Temat: Podstawowe operacje na sekwencji wideo

Dwa warianty przetwarzania video:

- przechwytywanie strumienia obrazów z kamery w czasie rzeczywistym
- odczytywanie zarejestrowanego wcześniej strumienia video w pliku

Przechwytywanie strumienia obrazów z kamery

- Do przechwytywania video służy obiekt **VideoCapture**.
- Obiekt ten standardowo posiada parametr, który przyjmuje ścieżkę do pliku wideo, który ma być analizowany.
- Jeżeli wartością parametru będą liczby 0, 1, … , to obiekt przechwyci obraz z lokalnych kamer podłączonych komputera (standardowo wartość ta ma 0).
- Metoda obiektu `cap.read()` zwraca wartość bool (`True/False`): jeżeli ramka zostanie odczytana poprawnie, to będzie `True`. Przy pomocy tej funkcji można sprawdzać zakończenie czytania pliku z filmem.
- Trzeba pamiętać, aby po zakończeniu zwolnić obiekt przechwytywania i zamknąć jawnie okno.
- Jeżeli z nieokreślonych powodów opiekt nie chce się otworzyć, to można próbować wymusić jego otwarcie poleceniem: `cap.open()`.

```
# Przechwytywanie strumienia obrazów RGB z kamery

import cv2

cap = cv2.VideoCapture(0) #przechwycenie strumienia wideo
if not cap.isOpened():
    print("Nie mogę otworzyć kamery")
    exit()

while cap.isOpened():
    # przechwytywanie ramka po ramce
    # jeśli ramka jest wczytana poprawnie ret ma wartość True
    ret, frame = cap.read()

    if not ret:
        print("Nie można odczytać ramki. Możliwe, że koniec filmu. Wyjście…")
        break

    cv2.imshow('Moja kamera', frame)
    if cv2.waitKey(1) == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Przechwytywanie strumienia obrazów z kamery i zamiana na obraz w skali szarości

```
# Przechwytywanie strumienia obrazów z kamery w skali szarości

import cv2

cap = cv2.VideoCapture(0) #przechwycenie strumienia wideo
if not cap.isOpened():
    print("Nie mogę otworzyć kamery")
    exit()

while cap.isOpened():
    # przechwytywanie ramka po ramce
    # jeśli ramka jest wczytana poprawnie ret ma wartość True
    ret, frame = cap.read()

    if not ret:
        print("Nie można odczytać ramki. Możliwe, że koniec filmu. Wyjście…")
        break

    frame_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) #zmiana na skalę sza
rości
    cv2.imshow('Moja kamera', frame_gray)
    if cv2.waitKey(1) == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Przykład manipulowania przychwytywanym obrazem

```
# Manipulacje obrazem

import cv2

cap = cv2.VideoCapture(0) #przechwycenie strumienia wideo
if not cap.isOpened():
    print("Nie mogę otworzyć kamery")
    exit()

while cap.isOpened():
    # przechwytywanie ramka po ramce
    # jeśli ramka jest wczytana poprawnie ret ma wartość True
    ret, frame = cap.read()

    if not ret:
        print("Nie można odczytać ramki. Możliwe, że koniec filmu. Wyjście…")
        break

    frame = frame[:,::2]
    cv2.imshow('Moja kamera', frame)
    if cv2.waitKey(1) == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Dostęp do określonych cech badanego strumienia/filmu

- Do odczytywania parametrów służy metoda `cap.get(propId)`, gdzie `propId` to liczba od 0 do 18 określająca wartość konkretnego parametru.
- Do ustawiania parametrów obiektu służy natomiast metoda `cap.set(propId, value)`, gdzie `value` to nowa wartość danego parametru.
- Przykłady:
    - sprawdzenie szerokości i wysokości ramki przez:
        - `cap.get(cv2.CAP_PROP_FRAME_WIDTH)`
        - `cap.get(cv2.CAP_PROP_FRAME_HEIGHT)`
    - modyfikacja rozmiaru odbywa się poprzez parametry:
        - `ret = cap.set(cv2.CAP_PROP_FRAME_WIDTH,320)`
        - `ret = cap.set(cv2.CAP_PROP_FRAME_HEIGHT,240)`

Zapisywanie sekwencji wideo

- Do zapisywania sekwencji video służy obiekt **VideoWriter**, któremu przekazujemy następujące parametry:
    - nazwę pliku wyjściowego (np. `output.avi`)
    - kod – `FourCC`
        - jest to czterobajtowy kod używany do określania kodeka wideo i jest zależny od platformy
        - najbardziej popularny jest MJPG (MP4)
    - liczbę klatek na sekundę (`FPS`)
    - rozdzielczość
    - opcjonalna flaga `isColor`: jeżeli jest `True`, koder oczekuje ramki kolorowej, w przeciwnym przypadku działa z ramką w skali szarości. Domyślnie jest ona włączona.

```
# Zapisywanie filmu

import cv2

cap = cv2.VideoCapture(0) # przechwycenie strumienia wideo
out = cv2.VideoWriter('Z_kamery.avi', cv2.VideoWriter_fourcc(*'MJPG'), 30, (6
40,  480)) #przygotowanie streamu do zapisu

while cap.isOpened():
    ret, frame = cap.read()
    # przechwytywanie ramka po ramce
    # jeśli ramka jest wczytana poprawnie ret ma wartość True

    if not ret:
        print("Nie można odczytać ramki. Możliwe, że koniec filmu. Wyjście…")
        break

    cv2.imshow('Moja kamera', frame)
    out.write(frame) # zapisywanie
    if cv2.waitKey(1) == ord('q'):
        break

cap.release()
out.release()
cv2.destroyAllWindows()
```

Wczytywanie wideo z pliku

- Odtwarzanie wideo z pliku jest takie samo, jak przechwytywanie go z kamery, wystarczy zmienić indeks kamery na nazwę pliku wideo.
- Również podczas wyświetlania ramki użyj odpowiedniego czasu dla `cv2.waitKey()`. Jeśli jest zbyt małe, wideo będzie bardzo szybkie, a jeśli jest zbyt wysokie, wideo będzie wolne (w ten sposób można wyświetlać filmy w zwolnionym tempie).
- W normalnych przypadkach wynosi 25 milisekund.

```
# Wczytywanie filmu z pliku i wyświetlenie go

cap = cv2.VideoCapture("videos/S1610010.MP4") #przechwycenie strumienia wideo

while cap.isOpened():
    ret, frame = cap.read()
    # jeśli ramka jest wczytana poprawnie ret ma wartość True

    if not ret:
        print("Nie można odczytać ramki. Możliwe, że koniec filmu. Wyjście…")
        break

    cv2.imshow('Film', frame)
    if cv2.waitKey(25) == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```