

УДК 004.93.12

Пантелєєв Олександр,
Пилипенко Павло,
Олійник Володимир

СИСТЕМА ВІЗУАЛЬНОГО МУЛЬТІТРЕКІНГА В РЕАЛЬНОМУ ЧАСІ REALTIMEVISUALMULTITRACKINGSYSTEMY

В статті розглянута проблема візуального мультітрекінга. Запропонована реалізація системи мультітрекінга на базі фреймворка ROS. В системі використовується модифікація підходу трекінга на основі детектування, у якій для побудови траєкторії цілей на проміжних кадрах використовується незалежні трекари на основі кореляційних фільтрів.

Ключові слова: Комп'ютерний зір, візуальний трекінг, кореляційний фільтр, мультітрекінг, трекінг на основі детектування.

Рис.: 2. Бібл.: 6.

The article proposes implementation of the multittracking system based on the ROS framework. It is suggested to use a modification of method of tracking by detection, in which trackers based on correlation filters are used to construct the target trajectory between the key frames.

Keywords: Computer vision, visual object tracking, correlation filter, multittracking, tracking by detection.

Fig.: 2. Bibl.: 6.

Актуальність теми дослідження. Задача відстеження множини об'єктів на відео в реальному часі є невід'ємною частиною багатьох прикладних областей комп'ютерного зору, таких як системи відеоспостереження, аналіз дорожнього трафіку. Особливий інтерес дана задача представляє для автономних роботизованих агентів, здатних активно взаємодіяти з навколишнім світом, зокрема для аналізу оточення безпілотним літальним апаратом, автопілотом автомобіля, промисловим або домашнім роботом.

Постановка проблеми. Задачу мультітрекінга можна сформулювати наступним чином: Необхідно виявити на відео об'єкти, що належать до певних класів, та побудувати для кожного виявленого об'єкта його траєкторію на кадрі. Основною проблемою існуючих систем, які працюють в режимі реального часу, є низька точність та надійність відстеження об'єктів, що рухаються нелінійно.

Аналіз останніх досліджень і публікацій. Найбільш поширеним методом вирішення задачі мультітрекінга на даний момент є трекінг на основі детектування, в якому для побудови траєкторії об'єкта використовується динамічна модель та фільтр Калмана[1, 2]. Такий підхід використовується через високу швидкість розрахунків. Проте через обмеження фільтра Калмана та фактичну відмову від використання візуальних ознак такий трекаер погано справляється з об'єктами, які рухаються нелінійно.

Виділення недосліджених частин загальної проблеми. Дана стаття присвячена дослідженню можливостей використання кореляційних трекаерів для вирішення задачі мультітрекінгу. Раніше використання кореляційних

трекерів у системах реального часу було неможливим, через велику кількість обчислень, проте нещодавно було запропоновано метод [3], який дозволяє значно збільшити швидкість роботи кореляційного трекера (300FPS для однієї цілі). Така швидкість дозволяє використовувати цей підхід для відстеження множини об'єктів.

Постановка завдання. Задачею даної роботи є розробка гібридної системи візуального мультітрекінга, яка поєднує в собі метод відстеження на основі детектування, та метод відстеження із допомогою кореляційних фільтрів. Розроблена система повинна в режимі реального часу виявляти цілі, та відстежувати траєкторію об'єктів, що рухаються нелінійно з більшою точністю, ніж у існуючих систем.

Викладення основного матеріалу. Для вирішення задачі мультітрекінга пропонується використовувати модифікацію підходу трекінгу на основі детектування, в якій для побудови траєкторії кожного об'єкта між ключовими кадрами використовуються незалежні трекери на основі кореляційних фільтрів. Після виявлення об'єктів детектором, для кожної цілі створюється кореляційний трекер, якій відстежує її до наступного повідомлення від детектора. Після отримання наступної множини об'єктів, список цілей оновлюється.

Для реалізації системи використовується фреймворк ROS [4], який дозволяє розділити систему на окремі вузли (Node). Кожен вузол являє собою незалежний програмний процес. Модулі обмінюються інформацією шляхом публікації повідомлень у відповідні топіки. Кожен вузол, який підписався на топик отримує копію повідомлення, яке відправив інший вузол у цей топик. Обмін даними здійснюється через TCPсокет, таким чином вузли можуть бути фізично розміщені на різних комп'ютерах. Запропонована система має три вузла: камера, детектор, трекер (рис. 1).

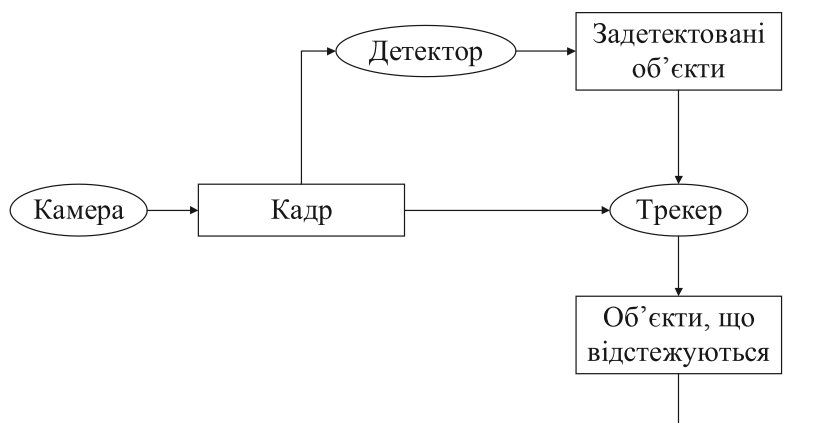


Рис. 1. Схема системи мультітрекінга.

Камера. Цей вузол відповідає за публікацію відеоряду. Як джерело даних може використовуватися веб-камера, відеофайл або директорія з набором кадрів. Кожному кадру присвоюється мітка часу, яка використовується для синхронізації.

Детектор відповідає за виявлення об'єктів на відео. При отриманні нового кадру, на ньому виявляються об'єкти за допомоги нейронної мережі SSD (Single Shot Detector) [5]. Для підвищення швидкості роботи, усі кадри, що надходять на детектор під час обробки поточного зображення, відкидаються.

Трекер. Трекер відповідає за відстеження об'єктів на проміжних кадрах, для цього для кожного об'єкта створюється незалежний кореляційний трекер, який визначає позицію цілі на кадрі. Цей вузол отримує як кадри відео, так і масиви об'єктів, що виявив детектор.

При отриманні нового кадру, оновлюються позиції усіх цілей на цьому кадрі із допомогою їх кореляційних трекерів.

При отриманні набору виявлених об'єктів від детектора, відбувається оновлення списку цілей. Для того щоб правильно розрахувати візуальні ознаки для кожного об'єкту повідомлення зі списком виявлених об'єктів синхронізується з кадром з камери, а потім поточної набір об'єктів зіставляється з об'єктами, які виявив детектор. Зіставлення відбувається за допомогою угорського алгоритму, критерієм подібності об'єктів виступає відношення площин обмежувальних рамок до їх об'єднання (IntersectionOverUnion). За результатами зіставлення можливо три випадки:

- Якщо два об'єкти зіставлені, трекер коригує свою модель, використовуючи позицію і візуальні ознаки відповідного об'єкта, виявленого детектором.
- Якщо об'єкт що відслідковується не був зіставлений з жодним об'єктом з детектора, він вважається втраченим і його відслідковування припиняється.
- Якщо виявлений детектором об'єкт не був зіставлений, вважається, що в кадрі з'явився, новий об'єкт. Для нього створюється новий трекер з новим унікальним ідентифікатором.

Синхронізація. Оскільки вузли системи є незалежними, і повідомлення надходять у довільний момент часу, у вузлі трекера треба здійснювати синхронізацію кадрів та масивів об'єктів які виявив детектор.

Для цього створюються три буфера, в перший додаються повідомлення які містять кадр, у другий – повідомлення, які містять масив об'єктів, що були знайдені на кадрі. Після цього повідомлення, які мають однакову мітку часу, об'єднуються у кортеж і додаються до третього буфера, який у подальшому використовує трекер (рис. 2).

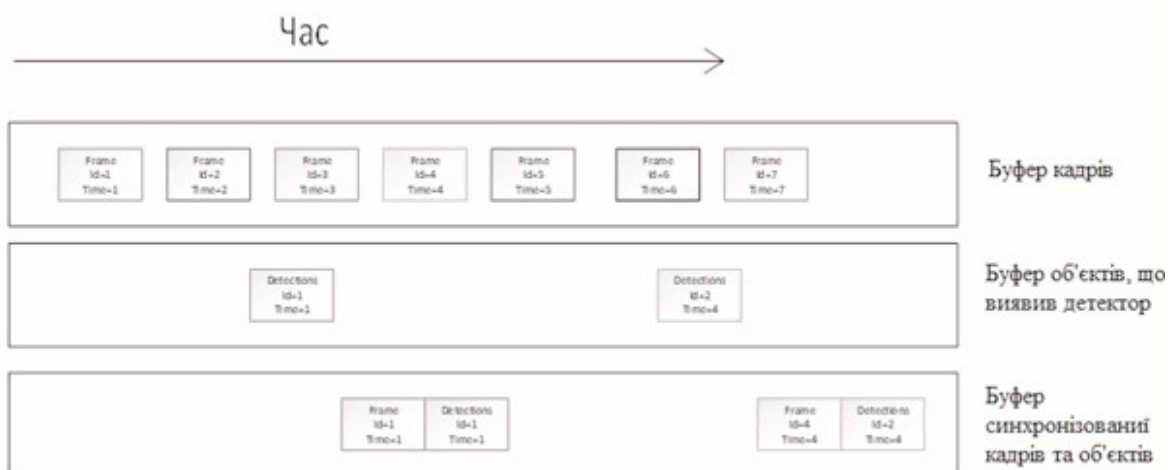


Рис. 2. Схема буферів для синхронізації повідомлень

Експерименти. Для оцінки результатів роботи системи використовується бенчмарк MOT[6]. Датасет містить 11 відео. Відео зняті в різноманітних умовах

(різна освітленість, кількість об'єктів, швидкість та характер руху камери тощо), для того щоб протестувати роботу системи у різних умовах та уникнути пристосування до якогось певного сценарію. Середня швидкість роботи трекара склала 73 FPS, а середня точність – 83%.

Висновки. Запропонована в статті система відстеження множини об'єктів в реальному часі, дозволяє підвищити точність і надійність трекінга об'єктів, що рухаються нелінійно. Це досягається за рахунок використання кореляційних фільтрів візуальних ознак для відстеження цілі на проміжних кадрах. Система мультитрекінга була реалізована на базі ROS, що дозволило зробити компоненти системи незалежними, та здійснювати розподілену обробку даних, завдяки чому підвищується швидкість та надійність системи.

Список використаних джерел

1. Bochinski E., Eiselein V., Sikora T. High-Speed tracking-by-detection without using image information // IEEE International Conference on Advanced Video and Signal Based Surveillance. 2017. С. 1-6.
2. Luo W. Multiple Object Tracking: A Literature Review. [Електронний ресурс] // arxiv.org URL: <https://arxiv.org/pdf/1409.7618.pdf> Дата звернення: 03.04.2018
3. Henriques J., Caseiro R. High-speed tracking with kernelized correlation filters // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015. С.583–596.
4. Morgan Q. ROS: an open-source Robot Operating System // ICRA Workshop on Open Source Software. 2009.
5. Liu W. SSD: Single shot multibox detector // European Conference on Computer Vision. Springer. 2016. С.21–37.
6. Milan A. MOT16: A benchmark for multi-object tracking [Електронний ресурс] // arxiv.org URL: <https://arxiv.org/pdf/1603.00831.pdf> Дата звернення: 29.04.2018

ДОВІДКА ПРО АВТОРІВ

Пантелєєв Олександр Сергійович – студент, кафедра технічної кібернетики, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Pantelieiev Oleksandr –student, Department of Technical Cybernetics, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: spn220v@gmail.com

Пилипенко Павло Геннадійович – студент, кафедра технічної кібернетики, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Pylypenko Pavlo –student, Department of Technical Cybernetics, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: pylypenko.pg@gmail.com

Олійник Володимир Валентинович – к.т.н., старший викладач, кафедра технічної кібернетики, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Oliinyk Volodymyr – Ph.D., associate professor, Department of Technical Cybernetics, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: oliinyk.volodymyr@gmail.com

**Panteliev Oleksandr,
Pylypenko Pavlo,
Oliinyk Volodymyr**

REALTIMEVISUALMULTITRACKINGSYSTEMY

Target setting. The problem of visual multiple object tracking is an integral part of many areas of computer vision, such as video surveillance systems, traffic analysis. This task is important for autonomous robotic agents, capable of active interaction with the outside world, in particular for the analysis of the environment by an unmanned aerial drone, autopilot of an autonomous car, an industrial or home robot.

Formulation of the problem. The problem of multitasking can be formulated as follows: On the video sequence, system must detect objects belonging to certain classes, and construct for each detected object its trajectory on a frame.

The main problem of existing systems operating in real time is the low accuracy and reliability of tracking objects that move nonlinearly.

Actual scientific researches and issues analysis. Currently the most common method of solving the multitasking problem is tracking based on detection, with a dynamical model and a Kalman filter. This approach is used because of the high speed of calculations. However, due to the limitation of the Kalman filter, such tracker does not handle objects that move nonlinearly.

Definition of uninvestigated parts of the problem. This article studies possibilities of using correlation trackers for solving the multitasking problem.

The research objective. The objective is to create the system that can automatically detect and track objects of particular classes on video.

Main results. Improved method of multitasker that uses correlational tracker was presented. Distributed system for visual multitasking was developed using ROS framework. Received results were accurate enough and the speed was sufficient for real-time applications.

Conclusions. The proposed multitasking method allows to increase the accuracy and reliability of tracking of the objects moving nonlinearly. This is achieved by using the correlation filter to track the target between keyframes. The multitasking system based on this method system was implemented using ROS framework, which allows independent components of the system and implements distributed data processing that increase speed and reliability of system.

Key words: Computer vision, visual object tracking, correlation filter, multitasking, tracking by detection.