

Кондратьев Игорь Иванович

студент

Національний технічний університет України

«Київський Політехнічний Інститут»

Кондратьев Игорь Иванович

студент

Национальный технический университет Украины

«Киевский Политехнический Институт»

Kondratiev I.

student

National Technical University of Ukraine

«Kyiv Polytechnic Institute»

**ВИКОРИСТАННЯ СТРУКТУРИ ДАНИХ R-TREE
В АЛГОРИТМІ ТРЕКІНГУ PREDATOR
ИСПОЛЬЗОВАНИЕ СТРУКТУРЫ ДАННЫХ R-TREE
В АЛГОРИТМЕ ТРЕКИНГА PREDATOR
R-TREE DATA STRUCTURE USAGE
IN PREDATOR TRACKING ALGORITHM**

Анотація. В роботі досліджується можливість використання структури даних R-Tree для оптимізації роботи алгоритму трекінгу «Predator».

Ключові слова: алгоритми трекінгу, онлайн трекінг.

Аннотация. В работе исследуется возможность использования структуры данных R-Tree для оптимизации работы алгоритма трекинга «Predator».

Ключевые слова: алгоритмы трекинга, онлайн трекинг.

Summary. This paper investigates R-Tree structure usage for «Predator» tracking algorithm optimizations.

Key words: tracking algorithms, on-line tracking.

Вступ

За останні двадцять років інформаційні технології стали невід'ємною частиною нашого життя. Технічні засоби, поява яких пов'язана зі стрімким розвитком науки і техніки, все глибше проникають у наше життя. Не останнє місце серед цих технічних засобів займають засоби відеоспостереження — різного роду пристрої відеозапису, як самостійні, так і у вигляді частин інших пристроїв. Зараз важко уявити собі людину без мобільного телефону з камерою, чи людину, яка ніколи не користувалась фотоапаратом чи відеокамерою.

Ріст популярності таких технічних засобів прямо пов'язаний з розвитком технологій на їх основі. Автоматичне розпізнавання облич, чи навіть посмішок, під час фотозйомки вже здається нам чимось звичним

і зовсім не викликає здивування. Фотокамери, що автоматично вибирають налаштування для зйомки і автоматично проводять обробку вже відзнятих фото для досягнення максимальної якості вже стали повсякденною нормою. На більшості перехрест'ях у кожному місті знаходяться камери відеоспостереження, які автоматично реєструють всі проїжджаючі автомобілі, при цьому автоматично розпізнаючи порушення правил дорожнього руху. Все це — лише декілька прикладів застосування в реальному житті результатів такої галузі сучасної науки як комп'ютерний зір.

З кожним днем, засоби відео зйомки та відеоспостереження знаходять нові застосування в нашому житті, що створює нові виклики для комп'ютерного зору. Задачі, що виникають, потребують нових підходів до свого вирішення, нових алгоритмів, мають все

більші вимоги до продуктивності. Однією з таких задач якраз і є задача супроводження об'єктів на відео, яка розглядається в даній роботі.

Метою даної роботи є дослідження можливості використання структури даних R-Tree для оптимізації алгоритму стеження «Predator».

1. Задача трекінгу об'єктів на відео

В загальному вигляді, постановка задачі супроводження об'єктів на відео формулюється наступним чином:

- Задана послідовність кадрів $\{I_t(x, y)\}_{t=1..T}$.
- Задано початкове положення цільового об'єкта.
- Потрібно визначити положення об'єкта на наступних кадрах.

Зазвичай використовують такі способи для позначення об'єкта на зображенні:

- точка — центр об'єкта;
- контури об'єкта;
- обмежуючий прямокутник або еліпс.

В даній роботі використовується третій спосіб, а саме визначення положення об'єкта за допомогою обмежуючого прямокутника.

Визначення об'єкта з допомогою обмежуючого прямокутника має ряд переваг над іншими способами:

- На відміну від центральної точки, дозволяє визначити розміри об'єкта.
- На відміну від контурного визначення не потребує додаткових обчислювальних витрат на точний розрахунок обмежуючого контуру.
- Контурне визначення може бути ускладненим внаслідок нечіткості зображень, що отримуються під час швидкої зміни положення об'єкта у кадрі.
- Дозволяє просто оцінювати якість роботи алгоритму на розміченій вибірці з використанням метрики, що базується на площі перетину прямокутників.

З врахуванням даних викладок, можна переформулювати постановку задачі наступним чином:

- Задана послідовність кадрів $\{I_t(x, y)\}_{t=1..T}$.
- Задано обмежуючий прямокутник, що визначає положення об'єкта на першому кадрі послідовності.
- Потрібно визначити обмежуючий прямокутник положення об'єкта на наступних кадрах.

До основних і найвідоміших алгоритмів трекінгу можна віднести такі:

- Median Flow Tracker.
- MeanShift Tracker.
- CamShift Tracker.
- Flock of Features Tracker.
- Predator.

2. Причини для використання структури R-Tree

Як відомо, алгоритм стеження за об'єктами на відео «Predator» базується на підході TLD — Tracking Learning Detection[1]. Найбільшу цікавість представляє етап detection — пошук цільового об'єкта у кадрі. На даному етапі алгоритм застосовує натренований детектор до частин зображення, в результаті чого приймається рішення про знаходження об'єкта в тій чи іншій частині зображення, або ж про його відсутність. Цей процес аналогічний до алгоритму ковзного вікна (sliding window), якщо врахувати, що множина вікон, які перебирає алгоритм зарані визначена. Генерація вікон відбувається на першій ітерації алгоритму після того, як стають відомі розміри зображення та розміри цільового об'єкта. Отримана множина вікон являє собою можливі положення об'єкта в кадрі з урахуванням можливої зміни розмірів в процесі роботи — для цього використовується масштабний коефіцієнт.

Зазвичай, на практиці, цільовий об'єкт займає невелику частину кадру, проте на кожній ітерації детектор перевіряє всю розраховану сітку вікон — цей факт можна використати для оптимізації роботи алгоритму. В залежності від конкретної задачі алгоритм може бути модифіковано розрахунками моделі переднього плану на зображеннях. В такому разі, для пошуку об'єкта використовується лише локалізована область переднього плану, проте навіть в такому випадку витрачається час на перевірку всіх вікон щодо належності їх до області переднього плану. Незважаючи на невисоку обчислювальну складність такої перевірки, кількість вікон є зазвичай дуже великою, що негативно впливає на швидкість роботи алгоритму. Для вирішення цієї проблеми пропонується використання структури R-Tree, яка дозволяє зберігати множину вікон для конкретного відеозапису та дозволяє швидко отримувати множину вікон, що в даний конкретний момент належать області переднього плану.

3. Використання структури R-Tree під час пошуку об'єкта на зображенні

R-Tree — структура даних у вигляді дерева, подібна до B-Tree, яка призначена для збереження просторових даних[2]. В даному випадку, дерево буде зберігати множину вікон, що відповідають потенційно можливим положенням об'єкта в кадрі. Таким чином, на першій ітерації алгоритму, після розрахунку положення всіх вікон, потрібно створити дерево і зберегти всі розраховані вікна в нього.

Як вже було сказано, для можливості використання дерева потрібно модифікувати алгоритм додавши до нього етап визначення переднього плану. Для цього можуть бути використані різні підходи, в залежності

від контексту конкретної задачі. В загальному вигляді можуть бути використані такі методи як:

- визначення переднього плану на основі розрахунку оптичного потоку,
- визначення переднього плану на основі обчислення різниці зображень.

Перевагою даних методів є простота та відносно невисока обчислювальна складність, що є критичним фактором для алгоритму стеження. Отож, на кожній ітерації перед початком роботи детектора потрібно розрахувати для кожної точки зображення приналежність її до переднього плану. Це можна зробити, наприклад, за допомогою наступного алгоритму:

1. Розрахувати оптичний потік для даного кадру за допомогою алгоритму Гунара Фарнебака [3]. В результаті, для кожної точки отримуємо вектор зміщення по відношенню до попереднього кадру.

$$OF^t(x, y) = OpticalFlow(I^t(x, y), I^{t-1}(x, y)).$$

2. Для визначення областей переднього плану потрібно розрахувати норму вектора в кожній точці. Для забезпечення високої швидкості роботи, розрахунок оптичного потоку та норм векторів краще виконувати на GPU з допомогою технологій CUDA або OpenCL.

$$OF_{norm}^t(x, y) = OF^t(x, y).$$

3. Варто зазначити, що у випадку статичної зйомки, замість розрахунку оптичного потоку можна використовувати попиксельну різницю між зображеннями — це дозволить отримати більшу швидкість розрахунків без значних втрат у якості роботи.

$$OF_{norm}^t(x, y) = |I^t(x, y) - I^{t-1}(x, y)|.$$

4. Після цього, варто провести фільтрацію отриманого зображення з допомогою порогового значення f_{min} , яке вибирається для кожної задачі окремо. Це дозволяє відкинути шуми, а також незначні зміни в фоновому зображенні.

$$FG_{mask}^t(x, y) = OF_{norm}^t(x, y) \text{ if } OF_{norm}^t(x, y) \geq f_{min} \text{ else } 0.$$

Після розрахунку маски для виділення області переднього плану на зображенні, виділені області переднього плану покриваються множиною прямокутників [4]. Позначимо її R_f . Тоді, вся область що належить до переднього плану визначається як

$$F = \bigcup_{r \in R_f} r.$$

Для визначення множини вікон, які слід перевіряти на наявність в них цільового об'єкта можна використати ступінь належності вікна до переднього плану як відношення площі вікна, що належить передньому плану до площі вікна

$$k_w = \frac{s(w \cap F)}{s(w)} = \frac{\sum_{r \in R_f} s(w \cap r)}{s(w)}.$$

Для перевірки слід залишити вікна, які мають ступінь належності більший за зарані заданий рівень k_{min} .

Оскільки R-Tree дозволяє швидко знайти прямокутники, що перетинаються з даним прямокутником, то для побудови фінальної множини варто зробити запити для кожного прямокутника з F , після чого відібрати лише ті, що мають достатній ступінь належності до переднього плану.

Використати даний підхід до обмеження множини пошуку цільового об'єкта можна і без використання R-Tree, можна просто перевіряти всі вікна на належність до області переднього плану і розглядати далі тільки ті, що пройшли перевірку. Проте, як буде видно в наступному розділі, за певних умов, використання дерева може дати суттєвий приріст швидкості роботи алгоритму.

4. Аналіз отриманих результатів

Для оцінки ефективності використання R-Tree можна порівняти швидкість роботи фільтрації вікон з використанням дерева та без нього в залежності від розмірів переднього плану. Для проведення експерименту були використані такі параметри: розміри зображення — 1280×720, кількість вікон — 401343, рівень $k_{min} = 0,6$. Результати наведено в наступній таблиці:

Таблиця 1

Порівняння отриманих результатів

$Q_{size}, \%$	N_w	$T_{tree}, \text{мс}$	$T_{notree}, \text{мс}$
25	13666	8	19
30	20179	10	21
35	27675	15	20
40	36972	17	21
45	47253	19	20
50	58569	22	21
55	71702	25	22
60	85552	30	21
65	100641	37	22

де Q_{size} — розмір прямокутника, що покриває передній план у відсотках відносно розмірів зображення;

N_w — кількість вікон, які потрапляють в область переднього плану;

T_{tree} — час виконання операції вибору вікон, що належать до переднього плану зі ступенем належності не менше k_{min} з допомогою структури R-Tree;

T_{notree} — час виконання тієї ж операції без використання дерева.

Аналізуючи отримані результати, можна зробити висновок, що швидкість роботи дерева напряму залежить від розмірів прямокутника, для якого виконується запит. Використання дерева має сенс

у тому випадку, коли розміри переднього плану не перевищують межу в 40% від розміру зображення, після якої час роботи дерева стає більшим за повний прохід всіх вікон. Це зумовлено тим, що зі збільшенням площі переднього плану збільшується кількість вікон, які потрапляють до нього, а отже, більше вершин дерева доводиться обійти для пошуку всіх варіантів.

Підсумовуючи, можна сказати, що практичний сенс має використання дерева як складової частини гібридного алгоритму — дерево використовується якщо розміри переднього плану не перевищують 40% від розмірів зображення, в інакшому випадку фільтрація вікон відбувається повним перебором.

Висновки

В даній роботі пропонується використання структури даних R-Tree на етапі детектування об'єктів в алгоритмі Predator. Визначено необхідні умови для використання даної структури, а також наведено можливий варіант модифікації алгоритму для застосування відповідної структури.

В ході дослідження було проведено порівняння використання структури R-Tree з повним перебором варіантів положення цільового об'єкта. Отримані результати свідчать, що застосування структури є доцільним у вигляді гібридного алгоритму, який використовує або дерево або повний прохід всіх вікон в залежності від структури та розмірів переднього плану.

Література

1. Kalal Z. Tracking-Learning-Detection / Z. Kalal, K. Mikolajczyk, J. Matas. // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2012. — № 7. — С. 1409–1422.
2. Guttman A. R-trees: a dynamic index structure for spatial searching / Antonin Guttman. // ACM SIGMOD Record. — 1984. — № 14. — С. 47–57.
3. Farneback G. Two-frame motion estimation based on polynomial expansion / Gunnar Farneback. // SCIA'03 Proceedings of the 13th Scandinavian conference on Image analysis. — 2003. — С. 363–370.
4. Chang F. A linear-time component-labeling algorithm using contour tracing technique / F. Chang, C. Chun-Jen, L. Chi-Jen. // Computer Vision and Image Understanding. — 2004. — № 93. — С. 206–220.