

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 853

**PRAĆENJE OBJEKATA U SLIKOVNIM
SEKVENCAMA ALGORITMOM CAMSHIFT**

Igor Bonači

Zagreb, lipanj 2009

Sadržaj

Sadržaj	1
1. Uvod.....	2
2. Opis algoritma.....	3
2.1. Pregled temeljnih postupaka	3
2.2. Izrada histograma.....	4
2.3. Histogram back-projection	6
2.4. Usporedba histograma	8
2.5. Opis mean-shift algoritma.....	8
2.6. Opis camshift algoritma	15
2.7. Opis ABCshift algoritma	18
3. Rezultati testiranja.....	22
4. Implementacija	29
5. Zaključak.....	31
6. Literatura.....	33
7. Naslov, sažetak i ključne riječi	34

1. Uvod

Praćenje objekata je proces određivanja položaja jednog ili više pokretnih objekata u vremenu. Zadatak algoritama za praćenje je analiza slikovnih sekvenci u svrhu izdvajanja korisne informacije. Praćenje osoba, vozila ili drugih objekata predstavlja važan aspekt današnjih i budućih aplikacija te stoga ima vrlo široku primjenu u praksi.

Danas su poznati mnogi algoritmi čija je svrha praćenje objekata u slikovnom zapisu. Možemo ih podijeliti na: *region-based*, *contour-based* i *model-based* algoritme.

Region based algoritmi su generički algoritmi koji koriste informaciju o boji i teksturi objekta kojeg prate. *Contour based* algoritmi izdvajaju i prate isključivo konture objekta. Generičko rješenje problema praćenja je još uvijek vrlo zahtjevan problem stoga se koriste i metode praćenja temeljene na modelu (*model-based* algoritmi) koje zahtijevaju apriorne informacije o obliku i tipu objekta.

Navedeni algoritmi razlikuju se po svojim karakteristikama, poglavito vremenskoj i prostornoj složenosti te domeni primjene.

U sklopu ovog rada bit će obrađena dva jednostavna i računalno vrlo učinkovita algoritma, iz skupine *region based* algoritama, *mean-shift* i *camshift*. Izloženi su rezultati testiranja navedenih algoritama u nekoliko realnih primjena s posebnim naglaskom na praćenje prometnih znakova. Kao poboljšanje *camshift* algoritma prikazan je i *ABCshift* algoritam koji u pojedinim slučajevima popravljajući performanse praćenja. Bolje performanse posebice se uočavaju prilikom praćenja objekata pomičnom kamerom. Jedan od takvih primjera je upravo praćenje prometnih znakova.

2. Opis algoritma

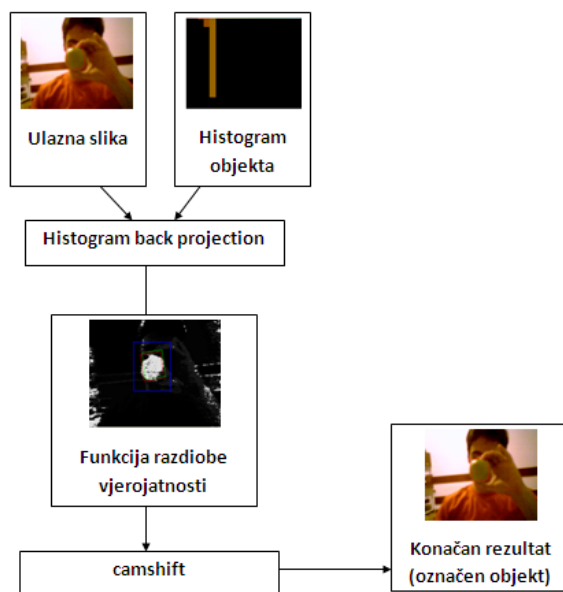
2.1. Pregled temeljnih postupaka

Algoritam *mean-shift* se koristi prilikom nalaženja ekstrema zadanog skupa uzoraka. Algoritam iterativno pomiče centralnu točku u skladu s *mean-shift* vektorom pomaka. Takav postupak rezultira pomicanjem centralne točke u smjeru gradijenta funkcije. Postupak završava konvergencijom u nekom ekstremu zadanog skupa uzoraka. Ukoliko zadani skup uzoraka predstavlja gustoću vjerojatnosti položaja objekta u sceni, a početna točka očekivani položaj objekta, centralna točka će konvergirati najvjerojatnijem položaju objekta u okolini početne točke.

Algoritam *camshift* je zasnovan na *mean-shift* algoritmu. Koristi se za nalaženje ekstrema skupa uzoraka, ali za razliku od *mean-shift*-a, radi se o algoritmu koji ima veću mogućnost prilagodbe promjenama. *Camshift* dinamički određuje veličinu prozora za traženje te se time bolje prilagođava nastalim promjenama, primjerice približavanju ili udaljavanju objekta od kamere.

Algoritmi *mean-shift* i *camshift* za svoj ulaz zahtijevaju razdiobu vjerojatnosti položaja objekta, a kao rezultat daju procjenu veličine i pozicije objekta u sceni. Stoga prije korištenja navedenih algoritama potrebno je poznavati funkciju koja opisuje vjerojatnost da se traženi objekt nalazi na određenoj poziciji u slici. U ovom radu koristi se jednostavan *region-based* algoritam za izračun te razdiobe.

Slika 1. ilustrira tijek rada algoritma. Prvi korak je označavanje objekta u ulaznoj slici i izrada histograma koji modelira objekt. Zatim se pomoću stvorenog modela postupkom *histogram back-projection* izračunava za svaki pojedini slikovni element vjerojatnost da se objekt nalazi upravo na tom mjestu. Posljednji korak algoritma je pronalazak najbližeg ekstrema izračunate razdiobe u okolini prethodnog položaja objekta. Taj posao obavljaju dva algoritma *mean-shift* te *camshift*.



Slika 1. Shematski prikaz rada algoritma

2.2. Izrada histograma

Histogram se izračunava brojanjem slikovnih elemenata određene boje nad regijom koja predstavlja objekt. Boje se kvantiziraju u odjeljke, prema *relaciji 1*. Sama operacija kvantizacije doprinosi otpornosti na pogreške (šum u slici) odnosno robusnosti cijelog postupka te manjoj memorijskoj složenosti.

$$q_u = \sum_{i=1}^n \sum_{j=1}^m \delta[c(pixel_{ij}) - u] \quad (1)$$

Relacija 1. prikazuje izračun jednostavnog histograma. Funkcija $c: N^3 \rightarrow \{1, \dots, k\}$ vrijednost slikovnog elementa na poziciji (i, j) preslikava u vrijednost indeksa odjeljka. Tipično funkcija c koristi *hue* komponentu iz *HSV* sustava boja. Česta je uporaba i normaliziranih *RGB* komponenti ili korištenje a, b komponenti iz *Lab* sustava boja. Odabir navedenih komponenti doprinosi otpornosti na promjenu osvjetljenja objekta te na mogući nastanak sjene.

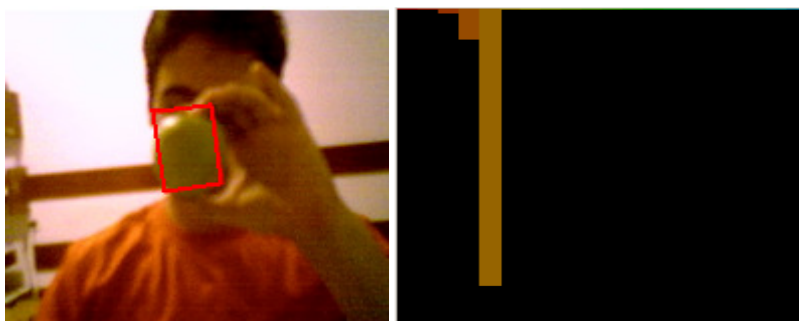
Posebnu pažnju zahtijeva slučaj u kojem slikovni element ima vrlo nisko zasićenje (*saturation*) ili vrlo nisko osvjetljenje (*brightness, value*) u *HSV* sustavu. U navedenim slučajevima, i u slučaju bijele boje, *hue* vrijednost nije precizno definirana te je stoga potrebno za ovakve boje pripremiti poseban slučaj prilikom izrade histograma.

Opisani histogram nad *hue* komponentom korišten je prilikom testiranja. *Slika 2.* prikazuje sliku scene te pripadni histogram nad *hue* komponentom. Može se primijetiti da gotovi svi slikovni elementi praćenog objekta pripadaju istom odjeljku. To je posljedica korištenja *hue* komponente koja određuje ton boje u skladu s ljudskim vizualnim sustavom.

Uz takav jednostavan jednodimenzionalan korišteni su i složeniji histogrami. Jednostavnim proširenjem operacije kvantizacije na *a*, *b* komponente iz *Lab* sustava boja dolazimo do dvodimenzionalnog histograma objekta. Ovakav histogram vjernije opisuje početni objekt, a zadržava dobro svojstvo otpornosti na promjenu osvjetljenja objekta zbog odbacivanja *L* komponente koja prikazuje osvjetljenje.

U ovom trenutku važno je istaknuti neke prednosti histograma, a to su: jednostavnost i računalna učinkovitost njegovog izračuna, otpornost na pogreške, otpornost na djelomičnu pokrivenost te invarijantnost na rotaciju i skaliranje objekta. Istaknute dobre karakteristike modeliranja objekta histogramom u najvećoj mjeri određuju kvalitetna dinamička svojstva *camshift* algoritma.

Nasuprot tome pojavljuje se problem prilikom korištenja histograma ukoliko je početna veličina objekta mala. U tom slučaju, zbog nedostataka podataka, može se dogoditi da histogram ne opisuje vjerno početni objekt.



Slika 2. Objekt te njegov pripadni jednodimenzionalni histogram.

2.3. Histogram back-projection

Kako je navedeno u prethodnom odjeljku histogram modelira određeni objekt. Ta ovisnost objekta i njegovog histograma prikazana je izrazom: $H(M|o_n, R, T)$.

M predstavlja lokalni skup mjera, R predstavlja rotaciju, a T translaciju objekta o_n . Preciznije rečeno histogram opisuje ovisnost između skupa mjera M te objekta o_n i njegovog položaja i rotacije. Moguće je odabrati različite skupove mjera M , primjerice boju slikovnog elementa ili gradijent slike. U prethodnom odjeljku navedeno je da je izrada histograma invarijantna na rotaciju i translaciju, stoga se navedeni izraz može pisati i na sljedeći način: $H(M|o_n)$. Postupkom normalizacije sa ukupnim brojem unosa u histogram dobije se $p(M|o_n)$, odnosno uvjetna vjerojatnost pojave skupa mjera M u slučaju pojave objekta o_n .

U slučaju detekcije i praćenja objekata potrebna je upravo suprotna uvjetna vjerojatnost, odnosno $p(o_n|M)$. Za izračun tražene vjerojatnosti iz već poznate vjerojatnosti $p(M|o_n)$ koristi se Bayes-ov teorem. Izravnom primjenom teorema slijedi *relacija 2*.

$$p(o_n|M) = \frac{p(M|o_n) * p(o_n)}{\sum_i p(M|o_i) * p(o_i)} \quad (2)$$

S obzirom na važnost *relacije 2*. u prikazanom algoritmu iznesen je i njen detaljniji opis. Vjerojatnosti $p(o_i)$ predstavljaju vjerojatnosti pojedinih objekata koje je potrebno pratiti. S obzirom da na početku analize ne postoji dodatna informacija o vjerojatnostima pojave pojedinih objekata, one se postavljaju na jednake iznose za svaki objekt.

Vjerojatnost $p(M|o_i)$ se izračunava pomoću histograma izrađenog za pojedini objekt o_i . Ukoliko se koristi najjednostavniji slučaj pri kojem je skup mjera M zapravo vrijednost slikovnog elementa na pojedinoj lokaciji tada, kao što je već prije prikazano, vjerojatnost $p(M|o_i)$ je jednaka vrijednosti normaliziranog histograma objekta u odjeljku tog slikovnog elementa.

Vrijednost nazivnika *relacije 2*, odnosno $\sum_i p(M|o_i) * p(o_i)$ je upravo jednaka vjerojatnosti $p(M)$ te ga je moguće izračunati čim budu poznati svi objekti koji se nalaze u sceni.

U daljnjem tekstu prikazan je najjednostavniji oblik *histogram back-projection*-a u kojem se koristi jednodimenzionalan normalizirani histogram H , te skup mjera M koji se izračunava nad *hue* vrijednosti pojedinog slikovnog elementa. Kako je prije rečeno vjerojatnosti svih objekta $p(o_i)$ su jednake te *relacija 2* prelazi u:

$$p(o_n|M) = \frac{p(M|o_n)}{\sum_i p(M|o_i)} \quad (3)$$

Vjerojatnosti $p(M|o_i)$ su izračunate stvaranjem histograma boja za objekt o_i . Preciznije rečeno vjerojatnost $p(M|o_i)$ se izračunava kao broj slikovnih elemenata objekta čija *hue* vrijednost pripada istom odjeljku kao i vrijednost slikovnog elementa na lokaciji (i, j) .

Ovim postupkom se za svaki slikovni element scene izračunava vjerojatnost da se objekt nalazi upravo na tom mjestu. Vremenska složenost opisanog postupka je $O(n^2)$, gdje n predstavlja vertikalnu odnosno horizontalnu rezoluciju slike. Primjer jedne takve razdiobe vjerojatnosti prikazan je na *slici 3*. Nastavak algoritma praćenja objekata ide u smjeru detekcije regije slike, odnosno veće nakupine slikovnih elemenata, u kojima je vjerojatnost pojave objekta visoka. Taj postupak provode dva algoritma: *mean-shift* te *camshift*, oba algoritma su opisana u nastavku.



Slika 3. Prikaz scene i funkcije razdiobe vjerojatnosti

Moguće je odrediti značajno složeniji skup mjera M , preciznije rečeno moguće je koristiti vektor koji se sastoji od više lokalnih mjera neke regije slike. Tada *relacija 3* prelazi u:

$$p(o_n|\Lambda_k m_k) = \frac{p(\Lambda_k m_k|o_n) * p(o_n)}{\sum_i p(\Lambda_k m_k|o_i) * p(o_i)} \quad (4)$$

Teško je izračunati združene vjerojatnosti $\Lambda_k m_k$ na nekoj regiji slike. Ali ukoliko se pretpostavi njihova međusobna neovisnost dobije se:

$$p(o_n | \Lambda_k m_k) = \frac{\prod_k p(m_k | o_n) * p(o_n)}{\sum_i \prod_k p(m_k | o_i) * p(o_i)} \quad (5)$$

2.4. Usporedba histograma

Alternativa *histogram back-projection* postupku za izračun funkcije razdiobe vjerojatnosti je usporedba histograma objekta te histograma u okolini slikovnog elementa. U svrhu usporedbe njihove sličnosti može se koristiti *Bhattacharyya-in koeficijent* definiran *relacijom 6*:

$$BC(p, q) = \sum_{i=1}^n \sqrt{p_i * q_i} \quad (6)$$

Geometrijska interpretacija *relacije 6* je kosinus kuta između n -dimenzionalnih jediničnih vektora $[\sqrt{p_1}, \dots, \sqrt{p_m}]^T$ i $[\sqrt{q_1}, \dots, \sqrt{q_m}]^T$

$BC(p, q)$ definira sličnost među dva histograma. Udaljenost se također može definirati pomoću *Bhattacharyya koeficijenta* :

$$d(p, q) = \sqrt{1 - BC(p, q)} \quad (7)$$

Na ovaj način definirana udaljenost zadovoljava uvjete metrike.

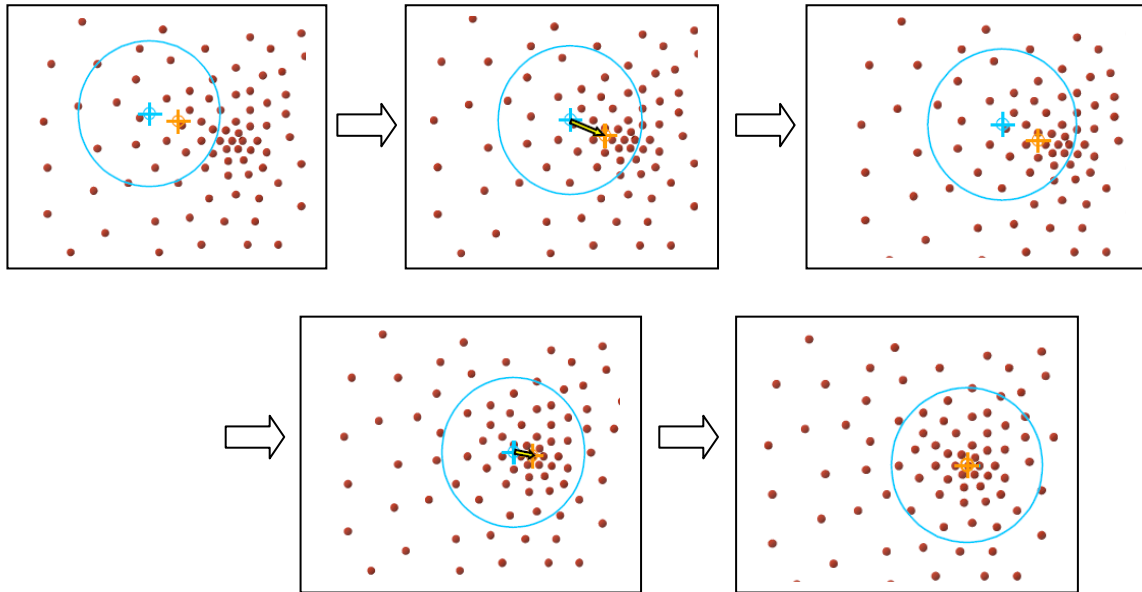
2.5. Opis mean-shift algoritma

Mean-shift (*mean-shift algorithm*, Comaniciu [4]) je algoritam koji se koristi za pronalaženje najbližeg ekstrema funkcije gustoće vjerojatnosti u prostoru R^N . Primjer jedne takve razdiobe sa označenim ekstremom prikazan je na *slici 3*.

Mean-shift algoritam može se opisati jednostavnim koracima:

1. Odaberi veličinu prozora za traženje
2. Odaberi početnu poziciju prozora za traženje. Najčešće se uzima pozicija prozora iz prethodnog okvira (*frame*) slikovne sekvence.
3. Izračunaj centar mase funkcije razdiobe vjerojatnosti unutar prozora za traženje.

4. Centriraj prozor na centar mase izračunat u trećem koraku.
5. Ponavljaj korake 3. i 4. do konvergencije.



Slika 4. Koraci rada mean-shift algoritma. Izvor: Y. Ukrainitz, B. Sarel

Slika 4. prikazuje postupak postupne konvergencije algoritma *mean-shift* ekstremu zadanog skupa uzoraka. Postupak završava postizanjem konvergencije, odnosno kada centralna točka izračunata u koraku 3. odgovara centralnoj točki prozora za traženje.

Gore prikazan algoritam predstavlja najjednostavniji mogući *mean-shift* algoritam. Naime svaka točka unutar prozora za traženje na jednak način sudjeluje u izračunu centra. Moguće je izvesti funkciju K (*kernel function*, *jezgra*) koja svakoj točki x_i pridjeljuje težinu, odnosno važnost, prilikom izračuna centra. Novi način izračuna centra $m(x)$ prikazuje *relacija 8*.

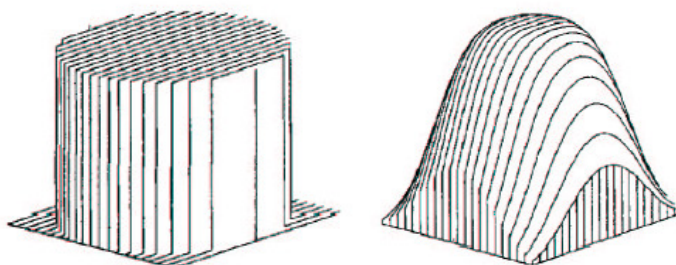
$$m(x) = \frac{\sum_{i=1}^n K(x - x_i) * x_i}{\sum_{i=1}^n K(x - x_i)} \quad (8)$$

Razlika $m(x) - x$ se naziva *mean shift vector*. Algoritam iterativno pomiče točku x na izračunati centar $m(x)$. Uvjet zaustavljanja algoritma glasi: $m(x) = x$.

Najčešće je jezgra K funkcija od $||x||^2$, odnosno piše se:

$$K(\mathbf{x}) = k\left(\|\mathbf{x}\|^2\right) \quad (9)$$

Funkcija k se naziva profil jezgre K . Neke česte jezgre su prikazane na slikama 5 i 6.



Slika 5. Ravna jezgra Slika 6. Gaussova jezgra
(izvor: Y. Ukrainitz, B. Sarel)

Ravna jezgra se definira funkcijom:

$$K(\mathbf{x}) = \begin{cases} C, & \text{za } \|\mathbf{x}\|^2 \leq 1 \\ 0, & \text{inače} \end{cases} \quad (10)$$

Gaussova jezgra funkcijom:

$$K(\mathbf{x}) = \frac{1}{\sqrt{(2 * \pi)^d}} \exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)$$

d je broj dimenzija prostora.

Pomoću jezgre moguće je doći do procjene funkcije razdiobe vjerojatnosti, ovaj postupak se naziva *Parzen window technique*. Za skup od n točaka u d -D prostoru procjena funkcije razdiobe vjerojatnosti pomoću jezgre K i radiusa h glasi:

$$f_k(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \quad (11)$$

Relacija 11. se može grafički predočiti kao suma jezgrenih funkcija postavljenih u svaki uzorak. Maksimum jezgrene funkcije je u točki $\mathbf{x} = 0$, a funkcija opada sa $\|\mathbf{x}\|^2$ što rezultira većim utjecajem bližih uzoraka, a što je uzorak udaljeniji od točke $\mathbf{x}$ njegov utjecaj na procjenu u toj točki opada.

Radius h služi kao parametar za podešavanje širine jezgre K . Naime povećavanjem radiusa h sve udaljenije točke pridonose procjeni razdiobe i time

čine izračunatu funkciju glađom. Ukoliko se uzme $h = 0$ tada samo lokalna točka pridonosi ukupnoj sumi, te će rezultatna procjena biti sastavljena od δ funkcija (*Dirac*-ovih funkcija).

Jezgra $G(x)$, sa pripadnim profilom $g(x)$, definirana je formulom:

$$g(x) = -k'(x), \quad G(x) = C * g(|x|^2) \quad (12)$$

Može se pokazati da *mean-shift* sa jezgrom G pomiče točku x u smjeru gradijenta funkcije $f_k(x)$. Ovaj rezultat je od velike važnosti za *mean-shift* algoritam. Ukoliko za procjenu razdiobe vjerojatnosti koristimo jezgru K , za ispravan rad *mean-shift* algoritma potrebno je koristiti jezgru G . Ukoliko kernel K ima konveksan i monotono padajući profil tada *mean-shift* algoritam s jezgrom G konvergira prema ekstremu funkcije $f_k(x)$.

Mean-shift algoritam prikazan na početku ovog odjeljka koristi ravnu jezgru G . Iz *relacije 12.* slijedi da jezgra K potrebna za izračun razdiobe vjerojatnosti jednaka integralu od $g(x)$ te glasi:

$$K(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d + 2) * (1 - |x|^2), & \text{za } |x| \leq 1 \\ 0, & \text{inače} \end{cases} \quad (13)$$

U *relaciji 13* c_d je obujam d dimenzionalne sfere. Kernel definiran *relacijom 13* naziva se *Epanechnikov kernel*. Karakterističan je zbog toga što minimizira pogrešku procjene i prave razdiobe.

Opisani postupak se koristi za procjenu funkcije gustoće vjerojatnosti. Jedan od mogućih pristupa problemu, nakon prikazane analize, je analitički pronalazak maksimuma izračunate razdiobe. Ovakav postupak je pretjerano kompliciran i zahtjevan, stoga je zadatak *mean-shift* algoritma pronalazak ekstrema funkcije bez analitičkog proračuna razdiobe.

Cilj *mean-shift* algoritma je maksimizirati *Bhattacharyya koeficijent* između histograma objekta i histograma okoline slikovnog elementa. Usporedbom histograma se izračunava vjerojatnost da se objekt nalazi baš na tom položaju.

Prvi korak je ponovno izrada histograma objekta. Neka su sa $\{x_i^*\}_{i=1 \dots n}$ označeni pozicije slikovnih elemenata objekta. Može se definirati funkcija $b : \mathbf{R}^2 \rightarrow \{1 \dots m\}$

koja povezuje slikovni element na poziciji $\mathbf{x}_i^*$ s odjeljakom histograma $b(\mathbf{x}_i^*)$ ovisnosti o njegovoj boji. Za što vjerniju izrada histograma objekta koriste se jezgre koje slikovnim elementima udaljenijim od centra objekta daju manju težinu s obzirom da postoji veća mogućnost da oni ne reprezentiraju objekt. Ova težinska funkcija omogućava robusniju procjenu.

$$\hat{q}_u = C * \sum_{i=1}^n k\left(\left\|\frac{\mathbf{x}_i^*}{h}\right\|^2\right) * \delta(b(\mathbf{x}_i^*) - u) \quad (14)$$

h je u ovom slučaju radius kernel funkcije, odnosno omogućava prilagodbu veličini objekta. Umjesto korištenja radiusa h moguće je koristiti normalizirane koordinate.

C u ovom slučaju je samo konstanta koja omogućava normalizaciju histograma. Može se izračunati sumiranjem vrijednosti svih odjeljaka histograma:

$$C = \frac{1}{\sum_{i=1}^n k\left(\left\|\frac{\mathbf{x}_i^*}{h}\right\|^2\right)} \quad (15)$$

Pretpostavimo da je jedna od mogućih pozicija objekta u trenutnom okviru slikovne sekvence $\mathbf{y}$. Neka su s $\{\mathbf{x}_i\}_{i=1 \dots n_h}$ označeni slikovni elementi u okolini od $\mathbf{y}$. Koristeći isti kernel k , ali u ovom trenutku s drugim radiusom h_2 moguće je izraditi histogram okoline slikovnog elementa $\mathbf{y}$.

$$\hat{p}_u(\mathbf{y}) = C_h * \sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y} - \mathbf{x}_i}{h_2}\right\|^2\right) * \delta(b(\mathbf{x}_i)) - u \quad (16)$$

Ponovno je C_h samo normalizacijska konstanta koju je moguće izračunati iz:

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y} - \mathbf{x}_i}{h_2}\right\|^2\right)} \quad (17)$$

Važno je primjetiti da C_h ne ovisi o $\mathbf{y}$ s obzirom da su točke $\mathbf{x}_i$ simetrično postavljene oko slikovnog elementa $\mathbf{y}$.

Već je prije navedeno da je cilj *mean-shift* pronalazak regije čiji histogram ima najveću sličnost s histogramom objekta. Korištenjem razvoja funkcije u Taylorov red moguće je zapisati:

$$BC(\hat{p}(\mathbf{y}), \hat{q}) \approx \frac{1}{2} * \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{\mathbf{y}}_0) \hat{q}_u} + \frac{1}{2} * \sum_{u=1}^m \hat{p}_u(\mathbf{y}) * \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{y}}_0)}} \quad (18)$$

Prikazana relacija uzima za pretpostavku da nije došlo do veće promjene histograma okoline između točke $\hat{\mathbf{y}}_0$ i $\mathbf{y}$. Uvrštavanjem *relacije 16* u *relaciju 18* dobije se:

$$BC(\hat{p}(\mathbf{y}), \hat{q}) \approx \frac{1}{2} * \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{\mathbf{y}}_0) \hat{q}_u} + \frac{C_h}{2} * \sum_{u=1}^m w_i k\left(\left|\frac{\mathbf{y} - \mathbf{x}_i}{h}\right|^2\right) \quad (19)$$

$$w_i = \sum_{u=1}^m \delta(b(\mathbf{x}_i) - u) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{y}}_0)}} \quad (20)$$

Iz ovog razmatranja je vidljivo da u ukoliko se želi maksimizirati *Bhattacharyya* koeficijent potrebno je maksimizirati *relaciju 19*. S obzirom da je prvi sumand u *relaciji 19* neovisan o $\mathbf{y}$ potrebno je maksimizirati samo drugi sumand.

Sada je moguće prikazati potpuniju sliku *mean-shift* algoritma po koracima:

1. Izraditi histogram q koji modelira objekt.
2. Postavi $\mathbf{y}$ na poziciju objekta u prethodnoj slici, tj $\mathbf{y} = \mathbf{y}_0$.
3. Izračunati histogram $\hat{p}$ u okolini točke $\mathbf{y}_0$, te pripadni *Bhattacharyya* koeficijent $\mathbf{BC}_0$ usporedbom s histogramom objekta.
4. Za svaku točku (x, y) u okolini od $\mathbf{y}$ izračunaj w :

$$c = b(x, y), \quad w_i = \sqrt{\frac{q_c}{\hat{p}_c}} \quad (21)$$

q je diskretna razdioba dobivena iz histograma objekta. *Relacija 21* je ekvivalentna *relaciji 20*.

5. Zatim koristeći *mean-shift* vektor izračunaj novu poziciju $\mathbf{y}'$ (n je broj točaka u okolini od $\mathbf{y}$):

$$\mathbf{y}' = \frac{\sum_{i=1}^n G\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) * \mathbf{x}_i * w_i}{\sum_{i=1}^n G\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) * w_i} \quad (22)$$

6. Izračunaj histogram p u okolini točke y' , te pripadni *Bhattacharyya* koeficijent BC_1
7. Ukoliko je $BC_1 < BC_0$ tada postavi $y' = \frac{1}{2}(y + y')$ te se vrati na korak 6, inače nastavi s korakom 8.
8. Ako je razlika $y' - y$ manja od jednog slikovnog elementa postupak je gotov, inače postavi $y = y'$ i vrati se na korak 3.

Korak 4. algoritma aproksimira vrijednost *Bhattacharyya* koeficijenta. Zatim se nad tom aproksimacijom primjenjuje *mean-shift* vektor te se izračunava nova pozicija y' . Ukoliko je $BC_1 < BC_0$ tada je pomak rezultiran *mean-shift* vektorom bio prevelik te se y' postavlja na aritmetičku sredinu između y i y' . Autori navode da je ovaj korak potreban tek u nešto manje od 0.1% slučajeva [4].

Glavna mana ovog pristupa je otežana detekcija promjene veličine objekta. Autori navode način moguće prilagodbe na takve promjene. Provodi se nekoliko *mean-shift* iteracija s različitim radiusima jezgre ($\pm 10\%$), te se među njima odabere onaj s najmanjom pogreškom. Upravo problem detekcije veličine objekta rješava *camshift* algoritam opisan u nastavku.

2.6. Opis camshift algoritma

Algoritam *camshift* (*Continuously Adaptive Mean Shift algorithm*, Bradsky [1]) se koristi za istu namjenu kao i *mean-shift* algoritam opisan u prethodnom odjeljku, odnosno za pronalaženje najbližeg ekstrema funkcije razdiobe vjerojatnosti u prostoru R^N . U najjednostavnijem slučaju primjenjuje se izravno na razdiobu dobivenu *histogram back-projection*-om. Primjer jedne takve razdiobe sa označenim ekstremom prikazan je na slici 3.

Za razliku od *mean-shift* algoritma *camshift* ima mogućnost bolje prilagodbe promjenama u funkciji razdiobe vjerojatnosti te samim time posjeduje bolje performanse prilikom praćenja objekata. Jedna od najvažnijih dodatnih mogućnosti je dinamička prilagodba prozora za traženje na novu veličinu objekta te jednostavan i efikasan izračun nagiba, središta te veličine objekta.

Camshift algoritam korake 3. i 4. iz prethodnog odjeljka o *mean-shift* algoritmu mijenja. Ukupan algoritam se može predstaviti koracima:

1. Iz ulazne slike modeliraj objekt pomoću histograma.
2. Postupkom *histogram back-projection* za svaki slikovni element slike izračunati vjerojatnost da se traženi objekt nalazi upravo na tom mjestu.
3. Odaberi veličinu prozora za traženje. Veličina prozora za traženje ovisi o veličini objekta u prethodnom koraku.
4. Odaberi početnu poziciju prozora za traženje, najčešće se uzima pozicija prozora iz prethodnog okvira slikovne sekvence.
5. a) izračunati nulti moment unutar prozora za traženje

$$M_{00} = \sum_x \sum_y I(x, y) \quad (23)$$

M_{00} je suma vjerojatnosti unutar prozora za traženje.

5. b) izračunati prve momente za x i y tj.

$$M_{10} = \sum_x \sum_y x * I(x, y); M_{01} = \sum_x \sum_y y * I(x, y) \quad (24)$$

5. c) izračunati druge momente za x i y tj.

$$M_{20} = \sum_x \sum_y x^2 * I(x, y) \quad (25)$$

$$M_{02} = \sum_x \sum_y y^2 * I(x, y) \quad (26)$$

$$M_{11} = \sum_x \sum_y x * y * I(x, y) \quad (27)$$

5. d) sredina prozora za traženje se izračunava na sljedeći način:

$$x_c = \frac{M_{10}}{M_{00}}; y_c = \frac{M_{01}}{M_{00}}; \quad (28)$$

Koordinate (x_c, y_c) su koordinate centra mase unutar prozora za traženje.

Veličina prozora je definirana s:

$$l = \sqrt{\frac{(a + c) + \sqrt{b^2 + (a - c)^2}}{2}}; \quad (29)$$

$$w = \sqrt{\frac{(a + c) - \sqrt{b^2 + (a - c)^2}}{2}} \quad (30)$$

Kut nagiba definiran je s:

$$\theta = \frac{1}{2} * \arctan \left(\frac{b}{a - c} \right) \quad (31)$$

Pritom vrijedi:

$$a = \frac{M_{20}}{M_{00}} - x_c^2; \quad (32)$$

$$b = 2 * \left(\frac{M_{11}}{M_{00}} - x_c * y_c \right); \quad (33)$$

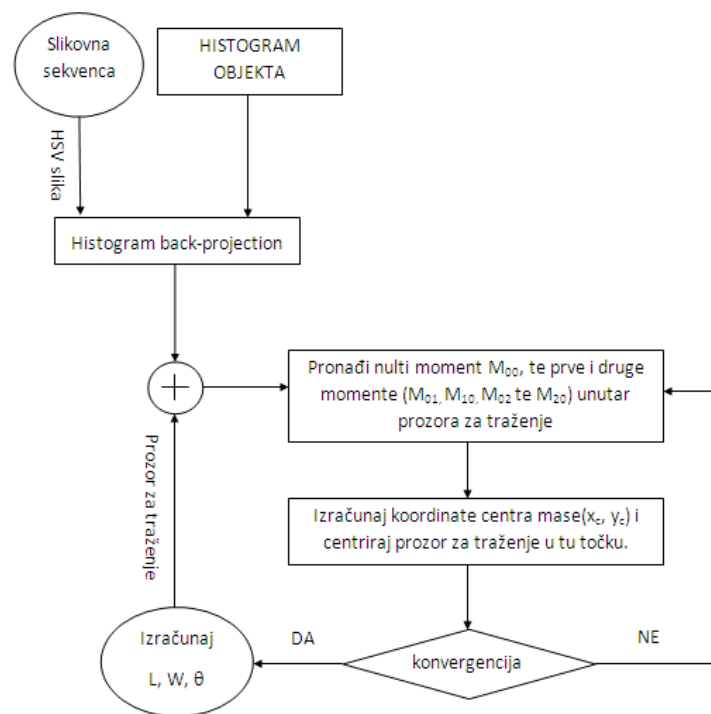
$$c = \frac{M_{02}}{M_{00}} - y_c^2; \quad (34)$$

6. centriraj prozor veličine (l, w) na poziciju (x_c, y_c) i ponavljaj korake 5 i 6 do konvergencije.

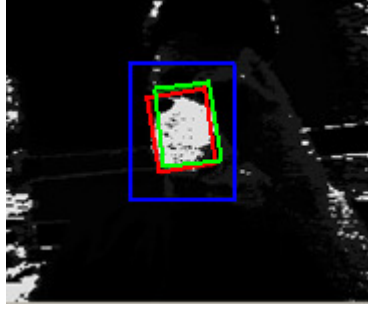
$I(x, y)$ je vjerojatnost pojava objekta na lokaciji (x, y) .

Veličina prozora za traženje se mijenja iz koraka u korak i ovisi o M_{00} u trenutnom koraku prema izrazu: $s = 2 * \sqrt{M_{00}}$. Ovaj izraz je važan jer omogućava *camshift* algoritmu da dinamički prilagodi promjenama u razdiobi. Ukoliko je ukupna suma vjerojatnosti unutar prozora za traženje velika tada će se u sljedećem koraku prozor za traženje proširiti i omogućiti detekciju povećanja objekta.

Važno je primjetiti da je algoritam *camshift* samo nadogradnja na *mean-shift* algoritam. Naime koordinate (x_c, y_c) se izračunavaju na potpuno isti način kao i kod *mean-shift* algoritma, ukoliko se koristi ravan kernel. Preciznije rečeno računa se centar mase funkcije razdiobe unutar prozora za traženje. Novitet su formule za određivanje visine, širine i nagiba objekta odnosno l, w, θ . Parametri a, b, c su uvedeni isključivo zbog kraćeg prikaza formula. Blok diagram *camshift* algoritma prikazan je na slici 8. Prikaz postupka postupne konvergencije ekstremu razdiobe prikazan je na slici 9. Plavi prozor je prozor za traženje, a zeleni odnosno crveni okvir prikazuju postupak postupne konvergencije *camshift* algoritma. Važno je primjetiti da se veličina prozora za traženje mijenja iz koraka u korak.



Slika 8. Blok diagram *camshift* algoritma



Slika 9. Prikaz rada CAMSHIFT algoritma u prostoru R^2 .

2.7. Opis ABCshift algoritma

ABCshift (*An Adaptive Background Model for Camshift Tracking with a Moving camera*, R. Stolkin, I. Florescu, G. Kamberov [3]) je algoritam zasnovan na Camshift algoritmu.

Prilikom testiranja Camshift algoritma nad slikovnim sekvencama u kojima kamera nije stacionarna pokazuje se da algoritam ne radi na zadovoljavajući način. Uzrok takvog ponašanja Camshift algoritma je statičan model pozadine. Preciznije rečeno histogrami boja praćenog objekta i pozadine izgrađeni su prilikom učenja objekta te se kasnije ne mijenjaju. Ukoliko se dogodi da se u pozadini praćenog objekta, uslijed pomaka kamere, pojavi razdioba boja slična zadanom objektu Camshift algoritam vrlo često ne uspijeva pratiti zadani objekt.

Navedeni problem moguće je riješiti izradom dinamičkog modela pozadine te samim time prilagodbom rada algoritma konkretnoj okolini objekta.

Odjeljak o *histogram back-projection-u* u *relaciji 2* navodi način izračuna uvjetne vjerojatnosti $p(o_n|M)$. Pokazano je da vrijedi $\sum_i p(M|o_i) = p(M)$. Camshift algoritam statički izgradi vjerojatnosti $p(M)$ prilikom učenja modela pozadine. Za razliku od toga ABCshift-a mijenja model pozadine promjenom okoline objekta. Relacija 2 se može dalje proširiti:

$$\begin{aligned}
 p(o_n|M) &= \frac{p(M|o_n)}{\sum_i p(M|o_i)} = \frac{p(M|o_n)}{p(M)} = \\
 &= \frac{p(M|o_n)}{p(M|o_n) * p(o_n) + p(M|B) * p(B)} \quad (35)
 \end{aligned}$$

$p(B)$ predstavlja vjerojatnost da slikovni element pripada pozadini. Očigledno vrijedi: $p(M|o_n) + p(M|B) = p(M)$ jer slikovni element pripada ili pozadini ili objektu, odnosno skupovi o_n i B su disjunktni. Iz ovog razmatranja slijedi da ukoliko pretpostavimo $p(o_n) = p(B)$ *relacija 35* u potpunosti odgovara *relaciji 3* korištenoj kod *camshift* algoritma. *ABCShift* odbacuje ovu pretpostavku i postavlja vjerojatnosti: $p(o_n) = \frac{1}{r}$; $p(B) = \frac{r}{r-1}$, r je omjer površine prozora za traženje i površine objekta u prethodnom koraku. Vjerojatnosti $p(M|o_n)$ su izračunate prilikom izrade histograma objekta i one se ne mijenjaju tokom vremena. Vjerojatnosti $p(M|B)$ se mijenjaju tokom vremena i izračunavaju se iz histograma prozora za traženje u trenutnom koraku. Upravo na ovaj način se *ABCshift* dinamički prilagođava promjenama u pozadini.

Prednosti ovog pristupa se mogu vidjeti na primjeru praćenja objekta čiji histogram u većoj mjeri korelira s histogramom trenutne pozadine. *Camshift* u tom slučaju veći dio pozadine zamjenjuje za objekt, dok *ABCshift* veću težinu pridaje slikovnim elementima koji razlikuju objekt od trenutne pozadine.

Adaptivno prilagođavanje pozadini omogućava uspješno praćenje u slučajevim u kojima *Camshift* ne uspijeva, ali istovremeno dovodi do moguće nestabilnosti. Naime ukoliko se dogodi da u pojedinom trenutku zbog manje pogreške izračuna veličine objekta, jedan dio objekta pripadne pozadini tada se model pozadine počinje prilagođavati razdiobi boja objekta. Ovo u nekoliko sljedećih slikovnih okvira dovodi do postupnog smanjenja prozora za traženje na vrlo malu veličinu i time se gubi mogućnost daljnjeg praćenja.

Rješenje navedenog problema je vrlo jednostavno. Smanjenje prozora za traženje dovodi do povećanja sličnosti izračunatog histograma pozadine te histograma objekta. U slučaju da ova sličnost pređe unaprijed određeni prag potrebno je povećati veličinu prozora za traženje. Jedan od mogućih načina izračuna sličnosti između dva histograma je *Bhattacharyya* metrika definirana *relacijom 36*.

$$d(p, q) = \sqrt{\sum_{i=1}^K (\sqrt{p_i} - \sqrt{q_i})^2}, 0 \leq d \leq \sqrt{2} \quad (36)$$

Relacija 36. se također može prikazati pomoću *Bhattacharyya* koeficijenta definiranog *relacijom 6*:

$$d(p, q) = \sqrt{2} * \sqrt{1 - BC(p, q)} . \quad (37)$$

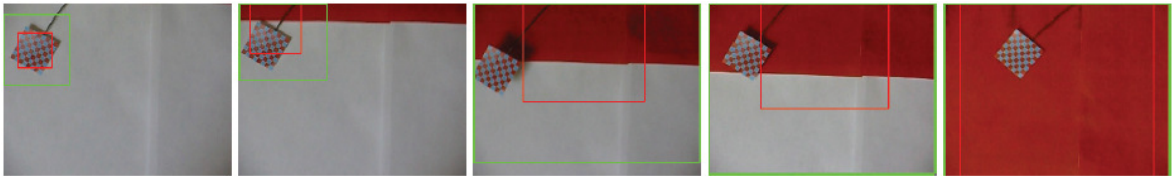
U svakoj iteraciji izračunava se *Bhattacharyya* udaljenost među histogramima objekta i pozadine te ukoliko izračunata vrijednost padne ispod unaprijed određenog praga potrebno je provesti skaliranje objekta i prozora za traženje faktorom r (omjer površine prozora za traženje i površine objekta).

Ostaje pitanje u kojem točno trenutku izračunati vjerojatnosti $p(M|B)$, odnosno je li dovoljno samo jednom prilikom obrade slikovnog okvira izračunati te vrijednosti ili je to potrebno ponavljati pri svakom pomaku prozora za traženje prilikom postupka konvergencije prema ekstremu funkcije gustoće razdiobe. Testiranjem se pokazalo da u realnim primjenama je dovoljno to učiniti samo jednom za svaki okvir slikovne sekvence jer oba postupka imaju ekvivalentne mogućnosti praćenja. Također važno je primjetiti da dodatan izračun ne usporava rad algoritma, nego kao što navode autori, u nekim slučajevima može dovesti čak i do ubrzanja rada s obzirom da bolji model pozadine rezultira manjim brojem koraka potrebnih za postizanje konvergencije [3].

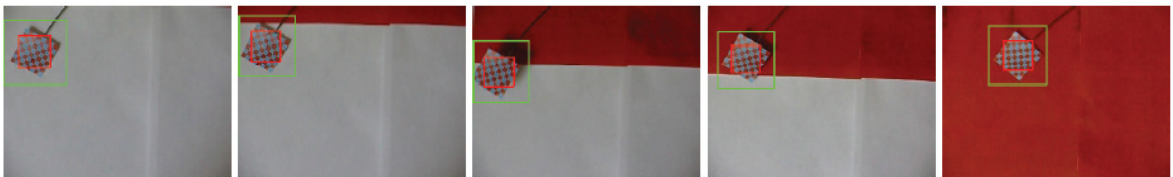
Sada je moguće iznijeti potpuni opis rada *abcsift* algoritma po koracima.

1. Iz ulazne slike modeliraj objekt pomoću histograma.
2. U ovom koraku za razliku od *camshift* algoritma izrađuje se i model pozadine. Preciznije rečeno, primjenom *relacije 35*, za svaki slikovni element, izračunava se vjerojatnost da se objekt nalazi upravo na tom mjestu.
3. Odaberi veličinu prozora za traženje. Veličina prozora za traženje ovisi o veličini objekta u prethodnom koraku.
4. Odaberi početnu poziciju prozora za traženje. Najčešće se uzima pozicija prozora iz prethodnog okvira slikovne sekvence.
5. Sljedeći koraci odgovaraju koracima iz *camshift* algoritma. Izračunavaju se momenti $M_{00}, M_{01}, M_{10}, M_{20}, M_{02}$, te zatim iz njih x_c, y_c te veličina, širina te nagib objekta pomoću *relacija 29, 30 i 31*.

6. Izračunaj razliku između histograma objekta i okoline pomoću *Bhattacharyya* udaljenosti. Ukoliko ta udaljenost bude manja od praga tada reskaliraj veličinu prozora za traženje s r .
7. Centriraj prozor na novo izračunati centar x_c, y_c te ponavljaj korake 5, 6 i 7 do konvergencije.



*Slika 10. Prikaz rada camshift algoritma na primjeru
(izvor: Stolkin, Florescu, Kamberov [3])*



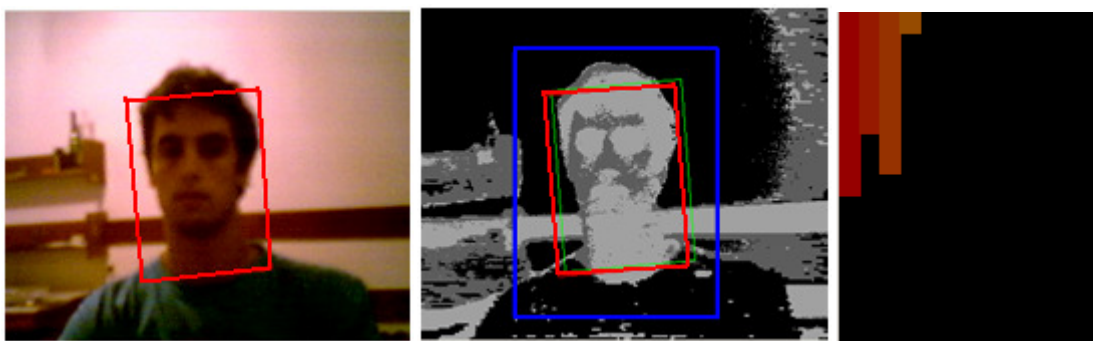
*Slika 11. Prikaz rada abcshift algoritma na primjeru
(izvor: Stolkin, Florescu, Kamberov [3])*

Primjer u kojem *abcshift* pokazuje svoje prednosti prilikom praćenja prikazan je *slikama 10, 11*. Razdioba boja objekta te okoline je u velikoj mjeri slična. Crveno bijeli objekt se kreće preko pozadine slične razdiobe boja. *Camshift* algoritma gubi mogućnosti praćenja čim objekt pređe preko regije s kojom ima sličnu razdiobu boja. *Abcshift* algoritma se pokušava prilagoditi nastalim promjenama u okolini i u skladu s tom informacijom uskladiti izračunate vjerojatnosti pojave objekta. Rad *abcshift* algoritma prikazan je na *slici 11*.

3. Rezultati testiranja

Izvorna namjena algoritma *camshift* je praćenje lica u slici. Dizajniran je s ciljem izrade sučelja između korisnika i računala. Velika pažnja je posvećena brzini i efikasnosti algoritma kao i mogućnosti rada s kamerama niske kvalitete.

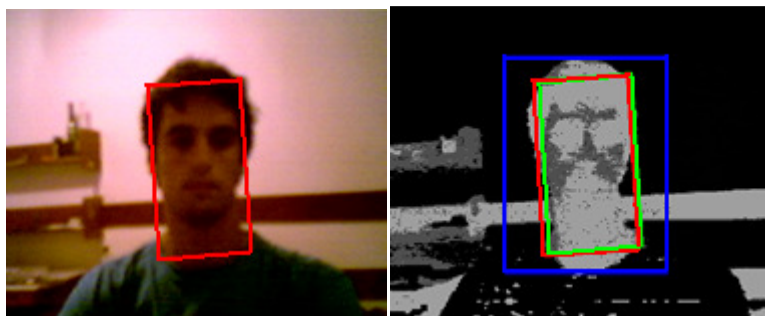
Prvi testovi u ovom radu će također biti posvećeni praćenju ljudskih lica u svrhu isporodbe s izvornim člankom.



Slika 12. Funkcija razdiobe vjerojatnosti i histogram lica.

Slika 12 prikazuje razdiobu vjerojatnosti položaja lica u sceni. Vidljiva je velika količina šuma koji se javlja većinom zbog loših karakteristika kamere. Plavom bojom je označen prozor za traženje *camshift* algoritma, a crvenom konačan objekt. U pozadini također postoje objekti vrlo slične razdiobe boja kao i ljudsko lice te time doprinose šumu. S obzirom da je lice dominantni objekt u sceni ovaj šum nema velikog utjecaja na praćenje te *camshift* algoritam gotovo savršeno prati lice. Prilikom udaljavanja lica od kamere pozadinski šum ima sve veći relativni utjecaj što za posljedicu ima sve lošije, ali ipak zadovoljavajuće praćenje.

Ukoliko se umjesto jednodimenzionalnog histograma (*slika 12*) koristi dvodimenzionalni histogram u Lab sustavu boja pozadinski šum se smanjuje, a praćenje lica postaje još i bolje. *Slika 13* prikazuje razdiobu vjerojatnosti prilikom korištenja dvodimenzionalnog histograma a , b komponenti iz Lab sustava boja. Primjetno je uži pravokutnik koji obuhvaća lice. Razlog tome je upravo u manjoj količini šuma, naime procjena veličine objekta se odvija preko nultog moment M_{00} koji se izračunava kao suma vjerojatnosti unutar prozora za traženje (plavi pravokutnik).

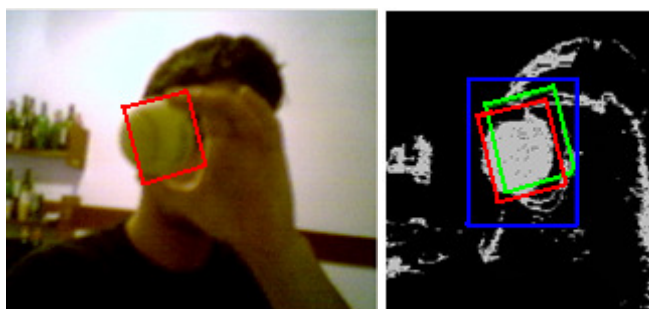


Slika 13. Funkcija razdiobe vjerojatnosti položaja prilikom korištenja dvodimenzionalnog histograma

Testiranje je provedeno uz razne vrste osvjetljenja, pokrete u pozadini, te uz djelomično pokrivanje lica. U svim slučajevima algoritam pokazuje odlične rezultate. Praćenje lica je narušeno tek uz prisustvo drugog lica u sceni čija je veličina bila veća ili jednaka početnom objektu. Naime korištenjem histograma nije moguće vršiti klasifikaciju između lica različitih osoba.

Promjena osvjetljenja prilikom testiranja nije utjecala na performanse praćenja. Time je pokazano da je odabir *hue* komponente iz HSV sustava boja, te *a*, *b* komponenti iz Lab sustava u većoj mjeri invarijantno na promjenu osvjetljenja. Ukoliko se tokom vremena događaju još veće promjene u razdiobi boja objekta moguće je povremeno obnavljati model.

Osim lica *camshift* algoritam pokazuje vrlo dobre rezultate kod praćenja objekata jednolike razdiobe boje. To se pokazuje na primjeru praćenja teniske loptice. Unatoč pozadinskom šumu praćenje objekta je gotovo savršeno.



Slika 14. Funkcija razdiobe vjerojatnosti položaja prilikom praćenja objekta

Testiranje je pokazalo da praćenje prometnih znakova predstavlja veći problem za *camshift* algoritam. Naime problem se pojavljuje kada okolina znaka sadrži neke od boja prometnog znaka. Jedan takav primjer prikazan je na *slici 15*.



Slika 15. Funkcija razdiobe vjerojatnosti prilikom praćenja prometnog znaka.



Slika 16. Histogram prometnog znaka

Prilikom praćenja prometnih znakova korištena je trokutasta regija za označavanje objekta umjesto tipične pravokutne. Taj pristup rezultira značajno boljim modelom objekta. Naime korištenje trokutaste regije za označavanje onemogućava da model objekta sadrži i neke slikovne elemente koji njemu ne pripadaju.

Prometni znak nema jednoliku razdiobu boja stoga njegov histogram ima značajno širu razdiobu. S obzirom da u histogramu postoji značajan udio bijele boje, slično kao i u okolini znaka, razdioba vjerojatnosti sadrži velike regije visoke vjerojatnosti u okolini znaka. Ovo rezultira vrlo lošim praćenjem prometnog znaka.

Puno bolje rezultate pokazuje *ABCshift* algoritam. Naime dinamički modela pozadine u okolini znaka sadrži veliku količinu bijele boje. Primjenom Bayes-ovog teorema utjecaj bijele boje na izračun funkcije vjerojatnosti se uvelike smanjuje. Crvena boja znaka time dobiva veći utjecaj. Ovaj primjer pokazuje da *ABCshift*

veću težinu pridaje slikovnim elementima koji razlikuju objekt od trenutne pozadine i time uvelike popravlja mogućnosti praćenja.

Slika 17. prikazuje jedan primjer praćenja prometnog znaka pomoću *ABCshift*. Vidljivo je da algoritam pravilno prati znak (crveni pravokutnik). Također je važno primjetiti na funkciji gustoće vjerojatnosti da najveći utjecaj na izračun maksimuma ima crvena boja.

Zanimljivo je primjetiti da se mogući nastanak nestabilnosti, opisan u odjeljku o *ABCshift* algoritmu, ne pojavljuje prilikom praćenja znakova. Razlog tome je što najveći utjecaj ima upravo crvena boja koja se nalazi na rubu znaka. Stoga procjena veličine objekta uvijek obuhvaća upravo tu regiju najveće vjerojatnosti te onemogućava nastanak opisane nestabilnosti.

Također može se primjetiti da prilikom praćenja prometnih znakova *ABCshift* algoritmom izračunati kut nagiba ne odgovara stvarnom kutu nagiba objekta. Razlog tome je oblik praćenog znaka. Naime zbog geometrijskog oblika znaka nagib slabo utječe na ukupnu sumu vjerojatnosti unutar prozora za traženje te time onemogućava njegovu procjenu (*slika 17*). Prilikom praćenja prometnih znakova, s obzirom na prirodu problema, moguće je potpuno isključiti izračun nagiba.



Slika 17. Prikaz rada ABCshift algoritma na primjeru praćenja prometnog znaka

Praćenje prometnog znaka ilustrirano je slikama 18, 19 i 20. Označeni objekt je gornji trokutasti znak. Oznaka praćenog objekta obuhvaća oba prometna znaka. Iako *ABCshift* algoritam modelira pozadinu objekta, velike sličnosti u razdiobi boja između praćenog objekta (trokutastog znaka), te susjednog objekta (okruglog znaka) nije moguće razdvojiti korištenjem isključivo histograma objekta.



Slika 18. Praćenje prometnog znaka algoritmom ABCshift



Slika 19. Praćenje prometnog znaka algoritmom ABCshift



Slika 20. Praćenje prometnog znaka algoritmom ABCshift

Prilikom praćenja objekta prikazanom na slikama 18, 19 i 20 *camshift* algoritam pokazuje gotovo identične rezultate. Naime u ovom slučaju okolina prikazanih prometnih znakova ne sadrži sličnu razdiobu boja te se praćenje odvija na način prikazan slikama.

Detekcija nestanka objekta iz scene također je provedena korištenjem usporedbe histograma. Preciznije rečeno provedena je usporedba histograma označenog objekta te histograma trenutno označenog područja. Ukoliko je ta sličnost pala ispod određene granice smatra se da je objekt nestao iz scene te se praćenje prekida.

Modeliranje objekta histogramom onemogućava obradu prostornih odnosa. Drugim riječima, s obzirom da je izrađen histogram nad značajkom boje, svi slikovni elementi iste boje imaju istu vjerojatnost pojave objekta. Iz tog razlika pomoću histograma nije moguće postići da bijele regije znaka imaju visoku vjerojatnost, a da okolina znaka koja je također bijela ima malu vjerojatnost pojave znaka. Ovo ograničenje posljedica je modeliranja objekta jednim histogramom boja, a ne samog algoritma *camshift*. Primjerice autori M. Mason i Z. Duric u svom radu [5] modeliraju objekt pomoću više histograma i to nad značajkama boje i smjera gradijenta. Pomoću tog modela, na sličan način kao što je prikazano u ovom radu, izračunavaju funkciju gustoće vjerojatnosti te nad njom primjenjuju algoritam *camshift*. Ovakav model u obzir uzima i grube prostorne odnose boja te oblik konture.

Druga moguća promjena u algoritmu je korištenje postupka izravne usporedbe histograma objekta te histograma okoline svakog pojedinog slikovnog elementa. Razlika između ovog postupka te *histogram back projection* metode je u tome što se za svaki slikovni element provodi usporedba cijelih histograma, a ne samo primjena Bayes-ov teorema za proračun vjerojatnosti. Za usporedbu histograma ponovno je korišten *Bhattacharyya koeficijent*. Ovaj postupak je računalno značajno zahtjevniji, ali uz korištenje strukture *integral histogram*, te uz manju rezoluciju slike, moguća je implementacija koja se izvodi u stvarnom vremenu.

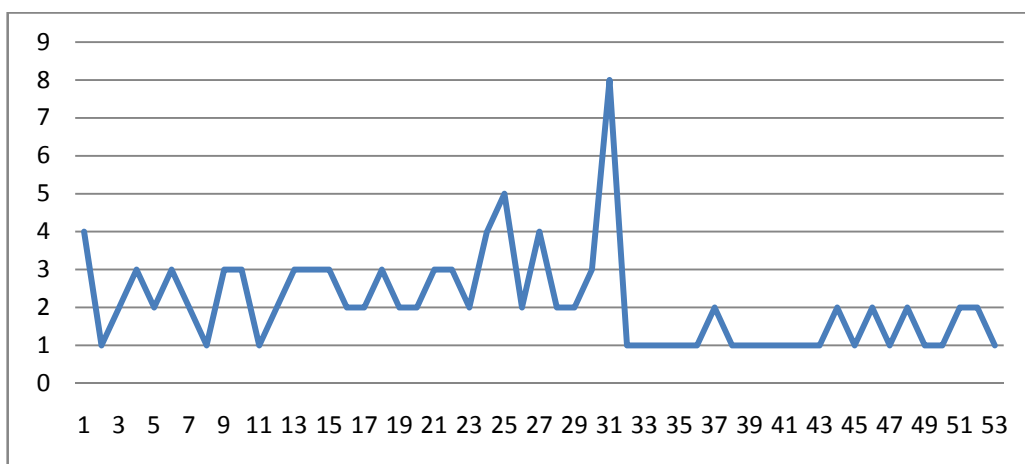
Posljedica složenije usporedbe je bolje praćenje prometnog znaka. S obzirom da je količina okolnog šuma i dalje vrlo visoka rezultati nisu zadovoljavajući. Prozor za označavanje objekta je značajno širi od prometnog znaka i obuhvaća područje iznad znaka. Nakon što prometni znak nestane iz scene, praćenje se nastavlja na području iznad znaka. Na *slici 21* prikazana je razdioba vjerojatnosti dobivena izravnom usporedbom histograma pomoću *Bhattacharyya koeficijenta*.



Slika 21. Razdioba vjerojatnosti prilikom praćenja prometnog znaka

Jedna od mogućih metoda izračuna razdiobe vjerojatnosti je i metoda čestičnog filtera (*particle filter*). Čestični filter je primjer *Monte Carlo* algoritma, preciznije rečeno metoda procjene razdiobe vjerojatnosti metodom uzimanja uzoraka. Razdioba vjerojatnosti aproksimira se pomoću skupa čestica.

Performanse algoritama najbolje se mogu prikazati grafom koji opisuje broj iteracija *abcshift* algoritma potrebnih za postizanje konvergencije. *Slika 22.* na apscisi prikazuje redni broj slikovnog okvira prilikom praćenja prometnog znaka, a ordinata grafa prikazuje potreban broj iteracija za pronalazak ekstrema funkcije razdiobe.



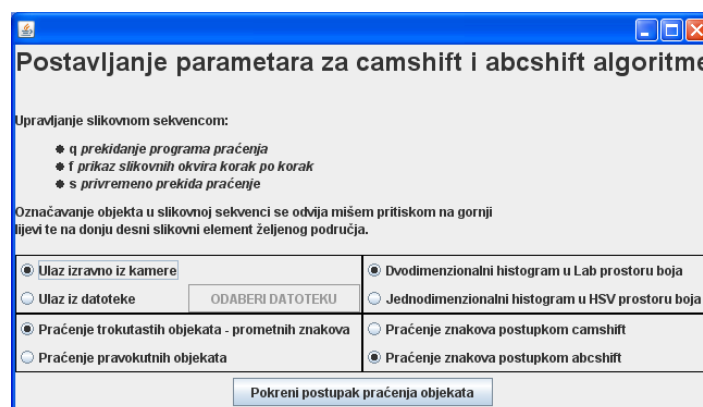
Slika 22. Broj iteracija abcshift algoritma

4. Implementacija

Algoritmi opisani u ovom radu implementirani su u programskom jeziku *C++*. Prilikom postupka učitavanja slikovne sekvence iz datoteke odnosno kamere korištena je biblioteka *OpenCV*. Implementacija opisanih algoritama izvedena je u sklopu završnog rada te se cjelokupni izvorni kod te izvršna datoteka nalaze u prilogu.

Prilikom obrade slikovne sekvence program omogućava obilježavanje objekta kojeg je potrebno pratiti. Nakon odabira prikazuje se izračunati histogram objekta (*slika 2*). Tokom postupka praćenja prikazana je izvorna slikovna sekvenca sa crvenim pravokutnikom oko praćenog objekta (*slika 3*). Uz izvornu slikovnu sekvencu prikazana je i funkcije gustoće vjerojatnosti izračunata postupkom *histogram back-projection* te na njoj prozori za traženje nastali postupkom postupne konvergencije *camshift* odnosno *abcshift* algoritma (*slika 3*).

Implementacija omogućava dinamički odabir različitih histogramima te različitih prostora boja korištenih za modeliranje objekta. Program omogućava označavanje pravokutnih te trokutastih objekta. Prilikom označavanja prometnih znakova važan je odabir trokutastih objekata jer time izrađeni histogram vjernije modelira početni objekt te time opisani algoritmi pokazuju značajno kvalitetnije rezultate prilikom praćenja.



Slika 23. Prikaz grafičkog korisničkog sučelja

U sklopu rada razvijeno je i jednostavno korisničko sučelje napisano u programskom jeziku *Java*. Sučelje omogućava jednostavnu konfiguraciju *camshift* i *abcshift* algoritama za praćenje objekata u svrhu lakšeg testiranja mogućnosti koje nude navedeni algoritmi (*slika 21*). Moguće je odabrati ulaznu datoteku ili obradu slikovne sekvence izravno s kamere. Također na jednostavan način omogućena je promjena korištenog sustava boja te samog postupka praćenja između *camshift* i *abcshift* algoritma.

5. Zaključak

Praćenje objekata je proces određivanja položaja jednog ili više pokretnih objekata u vremenu. Zadatak algoritama za praćenje je analiza slikovnih sekvenci u svrhu izdvajanja korisne informacije. Primjene su vrlo raznolike, poglavito u aplikacijama koje se izvode u realnom vremenu. Praćenje osoba, sigurnosni nadzor, izrada kvalitetnijeg korisničkog sučelja pa čak i kompresija slikovnih sekvenci obuhvaća i postupke praćenja objekata.

Algoritmi opisani u ovom radu *mean-shift*, *camshift* te *abcshift* koriste informaciju o boji objekta, koje modeliraju histogramom, za izračun funkcije koja opisuje vjerojatnost položaja objekta u sceni. Upravo određivanjem ekstrema te funkcije pronalazi se najvjerojatniji položaj objekta.

Mean-shift je algoritam koji se koristi za pronalaženje najbližeg ekstrema funkcije gustoće vjerojatnosti u prostoru R^N . Algoritam se temelji na postupnom pomicanju točke u smjeru gradijenta funkcije. Postupak završava konvergencijom u maksimumu zadane funkcije.

Algoritam *camshift* je zasnovan na *mean-shift* algoritmu. Koristi se za nalaženje ekstrema skupa uzoraka, ali za razliku od *mean-shift*-a, radi se o algoritmu koji ima veću mogućnost prilagodbe promjenama. *Camshift* dinamički određuje veličinu prozora za traženje te se time bolje prilagođava nastalim promjenama, primjerice približavanju ili udaljavanju objekta od kamere. Dizajniran je s ciljem izrade sučelja između korisnika i računala, točnije praćenju lica korisnika. Unatoč tome, pokazalo se da je uz manje izmjene sposoban pratiti objekte u raznolikim primjenama.

ABCshift izgrađuje model pozadine u svakom koraku te time u određenim slučajevima značajno poboljšava mogućnosti *camshift* algoritma. To se poglaviti odnosi na praćenje objekata u kojima je i kamera pokretna. U slučaju da su u nastaloj pozadini i objektu nalazi slična razdioba boja *ABCshift* modeliranjem pozadine veću važnost pridaje slikovnim elementima koji razlikuju pozadinu od objekta te time značajno poboljšava praćenje. Istovremeno se javlja i mogućnost nastanka nestabilnosti pri kojem se prozor za traženje značajno smanji, ali taj

problem je uspješno riješen korištenjem metrike zasnovane na *Bhattacharyya* koeficijentu.

Ono što posebice izdvaja algoritme *Camshift* i *ABCshift* je mogućnost rada u realnom vremenu s vrlo niskom potrošnjom računalnih resursa. Oba algoritma imaju mogućnost obrade uz vrlo visoki *frame-rate*. Time se omogućava praćenje vrlo brzih pokreta objekta uz potrošnju minimalnih računalnih resursa. Istovremeno se omogućava rad ovih algoritama u složenijim sustavima u kojima je praćenja objekata samo jedan od potrebnih postupaka. Ono što izdvaja većinu *region based* algoritama, uključujući *mean-shift*, *camshift* te *ABCshift*, od *contour based* algoritama je mogućnost praćenja objekta u slučajevima tek djelomične vidljivosti te u slučaju visoke količine šuma. Dobro svojstvo navednih algoritama je i jednostavan i računalno efikasan postupak izrade modela, odnosno histograma objekta.

Postignuti rezultati algoritmom *camshift* uvelike odgovaraju rezultatima prikazanim u izvornom članku[1]. Dodatno, uz praćenje lica, provedeno je testiranje na nekoliko drugih primjera u kojima *camshift* pokazuje vrlo dobre rezultate praćenja uz izuzetno nisku potrošnju računalnih resursa.

Praćenje prometnih znakova se pokazalo kao značajno teži problem koji nastaju zbog pomične kamere te nejednolike razdiobe boja zadanog objekta. *Camshift* algoritam u tim uvjetima nije pokazao dobre rezultate praćenja. Puno bolji rezultati postignuti su *ABCshift* algoritmom. Naime postupak modeliranja pozadine izuzetno je važan u slučaju praćenja objekata s pomičnom kamerom te objekata koji imaju određen stupanj sličnosti u razdiobi boja s okolinom. *ABCshift* postupkom modeliranja pozadine više ističe upravo razlike u modelima pozadine i objekta te rezultira kvalitetnim praćenjem objekta.

6. Literatura

- [1] Bradski, G. R.: Computer Vision Face Tracking as a Component of a Perceptual User Interface, In Proc. of the IEEE Workshop on Applications of Comp. Vision, (1998) 214–219
- [2] Boyle, Michael: The effects of capture conditions on the CAMSHIFT face tracker. Report 2001, Department of computer science, University of Calgary, Alberta, Canada.
- [3] R. Stolkin, I. Florescu, G. Kamberov: „An adaptive background model for Camshift tracking with a moving camera“. Proc. International Conference on Advances in Pattern Recognition, 2007., pp. 147-151
- [4] D. Comaniciu, V. Ramesh: Real-Time Tracking of Non-Rigid object using Mean Shift, IEEE CVPR 2000.
- [5] Mason, M., Duric, Z.: Using histograms to detect and track objects in color video, Applied Imagery Pattern Recognition Workshop, AIPR 2001, 2001, pp. 154-159

7. Naslov, sažetak i ključne riječi

Naslov:

Praćenje objekata u slikovnim sekvencama algoritmom camshift.

Sažetak:

Praćenje objekata u realnom vremenu korištenjem algoritama *mean-shift*, *camshift* i *abcshift*. Algoritmi opisani u ovom radu koriste informaciju o boji objekta, koje modeliraju histogramom, za izračun funkcije koja opisuje vjerojatnost položaja objekta u sceni. Upravo određivanjem ekstrema te funkcije pronalazi se najvjerojatniji položaj objekta. Navedeni algoritmi testirani su u nekoliko različitih primjena s posebnim naglaskom na praćenje prometnih znakova. Testiranja su pokazala da prilikom praćenja prometnih znakova *abcshift* algoritam, korištenjem modela objekta i njegove trenutne pozadine, pokazuje najbolje rezultate od svih opisanih algoritama. Dodatan izračun modela pozadine u svakom koraku ne utječe na računalnu učinkovitost, a pridonosi boljoj mogućnosti praćenja u slučaju pomične kamere.

Ključne riječi: camshift, mean-shift, abcshift, praćenje objekata, praćenje prometnih znakova u realnom vremenu.

Title:

Object tracking using Camshift algorithm

Abstract:

Description of real time object tracking using *mean-shift*, *camshift* and *abcshift* algorithm is given. Described algorithms use histogram modeled color information about objects from which they create probability distribution over an image region. Finding a mode of probability distribution yields maximum likelihood position of an object. Algorithms are tested in different scenarios. Testing with traffic sign objects proved superior performance of *abcshift* algorithm over other described algorithms. *Abcshift* uses object modeling and background modeling to separate object from its background. Background model is continuously relearned for every frame with minimal computational cost and also enables good tracking performance with a moving camera.

Keywords: camshift, mean-shift, abcshift, object tracking, real-time traffic sign tracking