

# Mean shift sledilnik

## 2. naloga pri predmetu Napredne metode računalniškega vida

Ivan Antešić<sup>1</sup>

Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, Večna pot 113,  
Slovenija  
`ia6382@student.uni-lj.si`

**Keywords:** Računalniški vid · Mean Shift · Sledilnik.

### 1 Uvod

V okviru druge domače naloge smo implementirali sledilnik, ki uporablja metodo mean shift. Algoritem smo nato testirali, poskusili nadgraditi in na koncu zapisali glavne ugotovitve.

#### 1.1 Mean shift

Mean shift je iterativna metoda za iskanje maksimuma verjetnostne funkcije, uporablja pa se lahko tudi za gručenje točk. Vsako iteracijo se malo pomaknemo v težišče sesednjih točk in tako postopoma dosežemo vrh funkcije. Formula za pozicijo naslednje iteracije je enaka:

$$x^{(k+1)} = \frac{\sum_{i=1}^N x_i * w_i * g(\| (x^{(k)} - x_i)/h \|^2)}{\sum_{i=1}^N w_i * g(\| (x^{(k)} - x_i)/h \|^2)}$$

kjer je  $x_i$  položaj  $i$ -te točke,  $w_i$  vrednost funkcije v tej točki,  $g$  pa odvod jedra, ki ga uporabljamo za "vzpenjanje" po funkciji.

#### 1.2 Mean shift tracker

Mean shift tracker je sledilnik, ki deluje na modelu barve. Z primerjanjem barvnega histograma regij slik, poiščemo tisto, ki je najbolj podobna začetni predlogi in tako lokaliziramo tarčo. Za določitev podobnosti se uporablja Bhattacharyya razdalja, ki jo uporabimo za določitev funkcije cene:

$$E(x) = 1/2Ch \sum_{i=1}^N w_i * k(\| (x - x_i)/h \|^2).$$

Na tej funkciji uporabimo mean shift algoritem in se premaknemo v lokacijo, ki vsebuje največjo podobnost.

Sledilnik deluje dobro, ker na model barvnih histogramov ne vplivajo transformacije kot so rotacija, delna pokritost tarče ipd. Ima pa svoje slabosti, ki smo jih odkrili s testiranjem in opisali v naslednjem poglavju.

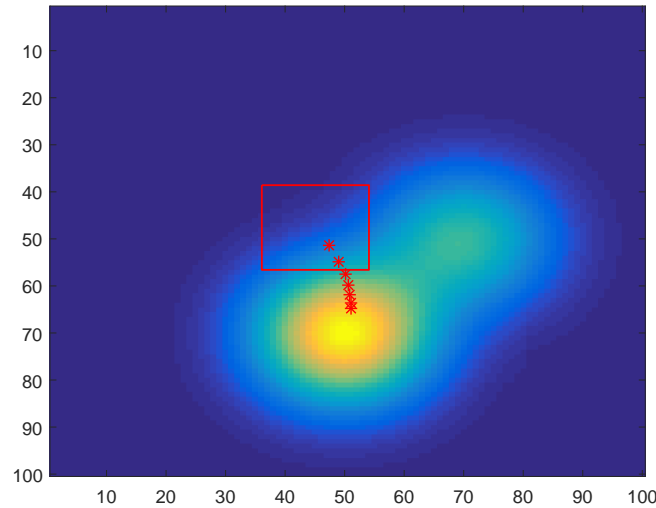
V implementaciji smo uporabili epanechnikovo jedro, saj se pri odvodu poenstavi v običajno seštevanje in ga lahko izpustimo iz omenjenih enacb. Uporabili smo tudi posodabljanje modela začetne predloge  $Q$  v vsakem okvirju (angl. frame), po formuli:  $Q = \text{alfa} * Q_{\text{stari}} + (1 - \text{alfa}) * Q_{\text{novi}}$ . Od parametra alfa je torej odvisno kako hitro se bo predloga prilagodila spremembam tarče in tako lažje sledila skozi sekvenco.

## 2 Testiranje

### 2.1 Mean Shift

Kako hitro (v kolikih iteracijah) in kako natančno bo skonvergirala algoritem je odvisno od parametrov, ki jih nastavimo z eksperimentiranjem.

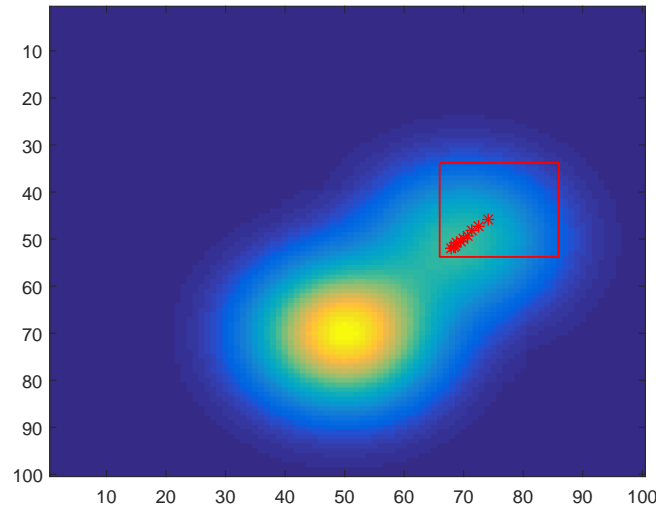
S parametrom  $e$  določimo kriterij za ustavitev - ko je premik manjši kot  $e$  prenehamo z izvajanjem. Ta mora biti majhen, da mean shift doseže vrhu funkcije in se ne ustavi malo pred njim (kot lahko vidimo na sliki 1). Vendar če povečamo  $e$  se poveča tudi število iteracij.



**Fig. 1.** Delovanje mean shift metode, na testni funkciji 1.  $e = 1.5$ ,  $h = 21$ . Za konvergenco je bilo potrebnih 7 iteracij.

Parameter  $h$  določa velikost jedra ter posledično točke, ki jih pregledujemo v vsaki iteraciji. Če je  $h$  majhense lahko na robovih funkcije sploh ne premakne.

Zgodi se lahko tudi, da se ustavi na lokalnem maksimumu namesto globalnemu (vidno na sliki 2).



**Fig. 2.** Delovanje mean shift metode, na testni funkciji 1.  $e = 0.5$ ,  $h = 21$ . Za konvergenco je bilo potrebnih 10 iteracij.

Če povečamo velikost jedra, lahko na sliki 3 vidimo, da mean shift doseže globalni maksimum, ker je jedro dovolj veliko, da upošteva točke dlje od lokalnega maksimuma.

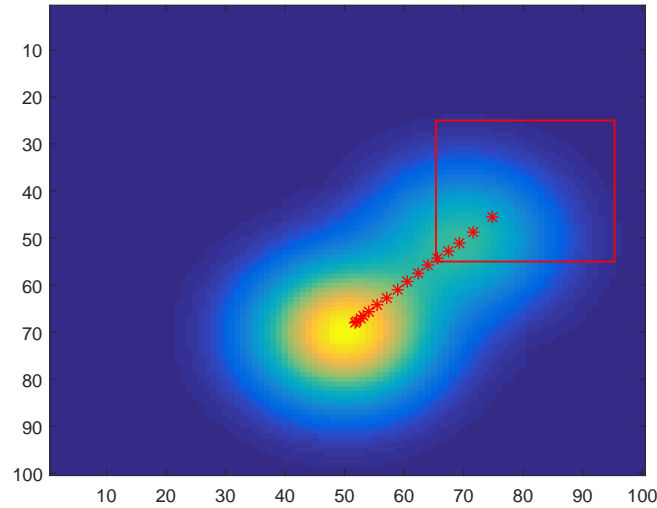
Tudi kriterij za ustavitev vpliva na preseg lokalnega maksimuma, saj prevelik  $e$  povzroči zaključek izvajanja, ko dosežemo lokalni vrh. Mean shift je namrec samo prilagojoč (angl. self adapting) - ko se približuje vrhu bo naredil manjše premike. Večji  $e$  zmanjša število iteracij vendar, ne skonvergira točno pri vrhu.

Pomembna je tudi začetna pozicija. Če začnemo v točki, ki je v popolnoma ravni okolici, algoritem ne bo deloval saj se ne mora premakniti nikamor.

Potrebno je narediti kompromis in določiti pravo razmerje med parametri, da bo algoritem deloval čimbolj hitro in pravilno. Algoritem sem testiral še na primeru svoje funkcije (slika 4). Ugotovil, sem da pri parametrih  $h = 23$   $e = 0.5$  deluje hitro (25 iteracij) in razmeroma natančno. še vedno se lahko včasih ustavi na lokalnem vrhu, ali obtici v globoki dolini. Najbolj zanesljivo delovanje je vidno na sliki 5, vendar ta zahteva veliko iteracij.

## 2.2 Mean Shift sledilnik

adsa [1] Sledilnik sem testiral na sekvencah slik zbirke VOT14 (dostopno na url: <http://box.vicos.si/vot/vot2014.zip>).



**Fig. 3.** Delovanje mean shift metode, na testni funkciji 1.  $e = 0.5$ ,  $h = 31$ . Za konvergenco je bilo potrebnih 16 iteracij.

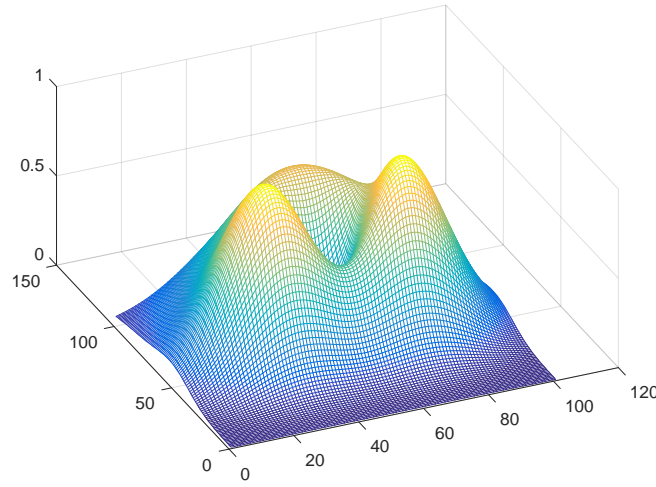
**Osnovni sledilnik** V uvodu opisan osnovni algoritem smo testirali na sekvencah 'basketball', 'ball' in 'car'. Pri vseh je sledilnik deloval brez padcev (angl. fail). Kljub temu je pri primerih 'ball' in 'car' komaj sledil, saj so tarče spreminjale velikost, sledilnik pa ne. To lahko vidimo na sliki 6. Še posebej dobro je sledil v primeru 'basketball' (slika 7). Dobro delovanje je posledica ozadja, ki je drugačno od ospredja in se ne razlikuje veliko skozi celotno sekvenco.

Na sekvenci 'gymnastics' smo opazili, da se algoritem preveč osredotoča na ozadje tarče - predvsem zgornji modri ograji. Ko se tarča premakne na drugo ozadje se zgodi, da sledilnik zaostane in pade. S nastavitvijo parametra alfa (ki skrbi za posodobitev modela) na 0.05, se delovanje malo izboljša vendar ne zadosti, da bi sledil brez napak.

Na sekvenci 'woman' se je algoritem odzval najslabše. Tarčo je izgubil petkrat. Vzrok je ponovna inicializacija predloge na različno obarvanih območjih (avtomobilih), ki nato izginejo iz slike. Začetna napaka se zgodi zaradi sledenja temnim hlačam in belim čevljem, ki jih kmalu zameša s podobnimi barvami ozadja, ko gre tarča čez pločnik. To lahko vidimo na sliki 8.

Delovanje sem izboljšal z spremembo barvnega prostora. V HSV prostoru z 8 koši (angl. bins) histograma sledilnik pade le dvakrat - v tem bravnem prostoru je večja razlika med pločnikom in tarčo (slika 9). Podobno je pri uporabi YCbCr prostora (slika 10).

Pri testiranju na sekvenci 'david' je razviden vpliv posodobitve modela. Braz posodobitve slabo sledi, ker se tarča premika iz temnega v svetlo okolje - intenziteta barv se spreminja in pride do 1 napake sledilnika. Če model vsak okvir posodobimo deluje bolje, saj se histogram predloge počasi privaja na spre-



**Fig. 4.** Vizualizacija testne funkcije 2.

membe. Sledilnik še vedno enkrat odpove, vendar se to zgodi zaradi podobnosti las z temnim ozadjem, ne pa zaradi spreminjanja intenzitete barv.

**Nadgradnje** Pri sekvencah 'ball' in 'car' smo opazili, da komaj uspe slediti tarči, ker le ta spreminja svojo velikost skozi sekvenco (se oddaljuje ali približuje). Zato sem implementiral nadgradnjo omenjeno v članku[1], ki vsak okvir izračuna razdaljo z tremi različnimi velikostmi jedra (+10% večji, manjši in enak kot prejšnji okvir), izbere najboljšo velikost glede na bhattacharyjevo razdaljo ter nato z njo izvede lokalizacijo tarče.

Da ne bi bil algoritem preobčutljiv na spremembe skale, smo novo velikost utežili z prejšnjo po naslednji formuli

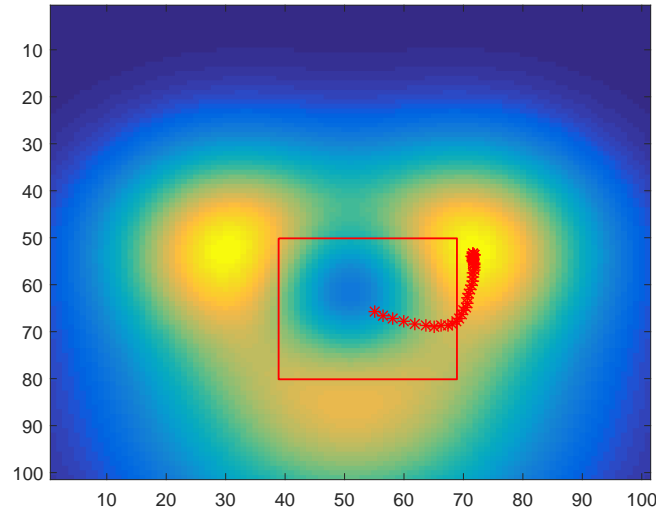
$$velikostjendraH = 0.9 * Hstari + 0.9 * Qnovi.$$

S tem smo dosegli boljše delovanje sledilnika.

Pri 'car' sekvenci se lepše prilagaja tarči (slika 11). Podobno deluje pri ostalih sekvencah.

Do težav pride, ko nadgradnjo kombiniramo s posodobitvijo modela. Tudi pri majhnemu parametru alfa (npr. 0,05), začne sledilnik v svoj model čedalje bolj vključevati ozadje. Zaradi tega nagradi, povečanje jedra in regije, ki vsebuje več ozadja. Posledično se regija povečuje, kar privede do se hujše težave, ko sledilnik v svojo regijo ujame podoben objekt - na primer drugega igralca, kar lahko vidimo na sliki 12. V takšnih primerih sledilnik deluje bolje če zelo pomanjšamo alfo (0.001) ali pa popolnoma odstranimo posodabljanje modela.

Pri sekvenci 'david' sledilnik ne odpove, vendar se brez posodobitve modela, območje sledenja skrči na lase tarče saj so najbolj podobni začetni predlogi (slika 13). Z posodobitvijo odpove enkrat ker se preveč razširi.



**Fig. 5.** Delovanje mean shift metode, na testni funkciji 2.  $e = 0.5$ ,  $h = 31$ . Za konvergenco je bilo potrebnih 39 iteracij

Vecino težav, kjer sledilnik preveč upošteva značilnosti ozadja, bi lahko odpravili z uteževanjem ozadja (opisano v članku[1]). Okoli tarče izračunamo histogram ozadja in utežimo značilnosti tarče. S tem zmanjšamo pomembnost tistih, ki so vsebovane v ozadju in povečamo tiste, ki so prisotne samo v tarči.

**Parametri** Vpliv parametrov velikosti jedra in alfe (s katerim posodabljam model) smo že opisali zgoraj pri analiziranju primerov. Še dodatno pa v nadaljevanju opišemo vpliv nekaterih drugih parametrov.

Parameter  $\epsilon$ , ki nastopa v formuli  $V = \sqrt{Q/(P + \epsilon)}$  za izračun uteži iz histogram predloge  $Q$  in regije  $P$  ima velik vpliv. Z prištevanjem majhne vrednosti kolicniku, se izognemo deljenju z nič. Poleg tega premajhna vrednost  $\epsilon$  preveč ojača vpliv zgornjega člena in vse vrednosti uteži si postanejo zelo podobne, kar lahko vidimo na sliki 14.

Če pa je prevelik nima nobenega vpliva in regija vsebuje preveč ključnih značilnosti v ozadju (slika 15 in slika 16). Z eksperimentiranjem sem ugotovil, da najboljše sledi pri vrednosti parametra  $\epsilon = 1e - 4$  (slika 17).

Manjše število košev (bins) histograma povzroči manjšo ločljivost uteži regije - več pikslov je uteženih z isto vrednostjo. Če ni pravilne velikosti, mean-shift težje najde težišče, ker potrebuje večje število iteracij. Pri skevencu 'basketball' s štirimi koši dobimo 3 odpovedi, z 64 koši pa 7 odpovedi. Pri tem velikokrat naredimo 5 ali več iteracij vsak okvir. Pri uporabi RGB prostora se je izkazalo, da je najboljša vrednost 16 (le redko vzame 5 iteracij), pri HSV in YCbCr pa 8.

Več mean shift iteracij kot dopustimo v vsakem okvirju, bolj bo natančen naš sledilnik. Ponavadi ne porabi več kot 5 iteracij, izjemoma pa ne skonvergira



**Fig. 6.** Lokalizacija tarče z osnovnim sledilnikom na testni sekvenci 'car'. Sledimo le delu avtomobila, namesto celotni tarči.

zato omejimo z največjim možnim številom iteracij 10. Mean shift algoritem konvergira, ko je izpolnjen pogoj, da je premik manjši kot 1 piksel. Natančnost manjša kot 1 piksel ne doprinese boljšega delovanja sledilnika, še posebej pri manjšem številu košev histograma.

### 3 Zaključek

Mean shift sledilnik je hiter in robusten sledilni algoritem - sama lokalizacija tarče v vsakem okvirju vzame povprečno 0.005 sekund. S prilagajanjem skali tarče, vidno izboljšamo sledenje vendar ne v kombinaciji z posodobitvijo modela. Parametri algoritma še posebej eps imajo velik vpliv na delovanje, zato jih je potrebno ustrezno nastaviti in testirati.

V nadaljevanju, bi bilo potrebno še več eksperimentirati in optimizirati parametre. Implementirali bi tudi nadgradnjo uteževanja ozadij in s tem rešili glavni problem trenutnega algoritma - prekomerno upoštevanja značilnosti ozadja, ki je odgovoren za večino odpovedi sledilnika.

### Literatura

1. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(5), 564–577 (May 2003). <https://doi.org/10.1109/TPAMI.2003.1195991>



**Fig. 7.** Lokalizacija tarče z osnovnim sledilnikom na testni sekvenci 'basketball'.



**Fig. 8.** Primer izgube tarče z osnovnim sledilnikom na testni sekvenci 'woman'.





**Fig. 9.** Težavni okvir/slika iz testne sekvenc 'woman' v HSV barvnem prostoru.



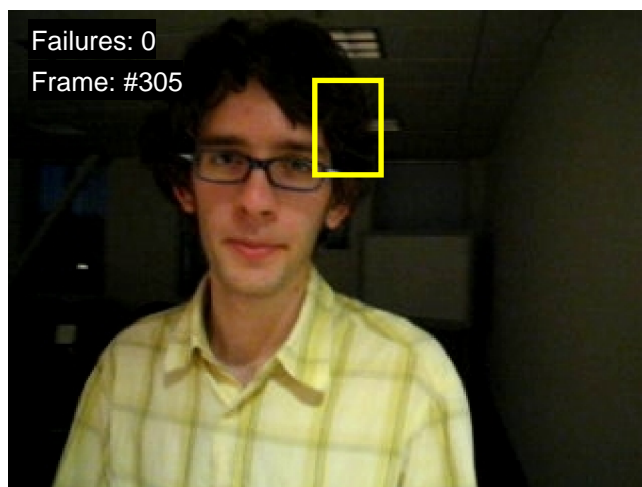
**Fig. 10.** Težavni okvir/slika iz testne sekvenc 'woman' v YCbCr barvnem prostoru.



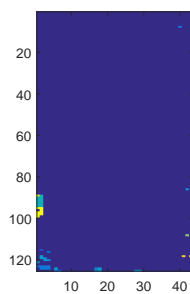
**Fig. 11.** Lokalizacija tarče z sledilnikom, ki se prilagaja velikosti tarče na testni sekvenci 'car'.



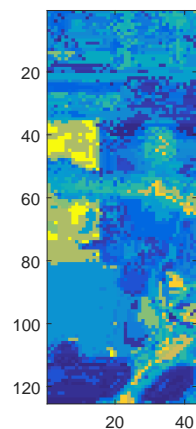
**Fig. 12.** Lokalizacija tarče z sledilnikom, ki se prilagaja velikosti tarče na testni sekvenci 'basketball'.  $\alpha = 0.05$ .



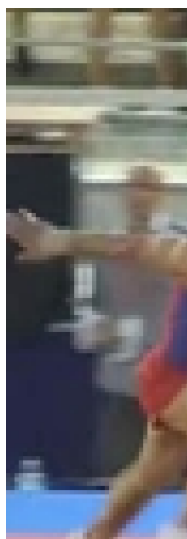
**Fig. 13.** Lokalizacija tarče z sledilnikom, ki se prilagaja velikosti tarče na testni sekvenci 'david'.  $\alpha = 0.01$ .



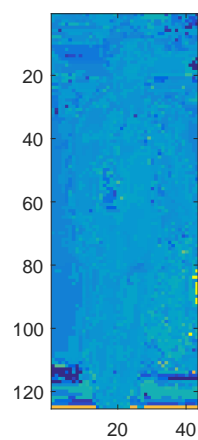
**Fig. 14.** Uteži za sliko izbrane regije pri sekvenci 'gymnastics'.  $\text{Eps} = 1\text{e-}10$ .



**Fig. 15.** Uteži za sliko16 izbrane regije pri sekvenci 'gymnastics'.  $Eps = 1e-1$ . Vidno je napačno osredotočanje na značilnosti ozadja namesto tarče na desni strani.



**Fig. 16.** Slika izbrane regije pri sekvenci 'gymnastics'. Sledilnik je skoraj izgubil tarčo na desni strani.



**Fig. 17.** Uteži za sliko izbrane regije pri sekvenci 'gymnastics'.  $\text{Eps} = 1\text{e-}4$ .