

# Zaawansowane algorytmy wizyjne

materiały do ćwiczeń laboratoryjnych

Piotr Pawlik, Tomasz Kryjak

Copyright © 2018 Piotr Pawlik, Tomasz Kryjak

PUBLISHED BY AGH

*First printing, March 2018*

# Contents

<b>1</b>	<b>Uogólniona transformata Hougha .....</b>	<b>5</b>
1.1	Cel zajęć	5
1.2	Implementacja uogólnionej transformaty Hougha	5
1.3	Wyszukiwanie wzorców różniących się orientacją (nieobowiązkowe)	6



# 1 — Uogólniona transformata Hougha

## 1.1 Cel zajęć

- implementacja uogólnionej transformaty Hougha
- wyszukiwanie wzorców za pomocą uogólnionej transformaty Hougha

## 1.2 Implementacja uogólnionej transformaty Hougha

1. Ze strony kursu pobierz archiwum z danymi do ćwiczenia i rozpakuj je we własnym katalogu roboczym.
2. Utwórz nowy skrypt. Na podstawie obrazu ze wzorcem 'trybik.jpg' stworzymy tablicę R-table. W tym celu wyznacz kontury oraz gradienty na obrazie wzorca. Kontury można wyznaczyć podobnie jak w ćwiczeniu z odległości Hausdorffa, natomiast do wyliczenia gradientów można wykorzystać filtry Sobela, przykładowo:

```
sobelx = cv2.Sobel(im, cv2.CV_64F, 1, 0, ksize=5)
sobely = cv2.Sobel(im, cv2.CV_64F, 0, 1, ksize=5)
```

Należy wyliczyć 2 macierze (obrazy) - zarówno wartość gradientu (pierwiastek sumy kwadratów Sobela pionowego i poziomego) jak i orientację (wykorzystując funkcję np.  $\arctan2$ ). Macierz wartości gradientu warto znormalizować dzieląc ją przez jej wartość maksymalną.

Przed wypełnieniem R-table należy wybrać punkt referencyjny - niech będzie to środek ciężkości wzorca wyznaczany ze zbinaryzowanego obrazu wzorca z wykorzystaniem momentów (jak w ćwiczeniu z odległości Hausdorffa).

Do wypełnienia R-table będą potrzebne wektory łączące punkty konturu/konturów (we wzorcu mogą wystąpić 'dziury') z punktem referencyjnym. Do R-table wpisujemy długości tych wektorów oraz kąty jaki tworzą z osią OX (tu znów przyda się funkcja np.  $\arctan2$ ). Miejsce wpisania do tablicy R-table wyznacza orientacja gradientu w punkcie konturu, **przy czym proszę przeliczyć radiany na stopnie** - R-table będzie miała 360 wierszy. R-table można zaimplementować jako lista 360 list:

```
Rtable = [[] for i in range(360)].
```

Wówczas np. `Rtable[30]` będzie listą współrzędnych biegunowych punktów konturu, których gradient orientacji wynosi około  $30^\circ$ .

3. Na podstawie obrazu 'trybiki2.jpg' oraz R-table z poprzedniego punktu wypełnij dwuwymiarową przestrzeń Hougha - wylicz gradient w każdym punkcie jak w poprzednim punkcie i dla punktów, których znormalizowana wartość gradientu przekracza 0.5 dodaj jeden w przestrzeni akumulacyjnej w punktach wynikających z wpisów do R-table - czyli w punktach:

```
x1 = -r*np.cos(fi) + x
y1 = -r*np.sin(fi) + y
```

gdzie r,fi - wartości z R-table, natomiast x,y - współrzędne punktu przekraczającego 0.5

4. Wyszukaj maksimum w przestrzeni Hougha i zaznacz je w obrazie wejściowym. Wyznaczanie współrzędnych maksimum można przeprowadzić za pomocą funkcji np.argmax lub konstrukcją np.where(hough.max() == hough) gdzie hough jest typu np.array. Z kolei do zaznaczania na obrazie można wykorzystać funkcję:  
plt.plot([m\_x], [m\_y], '\*', color='r')

### 1.3 Wyszukiwanie wzorców różniących się orientacją (nieobowiązkowe)

1. Teraz zwiększymy przestrzeń Hougha do 3 wymiarów dodając obroty. Przeształć kod z punktu 1.2 . 3 tak, aby dodawanie jedynki było powtórzone dla co dziesiątego kąta z zakresu [0-360) (kąty odejmujemy od fi). Powiększenie przestrzeni Hougha o trzeci wymiar można zrealizować przez powiększenie 'shape' jak poniżej;

```
new_hough_shape = hough.shape + (36,)
new_hough=np.zeros(new_hough_shape);
```

(36,) jest tu jednoelementową krotką - reprezentuje ona trzeci wymiar - co dziesiąty stopień kąta

2. Aby uwzględnić obroty można wykorzystać stworzoną dla wzorca R-table, przy czym przy wyliczaniu indeksu tej tablicy dla konkretnego punktu obrazu należy uwzględnić nie tylko orientację gradientu, ale także kąt obrotu (indeks = orientacja - kąt; wyrażone w stopniach)

Do wyznaczania punktów akumulacji należy zmodyfikować wzory z punktu 1.2 . 3:

```
x1 = -r*np.cos(fi + df) + x
y1 = -r*np.sin(fi + df) + y
```

gdzie df to kolejne kąty obrotu (wyrażone w radianach)

3. W celu wykrycia wszystkich wzorców na obrazie należałoby wykryć odpowiednio duże maksima lokalne. Jednakże dla ułatwienia zastąpimy tę operację kilkukrotnym wykrywaniem maksimum globalnego i wyzerowaniem jego otoczenia. Do znalezienia maksimum w 3-wymiarowej przestrzeni Hougha można wykorzystać metodę tablicy numpy - argmax(). Jednakże zwraca ona jeden indeks w tablicy 'spłaszczonej' do jednego wymiaru. Do uzyskania użytecznych dla nas indeksów (po 3 wymiarach) należy wynik argmax() podać do funkcji np.unravel\_index (wraz z 'shape' tablicy). Po znalezieniu maksimum zaznacz je w obrazie wejściowym (potrzebne są tylko 2 współrzędne maksimum, kąt na razie nie jest istotny).
4. Jeżeli znalezione zostało to samo maksimum co w punkcie 1.2 . 4 wyzeruj otoczenie maksimum w przestrzeni Hougha - przykładowo:  
hough[m[0]-delta:m[0]+delta, m[1]-delta:m[1]+delta, :] = 0, gdzie m zawiera indeksy maksimum, a delta to rozmiar otoczenia (może przyjąć wartość 30)  
Następnie ponów operacje z poprzedniego punktu. Wyszukaj i zaznacz w sumie 5 kolejnych maksimumów. Sprawdź czy znalezione zostały wzorce.
5. Dodatkowo można dla ładniejszej wizualizacji wyrysować wzorzec w znalezionych punktach. W tym celu należy wyrysować w obrazie wejściowym punkty odpowiadające

wszystkim wektorom z tablicy R-table używając wzorów bardzo podobnych do tych z punktu 1.2 . 3, przy czym  $x, y$  to będą współrzędne maksimum, a  $\theta$  kąt  $\theta$  wynika z 3 współrzędnej maksimum.







## Bibliography