

POLITECHNIKA ŚWIĘTOKRZYSKA
Wydział Elektrotechniki, Automatyki i Informatyki
Katedra Elektrotechniki Przemysłowej i Automatyki
Zakład Urządzeń i Systemów Automatyki

Budowa i Oprogramowanie Komputerowych Systemów Sterowania

Wprowadzenie do systemów wizyjnych z wykorzystaniem biblioteki OpenCV

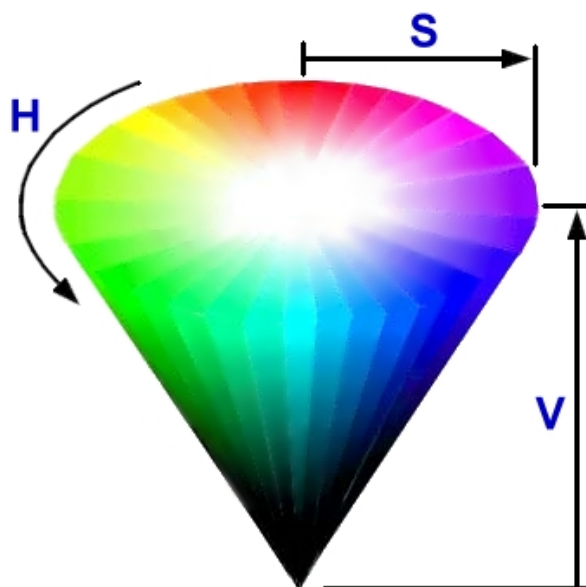
Instrukcja laboratoryjna

Paweł Strączyński

2019

Model HSV (hue – brawa, saturation – nasycenie, value – wartość) to model barw który znacznie lepiej nadaje się do rozpoznawania kolorowych obiektów. Model ten względem mode-

lu HSV charakteryzuje się lepszą relacją w stosunku do percepcji ludzkiego oka. Reprezentację graficzną modelu RGB przedstawiono na rysunku 1.2.



Rysunek 1.2: Model barw HSV

1.3 Wczytywanie obrazu i konwersja przestrzeni barw w OpenCV

Na listingu 1.3 przedstawiono sposób importu obrazu w języku Python z wykorzystaniem biblioteki OpenCV oraz konwersję do przestrzeni HSV oraz skali odcieni szarości.

Listing 1.1: Import obrazu w języku Python i konwersja obrazu

```
1 import cv2 #Import biblioteki OpenCV
2
3 img = cv2.imread('obraz.jpg') #Wczytanie obrazu z pliku
4 imgHSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV) #Konwersja BGR -> HSV
5 imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #Konwersja BGR -> GRAY
```

2 Filtracja obrazu

Filtracja obrazu to metoda cyfrowego przetwarzania sygnałów. Dokonywana jest dla poszczególnych pikseli obrazu. W wyniku jej działania otrzymuje się nowy, przekształcony (prze-filtrowany) obraz. Zastosowanie filtrów w przetwarzaniu obrazów oznacza, że do obliczenia

nowej wartości punktu (piksela) brane są pod uwagę wartości punktów z jego otoczenia. W zależności od wielkości zastosowanej maski filtru branych jest pod uwagę odpowiednia ilość pikseli sąsiadujących.

a)

1	1	1
1	1	1
1	1	1

b)

1	-2	1
-2	5	-2
1	-2	1

c)

1	1	2	1	1
1	2	4	2	1
2	4	8	4	2
1	2	4	2	1
1	1	2	1	1

Rysunek 2.1: Przykładowe maski filtrów: a) filtr uśredniający 3x3, b) filtr górnoprzepustowy 3x3, c) filtr Gaussa 5x5

Filtrację można wykonać dokonując operacji splotu (konwulsji) w postaci dyskretniej dwóch sygnałów (obrazów).

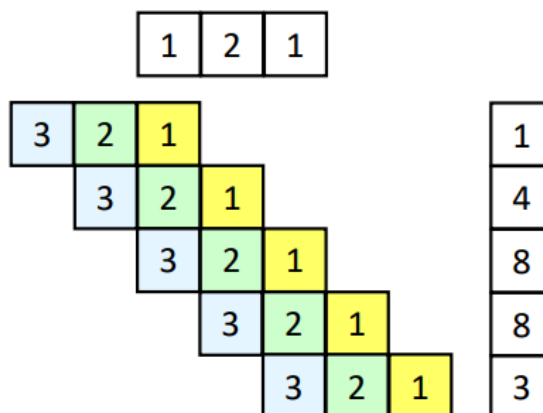
$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m] = f[n-m]g[m] \quad (1)$$

Z definicji splot to sumy iloczynów dwóch sygnałów, gdy jeden jest przesunięty względem drugiego, dla wszystkich przesunięć k takich, że przynajmniej jedna próbka się pokrywa. Graficzną interpretację operacji splotu przedstawiono na rysunku 2.3. W wyniku operacji splotu otrzymuje się sygnał o długości $N + M - 1$. Filtracji można dokonać również z wykorzystaniem szybkiej transformaty Fouriera. W takim przypadku należy:

- przekształcając obraz i filtr do dziedzin częstotliwości (FFT).
- dokonać mnożenia otrzymanych sygnałów,
- dokonać odwrotnego przekształcenia Fouriera (IFFT)

Biblioteka OpenCV oferuje funkcje realizujące operacje filtracji z użyciem popularnych filtrów m.in. uśredniającego, filtru Gaussa, medianowego itd. - `cv2.blur()`, `cv2.GaussianBlur()`,

$$a = [1, 2, 1]; b = [1, 2, 3]$$



Rysunek 2.2: Splot funkcji

`cv2.medianBlur()`. OpenCV posiada również uniwersalną funkcję która umożliwia wykonanie wykonanie dwuwymiarowego splotu dwóch sygnałów (obrazu i filtru) o określonej masce. Przykład takiej filtracji z użyciem filtru o postaci:

$$K = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (2)$$

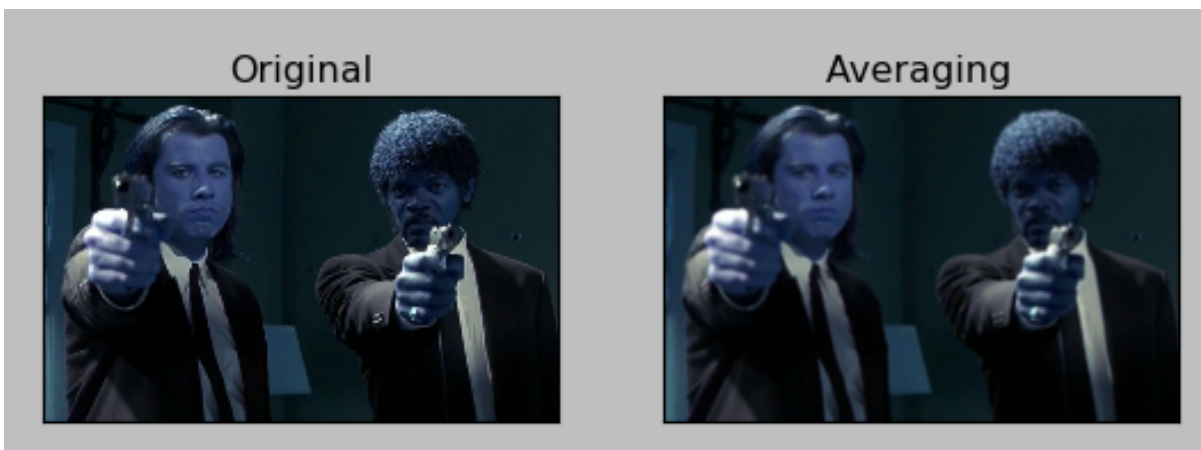
pokazano na listingu poniżej.

Listing 2.2: Przykład filtracji uśredniającej

```

1 import cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4 img = cv2.imread('image.jpg') #Odczyt danych z pliku
5 kernel = np.ones((3,3),np.float32)/3 #Maska filtru
6 dst = cv2.filter2D(img,-1,kernel) #Filtracja obrazu
7 plt.subplot(121),plt.imshow(img),plt.title('Original')
8 plt.xticks([], plt.yticks([]))
9 plt.subplot(122),plt.imshow(dst),plt.title('Averaging')
10 plt.xticks([], plt.yticks([]))
11 plt.show()

```



Rysunek 2.3: Filtracja obrazu

2.1 Progowanie obrazu (binaryzacja)

Binaryzacja obrazu to proces konwersji obrazów kolorowych lub w skali odcieni szarości do postaci binarnej (czarno-białej). Binaryzacji dokonuje się przez progowanie (*thresholding*). Progowanie polega na ustaleniu wartości progowej dla obrazu. Wszystkie piksele poniżej tej wartości zapisywane są jako 0 (białe) a powyżej jako 1 (czarne).

$$g(x, y) = \begin{cases} 1 & f(x, y) \geq T \\ 0 & f(x, y) < T \end{cases} \quad (3)$$

Funkcję odpowiedzialną za operację progowania przedstawiono poniżej.

```
ret,thresh = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
```

3 Zadania do wykonania

1. Napisać program który wczyta obraz z pliku oraz dokona konwersji koloru do przestrzeni barw hsv oraz skali odcieni szarości.
2. Zapoznać się z operacją filtracji obrazu. Zastosować różne rodzaje filtrów. Porównać wyniki operacji z użyciem filtrów o różnych wielkościach.
3. Przetestować działanie operacji progowania obrazu. Porównać wynik działania operacji o różnym poziomie binaryzacji.

4. Zapoznać się z tutorialiem dotyczącym wykrywania konturów - uruchomić przedstawiony przykład.
5. Napisać program wykrywający kontur na własnym obrazie.