Ukraine

COMPARATIVE ANALYSIS OF KEYPOINT DETECTION METHODS IN IMAGES

Summary. Introduction. *Image search – finding one image within another – is a fundamental problem in computer vision, with applications in object recognition, motion tracking, 3D reconstruction, aerial image analysis and autonomous navigation systems. The core challenge is identifying local image features, called keypoints, that remain stable under changes in scale, rotation, viewpoint and illumination. Since the pioneering work of Moravec and Harris on corner detection, and the subsequent development of SIFT by Lowe, the field has produced a number of competing methods that vary significantly in their theoretical foundations, accuracy and computational efficiency.*

Purpose. *This paper presents a systematic comparison of six keypoint detection methods – SIFT, SURF, BRISK, ORB, KAZE and AKAZE – evaluated on images of varying geometric complexity. Two evaluation criteria are used: the spatial configuration quality of detected keypoints relative to salient image features, and the average computation time per keypoint.*

Results. *The mathematical background of SIFT is described in detail, covering scale-space extrema detection via the Difference of Gaussians function, precise keypoint localization using Taylor series expansion of the scale-space function, orientation assignment and descriptor computation. For all six methods, experiments were conducted on four synthetic images (line segments, closed polylines, ellipses, and a color image with programming language logos) and two real-world images (book covers). Results show that BRISK and ORB are the fastest methods, with per-keypoint times of 0.032 ms and 0.070 ms respectively on real images. However, BRISK detects an excessive number of keypoints (148,300 over 20 trials), many of which are unstable. ORB provides a more balanced result: moderate keypoint count (10,000), the second-fastest computation time, and a spatial configuration concentrated near meaningful image features such as corners and intersections. SIFT, while the slowest method (0.220 ms per keypoint), provides a theoretically well-grounded and selective response. KAZE is the slowest across all test cases (0.861 ms per keypoint), limiting its practical use in time-critical applications.*

Conclusion. *No single method dominates across all evaluation criteria and image types. The optimal choice depends on the characteristics of the target images and the specific requirements of the application. For real-time systems with limited computational resources, ORB provides the best trade-off between speed and keypoint quality. For tasks where accuracy is prioritized over speed, SIFT remains competitive. These findings highlight the importance of method selection guided by systematic benchmarking rather than general assumptions.*

Keywords: *keypoints, image descriptor, SIFT, SURF, BRISK, ORB, KAZE, AKAZE, computer vision, scale-invariant feature transform.*

Одержано редакцією 09.11.2024 р.
Прийнято до публікації 11.12.2024 р.