**UNIVERZITET U BANJOJ LUCI**

**ELEKTROTEHNIČKI FAKULTET**

Srđan Jović

**IMPLEMENACIJA I OPTIMIZACIJA ALGORITMA ZA DETEKCIJU POMJERAJA OBJEKTA U VIDEU NA DSP PROCESORU**

Diplomski rad

**Banja Luka, oktobar 2019**

|  |  |
| --- | --- |
| **Tema:** | IMPLEMENTACIJA I OPTIMIZACIJA ALGORITMA ZA DETEKCIJU POMJERAJA OBJEKTA U VIDEU NA DSP PROCESORU |
|  | Ključne riječi:  Detekcija pomjeraja  Uparivanje blokova  DSP procesor |
| **Komisija:** | prof. dr Aleksej Avramović, predsjednik  prof. dr Vladimir Risojević, mentor  mr Vladan Stojnić, član |

**Kandidat:**

Srđan Jović

**UNIVERZITET U BANJOJ LUCI**

**ELEKTROTEHNIČKI FAKULTET**

**KATEDRA ZA RAČUNARSTVO I INFORMATIKU**

|  |  |
| --- | --- |
| **Predmet**: | MULTIMEDIJALNI SISTEMI |
| **Tema**: | IMPLEMENTACIJA I OPTIMIZACIJA ALGORITMA ZA DETEKCIJU POMJERAJA U VIDEU NA DSP PROCESORU |
| **Zadatak**: | Opisati arhitekturu TI DSP C66x procesora, kao i tehnike za optimizaciju te korištenje mogućnosti cl6x TI kompajlera u svrhu optimizacije koda. Opisati algoritam za detekciju pomjeraja objekta na videu korištenjem estimacije pokreta uparivanjem blokova u susjednim frejmovima. Realizovati, te prilagoditi algoritam za detekciju pomjeraja objekta korištenjem estimacije pokreta za TI DSP C66x procesor. Primijeniti taj algoritam na video koji dolazi sa kamere, te rezultate prikazati na HDMI displeju. Analizirati performanse inicijalnog i optimizovanog koda. |
| **Mentor**: | prof. dr Vladimir Risojević |
| **Kandidat**: | Srđan Jović |

**Banja Luka, oktobar 2019.**

**Sadržaj**

[1. UVOD 1](#_Toc21602140)

[2. RAČUNARSKI VID 3](#_Toc21602141)

[2.1 Istorija 4](#_Toc21602142)

[2.2 Primjena 5](#_Toc21602143)

[3. DETEKCIJA POMJERAJA OBJEKTA U VIDEU 6](#_Toc21602144)

[3.1 Razlika frejmova 7](#_Toc21602145)

[3.2 Estimacija pokreta uparivanjem blokova 9](#_Toc21602146)

[3.2.1 Iscrpna metoda pretrage 9](#_Toc21602147)

[3.2.2 Metoda pretrage od tri koraka 11](#_Toc21602148)

[3.2.3 Dijamanstka metoda pretrage 12](#_Toc21602149)

[4. UGRAĐENI SISTEMI 14](#_Toc21602150)

[5. DSP 15](#_Toc21602151)

[6. IMPLEMENTACIJA ALGORITMA DETEKCIJE POMJERAJA 16](#_Toc21602152)

[7. OPTIMIZACIJA ALGORITMA DETEKCIJE POMJERAJA 17](#_Toc21602153)

[8. TESTIRANJE I ANALIZA REZULTATA TESTIRANJA 18](#_Toc21602154)

[9. ZAKLJUČAK 19](#_Toc21602155)

[LITERATURA 20](#_Toc21602156)

# 1. UVOD

Još početkom sedamdesetih godina dvadesetog vijeka naučnici u polju računarske tehnike pokazivali su interesovanje za pravac koji danas nazivamo “Računarski vid” (*eng. Computer vision*). Računarski vid jeste naučna disciplina koja se bavi razvojem i istraživanjem metoda koje računar koristi da bi dobio određen nivo shvatanja digitalnih slika ili digitalnog videa. Iz perspektive inženjeringa, pravac istražuje načine koji će automatizovati procese koje ljudski sistem za vid može da obavlja. Računarski vid jeste veoma širok pojam, a samo neke od pod-disciplina kojima se računarski vid bavi jesu: rekonstrukcije scena, prepoznavanje objekata, detekcija i estimacija pokreta objekta u videu, kao i mnoge druge.

Pod-disciplina na koju se ovaj rad jednim dijelom fokusira jeste detekcija pomjeraja objekta u videu. Kada se priča o detekciji pomjeraja objekta misli se na proces detekcije promijene pozicije objekta u videu u odnosu na njegovo okruženje, kao i detekcije promjene položaja okruženja objekta u odnosu na stacionaran objekat. Primjena detekcije pomjeraja objekta na videu je raznolika, a samo neka od značajnijih mjesta gdje je detekcija zastupljena jesu: sigurnosni sistemi koji detekciju pokreta korsite da bi se detektovale određene aktivnosti krivičnog karaktera kao što su neautoritzovani pristupi prostoru koji se smatra zaštićenim, automotiv industrija koja detekciju pomjeraja objekata koristi da bi se iz vozila mogla detektovati ostala vozila koja učestvuju u saobraćaju, kao i detekcija kretanja pješaka koji se nalaze u neposrednoj blizini vozila. Metoda detekcije pomjeraja koja će biti obrađena u nastavku rada jeste metoda bazirana na estimaciji pokreta uparivanjem blokova iz susjednih frejmova.

U prethodnom paragrafu može se vidjeti da neki slučajevi upotrebe detekcije pomjeraja objekta na videu zahtijevaju da se proces detekcije izvršava na lokacijama na kojima nekada i nije moguće imati veliku procesorsku moć i gdje su dimenzije računarske platforme na kojoj se proces detekcije odvija često ograničene. Za primjer možemo uzeti detekciju pomjeraja objekata u okolini automobila gdje se sama platforma koja izvršava detekciju nalazi u automobilu. Zbog toga se u ovakvim situacijama koriste sistemi koji se nazivaju ugrađeni sistemi (*eng. Embedded systems*), a njihova primarna uloga jeste da obavljaju neku određenu funkciju, najčešće u sklopu nekog većeg mehaničkog ili električnog sistema. Nešto više o ugrađenim sistemima biće rečeno u glavi 4 nazvanom “Ugrađeni sistemi”. Ugrađeni sistem na kojem će se bazirati ovaj rad zasnovan je na TSM320C66x procesorskoj arhitekturi, koja je razvijena od strane kompanije pod nazivom Texas Instruments i koja je specifična po tome što se programi mogu izvršavati na procesoru za digitalnu obradu signala (*eng. DSP - Digital Signal Processor*) koji je optimizovan za operacije koje su česte kada se radi o obradi digitalnih signala.

Predmet ovog rada, kao što se može zaključiti iz samoga naziva rada, jeste implementacija algoritma za detekciju pomjeraja objekta u videu i optimizacija istog za platformu koja je bazirana na TMS320C66x procesorskoj arhitekturi. U drugom poglavlju rada dat je pregled pravca računarske tehnike koji se naziva računarski vid i kratak pregled pod-disciplina kojima se računarski vid bavi. Kao jedna od pod-disciplina kojom se računarski vid bavi jeste detekcija pomjeraja objekta u videu i ona je opisana u trećem po redu poglavlju, gdje se u prvi plan stavlja metoda detekcije zasnovana na estimaciji pokreta uparivanjem blokova između susjednih frejmova video zapisa. Navedeni metod je zansnovan na više različitih metoda pretrage tokom uparivanja blokova, a tri metoda koja su obrađena u ovoj glavi jesu: iscrpna metoda pretrage (*eng. Exhaustive Search*), metoda pretrage od tri koraka (*eng. Three Step Search*) i dijamantska metoda pretrage (*eng. Diamond Search*). Dodatno će biti riječi i o dodatne dvije metode, a to su: šestougaona pretraga (*eng. Hexagon Search*) i logoritamska pretraga (*eng. Logarithmic Search*). Implementacija će se bazirati na prve tri navedene metode, dok se druge dvije spominju samo idejno. Četvrta glava govori uopšteno o ugrađenim sistemima, kao i o platformi TDA2Px koja se koristi kao platforma za realizaciju rada ,a služi kao podloga za glavu koja joj slijedi, galvu pet, koja govori uopšteno o DSP procesoru, a nakon toga daje uvid u arhitekturu i karaktersistike TMS320C66x procesora. Detalji implementacije algoritma za detekciju pomjeraja na datoj hardverskoj arhitekturi dati su u glavi šest, koja daje informativan pregled arhitekture softerskog *framework*-a pod nazivom Processor SDK-Vision u kojem je realizovana implementacija praktičnog dijela. Glava sedam sadrži uvid u metode optimizacija koje su korištene pri implementaciji praktičnog dijela, a uključuju optimizacije na nivou algoritma i optimizacije na nivou arhitekture na kojoj je praktični dio razvijen. Testriranje i analiza rezultata dobijenih testiranjem dati su u glavi osam. U posljednoj glavi nalazi se zaključak ovog rada, gdje se ističu glavni rezultati rada, mogućnost primjene rada, kao i preporuke za dalji rad na obrađenom problemu.

# 2. RAČUNARSKI VID

Sam pojam detekcije pomjeraja objekta na videu ne bi se istraživao da naučnici u sedamdesetim godinama prošlog vijeka nisu pokazali određen nivo interesovanja u polju računarskog vida. Ljudska bića svakodnevno opažaju trodimenzionalnu strukturu svijeta oko sebe i to rade sa velikom lakoćom. Ljudi bez ikakvih problema mogu da zaključe kakvog je neki predmet obilika, koliko se ljudi nalazi na fotografiji, pa čak i kojega pola i koje starosti su ti ljudi sa fotografije.Perceptivni psiholozi proveli su decenije pokušavajući da shvate kako vizuelni sistem kod čovjeka funkcioniše i iako su uspjeli da shvate neke od osnonih principa, kompletan način funkcionisanja vizuelnog sistema čovjeka i danas ostaje misterija [1].

Naučnici u polju računarskog vida razvili su računarske metode za rekonstruisanje trodimenzionalnog oblika objekata sa slika na osnovu dovoljno velikog broja slika istog objekta iz različitih uglova posmatranja. Danas je moguće i pratiti kretanje objekata iza kojih se nalaze kompleksne pozadine. Moguće je čak i odrediti koje se osobe nalaze na fotografiji tako što se koriste kombinacije metoda za detekciju i prepoznavanje lica, odjeće i kose. Međutim, bez obzira na sva ova tehnološka dostignuća današnjice, i dalje je nezamslivo da računar analizira i interpretira fotografiju na nivou dvogodišnjeg djeteta [1].

Postavlja se pitanje zašto je računarski vid toliko komplkesan domen i težak za razumijevanje? Odgovor proizilazi iz činjenice da je vid „inverzan“ problem, što znači da je potrebno odrediti neke nepoznate varijable, a na raspolaganju su nam informacije koje nisu dovoljne da se te varijable odrede sa lakoćom. *Forward* modeli koji se korsite u polju računarskog vida potiču iz računarske grafike i fizike, polja koja modeluju kako se objekat kreće, kako se svjetlost prelama, odbija od reflektivne površine i na kraju projektuje na ravne ili zakrivljene površine. Sa druge strane u računarskom vidu isto tako pokušava se raditi obrnuta procedura, gdje se na osnovu jedne ili više fotografija dobijaju informacije kao što su oblik, osvjetljenost, distribucija boja na objektima, kao i način i pravac kretanja objekata sa fotografije ili videa [1].

|  |  |
| --- | --- |
|  |  |
| Slika 2.1 – Algoritam za praćenje kretanja osobe sa kompleksnom pozadinom | Slika 2.2 – Kombinacija algoritama za detekciju lica i odjeće |

## 2.1 Istorija

Ideja da bi računarski vid trebao biti jednostavan potiče još iz ranih dana vještačke inteligencije, kada se vjerovalo da je kognitivni dio inteligencije dosta kompleksniji za shvatanje od perceptualnog dijela. U to vrijeme naučnici vještačke inteligencije i robotike (na institutima kao što su MIT, Stanford i CMU) vjerovali su da će problem vizuelnog *inputa* biti samo jednostavan korak na putu do rješavanja nekih mnogo kompleksnijih problema kao što su svhvatanje na visokom nivou (*eng. high-level reasoning*).

U posljednjih četrdeset godina polje računarskog vida znatno je napredovalo, a kratak istorijski pregled najznačajnijih dostignuća u polju računarskog vida dat je u Tabeli 2.1.

Tabela 2.1 – Istorijski pregled dostignuća u polju računarskog vida [1]

|  |  |
| --- | --- |
| Period | Dosignuća |
| 1970e | Winston (1975) i Hanson i Riesman (1978) objavljuju radove na temu rekonstrukcije trodimenzionalne strukture svijeta na osnovu fotografija;  Huffman (1971), Clowes (1971), Waltz (1975) objavljuju radove na temu razvoja algoritama za linearno labeliranje |
| 1980e | Canny (1986), Nalwa i Binford (1986) objavljuju radove o detekciji ivica i kontura;  Procesiranje trodimenzoinalnih podataka (*eng. Three dimensional data processing*), uključuje pribavljanje, spajanje, modelovanje i prepoznavnaje, aktivno se istražuje tokom osamdesetih godina i obrađeno je u radovina koje objavljuju Besl i Jain (1985), Faugeras i Hebert (1987)i mnogi drugi; |
| 1990e | Metode optičkog toka (*eng. Optical Flow*) nastavljaju da se obrađuju u radovima koje objavljuju Nagel i Enkelmann (1986);  Aktivno se istražuju *multi-view stereo* algoritmi u radovima koje objavljuju Seitz i Dyer (1999), kao i Kutulakos i Seitz (2000);  Algoritmi za praćenje (*eng. tracking algorithms*) su se također unaprijedili i obrađeni su u radovima koje objavljuju Kass, Witkin i Terzopoulos (1988), kao i Balke i Isard (1998); |
| 2000e | Razvijaju se mnogobrojne tehnike koje danas spadaju pod dispilinu koja se naziva komputaciona fotografija (*eng. computational photography*), a u koje spadaju spajanje slika (*eng. image stiching*), fotografije visokog dinamičkog opsega (*eng. HDR – High Dynamic Range*);  Sinteza tekstura i *quilting* obrađeni u radovima koje objavljuju Efros i Freeman (2001), Kwatra, Shodl, Essa (2003);  Tehnike proepoznavanja zasnovane na odlikama (*eng. feature based recognition*) obrađene su u radovima Ponce, Hebert i Schmid (2006);  Metode računraskog vida bazirane na mašinskom učenju; |

## 2.2 Primjena

Pojam računarskog vida je širok i obuhvata veliki broj pod-disciplina, pa je samim tim i način na koji se znanja iz ove oblasti primjenjuju raznolik. Samo neke od primjena računarskog vida u današnje vrijeme jesu:

* Optičko prepoznavanje karaktera (*eng. OCR - Optical character recognition*) – korsiti se za automatsko prepoznavanje registarskih tablica na slikama;
* Inspekcija mašina (*eng. Machine Inspection*) – brza inspekcija dijelova koji se koriste u auto i avio industriji da bi se osigurao maksimalni kvalitet konačnog proizvoda;
* Izgradnja trodimenzionalnih modela (*eng. 3D Model Building*) – automatizovani sistemi koji generišu trodimenzionalne modele reljefa na osnovu aero fotografija;
* Automotiv sigurnost (*eng. Automotive safety*) – detekcija neočekivanih prepreka kao što su pješaci na ulici u uslovima gdje klasične tehnike vida nisu baš najbolje rješenje;
* Obrada medicinskih fotografija (*eng. Medical imaging*) – obrada slika u cilju proučavanja morfologije ljudkog mozga dok stari;
* Prepoznavanje otiska prsta i biometrija (*eng. Fingerprint Recognition and Biometrics*) – za automatsku autentikaciju baziranu na skeniranju otiska prsta, kao i za forenzičke aplikacije;
* Nadzor (*eng. Surveillnace*) – detekcija pomjeraja objekta u videu koristi se za nadgledanje uljeza, analizu saobraćaja, kao i na primjer za praćenje stanja bazena u svrhu detekcije utapanja;

Ovaj rad će se primarno fokusirati na posljednju primjenu navedenu u prethodnoj listi, a to je detekcija pomjeraja objekta u videu. Naredna glava daje detaljan uvid u detekciju pomjeraja objekta u videu, kao i u tehnike i algoritme koji se koriste za postizanje ovog cilja.

# 3. DETEKCIJA POMJERAJA OBJEKTA U VIDEU

Potreba da se vrši detekcija pomjeraja objekta u video zapisu postaje sve češća, a broj discpilina koje imaju koristi od metoda detekcije pomjeraja raste svakodnevno. Skoro svi moderni sistemi za nadgledanje (*eng. surveillance systems*) danas posjeduju mogućnost praćenja objekta koji se kreće na sceni, a da bi praćenje objekta bilo moguće potrebno je detektovati svaki vid kretanja tog objekta u videu. Video sistemi za nadzor koji rade u realnom vremenu imaju najviše koristi od metoda detekcije pomjeraja objekta, mada, nije ni rijetkost da se metodi detekcije pomjeraja koriste i za obradu već snimljenog video sadržaja. Na slici 3.1 je prikazan jedan interesantan primjer takve primjene - detekcija pomjeraja i kretanja igrača na video snimku fudbalske utakmice, gdje se dobijene informacije o kretanju objekata (u ovom slučaju igrača) korsiti za analizu i poboljšavanje pozicioniranja individualnih igrača, kao i cijelog tima.



Slika 3.1 – Detekcija pomjeraja igrača na fudbalskoj utakmici

Generalno, proces detekcije pomjeraja objekta na videu smatra se procesom kojim se potvrđuje promjena pozicije objekta ili objekata u odnosu na položaj pozadine, ili promjena pozadine u odnosu na objekat [3]. Kroz prethodnih nekoliko decenija, predloženo je i razvijeno nekoliko tehnika da bi se ispunio ovaj cilj. Naravno, ne postoji savršena metoda detekcije pomjeraja koja je otporna na sve probleme sa kojima se susreće tokom procesa detekcije, a koji se najčešće tiču uslova osvjetljena scene i dinamične promjene istog, brzine kretanja objekta ili objekata na sceni, kao i otpornost na razne vidove šuma koji neprestano ometaju proces detekcije. U nastavku glave obrađene su dvije metode detekcije pomjeraja objekta u videu: detekcija pomjeraja bazirana na razlici frejmova (*eng. Image Substraction*) i detekcija pomjeraja bazirana na estimaciji pokreta uparivanjem blokova (*eng. Block-matching Motion Estimation*). Prva metoda je obrađena samo teorijski, dok se implementacija u nastavku rada (Glava 6) bazira na drugoj metodi, koja je iz toga razloga obrađena u većem detalju.

## 3.1 Razlika frejmova

Razlika frejmova (*eng. Image substraction*) jeste jedna od jednostavnijih i popularnijih tehnika koje se danas koriste da bi se detektovao pomjeraj u videu. Jednostavno, razlika frejmova se može predstaviti izrazom:

|  |  |
| --- | --- |
| *∆I(i,j) = ICurr(i,j) – IPrev(i,j),* | (3.1) |

gdje *i* predstavlja ideks reda frejma, *j* predstavlja indeks kolone frejma, *ICurr(i,j)* predstavlja matricu intenziteta piskela trenutnog frejma, a *IPrev(i,j)* predstavlja matricu intenziteta piksela prethodnog frejma. *∆I(i,j)* predstavlja matricu koja se dobije razlikom intenziteta piksela na odgovarajućim pozicijama za dva uzastopna frejma.

|  |  |
| --- | --- |
|  |  |
|  | |

Slika 3.2 – Rezultat razlike dva uzastopna frejma u videu

Kao što se može vidjeti na Slici 3.2, rezultat razlike dva uzastopna frejma u videu jeste intenzitetska slika na kojoj vrijednost intenziteta piksela na pozicijama gdje nije bilo razlike iznosi nula ili približno nula, dok se pozicijama gdje je postojala određena razlika između frejmva nalaze pikseli intenziteta koji je veći od nula. Na osnovu intenziteta piskela na rezultantnoj slici može se zaključiti da li je došlo do pomjeraja objekata na slici, i ako jeste, na kojoj lokaciji frejma se taj pomjeraj desio.

Iako prilikom detekcije pomjeraja baziranoj na razlici dva uzastopna frejma dobijamo zadovoljavajuće rezulate kada se radi o statičnom frejmu (pomjeraj pozadine i objekata u pozadini je nepostojeći ili minimalan), ovaj način detekcije pomjeraja ne daje zadovoljavajuće rezultate kada se isti princip korsiti na video sa pozadinom koja se pomjera. Isto tako, problem predstavlja i promjena osvjetljenja scene, a tada je razlika intenziteta piksela postojeća na svim pozicijama na frejmu. Ovaj problem se može ublažiti tako što se korsiti metoda usrednjavanja razlika više uzastopnih frejmova, ali ovaj rad se u nastavku neće osvrtati na navedeno poboljšanje. Jedan od problema na koji ovaj pristup takođe nailazi jeste šum pozadine koji ne treba da bude detektovan kao pomjeraj. Način nakoji se ovaj šum može otkloniti jeste korištenjem praga intenziteta na frejmu rezlike intenziteta. Korištenje praga dato je izrazom:

|  |  |
| --- | --- |
| *∆I(i,j) = 0, ako važi ∆I(i,j) < t,* | (3.2) |
| *∆I(i,j) = 1, ako važi ∆I(i,j) > t,* |  |

gdje *t* predstavlja prag odbacivanja šuma. U zavisnosti od vrijednosti praga, rezultantna razlika će biti manje ili više tolerantna na šum. Prevelika vrijednost praga *t* može da odbaci i korsine informacije pa ne postoji najbolja vrijednost praga, te se ta vrijednost određuje eksperimentalno u zavisnosti od situacije.

|  |  |
| --- | --- |
|  |  |

Slika 3.3 – Rezultantni frejm prije i nakon filtriranja šuma uz pomoć praga

## 3.2 Estimacija pokreta uparivanjem blokova

Estimacijom pokreta smatra se proces kojim se određuju vektori pomjeraja koji opisuju pomijeranje objekata između dva uzastopna frejma. Isto tako se može reći da vektori pomjeraja opisjuju kako se jedan frejm transformiše u drugi. Ovaj problem nije dobro definisan zato što slike zapravo predstavljaju projekciju trodimenzionalne scene u dvodimenzionalnu ravan pa korespondencija između pokreta objekata u 3D okruženju i 2D ravni nije jednoznačna. Međutim ako je broj frejmova u videu veliki, onda se može pretpostaviti da je količina pomjeraja objekata između dva uzastopna frejma mala što omogućava dovoljno dobru estimaciju koja se može iskoristiti da bi se detektovao pomjeraj objekta u videu [2]. Veliki je broj algoritama koji se korsite da bi se odredili vektori pomjeraja kao što su: uparivanje blokova, određivanje optičkog toka, fazna korelacija, piksel-rekurzivni algoritmi, algoritmi zasnovani na detekciji i uparivanju obilježja, itd. Ovaj rad se u nastavku fokusira na algoritam uparivanja blokova. Estimacija pokreta pored detekcije pomjeraja objekta ima i druge primjene, a primarno se korsiti kao korak prilikom kompresije video sadržaja. U nastavku, rad se isključivo fokusira na primjenu estimcije pokreta radi detekcije pomjeraja objekta u videu.

Algoritam uparivanja blokova u opštem slučaju pronalazi vektore pomjeraja blokova između dva frejma – referentni frejm i trenutni frejm. U slučaju primjene estimacije pokreta radi detekcije pomjeraja, smatra se da su ta dva frejma uzastopna ako nije drugačije naglašeno, a referenciraće se kao trenutni i prethodni frejm. Prvi korak algoritma jeste da se trenutni frejm podijeli na blokove fiksne veličine. Veličina blokova se bira pod pretpostavkom da se prostorno bliski pikseli kreću na sličan način. Za svaki blok *B* na trenuntom frejmu potrebno je odrediti odgovarajući blok sa prethodnog frejma tako da mjera nesličnosti blokova poprima minimalnu vrijednost. Za mjeru nesličnosti blokova najčešće se koriste suma apsolutnih razlika intenziteta piksela između frejmova ili suma kvadrata razlika. Suma apsolutnih razlika definisana je izrazom:

|  |  |
| --- | --- |
| *D(d1, d2) =* | (3.3) |

dok je suma kvardata razlika data izrazom:

|  |  |
| --- | --- |
| *D(d1, d2) =* | (3.4) |

gdje *k* predstavlja indeks trenutnog frejma, a [*d1, d2*] predstavlja vektor pomjeraja između blokova koje posmatramo. U nastavku rada razmotrene su tri metode određivanja najsličnijeg bloka sa prethodnog frejma, a to su: iscrpna(sekvencijalna) metoda pretrage, metoda pretrage od tri koraka i dijamantska metoda pretrage.

### 3.2.1 Iscrpna metoda pretrage

U opštem slučaju da bi algoritam uparivanja blokova našao blok sa prethodnog frejma koji najbolje odgovara referentnom bloku sa trenutnog frejma, potrebno je da se poredi sa svim blokovima prethodnog frejma, što je veoma zahtjevno računarski pa se prostor pretrage u praksi ograničava takozvanim prozorom pretrage. Prozor pretrage je prostor koji se dobija tako što se na blok sa prethodnog frejma koji odgovara pozicijom bloku trenutnog frejma doda fiksno određeno proširenje sa svake od četiri strane bloka, čime površina prozora pretrage postaje veća od površine bloka. Ograničavanje oblasti koja se pretražuje zasniva se na pretpostavci da su pomjeraji objekta mali između susjednih frejmova i da će upareni blok biti pronađen u neposrednoj blizini referentnog bloka [2]. Uobičajeno je da se proširenje u horizontalnom i vertikalnom pravcu ograniči na veličinu bloka. Tako da ako se uzme da su dimenzije bloka *p* × *p,* onda horizontalne i vertikalne vrijednosti proširenja mogu imati vrijednosti iz intervala [-*p, p*], pa je oblast pretrađivanja dimenzija (2*p +* 1) × (2*p +* 1) [2].

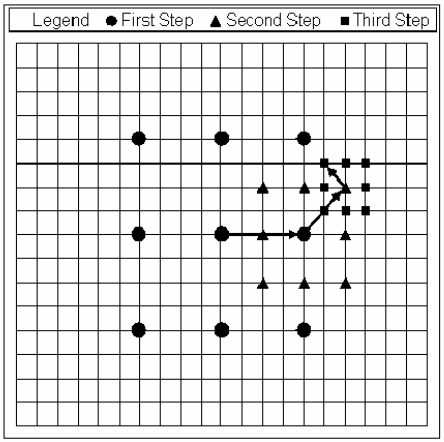


Slika 3.4 – Lijevo je prikazan treuntni frejm i blok koji se obrađuje, desno je prikazan prethodni frejm sa određenim prozorom pretrage i uparenim blokom [2]

Proces pretrage unutar prozora odvija se klizanjem referentnog bloka sa trenutnog frejma preko svih blokova koji se nalaze unutar prozora pretrage i za svaki par se računa mjera nesličnosti. Blok iz prozora pretrage koji je dao minimalnu vrijednost mjere nesličnosti uzima se kao najbolji rezultat, a vektor pomjeraja za referentni blok dobija se kao razlika koordinata u lijevom gornjem uglu referentnog bloka i bloka koji je određen kao najbolje poklapanje unutar prozora pretrage. Uzmimo za primjer referentni blok sa ugaonom koordiantom (200, 216), ako je najbolje poklapanje određeno za blok sa koordiantama (205, 212) vektor pomjeraja za dati referentni blok izosi [5, -4]. Na isti princip se određuje vektor pomjeraja za svaki blok koji je dobijen dijeljenjem referentnog frejma na blokove, a kao rezultat cijelog procesa dobija se matrica vektora pomjeraja čiji je broj redova jednak broju redova blokova kada se referentni frejm podijeli na blokove fiksnih dimenzija, a broj kolona samim tim jednak broju kolona blokova u referentnom frejmu nakon dijeljenja na blokove. I sa ograničenim prostorom pretrage iscrpna metoda je i dalje računarski zahtjevna metoda pa su predložene mnoge varijacije optimizacije pretrage, a u nstavku su obrađene pretraga od tri koraka i dijamantska metoda pretrage.

### 3.2.2 Metoda pretrage od tri koraka

Smatra se jednim od najranijih pokušaja da se implementira brza metoda za uparivanje blokova, a po prvi put se pojavljuje još sredinom osamdesetih godina prošloga vijeka [5]. Na Slici 3.5 prikazana je generalna ideja iza ovog algoritma.



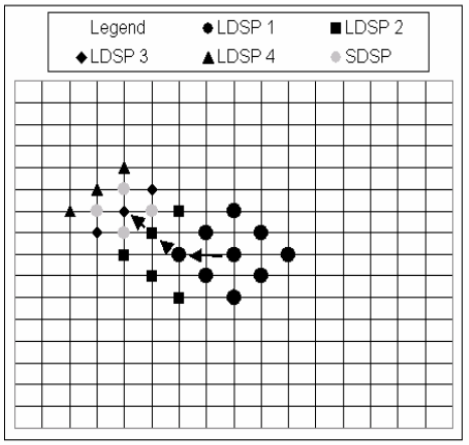
Slika 3.5 – Procedura metode pretrage od tri koraka. Tačkama su označene lokacije pretrage tokom prve faze, troulgovi predstavljaju lokacije pretrage tokom druge faze pretrage, a pravougaonici lokacije pretrage treće faze. Strelicama je označen pravac napredovanja algoritma kroz faze [5]

Pretraga bloka se ovaj put ne odvija unutar prozora pretrage, nego se u tri uzastopna koraka algoritma definišu fiksne lokacije na kojima se blokovi porede sa referentnim blokom sa trenutnog frejma. Pretraga na prostoru prethodnog frejma započinje na istoj lokaciji na kojoj se nalazi i referentni blok na trenutnom frejmu, tačnije iz centra lokacije referentnog bloka. Za izvšavanje algoritma bitan je parametar *step\_size (S)* koji se na početku postavlja na vrijednost *S =* 4. Pored izračunavanja mjere nesličnosti za centralnu lokaciju, mjera nesličnosti se računa i za sedam okružujućih blokova koji se od centralnog bloka nalaze na distancama – *S*  i *S* po horizontalnoj i vertikalnoj osi tako da se ukupno vrši devet poređenja u prvoj fazi algoritma. Blok za koji je izračunata minimalna vrijednost mjere nesličnosti uzima se kao novi centralni blok i prelazi se u drugu fazu algoritma. U drugoj fazi algoritma vrijednost koraka pomjera se polovi i dobija vrijednost *S =* 2. Kao i kod prve faze algoritma, računa se mjeru nesličnosti za referentni blok i za novi centralni blok sa svojih osam okružujućih blokova koji se kao i u prvoj fazi od centralnog bloka nalaze na distancama od – *S* i *S* po horizontalnoj i vertiklanoj osi. Kada se odredi blok sa minimalnom vrijednošću mjere nesličnosti prelazi se na finalnu treću fazu koja je ista kao i prve dvije faze, samo što se vrijednost koraka pomjeraja polovi i uzima vrijednost *S =* 1. Blok koji je u trećoj fazi određen kao blok sa minimalnom mjerom nesličnosti u odnosu na referentni blok uzima se kao najbolje poklapanje, a vektor pomjeraja se računa za njegovu poziciju i za poziciju referentnog bloka.

Ova implementacija algoritma daje značajno poboljšanje u perofrmansama u odnosu na iscrpnu metodu pretrage, a izvršava se čak i do devet puta brže [5]. U situaciji iscrpne metode pretrage sa proširenjem *p =* 7, da bi se pronašlo najbolje poklapanje za samo jedan blok sa referentnog frejma potrebno je izračunati vrijednosti mjere nesličnosti za 225 različitih blokova iz prozora pretrage sa prethodnog frejma. Sa druge strane, metoda pretrage od tri koraka svaki put fiksno radi dvadeset i sedam izračunavanja mjere nesličnosti prilikom pretrage jednog bloka. Za manu ovoga algoritma može se uzeti to što zbog ograničenog broja blokova nad kojima će se računati mjera nesličnosti ne pronalazi uvijek blok sa minimalnom mjerom nesličnosti. U slučaju detekcije pomjeraja objekata ova mana ne predstavlja veliki problem ali u slučaju primjene algoritma kao koraka kompresije video sadržaja (koja je prethodno navedena kao jedna od čestih primjena za algoritme estimacije pokreta) mogu se dobiti lošiji rezultati u odnsu na iscrpnu metodu pretrage. Poboljšanje ovog algoritma dato je u vidu takozvane nove pretrage od tri koraka (*eng. NTSS – New Three Step Search*) [5] koji je jedan od prvih široko rasprostranjenih brzih algoritama za uparivanje blokova ali koji neće biti obrađen u nastavku rada.

### 3.2.3 Dijamanstka metoda pretrage

Dijamanstka metoda pretrage jeste algoritam čiji je šablon pretrage sličan pretrazi kod metode pretrage od tri koraka, samo što se umjesto šablona pravougaonika koristi šablon dijamanta. Na Slici 3.6 prikazana je generalna ideja iza ovog algoritma.



Slika 3.6 – Procedura dijamantske metode pretrage. Na slici je crnim tačnakma označen šablon velikog dijamanta, dok je sivim tačkama označen šablon malog dijamanta. Strelicama je označen pravac napredovanja algoritma kroz faze [5]

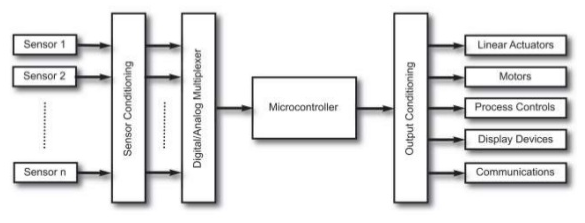
Kao i kod metode pretrage od tri koraka, ni ova metoda ne posjeduje prozor pretrage. Dijamantska metoda pretrage tokom pretraživanja korsiti dva fiksna šablona pretrage, a to su šablon velikog dijamanta pretrage (*eng. LDSP – Large Diamond Search Pattern*) i šablon malog dijamanta pretrage (*eng. SDSP – Small Diamond Search Pattern*). Slično kao i kod pretrage od tri koraka, u prvoj fazi algoritma vrši se izračunavnaje mjere nesličnosti između referentnog bloka sa trenutnog frejma i 9 blokova sa prethodnog frejma koji se nalaze na lokacijama koje odgovaraju tačkama šablona velikog dijamanta pretrage. Lokacija bloka sa najboljim poklapanjem uzima se kao nova centralna lokacija i prelazi se u drugu fazu algoritma u kojoj se izračunava mjera nesličnosti između referentnog bloka sa trenutnog frejma i 5 blokova sa prethodnog frejma koji se nalaze na lokacijama koje sada odgovaraju tačkama šablona malog dijamanta pretrage. Ako se ispostavi da je minimalna vrijendost mejere nesličnosti izračunata za centralni blok dijamanta, algoritam se prestaje izvršavati i blok se uzima kao blok sa najboljim poklapanjem. U suprotnom, lokacija sa najboljim poklapanjem postaje nova centralna lokacija i ponavlja se druga faza algoritma. Algoritam uobičajeno nema ograničen broj koraka da bi mogao sa što većom preciznošću odrediti blok sa najboljim poklapanjem.

U situacijama kada se pozicija objekata u videu mijenja sporo iz frejma u frejm, algoritam daje dobre rezultate uz znatno manji broj koraka potrebnih da se odredi najbolje poklapanje u odnosu na iscrpnu metodu pretrage. Sa druge strane, ako se pozicija objekata u videu mijenja brzo algoritam mora da obradi znatno veći broj koraka da bi pronašao najbolje poklapanje pa se u takvim situacijama performanse algoritma degradiraju [5].

# 4. UGRAĐENI SISTEMI

Paralelno sa napretkom računarskih tehnologija, sve više se javljaju električni uređaji koji zavise od interno ugrađenih računarskih sistema (*eng. Embedded System*). Mogućnosti koje ti električni uređaji koji zavise od ugrađenih sistema posjeduju prevazilaze mogućnosti koje se ostvaruju implementacijom na isključivo hardverskom nivou. Iz tog razloga, ugrađeni sistemi se danas mogu naći u širokom opsegu namjenskih uređaja, od onih najmanjih kao što su električni tajmeri gdje se ugrađeni sistemi korsite za male količine procesiranja, pa sve do znatno kompleksnijih kao što su igračke konzole i mnogi drugi industrijski proizvodi.

Ugrađeni sistem nije standardan sistem kao što je personalni računar koji može da izvršava veliki broj različitih programa i obavlja još veći broj pozadinskih zadataka. Ugrađeni sistem uvijek se fokusira na izvršavanje jednog specifičnog programa ili zadatka za koji je isprogramiran. Da bi se ispunila ova potreba, uređaji koji u sebi sadrže ugrađene sisteme na njima imaju preinstaliran softver koji se ne mijenja često ali postoji mogućnost ažuriranja sa vremena na vrijeme [6].



Slika 4.1 – Uprošćena arhitektura ugrađenog sistema

Generalno, ugrađeni sistemi definišu se kao bilo koji računarski sistemi sadržani unutar nekog proizvoda ili sistema koji se ne klasifikuje kao računarski sistem [6].

Softver ugrađenog sistema klasično se naziva *firmware.* Za razliku od standardnog softvera na računarima koji se čuva na diskovima, *firmware* se čuva na samome čipu [6]. Ugrađeni sistemi sastoje se od dvije glavne komponente, a to su: hardverska komponenta i softerska komponenta. Hardverska komponenta je hardverska platforma na kojoj se softverska komponenta izvršava. Hardvarska komponenta može biti bazirana na mikroprocesoru i na mikrokontroleru. Uz to, hardverska komponenta uključuje i ostale elemente kao što su: memorija, interfejsi za ulaz i izlaz i slično.

## 4.1 TDA2Px evaluacioni modul

# 5. DSP

# 6. IMPLEMENTACIJA ALGORITMA DETEKCIJE POMJERAJA

# 7. OPTIMIZACIJA ALGORITMA DETEKCIJE POMJERAJA

# 8. TESTIRANJE I ANALIZA REZULTATA TESTIRANJA

# 9. ZAKLJUČAK

# LITERATURA

|  |  |
| --- | --- |
| [1] | R. Szeliski, “*Computer Vision: Algorithms and Applications*”, 2011 |
| [2] | V. Risojević, “*Multimedijalni sistemi*”, 2018 |
| [3] | S. Alavi, “*Comparation of Some Motion Detection Methods in cases of Single and Multiple Moving Objects*”, 2012 |
| [4] | B. Pesquet-Popescu, M. Cagnazzo, “*Motion Estimation Techniques*”, 2015 |
| [5] | A. Barjatya, “*Block Matching Algorithms for Motion Estimation*”, 2004 |
| [6] | Underestanding Embedded Systems, <https://www.electronics-notes.com/articles/digital-embedded-processing/embedded-systems/basics-primer.php>, posjećeno 10.10.2019 |
| [7] |  |
| [8] |  |
| [9] |  |
| [10] |  |
| [11] |  |