Zaawansowne algorytmy wizyjne

materiały do ćwiczeń laboratoryjnych

Piotr Pawlik, Tomasz Kryjak

PUBLISHED BY AGH

First printing, March 2018

Contents

1 — Termowizja

1.1 Prosta analiza obrazu termowizyjnego

Obraz termowizyjne mają tą zaletę, że obiekty np. ludzie są na nich zwykle dość dobrze widoczni (tj. wyróżniają się od tła). Wyjątkiem jest sytuacja, gdy temperatura tła jest zbliżona do temperatury obiektu - w bardzo ciepły dzień.

W pierwszym etapie do detekcji sylwetek ludzi wykorzystamy następujące algorytmy:

- binaryzacja
- filtracja
- indeksacja
- analiza wyników indeksacji,

TeksT

1. Ze strony kursu pobierz sekwencję
2. Wczytywanie sekwencji w OpenCV jest względnie proste:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

cap = cv2.VideoCapture('vid1_IR.avi')

while(cap.isOpened()):
  ret, frame = cap.read()
  G = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
  cv2.imshow('IR',G)
  if cv2.waitKey(1) & 0xFF == ord('q'):  # przerwanie petli po
    wcisnieciu klawisza 'q'
    break
cap.release()
```

   Uwaga. Zamiast nazwy pliku można podać liczbę (np. 0) – wtedy aplikacja spróbuje się połączyć z kamerą (USB, wbudowaną), oczywiście o ile taka jest dostępna w systemie. Można też podać adres kamery IP działającej w protokole rtsp.
3. Binaryzacja – w pierwszym przybliżeniu proszę zastosować stały próg. Dla przypomnienia: cv2.threshold (szczegóły sprawdzić w dokumentacji). Wartość progu należy dobrać eksperymentalnie, tak aby sylwetki był jak najlepiej wyodrębnione, przy jednoczesnym braku szumów. Od razu należy zaznaczyć, że nie da się tego zrobić idealnie.

4. Filtracja – można wykorzystać cały „arsenał” środków – filtry medianowe oraz operacje morfologiczne.
5. Indeksacja – funkcje connectedComponents oraz connectedComponentsWithStats. Sugeruje się użycie drugiej wersji (obliczane są od razu pole, prostokąt otaczający oraz środek ciężkości). Sposób dostępu wyników proszę odszukać w dokumentacji.
6. Analiza wyników indeksacji:
   - w pierwszym kroku proponuje się wyświetlenie prostokątów otaczających dla wszystkich obiektów – funkcja cv2.rectangle
   - najprostsze kryterium to rozmiar – obiekty powinny być większe niż X (dobrać).
   - można też analizować kształty – szukamy pionowych sylwetek.
   - ostatni i najtrudniejszy element, to próba połączenia sylwetek, które są podzielone. Jest to często występujące zjawisko na obrazach termowizyjnych osób ubranych (ciepło, grubo). Odsłonięte części ciała, takie jak głowa, ręce są najjaśniejsze, a okolice pasa i nóg najciemniejsze. W tym przypadku warto rozważyć heurystykę, polegającą na sklejaniu leżących „jeden pod drugim” prostokątów otaczających – do samodzielnego zaprojektowania.

1.2 Detekcja obiektów z wykorzystaniem wzorca probabilistycznego (nieobowiązkowe)

Inne podejście do zagadnienia detekcji pieszych polega na obserwacji, że sylwetki stojących osób mają dość charakterystyczny i powtarzalny wygląd. Pozawala to na stworzenie „wzorca” sylwetki, a następie wyszukiwanie go na obrazie (technika okna przesuwnego).

W pierwszym etapie potrzebna będzie baza wycinków sylwetek o określonych rozmiarach – 192 x 64. W celu ich pozyskania wykorzystamy aplikację z części pierwszej – dodajmy do niej zapisywanie do pliku png zweryfikowanych sylwetek.

1. Wycięcie ROI:

   ```
   ROI = G[y1:y2, x1:x2]
   ```

2. Zapis ROI:

   ```
   cv2.imwrite('sample_%06d.png' % iPedestrian,ROI)
   ```

   iPedestrian to licznik globalny – należy inkrementować co zapis,
3. Wybieramy ok. 30 -50 zdjęć. Zasadniczo powinny one być frontalne, ale dopuszcza się również zrobione z boku.
4. W kolejnym kroku, w osobnym skrypcie należy wybrane zdjęcia wczytać, przeskalować i wyznaczyć wzorzec. Skalowanie cv2.resize – do rozmiaru 192 x 64
   Binaryzacja (jak wcześniej, choć można rozważyć zmianę progu). Uwaga. Należy zmienić przypisywaną wartość wyjściową z 255 na 1.
   W razie potrzeby, można rozważyć też filtrację.
   Ostatecznie wzorzec probabilistyczny powstaje na zasadzie DPM = DPM + B, przy czym B musi być binarny, a nie [0;1].
   Otrzymany wzorzec zapisujemy do pliku.

Do samej detekcji wykorzystujemy trzeci skrypt.

W nim realizujemy wczytywanie sekwencji podobne jak w pierwszym, przy czym w pierwszym etapie najlepiej przećwiczyć działania metody na wybranej ramce. Przykładowa jest umieszczona na stronie kursu, inne można pozyskać zapisując ramki do png w programie 1.

Wczytujemy wzorzec (może być potrzebna konwersja do odcieni szarości). Konwertujemy do typu float32, dzielimy przez liczbę obrazów, z których został utworzony (uzyskujemy skalowanie do zakresu 0-1). Wynik oznaczamy DMP_1 oraz obliczamy negację DPM_0 = (1-DPM) tj. prawdopodobieństwo przynależności do tła.

Binaryzujemy obraz z sekwencji testowej. Następnie prowadzimy analizę w oknie 192 x 64 (technika okna przesuwnego). Przykładamy takie okno to kolejnych lokalizacji na obrazie (można rozważyć krok większy niż 1). Dla maski binarnej (B) zliczamy prawdopodobieństwo, że jest sylwetką ludzką:

$\sum_y \sum_x B(y,x) * DPM\_1(y,x) + (1 - (By,x)) * DPM\_0(y,x)$

Wynik zapisujemy w obrazie pomocniczym: result = np.zeros((360,480), np.float32)

Do wizualizacji należy wyniki znormalizować:

result = result / np.max(np.max(result))

ruint8 = np.uint8(result*255)

Ostatni etap to odnajdywanie maksimów lokalnych.

Najprostsze rozwiązanie to szukanie maksimum „na bieżąco", podczas analizy poszczególnych okien. Przy czym to podejście pozwala znaleźć tylko jedno maksimum. Proszę sprawdzić i narysować prostokąt otaczający dla znalezionej postaci.

W ramach pracy własnej proszę opracować metodę znajdywania wielu maksimów. Należy przy tym pamiętać o dwóch sprawach: a) ustalić próg dolny, b) zastosować eliminację obszarów już odwiedzonych.

Bibliography