

# Аналіз швидкодії алгоритму обробки потокового відео в залежності від способу реалізації

Сорокопуд Владислав Ігорович

науковий керівник: Приставка П.О., д.т.н., проф.  
ННІ ІДС, Національний авіаційний університет  
Київ, Україна  
e-mail: vlad.sorokopud.i@gmail.com

Чирков Артем Валерійович

науковий керівник: Приставка П.О., д.т.н., проф.  
ННІ ІДС, Національний авіаційний університет  
Київ, Україна  
e-mail: a.chyrkov@nau.edu.ua

**Анотація** — порівняння способів підвищення швидкості роботи алгоритму на мові Python: вибір інтерпретатора, конвертація коду на мову C++.

**Ключові слова** — *python, c++, інтерпретатор, конвертація коду.*

Під час виконання наукових досліджень основною вимогою до мови програмування є вимога швидкості написання програмного коду, зокрема швидкість реалізації рутинних операцій. Прийнятним варіантом у випадку досліджень у сфері автоматизованої обробки потокового відео є мова програмування Python [1]. Недоліком є низька швидкість роботи на процесорах архітектури ARM, зокрема на Raspberry Pi 3, неприйнятна для використання алгоритму в реальному часі. Отже, виникає задача пошуку оптимального способу виконання алгоритму.

Особливістю алгоритму, задачу підвищення швидкості якого вирішували автори, є наявність залежностей від сторонніх бібліотек: OpenCV, NumPy.

Розроблений алгоритм було протестовано на цільовому пристрої. Результат – робота алгоритму на відео розмірністю 1920x1080 дала швидкість в середньому 1 fps (кадр/с) при середньоквадратичному відхиленні  $\sigma$  0,12 fps.

Такий результат не є прийнятним для обробки відео в режимі реального часу, тому була поставлена задача підвищити швидкість роботи програми декількома шляхами: 1) переведення програми з мови Python на C++, 2) підвищення швидкості в рамках обраної мови шляхом використання різних інтерпретаторів мови Python.

Перш за все було досліджено можливість *автоматичної трансляції алгоритму* з мови Python на C++. Було знайдено декілька застосунків, що підходять, а саме – Nuitka та PyInstaller.

Nuitka – програма для конвертації коду шляхом трансляції в мову C++ та компілювання програми. Однак вона не працює з бібліотекою OpenCV, тому її використання не є можливим.

PyInstaller – ще одна програма для конвертації, яка компілює python-код. З її допомогою ми вдало провели конвертування, однак швидкість становила в середньому 0,88 fps ( $\sigma=0,03$  fps), тобто приросту швидкості немає.

Іншим варіантом підвищення швидкості коду на Python є *використання специфічних інтерпретаторів*. Особливістю мови Python є той факт, що в залежності від типу інтерпретатора, який використовується, можна отримувати різні результати: інтерпретатор може запустити програму, скомпілювати її або провести JIT-компіляцію.

З великого списку різноманітних інтерпретаторів [2] був обраний PyPy, специфіка якого заключається в так званій гібридній інтерпретації: програма запускається на віртуальній машині, однак функції, які часто використовуються, компілюються і потім визивається уже скомпільований варіант. На жаль, така технологія на даний момент не працює з бібліотекою OpenCV, тому в даній задачі ми не можемо її використовувати.

Також було розглянуто можливість *конвертування коду на C++ вручну*. Тестування дало швидкість 6 fps в середньому ( $\sigma=0,86$  fps).

Результати практичної апробації використання вказаних підходів наведено в Таблиці 1.

ТАБЛИЦЯ 1. СЕРЕДНІ ШВИДКОСТІ РОБОТИ АЛГОРИТМУ

	Python 3	PyInstaller	C++
1920x1080	1,00 fps	0,88 fps	6,00 fps
1280x720	2,40 fps	1,97 fps	9,50 fps

Автоматична трансляція алгоритму на мову C++ не дала позитивних результатів. Використання специфічних інтерпретаторів також не вирішує поставлену задачу. Отже, найбільш прийнятним способом підвищення швидкості виконання алгоритму на Raspberry Pi 3 є трансляція коду на мову C++ вручну.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Чирков А.В. Методика реалізації програмних рішень для обробки даних на борту безпілотного літального апарату / Чирков А.В. // Математичне та програмне забезпечення інтелектуальних систем: тези доповідей XIII міжнар. наук.-практ. конф., м. Дніпропетровськ, 18–20 листопада 2015 р., Дніпропетровський національний університет ім. Олеся Гончара. – Дні-вськ: ДНУ, 2015. – С. 245–246.
- [2] Почему существует так много Питонов? [електронний ресурс] // <https://www.toptal.com/python/почему-существует-так-много-питонов/ru>

