

Progowanie i segmentacja

Ćwiczenie laboratoryjne nr 3

Analiza i przetwarzanie obrazów

Wprowadzenie

Celem ćwiczenia jest zapoznanie uczestników kursu z metodami progowania (ang. thresholding) którego celem jest uzyskanie obrazu zbinaryzowanego. Ponadto podejmujemy również temat segmentacji obrazu.

Progowanie i segmentacja – przydatne polecenia

Aby uzyskać binarną reprezentację obrazu zapisanego w skali szarości korzystamy z polecenia `cv2.threshold`.

Funkcja ta w podstawowej wersji – z globalnym statycznie ustalonym progiem – przyjmuje dwa parametry – progową wartość intensywności oraz sposób realizacji progowania (`cv2.THRESH_BINARY`, `cv2.THRESH_BINARY_INV`, `cv2.THRESH_TRUNC`, `cv2.THRESH_TOZERO` i `cv2.THRESH_TOZERO_INV`).

Ustalenie progu globalnie dla całego obrazu nie pozwala na uwzględnienie lokalnej charakterystyki obrazu (np. oświetlenia). Dla tego celu korzystamy z lokalnego adaptacyjnego progowania. Ustala on próg na podstawie wartości intensywności w zadanym otoczeniu danego piksela.

W tym wariancie polecenie `cv2.threshold` przyjmuje trzy parametry:

- metoda:
 - o `cv2.ADAPTIVE_THRESH_MEAN_C` – próg wyznaczany na podstawie średniej intensywności w otoczeniu
 - o `cv2.ADAPTIVE_THRESH_GAUSSIAN_C` – próg wyznaczany jest na podstawie średniej ważonej intensywności sąsiednich pikseli z wagami wyznaczonymi z rozkładu Gaussa
- rozmiar bloku – określa rozmiar otoczenia branego pod uwagę przy obliczaniu lokalnego progu
- `C` – stała odejmowana od średniej/średniej ważonej

Trzecią możliwością przeprowadzenia progowania jest znalezienie podziału który minimalizuje wariancję (zmiennosc) w ramach każdej z dwóch klas uzyskanych w ramach podziału. Tę metodę nazywamy metodą Otsu (od nazwiska jej twórcy).

W przypadku progowania metodą Otsu korzystamy z flagi metody `cv2.THRESH_OTSU` równocześnie jako progu możemy użyć 0. Funkcja zwraca poza binaryzacją również obliczoną wartość progu.

Segmentacja obrazu jest jego podziałem na części tzw. segmenty. Łączą one piksele o podobnych atrybutach. Binarizacja jest przykładem globalnej segmentacji na dwa segmenty.

Bardzo częstym sposobem segmentacji jest klasteryzacja – polega ona na zgrupowaniu pikseli o podobnej kolorystyce w klastry, maksymalnie jednolite pod kątem atrybutów pikseli (a skrajnie różne pomiędzy klastrami). Klasteryzacja może służyć również do kwantyzacji kolorystycznej obrazu – po zadaniu liczby klastrów i zrealizowaniu klasteryzacji możemy każdy piksel reprezentować przez kolor środka klastra.

Klasteryzację w OpenCV można zrealizować z użyciem procedury k-średnich i funkcji `cv2.kmeans`.

Funkcja ta przyjmuje dane w formacie – jeden piksel to jeden rząd, należy również podać zadaną liczbę klastrów, kryterium stopu oraz sposób generowania rozwiązania początkowego. Funkcja zwraca przypisanie punktów do klastrów oraz uzyskane środki klastrów.

Realizacja ćwiczenia

- a) Proszę wczytać przykładowy obraz i skonwertować go do skali szarości.
- b) Proszę sprawdzić i porównać wynik progowania ze statycznym ustalonym globalnie progiem zrealizowanego z użyciem każdego z dostępnych w OpenCV trybu progowania (`cv2.THRESH_BINARY`, `cv2.THRESH_BINARY_INV`, `cv2.THRESH_TRUNC`, `cv2.THRESH_TOZERO` i `cv2.THRESH_TOZERO_INV`)
- c) Proszę zastosować metody adaptacyjnego doboru progu. Jaki wpływ na wynik ma rozmiar sąsiedztwa? Która z metod daje lepsze rezultaty?
- d) Proszę zastosować metodę Otsu, wykreślić histogram obrazu i zlokalizować na nim uzyskaną wartość progu.
- e) Proszę wczytać drugi z obrazów i dokonać jego segmentacji z użyciem algorytmu k-średnich na 2, 4, 8 klastrów. Proszę zweryfikować wizualnie wynik klasteryzacji (kolor każdego piksela to kolor środka klastra do którego go przypisaliliśmy) i sprawdzić czy kolory odpowiadają jakimś cechom zdjęcia które analizujemy.

Uwaga: jeśli klasteryzacja zdjęcia trwa zbyt długo to proszę zmniejszyć jego rozmiar.

- f) Proszę samodzielnie zaimplementować metodę Otsu. Czy otrzymany próg różni się od bibliotecznej implementacji? Jeśli tak to dlaczego?

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4310076>