Usporedba dinamičnih slika zasnovana na izdvajanju i usporedbi ključnih točaka

Pul, Matija

Master's thesis / Diplomski rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:388356

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

USPOREDBA DINAMIČNIH SLIKA ZASNOVANA NA IZDVAJANJU I USPOREDBI KLJUČNIH TOČAKA

Diplomski rad

Matija Pul

Osijek, 2016.

SADRŽAJ

1. UVOD 1
2. DETEKCIJA KLJUČNIH TOČAKA 2
2.1. SIFT algoritam 2
2.1.1. Detekcija ekstrema 3
2.1.2. Lokalizacija ključnih točaka 6
2.1.3. Određivanje orijentacije 7
2.1.4. Oblikovanje deskriptora 8
2.2. ORB algoritam 10
2.2.1. oFAST algoritam 11
2.2.2. rBRIEF algoritam 12
3. NOVI ALGORITAM ZA USPOREDBU DINAMIČNIH SLIKA 14
3.1. Način rada algoritma 14
3.2. Odziv ključnih točaka 16
3.3. Pogrešna detekcija ključnih točaka 18
3.3.1. Detekcija objekata na slici 19
3.3.2. Filtriranje pogrešno detektiranih ključnih točaka 20
3.4. Usporedba cijelih slika 21
3.5. Traženje objekta 21
4. VREDNOVANJE REZULTATA NOVOG ALGORITMA ZA USPOREĐIVANJE DINAMIČNIH SLIKA 23
4.1. Test tolerancije na različite smetnje 23
4.1.1. Tolerancija na rotaciju 23
4.1.2. Tolerancija na skaliranje 24
4.1.3. Tolerancija na čestice 25
4.1.4. Tolerancija na translaciju 26
4.1.5. Tolerancija na višestruke distorzije 27
4.2. Testiranje na stvarnim sekvencama 28
4.2.1. Glavni izbornik 29
4.2.2. Aplikacije 34
4.2.3. Postavke 35
4.2.4. TV vodič 37
5. ZAKLJUČAK 39
LITERATURA 40
SAŽETAK 41
ŽIVOTOPIS 43

1. UVOD

Usporedba dinamičnih slika jedan je od značajnijih problema u području računalnog vida koji su se pojavili nedavno, zbog naglog razvoja tog područja tehnologije. Računalni vid se bavi obradom digitalnih slika i video zapisa, tj. automatizacijom ljudskog vizualnog sustava. Ljudski mozak bez poteškoća uspoređuje dinamične slike, iako se one razlikuju po sadržaju, zato što osim sadržaja mozak uočava i kontekst slike. Računalo prepoznaje slike samo po njihovom sadržaju (elementima slike) te prema vlastitom saznanju u svijetu trenutno ne postoji algoritam koji uspješno oponaša ljudsku funkciju uspoređivanja konteksta slika. Cilj ovog rada je predstaviti jedno od rješenja tog problema, tj. predstaviti algoritam koji uspoređuje slike po njihovom kontekstu.

Moderne aplikacije koriste animacije i efekte kako bi učinile svoj sadržaj zanimljivijim i ugodnijim oku korisnika. Testiranje funkcionalnosti takvih aplikacija vrši se uspoređivanjem slika preuzetih s testne verzije aplikacije s referentnim slikama ispravne aplikacije. Zbog animacija i raznih efekata, sadržaj slike ovisi o trenutku u kojem je preuzeta, tj. slike preuzete iz aplikacije su dinamične slike. Dakle, testiranje funkcionalnosti svodi se na uspoređivanje dinamičnih slika.

U sklopu ovog rada razvijen je algoritam za usporedbu dinamičnih slika čija je svrha testiranje funkcionalnosti spomenutih aplikacija. Algoritam se temelji na izdvajanju i usporedbi ključnih točaka slike. Ključne točke omogućavaju algoritmu detekciju objekata koji se poslije koriste za usporedbu dinamičnih slika. Glavna funkcija izrađenog algoritma je usporedba cijelih dinamičnih slika uz dodatnu mogućnost detekcije pojedinačnih objekata na slici.

Kroz rad su objašnjeni problemi detekcije ključnih točaka, princip rada algoritma te analiza rezultata uspoređivanja dinamičnih slika. Drugo poglavlje objašnjava postojeće algoritme korištene za detekciju ključnih točaka. U trećem poglavlju objašnjen je algoritam izrađen u sklopu diplomskog rada koji predstavlja rješenje za problem usporedbe dinamičnih slika. Četvrto poglavlje prikazuje analizu rada algoritma te njegovu otpornost na problematične pojave kao što su rotacija, skaliranje, čestice i translacija. Zadnje, peto poglavlje donosi zaključke.

2. DETEKCIJA KLJUČNIH TOČAKA

Dinamične slike su slike čiji je sadržaj promjenjiv u vremenu, ali sam kontekst slike ostaje nepromijenjen. Na primjer, slike preuzete sa sigurnosnih kamera su dinamične slike. Kada bi se uzele dvije slike snimljene u različitim trenucima pomoću sigurnosne kamere usmjerene prema parkiralištu, tada bi većinski dio slike bio jednak, ali bi se mijenjali položaji ljudi i vozila. Sadržaj slika se mijenja, ali kontekst ostaje isti. Standardni način računalne usporedbe slika zasnovan na usporedbi elemenata slike (engl. *pixel*) uspoređuje sadržaj slike te se zbog toga ne može primijeniti za usporedbu dinamičnih slika. Ovaj rad predstavlja rješenje uspoređivanja konteksta slika korištenjem ključnih točaka. Ključne točke su istaknuti dijelovi slike opisani svojom okolinom. Sve dok su okoline ključnih točaka dovoljno jednake, računalo ih može prepoznati. Stoga, ključne točke su otporne na skaliranje, rotiranje te promjene svjetline i položaja. Također, u sklopu ovog rada izrađen je algoritam za uspoređivanje dinamičnih slika zasnovan na detekciji i uspoređivanju objekata.

2.1. SIFT algoritam

SIFT (engl. *Scale Invariant Feature Transform*) [1] je algoritam za detekciju ključnih točaka i određivanje pripadajućih deskriptora kojeg je izmislio David G. Lowe 2004. godine. Algoritam određuje ključne točke koje su otporne na promjene u veličini, rotaciji, osvjetljenju i perspektivi slike, što ih čini povoljnim za detekciju objekata kod dinamičnih slika. Sastoji se od četiri ključne faze [1]:

1. **Detekcija ekstrema** – prva faza pretraživanja svih skala i lokacija na slici. Učinkovito je implementirana korištenjem funkcije Gaussovih razlika (engl. *difference-of-Gaussian*) za identificiranje potencijalnih ključnih točaka.

2. **Lokalizacija ključnih točaka** – određivanje lokacije i skale svakog potencijalnog kandidata. Ključne točke se eliminiraju na temelju njihove stabilnosti (otpornosti na utjecaj smetnji).

3. **Određivanje orijentacije** – svakoj ključnoj točki određuje se jedna ili više orijentacija ovisno o smjerovima lokalnih gradijenata slike. Sve buduće operacije izvršavaju se na podacima slike koji su transformirani ovisno o pripadajućoj orijentaciji, skali i lokaciji za pojedinu ključnu točku, čime se postiže njihova neovisnost o tim transformacijama.

4. **Oblikovanje deskriptora** – mjere se lokalni gradijenti slike za odabranu skalu u području oko svake ključne točke. Izmjereni gradijenti transformiraju se u oblik koji je otporan na značajne distorzije i promjene u osvjetljenju.

2.1.1. Detekcija ekstrema

Cilj prve faze je odrediti sve potencijalne lokacije ključnih točaka detekcijom ekstrema na skalama. Detekcija ekstrema zasniva se na principu kaskadnog filtriranja te primjene algoritma za identificiranje potencijalnih lokacija koje će poslije biti detaljnije obrađene. Prvi korak je određivanje piramide Gaussovih razlika (engl. *Difference of Gaussian pyramid*), dalje u tekstu DOG piramida. DOG piramida sastoji se od skala koje se sastoje od oktava. **Skala** predstavlja izvornu sliku u određenoj rezoluciji zamućenu pomoću Gaussove funkcije. Prva skala može biti dvostruko veće, jednake ili dvostruko manje rezolucije od izvorne slike. No, svaka slijedeća skala ima dvostruko manju rezoluciju od prethodne skale. **Oktave** predstavljaju slike skale s različitim intenzitetima Gaussovog zamućenja. Prva oktava skale ima najslabiji intenzitet zamućenja, a svaka slijedeća oktava ima sve veći intenzitet zamućenja. Kao rezultat dobije se piramida različitih verzija izvorne slike koje se razlikuju po rezoluciji i intenzitetu zamućenja.

Koenderink [2] i Lindeberg [3] utvrdili su da je Gaussova funkcija jedina funkcija za učinkovito obrađivanje prostora skale i uzoraka. Stoga, prostor skale slike definira se kao funkcija $L(x, y, \sigma)$ koja je produkt konvolucije Gaussove fukcije $G(x, y, \sigma)$ s ulaznom slikom $I(x, y)$ u prostornoj domeni [1]:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{2-1}$$

gdje je

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}, \tag{2-2}$$

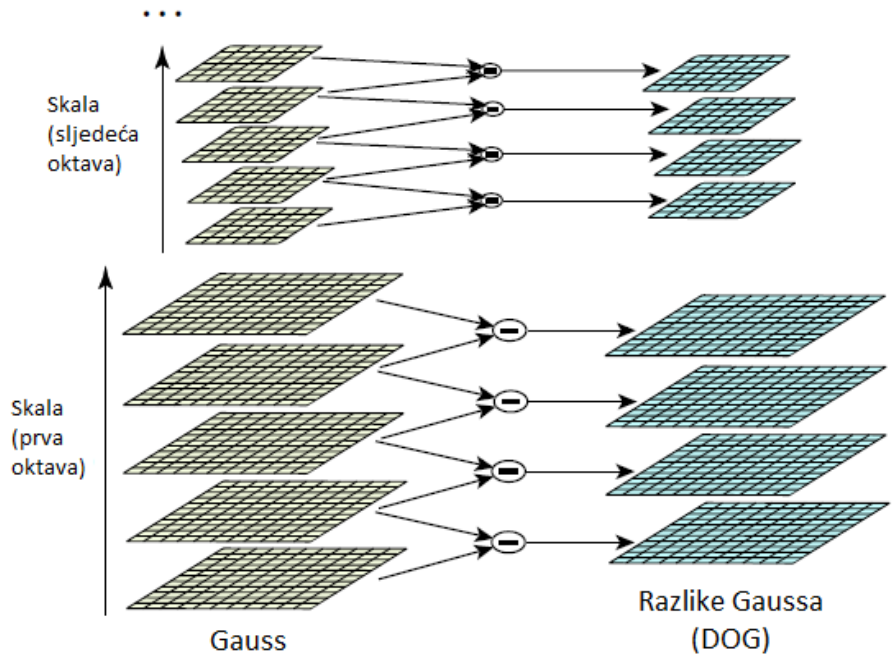
a x i y predstavljaju položaj promatrane točke u prostornoj domeni. Za učinkovitu detekciju lokacija stabilnih ključnih točaka u prostoru skala, Lowe [1] je predložio korištenje ekstrema u prostoru skala dobivenih konvolucijom DOG funkcije sa slikom $D(x, y, \sigma)$, koju je moguće dobiti iz razlike dviju susjednih skala odvojenih konstantnim faktorom množenja k [1]:

$$D(x, y, \sigma) = \big(G(x, y, k\sigma) - G(x, y, \sigma)\big) * I(x, y), \tag{2-3}$$

odnosno

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma). \tag{2-4}$$

Zamućene slike L svakako moraju biti obrađene tijekom izrade DOG piramide što, ovaj pristup čini osobito učinkovitim jer se D može dobiti jednostavnom razlikom slika kao što je prikazano na slici 2.1.



Slika 2.1. *Postupak određivanja DOG piramide* [1].

Osim toga, DOG funkcija daje približnu aproksimaciju skale normaliziranog Laplacea od Gaussa, $\sigma^2\nabla^2 G$, što je proučavao Lindeberg [3]. Lindeberg je pokazao kako je potrebno normalizirati Laplacea s faktorom σ^2 kako bi se postigla potpuna neovisnost o skali. Mikolajczyk [4] je izvršio detaljne eksperimentalne usporedbe te otkrio kako maksimumi i minimumi od $\sigma^2\nabla^2 G$ čine najstabilnije ključne točke u usporedbi s nizom drugih mogućih funkcija kao što su gradijent, Hessian ili Harris kutna funkcija. Jednadžba toplinske difuzije (parametrirana u smislu σ umjesto standardnog $t = \sigma^2$) najbolje opisuje odnos između D i $\sigma^2\nabla^2 G$ [1]:

$$\frac{\partial G}{\partial \sigma} = \sigma\nabla^2 G. \tag{2-5}$$

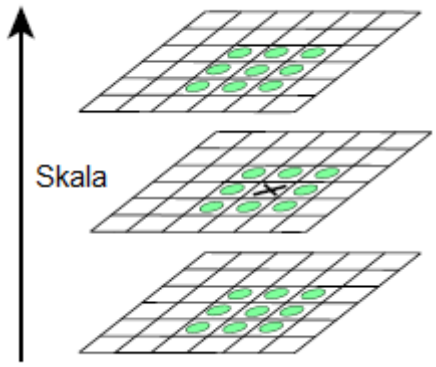
Iz izraza (2-5) uočljivo je da se $\nabla^2 G$ može dobiti iz aproksimacije konačnih razlika za $\frac{\partial G}{\partial \sigma}$, korištenjem razlike susjednih skala s $k\sigma$ i σ [1]:

$$\sigma\nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, a, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \tag{2-6}$$

iz čega slijedi,

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2\nabla^2 G. \tag{2-7}$$

Kada DOG funkcija ima skale koje se razlikuju za konstantni faktor, tada već uključuje σ^2 normalizaciju potrebnu za neovisnost Laplacea o skali. Faktor $(k-1)$ u jednadžbi je konstantan na svim skalama te ne utječe na lokacije ekstrema. Pogreška aproksimacije približava se nuli kako se *k* približava 1, ali u praksi se pokazalo da aproksimacija nema značajan utjecaj na lokalizaciju ili stabilnost detekcije ekstrema čak i kod značajnih razlika u skali, kao što je $k = \sqrt{2}$. Slika 2.2. prikazuje učinkovit pristup konstrukciji $D(x, y, \sigma)$[1].



Slika 2.2. *Postupak detekcije ekstrema* [1].

Početna slika inkrementalno se zamućuje pomoću Gaussove funkcije kako bi se proizvele slike s konstantnim faktorom razlike *k* u prostoru skale. Svaka oktava skale dijeli se na *s* intervala gdje je *s* cijeli broj, tako da se dobije $k = 2^{\frac{1}{s}}$. Potrebno je proizvesti $s + 3$ slika u stogu zamućenih slika za svaku oktavu da bi konačna detekcija ekstrema pokrivala kompletnu oktavu. Susjedne slike skale oduzimaju se radi generiranja DOG slika prikazanih na slici 2.2. Kada je obrađena kompletna oktava, uzimaju se slike s dvostruko većom inicijalnom vrijednošću σ (dvije slike s vrha stoga) te se uzorkuju s dvostruko manjim brojem uzoraka u vertikalnoj i dvostruko manjim brojem uzoraka u horizontalnoj dimenziji. Preciznost uzorkovanja je relativna

prema σ te se ne razlikuje od preciznosti na početku prethodne oktave, a broj potrebnih operacija je značajno smanjen [1].

2.1.2. Lokalizacija ključnih točaka

Nakon što je pronađen kandidat za ključnu točku postupkom uspoređivanja elementa slike s njegovim susjednim elementima, slijedi detaljno oblikovanje kandidata prema okolnim podacima za lokaciju, skalu i odnos glavnih zakrivljenosti. Dobivene informacije omogućavaju odbacivanje kandidata koji imaju slab kontrast (visoka osjetljivost na šum) ili su slabo lokalizirani uzduž ruba [1].

Početna implementacija ovog pristupa [5] oblikovala je kandidate samo prema lokaciji i skali središnje točke uzorka. No, Brown je razvio metodu [6] za pridruživanje 3D kvadratne funkcije lokalnim točkama uzorka kako bi odredio interpoliranu lokaciju maksimuma. Eksperimenti su pokazali značajno poboljšanje u stabilnosti i postotku poklapanja. Brownov pristup koristi Taylorov red funkcije prostora skale, $D(x, y, \sigma)$, pomaknute tako da se središte nalazi u točki uzorka [1]:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \tag{2-8}$$

gdje se D i njene derivacije pregledavaju u točki uzorka s pomakom $X = (x, y, \sigma)^T$[1]. Lokacija ekstrema, $\hat{x}$, dobije se određivanjem derivacije funkcije s obzirom na x i izjednačavanjem s nulom, čime se dobije [1]:

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x}. \tag{2-9}$$

Derivacije od D aproksimiraju se korištenjem razlika u susjednim točkama uzoraka. Dobiveni 3x3 linearni sustav moguće je riješiti s minimalnim troškovima. Ako je odstupanje $\hat{x}$ veće od 0,5 u bilo kojoj dimenziji, to znači da se ekstrem nalazi bliže nekoj drugoj točki uzorka. U tom slučaju, točka uzorka se mijenja te se interpolacija vrši na novoj točki. Konačno odstupanje $\hat{x}$ dodaje se lokaciji točke uzorka kako bi se dobila interpolirana procjena lokacije ekstrema. Vrijednost funkcije u točki ekstrema, $D(\hat{x})$, korisna je za odbacivanje nestabilnih ekstrema sa slabim kontrastom. To se postiže uvrštavanjem (2-9) u (2-8) [1]:

[1] Operator transponiranja matrice

$$D(\hat{x}) = D + \frac{1}{2}\frac{\partial D^T}{\partial x}\hat{x}. \tag{2-10}$$

2.1.3. Određivanje orijentacije

Dodjeljivanje konstantne orijentacije svakoj ključnoj točki na osnovu njene okoline omogućava predstavljanje pripadajućeg deskriptora relativno o orijentaciji ključne točke, čime se postiže otpornost na rotaciju slike. Ovaj pristup je u suprotnosti s orijentacijski neovisnim deskriptorima koje su razvili Schmid i Mohr [7], kod kojih se svako svojstvo slike zasniva na rotacijski neovisnoj mjeri. Nedostatak tog pristupa je ograničavanje deskriptora te neiskorištavanje podataka slike zato što ne zahtjeva da se sve mjere temelje na dosljednoj rotaciji.

Eksperimentiranje s više različitih metoda dodjeljivanja lokalne orijentacije pokazalo je da najstabilnije rezultate postiže sljedeći pristup. Skala ključne točke koristi se za odabir zamućene slike, L, s najbližom skalom tako da se sve operacije izvršavaju u području neovisnom o skali. Za svaki uzorak slike, $L(x, y)$, na odgovarajućoj skali, iznos gradijenta, $m(x, y)$, i orijentacije, $\theta(x, y)$, računa se korištenjem razlika elementa slike [1]:

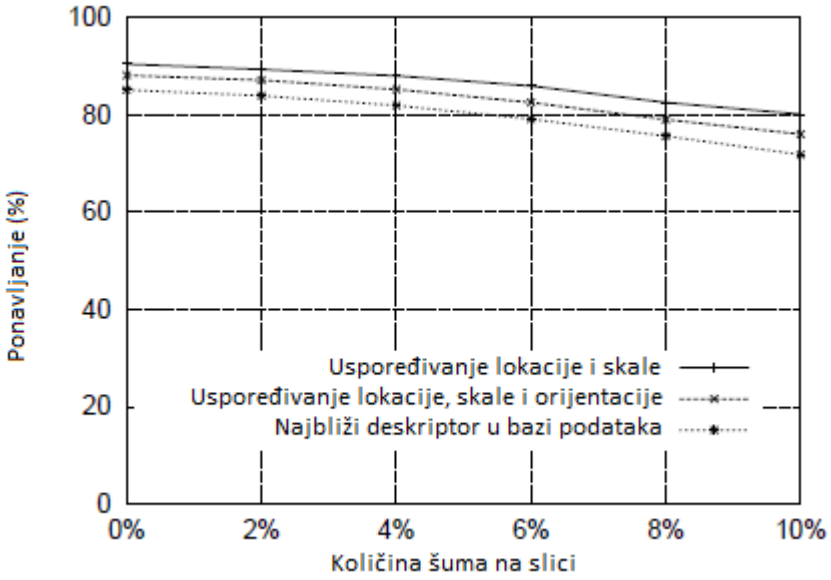
$$m(x,y) = \sqrt{\big(L(x+1,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2} \tag{2-11}$$

$$\theta(x,y) = tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right). \tag{2-12}$$

Orijentacijski histogram oblikovan je iz orijentacija gradijenata točaka uzoraka unutar područja oko ključne točke. Orijentacijski histogram sastoji se od 36 odjeljaka koji pokrivaju svih 360 stupnjeva orijentacije. Svaki uzorak dodan histogramu odmjerava se prema iznosu gradijenta i Gaussom ponderiranom kružnom okviru, gdje je iznos σ veći od skale ključne točke 1,5 puta [1].

Vrhovi u orijentacijskom histogramu odgovaraju dominantnim smjerovima lokalnih gradijenata. Pronalazi se najviši vrh u histogramu i svi ostali vrhovi koji se nalaze unutar 80% najvišeg vrha te se pomoću njih generira ključna točka s tom orijentacijom. Stoga, za lokacije s više vrhova sličnog iznosa generira se više ključnih točaka na istoj lokaciji i skali, ali s različitim orijentacijama. Otprilike 15% točaka bude dodijeljeno više orijentacija, ali one imaju značajan utjecaj na stabilnost poklapanja. Posljednje, određuje se parabola iz tri vrijednosti histograma koje su najbliže svakom vrhu kako bi se interpolirao položaj vrha za bolju preciznost [1].

Slika 2.3. prikazuje eksperimentalno dobivene stabilnosti lokacija, skale i orijentacija mjerenih za različite iznose šuma na slici. Slike u eksperimentima su nasumično skalirane i rotirane. Gornja linija prikazuje stabilnost dodijeljenih lokacija i skala ključnih točaka. Linija ispod prikazuje rezultate poklapanja kada je također zahtijevano da orijentacija bude unutar 15 stupnjeva. Kao što je vidljivo iz razmaka između gornjih dviju linija, dodjeljivanje orijentacije ostaje 95% precizno čak i nakon dodavanja $\pm$10% šuma elemenata slike (ekvivalentno kameri s preciznošću manjom od tri bita). Izmjereno odstupanje orijentacije za točna poklapanja je oko 2,5 stupnjeva, a za 10% šuma je oko 3,9 stupnjeva. Treća linija na slici 2.3. prikazuje konačnu preciznost točnog poklapanja deskriptora ključne točke s bazom od 40 000 ključnih točka. Graf pokazuje da su SIFT ključne točke otporne i na značajan iznos šuma te da je najveći uzrok pogrešaka izazvan početnom pogrešnom detekcijom lokacije i skale [1].



Slika 2.3. *Pouzdanost SIFT detekcije* [1].

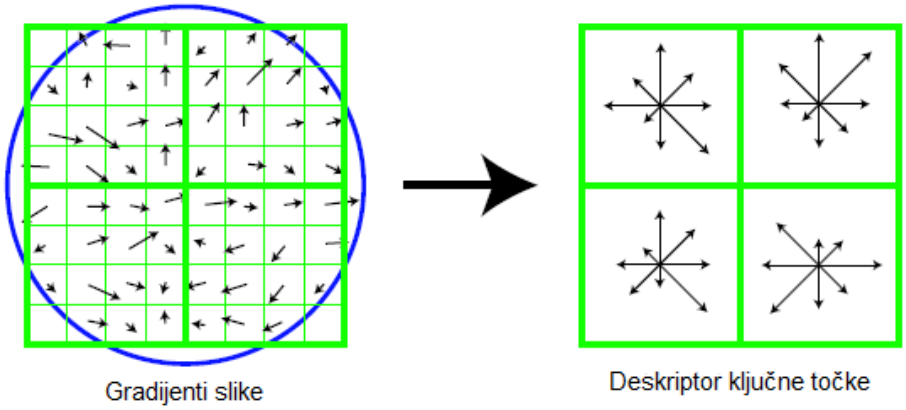
2.1.4. Oblikovanje deskriptora

Prethodne operacije odredile su lokaciju, skalu i orijentaciju svake ključne točke, što omogućava oblikovanje kvalitetnih deskriptora. Dobiveni parametri predstavljaju lokalni 2D koordinatni sustav u kojem opisuju lokalno područje slike te time ostvaruju neovisnost ključnih točaka o tim parametrima. Deskriptori se oblikuju za svako lokalno područje slike koje je vrlo karakteristično, a istovremeno otporno na distorzije kao što je promjena svjetline ili 3D gledišta.

Prva mogućnost je uzorkovanje lokalnih intenziteta slike oko ključne točke na zadanoj skali te usklađivanje korištenjem normalizirane mjere korelacije. Međutim, jednostavna korelacija dijelova slike je iznimno osjetljiva na promjene koje uzrokuju pogrešnu registraciju uzoraka, kao što su promjena 3D gledišta ili fizičke deformacije. Bolji pristup su demonstrirali

Edelman, Intrator i Poggio [8]. Njihova metoda temelji se na modelu biološkog vida, točnije neurona u primarnom vidnom korteksu. Ti neuroni reagiraju na gradijent određene orijentacije i prostorne frekvencije, ali lokacija gradijenta na mrežnici može se pomicati preko malog receptivnog polja umjesto da bude precizno lokalizirana. Pretpostavili su da je funkcija neurona omogućiti poklapanje i prepoznavanje 3D objekata iz više različitih gledišta. Stoga su izvršili detaljne eksperimente koristeći 3D računalne modele objekata i životinja koji su pokazali da je uspoređivanje gradijenta uz dopušten pomak u položaju značajno poboljšalo rezultate klasifikacije kod 3D rotacije. Na primjer, preciznost uspoređivanja za 3D objekte rotirane 20 stupnjeva se povećalo s 35% za korelaciju gradijenata na 94% korištenjem modela kompleksnih ćelija. Njihova ideja je inspirirala Lowea da razvije svoju implementaciju koja dopušta prostorni pomak korištenjem drugačijeg mehanizma računanja [1].

Slika 2.4. predstavlja oblikovanje deskriptora pomoću gradijenata slike. Prvo se uzorkuju iznosi gradijenata i orijentacije oko lokacije ključne točke korištenjem skale ključne točke za odabir razine Gaussovog zamućenja slike. Da bi se postigla neovisnost o orijentaciji, koordinate deskriptora i orijentacije gradijenata rotirane su u odnosu na orijentaciju ključne točke. Za učinkovitost, gradijenti su unaprijed izračunati za sve razine piramide. Gradijenti su prikazani pomoću malih strelica za svaku lokaciju uzorka na lijevoj strani slike 2.4. [1].



Slika 2.4. *Određivanje deskriptora iz gradijenata slike* [1].

Gaussova težinska funkcija čiji je σ jednak polovici širine okvira deskriptora koristi se za određivanje težine iznosa svake točke uzorka. To je ilustrirano pomoću plave kružnice na lijevoj strani slike 2.4. u kojoj težina uzorka opada s povećanjem udaljenosti od središta. Svrha ovog Gaussovog okvira je izbjegavanje naglih promjena u deskriptoru uzrokovanih laganim pomacima okvira te smanjivanje utjecaja gradijenata koji se nalaze daleko od središta deskriptora jer su oni najosjetljiviji na pogrešnu detekciju [1].

Deskriptor ključne točke prikazan je na desnoj strani slike 2.4. Stvaranjem orijentacijskih histograma na određenom području uzoraka (npr. 4x4 elementa slike) omogućavaju se značajni pomaci u položajima gradijenata. Slika prikazuje osam strelica za svaki orijentacijski histogram gdje duljina svake strelice odgovara iznosu pripadnog histograma. Gradijent prikazan na lijevoj strani slike može se pomaknuti do 4 mjesta, a da pri tome i dalje pridonosi istom histogramu prikazanom na desnoj strani slike. Na taj način postignuta je otpornost na veće lokalne pomake. Ključno je izbjeći utjecaj koji imaju rubovi gdje se deskriptor naglo mijenja kako uzorci prelaze iz jednog histograma u drugi ili iz jedne orijentacije u drugu. Stoga se koristi trilinearna interpolacija za distribuciju vrijednosti svakog gradijenta u susjedne histograme. Drugim riječima, svaki uzorak histograma se množi s faktorom $1 - d$ za svaku dimenziju, gdje je d udaljenost uzorka od središnje vrijednosti histograma [1].

Deskriptor je formiran iz vektora koji sadrži uzorke svih orijentacijskih histograma čije vrijednosti odgovaraju duljinama strelica na desnoj strani slike 2.4. Slika prikazuje 2x2 matricu orijentacijskih histograma, no testiranja su pokazala da najbolje rezultate postiže 4x4 matrica histograma s osam orijentacija. Stoga deskriptor ima $4 \cdot 4 \cdot 8 = 128$ elemenata koji opisuju ključnu točku [1].

Konačno, deskriptor se modificira tako da se smanji utjecaj promjene osvjetljenja. Prvo se vrši normalizacija vektora po jedinici duljine. Promjena u kontrastu slike kod koje se svaka vrijednost elemenata slike množi s konstantom ujedno množi i gradijente s istom konstantom. Stoga je takva promjena kontrasta negirana normalizacijom vektora. Promjena u svjetlini gdje se vrijednosti svakog elementa slike dodaje konstanta neće imati utjecaj na vrijednosti gradijenata jer se oni dobivaju iz razlike elemenata slike. Stoga, deskriptor je potpuno otporan na promjene svjetline cijele slike. No, i dalje se mogu pojaviti nelinearne promjene osvjetljenja kao što ih izaziva saturacija kamere. Ovakve pojave mogu izazvati nepoželjne promjene u relativnim iznosima nekih gradijenata, ali rijetko utječu na orijentaciju gradijenata. Stoga, utjecaj značajnih iznosa gradijenata smanjuje se odbacivanjem svih gradijenata čiji je iznos veći od 0,2 te se vrši ponovna normalizacija po jedinici duljine. To znači da uspoređivanje iznosa gradijenata više nije toliko značajno te da veću važnost ima distribucija orijentacija. Prag 0,2 je eksperimentalno određen korištenjem slika koje sadrže različite iluminacije za isti 3D objekt [1].

2.2. ORB algoritam

ORB (engl. *Oriented FAST and Rotated BRIEF*) [9] je algoritam za određivanje ključnih točaka zasnovan na kombinaciji FAST (engl. *Features from Accelerated Segment Test*) [10] i BRIEF (engl. *Binary Robust Independent Elementary Features*) [11] algoritama. FAST

algoritam služi za detekciju ključnih točaka, a BRIEF algoritam za oblikovanje njihovih deskriptora. SIFT je bio prvi algoritam u svojoj klasi te je prilikom njegovog dizajniranja naglasak bio na kvaliteti rezultata, a manje na potrošnji resursa. Kao posljedica toga SIFT postiže dobre rezultate, ali značajno opterećuje uređaj koji ga pokreće te se ne može koristiti u stvarnom vremenu. Razvojem tehnologije, kako računalnih, tako i mobilnih uređaja, pojavila se potreba za novim algoritmom koji zahtjeva manje resursa i radi u stvarnom vremenu. ORB je dizajniran kao alternativno rješenje za SIFT koje zahtjeva manje resursa, a postiže približno jednake rezultate.

2.2.1. oFAST algoritam

FAST algoritam se često koristi zbog visoke učinkovitosti, a malog opterećenja resursa. No, za razliku od SIFT-a, dobivene ključne točke ne sadrže orijentacijsku komponentu. Algoritam je vrlo učinkovit u detekciji ključnih točaka, ali su manje kvalitetne nego točke dobivene SIFT-om. Pod kvalitetom se smatra otpornost na distorzije tj. vjerojatnost točne detekcije. Stoga, nakon detekcije ključnih točaka potrebno im je dodatno odrediti orijentaciju kako bi se povećala kvaliteta. Postupak dodavanja orijentacijske komponente FAST ključnim točkama zove se oFAST (engl. *Oriented FAST*) algoritam.

Prvi korak je detekcija FAST ključnih točaka. FAST koristi samo jedan parametar - prag intenziteta između središnjeg elementa slike i ostalih elemenata slike koji se nalaze unutar kružnice oko središnjeg elementa. ORB algoritam koristi FAST-9 (radijus kružnice je 9 elemenata slike) jer se eksperimentalno pokazalo da on postiže najbolje rezultate. Kako FAST, tako i ORB algoritam ima parametar N koji predstavlja maksimalan broj ključnih točaka koje algoritam treba proizvesti. FAST algoritam radi na principu smanjivanja praga odziva kako bi dobio dovoljan broj ključnih točaka. Osim nedostatka orijentacijske komponente, FAST algoritam ne proizvodi nikakvu mjeru položaja koja ukazuje na blizinu kutova. No, pokazalo se da dobivene ključne točke imaju jak odziv uzduž rubova. Stoga se primjenjuje Harissova mjera kutova [12] za sortiranje ključnih točaka. To znači da algoritam prvo proizvede N ključnih točaka, a zatim ih sortira po redu ovisno o njihovoj Harissonovoj mjeri. Ključne točke dobivene FAST algoritmom detektiraju se posebno na svakoj razini DOG piramide, dok se kod SIFT-a istovremeno detektira na više razina.

ORB algoritam koristi jednostavnu, ali učinkovitu metodu mjerenja orijentacije kutova. Metoda se temelji na mjerenju intenziteta težišta. Točnije, intenzitet kuta smatra se odstupanjem od intenziteta težišta ključne točke te se dobiveni vektor između ta dva intenziteta koristi za računanje orijentacije. Količinu gibanja unutar promatranog polja definirao je Rosin [13]:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y), \tag{2-13}$$

što omogućava definiranje težišta u obliku:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right). \tag{2-14}$$

Nakon što je definirano težište promatranog područja, a težište kuta je unaprijed poznato, moguće je povući vektor od kuta prema težištu. Dobiveni vektor predstavlja orijentaciju područja koja se može prikazati kao [9]:

$$\theta = atan2(m_{01}, m_{10}), \tag{2-15}$$

gdje je *atan2* funkcija *arctan* koja osim iznosa orijentacije također vraća i pripadajući kvadrant. Kako bi se unaprijedila rotacijska neovisnost koriste se samo količine gibanja čije se *x* i *y* vrijednosti nalaze unutar kružnice radijusa *r*. Najbolji rezultati se postižu kada se za *r* uzme iznos jednak veličini promatranog polja, tj. kada je duljina *r* jednaka polovici širine promatranog polja.

2.2.2. rBRIEF algoritam

Deskriptor dobiven korištenjem BRIEF algoritma je niz opisnih bitova dobiven iz seta binarnih testova intenziteta. Neka je promatrano zamućeno polje slike *p*. Binarni test τ definiran je kao [9]:

$$\tau(p;x,y) \coloneqq \begin{cases} 1 & : p(x) < p(y) \\ 0 & : p(x) \geq p(y) \end{cases} \tag{2-16}$$

gdje je *p*(*x*) intenzitet promatranog polja *p* u točki *x*. Ključna točka je definirana kao vektor od *n* binarnih testova [11]:

$$f_n(p) \coloneqq \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; x_i, y_i). \tag{2-17}$$

Postoji više tipova distribucije testova, no najbolje rezultate postiže Gaussova distribucija oko središta promatranog polja. Za duljinu vektora *n* najčešće se uzima 256, isto kao i kod SIFT-a.

BRIEF deskriptori nisu neovisni o rotaciji slike u ravnini. Postotak poklapanja značajno opada već nakon par stupnjeva rotacije. Jedno od mogućih rješenja je dodavanje setova rotacija i pomaka za svako polje prilikom računanja deskriptora, no to bi značajno povećalo trošak

računanja. Stoga je razvijena učinkovitija metoda upravljanja BRIEF deskriptora s obzirom na orijentaciju pripadajućih ključnih točaka. Za svaku ključnu točku sa setom od *n* binarnih testova na lokaciji (x_i, y_i), definira se 2x*n* matrica [9]:

$$S = \begin{pmatrix} x_1, \ldots, x_n \\ y_1, \ldots, y_n \end{pmatrix}. \tag{2-18}$$

Korištenjem orijentacije θ promatranog polja i pripadajuće rotacije matrice R_θ izrađuje se usmjerena verzija S_θ od S [9]:

$$S_\theta = R_\theta S \tag{2-19}$$

iz čega se dobije usmjereni BRIEF operator:

$$g_n(p, \theta) := f_n(p) | (x_i, y_i) \in S_\theta. \tag{2-20}$$

Kut se diskretizira po koracima od 12 stupnjeva te se izrađuje tablica prethodno izračunatih BRIEF uzoraka. Sve dok je orijentacija θ ključne točke konzistentna iz svih smjerova, koristi se ispravan set točaka S_θ za izradu pripadajućeg deskriptora. Na taj je način usmjeren BRIEF deskriptor te je postignuta neovisnost o rotaciji slike u ravnini. Stoga se ovaj postupak nadogradnje BRIEF deskriptora naziva rBRIEF (engl. *rotated BRIEF*) algoritam.

3. NOVI ALGORITAM ZA USPOREDBU DINAMIČNIH SLIKA

Dinamične slike se ne mogu uspoređivati po svojem izvornom sadržaju, zato što je sadržaj opisan elementima slike. Problem kod dinamičnih slika je taj što se sadržaj dinamički mijenja u vremenu te nije direktno vezan uz elemente slike. Iz tog je razloga za izradu algoritma u sklopu ovog rada odabran pristup pomoću ključnih točaka. Ključne točke opisuju sadržaj slike na način koji nije direktno vezan uz položaj elemenata slike. To omogućava uspoređivanje sadržaja po njegovom kontekstu, a ne položaju elemenata slike. Ključna točka ukazuje računalu na karakterističan dio slike otporan na smetnje. Pojavljuje se novi problem - kako iskoristiti prepoznatljive podatke za opis konteksta slike.

Metoda koja je korištena u izrađenom algoritmu temelji se na detekciji objekata. Pretpostavka je da su objekti prepoznatljivi, kako čovjeku, tako i računalu. Detekcija ključnih točaka temeljena je na principu rada ljudskog oka, kao što je objašnjeno u prethodnom poglavlju. Objekti imaju istaknute rubove i nagle prijelaze što znači da se na njima treba pojaviti više ključnih točaka. Stoga, ako se na određenom području slike pojavi zbijena grupa ključnih točaka, onda se na tom mjestu vjerojatno nalazi nekakav objekt. Drugim riječima, samostalna ključna točka nije značajna, ali grupa ključnih točaka predstavlja objekt.

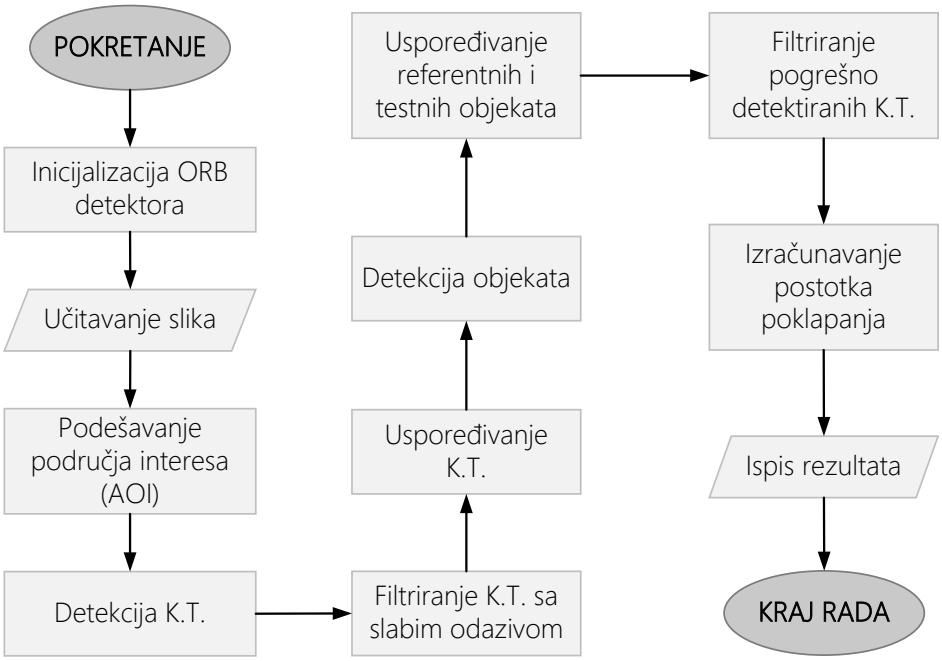
Objekti predstavljaju kontekst slike opisan elementima slike koji nisu direktno vezani za svoj položaj na slici, što ih čini idealnim za usporedbu dinamičnih slika. Algoritam vrši usporedbu dinamičnih slika provjerom jesu li se pojavili svi traženi objekti i jesu li se pojavili unutar očekivanog područja.

3.1. Način rada algoritma

Algoritam je pisan u programskom jeziku C++ korištenjem programa Visual Studio 2013. Podijeljen je u pet cjelina:

- inicijalizacija,
- obrada ključnih točaka,
- detekcija objekata,
- uspoređivanje slika,
- obrada rezultata.

Slika 3.1. prikazuje dijagram toka algoritma na kojem je vidljiv redoslijed svih aktivnosti potrebnih za usporedbu dinamičnih slika. Na slici K.T. označava ključne točke.



Slika 3.1. *Dijagram toka algoritma za usporedbu dinamičnih slika.*

Inicijalizacija obuhvaća postupke inicijalizacije ORB detektora, učitavanja slika te izdvajanje područja interesa. Inicijalizacija ORB detektora vrši se pomoću parametara optimiziranih za detekciju objekata na grafičkim sučeljima. Korišteni parametri imaju značajnu ulogu u detekciji ključnih točaka te ih je potrebno prilagoditi vrsti traženih objekata. U ovom algoritmu parametri su dobiveni empirijskom metodom, odnosno testiranjem na sekvencama preuzetim sa suvremenih grafičkih sučelja za televizijske uređaje. Izdvajanje područja interesa (engl. *Area of interest,* AOI) koristi se za izdvajanje samo određenog dijela slike. Na primjer, provjera pojave traženog objekta na određenom dijelu slike. Ako se objekt ne pojavi unutar područja interesa tada neće biti detektiran.

Obrada ključnih točaka obuhvaća njihovu detekciju i filtriranje s obzirom na odziv. Detekciju ključnih točaka vrši ORB detektor definiran u prethodnoj cjelini. Nakon detekcije izvršava se filtriranje ključnih točaka s obzirom na njihov odziv. Sve točke s odzivom slabijim od traženog se odbacuju.

Detekcija objekata se odnosi na proces dodjeljivanja svake ključne točke jednoj grupi (objektu). Nakon detekcije objekata svaku ključnu točku može se klasificirati po objektu kojem pripada.

Uspoređivanje slika predstavlja proces uspoređivanja objekata s referentne slike s objektima na testnoj slici. Nakon uspoređivanja filtriraju se svi pogrešni parovi ključnih točaka. Na taj se način osigurava da pojedini objekt s referentne slike može biti detektiran na samo jednom objektu testne slike.

Obrada rezultata vrši se izračunavanjem uspješnosti detekcije. Dvije dinamične slike su jednake ako dobiveni postotak poklapanja zadovoljava postavljeni prag. Postotak poklapanja računa se prema:

$$P[\%] = \frac{\text{broj K. T. referentne slike detektiranih na testnoj slici nakon primjene algoritma}}{\text{broj K. T. pronađenih na referentnoj slici}} \cdot 100\,. \qquad (3\text{-}1)$$

Prag koji mora biti zadovoljen ovisi o očekivanim smetnjama (deformacijama elemenata slike). Što je više smetnji, to je prag niži i obrnuto.

3.2. Odziv ključnih točaka

U drugom poglavlju objašnjen je rad ORB algoritma, tj. način na koji on generira ključne točke. Generira se dovoljan broj ključnih točaka koje se potom sortiraju po odzivu te se kao rezultat predaje N najboljih ključnih točaka, gdje je N maksimalan broj ključnih točaka. Odziv opisuje stabilnost ključne točke s obzirom na utjecaj smetnji. Stoga se stabilnost često koristi kao mjera kvalitete ključne točke. Mjera odziva temelji se na više faktora kao što su intenzitet ekstrema, blizina rubova i izraženost orijentacije. Algoritam izrađen u sklopu ovog rada koristi ORB algoritam za generiranje što većeg broja ključnih točaka.

Rezultat toga je velik broj nestabilnih ključnih točaka. Iz tog se razloga vrši filtriranje ključnih točaka s obzirom na njihov odziv. Prag po kojem se određuje hoće li ključna točka biti odbačena ovisi o karakteristikama slike. Drugim riječima, prag ovisi o kontekstu slike, odnosno o tome što se na slici nalazi. Na primjer, ako se na slici nalazi samo tekst, što znači puno naglih prijelaza i rubova, potrebno je koristiti veći prag jer će u prosjeku ključne točke imati jači odziv. Ako se pak radi o slici prirode s puno blagih tonova i prijelaza, potrebno je uzeti manji prag, zbog slabijeg odziva ključnih točaka. Naravno, potrebno je uzeti u obzir određenu minimalnu razinu praga kako ne bi došlo do propuštanja nestabilnih ključnih točaka.

Slika 3.2. prikazuje tri slučaja s različitim pragom odziva korištenim za filtriranje ključnih točaka. Na slici (a) je korišten previsok prag te je pronađeno samo 789 ključnih točaka, iako se na slici nalazi više objekata s istaknutom grafikom i tekstom. Za sliku (b) je korišten optimiziran prag koji se i koristi u konačnoj verziji algoritma te je detektirano 2792 ključnih točaka. Potrebno je uočiti da je većina ključnih točaka detektirana na objektima, što znači da će postotak poklapanja ispravno prikazivati samo postotak poklapanja objekata.



(a)



(b)



(c)

Slika 3.2. *Filtriranje ključnih točaka s obzirom na odziv, slučaj kada je prag odziva (a) previsok (b) optimalan (c) prenizak.*

Slika (c) prikazuje problem minimalne razine praga tj. detekcije neželjenih čestica kao ključnih točaka. Detektirana je 5301 ključna točka, no značajan broj tih ključnih točaka zapravo predstavlja smetnju. Pojava čestica na slici ne smije imati utjecaja na postotak poklapanja slika, što znači da ne smiju biti detektirane kao ključne točke. Iz tog razloga vrši se optimizacija algoritma ovisno o kontekstu slike. Prag odziva ima unaprijed postavljenu vrijednost čiji je iznos određen eksperimentiranjem na sekvencama s grafičkog sučelja za televizijske uređaje kao što je vidljivo na slici 3.2.

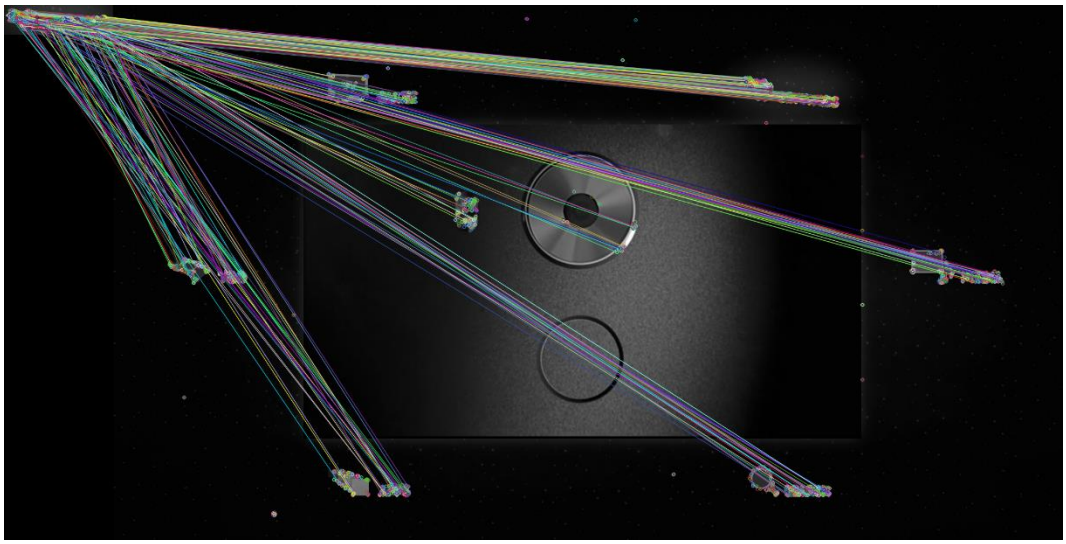
Algoritam je rađen tako da ima univerzalnu primjenu, što znači da se može koristiti na slikama svih konteksta i oblika objekata. S obzirom da postoji beskonačan broj mogućih oblika koji se mogu pojaviti na slici, algoritam mora imati visoku prilagodljivost. Iz tog je razloga korisniku do određene granice dopuštena manipulacija radom algoritma. Jedna od varijabli koje korisnik može mijenjati je prag odziva ključnih točaka. Ako se ustanovi da unaprijed definiran prag odziva nije dovoljno učinkovit za korištene slike, potrebno ga je povećavati ili smanjivati dok se ne procjeni da algoritam radi ispravno.

3.3. Pogrešna detekcija ključnih točaka

Glavni problem koji se pojavljuje tijekom usporedbe ključnih točaka je njihova pogrešna detekcija. Pogrešna detekcija se događa kada algoritam usporedi referentnu ključnu točku s testnom i dobije pozitivan rezultat, iako se radi o pogrešnoj točki (najčešće kod teksta kada se ista slova ili riječi pojave na više mjesta). Problem pogrešne detekcije ključnih točaka ima značajan utjecaj na rezultat usporedbe slika povećavanjem postotka poklapanja, što može rezultirati uspješnim poklapanjem dviju različitih slika. Slika 3.3. prikazuje slučaj detekcije objekta na slici pri kojem dolazi do velikog broja pogrešnih detekcija.



(a)



(b)

Slika 3.3. *Primjer pogrešne detekcije ključnih točaka, referentna (gornji lijevi ugao na slikama) i testna slika (a) prije detekcije (b) nakon detekcije.*

Slika (a) prikazuje referentni objekt i testnu sliku prije uspoređivanja ključnih točaka, a donja slika nakon uspoređivanja. Referentni objekt koji se traži je prikazan u gornjem lijevom

kutu slike (simbol zupčanika s tekstom „SETTINGS“). Na testnoj slici, objekt koji se traži nalazi se u gornjem desnom dijelu slike te je označen svijetlije od drugih objekata. Nakon uspoređivanja ključnih točaka uočljiv je veliki broj pogrešnih detekcija koje su se pojavile na svim ostalim objektima. Za ovaj primjer postignuto je 100% poklapanje, tj. pronađen je odgovarajući par za svaku od 474 referentnih ključnih točaka. Od svih poklapanja samo je 63,92% točno, a to znači da je čak 36,08% poklapanja zapravo pogrešna detekcija ključnih točaka, što nije prihvatljivo.

3.3.1. Detekcija objekata na slici

Učinkovito rješenje za pogrešnu detekciju ključnih točaka je detekcija objekata, gdje se svakoj ključnoj točki dodijeli objekt (grupa) kojoj pripada te se filtriraju sva poklapanja s pogrešnim objektima. Detekcija objekata i filtriranje odvojeni su procesi te je u ovom dijelu poglavlja objašnjena samo detekcija objekata.

Prvi korak u detekciji objekata je stvaranje matrice udaljenosti koja sadrži međusobne udaljenosti između svih ključnih točaka na slici. Ovaj proces je jednostavan, ali troši puno resursa jer je potrebno proći svaku ključnu točku i izračunati udaljenost u odnosu na sve druge točke za koje udaljenost još nije poznata. Jednadžba po kojoj se računa udaljenost je Euklidova jednadžba za udaljenost dviju točaka u dvodimenzionalnom prostoru:

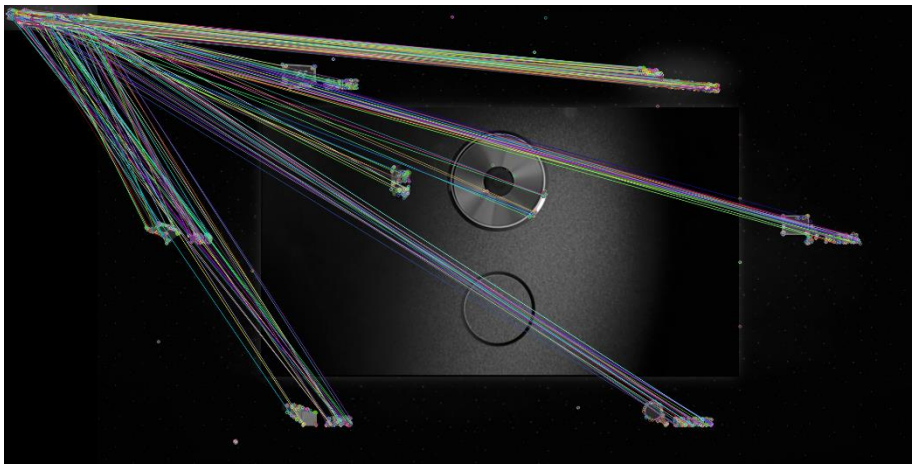
$$D(x, y) = \sqrt[2]{(y_1 - x_1)^2 + (y_2 - x_2)^2}, \tag{3-2}$$

gdje su x_1 i y_1 prostorne koordinate prve, a x_2 i y_2 druge točke.

Nakon izračuna matrice udaljenosti slijedi detekcija objekata. Glavni čimbenik kod detekcije objekata je maksimalna dopuštena udaljenost između dviju ključnih točaka. Algoritam za detekciju objekata radi na principu svrstavanja svih ključnih točaka u jedan objekt ako je ispunjen sljedeći uvjet. Svaka točka unutar objekta mora biti udaljena od neke druge točke unutar objekta za udaljenost manju od maksimalno dopuštene udaljenosti između dviju ključnih točaka. Algoritam pregledava ključne točke, a kada pronađe prvu slobodnu točku dodjeljuje joj oznaku grupe (objekta) i nastoji ju proširiti na sve okolne točke. Algoritam pretražuje vektor koji sadrži ključne točke sve dok ne naiđe na ključnu točku kojoj nije dodijeljena grupa (objekt). Pronađenoj točki dodjeljuje se nova oznaka grupe te se pregledava matrica udaljenosti za sve obližnje ključne točke koje se nalaze na udaljenosti manjoj od maksimalno dopuštene. Svakoj novoj pronađenoj točki dodjeljuje se ista oznaka grupe kao i izvornoj točki te se postupak ponavlja dok nisu pronađene sve točke koje zadovoljavaju uvjet objekta.

3.3.2. Filtriranje pogrešno detektiranih ključnih točaka

Nakon detekcije objekata svaka ključna točka ima svoju oznaku grupe, tj. objekta kojem pripada. Uspoređivanje ključnih točaka je prethodno odrađeno te su poznati svi parovi između referentnih i testnih ključnih točaka od kojih su mnogi pogrešni. Cilj filtriranja je odbaciti sve pogrešne parove. To je odrađeno definiranjem grupe referentnih objekata. Potrebno je obratiti pažnju da grupa referentnih objekata nije isto što i grupa objekata s referentne slike. Grupa referentnih objekata sadrži objekte s testne slike koji se najbolje poklapaju s objektima referentne slike. Preciznije rečeno, za svaki objekt s referentne slike definiran je samo jedan referentni objekt s testne slike za kojeg je postignuto najbolje poklapanje. Slika 3.4. prikazuje rezultate poklapanja prije i poslije filtriranja.



(a)



(b)

Slika 3.4. *Filtriranje pogrešno detektiranih ključnih točaka, rezultati uspoređivanja ključnih točaka (a) prije filtriranja (b) poslije filtriranja.*

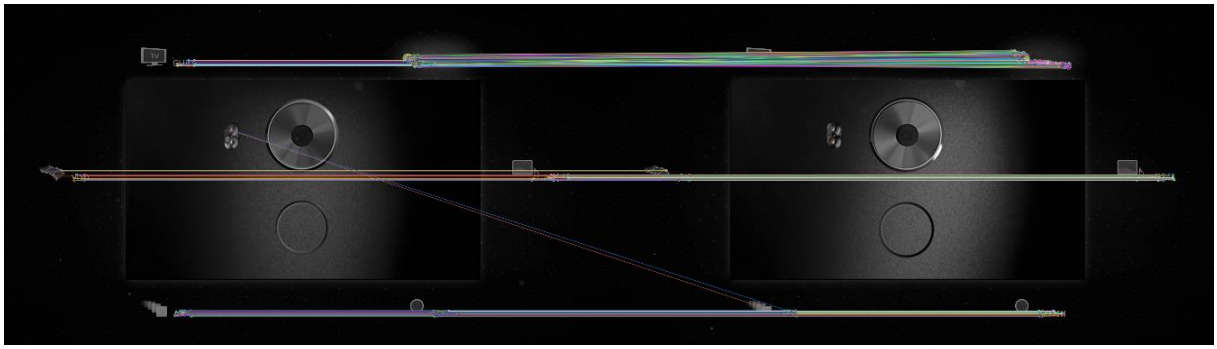
Referentni objekti se određuju s obzirom na broj detektiranih poklapanja. Svakom objektu s referentne slike promatraju se ključne točke, tj. grupe ključnih točaka s kojima su se

povezale. Vrši se prebrojavanje za svaku grupu (objekt) s testne slike tako da u konačnici bude poznato koliko ključnih točaka se povezalo s pojedinim objektom. Onaj objekt s testne slike na kojem je pojavilo najviše poklapanja definira se kao referentni objekt za taj objekt s referentne slike. Postupak se ponavlja sve dok se ne definiraju referentni objekti za svaki objekt referentne slike.

Nakon što je poznata grupa referentnih objekata, vrši se filtriranje pogrešnih detekcija ključnih točaka. Ponovno se pregledavaju sve ključne točke objekata s referentne slike te se provjerava je li pridružena ključna točka s testne slike član referentnog objekta za taj objekt referentne slike. Ako nije, onda se taj par smatra pogrešnim te se odbacuje.

3.4. Usporedba cijelih slika

Usporedba cijelih slika je osnovna funkcija algoritma. Cilj je usporediti dvije dinamične slike te odrediti jesu li one jednake. Algoritam koji ih uspoređuje radi na principu traženja objekata. Prvi korak je pretraživanje referentne slike i definiranje svih referentnih objekata, a zatim se pregledava testna slika te se određuju svi objekti testne slike. Nakon toga se pokreće algoritam za usporedbu objekata koji povezuje referentne objekte s njihovim parovima na testnoj slici, ukoliko oni postoje. Ako nisu pronađeni svi referentni objekti, algoritam odlučuje da se radi o različitim slikama. No, ako jesu pronađeni, onda se računa postotak poklapanja svih objekata te se određuje ukupni postotak poklapanja slike po kojem se odlučuje jesu li slike jednake.



Slika 3.5. *Detektirana poklapanja ključnih točaka cijele referentne i cijele testne slike.*

3.5. Traženje objekta

Traženje objekta je funkcija slična usporedbi cijelih slika, no u ovom slučaju referentna slika ima samo jedan objekt. Primjer traženja objekata je prikazan na slici 3.4. (b). Cilj je pregledati testnu sliku i odlučiti nalazi li se na njoj referentni objekt. Prvo se odrede sve ključne točke referentnog objekta, a zatim se određuju svi objekti na testnoj slici. Vrši se filtriranje objekata ovisno o broju pripadajućih ključnih točaka kako bi se odbacili svi preveliki i premali

objekti. Nakon toga vrši se uspoređivanje referentnog objekta sa svim potencijalnim kandidatima. Odluka o pronalasku traženog objekta donosi se pomoću najvišeg postotka poklapanja kandidata. Ovisno o zadanim parametrima za očekivanu translaciju, rotaciju i skaliranje, određuje se prag postotka poklapanja koji mora biti zadovoljen kako bi se sa sigurnošću moglo reći da je objekt pronađen.

4. VREDNOVANJE REZULTATA NOVOG ALGORITMA ZA USPOREĐIVANJE DINAMIČNIH SLIKA

Usporedba dinamičnih slika je jedinstven problem u području računalnog vida te je algoritam izrađen u sklopu ovog diplomskog rada jedno od prvih rješenja. Prema vlastitim saznanjima ne postoje druga globalno prihvaćena rješenja s kojima bi se novi algoritam mogao usporediti. Stoga je mjerena kvaliteta rada algoritma, odnosno otpornost na različite smetnje kao što su rotacija, skaliranje, pojava nepoželjnih čestica i translacija. Kada ne postoje smetnje, tada bi algoritam trebao imati *P* iz izraza (3-1) veći od 90%. Ako postoje smetnje onda se prag spušta na 70%, ovisno o intenzitetu smetnji. Osim mjerenja otpornosti algoritma na smetnje, također je izvršeno mjerenje učinkovitosti rada algoritma na stvarnim primjerima. Korištena su dva seta testova. Računalno generirani testovi za simulaciju smetnji te testovi preuzeti iz sekvence dinamičkog grafičkog sučelja.

4.1. Test tolerancije na različite smetnje

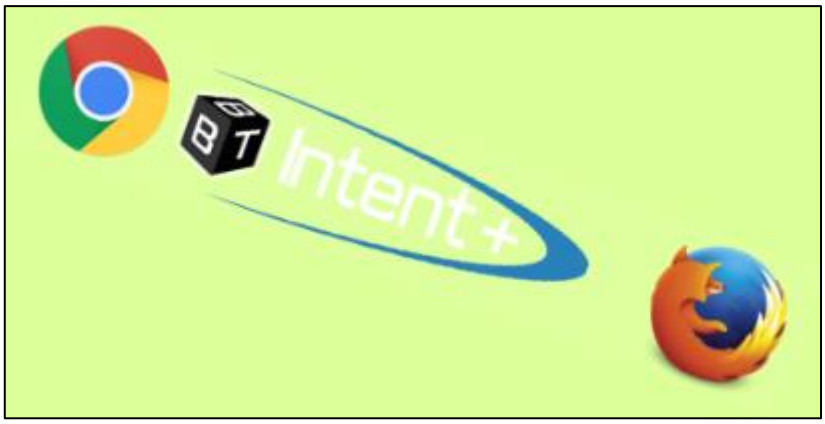
Prvi set testova je računalno generiran primjenom željenih transformacija (smetnji) na referentnu sliku prikazanu na slici 4.1. Svrha testova tolerancije je ispitati ponašanje algoritma u slučaju pojave određenih transformacija (smetnji) na objektima prikazanim na slici. Algoritam treba uspješno detektirati objekte prikazane na referentnoj slici nakon primjene transformacije.



Slika 4.1. *Referentna slika korištena u testovima.*

4.1.1. Tolerancija na rotaciju

Tolerancija na rotaciju podrazumijeva rad algoritma u slučajevima kada su slika ili traženi objekt rotirani oko svoje osi za ±20°. Ovo je najzahtjevnija vrsta smetnje, jer uzrokuje značajne promjene okoline ključnih točaka uzrokovane deformacijom objekta tijekom rotiranja. Primjer slike testa tolerancije na rotaciju prikazan je na slici 4.2.



Slika 4.2. *Primjer slike gdje je referentna slika zarotirana za 20°.*

S obzirom da postoji smetnja, prag koji se uzima za uspješnu detekciju je 70%. To znači da mora biti pronađeno više od 70% referentnih ključnih točaka. Rezultati testiranja su prikazani u tablici 4.1.

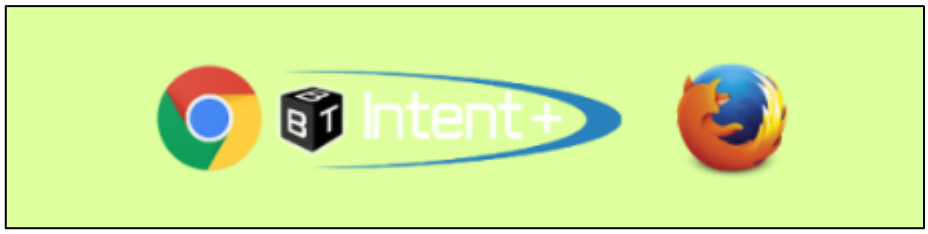
Tablica 4.1. *Rezultati testiranja tolerancije algoritma na rotaciju.*

Rotacija [°]	P [%]
-20	**94,22**
-10	**95,98**
-5	**94,05**
5	**94,05**
10	**93,10**
20	**93,06**

Analiza rezultata iz tablice 4.1. pokazuje da algoritam ima visoku toleranciju na rotaciju objekata. Postotak poklapanja u svakom testu prelazi 90% što zadovoljava postavljeni uvjet od minimalno 70%.

4.1.2. Tolerancija na skaliranje

Tolerancija na skaliranje podrazumijeva rad algoritma u slučajevima kada je traženi objekt skaliran s obzirom na svoje središte za ±20% svoje veličine. U nekim slučajevima skaliranje izaziva više deformacija na objektu nego rotacija. Ovisno o kvaliteti skaliranja mogu se pojaviti blokovi ili izgubiti pozadinski elementi slike koji značajno mijenjaju okolinu ključnih točaka. Primjer slike testa tolerancije na skaliranje je prikazan na slici 4.3.



Slika 4.3. *Primjer slike gdje je referentna slika smanjena za 20%.*

Prag koji se uzima za uspješnu detekciju jednak je 70%, kao i kod testa tolerancije na rotaciju. Rezultati testiranja su prikazani u tablici 4.2.

Tablica 4.2. *Rezultati testiranja tolerancije algoritma na skaliranje.*

Skaliranje [%]	P [%]
-20	**83,08**
-10	**89,26**
-5	**90,32**
5	**91,28**
10	**91,86**
20	**97,02**

Analiza rezultata iz tablice 4.2. pokazuje da algoritam ima visoku toleranciju na skaliranje objekata. Uočljivo je da su najbolji rezultati postignuti kada je objekt povećan za 20%, a najlošiji kada je smanjen za 20%. U slučaju povećavanja objekta elementi slike se dupliciraju te algoritam ima više informacija za definiranje ključnih točaka, što može povećati postotak poklapanja. Suprotno tome, kada se objekt smanjuje dolazi do gubitka podataka i informacija za definiranje ključnih točaka. Ako se objekt dovoljno smanji doći će i do gubitka ključnih točaka, što značajno smanjuje postotak poklapanja.

4.1.3. Tolerancija na čestice

Tolerancija na čestice podrazumijeva rad algoritma u slučajevima kada se na slici pojavljuju čestice koje nisu od nikakvog značaja, ali ometaju detekciju. Algoritam mora ispravno raditi kada je do 10% slike prekriveno česticama veličine 5x5 elemenata slike. Problem kod pojave čestica je njihova nepredvidljivost. Primjer slike testa tolerancije na čestice je prikazan na slici 4.4.



Slika 4.4. *Primjer slike gdje je 10% površine referentne slike prekriveno česticama.*

U nekim slučajevima se čestice pozicioniraju tako da ne preklapaju ni jednu ključnu točku, dok u drugim slučajevima uzrokuju dovoljne promjene okoline da se neke ključne točke uopće ni ne detektiraju. Ipak, pojava čestica ne uzrokuje toliko značajne deformacije objekata kao što ih uzrokuju rotacija i skaliranje te se za uspješnu detekciju uzima prag od 80%. Rezultati testiranja su prikazani u tablici 4.3.

Tablica 4.3. *Rezultati testiranja tolerancije algoritma na čestice.*

Čestice [%]	P [%]
2	**99,35**
4	**97,42**
6	**98,71**
8	**98,71**
10	**97,42**

Analiza rezultata iz tablice 4.3. pokazuje da algoritam ima visoku toleranciju na pojavu čestica. Postoji mogućnost da se čestice detektiraju kao ključne točke i uzrokuju lažne rezultate koji utječu na uspješnost poklapanja. Iz tog razloga algoritam, prije uspoređivanja, odbacuje sve ključne točke s niskim odzivom što osigurava da se same čestice ne detektiraju kao ključne točke.

4.1.4. Tolerancija na translaciju

Tolerancija na translaciju podrazumijeva uspješnu detekciju objekata ako su translatirani u bilo kojem smjeru do 20 elemenata slike. Ova vrsta smetnje najmanje utječe na postotak poklapanja jer ne uzrokuje nikakve deformacije objekta već samo promjenu pozadine iza objekta. Primjer slike testa na translaciju izgleda isto kao i referentna slika prikazana na slici 4.1., zato što je pozadina jednaka na cijelom testu te nisu vidljive nikakve promjene u slučaju pomaka. Stoga se za prag uspješne detekcije uzima 90%. Rezultati testiranja su prikazani u tablici 4.4.

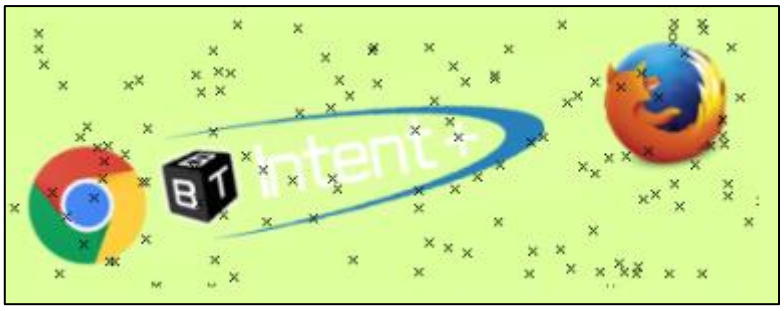
Tablica 4.4. *Rezultati testiranja tolerancije algoritma na translaciju.*

Horizontalna translacija	Vertikalna translacija	P [%]
-20	-20	**97,72**
-10	0	**96,71**
-20	20	**96,34**
20	20	**96,27**
0	-10	**99,36**
0	10	**97,67**

Analiza rezultata iz tablice 4.4. pokazuje da algoritam ima visoku toleranciju na pojavu translacije objekata. Poklapanje ni u jednom testu nije 100% zato što uvijek postoji nekoliko ključnih točaka koje su nestabilne, ali zadovoljavaju sve uvjete i budu detektirane. Prilikom filtriranja, algoritam odbacuje sva poklapanja na tim točkama zbog čega konačni rezultat uspješnosti poklapanja nije 100%.

4.1.5. Tolerancija na višestruke distorzije

Tolerancija na višestruke distorzije podrazumijeva ispravan rad algoritma kada se istovremeno pojavljuju sve prethodno navedene smetnje, što znači pojave rotacija, skaliranja, čestica i translacija. Uvjeti za pojedinu smetnju jednaki su uvjetima definiranim u prethodnom dijelu poglavlja. Primjer slike testa tolerancije na višestruke distorzije je prikazan na slici 4.5.



Slika 4.5. *Primjer slike gdje je referentna slika zarotirana za – 10°, smanjena za 10%, 5% površine prekriveno česticama i translatirana za 10 elemenata slike u lijevo i 10 elemenata slike prema dolje.*

S obzirom da postoji više smetnji od kojih neke uzrokuju značajne deformacije uzima se prag uspješne detekcije 70%. Rezultati testiranja su prikazani u tablici 4.5. Kod horizontalne translacije predznak "-" predstavlja translaciju u lijevo, a kod vertikalne translacije predstavlja translaciju prema gore.

Tablica 4.5. *Rezultati testiranja tolerancije algoritma na višestruke distorzije.*

Rotacija [°]	Skaliranje [%]	Čestice [%]	Horizontalna translacija [elemenata slike]	Vertikalna translacija [elemenata slike]	P [%]
-10	-10	5	-10	-10	**87,27**
10	10	6	0	0	**90,64**
-10	10	0	10	-10	**90,48**
10	0	2	20	20	**88,55**
0	-15	3	-20	20	**88,55**
20	20	0	0	0	**95,21**
-20	0	5	0	0	**95,38**
-10	0	0	-20	-10	**92,40**
0	-15	3	0	0	**85,14**
0	-12	0	20	20	**88,24**
0	0	3	-12	17	**92,53**

Analiza rezultata iz tablice 4.5. pokazuje da algoritam radi ispravno čak i kada se pojavljuju sve četiri vrste smetnji. Uočljivo je da postotak poklapanja nije konzistentan pojavama smetnji, tj. u nekim slučajevima se za više smetnji dobije bolji rezultat nego u slučaju s manje smetnji. Smetnje su nepredvidljive i njihov iznos nije proporcionalan s količinom deformacija na objektu. Na primjer, rotacija objekta za 10° može uzrokovati manje deformacija od rotacije objekta za 7°. Zato prag uspješne detekcije uzima u obzir samo je li se pojavila smetnja, a zanemaruje njezin iznos.

U ovom podpoglavlju 4.1. prikazani su rezultati otpornosti algoritma na razne smetnje za jedan tip sadržaja slike. Treba napomenuti da su isti testovi provedeni za pet različitih sadržaja i da su rezultati u svim slučajevima bili pozitivni, no zbog opsežnosti njihova prikaza neće svi oni biti prikazani.

4.2. Testiranje na stvarnim sekvencama

Institut RT-RK Osijek d.o.o. je isporučio drugi set testova kojeg su generirali iz sekvence vlastitog suvremenog grafičkog sučelja C-MORE UI/UX. Promotivni video grafičkog sučelja može se pogledati na [14]. Ovaj set testova predstavlja stvarne uvjete rada što znači da algoritam mora dobiti pozitivan rezultat na svakom testu kako bi se smatrao spremnim za tržište. Testovi su izvršeni na grafičkom sučelju u četiri različita slučaja:

1. Glavni izbornik,
2. Aplikacije,

3. Postavke,
4. TV vodič.

U svakom slučaju je potrebno detektirati je li se objekt pojavio na slici. Svaki slučaj sadrži različitu kombinaciju smetnji promjenjivih intenziteta.

4.2.1. Glavni izbornik

Glavni izbornik sadrži šest objekata od kojih je potrebno detektirati onoga koji je označen. Označeni objekt ima svjetliju pozadinu od drugih te se lagano rotira, povećava i smanjuje. Također, u pozadini slike se nalaze čestice koje se povremeno kreću po slici. Slika 4.6. prikazuje slučaj glavnog izbornika kada je odabran element „TV GUIDE“ u gornjem lijevom dijelu slike.



Slika 4.6. *Primjer glavnog izbornika s odabranim elementom „TV GUIDE“.*

S obzirom da se pojavljuju značajne smetnje, odabran je prag uspješne detekcije 70%. U ovakvom slučaju testa korišteno je područje interesa, što znači da je promatrana samo lokacija na kojoj se trenutno nalazi označeni objekt. Provedena su dva seta testova kod detekcije elementa „TV GUIDE“. Prvi set je dobiven nasumičnim odabirom slika iz sekvence u slučaju kada je odabran element „TV GUIDE“. To znači da će se pojaviti sve prethodno spomenute smetnje. Na slici 4.7. prikazano je 11 nasumično odabranih slika za ovaj set. Slika 4.6. korištena je kao referentna slika za ovaj set.


GUIDE
SETTINGS
VOD
MEDIA
APPS
SEARCH
LIVE
BELGIAN CAPITAL HIT BY DEADLY BLASTS: AIRPORT, METRO TARGETED
LOCAL MEDIA: BELGIAN INTELLIGENCE KNEW ABOUT POSSIBLE ATTACKS BUT NOT DETAILS

(a)


GUIDE
SETTINGS
VOD
MEDIA
APPS
SEARCH
LIVE
BELGIAN CAPITAL HIT BY DEADLY BLASTS: AIRPORT, METRO TARGETED
LOCAL MEDIA: BELGIAN INTELLIGENCE KNEW ABOUT POSSIBLE ATTACKS BUT NOT DETAILS

(b)


GUIDE
SETTINGS
VOD
MEDIA
APPS
SEARCH
LIVE
BELGIAN CAPITAL HIT BY DEADLY BLASTS: AIRPORT, METRO TARGETED
REPORTS: ISLAMIC STATE CLAIMS WAVE OF TERROR ATTACKS IN BRUSSELS ON TUESDAY

(c)


GUIDE
SETTINGS
VOD
MEDIA
APPS
SEARCH
LIVE
BELGIAN CAPITAL HIT BY DEADLY BLASTS: AIRPORT, METRO TARGETED
TWO EXPLOSIONS HIT BRUSSELS AIRPORT KILLING UP TO 17 PEOPLE. SCORES ALSO INJURED

(d)


GUIDE
SETTINGS
VOD
MEDIA
APPS
SEARCH
Politie
LIVE
ISIL TERRORISTS CLAIM VIOLENCE LEAVING UP TO 28 DEAD: REPORTS
BELGIUM-FRANCE BORDER REPORTEDLY CLOSED FOLLOWING ATTACKS ON BRUSSELS

(e)


GUIDE
SETTINGS
VOD
MEDIA
APPS
SEARCH
LIVE
ISIL TERRORISTS CLAIM VIOLENCE LEAVING UP TO 28 DEAD: REPORTS

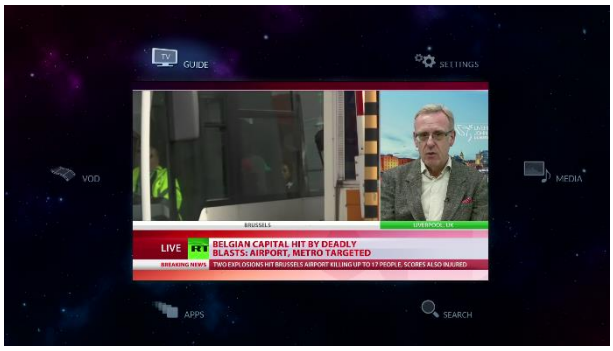
(f)


GUIDE
SETTINGS
VOD
MEDIA
APPS
SEARCH
LIVE
BELGIAN CAPITAL HIT BY DEADLY BLASTS: AIRPORT, METRO TARGETED
LOCAL MEDIA: BELGIAN INTELLIGENCE KNEW ABOUT POSSIBLE ATTACKS BUT NOT DETAILS

(g)


GUIDE
SETTINGS
VOD
MEDIA
APPS
SEARCH
LIVE
BELGIAN CAPITAL HIT BY DEADLY BLASTS: AIRPORT, METRO TARGETED
BELGIUM-FRANCE BORDER REPORTEDLY CLOSED FOLLOWING ATTACKS ON BRUSSELS

(h)



(i)



(j)



(k)

Slika 4.7. *Slike korištene za testiranje odabranog elementa glavnog izbornika, (a) Main_menu_selected_tv_guide_test1.bmp, (b) Main_menu_selected_tv_guide_test2.bmp, (c) Main_menu_selected_tv_guide_test3.bmp, (d) Main_menu_selected_tv_guide_test4.bmp, (e) Main_menu_selected_tv_guide_test5.bmp, (f) Main_menu_selected_tv_guide_test6.bmp, (g) Main_menu_selected_tv_guide_test7.bmp, (h) Main_menu_selected_tv_guide_test8.bmp, (i) Main_menu_selected_tv_guide_test9.bmp, (j) Main_menu_selected_tv_guide_test10.bmp, (k) Main_menu_selected_tv_guide_test11.bmp.*

Drugi set testova se odnosi na slučaj kada je odabran neki drugi element. U tom slučaju se na promatranom elementu ne događaju nikakve promjene, ali je sam element slabije izražen, što smanjuje odziv detektiranih ključnih točaka. Na slici 4.8. prikazano je 6 nasumično odabranih slika za ovaj set, kao i referentna slika.



(a)



(b)



(c)



(d)



(e)



(f)



(g)

Slika 4.8. *Slike korištene za testiranje neodabranog elementa glavnog izbornika, (a) Main_menu_not_selected_tv_guide_1.bmp, (b) Main_menu_not_selected_tv_guide_2.bmp, (c) Main_menu_not_selected_tv_guide_3.bmp, (d) Main_menu_not_selected_tv_guide_4.bmp, (e) Main_menu_not_selected_tv_guide_5.bmp, (f) Main_menu_not_selected_tv_guide_6.bmp, (g) Referent_picture_not_selected_tv_guide.bmp.*

Tablica 4.6. prikazuje rezultate za prvi set testova.

Tablica 4.6. *Rezultati testiranja kada je odabran element „TV GUIDE“.*

Referentna slika	Testna slika	P [%]
Referent_picture_tv_guide.bmp	Main_menu_selected_tv_guide_test1.bmp	**79,58**
	Main_menu_selected_tv_guide_test2.bmp	**79,58**
	Main_menu_selected_tv_guide_test3.bmp	**79,17**
	Main_menu_selected_tv_guide_test4.bmp	**88,33**
	Main_menu_selected_tv_guide_test5.bmp	**85,83**
	Main_menu_selected_tv_guide_test6.bmp	**78,75**
	Main_menu_selected_tv_guide_test7.bmp	**80,42**
	Main_menu_selected_tv_guide_test8.bmp	**82,08**
	Main_menu_selected_tv_guide_test9.bmp	**87,50**
	Main_menu_selected_tv_guide_test10.bmp	**83,33**
	Main_menu_selected_tv_guide_test11.bmp	**82,50**

Analiza rezultata iz tablice 4.6. pokazuje da je algoritam svaki puta uspješno detektirao traženi objekt. Dobiveni postotci poklapanja upućuju na pojavu značajnih smetnji kao što je i očekivano. Tablica 4.7. prikazuje rezultate drugog seta testova kada traženi element nije odabran.

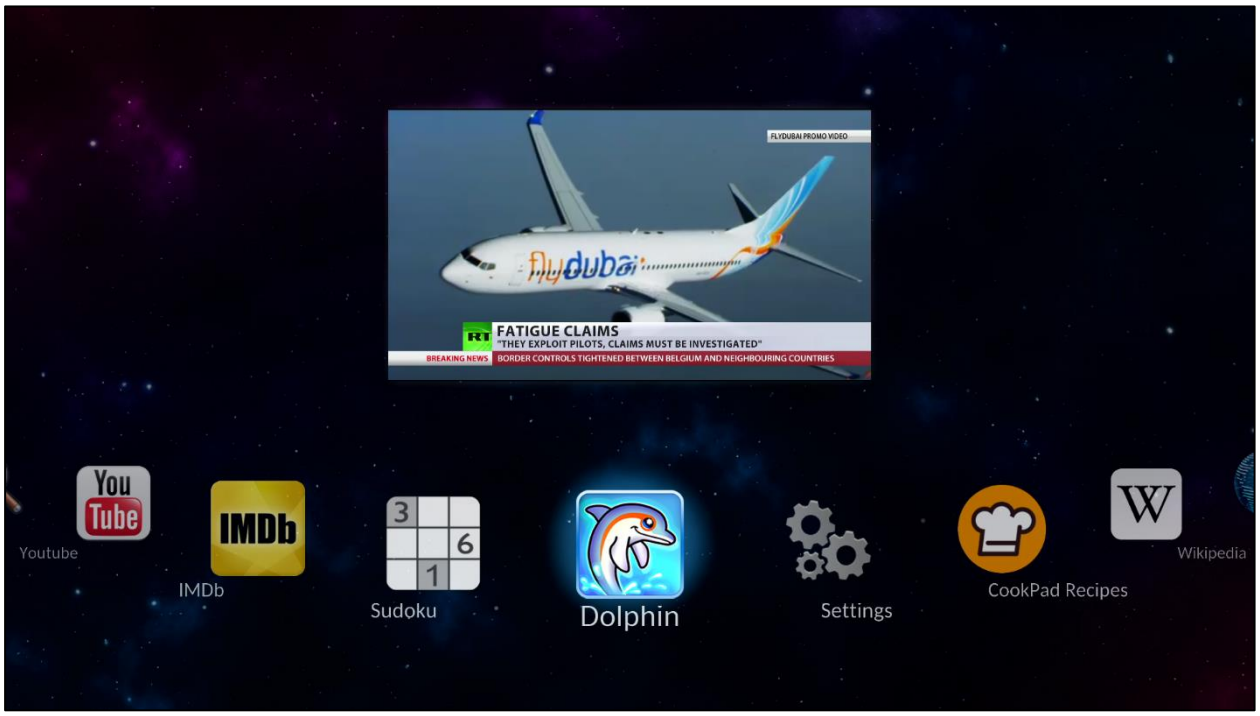
Tablica 4.7. *Rezultati testiranja kada traženi element „TV GUIDE“ nije odabran.*

Referentna slika	Testna slika	P [%]
Referent_picture_not_selected _tv_guide.bmp	Main_menu_not_selected_tv_guide_1.bmp	**97,06**
	Main_menu_not_selected_tv_guide_2.bmp	**94,12**
	Main_menu_not_selected_tv_guide_3.bmp	**94,12**
	Main_menu_not_selected_tv_guide_4.bmp	**100,00**
	Main_menu_not_selected_tv_guide_5.bmp	**94,12**
	Main_menu_not_selected_tv_guide_6.bmp	**97,06**

Rezultati drugog seta testova su značajno bolji od prvog seta. To znači da smetnje imaju značajniji utjecaj na postotak poklapanja od intenziteta, tj. istaknutosti objekta. Potrebno je obratiti pozornost da prilikom slabe istaknutosti objekta postoji mogućnost odbacivanja detektiranih ključnih točaka zbog nedovoljnog odziva. Zato je potrebno pažljivo odabrati minimalni prag odziva.

4.2.2. Aplikacije

Odabirom elementa „APPS“ na glavnom izborniku otvara se aplikacijski izbornik u kojem je izvršena druga cjelina testova. Ovdje je potrebno testirati je li se na zaslonu pojavila tražena aplikacija. Za razliku od glavnog izbornika, ovdje odabrana aplikacija nije animirana te nema značajnih smetnji poput rotacije i skaliranja. No, iza aplikacije se nalazi izvor svijetlosti čiji intenzitet varira. Primjer je prikazan na slici 4.9.

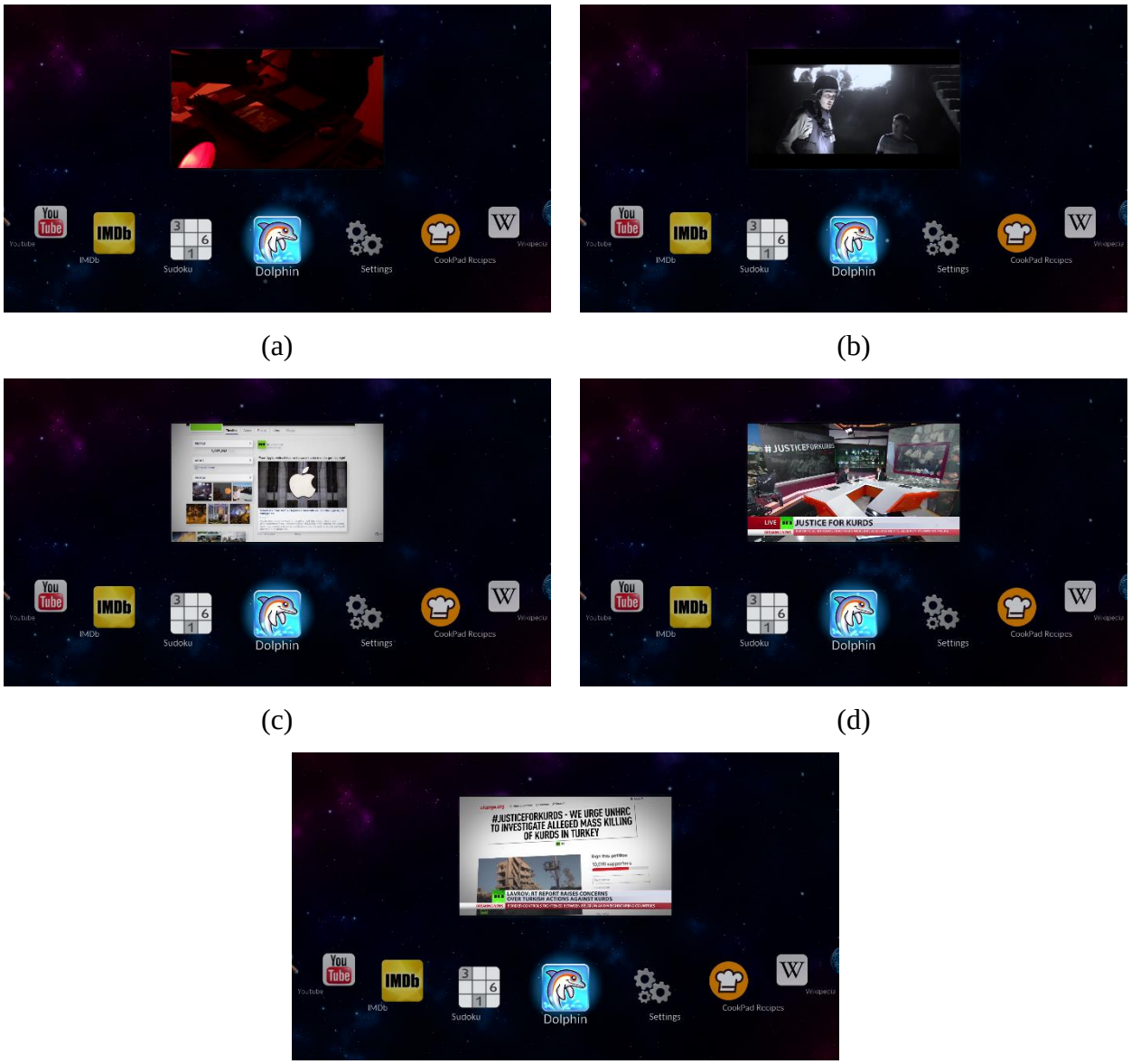


Slika 4.9. *Prikaz tražene aplikacije „Dolphin“.*

Osim promjene pozadinskog osvjetljenja također se pojavljuju i čestice, kao i kod slučaja glavnog izbornika. Slika 4.9. korištena je kao referentna slika, a testne slike dane su na slici 4.10. Rezultati testiranja su prikazani u tablici 4.8.

Tablica 4.8. *Rezultati detekcije aplikacije „Dolphin“.*

Referentna slika	Testna slika	P [%]
Referent_picture_dolphin.bmp	Dolphin_test_1.bmp	**88,17**
	Dolphin_test_2.bmp	**89,22**
	Dolphin_test_3.bmp	**87,57**
	Dolphin_test_4.bmp	**97,45**
	Dolphin_test_5.bmp	**95,21**

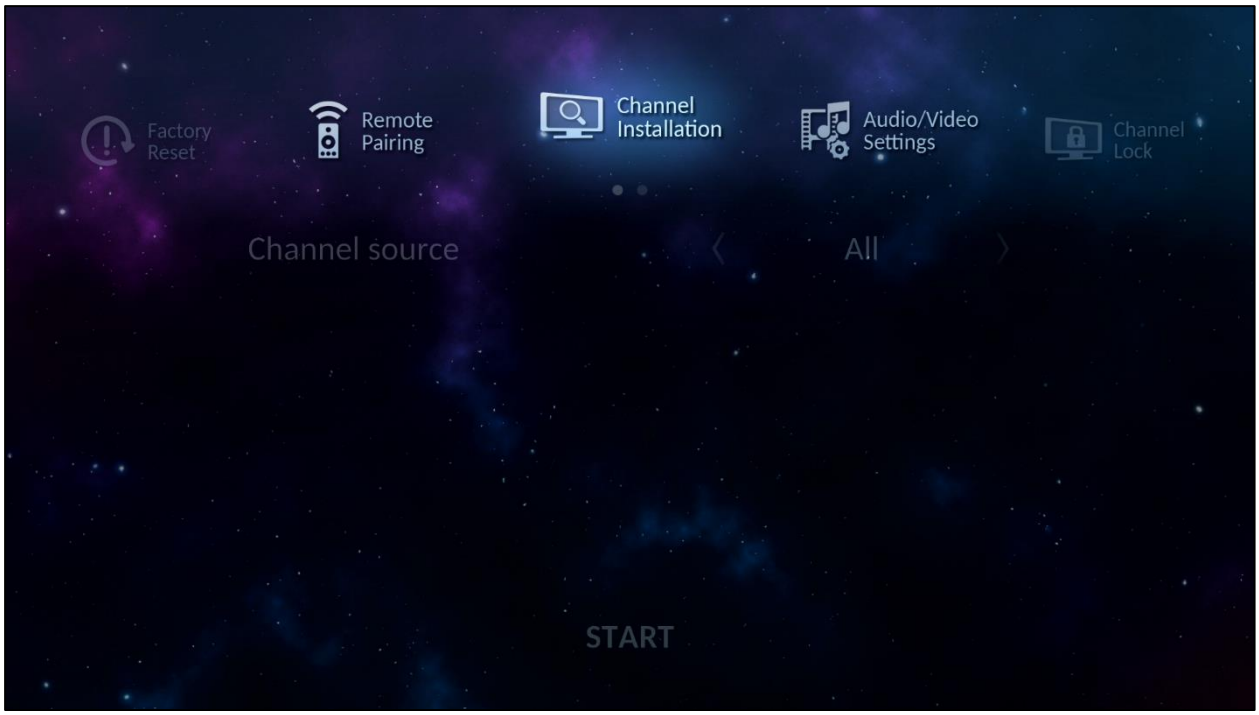


Slika 4.10. *Slike korištene za testiranje aplikacije „Dolphin“, (a) Dolphin_test_1.bmp, (b) Dolphin_test_2.bmp, (c) Dolphin_test_3.bmp, (d) Dolphin_test_4.bmp, (e) Dolphin_test_5.bmp.*

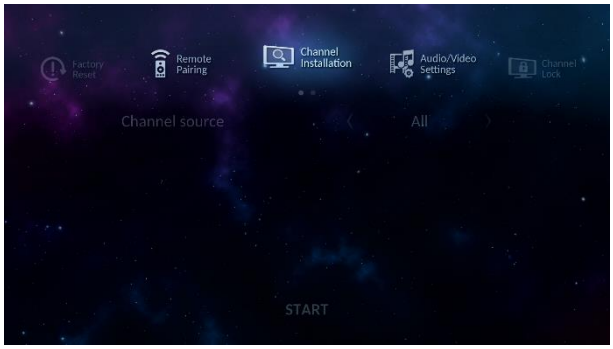
Zbog pojave čestica i promjena svjetline, korišten je prag uspješne detekcije 80%. Dobiveni rezultati pokazuju da je algoritam svaki put uspješno detektirao traženu aplikaciju.

4.2.3. Postavke

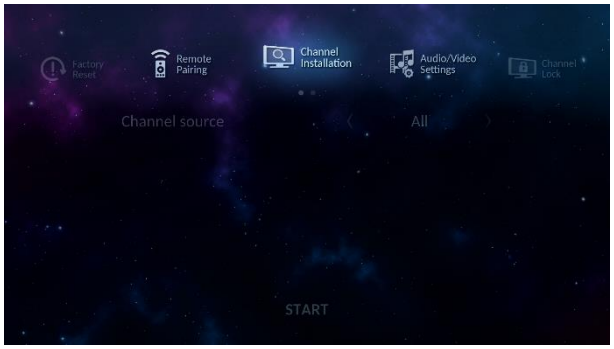
Treći set testova je izvršen na sekvenci dobivenoj iz glavnih postavki grafičkog sučelja. U ovom slučaju označeni traženi element je sličnog oblika kao i elementi glavnog izbornika, a pojavljuju se smetnje jednake kao i kod detekcije aplikacije. Primjer detekcije elementa „Channel Installation“ je prikazan na slici 4.11. koja je ujedno korištena kao referentna slika, a testne slike dane su na slici 4.12.



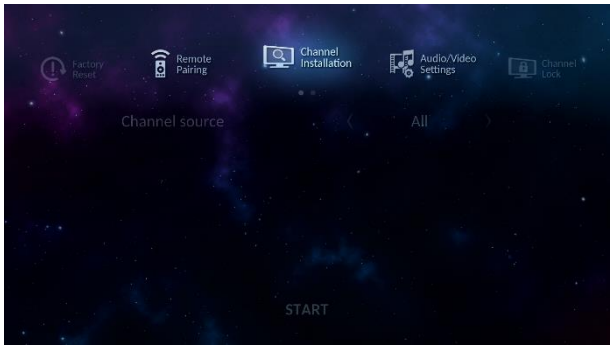
Slika 4.11. *Prikaz slike testa detekcije elementa „Channel Installation“.*



(a)



(b)



(c)

Slika 4.12. *Slike korištene za detekciju elementa „Channel Installation“, (a) Channel_installation_test_1.bmp, (b) Channel_installation_test_2.bmp, (c) Channel_installation_test_3.bmp.*

U ovom slučaju najveće smetnje izazivaju čestice koje mogu deformirati traženi objekt ili biti detektirane kao ključne točke. Rezultati testiranja su prikazani u tablici 4.9.

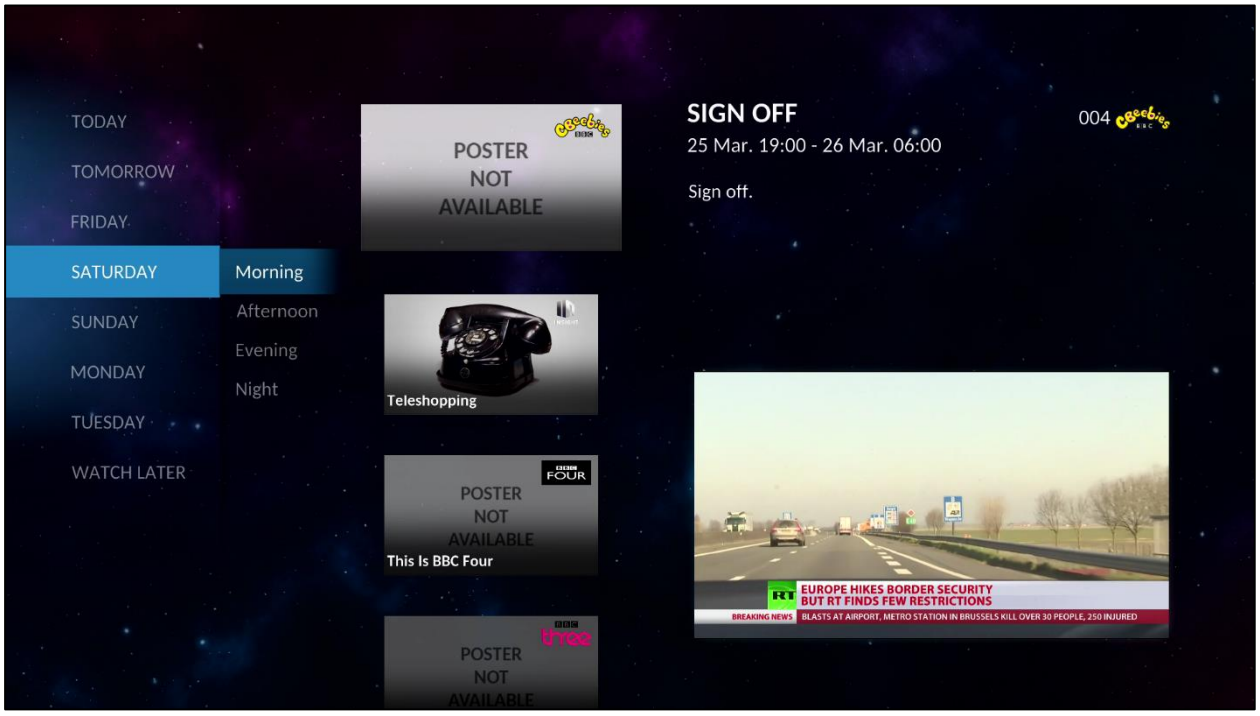
Tablica 4.9. *Rezultati detekcije elementa „Channel Installation“.*

Referentna slika	Testna slika	P [%]
Referent_picture_channel_installation.bmp	Channel_installation_test_1.bmp	**95,12**
	Channel_installation_test_2.bmp	**96,53**
	Channel_installation_test_3.bmp	**96,62**

S obzirom da se pojavljuju iste smetnje kao i u prethodnom testu, korišten je isti prag uspješne detekcije od 80%. Analiza rezultata pokazuje da je algoritam uspješno detektirao traženi element u sva tri slučaja.

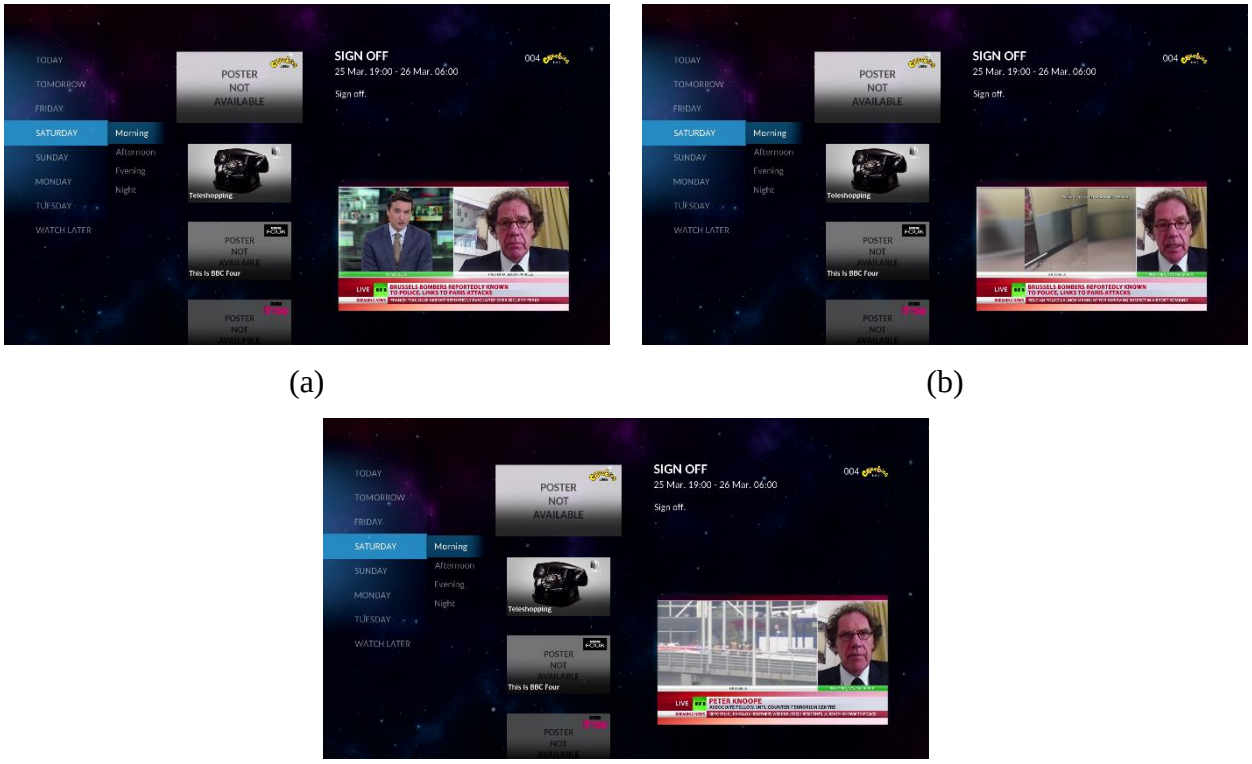
4.2.4. TV vodič

Posljednji set testova je izvršen na TV vodiču. Ovaj tip testa zahtjeva ispravnu detekciju teksta uz pojavu čestica. Tekst se nalazi na jednobojnoj pozadini bez ikakvih animacija što znači da se ne pojavljuju nepoželjne deformacije. Slika 4.13. prikazuje primjer slike testa na kojem je potrebno detektirati je li se pojavio tekst „SATURDAY“ na lijevom dijelu izbornika.



Slika 4.13. *Primjer slike testa za detekciju teksta „SATURDAY“.*

S obzirom da se ne očekuju značajne smetnje, korišten je prag uspješne detekcije 90%. Rezultati testiranja su prikazani u tablici 4.10., a testne slike na slici 4.14. Referentna slika prikazana je na slici 4.13.



(a) (b)

(c)

Slika 4.14. *Slike korištene za detekciju teksta „SATURDAY“, (a) Guide_Saturday_test_1.bmp, (b) Guide_Saturday_test_2.bmp, (c) Guide_Saturday_test_3.bmp.*

Tablica 4.10. *Rezultati detekcije teksta „SATURDAY“.*

Referentna slika	Testna slika	P [%]
Referent_picture_saturday.bmp	Guide_Saturday_test_1.bmp	**99,22**
	Guide_Saturday_test_2.bmp	**95,35**
	Guide_Saturday_test_3.bmp	**99,22**

Kao što je i očekivano, algoritam je uspješno detektirao traženi tekst s visokim postotkom poklapanja. Algoritam je prošao sve testove te se pokazao učinkovitim pri radu u stvarnim uvjetima rada.

5. ZAKLJUČAK

SIFT algoritam je prvo učinkovito rješenje za detekciju ključnih točaka na slici koje se još uvijek koristi kao temelj te tehnologije. No, razvoj računalne i mobilne tehnologije omogućio je rad programa u stvarnom vremenu što je izazvalo potrebu za novijim i boljim rješenjima. Stoga je, kao alternativno rješenje za SIFT, dizajniran ORB algoritam koji ne zauzima toliko resursa, a postiže približno jednake rezultate. Osim toga, ORB nije zaštićen licencom te se može slobodno koristiti u razvoju tehnologije, što je omogućilo izradu algoritma u sklopu ovog rada.

Problem koji se pojavljuje kod usporedbe dinamičnih slika korištenjem ORB algoritma je značajan broj pogrešnih detekcija ključnih točaka. Standardni algoritam za uspoređivanje ključnih točaka povezuje dvije točke za koje je detektirana najveća sličnost, iako se radi o pogrešnim točkama. To je problem koji se najčešće pojavljuje na slikama koje sadrže tekst, jer se ista slova pojavljuju na više mjesta, što znači da se pojavljuje više sličnih ključnih točaka. Algoritam izrađen u sklopu ovog rada rješava problem pogrešne detekcije pomoću detekcije objekata. Objekt je skupina ključnih točaka za koje vrijedi uvjet da svaka ključna točka unutar objekta mora biti udaljena od neke druge točke iz istog objekta za udaljenost manju od maksimalne dopuštene. Detekcija objekata osigurava povezivanje točaka iz određenog objekta s referentne slike s ključnim točkama samo jednog objekta s testne slike. Kao rezultat toga pojava pogrešnih detekcija ključnih točaka više nema značajan utjecaj na rezultat uspoređivanja te se algoritam može koristiti za preciznije usporedbe.

Analiza rezultata testiranja pokazuje da algoritam ima visoku toleranciju na smetnje, kao što su rotacija, skaliranje i translacija objekata te pojava sitnih čestica na slici. Algoritam radi ispravno čak i kada se sve spomenute smetnje pojavljuju istovremeno. Također, algoritam je moguće prilagoditi za detekciju specifičnih objekata kao što su ljudi ili automobili. U tom slučaju potrebno je definirati bazu referentnih ključnih točaka koje opisuju traženi objekt. Algoritam detektira objekte bilo kojih oblika što omogućuje široko područje primjene, a visoka tolerancija na smetnje osigurava visoku preciznost rada.

LITERATURA

[1] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, volume 60, str. 91 – 110, Nizozemska, 2004.

[2] J. J. Koenderink, The structure of images, Biological Cybernetics, 50: str. 363 – 396, 1984.

[3] T. Lindeberg, Scale-space theory: A basic tool for analysing structures at different scales, Journal of Applied Statistics, 21(2): str. 224 – 270, 1994.

[4] K. Mikolajczyk, Detection of local features invariant to affine transformations, Ph.D. thesis, Institut National Polytechnique de Grenoble, Francuska, 2002.

[5] D. G. Lowe, Object recognition from local scale-invariant features, International Conference on Coputer Vision, Corfu, Greece, str. 1150 – 1157, 1999.

[6] M. Brown, D. G. Lowe, Invariant features from interest point groups, British Machine Vision Conference, Cardiff, Wales, str. 656 – 665, 2002.

[7] C. Schmid, R. Mohr, Local grayvalue invariants for image retrieval, IEEE Trans. on Pattern Analysis and Machine Intelligence, 19(5): str. 530 – 534, 1997.

[8] S. Edelman, N. Intrator, T. Poggion, Complex cells and object recognition, unpublished manuscript, 1997.

[9] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: an efficient alternative to SIFT or SURF, International Conference on Computer Vision, str. 2564 – 2571, Barcelona, 2011.

[10] E. Rosten, T. Drummond, Machine learning for highspeed corner detection, European Conference on Computer Vision, volume 1, 2006.

[11] M. Calonder, V. Lepetit, C. Strecha, P. Fua, Brief: Binary robust independent elementary features, European Conference on Computer Vision, str. 778 – 792, 2010.

[12] C. Harris, M. Stephens, A combined corner and edge detector, Alvey Vision Conference, str. 147–151, 1988.

[13] P. L. Rosin, Measuring corner properties, Computer Vision and Image Understanding, 73(2): str. 291 – 307, 1999.

[14] C-MORE UI/UX: https://www.youtube.com/watch?v=Y6xv8wL0zDU (16.09.2016.)

.

SAŽETAK

Tema ovog diplomskog rada je problem usporedbe dinamičnih slika. Dinamične slike su slike sa stalnim kontekstom čiji je sadržaj promjenjiv u vremenu. Standardni način računalne usporedbe slika zasnovan na usporedbi elemenata slike uspoređuje sadržaj slike te se zbog toga ne može primijeniti za usporedbu dinamičnih slika. U sklopu ovog diplomskog rada izrađen je algoritam za detekciju objekata zasnovan na izdvajanju i uspoređivanju ključnih točaka. Algoritam koristi ORB algoritam za detekciju ključnih točaka. Glavni problem koji se pojavljuje je pogrešna detekcija ključnih točaka. Pogrešna detekcija se događa kada algoritam usporedi referentnu ključnu točku s testnom i dobije pozitivan rezultat, iako se radi o pogrešnoj točki (najčešće kod teksta kada se ista slova ili riječi pojave na više mjesta). Taj problem je riješen definiranjem objekata gdje se svakoj ključnoj točki dodjeljuje objekt (grupa) kojem pripada te se filtriraju sva poklapanja s pogrešnim objektima. Algoritmu je testirana otpornost na smetnje (rotaciju, skaliranje, pojavu čestica i translaciju) te učinkovitost rada na stvarnim sekvencama.

Ključne riječi: usporedba dinamičnih slika, ključne točke, ORB, dinamične slike

DYNAMIC PICTURE COMPARISON BASED ON EXTRACTION AND COMPARISON OF KEY POINTS

ABSTRACT

This Master's Thesis addresses the problem of dynamic image comparison. Dynamic images are images with static context and continuous content. Standard approach to image comparing based on pixels comparing only compares image content which makes it unusable for dynamic images. As part of this Master's Thesis, an object detection algorithm was designed based on the separation and comparison of key points. The algorithm uses ORB algorithm for key points detection. The main problem that appears is false key point detection. False key point detection happens when algorithm compares referent key point with test key point and gets a positive result although it's a wrong key point (mostly happens on images with text where same letters appear on multiple locations). That problem is solved by defining objects in a way that each key point is given an object (group) to which it belongs and then filtering all matches with wrong objects. Algorithm has been tested for distortion tolerance (rotation, scaling, particles and translation) and work efficiency on real actual sequences.

Key words: dynamic image comparison, key points, ORB, dynamic images

ŽIVOTOPIS

Matija Pul rođen je 23. listopada 1991. u Đakovu. Od 1998. do 2006. godine pohađa osnovnu školu Frana Krste Frankopana u Osijeku. Za to vrijeme sudjeluje u natjecanjima iz informatike u programskom jeziku LOGO te natjecanju iz tehničke kulture na Državnoj razini u disciplini robotike. Nakon osnovne škole, 2006. godine, upisuje Elektrotehničku i prometnu školu Osijek, smjer elektrotehnika. 2010. godine prima nagradu za odličan uspjeh i uzorno vladanje te doprinos razredu tijekom školovanja kroz četiri godine. Te iste godine ostvaruje pravo izravnog upisa na Elektrotehnički fakultet Osijek gdje odabire smjer komunikacije i informatika. Po završetku preddiplomskog studija 2014. upisuje diplomski studij, smjer komunikacije i informatika. U rujnu 2015. godine postaje stipendist Instituta RT-RK Osijek.

Potpis:

_____