2015-12-22

Sebastian Łasisz 183668

**Transmisja, transkodowanie oraz przekształcanie danych wideo**

Teoria i inżynieria ruchu teleinformatycznego

## Opis funkcjonalności projektu

**Launcher** umożliwia użytkownikowi wybranie filmu do przetworzenia oraz ustalenie liczby filmów wyjściowych. Następnie wysyła on do VideoRetriever ścieżkę pliku do przetworzenia oraz ilość filmów wyjściowych. Po wysłaniu Launcher czeka na wiadomość od Statistics, która informuję o zakończeniu przetwarzania filmów. Po tym Launcher umożliwia użytkownikowi wyświetlenie poprzednio wygenerowanych wykresów w celu porównania parametrów filmów wyjściowych.

**VideoRetriever** pobiera od Launcher ścieżkę do pliku wideo oraz liczbę operacji do wykonania na filmie. Następnie wysyła do Server (port 1234), Receiver (port 1235) oraz Statistics (port 1236) liczbę operacji. Następnie ścieżkę video wysyła po porcie 1233 do Client oraz po porcie 1236 do Statistics. Jeżeli dane zostały odebrane to VideoRetriever kończy działanie.

**Client** odbiera od VideoRetriever ścieżkę do pliku wideo, a następnie koduję je do pliku jpg w celu zapisania strumienia danych do tablicy, po czym zapisuję strumień jako bity. Po zapisaniu danej klatki do tablicy wysyła on portem 1234 do Server. Jeżeli wszystkie klatki zostały zakodowane oraz wysłane to Client kończy działanie.

**Server** odbiera od VideoRetriever liczbę filmów do przetworzenia, a od Client zakodowane klatki, a następnie zamienia strumień bitowy w daną klatkę. Następnie modyfikuję daną klatkę w oparciu o wybraną metodę, po czym koduję ją i wysyła po porcie 1235 do Receiver. Jeżeli wszystkie klatki zostały wczytane, a następnie przetworzone i wysłane to Server kończy działanie.

**Receiver** odbiera od VideoRetriever liczbę filmów do stworzenia , a od Server wszystkie klatki filmu wejściowego, a następnie zgodnie z ustalonymi parametrami zamienia klatki w materiał wideo. Następnie wysyła materiał wideo do Statistics po porcie 1236. Jeżeli wszystkie materiały wideo zostały wysłane, to Receiver kończy działanie.

**Statistics** zbiera wszystkie materiały wideo, a następnie analizuje je pod kątem najważniejszych parametrów, takich jak: długość wideo, przepływność wideo, liczba klatek na sekundę w danym wideo, szerokość klatki w wideo, długość klatki w wideo oraz rozmiar pliku wideo. Po zebraniu danych odnośnie wszystkich plików Statistics generuje wykresy. Jeżeli wszystkie wykresy zostały wygenerowane to wysyła wiadomość do Launcher, a następnie kończy pracę.

## Wykorzystane technologieś

### Python

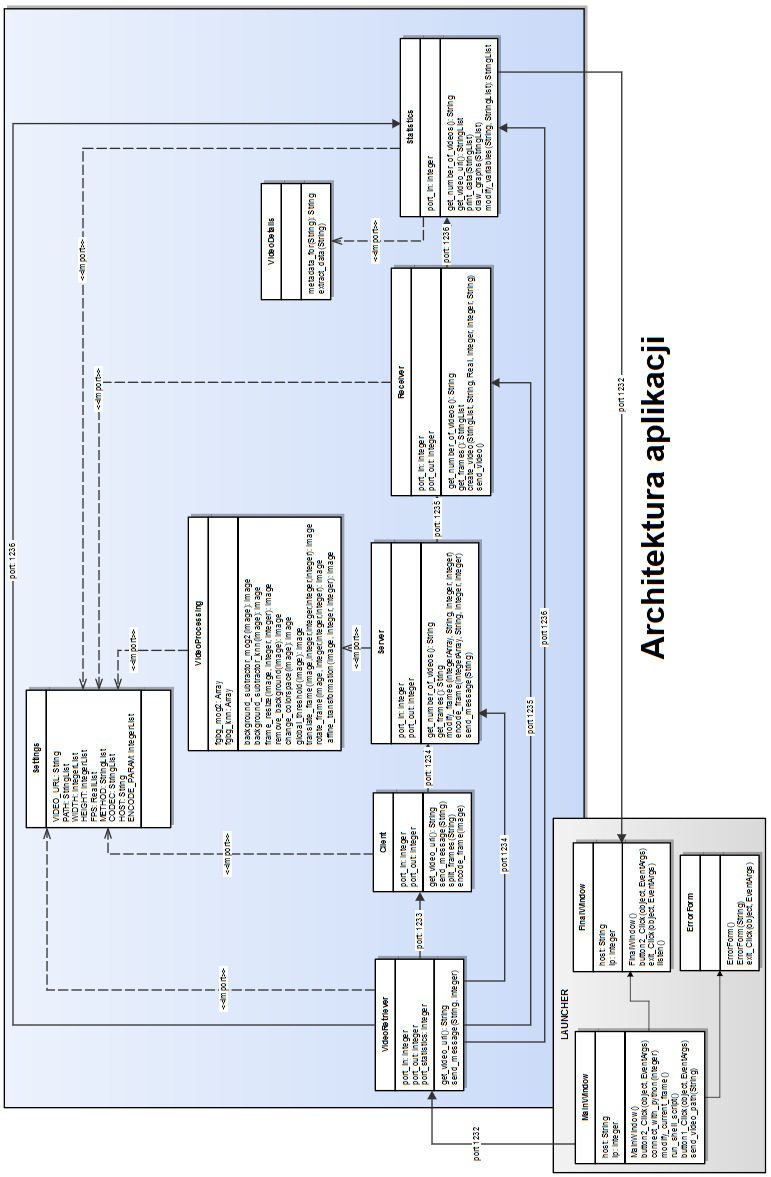
Funkcyjny język programowania, który został wykorzystany do stworzenia logiki aplikacji oraz komunikacji z interfejsem użytkownika.

### C#

Obiektowy język programowania, który został wykorzystany do stworzenia interfejsu użytkownika oraz do komunikacji z logiką aplikacji.

### Shell Script

Skryptowy język programowania, który został wykorzystany do uruchomienia wykorzystywanych funkcji logiki aplikacji.



## Wykorzystane moduły

### Socket

Jest to moduł, który zapewnia dostęp do interfejsu BSD socket.

### Os

Jest to moduł, który zapewnia dostęp do przenośnego sposobu korzystania z zależnych funkcji systemu operacyjnego. Zapewnia takie działania jak: otwieranie/zamykanie pliku, manipulowanie ścieżki pliku, czytanie wszystkich linii w pliku, tworzenie tymczasowych plików/folderów.

### Time

Jest to moduł, który zapewnia dostęp do różnych funkcji związanych z czasem.

## wykorzystane biblioteki

### OpenCV

Jest to biblioteka zawierające funkcję do wykorzystywania obróbki obrazu. Skupia się na przetwarzaniu obrazu w czasie rzeczywistym.

### Pygal

Jest to biblioteka, która pozwala tworzyć dynamiczne wykresy SVG.

### Hachoir

Jest to biblioteka, która pozwala oglądać i edytować binarne strumienie pole po polu. Pozwala ona przeglądać dany strumień podobnie do tego jak można przeglądać foldery i pliki. Plik jest rozdzielony na drzewo pól, gdzie najmniejszym polem jest jeden bit.

### NumPy

Jest to rozszerzenie do Python’a, które dodaje wsparcię dla dużych, wielowymiarowych tablic i macierzy, razem z dużą biblioteką funkcji matematycznych do wysokopoziomowych operacji nad tymi tablicami.

## wykorzystane metody

### Background subtractor mog2

Jest to Gaussian Mixture-based Background/Foreground Segmentation Algorythm. Wykorzystuje ona metodę modelowania każdego piksela tła przy pomocy mieszanki dystrybucji K Gaussiana, gdzie 3≤k≥5. Wagi tych mieszanin, są reprezentowane przez proporcjonalną ilość czasu jaki dany kolor pozostał na scenie. Ewentualnym kolorem tła będzie ten, który został dłużej i był bardziej statyczny. Metoda ta ma nieobowiązkowe parametry takie jak: długość historii, liczba mieszanin gaussiana, próg, etc.

### Background subtractor knn

Jest to również Gaussian Mixture-based Background/Foreground Segmentation Algorith. Jedną ważną cechą jest to, że algorytm wybiera odpowiedni współczynnik dystrybucji Gaussiana dla każdego piksela. Powoduję to, że jest on bardziej dostosowany do różnych scen spowodowanych oświetleniem. W tej metodzie jest możliwość wykrywania cieni.

### Remove background

Jest to metoda, która przy użyciu maski oraz prostokąta zamienia wszystkie piksele, które nie są czarne oraz wykraczają po za dany prostokąt na białe.

### Frame resize

Metoda polega na zmianie rozmiaru danej klatki.

### Change colorspace

Metoda polega na ekstrakcji kolorowego obiektu. Na początku zamieniany jest system barw z RGB na HSV. Następnie ustawiany jest próg na zasięg koloru niebieskiego. A na końcu, wydobyty jest obiekt o kolorze niebieskim.

### Global threshold

Metoda polega na zastąpieniu każdego piksela w obrazie czarnym pikselem jeżeli natężenie jest mniejsze od danej stałej, albo białym pikselem, jeżeli natężenie jest większe od stałej.

### Translate frame

Metoda polega na przesunięciu obiektu w danym kierunku (tx, ty) wykorzystując macierz transformacji .

### Rotate frame

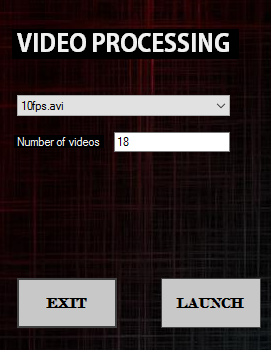
Metoda polega na obróceniu obrazu z regulacją środka obrotu o kąt . Można to osiągnąć wykorzystując macierz transformacji:

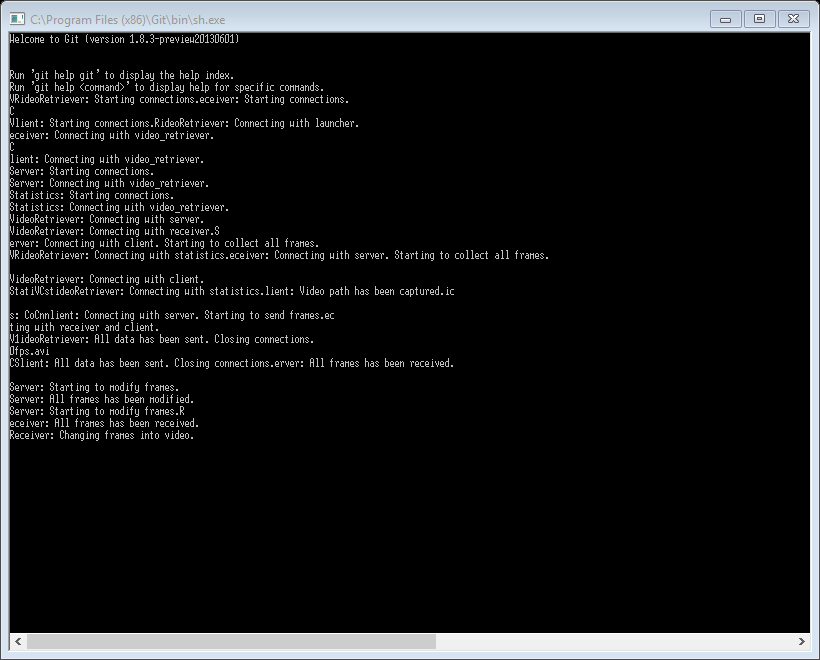
, gdzie .

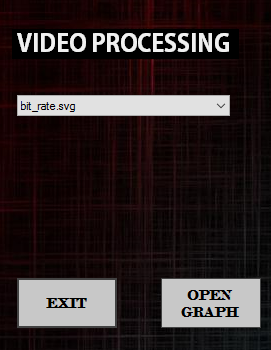
### Affine transformation

Metoda polega na przekształceniu geometrycznym przestrzeni euklidesowych odwzorowujących odcinki na odcinki.

## Wygląd interfejsu użytkownika

****Interfejs użytkownika został zaprojektowany w taki sposób, aby był przejrzysty i prosty w obsłudze. Po uruchomieniu Launcher.exe pojawia się okno, w którym użytkownik może wybrać dany materiał wideo, który zostanie wykorzystany do analizy oraz liczbę. Ponadto użytkownik może ustalić liczbę operacji które zostaną wykonane na danym materiale wideo (jest to również liczba wyjściowych filmów). Użytkownik ma do dyspozycji dwa przyciski: Exit, który zamyka aplikację, oraz Launch, który uruchamia analizę filmu. Po uruchomieniu analizy wideo wszystkie przyciski zostaną zablokowane, oraz pojawi się terminal, który jest odpowiedzialny za analizę wideo.

Gdy aplikacja zakończy analizę materiału wideo, zostanie odtworzony dźwięk, okno aplikacji oraz terminal zostaną zamknięte, a otworzy się drugie okno aplikacji. Użytkownik będzie miał możliwość wyboru pliku, który zawiera dany wykres. Ponadto do dyspozycji użytkownika będą dwa przyciski: Exit, który zamknie aplikację, oraz Open Graph, który otworzy wybrany wykres.

****

## Przykładowy wynik programu

Na wynik programu składają się trzy składowe:

##### **Zbiór plików filmowych**

W zależności od ustawień początkowych użytkownik może otrzymać od 1 do 19 filmów wideo. Każdy z tych plików różnych się od siebie parametrami. W zależności od użytej metody użytkownik może na przykład otrzymać: film obrócony względem filmu pierwotnego, bądź film czarnobiały.

**Przykładowe filmy wyjściowe.**

###### Background subtractor knn

****Global threshold

****

###### Rotate

****

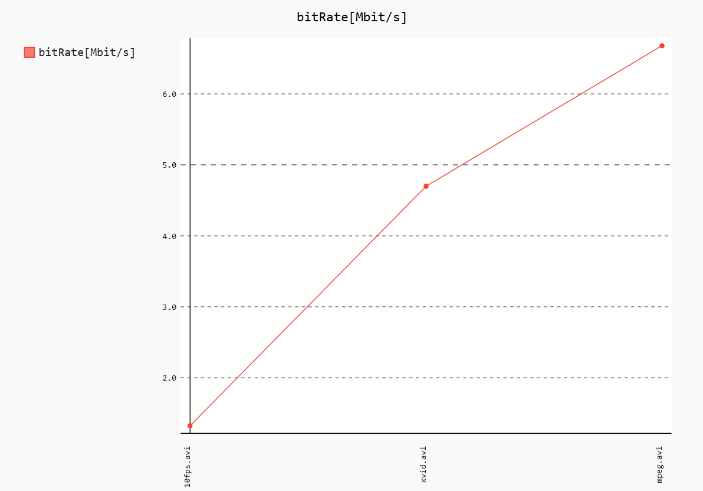
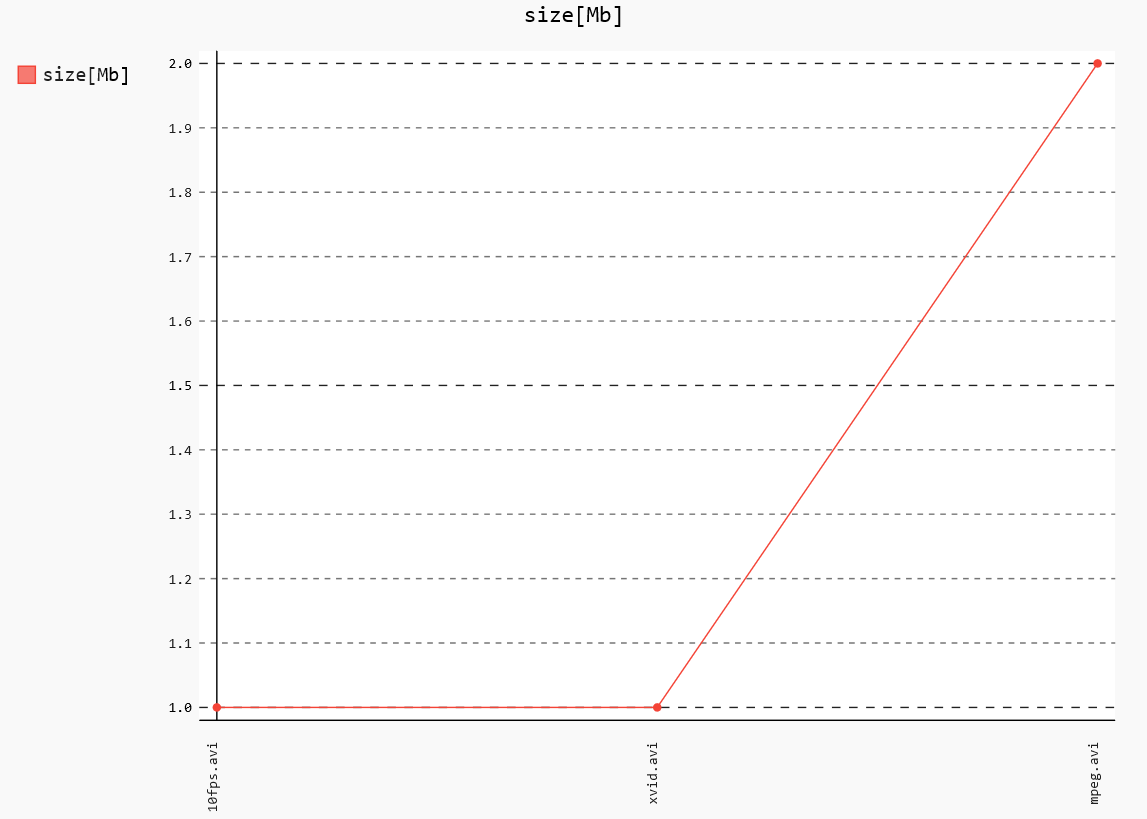
##### Parametry materiałów wideo

W zależności od ustawień początkowych użytkownik może otrzymać zbiór parametrów odnośnie przeanalizowanych filmów. Dla każdego z przeanalizowanych wideo zostaną wyznaczone takie parametry jak: przepływność (bit rate), liczba klatek na sekundę (fps), czas trwania, szerokość/wysokość filmu, etc. Wszystkie wyznaczone informację zostaną wykorzystane do rysowania wykresów. Przykładowe dane wideo:

##### 

##### Zbiór wykresów

Po wykonaniu analizy wideo użytkownik dostanie 9 wykresów (przepływność, długość filmu, final, liczba klatek na sekundę, wysokość klatek, rozmiar filmu, szerokość klatek), które przedstawiają porównanie wszystkich filmów, które zostały poddane analizie. Przykładowe wykresy:

****

## Wnioski

Szerokość wideo jest zależne od szerokości poszczególnych klatek. Wszystkie klatki w danym wideo muszą mieć dokładnie taką samą szerokość. Szerokość klatek jest szerokością wideo. Szerokość klatki nie jest zależna od innych parametrów wideo.

Wysokość wideo jest zależne od wysokości poszczególnych klatek. Wszystkie klatki w danym wideo muszą mieć dokładnie taką samą wysokość. Wysokość klatek jest wysokością wideo. Wysokość klatki nie jest zależna od innych parametrów wideo.

Liczba klatek na sekundę nie zależna od innych parametrów wideo.

Długość pliku jest zależna od liczby klatek na sekundę. Jeżeli liczba klatek rośnie, to długość spada oraz odwrotnie, jeżeli liczba klatek spada, to długość rośnie.

Wielkość pliku jest zależna od wysokości wideo, szerokości wideo, kodowania oraz jakości klatek. Jeżeli wysokość, bądź szerokość wideo rosną to wielkość pliku również rośnie. Jeżeli wysokość, bądź szerokość maleją to wielkość pliku również maleje. Wielkość pliku jest również zależna od rodzaju kodeka, którym zakodowany jest dany materiał wideo. Ponadto wielkość pliku jest zależna od wykonanych operacji na danej klatce (usunięcie tła zwiększyło wielkość pliku).

Przepływność wideo jest zależna od operacji, która została wykonana na danym fragmencie wideo. W zależności od operacji przepływność będzie rosła, bądź malała. Na przykład:

* jeżeli wysokość, bądź szerokość wideo rosną to przepływność wideo również rośnie
* jeżeli wysokość, bądź szerokość wideo maleją to przepływność wideo również maleje
* jeżeli wielkość wideo maleje, to przepływność również maleje
* w zależności od rodzaju kodeka, którym zakodowany jest dany materiał wideo, przepływność może rosnąć, bądź maleć

Przepływność pozostaje bez zmian tylko wtedy, kiedy obraz wejściowy i obraz wyjściowy zawierają dokładnie ten sam zbiór pikseli.