

Цікаві алгоритми



Зміст

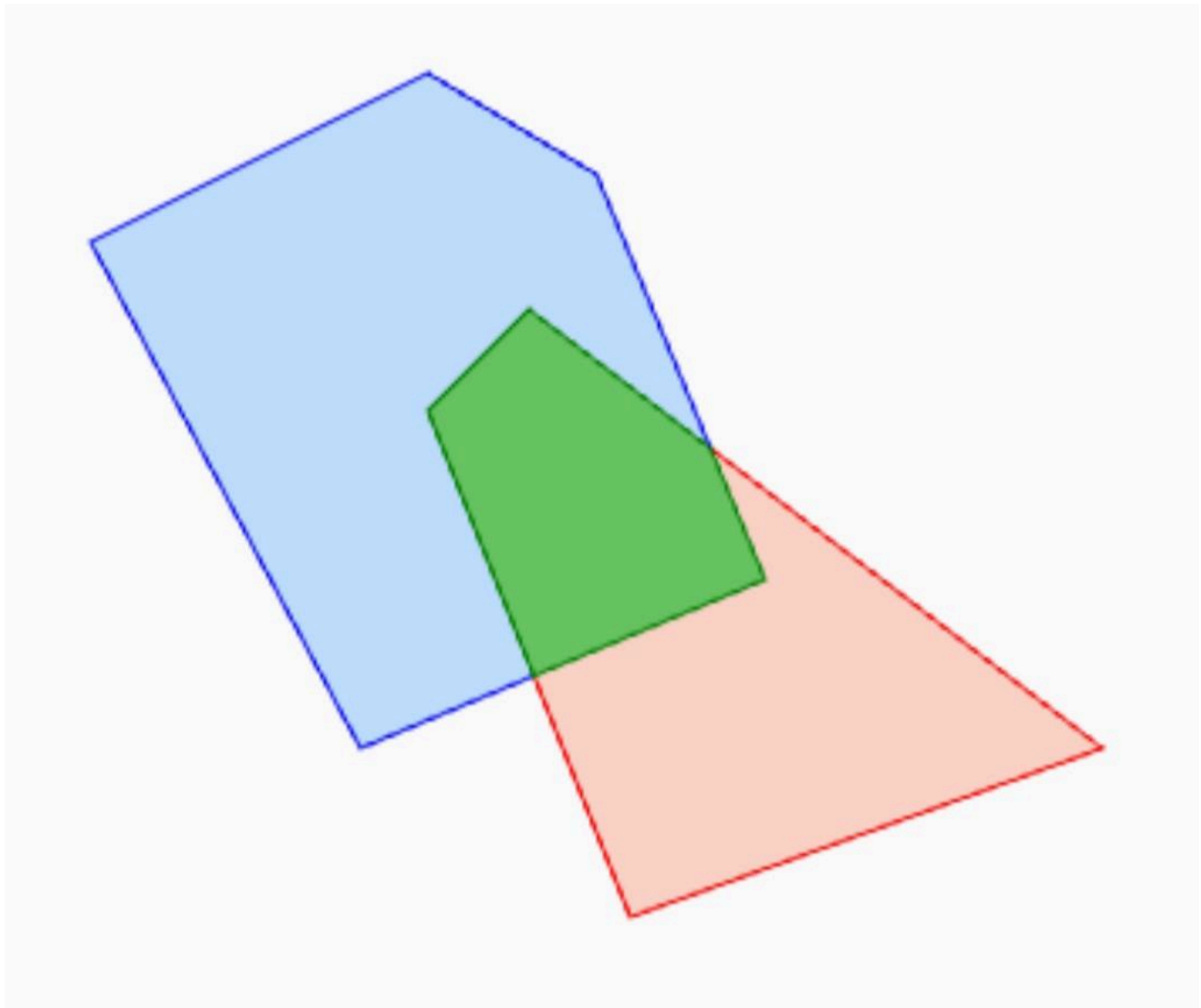
В статті описані алгоритми

- Sutherland–Hodgman clipping algorithm
- Alternating Tripod Gait
- Frame Differencing (motion detection)
- Classical OCR (Optical Character Recognition)
- Reticle-seeker (positioning)

Укладач: Дмитро Попов.

<https://www.linkedin.com/in/dima1popov>

Алгоритм відсікання Сазерленда-Ходжмена для опуклих багатокутників



У нас є два багатокутники (полігони) A та B, і ми хочемо знайти результат їхнього перетину, тобто новий багатокутник C.

Для цього потрібно виконати два основні кроки:

1. Визначити точки багатокутника B, які належать багатокутнику A.
2. Визначити точки перетину сторін багатокутників.

На основі цих точок ми побудуємо новий багатокутник. Розглядаються лише опуклі багатокутники (convex).

Щоб визначити, чи знаходиться точка всередині багатокутника, можна застосувати два алгоритми:

1. Метод променя.

Від заданої точки проводимо промінь. Якщо промінь перетинає багатокутник рівно один раз — точка належить багатокутнику. Якщо промінь взагалі не перетинає багатокутник або перетинає його два чи більше разів, тоді точка знаходиться поза межами багатокутника.

2. Метод орієнтації сторін.

Обходимо кожну сторону багатокутника проти годинникової стрілки та перевіряємо, чи досліджувана точка завжди розташована ліворуч від усіх сторін. Якщо так — точка всередині багатокутника; якщо для якоїсь зі сторін точка виявиться праворуч — вона зовні.

Багатокутники задані як масив точок, де кожна точка — це пара координат (x, y) . Кожні дві сусідні точки утворюють сторону.

Щоб визначити, чи точка розташована праворуч або ліворуч від сторони, застосовується спеціальна формула, яку ми наведемо нижче — після того, як розглянемо метод знаходження точки перетину двох відрізків.

Отже, крім перевірки належності точки багатокутнику, потрібно ще й знаходити точки перетину сторін. Для цього також існує спеціальна формула.

Нижче наведено код для функції визначення точки перетину та положення точки відносно сторони.

Тут параметри функції це точки: $[x, y]$.

```

function inside(p, a, b) {

    return (b[0]-a[0])*(p[1]-a[1]) - (b[1]-a[1])*(p[0]-a[0]) >=
0;
}

function intersection(p1,p2,a,b) {
    const A1 = p2[1]-p1[1];
    const B1 = p1[0]-p2[0];
    const C1 = A1*p1[0]+B1*p1[1];
    const A2 = b[1]-a[1];
    const B2 = a[0]-b[0];
    const C2 = A2*a[0]+B2*a[1];
    const det = A1*B2 - A2*B1;
    if (det === 0) return p1; // паралельні
    return [ (B2*C1-B1*C2)/det, (A1*C2-A2*C1)/det ];
}

```

Код

```

function inside(p, a, b) {

    return (b[0]-a[0])*(p[1]-a[1]) - (b[1]-a[1])*(p[0]-a[0]) >= 0;

}

```

```

function intersection(p1,p2,a,b) {

    const A1 = p2[1]-p1[1];

    const B1 = p1[0]-p2[0];

    const C1 = A1*p1[0]+B1*p1[1];

```

```
const A2 = b[1]-a[1];

const B2 = a[0]-b[0];

const C2 = A2*a[0]+B2*a[1];

const det = A1*B2 - A2*B1;

if (det === 0) return p1; // паралельні

return [ (B2*C1-B1*C2)/det, (A1*C2-A2*C1)/det ];

}
```

Для складних геометричних об'єктів задача визначення факту перетину та площі перетину може бути розв'язана за допомогою стохастичних методів. Зокрема, застосування методу Монте-Карло дозволяє апроксимувати площу спільної області шляхом випадкової генерації множини точок та статистичної оцінки частки тих із них, що належать обом множинам.

Альтернативним підходом є апроксимація складної області опуклим або неопуклим багатокутником, після чого задача зводиться до виявлення перетину багатокутників із використанням відомих алгоритмів обчислювальної геометрії.

Для простих фігур застосовуються спеціалізовані методи з нижчою обчислювальною складністю. Наприклад, для двох кіл достатньо обчислити відстань між їхніми центрами та порівняти її із сумою або різницею радіусів, що дозволяє встановити факт наявності або відсутності перетину.

Алгоритми ходи

Триподна хода — це спосіб пересування шестинокої системи, при якому ноги працюють двома групами: три стоять на землі, утворюючи стійкий трикутник опори, а три інші роблять крок. Потім групи міняються місцями. Завдяки такій схемі завжди зберігається рівновага, бо платформа постійно має три точки опори.

Цей принцип використовується в гексаподних роботах і спостерігається у комах, зокрема мурах та жуків. Він забезпечує стабільний рух уперед, а також дає можливість виконувати повороти: з одного боку ноги роблять коротші кроки, а з іншого — довші, або ж групи ніг рухаються в протилежних напрямках, створюючи розворот на місці.

Триподна хода — це спосіб пересування, при якому у будь-який момент часу три кінцівки торкаються поверхні, а три інші підняті. Ноги, що стоять, утворюють трикутник опори, завдяки чому тіло зберігає рівновагу. Підняті кінцівки переносяться вперед без контакту з опорною поверхнею, тому відсутнє волочіння чи зайве буксування. Такий принцип характерний для комах і використовується в мобільній робототехніці для підвищення стійкості.

Для повороту на місці триподна хода також ефективна: ноги, що знаходяться на поверхні, прикладають сили з різним напрямком, утворюючи крутий момент навколо центра тіла. При цьому ковзання не виникає, бо відбувається чітке чергування фаз опори та переносу. Таким чином, триподна хода поєднує стійкість із високою маневровістю, що робить її однією з найефективніших схем багатоногого пересування.

Це **шестиногий хід (триподна хода)**, який використовується у робототехніці та в природі (наприклад, у комах).

- ♦ Маємо платформу (тіло робота/механізму) з **6 ногами**:
 - з одного боку — ноги **1, 2, 3**,
 - з іншого боку — ноги **4, 5, 6**.
- ♦ Рух організовано так, що **одночасно підіймаються три ноги, які не сусідні, а утворюють трикутник опори**.

Наприклад:

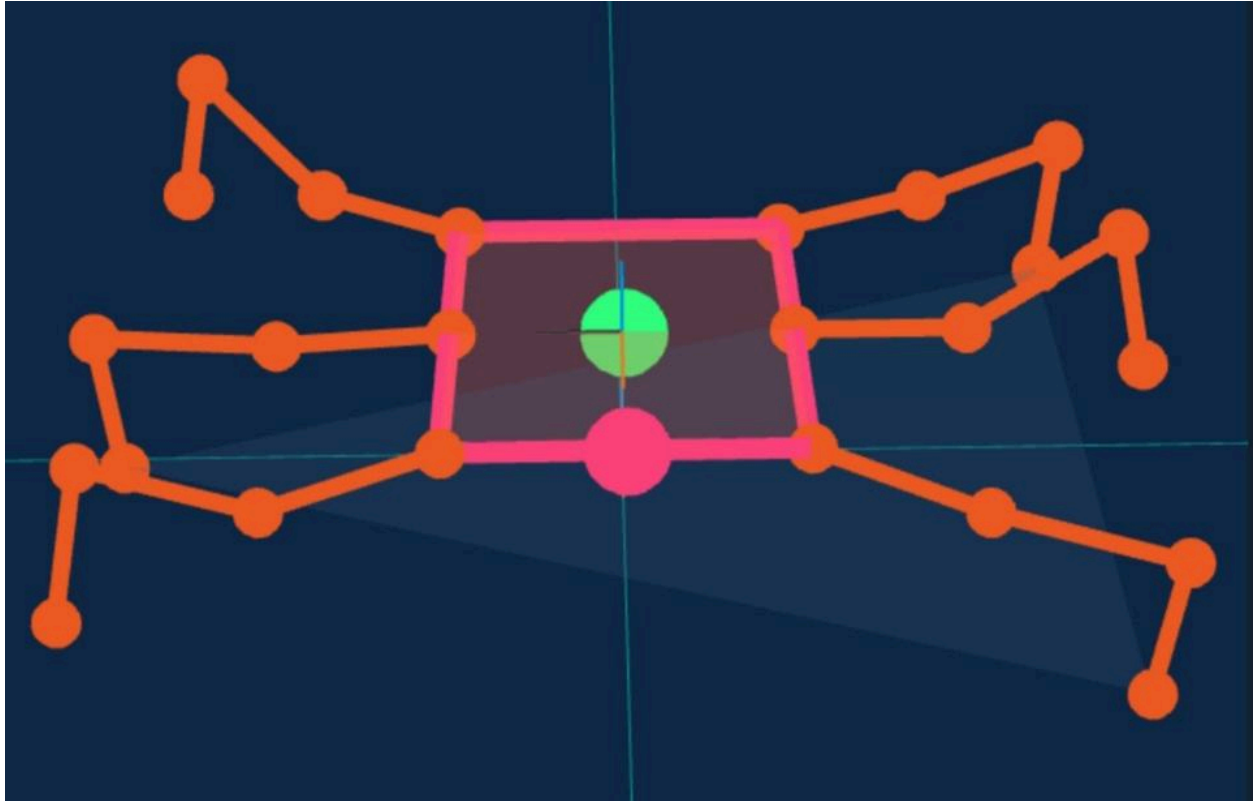
- Підняті **1, 5, 3**,
- На землі залишаються **2, 4, 6**.

Таким чином у будь-який момент часу є **три точки опори**, що гарантує стійкість конструкції. Потім фази міняються: ноги, які стояли, піднімаються, а ті, що були в повітрі, стають опорними.

✚ Це називається **альтернативна триподна хода** (англ. *alternating tripod gait*). Її використовують у:

- гексаподних роботах (шестиногих роботах),
- мурах, павуків, жуків та інших комах.

✓ Перевага — максимальна стійкість і плавність руху, бо центр мас завжди знаходиться всередині трикутника опори.



<https://hexapod.netlify.app/walking-gaits>

У гексаподів (6-ногих роботів) поворот робиться не як у колісних машин, а за рахунок зміни напрямку кроку ніг.

- ♦ Якщо прямо — то триподна хода: $(1,3,5) \leftrightarrow (2,4,6)$.
- ♦ Якщо поворот — треба змінити траєкторію ніг. Є два основні способи:

1. Поворот як кермо (Yaw-rotation)

Ноги з одного боку рухаються коротшим кроком, а з іншого — довшим.

Наприклад, щоб повернути вліво, ліва трійка (1,2,3) робить короткий крок уперед, а права трійка (4,5,6) — довший.

В результаті платформа плавно повертає, як автомобіль.


2. Розворот на місці (Pivot-turn)

Платформа може обертатися майже на місці, якщо:

Ноги однієї трійки роблять крок вперед і трохи вбік,
А інша трійка — назад і трохи вбік у протилежний бік.
Так створюється крутний момент, і робот розвертається на місці, як танк.

3. Диференційований поворот (для точності)

Можна комбінувати — наприклад, центральні ноги (2 і 5) рухаються по прямій, а крайні (1,3,4,6) описують дугу.
Це дає більш плавний поворот із контролем радіуса.

 У комах це виглядає так само: мураха може або плавно змінювати напрям, або крутитися на місці, перебираючи лапками.

Тринога хода:

Опис: Три ноги (по одній з кожного боку, зазвичай ліва передня, права середня, ліва задня і навпаки) рухаються разом, тоді як інші три залишаються на землі, утворюючи стабільну триногу.

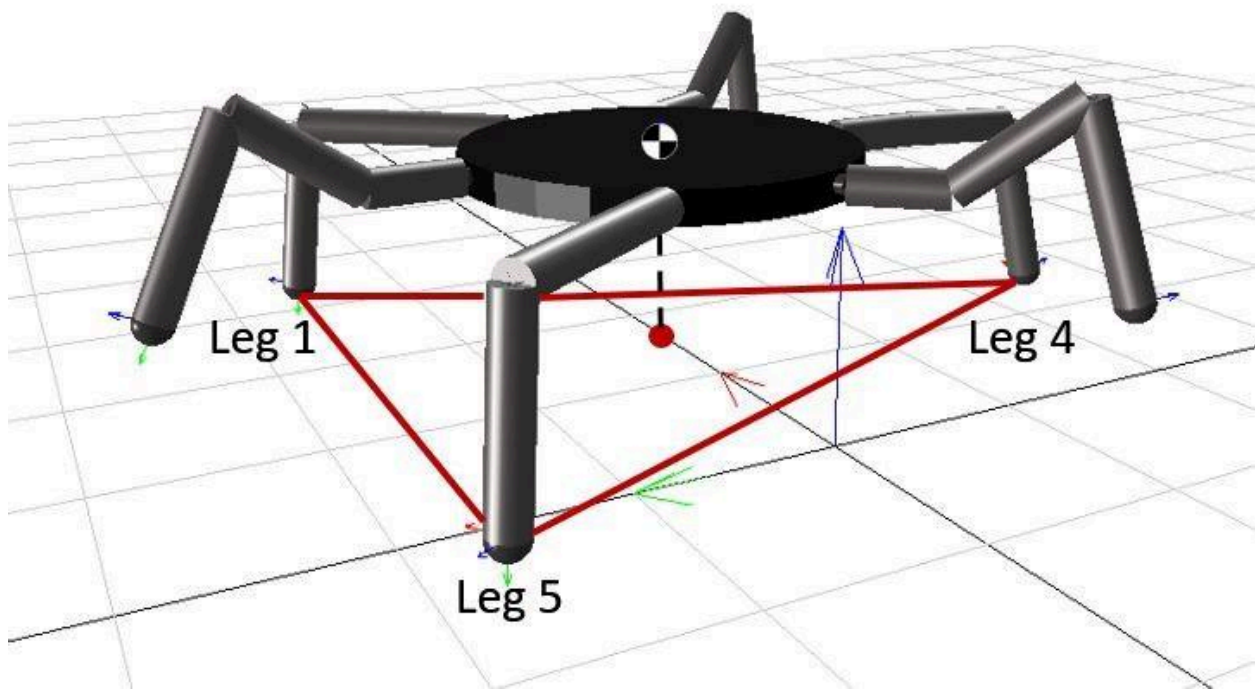
Характеристики:

Швидка та ефективна для гладких поверхонь.

Забезпечує хорошу стійкість, оскільки три ноги завжди контактують із землею.

Поширена у комах, таких як таргани, що рухаються з високою швидкістю.

Хід: Чергує два набори з трьох ніг (наприклад, ліва передня + права середня + ліва задня, потім права передня + ліва середня + права задня).



(Image from:
<https://hackaday.io/project/21904-hexapod-modelling-path-planning-and-control/log/62326-3-fundamentals-of-hexapod-robot>)

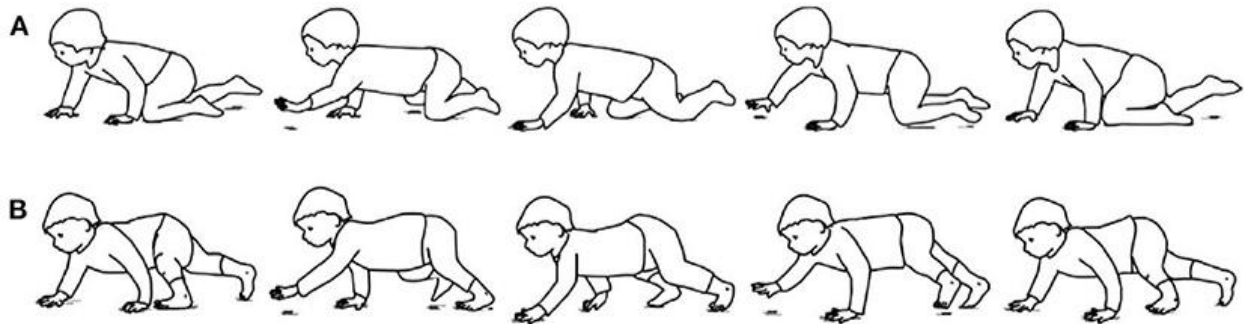
Чотириногі (Tetrapods)

Алгоритм руху чотириногих базується на принципі чергування фаз опори та перенесення для кожної кінцівки. При повільній ході (так званий *crawl gait*) завжди три ноги залишаються на землі, а одна виконує крок уперед. Це забезпечує статичну стійкість: центр мас тіла завжди знаходиться всередині опорного трикутника. Послідовність руху ніг зазвичай організовують циклічно (наприклад: ліва передня → права задня → права передня → ліва задня), де кожна нога піднімається, переноситься вперед по заданій траєкторії та знову торкається опори.

Для швидшого пересування використовують інші види ходи (*gait*): *рись* (рух діагональних пар ніг), *інохідь* (рух бічних пар), *галопа* (стрибкоподібний рух із фазою, коли всі ноги у повітрі). Вибір алгоритму залежить від задачі: при русі по складному рельєфу перевагу надають повільній ході для стабільності, тоді як для швидкості застосовують більш динамічні схеми. Таким чином, управління чотириногими будується на поєднанні фазових зсувів між ногами та контролю балансу тіла.

Повзання людини — це спосіб пересування, коли тіло знаходиться низько до землі, а руки й ноги по чергову виконують рухи для просування вперед. Найбільш типовим є так зване перехресне повзання: права рука і ліва нога рухаються одночасно, потім ліва рука й права нога. Така схема дозволяє зберігати рівновагу й плавно переміщатися.

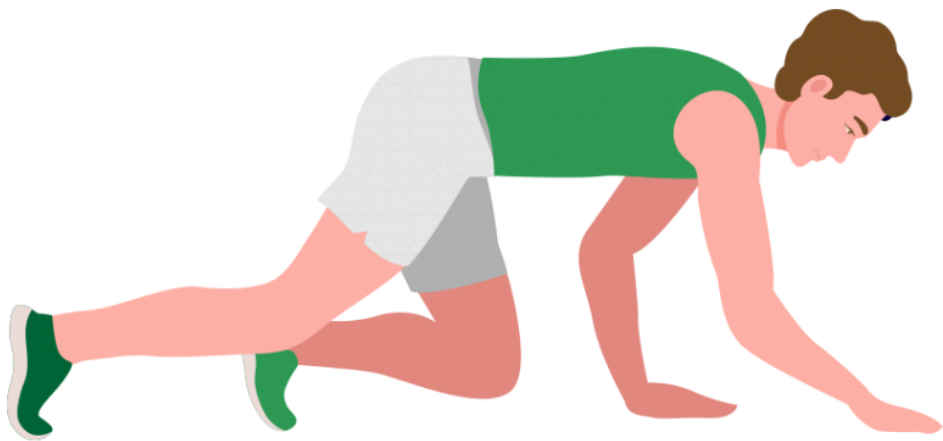
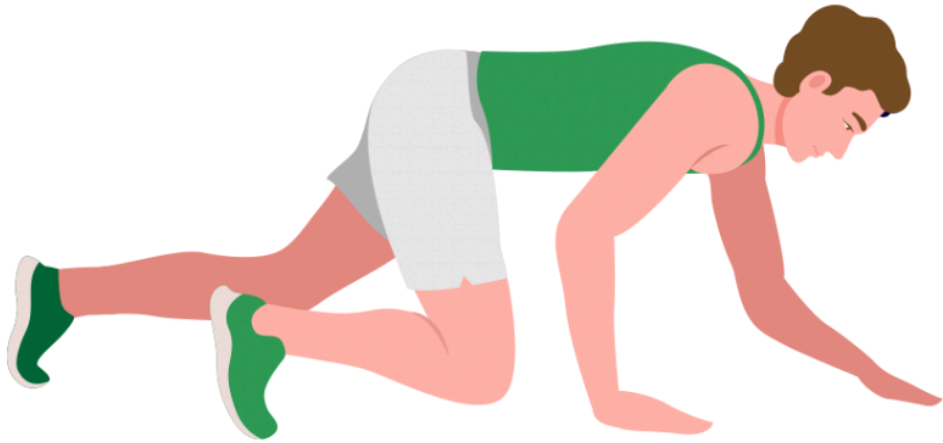
У немовлят повзання — важливий етап розвитку, що допомагає зміцнити м'язи спини, рук і ніг та координувати роботу обох півкуль мозку. У дорослих повзання використовується у військовій чи рятувальній підготовці, коли потрібно рухатися малопомітно або в умовах обмеженого простору.



(Image from: <https://www.frontiersin.org/journals/neurology/articles/10.3389/fneur.2021.731374/full>)

Повзання людини базується на чергуванні рухів рук і ніг у перехресному порядку. Тіло перебуває низько до землі, спирається на долоні та коліна (або на долоні й ступні). Спочатку вперед висунуться права рука та ліва нога, тоді вага тіла переноситься на них, а потім рухаються ліва рука й права нога. Такий цикл повторюється ритмічно.

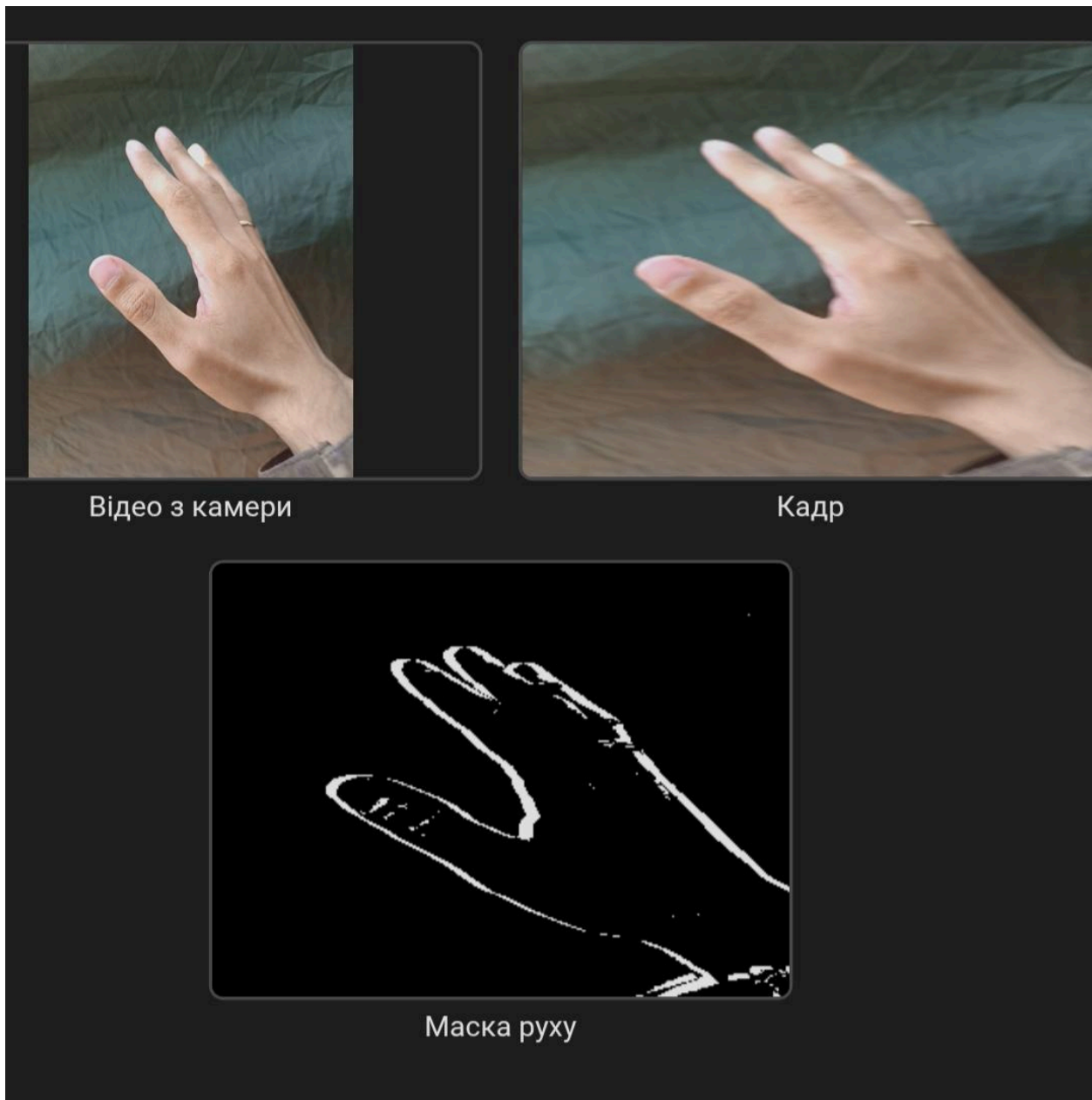
Завдяки перехресному руху кінцівок зберігається рівновага й плавність пересування. Цей алгоритм лежить в основі як дитячого повзання, що розвиває координацію та силу, так і спеціальних технік у дорослих, наприклад у військовій чи рятувальній підготовці для руху в умовах обмеженого простору.



Детекції руху на відео

Алгоритм виявлення рухомих об'єктів на відео базується на порівнянні послідовних кадрів. Кожен кадр представляється як матриця пікселів, і якщо відняти попередній кадр від поточного, то нерухомий фон зникає, а зміни, викликані рухомими об'єктами, залишаються. Щоб виділити рух, різницю кадрів можна порогово обробити, перетворивши

пікселі з великою зміною яскравості на білий, а всі інші — на чорний, отримуючи маску руху.



Для підвищення стабільності алгоритму часто використовують фонове усереднення: фон оновлюється поступово як середнє попередніх кадрів. Різниця між поточним кадром і фоном дозволяє ігнорувати дрібні коливання або шум, а морфологічні операції допомагають об'єднати фрагменти руху та видалити артефакти. Після цього можна визначати контури рухомих об'єктів або будувати bounding boxes для їхнього відстеження.

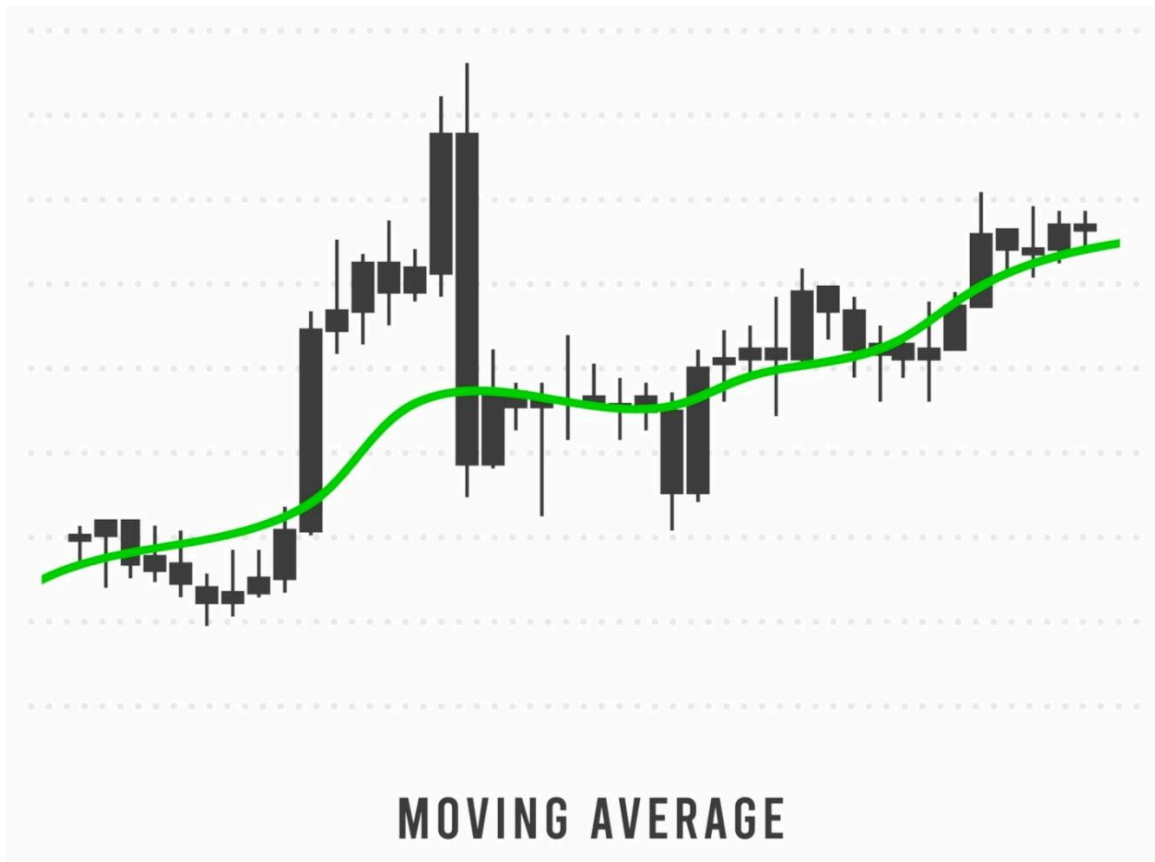
У масці руху завжди будуть “шуми” (білі точки навіть там, де немає справжнього руху) — це дрібні коливання світла, автофокус, компресія камери.

Є кілька простих прийомів, щоб покращити результат:

- ◆ 1. Усереднення (згладжування кадрів)

Замість порівняння зразу з попереднім кадром, можна брати усереднене зображення за кілька кадрів → випадкові шуми зникають.

Простий варіант — робити “ковзне середнє” (running average, moving average).



Ковзне середнє — це статистичний показник, який обчислюється шляхом усереднення значень певної кількості попередніх значень. Для обчислення ковзного середнього, ви берете середнє арифметичне значень певної кількості останніх спостережень. Потім з кожним новим спостереженням ви оновлюєте це середнє значення, викидаючи найстаріше спостереження і додаючи нове.

- ◆ 2. Збільшення порогу

Звичайно, можна ще трохи підняти threshold (наприклад, 50–60) — дрібні зміни не вважатимуться рухом.

Принцип дуже схожий на MTI-радар: там попередній сигнал зберігається в ртутній лінії затримки, і віднімання нового сигналу від старого дозволяє виділити тільки рухомі об'єкти, ігноруючи нерухомий фон. У відео аналогічно виділяється зміна пікселів: статичний фон зникає, а рух помітний, що дозволяє відстежувати об'єкти та аналізувати їхній рух.

1. Що таке ртутна лінія затримки

У старих радарах (50-60-ті роки) для обробки сигналу застосовували лінії затримки — спеціальні пристрої, які фізично затримували електричний імпульс на певний час. Одним з поширених варіантів були ртутні акустичні лінії затримки:

- електричний сигнал перетворювався на ультразвукову хвилю,
- ця хвиля поширювалася через довгий стовп ртуті зі сталою швидкістю (~1450 м/с),
- на виході знову перетворювалася в електричний сигнал.

Таким чином можна було "зберегти" сигнал на мікросекунди чи мілісекунди й використати його для порівняння з новим сигналом.

2. Як це допомагало відсіяти нерухомі об'єкти

Принцип був схожий на те, що відбувається з камерою:

- сигнал поточного імпульсу порівнювали з сигналом попереднього (збереженого в лінії затримки),
- якщо об'єкт був нерухомий (наприклад, будівля, гора), відбиті сигнали майже не змінювалися → віднімання давало нуль або дуже малий результат,
- якщо об'єкт рухався (літак, корабель) → відбиття трохи зсувалося у фазі та частоті → при відніманні залишався різницевий сигнал, і його можна було виділити.

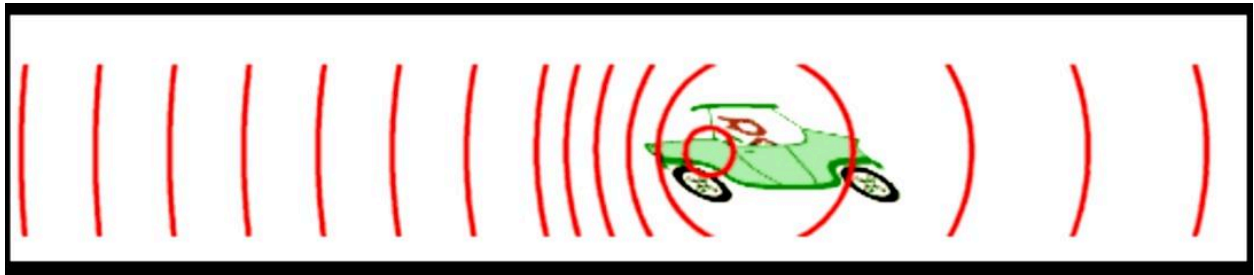
Це називалося MTI – Moving Target Indication (індикатор рухомих цілей).

3. Чи враховувався тут ефект Доплера

Так, ефект Доплера був ключовим фізичним явищем.

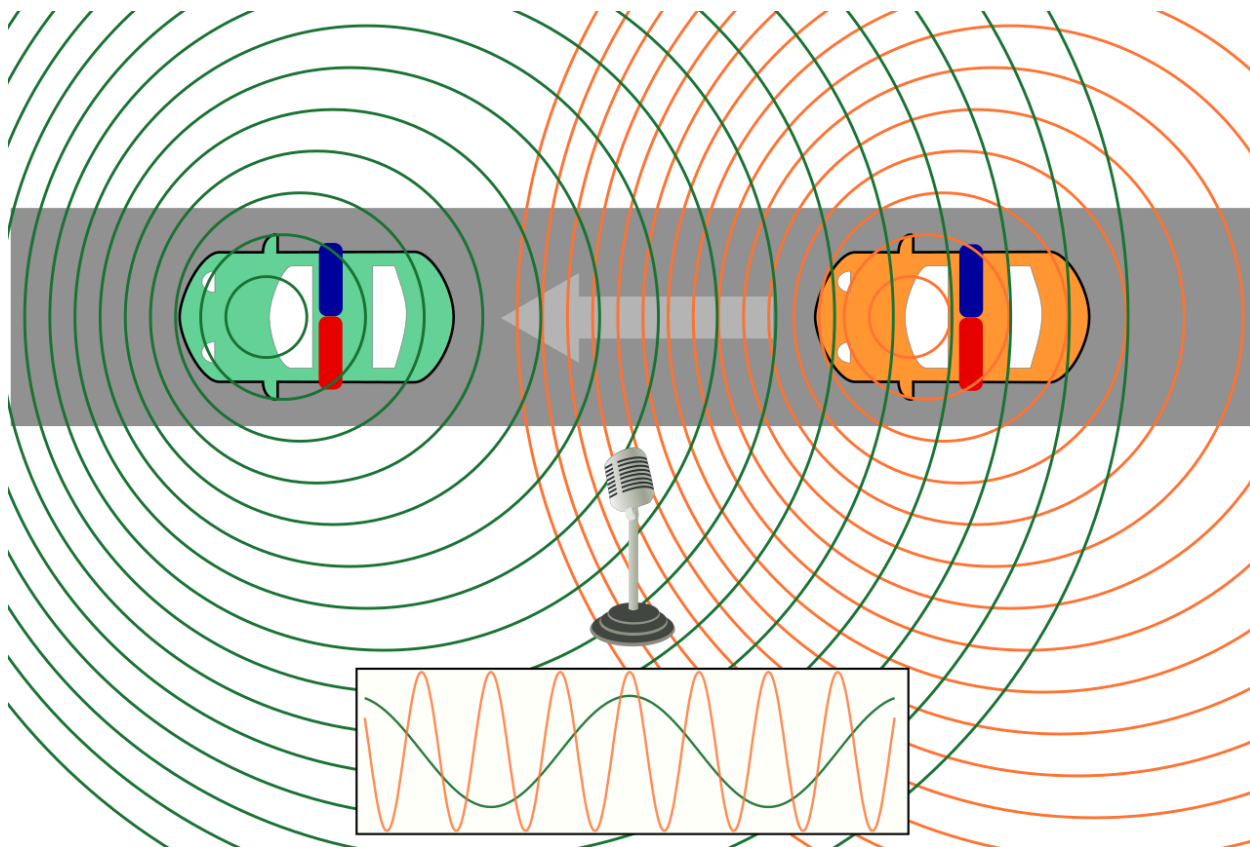
- Рухомий об'єкт зсуває частоту відбитого сигналу (вгору або вниз).
- Коли сигнал порівнювали з попереднім у лінії затримки, цей частотний зсув призводив до того, що віднімання не занулялося повністю, а залишався корисний "доплерівський" сигнал.
- Саме завдяки цьому можна було розрізнити, хто рухається, а хто ні.

Ефект Доплера — явище зміни частоти хвилі, яку реєструє приймач, викликане переміщенням джерела або приймача.



(Charly Whisky 18:20, 27 January 2007 • CC BY-SA 3.0)

Ефект Доплера використовується в радіолокації для розпізнавання рухомих об'єктів, наприклад, літаків, на фоні нерухомих (гір, хмар). За червоним зміщенням світла від астрономічних об'єктів вимірюється їхня швидкість і розраховується відстань до них. Ефект Доплера широко використовується в медицині. На базі ефекту створені комп'ютерні комплекси ультразвукової доплерографії. Зміна характеристик ультразвуку при проходженні через судини дозволяє визначати стан кровообігу як у поверхневих, так і у внутрішніх судинах.



(Doppler-Effekt am Beispiel zweier sich bewegender Polizeiwagen und eines ortsfesten Mikrophons, Original: Ekko Vektor: kismalac • CC BY-SA 3.0)

4. Аналогія з відео

Камера: віднімаємо кадр від попереднього → бачимо тільки рух.

Радар із ртутною лінією затримки: віднімаємо новий радіоімпульс від попереднього → бачимо тільки те, що змінилося (об'єкти з доплерівським зсувом, тобто рухомі).

Оптическое распознавание текста (OCR)

OCR (англ. optical character recognition - оптическое распознавание текста) на практиці часто реалізується через кілька послідовних кроків обробки зображення. Спочатку застосовують Gaussian Blur — розмиття зображення гаусовим фільтром, щоб зменшити шум і зробити контури символів більш чіткими. Це дозволяє наступним алгоритмам працювати стабільніше, не “спотикаючись” на маленьких артефактах сканування чи фотографії.

California State University

Hello

San Bernardino

World

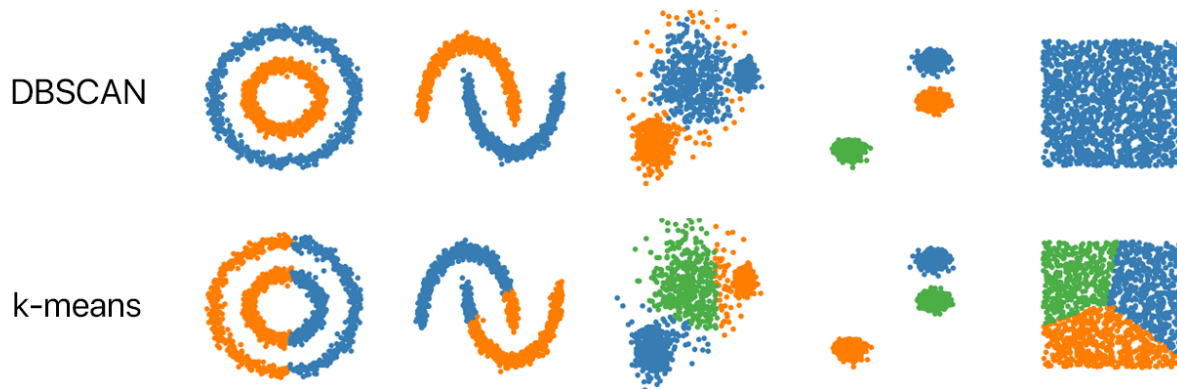
Department of Computer Science

Original Image



Після цього іноді застосовують Edge Detection, тобто виділення країв. Це допомагає знайти контури букв на зображенні, особливо якщо фон і текст близькі за кольором або контраст невисокий. Алгоритми на кшталт Canny чи Sobel дозволяють отримати чіткі межі символів, які потім легко аналізувати.

Далі йде сегментація — розподіл зображення на окремі символи або рядки. Тут можуть використовувати методи кластеризації, наприклад DBSCAN, який групує пікселі, що близько розташовані, або просту сегментацію по рядкам, яка виділяє горизонтальні смуги тексту. Мета цього етапу — отримати кожен символ або слово окремо для подальшого розпізнавання.



Останній крок — порівняння символів із шаблонами. Кожен сегментований символ порівнюють із базою готових зображень букв і цифр. Вибирають той шаблон, який найбільше схожий на знайдений символ. Це дозволяє ідентифікувати текст навіть без складних нейронних мереж, особливо якщо шрифт стандартний і символи не деформовані.

У цілому, процес виглядає як: розмиття зображення для зменшення шуму → виділення контурів → сегментація символів або рядків → порівняння кожного символу з шаблоном і визначення тексту. Ця послідовність дає стабільний результат на документних зображеннях, рахунках, номерах або будь-яких чітких шрифтах.

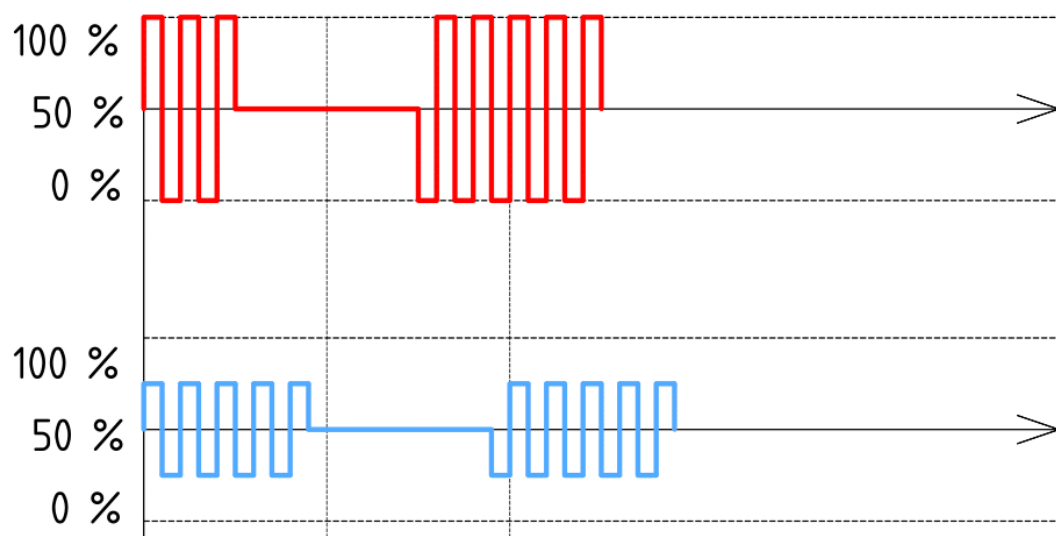
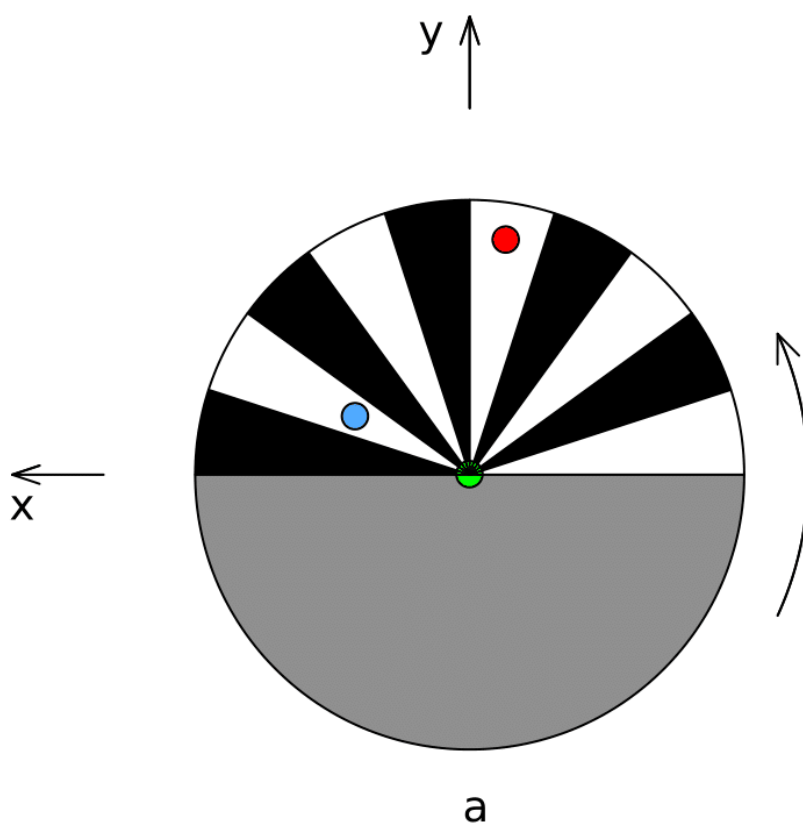
У класичному OCR із шаблонами порівняння символів зазвичай виконується у відсотках схожості: для кожного сегментованого символу його зображення порівнюють з усіма

шаблонами в базі, обчислюючи метрику схожості (наприклад, кореляцію пікселів або суму квадратів різниць), і результат нормалізується до 0–100%, де 100% означає ідеальну відповідність; символ визнається розпізнаним як той шаблон, з яким схожість максимальна, і часто використовується поріг (наприклад, 80%), нижче якого символ вважається невизначеним.

Військові алгоритми ППО - Reticle-seeker

Reticle-seeker

Механізм reticle-seeker (обертювий візир) дозволяє переводити ІЧ-випромінювання від цілі в сигнал з певною сигнатурою. Замість того, щоб робити відеофіксацію цілі та аналізувати зображення, ви просто фокусуєте сигнал і пропускаєте його через reticle-seeker, отримуючи сигнатуру, по якій точно можна дізнатися місце знаходження цілі в вашому полі зору.



У найпростіших системах з візирним пошуковим пристроєм інфрачервона енергія, випромінювана ціллю, збирається оптичною системою та фокусується на детекторі через обертовий візир (reticle-seeker).

Обертовий візир (reticle-seeker) — це оптичний модулятор, що складається з круглого елемента, на якому розташовані послідовно прозорі та непрозорі сектори. Візир розділяє промінь. Вихідний сигнал детектора — це послідовність імпульсів, амплітуда яких пропорційна величині інфрачервоної енергії, а частота дорівнює добутку швидкості обертання візира на кількість пар прозорих/непрозорих спиць. По частоті (або амплітуді) та по зміщенню фази можна виявити де ваша ціль, в центрі поля зору, праворуч чи ліворуч.

Крім того, сучасні тепловізійні ракети з обертовим візиром, використовують фазу сигналу детектора відносно еталонної фази для визначення кута положення осі пошукового пристрою щодо цілі. Кутова похибка потім використовується для того, щоб ось пошукового пристрою залишалася спрямованою на ціль в межах поля зору, а також для коригування траєкторії польоту ракети з метою перехоплення цілі.

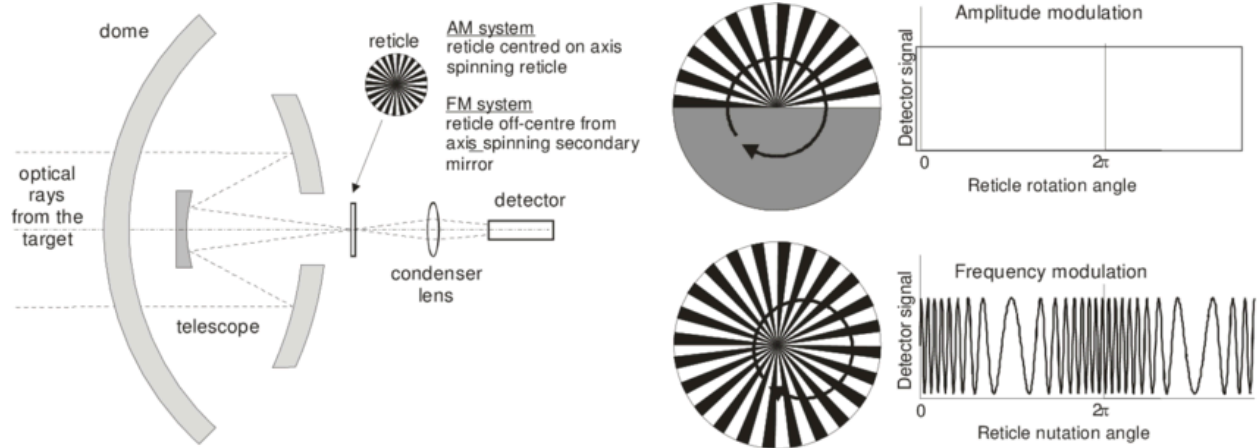
AIM-9 Sidewinder — це одна з найпоширеніших ракет класу "повітря-повітря", яка була розроблена в США. Вона використовує інфрачервоний візир для наведення на теплове випромінювання цілі. Ракета має система самонаведення і дуже високу маневровість, що дозволяє їй вражати повітряні цілі на близьких та середніх дистанціях. AIM-9 активно використовує теплові сигнали двигунів літаків для захоплення і знищення цілі, навіть при високих маневрах. Це одна з найстаріших та найвідоміших ракет в історії, що використовувалась з 1950-х років і по сьогоднішній день, з численними модернізаціями.

Це щоб не використовувати методи комп'ютерного зору. Якщо ж використовувати камери з ПЗЗ-матрицями, то з допомогою фільтра Гаусса та кластеризації типу k-means, або DBSCAN, можна виявляти цілі на зображенні. Це якщо не використовувати нейромережі. Бо то все накручує обчислювальні потужності.

До речі, DBSCAN описаний в моїй книзі "The Essence of Algorithms" by Dmitry Popov. А фільтр Гаусса описаний в моєму конспекті під назвою "Комп'ютерна наука в питаннях та відповідях".

В системах типу Петріот, в ракетах, є високоточні радары. Це вже не комп'ютерний зір та не виявлення по ІЧ-випромінювання, це радіолокація в ракетах. Високоточні радары, які можуть виявляти цілі, що летять зі швидкістю 5-10 махів, дуже дорогі, бо вимагають врахування ефекту Доплера, трансформацію Фур'є, не рахуючи самих фазових решіток і поправки на рух самої ракети.

Начинка високоточними міні радарыми робить ракети дуже дорогими.



(Image from: <https://www.researchgate.net/publication/46063482>)

Sources:

“THE IR MISSILE (SPIN-SCAN AND CON-SCAN SEEKERS) COUNTERMEASURES”, Ting Li Chang, 1994.

“Computer Vision: Algorithms and Applications 2nd Edition”, Richard Szeliski, 2022.

“Algorithms for Image Processing and Computer Vision 2nd Edition”, J.R. Parker, 2011.

“Computational Geometry: Algorithms and Applications 3rd edition”, Marc van Kreveld, Mark Overmars, and Mark de Berg, 2008.

“Deep Learning”, Aaron Courville, Ian Goodfellow, and Yoshua Bengio, 2015.