Osnove racunalniskega vida, Detekcija gibanja

Rene Jaušovec

2023/24

```
1 usage  Rene Jausovec *
def camshift(slika, sablone, lokacije_oken, iteracije, napaka):
    hsv = cv.cvtColor(slika, cv.COLOR_BGR2HSV)
    nova_lokacija_oken = []
    for sablona, lokacija_okna in zip(sablone, lokacije_oken):
        #povratna_projekcija = calc_back_project(hsv, [0, 1], sablona, [0, 180, 0, 256])
        #povratna_projekcija = cv.calcBackProject([hsv], [0, 1], sablona, [0, 180, 0, 256], 1)
        povratna_projekcija = izracunaj_povratno_projekcijo(hsv, sablona)
        lokacija_okna = mean_shift(povratna_projekcija, lokacija_okna, iteracije, napaka)
        nova_lokacija_oken.append(lokacija_okna)
    return nova_lokacija_oken
```

Funkcija camshift sprejme sliko, seznam šablon, lokacije oken, število iteracij in toleranco napake. Najprej se vhodna slika pretvori v barvni prostor HSV. Nato se za vsako šablono in njeno pripadajočo lokacijo okna izračuna povratna projekcija. To je postopek, ki oceni verjetnost, da vsaka točka v sliki pripada določeni šabloni. Nato se uporabi algoritem mean shift za iskanje najverjetnejše lokacije šablone v sliki. Nova lokacija oken se shranjuje v seznam nova_lokacija_oken in se na koncu funkcije vrne.

```
1 usage  Rene Jausovec
def mean_shift(back_projected_image, window_location, iterations, error_threshold):
    window_x, window_y, window_width, window_height = window_location
    for _ in range(iterations):
        # Calculate the centroid of the window
        region_of_interest = back_projected_image[window_y:window_y + window_height, window_x:window_x + window_width]
        moments = cv.moments(region_of_interest)
        if moments["m00"] != 0:
            centroid_x = int(moments["m10"] / moments["m00"])
            centroid_y = int(moments["m01"] / moments["m00"])
        else:
            centroid_x, centroid_y = window_width // 2, window_height // 2

        # Move the window to the centroid
        delta_x = centroid_x - window_width // 2
        delta_y = centroid_y - window_height // 2

        # Check for convergence
        if abs(delta_x) < error_threshold and abs(delta_y) < error_threshold:
            break

        window_x += delta_x
        window_y += delta_y

    return window_x, window_y, window_width, window_height
```

Funkcija mean_shift izvaja algoritem mean shift za sledenje objekta v sliki. Sprejme sliko povratne projekcije, lokacijo okna, število iteracij in prag napake. Na začetku razčleni lokacijo okna na koordinate in dimenzije. Nato v vsaki iteraciji izračuna centroid območja interesa znotraj okna in premakne okno proti centroidu. Preveri konvergenco z napako in prekine iteracije, če je dosežena. Na koncu vrne posodobljeno lokacijo okna.

```python
1 usage  Rene Jausovec
def izracunaj_histogram_in_normaliziraj(image, mask, bins, ranges):
    """Docstring."""
    # Apply the mask to the image
    masked_image = cv.bitwise_and(image, image, mask=mask)

    # Calculate the histogram
    histogram, bin_edges = np.histogram(masked_image.ravel(), bins=bins, range=ranges)

    # Normalize the histogram
    histogram = histogram.astype('float')
    histogram /= (histogram.sum() + np.finfo(float).eps)

    return histogram
```

```python
1 usage  Rene Jausovec
def izracunaj_povratno_projekcijo(slika, histogram):
    # Iz slike izlušči kanal odtenka (hue)
    kanal = slika[:, :, 0]

    # Uporabi vrednosti odtenka kot indekse za iskanje ustreznih vrednosti v histogramu
    # Opomba: Prepričajmo se, da so indeksi celoštevilske vrednosti
    povratna_projekcija = histogram[kanal.astype(int)]

    # Normaliziraj povratno projekcijo, da imajo vrednosti v obsegu [0, 255]
    cv.normalize(povratna_projekcija, povratna_projekcija, alpha: 0, beta: 255, cv.NORM_MINMAX)

    # Pretvori povratno projekcijo v 8-bitno sliko
    povratna_projekcija = povratna_projekcija.astype(np.uint8)

    return povratna_projekcija
```

izracunaj_povratno_projekcijo sprejme sliko in histogram ter izračuna povratno projekcijo slike na podlagi histograma. Najprej iz slike izlušči kanal odtenka (hue), nato uporabi vrednosti odtenka kot indekse za iskanje ustreznih vrednosti v histogramu. Nato normalizira povratno projekcijo na obseg [0, 255] in jo pretvori v 8-bitno sliko, preden jo vrne.

izracunaj_histogram_in_normaliziraj sprejme sliko, masko, število predalov histograma (bins) in območje vrednosti (ranges), nato pa izračuna in normalizira histogram slike glede na dano masko. Najprej uporabi masko na sliki, nato izračuna histogram s pomočjo np.histogram. Nato normalizira histogram, da se vsote vrednosti histograma enašajo s 1, preden ga vrne.

Delovanje mean shift/camshift algoritma lahko pokvari več dejavnikov:

Slabo izbrana začetna lokacija okna: Če začetna lokacija ni ustrezno določena, lahko algoritem konvergira v napačno rešitev ali pa potrebuje več iteracij za dosego pravilne rešitve.

Slaba izbira prostorskih dimenzij: Če prostorske dimenzije niso ustrezno izbrane, lahko algoritem nepravilno sledi objektom ali pa ne zazna premikov.

Spreminjanje osvetlitve in barv: Spremembe osvetlitve ali barv v okolici objekta lahko vplivajo na natančnost sledenja, saj se lahko zaznavajo kot premiki objekta.

Zastoji v gibanju objekta: Če se objekt premika zelo hitro ali pa obstajajo zastoji v gibanju, lahko algoritem izgubi sled ali pa sledi napačnemu objektu.

Spreminjanje začetne velikosti okna lahko močno vpliva na delovanje sledenja. Pri pomanjševanju okna lahko algoritem postane bolj občutljiv na lokalne podrobnosti, kar lahko privede do napačnega sledenja zaradi prevelikega vpliva šuma. Pri povečevanju okna pa lahko algoritem postane manj natančen in lahko izgubi sled objektu zaradi prevelikega povprečenja informacij v okolici. Zelo pretirano pomanjševanje okna lahko privede do izgube sledi objektu, saj algoritem ne bo imel dovolj informacij za zaznavanje premikov. Zelo pretirano povečevanje okna pa lahko povzroči, da algoritem sledi okoliškim podrobnostim namesto ciljnemu objektu.

Slabosti algoritma Camshift vključujejo:

Občutljivost na začetne parametre: Algoritem je občutljiv na začetne parametre, kot so začetna lokacija okna in velikost okna. Nepravilno izbrane začetne parametre lahko privedejo do napačnega sledenja ali pa zahtevajo več iteracij za dosego pravilne rešitve.

Omejitev na en objekt: Algoritem je zasnovan za sledenje samo enemu objektu naenkrat. Če je v sliki več objektov ali pa se objekti prekrivajo, lahko algoritem izgubi sled ali pa sledi napačnemu objektu.

Občutljivost na spremembe osvetlitve in barv: Spremembe osvetlitve ali barv v okolici objekta lahko vplivajo na natančnost sledenja, saj algoritem temelji na histogramih barv v sliki.