

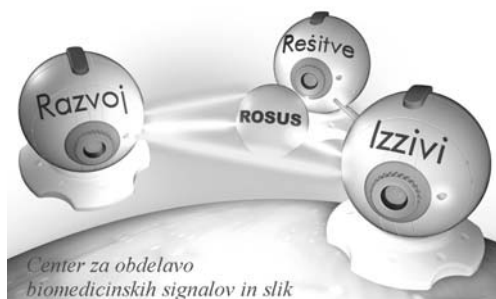


Univerza v Mariboru

*Fakulteta za elektrotehniko,
računalništvo in informatiko*



Inštitut za
računalništvo



ROSUS 2010
Računalniška obdelava slik in njena
uporaba v Sloveniji 2010
Zbornik 5. strokovne konference

Urednik
Božidar Potočnik

Maribor, 18. marec 2010

Zbornik prispevkov strokovne konference
ROSUS 2010 Računalniška obdelava slik in njena uporaba v Sloveniji 2010
Maribor, 18. marec 2010
Elektronska izdaja

CIP - Kataložni zapis o publikaciji
Univerzitetna knjižnica Maribor

004.93 (082)

STROKOVNA konferenca ROSUS (5 ; 2010 ; Maribor)

ROSUS 2010 [Elektronski vir] : računalniška obdelava slik in njena uporaba v Sloveniji 2010 : zbornik 5. strokovne konference, Maribor, 18. marec 2010 / urednik Božidar Potočnik. – El. zbornik. – Maribor : Fakulteta za elektrotehniko, računalništvo in informatiko, Inštitut za računalništvo, 2010

Dostopno tudi na :
<http://cobsis.uni-mb.si/rosus2010/>

ISBN 978-961-248-206-0 (CD-ROM)

1. Potočnik, Božidar
COBISS.SI-ID 64471553

ISBN 978-961-248-206-0



Programski odbor ROSUS 2010:

dr. Božidar Potočnik
dr. Damjan Zazula
dr. Boris Cigale
dr. Viljem Žumer
dr. Baldomir Zajc
Tomaž Tomažič
dr. Bruno Stiglic
dr. Konrad Steblovnik
dr. Slobodan Ribarič
dr. Veljko Vlasisavljevič

dr. Saša Divjak
dr. Stanislav Kovačič
dr. Zdravko Kačič
dr. Aleš Leonardis
dr. Janez Perš
dr. Mitja Lenič
dr. Franc Solina
dr. Andrej Šoštarich
dr. Franjo Pernuš
dr. Eldar Gadžijev

Organizacijski odbor 2010:

dr. Božidar Potočnik
dr. Damjan Zazula
dr. Danilo Korže
dr. Boris Cigale
Jurij Munda
Smiljan Šinjur
Vojko Glaser
Rok Istenič
Sebastijan Šprager

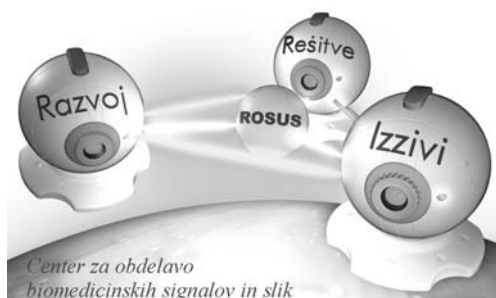
Spoštovani!

Po štirih uspešnih konferencah ROSUS 2006–2009 želimo s konferenco ROSUS 2010 spet promovirati pomembnost ekonomske koristnosti računalniške obdelave slik na področjih industrije, biomedicine in drugih poslovnih procesov. Vezi, ki smo jih na prejšnjih konferencah stkali med raziskovalci, razvijalci, ponudniki rešitev ter uporabniki računalniške obdelave slik v slovenskem prostoru, želimo še dodatno okrepiti, ob tem pa nuditi tudi možnosti sklepanja novih sodelovanj in svetovanja pri razreševanju konkretnih poslovnih oziroma raziskovalnih problemov.

Tudi letos namenjamo glavni poudarek aplikacijam s področja računalniške obdelave slik, ki so že integrirane oziroma pripravljene za integracijo v poslovne procese. S tem želimo ne le demonstrirati, da takšna avtomatska obdelava lahko zaradi svoje natančnosti in hitrosti prinaša velike ekonomske koristi, ampak tudi dokazati, da aplikacije računalniške obdelave slik nudijo nove priložnosti za uveljavitev na trgu visokih tehnologij. Seveda ne smemo pozabiti na možnost diskusije ter predstavitev konkretnih problemov in potreb, ki se porajajo pri uporabnikih, s katerimi bomo računalniško obdelavo slik in njeno koristnost še bolj približali avditoriju.

Naj sklenemo uvodne misli s pristrčno zahvalo Slovenskemu podjetniškemu skladu, ki je v okviru konference ROSUS 2010 predstavil zanimive finančne instrumente za spodbujanje prenosa tehnoloških rešitev v podjetniško sfero.

dr. Božidar Potočnik
predsednik konference
ROSUS 2010



POKROVITELJI



ROSUS 2010
<http://cobsis.uni-mb.si/rosus2010/>

KAZALO

VABLJENA PREDAVANJA.....	1
Janez Primc, Emil Jerman, Mitja Zabavnik, Technix d.o.o., Mengeš <i>Avtomatska detekcija prometa v slovenskih predorih.....</i>	3
Franjo Pernuš, FE, Ljubljana <i>Poravnava slik v slikovno vodenih medicinskih postopkih</i>	24
Sergej Rinc, Miha Gaberšek, spletin.net d.o.o., Škofja vas <i>Možnosti obdelave slik v aplikacijah za mobilne telefone</i>	25
INDUSTRIJSKE APLIKACIJE	35
Domen Mongus, Damjan Zazula, FERI, Maribor <i>Vpliv učne množice na uspešnost razpoznavanja obrazov</i>	37
Peter Podbreznik, Božidar Potočnik, FG, FERI, Maribor <i>Identifikacija in korekcija napačno vzpostavljenih korespondenčnih točk za velike odmike dveh kamer</i>	47
Matjaž Kosec, Dejan Tomažević, Franjo Pernuš, Boštjan Likar, Sensum d.o.o., FE, Ljubljana <i>Analiza NIR hiperspektralnih slik.....</i>	53
Jurij Rakun, Damjan Zazula, FK BV, FERI, Maribor <i>Primernost postopkov poravnave slik za določanje gosto posejanih korespondenčnih točk</i>	61
Janez Perš, Matej Kristan, Vildana Sulič, Stanislav Kovačič, FE, Ljubljana <i>Računalniški vid za vodenje avtonomnega plovila</i>	69
MEDICINSKE IN BIOMEDICINSKE APLIKACIJE.....	77
Sebastijan Šprager, Jurij Munda, Damjan Zazula, FERI, Maribor <i>Uporaba porazdeljenega procesiranja pri obdelavi ultrazvočnih volumnov jajčnikov</i>	79
Tina Klenovšek, Franc Janžekovič, Boris Kryštufek, FNM, ZRCUP, Maribor, Koper <i>Variabilnost lobanje dinarske voluharice, Dinaromys bogdanovi.....</i>	89

Peter Usenik, Tomaž Vrtovec, Franjo Pernuš, Boštjan Likar, FE, Ljubljana <i>Sledenje oblik fosfolipidnih veziklov zajetih s fazno kontrastno mikroskopijo</i>	<i>94</i>
Boris Cigale, Smiljan Šinjur, Damjan Zazula, FERI, Maribor <i>Kvantizacija in rekonstrukcija perifolikularnega ožilja v ultrazvočnih volumnih</i>	<i>101</i>
DRUGO	109
Milan Čoh, FŠ, Ljubljana <i>Registracija in objektivizacija hitrih gibanj v športu.....</i>	<i>111</i>
ŠTUDENTSKE APLIKACIJE	121
Uroš Klopčič, Peter Peer, FRI, Ljubljana <i>Sistem za verifikacijo osebe na podlagi prstnega odtisa</i>	<i>123</i>
Roman Avsec, Peter Peer, FRI, Ljubljana <i>OpenCV in iPhone: Detekcija obrazov v realnem času.....</i>	<i>131</i>

VABLJENA PREDAVANJA

AVTOMATSKA DETEKCIJA PROMETA V SLOVENSКИH PREDORIH

Janez Primc, Emil Jerman, Mitja Zabavnik

Technix d.o.o., Topole 51, 1234 Mengeš

E-pošta: info@technix.si

URL: www.technix.si

POVZETEK: Predori veljajo za ene bolj nevarnih avtocestnih odsekov. Manjša prometna nesreča lahko hitro preraste v veliko tragedijo, ko ljudje ostanejo ujeti v predoru brez možnosti za pobeg. Zato so naložbe v obvladovanje nesreč v predorih nujne, saj njihovo učinkovito upravljanje rešuje življenje ljudi. Učinkovitost takšnega upravljanja pa je odvisna predvsem od tega, kako hitro bo nesreča zaznana in kako hitro bodo operaterji ob pomoči sistemov ustrezno reagirali. Vsaka izgubljena minuta povečuje možnost nove nesreče, ki bi lahko drastično povečala število žrtev. To bi povzročilo še večjo materialno škodo in podaljšalo čas za vzpostavitev normalnega stanja. S pomočjo sistema avtomatske detekcije prometa (ADP) in ostalih sistemov, ki sodelujejo pri nadzoru in vodenju prometa, se čas za detekcijo nesreče lahko zelo zmanjša, odziv operaterja, ki bo poklical ustrezne intervencijske reševalne ekipe, bo hitrejši, možnost za nastanek nesreče in nadaljnjih nesreč pa manjša. V nadaljevanju bo predstavljen sistem avtomatske detekcije prometa, njegova zgradba, delovanje in povezava z ostalimi sistemi. Poleg tega bo predstavljeno, kje se sistemi video detekcije še pojavljajo in primer detekcijskega senzorja, ki pri 3D detekciji uporablja tehnologijo stereoskopije.

1. UVOD

Družba Technix je bila ustanovljena leta 1992. V začetnem obdobju se je ukvarjala zlasti s snovanjem in izvajanjem avtomatizacije različnih tehnoloških procesov in objektov ("na ključ").

V kasnejšem obdobju se je področje dejavnosti razširilo na nekatere segmente telekomunikacij in informatike (CTI, omrežje ethernet, tehnično-komercialna programska orodja) ter celoten investicijski inženiring (načrtovanje in vodenje investicij).

Predhodno delovanje na področju avtomatike in informatike je botrovalo tudi navezavi poslovnih stikov s firmo Velec, ki je poleg redne proizvodnje informacijskih tabel, radijskih komunikacij za TGV ter posebnih namenskih produktov za evropski satelitski program in za vojaške namene že od leta 1980 razvijala tudi analizatorje slike ter bila vključena v evropski pilotski projekt »inteligentne avtoceste« – Melisa.

Technix je leta 1995 izdelal PGD/PZR dokumentacijo za elektro-strojno opremo (ESO) predorov Barnica in Tabor (po tedanji določitvi trase) in kot priporočen varnostni sistem navedel tudi avtomatsko detekcijo prometa. V tem času je Velec komercialno že tržil svoj AID sistem VisiRout. Leta 1996 je Technix v PGD/PZI predora Tabor že predvidel avtomatsko detekcijo, kar pa niti pri revidentih niti pri investitorju ni naletelo na odobravanje. Ideja o implementaciji avtomatske detekcije v slovenskih predorih je tako skupaj s spremembo same avtocestne trase za nekaj let zamrla.

Avtomatska detekcija se je v slovenskem cestnem programu ponovno pojavila po letu 2000 in sicer v razpisu za ESO predorov Ločica in Jasovnik. Technix je tedaj nastopal kot član ponudbenega konzorcija z namenom izvedbe sistema avtomatske detekcije ter sistema nadzora in krmiljenja. Velec je bil v vmesnem obdobju prodan firmi Sagem, ki pa je kmalu opustila proizvodnjo detektorjev. Del znanja in izkušenj je prešel tudi v belgijsko podjetje Traficon, ki se je med tem razvil v vodilnega proizvajalca opreme za avtomatsko detekcijo prometa. Technix je na koncu kot podizvajalec izdelal sistem video nadzora in sistem avtomatske detekcije v predorih Ločica in Jasovnik ter leta 2002 v svojo rešitev vključil prve detektorje VIP/I (video image processor/incident).

Podjetje se je tako v zadnjem času preusmerilo zlasti in izvajanje projektov video nadzora in detekcije, predvsem za področje prometa in tehnoloških procesov; precej truda pa Technix vlaga tudi v razvoj lastnih produktov na področju video tehnologije.

1.1 Razlogi za uporabo avtomatske detekcije prometa

Video sistem je namenjen vizualni kontroli prometnega dogajanja v predorski cevi. Vid je človeku prirojen in poleg sluha predstavlja osnovni način zaznavanja informacij iz okolice. Človeški možgani se ne morejo odzivati na vse dražljaje, zato jih selekcionirajo. Proces osredotočenosti na omejeno število dražljajev se imenuje pozornost.

Pozornost opredeljujeta dve značilnosti:

- dražljaji, na katere je usmerjena pozornost, postanejo zavesti jasni, ostalim pa je dostop do zavesti preprečen;
- v vedenju nastanejo spremembe: telo je obrnjeno proti dražljaju, mišice so napete, gibov je manj.

(primer: študent, ki na izpitu prepisuje, ne opazi bližajočega se asistenta)

Pri odraslih Tahistoskop določa, da je maksimalno 10 objektov smiselnih za opazovanje. Poleg "mentalnih" sposobnosti posameznika je obseg pozornosti odvisen tudi od vrste in razporejenosti opazovanih objektov. Trajanje pozornosti je odvisno predvsem od motivacije, starosti in motilnih dražljajev, ki preusmerjajo pozornost. Pozornost je torej dinamičen proces, ki se nenehno spreminja.

Dejavniki pozornosti:

Obseg, trajanje, smer in intenzivnost pozornosti so odvisni od dveh vrst dejavnikov:

- zunanjih
- notranjih.

Med zunanje dejavnike štejemo lastnosti dražljajev:

- intenzivnost
- spreminjanje
- gibanje
- trajanje
- pogostost
- razlikovanje od okolice
- velikost.

Vrivanje dražljajev v zavest se veča z:

- intenzivnostjo (npr. močna svetloba, glasen zvok, intenziven okus)
- kontrastnostjo (npr. grenak okus med sladkimi, črno-bela fotografija med barvnimi)
- gibanjem (npr. premikajoči napisi)
- modalnostjo (npr. visoki toni, rumena in rdeča barva).

Med notranje dejavnike med drugim uvrščamo:

- potrebe in motive, ki usmerjajo pozornost (bolj smo pozorni na stvari, ki jih želimo)
- izkušnje in znanje (bolj pozorni smo na stvari, s katerimi imamo izkušnje)
- čustva (bolj pozorni smo na stvari, ki so nam čustveno pomembne).

(primer: pričakovanje prijateljice na postaji poveča pozornost, žalost pa zmanjšuje pozornost za številne dražljaje)

Če želimo torej "vidno" uporabljati za nadzor in upravljanje kritičnih procesov kot npr. prometa, potem moramo uvodoma ugotoviti pomanjkljivosti:

- delo je monotono in traja osem ur (do pred kratkim celo 12)
- koncentracija upade v povprečju dvakrat na uro
- sama video slika je kljub gibajočemu prometu za možgane statična
- barve na video sliki ne odražajo kritičnih situacij
- število monitorjev naj ne bo večje od 10
- obstaja potreba po opazovanju npr. 200 ali več kamer.

Ugotovimo torej lahko, da je vizualno spremljanje za človeka sicer naraven, vendar zaradi stotih ali več kamer v obdobju osmih ur relativno težko zaznaven proces.

Kot posledica navedenih naravnih omejitev se je pojavila potreba po preoblikovanju zunanjih zaznavnih dejavnikov oziroma procesiranju realnih vidnih dražljajev tako, da bi se lažje vrivali v zavestni del operaterjevih možganov. Tako je prišlo do ideje o **avtomatski detekciji prometa**.

1.2 Realnost uporabe avtomatske detekcije prometa

Ena od specifičnih lastnosti (cestnega in železniškega) prometa je njegova geografska razprostrtnost, zaradi česar je daljinski prenos slike edini način za zagotavljanje vidne povezave med objektom nadzora in operaterjem.

V izvedenih primerih je zajem slike na objektu »klasičen«, to je preko CCTV kamere, vgrajene v ohišje z visoko korozijsko in mehansko odpornostjo in ustrezno stopnjo IP zaščite. Kamere morajo biti nameščene tako, da kar najbolje pokrivajo nadzorovani tehnološki proces, kar v realnem okolju pomeni, da so ob normalnem obratovanju objekta, to je pod prometom, nedostopne za morebitne vzdrževalne posege. Navedeno nas sili k temu, da je neposredno v ohišju kamere nameščeno čim manj aktivne opreme, zato originalni video signal vodimo prek koaksialnega kabla do najbližje dostopne lokacije. V primeru predorov je to niša za klic v sili, kjer se na optičnih oddajnikih izvede pretvorba in prenos po optičnem vlaknu do pogonske centrale. V pogonski centrali se video signal rekonvertira v izvorno obliko in razdeli med dva osnovna sistema, to je nadzor in detekcijo.

Za potrebe sistema video nadzora prometa (VNP), to je prikazovanje in krmiljenje živega videa, alarmnega videa in arhivskih posnetkov, z »video zajemalniki« pretvorimo video signal v digitaliziran format in izvedemo kompresijo ter prehod na regionalno ethernet tehnološko omrežje predorov, na katerega je povezan tudi pristojni regionalni center vodenja z ustrežno opremo in osebjem za neprekinjeno dežurno vodenje podrejenih predorov in trase.

Za potrebe sistema ADP, to je kontinuirano samodejno spremljanje in analiza prometa ter posledično sporočanje izrednih prometnih dogodkov, se video signal vodi na tako imenovane detektorje, to je namenske video-detekcijske procesne platforme.

Vsaki kameri je namenjen svoj detektor, ki vhodni analogni video signal najprej digitalizira in sfiltrira, potem pa s pomočjo verige kompleksnih matematičnih transformacij ugotavlja odstopanja od vzorca normalnega prometnega toka. Zaznana stanja se na osnovi TCP/IP komunikacije prenašajo na regionalno ethernet tehnološko omrežje predorov, s katerim je povezan tudi pristojni RCV z ustrežno opremo in osebjem za neprekinjeno dežurno vodenje podrejenih predorov in trase.

Zaradi varnostnih razlogov se zajem in shranjevanje podatkov običajno izvaja na dveh ADP strežnikih, ki delujeta paralelno, eden od njiju pa v primeru izpada dela mreže ali same strojne platforme služi kot aktivna rezerva. Tudi aplikativni program AdpTec se običajno izvaja na dveh identičnih uporabniških delovnih postajah. Če pride do

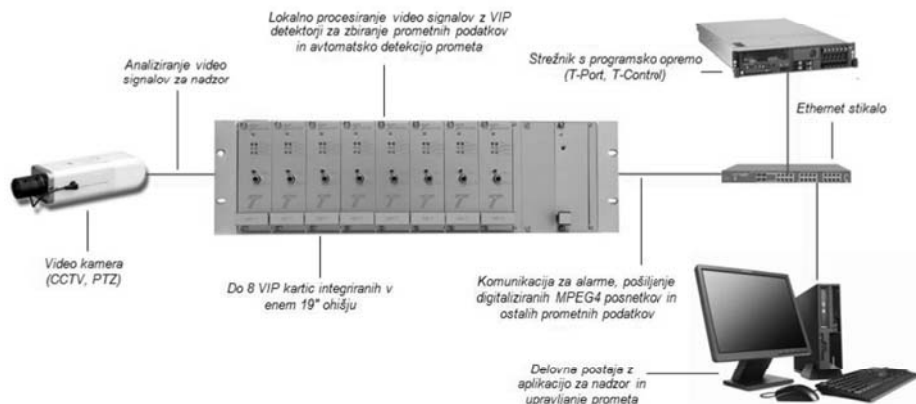
kakršnekoli napake na komunikaciji ali strojni platformi glavnega delovnega mesta, se operater preseli na pomožno lokacijo. Sistem omogoča pregledovanje in filtriranje shranjenih podatkov po različnih merilih ter seveda prenos alarmov na tretje sisteme, ki neposredno krmilijo prometno opremo v predoru ali na trasi.

2. SISTEM AVTOMATSKE DETEKCIJE PROMETA (ADP)

Sistem je namenjen spremljanju prometa in alarmnih stanj v predorih ali prometnih križiščih. Detekcija zaznava ustavljena vozila, nasproti vozeča vozila, pešce, spremembe hitrosti in zadimljenost na vozišču.

2.1 Splošen opis delovanja celotnega sistema

Promet nadzorujejo analogne video kamere, ki so nameščene v predorih. Video signal najprej vodimo prek koaksialnih kablov do predorskih niš, kjer ga digitaliziramo in prek optične prenosne opreme vodimo do pogonskih central pred predori. V pogonskih centralah ga pretvorimo nazaj v analognega ter ga vodimo na detektorske kartice (VIP-T). Vsak VIP-T ima tudi video izhod, od koder video signale vodimo naprej na video kodirne naprave. Vse naprave (VIP-T detektorji, video kodirniki, snemalniki, delovne postaje, strežniki itd.) so povezane na več ethernet stikal, ki so med seboj povezana v topologiji obroča. Za komunikacijo med detektorskimi karticami in delovnimi postajami ter za periodično kopiranje vseh alarmov in podatkov nam služita dva redundančna strežnika. V nadzornem centru se nahaja delovna postaja, na kateri teče aplikacija za avtomatsko detekcijo prometa. Običajno aplikacija teče še na eni delovni postaji v pomožnem centru (pogonski centrali), ki služi za redundanco v primeru izpada glavnega nadzornega centra. Aplikacija omogoča grafični prikaz alarmov za vsak odsek, ki ga pokriva določena kamera. Shranjene podatke lahko pregledujemo, razvrščamo in združujemo po različnih kriterijih.



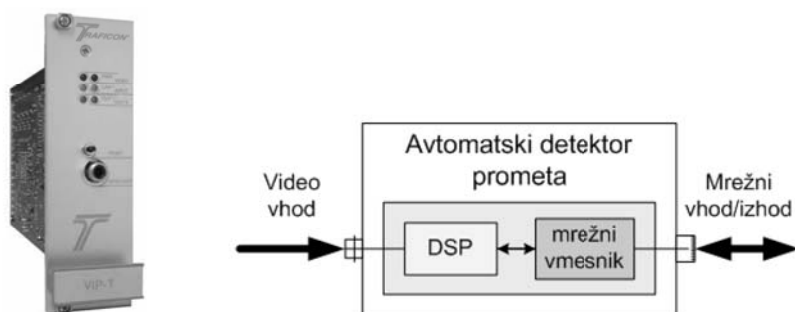
Slika 1: Blok shema ADP sistema

2.2 Opis delovanja pomembnejših gradnikov

2.2.1 VIP-T detektor

VIP-T detektor je računalnik, ki s pomočjo različnih kompleksnih algoritmov analizira sliko v realnem času in ugotavlja dogodke, kot so ustavljeno vozilo, vožnja v nasprotno smer, detekcija pešcev, nihanje hitrosti vožnje, detekcija dima, izguba video slike itd. VIP-T združuje avtomatsko zaznavanje nezgod, zbiranje podatkov, detekcijo prisotnosti in digitalno snemanje pred nezgodnimi posnetki in po njih. Detekcija prometnih dogodkov in MPEG-4 video kodiranje se izvajata neodvisno.

Algoritem v detektorju potrebuje vrsto parametrov, ki mu jih moramo nastaviti ob postavitvi kamere. Nastaviti mu moramo tako imenovane cone, s pomočjo katerih detektor razloči vozni pas od okolice, kakšna je velikost predmetov, katera je pravilna smer vožnje itd. Ko so vsi ti parametri pravilno nastavljeni, je tovrsten detektor sposoben zaznati dogodke z zanesljivostjo 98 %.



Slika 2: VIP-T detektor in njegova blok shema

Na vhodu detektorja je video signal, ki ga najprej pretvori v digitalno obliko. Ko je slika v digitalni obliki, jo analizira v realnem času in v primeru kakšnega dogodka sproži alarm. V omrežje je povezan prek mrežnega izhoda. S stališča kakovosti zaznavanja dogodkov je detektor najboljšo postaviti čim bližje kameri, vsekakor pred kodiranjem. Tako dobimo na vhodu najboljšo možno kakovost slike brez motenj, ki bi lahko vplivale na zanesljivost detekcije.

2.2.2 Programsko orodje TCT (Traficon Configuration Tool) za nastavitve VIP-T detektorjev

TCT programsko orodje je preprost program, ki ga namestimo na PC računalnik, povezan na isto ethernet omrežje. V programu je možen pregled vseh video signalov v

živi sliki, same nastavitve pa izvajamo na statični sliki. Na ta način lahko detektor skalibriramo, kar pomeni, da mu povemo širino cestišča ter višino kamere. S pomočjo teh podatkov detektor lahko preračuna velikost predmeta na začetku in na koncu vidnega polja kamere. Nastavimo tudi vrsto parametrov in con, posebej za zajem podatkov (število vozil, povprečne razdalje med vozili, itd.), posebej za detekcijo ustavljenih vozil, za detekcijo nasprotne smeri vožnje, za detekcijo pešcev in za detekcijo zadimljenosti v predoru.



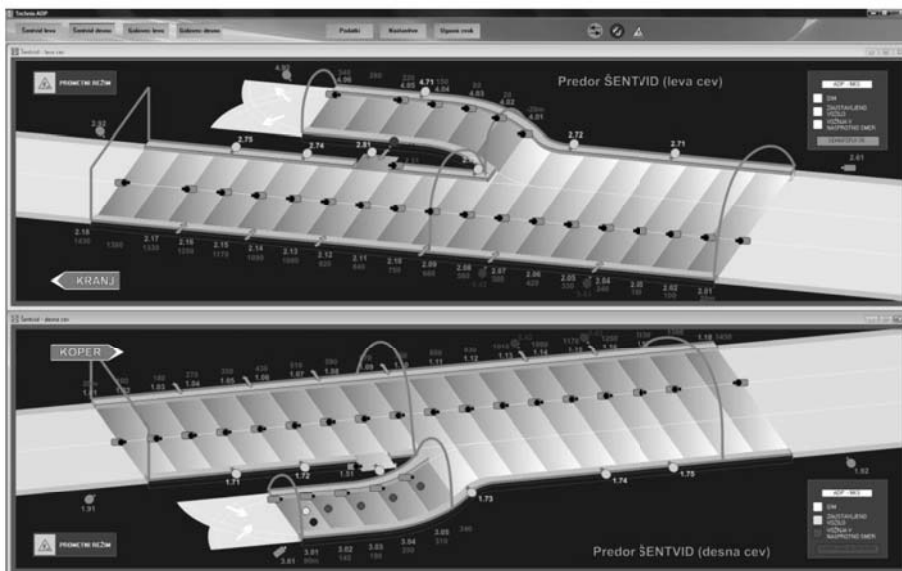
Slika 3: Nastavitev detekcijskih con

2.2.3 T-Port programska oprema

T-Port programska oprema je nameščena na obeh redundančnih strežnikih in po ethernet omrežju komunicira s posameznimi VIP-T detektorskimi karticami, od katerih prejema različne podatke o stanju prometa v predoru. Vse podatke (vključno s alarmnimi posnetki) shranjuje na trde diske v RAID tehnologiji. Na obeh strežnikih se izvajajo enaki procesi in v primeru izpada enega prevzame vse naloge drugi.

2.2.4 Aplikacija ADP

Aplikacija ADP teče vsaj na dveh delovnih postajah. Glavna delovna postaja je locirana v nadzornem centru in operaterju služi za grafični prikaz vseh prometnih dogodkov za vsak odsek posebej. Omogoča tudi potrjevanje alarmov ter sproščanje prometnih semaforjev (možna povezava na nadzorni krmilni sistem), vsa alarmna stanja pa je možno tudi pregledovati, razvrščati in izvažati v excelove datoteke.



Slika 4: Grafični vmesnih na ADP delovni postaji

2.3 Prednosti in slabosti

2.3.1 Prednosti:

- visoka stopnja detekcije in minimalen detekcijski čas (do 10 sekund)
- dokazano hitra in zanesljiva detekcija (25 let – 250 predorov – 50.000 detektorjev)
- preprosta instalacija, dolg povprečni čas med okvarami (MTBF) in kratek povprečen čas odprave okvare (MTTR)
- hitro in uporabniku prijazno nastavljanje (TCT – Traficon Configuration Tool)
- deluje na fiksnih in vrtljivih kamerah (PAL ali NTSC signal), novih ali že obstoječih infrastrukturah.

2.3.2 Slabosti:

- lažni alarmi, ki jih ob pravi izbiri kamer in dobri postavitvi le teh ter ob pravilni nastavitvi parametrov ni veliko (cca. 0,1 alarm/dan/kamero)

2.4 Delovanje v praksi

2.4.1 Nasprotna smer vožnje v predoru Dekani

Dne 1. decembra 2004 ob 6:13:44 se je v predoru Dekani v odstavni niši ustavilo osebno vozilo in po 10 sekundah je avtomatska detekcija prometa zaznala stoječe vozilo;

prometna signalizacija se je avtomatsko preklonila na rumeno utripajoče stanje (omejitev hitrosti 60). Potem je vozilo obrnilo in se odpeljalo po predoru v nasprotni smeri. Dogodek je zopet zaznala avtomatska detekcija prometa in po nekaj sekundah se je prometna signalizacija avtomatsko preklonila na rdečo luč in predor zaprla. Velja omeniti, da se je nekaj voznikov kljub rdečim lučem peljalo skozi predor, kar je bilo zelo nevarno početje. Na srečo ni prišlo do prometne nesreče; dogodek se je srečno končal.

Opozorilo !

Če so pred predorom in v njem vklopljene rdeče luči, je dejstvo, da se nekdo pelje v nasprotni smeri ali pa je v predoru požar! (lažni alarmi ne morejo sprožiti rdečih luči, ker je sistem tako zasnovan)



Slika 5: Nasprotna smer vožnje v predoru Dekani

2.4.2 Odpadanje ometa v predoru Šentvid

Dne 6 avgusta 2008 ob 1:55:05 je v stari galeriji predora Šentvid s stropa odpadel protipožarni omet in padel na osebni avtomobil nemškega turista. Ker je bila na cestišču velika količina betona, je bilo to za sistem avtomatske detekcije prometa enako, kot bi se ustavilo vozilo. Po 10 sekundah je zaznal »ustavljeno vozilo« ter prometno signalizacijo avtomatsko preklonil na rumene utripajoče luči (omejitev hitrosti 60). Hkrati se je na alarmni monitor prikazala živa video slika iz kamere, kjer se je zgodila nesreča, in na podlagi tega je operater predor zaprl (ročni vklop rdečih luči). Tak dogodek bi operater sam zelo težko opazil, saj je nemogoče ves čas spremljati 100 kamer hkrati. To pomeni, da bi bile posledice verjetno veliko hujše, če v predoru ne bi bila nameščena avtomatska detekcija prometa, ki je tudi tokrat reševala življenje ljudi.



Slika 6: Odpadanje ometa v predoru Šentvid

2.4.3 Nasprotna smer vožnje v predoru Šentvid

Dne 18. januarja 2010 ob 14:55:05 se je v predoru Šentvid pripeljalo osebno vozilo v nasprotni smeri vožnje. Sistem avtomatske detekcije prometa je dogodek zaznal in že po nekaj sekundah signalizacijo avtomatsko preklapljal na rumene utripajoče luči, nekaj sekund za tem pa na rdeče luči. V cca. 10 sekundah je bil predor zaprt. Nekateri vozniki zopet niso upoštevali prometne signalizacije in se kljub rdečim lučem peljali naprej. Vozilo v nasprotni smeri se kljub bližnjim srečanjem z več vozili, ki so vozili v pravilni smeri, ni ustavilo in se je v nasprotni smeri peljalo skozi ves predor in še naprej, dokler ga niso na izvozu pričakali policisti. Na srečo tudi v tem primeru ni prišlo do hujših poškodb, poškodovani sta bili le dve vozili.



Slika 7: Nasprotna smer vožnje v predoru Šentvid

2.4.4 Požar v predoru Trojane

Dne 20. januarja 2010 ob 15:12:16 se je v trojanskem predoru zgodila prometna nesreča, v kateri je bilo udeleženih več tovornih vozil. 10 sekund po trku je sistem avtomatske detekcije zaznal ustavljeno vozilo in prometna signalizacija se je avtomatsko preklpila v stanje rumenih utripajočih luči (omejitev hitrosti 60, utripajoči led smerniki), na alarmnem monitorju pa se je prikazala živa video slika iz kamere, ki je snemala nesrečo. Ker je operater opazil, da je pri nesreči prišlo do požara, je predor ročno postavil na rdečo luč in ga zaprl za ves promet, nato pa poklical gasilce. V tem primeru je operater s pomočjo avtomatske detekcije prometa tako hitro ugotovil dogodke, da se je odzval celo hitreje kot požarni sistem, ki bi sicer nekaj trenutkov po tem (ob dovolj visoki temperaturi) tudi avtomatsko zaprl predor. Vseeno pa se je avtomatsko sprožil požarni algoritem, ki je v prvi vrsti namenjen reševanju ljudi (evakuacija ljudi v sosednjo predorsko cev) ter izpihovanju dima iz predora. Zahvaljujoč avtomatskemu delovanju sistemov so bili vozniki in operaterji dovolj hitro obveščeni o nesreči, da so se lahko preprečile nadaljnje nesreče ter da ni bilo smrtnih žrtev.



Slika 8: Požar v predoru Trojane

3. POVEZAVA MED SISTEMI ZA NADZOR IN VODENJE PROMETA IN NAČIN ALARMIRANJA V NADZORNIH CENTRIH

Celoten sistem nadzora prometa na avtocestah je sestavljen iz več podsistemov, in sicer:

Avtomatska detekcija prometa – ADP, ki ob pomoči video slike iz kamere in detekcijskih kartic zazna nevarnost (nasprotna smer vožnje, ustavljeno vozilo, detekcija dima, detekcija pešca itd.), nanjo opozori operaterja prek ADP aplikacije na delovni postaji in alarm o nevarnosti posreduje naprej na ostale sisteme.

Nadzorni krmilni sistem – NKS predora, ki v primeru pomembnejših alarmov s strani sistema ADP in s strani krmilnikov sistema NKS (javljanje požara, odpiranje vrat v niši

in prečniku, prevelika koncentracija CO in zmanjšana vidljivost, pritisk SOS tipke itd.) sproži algoritem za krmiljenje semaforjev (utripajoče rumene luči, led smerniki ter omejitve hitrosti).

Video nadzorni sistem – VNP, ki na alarmnem monitorju prikaže video sliko iz kamere, ki pokriva območje, v katerem je bil zaznan alarm, vzporedno pa se alarm zapiše tudi v podatkovno bazo na VNP alarmni strežnik in izdela se posnetek dogodka za potrebe kasnejše analize.

Sistem nadzora in vodenja prometa – SNVP, ki na SPIS portalih pred predorom izpiše ustrezen napis o nevarnosti in znak omejitve hitrosti prometa.

Vsi ti sistemi, ki med seboj komunicirajo in si izmenjujejo podatke o nevarnih stanjih, skrbijo za varnost vseh udeležencev v avtocestnem prometu in omogočajo hitro obveščanje operaterjev o nevarnosti. Operaterji lahko zato pravočasno opozorijo voznike na nevarnosti, ki pretijo na njih, in sicer prek informacijskih portalov, radijskih valov in elektronskih medijev kot tudi prometnih znakov in svetlobnih opozoril. Hitrost zaznavanja nesreče in obveščanja voznikov je odvisna od tega, kako hitro bodo sistemi samodejno reagirali in javili alarm v nadzorni center. Vsaka izgubljena minuta namreč povečuje možnost za nastanek nove nesreče, ki bi lahko drastično povečala število žrtev, materialno škodo in čas za vzpostavitev normalnega stanja.

Samodejno ustvarjeni alarmi opozorijo operaterja takoj, ko se alarmni dogodek zgodi. Operater iz sistemov razbere vrsto alarma, njegovo resnost, lokacijo in ustrezne podatke o prometu oz. kaj se je zgodilo, koliko je udeleženih vozil, kakšne so posledice in kakšno je splošno stanje prometa na mestu alarmnega dogodka. Takšna kombinacija podatkov pa omogoča ustrezno javljanje intervencijskim reševalnim ekipam in udeležencem v prometu, da se prepreči nadaljnje nesreče in daljše zaprtje avtocestnega odseka.

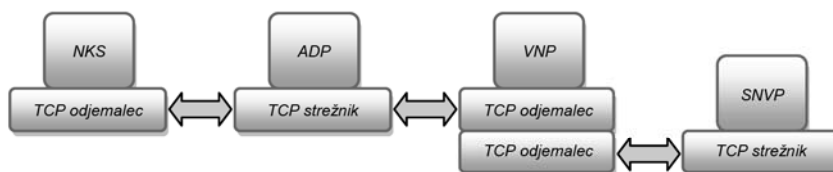
3.1 Povezava med sistemi ADP, NKS, VNP in SNVP

Ključni pomen povezave med sistemi je izmenjava podatkov o alarmnem stanju, obveščanje operaterja in opozarjanje voznikov o zaznani oz. javljeni nevarnosti. Za ta namen so bili v podjetju Technix in ostalih podjetjih, ki gradijo sisteme NKS in SNVP, razviti namenski TCP programski vmesniki za komunikacijo in algoritmi, ki sprožijo različna stanja za različno zaznane ali javljene alarme.

3.1.1 Opis komunikacije

Izmenjava podatkov o izrednih dogodkih in komunikacija med sistemi temelji na TCP/IP protokolu. Sporočila prek protokola TCP se zaradi vzpostavljene povezave med odjemalcem in strežnikom zanesljivo prenašajo v obe smeri, so brez napak, podvajanja in v pravem vrstnem redu. Ker je TCP povezovalni protokol, se najprej vzpostavi povezava med odjemalcem in strežnikom. Pri povezavi je določen odjemalčev naslov IP in vrata (vrata lahko zavzemajo vrednost od 1 do vključno 65535) ter strežnikov naslov

IP in vrata, na katerih posluša servis strežnika. Naslov IP, povezan z določenimi vrati, tvori vtičnico (ang. socket). Par odjemalčeve in strežnikove vtičnice pa tvori povezavo TCP, ki je edinstveno določena. Glava (ang. header) paketa TCP vsebuje izvorni naslov IP in vrata, ciljni naslov IP in vrata, zaporedno številko paketa, številko potrditve in kontrolne zastavice.



Slika 9: Komunikacija med sistemi

3.1.2 Sestava sporočila

Še preden se sistemi zgradijo, se za vsak predor pripravi podrobna dokumentacija, ki opisuje sestavo vseh vrst sporočil, ki si jih bodo sistemi med seboj izmenjavali, in okvirna navodila za delovanje programske opreme, ki bo skrbel za komunikacijo med sistemi. Splošna oblika sporočila, ki si ga sistemi med seboj izmenjujejo, je prikazana v Tabeli 1.

Tabela 1: Sestava sporočila

#	ŠT. BAJTOV	IME	OPIS
1	2B	DOLŽINA SPOROČILA	<i>Dolžina sporočila je enaka številu bajtov poslanega sporočila.</i>
2			
3	1B	LOKACIJA	<i>Številkska vrednost, ki predstavlja ime predora.</i>
4	1B	CEV	<i>Številkska vrednost, ki predstavlja predorsko cev (leva, desna, uvozna in izvozna cev).</i>
5	2B	OBMOČJE	<i>Celoten predor je razdeljen na območja, ki jih pokriva posamezna kamera (območje pred predorom, ki ga največkrat pokriva vrtljiva kamera, je območje 0, prva kamera v predoru je območje 1 itd.).</i>
6			
7	1B	SISTEM	<i>Številčna vrednost, iz katerega sistema je bilo javljeno alarmno sporočilo (NKS je 1, ADP je 2 itd.).</i>
8	1B	VRSTA	<i>Številkska vrednost, ki ponazarja vrsto informacije: sistemska (pošiljanje testnih sporočil, javljanje napak na opremi) je vrednost 1, tehnološka (alarmi iz sistema NKS) je vrednost 2, prometna (alarmi iz sistema ADP) je vrednost 3. Če bodo v sporočilu poslani številčne vrednosti ali izmerjene količine, so za vrsto informacije uporabljene druge dogovorjene vrednosti.</i>

9	1B	PODVRSTA	<i>Podvrste sistemskih, tehnoloških in prometnih informacij. To so številске vrednosti, ki predstavljajo vrsto sporočila. Sistemske: splošna napaka sistema, testni signal itd.; tehnološke: požar, odprta vrata, CO, vidljivost itd.; prometne: ustavljeno vozilo, pešec, detekcija dima, vožnja v nasprotno smer itd.</i>	
10	1B	REZERVA	<i>Uporablja se za spremembo pozicije vrtljive kamere, lahko pa se uporabi za kateri drug dogovorjen parameter.</i>	
11	2B	STANJE / VREDNOST	<i>0, če gre za normalno stanje, 1, če gre za alarm, katere druga številčna vrednost določenega parametra.</i>	
12				
13	7B	ČAS	<i>Zapiše se točen čas, kdaj je bilo sporočilo poslano.</i>	
14				Leto
15				Mesec
16				Dan
17				Ura
18				Minuta
19				Sekunda
20	1B	CHECKSUM	<i>Se uporablja za redundantno preverjanje napak.</i>	

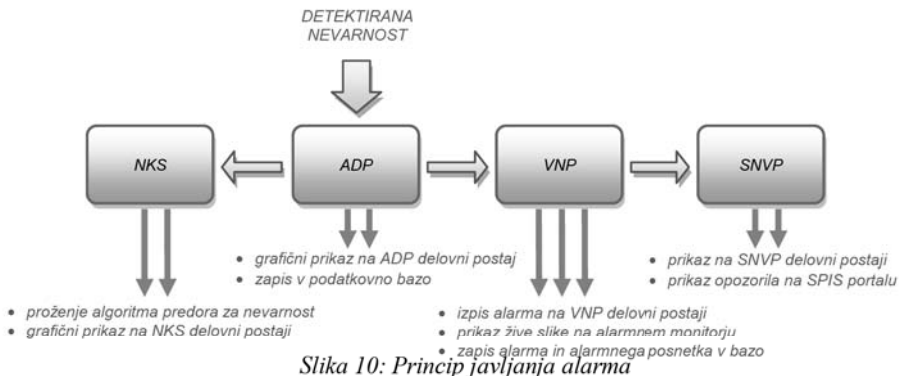
3.1.3 Preverjanje stanja povezave in poslanih podatkov

Za dodatno preverjanje stanja povezave si sistemi na vsakih 15 sekund izmenjujejo paket z vsebino: vrsta inf.=1, podvrsta inf.=3. Če po 30 sekundah ni nobenega paketa, TCP strežnik ali TCP odjemalec prekine povezavo, TCP strežnik preide v stanje poslušanja, TCP klient pa ponovno poizkuša vzpostaviti povezavo s TCP strežnikom.

Da preprečimo napačno interpretiranje poslanega sporočila, se pravilnost prenesenih podatkov dodatno preverja še z operatorjem XOR (bitni) med bajti 1 do 19.

3.1.4 Proženje algoritmov in način alarmiranja v nadzornem centru

V nadzornem centru operaterji upravljajo s sistemi za nadzor in vodenje prometa s pomočjo delovnih postaj, na katerih so nameščene namenske aplikacije, ki prikazujejo stanja in dogajanje v predorih in na avtocestni trasi. V pomoč jim je tudi stenski prikazovalnik, na katerem se prikazujejo sekvenca, alarmne in žive video slike iz poljubnih kamer ter globalna stanja predorov in avtocestne trase. Ker je operater stalno izpostavljen ogromni količini informacij, je pomembno, da ga sistemi sami čim bolj opozorijo na nevarnost, ki je bila zaznana oz. javljena z določenega sistema. Za ta namen je bila razvita vrsto algoritmov, ki ob pomoči podatkov iz drugih sistemov operaterja natančno obveščajo o izrednih dogodkih.

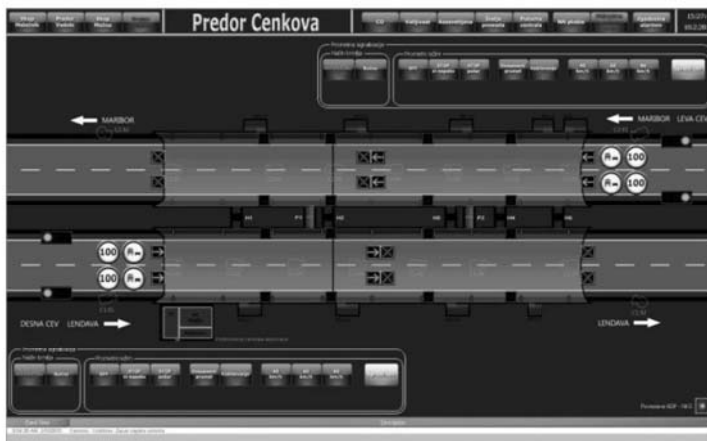


Tako bo na primer sistem avtomatske detekcije prometa ob pomoči video slike in algoritmov na detekcijski kartici zaznal ustavljeno vozilo. Detekcijska kartica bo podatke o ustavljenem vozilu poslala na ADP strežnik. Tu se bo generiralo alarmno sporočilo, ki bo vsebovalo vse podatke o lokaciji in vrsti zaznane nevarnosti. Sporočilo se bo hkrati poslalo na ostale tri sisteme NKS, VNP in SNVP. Na ADP delovni postaji pa bo zvočni efekt in grafični prikaz opozoril operaterja na nevarnost v predoru. Hkrati se bo alarm in posnetek iz kamere zapisal še v podatkovno bazo na ADP strežniku.



Slika 11: Grafični prikaz javljenih alarmov na ADP delovni postaji

Sistem NKS bo sprejel alarmno sporočilo, ga razbral in sprožil algoritme, ki bodo predor preklpili v stanje 60 (omejitev hitrosti 60, utripajoča rumena luč in utripajoči led smerniki). Prav tako bo prek delovne postaje javljen zvočni efekt in prikazan grafični prikaz novega stanja predora.



Slika 12: Grafični prikaz na NKS delovni postaji

Na VNP delovni postaji bo javljen zvočni efekt in izpisali se bodo podatki o prejemu alarmnem sporočilu. Na alarmnem monitorju se bo prikazala živa video slika iz kamere, ki pokriva območje ustavljenega vozila. Alarm se bo zapisal še v podatkovno bazo na VNP alarmnem strežniku in izdelal se bo 3 minutni posnetek (1 minuto pred javljenim alarmom in 2 minute po njem) za kasnejše potrebe analize.



Slika 13: Izpis podatkov o javljenih alarmih na VNP aplikaciji

Na podoben način sistemi reagirajo tudi ob ostalih zaznanih nevarnostih, javljenih iz sistema ADP ali sproženih iz sistema NKS, pri čemer se ob pomembnejših dogodkih, kot so vožnja v nasprotno smer, zaznava dima in javljanje požara, sprožijo algoritmi, ki predor ustavijo za ves promet.



Slika 14: Nadzorni center Vranksko

4. KJE SISTEME ZA AVTOMATSKO DETEKCIJE PROMETA ŠE SREČAMO IN KJE JIH ŠE BOMO

4.1 Sistem detekcije prisotnosti vozil v semaforiziranih križiščih

Sistem je namenjen spremljanju prometa in bolj dinamičnemu spreminjanju signalizacijskih stanj v prometnih križiščih. Detekcija zaznava prisotnost premikajočih in ustavljenih vozil pred križiščem. Glede na gostoto vozil na posameznih prometnih pasovih se krmilijo semaforji.

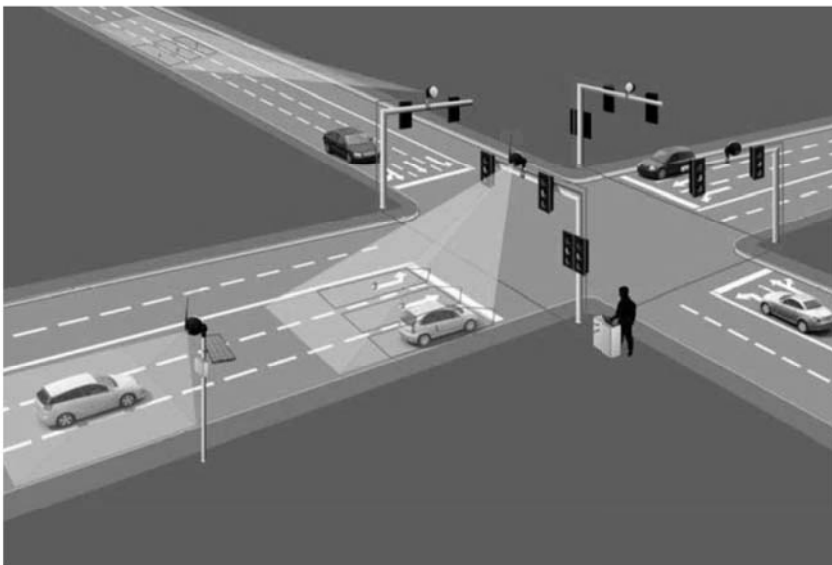
4.4.1 Traficam X-stream

Za sistem detekcije prisotnosti vozil v križiščih se uporablja detekcijska kamera Traficam X-stream. To je integracija kamere in detektorja, ki omogoča zaznavanje prisotnosti vozil in MPEG-4 kodiranje. Ta inteligentni sistem omogoča detekcijo in pregled premikajočih in stoječih vozil v semaforiziranem križišču. Skozi detekcijske izhode informacijo o prisotnosti vozil posredujemo krmilnemu sistemu prometne signalizacije. Tako je lahko časovno spreminjanje signalizacije bolj dinamično. S tem zmanjšamo čakanje vozil, povečamo prometno pretočnost in zmanjšamo onesnaževanje ozračja.



Slika 15: Traficam X-Stream detekcija prisotnosti vozil

Traficam X-stream je IP video detekcijska kamera, od katere lahko vodimo video signal (živo sliko) prek ethernet omrežja do nadzornega centra. Nastavitve in parametrisiranje je možno na sami lokaciji ali prek oddaljenega dostopa. Nastavljanje je zelo hitro in preprosto, saj je sistem lahko funkcionalen že v nekaj minutah. Poleg tega nam lahko privarčuje kar nekaj stroškov vzdrževanja in instalacij običajnih zank v vozišču.[1]



Slika 16: Primer semaforiziranega križišča z vgrajenimi kamerami Traficam

4.4.2 TraftiCam Collect-R

To je kamera, ki spremlja promet in nam posreduje podatke o hitrosti vozil, zasedenosti vozišča in številu vozil, ločenih po razredih – tovorna, kombinirana, osebna. S postavitvijo navideznih con lahko simuliramo navidezne zanke na posameznih pasovih. Kamera nadzoruje prometno pretočnost.[2]



Slika 17: Montirana visoko nad cestiščem omogoča spremljanje prometa na 4 voznih pasih hkrati

4.4.2 TraftiCam C-Walk

Ta kamera ima vgrajene algoritme za detekcijo prisotnosti pešcev na cestnem prehodu. Namenjena je izboljšanju varnosti pešcev pri prečkanju ceste predvsem v okolici šol, bolnišnic in nakupovalnih centrov. V povezavi s prometno signalizacijo omogoča bolj dinamično upravljanje s semaforji, pri čemer se zmanjšujejo nepotrebne zakasnitve pri čakanju na zeleno luč.[3]



Slika 18: TraftiCam C-Walk detekcija prisotnosti pešcev

4.1 SafeWalk - 3D video detekcija

SafeWalk, ki ga je razvil Traficon, je popolnoma nov produkt, ki združuje tehnologijo obdelave stereoskopskih video slik in pametnih algoritmov za zaznavanje pešcev.

Pešci so poleg kolesarjev in motoristov najbolj ogroženi udeleženci v cestnem prometu in predstavljajo letno približno 46 % od 1,3 milijona smrtnih žrtev v cestnem prometu. Po podatkih Svetovne zdravstvene organizacije znaša število smrtnih žrtev teh udeležencev v manj razvitih državah Jugovzhodne Azije celo 80 %. Posledica teh pretresljivih statistik je vedno večja osredotočenost na ogrožene skupine v cestnem prometu, in sicer na področju urbanističnega načrtovanja, upravljanja prometa in avtomobilske varnosti. SafeWalk je nastal z namenom soočanja z omenjenimi trendi in je tako na področju inteligentnih transportnih sistemov najnovejša rešitev video detekcije. Glavni cilj sistema SafeWalk je nadzorovati in izboljšati delovanje ter varnost prometne signalizacije na območju prehodov za pešce. Z bolj dinamičnim upravljanjem in nadzorom prometne signalizacije pa se zmanjšujejo nepotrebne izgube časa tako za udeležence, ki prečkajo cesto, kot tiste, ki se po cesti peljejo.

SafeWalk z uporabo dveh CMOS kamer hkrati pretvori dvodimenzionalne video slike v tridimenzionalno informacijo. Takšna informacija omogoča razlikovanje med dejanskimi objekti in nepomembnim ozadjem, kot so sence in odsevi. Na osnovi teh 3D informacij Traficonovi algoritmi zagotavljajo še bolj poglobljeno analizo in natančnejše informacije o prisotnosti in vedenju pešcev. Visoka stopnja zaznavanja in imunosti na lažne alarme omogoča zaznavanje in nadzor pešcev, ki čakajo na prehod čez cesto v okviru določenega območja. Takoj, ko pešec vstopi v določeno območje detekcije in čaka nastavljivo določen čas, krmilnik semaforja dobi informacijo o čakajočem pešču in aktivira zeleno luč za pešce. Poleg tega pa lahko sistem zadrži rdečo luč, dokler ni prisotnega nobenega pešca več, zaradi česar promet teče bolj tekoče in učinkoviteje. S pomočjo te inovativne detekcije za pešce lahko naredimo prečkanje cest v bližini šol, nakupovalnih središč in bolnišnic še varnejše in učinkovitejše.[4]



Slika 19: SafeWalk 3D detekcijski senzor in primer namestitve

5. ZAKLJUČEK

Od leta 2002 do danes je Technix vgradil v slovenske predore (Ločica, Jasovnik, Kastelec, Dekani, Trojane, Podmilj, Močna, Šentvid, Cenkova, Barnica, Podnanos, Vodole in Malečnik) več kot 300 VIP detektorjev, ki delujejo 24 ur na dan in 365 dni na leto. Okvar modulov je bilo manj kot pet, kar pomeni, da se je oprema izkazala za zelo zanesljivo. Sama primeri zaznanih dogodkov pa dokazujejo, da je avtomatska detekcija prometa iz vidika varnosti v predorih eden najpomembnejših sistemov.

Izmenjava informacij o alarmnem stanju med sistemi je ključnega pomena, saj lahko sistemi ob zaznanih alarmih sami reagirajo in preprečijo nadaljnje nesreče, pa tudi operater hitreje pridobi več informacij o nastali situaciji. V prihodnosti je cilj, da ob javljanju nevarnosti v predoru vključimo tudi ostale sisteme, kot so sistem prepoznavanja nevarnega tovora, kar bi operaterju omogočalo še popolnejše obveščanje intervencijskih ekip.

Tudi na ostalih področjih je video detekcija doživela velik napredek. Pametnih senzorjev, ki za osnovo uporabljajo video sliko, je čedalje več: Na tovornih vozilih za spremljanje budnosti voznika, pri avtomobilih za spremljanje mrtvega kota pri menjavi voznih pasov, opozarjanje na nevarnost naleta, prepoznavna prometnih znakov in opozarjanje voznika na omejitve hitrosti, detekcija pozabljene prtljage na letališčih, detekcija registrskih tablic, štetje obiskovalcev, prepoznavna obrazov itd. Pospešeno se razvija tudi stereoskopija ali 3D detekcija, pri kateri je zaznavanje objektov še boljše, lažnih alarmov pa posledično manj. 3D detekcija bo v prihodnosti najverjetneje zamenjala obstoječe video detektorje in nam omogočala še varnejše bivanje v tem hitrem načinu življenja.

LITERATURA

1. <http://www.traficon.com/solutions/product.jsp?id=36&parentType=ProductCategory>; TrafiCam X-Stream; februar 2010.
2. <http://www.traficon.com/solutions/product.jsp?id=26&parentType=ProductCategory>; TrafiCam Collect-R; februar 2010.
3. <http://www.traficon.com/solutions/product.jsp?id=27&parentType=ProductCategory>; TrafiCam C-Walk; februar 2010.
4. http://www.traficon.com/products/spotlight_detail.jsp?id=15; SafeWalk; februar 2010.

PORAVNAVA SLIK V SLIKOVNO VODENIH MEDICINSKIH POSTOPKIH

Franjo Pernuš

Fakulteta za elektrotehniko, Univerza v Ljubljani

E-pošta: franjo.pernus@uni-lj.si

POVZETEK: *Medicinske slike, ki so zajete z različnimi tehnikami slikanja, so danes nepogrešljive za postavitev diagnoze, načrtovanje in simulacijo posega, izvedbo posega ter za spremljanje učinkov posega ali napredovanja bolezni. Vsak poseg (operacija, radioterapija, radiokirurgija, intervencijska radiologija, itd.) naj bi bil čim natančnejši in minimalno invaziven. Ti dve zahtevi lahko dosežemo s slikovno vodenimi posegi, med katerimi kirurg, radioterapevt ali kemoterapevt s pomočjo slik, zajetih pred posegom, minimalno invazivno in natančno usmerja različna kirurška orodja, izvore energij ali kemikalije proti ciljani anatomske strukturi in se pri tem izogne kritičnim strukturam, kot so pomembne žile in živci.*

Osnovne komponente slikovno vodenih posegov so: zajem tridimenzionalne (3D) slike pred posegom z računalniško tomografijo (CT) ali magnetno resonančnim (MR) slikanjem, zajem ena ali več dvodimenzionalnih (2D) fluoroskopskih slik tik pred in/ali med posegom, poravnava (registracija) 3D pred in 2D medoperativnih slik, sledenje kirurških orodij v prostoru ter istočasno prikazovanje poravnanih slik, načrta posega in kirurških orodij na zaslonu. Podan bo pregled postopkov za poravnavo 3D slik zajetih pred in 2D slik zajetih med posegom.

MOŽNOSTI OBDELAVE SLIK V APLIKACIJAH ZA MOBILNE TELEFONE

Sergej Rinc, Miha Gaberšek

spletin.net d.o.o.

E-pošta: podjetje@spletin.net

URL: <http://www.spletin.net>

POVZETEK: Mobilni telefon ni le bolj razširjena naprava za komuniciranje kot klasični telefon, ampak vse bolj tudi osrednja naprava za komuniciranje ljudi. Poleg telefoniranja in sporočanja s kratkimi obvestili pametni mobilniki omogočajo tudi izmenjavo elektronske pošte ter spletno brskanje, vključujejo pa tudi digitalne fotoaparate. Vse navedeno predstavlja zadostne pogoje za smiselnost razvoja dodatnih aplikacij tudi s področja obdelave digitalnih slik.

Za razvijalce aplikacij je najbolj poskrbela družba Apple z možnostmi, ki jih ponuja model iPhone. Pregledali smo zahteve in orodja za razvoj aplikacij za iPhone ter pripravili zabavno aplikacijo za uvrstitev v spletno trgovino AppStore. Primerjali smo rezultate obdelave slik glede na osebne računalnike in ugotovili smiselnost razvoja ter komercialni model multimedijskih aplikacij za obdelavo slik v mobilnih telefonih.

1. UVOD

Mobilna telefonija je od leta 2005 de facto standard in gonilo razvoja telekomunikacij. Prodaja na mobilnem trgu v Sloveniji je leta 2006 znašala 502 milijona EUR, kar je pomenilo glede na prodajo na fiksnem trgu – ta vključuje tudi telefonijo VoIP – v obsegu 382 milijonov EUR že 31,4% več [1]. Leta 2009 je storitve mobilne telefonije uporabljalo 5 milijard prebivalcev Zemlje in po predvidevanjih ITU bo do leta 2015 več ljudi dostopalo do spleta prek prenosnikov, mobilnikov in podobnih naprav kot pa prek namiznih računalnikov [2].

Mobilni telefon oz. mobilnik ni le bolj razširjena naprava za komuniciranje kot klasični telefon, ampak vse bolj tudi osrednja naprava za komuniciranje ljudi. Pametni mobilniki omogočajo telefoniranje, sporočanje s kratkimi obvestili ter izmenjavo elektronske pošte in spletno brskanje, vključujejo pa tudi digitalne fotoaparate. Za mobilnik z integriranim digitalnim fotoaparatom se vse bolj uveljavlja izraz *fotofon*. Ta izraz odstopa od starega slovenskega izraza za napravo, ki jo je izumil G. Bell in ki je omogočala prenos govora s pomočjo svetlobnih signalov [3].

Trg in tehnične možnosti modelov fotofonov predstavljajo zadostne pogoje za smiselnost razvoja dodatnih aplikacij tudi s področja obdelave digitalnih slik. Pri tem ne mislimo na zajem slik na fotofonu in pošiljanje v obdelavo v namenski strežnik v spletu (npr. [4]) ampak na lokalno obdelavo slik v fotofonu.

2. RAZVOJNA OKOLJA ZA MOBILNE APLIKACIJE

V tem poglavju pregledamo mobilne standarde in razvojna okolja za mobilne aplikacije, da spoznamo razlike glede na razvoj namiznih aplikacij. Razvijalci programske opreme jih moramo upoštevati, za obdelavo slik pa moramo preveriti tudi strojne zmogljivosti fotofona ter možnosti programske arhitekture.

2.1 Mobilni standardi za grafični prikaz lastnih aplikacij

Osnovni in preprost prikaz multimedijskih vsebin omogočajo specifikacije, ki izhajajo iz označevalnega jezika HTML. Opisali bomo vlogo WAP, chtml in XHTML.

Glavni avtor tega prispevka je razvil mobilni portal wap.skb.si (angleška verzija je bila izdelana leta 2000). Zunanji izvajalec je dodal še mobilno verzijo elektronskega bančništva za komunikacijski protokol WAP [5] a je bilo že leta 1999 jasno, da je WAP rešitev za problem, ki ga še ni bilo. Dostopni mobilni telefoni na trgu so imeli male ekrane brez prikaza barv, kar ni bilo privlačno za uporabnike. Mobilni operaterji niso sledili uspešnemu japonskemu modelu imode [6], ki je stavil na storitve in razvijalcem prijazno verzijo označevalnega jezika *chtml*. Ta je omejena in mobilnim telefonom prirejena verzija označevalnega jezika HTML in ne na novo napisan jezik kot WML (ang. Wireless Markup Language, označevalni jezik za brezžične naprave).

Mobilni telefoni na japonskem trgu so hitro dobili tudi podporo prikazu do 32.768 barv in vgrajen fotografski del. Drugje v svetu so uporabniki in delno tudi mediji pričakovali, da je WAP dejansko internet na mobilnem telefonu. To brez naprednih telefonov in storitev ni primerljivo. Razvoj mobilnikov je kasneje doprinesel k zmogljivi strojni zasnovi, ki omogoča vključitev podpore za standardni označevalni jezik XHTML. Razvijalci spletnih aplikacij tako nimajo težav pri implementaciji mobilne verzije aplikacije vsaj pri njenem uporabniškem vmesniku (ang. »front end«). Sedaj je de facto standard uporaba odprtokodnega ogrodja WebKit za prikaz spletnih strani (Nokia v operacijskem sistemu Symbian, ki je od 4. februarja 2010 odprtokoden [7] in je najbolj razširjen v mobilnikih, nato SonyEricsson, Apple idr.).

Trije omenjeni standardi so evolucija označevalnega jezika HTML in predvsem primerni za aplikacije, ki omogočajo osveževanje podatkov s strežnika ponudnika aplikacije. Za bolj zmogljive in samostojne aplikacije niso primerni. V naslednjem poglavju si bomo ogledali možnosti, ki so na voljo za razvoj domorodnih mobilnih aplikacij.

2.2 Razvojna okolja in uporabniški vmesniki za mobilne aplikacije

Razvoj naprednih mobilnih aplikacij omogočajo razvojna okolja, prirejena za mobilnike. Pregledali bomo najbolj razširjena okolja in vpliv uporabniških vmesnikov nanje.

Vodilno vlogo pri razvojnih okoljih za mobilne aplikacije ima operacijski sistem Symbian [7] in zanj grafično razvojno okolje (SDK), ki vsebuje prevajalnik za programski jezik C++, na voljo pa je tudi okolje za java. Pobudnik industrijske skupine

Symbian je bila družba Nokia. Ko pripravljamo aplikacijo za določeni telefon znamke Nokia, moramo poznati verzijo Symbian OS v tem telefonu, da nato zanjo prevedemo aplikacijo. Razvojno okolje dopušča več hkrati nameščenih različnih verzij Symbian OS SDK-ja in s tem testiranje aplikacije za več telefonov in verzij vgrajene tovarniške programske opreme (ang. »firmware«).

Če le omenimo standardno podporo za arhitekturo Java v vseh mobilnikih z imode, je podpora Jave odvisna od proizvajalca mobilnih telefonov. SonyEricsson tako ponuja mobilnike in razvojna okolja za Javo, C in C++ in to za lastne implementacije operacijskega sistema kot tudi za Microsoft Windows Mobile itd. Samsung in LG se v zadnjem času posvečata Windows Mobile a z lastnim grafičnim uporabniškim vmesnikom, ki zamenja privzete Microsoftove ekranske operacije z uporabniku bolj prijaznimi npr. z dotikanjem s prsti.

Microsoft Windows Mobile v trenutno aktualni verziji 6.5 še ne omogoča prave uporabniške izkušnje z zasloni, občutljivimi na dotik. Čeprav je možno izbrati veliko operacij z dotikom na grafično ikono, je pri izbiri nadaljnjih možnosti pogosto potrebno poseči po pisalu ali navigaciji s tipkami. V letu 2010 naj bi verzija 7.0 odpravila tudi to nedoslednost, ki je ne morejo zakriti tudi ponudniki mobilnikov z izboljšanim uporabniškim vmesnikom (npr. HTC z modeli Touch). Ta se za opravila uporabnika prikaže namesto vmesnika Windows Mobile in s tem omogoča hitrejši in bolj uporaben dostop do pogosto uporabljenih izbir.

Microsoftov primer kaže na to, da uporabnikom ne ustreza le verzija namiznega operacijskega sistema, ki je prirejena za mobilnike. To je spoznala družba Apple, ki je junij 2007 na trg lansirala mobilnik iPhone. Že prvi model iPhone je imel zaslon, občutljiv na dotik, za uporabo s prsti pa tudi poseben operacijski sistem iPhone OS. iPhone OS je verzija jedra operacijskega sistema Darwin, ki ga ima operacijski sistem Mac OS X za Appleove računalnike, vsebuje pa tudi dodatne funkcije (npr. za dotik). Skupen razvoj za namizne računalnike Mac in mobilnike iPhone ni možen. Aplikacije iPhone zahtevajo posebno pisanje in prevajanje kode z orodji iPhone SDK [8] za izgradnjo končne aplikacije. Razvoj mobilnih aplikacij za iPhone opišemo v poglavju 4.

Za popolno sliko moramo omeniti še okolje Google Android. Z njim želi družba Google Inc. doseči večji del oglaševalskega trga tudi na mobilnih napravah. Android je verzija operacijskega sistema Linux vendar je Google odstranil Android iz razvojnega drevesa Linuxa. To je verjetno posledica vključenosti dodatnih, tudi s patenti zaščitene aplikacij Googla v distribucijo Androida. Razvijalci, ki podpirajo odprtokodne rešitve, so s tem lahko prikrajšani za razumevanje delovanja kode Androida in njeno prilagoditev za lastne aplikacije. Za razvoj je na voljo brezplačno okolje Android SDK.

Tako smo pregledali vodilna razvojna okolja za mobilne aplikacije. Odločitev o razvoju mobilne aplikacije za trg je odvisna od možnosti za njeno uspešno prodajo, zato vedno upoštevajmo tudi razvijalski vidik. Opis je v naslednjem podpoglavju.

2.3 Razvijalski vidik

Razvijalci mobilnih aplikacij za prodajo na trgu potrebujejo zmogljivo razvojno orodje, ki omogoča hiter razvoj in časovno nezamudno preverjanje delovanje aplikacij v končnih napravah – mobilnikih. Odločitev za razvoj aplikacij za določene družine mobilnikov ali celo le za en model mobilnika je odvisna od razširjenosti teh mobilnikov na trgu ali naročnika aplikacije. Tretji vidik je možnost izvajanja želenih operacij v mobilniku.

V preteklosti smo za razvoj naših aplikacij uporabljali predvsem Symbian SDK in programski jezik java. Specifikacije WAP predpisujejo dober način distribucije in namestitve aplikacij iz javanskih arhivov (datoteke s končnico .jar). Težava tega idealnega načina distribucije aplikacij je njegovala podprtost v mobilnih telefonih. Tako smo morali opustiti razvoj aplikacij s praktično avtomatsko namestitvijo na uporabnikov telefon, ker predpisani model telefona Nokia s strani naročnika ustrezne specifikacije WAP ne podpira! Tržna moč največjih ponudnikov mobilnikov žal pomeni tudi (pre)počasno podporo standardom in specifikacijam, ki bi jih razvijalci potrebovali.

Leta 2009 smo začeli testirati okolje iPhone SDK [8]. Za razvoj aplikacij za iPhone se uporablja predmetno usmerjena verzija prog. jezika C (Objective C [9]). Morda družba Apple, ki je v osnovi ponudnik strojne opreme (računalniki, mobilniki, predvajalniki MP3), prav zaradi pospeševanja prodaje te opreme ponudi razvijalcem zelo zmogljiva orodja. Zaradi hitrega razvoja in razdroščevanja aplikacij z izvajanjem v simuliranem okolju ali s povezanim mobilnikom iPhone smatramo iPhone SDK za dejansko najboljšo razvojno okolje za mobilne aplikacije. Orodje Interface Builder je prikazano na sliki 1.



Slika 1: Orodje Interface Builder za razvoj aplikacij za iPhone, viden je simuliran mobilnik.

Mobilniki iPhone imajo pomemben delež trga mobilnikov. Podatek o prodanih cca. 7,83 milijona mobilnikov iPhone do avgusta 2009 in vodilni tržni delež 32,9 % na trgu mobilnikov z zaslonom, občutljivim na dotik [10], je dovolj zgovoren. Število aplikacij v spletni tržnici AppStore [11] 15. januarja 2010 je znašalo 133.979 [12], skupaj pa je bilo do 5. januarja 2010 opravljeno več kot 3 milijarde prenosov aplikacij [13].

V aplikaciji iPhone želimo izvajati tudi napredne operacije obdelave slik. Namenska vezja v modelu iPhone 2G že omogoča obdelavo posameznih slik v realnem času oz. času, ki je sprejemljiv za uporabnika po splošnih načelih uporabnosti (3-4 s za odziv naprave po sprožitvi operacije, podrobnosti so v podpoglavju 3.2). Bistvena strojna oprema zadnje verzije mobilnika iPhone 3GS [14] je navedena v tabeli 1.

Tabela 1: Bistvena strojna oprema mobilnika iPhone 3GS

<i>KOMPONENTA</i>	<i>LASTNOSTI IN POMEMBNOST ZA OBDELAVO SLIK</i>
CPE ARM Cortex A8	600 MHz, izboljšana enota za operacije s plavajočo vejico, prvonivojski predpomnilnik L1 32/32 KB (za ukaze in podatke) ter dodan drugonivojski predpomnilnik L2 256 KB. Poleg tega je dodan tudi paralelni pogon SIMD (ang. Single Instruction Multiple Data) z imenom NEON. - za obdelavo slik potrebujemo hitro CPE.
GPE PowerVR SGX 535	200 MHz, prikaz 28 milijonov poligonov/s oz. do 400 milijonov pik/s, podpora OpenGL ES 2.0 - lepljenje tekstur prek objekta, hitrejši prikaz slike.
eDRAM	256 MB dinamičnega pomnilnika - uporabnik ima lahko odprtih več aplikacij in čeprav so med preklopi te aplikacije v »zamrznjenem stanju«, je modelom iPhone pred verzijo 3GS lahko zmanjkalo pomnilnika.

Po naših izkušnjah je z modelom iPhone 3GS in optimizacijo aplikacije možno doseči čas obdelave slike (npr. razpoznavanje obraza, lepljenje texture), ki znaša 1-2 s.

3. MOŽNOSTI OBDELAVE SLIK V APLIKACIJAH IPHONE

Opišimo možnosti obdelave slik v mobilniku iPhone na zgledih nekaterih obstoječih aplikacij v AppStore [11]. Oglejmo si naš primer razvoja mobilne aplikacije za iPhone in rezultate. Za druge razvijalce smo navedli tudi priporočene korake za razvoj, dodali pa tudi pomemben vidik komercialnega uspeha aplikacij za iPhone – promocija.

3.1 Pregled aplikacij iPhone v kategoriji fotografija v AppStore

Izmed aplikacij za osnovno obdelavo slik omenimo aplikacijo PhotoGene. Gre za aplikacijo, ki ima običajne operacije nad slikami kot v grafičnih namiznih programih. Omenimo izenačevanje histogramov, spremembe intenzitete barv in črnobeli efekt. Podobnih aplikacij je še nekaj deset.

Aplikacij, ki bi zares razpoznavale objekte na slikah za napredne operacije nad slikami, je v AppStore malo. Od pribl. 2.700 aplikacij v kategoriji fotografija dejansko nismo našli aplikacije, ki razpozna obraz avtomatsko. V aplikacijah kot so MixFace (menjava delov obraza s slikovnimi izseki portretov drugih ljudi), Face IT! (dodajanje zabavnih dodatkov na obraz npr. lasulja, očala, klobuk ipd) in Faces! (menjava slike obraza z drugo) mora uporabnik sam določiti področje obraza, ki se nato potem uporabi za zamenjavo ali lokalno transformacijo.

Ker smo s simulatorjem mobilnika iPhone izračunali zadovoljivo hitrost razpoznavanja obraza in človeške figure (manj kot 4 s), smo to izkoristili za pripravo aplikacije z razpoznavo obraza in telesa z vključeno zabavno noto. Vsekakor nismo nameravali izrabiti popularnost portala YouTube, ki jo je neko podjetje izkoristilo za aplikacijo Nude It z (lažno) obdelavo slik (menjava obleke brez razpoznavanja figure).

3.2 Zabavna mobilna aplikacija iDelude

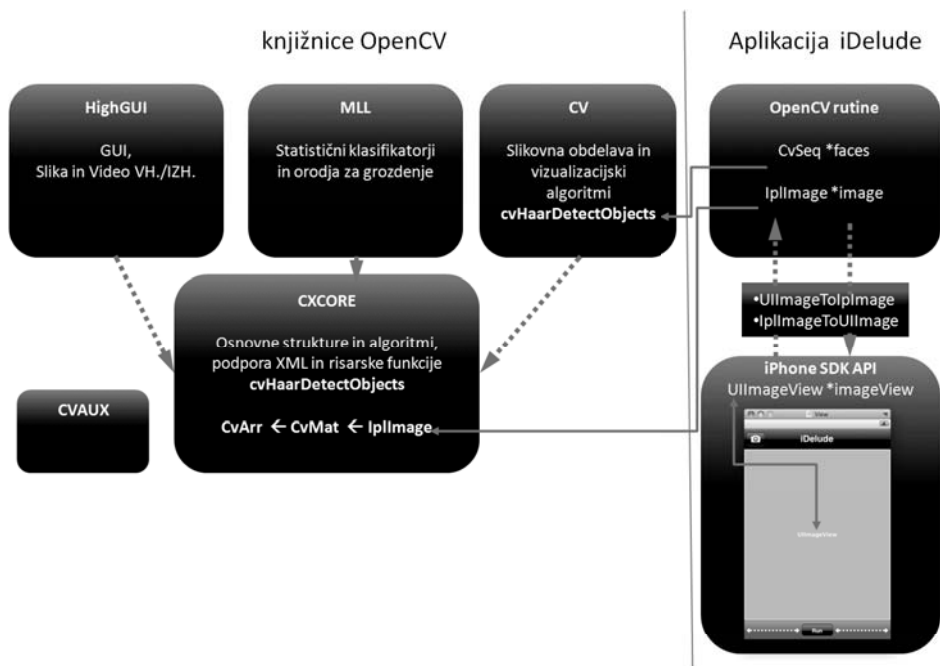
Izdelali smo mobilno aplikacijo iDelude [15], ki omogoča razpoznavanje obraza in človeške figure na digitalni sliki. Zabavna nota je v tem, da po razpoznavi zamenjamo obleko človeka s kopalkami.

Več podrobnosti o aplikaciji iDelude ter rezultati bodo na voljo v diplomski nalogi drugega avtorja. Zato so v nadaljevanju le osnovne ideje, ki smo jih implementirali.

Odločili smo za uporabo odprtokodne knjižnice OpenCV družbe Intel. To je znan skupek programskih metod za procesiranje slik v realnem času. Ker uporablja knjižnica OpenCV programski jezik C++, za iPhone pa moramo uporabljati prog. jezik Objective C, smo morali implementirati povezavo med njima. Ker iPhone v osnovni arhitekturi ne omogoča dodatnega odjemalca metod knjižnic torej dinamično povezovanje, smo morali vpeljati statično povezovanje ustreznih knjižnic OpenCV. Arhitektura pri klicu funkcije cvHaarDetectObjects je prikazana na sliki 2.

Implementirali smo tri metode. Z metodo *FaceDetect* razpoznamo obraz človeka na sliki. Osnova metode je uporaba funkcije cvHaarDetectObjects iz OpenCV in postopek razpoznavave pravega »kvadrata« iz množice potencialnih obrazov v sliki dodatno optimizirali. Ta postopek je poslovna skrivnost, zato podrobnosti ne moremo navesti.

Metoda *SkinColor* določi oz. ugotovi barvo kože obraza na sliki. Za pridobljeno kvadratno področje iz metode FaceDetect izračunamo barvni histogram v barvnem modelu HSV. Z združevanjem kvantizacijskih nivojev histograme zmanjšamo različna področja na 32 barvnih odtenkov (H), 32 nivojev nasičenja (S) in 32 nivojev svetilnosti (V). Izhod metode nam nato vrne skalarno vrednost HSV dominantne barve obraza.



Slika 2: Povezava knjižnice OpenCV s prog. vmesnikom iPhone SDK. Prikazan je klic funkcije `cvHaarDetectObjects` za razpoznavo obraza na izbrani sliki.



Slika 3: Primer slike z označenim območjem modrega kvadrata razpoznanega obraza in položaji izločitvenih kvadratkov na vsaki strani ob telesu

Na koncu z metodo *Dress* »preoblečemo« figuro telesa z barvo obraza t.j. kože. Za razpoznavo mej telesa smo uporabili kar pravilo zlatega reza in upoštevamo višino telesa brez glave v vrednosti petkratne razpoznane višine glave (kvadratno področje, ki je izhod metode FaceDetect). Naš algoritem detekcije telesa označuje iskani objekt z izključevanjem njegovega ozadja. To storimo z računanjem odločitvenih kvadratkov na levem in desnem robu zaznane figure telesa glede na zaznani obraz.

Velikost tega kvadratka smo določili fiksno na 15 pik. Za kvadratke in za celotno dolžino pasa smo se odločili zaradi konveksne oblike oklepajoče škatle v predelu komolcev. Tako smo pridobili boljšo zaznavo neposrednega ozadja v bližnji okolici osebe.

Postavitev navideznih modrih kvadratkov na levem in desnem robu vidimo na sliki 3. Dobro je viden tudi primer, ko je telo pomaknjeno k robu slike in za pravilno konveksno obliko okolico komolca ni prostora. V tem primeru smo zagotovili, da se kvadrati postavijo ob robu slike kolikor je to mogoče.

3.3 Priporočeni koraki za razvoj mobilne aplikacije za iPhone

Pri razvoju aplikacije iDelude smo spoznali in zapisali priporočene korake za razvoj mobilnih aplikacij za iPhone. Nato razmišljamo še o promociji aplikacij za iPhone.

Tabela 2: Priporočeni koraki za razvoj mobilne aplikacije za iPhone

<i>KORAK</i>	<i>OPIS</i>
1. Prenos in namestitev razvojnega okolja	iPhone SDK je na voljo v spletu [8]. Razvojni računalnik naj bo vrste Mac s CPE Intel (zaradi generiranja certifikata aplikacij).
2. Razvoj aplikacije in uporaba simulatorja	Aplikacijo razvijemo v Interface Builderju in drugih orodjih, simuliramo in razhroščimo itd.
3. Včlanitev v program za razvijalce	S prijavo v program za razvijalce iPhone pridobimo možnost prenosa zadnjih verzij orodij ter uvrstitve aplikacije v AppStore.
4. Pridobitev testnega certifikata	Testni certifikat lahko uporabimo za izvajanje aplikacije v mobilniku, licenčni sporazum omogoča razširjanje aplikacije za testne potrebe do 100 drugim uporabnikom.
5. Testiranje v telefonu in razhroščevanje	Aplikacijo testiramo v mobilniku in pri tem v realnem času analiziramo delovanje ter uporabljamo razhroščevalnik.
6. Pridobitev pravega certifikata in uvrstitev v AppStore	Aplikacijo testiramo s pravim certifikatom. Po verifikaciji aplikacijo naložimo v AppStore in s tem naročimo inženirjem družbe Apple, da pregledajo aplikacijo in odobrijo ali zavrnejo njeno uvrstitev v AppStore.
7. Nadaljevanje razvoja	Spremljamo odzive in želje uporabnikov, odpravljamo napake, smiselne zahteve pa implementiramo v naslednji verziji aplikacije.

Ker želimo prihraniti čas drugim razvijalcem, smo dokumentirali naše priporočene korake za razvoj mobilne aplikacije za iPhone dokumentirali. Navedeni so v tabeli 2.

Omenimo dva koristna nasveta. Praktični tržni model aplikacij za iPhone je imeti brezplačno verzijo aplikacije. V tej lahko dodamo oglase sistema AdMob [16], ki omogoča menjavo razmerja prikaza oglasov po naši izbiri. Smiselno je, da je eden od oglasov kar naš, ki vabi k nakupu plačljive verzije aplikacije z več možnostmi. Razmerje prikaza našega oglasa glede na druge lahko določimo npr. 50-50% (polovica časa v delujoči aplikaciji se prikazuje naš oglas, polovico pa vsi drugi).

Koristen je tudi dobro oblikovan lasten blog (spletni dnevnik) aplikacije. Glavni cilj je vabilo k nakupu plačljive verzije aplikacije. Ideje se nahajajo v periodiki [17] in drugje.

Še zadnja opomba: družba Apple je začela februarja 2010 umikati aplikacije v AppStore, če vsebujejo goloto ali možna erotična namigovanja ipd. O razlogih ne bi razmišljali na tem mestu, se je pa dobro tega zavedati pri razvoju aplikacij za razpoznavo človeške kože npr. obraza ali telesa. Preveliko prikazane kože v aplikaciji lahko pomeni zavrnitev uvrstitve aplikacije v AppStore in s tem nedosežene (poslovne) načrte. Načrtujte aplikacijo tako, da je možno omenjene sporne dele umakniti ali zmanjšati na – za Apple – sprejemljivo mero. Podobne pogoje postavlja tudi Google v tržnici Android Market.

4. ZAKLJUČEK

Mobilniki iPhone so zgled za druge izdelovalce, ki v svoje mobilnike dodajajo zaslone, občutljive na dotik, hitre uporabniške grafične vmesnike z ikonami in druge praktične lastnosti. S tem spreminjajo mobilnike v bolj uporabne in intuitivne naprave.

iPhone že z modelom 2G omogoča dovolj hitro obdelavo slik za uporabo v slikovnih aplikacijah. Model 3GS poskušajo posnemati tudi drugi izdelovalci. Za razpoznavo obraza so od leta 2009 dostopni zmogljivi mobilniki za prijetno uporabniško izkušnjo.

Upamo, da bodo ta prispevek in nasveti za razvoj aplikacij za iPhone prihranili čas drugim razvijalcem ter jih vzpodbudili k izdelavi novih aplikacij za obdelavo slik. Morda bi bilo dobro uvesti dodatni predmet o razvoju mobilnih aplikacij v študijske programe računalništva. Nove aplikacije so namreč lahko tudi začetek podjetniške poti študentov.

LITERATURA

1. http://www.apek.si/sl/13_implementacijsko_porocilo_evropske_komisije
13. Implem. poročilo Evr. komisije o razvoju trga in regulacije elektr. komunikacij.
2. http://www.itu.int/net/pressoffice/press_releases/2010/06.aspx
ITU sees 5 billion mobile subscriptions globally in 2010, Barcelona, 15.02.2010.

3. <http://www.dlib.si/v2/StreamFile.aspx?URN=URN:NBN:SI:doc-EIU9ZXUP&id=6bc13393-e640-494c-8681-d2db4514b5b6&type=PDF>
Elektronska kopija glasila Ljubljanski zvon, 1881, str. 52-53, 122-123.
4. <http://search.japantimes.co.jp/cgi-bin/nb20070626a1.html>
New cell phone services tap image-recognition technologies, 26.06.2007.
5. <http://www.openmobilealliance.org/Technical/wapindex.aspx>
WAP Forum conformance release
6. http://www.eurotechnology.com/market_reports/imode/faq.shtml
The unofficial independent i-Mode FAQ
7. <http://www.symbian.org>
Innovation in HD, 3D, and Apps, Now Available to All
8. <http://developer.apple.com/iphone/program/>
iPhone Developer Program
9. <http://developer.apple.com/Mac/library/documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>
Introduction to The Objective-C Programming Language
10. http://www.comscore.com/Press_Events/Press_Releases/2009/11/Touchscreen_Mobile_Phone_Adoption_Grows_at_Blistering_Pace_in_U.S._During_Past_Year
Touchscreen Mobile Phone Adoption Grows at Blistering Pace in U.S.
11. <http://www.apple.com/iphone/iphone-3gs/app-store.html>
AppStore
12. <http://gigaom.com/2010/01/12/the-apple-app-store-economy/>
The Apple App Store Economy
13. <http://www.apple.com/pr/library/2010/01/05appstore.html>
Apple's App Store Downloads Top Three Billion
14. <http://www.anandtech.com/gadgets/showdoc.aspx?i=3579>
The iPhone 3GS Hardware Exposed & Analyzed
15. <http://www.idelude.com>
Spletna stran aplikacije iDelude
16. <http://www.admob.com/>
Oglaševalski sistem AdMob
17. <http://www.netmag.co.uk/zine/latest-issue/issue-198>
.net, februar 2010, str. 92-95.

INDUSTRIJSKE APLIKACIJE

VPLIV UČNE MNOŽICE NA USPEŠNOST RAZPOZNAVANJA OBRAZOV

Domen Mongus, Damjan Zazula

Fakulteta za elektrotehniko, računalništvo in informatiko

E-pošta: domen.mongus@uni-mb.si

POVZETEK: *Ustrezna izbira značilnosti obrazov je ključna za učinkovitost postopkov njihove razpoznavne. Ker preprosta izbira značilnic običajno ne zagotavlja ustrezne zanesljivosti, se snovalci tovrstnih sistemov pogosto naslanjajo na pristope, ki definirajo nabor značilnic na osnovi statistične analize učne množice. V članku predstavljamo vpliv učne množice na učinkovitost razpoznavne obrazov dveh izmed najpogosteje uporabljenih statistično podprtih pristopov PCA in LDA, ter pristopa, ki temelji na Gaborjevi valčni transformaciji in pri katerem izberemo značilnosti glede na človeško fiziognomijo. Vse tri postopke podrobno opišemo in z rezultati pokažemo, da sta oba statistična pristopa občutno zanesljivejša, a hkrati zelo odvisna od izbora učnih vzorcev, medtem ko ta nima posebnega vpliva na postopek, ki temelji na valčni transformaciji z Gaborjevimi valčki.*

1. UVOD

Razpoznavna obrazov je, navkljub številnim že obstoječim aplikacijam, še vedno zelo atraktivno raziskovalno področje. Visoke zahteve tržišča, kjer danes močno prevladujejo težnje po varnih sistemih, v ospredje postavljajo zanesljivo identifikacijo oseb s pomočjo biometričnih podatkov. Razpoznavna obraza je pri tem prav gotovo ena izmed elegantnejših rešitev, saj velja za cenen in uporabnika prijazen način biometrične identifikacije. Ugotovitve na tem področju pa so pogosto uporabljene tudi za industrijsko razpoznavanje vzorcev [1]. V obeh primerih se ključ učinkovitosti skriva v uspešno definiranih značilnostih in njihovi predstavitvi v bazi modelov [2]. Načinov za to je več [2,3], vseeno pa je največkrat uporabljena predstavitev modelov s tako imenovanimi vektorji značilnosti (ang. feature vector). Značilnost pri tem predstavlja pomembno lastnost, na osnovi katere je mogoče obraze med seboj razlikovati. V najbolj preprostem primeru lahko kot značilnice obravnavamo barvo oči, barvo kože ali obliko obraza, a na žalost takšne preproste značilnosti običajno ne zagotavljajo učinkovite razpoznavne.

Če se pri izločanju značilnosti opremo na statistične lastnosti izbrane populacije obrazov, je lahko končni nabor značilnic zelo odvisen od podmnožice obrazov, ki jih vključimo v postopek. Ena izmed najširše uporabljenih tovrstnih metod je analiza glavnih komponent (ang. principle component analysis – PCA). Glavne komponente močno zmanjšajo dimenzije iskalnega prostora, hkrati pa tudi zelo vplivajo na izločanje značilnic in s tem

na bolj ali manj uspešno razpoznavanje. V pričujoči raziskavi nas je zanimalo, kako vpliva na tvorbo glavnih komponent izbor učne množice obrazov in kako je od tega izbora nato odvisna uspešnost razpoznavanja. Zato smo vzeli standardno bazo obrazov [4] in analizirali njihovo razpoznavanje s tremi različnimi pristopi: z grafi značilnosti in valčno transformacijo, z linearno diskriminanto analizo (ang. linear discriminant analysis – LDA) in z omenjeno PCA. Primerjali smo uspešnost razpoznavanja in ugotavljali odvisnost od izbora učne množice obrazov.

V poglavju 2 podrobneje predstavljamo vse tri preizkušene postopke za razpoznavanje obrazov, v poglavju 3 pa sledi predstavitev rezultatov, dobljenih s temi postopki. V poglavju 4 poudarimo najpomembnejše ugotovitve.

2. METODE ZA RAZPOZNAVANJE OBRAZOV

Preden se lotimo razpoznavanja obrazov, moramo slike normalizirati [5,6]. Ta postopek poskrbi, da sta kontrastna in ploskovna ločljivost primerjanih slik izenačeni. Normalizacija slik, ki je tukaj ne bomo podrobneje opisovali, nas intuitivno vodi k možnosti razpoznave obrazov glede na graf značilnosti. Zato v nadaljevanju najprej predstavljamo razpoznavanje obrazov, ki temelji na zaznavi poglavitnih značilnosti obraza in njihovi primerjavi z uporabo Gaborjeve valčne transformacije. Nato se posvetimo še dvema postopkoma, ki temeljita na linearni transformaciji slike v podprostor značilnosti, imenovanima PCA in LDA.

2.1 Graf značilnic z Gaborjevo valčno transformacijo

Pri tem pristopu najprej na obrazu določimo ključne točke za razpoznavo. Značilnosti v teh točkah nato sestavimo v graf in izvedemo primerjavo med grafi. V našem primeru smo v ta namen uporabili nekoliko modificiran pristop, predstavljen v [7], ki uvaja tako imenovan vektor valčnih koeficientov (ang. jet) kot nosilce značilnosti obraza v določeni točki. Valčno transformacijo pri tem izvajamo z Gaborjevimi valčki, ki so znani po svoji robustnosti na spremembe osvetlitve in majhne spremembe v zamiku. V našem primeru uporabimo 40 različnih valčkov (5 različnih frekvenc v 8 različnih smereh).

Glede na fiziognomijo obrazov [8] izberemo za razpoznavo 4 točke: po eno na levi in desni ličnici, na čelu in nad brado. V vsaki izmed izbranih točk nato izračunamo prej opisani vektor valčnih koeficientov glede na 40 Gaborjevih valčkov, kot to prikazuje slika 1. Ker lahko majhne spremembe položaja hitro povzročijo fazni zamik, dejansko izvedemo valčno transformacijo za izbrano točko v njeni okolici 7×7 in tako predstavimo obraz v bazi modelov z 49 vektorji valčnih koeficientov, in to za vsako izmed štirih izbranih točk. V procesu razpoznave nato vektorje valčnih koeficientov izračunamo le v izbranih točkah testnega vzorca in izvedemo najboljše prilagajanje grafa z modelom v bazi. Razdaljo med dvema vektorjema valčnih koeficientov pri tem merimo z L^1 ali manhattansko razdaljo, ki jo določa naslednja enačba:

$$d(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^n |p_i - q_i|, \quad (1)$$

kjer $d(p, q)$ predstavlja razdaljo med vektorjema p in q .

Gaborjevi valčki

Vhodna slika

Vektor valčnih koeficientov



Slika 1: Tvorba vektorja valčnih koeficientov za okolico ključnih točk s pomočjo valčne transformacije z Gaborjevimi valčki, pri čemer so ključne točke izbrane glede na fiziognomijo obrazov

2.2 PCA

PCA predstavlja statistično orodje za identificiranje vzorcev v podatkih. Omogoča predstavitev različnih naborov podatkov na način, da poudari razlike med njimi. Ker je sicer iskanje vzorcev v večdimenzionalnih podatkih zahtevno, PCA predstavlja pomembno orodje za njihovo analizo in danes velja za enega najpopularnejših pristopov pri razpoznavanju obrazov [9,10,11], pa tudi vzorcev na splošno [12,13]. Izvedbo PCA za razpoznavanje obrazov lahko opišemo v šestih korakih:

Korak 1: Vhod v postopek predstavljajo normalizirane slike. Te slike najprej vektoriziramo, tako da zložimo vrstice (ali stolpce) eno za drugo. V našem primeru je dimenzija vhodnih slik 256×256 pikslov, zato ima vsak vektor 65536 komponent. Takšni vektorji sedaj definirajo vektorski prostor, ki ima toliko dimenzij, kolikor je število vhodnih slik. Zaradi boljše predstavitve pa lahko vektorje sestavimo v matriko, v kateri vsak izmed njih predstavlja svoj stolpec.

Korak 2: Od vhodnega nabora podatkov nato odštejemo povprečje, v vsaki dimenziji posebej. To praktično pomeni, da od vsakega vektorja (vhodne slike) odštejemo povprečno vrednost njegovih komponent, zato ima dobljeni vektorski prostor sedaj izhodišče v točki $(0,0,\dots,0)$.

Korak 3: Dobljenemu vektorskemu prostoru zmanjšamo dimenzionalnost tako, da najprej izrazimo medsebojno odvisnost baznih vektorjev s kovariančno matriko

$$C_{i,j} = \frac{\sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}}_i)(\mathbf{x}_j - \bar{\mathbf{x}}_j)}{(N_{PCA} - 1)}, \quad (2)$$

kjer $C_{i,j}$ označuje (i,j) -ti element kovariančne matrike, \mathbf{x}_i in \mathbf{x}_j predstavljata vektorja, med katerima v danem koraku izračunavamo kovarianco, $\bar{\mathbf{x}}_i$ in $\bar{\mathbf{x}}_j$ njuni povprečni vrednosti, ki sta zaradi koraka 2 v našem primeru vedno enaki 0 in N_{PCA} dimenzionalnost vektorjev.

Korak 4: Kovariančno matriko zapišemo z njenimi lastnimi vrednostmi in pripadajočimi lastnimi vektorji. Ker lastni vektorji predstavljajo soodvisnost podatkov glede na dano dimenzijo, jih lahko interpretiramo kot značilnosti obraza, glede na katere so si vzorci v učni množici podobni oziroma glede na katere se razlikujejo (slika 2). Čeprav smo tako tvorili vektorski podprostor, v katerem je mogoče izvesti učinkovito razpoznavo obrazov, pri tem običajno ne uporabimo vseh lastnih vektorjev, saj so tisti, ki pripadajo največjim lastnim vrednostim, pogosto podvrženi vplivom osvetlitev in jih zato raje izločimo [11]. V ta namen najprej uredimo vektorje glede na padajoče lastne vrednosti in izločimo začetna dva vektorja. Preostali lasni vektorji tvorijo vektorski podprostor E , in jih predstavimo v obliki matrike, kjer vsak vektor predstavlja svoj stolpec.



Slika 2: Lastni vektorji, kjer je opazen vpliv osvetlitev na vektorje z višjimi lastnimi vrednostmi (zgornja vrsta), in značilnosti, ki jih predstavljajo (spodnji dve vrsti).

Korak 5: V tako dobljeni vektorski podprostor sedaj po enačbi (3) preslikamo še vhodne slike in s tem zgradimo bazo modelov za znane obraze:

$$\mathbf{y}_i = \mathbf{E}^T \cdot \mathbf{x}_i, \quad (3)$$

ker \mathbf{y}_i predstavlja projekcijo vhodne slike \mathbf{x}_i , v vektorski podprostor PCA definiran z reducirano matriko lastnih vektorjev kovariančne matrike E .

Korak 6: Za uspešno razpoznavanje obrazov moramo v zadnjem koraku definirati še način merjenja razdalje med slikami, projiciranimi v vektorski podprostor PCA. V našem primeru to izvedemo s tako imenovano normalizirano evklidsko razdaljo ali Mahalanobisovo razdaljo:

$$d(\mathbf{y}_i, \mathbf{y}_j) = \sqrt{\sum_{n=1}^{N_{PCA}} \frac{(y_{in} - y_{jn})^2}{\sigma_n^2}}, \quad (4)$$

kjer $d(\mathbf{y}_i, \mathbf{y}_j)$ predstavlja razdaljo med vektorjema \mathbf{y}_i in \mathbf{y}_j , σ_n pa standardni odklon, ki ga v našem primeru nadomestimo z lastno vrednostjo, povezano z n -tim lastnim vektorjem.

2.3 LDA

Podobno kot PCA tudi LDA omogoča klasifikacijo podatkov, pri čemer pa LDA ne išče vzorcev v vhodnih podatkih in podatkov posledično tudi ne spreminja. LDA tvori klasifikacijske meje med razredi, pri tem pa temelji na maksimiranju razmerja med medrazredno varianco in varianco znotraj posamičnega razreda [14]. Oba postopka lahko torej izvajamo zaporedno in takšen pristop lahko opišemo v petih korakih:

Korak 1: Vhod v postopek LDA predstavljajo vektorji slik, že projiciranih v vektorski podprostor PCA. Vsakemu vektorju dodamo še komponento, s katero povemo, kateremu razredu dani vektor pripada.

Korak 2: Ker lahko v postopku LDA predstavimo posamičen razred z več vzorci, nato izračunamo povprečje znotraj vsakega razreda posebej ($\mu_1, \mu_2, \dots, \mu_n$) in povprečje med vsemi razredi μ . Povprečje vseh razredov lahko preprosto izrazimo iz povprečja posamičnih razredov z naslednjo enačbo:

$$\mu = \sum_{i=1}^n p_i \mu_i, \quad (5)$$

kjer p_i predstavlja verjetnost posamičnega razreda, določeno kar kot $p_i = \frac{1}{N_R}$ za vse razrede, pri čemer N_R predstavlja število vseh razredov.

Korak 3: Iz nabora podatkov lahko sedaj izračunamo po enačbi (6) dve matriki razpršenosti (ang. scatter matrix), kjer \mathbf{S}_w opisuje kovarianco znotraj posamičnega razreda R_j ; $1 \leq j \leq N_R$, \mathbf{S}_b pa razpršenost med razredi. Kadar za R_j obstaja več vzorcev \mathbf{y}_i^j ; $1 \leq i \leq M_j$, lahko matriko \mathbf{S}_b razumemo tudi kot opis kovariance med povprečnimi vektorji μ_j razredov.

$$\mathbf{S}_w = \sum_{i=1}^{N_R} \sum_{j=1}^{M_j} (\mathbf{y}_i^j - \mu_j)(\mathbf{y}_i^j - \mu_j)^T, \quad (6)$$

$$\mathbf{S}_b = \sum_{j=1}^n (\mu_j - \mu)(\mu_j - \mu)^T.$$

Korak 4: Optimizacijski kriterij LDA določa razmerje med \mathbf{S}_w in \mathbf{S}_b . Ker v našem primeru uporabljamo Fisherjev LDA, lahko optimizacijski kriterij zapišemo kot:

$$J(\mathbf{W}) = \frac{\mathbf{W}^T \mathbf{S}_b \mathbf{W}}{\mathbf{W}^T \mathbf{S}_w \mathbf{W}}, \quad (7)$$

pri čemer iščemo matriko \mathbf{w} tako, da maksimiramo vrednost $J(\mathbf{W})$. Možnih načinov za to je več, v našem primeru pa maksimiramo razmerje $|\det \mathbf{S}_b| / |\det \mathbf{S}_w|$ [15], saj je bilo že pokazano da lahko \mathbf{W} dobimo kar z lastnimi vektorji matrike $\mathbf{S}_b / \mathbf{S}_w$, če \mathbf{S}_w ni singularna matrika [16]. Čeprav v realnih primerih \mathbf{S}_w večinoma ni singularna, lahko njeno nesingularnost v vseh primerih zagotovimo tako, da za vsak razred uporabimo vsaj dva učna vzorca [15]. V vsakem primeru pa nato matriko \mathbf{W} še normaliziramo.

Korak 5: Matrika \mathbf{w} sedaj predstavlja vektorski podprostor, v katerem lahko optimalno klasifikaciramo glede na dani nabor učnih vzorcev. Formalno lahko postopek razpoznave v prostoru LDA definiramo z naslednjo enačbo:

$$\mathbf{y}_i = \mathbf{E}^T \mathbf{x}_i, \quad (8)$$

$$\mathbf{z}_i = \mathbf{W}^T \mathbf{y}_i,$$

kjer \mathbf{E} predstavlja bazo podprostora PCA, \mathbf{W} bazo podprostora LDA, \mathbf{y}_i sliko \mathbf{x}_i projicirano v podprostor PCA in \mathbf{z}_i projekcijo \mathbf{y}_i v podprostor \mathbf{W} . Vzorce primerjamo glede na evklidsko ali L^2 razdaljo, podano kot:

$$\mathbf{d}(\mathbf{z}_i, \mathbf{z}_j) = \sqrt{(z_{i1} - z_{j1})^2 + (z_{i2} - z_{j2})^2 + \dots + (z_{iN_{LDA}} - z_{jN_{LDA}})^2}, \quad (9)$$

kjer $\mathbf{d}(\mathbf{z}_i, \mathbf{z}_j)$ predstavlja razdaljo med N_{LDA} -dimenzionalnima vektorjema \mathbf{z}_i in \mathbf{z}_j .

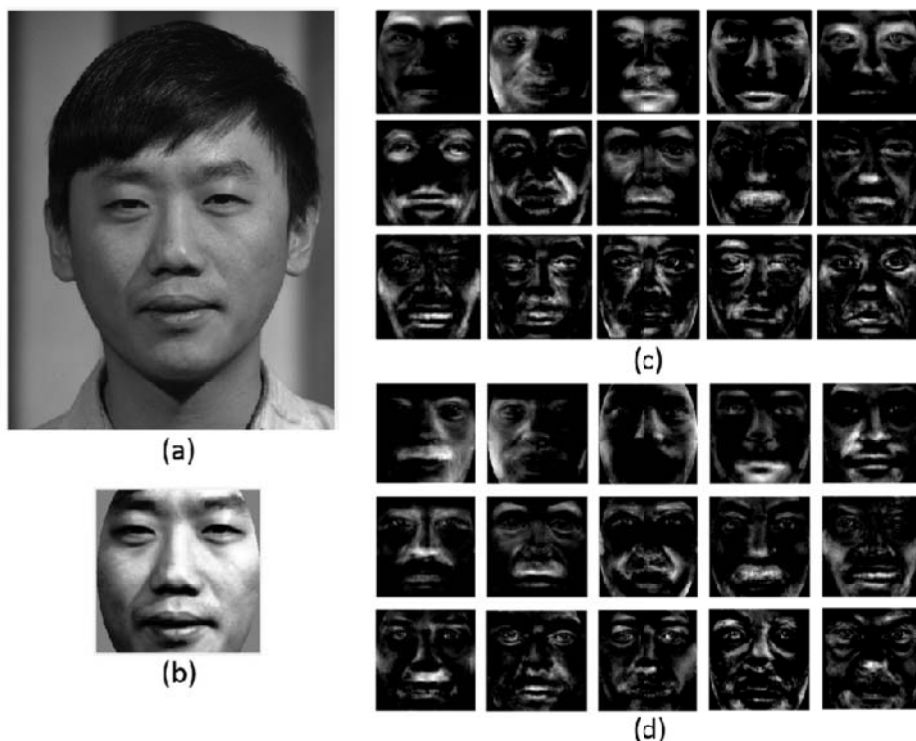
3. REZULTATI RAZPOZNAVANJA

Za testiranje smo izbrali standardno bazo slik FERET [4], iz katere smo ustvarili tri baze modelov. V prvo bazo smo vključili 10 naključnih posameznikov, v drugo 20 in tretjo 40, pri čemer smo vsakega posameznika predstavili z enim učnim vzorcem. Uspešnost predstavljenih postopkov smo nato preverjali s testno množico, v katero smo vključili 5 testnih vzorcev vsakega posameznika. V tabeli 1 je v odstotkih predstavljena povprečna uspešnost postopkov, s standardnim odklonom pa je opisano odstopanje od povprečja glede na posamične razrede (osebo).

Tabela 1: Uspešnost razpoznavanja obrazov z Gaborjevimi valčki, PCA in LDA.

<i>Pristop</i>	<i>Baza 1 (10×5 slik)</i>		<i>Baza 2 (20×5 slik)</i>		<i>Baza 3(40×5 slik)</i>	
	Uspešnost razpoznavanja	Standardni odklon	Uspešnost razpoznavanja	Standardni odklon	Uspešnost razpoznavanja	Standardni odklon
Gaborjevi valčki	68 %	0.9940	56 %	1.0940	56 %	1.3940
PCA	80 %	0.5164	88 %	0.9987	82 %	1.1873
LDA	77 %	0.4830	85 %	0.9679	79 %	1.4118

Prvi, mogoče presenetljiv rezultat je dejstvo, da sta postopka PCA in LDA uspešneje razpoznavala osebe v bazi 2 kot v bazi 1. Glavni razlog za to se skriva v dejstvu, da pri majhnem številu učnih vzorcev oba postopka hitro postaneta preveč determinirana (ang. overdeterminate) [11, 14]. Ker je postopek PCA del LDA, je ta efekt pri postopku LDA še posebej opazen in zato je njegova učinkovitost nižja. Ker je očitno, da statistične metode razpoznavanja potrebujejo več učnih vzorcev, da lahko izluščijo več pomembnih značilnosti, bi lahko pričakovali, da bosta oba postopka v primeru baze 3 najuspešnejša. Vseeno pa ni tako. Vidimo celo nepričakovano visok padec učinkovitosti. Pri analizi tega pojava pa smo naleteli na zanimivo dejstvo. Ko smo iz učne množice izločili specifičnega posameznika (slika 3), se je učinkovitost razpoznavanja s PCA povečala celo na 90 % (s standardnim odklonom 1,0671), pri LDA pa na 87 % (s standardnim odklonom 0,9901). Na slikah 3c in 3d je prikazan vpliv tega učnega vzorca na celoten projekcijski podprostor. Opazimo lahko, da značilnosti v prvem primeru (slika 3c) niso tako izrazite in nanje vpliva porazdelitev svetlobe, ki je opazna tudi pri lastnih vektorjih z višjimi lastnimi vrednostmi. Običajno tak učinek pričakujemo le pri nekaj lastnih vektorjih, povezanih z najvišjimi lastnimi vrednostmi (slika 3d).



Slika 3: Vpliv učnega vzorca na lastne vektorje: (a) prikazuje sliko specifičnega posameznika, (b) prikazuje to sliko v normalizirani obliki, (c) prikazuje prvih 15 lastnih vektorjev, ko je izbrani posameznik vključen v učno množico in (d) prikazuje istoležne lastne vektorje, ko omenjenega posameznika ni med učnimi vzorci.

Izbira učne množice pa nima posebnega vpliva na razpoznavo z Gaborjevimi valčki. Razvidno je sicer, da se zanesljivost tovrstne razpoznave znižuje z večanjem števila testnih oseb, kar sicer kaže na omejene zmožnosti razločevanja predstavljenega postopka, a to je pričakovano. Poleg majhnega izbora točk, v katerih tvorimo vektorje valčnih koeficientov, predstavlja glavno težavo našega pristopa merjenje razdalje. S preprostimi metrikami tudi v tem primeru ni mogoče doseči visoke stopnje razpoznave, zato so različni avtorji večkrat predlagali pristop z Gaborjevimi valčki, ki mu sledi klasifikacija z eno izmed statističnih metod, kot sta PCA ali LDA [17, 18].

V nekaterih realnih aplikacijah pa naletimo tudi na primere, ko iskane osebe ni v bazi modelov in jo moramo zato označiti kot neznano. Zato pri klasifikaciji obrazov ne iščemo le najbližjega razreda za dani testni vzorec, ampak nas zanima tudi razlika (razdalja) med njim in njegovim najbližjim razredom. Če je ta razdalja večja od določenega praga, moramo vzorec identificirati kot neznan. Za poskus smo v testno bazo 3 vključili še 100 neznanih vzorcev (slik neznanih oseb), pri čemer smo razpoznavni prag določili glede na aritmetično sredino med povprečno razdaljo pravilno razpoznanih vzorcev in povprečno razdaljo neznanih vzorcev. Rezultate tega testiranja prikazuje tabela 2, kjer TP (ang. true positive) opisuje število pravilno razpoznanih vzorcev, FN (ang. false negative) število pravilno odkritih neznanih vzorcev, FP (ang. false positive) število napak, kjer neznan vzorec prepoznamo kot osebo iz baze, in TN (ang. true negative) število napak, kjer znano osebo označimo kot neznano.

Tabela 2: Uspešnost postopkov za razpoznavanje obrazov v razširjeni bazi 3.

<i>Pristop</i>	<i>TP</i>	<i>FN</i>	<i>FP</i>	<i>TN</i>
Gaborjevi valčki	50 %	81 %	19 %	6 %
PCA	80 %	96 %	4 %	2 %
LDA	76 %	94 %	6 %	3 %

Iz rezultatov, predstavljenih v tabeli 2 lahko hitro ugotovimo, da že preprosta pragovna omejitev omogoča učinkovito identificiranje neznanih vzorcev, hkrati pa ne poslabša učinkovitosti razpoznavanja. Še zlasti je to očitno pri metodah PCA in LDA, kar dodatno potrjuje omenjeno dejstvo o vplivu učnih vzorcev na učinkovitost razpoznavanja. Običajno je namreč razdalja med neznanim vzorcem in njemu najbližjim razredom celo večja od razdalje med znanim vzorcem in njemu najbolj oddaljenim razredom. Zato se pojavlja večja napaka pri razpoznavi znanih oseb kot pri identificiranju neznanih. Našteta dejstva potrjujejo, da je izbira učnih vzorcev zelo pomembna za učinkovitost statističnih klasifikacijskih postopkov, kot sta PCA in LDA. Posledično velja to tudi za metode, ki temeljijo na Gaborjevimi valčki, ki jim sledi klasifikacija s kakšnim izmed statističnih pristopov.

4. ZAKLJUČEK

V članku predstavljamo postopke za razpoznavo obrazov, ki temeljijo na Gaborjevi valčni transformaciji ter metodah PCA in LDA. Predstavljene postopke smo testirali z različnimi testnimi množicami in pri tem posebej preučevali vpliv učnih vzorcev na njihovo učinkovitost. Prva, pravzaprav pričakovana ugotovitev je, da se uspešnost postopkov PCA in LDA izboljšuje s povečanjem učne množice. Ker postopka temeljita na statističnih zakonitostih, zahtevata večji nabor učnih vzorcev, saj tako povečamo njihovo reprezentativnost. Še več, pri postopkih PCA in LDA je pomembna tudi izbira lastnih vektorjev, s katerimi tvorimo projekcijski podprostor. Nekaj lastnih vektorjev, povezanih z največjimi lastnimi vrednostmi, namreč poudarja porazdelitev svetlobe na vhodnih slikah in jih je zato smiselno izločiti. Smiselno je preučiti tudi vpliv posameznega učnega vzorca na učinkovitost postopkov. Pokazali smo, kako lahko en učni vzorec občutno spremeni projekcijski prostor in zniža učinkovitost postopkov PCA in LDA. Hkrati pa lahko s projekcijo v podprostor lastnih vektorjev razmeroma enostavno identificiramo neznanih primerke. Ker ti niso bili vključeni v tvorbo projekcijskega podprostora, njihove značilnosti niso poudarjene in je njihova oddaljenost od definiranih razredov občutno večja. Čeprav se večini neželenih učinkov zaradi sestave učne množice uspemo izogniti v postopku, ki temelji na valčni transformaciji z Gaborjevimi valčki, pa je njegova uspešnost razpoznavanja kljub temu občutno nižja kot pri metodah PCA in LDA.

LITERATURA

1. C. Liu, H. Wechsler (1999) Comparative Assessment of Independent Component Analysis (ICA) for Face Recognition, *Second International Conference on Audio- and Video-based Biometric Person Authentication*, Washington, str. 211-216.
2. R. Jain, R. Kasturi, B. G. Schinck (1995), *Machine Vision*, New York: McGraw-Hill.
3. H. F. Liao, K. P. Seng, Y. W. Wong, L.-M. Ang (2007), New Parallel Models for Face Recognition, *International Conference on Computational Intelligence and Security (CIS 2007)*, Washington: IEEE Computer Society, str. 306-309.
4. P. J. Phillips, H. Moon, S. A. Rizvi, P. J. Rauss (2000), The FERET Evaluation Methodology for face-recognition algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22(10), str. 1090-1103.
5. Y. Zana, R. M. Cesar (2006), Face recognition based on polar frequency features, *ACM Transactions on Applied Perception*, vol. 3(1), str. 62-82.
6. M. Lee, C. H. Park (2008), An efficient image normalization method for face recognition under varying illuminations, *Proceeding of the 1st ACM international conference on Multimedia information retrieval*, New York: ACM Press, str. 128-133.

7. L. Wiskott, J.-M. Fellous, N. Krueger, C. von der Malsburg (1997), Face Recognition by Elastic Bunch Graph Matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19(7), str. 775-779.
8. W. Zhao, R. Chellappa, A. Rosenfeld, P.J. Phillips (2003), *Face Recognition: A Literature Survey*, ACM Computing Surveys, vol. 35(4), str. 399-458.
9. M. Turk, A. Pentland (1991), Eigenfaces for Recognition, *Journal of Cognitive Neuroscience*, vol. 3(1), str. 71-86.
10. A. Pentland, B. Moghaddam (1994), T. Starner, View-Based and Modular Eigenspaces for Face Recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle: IEEE Computer Society, str. 84-91.
11. H. Moon, P.J. Phillips (2001), Computational and Performance aspects of PCA-based Face Recognition Algorithms, *Perception*, vol. 30, str. 303-321.
12. A. Ferraz, E. Esposito, R.E. Bruns and N. Durán (1998), The use of principal component analysis (PCA) for pattern recognition in Eucalyptus grandis wood biodegradation experiments, *World Journal of Microbiology and Biotechnology*, vol 14(4), str. 487-490.
13. R. Saegusa and S. Hashimoto (2005), Pattern Recognition Using a Nonlinear PCA, *Proceedings of GVIP 05 Conference*, Cairo: CICC, str. 73-80.
14. W. Zhao, A. Krishnaswamy, R. Chellappa, D.L. Swets, J. Weng (1998), Discriminant Analysis of Principal Components for Face Recognition, *Face Recognition: From Theory to Applications*, H. Wechsler, P.J. Phillips, V. Bruce, F.F. Soulie, T.S. Huang (eds.), Berlin: Springer-Verlag, str. 73-85.
15. A. M. Martinez, A.C. Kak (2001), PCA versus LDA, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23(2), str. 228-233.
16. R.A. Fisher (1938), The Statistical Utilization of Multiple Measurements, *Annals of Eugenics*, vol. 8, str. 376-386.
17. C. Liu (2004), Gabor-Based Kernel PCA with Fractional Power Polynomial Models for Face Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligenc*, vol. 26(5), str. 572-581.
18. W. Li, W. Cheng (2008), Face Recognition Based on Adaptively Weighted Gabor-LDA, *Proceedings of the 2008 Fourth International Conference on Natural Computation*, Washington: IEEE Computer Society, vol. 4, str. 130-134.

IDENTIFIKACIJA IN KOREKCIJA NAPAČNO VZPOSTAVLJENIH KORESPONDENČNIH TOČK ZA VELIKE ODMIKE DVEH KAMER

Peter Podbreznik¹, Božidar Potočnik²

¹ Center za gradbeno informatiko, Fakulteta za gradbeništvo

² Fakulteta za elektrotehniko, računalništvo in informatiko

Univerza v Mariboru

E-pošta: {peter.podbreznik, bozo.potocnik}@uni-mb.si

POVZETEK: *Naš algoritem ASIFT-SH za vzpostavljanje korespondence za velike odmike dveh kamer, temelječ na algoritmu ASIFT, segmentaciji in homografiji v splošnem dobro izračuna korespondenčno točko na drugi sliki za poljubno izbrano točko na prvi sliki. Identificirali smo primere, ko korespondenčne točke ni mogoče izračunati, ali pa je njena lokacija določena napačno. Predstavili bomo način detekcije napačno vzpostavljenih korespondenčnih točk in ponovno določanje njihove prave lokacije s pomočjo 3D-rekonstrukcije.*

1. UVOD

Pri avtomatskem spremljanju poteka gradnje s pomočjo kamer [7] je potrebno identificirati posamezne gradbene elemente, s čimer omogočimo realno časovno spremljanje procesa gradnje in avtomatsko ugotavljanje odstopanj od terminskega plana [9]. Eden od problemov avtomatskega razpoznavanja je prekrivanje gradbenih elementov, zato je potrebno zajemanje slik zagotoviti z več fiksno nameščenimi kamerami. Identifikacija istega elementa iz več slik gradbišča zahteva poznavanje relacij med pogledi kamer. Kot prvo je potrebno zagotoviti zadostno množico začetnih korespondenčnih točk. Obstajajo različne metode za avtomatsko vzpostavljanje korespondence, ki večinoma temeljijo na iskanju podobnosti značilk. Najbolj uporabljeni metodi za vzpostavitev korespondence stereo pogledov sta relaksacijska metoda [12] in metoda zasnovana z uporabo singularnega razcepa (SVD) [6]. Obe delujeta zanesljivo le za majhne odmike kamer. Metoda, ki temelji na singularnem razcepu, je bila kasneje tudi nadgrajena z uporabo značilk SIFT [3] in objavljena v [1].

Slike gradbišča zajemamo z več kamerami, pri čemer se pogledi kamer lahko tudi močno razlikujejo. Za uspešno postavitve korespondence med njimi je zato potrebno na slikah poiskati značilke, ki so invariantne na zasuk, premik, skaliranje in intenziteto pikselov. Potreba po takšnem postopku se je najprej pojavila pri iskanju izbranih predmetov po slikovnih podatkovnih bazah [10], kjer so iskani predmeti na slikah zasukani, skalirani in premaknjeni. V [3] je opisana značilka SIFT, ki je še dodatno invariantna na intenziteto

piksla. Omenjeno značilko je isti avtor uporabil v [4] za določitev korespondence med dvema močno različnima pogledoma. Ker značilka SIFT nima vseh afinih invariantnih značilnosti je avtor Morel predlagal značilko ASIFT in jo prvič predstavil v [11]. Zaradi robustnosti značilke ASIFT so jo uporabili v metodi za vzpostavitev korespondence [5], kjer so pogledi kamer podvrženi zlasti zasuku.



Slika 1: Začetne korespondenčne točke izračunane z metodo ASIFT: na levi sliki pogled prve kamere in na desni sliki pogled druge kamere. Bele črte na sliki prikazujejo povezave med korespondenčnimi točkami.

2. METODA ASIFT-SH

Zagotavljanje dovolj dobre korespondence med točkami slik različnih pogledov in njihovo zadostno število je osnovni problem računalniškega vida. Naša metoda ASIFT-SH sestoji iz štirih korakov [14]:

1. Določimo začetne korespondenčne točke na slikah obeh pogledov z metodo [5], ki uporablja značilke ASIFT. Le-te so predstavljene kot invariantne značilke. Primer takšnega iskanja prikazuje slika 1. Za dobljene začetne korespondenčne točke izračunamo temeljno matriko F in za vse točke iz množice začetnih korespondenčnih točk, s pomočjo temeljne matrike, izračunamo epipremico. Izločimo samo tiste korespondenčne pare, katerih točka odstopa več kot $2,5 \cdot \alpha$ od epipremice, kjer je α standardni odklon odstopanja vseh točk množice od epipremice.
2. Združevanje začetnih korespondenčnih točk v podmnožice, ki predstavljajo koplanarne točke v prostoru. To zagotovimo s pomočjo ustrezne segmentacije. Predpostavili smo, da so posamezni deli gradbenih elementov na sliki segmentirani tako, da segmentirane regije predstavljajo ravne ploskve objekta (primer je prikazan na sliki 2). Odvisno od števila regij segmentacije, se pripravi ustrezno število podmnožic. Začetna korespondenčna točka postane članica ustrezne podmnožice, če leži znotraj regije, pri čemer lahko posamezna korespondenčna točka postane članica tudi drugih podmnožic, v kolikor meji na sosednje regije. Ustrezno združene začetne korespondenčne točke v podmnožice omogočajo izračun lokalnih homografij.
3. Izračun lokalnih homografij se izvede za posamezno podmnožico korespondenčnih točk. V splošnem je homografija preslikava, ki je definirana

samo za koplanarne točke. Če izbrana točka ni koplanarna glede na točke, ki določajo matriko H , potem izračunana korespondenčna točka odstopa od prave lokacije, kar imenujemo paralaksa. Ker najdene začetne korespondenčne točke na sliki 1 v splošnem niso koplanarne, je potrebno korespondenčne pare iz množice začetnih korespondenčnih točk združiti v podmnožice tako, da so točke vseh parov posamezne podmnožice koplanarne. Če posamezna podmnožica vsebuje zadostno število korespondenčnih parov, t.j. vsaj štiri, lahko zanjo izračunamo matriko H , pri čemer imamo na voljo različne metode [2] (npr. metoda DTL, Sampson). V našem postopku smo uporabili Sampsonovo metodo.

4. Korespondenčno točko lahko sedaj določimo s pomočjo lokalne homografije. Za vsako regijo, ki vsebuje vsaj štiri začetne korespondenčne točke, določimo matriko H , ki omogoča enolično preslikavo vseh pikslov znotraj regije. Za poljubno izbrano točko na prvi sliki je tako potrebno zgolj ugotoviti, kateri regiji pripada. Za dobljeno regijo se nato uporabi pripadajoča matrika H , ki enolično določa korespondenčno točko na drugi sliki po enačbi:

$$\mathbf{p}' = H\mathbf{p} \quad (1)$$

kjer je \mathbf{p} poljubno izbrana točka na prvi sliki, \mathbf{p}' korespondenčna točka točke \mathbf{p} na drugi sliki in matrika H predstavlja lokalno homografijo regije, katere članica je točka \mathbf{p} .



Slika 2: Segmentirana slika prve kamere.

Predhodne meritve naše metode ASIFT-SH so pokazale, da izračunane korespondenčne točke odstopajo le nekaj pikslov od njihove prave lokacije. Povprečna odstopanja naše metode za posamezno sliko so v razponu med 0,36 in 68,56 pikslov. Odstopanje 15 pikslov na sliki v naravi predstavljata okoli 6 mm za objekte oddaljene 15 m. Podrobnejši rezultati meritev so navedeni v [14].

Rezultate prejšnjih meritev smo podrobneje analizirali in se osredotočili na primere, ko je standardni odklon velik glede na povprečno odstopanje. Takšen rezultat nakazuje, da so korespondenčne točke v splošnem dobro vzpostavljene, vendar pa obstaja manjša množica točk, za katere so bile določene korespondenčne točke z veliko napako. V tem prispevku se bomo osredotočili prav na takšne množice točk in bomo predstavili princip

avtomatske identifikacije in transformacije, s čimer omogočimo da metoda ASIFT-SH tudi zanje zagotavlja dobre rezultate.

3. ADAPTIVNA VZPOSTAVITEV KORESPONDENCE Z METODO ASIFT-SH

Omenili smo, da se pojavlja manjša množica točk za katere metoda ASIFT-SH izračuna korespondenco z veliko napako. Korespondenčna točka se izračuna s pomočjo ustrezno izbrane matrike H po enačbi (1), pri čemer se posamezna matrika H izračuna iz začetnih korespondenčnih točk določene podmnožice. Začetne korespondenčne točke, ki ležijo znotraj iste regije, tvorijo posamezno podmnožico S_i . Ker segmentacija bistveno vpliva na obliko regij lahko sklepamo, da s tem vpliva tudi na izračun matrik H . Prvotna izvedba metode ASIFT-SH temelji na predpostavki, da segmenti slike predstavljajo koplanarne regije. To je sicer v veliki meri mogoče doseči z izbiro ustrezne segmentacijske metode [13] in nastavitvijo ustreznih parametrov uporabljene metode. Regije, ki niso koplanarne, pa identificiramo s pomočjo 3D-rekonstrukcije začetnih korespondenčnih točk. Iz rekonstruiranih začetnih korespondenčnih točk izračunamo njihovo pripadajočo ravnino v prostoru z uporabo singularnega razcepa (SVD). Rekonstruirane točke zapišemo v matriko M :

$$M = \begin{bmatrix} p_{1_x} & p_{1_y} & p_{1_z} & 1 \\ p_{2_x} & p_{2_y} & p_{2_z} & 1 \\ \dots & \dots & \dots & \dots \\ p_{n_x} & p_{n_y} & p_{n_z} & 1 \end{bmatrix}, \quad (2)$$

kjer je p_{1_x} koordinata x prve točke in n je število vseh točk podmnožice. Metoda SVD razcepi matriko M na produkt treh matrik, in sicer:

$$[U, D, V] = SVD(M), \quad (3)$$

kjer sta matriki U in V ortogonalni, matrika D pa je diagonalna matrika. Lastni vektor iz matrike V , ki pripada najmanjši vrednosti iz matrike D implicitno predstavlja enačbo ravnine, za katero velja, da je povprečna razdalja vseh točk podmnožice najmanjša. Ker so lastne vrednosti v matriki D urejene po velikost, potem je zadnji stolpec v matriki V lastni vektor, ki predstavlja našo rešitev.

Identifikacija koplanarnosti temelji na opazovanju povprečne oddaljenosti \bar{d} in standardnega odklona oddaljenosti σ rekonstruiranih točk podmnožice od njihove ravnine, izračunane po enačbah (2) in (3). Sklepamo lahko dvojce:

1. če je povprečno odstopanje \bar{d} zmerno veliko in standardni odklon σ velik, potem upravičeno sklepamo, da znotraj podmnožice začetnih korespondenčnih točk obstaja majhen nabor točk, ki niso koplanarne, glede na ostale točke podmnožice in
2. če sta povprečno odstopanje \bar{d} in standardni odklon σ velika, sklepamo, da znotraj podmnožice začetnih korespondenčnih točk obstaja večji nabor ali več manjših naborov točk, ki niso koplanarni glede na ostale točke podmnožice.

V prvem primeru je potrebno osnovno podmnožico začetnih korespondenčnih točk S_i razdeliti na dve manjši podmnožici S_i in S_j , ki vsaka zase ustreza kriteriju koplanarnosti. Delitev poteka tako, da ustvarimo novo podmnožico S_j in iz množice S_i prestavimo le tiste točke, ki so nadpovprečno oddaljene od izračunane ravnine. Ker smo podmnožico S_i razdelili na dve podmnožici S_i in S_j , je enako potrebno storiti tudi z regijo R_i . Segmentacija obravnavane regije je bila izvedena z metodo rast regij, kjer smo točke podmnožic S_i in S_j uporabili kot začetne točke segmentacije. Prvotno regijo R_i smo tako razdelimo v dve novi regiji R_i in R_j , ki vsebujeta točke iz podmnožic S_i in S_j . Na koncu je potrebno še izračunati lokalni homografiji za podmnožici S_i in S_j .

Drugi problem je težje rešljiv. Za začetne korespondenčne točke iz podmnožice S_i je potrebno ugotoviti koplanarnosti znotraj podmnožice same. Tako se morajo 3D točke iz podmnožice S_i ponovno razvrstiti v podmnožice na način, da bodo 3D točke znotraj posamezne novo nastale podmnožice koplanarne. Opisani problem je tema našega nadaljnjega raziskovalnega dela.

3. ZAKLJUČEK

Nadgradnja naše metode ASIFT-SH s pomočjo 3D-rekonstrukcije omogoča identifikacijo podmnožic točk, ki niso koplanarne. Če je standardni odklon oddaljenosti točk od njihove ravnine velik, se podmnožica razbije na dve manjši podmnožici, pri čemer so točke v posamezni podmnožici koplanarne oz. je njihova oddaljenost od ravnine manjša. Nad novo nastalimi podmnožicami ponovno izračunamo lokalne homografije, ki sedaj zagotavljajo bolj natančno določitev korespondenčne točke. Postopek lahko tudi iteriramo. Sledi še postopek segmentacije, kjer točke iz novo nastalih podmnožic uporabimo kot začetne točke za segmentacijo. Za segmentacijo smo uporabili metodo rast regij. Prihodnje raziskovalno delo bo usmerjeno v reševanje problema, ko sta povprečno odstopanje in standardni odklon oddaljenosti točk od ravnine velika in je obravnavano podmnožico potrebno razbiti na več podmnožic za zagotavljanje koplanarnosti.

LITERATURA

1. E. Delponte, F. Isgro, F. Odone, A. Verri (2006), SVD-matching using SIFT features, *Graph. Models*, vol. 68, str. 415-431.
2. R. Hartley, A. Zisserman (2004), *Multiple View Geometry in Computer Vision*, Cambridge University Press, druga izdaja.
3. D. G. Lowe (1999), Object recognition from local scale invariant features, *International Conference on Computer Vision (ICCV)*, str. 1150-1157.
4. D. G. Lowe (2004), Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, vol. 60, str. 91-110.

5. J. M. Morel, G. Yu (2009), ASIFT: A new framework for fully affine invariant image comparison, *SLAM Journal on Imaging Science*, vol. 2.
6. M. Pilu (1997), Uncalibrated stereo correspondence by singular value decomposition, *Tehnično poročilo HPL-97-96*, Digital Media Department HP Laboratories Bristol.
7. P. Podbreznik, D. Rebolj (2005), Automatic comparison of site images and the 4D model of the building, *CIB W78 22nd conference on information technology in construction*, str. 235-239.
8. P. Podbreznik, D. Zazula. Določanje uporabnosti homografij, *Zbornik strokovne konference ROSUS 2009 and Računalniška obdelava slik in njena uporaba v Sloveniji 2009*, str. 57-62.
9. D. Rebolj, N. Čuš-Babič, A. Magdič, P. Podbreznik, M. Pšunder (2008), Automated construction activity monitoring system, *Advanced Engineering Informatics*, vol. 22, str. 493-503.
10. C. Schmid, Mohr (1997), Local greyvalue invariants for image retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 19, str. 872-877.
11. G. Yu, J. M. Morel (2009), A fully affine invariant image comparison method, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
12. Z. Zhang, R. Deriche, O. Faugeras, Q. T. Luong (1994), A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry, *Tehnično poročilo 2273, Institut National de Recherche en Informatique et en Automatique*.
13. P. F. Felzenszwalb, D. P. Huttenlocher (2004), Efficient Graph-Based Image Segmentation, *International Journal Of Computer Vision*, vol. 59, str. 109-131.
14. P. Podbreznik, B. Potočnik (2009), Določanje korespondence poljubno izbrane točke za velike odmike dveh kamer, *Zbornik mednarodne elektrotehniške in računalniške konference (ERK) 2009*, vol. B, str. 287-290.

ANALIZA NIR HIPERSPEKTRALNIH SLIK

Matjaz Kosec¹, Dejan Tomažević^{1,2}, Franjo Pernuš^{1,2}, Boštjan Likar^{1,2}

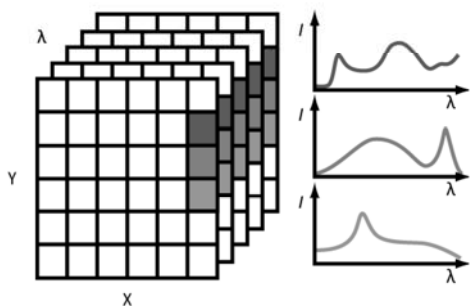
¹Sensum, sistemi z računalniškim vidom d.o.o.,
Tehnološki park 21, 1000 Ljubljana

²Univerza v Ljubljani, Fakulteta za elektrotehniko,
Laboratorij za slikovne tehnologije, Tržaška 25, 1000 Ljubljana
E-pošta: matjaz.kosec@sensum.si
URL: <http://www.sensum.si>

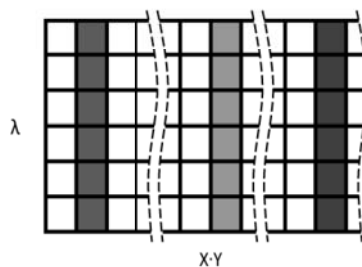
POVZETEK: *Hiperspektralno slikanje v bližnjem infrardečem delu spektra (NIR) je tehnika za neinvazivno opazovanje merjenca v realnem času. Zaradi kompleksnih mehanizmov nastanka spektrov so potrebni zahtevnejši postopki kalibracije, obdelave in analize. Hiperspektralne slike lahko razumemo kot zbirko slik zajetih pri različnih valovnih dolžinah ali pa kot sliko, ki ima v vsakem slikovnem elementu zbrano informacijo o opazovanem spektru. Običajni postopek se prične s preureditvijo hiperspektralne slike v matriko spektrov. Temu sledijo predobdelava, multivariatna analiza in izračun kvalitativnega parametra ter preureditev niza parametrov v prostorsko definirano matriko oz. sliko. Med posameznimi slikami pri različnih valovnih dolžinah lahko obstaja močna korelacija kar privede do problema multikolinearnosti. Najpogosteje uporabljene metode za redukcijo kompleksnosti temeljijo na izračunu glavnih komponent. V prispevku razložimo osnovne pojme in metode za analizo hiperspektralnih slik ter na primeru prikažemo postopek kvalitativne analize hiperspektralnih slik binarne mešanice pomožnih farmacevtskih sestavin.*

1. UVOD

NIR spektroskopija (NIRS) je spektroskopska metoda, ki sloni na opazovanju spektra v bližnjem infrardečem delu elektromagnetnega valovanja (780...2500nm). Pojav in spektralne lastnosti absorpcijskih trakov so večinoma pogojene z anharmoničnostjo in Fermijevo resonanco. V približku lahko nihajočo molekulo opišemo s kvantnim modelom anharmohičnega oscilatorja, kjer so možni prehodi med stanji, katerih kvantna števila se razlikujejo za celo število. Takšni prehodi med energijskimi nivoji so osnova pojava spektroskopskih pasov v NIR območju. Zaradi Fermijeve resonance medmolekularnih vezi in dipolnih interakcij je spektralni odziv kompleksen. Absorpcijski trakovi so običajno široki, prekrivajoči in 10-100 krat šibkejši kot pri pripadajočih osnovnih prehodih v MIR (Middle InfraRed) absorpcijskih območjih. Te lastnosti so resna omejitev občutljivosti metode ter narekujejo uporabo zahtevnejših metod za analizo podatkov. Po drugi strani pa majhen absorpcijski koeficient omogoča globoko



Sl. 1 V vsakem slikovnem elementu hiperspektralne slike je zbrana celotna spektralna informacija.



Sl. 2 Preureditev 3D podatkovne kocke hiperspektralne slike v matriko spektrov.

prodornost NIR sevanja v vzorec in s tem boljše prilagodljivost na njegovo debelino. Mogoče je opazovanje zelo absorptivnih ali zelo sipalnih vzorcev, tako v presevnem kot odbojnem režimu. Priprava vzorcev je običajno nezahtevna ali celo nepotrebna, kar omogoča neinvazivno analizo v realnem času.

NIRS se je izkazala kot primerno orodje za preučevanje tako kemičnih (npr. identifikacija sestavin, sestava, vsebnost vlage) kot fizikalnih (npr. velikost in porazdelitev delcev, trdnost, kompaktnost) zanimivih parametrov.

V tradicionalnih merilnih sklopih se NIR spekter zajema v posameznih območjih. Z razvojem 2D slikovnih detektorjev in pripadajoče opreme za izbiro opazovane valovne dolžine, so se v zadnjih letih odprle možnosti istočasnega zajemanja spektrov v več prostorsko ločenih točkah. Iz izmerkov je mogoče tvoriti 3D podatkovno kocko oz. NIR hiperspektralno sliko. Dve dimenziji predstavljata prostorsko razsežnost, tretja pa je namenjena spektralnemu delu (Sl. 1). Uvedba nove informacije omogoči vrednotenje prostorske porazdelitve zanimivega parametra.

Glavne prednosti NIR hiperspektralnega slikanja so neinvazivnost, hitrost in relativna preprostost. Število uporabnih izvedb za industrijsko rabo je v porastu, predvsem na področju identifikacije farmacevtskih surovin, prehrambnih izdelkov, petrokemični industriji itd. Z uvedbo PAT (Process Analytical Technology) [1] pobude postaja opisana metoda vse bolj zanimiva za farmacevtsko industrijo, in sicer kot pomožno orodje kontrole posameznih proizvodnih procesov. Obstoječe paradigme kontrole proizvodnega procesa namreč temeljijo na preverjanju končnega izdelka in kontrole vhodnih sestavin. Takšen pristop ne omogoča sprotne povratne informacije, ki bi služila kot parameter za takojšnje regulacijo procesa. V prihodnosti se zato s strani proizvajalcev farmacevtskih izdelkov pričakuje velik porast povpraševanja po tej tehnologiji.

2. PREDSTAVITEV PROBLEMA IN PRIČAKOVANE REŠITVE

Glavna težava pri razvoju algoritmov za obdelavo hiperspektralnih slik izhaja iz kompleksnosti fizikalnih mehanizmov nastanka NIR spektrov. Ta poleg informacije o sestavi merjenca vsebuje še podatke o njegovih fizikalnih lastnostih, kot so velikost

delcev, trdnost, kompaktnost, vsebnost vode in celo podatke o posameznih fazah izdelave vzorca. To lahko pomeni večji nabor možnih opazovanih parametrov ali pa predstavlja dodatno neželjeno informacijo, ki močno poveča kompleksnost sistema ali poslabša njegovo občutljivost.

Multivariatna analiza je veja kemometrije, ki se osredotoča na iskanje zvez med setom meritev pridobljenih z enostavnimi metodami in izmerki, zbranimi s kompleksnejšimi, robustnejšimi in uveljavljenimi postopki, npr. s HPLC (High Performance Liquid Chromatography). Cilj analize je najti zvezo, ki omogoča napoved rezultatov kompleksnejših postopkov iz izmerkov enostavnejših merilnih metod.

Hiperspektralno sliko je mogoče opazovati kot zbirko prostorsko lociranih spektrov ali kot serijo slik pri različnih valovnih dolžinah. Obravnava primerno se posamezne spektre analizira z orodji za spektralno analizo oziroma z algoritmi za obdelavo slik pri različnih valovnih dolžinah. Večina najpopularnejših analitičnih metod izvira iz eno-točkovne spektroskopije, zato se v nadaljevanju prispevka posvetimo samo obravnavi s spektroskopskega vidika. Za takšen pristop je v prvem koraku potrebno hiperspektralne slike prepisati v obliko 2D matrike, vendar tako, da je mogoča kasnejša rekonstrukcija slike.

Analizo NIR spektrov lahko izvajamo na zelo različne načine. Včasih zadostuje že preprosta analiza pri valovni dolžini, kjer je variacija absorpcije najbolj povezana z variacijo zanimivega parametra. V splošnem univariatna statistika ni robustna in je močno odvisna od vseh mogočih vplivov: bodisi od spremenjenih pogojev merjenja bodisi od sprememb samega vzorca. V praktičnih primerih se izkaže, da je potreben pristop z multivariatnimi analitičnimi postopki, npr. s PCR (Principal Component Regression) ali PLSR (Partial Least Squares Regression) .

Običajni potek matematične analize hiperspektralnih slik se začne s kalibracijo slik, sledijo: predobdelava spektrov [2], izračun kvalitativne vrednosti zanimivega parametra in na koncu preureditev izračunanih elementov v obliko slike.

Kalibracija slik

Zajete hiperspektralne slike so lahko zaradi različnih vplivov okolice ali pa samih lastnosti merilnega sklopa degradirane. Značilne degradacije, ki se pojavljajo so: različne občutljivosti ali celo nedelovanje posameznih slikovnih elementov, neenakomerna spektralna občutljivost, spektralno odvisne geometrijske napake leč.

Po Bouguer-Lambert-Beerovem zakonu je absorbanca A odvisna od optične poti l , koncentracije snovi v opazovani mešanici c in molarne absorptivnosti k : $A = k l c$. V primeru zajema slik difuznega odboja je količina odbite svetlobe R definirana kot razmerje med hiperspektralnima slikama odbite svetlobe G in reference G_0 : $R = G / G_0$. Iz česar lahko z izračunom logaritma v vsakem slikovnem elementu dobimo sliko psevdo-absorbance¹ $A = -\log_{10}(R)$. V kolikor je na voljo referenčni standard z visoko

¹ Termin je značilen za področje NIRS.

odbojnostjo preko celotnega spektra lahko kalibracijo intenzitete izračunamo z enačbo $R_0 = (G - D) / * (G_0 - D)$, kjer $/*$ označuje deljenje med soležnimi elementi hiperspektralne slike, D pa temni tok senzorja. Podrobnejši opis problematike lahko bralec najde v [3].

Standardi z ostrimi spektralnimi vrhovi lahko služijo za oceno valovne točnosti. Kadar je znan fizikalni model prenosne funkcije merilnega sistema je mogoče z opazovanjem referenčnega svetila obenem oceniti spektralno ločljivost in določiti parametre umeritvene krivulje [4].

Za geometrijsko kalibracijo hiperspektralnega sistema se lahko uporabi fizični kaliber znanih dimenzij. Slike kalibra so zajete pri več valovnih dolžinah in nato prostorsko poravnane na sliko modela kalibra. Za poravnavo se uporabi preslikavo z B-zlepki in korelacijski koeficient za mero podobnosti. S tem je dosežena prostorska natančnost približno desetine velikosti slikovnega elementa, istočasno pa se izboljša tudi prostorska ločljivost hiperspektralnega sistema [5].

Predobdelava spektrov

Predvsem zaradi velikega vpliva nelinearnih pojavov sipanja svetlobe je NIRS področje bogato z zbirko metod predobdelave. Pri trdnih vzorcih so neželene variacije v večji meri posledica sipanja ter razlik dolžine optične poti svetlobe in predstavljajo večji del variabilnosti danih vzorcev. Difuzno odbita svetloba, ki je glavni vir informacij, vsebuje tako informacijo o kemični sestavi kot tudi mikro-strukturi. Kadar so opazovani delci večji od valovne dolžine vpadne svetlobe ima največjo vlogo Lorentz-Mie sipanje, ki je anizotropno, odvisno od oblike delcev in le malo odvisno od valovne dolžine. Iz teorije sipanja je razvidno, da ima sipanje multiplikativni učinek na odbojni spekter. Najpogosteje uporabljene metode za ločevanja kemijskih in fizikalnih pojavov so: MSC (ang. Multiplicative Scatter Correction), njena izboljšava EMSC (ang. Extended Multiplicative Scatter Correction) [6], SNV (ang. Standard Normal Variate) in odvajanje spektrov.

Najpogosteje uporabljena metoda predobdelave na področju NIRS je MSC. Sestavljena je iz dveh korakov. Najprej se s prilagajanjem linearnega modela med referenčnim x_{ref} (običajno kar povprečni spekter) in opazovanim spektrom x_{izm} ocenijo koeficienti popravka $x_{izm} = b_0 + b_{ref,1} \cdot x_{ref} + e$. Čemur sledi izračun popravljenega spektra x_{cor} :

$$x_{cor} = \frac{x_{izm} - b_0}{b_{ref,1}} = x_{ref} + \frac{e}{b_{ref,1}} \quad (1)$$

Multivariatna analiza

PCR

Zaradi visoke korelacije oz. kolinearnosti je med različnimi spremenljivkami v spektru potreben opis podatkov z manjšim številom neodvisnih spremenljivk, ki vsebujejo zgolj relevantno informacijo.

PCR (Principal Component Regression) je regresijska analiza, ki za oceno regresijskih koeficientov uporabi PCA (Principal Component Analysis). Metoda PCA je ortogonalna linearna transformacija, ki spektralne podatke preslika v nov koordinatni sistem. Osi novega ortogonalnega koordinatnega sistema so usmerjene v smeri največjih varianc in medsebojno pravokotne. Nove spremenljivke se imenujejo glavne komponente (ang. Principal Components - PC). PC so lastni vektorji, ki sovpadajo z največjimi lastnimi vrednostmi kovariančne matrike. Prvi lastni vektor predstavlja največjo variacijo med vsemi linearnimi kombinacijami, vsak naslednji opiše tem večji del preostale variabilnosti. V analizi NIR spektrov je pogosto mogoče stisniti večino spektralne variabilnosti v le nekaj faktorjev. Matriko zajetih spektrov X najprej centriramo okrog povprečnega spektra in jo nato razcepimo na matriki P (ang. loadings) in T (ang. scores):

$$X = TP^T + E \quad (2)$$

Odziv y je spremenljivka, ki kvantitativno opiše zanimivo informacijo, npr. koncentracijo. Model za napoved odziva je opisan z linearno kombinacijo spektra T in vektorja q (ang. loadings) ter ostankom f :

$$y = Tq + f \quad (3)$$

Za dejansko izvedbo redukcije števila spremenljivk je potrebno iz matrik (T,P) vzeti samo n_{PC} glavnih komponent, s čimer se definira novi matriki (T_{nPC}, P_{nPC}) . Z združitvijo (2) in (3) ter upoštevanjem samo izbranih komponent, lahko iz zbirke neznanih zajetih spektrov \tilde{X} dobimo napoved za :

$$\tilde{y} = \tilde{X}P_{nPC}q_{nPC} \quad (4)$$

Osnovne komponente (P_{nPC}, T_{nPC}) dobimo z razcepom učnega seta spektrov. Linearno funkcijsko odvisnost med T_{nPC} y lahko opišemo s q_{nPC} :

$$q_{nPC} = (T_{nPC}^T T_{nPC})^{-1} T_{nPC}^T y \quad (5)$$

Popularni metodi razcepa matrike X sta: direktni razcep kovariančne matrike XX' in iterativni algoritem NIPALS. S slednjim izračunamo vsako komponento posebej in lahko izvajanje prekinemo, ko imamo želeno število komponent. Pravilna izbira osnovnih komponent je ključnega pomena za uspešno napoved iskanega parametra.

PLSR

PLSR je regresijska analiza, ki je podobna PCR le, da za oceno koeficientov uporabi PLS (ang. Partial Least Squares). PLS poišče smeri največje variabilnosti z primerjavo tako spektralne informacije kot informacije o ciljnem parametru [7]. Medtem, ko glavne komponente PCA predstavljajo smeri največje variabilnosti, so pri PLS usmerjene v smeri največje variabilnosti glede na variacijo zanimivega parametra. Prva komponenta opiše največjo korelirano variacijo in vsaka naslednja tem večji preostali del.

Nastavek je podoben nastavku za PCA (2,3). Vendar sta T (PLS scores) in P (PLS loadings) drugačna kot pri PCA. Najpopularnejši iterativni metodi za iskanje glavnih

komponent sta NIPALS in SIMPLS. Kadar iščemo več zanimivih parametrov (npr. koncentracije različnih sestavin v mešanici) vsakemu pripada lastni set T , P , q .

3. PRIMER ANALIZE BINARNE MEŠANICE MIKROKRISTALINIČNE CELULOZE IN LAKTOZE

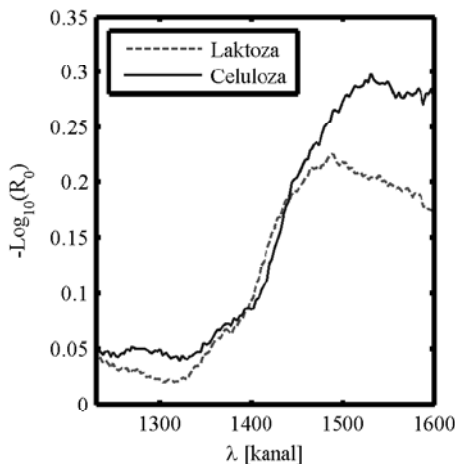
Sistem za zajem hiperspektralnih slik smo sestavili iz AOTF (Acoustically Tunable Optical Filter) (Brimrose VA200-9-1.7-L), NIR kamere (Xenics XEVA-1.7-320). Za osvetlitev smo uporabili dve difuzni halogenski svetili.

Vzorce smo pripravili iz mešanice mikrokristalinične celuloze in laktoze. Masni delež laktoze v mešanici se je v 10% korakih spreminjal od 0% (čista celuloza) do 100% (čista laktoza).

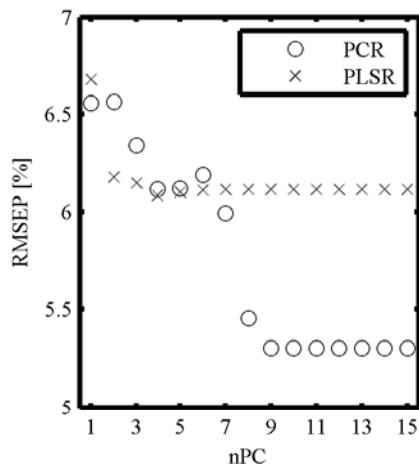
Zajem slik smo izvedli v dveh zaporednih dneh in sicer štiri hiperspektralne slike pri vsaki koncentraciji. Skupaj smo tako zajeli 88 hiperspektralnih slik, pri prostorski ločljivosti 320×256 slikovnih elementov in 124-ih valovnih dolžinah. Slike so bile kasneje pomanjšane na velikost 170×225 slikovnih elementov.

V prvem koraku smo izločili pokvarjene slikovne elemente in izvedli kalibracijo intenzitete slik s hiperspektralno sliko belega standarda. Hiperspektralno sliko smo preuredili v 2D matriko in spektre izboljšali z metodo MSC. Hiperspektralne slike smo razdelili na učni in testni set, ter izvedli kvantitativno analizo s PCR in PLS. Običajno se za merilo kakovosti modela napovedovanja uporabi RMSEP (Root Mean Square Error of Calibration):

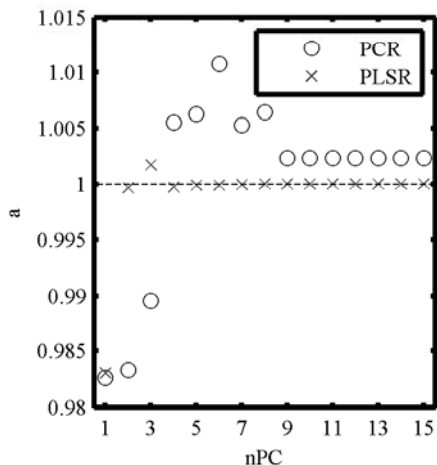
$$RMSEP = \left(\sum_{i=1}^N (\hat{y}_i - y_i)^2 / N_p \right)^{1/2}, \quad (6)$$



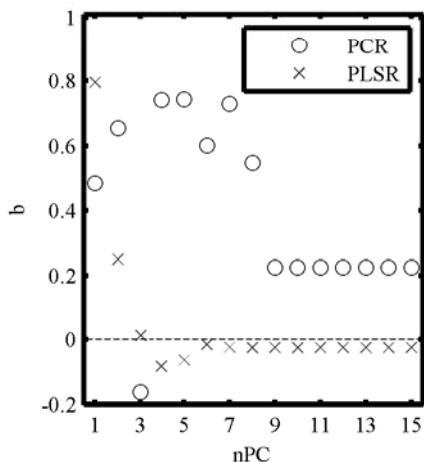
Sl. 3 Spektra čiste laktoze in celuloze.



Sl. 4 RMSEP pri različnih številih izbranih glavnih komponent.



Sl. 5 Vrednost parametra a pri različnih številih izbranih glavnih komponent.



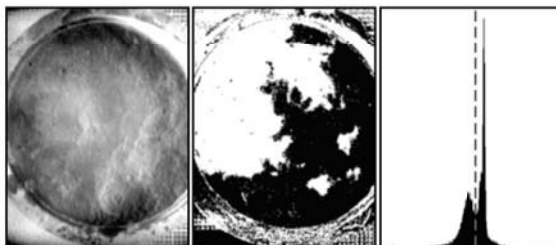
Sl. 6 Vrednost parametra b pri različnih številih izbranih glavnih komponent.

Kjer sta \hat{y}_i in y_i napovedana ter dejanska koncentracija, N_p pa je število uporabljenih spektrov. Ker PCA v splošnem bolje opiše celotno variabilnost, so ob pravilni izbiri komponent napovedi s PCR natančnejše kot s PLSR (Sl. 4). Celotno variabilnost izmerjenih spektrov pri 124 valovnih dolžinah lahko opišemo z 9 PCA ali 5 PLS komponentami. Zaradi boljšega opisa razlike med metodama smo izračunali še dodatno primerjavo koeficientov linearne zveze ($\hat{y} = ay + b$) med napovedano in dejansko vrednostjo koncentracije. V idealnih okoliščinah bi bila parametra $a = 1$ in $b = 0$. Točnost ocene modela napovedi je boljša pri PLSR (Sl. 5, Sl. 6).

Poleg izmerkov homogenih mešanic smo zajeli tudi sliko slabo premešanega vzorca. Na sliki (Sl. 7) je prikazana uprakovljena slika izračunane koncentracije. Kljub temu, da pri eni valovni dolžini ni slutiti območij, ki jih zavzemata sestavini oz. njuni mešanici, ju je z multivariatno analizo mogoče dobro ločiti.

4. ZAKLJUČEK

V prispevku je predstavljen postopek multivariatne analize NIR hiperspektralnih slik s katerim je mogoče določiti prostorsko porazdelitev informacije o kemičnih ali fizikalnih lastnostih opazovanih vzorca.



Sl. 7 Na levi je prikazana slika pri eni valovni dolžini. Sredinska slika prikazuje upragovljeno prostorsko porazdelitev laktoze (svetlo področje) in mikrokristalinične celuloze (temna področja). Skrajno desno je prikazan histogram slike izračunanih koncentracij. Črtkana črta označuje nivo upragovljanja.

5. LITERATURA

1. FDA (2004), "Guidance for Industry: PAT - A Framework for Innovative Pharmaceutical Development, Manufacturing, and Quality Assurance,".
2. Å. Rinnan, F.V.D. Berg, and S.B. Engelsen (2009), "Review of the most common pre-processing techniques for near-infrared spectra," *TrAC Trends in Analytical Chemistry*, vol. 28, str. 1201-1222.
3. P. Geladi (2007), *Techniques and Applications of Hyperspectral Image Analysis*, John Wiley and Sons.
4. J. Katrašnik, M. Bürmen, F. Pernuš, and B. Likar (2009), "Spectral characterization and calibration of AOTF spectrometers and hyper-spectral imaging systems," *Chemometrics and Intelligent Laboratory Systems*.
5. Ž. Špiclin, J. Katrašnik, M. Bürmen, F. Pernuš, and B. Likar (2010), "Geometrical calibration of an AOTF hyperspectral imaging system," , SPIE Photonics West, SF, CA, USA.
6. T. Isaksson and T. Næs (1988), "The Effect of Multiplicative Scatter Correction (MSC) and Linearity Improvement in NIR Spectroscopy," *Applied Spectroscopy*, vol. 42, str. 1273-1284.
7. S. Wold, M. Sjöström, and L. Eriksson (2001), "PLS-regression: a basic tool of chemometrics," *Chemometrics and intelligent laboratory systems*, vol. 58, str. 109–130.

PRIMERENOST POSTOPKOV PORAVNAVE SLIK ZA DOLOČANJE GOSTO POSEJANIH KORESPONDENČNIH TOČK

Jurij Rakun, Damjan Zazula

Fakulteta za elektrotehniko, računalništvo in informatiko
E-pošta: jurij.rakun@uni-mb.si

POVZETEK: *V delu proučujemo možnost uporabe postopkov za poravnavo slik pri določanju gosto posejanih korespondenčnih točk na paru deloma podobnih slik. Postopek poravnave smo povzeli s pomočjo dveh algoritmov poravnave, kjer prvi pare slik poravnava togo, drugi pa nato še elastično. S pomočjo dobljenih deformacijskih matrik lahko nato določimo ujemanje najbolje ocenjenih korespondenčnih točk. Postopek smo preverili na naključno izbranih primerih nepravilnih oblik. Preliminarni rezultati kažejo, da je odstopanje med več kot polovico korespondenčnih točk manjše kot en piksel. Hkrati z njimi dosežemo do 30 % pokritost, kar pomeni gosto mrežo korespondenčnih parov točk na iskanih objektih, ki jih tako lahko natančno rekonstruiramo v 3D.*

1. UVOD

Na področju računalniškega vida in digitalne obdelave slik se pogosto srečamo s problemom iskanja korespondenčnih zvez oziroma iskanja korespondenčnih točk [1, 4, 10]. To so točke, ki na paroma enakih ali vsaj podobnih slikah opišejo isto vsebino in imajo ključen pomen, ko želimo poravnati pare slik [1, 2, 3, 4, 6], rekonstruirati objekte v 3D [11] ali pa preprosto združiti več zaporednih, deloma prekrivajočih se posnetkov v skupno kompozitno sliko [4]. Postopki določanja korespondenčnih točk navadno temeljijo na lastnostih, kot so intenzivnost barvnih ravnin [2], geometrijske lastnosti [4], slikovno-frekvenčne lastnosti [6] in statistične lastnosti [7].

Poravnavo slik s pomočjo korespondenčnih parov točk navadno uporabimo, ko je pomembna hitrost poravnave [2]. Predpogoj je seveda postopek določanja korespondenčnih parov, glede na katere se tvori transformacijska matrika, ki opiše preslikavo ene slike v drugo.

Problem iskanja korespondenčnih parov srečamo tudi pri uporabi stereo vida [11], ko opazujemo sceno iz dveh različnih zornih kotov. Glede na deformacijo korespondenčnih parov točk lahko nato sklepamo o oddaljenosti njune 3D rekonstrukcije in tako

rekonstruiramo prostorsko globino, ki smo jo neposredno izgubili z zajemom 2D posnetkov scen.

Korespondenčne točke pomagajo tudi pri sestavljanju kompozitnih posnetkov [4], ki jih kot mozaik sestavljamo v skupno celoto. Seveda se morajo posnetki scen deloma prekrivati. Poiščemo značilne korespondenčne pare in posnetke premaknemo tako, da se korespondenčne točke ujamejo. Iz množice delnih posnetkov sestavimo večjo, recimo panoramsko sliko.

Vsi trije predstavljeni problemi zahtevajo predhodno dobro določene korespondenčne pare točk. Za njihovo iskanje obstaja precej pristopov [1, 4, 10], pri katerih se navadno zanašamo na geometrijske lastnosti, kot so oglišča, robovi, konture ipd. Na žalost ti postopki ne nudijo dovolj dobre pokritosti za manj izrazite dele posnetkov, kot so npr. plodovi sadnih dreves, in zato v primerih, kot je 3D rekonstrukcija, niso najboljši. Zato smo se odločili, da preizkusimo obratno pot, kot jo uporablja poravnava slik s pomočjo korespondenčnih točk. Proučili smo možnost, da poravnavamo slike s postopki, ki ne zahtevajo korespondenčnih točk, nato pa iz vektorjev premikov za posamezne slikovne točke določimo korespondenco med njimi.

Članek v naslednjem poglavju opisuje poravnavo slik in metodo za ocenitev dobljenih korespondenčnih točk. V tretjem poglavju interpretiramo eksperimente, v četrtem pa nakažemo smernice za nadaljnje raziskave in sklenemo članek.

2. POSTOPKI ZA PORAVNAVO SLIK

Z namenom odkriti množico korespondenčnih parov točk, par slik najprej poravnamo. Ob tem postavljamo domnevo, da je par slik predhodno že grobo poravnan (kar lahko npr. dosežemo s poravnavo težišč [12] ali poravnavo s pomočjo frekvenčnega prostora [2]). Par slik nato poravnamo v dveh korakih: najprej fino togo in nato še elastično. Toga poravnava deluje globalno in ne spreminja intenzitete pikslov, medtem ko elastična deluje lokalno in spreminja svetlost in kontrast. Toga poravnava tako par slik približa enega drugemu, kolikor je le mogoče, nato pa elastična poravnava sliki dodatno poravnava na območjih, ki so lokalno deformirana.

2.1. TOGA PORAVNAVA

Fina toga poravnava [2, 3, 12] uvaja globalno afino prilagajanje parov slik. Omogoča tudi poravnavo deloma neenakih slik, kot so npr. posnetki naravnih scen, zajeti iz različnih zornih kotov.

Po [3, 5] izpeljemo postopek fine poravnave na naslednji način. Vzemimo, da sta ekvivalentna slikovna elementa s prve in druge slike označena kot $\mathbf{p}(x,y,t)$ in $\mathbf{p}(x,y,t-1)$. Parameter t določa referenčno, parameter $t-1$ pa primerjano sliko. Če gre za sliki z enakim motivom, lahko slikovne elemente ene slike izrazimo tako, da ustrezno premaknemo slikovne elemente druge slike. To lahko zapišemo kot:

$$\mathbf{p}(x, y, t) = \mathbf{p}(m_1x + m_2y + m_5x, m_3x + m_4y + m_6x, t - 1). \quad (2.1)$$

Parametri m_i do m_6 so v tem primeru koeficienti afixne transformacije. Zapišimo sedaj oceno trenutne poravnave kot vsoto kvadratov razlik med istoležečimi slikovnimi elementi:

$$E(\bar{\mathbf{m}}) = \sum_{x=1}^M \sum_{y=1}^N [\mathbf{p}(x, y, t) - \mathbf{p}(m_1x + m_2y + m_5, m_3x + m_4y + m_6, t - 1)] \quad (2.2)$$

kjer M in N pomenita dimenziji slik. Želimo oceniti vrednosti parametrov m_i , ki pa v enačbi (2.2) ne nastopajo eksplicitno. Zato izraz razvijemo v Taylorjevo vrsto in jo aproksimiramo samo s členi prvega reda:

$$E(\bar{\mathbf{m}}) \approx \sum_{x=1}^M \sum_{y=1}^N [\mathbf{p}(x, y, t) - \{\mathbf{p}(x, y, t) + (m_1x + m_2y + m_5)\mathbf{p}_x(x, y, t) + (m_3x + m_4y + m_6)\mathbf{p}_y(x, y, t) - \mathbf{p}_t(x, y, t)\}^2], \quad (2.3)$$

kjer so p_x , p_y in p_t parcialni odvodi po x , y in t , ki jih ocenimo z diferencami. Enačbo (2.3) lahko razvijamo naprej in dobimo:

$$E(\bar{\mathbf{m}}) = \sum_{x=1}^M \sum_{y=1}^N [\mathbf{p}(x, y, t) - (m_1x + m_2y + m_5)\mathbf{p}_x(x, y, t) - (m_3x + m_4y + m_6)\mathbf{p}_y(x, y, t)]^2. \quad (2.4)$$

Enačbo (2.4) lahko še poenostavimo, tako da uvedemo vektorski zapis, v katerem je $\mathbf{c} = [xp_x \ yp_x \ xp_y \ yp_y \ p_x \ p_y]^T$ in skalar $k = p_t + xp_x + yp_y$. Zaradi preglednosti smo izpustili parametre parcialnih odvodov.

$$E(\mathbf{m}) = \sum_{x=1}^M \sum_{y=1}^N (k - \mathbf{c}^T \mathbf{m})^2. \quad (2.5)$$

Enačbo (2.5) sedaj še parcialno odvajamo po \mathbf{m} , tako da dobimo:

$$\frac{dE(\bar{\mathbf{m}})}{d\bar{\mathbf{m}}} = \sum_{x=1}^M \sum_{y=1}^N -2\mathbf{c}(k - \mathbf{c}^T \mathbf{m}). \quad (2.6)$$

Na koncu, dobljeno še enačimo z 0 in izpostavimo \mathbf{m} :

$$\mathbf{m} = [\sum_{x=1}^M \sum_{y=1}^N \mathbf{c}\mathbf{c}^T]^{-1} [\sum_{x=1}^M \sum_{y=1}^N \mathbf{c}k]. \quad (2.7)$$

Tako dobimo koeficiente, ki jih uporabimo v afixni transformaciji. Postopek pa omogoča le manjša približevanja h končni rešitvi. Zato ga je treba izvesti iterativno in za vsako iteracijo izračunati optimalnejše parametre m . Dogodi se tudi, da se najprej približujemo

optimalni rešitvi, nato pa jo preskočimo. V tem primeru je priporočljivo izvesti še nekaj dodatnih iteracij, ki nas pripeljejo nazaj v optimum. Uspešnost poravnave slik ocenjujemo s kriterijsko funkcijo, ki meri podobnost med slikama. Pri izpeljanih postopkih smo uporabljali evklidsko razdaljo.

2.2. ELASTIČNA PORAVNAVA

Nadaljujmo z elastično poravnavo, za katero bomo uporabili postopek iz poglavja 2.1, ki namesto globalne afine poravnave v tem primeru deluje lokalno afino [2, 3, 5]. V postopek pa bomo vpeljali še dva parametra, ki omogočata, da poravnamo mestoma neenako osvetljene in deloma manjkajoče, nepopolne motive [6], kot to prikazuje enačba (2.8):

$$m_7 \mathbf{p}(x, y, t) + m_8 = \mathbf{p}(m_1 x + m_2 y + m_5 x, m_3 x + m_4 y + m_6 x, t - 1). \quad (2.8)$$

Uvedli smo dodatna parametra m_7 in m_8 , s katerima opišemo spremembe v kontrastu in osvetlitvi. Podobno kot smo storili pri enačbi (2.1), lahko s pomočjo Taylorjeve vrste prvega reda razvijemo tudi enačbo (2.8). Dobimo približek napake, ki ga zapišemo kot:

$$E(\mathbf{m}) = \sum_{x=1}^M \sum_{y=1}^N (k - \mathbf{c}^T \mathbf{m})^2. \quad (2.9)$$

Enačbi za izračun skalarja k in vektorja \mathbf{c} tako dobita nekoliko drugačno obliko:

$$k = p_t - p + x p_x + y p_y \quad (2.10)$$

in

$$\mathbf{c} = [x p_x \quad y p_x \quad x p_y \quad y p_y \quad p_x \quad p_y \quad -p - 1]^T. \quad (2.11)$$

Ker lahko s postopkom elastične poravnave poravnavamo tudi deloma različne motive slik, vpeljimo v postopek iz [8] dodaten parameter w , ki ponazarja utež zastopanosti izbranega področja na posnetku. Če izpostavimo rešitev za \mathbf{m} , kot smo to storili v podpoglavju 2.1, dobimo:

$$\mathbf{m} = \left[\sum_{x=1}^M \sum_{y=1}^N (\mathbf{c} \mathbf{c}^T) \mathbf{w} \right]^{-1} \left[\sum_{x=1}^M \sum_{y=1}^N (\mathbf{c} k) \mathbf{w} \right]. \quad (2.12)$$

Seveda za izračun parametrov \mathbf{m} potrebujemo uteži w , za w pa potrebujemo parametre \mathbf{m} . Problem lahko rešimo z uporabo algoritma največje verjetnosti [9]. Ta deluje tako, da najprej grobo nastavimo uteži w glede na parametre predhodno izvedene grobe poravnave. Nadaljujemo z oceno parametrov elastične poravnave \mathbf{m} , kot smo jo spoznali v tem poglavju, nato pa ponovno popravimo uteži w . Postopek ponavljamo, dokler lahko zmanjšujemo napako približkov \mathbf{m} in uteži w .

2.3. POSTOPEK DOLOČANJA KORESPONDENČNIH TOČK

Predlagani postopek določanja korespondenčnih točk temelji na lastnosti barvnih intenzivnosti. Iskanje korespondenčnih točk izpeljemo s poravnavo slik, ki smo jo opisali v podpoglavjih 2.1 in 2.2. Kombinacija afinih parametrov toge in deformacijskih parametrov elastične poravnave opiše preslikavo od izvorne pozicije posameznega slikovnega elementa do njegove korespondenčne pozicije na drugem posnetku. Preslikava je za vsak slikovni element različna. Ujemanja so pri nekaterih boljša, pri drugih zaradi vsebinske različenosti slik slabša. Pomembno je, da izberemo tiste točke, ki preslikavo dobro opišejo.

Iskanje korespondenčnih točk opravimo tako, da ocenimo preslikavo vsakega slikovnega elementa glede na vsebinsko raznolikost njegove najbližje okolice. Če z \mathbf{I}_1 in \mathbf{I}_2 označimo okna velikosti $M \times N$ z okolico izbranih korespondenčnih točk, lahko oceno podobnosti izbranih oken zapišemo kot:

$$D(\mathbf{I}_1, \mathbf{I}_2) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |I_1(i, j) - I_2(i, j)| \quad (2.13)$$

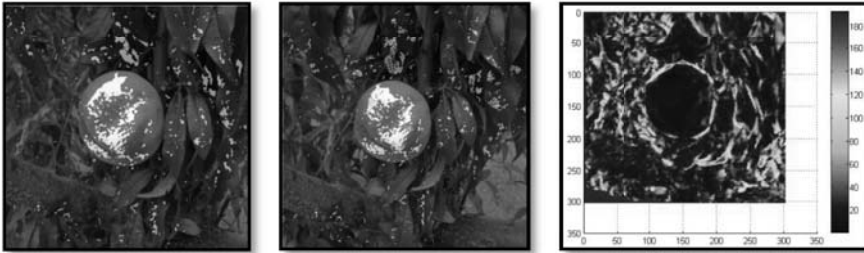
Ujemanje korespondenčnih parov točk, določeno po enačbi (2.13), razkrije dobre pare, katerih razlika bo čim bližje 0.

3. REZULTATI

Poglejmo si dva naključno izbrana primera naravnih scen in na njih poiščimo množice korespondenčnih točk. Za oba primera smo predstavili po dva podobna posnetka, posneta iz različnih zornih kotov, in rezultat njune poravnave (slika 1 in slika 3). Omejili smo se na ocenitveno okno velikosti 3×3 slikovnih elementov in za oba primera smo prikazali rezultat določanja korespondenčnih točk (slika 2 in slika 4), ki se po enačbi (2.13) razhajajo za manj kot 5 pikslov. Korespondenčne točke smo označili kot bele pike.



Slika 1: Prvi primer: par posnetkov iste scene (sliki levo in na sredini), posnet iz različnih zornih kotov, in rezultat njune poravnave na skupni slikovni ravnini (desno).



Slika 2: Porazdelitev korespondenčnih točk (sliki levo in na sredini) in ocena ujemanja (slika desno) za par posnetkov s slike 1; čim bolj temno modre so ocene, tem boljše je ujemanje.



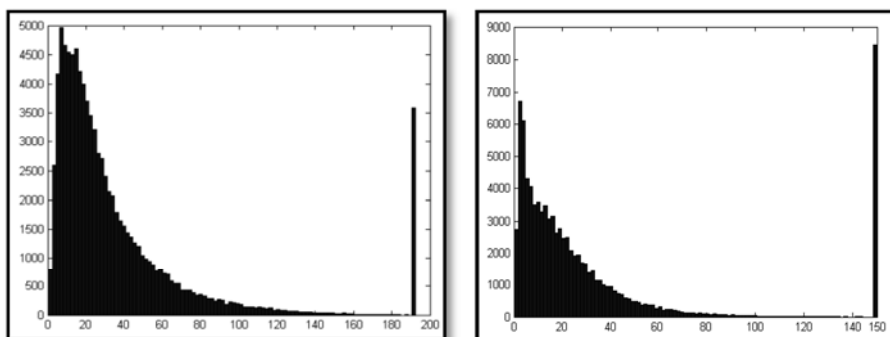
Slika 3: Drugi primer: par posnetkov iste scene (sliki levo in na sredini), posnet iz različnih zornih kotov, in rezultat njune poravnave na skupni slikovni ravnini (desno).



Slika 4: Porazdelitev korespondenčnih točk (sliki levo in na sredini) in ocena ujemanja (slika desno) za par posnetkov s slike 3; čim bolj temno modre so ocene, tem boljše je ujemanje.

Ujemanje korespondenčnih točk merimo z razdaljo med slikovnimi elementi v primerjalnem oknu velikosti 3×3 . Za točki trdimo, da se ujemata, če kumulativna vrednost absolute napake v oknu ne preseže 9. Za prvi primer smo z napako 9 sivinskih vrednosti ali manj zajeli 14716 (16 % vseh slikovnih) korespondenčnih točk, v drugem pa kar 27199 (30 % vseh). Vidimo, da je določanje korespondenčnih točk precej bolj zanesljivo za uniformna področja, kar smo lahko pričakovali. Ta so po enačbi (2.13) ocenjena bolje in različni pogled na sceno ni tako opazen, saj ne prihaja do hitrih

sprememb. Za konec si oglejmo še histograma napak, ki ju določajo izbrani korespondenčni pari točk in ju prikazuje slika 5.



Slika 5: Histograma absolutnih razlik med korespondenčnimi točkami za posnetka s slike 1 (levo) in slike 3 (desno).

V obeh testiranih primerih dobimo približno enako obliko histograma napak. Največ točk je zbranih do napake 20, večjih napak pa je bistveno manj. Na koncu (skrajno desno) dobimo še en izrazit vrh, ki pripada odstopanjem ob robu poravnanih slik, kjer naletimo na točke, prisotne samo na eni od slik. V obeh izbranih primerih smo našli tudi točke, kjer napake nismo zaznali, a je teh razmeroma malo, saj imamo opravka s posnetki deloma različnih naravnih scen, ki ju pokvari še interpolacija, nujni sestavni poseg pri poravnavi slik.

4. ZAKLJUČEK

Glede na zbrane podatke sklepamo, da lahko določimo korespondenčne pare točk s pomočjo poravnave slik in da se precejšnje število točk popolnoma ujema. Še več, določanje korespondence na predlagani način omogoča iskanje korespondence za gruče sosednjih točk, česar klasični pristopi, ki temeljijo na geometrijskih značilnostih, ne nudijo.

Eksperimenti so potrdili, da je določanje korespondenčnih točk na predstavljeni način možno tudi za nekoliko bolj problematične primere, kot so posnetki naravnih scen. Ob razhajanju v oknu okoli korespondenčnih točk kumulativno za 9 ali manj smo v prvem poskusu odkrili 16 % vseh točk, ki lahko nastopajo kot korespondenčne točke. V drugem smo našli kar 30 % vseh točk. Število dobro določenih korespondenčnih točk seveda variira od primera do primera in je odvisno od različnosti oziroma deformiranosti scen na posnetkih.

Nadaljnje raziskave bodo posvečene iskanju izboljšanih primerjalnih ocen oziroma ocen odstopanj med korespondenčnimi točkami. Ta je bila v opisani raziskavi omejena na razmeroma majhno področje 3×3 slikovnih elementov. Razlog, zakaj smo so omejili na tako majhno področje, leži v dejstvu, da primerjava ni rotacijsko neodvisna in z večjimi

ocenitvenimi okni napravimo večjo napako. V prihodnje zato velja razmisliti o vpeljavitih rotacijsko neodvisne ocenitvene metode.

LITERATURA

1. R. C. Gonzales, R. E. Woods, *Digital image processing*, 3. izdaja, Prentice Hall, Upper Saddle River, New Jersey, 2008.
2. J. Rakun, D. Zazula, Pospešitev iskanja tiskarskih napak z uporabo poravnave v frekvenčnem prostoru, *ROSUS 2007*, FERI Maribor, str. 88-95, 2007.
3. S. Periaswamy, H. Farid, Elastic Registration in the Presence of Intensity Variations, *IEEE Transactions on medical imaging*, vol. 22, št. 7, str. 865-874, 2003.
4. A. A. Goshtasby, *2-D and 3-D image registration*, Wiley-Interscience publication, 2005.
5. S. Periaswamy, H. Farid, *Differential Elastic Image Registration*, Department of Computer Science, Dartmouth College, št. TR2001-413, 2001.
6. H. Foroosh, W. S. Hoge, *Motion information in the phase domain*, *Video registration*, Kulwer Academic Publishers, 2003.
7. W. J. Conover, *Practical nonparametric statistics*, John Wiley & Sons, str. 344 – 386, 1980.
8. S. Periaswamy, H. Farid, Medical Image Registration with Partial Data, *Medical Image Analysis*, št. 10, str. 452 – 464, 2006.
9. A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society* 99, str. 1–38, 1977.
10. D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *International Journal of Computer Vision*, vol. 47, str. 7-42, 2002.
11. R. Hartley, *Multiple View Geometry in Computer Vision*, 2. izdaja, Cambridge University Press, 2003.
12. J. Rakun, D. Zazula, Odkrivanje napak v industrijskih tiskovinah s pomočjo slikovne poravnave, *Elektrotehniški vestnik*, vol. 74, št. 3, str. 99-106, 2007.

RAČUNALNIŠKI VID ZA VODENJE AVTONOMNEGA PLOVILA

Janez Perš, Matej Kristan, Vildana Sulič, Stanislav Kovačič

Laboratorij za strojni vid
Fakulteta za elektrotehniko, Univerza v Ljubljani
E-pošta: janez.pers@fe.uni-lj.si
URL: <http://vision.fe.uni-lj.si>

POVZETEK: *Opisana je izvedba algoritma računalniškega vida, ki je v pomoč avtomatskemu vodenju avtonomnega plovila. Algoritem detektira morebitne ovire na morski površini, in posreduje informacije o relativni poziciji ovir centralnemu krmilniku plovila. Cilj projekta je izdelati plovilo, ki se bo pred oviro samodejno ustavilo, ali pa se ji bo samodejno izognilo, izključno na podlagi vizualne informacije.*

1. UVOD

V okviru tehnološkega projekta TP-MIR APSIS¹ je konzorcij partnerjev iz industrije (Harpha Sea d.o.o., Ames d.o.o., IFB d.o.o., Xenya d.o.o., Acorn d.o.o.), in akademskih institucij (UP Pint Primorske univerze in Fakulteta za elektrotehniko Univerze v Ljubljani) zasnoval avtonomno plovilo, ki predstavlja senzorsko platformo za različne civilne in vojaške namene, med drugim:

- analizo vode in zraka,
- analizo morfologije dna,
- določitev objektov v vodi,
- posredovanje komunikacij med objekti pod in nad vodo.

Ker gre za avtonomno plovilo, ki bo delovalo v misijah v obalni coni morja, se je pojavila potreba po senzorju, ki bi znal detektirati različne ovire na morski gladini. Tehnologije, kot sta ultrazvok in radar, so v takšnem plovilu težko uporabne, saj zaradi bližine morske gladine prihaja do motečih odbojev od površine vode. Ker je bil cilj plovilo opremiti tudi z videokamero za prenos slike v nadzorni center, smo v Laboratoriju za strojni vid izdelali algoritem – softverski „senzor“ za zavedanje okolice, ki bi sliko obdelal že na krovu plovila in poslal ustrezne informacije o ovirah centralnemu krmilniku plovila. Centralni krmilnik, na katerem teče tudi program, ki ima

¹ Zahvaljujemo se sofinancerjema, MORS in agenciji TIA, ter vsem ostalim partnerjem na projektu.

funkcijo avtopilota, ima tako na razpolago informacije o morebitnih ovirah v smeri vožnje, ki jih lahko uporabi po lastni presoji.

2. PLOVILO

Plovilo je bilo izdelano v dveh prototipnih izvedbah; prva, ki je prikazana na slikah v tem članku, je nekoliko manjša, saj je temeljila na prilagojenem trupu kajaka, in je služila osnovnim meritvam in razvoju. Po testiranju se je izkazalo, da je zaradi stabilnosti in nosilnosti primeren nekoliko večji trup, ki ima sedaj obliko katamarana. Slika 1 prikazuje plovilo s trupom prve generacije med vožnjo.



Slika 1: Postavitev slikovnega procesorja v prvem prototipu plovila in prototip plovila med vožnjo (zavoj v levo)

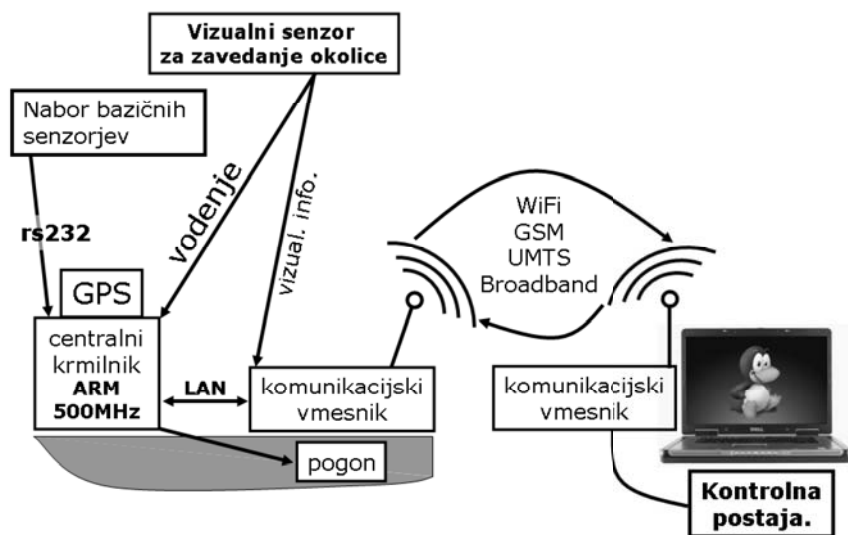
2.1 Oprema plovila

Na krovu druge generacije plovila je naslednja oprema:

- Vzorčevalnik vode z vitlom in črpalko za zajem iz globine nekaj metrov.
- Vremenska postaja z merilniki vetra, sončnega sevanja, in temperature, montiranimi na sprednji jambor.
- Broadband in narrowband digitalna radijska postaja za prenos digitalnega videa in telemetrijo z ustreznimi antenami. Za potrebe testiranja je dodan še UMTS modem, ki se poveže v internet preko Mobitelovega omrežja, tako da je omogočen dostop na daljavo tudi med samimi testnimi vožnjami.
- Električni motor, ki poganja vijak, s servomotorjem za krmiljenje smeri pogona in električnimi baterijami

- Centralni krmilnik z elektronskim kompasom, GPS sprejemnikom in žiroskopskimi senzorji
- Sistem za »zavedanje okolice«, sestavljen iz digitalne omrežne kamere, omrežnih povezav, procesne enote, in lastnega baterijskega napajanja.

Konceptualno strukturo celotnega sistema, skupaj z nadzorno centralo na obali prikazuje Slika 2.



Slika 2: Konceptualna struktura celotnega sistema

3. SISTEM ZA ZAVEDANJE OKOLICE

3.1 Strojna in programska oprema

Strojna oprema je zasnovana modularno, tako da je mogoče plovilo uporabiti tudi v enostavnejši konfiguraciji. Na prednjem jamboru je montirana videonadzorna digitalna kamera Axis 207 v vodoodpornem ohišju. Usmerjena je v smer vožnje. Napajanje prihaja iz centralnega krmilnika, podatki pa se prenašajo v trup plovila po lokalni Ethernet povezavi. Kamera ob porabi cca 5W izvaja kompresijo videa v MPEG4 in MJPEG formatu in ga posreduje vsem zainteresiranim odjemalcem. IP omrežje plovila sega do kontrolne postaje na obali, kjer je mogoče ob delovanju širokopasovne povezave spremljati video v polni ločljivosti.

Slikovni procesor je zgrajen iz »embedded PC« izvedbe osebnega računalnika z napetostnim regulatorjem in lastnim napajanjem, ki omogoča 4-5 ur avtonomije, kljub porabi, ki znaša, odvisno od obremenitve procesorja, v prototipni varianti od 25W do 60W. Slikovni procesor je zgrajen okrog Intel Core 2 duo dvojedrnega procesorja in 2

GB pomnilnika, operacijski sistem in programska oprema pa je nameščena na CompactFlash kartici. Slikovni procesor kamere odjema digitalni video tok v formatu MJPG prek lokalnega omrežja in ga dekodira za nadaljnjo obdelavo.

Na slikovnem procesorju je nameščen Linux in, za prototipno testiranje, Matlab za Linux s prevajalnikom. Preko UMTS povezave v internet je možno delati interaktivno z Matlabom na neomejene razdalje in opazovati živo sliko iz kamere, tudi v času testnih voženj. Ponavadi teče v času vožnje vnaprej prevedena različica programske kode, ki obdeluje sliko, ter arhivira sliko ter rezultate. Na daljavo je po končani vožnji zlahka mogoče prenesti nekaj GB nastalih posnetkov v naš laboratorij v Ljubljani.

3.2 Obdelava slike

Zmogljiv slikovni procesor omogoča testiranje kode, razvite za namizne osebne računalnike, vendar bo v morebitni komercialni različici glede na potrebe koda optimizirana do stopnje, ko ne bo uporabljala Matlaba.

Cilj obdelave slike je pridobiti podatke o ovirah, ki se nahajajo pred plovilom. Uporabnost takšnih podatkov je dvojna:

- Prenos izjemno kompaktne informacije o prizoru na obalo preko ozkopasovnih povezav.
- Avtomatsko izogibanje oviram, ki ga na podlagi teh podatkov izvaja centralni krmilnik plovila, oziroma njegov avtopilot.

4. DETEKCIJA OBJEKTOV NA VODNI GLADINI

Detekcija objektov na vodni gladini je sestavljena iz dveh kritičnih korakov:

- Detekcija in dobra ocena pozicije horizonta, oziroma zgornjega roba gladine morja.
- Detekcija *od površine morja vizualno različnih* objektov v navideznem štirikotniku, ki ga tvorijo rob morja, spodnji rob slike, ter spodnja dela levega in desnega roba slike.

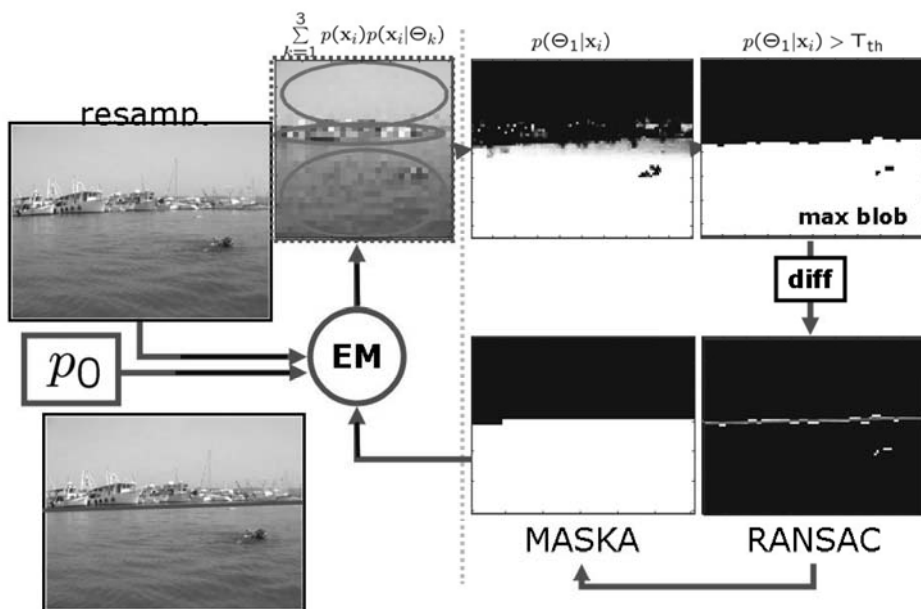
Za izvedbo obeh korakov je pomembno, da imamo na razpolago dober model prizora. Za razliko od mobilnih robotov, ki se premikajo skozi raznoliko okolico, je model prizora, ki ga opazuje plovilo, sorazmerno enostaven: na dnu slike lahko vedno pričakujemo vidno površino morja, ki sega navzgor vse do roba morja, kjer preide ali v pas obale (če plovilo gleda proti obali), ali pa neposredno v pas neba. Tudi obalni pas se načeloma zaključuje z vidnim pasom neba na skrajnem zgornjem robu slike

Težavnost modeliranja tega prizora pa ni v njegovi kompleksnosti, ampak v njegovi *dinamičnosti*. Za razliko od mobilnih robotov, ki lahko praktično kadarkoli *umirijo* svoj pogled na okolico (če se popolnoma ustavijo) imamo v primeru plovila opraviti z neprestanim premikanjem. Plovilo se tudi v ustavljenem stanju ziba in nagiba najmanj

okrog dveh osi, kar se prenese v ustrezno dinamiko slike. Površina morja se konstantno spreminja zaradi valovanja morja, in je na njej nemogoče najti dobre oporne točke s standardnimi metodami za poravnavo slik (npr. SIFT deskriptorji [1]).

4.1 Model prizora

Model prizora je sestavljen iz treh segmentov, ki predstavljajo morje, pas obale, in nebo, kot je to razvidno iz diagrama obdelave slike, ki ga prikazuje Slika 3. Sliko močno podvzorčimo in s pomočjo tri-segmentnega modela ocenimo, kje približno se nahajajo posamezne komponente prizora (morje, obala, nebo). Pri obdelavi prve slike model vodi samo apriori podano znanje, pri obdelavi naslednjih slik pa algoritem upošteva znanje, ki ga je pridobil pri obdelavi prejšnje slike.



Slika 3: Potek obdelave slike.

Model ima vgrajeno apriori znanje, ki onemogoča poljubno prilagajanje pozicije komponent. Tako velja, da je morje vedno v spodnjem delu slike, nebo pa v zgornjem delu. Postopek je iterativen. Pozicijo posameznih komponent prizora ocenimo z EM algoritmom (Expectation Maximization), in tako dobimo za vsak slikovni element aposteriori verjetnost, da je barve vode. Ta verjetnost je uporabljena za segmentacijo roba morja, ki jo prikazuje zaporedje slik na desni strani Slike 3. Po binarizaciji slike je na robu največjega področja uporabljen algoritem RANSAC [2], s čimer dobimo linijo roba morja, iz te pa je generirana maska, ki v naslednji iteraciji predstavlja apriori verjetnost, da slikovni elementi vsebujejo vodo. Ves postopek, vključno z EM se ponavlja, dokler ne konvergira, število iteracij pa je omejeno. Tako dobimo zgornji rob morja. »Znanje« o barvni porazdelitvi slikovnih elementov na sliki in poziciji področij na

sliki pa propagiramo v obdelavo naslednje slike, kjer služi kot začetni približek iskanja segmentov na sliki.

Ko je rob morja določen, preverimo, kateri slikovni elementi pod robom morja po barvi odstopajo od večine. Detekcijo regij takšnih odstopajočih slikovnih elementov izvedemo z detektorji MSER [3].

4.3 Kompresija slike

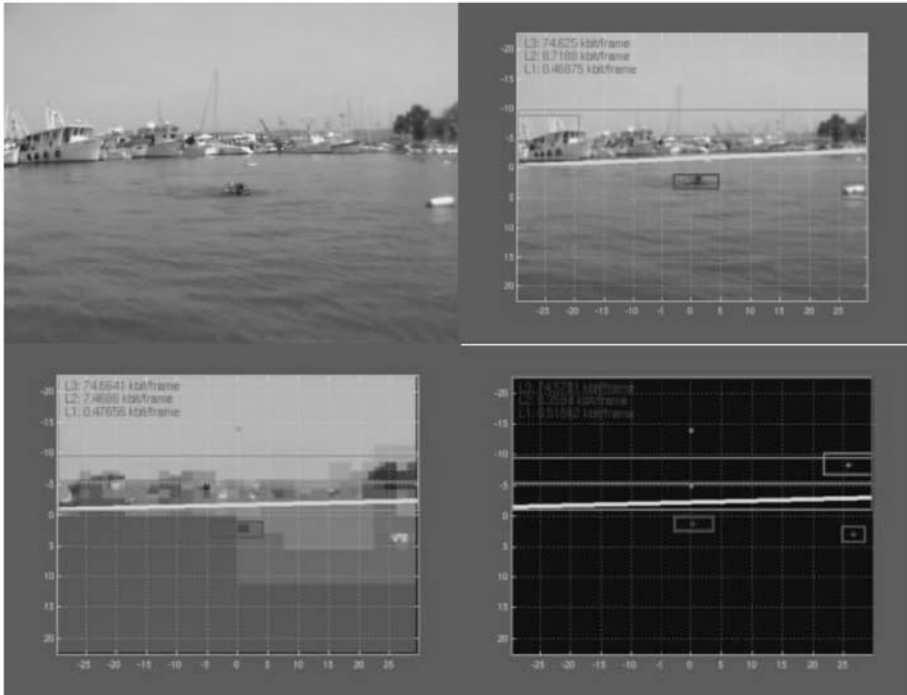
Širokopasovna radijska povezava ima omejen doseg, in slike preko MPEG4 toka ne bo vedno mogoče prenašati. Zato je detekcija objektov uporabljena tudi kot del večstopenjske izgubne slikovne kompresijske sheme, ki bi v skrajni sili omogočala tudi ročno vodenje plovila, tudi če prenos slike preko širokopasovne povezave ni na voljo. Slikovni procesor vsako obdelano sliko stisne v JPEG zapis s parametri, ki generirajo približno 12 KB na sliko, in jo pošlje preko podatkovne povezave kontrolni postaji na kopno. Potem izvede še štiriško dekompozicijo (angl. quadtree decomposition) izvorne slike, in po ustrezni kvantizaciji dobi kodirano predstavitev, dolgo približno 1KB, ki jo ravno tako pošlje na kopno. Prednost takšne predstavitve je, da je slika kvalitetnejša kot JPEG kodirana slika pri isti dolžini. Nazadnje ustrezno kvantizira še pozicije detektiranih objektov in roba morja, in jih pošlje kontrolni postaji, ti podatki pa zajemajo v običajnih okoliščinah le kakšnih 100 bajtov. Glede na zmožljivost prenosne poti lahko komunikacijski vmesnik zavrže nekatere podatkovne pakete, in postaja na kopnem rekonstruira sliko na najboljši možni način. V skrajnem primeru, pri najpočasnejšem delovanju povezave bi tako postaja prikazovala operaterju le detektirane pozicije, v primeru zmogljivejše povezave pa bi dodala še ostale podatke.

5. DELOVANJE SISTEMA

Uporaba Matlab a onemogoča hitro procesiranje slike, čeprav so za najbolj potratne dele algoritma (npr. MSER) uporabljene knjižnice, pisane v jeziku C. Trenutna hitrost obdelave je 2 sliki na sekundo, kar pa je dovolj za delovanje pri zmernih hitrostih – plovilo se pred dobro vidnimi objekti pravočasno ustavi.

Za razvijalce sistema računalniškega vida je bilo veliko presenečenje delovanje algoritma na mirnem in razburkanem morju. V nasprotju s pričakovanji algoritem neprimerno bolje deluje ob zmerno vzvalovanem, ali pa celo močnejše vzvalovanem morju, kot pa ob poponoma mirnem morju. Izkazalo se je, da je pri vzvalovanem morju površina dokaj enakomerne modro-zelene barve, kar daje dober kontrast z nebom (kar koristi iskanju roba morja) in ustreza predpostavkam, vgrajenim v model prizora. V mirnem morju pa prihaja do pojava izrazitega zrcaljenja okolice v morje, tako da sistem izjemno težko razloči objekte od odsevov na vodni površini. V zgodnjih jutranjih urah v poletnem času je ob mirnem morju horizont praktično neviden, kar pravzaprav onemogoča delovanje algoritma. Rešitev tega problema je potencialno v uporabi žiroskopskih meritev, vendar pa so ti podatki bistveno manj natančni (med razvojem plovila je bil horizont, določen s strani našega algoritma celo uporabljen za umerjanje žiroskopa, saj predstavlja, v

primeru dobre vidljivosti najbolj primerno referenco. Tudi uporaba polarizacijskega filtra ni odstranila večine odsevov.



Slika 4: Levo zgoraj: zajeta slika, s potapljačem, ki je priplaval na površje. Desno zgoraj: prikaz detekcije potapljača in bele boje, ozadje je JPEG kodirana slika. Levo spodaj: prikaz detekcij, položen čez štiriško kodirano sliko. Desno spodaj: prikaz detekcij na praznem ozadju. V vseh primerih je z zeleno barvo označen prikaz objektov nad robom morja – gre za funkcionalnost, ki v članku ni opisana in večinoma ni uporabljana.

LITERATURA

1. Lowe, D. G. (2004), Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, vol 60, str. 91-110.
2. M.A. Fischler in R.C. Bolles (1981), Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography.
3. J. Matas, O. Chum, M. Urban, in T. Pajdla (2004), Robust wide-baseline stereo from maximally stable extremal regions, *Image and Vision Computing*, vol. 22, str. 761-767.

MEDICINSKE IN BIOMEDICINSKE APLIKACIJE

UPORABA PORAZDELJENEGA PROCESIRANJA PRI OBDELAVI ULTRAZVOČNIH VOLUMNOV JAJČNIKOV

Sebastijan Šprager, Jurij Munda, Damjan Zazula

Laboratorij za sistemsko programsko opremo

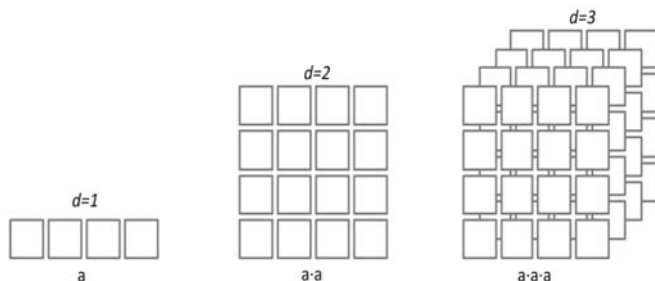
Fakulteta za elektrotehniko, računalništvo in informatiko Maribor

E-pošta: {sebastijan.sprager, jurij.munda, zazula}@uni-mb.si

POVZETEK: *Prispevek govori o uporabi porazdeljenega procesiranja pri obdelavi ultrazvočnih volumnov jajčnikov. Obdelava tridimenzionalnih posnetkov v večini primerov predstavlja časovno zahtevno operacijo, zaradi različnih potreb pa je zaželeno čim hitrejšo izvajanje postopkov obdelave. Ena izmed poglavitnih možnosti za pohitritev računanja predstavlja porazdeljeno procesiranje. S pomočjo matlabovega orodja smo zgradili sistem za porazdeljeno računanje. Učinkovitost smo preverili z dvema časovno potratnima postopkoma za obdelavo ultrazvočnih volumnov jajčnikov, in sicer s programskim simulatorjem za tvorbo ultrazvočnih volumnov ter algoritmom za njihovo elastično poravnavanje. Rezultati kažejo, da se z uporabo porazdeljenega procesiranja časi računanja znižajo tudi na 20 % časa, sicer potrebnega za enoprocesorsko računanje.*

1 UVOD

Obdelava tridimenzionalnih posnetkov (volumnov) na področju biomedicine postaja vse bolj prisotna. Omogoča dodano vrednost pri medicinski diagnostiki in podporo pri zdravljenju. V bistvu gre za procesiranje tridimenzionalnih signalov, pri čemer imamo opravka z veliko količino podatkov. Če enodimenzionalni signal preslikamo v tridimenzionalnega z enako dolžino, njegova razsežnost narašča s tretjo potenco, kot vidimo na sliki 1. Zato obdelava volumnov lahko predstavlja časovno zahtevno operacijo. Nekateri problemi so tako zahtevni, da procesiranje traja nekaj ur, celo dni. V takih primerih razvijalcem postopkov za obdelavo predstavlja velik izziv optimizacija postopka, tako da bi minimizirali čas procesiranja. To lahko izvedemo na dva načina. Prvi način predstavlja optimizacija programske kode, pri kateri z modifikacijo programske kode njene težke dele spremenimo tako, da zmanjšamo njihovo časovno zahtevnost. Velikokrat pa ta način ne zadostuje. Takrat se moramo zateči k drugemu načinu, pri katerem uporabimo porazdeljeno procesiranje. V tem primeru programsko kodo oziroma težke del programske kode razbijemo na več podenot, ki se lahko izvajajo sočasno, vsak del na svojem računalniku oziroma jedru.



Slika 1: Shematski prikaz količine informacij pri različnih dimenzijah signala dolžine a .

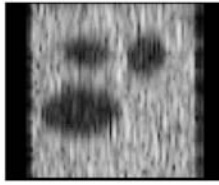
V Laboratoriju za sistemsko programsko opremo na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru se ukvarjamo z računalniško podprtim obdelavanjem ultrazvočnih posnetkov jajčnikov. Razvili smo številne postopke za njihovo obdelavo [1, 2, 3, 4]. Ker sta dva izmed njih precej časovno potratna in dokaj pogosto uporabljena, se je pojavila potreba po njunem hitrejšem izvajanju. To smo dosegli z uporabo porazdeljenega procesiranja.

Omenjena postopka, torej postopek za simulacijo ultrazvočnih posnetkov jajčnikov ter postopek za elastično poravnavanje njihovih ultrazvočnih volumnov, sta opisana v drugem poglavju. Čas izvajanja omenjenih postopkov smo zmanjšali s sistemom za porazdeljeno procesiranje, zgrajeno s pomočjo programskega orodja matlab. Sistem je opisan v tretjem poglavju. V četrtem poglavju sledi predstavitev rezultatov, prispevek pa sklenemo z razpravo v petem poglavju.

2 POSTOPKA ZA SIMULACIJO IN ELASTIČNO PORAVNAVANJE ULTRAZVOČNIH VOLUMNOV

2.1 Simulacija ultrazvočnih posnetkov jajčnih mešičkov

Za preverjanje učinkovitosti postopkov za obdelavo ultrazvočnih posnetkov jajčnikov, v našem primeru segmentacije in spremljanja rasti jajčnih mešičkov, potrebujemo nabor testnih posnetkov [1]. Ker za realne posnetke ne moremo vnaprej poznati njihove natančne vsebine in strukture, niso neposredno primerni za preverjanje in potrjevanje novo razvitih analizijskih postopkov. Zato smo svoje postopke preverjali s posnetki, ki smo jih izdelali s pomočjo simulatorja. Jajčnike smo predstavili z elipsoidi, ki smo jih postavili v prostor vokslov, nanj pa aplicirali ultrazvočni šum. Omenjeni model ima velikost $100 \times 100 \times 100$ vokslov, kar je podobno velikosti realnih posnetkov z volumni jajčnikov, ki jih imamo na voljo. Primer prereza simuliranega volumna jajčnika je prikazan na sliki 2.



Slika 2: Prerez skozi simulirani volumen jajčnika: temna območja predstavljajo jajčne mešičke.

Za simulacijo ultrazvočnih volumnov smo uporabili programski simulator Field II [8], ki je najpogosteje uporabljeni simulator glede na navedbe v literaturi. Teče v okolju matlab, njegovo jedro pa je pisano v programskem jeziku C.

Ultrazvočne slike se tvorijo s pomočjo točk, ki raztresajo ultrazvok (*scatter*). Reprezentativnost rezultata je v veliki meri odvisna prav od števila in odbojnosti teh točk, ki so izbrane naključno znotraj simuliranih jajčnikov v obliki elipsoidov. Prav ta postopek pa vnaša veliko računsko zahtevnost, ki narašča sorazmerno s številom teh točk. V našem primeru je točk, na katerih se odbijajo ultrazvočni valovi, okrog 70 % vseh točk v modelu jajčnika, kar pomeni pri izbrani velikosti jajčnikov 700.000 točk.

Računanje v simulatorju predstavlja zelo dolgotrajen postopek. Pri predstavljenih parametrih traja simulacija posameznega volumna kar 11 ur. Postopek za izračun odbojnih valov, ki predstavlja najtežji del programske kode, je zasnovan tako, da je odboje mogoče računati ločeno in hkrati sočasno, kar nas je napeljalo k uporabi porazdeljenega procesiranja, opisanega v tretjem poglavju.

2.2 Elastično poravnavanje ultrazvočnih volumnov jajčnih mešičkov

Postopek za elastično poravnavanje ultrazvočnih posnetkov smo uporabili pri spremljanju rasti jajčnih mešičkov in pri ugotavljanju lokalnih sprememb mešičkov [4]. Pristop temelji na enakosti povprečne vrednosti pegastega šuma v primerjanih ultrazvočnih posnetkih [6]. Razliko med primerjanima volumnoma lahko opišemo z naslednjim deformacijskim modelom:

$$\mathbf{u}(\mathbf{x}) = \mathbf{x} + \sum_{\mathbf{k} \in \mathbb{Z}^3} \mathbf{c}_{\mathbf{k}} \beta^3 \left(\frac{\mathbf{x}}{\mathbf{h}} - \mathbf{k} \right), \quad (1)$$

kjer \mathbf{x} predstavlja izbrani voksel, $\mathbf{c}_{\mathbf{k}} = (c_{k,x}, c_{k,y}, c_{k,z})$ deformacijske parametre, $\mathbf{h} = (h_x, h_y, h_z)$ pa razdaljo v vokslah med dvema vozliščema v B-zlepkih, označenih z β^3 . Enačba (1) določa gostoto mreže 3D B-zlepkov, ki jih umestimo v poravnani volumen [5].

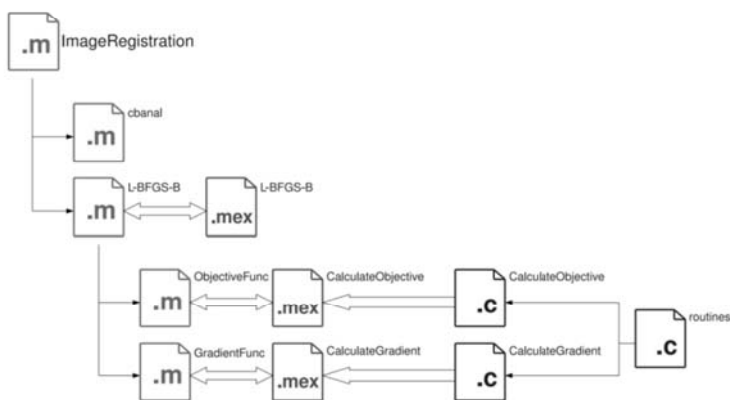
Optimalno elastično poravnavo posnetkov iščemo s prilagajanjem deformacijskih parametrov, pri čemer vrednost posameznega deformacijskega parametra vpliva na

deformacije volumna v okoli opazovanega voksla \mathbf{x} . Ustrezne vrednosti deformacijskih parametrov iščemo s pomočjo optimizacijskega algoritma L-BFGS-B [7]. Ocenitvena funkcija je predstavljena z naslednjo enačbo:

$$E = \frac{1}{N} \sum_{\mathbf{x} \in \mathbb{Z}^3} (\ln(\exp(2\mathbf{r}(\mathbf{x})) + 1) - \mathbf{r}(\mathbf{x})), \quad (2)$$

kjer $\mathbf{r}(\mathbf{x})$ predstavlja razliko med deformiranim oziroma končnim volumnom in začetnim oziroma referenčnim volumnom [6]. Postopek deluje iterativno, tako da skušamo začetni volumen približati končnemu, in se zaključi, ko je razlika med volumnoma (ocena E) dovolj majhna.

Tudi algoritem za elastično poravnavanje je pretežno programiran v programskem jeziku matlab. Na sliki 3 je prikazana struktura celotnega algoritma. Skript `cbanal` uporabimo za izračun interpolacijskih koeficientov. Ključni del algoritma predstavlja skript `L-BFGS-B`, s pomočjo katerega zaženemo optimizacijski algoritem L-BFGS-B. Izračun ocenitvene funkcije in gradienta optimizacijski algoritem opravi s klicema skriptov `ObjectiveFunc` ter `GradientFunc`.



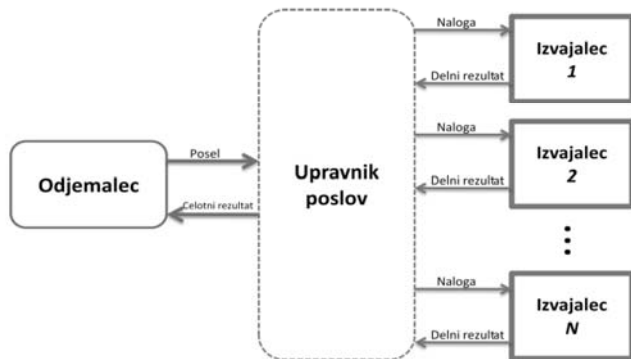
Slika 3: Struktura za izvedbo algoritma za elastično poravnavanje ultrazvočnih volumnov.

Že s pomočjo matlabovega profilirnika smo ugotovili, da se pri izvajanju algoritma pojavljajata dva odseka težke kode, in sicer pri izračunu ocenitvene funkcije ter določanju deformacijskih parametrov, ko iterativno računamo gradient. Njuno računanje se izvaja celo na nivoju posameznega voksla. Zato smo te dele kode programirali v programskem jeziku C in jih prevedli s pomočjo orodja MEX in prevajalnika za C (GCC). Že s tem korakom smo pridobili na zmogljivosti, kar je zadostovalo za računanje ocenitvene funkcije v posamezni iteraciji, ne pa tudi za računanje deformacijskih parametrov pri računanju gradienta.

Zasnova algoritma za računanje omenjenih parametrov pa omogoča vzporedno izvajanje, kar smo s pridom izkoristili in vpeljali porazdeljeno procesiranje.

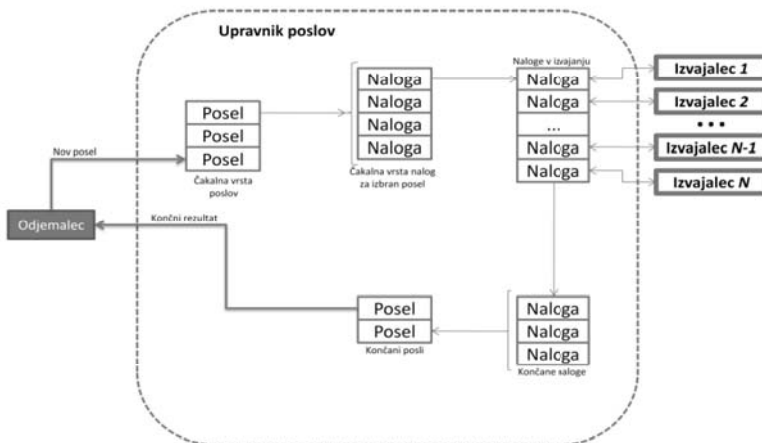
3 SISTEM ZA PORAZDELJENO PROCESIRANJE S POMOČJO PROGRAMSKEGA ORODJA MATLAB

Da bi pohitrili izvajanje postopkov, opisanih v prejšnjem poglavju, smo s pomočjo dveh matlabovih orodij, ki delujeta z roko v roki, sestavili sistem, s pomočjo katerega smo izvedli porazdeljeno procesiranje omenjenih postopkov [9, 10]. Orodje za vzporedno računanje (*Parallel Computing Toolbox*) in strežnik za porazdeljeno računanje (*Distributed Computing Server*) omogočata računanje računsko in podatkovno zahtevnih problemov na večjedrnih oziroma večprocesorskih sistemih. Številni sestavni deli orodja omogočajo vzporedno računanje na višjem nivoju abstrakcije brez dodatnega programiranja za specifično strojno opremo oziroma mrežno arhitekturo.



Slika 4: Logična arhitektura sistema za porazdeljeno procesiranje s pomočjo programskega orodja matlab.

Logična arhitektura uporabljenega sistema za porazdeljeno procesiranje je prikazana na sliki 4. Celotni sistem lahko razdelimo na tri sloje. Najvišji sloj predstavlja uporabnikov računalnik, imenovan odjemalec (*client*), v najnižjem sloju je skupina izvajalcev (*workers*), upravnik poslov (*job manager*) pa predstavlja vmesni sloj. Posel (*job*) je večja operacija, ki jo uporabnik želi izvesti. Definira se znotraj odjemalca v navadni matlabovi seji. Posel se s strani uporabnika vnaprej razdeli na manjše podenote, imenovane naloge (*tasks*). Uporabnik znotraj odjemalca s pomočjo orodja za vzporedno procesiranje definira posel in posledično tudi naloge. Matlabov strežnik za porazdeljeno procesiranje, ki teče na izvajalcih, izvede računanje posameznega posla z ovrednotenjem vsake izmed njegovih nalog, rezultati izračunov se kot končan posel vrnejo na uporabnikov računalnik – odjemalec.

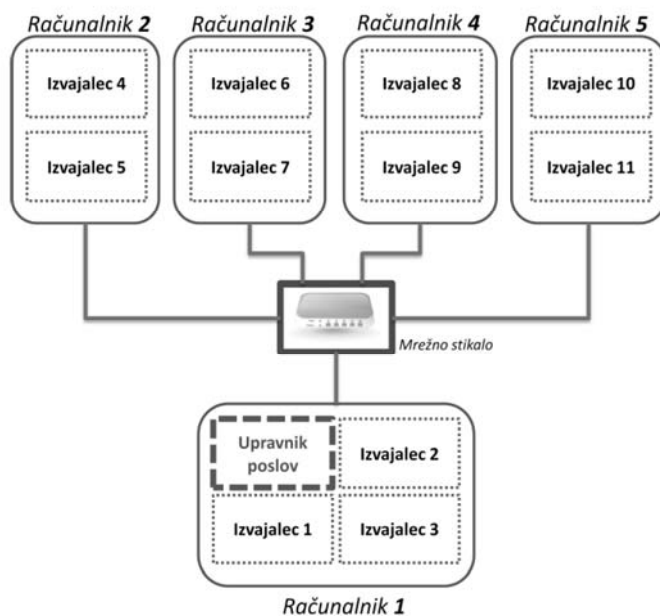


Slika 5: Vloga upravnika poslov v sistemu za porazdeljeno procesiranje.

Na sliki 5 je podrobneje prikazana vloga upravnika poslov znotraj našega sistema za porazdeljeno procesiranje. Uporabnik na odjemalcu znotraj matlaba definira posel, ki ga razdeli na naloge. Prispeli posli se pri upravniku poslov postavijo v čakalno vrsto. V naslednji fazi se v čakalno vrsto postavijo naloge posla, vzetega iz čakalne vrste. Upravnik posreduje posamezne naloge posla iz čakalne vrste prostim izvajalcem. Ko izvajalec nalogo izvede, rezultat posreduje upravniku poslov, ta pa mu dodeli naslednjo nalogo, ki čaka v vrsti. Rezultate nalog končanega posla upravnik nato v zadnji fazi posreduje odjemalcu v obliki končnih rezultatov. V našem primeru smo v posel vključili tvorbo ultrazvočnih odbojev pri simulaciji ultrazvočnih posnetkov ter računanje gradienta v ocenitveni funkciji pri elastičnem poravnavanju teh posnetkov. V prvem primeru pomeni posamezna naloga računanje posameznega odboja, v drugem primeru pa računanje posameznega parametra za določanje gradienta v ocenitveni funkciji.

3.1 Namestitev strojne opreme

Sistem za porazdeljeno procesiranje smo namestili na pet računalnikov. Računalnike smo povezali med seboj s pomočjo mrežnega stikala. Na vseh je tekel operacijski sistem Ubuntu Linux 9.10 z nameščenim matlabom različice 7.6. Prvi računalnik vsebuje štirijedrni procesor Intel Core 2 Quad Q6600, kjer posamezno jedro deluje s taktom 2,4 GHz in dosega 8 GB delovnega pomnilnika. Na enem jedru je tekel upravnik poslov, na preostalih treh pa trije izvajalci. Tudi na preostalih štirih računalnikih je tekel enak operacijski sistem z enako različico matlaba. Ti računalniki so enakega tipa, vsi vsebujejo dvojedrni procesor Intel Core 2 Duo E7500, kjer posamezno jedro deluje s taktom 2,93 GHz, imajo pa vgrajenega 2 GB delovnega pomnilnika. Na vsakem računalniku sta tekla dva izvajalca, na vsakem jedru eden.



Slika 6: Arhitektura sistema za porazdeljeno procesiranje.

4 REZULTATI

4.1 Simulacija ultrazvočnih volumnov s pomočjo porazdeljenega računanja

Simulacijo ultrazvočnega volumna z lastnostmi, opisanimi v podpoglavju 2.1, smo izvedli s porazdeljenim računanjem. Simulirali smo dva volumna z različnim številom odbojev (*scatter*). Čas simulacije na enem procesorju smo ocenili glede na prej opravljene meritve [8], saj bi tako računanje sicer trajalo predolgo. Časi računanja so prikazani v tabeli 1. Opazna je bistvena razlika v zmogljivosti, saj sistem s porazdeljenim procesiranjem svojo nalogo opravi v povprečju zgolj v približno 3,5 % časa, ki bi bil potreben za izračun na enoprocetni strojni oprepi.

Število točk za odboj ultrazvočnih valov	Ocenjeni čas enoprocorskega računanja [h]	Čas porazdeljenega računanja [h]
175.000	38,5	0,9 (55 min)
700.000	154 (6 dni)	7

Tabela 1: Primerjava časov računanja pri simulaciji ultrazvočnih volumnov.

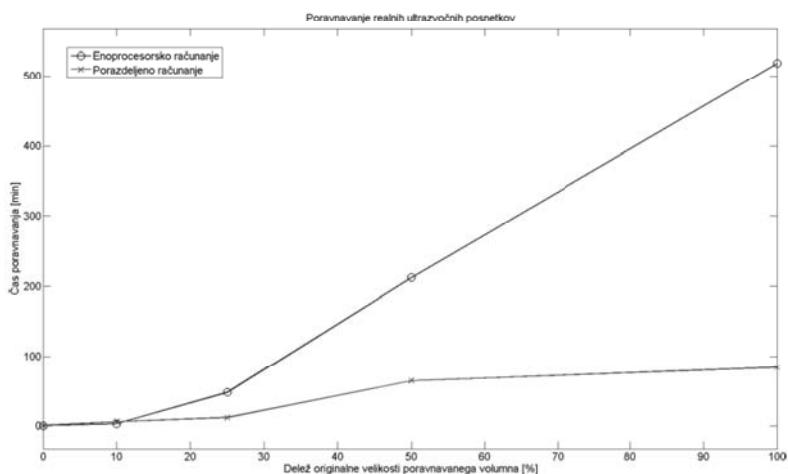
4.2 Elastično poravnavanje ultrazvočnih volumnov s pomočjo porazdeljenega računanja

Elastično poravnavanje ultrazvočnih volumnov smo izvajali z dvema paroma volumnov. Prvi par je predstavljal realna ultrazvočna volumna jajčnikov, drugi par pa je bil izdelan s pomočjo simulacije. Vsak par smo nato pretvorili v tri pare s spremenjeno velikostjo, in sicer na 10 %, 25 % in 50 % velikosti. Vsak pomanjšani par smo poravnavali z enoprocesorskim računalnikom, nato pa še s pomočjo sistema za porazdeljeno procesiranje. Časi poravnavanja realnih posnetkov so prikazani v tabeli 2.

<i>Velikost volumna (delež originalne velikosti)</i>	<i>Čas enoprocesorskega računanja [min]</i>	<i>Čas porazdeljenega računanja [min]</i>
12 x 10 x 15 (10%)	2,6	5
30 x 26 x 37 (25%)	47	11
60 x 51 x 74 (50%)	212 (3,5 h)	64 (1 h)
121 x 103 x 149 (100%)	518 (8,6 h)	83 (1,4 h)

Tabela 2: Primerjava časov za poravnavanje različno velikih realnih ultrazvočnih volumnov.

Vidimo, da porazdeljeno računanje precej skrajša čase, potrebne za poravnavanje. Algoritem za elastično poravnavanje tako v povprečju teče le 23,1±12,5 % časa, potrebnega za enoprocesorsko računanje, če upoštevamo poravnavanje parov pri volumnih s 50 % in 100 % originalne velikosti. Upoštevati je namreč treba, da se pri manjših volumnih bolj splača enoprocesorsko računanje, saj tudi komunikacija in prenos podatkov med upravnikom poslov, izvajalci in odjemalcem vnaša časovne zakasnitve, kar pri manjši količini podatkov postane precej izrazito. Graf s časi za poravnavanje realnih ultrazvočnih volumnov jajčnikov je prikazan na sliki 7.

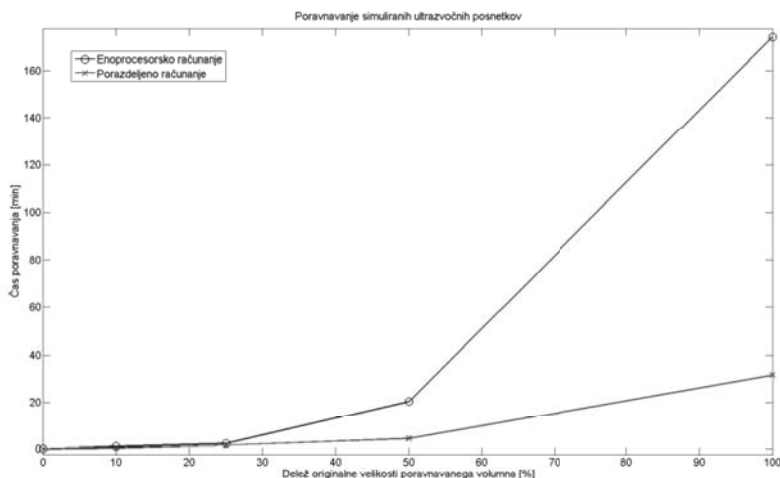


Slika 7: Graf časov za poravnavanje realnega ultrazvočnega volumna pri različnih velikostih.

Podobne rezultate smo dobili tudi pri poravnavanju sintetičnih ultrazvočnih volumnov (tabela 3). Tudi v tem primeru so rezultati dokaj podobni, saj je poravnava s pomočjo porazdeljenega procesiranja trajala v povprečju $20,15 \pm 7,8$ % celotnega časa, potrebnega za enoprocesorsko računanje. Graf s časi za poravnavanje simuliranih ultrazvočnih volumnov jajčnikov je prikazan na sliki 8.

Velikost volumna (delež originalne velikosti)	Čas enoprocesorskega računanja [min]	Čas porazdeljenega računanja [min]
10 x 10 x 10 (10%)	1,1	0,5
25 x 25 x 25 (25%)	2,3	1,6
50 x 50 x 50 (50%)	20	4,5
100 x 100 x 100 (100%)	174 (2,9 h)	31

Tabela 3: Primerjava časov za poravnavanje simuliranega ultrazvočnega volumna pri različnih velikostih.



Slika 8: Graf časov za poravnavanje simuliranega ultrazvočnega volumna pri različnih velikostih.

5 ZAKLJUČEK

V prispevku smo opisali sistem za porazdeljeno procesiranje s pomočjo programskega orodja matlab, s katerim smo pospešili dva časovno zahtevna postopka obdelovanja ultrazvočnih volumnov jajčnikov. Pri tvorbi sintetičnih ultrazvočnih posnetkov jajčnikov smo pokazali, da se je čas računanja drastično zmanjšal, in sicer na 2 % časa, potrebnega pri računanju z enoprocesorskim sistemom. Tako računanje v tem primeru niti ne bi bilo smiselno, zato je uporaba porazdeljenega procesiranja edina uporabna pot. Tudi pri elastičnem poravnavanju ultrazvočnih posnetkov jajčnikov se je izkazala smiselnost uporabe porazdeljenega računanja, saj predstavljajo časi za poravnavanje v

povprečju le okoli 20 % časa, potrebnega pri poravnavanju z enoprocesorskim računalnikom. Pri nadaljevanjih raziskavah bi kazalo preizkusiti vplive različnih konfiguracij strojne opreme (večji oziroma spremenjen nabor računalnikov) ter spremenjeno konfiguracijo za tvorbo nalog znotraj poslov, kjer bi ugotavljali, ali se spleča večja razdrobljenost (veliko število manjših nalog) ali nasprotno (število nalog je enako številu izvajalcev). Smo pa že sedaj z uporabo porazdeljenega procesiranja dobili učinkovit pripomoček, s katerim smo pri raziskavah prihranili veliko časa in omogočili izvedbo nalog, ki sicer ne bi bile realno izvedljive.

LITERATURA

1. B. Cigale, Odkrivanje in spremljanje rasti struktur z večmerilno obdelavo 3D ultrazvočnih posnetkov, doktorsko delo, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, 2007
2. B. Cigale, D. Zazula, Segmentation of Ovarian Ultrasound Images Using Cellular Neural Networks, *International Journal of Pattern Recognition and Artificial Intelligence*, 18, (2004), 4, str. 563-581
3. S. Šprager, D. Zazula, Poravnavanje ultrazvočnih posnetkov s pomočjo lokalno nespremenljivega povprečja pegastega šuma, *ERK 2008*, zvezek B, (2008), str. 261-264
4. S. Šprager, B. Cigale, D. Zazula, Estimation of Growth of Ovarian Follicles using Rigid and Elastic Ultrasound Registration, *BIOSIGNALS 2010 proceedings - Valencia, Spain*, (2010), str. 333-338
5. J. Kybic, M. Unser, Fast Parametric Elastic Image Registration, *IEEE Transactions on Image Processing*, 12, (2003), 11, str. 1427-1442
6. Y. Yue, J. W. Clark, D. W. Khoury, Speckle Tracing in Intracardiac Echocardiography for the Assessment of Myocardial Deformation, *IEEE Transactions on Biomedical Engineering* (v tisku)
7. C. Zhu, R. H. Byrd, J. Nocedal, L-BFGS-B – FORTRAN routines for large scale bound constrained optimization, *ACM Transactions on Mathematical Software*, 23, (1997), 4, str. 550-560
8. J. A. Jensen, Field: A Program For Simulation Ultrasound Systems, *10th Nordic-Baltic Conference on Biomedical Imaging*, (1996), str. 351-353
9. MATLAB, Parallel Computing Toolbox 4, *User's Guide*, (2009)
10. MATLAB, Distributed Computing Server 4, *System Administrator's Guide*, (2009)

VARIABILNOST LOBANJE DINARSKE VOLUHARICE, *DINAROMYS BOGDANOVI*

Tina Klenovšek², Franc Janžekovič², Boris Kryštufek¹,

¹Znanstvenoraziskovalni center, Univerza na Primorskem, Garibaldijska 1, 6000 Koper, E-pošta: boris.krystufek@zrs.upr.si

²Fakulteta za naravoslovje in matematiko, Univerza v Mariboru, Oddelek za biologijo, Koroška cesta 160, 2000 Maribor
E-pošta: tina.klenovsek@uni-mb.si, franc.janzekovic@uni-mb.si

POVZETEK: V prispevku predstavljamo morfometrično variabilnost ventralne strani lobanje dinarske voluharice, *Dinaromys bogdanovi*. Na osnovi geografske razširjenosti in polimorfnosti mitohondrijske DNA so definirani trije filogeografski vzorci. Z metodami geometrijske morfometrije in multivariatne statistike smo preverili skladnost med vzorcem razširjenosti in genetske variabilnosti v primerjavi z morfometričnimi značilnostmi ventralne strani lobanje. Ugotovili smo, da se morfometrične lastnosti lobanje skladajo z dosedaj opisano genetsko variabilnostjo in geografsko razširjenostjo. Prav tako ugotavljamo, da na vzorcih dinarske voluharice lahko z metodami geometrijske morfometrične analize ustrezno opišemo filogenetske odnose med evolucijskimi skupinami.

1. UVOD

Dinarska voluharica, *Dinaromys bogdanovi* (V. & E. Martino, 1922), je endemični glodavec dinarskega krasa. Telo skupaj z glavo v povprečju meri med 99 in 152 mm. Dolžina repa, ki je navadno daljši od polovice dolžine glave in telesa, pa meri med 74 in 119 mm, tehta od 30 do 82 gramov. Kožuh je gost, mehak, dlaka je zmerno dolga, najdaljša je na hrbtu (17 do 18 mm), po hrbtu je rjavkasto sive barve, trebušna stran je vedno sivkasto bela (Petrov & Todorović 1982).

Današnji areal razširjenosti sega od severnega Velebita in Male Kapele na Hrvaškem, preko Bosne in Hercegovine, Črne gore in Kosova do Galičice v Makedoniji in do Albanije, morda sega celo v Grčijo (Petrov & Todorović 1982).

Življenjsko okolje dinarske voluharice so ozke špranje med skalovjem, naseljuje pa tudi vhode v kraške jame. Ponavadi sega bivališče zelo globoko v skalovje. Vegetacija voluharičinega življenjskega prostora se spreminja od makije, preko listnatega do iglastega gozda vse do zgornje gozdne meje (Petrov & Todorović 1982).

Lobanja dinarske voluharice je značilna glodavska z izrazito diastemo, brez podočnikov in predmeljakov, glede na potek žvekalnih mišic sodi v skupino miomorfa. Za lobanje

dinarske voluharice so že značilne morfološke strukture: postorbitalni greben oziroma čvrš na skvamozumu in prizmatični hipsodontni meljaki. Prvi spodnji meljak (m1) ima poleg zadnje zanke in prednje pentlje še 5 trikotnikov, njegova žvekalna površina meri v dolžino 3,2 do 3,9 mm, celotna dolžina spodnjih meljakov pa znaša 7,2 do 8,5 mm.

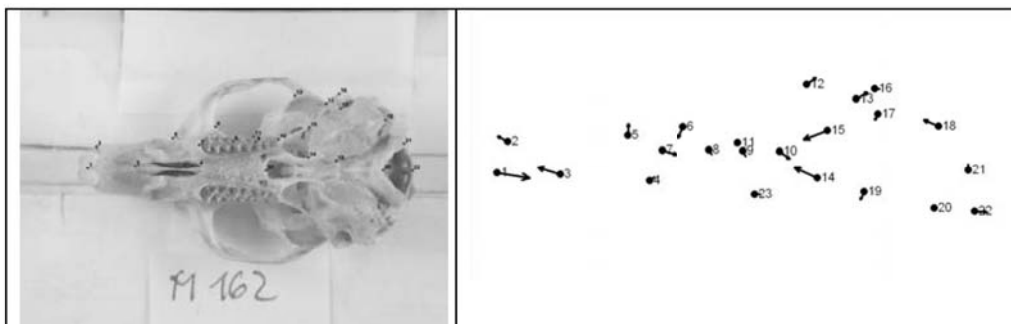
V starejši literaturi (Bolkay 1924, Đulić & Mirić 1967, Martino 1948, Todorović 1956) je opisanih več geografskih oblik dinarske voluharice, nekateri od navedenih avtorjev jim celo pripisujejo status podvrst. V novejši analizi na osnovi mitohondrijske DNA ugotavljajo samo tri genotipske skupine, ki se med seboj tudi geografsko dobro razmejene, severozahodna skupina sega od Velike Kapele na Hrvaškem in proti jugu do reke Neretve, centralna skupina je razširjena na širšem območju Dinaridov med rekama Neretva in Drim, jugovzhodna skupina pa je razširjena južno od reke Drim, predvsem v Makedoniji in Albaniji (Kryštufek s sod. 2007). Na podlagi polimorfnosti mitohondrijske DNA, deloma pa je podprto tudi s fosilnimi najdbami, se je dinarska voluharica iz prvotne razširjenosti (jugovzhodni obronki Alp širila v južni smeri po Dinarskem gorstvu ob jadranski obali). Geografska in reprodukcijska izolacija se je na reki Neretvi vzpostavila pred približno 1 milijonom let, pred približno 300.000 leti je voluharica napredovala južneje od reke Drim.

Na podlagi ugotovitev o treh dobro definiranih filogeografskih skupinah, ter s pomočjo geometrijske morfometrične analize, predstavljamo morfološko variabilnost ventralne strani lobanje med tremi skupinami dinarske voluharice.

2. METODE GEOMETRIJSKE MORFOMETRIJE

V analizo geometrijske morfometrije dinarske voluharice smo vključili 99 lobanj odraslih dinarskih voluharic, material izvira iz celotnega areala vrste. Lobanje voluharic so izposojene iz zbirk Prirodoslovnega muzeja Slovenije, Oddelka za zoologijo Univerze v Zagrebu in Narodnega muzeja Bosne in Hercegovine, Sarajevo.

V analizo so bile vključene odrasle živali, mlade (mlajše od 3 mesecev) smo izločili zaradi izrazitega vpliva rasti, ki v tej fazi še ni zaključena. Spolni dimorfizem smo preverjali z enosmerno analizo variance.



Slika 1. Ventralna stran lobanje dinarske voluharice, *Dinaromys bogdanovi*, s 23 oslonilnimi točkami (levo) in smeri vektorjev kot jih izkazujejo morfološke značilnosti ventralne strani lobanje severozahodne filogeografske skupine (desno).

Geometrijsko morfološko analizo smo izvedli samo za ventralno stran lobanje, saj je ventralna stran morfološko in genetsko veliko bolj raznolika in vsebuje različne neodvisne filogenetske komponente variabilnosti (Caumul & Polly 2005).

Lobanje smo fotografirali z digitalno kamero povezano z osebnim računalnikom, oslonilne točke pa določali s pomočjo računalniškega programa tpsDig (Rohlf 2006). Podatke smo v predhodnem postopku posplošene Prokrustove analize, ki zajema premik, rotacijo in skaliranje konfiguracij oslonilnih točk (Rohlf & Slice 1990, Rohlf 2007), pripravili za ugotavljanje morfološke variabilnosti. Vrednosti Prokrustove analize, delne varpe, smo uporabili kot vhodne podatke o obliki lobanj v diskriminantni analizi (Janžekovič s sod. 2007). Vizualizacijo primerjav morfoloških značilnosti med skupinami smo izvedli z regresijsko analizo med posamezno diskriminantno funkcijo in delnimi varpi s pomočjo računalniškega programa tpsReg (Rohlf 2000).

3. REZULTATI IN DISKUSIJA

Morfološke razlike v sekundarnem spolnem dimorfizmu niso bile statistično značilne, zato smo v nadaljnjih analizah spola obravnavali združeno v enotnem vzorcu.

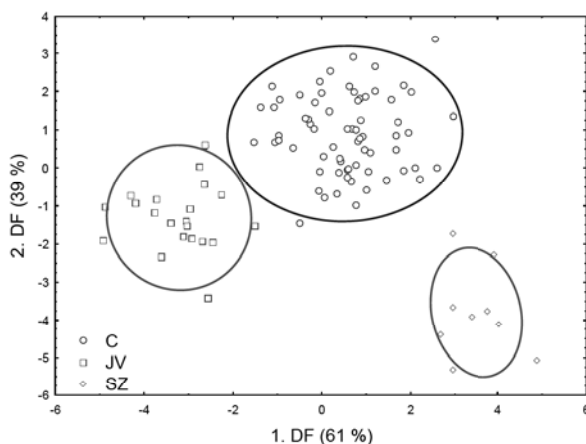
Iz 23 oslonilnih točk ventralne strani lobanje smo izračunali 44 delnih varpov in uniformni komponenti za x in y os, te vrednosti povzemajo variabilnost o obliki lobanje. Vrednosti varpov in uniformne komponente smo kot vhodne podatke vpeljali v diskriminantno analizo, v kateri so bili trije filogeografski vzorci dinarske voluharice (severozahodni, centralni in jugovzhodni) a priori določene skupine. Diskriminantna analiza je izkazala signifikantne morfološke razlike med vsemi tremi skupinami Wilks' Lambda: 0,066 $F_{84,110}=3,79$ $p < 0,001$. V klasifikacijski analizi je bilo iz dejanskih v napovedane skupine pravilno uvrščenih 98,9% osebkov (tabela 1), nepravilna klasifikacija je bila samo v enem primeru, kjer je bila žival iz centralne skupine uvrščena v jugovzhodno skupino. Morfološki odnosi med filogeografskimi skupinami so predstavljene na sliki 2. Severozahodna, evolucijsko najstarejša skupina katere starost je

ocenjena na 1.000.000 let, je pričakovano jasno ločena od preostalih dveh in to na osnovi 1. in 2. diskriminantne funkcije. Med centralno in jugovzhodno skupino, med katerima je cca. 300.000 let ločenega razvoja, je morfometrično razlikovanje sicer dobro vzpostavljeno, prehod od ene do druge skupine pa je praktično zvezen.

Morfološke razlike med severozahodno in centralno skupino se nanašajo predvsem na dolžino trdega neba (točki 1 in 3) ter na relativno velikost bobničnega mehurja, opisujejo ga točke 13, 14, 15, 18 in 19 (slika 1).

Tabela 1: Rezultati klasifikacije osebkov dinarske voluharice iz dejanskih v napovedane skupine.

	<i>Delež pravih klasifikacij</i>	<i>C</i>	<i>J</i> <i>V</i>	<i>SZ</i>
Centralna skupina - C	99	68	1	0
Jugo vzhodna skupina - JV	100	0	21	0
Severo zahodna skupina - SZ	100	0	0	9
Skupaj	99	68	22	9



Slika 2: Razmerja med morfometričnimi prostori pri projekciji osebkov dinarske voluharice na 1. in 2. diskriminantno funkcijo.

LITERATURA

1. Bolkay, S., 1924. Catalogue of the mammals occurring in the Bosnia-Hercegovina and preserved in the Land-Museum in Sarajevo. Biol. Hung. 1, 1–7.
2. Caumul, R., Polly, P.D., 2005. Phylogenetic and environmental components of morphological variation: skull, mandible, and molar shape in marmots (*Marmota*, Rodentia). *Evolution* 59, 2460–2472.

3. Đulić, B., Mirić, Đ., 1967. *Catalogus Faunae Jugoslaviae. IV/4 Mammalia.* Academia Scientiarum et Artium Slovenica, Ljubljana.
4. Janžekovič, F., Klenovšek, T., Kryštufek, B., Uporaba metod geometrijske morfometrije in multivariatne statistike v bioloških raziskavah filogenije in ontogenije. V: Potočnik, B., 2007 ROSUS, Računalniška obdelava slik in njena uporaba v Sloveniji. Maribor; str. 135-144.
5. Krystufek, B., Buzan, E.V., Hutchinson, W.F., Hänfling, B., 2007. Phylogeography of the rare Balkan endemic Martino's vole, *Dinaromys bogdanovi*, reveals strong differentiation within the western Balkan Peninsula. *Mol. Ecol.* 16, 1221–1232.
6. Martino, K., 1948. Prilog poznavanju recentnih voluharica roda *Dolomys*. *Godišnjak Biološkog Instituta, Sarajevo* 1, 71–78.
7. Petrov, B., Todorović, M., 1982. *Dinaromys bogdanovi* (V. et E. Martino, 1922) – Bergmaus. In: Niethammer, J., Krapp, F. (Eds.), *Handbuch der Säugetiere Europas*, Bd. 2/I. Akademische Verlagsgesellschaft, Wiesbaden, pp. 193–208.
8. Rohlf, F.J., 2000. *tpsRcgr – Thin Plate Spline Regression*, version 1.22. State University of New York at Stony Brook, New York.
9. Rohlf, J., 2006. *tpsDig2*, Version 2.10.. SUNY at Stony Brook.
10. Rohlf, J. (2007) *Relative warps*, Version 1.45. *Ecology & Evolution*. SUNY at Stony Brook.
11. Todorović, M., 1956. Variability of the endemic genus *Dolomys* Nehring (*Microtinae*, *Rodentia*). *Arhiv biol. nauka* 8, 93-109.

SLEDENJE OBLIK FOSFOLIPIDNIH VEZIKLOV ZAJETIH S FAZNO KONTRASTNO MIKROSKOPIJO

Peter Usenik, Tomaž Vrtovec, Franjo Pernuš, Boštjan Likar

Laboratorij za slikovne tehnologije
Fakulteta za elektrotehniko, Univerza v Ljubljani
E-pošta: peter.usenik@fe.uni-lj.si

POVZETEK: Predlagamo algoritem, ki samodejno sledi konturam fosfolipidnih veziklov na slikah zajetih s fazno kontrastno mikroskopijo, in ki ga lahko uporabimo za določanje moči, amplitudne in frekvenčne porazeditve konturnih premikov, ki so posledica termičnih fluktuacij. Algoritem smo testirali na veziklih z različnimi membranskimi strukturami, ki so bili izpostavljeni različnim temperaturnim okoljem. Rezultati kažejo, da je predlagani algoritem hiter, robusten, zanesljiv in omogoča direktno spektralno analizo termičnih fluktuacij, preko katere lahko določimo ostale lastnosti veziklov, kot sta upogibna elastičnost in spontana ukrivljenost.

1. UVOD

Proučevanje biomehanskih lastnosti fosfolipidnih vezikov omogoča boljše razumevanje bioloških membran in s tem biologije celic. S tehnološkim napredkom je postala video mikroskopija ena od standardnih tehnik na področju proučevanja oblik celic. Z merjenjem in sledenjem oblik celic dobimo kvantitativne podatke, ki nudijo odgovore na številna znanstvena vprašanja.

Fosfolipidne membrane imajo poseben mikroskopski značaj, ki mu pravimo termične fluktuacije [1]. Fluktuacije lahko opazujemo in jih analiziramo na dvo-dimenzionalnih (2-D) mikroskopskih posnetkih. Membrano tako predstavlja 2-D kontura, iz katere lahko določimo nekatere biomehanske lastnosti membran, kot sta membranska upogibna elastičnost in spontana ukrivljenosti membrane [2, 3, 4]. Za proučevanje biomehanskih lastnosti membran največkrat uporabljamo velikanske vezikle (GUV), ki so dovolj veliki ($\sim 1\text{--}100\ \mu\text{m}$), da jih lahko direktno opazujemo z optično vidno mikroskopijo. Ker pa je debelina membrane veziklov ($4\text{--}5\ \text{nm}$) precej pod resolucijo optičnih mikroskopov ($200\ \text{nm}$), je potrebno uporabiti primerno mikroskopsko tehniko, ki poveča kontrast, kot so fazni kontrast, diferencialno interferenčni kontrast ali fluorescenca.

V tem članku predlagamo algoritem, ki avtomatično sledi konturam veziklov na posnetkih zajetih z metodo fazno kontrastne mikroskopije. Kljub temu, da je algoritem osnovan na relativno enostavnih in intuitivnih metodah, predstavlja učinkovit pristop opisa konture, ki je primeren za nadaljno analizo kontur.

2. MATERIALI IN METODA

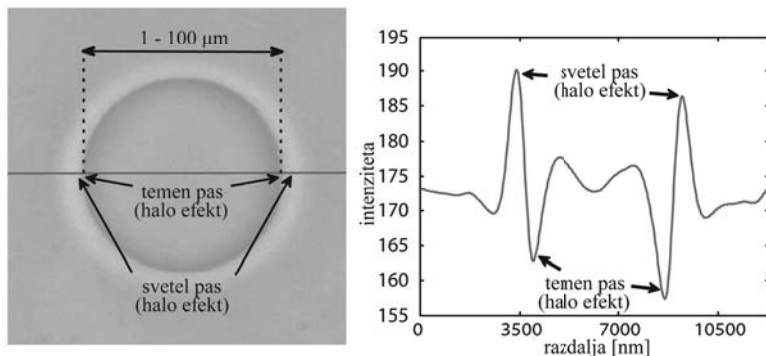
Vezikle smo pripravili pri sobni temperaturi (23 °C) po modificirani metodi elektroformacije [5]. Uporabili smo sintetični lipid palmitoil-oleil-fosfatidilholin (POPC) in holesterol (CHOL) (oba Avanti Polar Lipids, Alabaster, AL, USA). V organskem topilu kloroforma in metanola (2:1) smo raztopili različna razmerja POPC in holesterola (čisti POPC, POPC:CHOL = 4:1 in POPC:CHOL = 1:4). Zmes smo v stekleni bučki intenzivno premešali nato pa 20 μ l zmesi nanseli na platinaste elektrode. Nanešeno zmes smo sušili v nizkem vakumu približno 30 min, nato pa smo jih postavili v elektroformacijsko posodo, v kateri smo imeli 3 ml 0,2 M saharozne raztopine. Potem smo vzpostavili izmenično električno polje velikosti 5 V/mm pri frekvenci 10 Hz za približno dve uri. Na koncu smo dodali še 4 ml 0,2 M raztopine glukoze in tako dobili raztopino, ki je vključevala velikanske vezikle (GUV).

Pripravljene vezikle smo nanесли na objektivno steklo, jih prekrili z krovnim stekelcem in zatesnili s silikonsko mastjo, da smo preprečili morebitno izhlapevanje vode. Preparat smo vstavili v termično kletko, ki je del temperaturnega krmilnika (HCC-100A, Dagan Corporation, Minneapolis, USA) in vse skupaj postavili pod invertirani fazno kontrastni mikroskop (Nikon Eclipse TE2000-S, Tokyo, Japan) 100 \times povečave. Temperaturo termične kletke smo zaznavali z dvema termočlenoma na preparatu in na samem ohišju termične kletke. S krmiljenjem temperature preparata smo ustvarili temperaturna okolja pri 5, 20, 35 in 50 °C (\pm 0.1 °C). Mikroskopske posnetke (velikosti 640 \times 480 slikovnih elementov, kjer je velikost slikovnih elementov 35 nm) smo zajemali pri hitrosti 30 Hz z 8-bitno sivinsko digitalno kamero (BCi4-4400, C-cam Technologies, Leuven, Belgium).

2.1 Sledenje kontur vezikla

Slike zajete s fazno kontrastno mikroskopijo imajo posebno značilnost, ki se kaže v posvetljenih in otemnjenih sivinskih intenzitetah v okolici objekta. Temu fenomenu pravimo "halo efekt" in je posebej izrazit pri preparatih, ki imajo v strukturah velike razlike v lomnih količinah. V primeru fosfolipinega vezikla, se halo efekt kaže kot osvetljen pas na zunanji in kot otemnjen pas na notranji strani membrane (Slika 1). Dejanski položaj membrane pa predstavlja prehod med tema dvema pasovama. Predstavljeni algoritem bazira na samodejnem iskanju prehoda med pasovoma, ki definira konturo vezikla za vsako od slik v zajeti video sekvenci.

V prvi sliki izbrane video sekvence ročno določimo točko znotraj vezikla, ki predstavlja začetno izhodišče polarnega koordinatnega sistema (r, φ). Nato generiramo K radialnih žarkov, ki potekajo od izbranega koordinatnega izhodišča proti membrani vezikla. Žarki so razporejeni tako, da so med seboj razmaknjeni za enak kot $\Delta\varphi = 2\pi/K$ (Slika 2). Vsakemu žarku pripada profil sivinskih vrednosti $f_k(r, \varphi_k)$; $k = 1, 2, \dots, K$ in ima potek značilen za halo efekt (Slika 2), kar se sprva kaže v zmanjšanju sivinskih vrednostih (otemnjen pas) in kasneje hitrem zvečanju sivinskih vrednosti (osvetljen pas).

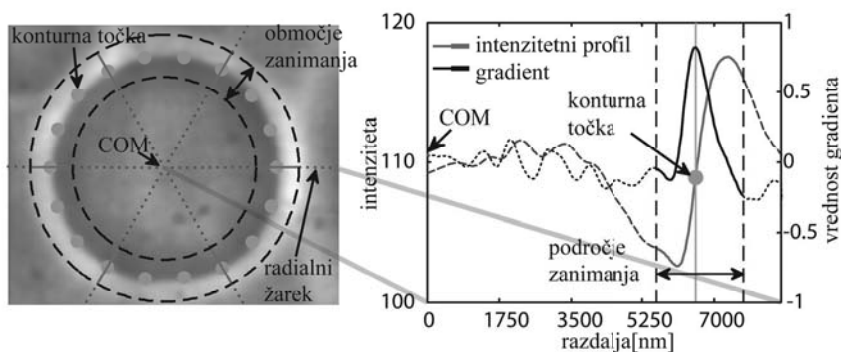


Slika 1: Slika fosfolipidnega vezikla zajetega s fazno kontrastno mikroskopijo (levo). Profil intenzitnih vrednosti (desno).

Ker vsak k -ti žarek prebada membrano natanko enkrat, lahko določimo položaj konturne točke $C_k(r_k, \varphi_k)$ s položajem maksimalnega gradienta izbranega profila sivinskih vrednosti $f_k(r, \varphi_k)$ (Slika 2):

$$C_k(r_k, \varphi_k) = \arg \max_{r, \varphi_k} \left(\frac{df(r, \varphi_k)}{dr} \right); \quad k = 1, 2, \dots, K \quad (1)$$

Konturne točke lahko določimo avtomatično za vsak žarek in jih shranimo kot serijo točk $C_k(r_k, \varphi_k)$; $k = 1, 2, \dots, K$, ki definirajo prostorski obris ali konturo membrane vezikla.



Slika 2: Določitev konturnih točk na podlagi maksimalne vrednosti gradienta iz pripadajočega intenzitetnega profila..

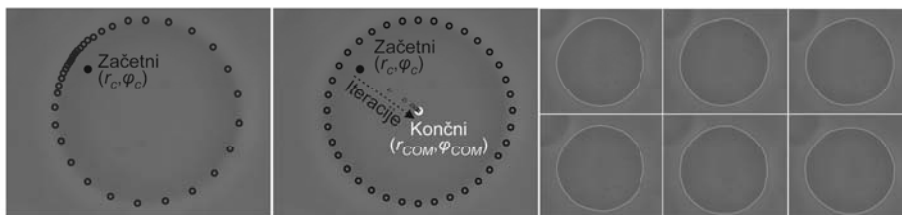
Točko, ki predstavlja izhodišče polarnega koordinatnega sistema (r, φ) , smo določili ročno, zato povsem verjetno ne leži v težišču (COM) vezikla in zato konturne točke niso med seboj enakomerno razmaknjene (slika). Da bi membrano opisali z enakomerno razmaknjenimi konturnimi točkami, se mora koordinatno izhodišče (r, φ) nahajati ravno

v težišču (COM) vezikla. Iz sprva dobljenih konturnih točk izračunamo njihovo težišče (r_c, φ_c) :

$$\begin{aligned} r_c \cos \varphi_c &= \frac{1}{K} \sum_{k=0}^K r_k \cos \varphi_k \\ r_c \sin \varphi_c &= \frac{1}{K} \sum_{k=0}^K r_k \sin \varphi_k \end{aligned} \quad (2)$$

Potrebno je poudariti, da dobljeno težišče konturnih točk (r_c, φ_c) ne predstavlja težišča vezikla (r_{com}, φ_{com}) , predstavlja pa točko, ki leži bližje težišču (r_{com}, φ_{com}) kot predhodno določeno izhodišče koordinatnega sistema. Da postavimo koordinatno izhodišče v zadovoljivo okolico težišča vezikla (r_{com}, φ_{com}) , je potrebno opisani postopek ponavljati, dokler koordinate (r_c, φ_c) ne dosežejo izbranega konvergenčnega območja. V vsakem koraku takšne iteracije generiramo nove radialne žarke in določimo nove konturne točke. Končne konturne točke, ki predstavljajo opis konture vezikla, določimo iz dobljenih končnih kordinat (r_c, φ_c) , ki sovpadajo z vrednostjo težišča vezikla (r_{com}, φ_{com}) . Dejstvo, da so končne konturne točke določene iz težišča vezikla omogoča direktno aplikacijo Fourierjeve transformacije, ki jo uporabimo v nadaljni analizi konture.

Opisani postopek izvedemo za vsako sliko v izbrani video sekvenci. Začetno koordinatno izhodišče (r, φ) izberemo ročno le v prvi sliki, za vse ostale slike v sekvenci pa je začetno koordinatno izhodišče določeno z položajem težišča vezikla v predhodnji sliki (r_{com}, φ_{com}) . Z avtomatskim sledenjem konture vezikla skozi celotno izbrano video sekvenco dobimo serijo konturnih točk v času. Tako lahko časovno analiziramo obliko vezikla.



Slika 3: Iterativni postopek iskanja (r_{com}, φ_{com}) (levo, sredina). Slike uspešno zaznanih kontur na primerih znotraj ene sekvence (desno).

2.2 Analiza konture vezikla

Zaradi termičnih fluktuacij se položaj vsake od konturnih točk v času spreminja. Za vsako konturno točko C_k ; $k = 1, 2, \dots, K$, lahko definiramo funkcijo $c_k(t)$; $k = 1, 2, \dots, K$, ki opisuje položaj točke v času. Moč P_k lahko izračunamo za vsako od funkcij kot $P_k = c_k^2(t)$ in močnostni spekter $S_k(\omega)$ za vsako od funkcij z uporabo hitre Fourierjeve transformacije (FFT). Povprečni moči in povprečnemu močnostnemu spektru vseh

funkcij $c_k(t)$; $k = 1, 2, \dots, K$, lahko rečemo moč membrane in močnostni spekter membrane in ju uporabimo za karakterizacijo termičnih fluktuacij vezikla (Slika 3).

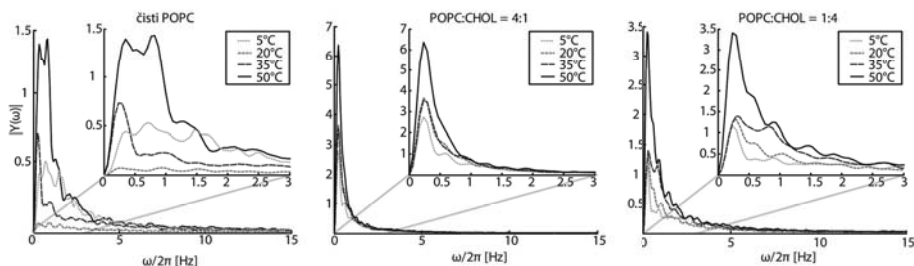
2.3 Podrobnosti izvedbe

Predlagani algoritem je bil implementiran v programskem jeziku MATLAB (The Mathworks, Natick, MA, USA). Vsako video sekvenco smo predhodno filtrirali z Gaussovimi filtrom ($\sigma = 175$ nm). Število K radialnih žarkov je poljubno nastavljivo, za analizo pa smo uporabili $K = 360 \Rightarrow \Delta\varphi = 1^\circ$. Profile sivinskih vrednosti v domeni žarkov, smo določili z bilinearno interpolacijo. Iskanje konturnih točk vezikla smo iterativno ponavljali dokler se koordinate (r_c, φ_c) niso približale težišču za manj kot en slikovni element (35nm), za kar so bile v povprečju potrebne štiri iteracije ($N = 4$). Da smo povečali hitrost algoritma, smo avtomatsko definirali območje velikosti dolžin žarkov v obliki kolobarja z prilagodljivo debelino tako, da se je membrana zagotovo nahajala znotraj tega območja.

3. REZULTATI IN DISKUSIJA

Predstavljeni algoritem smo preiskovali na velikanskih veziklih treh različnih membranskih struktur, ki so bili izpostavljeni štirim različnim temperaturnim okoljem (Poglavje 2). Tako smo zajeli 12 mikroskopskih video sekvenc. Dobljeni rezultati kažejo, da predlagani algoritem uspešno in samodejno sledi konturam veziklov, da je robusten z ozirom na šum, da je časovno nezahteven in da je učinkovit za analizo fluktuacij. Dobljen kvantitativni podpis fluktuacij vezikla omogoča direktno spektralno analizo, preko katere lahko določimo biomehanske lastnosti vezikla, kot sta upogibna elastičnost in spontana ukrivljenost [6].

Za vsako od sekvenc smo določili povprečno moč membrane (Tabela 1) in povprečen močnostni spekter membrane (Slika 4) iz funkcij $c_k(t)$; $k = 1, 2, \dots, K$, ki predstavljajo položaj vsake od konturnih točk v času. Rezultati kažejo, da moč membrane, ne glede na membransko strukturo veziklov, relativno počasi in konstantno narašča v temperaturnem območju do 35°C , medtem ko lahko opazimo izrazito nelinearno povečanje moči membrane v temperaturnem območju nad 35°C . Potek močnostnega spektra membrane opisuje frekvenčno razporeditev fluktuacij membrane. Močnostni spektri na sliki 4, kažejo na to, da se nahaja pri vseh membranskih strukturah, večina fluktuacij membrane v pasu nizkih frekvenc (0–3 Hz), amplituda spektrov pa se z višjo temperaturo veča. Če upoštevamo Shannonov teorem, mora biti frekvenca vzorčenja vsaj dvakrat večja kot najvišja frekvenc fluktuacij. Zmožnost vzorčenja naše kamere (30 Hz) omogoča zaznavo fluktuacij do 15Hz. Kljub temu, da je večina frekvenc fluktuacij v omenjenem nizkem pasu, je možno, da obstajajo tudi fluktuacije, katerih frekvence so večje kot 15 Hz. Membranska struktura veziklov se je razlikovala po deležu holesterola v POPC. Holesterol je razširjena membranska molekula, ki vpliva na viskoznost in ostale lastnosti fosfolipidnih membran [1]. Kljub temu pa iz naših meritev, ne moremo potrditi vloge holesterola.



Slika 4: Amplituda močnostnega spektra membrane za različne membranske strukture pri različnih temperaturnih okoljih.

Tabela 1: Moč membrane (\pm standardna deviacija) za različne membranske strukture pri različnih temperaturnih okoljih.

T [°C]	<i>čisti POPC</i>	<i>POPC:CHOL</i> <i>razmerje 4:1</i>	<i>POPC:CHOL</i> <i>razmerje 1:4</i>
5	25.43 \pm 2.91	32.89 \pm 3.25	29.29 \pm 7.33
20	5.34 \pm 0.74	42.82 \pm 3.43	28.88 \pm 7.69
35	17.06 \pm 2.16	43.72 \pm 4.64	41.89 \pm 7.56
50	40.24 \pm 4.43	63.15 \pm 8.91	61.32 \pm 6.76

Algoritem lahko uporabljamo za sledenje kontur veziklov poljubnih ovalnih oblik, sicer pa je najučinkovitejši pri skoraj okroglih veziklih, ki tudi sicer predstavljajo največji znanstveni interes, saj predstavljajo ravnovesno stanje zaprtih lipidnih struktur [1]. Kljub temu pa ima algoritem nekaj omejitev. Če opazovani vzorec ni popolnoma v termičnem ravnovesju, se lahko vezikli premikajo v fokusni ravnini ali vertikalno glede na fokusno ravnino. Premiki v fokusni ravnini povzročijo, da se vezikel ne nahaja več v vidnem območju kamere. Tako algoritem ne more določiti celotne konture vezikla, zato del konturnih točk ne predstavlja dejanske membrane. Premiki pravokotno glede na fokusno ravnino pa povzročijo, da slika vezikla ni več ostra. Tako algoritem zazna določene konturne točke, ki pa ne predstavljajo dejanske membrane. Tudi vezikli, ki se nahajajo v neposredni bližini opazovanega vezikla lahko prav tako onemogočijo pravilno sledenje, saj s prekrivanjem povzročijo, da se del membrane opazovanega vezikla premakne izven fokusne ravnine. Ob dejstvu da so nekatere geometrijske lastnosti veziklov v stabilnem temperaturnem okolju konstantne, lahko zanesljivost algoritma kvalitativno ovrednotimo s primerjanjem obsega in površine veziklov med seboj. Velike spremembe geometrijskih lastnosti od povprečnih vrednosti, kažejo kdaj je bilo sledenje konture napačno. Takšne spremembe lahko zaznavamo samodejno in slike z napačno določeno konturo izločimo pred izvedbo same analize konture ter se s tem izognemo prej omenjenim omejitvam.

4. ZAKLJUČEK

V tem prispevku predlagamo algoritem za avtomatsko sledenje kontur fosfolipidnih veziklov, ki je hiter, robusten in zanesljiv. Oblike kontur določajo podpise oblik vezikla, ki so primerni za analizo termičnih fluktuacij veziklov, za določanje moči in frekvenčne porazdeljenosti fluktuacij in nadalje omogočajo določanje biomehanskih lastnosti vezikla, kot sta ukrivljenost in spontana ukrivljenost.

LITERATURA

1. H.G. Döbereiner, E. Evans, M. Kraus, U. Seifert, and M. Wortis (1997), Mapping vesicle shapes into the phase diagram: A comparison of experiment and theory, *Physical Review E*, vol. 55, str. 4458-4474.
2. J. Pécéréaux, H. G. Döbereiner, J. Prost, J. F. Joanny, and P. Bassereau (2004), Refined contour analysis of giant unilamellar vesicles, *The European Physical Journal E - Soft Matter and Biological Physics*, vol. 13, str. 277-290.
3. M.B. Schneider, J.T. Jenkins, and W.W. Webb (1984), Thermal fluctuations of large quasi-spherical bimolecular phospholipid vesicles, *Journal de Physique*, vol. 45, str. 14571-1472.
4. V. Kralj-Iglič, G. Gomišček, J. Majhenc, V. Arrigler, S. Svetina (2001), Myelin-like protrusions of giant phospholipid vesicles prepared by electroformation, *Colloids and Surfaces A*, vol. 181, str. 315-318.
5. H. Engelhardt, H.P. Duwe and E. Sackmann (1985), Bilayer bending elasticity measured by Fourier analysis of thermally excited surface undulations of flaccid vesicles, *Journal de Physique Letters.*, vol. 46, str. L-395-400.

KVANTIZACIJA IN REKONSTRUKCIJA PERIFOLIKULARNEGA OŽILJA V ULTRAZVOČNIH VOLUMNIH

Boris Cigale, Smiljan Šinjur, Damjan Zazula

**Laboratorij za sistemsko programsko opremo
Fakulteta za elektrotehniko, računalništvo in informatiko**

E-pošta: boris.cigale@uni-mb.si

URL: <http://storm.uni-mb.si>

POVZETEK: *Žilna plast okoli jajčnega mešička igra pomembno vlogo pri ustvarjanju pogojev za zanositev. Naš prototipni postopek za ocenitev perifolikularne vaskularizacije temelji na uporabi ultrazvočnih posnetkov. Pri tem najprej poiščemo položaj dominantnega mešička s pomočjo valčne transformacije. Nato se osredotočimo na pas do oddaljenosti 5 mm okoli mešička, saj se v tem pasu nahajajo žile, pomembne za razvoj mešička. Položaj žil na posnetkih razberemo iz ultrazvočnih dopplerjevih posnetkov, opišemo pa jih z dvema histogramoma. Celotno ožilje rekonstruiramo z novim postopkom, ki temelji na minimalnih vpetih drevesih.*

1. UVOD

Žilna plast okoli jajčnih mešičkov igra pomembno vlogo pri ustvarjanju pogojev za zanositev. Pri postopkih oploditve z biomedicinsko pomočjo zdravniki izsrkajo vsebino mešička, pri tem pa pride do poškodbe žil, ki so ob mešičku (perifolikularna vaskularizacija). Kakšne so poškodbe ožilja med punkcijo mešička, ni popolnoma jasno, kljub temu pa vse pacientke dobijo zdravila, ki pospešijo delovanje rumenega telesa.

Cilj naše raziskave je razviti postopek, ki bi omogočil ocenitev perifolikularne vaskularizacije. S tem bi dobili možnost objektivne presoje, ali je uporaba hormonske terapije potrebna in smiselna.

Razviti postopek, ki je trenutno v fazi prototipa, obdeluje ultrazvočne volumne jajčnikov. Pri tem uporabi tako navadne ultrazvočne posnetke kakor tudi močnostne dopplerjeve posnetke. Najprej na navadnem ultrazvočnem posnetku poiščemo položaj mešička. Pri tem uporabimo valčno transformacijo, kakor je opisano v drugem poglavju. Ko najdemo mešiček, se osredotočimo na pas v oddaljenosti 5 mm okoli njega. V tem pasu se nahajajo žile, ki so povezane z mešičkom in njegovim razvojem. Ožilje opišemo s pomočjo dveh histogramov, kakor prikazujemo v tretjem poglavju. Ker ima močnostni doppler nizko ločljivost, smo manjkajočo informacijo poskušali nadomestiti z rekonstrukcijskim postopkom za prikaz ožilja. Ta temelji na minimalnih vpetih drevesih, kot je zapisano v četrtem poglavju. Prispevek sklenemo s kratko razpravo v 5. poglavju.

2. DETEKCIJA MEŠIČKOV

Jajčni mešički so v ultrazvočnih volumnih vidni kot sferična homogena območja, katerih povprečna sivina je temnejša od okolice. Njihova rast se ponavadi zaustavi, ko dosežejo premer od 8 do 10 mm, le dominantni mešički lahko dosežejo premer od 17 do 25 mm. V tej raziskavi smo se omejili le na iskanje dominantnih mešičkov. Premer posameznega voksla v volumnih, ki smo jih analizirali, znaša okoli 0,2 mm, torej imajo mešički premer okoli 85 voksloraz.

Za detekcijo mešičkov smo uporabili postopek, ki temelji na zvezni valčni transformaciji [1]. Postopek je razdeljen na dva koraka. V prvem poiščemo središče dominantnega mešička, v drugem pa določimo njegovo obliko.

2.1 Iskanje središča mešička

Ker imajo mešički sferično obliko, so še posebej primerni za detekcijo s pomočjo 3D valčka, imenovanega mehniški klobuk, saj je izotropen.

Če modeliramo mešiček kot sfero C s sivino 1 in okoliško tkivo s sivino 0, smo v [1] pokazali, da imamo v valčni transformiranki lokalni ekstrem v središču mešička. Ali gre za lokalni minimum ali maksimum, je odvisno od merila in polmera sfere C . V [2] smo pokazali, da so najbolj primerna merila a za iskanje središča mešička tista, kjer velja

$$\frac{C-0.5}{3.5} < a < \frac{C}{2}. \quad (1)$$

Pri manjših merilih se v okolici središča pojavi mnogo ekstremov, pri višjih merilih pa na rezultat preveč vpliva okolica mešička.

Naš algoritem temelji na sledenju lokalnim minimumom iz merila v merilo. Postopek začnemo v najvišjem merilu a_b , v katerem poskušamo zaznati največje mešičke, končamo pa v merilu a_e , kjer poskušamo zaznati manjše mešičke. Koordinate vseh lokalnih minimumov pri merilu a_t , $t=e, \dots, b$, shranimo v množico D_t . Najprej za vsak minimum preverimo smiselnost nadaljnje obdelave. Tako v originalnem posnetku preverimo sivino voksla, kjer leži minimum valčne transformiranke. Če sivina voksla presega poprej določen prag T , minimum zavržemo. Nato za vsako koordinato k_i^t v D_t poiščemo najbližji minimum s koordinato k_j^{t+1} , ki je hranjen v množici D_{t+1} . Koordinata k_j^{t+1} in merilo $t+1$ sta dodana v množico lokalnih minimumov, če je evklidska razdalja $|k_i^t - k_j^{t+1}| < H$, kjer H predstavlja največji dovoljeni odmik (ponavadi 5). Minimum, ki mu lahko sledimo skozi tri merila, je dober kandidat za središče mešička. Ker iščemo le dominantni mešiček, lahko varno domnevamo, da je to tisti, ki ga prvega uspemo izslediti skozi tri največja merila.

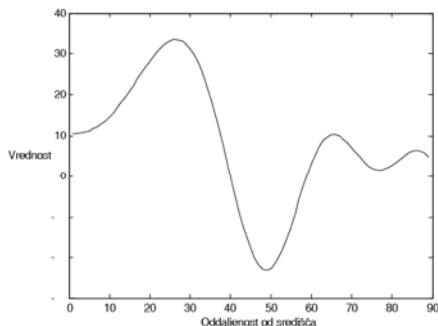
2.2 Določanje oblike mešička

Potem ko smo detektirali središče mešička, želimo določiti še njegovo obliko. V idealnem primeru bi lahko le orisali sfero okoli zaznanega središča, vendar pa realni dejanski mešički nimajo pravilnih oblik.

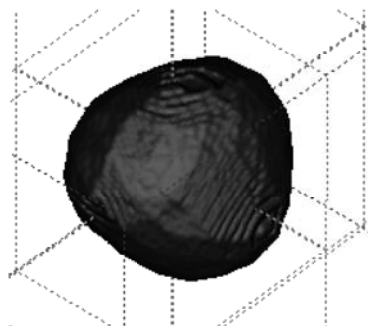
Iz lastnosti valčne transformacije vemo, da se na različnih odmikih od središča analiziranega mešička pojavijo različne karakteristične točke. Če pogledamo sliko 1(a), vidimo, da se na poti iz središča proti robu površini mešička najprej pojavi lokalni maksimum. V [1] smo pokazali, da je mesto maksimuma, označeno z b , povezano z oddaljenostjo r do površine in merilom a , pri katerem je bila dobljena valčna transformiranka. Oddaljenost površine mešička od njegovega središča tako lahko izračunamo z enačbo

$$r = b + a. \quad (2)$$

To dejstvo izkoristimo pri metodi pošiljanja žarkov, ki pošlje snope žarkov iz središča mešička v različnih smereh. Površinske točke na mešičku se izračunajo glede na merilo a , v katerem smo odkrili središče mešička. Za določanje mej mešička smo uporabili 132 snopov žarkov, ki so enakomerno porazdeljeni po sferi. Dobljene površinske točke organiziramo v trikotniško mrežo. Ker se zgodi, da nekatere razpoznane točke zelo odstopajo od okolice, trikotniško mrežo še zgladimo in se s tem znebimo napak pri rekonstruirani obliki mešička. Zgrajeno trikotniško mrežo lahko preprosto spremenimo tudi v volumensko obliko.



(a)



(b)

Slika 1: Spreminjanje vrednosti v valčni transformiranki dejanskega ultrazvočnega posnetka jajčnega mešička glede na oddaljenost od središča (a), rekonstruirani mešiček (b).

3. KVANTIZACIJA PERIFOLIKULARNEGA OŽILJA

Kvantizacijo perifolikularnega ožilja lahko odčitamo iz dopplerjevih ultrazvočnih posnetkov, ki kažejo območja z zaznavnim pretokom krvi. Pri kvantizaciji upoštevamo le pas v oddaljenosti 5 mm okoli mešička. Dopplerjeve posnetke najprej binariziramo.

Uporabljen prag se nastavi ročno in je odvisen od nastavitve občutljivosti ultrazvočnega aparata.

V nadaljevanju ocenimo debelino posameznih žil, to pa potem uporabimo za preprosto ovrednotenje ožilja.

Označimo množico vseh vokslov, ki ležijo znotraj žil, s F , voksel i , ki pripada tej množici, pa s p_i . Okolico $3 \times 3 \times 3$ voksla p_i označimo z $N(p_i)$. $N_{26}(p_i) = N(p_i) - p_i$ označuje množico 26 vokslov, ki so sosedi p_i .

3.1 Ocenitev debeline žil

Avtorji v [3] definirajo lokalno debelino voksla p_i kot polmer največje sfere, ki vključuje voksel p_i in hkrati leži v celoti znotraj strukture. V našem primeru so te strukture seveda žile. Če bi take sfere iskali z grobo silo, bi bil postopek časovno zelo zahteven. Boljši je pristop z uporabo maksimalnih sfer, to je sfer, ki ležijo v celoti znotraj območja, opredeljenega z dopplerjevim posnetkom žile, in so medsebojno disjunktne. Z maksimalnimi sferami je možno v popolnosti rekonstruirati območje žile, zato je pri oceni debeline treba dodati le test vsebovanosti, ki pokaže, kateri maksimalni sferi pripada določen voksel.

3.1.1 Središča maksimalnih sfer

Vsakemu vokslu p_i v F lahko priredimo sfero s_i , ki ima središče v p_i in polmer r_i , ki predstavlja evklidsko razdaljo voksla p_i od površine sfere. S tem je zagotovljeno, da vse točke sfere ležijo znotraj območja, ki ga preiskujemo.

Maksimalne sfere predstavljajo podmnožico teh sfer. Maksimalne sfere niso v popolnosti vsebovane v nobeni drugi sferi. Pokaže se, da so središča teh sfer ponavadi umeščena v središča preiskovanih območij. Za izračun maksimalnih sfer smo uporabili postopek [4]. Ta algoritem nadomešča evklidsko razdaljo z razdaljo chamfer. Ta se lahko izračuna zelo hitro, saj je potreben le dvojni prehod skozi vse voksle.

Razdalja chamfer $d(p_i)$ izrazi oddaljenost voksla p_i roba žile s pomočjo sosednjih točk $d(p_j)$ iz $N_{26}(p_i)$ [5]:

$$d(p_i) = \begin{cases} \min\{d(p_j) + D(p_i, p_j)\} & p_i \in F \\ 0 & p_i \in \bar{F} \end{cases} \quad (3)$$

kjer je $D(p, q)$ utež, določena z evklidsko razdaljo do sosednjih točk:

$$D(p_i, p_j) = \begin{cases} 3, & \|p_i - p_j\|^2 = 1 \\ 4, & \|p_i - p_j\|^2 = 2 \\ 5, & \|p_i - p_j\|^2 = 3 \end{cases} \quad (4)$$

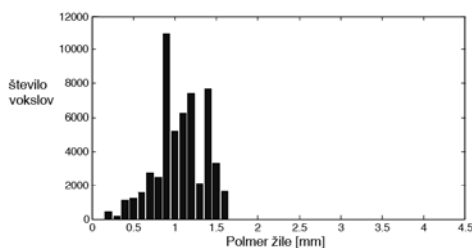
V [4] je pokazano, da središča c_i maksimalnih sfer izpolnjujejo naslednji pogoj:

$$d(c_i) \neq d(p_j) - D(c_i, p_j), \forall p_j \in N_{26}(c_i). \quad (5)$$

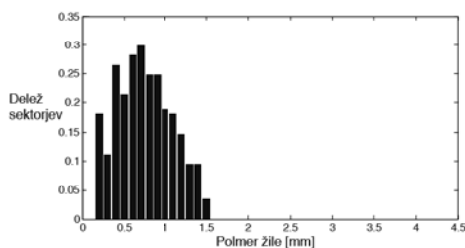
3.2 Izračun statistike

Statistično oceno perifolikularnega ožilja podamo s pomočjo dveh histogramov. Prvi histogram prikazuje število vokslov, ki pripadajo žili z določeno debelino. Tako dobimo splošni vtis o strukturi ožilja okoli dominantnega mešička (razmerje med debelimi in tankimi žilami). Primer histograma, odvisnega od ocenjene debeline žil, prikazuje slika 2(a).

Drugi histogram prikazuje prostorsko porazdelitev ožilja okoli mešička. Ta pokaže, ali je ožilje enakomerno porazdeljeno okoli mešička ali pa je koncentrirano le na določenem mestu. Histogram je upošteva delitev okolice mešička na 117 enako velikih sektorjev. Za vsako debelino žile prikazujemo, v kolikšnem deležu sektorjev je prisotna – slika 2(b).



(a)



(b)

Slika 2: Število vokslov, ki pripadajo žili z določeno debelino (a), in delež sektorjev, ki vsebujejo žile z določeno debelino (b).

4. REKONSTRUKCIJA OŽILJA

Zajem dopplerjevega ultrazvočnega posnetka traja 3-5 sekund, pri tem pa mora pacientka biti povsem pri miru in sproščena. V času zajemanja posameznega posnetka požene srce kri po telesu 3-5 krat. Tlak krvi v ožilju se zaradi utripanja spreminja, zato se deli ožilja na posnetkih, ki nastanejo med diastolami, slabše vidijo ali pa sploh niso zaznavni. Taka mesta v ultrazvočnih volumnih je treba rekonstruirati glede na informacije iz njihove okolice.

Rekonstrukcija ožilja je dodatek k statistični obdelavi zajetih podatkov, saj grafično prikaže porazdeljenost in razraslost ožilja okoli jajčnega mešička.

4.1 Postopek

Vhod v postopek rekonstrukcije je dopplerjev ultrazvočni volumen. Najpomembnejši korak postopka je tvorba minimalnega vpetega drevesa [6]. Z njim združimo dele ožilja, ki smo ga zaznali na posameznih rezinah volumna s pomočjo maksimalnih sfer, v celoto.

Rekonstrukcija ožilja poteka v več korakih: izračun razdalj chamfer, iskanje lokalnih maksimumov, izračun minimalnega vpetega drevesa, rasterizacija po Bresenhamu in morfološka operacija z razširitvijo.

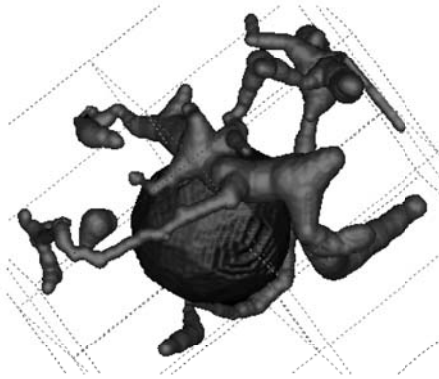
V prvem koraku izračunamo razdaljo chamfer za vsako točko ožilja, kot je opisano v sekciji 3.1.1. Najpomembnejše so točke, ki so najbolj oddaljene od površine mešička, saj predstavljajo središča žil. Točke, ki so znotraj razpoznanih območij najbolj oddaljene od robov območij, so lokalni maksimumi in hkrati vozliščne točke drevesa.

Število lokalnih maksimumov je v primerjavi s številom vokslov volumna razmeroma majhno. Zapišemo jih v redko matriko. Predpostavimo, da zaznano ožilje tvori samo eno drevo [7]. Vozliščne točke drevesa so že prej določeni lokalni maksimumi. Za tvorbo minimalnega vpetega drevesa smo uporabili Primov algoritem [8]. Kriterij za povezovanje vozlišč je evklidska razdalja med lokalnimi maksimumi v volumnu.

Rezultat Primovega algoritma je množica daljic, ki povezujejo vozlišča drevesa. Vsaka povezava nakazuje potek posamezne žile, celotno drevo v volumnu pa tvori ožilje. Izhod iz rekonstrukcije je nov volumen, zato je treba potek vsake daljice preslikati v volumenski posnetek. Daljico rasteriziramo, za kar uporabimo Bresenhamov algoritem [9,10]. Rezultat rasterizacije je množica 3D točk, ki jim informacijo o debelini priredimo iz prvega koraka, pri katerem smo tvorili matriko z razdaljami chamfer.

Seveda matrika definira samo debelino žile za vozliščne voksle. Zato vsakemu rasteriziranemu vokslu določimo debelino z linearno interpolacijo glede na debelino obeh vozlišč, ki sta z njim povezani. Vsak voksel nato v zadnjem koraku nadomestimo s sfero, katere polmer je enak izračunani debelini žile.

Slika 3 prikazuje rekonstruirano ožilje in pripadajoči jajčni mešiček.



Slika 3: Rekonstruirano ožilje in mešiček

4.2 Analiza

Opisani postopek rekonstruira perifolikularno ožilje iz ultrazvočnih dopplerjevih posnetkov jajčnikov. Rekonstruirano ožilje je seveda le približek dejanskega ožilja. Kljub temu lahko trdimo, da je večina vozlišč in povezav med njimi sovпада z dejanskimi žilami, saj so bili določeni z območji, ki na ultrazvočnih slikah predstavljajo prekrvavljene dele tkiva. Tudi primerjava rekonstruiranega ožilja s histogramom porazdelitve ožilja okoli mešička potrjuje to trditev.

Časovna zahtevnost opisanega algoritma je $O(n^2)$, kjer je n število lokalnih maksimumov, dobljenih z določanjem maksimalnih sfer na dopplerjevih posnetkov pretokov krvi okoli jajčnega mešička. Najzahtevnejši korak v postopku predstavlja izračun medsebojnih razdalj med najdenimi lokalnimi maksimumi (vozlišči). Za vsako vozlišče izračunamo razdaljo do vseh ostalih vozlišč. V prihodnje načrtujemo pohitritev tega izračuna, saj bi zadostovalo, če računanje razdalj za posamezno vozlišče omejimo le na njegovo lokalno okolico.

5. ZAKLJUČEK

V prispevku opisujemo prvi prototip za kvantizacijo perifolikularnega ožilja. Postopek deluje zadovoljivo na testnih posnetkih, ki jih imamo na voljo. Preliminarne rezultate so pregledali medicinski eksperti z Oddelka za ginekologijo in perinatologijo v Kliničnem centru Maribor. Seveda pa bodo sledila še izčrpnjša potrjevanja v medicinskem okolju in klinični praksi.

Delo v prihodnosti bo zato usmerjeno ravno na izdelavo prijaznega uporabniškega vmesnika, namenjenega medicinskemu osebju. Če se bo pojavila potreba s strani uporabnikov, bomo vpeljali tudi nove metrike.

LITERATURA

1. B. Cigale (2007), Odkrivanje in spremljanje rastistruktur z večmerilno obdelavo 3D ultrazvočnih posnetkov, *Doktorska disertacija*, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor.
2. B. Cigale, D. Zazula (2007), Segmentation of 3D ovarian ultrasound volumes using continuous wavelet transform, *Mediterranean Conference on Medical and Biological Engineering and Computing*, Ljubljana, str. 1017-1020.
3. T. Hildebrand, P. Rügsegger (1998), A new method for the model-independent assessment of thickness in three-dimensional images, *Journal of Microscopy*, Vol. 185, No. 1, str. 67-75.
4. C. Pudney (1998), Distance-Ordered Homotopic Thinning: A Skeletonization Algorithm for 3D Digital Images, *Computer Vision and Image Understanding*, Vol. 72, No. 3, str. 404-413.
5. M. W. Jones, J. A. Baerentzen, M. Sramek (2006), 3D Distance Fields: A Survey of Techniques and Applications, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 12, No. 4, str. 581-599.
6. E. Milková (2009), Constructing knowledge in graph theory and combinatorial optimization, *WSEAS Transactions on Mathematics*, Vol. 8, No. 8, str. 424-434.
7. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein (2001), *Introduction to Algorithms, Second Edition*, MIT Press and McGraw-Hill.
8. D. R. Karger, P. N. Klein, R. E. Tarjan (1995), A randomized linear-time algorithm to find minimum spanning trees, *Journal of the ACM*, Vol. 42, No. 2, str. 321-328.
9. J. E. Bresenham (1965), Algorithm for computer control of a digital plotter, *IBM Systems Journal*, Vol. 4, No.1, str. 25-30.
10. L. I. Dimitrov, M. Sramek (2004), Using 3D-Bresenham for Resampling Structured Grids, *Proceedings of the 3D Data Processing, Visualization, and Transmission*, 2nd International Symposium, str. 926-930.

DRUGO

REGISTRACIJA IN OBJEKTIVIZACIJA HITRIH GIBANJ V ŠPORTU

Milan Čoh, Mitja Bračič

Center za biomedicinske signale in slike
Fakulteta za šport, Laboratorij za biomehaniko, Univerza v Ljubljani
E-pošta: milan.coh@fsp.uni-lj.si
URL: <http://www.fsp.uni-lj.si/coh>

POVZETEK: *Sodobna športna diagnostika temelji na povezovanju različnih biomehaničnih metod s katerimi skušamo čim bolj objektivno oceniti gibanje in posamezne gibalne strukture. Za športna gibanja je značilna velika hitrost izvedbe in pravilna koordinacija. Med te gibalne strukture prištevamo skoke s katerimi merimo stopnjo eksplozivne moči. Eksplozivna moč pa je izjemno pomembna biomotorična sposobnost, ki generira rezultate v mnogih športnih panogah. S pomočjo metod kinematike, dinamike in elektromiografije smo ugotavljali parametre vertikalnih in globinskih skokov za oceno razvoja eksplozivne moči. Pridobljeni podatki nam služijo za kontrolo in modeliranje procesa treninga športnikov.*

1. UVOD

Hitra moč je ena od najpomembnejših biomotoričnih sposobnosti v napovedovanju rezultatov v različnih športnih panogah. Z močjo so bolj ali manj povezane tudi ostale biomotorične sposobnosti. Zato ni čudno, da so mnoge kineziološke raziskave fokusirane prav na področje moči z aspekta proučevanja njene strukture, metodologije treninga, uporabe novih metod in diagnostičnih postopkov. Zanesljivo je moč v sedanjem trenutku kineziološke znanosti med najtemeljiteje raziskanimi biomotoričnimi sposobnostmi. Kljub temu da obstajajo v tem prostoru še vedno številna odprta vprašanja. Moč lahko klasificiramo glede na različne kriterije. Z vidika akcijskega kriterija avtorji [1,3,5,7,8] ločijo maksimalno moč, hitro moč, eksplozivno moč in vzdržljivostno moč. Poleg tega kriterija obstaja še kriterij mišično – živčnega delovanja, kjer se moč manifestira v obliki izometrične kontrakcije, koncentrične, ekscentrične ali ekscentrično-koncentrične kontrakcije. Izometrična kontrakcija je prisotna v tistih situacijah, ko je sila mišice enaka zunanji sili, zato ne eksistira gibanje med prijemališčema mišice. V realnih motoričnih situacijah pa se najbolj pogosto pojavlja ekscentrično-koncentričen tip mišične kontrakcije, ki se manifestira v obliki odzivne moči. Odzivna moč je poseben primer eksplozivne moči v ekcentrično-koncentričnih pogojih in najbolj pogosto prisotna v cikličnih, acikličnih in kombiniranih gibalnih strukturah. Njena glavna značilnost je utilizacija elastične energije v ekcentrično-

koncentričnem ciklusu mišičnega naprežanja. Prispevek elastičnih značilnosti mišično-tetivnega kompleksa je odvisen od hitrosti prehoda.

Glede na strukturo gibanja so vertikalni in globinski skoki zelo podobni realnim motoričnim situacijam v športni praksi. Za diagnosticiranje eksplozivne moči spodnjih ekstremitet uporabljamo različne baterije testov, ki so lahko laboratorijskega ali situacijsko-terenskega tipa. Bosco (1992) je razvil klasični protokol monitoringa odzivne moči na osnovi vertikalnih skokov. Odrivno moč v koncentričnih pogojih živčno mišičnega delovanja merimo z vertikalnim skokom s pol-čepa (SQUAT JUMP). Odrivno moč pri katerem se aktivne mišice najprej raztegnejo (ekscentrična kontrakcija) nato pa skrčijo (koncentrična kontrakcija) merimo z globinskimi skoki (DROP JUMPS). V prispevku bomo prikazali kako je mogoče gibanja pri teh skokih objektivizirati z vidika kinematike, dinamike in elektromiografije

2. PREDSTAVITEV MERILNE PROCEDURE

V eksperimentalnih procedurah sta sodelovali dve vrhunski atletinji, skakalki troskoka (M.Š.: starost 28 let, višina 171, 5 cm, masa 66,3 kg, in M. G.: starost 23 let, višina 178.4 cm, masa 67.8 kg). Protokol meritev vertikalnih in globinskih skokov smo izvedli v Biomehanskem laboratoriju, Polyclinic for Physical Medicine and Rehabilitation Peharec v Puli. Za 3-D kinematično analizo vertikalnih skokov in globinskih skokov smo uporabili sistem 9 CCD kamer SMART-e 600 (BTS Bicengineering, Padova), ki so imele frekvenco 50 Hz in resolucijo slike 768 x 576 pixlov. Za obdelavo kinematičnih parametrov smo uporabili program BTS SMART Suite. Definirali smo kinematični model s sistemom 17 markerjev občutljivih na infra rdečo svetlobo (glava, ramena, podlakt, nadlakt, trup, boki, stegno, golen stopalo) – Slike 1,2,3. Veljavnost modela smo preverili z sekvenco hoje v sagitalni in frontalni ravnini. Na osnovi kinematične analize smo ugotovili naslednje parametre vertikalnih in globinskih skokov: višina odriva, čas letne faze, čas odzivne faze, čas ekscentrične faze, čas koncentrične faze, hitrost odriva, kot v skočnem, kolenskem in kolčnem sklepu.

KINEMATIČNA PROCEDURA

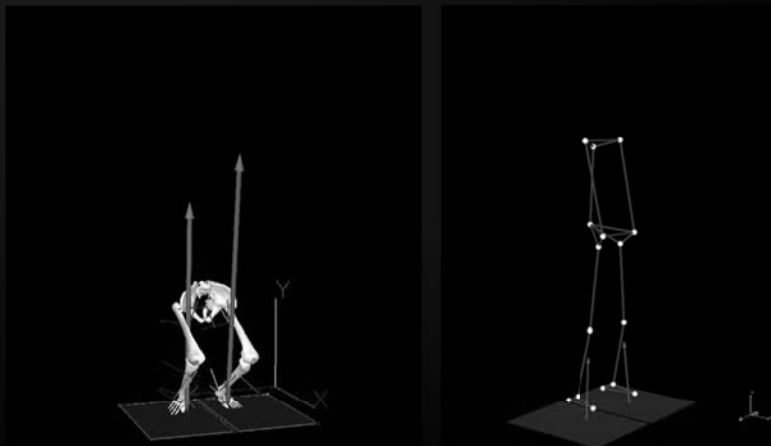
3-D KINEMATIČKA ANALIZA

(sistem 9 kamer CCD SMART – 250 Hz)



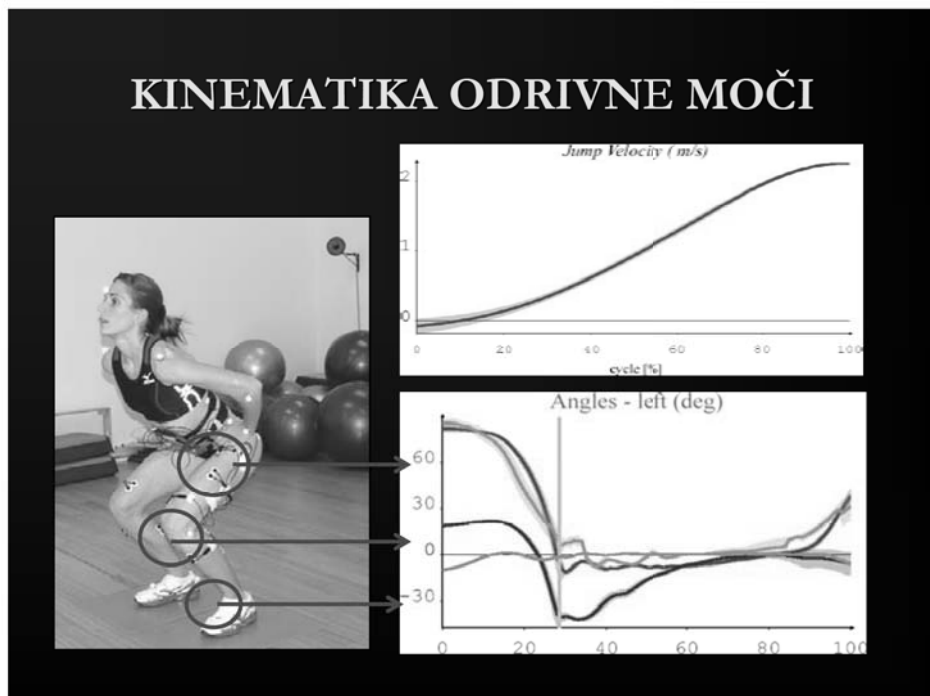
Slika 1. Sistem za 3 D kinematično analizo vertikalnih in globinskih skokov (BTS SMART – 600)

MONITORING EKSPLOZIVNE MOČI SPODNJIH EKSTREMITET



Slika 2: Kinematični model vertikalnega skoka definiran s 17 aktivnimi markerji

KINEMATIKA ODIVNE MOČI

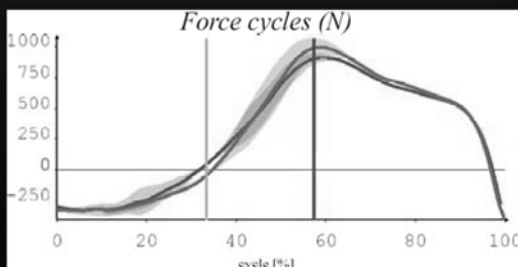


Slika 3: Registracija kinematičnih parametrov (vertikalna hitrost odziva, koti v skočnem, kolenskem in kolčnem sklepu) pri vertikalnem skoku.

Dinamične parametre vertikalnih in globinskih skokov smo ugotavljali s pomočjo dveh neodvisnih tenziometrijskih platform – force plate (Kistler Type 9286A). Frekvenca zajemanja podatkov je bila 1000 Hz. V analizi smo upoštevali naslednje dinamične parametre: maksimalno silo reakcije podlage, impulz sile in velikost dela na kg telesne teže. Silo reakcije podlage smo merili unilateralno in bilateralno (Slika 4).

DINAMIKA ODRIVNE MOČI

2 TENZIO PLATFORME
– KISTLER TYPE 9286 A
FREKVENCA 1000 HZ
MERENJE SILE
BILATERALNO I UNILATERALNO



Slika 4: Registracija dinamičnih parametrov vertikalnega skoka

Za analizo elektromiografske aktivnosti (EMG) smo uporabili 16 kanalni elektromiograf (BTS Pocket EMG, Myolab). Sestavljen je bil iz dveh enot: mobilne enote (HP Ipaq 4700), ki zajema vse EMG signale in jih s pomočjo brezžične tehnologije (Wi-Fi) prenaša v stacionarno enoto (Slika 5). Spremljali smo EMG aktivacijo 7 mišic na levi i 7 mišic na desni nogi (m. gluteus maximus, m. rectus femoris, m. vastus medialis, m. vastus lateralis, m. tibialis, m. biceps femoris, m. gastrocnemius – medialis) in eno od mišic iztegovalk trupa (m. erector spinae). Za detekcijo površinske elektromiografske aktivnosti mišic smo uporabili bipolarne površinske elektode Ag-AgCl (Ambu Blue Sensor SE – 00-S/50, Denmark), ki so bile nameščene na ustrezno lokacijo motorične enote mišic po temeljiti preparaciji kože (Slika 6).

ELEKTROMIOGRAFIJA – EMG ODRIVNE MOČI

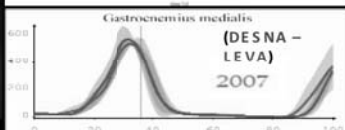
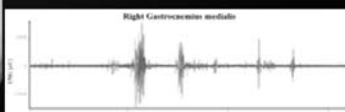
16 KANALNI ELEKTROMIOGRAF
(BTS POCKET EMG, MYOLAB)

1. Mobilna enota (HP Ipaq 4700) za zajam EMG signala
2. Telemetrijski prenos signala (Wi-Ei) u stacionarno jedinico



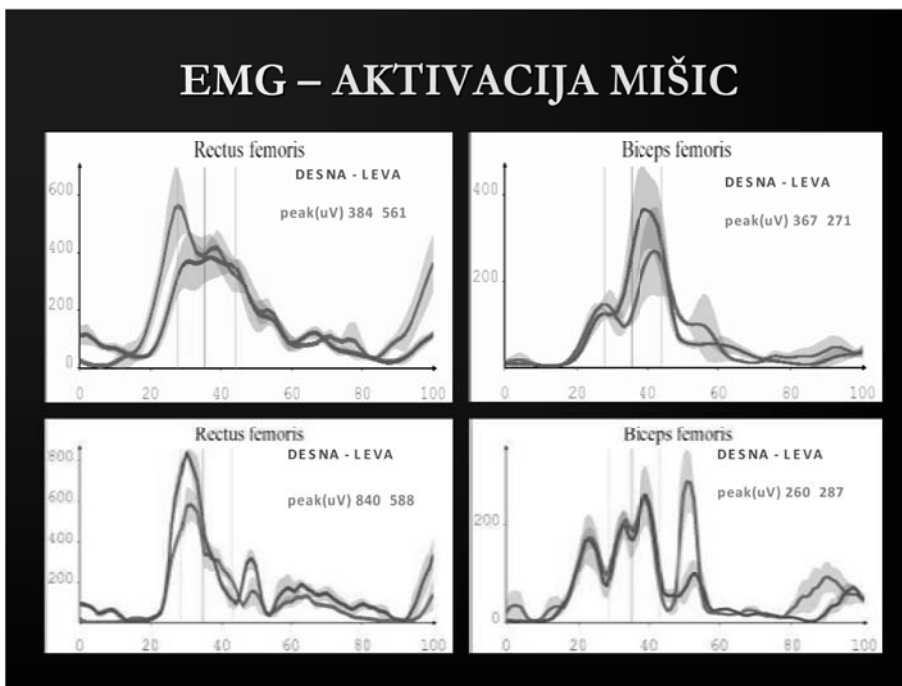
Slika 5: Elektromiografija odrivne moč pri vertikalnih in globinskih skokih.

EMG APLIKACIJA



Slika 6: Površinska aplikacija elektrod na posamezne mišične skupine, prenos v stacionarno enoto z surovim in filtriranim EMG signalom.

Posnete signale smo v nadaljevanju eksperimenta ustrezno filtrirali in pogladili (ang. smoothing). Najprej smo uporabili visokopropustni (high-pass filter) filter Hamming pri frekvenci 30 Hz za odstranitev artefaktov. Nato smo signal integrirali z algoritmom RMS s časovno osnovo 20 milisekund. Nadaljnje glajenje smo izvedli z nizkopropustnim filtrom Hamming pri frekvenci 10 Hz (slika 7). Statistična analiza rezultatov je bila v nadaljevanju obdelana z računalniškim paketom SPSS.



Slika 7: EMG aktivacija mišic rectus femoris in biceps femoris pri vertikalnem skoku

3. REZULTATI MERILNE PROCEDURE

Po standardnem protokolu testiranja izvajamo vertikalni skok iz statičnega položaja, pri čemer je kot v kolenu okoli 90° . Na ta način je izločen vpliv elastične energije v mišicah in tetivah ter refleksnih mehanizmov, ki mišice dodatno aktivirajo. Skok izvajamo brez pomoči rok, ki so fiksirane v višini bokov. S testom ocenjujemo koncentrično komponento hitre moči. Višino skoka je odvisna od iztegovalk skočnega sklepa, kolenskega sklepa in bokov. Na osnovi sile reakcije podlage izmerjene s pomočjo tenziometrijskih platform in 3 – D kinematične analize (sistem 9 kamer SMART) smo analizirali naslednje kinematične in dinamične parametre (Tabela 1).

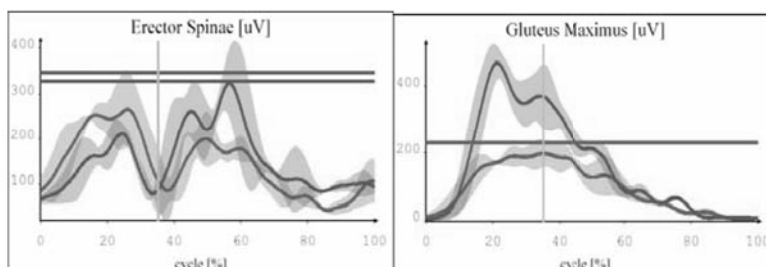
Pri pregledu tabele 1 lahko ugotovimo tiste dinamične in kinematične parametre, ki generirajo višino skoka. Ključni parameter je hitrost odziva, ki je odvisen od impulza sile v koncentrični fazi, maksimalne sile reakcije podlage in velikosti moči (W) na kg

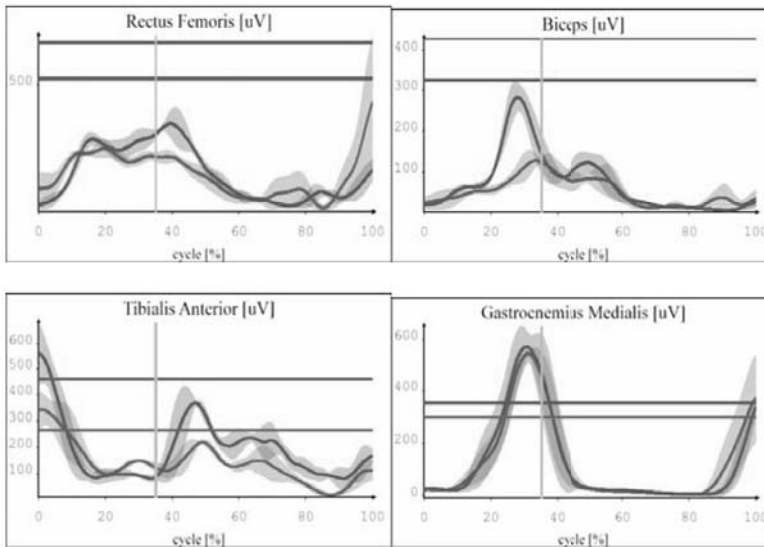
telesne teže . Pri vseh vertikalnih skokih je eden od limitirajočih dejavnikov razpoložljivi čas za izvedbo eksplozivnega gibanja. Čas generiranja sile v koncentrični fazi znaša 314 ± 6 do 326 ± 17 milisekund. Glede na raziskave nekaterih avtorjev [2,4,6] so vertikalni skoki s pol čepa v največji meri odvisni od kontraktilnih karakteristik mišic in bistveno manj od refleksnih mehanizmov in predaktivacije. Optimalni kot v kolenu je 89° do 90° in v bokih 71° do 84° .

Tabela 1: Dinamični in kinematični parametri vertikalnega skoka (squat jump)

TEST	UNIT	A	B
SQUAT JUMP			
Višina skoka	cm	45.8 ± 0.5	37.7 ± 0.6
Čas koncentrične faze	ms	314 ± 6	326 ± 17
Koncentrično delo	J/kg	6.5 ± 0.2	5.4 ± 0.4
Učinkovitost skoka	cm/J	7.0 ± 0.3	7.1 ± 1.2
Maksimalna moč	W/kg	48.7 ± 1.1	41.7 ± 0.3
Čas leta	ms	572 ± 4	504 ± 4
Hitrost odriva	$m \cdot s^{-1}$	2.37 ± 0.3	2.30 ± 0.3
Maksimalna sila	N	846 ± 1.4	744 ± 1.2
Impulz sile	Ns	172 ± 3	154 ± 6
Kot v kolčnem sklepu	deg	84 ± 2	71 ± 2
Kot v kolenskem sklepu	deg	89 ± 2	90 ± 4
Kot v skočnem sklepu	deg	30 ± 1	38 ± 1

Ti koti so pomembni, saj je odzivna akcija v začetni fazi povezana prav z delovanjem iztegovalk kolka in trupa. Končna hitrost odriva pa je odvisna predvsem od iztegovalk kolenskega in skočnega sklepa.





Slika 8: EMG aktivacija mišic spodnjih ekstremitet pri vertikalnem skoku

Pri vertikalnem skoku s dominantno koncentrično komponento (Slika 8) eksplozivne moči (squat jump) se v realizacijo odriava prvo vključuje m. erector spinae in m. gluteus maximus, nato m. vastus lateralis in medialis ter na koncu m. gastrocnemius - medialis. Največjo stopnjo aktivacije lahko ugotovimo pri m. gluteus maximusu in m gastrocnemiusu – medialisu. Aktivacije mišic v squat jumpu poteka po proksimalno – distalnemem principu.

4. ZAKLJUČEK

Biomehanska diagnostika eksplozivne moči spodnjih ekstremitet je izjemno pomemben sestavni del monitoringa sodobnega procesa treninga športnikov. Rezultati meritev raznih tipov vertikalnih in globinskih skokov nam dajejo temeljne informacije o stanju in delovanju mišično-živčnega sistema. Na osnovi teh informacij je mogoče bistveno bolj natančno programirati in kontrolirati proces treninga v prostoru moči. Predstavljena je naj sodobnejša tehnologija in merilni postopki za diagnostiko eksplozivne moči, predvsem z vidika spremljanja dinamičnih in kinematičnih in EMG parametrov hitrih gibanj kot so vertikalni in globinski skoki.

LITERATURA

1. Bührle, M., Schmidtbleicher, D., & Ressel, H. (1983). Die spezielle Diagnose der einzelnen Kraftkomponenten in Hochleistungssport. *Leistungssport*, 3, 11-16.

2. Bobbert, M., Huijing, P., van Ingen Schenau, G. (1987). Drop jumping I. The influence of jumping technique on the biomechanics of jumping. *Medicine and Science in Sport and Exercise*, 19:332-338.
3. Bosco, C. (1992). L'evaluation de la force par le test de Bosco. Roma, Societa Stampa Sportive.
4. Čoh., M. (2008). Biomechanical diagnostic methods in athletic training. Inštitut za kineziologijo, Fakulteta za šport.
5. Dolenc, A. (1999). Vpliv treniranja globinskih skokov s plantarno in dorsarno tehniko na delo gležnja pri globinskih skokih. Doktorska disertacija, FŠ, Ljubljana.
6. Gollhofer, A., Kyrolainen, H. (1991). Neuromuscular control of the human leg extensor muscles in jump exercises under various stretch-load conditions. *International Journal of Sports Medicine*, 12, 34-40.
7. Luhtanen, P., & Komi, P. (1980). Force-, power- and elasticity-velocity relationship in walking, running and jumping. *European Journal of Applied Physiology* 44 (3), 79-289.
8. Šarabon, N., Fajon, M., Zupanc, O., & Draksler, J. (2005). Stegenske strune (Hamstrings Injuries). *Šport*, 3, (53), 45-52.

ŠTUDENTSKE APLIKACIJE

SISTEM ZA VERIFIKACIJO OSEBE NA PODLAGI PRSTNEGA ODTISA

Uroš Klopčič, Peter Peer

Laboratorij za računalniški vid

Fakulteta za računalništvo in informatiko

E-pošta: uros.klopacic@gmail.com, peter.peer@fri.uni-lj.si

POVZETEK: V tem delu je predstavljen sistem za verifikacijo oseb na podlagi prstnega odtisa. V osnovi gre za prototipni sistem, katerega namen je postavitev temeljev in ogrodja, katero bi omogočalo nadaljnji razvoj in testiranje. Sistem sestoji iz korakov segmentacije, izboljšave kvalitete prstnega odtisa z Gaborjevimi filtri, tanjšanja, iskanje značilk, klasifikacije ter primerjanja. Algoritem smo testirali na štirih testnih množicah s tekmovanja Fingerprint Verification Competition 2002 (FVC2002). Algoritem daje zadovoljive rezultate, več težav se pojavi predvsem pri prstnih odtisih slabe kvalitete v koraku izboljšanja kvalitete prstnega odtisa. Čas procesiranja 3 sekund ustreza zahtevam enostavnega sistema za verifikacijo oseb na podlagi prstnega odtisa.

1. UVOD

Prstni odtisi že dolgo veljajo za eno od najbolj zanesljivih in natančnih biometričnih značilnosti, zato so se že zgodaj uveljavili pri procesu ugotavljanja identitete posameznika. S pojavom ter razvojem računalniške tehnologije, se je zamudna ročna identifikacija posameznika zamenjala s hitrimi in zmogljivimi sistemi za identifikacijo oseb na podlagi prstnega odtisa (angl. *AFIS*). Danes so prstni odtisi že popolnoma uveljavljeni, uporabljajo se predvsem pri kontroli dostopa ter policijskem delu.

Za namene kontrole dostopa v laboratorij smo tudi sami želeli razviti sistem za verifikacijo oseb na podlagi prstnega odtisa. Pri tem si kot glavni cilj nismo zastavili 100% zanesljivost sistema, temveč dobro ogrodje ter modularno zgradbo, na kateri bi bilo uveljavljanje prihodnjih nadgradenj čim bolj enostavno. Šele skozi proces posodobitev pa bi sistem pridobil potrebno zanesljivost ter natančnost.

Sistem bi lahko v grobem ločili v tri korake: zajem prstnega odtisa, iskanje značilk ter odločanje. Za zajem prstnega odtisa smo uporabili dva čitalca prstnih odtisov, vendar smo zaradi lažjega dela pri samem razvoju sistema uporabili v naprej zbrano zbirko slik prstnih odtisov. Večji poudarek je bil predvsem na koraku iskanja značilk, od katerega je odvisen tudi končni rezultat v koraku odločanja.

2. OBDELAVA PRSTNEGA ODTISA

2.1 Segmentacija

Pri segmentaciji poskušamo čim bolj ločiti področje, na katerem se nahaja prstni odtis, od ozadja. Pri tem smo uporabili metodo, ki je modificirana verzija pristopa opisanega v [3]. Kot prvo se poišče vrednost, ki predstavlja ozadje. V ta namen izračunamo histogram negativa slike. Nato potujemo po histogramu od indeksa 0 do 255 in opazujemo vrednost histograma. Indeks, kjer je ta vrednost prvič večja od praga $ThrF$ (1000 v našem primeru), predstavlja vrednost ozadja. Ta vrednost se odšteje od vsakega slikovnega elementa (i, j) na sliki. Nato z enačbo (1) povečamo dinamično razdaljo med grebeni in dolinami in še enkrat izračunamo negativno sliko.

$$neg_stretched(i, j) = \frac{neg_subtracted(i, j)}{\max_{i, j}(neg_subtracted(i, j))} \cdot 255 \quad (1)$$

V drugem delu sliko razdelimo na bloke enake velikosti $W \times W$ ($W = 9$ v našem primeru). Za vsak blok s središčem v (i, j) se izračuna štiri parametre: srednja vrednost, varianca, gradient in ROI , prikazan v enačbi (2).

$$ROI(i, j) = \sqrt{\frac{1}{W^2} \cdot \frac{(V_x(i, j)^2 + V_y(i, j)^2)}{V_c(i, j)}}, \quad (2)$$

kjer velja

$$V_x(i, j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} 2 \cdot \partial_x(u, v) \cdot \partial_y(u, v), \quad (3)$$

$$V_y(i, j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} (\partial_x^2(u, v) - \partial_y^2(u, v)), \quad (4)$$

$$V_c(i, j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} (\partial_x^2(u, v) + \partial_y^2(u, v)) \quad (5)$$

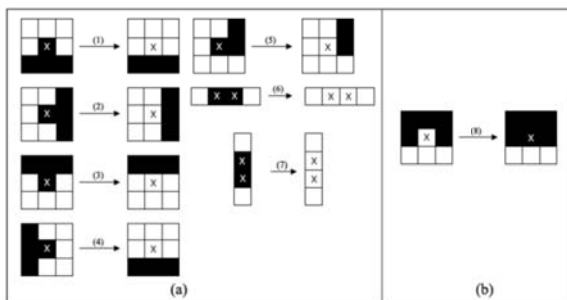
ROI poda stopnjo gotovosti, ki nam glede na prag pove, ali slikovni element pripada ozadju ali ne. ∂_x in ∂_y sta jakosti gradienta v x in y smeri, katera pridobimo s Sobelovim operatorjem [8]. Če je srednja vrednost večja in ostali parametri manjši od nastavljenih pragov, potem je blok označen za ozadje.

S tem korak segmentacije še ni končan. Pogosto se namreč zgodi, da so nekateri bloki napačno označeni za ozadje oziroma ospredje. Zato je potrebna še naknadna obdelava, kjer se take bloke prepozna in popravi. V ta namen se uporabi množica hevrističnih pravil (slika 1).

2.2 Izboljšava kvalitete slike prstnega odtisa

Pri prepoznavanju prstnih odtisov je zelo pomembna predhodna obdelava zajete slike prstnega odtisa, saj slaba kvaliteta negativno vpliva na postopke iskanje značilki. Pri

zajemu namreč pogosto pride do napak kot so šum in smeti, vzrok je lahko tudi prevelik pritisk na čitalec, pogoste pa so tudi mehanske poškodbe samega prsta (urez, praska). Zaradi tega smo v našem sistemu uporabili postopek izboljšanja kvalitete opisan v [4].



Slika 1: Hevristična pravila za odstranjevanje (a) blokov, označenih za ospredje, (b) blokov, označenih za ozadje

Nad prstnim odtisom se najprej izračuna orientacijsko polje s pomočjo metode gradientov. Orientacijsko polje nam za vsak del slike pove, v katero smer je prstni odtis usmerjen. Dobljeno orientacijsko polje pa ni povsem zanesljivo, saj vsebuje preveč šuma. Zato se izvede glajenje orientacijskega polja. Pri tem smo uporabili metodo [5], ki velikost okolice glajenja $\Omega(s)$ pametno prilagaja glede na doslednost orientacije. Za izračun slednje se uporabi enačba (6).

$$Cons(s) = \frac{\sqrt{(\sum_{(i,j) \in \Omega(s)} \cos(2\theta(i,j)))^2}}{O(s)} \quad (6)$$

$O(s)$ predstavlja število orientacij $\theta(i,j)$ v $\Omega(s)$. Kadar je doslednost majhna, se uporabi veliko okolico glajenja, s čimer omejimo vpliv šuma, obratno pa uporabimo majhno okolico, saj vemo, da je orientacija zanesljiva. Za vsak blok izračunamo $Cons(s)$ in njegovo orientacijo na podlagi zunanjih $8 \cdot s$ blokov iz njegove $(2 \cdot s + 1) \times (2 \cdot s + 1)$ velike okolice. Koraki glajenja so naslednji:

1. Izračunaj enotski vektor z dvakratno orientacijo $[\cos(2\theta(i,j)), \sin(2\theta(i,j))]$. Nastavi $s = 1$ in uporabi vrednosti enotskega vektorja pri izračunu $Cons(s)$.
2. Povečaj s za 1. Izračunaj $Cons(s)$ v $\Omega(s)$, kjer je $\Omega(s)$ $8 \cdot s$ zunanjih blokov.
3. $Cons(s)$ doseže vrednosti na intervalu $[0,1]$. Če je $Cons(s)$ manjši od praga (0,5 v našem primeru) ali pa manjši od $Cons(s - 1)$, potem pojdi na korak 2; na korak 2 se vračaš, dokler ne dosežeš maksimuma vrnitev (5 v našem primeru).
4. Če je s enak maksimumu, nastavi $\Omega(s)$ na okolico velikosti 3×3 blokov, sicer pojdi na korak 5.
5. Izračunaj zglajeno orientacijo po sledeči enačbi:

$$\bar{\theta}(i_c, j_c) = \frac{1}{2} \arctan\left(\frac{\sum_{(i,j) \in \Omega(s)} \sin(2\theta(i,j))}{\sum_{(i,j) \in \Omega(s)} \cos(2\theta(i,j))}\right) \quad (7)$$

kjer sta i_c in j_c koordinati točke, ki predstavlja središče trenutno obravnavanega bloka.

Izračunu orientacije sledi še izračun frekvence [4], nato pa se izvede dejansko filtriranje s pomočjo Gaborjevih filtrov.

2.3 Binarizacija in tanjšanje grebenov

Binarizacija se izvede s pomočjo prosto dostopnega orodja AForge .NET. Kot rezultat dobimo sliko, kjer so grebeni predstavljeni z bitom 1 in ozadje z bitom 0.

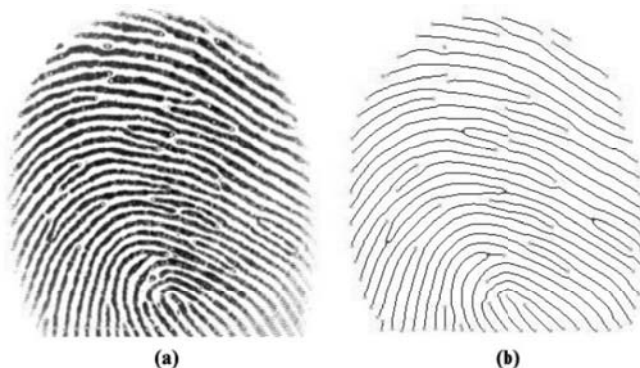
Namen tanjšanja grebenov je pridobiti skelet prstnega odtisa, kjer so grebeni široki natanko 1 slikovni element. V ta namen smo uporabili algoritem v [6], ki prične odstranjevati slikovne elemente na zunanjih robovih grebenov, dokler le-ti niso debeli zgolj 1 slikovni element. To dosežemo z uporabo 21 pravil, ki jih uporabimo v vsaki iteraciji nad vsakim slikovnim elementom. Algoritem uporablja tudi 4 posebna pravila za tanjšanje diagonalnih črt, poleg tega pa vsebuje tudi 12 pravil za predpripravo na tanjšanje.

2.4 Iskanje značilk

Naloga algoritma iskanja značilk je poiskati osnovne značilke: razcep (angl. *bifurcaton*) in zaključek (angl. *ridge ending*). Kadar imamo na voljo skelet prstnega odtisa, je naloga razmeroma enostavna. Pri iskanju zaključkov grebenov se opiramo na število križišč [10] v trenutni točki (angl. *Crossing Number*). Enačba (8) definira število križišč $cn(\mathbf{p})$ točke \mathbf{p} v binarni sliki kot polovico vsote razlik med pari sosednjih točk v okolici točke \mathbf{p} .

$$cn(\mathbf{p}) = \frac{1}{2} \sum_{i=1}^8 |\text{val}(\mathbf{p}_{i \bmod 8}) - \text{val}(\mathbf{p}_{i-1})|, \quad (8)$$

kjer so $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_7$ točke, ki pripadajo okolici točke \mathbf{p} , $\text{val}(\mathbf{p}) \in \{0, 1\}$ pa predstavlja vrednost točke. Za točko \mathbf{p} velja, da je zaključek grebena, če je $cn(\mathbf{p}) = 1$.



Slika 2: Najdene značilke, prikazane na (a) sivinski sliki in (b) sliki s skeletom prstnega odtisa

Pri iskanju razcepov si pomagamo z množico pravil. Za vsako točko se njena 3×3 okolica primerja s 24 možnimi maskami za razcep na sliki 2. Če najdemo ujemanje, potem je ta kandidat dejansko razcep.

Kljub temu, da smo kvaliteto slike prstnega odtisa izboljšali in da smo iskali značilke le na področju, kjer se nahaja prstni odtis, algoritem še vedno najde večje število nepravilnih značilk, ki niso primerne za nadaljnjo obdelavo. Zato vse značilke preverimo še z algoritmom verifikacije značilk [1]. S tem odstranimo nepravilne strukture (špice, mostove, kratke greben, itd.) in posledično povečamo natančnost sistema.

2.5 Klasifikacija

V postopku klasifikacije poskušamo prstnemu odtisu določiti razred glede na obliko grebenov. Pri tem smo uporabili 5 najbolj uporabljenih razredov Galton-Henryevega sistema: lok, štorast lok, leva in desna zanka ter spirala. Določitev razreda temelji na lokaciji in številu singularnih točk (jedra in delte). Te lahko določimo z uporabo polja ukrivljenosti (angl. *curvature map*) [7].

Za izračun polja ukrivljenosti sliko razdelimo na bloke velikosti $W \times W$ ($W = 7$ v našem primeru). Ukrivljenost bloka se izračuna glede na ukrivljenost 8 sosednjih blokov v njegovi 3×3 okolici. Natančneje, ukrivljenost bloka je enaka polovici razlike ukrivljenosti blokov, na katere kaže smer obravnavanega bloka (enačba 9).

$$B_k = \frac{|O(B_1) - O(B_2)|}{2}, \quad (9)$$

kjer sta B_1 in B_2 bloka, na katera kaže smer središčnega bloka. Singularne točke iz polja ukrivljenosti dobimo na naslednji način:

1. Določimo prag ukrivljenosti T_c ter prag razdalje T_d (0,35 in 4 v našem primeru)
2. Označimo z S_c množico singularnih točk, katerih ukrivljenost je večja od T_c .
3. Izberemo katerakoli dva bloka B_i in B_j iz množice S_c . Če je razdalja med njima manjša od T_d , se blok z manjšo ukrivljenostjo odstrani iz množice S_c .
4. Pojdi na korak 3, dokler velikost množice S_c ni stalna.
5. Množica S_c vsebuje resnične singularne točke.

Za prstne odtise tipa lok je znano, da ne vsebujejo izrazitih singularnih točk. Ker pa naš postopek zahteva določitev referenčne točke prstnega odtisa, se za jedro vzame kar blok z najvišjo ukrivljenostjo.

Tip dobljenih singularnih točk (jedro, delta) določimo z metodo Poincare Index [9]. Ko poznamo število jeder N_c in delt N_d , lahko klasificiramo prstni odtis glede na naslednja pravila:

1. Če je $N_c = 2$, potem je prstni odtis tipa spirala.
2. Če je $N_c = 0$ in $N_d = 0$, potem je prstni odtis tipa lok.
3. Če je $N_c = 1$ in $N_d = 1$, potem je prstni odtis lahko tipa štorast lok ali zanka. Potrebna je nadaljnja analiza. Med jedrom in delto potegnemo ravno črto. Pri štorastem loku je črta vzporedna z lokalnimi orientacijami, med tem ko jih pri

zanki seka. Naj bo λ kot, ki ga tvori daljica med jedrom in delto in $\eta_1, \eta_2, \dots, \eta_n$ koti lokalnih usmeritev vzdolž omenjene daljice. Če je povprečna vsota (enačba 10)

$$\sum_{i=1}^n \sin(\eta_i - \lambda) \quad (10)$$

manjša od pragu (0,3 v našem primeru), potem je prstni odtis tipa šotorast lok, sicer je tipa zanka. Leve zanke ločimo od desnih s pomočjo enačbe (11) - če je rezultat manjši od 0, potem je leva, sicer desna zanka.

$$(B_x - C_x)(D_y - C_y) - (B_y - C_y)(D_x - C_x) \quad (11)$$

kjer indeksa x in y označujeta x in y koordinati točk.

4. Sicer tipa prstnega odtisa ni mogoče določiti.

Referenčno točko tako predstavlja kar jedro. V primeru dveh jeder pa v ta namen izberemo kar središčno točko med obema jedroma.

2.6. Primerjanje

Primerjanje temelji na primerjavi lokacije, kota, usmeritve ter tipa značilk. V našem algoritmu se značilke najprej pretvorijo v polarni koordinatni sistem, kjer se za središče vzame referenčno točko.

Zadnji manjkajoči podatek je orientacija referenčne točke. Če obstaja dve jedri, je orientacija enaka orientaciji premice skozi ti dve točki. Sicer se okoli referenčne točke definira 16 smeri, ločenih s korakom $\pi/8$. Orientacija referenčne točke je smer, kateri se lokalne orientacije najbolj prilagajajo. Značilke nato predstavimo v polarnem koordinatnem sistemu s štirimi parametri (r, e, s, Φ): razdaljo od referenčne točke, polarnim kotom, tipom značilke ter kotom med r in smerjo značilke.

Primerjanje med vhodnim vektorjem značilk $P_i = \{(r_{i1}, e_{i1}, s_{i1}, \Phi_{i1})^T, \dots, (r_{iM}, e_{iM}, s_{iM}, \Phi_{iM})^T\}$ in tistim iz podatkovne zbirke $Q_j = \{(r_{j1}, e_{j1}, s_{j1}, \Phi_{j1})^T, \dots, (r_{jN}, e_{jN}, s_{jN}, \Phi_{jN})^T\}$ poteka na naslednji način:

1. Vsaka značilka iz P_i se primerja z vsako značilko iz Q_j .
2. Naj M_0 in N_0 označujeta število značilk v prvem in drugem prstnem odtisu, M_N pa število uspešnih ujemanj. Kadar je $M_N / \sqrt{N_0 M_0}$ večji od praga β , potem smatramo obravnavana prstna odtisa za enaka.
3. Vzemi naslednji vektor Q_{j+1} .

Zaradi šuma in nelinearnih deformacij je nemogoče natančno določiti lokacijo značilk glede na referenčno točko. Zato mora primerjalni algoritem upoštevati določeno toleranco. V našem primeru to dosežemo z uporabo dinamične mejne škatle (angl. *Variable Bounding Box*). Naj kot α_b in radij r_b označujeta velikost škatle. Kadar je radij značilke majhen, lahko majhna deformacija povzroči veliko spremembo v kotu radija značilke, medtem ko na radij nima vpliva. V tem primeru bi moral biti α_b večji in r_b manjši. Kadar pa je radij značilke velik, lahko majhna sprememba v kotu radija značilke povzroči veliko spremembo v lokaciji značilke. Takrat α_b pomanjšamo in r_b povečamo,

saj se vse deformacije med referenčno točko ter značilko seštevajo, kar se pozna na radiju.

3. REZULTATI

Sistem smo testirali na štirih množicah slik prstnih odtisov iz FVC2002. Vsaka množica je vsebovala 800 slik. Slike so bile pridobljene z optičnim (DB1, DB2) in kapacitivnim (DB3) čitalcem, zbirka DB4 pa je vsebovala sintetično generirane prstne odtise.

Vsi koraki z izjemo koraka izboljšanja kvalitete so dali zadovoljive rezultate. Pri izboljšanju kvalitete smo imeli probleme predvsem s prstnimi odtisi slabe kvalitete, saj algoritem ni bil zmožen ločevati med sosednjimi grebeni. V nekaterih primerih prstnih odtisov slabe kvalitete naš sistem ni bil sposoben podati natančne lokacije in števila singularnih točk, zaradi česar je bila klasifikacija prstnega odtisa nemogoča. Za primerjavo z ostalimi sistemi smo izmerili stopnji FNMR (angl. *False Non-Match Rate*) - in FMR (angl. *False Match Rate*). Prva označuje neujemanje dveh vzorcev istega osebkva, druga pa ujemanje dveh vzorcev različnih osebkov. Rezultati v tabeli 1 nam povejo, da naš sistem ni popolnoma zanesljiv. Zaradi velikega vpliva kvalitete na rezultate smo testirali sistem tudi s 64 slikami boljše kakovosti. Slednji rezultati so podani v tabeli 2. Opazimo veliko izboljšanje FNMR pri DB1 ter DB2, malo manj pa pri ostalih množicah, saj so najboljše slike iz teh množic še zmeraj vsebovale preveč šuma. Na koncu smo s sliko 3 podali še skupne rezultate za vsako testno množico. Predstavljeni so bili s krivuljo ROC (angl. *Receiver Operating Characteristic*), ki prikazuje FNMR in FMR v odvisnosti od praga. Kjer se krivulji FNMR in FMR sekata, je t.i. točka EER (angl. *Equal Error Rate*). Na sliki sta s praznim kvadratom in polnim krogcem označeni oceni EER tudi za predzadnji in najboljši algoritem iz FVC2002 [2]. Povprečen čas procesiranja je znašal 3 sekunde.

Tabela 1: Stopnji FNMR in FMR pri pragu $\beta = 0.3$.

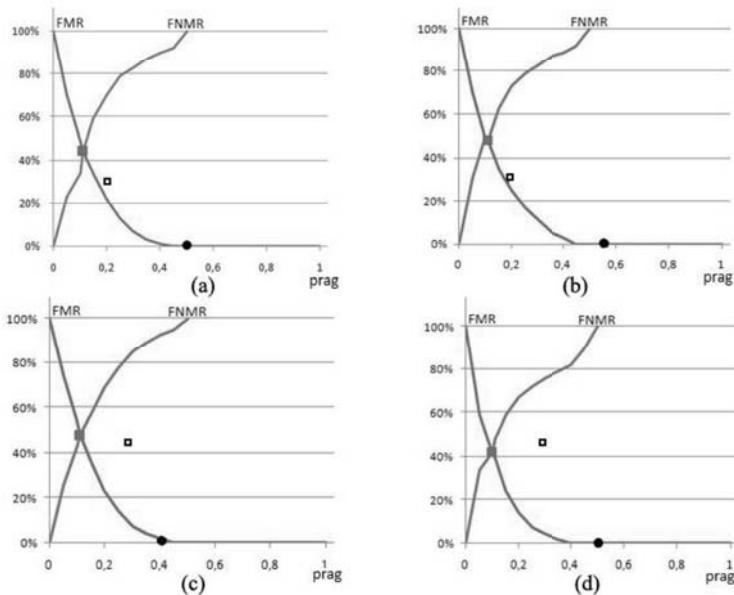
Stopnja	TESTNE MNOŽICE			
	DB1	DB2	DB3	DB4
FNMR (%)	89.3	88.6	91.2	81.3
FMR (%)	1.7	3.7	2.4	0.9

Tabela 2: Stopnji FNMR in FMR pri pragu $\beta = 0.3$ za slike dobre kakovosti.

STOPNJA	TESTNE MNOŽICE			
	DB1	DB2	DB3	DB4
FNMR (%)	6.1	41.1	42.6	8.2
FMR (%)	0	0	0	0

4. ZAKLJUČEK

Sistem še ni dovolj zanesljiv in potrebuje nekaj popravkov. Največ napak se zgodi predvsem zaradi slik slabe kakovosti. Zato bo poudarek v nadaljnjem razvoju predvsem pri jasni določitvi frekvence grebenov, ki mora biti robustna in odporna na šum. Prav tako potrebuje nekaj izboljšav tudi postopek določitve lokacije singularnih točk, kjer trenutno obravnavamo nekaj različnih pristopov. Izdelan sistem predstavlja dobro osnovo in ponuja še veliko možnosti za nadgradnjo.



Slika 3: ROC krivulje za testne množice (a) DB1, (b) DB2, (c) DB3 in (d) DB4.

LITERATURA

1. M.U. Akram, A. Tariq, S.A. Khan, S. Nasir, Fingerprint image: pre and post-processing, *International Journal of Biometrics* 2008, vol. 1(1), str. 63-80, 2008
2. <http://bias.csr.unibo.it/fvc2002/>
Informacije o temovanju Fingerprint Verification Competition 2002
3. M.M. Hadhoud, W.S. El Kilani, M.I. Samaan, An adaptive algorithm for fingerprints image enhancement using gabor filters, *ICCES '07. International Conference on Computer Engineering & Systems*, str. 227-236, November 2007
4. L. Hong, Y. Wan, A. Jain, Fingerprint image enhancement: algorithm and performance evaluation, *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 20(8), str. 777-789, 1998
5. M. Liu, X. Jiang, A.C. Kot, Fingerprint Reference Point Detection, *EURASIP Journal on Applied Signal Processing*, vol. 2005, str. 498-509, Januar 2005
6. P.M. Patil, S.R. Suralkar, F.B. Sheikh, Rotation invariant thinning algorithm to detect ridge bifurcations for fingerprint identification, *17th IEEE International Conference on Tools with Artificial Intelligence*, str. 641-649, 2005
7. J. Qi, M. Xie, A robust algorithm for fingerprint singular point detection and image reference direction determination based on the analysis of curvature map, *IEEE Conference on Cybernetics and Intelligent Systems*, str. 1051-1054, 2008
8. J. Qi, M. Xie, Segmentation of fingerprint images using the gradient vector field, *IEEE Conference on Cybernetics and Intelligent Systems*, str. 543-545, September 2008
9. T. Tang, X. Wu, M. Xiang, An improved fingerprint singular point detection algorithm based on continuous orientation field, *International Symposium on Computer Science and Computational Technology ISCCST, 2nd Edition*, str. 454-457, 2008
10. F. Zhao, X. Tang, Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction, *Pattern Recognition*, vol. 40(1), str. 1270-1281, 2007

OPENCV IN IPHONE: DETEKCIJA OBRAZOV V REALNEM ČASU

Roman Avsec, Peter Peer

GameTeam

Fakulteta za računalništvo in informatiko, Tržaška c. 25, 1000 Ljubljana

E-pošta: roman.avsec@gmail.com, peter.peer@fri.uni-lj.si

URL: <http://gameteam.fri.uni-lj.si/>

POVZETEK: *V č lanku bo predstavljenih nekaj adaptacij obstoječih in implementacij novih optimizacijskih tehnik, s katerimi lahko optimiziramo metode detekcije obraza v odprtokodni knjižnici za računalniški vid OpenCV. Na kratko bomo predstavili iPhone procesor ter iPhone razvojno okolje, vanj vključili knjižnico OpenCV in jo poskusili optimizirati do te mere, da se bo detekcija obraza izvršila v realnem času. Celoten postopek bo dokumentiran tako, da bo lahko bralec, ki ga tematika zanima, s pomočjo navodil in izvorne kode tudi samostojno vključil knjižnico v svoj projekt.*

1. UVOD

V okviru skupine GameTeam, ki deluje na Fakulteti za računalništvo in informatiko Ljubljana, smo poleti 2009 za iPhone razvili zabavno aplikacijo "15 seconds of fame" [1], ki iz posnete fotografije s pomočjo OpenCV knjižnice detektira obraz, ga izreže, poveča in transformira v abstrakten portret. Ugotovili smo, da je sicer OpenCV knjižnico relativno enostavno vključiti v iPhone razvijalno okolje, je pa detekcija zelo počasna: na sliki velikosti 400×400 slikovnih elementov se je izvajala od 6 do 10 sekund, odvisno od slike in parametrov detekcije. Zanimalo nas je, ali lahko detekcijo z uporabo OpenCV knjižnice na tej mobilni platformi optimiziramo do te mere, da čas procesiranja obraza skrajšamo v realni čas (real-time).

Na različnih virih po internetu smo naše začetne rezultate primerjali z rezultati ostalih poskusov in ugotovili, da je 6 do 10 sekund čas primerljiv s tistim, ki so ga dosegali drugi posamezniki pri detekciji. Naleteli pa smo na dva zapisa ([2], [3]), kjer poročajo, da so detekcijo na sliki dimenzij iPhone ekrana uspeli skrajšati na 100 – 300ms. Naš cilj je bil poskusiti, če in koliko se lahko temu času približamo z implementacijo predlaganih optimizacijskih tehnik in uvedbo novih.

V pomoč nam je po naključju priskočil Apple, ki je decembra 2009 odobril uporabo do takrat neuradne funkcije `UIImageScreenImage()`, ki preko zajema slike ekrana omogoča ob primerni pripravi tudi zajem videa s kamere. Pred to odobritvijo te funkcije je bil edini način zajema videa uporaba internih API klicev, ki bi povzročili zavrnitev aplikacije, če bi jo poskusili predstaviti na AppStore.

Video se zajema v obliki posameznih nekomprimiranih sličic, celoten zajem ene sličice pa traja cca 50 ms, zato pričakujem, da bo "realni čas" pomenil precej manj kot 20 sličic na sekundo.

2. OPENCV, NAMEN IN UPORABA

OpenCV (Open Computer Vision) je odprtokodna knjižnica z začetki razvoja v podjetju Intel in združuje čez 500 optimiziranih algoritmov za pomoč pri računalniškem vidu in obdelavi slik. Zabeleženih je čez 2 milijona prenosov, uporabna pa je na področjih od splošnega procesiranja računalniških slik, geometričnih opisov, transformacij, strojnega učenja, sledenja in mnogih drugih (povzeto po [5]). Za naše potrebe bomo uporabili funkcije za detekcijo objektov, specifično obrazov, ki v OpenCV knjižnici temeljijo na klasifikaciji Haarovih značilnosti.

3. IPHONE RAZVOJNO OKOLJE

Razvojno za iPhone okolje se imenuje Xcode in teče na operacijskem sistemu OS X. Avtor okolja je podjetje Apple in ga daje na voljo brezplačno vsem, ki se na njihovi spletni strani <http://developer.apple.com> registrirajo kot razvijalci. Razvijalci lahko aplikacije testirajo v priloženem iPhone simulatorju, ki pa nima implementiranih funkcij, kot so: pospeškometer, GPS, kamera. Ker je za naš primer kamera ključnega pomena, smo razvoj in testiranje opravljali direktno na napravi iPhone 3G, za kar je potreben *Developer certificate*, ki je sicer pri Applu na voljo proti letnemu plačilu.

4. IPHONE PROCESOR

iPhone poganja procesor iz ARM družine. Starejša modela (2G in 3G) imata vgrajen procesor iz družine ARM 11 z ARMv6 arhitekturo, novejši model (3GS) pa procesor družine Cortex z arhitekturo ARMv7. Oba procesorja sta sicer binarno kompatibilna, med njima pa so razlike, ki jih prikazuje Tabela 1 (povzeto po [8]).

Tabela 1: razlike med procesorjema iPhone 3G in iPhone 3GS


	iPhone 2G, 3G (ARM11)	iPhone 3GS (ARM Cortex A8)
Proces izdelave	90nm	65nm
Velikost cevovoda	8-stopenjski	13-stopenjski
Frekvenca ure	412MHz	600MHz
Velikost predpom. L1	16KB I-Cache + 16KB D-Cache	32KB I-Cache + 32KB D-Cache
Velikost predpom. L2	brez	256KB

Vidimo lahko, da je procesor novejšega iPhone-a hitrejši in z več tako prvnivojskega kot drugonivojskega predpomnilnika. Pred starejšim procesorjem ima še veliko prednost, razširitev NEON, ki je hibridna 64/128-bitna SIMD arhitektura, razvita za pospeševanje multimedijskih vsebin, vključno z kodiranjem/dekodiranjem video vsebin, 3D grafike, prepoznavo govora, dekodiranje in sinteza zvoka ter nenazadnje za računalniško obdelavo slik [9].

Zavoljo kompatibilnosti v tej fazi ukazov NEON razširitve ne bomo uporabljali, saj bodo vsi testi delani na napravi iPhone 3G, ki jih ne podpira.

5. INTEGRACIJA OPENCV V IPHONE XCODE PROJEKT

Osnovni jezik iPhone razvojnega okolja je programski jezik Objective-C, ki je subset C-ja, zato ima polno podporo za vključevanje C knjižnic ter prav tako polno podporo za vključevanje C++ izvorne kode. OpenCV tako lahko vključimo v razvojno okolje na več načinov, a predvsem zaradi možnosti, da lahko sami prevajamo in eksperimentiramo z izvorno kodo knjižnice, smo se odločili za postopek, ki je opisan na spletni strani [4] ter uporabo OpenCV 2.0 v iPhone SDK 3.1.2. Po končanem postopku, ki ga v članku sicer ne bomo opisovali podrobneje, se dva nivoja višje od trenutne lokacije ("../") nahajata mapi s prevedeno knjižnico in potrebnimi header datotekami – `opencv_simulator/` ter `opencv_device/`. Obe mapi prekopiramo v korensko mapo našega Xcode projekta.

Sedaj še v Xcode projektu nastavimo poti, kjer se nahajajo statične knjižnice in pripadajoče header datoteke. To storimo v meniju Project > Edit Active Target in v razdelku Linking > Other Linker Flags. S pomočjo ikone v levem spodnjem kotu () izberemo Add Build Setting Condition in dodamo dva pogoja:

- Any iPhone OS Device / Any Architecture
- Any iPhone OS Simulator / Any Architecture

V pogoj "Any iPhone OS Device / Any Architecture" bomo z dvojnimi klikom na polje vrednosti ter nato s klikom na ikono + dodali 4 zastavice in sicer:

- `-lstdc++`
- `-lz`
- `$(SRCROOT)/opencv_device/lib/libcv.a`
- `$(SRCROOT)/opencv_device/lib/libcxcvcore.a`

V pogoj "Any iPhone OS Simulator" dodamo enake štiri zastavice, le pri zadnjih dveh mapo "opencv_device" ustrezno spremenimo na "opencv_simulator". Vse, kar preostane je, da v naš projekt vključimo sledečo header datoteko:

```
#import <opencv/cv.h>
```

6. OPTIMIZACIJA OPENCV ZA IPHONE PLATFORMO

6.1 Optimizacija knjižnice

S pomočjo orodja Instruments, ki je sestavni del okolja Xcode, smo poskusili ugotoviti, kje aplikacija porabi največ časa. Slika 1 prikazuje funkcije, v katerih se aplikacija zadržuje najdlje; po Amdahlovem zakonu bomo največje pohitritve pridobili, če uspemo pohitriti prav te funkcije.

Self Run %	Running %	ms Running	Library	Symbol Name
35,6	35,6	7500,0	RT_OpenCV	▶ ARMcvRunHaarClassifierCascade
35,5	35,5	7480,0	RT_OpenCV	▶ ARMcvHaarDetectObjects
14,8	14,8	3120,0	RT_OpenCV	▶ ARMcvSetImagesForHaarClassifierCascade
2	2	430,0	libSystem.B.dylib	▶ mach_msg_trap
1,7	1,7	370,0	libSystem.B.dylib	▶ Irintl
1,4	1,4	310,0	RT_OpenCV	▶ void cv::pyrDown_<cv::FixPtCast<unsigned char, 8>, cv::NoVec>(cv::Mat const&, cv::Mat&)
1,1	1,1	250,0	CoreGraphics	▶ CGSBlendARGB8888toRGBA8888
0,9	0,9	190,0	RT_OpenCV	▶ icvBGRx2BGR_8u_CnC3R(unsigned char const*, int, unsigned char*, int, CvSize, int, int)
0,7	0,7	160,0	CoreGraphics	▶ CGSFillDRAM8by1
0,7	0,7	150,0	RT_OpenCV	▶ dyld_stub___ledf2vfp
0,6	0,6	140,0	RT_OpenCV	▶ dyld_stub___floatsisfvfp
0,5	0,5	110,0	libSystem.B.dylib	▶ sqrt
0,5	0,5	110,0	RT_OpenCV	▶ dyld_stub___extendsfdf2vfp
0,4	0,4	100,0	RT_OpenCV	▶ dyld_stub___muldf3vfp
0,3	0,3	70,0	libSystem.B.dylib	▶ pthread_setspecific
0,3	0,3	70,0	RT_OpenCV	▶ dyld_stub___mulsf3vfp

Slika 1: prikaz funkcij, kjer se aplikacija zadržuje najdlje

Vidimo, da se aplikacija najdlje zadržuje v funkcijah `cvRunHaarClassifierCascade` ter `cvSetImagesForHaarClassifierCascade`. Ti dve funkciji sta del implementacije Viola-Jones algoritma za detekcijo objektov. Viola-Jones algoritem uporablja t.i. klasifikatorje, ki so pridobljeni s strojnim učenjem na primerih objektov. Pozitivni klasifikatorji s primeri posameznega objekta (obraz, avto, roka, ...) so po učenju skrčeni na enake dimenzije, največkrat 20×20 slikovnih elementov. Ti klasifikatorji so nato uporabljeni na vhodni sliki na področjih enake velikosti, kot so bili uporabljeni pri učenju. Klasifikatorji se lahko enostavno povečujejo in pomanjšujejo, zato se za detekcijo objekta v določeni interesni regiji (Region of Interest – ROI) velikost iskalnega okna povečuje od začetne izbrane velikosti do velikosti celotne regije [5]. Takšen način je zelo uporaben za iskanje objekta neznane velikosti v določeni regiji, hkrati pa je tudi zelo potraten, saj se mora za vse velikosti iskalnega okna in vse pozicije tega okna v interesni regiji izvesti funkcija `cvRunHaarClassifierCascade`. To lahko v praksi pomeni tudi do nekaj tisoč klicev funkcije za posamezno sliko.

Pri pregledu teh funkcij smo ugotovili, da je večina aritmetika detekcije v plavajoči vejici, kar je šibka točka ARMv6 arhitekture. Za optimizacijo same knjižnice smo za zgled vzeli primer [1] in izbrali naslednja področja:

- faktor uteži (weight) v strukturi `CvHidHaarFeature` je bil pretvorjen iz float v integer
- prag tolerance (threshold) v strukturi `CvHidHaarTreeNode` je bil pretvorjen iz float v integer

- izračuni s pragom tolerance, utežmi in faktorjem normalne variance v funkciji `cvRunHaarClassifierCascade` so bili iz aritmetike plavajoče vejice pretvorjeni v celoštevilsko aritmetiko
- za izračun celoštevilčnega kvadratnega korena v funkciji `cvRunHaarClassifierCascade` je bil uporabljen algoritem [10]
- izračun uteži v funkciji `cvSetImagesForHaarClassifierCascade` je bil iz aritmetike plavajoče vejice pretvorjen v celoštevilsko aritmetiko

Na račun teh omejitev bo sicer natančnost prepoznavne slabša, a po vmesnih testih še vedno dovolj dobra za večino aplikacij.

6.2. Optimizacija postopkov

Pri detekciji v realnem času delamo načeloma z video vhomom in se zato lahko praviloma zanašamo na dva faktorja, ki nam pomagata odločitvi optimizacije:

- velika verjetnost je, da pozicija obraza ne bo veliko odstopala glede na prejšnjo lokacijo;
- velika verjetnost je, da velikost obraza ne bo veliko odstopala glede na prejšnjo velikost.







V duhu teh predvidevanj smo postopek priredili tako, da smo uporabili podatke prejšnje detekcije in sicer:

1. interesno področje (Region of Interest - ROI) sedaj izračunamo glede na pozicijo obraza v prejšnji detekciji, zraven pa vključimo še nekaj točk tolerance v vsaki smeri, da dovolimo odstopanja od prejšnje pozicije zaradi premikov. Odvisno od velikosti obraza in časa posamezne detekcije bi to pomenilo, da se objekt v oddaljenosti 3m lahko premika s hitrostjo cca 1m/s, pa bo algoritmu še vedno uspela hitra detekcija s pomočjo predhodne regije;
2. začetna velikost iskalnega okna ni konstantna, temveč jo v vsaki iteraciji izračunamo znova iz velikosti obraza prejšnje detekcije in vključimo nekaj točk za odstopanje.

6.3. Preverjanje vpliva posamezne tehnike (benchmark)

Čas posamezne detekcije je odvisen od več faktorjev – velikosti vhodne slike, velikosti obraza, pozicije obraza ter parametrov detekcije, kot so velikost iskalnega okna, faktor povečevanja iskalnega okna itd. Za potrebe primerjave smo izbrali generični obraz v velikosti cca 100×100 slikovnih elementov na sredini slike. Tabela 2 prikazuje čas, ki je bil potreben za detekcijo tega obraza v vsaki stopnji optimizacije.

Tabela 2: prikaz časovne zahtevnosti detekcije po posameznih stopnjah optimizacije

 <p>Brez optimizacij: 10s</p>	 <p>Enkratna inializacija spremenljivk in deljen cache: 6s</p>	 <p>Uporaba pozicije obraza prejšnje detekcije: 1s</p>
 <p>Uporaba velikosti iskalnega okna prejšnje detekcije: 650ms</p>	 <p>Uporaba aritmetike s celimi števili: 350ms</p>	 <p>Nastavitev parametrov detekcije: 260ms</p>

Vidimo lahko, da sta imeli največji doprinos enkratna inializacija deljenih spremenljivk ter uporaba pozicije obraza prejšnje detekcije. Z drugo tehniko smo uspeli ROI (Region of Interest) zmanjšati s celotnega ekrana na le del originalnega področja, kar je precej zmanjšalo število klicev funkcije `cvRunHaarClassifierCascade`, za katero smo ugotovili, da je funkcija, v kateri aplikacija preživi največ časa. Velik odstotek pohitritve je prispeval tudi prehod na aritmetiko s celimi števili, saj ARMv6 arhitektura v osnovi ne podpira hitrega računanja s plavajočo vejico.

Pod optimizacijo parametrov detekcije je zajeta izbira datoteke s haar-ovimi kaskadami, začetna velikost iskalnega okna, faktor povečave iskalnega okna ter metoda detekcije. Po

empiričnih ocenah zadovoljivih rezultatov so bili izbrani parametri za prikazani primer naslednji:

- datoteka s haar-ovimi kaskadami: haarcascade_frontalface_alt.xml
- začetna velikost iskalnega okna: 30×30 slikovnih elementov
- faktor povečave iskalnega okna: 1.4
- metoda detekcije: CV_HAAR_FIND_BIGGEST_OBJECT in CV_HAAR_DO_ROUGH_SEARCH

Pri izbrani metodi detekcije CV_HAAR_FIND_BIGGEST_OBJECT funkcija najde in vrne le največji objekt v sceni. Z metodo CV_HAAR_DO_ROUGH_SEARCH, ki se uporablja le s prejšnjo metodo iskanja največjega objekta, funkcija ne išče več manjših kandidatov za obraz po tem, ko že enkrat najde primerne kandidata. Ker je bila uporabljena metoda iskanja enega objekta, po mnogih priporočilih nismo uporabil metode CV_HAAR_DO_CANNY_PRUNNING. Le-ta z uporabo detekcije robov ovrže nekatere segmente, kar načeloma pohitri detekcijo, razen v primeru metode iskanja enega objekta.

Z uporabljenimi parametri se sicer zmanjša zanesljivost detekcije, hitrost pa, kot je razvidno iz zgornje tabele, ustrezno poveča.

7. MOŽNOSTI IMPLEMENTACIJE

Detekcija obrazov v realnem času s knjižnico OpenCV je na iPhone-u že skoraj realnost. Kljub temu pa bo zaradi trenutnega stanja – bolj groba detekcija, počasnejše delovanje, kamera le na zadnjem delu telefona – verjetno doma bolj v zabavnih aplikacijah. Praktični primeri uporabe bi lahko zajemali fotoaparata z detekcijo nasmeškov, prepoznavna gest, portretni način fotoaparata z avto fokusom na obraz (če Apple objavi API za avtofokus), aplikacije z nadgrajeno resničnostjo (augmented reality) in drugi. Če bo Apple kdaj dodal kamero na sprednji del telefona, bi bili lahko primer uporabe stabilizacija slike pri video konferenci, nadzor aplikacije s premiki glave in podobno.

Celotna koda primera uporabe sledenja obrazu, ki nadgradi resničnost z zelenim kvadratom okoli osebe, se nahaja na spletnem naslovu:

http://gameteam.fri.uni-lj.si/code/rt_opencv_iphone.zip

8. SKLEP

Z uporabo relativno enostavnih tehnik optimizacije smo časovno zahtevnost detekcije obrazov na iPhone 3G platformi zmanjšali za faktor skoraj 40. Prvotnemu cilju pa smo se s tem vseeno le približali, saj bi si za detekcijo v realnem času želel vsaj 10 sličic na sekundo, kar pomeni zmanjšanje časa na 100 ms.

Kot možnost nadaljne optimizacije se ponuja uporaba SIMD razširitve ARMv7 arhitekture, imenovana NEON, ki podpira do izvajanje 16 ukazov hkrati. Takšna

optimizacija bi zahtevala specifično vektorizacijo parametrov, zato presega okvirje tega članka, daje pa možnost za nove izzive. V izvorni kodi verzije 2.0 OpenCV knjižnice je več funkcij že optimiziranih za Intelove SIMD razširitve SSE1 do SSE4, zato bi lahko za navdih in primer lahko uporabili kar to implementacijo. NEON razširitev pa žal ni prisotna na ARMv6 arhitekturi, kar v praksi pomeni le optimizacijo na iPhone 3GS. iPhone 2G in 3G sicer vsebujeta Vector Floating Point (VFP) enoto, za katero obstaja knjižnica `vfpmathlibrary` [7], ki bi jo bilo mogoče uporabiti za paralelizacijo izračunov in dodatne pohitritve.

LITERATURA

1. B. Batagelj, F. Solina, P. Peer (2004), 15 seconds of fame - an interactive, computer-vision based art installation, ACM International Conference on Multimedia, pp. 764-765, New York, NY, USA
2. <http://www.morethantechnical.com/2009/08/09/near-realtime-face-detection-on-the-iphone-w-opencv-port-wcodevideo>
Near realtime face detection on the iPhone w/ OpenCV port
3. <http://www.computer-vision-software.com/blog/2009/06/fastfurious-face-detection-with-opencv>
Fast & Furious face detection with OpenCV
4. <http://niw.at/articles/2009/03/14/using-opencv-on-iphone/en>
Using OpenCV on iPhone
5. <http://opencv.willowgarage.com/wiki/FullOpenCVWiki>
FullOpenCVWiki
6. G. Bradski, A. Kaehler (2008), Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly
7. <http://code.google.com/p/vfpmathlibrary>
VFP based math library for the iPhone / iPod touch
8. <http://www.anandtech.com/printarticle.aspx?i=3579>
iPhone 3GS Hardware Exposed & Analyzed
9. <http://www.arm.com/products/CPUs/architecture.html>
ARM Processor Instruction Set Architecture
10. http://www.codecodex.com/wiki/index.php?title=Calculate_an_integer_square_root
Calculate an integer square root

KDO SMO?

Slovenski podjetniški sklad (v nadaljevanju Sklad) je javna finančna institucija Republike Slovenije, ustanovljena z namenom izboljšanja dostopa do finančnih sredstev za različne razvojno - poslovne projekte mikro, malih in srednje velikih podjetij (v nadaljevanju MSP) vključno s finančnimi sredstvi za zagon podjetij in mikrofinanciranje v RS Sloveniji.

KAJ PONUJAMO?

Sklad ponuja ustrezne finančne instrumente v okviru treh finančnih linij in sicer:

1. Garancijsko-kreditna linija za MSP

- P1 - Klasične garancije
- P1 TP - Garancije za tehnološke projekte
- P3 - Pogarancije za regionalne garancijske sheme

Produkta P1 in P3 uživata ugodnosti jamstva, ki je bilo izdano v okviru Okvirnega programa za konkurenčnost in inovativnost Evropske skupnosti.

2. Subvencijska linija za MSP

- P2 – subvencije za zagon podjetij v subjektih inovativnega okolja

3. Linija za lastniško financiranje za MSP

- LF - kapitalske naložbe republike Slovenije v zasebne družbe tveganega kapitala - DTK

Za pravočasne informacije o razpisih Sklada in drugih aktivnosti, se prijavite na e-novice na spletni strani: www.podjetniskisklad.si

VEČ INFORMACIJ: Slovenski podjetniški sklad, Trubarjeva 11, 2000 Maribor,
tel.: 02/234 12 60

