

## PRIMENA ALGORITAMA ZA DETEKCIJU REFERENTNIH ELEMENATA OBJEKATA U VIDEO SEKVENCI

Bratislav Milovanović<sup>1</sup>, Jugoslav Joković<sup>1</sup>, Nikola Dojčinović<sup>2</sup>, Nikola Božinović<sup>3</sup>

<sup>1</sup>Elektronski fakultet u Nišu

<sup>2</sup>Inovacioni centar ETF Beograd

<sup>3</sup>Motion DSP Niš

**Sadržaj:** U radu su prezentovani metodi za detekciju referentnih elemenata objekata na slici (*feature detection*), koji se koriste za predstavljanje scene u video sekvenci. Predstavljeni su algoritmi za detekciju različitih tipova elemenata i razmatrani su aspekti njihove realizacije i primene na odgovarajućim primerima. Posebno su razmatrani savremeni algoritmi za detekciju tzv. blobova. Data je analiza ponašanja algoritama u kontekstu ponovljivosti, mogućnosti uspostavljanja i tačnosti korespondencije.

**Ključne reči:** Referentni elementi, Detekcija ivice, Detekcija ugla, SIFT, MSER

### 1. Uvod

Intenzivni razvoj uređaja za zapisivanje i prezentaciju videa, praćen istovremenim smanjenjem cena, odnosno povećanjem njihove dostupnosti korisnicima, postavio je i znatno veće zahteve kod samog video procesiranja, naročito u smislu smanjenja kompleksnosti izračunavanja. Bolje performanse procesora omogućile su ekstrakciju informacija iz video signala u realnom vremenu. Ekspertski sistemi zasnovani na obradi video signala postaju sve samostalniji i efikasniji, obavljajući sve složenije zadatke. Međutim, scena snimljena video kamerom i dalje predstavlja redundantno okruženje nad kojim je teško vršiti procesiranje u realnom vremenu, pa su iz tog razloga razvijene tehnike predstavljanja scene na osnovu informacija o samoj njenoj strukturi.

Detekcija referentnih elemenata (*feature detection*) predstavlja jedan od najviše promenjivanih postupaka za opis scene. Određivanjem referentnih elemenata možemo apstrahovati scenu i učiniti je manje redundantnom. Algoritmi koji su zbog kompleksnosti izračunavanja nad celom scenom bili neprimenjivi, nad setom referentnih elemenata daju zadovoljavajuće rezultate u realnom vremenu: razvijeni su metodi za indirektnu estimaciju pokreta [1]; kompleksna preslikavanja između scena se izračunavaju na osnovu seta odgovarajućih referentnih elemenata [2]; omogućeno je i povezivanje strukture scene sa višestrukim projekcijama, odnosno izračunavanje fundamentalne matrice i trifokalnog tenzora [2], koje daje osnovu za trodimenzionalni opis scene.

Univerzalna i opšteprihvaćena definicija referentnih elemenata na slici ne postoji. Koje tačke konstituišu referentne elemente, kao i njihov tip, u velikoj meri zavise od same definicije problema i konkretne aplikacije. Najefikasniji način za opisivanje objekata je preko osnovnih geometrijskih apstrakcija, kao što su ivice, uglovi, stranice, odnosno površine, kod nepoligonalnih i amorfnihi formi, itd. S toga se kao referentni elementi nekog objekta uzimaju upravo ovi elementi. Pored toga što ovakvo shvatanje najviše odgovara ljudskoj percepciji objekta, takođe daje matematičku osnovu za opis referentnog elementa. Ukoliko sliku, odnosno frejm u video sekvenci, shvatimo kao dvodimenzionalnu neparametarsku funkciju  $f(x,y)$  dobijenu preslikavanjem scene  $S(X,Y,Z)$  na dvodimenzionalnu projekciju, onda ivice, uglove i strane, koji predstavljaju diskontinuitete u sceni u fizičkom prostoru  $R^3$ , možemo detektovati kao diskontinuitete u projekciji  $f(x,y)$ . U zavisnosti od fizičkih karakteristika objekta, diskontinuiteti će se manifestovati na različite načine. Tako, npr. ivice možemo detektovati kao nagle promene strmine funkcije  $f(x,y)$ , uglove kao presek dve ivice itd.

Osnovni zadatak algoritama za detekciju referentnih elemenata objekata na slici jeste da pronađe elemente koji se istaknutošću i perzistentnošću razlikuju od ostalih elemenata objekta. Pronalaženje referentnih elemenata je inicijalni korak za veliki broj algoritama za procesiranje videa ili slike koji koriste referentne elemente da bi opisali samu scenu. Položaj i međusobni odnos referentnih elemenata na sceni može pružiti mnogo informacija o geometriji scene. Promena pozicija referentnih elemenata kao i njihovog međusobnog odnosa predstavlja jedan od najefikasnijih mehanizama za detektovanje promena na sceni. Apstrakovanje objekata preko referentnih elemenata smanjuje redundantnost i numeričku kompleksnost njihovog predstavljanja i operacija nad njima. Osobine objekata se izvode iz osobina pojedinihi elemenata; npr. koincidentno pomeranje elemenata koji opisuju određeni objekat ukazuje na kretanje objekta, dok klasterizacija pokreta tačaka koje čine određeni objekat ukazuje na deformaciju objekta.

Prethodno opisani postupci podrazumevaju da su referentni elementi detektovani u različitim frejmovima video sekvence (slikama) projekcije istog elementa u prostoru. Da bi ispunili ovu pretpostavku, moramo osigurati da algoritam detekcije referentnih elemenata na različitim projekcijama iste scene detektuje iste referentne elemente. Ako pretpostavimo da je isti element prisutan na dve projekcije iste scene,  $F_1 = f_1(x,y)$  i  $F_2 = f_2(x,y)$ , vezu između njih možemo predstaviti matričnom jednačinom:

$$F_2 = H * F_1$$

gde  $H$  predstavlja transformaciju projekcije scene, čija priroda nije unapred poznata. Kvalitet algoritma za detekciju referentnih elemenata ogleđa se u imunosti na transformaciju  $H$ . Iako je većina algoritama imuna na translacionu i rotacionu transformaciju, imunost na tzv. *affine* transformaciju kompenzuje se kompleksnošću algoritma za detekciju, te njegovim temporalnim karakteristikama.

Upoređivanje referentnih elemenata je neophodno kako bi se utvrdio njihov identitet. Skoro da svaki algoritam za detekciju koristi sopstveni deskriptor elementa koji se, uglavnom, bazira na informacijama o samom referentnom elementu i njegovom lokalnom susedstvu. U zavisnosti od tipa referentnih elemenata, kao i od konkretne aplikacije i željene preciznosti detekcije, varira složenost deskriptora.

Kao jedni od osnovnih postupaka video procesiranja, algoritmi za detekciju referentnih elemenata pružaju nam koristan okvir za dalju analizu video sekvenci ili

ekstrakciju informacija. U drugom poglavlju će biti opisani algoritmi za detekciju ivica i uglova, dok će u trećem poglavlju detaljnije biti prikazana dva algoritma za detekciju kompleksnijih struktura. Aspekti primene opisanih algoritama razmatrani su na odgovarajućim video snimcima visoke rezolucije. Algoritmi su realizovani i testirani na računaru bez GPU akceleracije (procesor Core2Duo 2.2GHz, 3 GB RAM).

## 2. Algoritmi za detekciju referentnih elemenata na slici

Kao što je istaknuto, primarni zadatak algoritama za detekciju referentnih elemenata na slici jeste pronalaženje vizuelno istaknutih tačaka koje su pogodne za opis scene. U zavisnosti od tipa elementa, postupci za njegovu detekciju se znatno razlikuju.

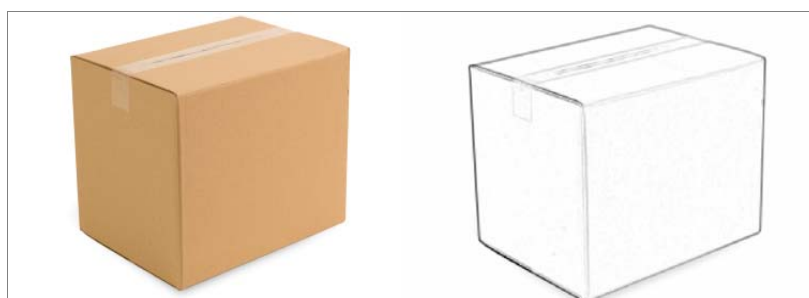
### 2.1. Detekcija ivica

Ivicu predstavljaju tačke na granici (ivici) između dva regiona na slici. Oblik ivice može biti različit i zavisi od oblika samog objekta. Na slici se ivice detektuju kao nagle promene u intenzitetu, odnosno kad skup povezanih tačaka gde dolazi do značajne promene magnitude gradijenata. Većina algoritama polazi od ovog svojstva, povezujući tačke velike magnitude gradijenta, dajući opis ivice. Kao dobri deskriptori mogu se koristiti orijentacija ivice, oblik i izraženost.

Među prvim radovima vezanim za detekciju ivice je *Canny*-jev algoritam [3], baziran na korišćenju gradijentnih operatora. Pozicije na kojima bi strmina gradijenta bila veća od praga smatrane bi ivicama. U radu su predloženi postupci za postprocesiranje, kao i računanje vrednosti praga strmine sa histerezisom. Sažeto, algoritam se nakon redukcije šuma preko *Gaussian* filtara svodi na pronalaženje intenziteta i strmine gradijenata; ako sa  $G_y$  i  $G_x$  označimo horizontalni i vertikalni gradijent, intenzitet gradijenta i pravac ivice mogu biti izračunati kao:

$$G = \sqrt{G_x^2 + G_y^2} \quad \Theta = \arctan(G_y / G_x)$$

U narednom koraku vrši se odstranjivanje ne-maksimuma, tj. odstranjuju se sve one ivice kod kojih je komponenta gradijenta normalna na pravac ivice slabijeg intenziteta od komponente koja se poklapa sa pravcem ivice. Na kraju se određuje prag sa histerezisom.



Slika 1: Originalna slika i slika sa izdvojenim ivicama

Razvijeno je više varijacija Canny-jevog algoritma koje se međusobno razlikuju u operatoru za određivanje direkcionalnih gradijenata,  $G_x$  i  $G_y$ , među kojima se najčešće koriste *Sobel-ov*, *Roberts-ov* [4] i *Prewit-ov* [5] operator. Na Slici 1 prikazani su rezultati detekcije ivica *Canny-jevim* algoritmom korišćenjem *Sobel-ovog* operatora.

## 2.2. Detekcija uglova

Algoritmi za detekciju uglova su razvijeni posle algoritama za detekciju ivica. Zasnovani su na analizi ivica, odnosno na detekciji brze promene pravca gradijenta duž ivice. Kasniji algoritmi su redukovali detekciju uglova izbegavajući detekciju ivica, direktno analizirajući zakrivljenost gradijenta slike, odnosno smatrajući ugao tačkom u čijoj okolini postoje dva ili više dominantna različita pravca gradijenta. Zapaženo je da se ovim pristupom detektuju i referentni elementi koje nisu uglovi u tradicionalnom smislu (npr. male svetle tačke na tamnoj pozadini vrlo verovatno će biti detektovane). Ove tačke se nazivaju tačkama od interesa (*interest points*). Danas se u literaturi referentne oblasti pojmovi “ugao” i “tačka od interesa” koriste kao sinonimi.

Prvi algoritmi su tretirali uglove kao tačke sa niskom samo-sličnošću [7]. Za svaki piksel je uvedena metrika sličnosti sa lokalnim susedstvom datog piksela sa lokalnim susedstvom koje ostvaruje što veće preklapanje. Radi uprošćavanja matematičkog aparata, *Moravec* je predložio sumu kvadrata razlike [7]. *Harris* i *Stephens* su upotrebili drugačiju metriku za izračunavanje sličnosti [8], koju nisu izračunavali na lokalnom susedstvu, već u pravcu diferenciranja. Iako se u [6] govori o autokorelaciji, matematički je u pitanju težinska suma razlike kvadrata.

$$S(x, y) = \sum \sum w(u, v) (F(u+x, v+y) - F(u, v))^2 \quad (1)$$

gde  $F$  predstavlja sliku, a  $w(u, v)$  težinsku funkciju. Ukoliko se za  $w(u, v)$  izabere funkcija sa kružnim kernelom, kao što je *Gaussian*, postiže se izotropnost, koja ne krasi *Moravec-ov* metod. Ako pretpostavimo da su  $u$  i  $v$  mali pomeraji kernela pri izračunavanju, desni deo jednačine (1) se može razviti u *Taylor-ov* red, koristeći parcijalni izvod  $F$  po obe ose slike, pa dobijamo:

$$S(x, y) \approx \sum \sum w(u, v) (F_x(u, v)x + F_y(u, v)y)^2$$

odnosno, u matričnom obliku

$$S(x, y) \approx \begin{pmatrix} x & y \end{pmatrix} A \begin{pmatrix} x \\ y \end{pmatrix}$$

gde je  $A$  strukturni tenzor:

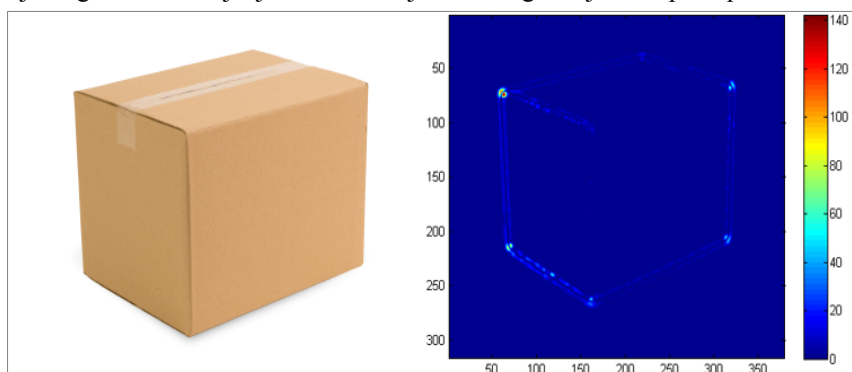
$$A = \sum \sum w(u, v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (2)$$

Matrica  $A$  je u literaturi poznata kao *Harris-ova* matrica. Ukoliko funkcija  $S$  u tački  $(x, y)$  ima veliku varijaciju vrednosti u pravcu vektora  $\begin{pmatrix} x \\ y \end{pmatrix}$ , onda tačku  $(x, y)$  možemo smatrati uglom, odnosno tačkom od interesa. Posmatrajući samo u tački koja predstavlja ugao, matrica  $A$  treba da ima visoke sopstvene vrednosti. Analizom sopstvenih vrednosti  $\lambda_1$  i  $\lambda_2$ ,

zaključujemo da je pronađen ugao ako  $\lambda_1$  i  $\lambda_2$  imaju velike pozitivne vrednosti. Ako su  $\lambda_1 \approx 0$  i  $\lambda_2 \approx 0$ , onda u tački  $(x,y)$  ne postoji ugao, a ako je  $\lambda_1 \approx 0$  i  $\lambda_2$  ima neku pozitivnu vrednost, pronađena je ivica. Kako izračunavanje sopstvenih vrednosti matrice zahteva veliki broj izračunavanja, *Harris* i *Stephens* su predložili metriku čiji tok funkcije odgovara prethodno iznetim zaključcima:

$$M_c = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(A) - k * \text{trace}(A)^2$$

gde su  $\lambda_1$  i  $\lambda_2$  sopstvene vrednosti matrice  $A$  definisane u (2),  $\det(A)$  označava determinantu matrice  $A$  i  $\text{trace}(A)$  označava trag matrice  $A$ . Podesivim parametrom  $k$  možemo kontrolisati osetljivost detekcije. Tipične vrednosti za  $k$  se kreću u intervalu 0.04 - 0.15. Na Slici 2 je prikazana originalna slika i vrednost dvodimenzionalne funkcije  $M_c$ . Može se primetiti da funkcija ima maksimume na uglovima, dok u delovima sa uniformnom raspodelom ima nultu vrednost. Takođe, zapaža se da je vrednost funkcije na ivicama različita od nule, ali opet daleko niža od vrednosti u uglovima. Uvođenjem praga, otklanjamo neželjene pozitivne vrednosti, koje delom imaju poreklo od kvantizacionog šuma, i izdvajamo tačke od interesa. Ukoliko se uvede dvostruki prag, postoji mogućnost izdvajanja i klasifikacije ivica i uglova jednim postupkom.



Slika 2: Originalna slika i vrednost 2D funkcije  $M_c$  izračunate nad slikom

### 3. Detekcija na više skala

Detekcija blobova (eng. *blob* – mehur, kap) daje daleko više informacija po pitanju strukture regiona od ivica i uglova. Blob se definiše kao region koji se najbolje može opisati kružnom strukturom. Idealno, pod blobom podrazumevano elipsu slabog ekcentriciteta, približno krug. Matematička interpretacija bloba najviše odgovara dvodimenzionalnom *Gaussian*-u sa  $\mu_x = 0$  i  $\mu_y = 0$ :

$$g(x, y) = \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right) \quad (3)$$

Slično kao i kod prethodno opisanih postupaka, mehanizam detekcije blobova se zasniva na merenju sličnosti elemenata slike sa modelom bloba. S obzirom da se radi o kompleksnijem modelu, poređenje se vrši preko korelacionih metoda. Kao rezultat poređenja dobija se dvodimenzionalna funkcija čiji lokalni maksimumi predstavljaju referentne elemente.

Opisani postupak detekcije meri sličnost elemenata slike sa modelom definisanim u (3). Ukoliko poredimo sličnost modela sa više blobova različitih dimenzija, najveću sličnost sa modelom pokazaće onaj blob čije se dimenzije najmanje razlikuju od dimenzije modela definisanih sa  $\sigma$ . Na ovaj način blobovi čije dimenzije nisu merljive sa dimenzijama modela, neće biti detektovani. Rešenje ovog problema predstavljaju metodi zasnovani na analizi na više različitih skala.

### 3.1. Prostor skala

Analizu slike na više različitih skala i konstrukciju prostora skala definisao je *Lindeberg* [9,10]. Prostor skala omogućava analizu slika na različitim skalama potiskujući između susednih skala detalje manje od  $\sigma^{1/2}$ , gde je  $\sigma$  faktor skaliranja između dve skale. Na ovaj način se omogućava da elementi različitih dimenzija budu dominantni na različitim skalama. Najčešće korišćeni prostor skala je linearni (*Gauss-ov*). Dobija se filtriranjem slike normalizovanim *Gauss-ovim* kernelom:

$$g(x, y, \sigma) = \frac{1}{(2\pi\sigma)} \exp\left(-\frac{(x^2 + y^2)}{\sigma}\right) \quad (4)$$

$$L(:, :, \sigma) = g(x, y, \sigma) * f(x, y)$$

gde \* predstavlja konvoluciju, a  $\sigma$  konstantni faktor koji gradira od manjih ka većim vrednostima. Kako povećanje  $\sigma$  povećava i dimenzije *Gauss-ovog* kernela, izračunavanja konvolucije postaju sve zahtevnija. Zbog toga  $\sigma$  varira samo u okviru jedne oktave, posle čega se vrši skaliranje dimenzija slike, odnosno ponovno semplovanje sa duplom većim faktorom, kako bi se dobila slika dimenzija koje su polovina dimenzija prethodne slike. Na ovaj način, koristeći opet iste kernele kao u prethodnoj oktavi, daleko smanjujemo broj izračunavanja, jer je broj piksela slike koja se konvoluirala sada četiri puta manji. Pored toga što omogućava analizu struktura različitih dimenzija, korišćenje prostora skala redukuje inter-pikselski i kvantizacioni šum koji se iznova javlja svaki put kada se slika resempljuje. Na Slici 3 prikazana je slika *mandi.tif* iz standardnog paketa Matlaba sa ugrađenim šumom i prostor skala izvedenom na 8 oktava sa faktorom skaliranja  $\sigma = 5$ .

### 3.2. SIFT

Postoji veći broj algoritama koji, koristeći prostor skala, nastoje da gradijentnim metodama detektuju referentne elemente u obliku blobova. Jedan od najkorišćenijih jeste tzv. SIFT (*Scale-Invariant Features*), u okviru koga je razvijen i deskriptor za opis bloba koji omogućava poređenje referentnih elemenata.

Početni korak SIFT algoritma je izračunavanje prostora skala. Zatim se izračunava razlika *Gaussian-a* (*DoG - Difference of Gaussians*) u okviru svake oktave. Ekvivalentan postupak opisan u [10] koristi umesto DoG normalizovani *Laplacian*.

$$DoG(x, y, \sigma) = L(x, y, k_1\sigma) - L(x, y, k_2\sigma) \quad (5)$$

gde se  $L$  izračunava na osnovu jednačine (4).



Slika 3: *Prostor skala na 8 oktava sa  $\sigma = 5$*

Nakon izračunavanja *DoG*, referentni elementi se dobijaju pronalaženjem lokalnih ekstremuma (maksimuma i minimuma) u okviru svake skale. Ekstremumi se pronalaze poređenjem svakog piksela sa lokalnim susedstvom (osam okolnih piksela). Pronađeni ekstremumi se porede sa vrednostima piksela na istim pozicijama, ali u susednim skalama. Ukoliko ekstremum i dalje predstavlja maksimum ili minimum, proglašava se kandidatom za referentni element. Klasa kandidata referentnih elemenata dodatno se proređuje eliminacijom tačaka sa niskim kontrastom, odstranjivanjem svih onih čiji je intenzitet manji od 3% od maksimalne vrednosti. Često se kao prag uzima i  $\mu_0$ .

*DoG* će imati i jak odziv po ivicama. Efektan način za odstranjivanje kandidata koji su pozicionirani na ivicama je preko *Hessian*-ova matrice:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

Ukoliko je jedna sopstvena vrednost matrice *H* približno jednaka 0, a druga pozitivna i velika, onda smatramo da se tačka nalazi na ivici. Pristup je sličan kao kod *Harris*-ovog operatora, s tim što *Harris* koristi strukturni tenzor, a ne *Hessian*-ovu matricu. Koristeći odnos  $r = \lambda_1 / \lambda_2$  može se pokazati da je

$$R = \text{Tr}(H)^2 / \det(H) = (r+1)^2 / r \quad (6)$$

Ako smatramo da vrednost  $r > 10$  označava da se tačka kandidat nalazi na ivici, onda odbacujemo sve tačke kandidate za koje je odnos  $R > 12.1$ . Tačke kandidati koji su prošli eliminaciju zbog slabog kontrasta i lokalizacije na ivici predstavljaju referentne elemente. Na Slici 4 se nalazi slika sa blobovima detektovanim SIFT algoritmom.

Opis referentnih elemenata SIFT vrši preko histograma orijentacije u 16x16 susedstvu. Izračunava se dominantna orijentacija, a zatim se oko nje računaju orijentacije subregiona dimenzija 4x4. Deskriptor postaje vektor čiji su elementi vrednosti svih ovih histograma. Kako ima 4x4=16 histograma od kojih se svaki sastoji od 8 elementa, deskriptor ukupno ima 128 elemenata.



Slika 4: Referentni elementi detektovani SIFT algoritmom

### 3.3. MSER

Svi prethodno opisani algoritmi koriste gradijente za detekciju referentnih elemenata. MSER (*Maximally Stable Extremal Regiona*) polazi od algoritma za segmentaciju slike [9]. Regioni koje MSER smatra referentnim tačkama su oni regioni koji pokazuju najveću stabilnost.

Algoritam polazi od sortiranja svih piksela slike po intenzitetu. Vršiti se izdvajanje niza pomoću praga koji menja vrednosti od najmanjih do najviših intenziteta na slici. Izdvajaju se pikseli koji imaju višu vrednost od piksela koji imaju nižu vrednost od praga. Na taj način se formiraju regioni koji se povećanjem praga međusobno spajaju, sve dok se ultimativno se spoje u jedan region. Pri tom procesu se svakom regionu dodeljuje jedinstveni identifikator. Kada se regioni spoje, region manje površine preuzima identifikator većeg regiona.

Stabilnost regiona se definiše kao konstantnost površine regiona sa promenom praga. Ako sa  $R$  označimo posmatrani region, a sa  $R \pm \Delta$  regione koji su dobijeni povećanjem (smanjenjem) praga za  $\Delta$ , uvodimo metriku  $\rho$ :

$$\rho(R, \Delta) = \frac{|R + \Delta| - |R - \Delta|}{|R|}$$

gde  $|R|$  označava površinu regiona. Ako je  $\rho(R, \Delta)$  manje od  $\rho(R + \Delta, \Delta)$  i  $\rho(R - \Delta, \Delta)$ , region  $R$  smatramo maksimalno stabilnim ekstremnim regionom, odnosno referentnim elementom.



Ceo proces se ponavlja i na komplementarnoj slici. Na Slici 5 prikazani su rezultati detekcije MSER algoritmom. Na slici levo, plusevima su označeni referentni elementi dobijeni primenom MSER-a na originalnoj slici, dok su na slici u sredini prikazani rezultati dobijeni primenom MSER-a na komplementarnoj slici. Krajnje desna slika prikazuje objedinjene rezultate detekcije MSER algoritmom.

Univerzalnost MSER-a ogleda se u tome da podjednako dobro detektuje blobove kao i amorfne strukture. Pored toga, MSER obezbeđuje odličan mehanizam za deskripciju, jer kao pamti sve piksele regiona, te je poređenje znatno olakšano. Za opis se najčešće koriste momenti drugog reda.



Slika 5: Detekcija referentnih elemenata MSER algoritmom

#### 4. Zaključak

U radu su opisani algoritmi za detekciju referentnih elemenata. U zavisnosti od konkretne potrebe za detekcijom određenog tipa elemenata, koriste se odgovarajući detektori [12]. Metrika koja je korišćena za komparativnu analizu zasniva se na ponovljivosti detekcije istih referentnih elemenata pod različitim transformacijama, pronalaženju korespondencija i tačnošću pronađenih korespondencija. Transformacije nad slikom na čiju imunost su detektori analizirani odnose se na promenu parametara za strukturnu i teksturisanoj sceni (promena tačke gledišta, promena skale – zumiranje, zamagljenje), kao i kompresija i promena iluminacije. Iste referentne elemente na slikama koje su pretrpele transformaciju najbolje je detektovao MSER skoro u svim slučajevima. Takođe, preko MSER deskriptora je pronađeno i najviše korespondencija. Međutim, najveći broj tačnih korespondencija ostvaruje se preko detektora koji koriste *Hessian* i pri tome su otporni na tzv. *affine* transformaciju (modifikovani SIFT).

Korišćenje detektora referentnih elemenata u video sekvencama nameće ograničenje vremena izvršenja, za rad u realnom vremenu. Arhitektura današnjih računara omogućava ubrzanje procesa kroz višestruku paralelizaciju podelom zadataka na više procesorskih jezgara. SIFT koristi četiri kompleksna izračunavanja: određivanje lokalnog maksimuma, konvolucija pri filtriranju, izračunavanje *DoG* i *DoH*. S obzirom da postoje algoritmi koji paralelizuju izvršenje ovih zadataka, SIFT je pogodan za rad u realnom vremenu. Izvršenje SIFT-a zahteva oko 18ms. Sa druge strane, u literaturi ne postoji algoritam za paralelizaciju MSER-a. Postoje realizacije koje omogućavaju izračunavanje MSER-a u realnom vremenu [13], koje detektuju referentne elemente za približno 30ms. No, kako je detekcija referentnih elemenata postupak koji se izvršava u predprocesiranju, ovakvo izračunavanje ne ostavlja dovoljno vremena za dodatno procesiranje.

Buduća istraživanja biće usmerena ka ubrzanju algoritama, kako bi vreme izvršenja omogućilo dodatno procesiranje u realnom vremenu. Generalno, preciznost primenjenih algoritama po pitanju ponovljivosti detekcije i tačnosti uspostavljanja korespondencije su zadovoljavajući u odnosu na posprocesiranje.

## Literatura

- [1] P.H.S. Torr and A. Zisserman: "Feature Based Methods for Structure and Motion Estimation", *ICCV Workshop on Vision Algorithms*, pp. 278-294, 1999.
- [2] R. Hartley and A. Zisserman: *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003, UK
- [3] J. Canny, "A Computational Approach to Edge Detection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679-698, 1986.
- [4] L.G. Roberts, "Machine Perception of Three-Dimensional Solid", in *Optical and Electro-Optical Processing*, *IEEE Computer Society Press*, Los Angeles, 1977.
- [5] J.M.S. Prewit, "Object Enhancement and Extraction", In *Picture Processing and Psychopictorics*, *Academic Press*, New York, 1970.
- [6] T. Lindeberg, "Edge detection and ridge detection with automatic scale selection", *International Journal of Computer Vision*, 30, 2, pp 117-154, 1998.
- [7] H. Moravec "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover". *Tech Report CMU-RI-TR-3 Carnegie-Mellon University*, 1980
- [8] C. Harris and M. Stephens, "A combined corner and edge detector", *Proceedings of the 4th Alvey Vision Conference*. pp.147-151, 1988.
- [9] T. Lindeberg, "Discrete Scale-Space Theory and the Scale-Space Primal Sketch", PhD thesis, Department of Numerical Analysis and Computer Science, KTH, 1991
- [10] T. Lindeberg "Feature detection with automatic scale selection", *International Journal of Computer Vision*, 30(2): pp 77-116, 1998.
- [11] J. Matas, O. Chum, M. Urba, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions." *Proc. of British Machine Vision Conference*, 2002.
- [12] K. Mikolajczyk and group of authors, "A comparison of Afine Region Detectors", *International Journal of Computer Vision*, 2006.
- [13] A. Vedaldi and B. Fulkerson, "VLFeat - An open and portable library of computer vision algorithms", in *Proceedings of the 18th ACM International Conference on Multimedia*, 2010.

**Abstract:** *This paper presents methods for feature detection used for scene apstraction in video sequences. Diferent type of feature detectors are presented and aspect of application and realization are considered on examples. Special attention was paid on state-of-the-art blob detection algorithms. Analysis of algorithms, in terms of repetability, posibility and accuracy of correspondences, was provided.*

**Keywords:** *features detection, edge detection, corner detection, SIFT, MSER*

## APPLICATION OF FEATURE DETECTION ALGORITHMS IN VIDEO

Bratislav Milovanović, Jugoslav Joković, Nikola Dojčinović, Nikola Božinović