"Równoległa, realizowana w czasie rzeczywistym implementacja wykrywania ruchu w sekwencji klatek filmowych"

08.06.2022

Spis treści

1 Opis ogólny

1.1 Nazwa programu

Wykrywanie ruchu na filmie - sekwencji klatek.

1.2 Cel

Głównym celem projektu było stworzenie aplikacji zaimplementowanej równolegle, która w czasie rzeczywistym wykrywa ruch obiektów w sekwencji klatek filmowych.

Kluczowym czynnikiem w projekcie była odpowiednia realizacja zrównoleglenia.

2 Opis technologii

2.1 Wybrane technologie

W celu realizacji zadania zostaną wykorzystane:

- Język Python w wersji 3
- Technologia MPI: mpi4py
- Biblioteka OpenCV oraz NumPy

3 Opis funkcjonalności

3.1 Uruchomienie programu

Program był uruchamiany z użyciem **venv** Pythona. W pliku **requirements.txt** podane są potrzebne biblioteki. Wystarczy użyć w swoim środowisku komendę: `pip install -r requirements.txt`

Program należy uruchomić używając linii komend. W tym celu należy podać nazwę programu oraz argumenty:

```
  mpiexec -n PROCESSES python move_detect.py [--fn FILENAME]
[--fps FRAMES_PER_SECOND] [--area MIN_AREA]
```

Opis wywołania:

- mpiexec - komenda do uruchomienia programu z MPI
- -n PROCESSES - liczba procesów używanych przez program
- move_detect.py - nazwa głównego programu
- -fn FILENAME - ścieżka do pliku filmowego. Jeśli parametr ten nie zostanie podany to zamiast pliku filmowego będzie używana zainstalowana kamera internetowa.
- -fps FRAMES_PER_SECOND - liczba klatek na sekundę do przetworzenia. Domyślną wartością jest bazowa liczba klatek na sekundę źródła.
- -area MIN_AREA - minimalna powierzchnia wykrywanych obiektów. Domyślną wartością jest 300.

W katalogu **/files** umieszczony jest plik **aquarium.mp4** pozwalający na szybkie przetestowanie możliwości parametru *-fn*

3.2 Przykłady uruchomienia

- Uruchomienie z podaniem źródła do analizowanego pliku filmowego, liczbą fps, oraz minimalnym obszarem wykrywania: **mpiexec -n 4 python move_detect.py --fn .\files\aquarium.mp4 --fps 10 --area 400**
- Uruchomienie z minimalnym obszarem wykrywania, domyślnie włączona będzie kamerka: **mpiexec -n 4 python move_detect.py --area 400**
- Uruchomienie bez parametrów, domyślnie włączona będzie kamerka, liczba fps będzie domyślna dla źródła, a minimalny obszar wykrywania będzie równy 300: **mpiexec -n 4 python move_detect.py**

3.3 Wyłączenie programu

Program można wyłączyć klikając klawisz **q** albo wymuszając **CTRL+C** w konsoli.

3.4 Możliwości programu

- Równoległe wykrywanie ruchu obiektów na podstawie podanego pliku filmowego.
- Równoległe wykrywanie ruchu obiektów na podstawie odczytu z kamerki internetowej (WebCam).
- Wyświetlanie zwrotnego wideo z graficznym oznaczeniem obiektów, które się poruszają.
- Działanie w czasie rzeczywistym.

3.5 Ograniczenia

- Wgrywane filmy powinny być nagrane najlepiej w bezruchu. Np. na statywie.
- Zaleca się, aby obiekty nie zlewały się bardzo z tłem

4 Format danych

4.1 Dane wejściowe

Plik filmowy .mp4 albo odczyt z kamerki internetowej (WebCam).

4.2 Dane wyjściowe

Okno z wyświetlanym obrazem i zaznaczonymi ramką wykrytymi obiektami.

5 Działanie programu

5.1 Realizacja wykrywania ruchu obiektów

1. Klatki są przerabiane na skalę szarości.

Rysunek 1: Obraz w skali szarości

2. Uzyskany obraz podaje się efektowi *blur* w celu usunięcia szumów.

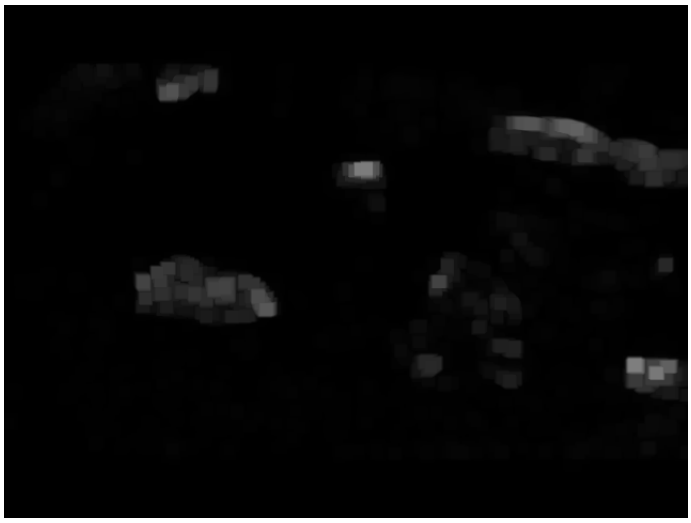
Rysunek 2: Obraz z blurem

3. Następuje odjęcie dwóch klatek (poprzedniej i obecnej) metodą *absdiff* (absolutna różnica).

Rysunek 3: Odjęcie absdiff

4. W celu wyeksponowania pomniejszych elementów dodawana jest dodatkowa macierz *kernel* i używana jest metoda *dilate* która powiększa

powierzchnię elementów

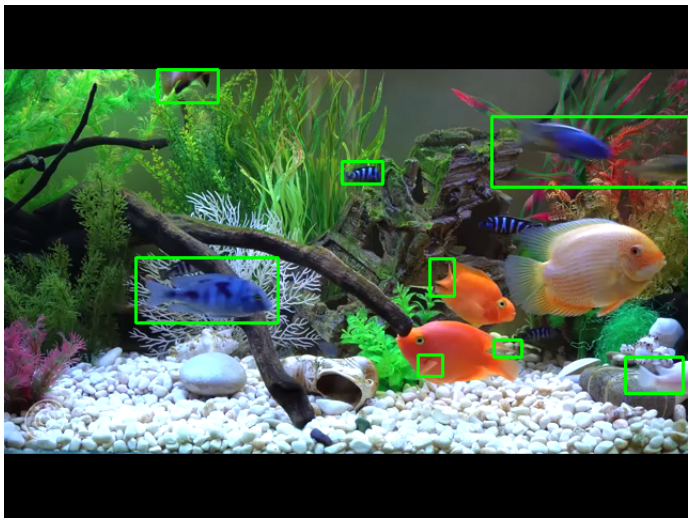


Rysunek 4: Operacja dilate

5. Kolejnym krokiem jest uzyskanie macierzy 0/1 używając metody *threshold* (metoda progowa)

Rysunek 5: Operacja threshold

6. Ostatecznie następuje sprawdzenie konturów obiektów. Za małe części są pomijane. Kontury są następnie nanoszone na nieprzetworzony (oryginalny) obraz.



Rysunek 6: Finalny obraz

6 Pomiary

6.1 Porównanie działania programu sekwencyjnego i równoległego

Id	Nazwa	Rozmiar	Rozdzielczość
0	Akwarium	5mb	około 480p
1	Webcam	-	około 480p
2	Duże nagranie	200 mb	1080p

Tabela 1: Badane źródła.

Webcam			
	Sekwencyjnie	Równolegle N=3	Równolegle N=8
ok. 25 FPS	0,012411 s	0,012092 s	0,013129 s
5 FPS	0,062777 s	0,083092 s	0,045694 s

Tabela 4: Porównanie czasów działania dla źródła "Webcam".

Akwarium			
	Sekwencyjnie	Równolegle N=3	Równolegle N=8
30 FPS	0,005602 s	0,002252 s	0,002284 s
5 FPS	0,005202 s	0,002078 s	0,002208 s

Tabela 2: Porównanie czasów działania dla źródła "Akwarium".

Duże nagranie			
	Sekwencyjnie	Równolegle N=3	Równolegle N=8
30 FPS	0,122428 s	0,163565 s	0,125910 s
5 FPS	0,136907 s	0,124880 s	0,144244 s

Tabela 3: Porównanie czasów działania dla źródła "Duże nagranie".