

Zadania rozpoznawania

Ćwiczenie laboratoryjne nr 7

Analiza i przetwarzanie obrazów

Wprowadzenie

Celem ćwiczenia jest zapoznanie uczestników kursu z metodami lokalizacji cech charakterystycznych i rozpoznawania obiektów (na przykładzie twarzy)

Cechy charakterystyczne – przydatne polecenia

OpenCV udostępnia szereg algorytmów pozwalających na lokalizację charakterystycznych elementów obrazu:

Harris corner detection – funkcja `cv2.cornerHarris()`

Shi-Tomasi corner detection – funkcja `cv2.goodFeaturesToTrack()`

Scale-Invariant Feature Transform (SIFT) – funkcja `sift.detect()`

Speeded-Up Robust Features (SURF) – funkcja `surf.detectAndCompute()`

- wymaga `python-contrib 3.4.2.17`

Features from Accelerated Segment Test (FAST) – funkcja `fast.detect()`

Oriented FAST and Rotated BRIEF (ORB) – funkcja `orb.detect()`

Dla zadania detekcji obiektów stosuje się standardowo klasyfikator kaskadowy Haara powołany do życia funkcją `cv2.CascadeClassifier`. Jej parametrem jest jeden z predefiniowanych modeli klasyfikujących z pliku XML. Klasyfikator taki można samemu wyuczyć – dostarczając pliki uczące albo skorzystać z jednego z predefiniowanych modeli:

<https://github.com/opencv/opencv/tree/master/data/haarcascades>

Po zainicjowaniu kaskady możemy przystąpić do wykrywania funkcją `detectMultiScale`. Przyjmuje ona 3 parametry: obraz (w skali szarości), parametr skali mówiący o tym o jaki współczynnik zmniejszane są obrazy w kolejnych fazach pracy algorytmu oraz liczba sąsiadów – parametr mówiący ile sąsiednich ramek z wykrywanym obiektem musi się pojawić by obiekt był uznany za wykryty. Im większa wartość tym lepsza jakość rozpoznawania – ale równocześnie szansa na pominięcie obiektów niejednoznacznych (np. twarzy częściowo zasłoniętej itp.).

Realizacja ćwiczenia

- A. Proszę porównać metody Harrisa, SIFT, SURF, FAST i ORB w zadaniu rozpoznawania punktów charakterystycznych dla obrazu *lab5_2.png*. Dla której metody wyniki wydają się bliższe intuicji? Która wydaje się być bardziej „aplikowalna” (np. ze względu na czas obliczeń)?

