Detekcija usmjerenosti pogleda i upravljanje računalom primjenom računalnog vida

Gizdić, Marija

Undergraduate thesis / Završni rad

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, University of Split, Faculty of science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:166:738151

Rights / Prava: Attribution-NonCommercial 4.0 International/Imenovanje-Nekomercijalno 4.0 međunarodna

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Repository of Faculty of Science

zir.nsk.hr

UNIVERSITY OF SPLIT

dabar

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

ZAVRŠNI RAD

DETEKCIJA USMJERENOSTI POGLEDA I UPRAVLJANJE RAČUNALOM PRIMJENOM RAČUNALNOG VIDA

Marija Gizdić

Split, srpanj 2015.

Sadržaj

Uvod 4
1. Prepoznavanje 5
1.1. Prepoznavanje objekata 5
1.2. Prepoznavanje lica 6
2. Praćenje pogleda 8
2.1. Pogled i komunikacija 9
2.1.1. Pogled kao ulazna informacija 10
2.2. Povijest praćenja pogleda 11
3. Temelji praćenja usmjerenosti pogleda 13
3.1. Fiziologija oka 13
3.2. Dinamika oka 14
3.3. Tehnike praćenja oka 15
3.3.1. Nametljive tehnike 15
3.3.2. Nenametljive tehnike 16
4. Opis razvijenog rješenja i implementacija 19
4.1. Opis problema 19
4.2. Razrada rješenja problema 19
4.2.1. Detekcija lica 20
4.2.2. Određivanje regije oka 21
4.2.3. Detekcija oka 22
4.2.4. Detekcija centra oka i čitanje koordinata 23
4.2.5. Prebacivanje koordinata na prozor videa 24
4.2.6. Prebacivanje koordinata na ekran 24
4.2.7. Postavljanje pokazivača miša 25
4.3. Implementacija 25
Zaključak 27
Literatura 28
Popis slika 29
Sažetak 30
Summary 31
Skraćenice 32
Privitak 33

Uvod

Računalni vid je područje računalne znanosti koje obuhvaća metode za stjecanje, obradu, analizu i razumijevanje slike te, općenito, višedimenzionalne podatke iz realnog svijeta u cilju dobivanja numeričkih ili simboličkih informacija.

Ovo područje se fokusira na obradu i analizu 2D slike, poput operacija nad pikselima kao što su poboljšanje kontrasta, lokalne operacije, izdvajanje rubova ili uklanjanje šuma, ili geometrijskih transformacija kao što je rotacija slike.

Primjena računalnog vida je prepoznavanje lica. Prepoznavanje lica je računalni program za automatsko prepoznavanje i provjeru osoba iz digitalne slike ili video izvora. Slično, može se prepoznati položaj očiju te usmjerenost pogleda što je usko vezano za temu ovog završnog rada.

U ovom radu je prikazan jednostavan sustav upravljanja pokazivačem miša praćenjem pogleda koristeći samo običnu web kameru. Opisani algoritam omogućuje prepoznavanje pogleda u sustavu obične web kamere, računala i korisnika te prepoznatu točku pretvara iz područja koordinata pozicije oka na koordinate kursora miša. Pomicanjem oka upravlja se pokazivačem miša na ekranu.

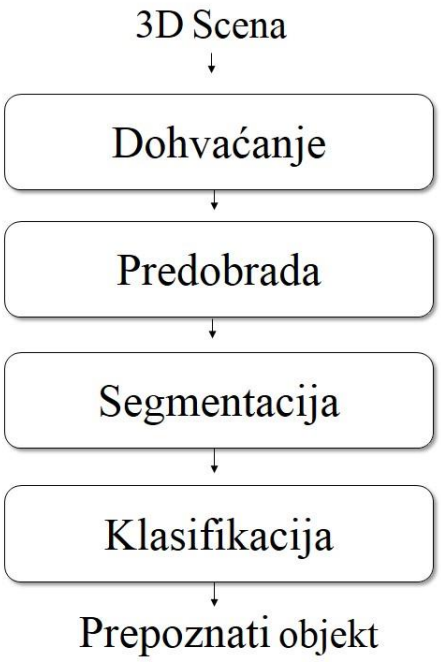
U prvom poglavlju dan je osvrt na dio računalnog vida koji se bavi prepoznavanjem objekata i lica. Drugo poglavlje se bavi praćenjem pogleda, opisana je povijest, raširenost, primjene i izazovi. Treće poglavlje pruža pregled fiziologije oka, osnovne vrste pokreta oka, određena ograničenja i probleme prilikom estimacije pogleda, kao i tehnička ograničenja. Četvrto poglavlje nudi opis problema, razradu rješenja i implementaciju.

1. Prepoznavanje

1.1. Prepoznavanje objekata

Klasični problem računalnog vida, obrade slika, i strojnog vida je utvrđivanje sadrži li slika određeni objekt, značajku ili aktivnost. Ovaj zadatak se obično može riješiti trivijalno i bez truda od strane čovjeka, ali još uvijek nije zadovoljavajuće riješen problem računalnog vida za opći slučaj - proizvoljni objekt u proizvoljnim situacijama. Razvijene metode za rješavanje ovog problema se u najboljem slučaju mogu primijeniti samo za točno određene objekte, kao što je jednostavni geometrijski oblik (npr. poliedar), lice čovjeka, vozila, tiskani ili ručno pisani znakovi. U određenim situacijama dobro postavljenog osvjetljenja, moguće je prepoznati i pozadine i položaj predmeta u odnosu na kameru.

Općenitog rješenja nema jer su zadaci koje bi takav sustav trebao rješavati raznovrsni i složeni. Slika 1.1 prikazuje tipični algoritam za prepoznavanje objekata na slikama. Problem prepoznavanja objekata usko je povezan s problemom segmentacije slike koja podrazumijeva da je izvršeno barem djelomično prepoznavanje elemenata slike. Također, ako nije obavljena segmentacija koja, ako je uspješno obavljena, za rezultat daje elemente slike koji predstavljaju određene objekte ili dijelove objekata, ni prepoznavanje nije moguće.



Slika 1.1 Tipična shema prepoznavanja objekta u digitalnoj obradi slika

1.2. Prepoznavanje lica

Prepoznavanje lica je jedan od najizazovnijih problema računalnog vida.

Postoji više metoda za prepoznavanje lica. Jedna od metoda je usporedba odabranih lica iz slike i lica iz baze podataka. Ovo se obično koristi u sigurnosnim sustavima i može se usporediti s drugim biometrijskim podacima kao što su otisak prsta ili zjenica oka. Neki algoritmi vrše prepoznavanje pomoću određenih značajki lica. Na primjer, algoritam može analizirati relativnu poziciju, veličinu ili oblik očiju, nosa, jagodice i čeljusti. Ove značajke tada koristi za traženje drugih slika s istim značajkama. Drugi algoritmi normaliziraju galeriju slika lica, a zatim komprimiraju podatke lica te spremaju podatke slike koji su korisni za otkrivanje lica. Prototipna slika se zatim uspoređuje s podacima lica.

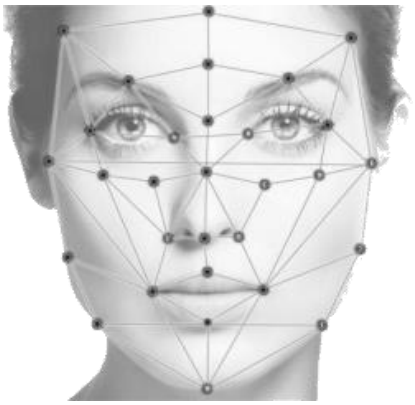
Jedan od prvih uspješnih sustava se temelji na predlošku podudarajućih tehnika koji se primjenjuju na niz izraženih crta lica. Priznavanje algoritama može se podijeliti u dva glavna pristupa, geometrijski, koji promatra razlikovna obilježja, ili fotometrijska, što je statistički pristup koji pretvara sliku u vrijednosti koje se uspoređuju sa vrijednostima predložaka da bi se otklonila odstupanja.

Formalnu metodu klasifikacije lica prvi je iznio Francis Galton 1888. koji je predložio skupljanje profila lica kao krivulja. Ova metoda, iako je imala mnoga odstupanja, rezultirala je vektorima nezavisnih mjera koji su se mogli usporediti s drugim vektorima u bazi podataka. Prepoznavanje lica je vrlo izazovan zadatak zbog varijabilnosti obilježja u fotografiji. Različite varijance skaliranja, orijentacije, pozicije lica su samo neke poteškoće za izgradnju takvog sustava.

Glavni izazovi povezani s detekcijom lica navedene su u Yangovom istraživanju iz 1939. godine [1]. Primijetio je probleme kao što su položaj lica u odnosu na kameru, slika profila ili obrnuta slika. Također, neke značajke očiju, usta koja mogu biti otvorena ili zatvorena. Prisutnost ili odsutnost strukturalnih komponenti. Značajke lica kao što su brada, brkovi, naočale mogu i ne moraju biti prisutni. Lice može biti djelomično zatvoreno od drugi objekata. I na kraju treba uzeti i čimbenike kao što su rasvjeta, karakteristike kamere koje isto utječu na izgled lica. Postoji više od sto različitih tehnika za prepoznavanje lica na slikama. Većina tih metoda koriste iste tehnike za ekstrakciju značajki.

Zadnjih 15 godina prepoznavanje lica je postalo popularno ne samo istraživačima računalnog vida već i u neuro-znanosti zbog jedinstvenosti lica. Nadolazeći trend, za koji se tvrdi da postiže

bolju točnost, je trodimenzionalno prepoznavanje lica (Slika 1.2). Ova tehnika koristi 3D senzore za hvatanje podatka o obliku lica. Te se informacije koriste za identifikaciju razlikovnih obilježja na površini lica, kao što su konture očne šupljine, nosa i brade, na svako se obilježje stavlja točka. Svako ljudsko lice ima 80 takvih točaka. Jedna od prednosti 3D prepoznavanje lica je to da na njega ne utječu promjene u rasvjeti kao na druge tehnike.



Slika 1.2 Trodimenzionalno prepoznavanje lica

Sustavi za kontrolu računala bez potrebe korištenja miša i tipkovnice, dakle implementacija lica i pogleda dobivaju sve veću pozornost u novijoj literaturi.

2. Praćenje pogleda

Razvojem računala razvila se i potreba za novim načinima komuniciranja između korisnika i računala što je dovelo do razvoja posebnog područja HCI (*eng. Human computer interaction*) sa ciljem razvoja lakše, intuitivnije i učinkovitije komunikacije sa računalima. Praćenje pogleda je jedini način komuniciranja za osobe sa određenim bolestima, ali osim toga postoji niz područja u kojima se primjenjuje praćenje pogleda, najčešće u kombinaciji sa drugim ulaznim jedinicama.

Pojam praćenja pogleda označava procjenu usmjerenosti pogleda neke osobe. Većinom se pritom misli na određivanje objekta na koje se pogled fokusira, ali kada govorimo o računalima najčešće mislimo na određivanje piksela odnosno regije na ekranu. Postoje i složeniji sustavi koji prate usmjerenost pogleda u 3D prostoru.

Praćenje pogleda se koristi u istraživanju u vizualnim sustavima, u psihologiji, u kognitivnoj lingvistici i dizajnu proizvoda. Postoji niz metoda za mjerenje kretanja očiju. Najpopularnija varijanta koristi video snimke iz kojih se vadi pozicija oka.

Većina današnjih uređaja dolazi sa već ugrađenim kamerama; prijenosna računala, mobiteli, tableti, pa i neki monitori. Povećanjem snage računala, odnosno procesora, obrada signala u realnom vremenu postaje moguća i na najmanjim uređajima, što omogućuje i širenje tehnologije praćenja oka. Veliki broj ljudi većinu radnog vremena provode uz računala, uostalom kao i slobodnog, koriste se tipkovnicama i miševima, koristeći obje ruke, što dovodi do različitih zdravstvenih problema. S obzirom na taj stalni porast interakcije sa računalima, postoji potreba i za alternativnim načinima upravljanja računalom. Unatoč padu cijena, i procjenama kako je širenje tehnologije samo stvar vremena, i kako će uskoro svatko imati uređaj za praćenje oka po cijeni miša ili današnje web kamere to se još nije dogodilo.

2.1. Pogled i komunikacija

Lice je najekspresivniji dio tijela. Mnogo je dijelova lica koji svojim pomakom odaju emocije, stavove, fizičko i psihičko stanje osobe. Promatraju se oči i dio oko očiju, usta, obrve, očni kapci, nos, boja i zategnutost kože lica, kao i lice u cjelini.

Kaže se da je oko zrcalo duše. U mnogim životnim situacijama pogledi su dostatni kao jednoznačni znakovi da bi se uspostavila komunikacija između osoba. Kontakt pogledom je važno sredstvo. Oči mogu odašiljati poruke. Oči ih i primaju.

Kao važan detalj u komunikaciji pogledom je i kulturalna razlika. Npr. neki Južnoamerikanci ne gledaju sugovornika u oči tijekom razgovora, nego nevažne predmete oko sebe. Nepristojno je dugo uzvraćati pogled jer to pokazuje nepoštovanje [2].

Kontakt očima i pogled su vrlo moćni neverbalni znakovi: širom otvorene oči i proširene zjenice pokazatelj su sviđanja. Smještaj i duljina pogleda ukazuju na interes i emocije i pomaže nam u usklađivanju komunikacije s drugom osobom. Skrivanje pogleda često je znak neiskrenosti, ali može biti i znak neugode.

Čak i bez ikakvih istraživanja, znamo da će osoba drugačije proučavati sliku ukoliko se na njoj nalazi neki tekst kojeg zna čitati nego ukoliko nije upoznata sa jezikom tog teksta. Još su davno obavljena istraživanja koja pokazuju kako osobe drugačije promatraju fotografije ukoliko se fokusiraju na neki od zadataka. Unatoč nekim univerzalnim biološkim datostima trzaja oka, takva kretanja ovise o osobi.

Činjenica da ljudi koriste oči kao sredstvo komunikacije dovela je do mnogih zaključaka za interakciju čovjeka i računala. Imamo sposobnost kontroliranja pogleda (oka) zato ga možemo koristiti kao ulaznu jedinicu.

2.1.1. Pogled kao ulazna informacija

Tradicionalno, nepokretni ljudi koriste sustave praćenja kako bi komunicirali sa okolinom [3]. U posljednje vrijeme, čini se kako bi praćenje pogleda moglo zamijeniti, barem dio, postojećih interakcija sa računalom. Iako su sustavi koji pomažu nepokretnima velika pomoć, takvi sustavi su u pravilu sporiji za korištenje od miša ili tipkovnice. Neka istraživanja uspješnosti određenih akcija pokazuju kako upravljanje pogledom može biti brže. Činjenica da su takvi sustavi ipak većinom sporiji od tradicionalnih ulaza, nije u skladu sa činjenicom kako je pogledom lagano i brzo upravljati [4]. Sustavi praćenja pogleda mogu ponuditi više prednosti:

- lagano korištenje
- ubrzanje interakcije
- nema potrebe za održavanjem
- daljinsko upravljanje
- sigurnija interakcija
- higijensko sučelje
- više informacija o korisnikovim aktivnostima

Naravno, postoje i problemi:

- mogućnost kontrole

  Oči imaju niz nekontroliranih pokreta što bi moglo ozbiljno narušiti mogućnost oka kao ulaza. Nije jasno do kojeg stupnja ljudi mogu kontrolirati svoj pogled. Jasno je kako oči možemo kontrolirati u društvenim interakcijama, ali nije jasno možemo li istrenirati oko kao što možemo istrenirati prste da sviraju klavir.

- konflikt ulaza i vida

  Primarna funkcija oka je vid. Korištenje oka za ulazne podatke može dovesti do konflikta sa vidom. Sustav mora biti u stanju razlikovati prirodne geste od namjernih. Različiti objekti koji se kreću ili na neki drugi način odvlače pažnju mogu rezultirati sa dodatnim problemima na koje bi trebalo paziti prilikom osmišljavanja.

- zamor očnih mišića

  Na temelju drugih ulaznih jedinica poznato je kako neke dovode do zamora određenih mišića odnosno mišićnih grupa što rezultira fizikalne probleme, takozvani RSI (*eng.*

repetitive strain injury). Postoji bojazan, da je isto moguće i za očne mišiće, ali kako su oni stalno u uporabi, opasnost se čini manjom.

2.2. Povijest praćenja pogleda

Kao što je već spomenuto, Yang je 1939. godine iznio glavne izazove vezane uz prepoznavanje lica. Mjerio je horizontalne i vertikalne pomake oka s elektrodama na koži blizu oka. Nakon Drugog svjetskog rata, točnije 1947. godine Fitts, Jones i Milton [5] objavljuju podatke sa više od 500,000 filmskih okvira od preko 40 pilota. Naime, koristili su kamere kako bi snimali pokrete oka pilota prilikom spuštanja aviona. I došli do zaključka: „Ako znamo gdje pilot gleda, nije nužno znati što on razmišlja jer već znamo nešto o čemu on misli.“ [5].

Velika inovacija u praćenje oka je uređaj koji se montira na glavu Eye-tracker (Slika 2.1), ova tehnika se još uvijek naširoko koristi. [6] Još jedan značajni rad u svijetu praćenja pogleda je objavio Yarbus. Yarbus je ruski psiholog koji je proučavao pokrete oka i istraživao složene slike 1950-ih i 1960-ih. Snimao je pokrete očiju koje obavlja promatrač dok gleda prirodne objekte i prizore [7]. U svom radu, Yarbus, nastoji pokazati da je smjer pogleda presudan u interaktivnosti, odnosno da putanja pogleda ovisi o zadatku koji promatrač treba izvršiti. Veliki dio važnih radova u 1970-ima fokusiran je na tehnička poboljšanja.

Slika 2.1 Eye-tracker

Tijekom 80-ih, interes za ovim područjem sve više raste. Veliki napredak osobnih računala donosi nove načine razmišljanja u odnosu nas i računala. Još jedan pionir ovoga područja je i Levine koji je među prvima vidio potencijal praćenja oka u interaktivnim aplikacijama. Kroz 15 godina, oči i praćenje pogleda postali su i od industrijskog značenja, mnogi proizvođači su

razvili proizvode za ovo područje istraživanja. Dramatično povećanje broja publikacija oko ove teme, mnogi časopisi, konferencije se počinju zanimati za ovo područje (konferencije: ECEM - *the European Conference on Eye Movements*, SWAET - *the Scandinavian Workshop on Applied Eye-tracking*, ETRA - *Eyetracking Research and Applications*...).

Iako sustavi za praćenje pogleda postoje već jako dugo vremena, praćenje pokreta oka i smjera pogleda je vrlo složen i skup zadatak, rezerviran za istraživanja u vojnim laboratorijima. Uređaji za praćenje očiju su neugodni i teški za korištenje. Međutim, brz tehnološki napredak (povećana brzina procesora, napredna digitalna video obrada) omogućuju poboljšanje opreme za praćenje pa tako i dramatično povećavaju učinkovitost oka.

3. Temelji praćenja usmjerenosti pogleda

Oko je glavni organ kojim percipiramo svijet oko sebe. Koristimo oči kako bi prikupili informacije o okolini tako što usmjerimo pogled na neki objekt. Osim toga, oči imaju važnu ulogu u društvenoj interakciji; iz njih možemo saznati o nečijem raspoloženju, pažnji i emocionalnom stanju. Oko je organ koji tradicionalno promatramo kao, računalnim rječnikom, ulaznu jedinicu, naravno za osobu čije je oko, ali ono također može poslužiti kao vanjska jedinica koja signalizira okolini o našem stanju i fokusiranosti.

3.1. Fiziologija oka

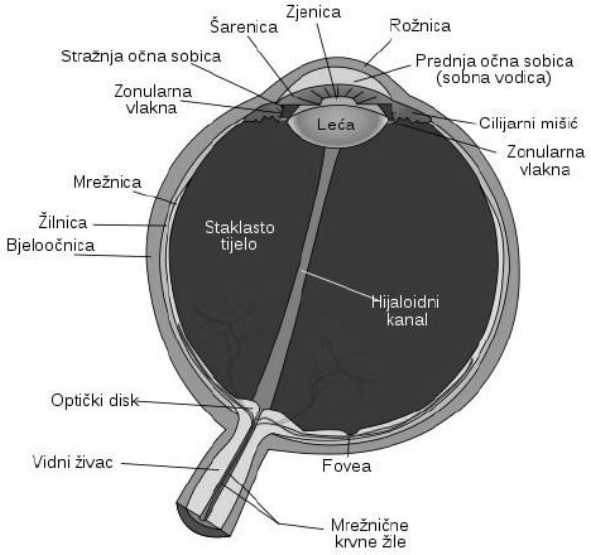
Oko je organ mnogih životinja i čovjeka koji služi pretvaranju svjetlosti u živčane impulse. Ljudsko oko djeluje slično fotoaparatima i kamerama: prozirni prednji dijelovi oka lome zrake svjetlosti projicirajući umanjenu i obrnutu sliku na fotosenzitivnu mrežnicu gdje se u specijaliziranim živčanim stanicama obavlja pretvorba u električne živčane impulse. Oko je najvažnije ljudsko osjetilo jer njime primamo 90% svih informacija iz okoline. Omogućuje svjesnu percepciju svjetla, vid, koji, među inim, omogućava razlikovanje boja i percepciju dubine. Ljudsko oko ima vidni kut od 200˚ i može razlikovati 10 milijuna nijansi boja.

Svako oko (Slika 3.1) pokreće po tri para očnih mišića: dva para ravnih, i jedan par kosih mišića. Očna jabučica je pokretljiva oko sve tri osi, poput kardanskog zgloba. Očna jabučica ima tri ovojnice. Vanjsku čine bjeličasto-poluprozirna bjeloočnica i prozirna rožnicu. Bjeloočnica daje oku stanovitu čvrstoću i oblik. Na nju se pripajaju vanjski očni mišići (druga njihova hvatišta su na stjenkama očne šupljine: ravnih i donjeg kosog na vezivnom prstenu oko optičkog živca u vrhu očne šupljine, dok se gornji kosi mišić pripaja na gornju koštanu stjenku očne šupljine).

Dublje od bjeloočnice je srednja očna ovojnica ili uvea, koju čine žilnica, šarenica i cilijarno tijelo. Ona je prokrvljena mnogim krvnim žilama. Žilnica sadrži pigment koji sprečava prodiranje svjetlosti u očnu jabučicu na bilo kom mjestu osim zjenice. Boja šarenice ovisi o količini pigmenta: što je više pigmenta, to je oko tamnije: najviše pigmenta sadrže tamnosmeđe šarenice, potom svijetlosmeđe, zelene, a najmanje pigmenta imaju plave šarenice. Albino ljudi u očima nemaju pigmenta i šarenice su im prozirne, pa nam se zbog odraza svjetlosti s krvnih žila šarenice, žilnice i mrežnice čini da su im šarenice sivkastoružičaste. Unutarnji sloj očne jabučice čini mrežnica koja ima dva dijela: optički i slijepi. Mrežnica je poluprozirna membrana sastavljena od četrdesetak vrsta živčanih stanica. Dio mrežnice odgovoran za oštrinu vida je

žutu pjegu. Žuta pjega je središnji dio mrežnice gdje su živčane stanice najgušće raspoređene. Pored žute pjege nalazi se početak vidnog živca koji je neosjetljiv na svjetlo, pa se njegova projekcija u vidnom polju naziva slijepa pjega.

Sadržaj očne jabučice čine očna vodica, leća i staklasto tijelo.



Slika 3.1 Fiziologija oka

3.2. Dinamika oka

Početnoj prilagodbi oka u stalnom, neprekinutom mraku potrebno je otprilike 4 sekunde. Potpuna prilagodba pomoću kemijske prilagodbe rožnice (Purkinjeov efekt) većinom završava u prvih 30 minuta. Stoga, dinamički omjer kontrasta od otprilike 1 000 000:1 je moguć. Proces je nelinearan, zato svaki prekid svjetlom započinje novu prilagodbu. Dobra adaptacija ovisi o dobrom protoku krvi; stoga prilagodba u mraku može biti poremećena lošom cirkulacijom, te vazokonstriktorima poput alkohola ili duhana. Oko uključuje leću ne toliko drugačiju od onih koje nalazimo u optičkim instrumentima poput kamera, i zato se ista načela mogu upotrijebiti. Zjenica oka je njegova aparatura; koju ograničava šarenica. Refrakcija u rožnici omogućuje da se aparatura (zjenični otvor) jako malo razlikuje od pravog dijametra zjenice. Ulaz zjenice ima obično promjer od 4 mm, iako mu je raspon od 2 mm u svijetloj prostoriji do 8 mm u mraku.

Vidni sustav u mozgu je prespor da bi obradio informacije ako slike prolaze preko mrežnice u više od par stupnjeva u sekundi. Stoga, da bi vidjeli kada se krećemo, mozak mora kompenzirati kretnje glavom kretnjama očiju. Još jedan od specijaliziranih dijelova optičkog sustava

životinja s naprijed smještanim očima je dio mrežnice s vrlo visokom oštrinom. Taj dio se zove pjega, i pokriva oko dva stupnja vidnog kuta u ljudi. Da bi dobili jasnu sliku, mozak mora usmjeriti oči tako da slika promatranog predmeta dođe na pjegu. Pokreti oka su stoga vrlo važni za vizualnu percepciju i bilo koje oštećenje vodi do ozbiljnih vidnih invalidnosti. Mozak mora usmjeriti oba oka precizno da bi objekt pao na obje mrežnice na isto mjesto, tj. korespondentno područje obiju mrežnica, da bi se izbjegla dvoslika. U primata (majmuna, ljudi) pokreti različitih dijelova tijela su kontrolirani mišićima koji djeluju sinkrono. Pokreti očiju su nešto različitiji, oči nisu čvrsto vezane za nešto, ali su zadržane u orbiti uz pomoć šest vanjskih očnih mišića.

3.3. Tehnike praćenja oka

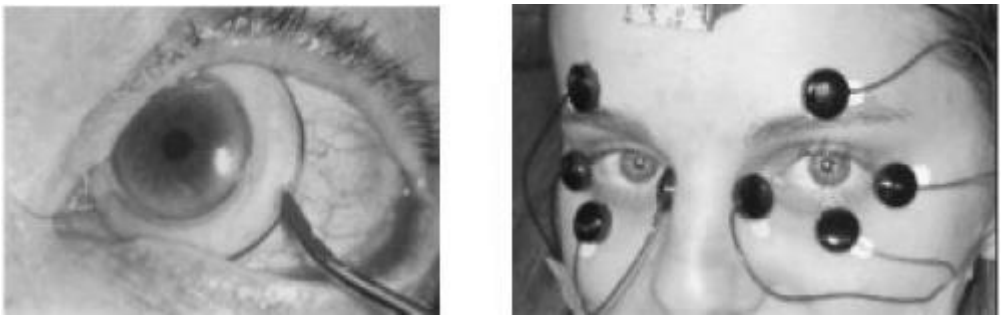
Mnoge tradicionalne tehnike praćenja pogleda su nametljive, tj. zahtijevaju opremu koja je u fizičkom kontaktu s korisnikom. Takve tehnike uključuju npr. kontaktne leće, elektrode itd. Nenametljive (daljinske) tehnike se baziraju na snimanju oka kamerom.

3.3.1. Nametljive tehnike

Nametljive tehnike (Slika 3.2), generalno su preciznije od nenametljivih (daljinskih). Jedna od najtradicionalnijih metoda je bazirana na kontaktnim lećama. Robinson [8] koristi malu zavojnicu ugrađenu u kontaktne leće koja čvrsto prianja na bjeloočnicu da bi se izbjeglo pomicanje tijekom brzog pomaka oka. Korisnikov pogled proračuna se iz mjerenja voltaže inducirane u zavojnici pomoću vanjskog elektro-magnetskog polja. Premda je sustav vrlo nametljiv, također je i vrlo točan (približno 0.08°).

Jeftinija tehnika bazira se na mjerenju potencijala kože. „Electro-oculogram - (EOG) je uobičajena tehnika snimanja pokreta očiju za bolničku primjenu zbog svoje tehničke jednostavnosti. Postavljanjem elektroda oko očiju, moguće je mjeriti male promjene potencijala kože koje se podudaraju s pokretima oka. Ova tehnika također nije prikladna za svakodnevnu uporabu, a točnost joj je oko 2°. Kamere ili neki ostali optički uređaji mogu se koristiti za mjerenje pozicije oka bez direktnog kontakta s korisnikom ali zahtijevaju da se oko nalazi u

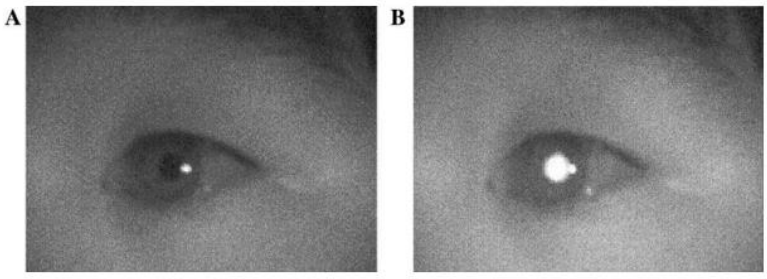
neposrednoj blizini optičkog uređaja, stoga glava mora biti fiksirana.



Slika 3.2 Primjer leće i EOG metode

3.3.2. Nenametljive tehnike

Tehnike bazirane na kamerama oslanjanju se na neka svojstva i karakteristike oka koje kamera ili neki drugi optički ili fotoosjetljivi uređaj mogu otkriti i pratiti. Uglavnom se te tehnike mogu implementirati na nenametljiv način. Za praćenje se često koriste zjenica i limbus. Limbus je granica između bjeloočnice i šarenice oka. Zahvaljujući kontrastu ta dva područja, lako se može pratiti horizontalno, ali pošto kapci uglavnom pokrivaju dio šarenice, tehnike praćenja limbusa vertikalno su dosta neprecizne. Zjenice je teže detektirati i pratiti zbog malog kontrasta na granici između zjenice i šarenice, ali tehnike praćenja zjenica su preciznije budući da one nisu prekrivene kapcima (osim za vrijeme treptaja). Kako bi poboljšali kontrast između zjenice i šarenice, mnogi koriste infracrveni (IC) izvor svjetla. Budući da IC svjetlo nije vidljivo, ono ne ometa korisnika. U praksi, većina implementacija koristi izvore svjetlosti blizu IC spektra, sa valnom duljinom oko 880 nm, što je gotovo nevidljivo ljudskom oku, ali ga svejedno može otkriti većina komercijalnih kamera. Katkad se izvor IC svjetlosti postavi blizu optičke osi kamere. Budući da kamera tada može „vidjeti“ - svjetlo koje se odbija nazad od oka, slično efektu crvenih očiju na noćnim fotografijama snimljenima jakim blicom, kamera vidi jarku zjenicu (Slika 3.3 B), umjesto normalne tamne zjenice (Slika 3.3 A).



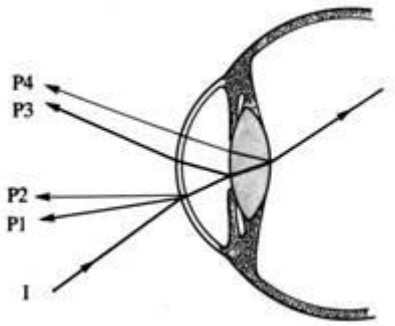
Slika 3.3 A. Normalna zjenica B Zjenica pod IC svjetlosti

Nguyen i ostali [9] proveli su eksperimente koji pokazuju ponašanje ljudskih očiju i njihov odgovor IC svjetlosti, jarkih zjenica, razmatrajući faktore koji mogu izazvati velike varijacije u odgovoru na IC svjetlost kod različitih subjekata. Svjetlosni izvor također generira odraz, odnosno odsjaj na površini rožnice, kraj zjenice.

Reulen i ostali [10] koriste varijaciju tehnika za praćenje limbusa sa IC svjetlosti zvanu infracrvena okulografija (eng. *IROG*). Postave diode koje emitiraju IC svjetlost i fototranzistore osjetljive na svjetlost u međuodnos iznad i ispod oka. Nekoliko takvih IC parova može se postaviti na naočale ili kacige, tako da je limbus uvijek osvijetljen, konkretno, nosna i sljepoočna strana limbusa. Fototranzistor transformira odraženu IC svjetlost u napon. Napon na fototranzistoru smještenom kraj nosa uspoređuje se s onim na fototranzistoru smještenom kraj sljepoočnice. Rezultirajuča razlika u naponu proporcionalna je kutnoj devijaciji oka. Svoj sustav nazvali su *IRIS eye tracker*, koji je originalno dizajniran za kliničke dijagnoze na okulomotornom živcu u ljudi. Međutim, zbog svoje velike preciznosti (do u 2 kutne minute) i propusnosti također omogućava fundamentalno proučavanje ponašanja oka, poput raznih vrsta pomicanja oka. Sustav IROG IRIS se postavlja na glavu i teži oko 300 g.

Cornsweet i Crane [11] opisuju drugi veoma precizan sustav za praćenje oka koji koristi prvu i četvrtu Purkinju sliku. Purkinje slike (Slika 3.4) su odrazi koji nastaju na različitim slojevima strukture oka. Prva Purkinja slika odgovara odrazu vanjske površine rožnice. Taj odraz je najsjajniji i najlakše ga je detektirati i pratiti. Detekcija ostalih Purkinjih slika zahtijeva posebnu opremu, ali omogućava procjenu 3D točke gledanja pomoću treće i četvrte Purkinje slike, koje odgovaraju opuštanju očne leće, kao što su opisali Crane i Steele [12]. U DPI (*eng. dual Purkinje image*) sustavu za praćenje oka, kada se oko translatira, obje slike se pomiču skupa; ali kad oko rotira, one premašuju različite udaljenosti te se razdvajaju. To razdvajanje daje mjeru kutne orijentacije oka. Autori navode impresivnu preciznost, do u 1 kutnu minutu. Umjesto korištenja eksplicitnih geometrijskih osobina poput kontura limbusa ili zjenice, alternativan pristup procjene položaja objekta je tretiranje slike kao točke u

višedimenzionalnom prostoru. Tehnike koje koriste ovu metodu bazirane su na izgledu, odnosno perspektivi.



Slika 3.4 Purkinje slike

Tan i ostali [13] koriste 252 slike od tri korisnika kako bi konstruirali metodu za procjenu pogleda, uz preciznost do u 0.5°.

Ni metoda koju predlažu Baluja i Pomerleau [14] ne koristi eksplicitne geometrijske osobine. Oni opisuju sustav za praćenje oka baziran na umjetnim neuronskim mrežama ANN *(eng. artificial neural networks*). Kad se jednom detektira oko, slika oka se izreže i koristi kao input za ANN. Uzimaju se slike za uvježbavanje sustava kad korisnik gleda u specifičnu točku na zaslonu. U svojim eksperimentima koristili su 2000 testnih uzoraka. Njihov prototip sustava za praćenje oka radi na 15Hz, a precizan je do u 2° te djelomično omogućava pomicanje glave.

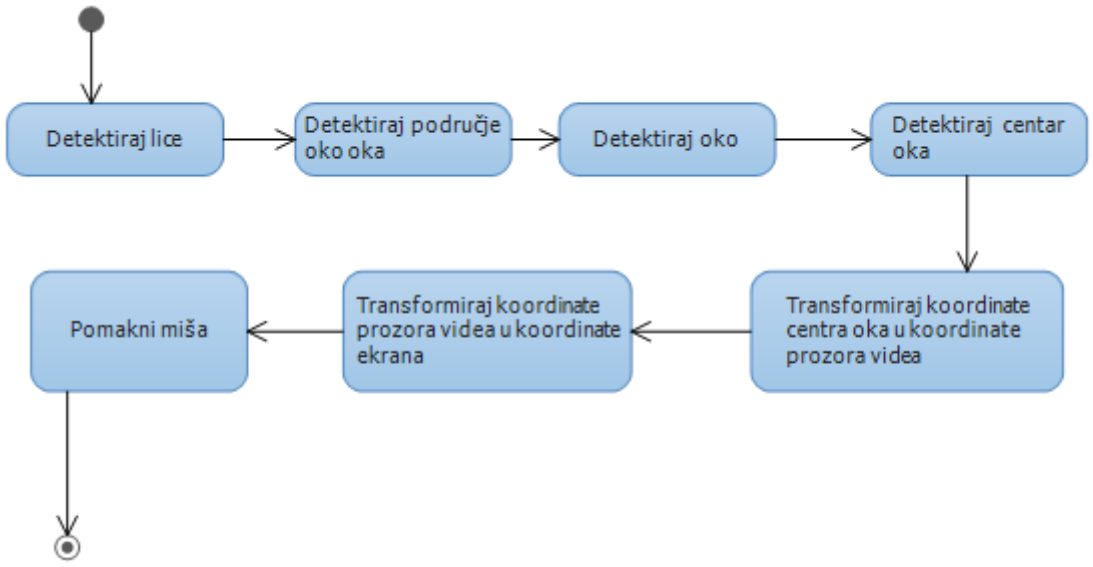
4. Opis razvijenog rješenja i implementacija

4.1. Opis problema

U ovom radu, osim same detekcije pogleda cilj je bio i upravljanje računalom pomoću obične web kamere. Iako detekcija pogleda, ima budućnost korištenja u mnogim područjima, najčešće je to upravljanje računalom. Pod upravljanje računalom se podrazumijeva kontrola perifernih jedinica poput miša. Potrebno je odrediti u kojem smjeru je pogled usmjeren te pomicati pokazivač miša.

4.2. Razrada rješenja problema

Rješavanju problema je pristupljeno analitičkim pristupom. Razlog za to je što su upravljanje pokazivačem miša i detekcija pogleda dva odvojena problema koja se ne mogu gledati u cjelini. Sama razrada je podijeljena na manje probleme (Slika 4.1), koji su razvijeni svaki posebno.



Slika 4.1 Razrada rješenja

4.2.1. Detekcija lica

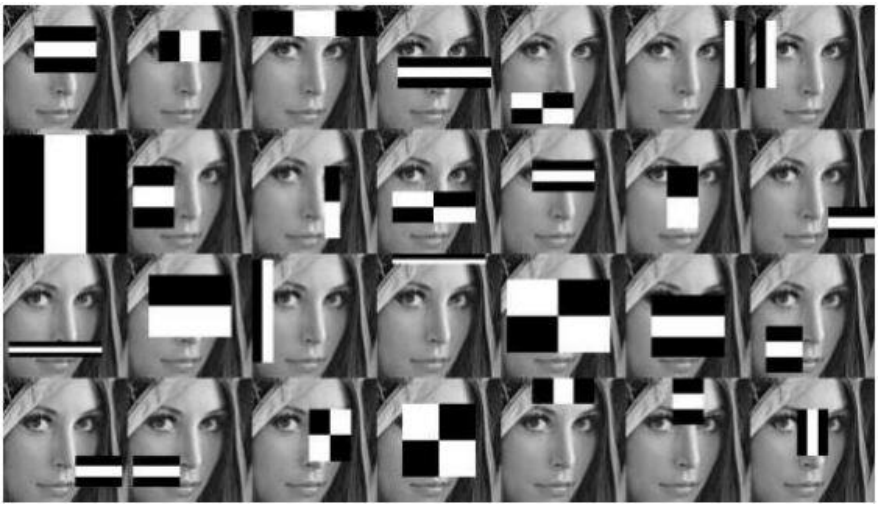
Iz ulaznih podataka prvi korak je bio prepoznavanje lica. Za prepoznavanje lica iz videa je korišten najpoznatiji algoritam za detekciju lica koji su razvili Paul Viola i Michael Jones. Chellappa i ostali [15] su predstavili pristup strojnog učenja za detekciju objekata prema učenju jakog klasifikatora kroz opterećenu kombinaciju nekoliko slabih učenja. Algoritam koji uči, temeljen je na Adaboost-u, odabire mali broj kritičkih vizualnih svojstava (Slika 4.2) koji pružaju najbolju preciznost kod klasifikacije.

Slika 4.2 Primjer tipične detekcije lica prema Chellappa-u

Ova procedura za detekciju lica, kako objašnjavaju Viola i Jones [16], klasificira slike na temelju vrijednosti nekoliko jednostavnih svojstava. Postoji više razloga za upotrebu svojstava umjesto direktne upotrebe piksela. Najčešći razlog je da se svojstva mogu praviti da kodiraju znanje u ovu svrhu, koje je teško inače naučiti upotrebljavajući konačnu količinu podataka za učenje. Za ovaj sustav upotrebe svojstava također postoji i drugi kritični razlog. Sustavi koji su temeljeni na svojstvima operiraju puno brže nego sustavi temeljeni na pikselima, a samim time su i jeftiniji.

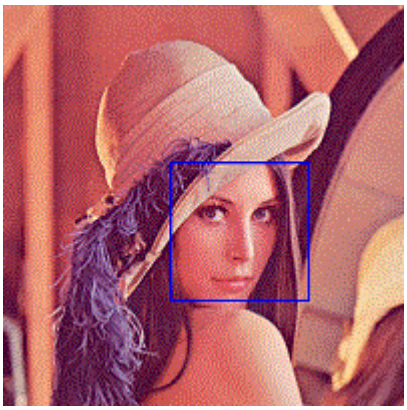
Ta jednostavna svojstva su dio tzv. Haar-bazirane funkcije (algoritam koji se bazira na pojavi - u usporedbi s algoritmima koji su bazirani na učenju. Dakle to su XML dosjei koji definiraju kaskadni set svojstava (Haar-svojstva) na digitalnim slikama upotrebljavane za prepoznavanje objekata. Viola i Jones su adaptirali ovu ideju i nazvali ta svojstva Haar-like svojstva. Slika 4.3 prikazuje primjer usporenog izračuna Haar-like svojstva prema Viola – Jones algoritmu, koji je

i korišten u ovom radu. Moraju postojati barem 3 preklapajuća crna kvadrata preko lica da bi se dobila 99% točnost da je na slici lice [17].



Slika 4.3 Usporeni izračun Haar-like svojstava

Za video, kao i za sliku, ovaj se algoritam pokazao jako dobar. Slika 4.4 prikazuje rezultat algoritma.



Slika 4.4 Regija lica

4.2.2. Određivanje regije oka

Kada je algoritam pronašao lice, potrebno je odrediti regiju oka (Slika 4.5) iz koje će se poslije prepoznavati samo oko. Odabrana je regija oko samo jednog oka zbog detekcije oka, koje je objašnjeno u sljedećem poglavlju. Regiju u kojoj se poslije tražilo oko nije bilo problem detektirati jer se ranije dobila točno određena regija lica.

Slika 4.5 Regija oko oka

4.2.3. Detekcija oka

Iz regije oko oka određuje se pozicija oka. U radu je uzeto samo jedno oko jer je traženje oba oka i njihovih centara puno zahtjevnije i nepotrebno za zadani problem, pa se pomicanje pokazivača miša odvijalo primjetno sporije. Sama detekcija oba oka i u videu i slikama radi dobro.

Za određivanje pozicije oka korišten je sličan algoritam kao i kod detekcije lica. Haar-svojstva za oči su se pokazale jednako dobre kao i za lice. Slika 4.6 prikazuje detektirano oko.

Slika 4.6 Detektirano oko

4.2.4. Detekcija centra oka i čitanje koordinata

Iz regije oka se određuje centar oka. Centar oka je centar same regije koju smo dobili prijašnjom detekcijom. Koordinate regije se spremaju u liste i pomoću njih se traži centar oka. Za rješenje ovog problema su testirane razne metode, detektori rubova (Harris, Canny), sjenčenja. Na slikama su neka od tih rješenja i davali bolje rezultate, ali ovakav pristup se pokazao najbolji za konkretan problem.

Cilj je bio pomicati pokazivač miša pomoću oka, ali opet uz neke lagane pokrete glavom da bi to korisniku bilo što prirodnije ponašanje. Korisnik sam instinktivno okrene glavu za pokazivačem miša. Kada se sve to uzme u obzir oko koordinata centra oka su postavljene 4 točke (Slika 4.7) do kojih oči i glava mogu doći. Prve očitane koordinate određuju mjesto odakle se upravlja pokazivačem miša. Postupak se ponavlja za svaki pomak pomicanja oka. Detekcija nam daje točne koordinate (x,y) centra oka.

Slika 4.7 Centar oka i 4 točke

4.2.5. Prebacivanje koordinata na prozor videa

Kako oko ima ograničeno pomicanje treba prebaciti koordinate na prozor videa u kojem se sve to odvija. Znači da treba prebaciti koordinate iz jednog koordinatnog sustava u drugi.

$$X1 = k1X * maliX - k1X * MsredX + VsredX \quad (1)$$

$$Y1 = k1Y * maliY - k1Y * MsredY + VsredY \quad (2)$$

Gdje su $(maliX, maliY)$ koordinate koje se dobiju očitavanjem centra oka, $(MsredX, MsredY)$ koordinate centra zamišljenog koordinatnog sustava oko oka koje se očitaju pri uključivanju kamere, $(VsredX, VsredY)$ koordinate centra prozora u kojem se prikazuje video. $k1X$ i $k1Y$ su omjeri dužina koordinatnog sustava oko oka i veličine prozora kako po x-u, tako i po y-u. Konačno $(X1, Y1)$ su koordinate nove točke u prozoru videa.

4.2.6. Prebacivanje koordinata na ekran

Do sada se sve odvijalo u prozoru u kojem se odvija video. Iako se u pravilu tu odvija sve što se tiče prikaza. Naravno da nema smisla da se pokazivač miša pomiče samo u tako ograničenim koordinatama. Zato je potrebno prebaciti koordinate prozora videa u koordinate stvarnog ekrana.

Koordinate ekrana se razlikuju pa je trebalo obratiti i pažnju da program radi na svakom ekranu, a ne samo iste veličine kao i na kojem je rađen sami program. Za veličinu ekrana je korišteno `GetSystemMetrics(0)` za x koordinatu i `GetSystemMetrics(1)` za y koordinatu.

Samo prebacivanje je slično kao i prethodno prebacivanje u prozor videa.

$$X = kX * X1 - kX * VsredX + EsredX \quad (3)$$

$$Y = kY * Y1 - kY * VsredY + EsredY \quad (4)$$

Gdje su $(X1, Y1)$ koordinate primljene iz ranijeg prebacivanja, $(EsredX, EsredY)$ sredina ekrana, kX i kY omjeri između stvarnog ekrana i prozora videa. I konačno (X, Y) stvarne koordinate za pokazivač miša.

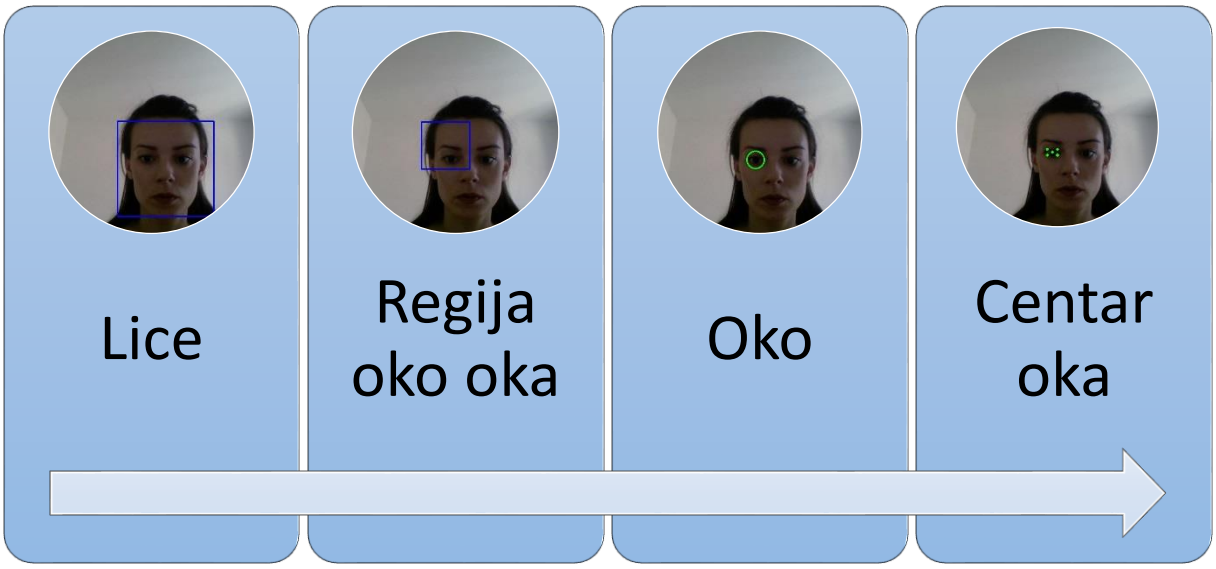
4.2.7. Postavljanje pokazivača miša

Postaviti pokazivač miša nije bio problem nakon što smo izračunali točne koordinate na kojem se on treba nalaziti. Postoje jako elegantni paketi za pokazivač, ja sam koristila `SetCursorPos()` koji se pokazao odličan izbor. Samo je trebalo voditi računa o tome da je slika na web kameri zrcalna te da se treba od same veličine ekrana oduzeti x koordinata za stvarnu poziciju po x-u.

4.3. Implementacija

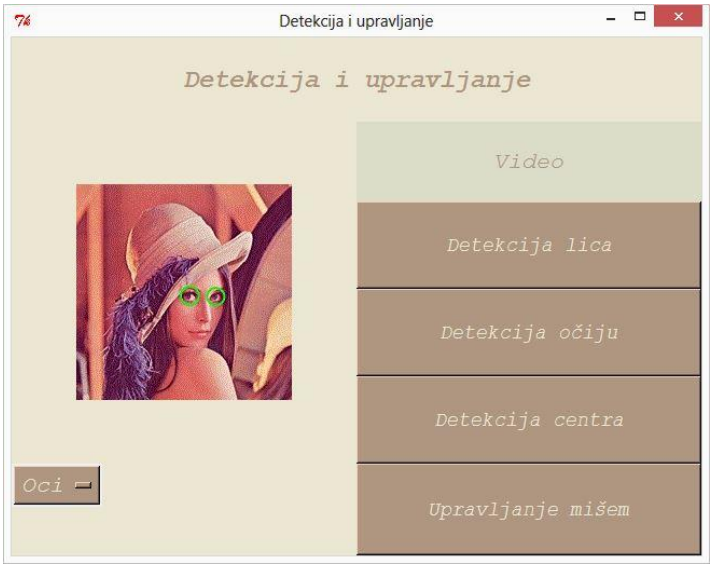
Program je implementiran u Pythonu 2.7.9. pomoću OpenCV-a. OpenCV (*eng. Open Source Computer Vision*) je biblioteka programskih funkcija uglavnom usmjerena računalnom vidu. Korišteni su i paketi iz Pythona i to: *numpy* za računske operacije (npr. popunjavanje matrice sive slike), *win32api* za koordinate ekrana i postavljanje miša na određene koordinate, *Tkinter* za grafičko sučelje i *PIL* za slike u grafičkom sučelju.

Slika 4.8 prikazuje korake do pronalaska centra oka to jest primjena algoritma koji je opisan u poglavlju 4.2.



Slika 4.8 Postupak traženja centra oka

Sučelje (Slika 4.9) na jednostavan način prikazuje određene korake kroz razradu problema. S opcijama na slici za brže pregledavanje prepoznavanja lica, očiju, centara očiju. I kroz videa za svaki od tih manjih problema. Na taj način se mogu i usporediti algoritmi kako rade na zadanoj slici ili uživo na videu, do na kraju samog upravljanja pokazivačem miša.



Slika 4.9 Sučelje

Zaključak

Upravljanje računalima i elektroničkim uređajima već dugo godina nije samo ograničeno na miša i tipkovnicu. U posebnim slučajevima, oko se pokazuje odličan izbor.

Mnogi nastoje razviti algoritme sa što većom preciznošću. Do sada sigurna rješenja za ovaj problem su se razvila samo uz odgovarajuće IC svjetlo ili određene alate. Takvi alati su skupi i nisu dostupni svima. Iako u zadnje vrijeme svi uređaji od mobitela, tableta, prijenosnih računala, imaju u sebi već ugrađenu jednostavnu web kameru, do pravog rješenja nije došlo.

U ovome radu razvijeni sustav omogućuje tek primitivnu primjenu. Iako krajnji rezultat nije odličan, daje temelj za nastavak i poboljšanje u mnogim pogledima. U budućim radovima se može uzeti u obzir udaljenost od ekrana, uz centar oka tražiti i centar šarenice pa uspoređivati udaljenost između ta dva parametra, smanjiti ili povećati pokrete glave, da korisnik po sebi može odrediti što mu najviše odgovara, nekim pokretima oka (npr. treptajem) kliktati mišem.

Svakodnevno se sve više naginjemo nad mišem i tipkovnicom ovakav dobro razvijeni sustav bi uvelike pomogao korisnicima u tom pogledu. Najvažnija primjena je naravno za osobe sa invaliditetom koje su pokretne samo u dijelu glave ili očiju ovakav sustav bi im omogućio korištenje samog računala. Iako je to nama u današnje vrijeme svakodnevnica njima bi donio jednu sasvim novu dimenziju u životu.

Literatura

[1] M. H. Yang i N. Ahuja, Detecting Faces in Images: A Survey IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No.1., 2002.

[2] Z. Orman, A. Battal i E. Kemer, »A STUDY ON FACE, EYE DETECTION AND GAZE,« *International Journal of Computer Science & Engineering Survey,* 2011.

[3] K. A., B. E. i B. W., »Why do we look at people's eyes?,« *Journal od Eye Movement Research.*

[4] S. L. i J. Robert, u *Evaluation of Eye Gaze Interaction*, pp. 281-288.

[5] J. R.E., P. Fitts i J. Milton., Eye movements of aircraft pilots during instrument landing approaches, 1950..

[6] Thompson., H. Hartridge i L.C., »Methods of investigating eye movements,« *British Journal of Ophthalmology*, p. 581–591, 1948..

[7] A. L. Yarbus, »Eye Movements and Vision,« *Plenum Press,* 1967.

[8] D. Robinson, u *A method of measuring eye movements using a scleral search coil in a magnetic field* , 1963, p. 137–145.

[9] K. Nguyen, C. Wagner, D. Koons i M. Flickner, Differences in the infrared bright pupil response of human eyes.

[10] J. Reulen, j. Marcus, D. Koops, F. d. Vries, G. Tiesinga, K. Boshuizen i J. Bos, *Precise recording of eye movement: the iris technique,* p. 20–26, 1988.

[11] T. Cornsweet i H. Crane, u *Accurate two-dimensional eye tracker using first and fourth purkinje images*, 1973, p. 921–928.

[12] H. Crane i C. Steele, u *Accurate three-dimensional eyetracker* , 1978, p. 691–705.

[13] K. Tan, D. Kriegman i H. Ahuja, »Appearance based eye gaze estimation,« *Workshop on Applications of Computer Vision—WACV02*, p. 191–195, 2002.

[14] S. Baluja i D. Pomerleau, »Non-intrusive gaze tracking using artificial neural networks,« 1994.

[15] R. Chellappa, P. Sinha i P. Phillips, »Face Recognition by Computers and Humans,« pp. 46-55, 2010.

[16] P. Viola i M. Jones, »Robust Real-Time Face Detection,« *International Journal of Kluwer Academic Publishers,* pp. 137-154, 2004.

[17] »http://docs.opencv.org/,« [Mrežno].

Popis slika

Slika 1.1 Tipična shema prepoznavanja objekta u digitalnoj obradi slika 5
Slika 1.2 Trodimenzionalno prepoznavanje lica 7
Slika 2.1 Eye-tracker 11
Slika 3.1 Fiziologija oka 14
Slika 3.2 Primjer leće i EOG metode 16
Slika 3.3 A. Normalna zjenica B Zjenica pod IC svjetlosti 16
Slika 3.4 Purkinje slike 18
Slika 4.1 Razrada rješenja 19
Slika 4.2 Primjer tipične detekcije lica prema Chellappa-u 20
Slika 4.3 Usporeni izračun Haar-like svojstava 21
Slika 4.4 Regija lica 21
Slika 4.5 Regija oko oka 22
Slika 4.6 Detektirano oko 22
Slika 4.7 Centar oka i 4 točke 23
Slika 4.8 Postupak traženja centra oka 25
Slika 4.9 Sučelje 26

Sažetak

Detekcija usmjerenosti pogleda i upravljanje računalom primjenom računalnog vida

Računalni vid je područje koje obuhvaća metode za stjecanje, obrade, analiziranja i razumijevanja slike i, općenito, višedimenzionalnih podataka iz realnog svijeta u cilju dobivanja numeričkih ili simboličkih informacija. Klasični problem računalnog vida, obrade slika, i strojnog vida je utvrđivanje sadrži li slika određeni objekt, značajku ili aktivnost. Prepoznavanje lica je jedan od najizazovnijih problema računalnog vida. Detekcija oka i pogleda ima mnoge primjene u svakodnevnom životu. Zato je cilj bio razviti jednostavan sustav temeljen na analitičkim tehnikama za upravljanje pokazivačem miša pomoću obične web kamere.

Ključne riječi: detekcija oka, praćenje oka, detekcija pogleda, upravljanjem kursorom miša

Summary

Eye-gaze detection and computer control using computer vision

Computer vision is the area that includes methods for acquiring, processing, analyzing and understanding the images and, in general, multidimensional data from the real world in order to obtain a numerical or symbolic information. The classic problem of computer vision, image processing and machine vision is to determine if the image contains a specific object, feature or activity. Face recognition is one of the most challenging problems of computer vision. Eye detection and eye-gaze can be used in a lot of different occasions in everyday life. Therefore, the goal was to develop a simple system based on analytical techniques for mouse control using an ordinary Web camera.

Keywords: eye detection, eye tracking, eye-gaze detection, mouse control

Skraćenice

HCI	*Human Computer Interaction*	Interakcija čovjeka i računala
RSI	*Repetitive Strain Injury*	
EOG	*Electrooculography*	
IC(IR)	*Infrared*	Infracrveno
IROG	*InfraredOculography*	
ANN	*Artificial Neural Network*	Umjetna neuronska mreža

Privitak

Instalacija programske podrške

Uz Python 2.7.9. treba instalirati sve potrebne pakete koje sam navela u poglavlju 4.3. *OpenCv, numpy, win32api. Tkinter i PIL* su paketi uključeni u Python 2.7.9. treba ih samo pozvati.

Kod u Pythonu

Sučelje

```
# -*- coding: utf-8 -*-

from Tkinter import *

from PIL import ImageTk, Image

import cv2

import numpy as np

import win32api, win32con, win32gui

from win32api import GetSystemMetrics

def callback():

    return execfile("Upravljanje_Misem.py")

def callback1():

    return execfile("Detekcija_lica.py")

def callback2():

    return execfile("Detekcija_ociju.py")

def callback3():

    return execfile("Detekcija_centra.py")


def change_image(*args):

    # Change image of label accordingly

    temp = -1

    if (studFiles.get() == "Slika"):

        temp = 0

    elif (studFiles.get() == "Lice"):
```

```
        temp = 1
    elif (studFiles.get() == "Oci"):
        temp = 2
    elif (studFiles.get() == "Centar Ociju"):
        temp = 3
    label.config(image=photos[temp])
#kreiranje praznog prozora
root = Tk()
root.wm_title("Detekcija i upravljanje")

content = Frame(root)
frame = Frame(content, borderwidth=2,  width=640, height=480, bg="#ECE7D3")
content.grid(column=0, row=0)
frame.grid(column=0, row=0, columnspan=2, rowspan=6)

label1 = Label(content, text="Detekcija i upravljanje", bg="#ECE7D3",
fg="#AF9680")
label1.config(font=("Courier", 18, "italic", "bold"))
label1.grid(columnspan=2, row=0, sticky=W+E+N+S)

studFiles = StringVar()
studFiles.set('Opcije')
files = ['Slika', 'Lice', 'Oci', 'Centar ociju']
studDropDown = OptionMenu(content, studFiles, *files)
studDropDown.config(font=("Courier",    16,    "italic"),    fg="#ECE7D3",
bg="#AF9680")
studDropDown["menu"].config(font=("Courier", 16, "italic"), fg="#ECE7D3",
bg="#AF9680")
studDropDown.grid(row=5, sticky=W+N)
```

```
studFiles.trace("w", change_image)


photos     =(PhotoImage(file='lena1.gif'),     PhotoImage(file='lice1.gif'),
PhotoImage(file='oci1.gif'), PhotoImage(file='centar1.gif'))

label = Label(content, image=photos[0], bg="#ECE7D3")

label.grid(row=1, rowspan=4, column=0, sticky=W+E+N+S)


label2 = Label(content, text="Video", bg="#DBDDC8", fg="#AF9680")

label2.config(font=("Courier", 16, "italic"))

label2.grid(column=1, row=1, sticky=W+E+N+S)


b1  =  Button(content,  text="Detekcija  lica",  fg="#ECE7D3",  bg="#AF9680",
command=callback1)

b1.config(font=("Courier", 14, "italic"))

b1.grid(row=2,column=1, sticky=W+E+N+S)

b2  =  Button(content,  text="Detekcija  očiju",   fg="#ECE7D3",  bg="#AF9680",
command=callback2)

b2.config(font=("Courier", 14, "italic"))

b2.grid(row=3,column=1, sticky=W+E+N+S)

b3  =  Button(content,  text="Detekcija  centra",  fg="#ECE7D3",  bg="#AF9680",
command=callback3)

b3.config(font=("Courier", 14, "italic"))

b3.grid(row=4,column=1, sticky=W+E+N+S )

b4  =  Button(content,  text="Upravljanje  mišem",  fg="#ECE7D3",  bg="#AF9680",
command=callback)

b4.config(font=("Courier", 14, "italic"))

b4.grid(row=5,column=1, sticky=W+E+N+S)

#da uvijek stoji jer inače se samo pojavi i zatvori da i ne vidimo

root.mainloop()
```

Upravljanje mišem

```
# -*- coding: utf-8 -*-

import cv2

import numpy as np

import win32api, win32con, win32gui

from win32api import GetSystemMetrics


cascPath1="C:\\opencv\sources\data\\haarcascades\\haarcascade_frontalface_a
lt.xml"

face_cascade = cv2.CascadeClassifier(cascPath1)

cascPath2 = "C:\\opencv\sources\data\\haarcascades\\haarcascade_eye.xml"

eye_cascade=cv2.CascadeClassifier(cascPath2)

vid = cv2.VideoCapture(0)

xc=[]

yc=[]

def click(x,y):

          win32api.SetCursorPos((x,y))


def pomak(malix,maliy,sredx,sredy):

    #pomicanje u framu radeno po omjeru dužina!

    omjer1X=640/24

    omjer1Y=480/14

    X=omjer1X*malix-omjer1X*sredx+(640/2)

    Y=omjer1Y*maliy-omjer1Y*sredy+(480/2)

    X=int(X)

    Y=int(Y)

    #pomicanje za cijeli ekran

    omjer2X=GetSystemMetrics(0)/640
```

```
    omjer2Y=GetSystemMetrics(1)/480
    praviX=int(omjer2X*X-omjer2X*(640/2)+(GetSystemMetrics(0)/2))
    praviY=int(omjer2Y*Y-omjer2Y*(480/2)+(GetSystemMetrics(1)/2))
    return(praviX,praviY)

while True:
    flag, frame = vid.read()

    if flag:
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, 1.3, 5)
        eyes = None
        # Nađi lice
        for (x, y, w, h) in faces:
            roi_gray = np.zeros(gray.shape, np.uint8)
            #iz regije oko oka uzime samo jedno oko
            roi_gray[y:y+h/2, x:x+w/2] = gray[y:y+h/2, x:x+w/2]
            eyes = eye_cascade.detectMultiScale(roi_gray)
            for (ex, ey, ew, eh) in eyes:
               #xcenter i ycenter su trenutne koordinate jednog oka kojeg
gledamo
                xcenter=(ex+(ex+ew))/2
                ycenter=(ey+(ey+eh))/2
                xc.append(xcenter)
                yc.append(ycenter)
                (x,y)=(xc[0],yc[0])
```

```
    (x,y)=pomak(xcenter,ycenter,x,y)
    try:
        click(GetSystemMetrics(0)-x,y)
    except:
        click(GetSystemMetrics(0)/2,GetSystemMetrics(1)/2)
    cv2.imshow('Video', frame)


    if cv2.waitKey(5) == 27:
        break
vid.release()
cv2.destroyAllWindows()
```