Projekt 2: Sledenje z MeanShift

Ema Leila Grošelj

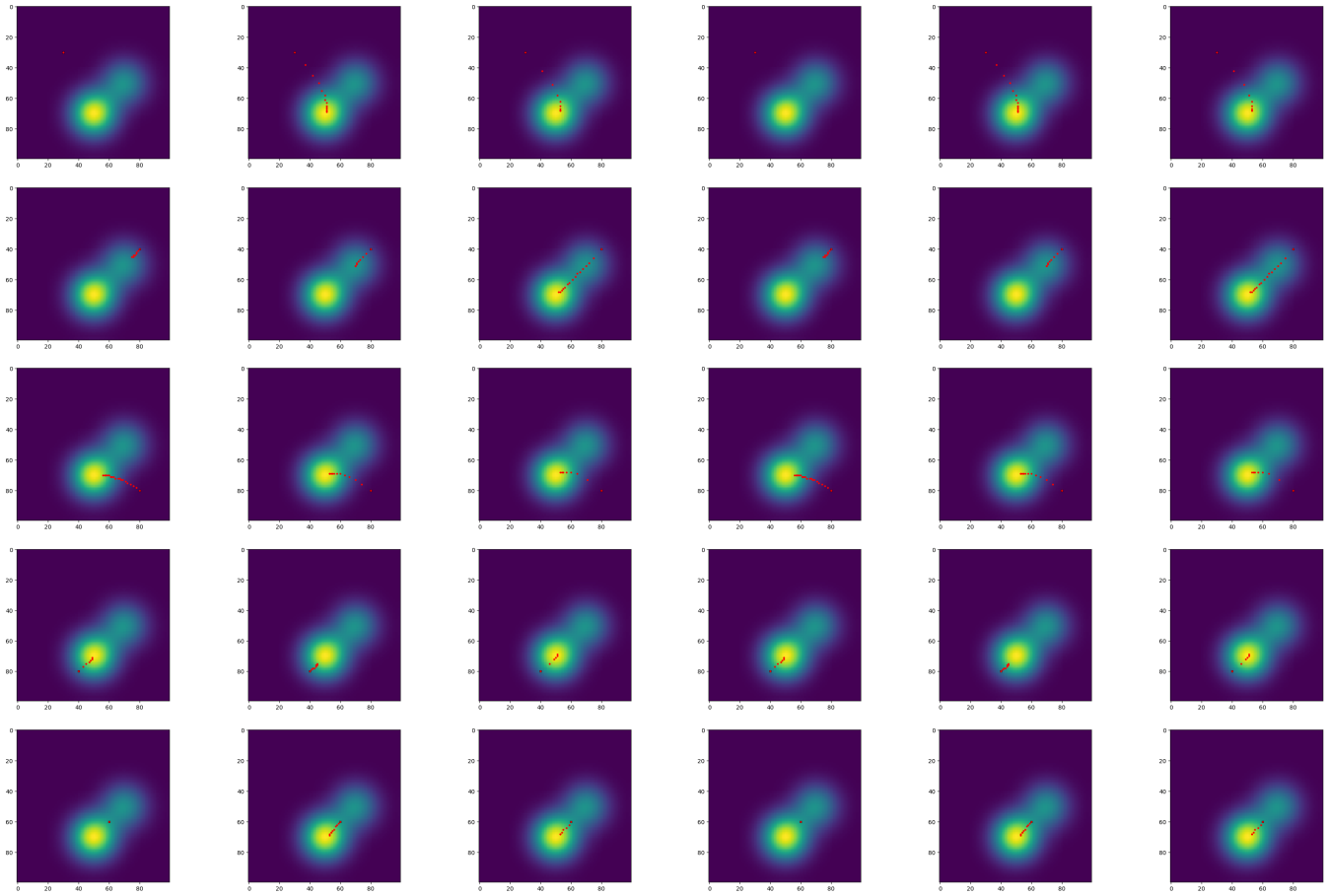

Figure 1: Konvergenca mean-shift algoritma za $x_0 = [(30,30),(80,40),(80,80),(40,80),(60,60)]$ in $velikost\_jedra = [11,21,31]$

Figure 2: Konvergenca mean-shift algoritma za $x_0 = [(30,30),(80,40),(80,80),(40,80),(60,60)]$ in $velikost\_jedra = [11,21,31]$

x_0 vel_jedra	(30,30)	(80,40)	(80,80)	(40,80)	(60,60)
11	0	7	20	7	0
21	12	8	11	6	9
31	8	15	7	5	6

Table I: Število iteracij, do konvergence. Obtičali smo na začetku (rdeče) in skonvergirali smo v neglobalni maksimum (modro).

I. Uvod

Implementirala sem mean-shift algoritem in sledilnik. Za mean-shift jedro sem vzela epanechnik. Njegov odvod je konstanta.

II. Mean-shift algoritem

Mean-shift algoritem sem preizkusila na *responses_1*. Uporabila sem razlicne dimenzije jeder in različne začetke x_0. Rezultati so prikazani na sliki 2 in v tabeli I. Fajli so dveh vrst: Obticimo na mestu, ker imamo pemajhno jedro, da bi se zavedali malo bolj oddaljene okolice. Po drugi strani pa lahko skonvergoramo v napačen maksimum, vendar je težko reči, kateri maksimum je pravilen. Načeloma je to najbližji maksimum, ampak a je to samo naključen šum, ki ima zraven veliko bolj verjeten maksimum.

Kar se ustavitvenega pogoja tiče, se mi zdi najbolj smiselno, da končamo, ko je meanshift enak 0. Alternativno bi lahko končali po nekaj iteracijah, na primer 10. Vendar bi s tem izgubili nekaj natančnosti (imamo primere z 11 pa vse do 20 potrebnimi iteracijami). Poleg tega je optimalno število iteracij težko oceniti, saj je odvisno od velikosti jedra in velikosti slike/regije/tarče. Število iteracij kot pogoj bo prišel prav kasneje pri zaporedju več slik (zaradi hitrosti).

Da, lahko pohitrimo konvergenco. Premišljeno moramo zmanjšati dimenzije pdf. Rezultati s polovičnega pdf (razpolovljene so tudi velikosti jeder) so prikazani v tabeli II. Mean-shift algoritem sem preizkusila tudi na *responses_2* in *responses_3* (moja primera gl. sliko 3). Vidimo, da v prvem primeru hitreje deluje jedro velikosti 11, v drugem pa jedro velikosti 5. Pomembna je skala.

III. Preizkusi

Grafi 4 prikazujejo odvisnost števila neuspehov od vrednosti posameznih parametrov za 5 različnih VOT(14) zaporedij slik.

x_0 vel_jedra	(30,30)	(80,40)	(80,80)	(40,80)	(60,60)
11 (5)	0	0	11	0	0
21 (11)	9	5	9	4	5
31 (15)	7	14	6	4	5

Table II: Zmanjšani dimenziji pdf za polovico: Število iteracij, do konvergence. Obtičali smo na začetku (rdeče) in skonvergirali smo v neglobalni maksimum (modro).

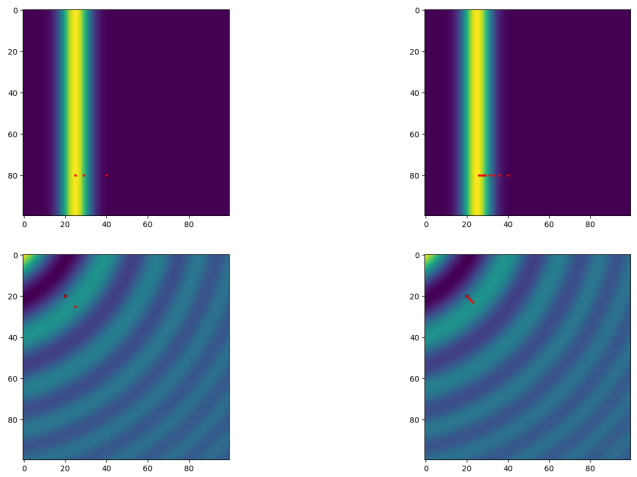

Figure 3: Konvergenca mean-shift algoritma na *responses_2* za $x_0 = (30, 30)$ in *velikost_jedra* $= [11, 31]$ (zgoraj) ter za *responses_3* za $x_0 = (20, 20)$ in *velikost_jedra* $= [5, 9]$

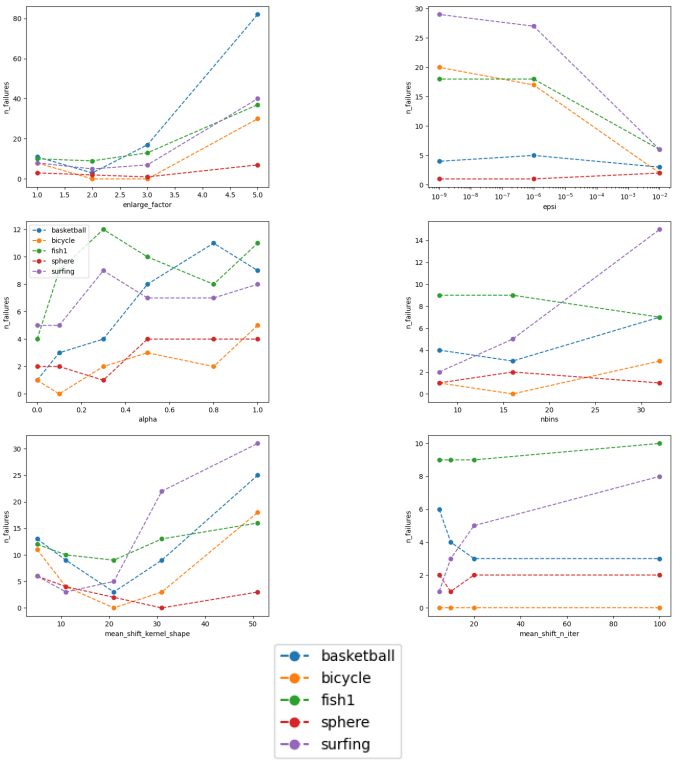

Figure 4: Kako parametri *enlarge_factor*, *epsi*, *alpha*, *nbins*,*mean_shift_kernel_shape* in *mean_shift_n_iter* vplivajo na število neuspehov na 5 različnih zaporedjih.

Vsak graf prikazuje spremembe ob spreminjanju enega samega parametra, pri čemer so ostali nastavljeni na privzete vrednosti, ki so razvidne iz tabele III, kjer so zapisane s črno barvo. Na podlagi 4 sem privzete vrednosti prilagodila (zelena v III), da bi zmanjšala število neuspehov.

Zanimivo mi je, da delovanje pri *sphere* izboljša tako povečanje kot zmanjšanje *nbins*. Nepričakovano je tudi, da je v večini primerov najboljša izbira za *alpha* kar 0, kar ustreza temu, da je histogram q vse skozi enak, saj bi osebno pričakovala, da bi bilo dobro. da se prilagaja spreminjajočemu se objektu. Zdi se mi, da je pri *fish*1 glavni razlog, za to, da je bolje imeti konstanten q, spremenljivo ozadje ribe, ki bi pokvarilo predstavitev ribe. Vendar je vseeno skoraj težko verjeti, da z *alpha* $= 0$ (ne smemo sicer pozabiti tudi na 2 reinicializaciji) lahko sledimo tako spremenljivemu objektu. Izkaže se tudi, da ni slabo, če imamo za ustavitveni pogoj omejitev števila iteracij (gl. graf 4 (desno spodaj).

Sledi komentar neuspehov iz tabele III. Napaka pri *basketball* je, da sledilnik preide iz našega košarkarja na sodnika. Odpravljanja te težave bi se, če smo že izčrpali prilagajanje parametrov lahko lotili:

- z uvedbo drugačnih robustnih reprezentacij tarče (na primer več njih tako, da se en osredotoča na oči drugi na superge ... in bi jih združili v ansambel)
- tako, da bi upoštevali verjetnost, da košarkar tako nenadno spremeni smer premikanja

	EF	eps	alf	nbins	MSkernel	MSiter	**nfail**
basketball	2	10e-3	0	16	(21,21)	20	**1**
bicycle	2	10e-3	0.1	16	(21,21)	20	**0**
fish1	2	10e-3	0	32	(21,21)	10	**2**
sphere	2	10e-3	0.3	8	(31,31)	20	**0**
surfing	2	10e-3	0	8	(11,11)	4	**0**

Table III: Parametri, ki minimizirajo število neuspehov za 5 zaporedij slik in število neuspehov. Prilagoditev parametrov (zelena), nespremenjene privzete vrednosti (črna).

Napaki pri *fish*1 se zgodita, ker se ribina površina zelo spremeni. Najprej se zmanjša, ko nas riba pogleda naravnost, in nato poveča, ko nam riba pokaže bok. Zanimivo ob vrtenju, ki sledi ne pride do napak. Menim, da je to zato, ker riba vedno ostane lepo na sredini našega območja. Tam je namreč res pomembno, da ostane ista barvna distribucija. Takšnih napak bi se lotila z bolj prilagodljivo regijo, ki bi omogočala prilagajanje višine in širine regije oziroma, če to ne bi zadostovalo, bi poskusila bitno masko oz. uteži, ki so povsem prilagodljive znotraj območja tako, da lahko za vsak piksel povemo verjetnost, ali je v ozadju ali ospredju.

IV. Dodatek

Implementirala sem $c = extract\_background\_histogram(patch, nbins, k)$, kjer k predstavlja razmerje med širino okvirja in širino slike ter višino okvirja in višino slike. Nastavila sem ga na 0.2. Okvir predstavlja bitno masko, ki jo podam kot *weights* v *extract_histogram*. Potem kot na prosojnicah s predavanj elementoma pomnožim, da dobim $c * p$ in $c * q$. Metoda pomaga premostiti okluzijo v primeru *basketball*, rezultata pri *fish*1 pa žal ne izboljša.

Poizkusila sem tudi druge zapise slik. Meritve sem izvedla na privzetih parametrih. Rezultati so prikazani na grafu 5. Edina resna konkurenca *BGR* po mojih meritvah je *HSV*, ki zniža število napak pri dveh zaporedjih.

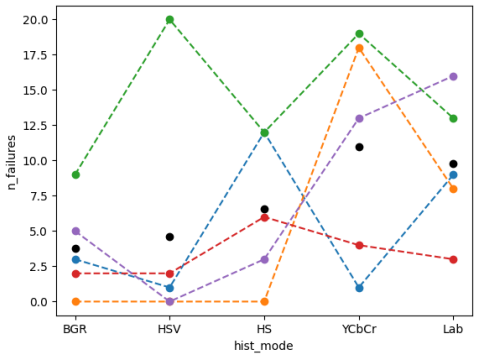

Figure 5: Graf števila neuspehov v odvisnosti od izbire barvnega prostora. Povprečje števila neuspehov (črna).

V. Zaključek

Mean-shift metoda odlično deluje na kratkih posnetkih. Za kaj daljšega z več okluzije, bi verjetno zraven rabili še kakšno metodo za odkrivanje targeta na sliki oz. reševanje neuspehov.