

# Mean shift sledilnik s Kalmanovim filtrom

Ivan Antesič

Fakulteta za računalništvo in informatiko,  
Univerza v Ljubljani,  
Večna pot 113, Ljubljana  
E-pošta: ivan.v.antesic@gmail.com

## Povzetek

Izpeljali smo Kalmanov filter in pokazali njegovo delovanje na praktičnem primeru sledenja tarči z Mean shift sledilnikom. Da bi izboljšali delovanje in preprečili odpovedi pri zakritju tarče smo sledilniku dodali Kalmanov filter. S testiranjem različnih modelov gibanja in parametrov smo ugotovili, da lahko z dinamičnim modelom izboljšamo delovanje pri specifičnih primerih.

## 1 Uvod

Kalmanov filter je algoritem poimenovan po Rudolfu E. Kálmánu, ki za določitev stanja tarče uporablja Bayesove principe verjentosti. S kombiniranjem meritve in predikcije zgladi morebitno šumne meritve ter tako bolj natančno določi naslednje stanje. Pogosto se uporablja se pri navigaciji, kontroli vozil, vodenju procesov, procesiranju signalov in ekonometriki. V okviru seminarške naloge smo Kalmanov filter uporabili na problemu računalniškega vida in v Mean shift sledilnik vpeljali Kalmanov filter.

Mean shift sledilnik, v večini primerov deluje razmeroma dobro. Barvni vizuelni model histograma je robusten in neobčutljiv na različne transformacije tarče. Še vedno pa ima sledilnik težave z delno ali popolno zakritostjo tarče (angl. occlusion) ter zamenjavo tarče za podobno obarvan predmet v bližini. Te težave, bi lahko ublažili z uvedbo modela gibanja. Zato smo se odločili, da nadgradimo sledilnik s Kalmanovim filtrom.

Kalmanov filter omogoča predvidenje dobre začetne točke Mean shift algoritma, kar se odraža v boljši lokализaciji - laže dosežemo globalni maksimum. Ko so meritve Mean shifta nezanesljive (zakrita tarča ali napačna zaznavna), zaradi upoštevanja dinamičnega modela predvidimo kje se v resnici nahaja tarča in tako ohranimo sledenje. Sledenje je manj šumno, saj se povprečijo meritve Mean shift sledilnika in napovedi dinamičnega modela.

## 2 Kalmanov filter

Delovanje in teorija Kalmanovega filtra je opisana v knjigi [1] s katero smo si pomagali pri razumevanju in implementiraju algoritma.

Definirajmo spremenljivki stanja  $w_t$  za dinamični model in meritve  $x_t$  za model meritve.

### 2.1 Dinamični model

Dinamični model definira povezavo stanja  $w_t$  ob času  $t$  s stanjem ob času  $t-1$  kot:

$$w_t = \Psi * w_{t-1} + \epsilon_p$$

Stanje smo predstavili z vektorjem x in y komponent:

- položaja
- hitrosti (časovnim odvodom položaja)
- pospeška (časovnim odvodom hitrosti)
- pri modelu NCA še dodatno s sunkom (časovnim odvodom pospeška)

$$w_t = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix}, \dot{w}_t = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \\ \dddot{x} \\ \dddot{y} \end{bmatrix}$$

$\Psi$  je matrika, ki določi katere komponente stanja upoštevamo pri predikciji in je odvisna od modela gibanja, ki ga uporabljamo.  $\epsilon_p$  je naključna spremenljivka, ki predstavlja šum določen z matriko kovariance šuma dinamičnega modela  $\Sigma_p$ .

Da lahko določimo  $\Psi$  je torej potrebno najprej definirati model gibanja, ki opisuje gibanje po formuli  $\dot{w} = F * w + L * \epsilon$ . Za vsak model določimo matriki  $F$  (katere komponente upoštevamo) in  $L$  (katere komponente so šum).

#### 2.1.1 RW

Pri modelu naključne hoje (angl. random walk) je hitrost definirana le s šumom, pospeška in sunka ne upoštevamo. Model je primeren za opis hitrih nenadnih in nepričakovanih sprememb.

$$\dot{w}_t = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix}, F = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

### 2.1.2 NCV

Model skoraj konstantne hitrosti (angl. near constant velocity) opisuje gibanje, ki ima konstantno hitrost in pospešek definiran s šumom. Primeren za objekte, ki se premikajo brez pospeška (npr. avto, ki vozi z isto hitrostjo).

$$\dot{w}_t = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix}, F = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

### 2.1.3 NCA

Model skoraj konstantnega pospeška (angl. near constant acceleration) je predstavljen s konstantno hitrostjo in pospeškom, nenadni sunki pa so definirani s šumom. Primereno za objekte, ki postopoma pospešujejo ali zavirajo svoje gibanje (npr. vzlet letala).

$$\dot{w}_t = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix}, F = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Sedaj lahko določimo matriko  $\Psi$  po enačbi:  $\Psi = e^{L * \Delta t}$ . Enačbo smo izračunali, kar z uporabo Matlab paketa *Symbolic toolbox*. S paketom izračunamo tudi kovariančno matriko  $\Sigma_p = \int_0^{\Delta t} (\Psi(\xi)L)k_q(\Psi(\xi)L)^T d\xi$ . Za npr. NCV model in  $\Delta t = 1$  bi bili matriki enaki:

$$\Psi = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \Sigma_p = k_q * \begin{bmatrix} 1/3 & 0 & 1/2 & 0 \\ 0 & 1/3 & 0 & 1/2 \\ 1/2 & 0 & 1 & 0 \\ 0 & 1/2 & 0 & 1 \end{bmatrix}$$

$k_q$  je parameter s katerim določimo velikost  $\Sigma_p$ .

## 2.2 Model meritev

$x_t$  je meritev, ki jo vrne Mean shift sledilnik, definirana kot

$$x_t = \Phi * w_t + \epsilon_m$$

V našem primeru meritev vsebuje le položaj v x in y koordinatah.

$$x_t = \begin{bmatrix} x \\ y \end{bmatrix}$$

Matrika  $\Phi$  iz stanja  $w_t$  izloči položaj tarče,  $\epsilon$  pa predstavlja šum določen z matriko kovariance šuma meritvenega modela  $\Sigma_m$ . Za npr. NCV model, bi bili matriki enaki:

$$\Phi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \Sigma_m = k_r * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$k_r$  je parameter s katerim določimo velikost  $\Sigma_m$ .

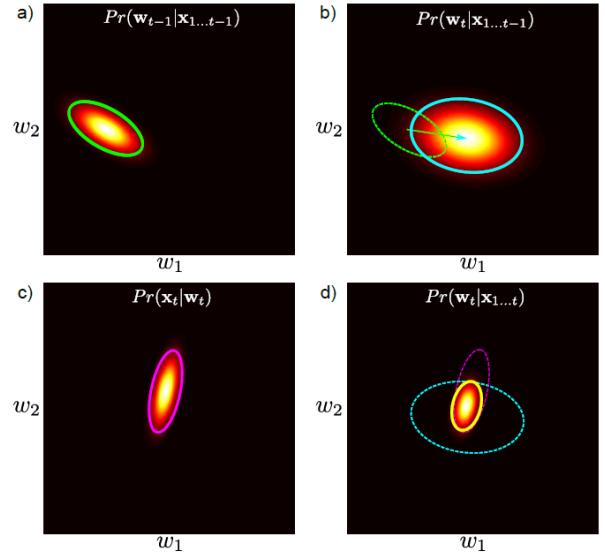
## 2.3 Inferenca

Kalmanov filter po naslednji formuli definira verjetnost, da smo prišli v stanje  $w_t$  glede na meritve  $x_{1:t}$ :

$$p(w_t | x_{1:t}) \propto p(x_t | w_t) \int p(w_t | w_{t-1}) p(w_{t-1} | x_{1:t-1}) dw_{t-1}$$

Iz prejšnjega stanja (previous posterior)  $p(w_{t-1} | x_{1:t-1})$  z modelom gibanja  $p(w_t | w_{t-1})$  naredi predikcijo stanja (prior)  $p(w_t | x_{1:t-1})$ . To nato združimo z meritvijo  $p(x_t | w_t)$  v novo stanje (posterior). V naslednji iteraciji postane posterior prejšnje stanje. Postopek je vizuelno predstavljen na sliki 1.

Kalmanov filter predpostavi, da so vse verjetnosti formule predstavljene s funkcijo Gaussove porazdelitve in s tem upošteva negotovost stanj tarče. Tako je npr. meritev predstavljena kot:  $p(x_t | w_t) = G(\mu_m = \psi * w_t, \Sigma_m)$ .



Slika 1: Slika prikazuje delovanje Kalmanovega filtra. V a.) vidimo prejšnje stanje, iz katerega predvidimo prior v b.). c.) je meritev, ki skupaj z b.) določa novo stanje v d.).

Nov posterior izračunamo z Bayesovim pravilom verjetnosti:

$$\begin{aligned} p(w_t | x_{1:t}) &= \frac{p(x_t | w_t)p(w_t | x_{1:t-1})}{p(x_{1:t})} \\ &= \frac{G(\mu_m + \Phi w_t, \Sigma_m)G(\mu_+, \Sigma_+)}{p(x_{1:t})} \\ &= G[(\Phi^T \Sigma_m^{-1} \Phi + \Sigma_+^{-1})^{-1} \\ &\quad ((\Phi^T \Sigma_m^{-1} (x_t - \mu_m) + \Sigma_+^{-1} \mu_+)^{-1}), \\ &\quad (\Phi^T \Sigma_m^{-1} \Phi + \Sigma_+^{-1})^{-1}] \\ &= G(\mu_t, \Sigma_t) \end{aligned}$$

$\Sigma_+$  in  $\mu_+$  sta varianca in povprečje stanja predikcije (prior). Ker je računanje novega stanja v tej obliki numerično zahtevno, preoblikujemo enačbo in definiramo Kalmanovo pridobitev.

## 2.4 Kalmanova pridobitev

S Kalmanovo pridobitvijo

$$K = \Sigma_+ \Phi^T (\Sigma_m + \Phi \Sigma_+ \Phi^T)^{-1}$$

lahko lažje določimo povprečje  $\mu_t$  in varianco  $\Sigma_t$  za novo stanje (posterior)  $p(w_t | x_{1:t}) = G(\mu_t, \Sigma_t)$ .

$$\begin{aligned}\mu_+ &= \mu_p + \Psi \mu_{t-1} \\ \Sigma_+ &= \Sigma_p + \Psi \Sigma_{t-1} \Psi^T \\ \mu_t &= \mu_+ + K(x_t - \mu_m - \Phi \mu_+) \\ \Sigma_t &= (I - K\Phi) \Sigma_+\end{aligned}$$

Če je Kalmanova pridobitev velika pomeni, da so meritve bolj zanesljive, kot predikcija in jih bolj upoštevamo. To lahko vidimo tudi iz enačbe za varianco: ko so meritve bolj zanesljive se varianca zmanjša. Kalmanov filter si lahko predstavljamo kot zelo dobro uteženo povprečje meritve in napovedi modela.

Na delovanje Kalmanovega filtra ponavadi vplivamo z nastavljivo parametra  $k_q$  s katerim določamo velikost kovariančne matrike za dinamični model  $\Sigma_p$  in s parametrom  $k_r$  s katerim določamo velikost kovariančne matrike za meritveni model  $\Sigma_m$ . Večji  $k_q$  bo povečal šum dinamičnega modela in predikcija se bo upoštevala manj. Enako velja za  $k_r$  in šum meritov.

## 2.5 Možne izboljšave

Kalmanov filter deluje dobro dokler model gibanja linearen in imamo le eno hipotezo (Gaussovo funkcijo z enim vrhom), kje se nahaja naše novo stanje. Ker se v praksi velikokrat želimo opisati nelinearna gibanja, na primer. v avtomatiki sisteme, ki poleg položaja in hitrosti upoštevajo še maso, zračni upor, kotno hitrost, ipd., potrebujemo rešiti te probleme z razširitvijo običajnega algoritma.

V našem primeru sledenja tarči skozi sekvenco slik nimamo potrebe po opisu nelinearnega gibanja. Posamezni premiki med zaporednimi slikami so zelo majhni, sistem sledilnika ni kompleksen poleg tega pa ne moremo določiti nelinearni model gibanja, ki bi dobro opisal vse sekvence (saj so si zelo različne). Kljub temu omenimo glavne ideje posameznih izboljšav.

### 2.5.1 Razširjeni Kalmanov filter

Pri razširjenemu Kalmanovem filtru (angl. Extended Kalman filter) lahko opišemo poljubni nelinearni funkciji  $f$  in  $g$ , ki določata stanje in meritev v času  $t$ :

$$w_t = f(w_{t-1}, \epsilon_p), x_t = g(w_t, \epsilon_m)$$

To je možno, ker funkcijo linearno aproksimiramo s Taylorjevim razvojem okoli njenega vrha v  $\mu_t$ . Dodatno lahko funkcijo zgladimo s ponavljanjem postopka na novih podatkih. Ta pristop imenujemo iterativno razširjeni Kalmanov filter.

### 2.5.2 Neodišavljeni Kalmanov filter

Z neodišavljenim Kalmanovim filtrom (angl. unscented Kalman filter) lahko opišemo bolj nelinearna gibanja kot z raširjeno obliko. Vendar je potrebno, da je šum gibanja porazdeljen normalno (Gauss).

$$w_t = f(w_{t-1}) + \epsilon_p, x_t = g(w_t) + \epsilon_m$$

Poleg tega postopek ne vsebuje računanja odvodov. Porazdelitve predstavimo z množico sigma točk (Diracovih

delta funkcij), ki jih preslikamo z nelinearno funkcijo. Na preslikani množici nato določimo povprečje in kovarianco za normalno porazdelitev.

### 2.5.3 Filter z delci

Kalmanov filter predpostavi da je verjetnostna funkcija Gaussove oblike. Velikokrat to ne drži, lahko dobimo težjo funkcijo z večimi vrhi. V teh primerih posterior opišemo kot model večih Gaussov (mixture model). Ker ne vemo vnaprej koliko Gaussovih funkcij potrebujemo, iz verjetnostne funkcije vzorčimo veliko Diracovih funkcij. Vzorec vsebuje svojo hipotezo za stanje in utež. Integriranje nadomestimo z Monte Carlo integracijo ter z računanjem uteži vzorcev dosežemo enako (ampak lažje) vzorčenje težkih funkcij.

## 3 Mean shift sledilnik

Za lažje razumevanje primera, na kratko opišimo tudi sledilnik, ki smo ga nadgradili s Kalmanovim filtrom. Pri implementaciji smo se zgledovali po članku [2], ki opisuje podoben pristop nadgradnje Mean shift algoritma s prilagajanjem skale in uteževanjem ozadja.

Mean shift je iterativna metoda za iskanje maksima verjetnostne funkcije, uporablja pa se lahko tudi za gručenje točk. Vsako iteracijo se malo pomaknemo v težišče sodnjih 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 "vzepanje" po funkciji.

Mean shift (MS) sledilnik deluje na modelu barve. S primerjanjem barvnih histogramov regij  $P$ , poiščemo tisto, ki je najbolj podobna histogramu začetne predloge  $Q$  in tako lokaliziramo tarčo. Ker so histogrami normalizirani, jih lahko obravnavamo kot porazdelitveno funkcijo in za določitev podobnosti uporabimo Bhattacharya koeficient

$$BC(P, Q) = \sum_{x \in X} \sqrt{P(x) * Q(x)}.$$

Z njim lahko nato določimo 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. V implementaciji smo uporabili epanechnikovo jedro, saj se pri odvodu poenostavi v običajno seštevanje in ga lahko izpustimo iz omenjenih enacb. Uporabili smo tudi posodabljanje modela začetne predloge  $Q$  v vsaki sliki (angl. frame) po formuli:

$$Q = (1 - \alpha) * Q_{star} + \alpha * Q_{nov}.$$

Od parametra  $\alpha$  je torej odvisno kako hitro se bo predloga prilagodila spremembam tarče in tako lažje sle-

dila skozi sekvenco. Da se model predloge ne bi napačno prilagodil na regijo, v kateri tarča ni dobro prisotna se alfa določi dinamično glede na podobnost regije s predlogo:

$$\alpha = \alpha_{Max} * (1 - BC(P, Q)).$$

Pri testiranju smo za parameter  $\alpha_{Max}$  določili vrednost 0,005.

### 3.1 Prilagajanje skale

Za prilagajanje skale smo implementirali preprost algoritmom. Vsako sliko (angl. frame) lokaliziramo tarčo s tremi različnimi velikostmi regije (prejšnja velikost, za 10% večja in manjša velikost) in izberemo najboljšo glede na podobnost histograma tarče s histogramom predloge.

Da ne bi bil algoritem preobčutljiv na spremembe velikosti, smo novo velikost utežili s prejšnjo po naslednji formuli:

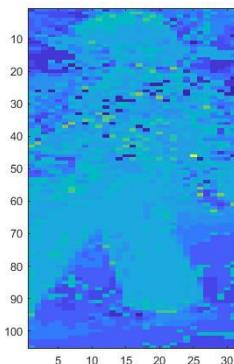
$$velikostH = (1 - gama) * Hstar + gama * Hnovi.$$

V robnih primerih kjer se tarča premakne ali poveča izven meja slike ohranimo dimenzije, le da izven ležeče piksele utežimo z vrednostjo 0 pri Epanechnicovem jedru, tako da jih ne upoštevamo pri lokalizaciji.

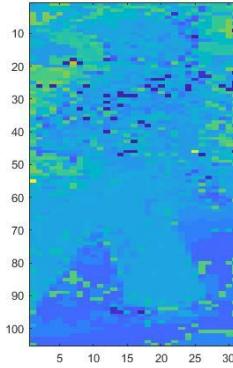
### 3.2 Upoštevanje ozadja

V večini sekvenč tarča ne obsega celotne regije v kateri se nahaja, zato so v vizuelnem modelu prisotne tudi barve, ki se nahajo v ozadju tarče. Če se ozadje tarče spreminja se lahko zgodi, da naš sledilnik odpove, ker je preveč osredotočen na značilnice ozadja, da bi upošteval premik tarče. Tudi posodabljanje modela ne pomaga dosti saj se zaradi upoštevanja ozadja pri skaliranju regija postopoma povečuje dokler ne zajeme celotnega ozadja.

Zato iz območja 10 piksov okoli regije tarče generiramo histogram ozadja ter ga delimo z vrednostjo najbolj prisotne barve. Tako ima najmanj prisotna barva največjo vrednost, najbolj prisotna pa najmanjšo. Histogram ozadja uporabimo kot utež za histogram tarče ter tako dosežemo, da se barve v ozadju med sledenjem upošteva čim manj. Rezultat uteževanja s histogramom ozadja je viden na sliki 2 v primerjavi s sliko 3.



Slika 2: S histogramom ozadja utežena regija tarče po kriteriju Mean shift koraki iščemo maksimum. V primerjavi s sliko 3 je tarča bolj razločna. Sekvenca 'woman'.



Slika 3: Neutežena regija tarče po kriteriju Mean shift koraki iščemo maksimum. Sekvenca 'woman'.

Tudi histogram ozadja B skozi sekvenco postopoma posodabljamо glede na podobnost regije s predlogo:

$$B = (1 - beta) * Bstar + beta * Bnovi,$$

kjer je beta določen kot :  $beta = alpha * 10$ .

### 3.3 Parametra Kalmanovega filtra

Kot smo omenili v poglavju 2.4 na delovanje vplivamo s parametrom  $k_q$  in  $k_r$ .  $k_q$  smo določili v inicializaciji glede na delež slike, ki ga pokriva tarča. To je potrebno, da je velikost kovariance razmeroma enaka v vseh sekvenkah glede na velikost tarče:

$$k_q = (wTarget * hTarget) / (wFrame * hFrame) * delta.$$

Pri modelih NCV in NCA ima delta vrednost 5, pri modelu RW pa 50, ker je velikost variance manjša pri RW. Pri RW namreč upoštevamo le položaj tarče.

Vrednost  $k_r$  določimo v vsaki sliki glede na podobnost histograma izbrane regije s predlogo  $k_r = omega * (1 - simil)$ . Omega je faktor odvisen od velikosti tarče:

$$omega = wTarget * hTarget / (wFrame * hFrame) * 2000.$$

Če se regija tarče razteza se mora povečati tudi kovarianca, saj lahko zgrešimo za več piksov kot pa če je tarča majhna. Poleg tega nam to pomaga, ko se regija raztegne zaradi bližine tarči podobnega objekta. Takrat je koristno zmanjšati vpliv meritve in zaupati dinamičnemu modelu. Množenje z 2000 poveča vpliv raztezanja na kovarianco.

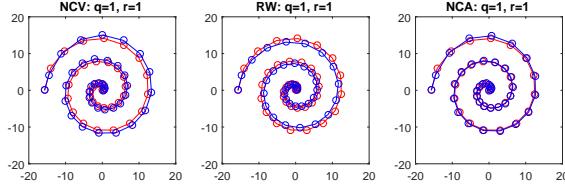
## 4 Testiranje

### 4.1 Glajenje funkcije

Najprej bomo testirali sam Kalmanov filter na primeru glajenja točk funkcije. Spodaj so predstavljeni rezultati testiranja vpliva različnih modelov gibanja in vrednosti parametrov  $k_q$ ,  $k_r$  na Kalmanov filter. Za časovni korak  $T$  smo predpostavili, da ima vedno vrednost 1.

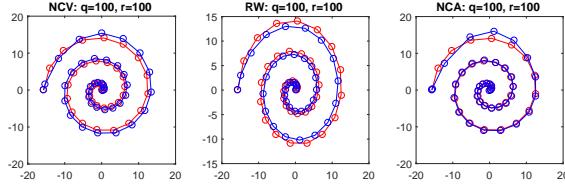
V primeru spiralne krivulje s 40 točkami in vrednosti parametra  $k_q = k_r = 1$  je vpliv šuma majhen tako na meritve kot na model gibanja. Na sliki 4 lahko vidimo vpliv modela gibanja na Kalmanov filter. Razvidno je da model

NCV malo zaostaja za meritvami, saj ne upošteva dovolj pospeška (krivulja meritov se postopoma zmanjšuje).



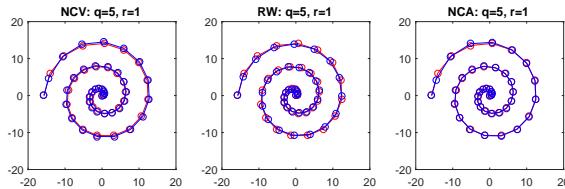
Slika 4: Glajenje meritov (rdeča krivulja) s Kalmanovim filtrom (modra krivulja) pri modelih gibanja RW, NCV in NCA ter parametrom  $k_q = 1, r = 1$ .

RW model upošteva le položaj in zato le približno sledi krivulji meritov. NCA na začetku sledi podobno kot NCV, kasneje pa se zaradi upoštevanja pospeška lepše prilagodi krivulji meritov. Enak graf dobimo za vse vrednosti pri katerih sta  $k_q$  in  $k_r$  enaka - Kalmanov filter za napovedovanje enakovredno upošteva meritve in model gibanja, saj je varianca (posledično šum) pri obeh ista (slika 5).



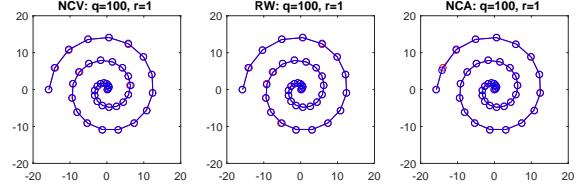
Slika 5: Glajenje meritov (rdeča krivulja) s Kalmanovim filtrom (modra krivulja) pri modelih gibanja RW, NCV in NCA ter parametrom  $k_q = 100, k_r = 100$ .

Če  $k_r$  pustimo majnen, povečamo pa vrednost parameta  $k_q$  in s tem šum modela gibanja opazimo (na sliki 6), da se krivulje modelov obnošajo enako kot prej, le da se sedaj bolj prilegajo krivulji meritov. Če efekt pojačamo in vrednost  $k_q$  močno povečamo, lahko na sliki 7 vidimo, da se Kalmanove krivulje popolnoma ujemajo z meritvami - vpliv modela je izničen. Takšno delovanje je posledica velikega šuma modela, zaradi katerega pri filtriranju upoštevamo le meritve - vrednosti modela se preveč razlikujejo od meritov, da bi jih še upoštevali.



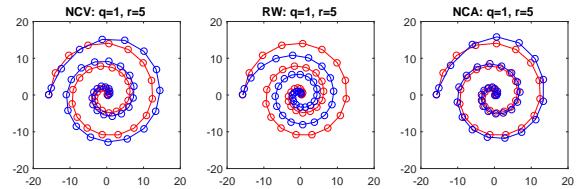
Slika 6: Glajenje meritov (rdeča krivulja) s Kalmanovim filtrom (modra krivulja) pri modelih gibanja RW, NCV in NCA ter parametrom  $k_q = 5, k_r = 1$ .

Obratno delovanje lahko opazimo na sliki 8 in 9. Vrednosti meritov so preveč šumne v primerjavi z napovedmi, zato se upošteva bolj model gibanja. NCA in NCV še

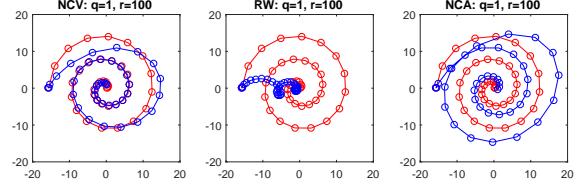


Slika 7: Glajenje meritov (rdeča krivulja) s Kalmanovim filtrom (modra krivulja) pri modelih gibanja RW, NCV in NCA ter parametrom  $k_q = 100, k_r = 1$ .

vedno približno ohranita obliko spirale (NCV bolje kot NCA saj pospešek NCA pokvari obliko), medtem ko se model RW popolnoma izrodi, ker je večinsko odvisen le od položaja.



Slika 8: Glajenje meritov (rdeča krivulja) s Kalmanovim filtrom (modra krivulja) pri modelih gibanja RW, NCV in NCA ter parametrom  $k_q = 1, k_r = 5$ .



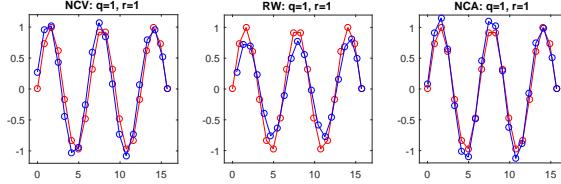
Slika 9: Glajenje meritov (rdeča krivulja) s Kalmanovim filtrom (modra krivulja) pri modelih gibanja RW, NCV in NCA ter parametrom  $k_q = 1, k_r = 100$ .

Pri sinusni krivulji s polovicno številom točk ( $N = 20$ ) je delovanje podobno, le da se Kalmanova krivulja bolje prilega meritvam, saj v primerjavi s spiralo, ne spreminja konstantno smeri (slika 10). Na sliki 11 lahko vidimo upoštevanje samo dinamičnega modela. NCV zgladi meritve v bolj položno in manjšo krivuljo, RW preveč zgladi in se izroji, NCA pa se približno prilagodi meritvam.

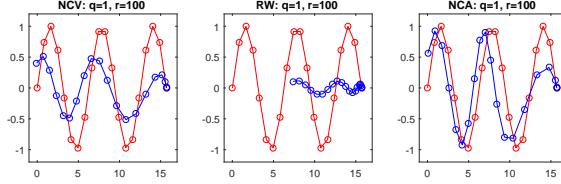
Pri testiranju na gaussovi krivulji, z 20 točkami, lahko lepo vidimo vpliv modelov gibanja (slika 12). Model RW zgladi meritve krivulje in se ustavi pravočasno, ko ta pada na desni strani. NCV zaradi upoštevanja hitrosti nadaljuje glajenje čez spodnjo mejo krivulje preden se vrne v ravnovesno lego. Model NCA zaradi upoštevanja pospeška, nadaljuje glajenje čez spodnjo mejo ter nato preveč narašte preden doseže ravnovesno lego (naredi nekakšno nihanje).

## 4.2 Sledenje

Kalmanov filter smo na primeru sledenja testirali z različnimi modeli gibanja. Testi so bili izvedeni na testni sekvencah



Slika 10: Glajenje meritev (rdeča krivulja) s Kalmanovim filtrom (modra krivulja) pri modelih gibanja RW, NCV in NCA ter parametrom  $k_q = 1, k_r = 1$ .



Slika 11: Glajenje meritev (rdeča krivulja) s Kalmanovim filtrom (modra krivulja) pri modelih gibanja RW, NCV in NCA ter parametrom  $k_q = 1, k_r = 100$ .

slik vot2014 (<http://box.vicos.si/vot/vot2014.zip>) z uporabo ogrodja Tracking Evaluation Toolkit (<https://github.com/alanlukezic/tracking-toolkit-lite>). Za ocenjevanje delovanja sta definirani meri prekrite (angl. overlap) - kolikšen delež tarče pokriva naš sledilnik skozi sekvenco in število odpovedi (angl. failures) - kolikokrat naš sledilnik odpove, ker izgubi tarčo in ga je potrebno reinicializirati.

#### 4.2.1 Brez dinamičnega modela

Najprej smo testirali delovanje Mean shift (MS) sledilnika brez Kalmanovega filtra, da lahko v nadaljevanju primerjamo delovanje in vidimo, kje običajni sledilnik deluje boljše in slabše.

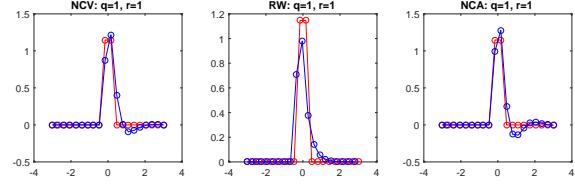
V tabeli 1 lahko razberemo delovanje slednilnika. V večini primerov se obnese dobro. Brez odpovedi prenese tudi sekvence kot je npr. 'sunshade', kjer je veliko hitrih gibov. Odpove pa v primerih, kjer pride do zakritja tarče (slika 13 ali ob kratkemu zaporedju napačnih zaznav (slika 14) ter bližini tarči podobnega predmeta (slika 15).

#### 4.2.2 RW model

Kalmanov filter z modelom gibanja RW (random walk) deluje kot nekakšna zakasnitev. RW model namreč pri predvidevanju ne upošteva hitrosti ali pospeška ampak samo položaj. Tako za predikcijo izbere lokacijo prejšnje slike (angl. frame) s kovarianco določeno s parametrom  $k_q$ .

Važno je, da je kovarianca dinamičnega modela malo manjša od kovariance meritev, tako da ob pravilnih meritvah le malo upošteva predikcijo. Ko algoritem zazna negotovost meritev se kovarianca meritev poveča in upoštevam predikcijo.

V tabeli 2 lahko razberemo delovanje slednilnika z RW modelom. Na splošno deluje slabše saj pri sekvenkah s hitrimi premiki, zaradi predikcije preveč zaostaja



Slika 12: Glajenje meritev (rdeča krivulja) s Kalmanovim filtrom (modra krivulja) pri modelih gibanja RW, NCV in NCA ter parametrom  $k_q = 1, k_r = 1$ .

Sequence	Overlap	Failures
ball	0.75	0
basketball	0.63	2
bicycle	0.40	0
bolt	0.50	2
car	0.57	0
david	0.29	1
diving	0.37	0
drunk	0.53	0
fernando	0.39	1
fish1	0.18	3
fish2	0.36	1
gymnastics	0.62	0
hand1	0.48	0
hand2	0.49	4
jogging	0.68	2
motocross	0.48	4
polarbear	0.58	0
skating	0.42	1
sphere	0.51	0
sunshade	0.56	0
surfing	0.52	0
torus	0.52	1
trellis	0.42	1
tunnel	0.43	6
woman	0.46	1
<b>Average</b>	0.49	1.20

Tabela 1: Rezultati testiranja sledilnika brez modela gibanja na testnih sekvenkah vot2014.

in izgubi tarčo (slika 16). Zelo dobro pa se obnese v sekvenci 'jogging', ki vsebuje pokritje - dovolj dolgo ohranja prejšnje stanje, da se tarča pojavi spet pojavlji, meritev postanejo zanesljive in lahko sledimo naprej (slike 17, 18, 19).

#### 4.2.3 NCV model

Za razliko od RW modela, NCV pri predikciji upošteva še hitrost ter pospešek in ima zato večjo kovarianco predikcije. Zmožen je za nekaj sličic tudi nadaljevati pot tarče. To je koristno pri sekvenkah, kot je 'motocross', kjer osnovni MS sledilnik izgubi tarčo zaradi nenačnega napačnega zaznavanja, z NCV modelom pa zaradi predikcije uspe slediti premikanju tarče (slika 20).

V tabeli 3 lahko razberemo delovanje slednilnika z



Slika 13: Primer zakritja tarče za drogom in posledične odpovedi sledilnika brez dinamičnega modela. Sekvenca 'jogging'.



Slika 14: Sledilnik brez dinamičnega modela je v prikazani sliki napačno zaznal tarčo, kar privede do odpovedi. Sekvenca 'motocross'.

NCV modelom. Delovanje je malo boljše od RW modela, ker ne ohranja le prejšnjega položaja. Zaradi tega pa tudi ni zmožen prenesti popolnega prekritja tarče pri sekvenci 'jogging'. Malo slabši je tudi od osnovnega MS sledilnika, ker zaradi večje velikosti kovariance ni zmožen dovolj popraviti slabe lastnosti sledilnika, hkrati pa upošteva malo varianco dovolj da pokvari hitre nenehne premike.

#### 4.2.4 NCA model

Sledilnik z NCA modelom upošteva hitrost, pospešek in sunke ter ima tako največjo varianco predikcije. V večini primerov upošteva meritve in le malo popravlja lokalizacijo. Zaradi tega lažje sledi hitrim premikom v primerjavi z NCV in RW ter ni zmožen tako dobro popraviti pokritosti.

Kljub temu lahko v tabeli 4 vidimo, da ima najmanj odpovedi. To je predvsem zato, ker se obnaša zelo podobno kot osnovni MS sledilnik, v nekaterih primerih pa ga le malo popravi, ko je kovarianca meritve velika (slika 21).



Slika 15: Sledilnik brez dinamičnega modela zaradi bližine tarči podobnega objekta odpove. Sekvenca 'hand2'.

Sequence	Overlap	Failures
ball	0.69	1
basketball	0.61	2
bicycle	0.33	0
bolt	0.46	2
car	0.52	0
david	0.40	1
diving	0.38	0
drunk	0.53	0
fernando	0.36	1
fish1	0.15	2
fish2	0.33	1
gymnastics	0.62	0
hand1	0.41	4
hand2	0.46	7
jogging	0.59	0
motocross	0.31	1
polarbear	0.59	0
skating	0.43	1
sphere	0.43	0
sunshade	0.48	6
surfing	0.46	0
torus	0.41	1
trellis	0.41	1
tunnel	0.36	4
woman	0.43	1
<b>Average</b>	<b>0.45</b>	<b>1.44</b>

Tabela 2: Rezultati testiranja sledilnika z RW modelom gibanja na testnih sekvenkah vot2014.

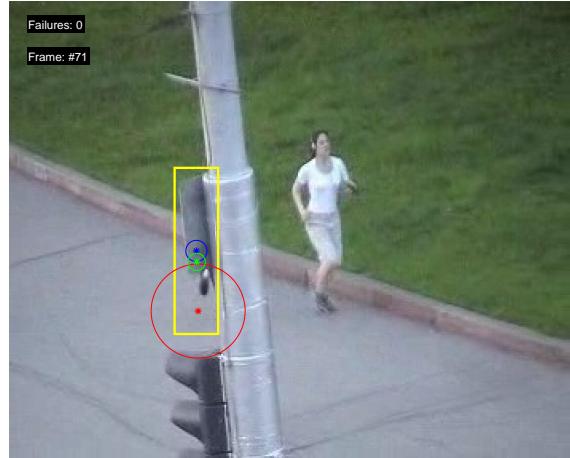
## 5 Zaključek

Izpeljali in implementirali smo Kalmanov filter ter na praktičnem primeru preizkusili njegovo delovanje in s tem potrdili naše razumevanje teorije.

Dobro implementiran Mean shift sledilnik, z upoštevanjem ozadja in prilagajanjem skale se že sam obnese razmeroma dobro v večini sekvenc. Z dodajanjem Kalmanovega filtra smo poskusili izboljšati še tiste primere v ka-



Slika 16: Sledilnik z modelom gibanja RW zaradi velikega premika neuspešno lokalizira tarčo. Modri krog prikazuje velikost variance (negotovosti) predikcije, rdeči varianco meritve, zeleni pa končno stanje in njegovo varianco. Sekvenca 'ball'.



Slika 18: Nadaljevanje slike 17: delovanje sledilnika z modelom gibanja RW na sekvenci 'jogging'. Modri krog prikazuje velikost variance (negotovost) predikcije, rdeči varianco meritve, zeleni pa končno stanje in njegovo varianco. Tarča je prekrita z večjo gotovostjo upoštevamo predikcijo.



Slika 17: Delovanje sledilnika z modelom gibanja RW na sekvenci 'jogging'. Modri krog prikazuje velikost variance (negotovost) predikcije, rdeči varianco meritve, zeleni pa končno stanje in njegovo varianco. Ker je regija dobro opisuje tarčo je negotovost vseh členov majhna.

terih odpove - predvsem zakritost tarče in nenađene slabe detekcije.

Pri testiranju modelov RW, NCV in NCA smo ugotovili, da će dovolj upoštevamo dinamični model, lahko popravimo odpovedi v željenih primerih ampak pri tem lahko pokvarimo tudi že prej delujoče primere. Še posebno tiste pri katerih se tarča hitro premika.

Zato se je najbolje obnesel NCA model, ki kljub temu, da ne popravi (popolnoma) težkih primerov, s svojo predikcijo malo izboljša osnovni algoritem. Primerjavo vseh modelov lahko vidimo na sliki 22 in tabeli 5.

## Literatura

- [1] S. Prince, *Computer vision: models, learning and inference*. Cambridge University Press, 2012. (<http://web4.cs.ucl.ac.uk/staff/s.prince/book/book.pdf>)



Slika 19: Nadaljevanje slike 18: delovanje sledilnika z modelom gibanja RW na sekvenci 'jogging'. Modri krog prikazuje velikost variance (negotovost) predikcije, rdeči varianco meritve, zeleni pa končno stanje in njegovo varianco. Tarča je prišla preko ovire zato nadaljujemo z večjim upoštevanjem meritve.

- [2] D. Comaniciu, V. Ramesh, P. Meer *Kernel-based object tracking*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003



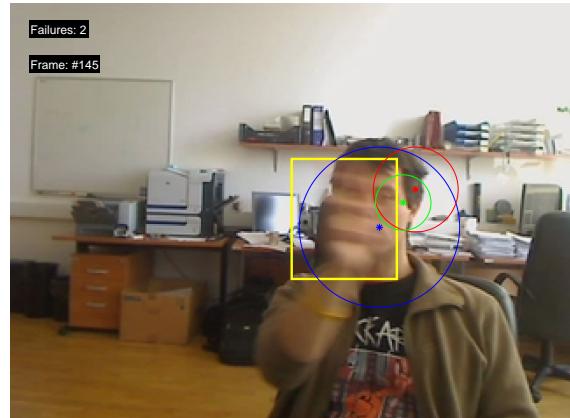
Slika 20: Sledilnik z modelom gibanja NCV zaradi upoštevanja predikcije uspešno lokalizira tarčo, ki nadaljuje svoj pot. Modri krog prikazuje velikost variance (negotovosti) predikcije, rdeči varianco meritve, zeleni pa končno stanje in njegovo varianco. Sekvenca 'motocross'.

MSNCV		
Sequence	Overlap	Failures
ball	0.73	0
basketball	0.63	2
bicycle	0.44	1
bolt	0.44	2
car	0.58	0
david	0.42	1
diving	0.37	0
drunk	0.53	0
fernando	0.37	1
fish1	0.27	3
fish2	0.36	1
gymnastics	0.59	1
hand1	0.49	0
hand2	0.31	4
jogging	0.48	1
motocross	0.40	3
polarbear	0.56	0
skating	0.40	1
sphere	0.44	0
sunshade	0.46	3
surfing	0.56	0
torus	0.50	1
trellis	0.35	1
tunnel	0.34	6
woman	0.45	1
<b>Average</b>	0.46	1.32

Tabela 3: Rezultati testiranja sledilnika z NCV modelom gibanja na testnih sekvencah vot2014.

MSNCA		
Sequence	Overlap	Failures
ball	0.75	0
basketball	0.63	2
bicycle	0.38	0
bolt	0.39	2
car	0.57	0
david	0.31	1
diving	0.38	0
drunk	0.53	0
fernando	0.36	1
fish1	0.06	0
fish2	0.37	2
gymnastics	0.61	0
hand1	0.45	0
hand2	0.44	2
jogging	0.51	1
motocross	0.48	4
polarbear	0.57	0
skating	0.42	3
sphere	0.48	0
sunshade	0.55	0
surfing	0.58	0
torus	0.53	1
trellis	0.40	2
tunnel	0.39	5
woman	0.42	2
<b>Average</b>	0.46	1.12

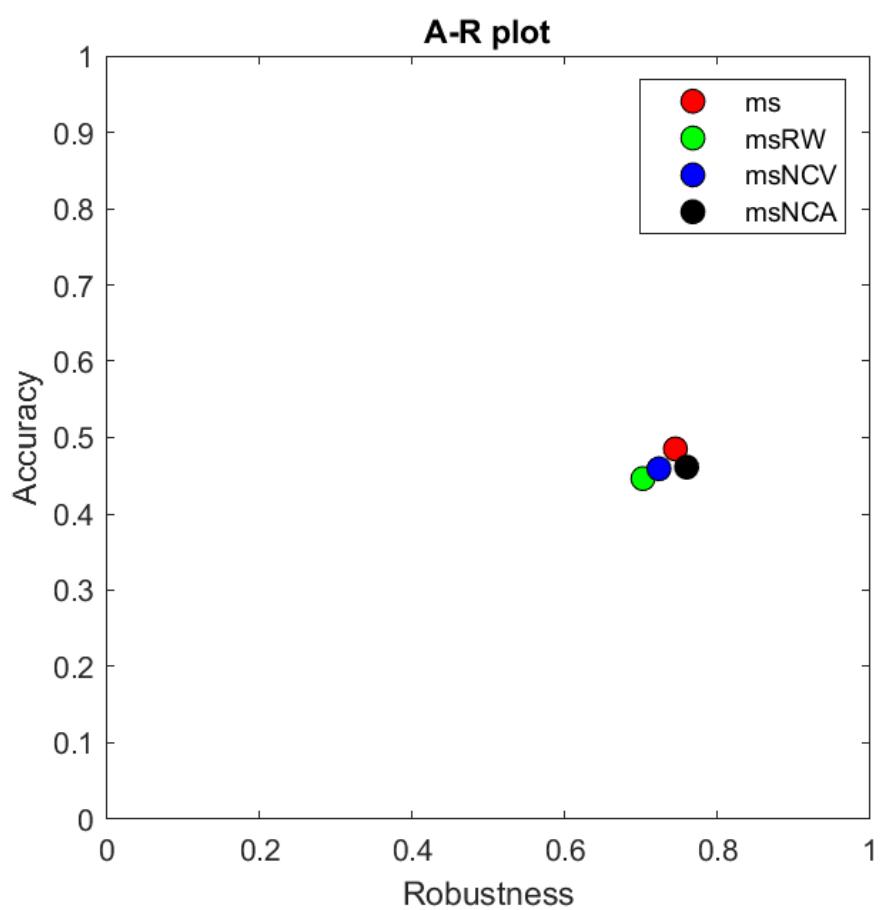
Tabela 4: Rezultati testiranja sledilnika z NCV modelom gibanja na testnih sekvencah vot2014.



Slika 21: Kljub veliki kovarianci predikcije sledilnika z modelom gibanja NCA, uspemo premakniti končno stanje dovolj, da se izognemo napačni meritvi. Modri krog prikazuje velikost variance (negotovosti) predikcije, rdeči varianco meritve, zeleni pa končno stanje in njegovo varianco. Sekvenca 'hand2'.

Sequence	MS		MSRW		MSNCV		MSNCA	
	Overlap	Failures	Overlap	Failures	Overlap	Failures	Overlap	Failures
ball	0.75	0	0.69	1	0.73	0	0.75	0
basketball	0.63	2	0.61	2	0.63	2	0.63	2
bicycle	0.40	0	0.33	0	0.44	1	0.38	0
bolt	0.50	2	0.46	2	0.44	2	0.39	2
car	0.57	0	0.52	0	0.58	0	0.57	0
david	0.29	1	0.40	1	0.42	1	0.31	1
diving	0.37	0	0.38	0	0.37	0	0.38	0
drunk	0.53	0	0.53	0	0.53	0	0.53	0
fernando	0.39	1	0.36	1	0.37	1	0.36	1
fish1	0.18	3	0.15	2	0.27	3	0.06	0
fish2	0.36	1	0.33	1	0.36	1	0.37	2
gymnastics	0.62	0	0.62	0	0.59	1	0.61	0
hand1	0.48	0	0.41	4	0.49	0	0.45	0
hand2	0.49	4	0.46	7	0.31	4	0.44	2
jogging	0.68	2	0.59	0	0.48	1	0.51	1
motocross	0.48	4	0.31	1	0.40	3	0.48	4
polarbear	0.58	0	0.59	0	0.56	0	0.57	0
skating	0.42	1	0.43	1	0.40	1	0.42	3
sphere	0.51	0	0.43	0	0.44	0	0.48	0
sunshade	0.56	0	0.48	6	0.46	3	0.55	0
surfing	0.52	0	0.46	0	0.56	0	0.58	0
torus	0.52	1	0.41	1	0.50	1	0.53	1
trellis	0.42	1	0.41	1	0.35	1	0.40	2
tunnel	0.43	6	0.36	4	0.34	6	0.39	5
woman	0.46	1	0.43	1	0.45	1	0.42	2
<b>Average</b>	0.49	1.20	0.45	1.44	0.46	1.32	0.46	1.12

Tabela 5: Rezultati testiranja sledilnikov z različnimi modeli gibanja (brez modela gibanja, RW, NCV, NCA) na testnih sekvencah vot2014.



Slika 22: A-R graf primerjave sledilnikov z različnimi modeli gibanja.