

*Драган Д.Д., Куц П.О., Петренко О.О. – рецензент Петренко А.І.  
ННК “Інститут прикладного системного аналізу” НТУУ “КПІ”, Київ, Україна*

## Алгоритми оцінки руху об'єктів при обробці відео

Задача оцінки руху (ME- motion Estimation) в відеопотоці, яка важлива для багатьох застосувань (наприклад, відслідкування автомобіля, що рухається, людини, що іде, та інш.) вирішується, зазвичай, через оцінку змін між сусідніми кадрами чи між базовим (опорним) кадром і поточним кадром. При стискуванні відео чи конвертації частоти кадрів кодер використовує цю модель руху, щоб переміщати контент опорного кадру для забезпечення кращого прогнозування поточного кадру [1].

Поточний відеокادر розбивається на блоки пікселів (скажімо, розміром 16x16 чи 4x4), і оцінка руху через вектор руху (MV) здійснюється незалежно для кожного блоку пікселів шляхом визначення блоку пікселів опорного кадру, який в найбільшій мірі співпадає з поточним блоком. Найбільш популярним алгоритмом оцінки руху є алгоритм співставлення блоків (Block Matching Algorithm - BMA), з допомогою якого обчислюється вектор руху для всього блоку пікселів, а не для окремих пікселів

Для зниження обчислювальних витрат оцінку руху можна відеоряд розділяти не на блоки пікселів, а безпосередньо розглядати рухомі об'єкти в кадрі. Виявлення рухомих об'єктів забезпечується розділенням пікселів відеоряду на пікселі переднього плану (рухомі об'єкти) та фон. Такий підхід, заснований на видаленні фону, іноді називають відніманням фону, яке полягає в усуненні фону шляхом порівняння кожного кадру з опорним або моделлю фону, при цьому пікселі, які суттєво відрізняються від фону, вважаються такими, що належать рухомих об'єктам [2].

Рухомий об'єкт можна відстежувати більш точно, обчислюючи суми абсолютної різниці (SAD)

$$\sum_{x=1}^{x=N} \sum_{y=1}^{y=N} |C(x, y) - R(x, y)|$$

між кадрами, що порівнюються, та встановлюючи порогові значення для фільтрації незначних відхилень. Якщо рухомий об'єкт існує в декількох сусідніх кадрах, необхідно оновлювати область супроводження об'єкту за допомогою сучасних модифікацій блочних алгоритмів оцінки руху (BMA), розглянутих раніше, а також коригувати розмір області супроводження. Модифікації BMA зводяться до вимоги розгляду і порівняння всіх блоків пікселів, на які розбиваються прямокутні області апроксимації конфігурацій рухомих об'єктів (Bounding boxes), і переходу до спеціальних шаблонів пошуку кандидатів для порівняння блоків [6].

В алгоритмі тріступеневого пошуку (TSS) на першій ітерації оцінюються дев'ять блоків - кандидатів [3]. Кандидати зосереджені навколо позиції опорного блоку. Розмір кроку для кожної ітерації зазвичай встановлюється рівним половині діапазону пошуку, що визначається розміром рухомого об'єкту. Під час наступної ітерації центр пошуку зміщується в бік найбільш підходящого кандидата із першої ітерації. Крім того, розмір кроку зменшується у два рази. Той же самий процес триває, доки крок стає рівним одному пікселю. Найбільш підходящий кандидат від останньої ітерації вибирається як остаточний кандидат. Вектор руху, відповідний цьому кандидатові, обирається для поточного. Алгоритм DS (Diamond Search) є іншим алгоритмом, який перевіряє обмежену кількість кандидатів. Він схожий на тріступеневий пошук. Під час першої ітерації перевіряються 5 кандидатів [4]. Кандидати зосереджені навколо поточного положення блоку у формі діаманту. Це перша модель, так звана велика схема пошуку (LDSP). Розмір кроку для першої ітерації встановлюється рівним половині діапазону пошуку. Для другої ітерації центр ромба зміщується в бік найбільш схожого кандидата. Розмір кроку зменшується у два рази, тільки якщо кращий кандидат виявиться в центрі діаманту. У цьому випадку використовується так звана мала схема пошуку (SDSP). Якщо найкращий кандидат не у центрі алмазу, тоді той самий розмір кроку використовується і

для другої ітерації. У цьому випадку деякі з кандидатів виявляються вже оціненими в ході першої ітерації. Отже, немає необхідності проводити відповідні розрахунки для цих кандидатів у другій ітерації. Результати першої ітерації можуть бути використані для цих кандидатів блоку. Процес триває, доки крок стане рівним одному пікселю. Для цієї ітерації всіх вісьмох навколишніх кандидатів вже оцінено. Найбільш підходящий кандидат цієї ітерації вибирається як відповідний поточному блоку. В алгоритмі DS число кандидатів, що оцінюються, є змінним. При цьому можуть бути визначені найгірший та найкращий кандидати.

Алгоритм ієрархічного співставлення блоків є більш складним методом оцінки руху. У цьому алгоритмі оцінка руху забезпечує постійну корекцію векторів руху шляхом послідовної переоцінки вектора руху при різних значеннях роздільної здатності. Процес оцінки руху починається при найнижчій роздільній здатності. Як правило, повний пошук оцінки руху виконується для кожного блоку при найнижчій роздільній здатності. Оскільки розмір блоку і діапазон пошуку скорочуються, процедура пошуку не вимагає великих обчислень. Вектори руху, отримані з низькою щільністю, масштабуються і передаються в якості кандидатів векторів руху для кожного блоку на наступний рівень. На наступному рівні вектори руху уточнюються при використанні меншої області пошуку. Простий алгоритм оцінки руху і невеликий діапазон пошуку є достатніми для оцінки векторів руху при роздільних здатностях, близьких до максимальної, тому що на цих роздільних здатностях значення векторів руху близьки до точних [5].

В доповіді наводяться результати моделювання розглянутих підходів до оцінки руху з допомогою програмної бібліотеки OpenCV, призначеної для розв'язання задач комп'ютерного зору [6].

**Література.** 1. Milind Phadtare. "Motion estimation techniques in video processing", Electronic Engineering Times India, August 2007, pp.1-4. 2. S. Immanuel Alex Pandian, G. Josemin Bala, Becky Alma George. "A Study on Block Matching Algorithms for Motion Estimation", International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 1 Jan 2011, pp.34-44. 3. T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," Proc. NTC81, pp. C9.6.1-9.6.5, New Orleans, LA. Nov. 1981. 4. S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," in Proc. Int. Conf. Inf. Commun. Signal Process. (ICICS '97), vol. 1, Sep. 9-12, 1997, pp.292-296. 5. L. W. Lee, J. F. Wang, J. Y. Lee, and J. D. Shie, "Dynamic search-window adjustment and interlaced search for block matching algorithm," IEEE Trans. Circuits Syst. Video Technol., vol. 3, no. 1, Feb. 1993. 6. OpenCV library, <http://sourceforge.net/projects/opencvlibrary>