



ROSUS 2016
Računalniška obdelava slik in njena
uporaba v Sloveniji 2016
Zbornik 11. strokovne konference

Urednik
Božidar Potočnik

Maribor, 17. marec 2016

Zbornik prispevkov strokovne konference
ROSUS 2016 Računalniška obdelava slik in njena uporaba v Sloveniji 2016
Maribor, 17. marec 2016.

Elektronska izdaja

CIP - Kataložni zapis o publikaciji
Univerzitetna knjižnica Maribor

004.932(497.4)(082)(0.034.2)

STROKOVNA konferenca ROSUS (11 ; 2016 ; Maribor)

ROSUS 2016 [Elektronski vir] : računalniška
obdelava slik in njena uporaba v Sloveniji 2016 :
zbornik 11. strokovne konference, Maribor, 17.
marec 2016 / urednik Božidar Potočnik. - El.
zbornik. - Maribor : Fakulteta za elektrotehniko,
računalništvo in informatiko, Inštitut za
računalništvo, 2016

Način dostopa (URL): <http://rosus.feri.um.si/rosus2016/ZbornikROSUS2016.pdf>

ISBN 978-961-248-506-1

l. Potočnik, Božidar

COBISS.SI-ID 86069761

ISBN 978-961-248-506-1



Programski odbor ROSUS 2016:

dr. Damjan Zazula
dr. Boris Cigale
dr. Marjan Mernik
dr. Peter Peer
dr. Tomaž Tomažič
dr. Matjaž Colnarič
dr. Slobodan Ribarič
dr. Vojko Flis
dr. Saša Divjak
dr. Stanislav Kovačič

dr. Zdravko Kačič
dr. Aleš Leonardis
dr. Simon Dobrišek
dr. Mitja Lenič
dr. Franc Solina
dr. Andrej Šoštarič
dr. Franjo Pernuš
dr. Aleš Holobar
dr. Božidar Potočnik

Organizacijski odbor 2016:

dr. Damjan Zazula
dr. Danilo Korže
dr. Boris Cigale
Jurij Munda
Vojko Glaser
Gašper Sedej
Martin Šavc
dr. Božidar Potočnik

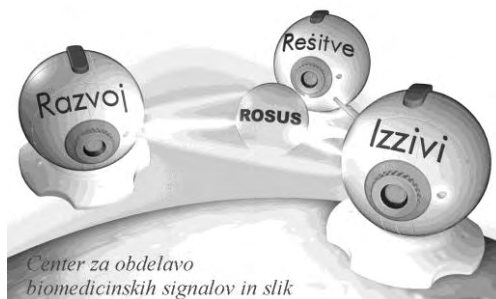
Spoštovani!

Po desetih uspešnih konferencah ROSUS 2006–2015 želimo tudi s pojubilejno konferenco ROSUS 2016 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 Javnemu skladu Republike Slovenije za podjetništvo, ki je v okviru konference ROSUS 2016 predstavil zanimive finančne instrumente za spodbujanje prenosa tehnoloških rešitev v podjetniško sfero.

dr. Božidar Potočnik
predsednik konference
ROSUS 2016



POKROVITELJI



ROSUS 2016
<http://rosus.feri.um.si/rosus2016/>

KAZALO

| | |
|--|-----------|
| VABLJENA PREDAVANJA | 1 |
| Jure Skvarč , Kolektor Orodjarna, Idrija <i>Strojni vid – ključna tehnologija Industrije 4.0</i> | 3 |
| Jernej Bule , KLS Ljubno, Ljubno ob Savinji <i>Kontrola ustreznosti obročev z avtomatskimi merilnimi stroji</i> | 9 |
| Domen Mongus , FERI, Maribor <i>Izločanje informacij iz 3D slik površja</i> | 12 |
| Rok Kreslin, Rok Irt, Luka Kumelj, Mitja Pugelj , Comland, Ljubljana <i>Izboljšanje uporabniške izkušnje spletnih kazinojev s prepoznavo obraznih izrazov</i> | 22 |
| INDUSTRIJSKE APLIKACIJE | 29 |
| Martin Šavc, Božidar Potočnik , FERI, Maribor <i>Barvna vztrajnost na osnovi ocenjevanja povprečne reflektivnosti scene</i> | 31 |
| Matej Babič , IJS, Ljubljana <i>Nov algoritem grafov vidljivosti v 3D prostoru in njegova aplikacija za analizo slik</i> | 39 |
| Jagdish Lal Raheja, Arun Rajsekhar Gadula , CEERI, Pilani, Indija <i>Real Time Pellet Size Distribution using a DSP Embedded Platform</i> | 45 |
| MEDICINSKE IN BIOMEDICINSKE APLIKACIJE | 55 |
| Dejan Knez, Boštjan Likar, Franjo Pernuš, Tomaž Vrtovec , FE, Ljubljana <i>Avtomatska metoda načrtovanja velikosti in trajektorije pedikularnih vijakov</i> | 57 |
| DRUGO | 69 |
| Milan Čoh, Peter Planjšek , FŠ, Ljubljana <i>Diagnostika hitrosti z laserskim merilnikom LDM-301</i> | 71 |
| Darijan Marčetić, Branko Samaržija, Martin Soldić, Slobodan Ribarić , FER, Zagreb, Hrvaška <i>Face de-identification for privacy protection in surveillance systems</i> | 79 |

| | |
|---|-----------|
| ŠTUDENTSKE APLIKACIJE | 87 |
| Leon Ropoša, Borut Batagelj, Franc Solina, FRI, Ljubljana <i>Določanje odziva na zaužito hrano z metodami za prepoznavo obraznega izraza.....</i> | <i>89</i> |
| Jaka Konda, Peter Peer, FRI, Ljubljana <i>Prepoznavna poslovnih vizitk in klasifikacija besedila.....</i> | <i>97</i> |

VABLJENA PREDAVANJA

STROJNI VID – KLJUČNA TEHNOLOGIJA INDUSTRIJE 4.0

Jure Skvarč

Kolektor Orodjarna d.o.o., PC VISION

E-pošta: jure.skvarc@kolektor.com

URL: <http://www.kolektorvision.com>

POVZETEK: *Živimo v času, ko izginjajo meje med fizičnim in kiber svetom. Z uvajanjem interneta stvari so fizične naprave vse bolj nevidno vgrajene v informacijsko mrežo – priča smo četrti industrijski revoluciji. Pri tem procesu je Zahodna Evropa kot del tradicionalnih industrijskih ekonomij v zadnjih 20 letih izgubila znaten delež industrijske dodane vrednosti [3]. Ker je industrija ključni element v vrednostni verigi, je ključno vprašanje, kako nadoknaditi zamujeno? Na ta izziv Evropa odgovarja z Industrijo 4.0 in obsežnim investicijskim programom v prihodnjih letih. Pri tem bo imelo vizualno računalništvo ključno, povezovalno vlogo. Vizualno računalništvo [8] je širši pojem, ki združuje računalniški/strojni vid, računalniško grafiko, obogateno in navidezno resničnost ter vizualizacijo. V prispevku so podrobneje predstavljeni strojni vid v industrijskem okolju, trendi razvoja in potencialni novi primeri uporabe.*

1. UVOD

Industrija ima osrednjo vlogo v evropskem gospodarstvu, saj prispeva 15 % delež dodane vrednosti, ustvari 80 % inovacij in predstavlja 75 % delež izvoza. Če k temu dodamo še storitve, povezane z industrijo, je industrija motor evropskega socialnega gospodarstva. Zaradi upadanja konkurenčnosti spričo novih akterjev na trgu, zlasti iz Azije, se je v zadnjih 10 letih znižalo število delovnih mest na sicer uveljavljenih trgih, kot so Velika Britanija, Francija in Nemčija [4].

Razvoj v evropskih državah ne poteka enako hitro. Medtem ko Nemčija in Vzhodna Evropa še naprej povečujeta svoj delež na industrijskem trgu, se druge članice EU soočajo z deindustrializacijo. Nadaljevanje takšnega trenda bi povzročilo nadaljnjo slabitev Evrope in posledično dodatno izgubo delovnih mest in znanja.

Po avtomatizaciji, elektrifikaciji in digitalizaciji industrije zaznamuje uvajanje interneta stvari v tovarne prihod četrte industrijske revolucije [4]. Internet je drastično spremenil naša življenja. Močno vpliva na industrijo in revolucionarno spreminja način organizacije proizvodnje. V scenariju, imenovanem Industrija 4.0, je realna proizvodnja integrirana z virtualno realnostjo, kar omogoča izjemno fleksibilnost proizvodnje. Tako

oblikovano digitalno omrežje omogoča uporabnikom prilagodljivo proizvodnjo in razvoj novih poslovnih modelov.

Globalna konkurenca postaja vse ostrejša, kar predstavlja dodaten izziv proizvajalcem. Le tisti, ki bodo med vodilnimi v produktivnosti in obvladovanju stroškov, bodo sposobni preživeti. Tehnologije, vključene v strojni vid, so komplementarne z internetom stvari in industrijskim internetom, vendar bolj usmerjene v vidike bližje avtomatizaciji, povezljivosti, povsod navzočim informacijam, kiber varnosti, inteligentni robotiki, upravljanju življenjskega cikla produktov (angl.: PLM), semantičnim tehnologijam in industrijskim masovnim podatkom. Strojni vid je tehnologija, ki ponuja rešitve za številne izzive v prihodnosti [6].

Že danes sistemi strojnega vida niso več preprosti sistemi, ki le kontrolirajo izdelke, temveč lahko hkrati tudi prepoznavajo trende v proizvodnem procesu, ustvarjajo podatkovne baze, generirajo statistične podatke o procesu in podobno. Tako se nadzor kakovosti razvija v optimizacijo proizvodnje, ki omogoča dinamično odzivanje na nepravilnosti v začetnih in vmesnih fazah proizvodnega procesa.

Tehnologija strojnega vida ima številne pozitivne učinke tudi na mnogo drugih področjih izven klasične industrijske proizvodnje. Izboljšana kakovost, večja zanesljivost, večja varnost in stroškovna učinkovitost so dodane vrednosti, ki so ključnega pomena tudi v neproizvodnih okoljih.

2. INDUSTRIJA 4.0

V preteklosti smo bili priča trem industrijskim revolucijam, ki bi jih lahko opisali kot nenadne skoke v industrijskih procesih, ki so povzročili občutno višjo produktivnost. Prva je izboljšala učinkovitost z uporabo vodne energije, povečano uporabo parne moči in razvojem strojev orodij. Druga je vpeljala uporabo električnih strojev in množično proizvodnjo (montažne linije), tretja pa je pospešeno vpeljala avtomatizacijo z uporabo elektronike in informacijskih tehnologij.

Industrija 4.0 ali četrta industrijska revolucija je skupni izraz, ki označuje tehnologije in koncepte organizacije vrednostne verige, ki skupaj povezuje kiberfizične sisteme, internet stvari (angl.: Internet of Things – IoT) in internet storitev (angl.: Internet of Services – IoS).

V modularno strukturirani pametni tovarni Industrije 4.0 kiberfizične naprave nadzirajo fizične procese, ustvarjajo virtualno kopijo fizičnega sveta in izvajajo decentralizirano odločanje. Pametna tovarna 4.0 je kot socialno omrežje – preko interneta stvari kiberfizični sistemi komunicirajo in sodelujejo med seboj in z ljudmi v realnem času [2]. Interne in medorganizacijske storitve so prek interneta storitev na voljo vsem udeležencem v vrednostni verigi.

Proizvodnja bo vse bolj uporabljala nove tehnologije, kot so senzorji, 3D-tiskalniki in roboti naslednje generacije. Proizvodni procesi bodo modularni, dobro usklajeni, fleksibilni, z možnostjo prilagajanja v realnem času.

Vse to bo spremenilo tudi miselnost uporabnikov ne le glede njihovih zahtev, temveč tudi glede njihove predstave o proizvodnem procesu. Stroji ne bodo več slepi izvajalci preprostih opravil, temveč bodo del nekega celovitega procesa, v katerem komunicirajo tudi z drugimi stroji in napravami z namenom čim učinkovitejšega izpolnjevanja celovite

naloge. Uporabnike bodo namreč vedno manj zanimale tehnologije, saj bodo usmerjeni predvsem k temu, da dobijo to, kar želijo in pričakujejo.

2. STROJNI VID

Strojni vid so tehnologije in metode, ki na osnovi slikovne informacije omogočajo avtomatsko kontrolo in analizo v aplikacijah, kot so optična kontrola izdelkov, nadzor in avtomatizacija procesov.

Strojni vid ni industrija "per se", ampak gre za integracijo tehnologij in izdelkov, ki zagotavljajo storitve in rešitve drugim industrijam in neindustrijskim primerom uporabe.

Razvoj področja je tesno povezan z razvojem računalniških sistemov in dvigom procesnih zmogljivosti računalnikov. Področje strojnega vida se je pričelo hitro razvijati skupaj s sistemi za avtomatizacijo proizvodnje, s podporo elektronike in IT. Prvo praktično uporabo strojnega vida so leta 1968 demonstrirali na MIT v Laboratoriju za umetno inteligenco. Strojni vid je bil uporabljen za prepoznavo preprostih objektov in krmiljenje mehanske roke.

Sistemi strojnega vida so vsestranski. Uporabljajo se za preverjanje kakovosti, vodenje naprav, nadzor procesov, identifikacijo sestavin, branje kod in pridobivanje podatkov za uporabo v optimizaciji proizvodnje.

Motivatorji razvoja strojnega vida v današnjem proizvodnem okolju so izboljšanje kakovosti, znižanje stroškov, povečanje obsega proizvodnje, večja kompleksnost izdelkov in krajši taktni časi.

Potencial strojnega vida so prepoznale tudi velike korporacije. Google na primer pod svojim okriljem združuje podjetja s kompetencami na področju robotike in strojnega vida. Razvijajo in preizkušajo tudi humanoidne robote, ki se samostojno gibljejo v naravnem okolju.

Končni cilj strojnega vida je modelirati, posnemati in, še bolje, preseči zmogljivosti človeškega vida s pomočjo računalniške programske opreme in strojne opreme na različnih ravneh.

2.1 Vloga strojnega vida v Industriji 4.0

Predstavljajmo si svet, v katerem vse težavne in ponavljajoče se naloge opravljajo avtonomni stroji. Svet, v katerem stroji samodejno pobirajo poljske pridelke, jih pregledujejo, razvrščajo, predelajo, pakirajo in dostavijo, brez posredovanja človeka. Čeprav se v tem trenutku omenjeno zdi težko predstavljivo, številni proizvajalci že uporabljajo avtomatizirane stroje, ki opravljajo vsaj del naštetih funkcij. Pri teh strojih so sistemi strojnega vida "zaznavala", ki v najboljši meri posnemajo sposobnosti človeškega vida. Z namenom vedno večje učinkovitosti in konkurenčnosti bodo sistemi strojnega vida uporabljeni v sistemih za avtomatizacijo celotnih proizvodnih procesov, ki bodo odpravljali potrebo po delovni sili in napake, ki iz tega izhajajo [7].

Obstoječi sistemi strojnega vida predstavljajo le majhen del strojnega vida na trgu, ki se bo razširil na vsa področja življenja, od obiranja poljskih pridelkov, sortiranja kamnin, do dostave izdelkov od proizvodnje do potrošnika.

2.2 Trendi na področju strojnega vida

Strojni vid je tehnologija, ki se hitro razvija, vendar večinoma sledi eni inovacijski krivulji. Nenehno se izboljšujejo lastnosti komponent in zmogljivosti algoritmov.

2.2.1 Optični in električni gradniki

Na področju razvoja komponent so opazni trendi:

- *Miniaturizacija*: zmanjševanje velikosti slikovnih elementov optičnih senzorjev in s tem možnost miniaturizacije kamer.
- *Nadomeščanje senzorjev CCD s senzorji CMOS*: cenejši, nižja poraba, večje število slikovnih elementov – višja ločljivost, napredno procesiranje, višje hitrosti.
- *3D-senzorji*: razvoj zmogljivejših kamer za 3D-zaznavanje prostora (stereo, triangulacija, čas preleta).
- *Pametne kamere in vgrajeni sistemi*: »pametni sistemi« bodo postali še bolj »pametni«.
- *Barve*: Povečana uporaba barvnih senzorjev in senzorjev, ki zaznavajo nevidno elektromagnetno valovanje (UV, NIR).
- *Optične leče*: cenejše, kakovostnejše leče. Prilagodljive, deformabilne leče – možnost hitrega spreminjanja optičnih parametrov.
- *Vmesniki za prenos podatkov*: hitre kamere in kamere z več slikovnimi elementi potrebujejo hitrejšo povezavo med kamero in procesno enoto (USB3 Vision, CoaXPress). [5]
- *Procesorji*: Neuskklajenost med zmogljivostjo strojne opreme in potrebami programskih algoritmov ovira realno implementacijo algoritmov. Strojna oprema mora omogočati izvajanje algoritmov v realnem času. Sodobna, široko dostopna procesna oprema je ravno dosegla nivo, ki omogoča implementacijo nekaterih zahtevnejših algoritmov v industrijskem okolju v realnem času (prepoznavna 3D-objektov v oblaku točk – nekaj sekund).
- *Standardizacija*: Standardizacija vmesnikov omogoča lažje vključevanje posameznih komponent strojnega vida v splošno uporaben sistem.

2.2.2 Programska oprema in uporabniška izkušnja

- *Algoritmi*: Velik napredek v procesni zmogljivosti računalnikov, ki so prosto dostopni na trgu, je spodbudil nadaljnji razvoj računsko potratnih algoritmov, kot so 3D-modeliranje in prepoznavanje, rekonstrukcija iz analize gibanja, analiza gibanja in sledenje, prepoznavanje vzorcev, statistične analize, umetna inteligenca. Pri večini obstoječih rešitev strojnega vida je postopek učenja (razvrščanje vzorcev na osnovi značilk) še vedno odvisen od človeka. Z uporabo mreže velikega števila povezanih procesorjev pa lahko ustvarimo nevronske mreže, ki je sposobna s

postopki globokega učenja (angl.: deep learning) z obdelavo več milijonov podatkov (slik) samodejno opraviti postopek učenja.

- *Uporabniški vmesnik*: Uporabniški vmesnik je pomemben del sistema strojnega vida. Desetletje nazaj je napako zaradi neželenega premika kamere na stroju lahko odpravil le izkušen sistemski inženir dobavitelja. Danes programska oprema sistemov strojnega vida izkorišča potencial zmogljivih procesnih enot in vse kompleksne funkcije sistema skriva za preprostim uporabniškim vmesnikom. Upravljanje s sistemi strojnega vida v prihodnosti bo preprosto in intuitivno.

2.2.3 Novi in napredni primeri uporabe

- *3D-vid in vizualno vodenje robotov*: Z zmogljivimi sodobnimi procesorji in 3D-senzorji postaja uporaba tridimenzionalnega strojnega vida realnost. Proizvajalci v avtomobilski industriji npr. pospešeno uvajajo tehnologije 3D-vida v aplikacijah, ki segajo od montaže avtomobilskih vrat in pokrovov, sestave avtomobilskih motorjev, sestave LED-zaslonov do pregledovanja pnevmatik (tekalni profil, oznake na boku). Izven avtomobilске industrije je potencial za 3D-vid npr. v pregledovanju polprevodniških rezin, pregledovanju hrane in farmacevtskih izdelkov ter na področju avtonomnih vozil. Nove, zmogljivejše kamere in 3D-senzorji, boljše programske knjižnice in nadaljnja rast procesne zmogljivosti računalnikov bodo omogočili širšo uvedbo tehnologije in nadaljno rast v prihodnosti.
- *Zaznavanje in zmožnost akcije*: Robotiko pogosto opisujemo kot inteligentno povezavo med zaznavanjem in akcijo. Akcijsko funkcijo omogočajo robotski aktuatorji, številni senzori pa omogočajo zaznavanje. Računalniki zagotavljajo okvir za integracijo/povezavo in inteligenco, ki je potrebna za koordiniranje v smiselno interpretacijo informacij in krmiljenje akcije. Sistemi strojnega vida so med najbolj zmogljivimi mehanizmi zaznavanja. Vključujejo pridobivanje, karakterizacijo in interpretacijo slikovnih informacij z namenom prepoznave in popisa objektov v okolju.
- *Prepoznavanje objektov in navigacija*: Primeri uporabe vključujejo mobilne robote (kopno, voda, zrak) z integriranimi sistemi strojnega vida, navigacijske sisteme za robote v gospodinjstvu, pametne kamere za sledenje pozicije glave in smeri pogleda oči, učljive sisteme za prepoznavanje objektov, sledenje človeškim gestam, interaktivno oglaševanje in podobno.
- *Nekontrolirana okolja, neindustrijski primeri uporabe*: Sistemi strojnega vida se trenutno uporabljajo pretežno v kontroliranih okoljih, kjer je možno lokalno nadzorovanje svetlobnih pogojev [1]. Vizualno vodeni robotski sistemi, ki bi izvajali npr. kmetijska opravila na prostem, kot so sajenje, gojenje in obiranje pridelkov, morajo delovati v nekontroliranih okoljih, kjer se svetlobne in vremenske razmere drastično spreminjajo.

7. ZAKLJUČEK

Implementacija principov Industrije 4.0 je najbolj opazna pri nemških podjetjih, ki že postopno spreminjajo strukturo vrednostne verige. Trenutno je največji poudarek na

pridobivanju informacij, povezljivosti, diagnostiki in ustvarjanju obsežnih baz podatkov (primer Siemens: MindSphere – Siemens Cloud for Industry). Pri tem sta pomembna standardizacija komunikacijskih protokolov in zagotavljanje varnosti na vseh nivojih. Virtualizacija (ustvarjanje virtualne slike realnega sveta) se že uporablja pri razvojnih orodjih za načrtovanje proizvodnih procesov, vizualizaciji in kot pomoč pri izobraževanju.

Primarna vloga sistemov strojnega vida je še vedno zagotavljanje kakovosti in avtomatizacija (del koncepta »Zero Defect Production«). Ključni problemi, s katerimi se morajo spoprijeti proizvajalci optičnih sistemov, so vezani na slabo definirana okolja delovanja in raznolikost izdelkov, ki so predmet optične kontrole. Kljub pričakovanemu razvoju v bližnji prihodnosti bo potrebno rešitve iskati v kombinaciji z drugimi tehnološkimi rešitvami.

Z napredkom tehnologije strojnega vida in razvojem ekosistema Industrije 4.0 bodo ti sistemi prehajali na nova področja, kjer do sedaj niso bili prisotni. Strojni vid bodo "oči" in ključna tehnologija Industrije 4.0.

LITERATURA

- [1] Shivleela, S. in ostali: Recent trends in machine vision applications, Proceedings of the 7th National Conference; INDIACOM, New Delhi, 2013
- [2] Morris, H.: What will the future look like under Industry 4.0 and digital transformation in the healthcare space, Stuttgart, MedTec Europe, 2015, http://www.medteceurope.com/sites/default/files/6.morris_hosseini_-_presentation_to_share_0.pdf [preneseno 17. 11. 2015]
- [3] Bloching, B in ostali: The Digital transformation of Industry, Munchen, Roland Berger Strategy Consultants GmbH, 2015, str. 1–52, http://www.rolandberger.com/media/pdf/Roland_Berger_digital_transformation_of_industry_20150315.pdf [preneseno 17. 11. 2015]
- [4] Dujin, A. in ostali: Think Act Beyond Mainstream, Munchen, Roland Berger Strategy Consultants GmbH, 2014, str. 1–24, https://www.rolandberger.com/media/pdf/Roland_Berger_TAB_Industry_4_0_2014_0403.pdf, [preneseno 17. 11. 2015]
- [5] Gretchen Alper, Top 5 predictions for the future in machine vision, 2015, <http://info.adimec.com/blogposts/top-5-predictions-for-the-future-in-machine-vision>, [preneseno 15. 11. 2015]
- [6] VDMA, Key Technology for Automation Solutions – Machine Vision 2015/16, http://ibv.vdma.org/documents/256550/0/IBV_2014_GB_LR.pdf/9874fc33-254a-4125-b22f-068cc70cec18, [preneseno 15. 11. 2015]
- [7] Andrew Wilson, Machine Vision: The past, the present and the future, Vision System Design, 2015, <http://www.vision-systems.com/articles/2015/06/machine-vision-and-image-processing-past-present-and-future.html>, [preneseno 17. 11. 2015]
- [8] Visual computing, https://en.wikipedia.org/wiki/Visual_computing

KONTROLA USTREZNOSTI OBROČEV Z AVTOMATSKIMI MERILNIMI STROJI

Jernej Bule

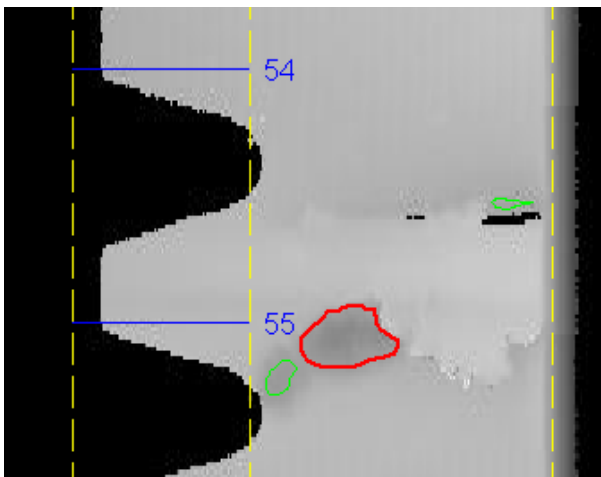
KLS Ljubno d.o.o.,
Ljubno ob Savinji
E-pošta: jernej.bule@kls.si

POVZETEK: *KLS Ljubno je eno izmed vodilnih globalnih podjetij na področju proizvodnje obročev za vztrajnike motorjev. V KLS načrtujejo in razvijajo proizvodne procese z veliko stopnjo zanesljivosti. Posledično imajo zelo nizek delež dobavljenih proizvodov, ki so slabi, saj le te odkrivajo že med samo izdelavo. Trenutno pri odjemalcih dosegajo »6 sigma« pri PPM in manj kot dve reklamaciji na milijon dobavljenih proizvodov. Zaradi navedenih dejstev je KLS v svetu priznan kot najbolj kakovosten proizvajalec obročev za vztrajnike motorjev.*

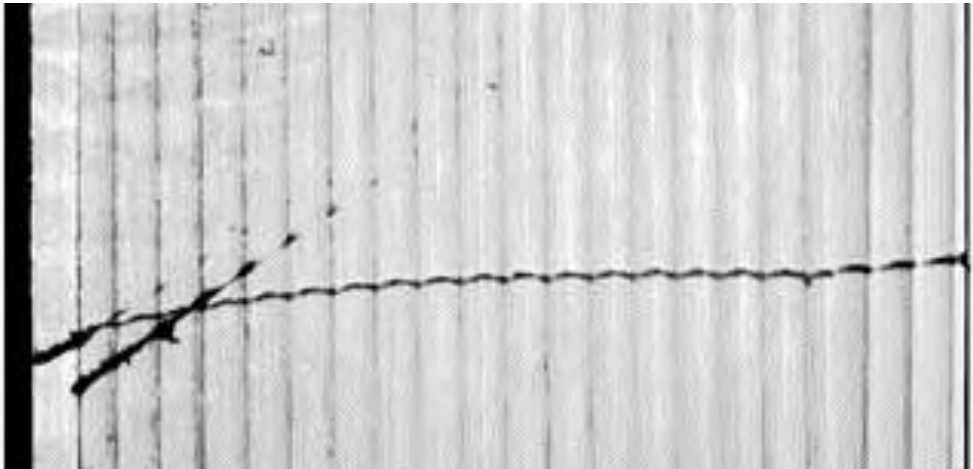
Za doseganje takšnih kazalnikov, v podjetju KLS Ljubno izvajajo 100 % kontrolo vsakega obroča, ki ga odpremo. Manjši delež proizvodov je še vedno pregledan ročno, večino pa avtomatsko pregledajo stroji, brez posredovanja delavca. Avtomatsko optično kontrolo izvajajo s pomočjo namenskih merilnih strojev, ki so izredno kompleksni, natančni, učinkoviti. Na predavanju bo predstavljen koncept delovanja merilnih strojev, uporabljene metode, principi merjenja ter problemi, s katerimi se vsakodnevno srečujejo.



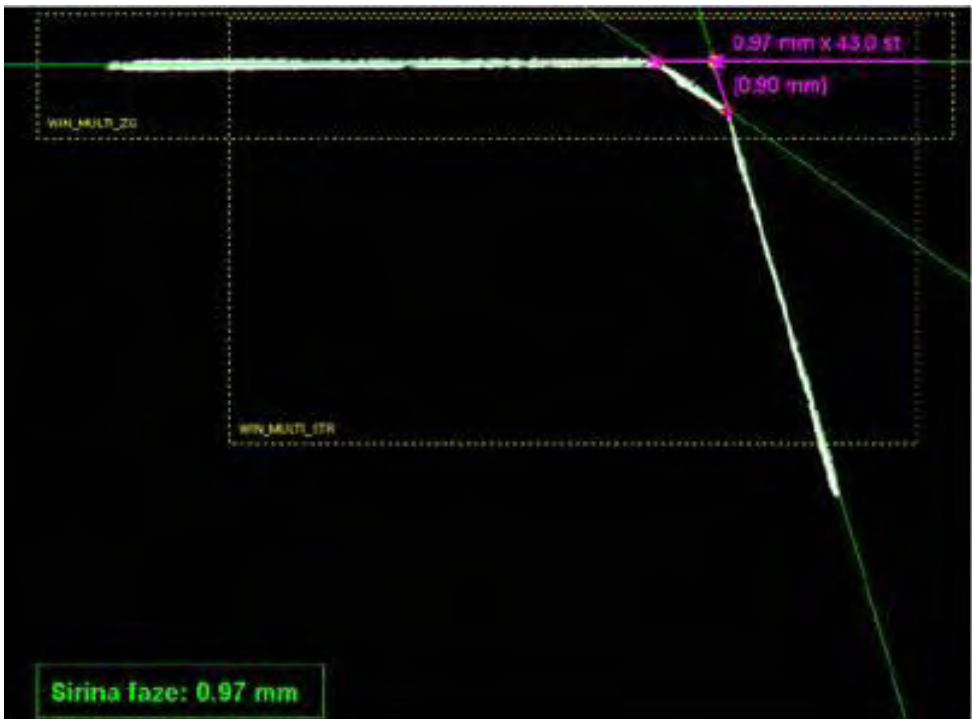
Slika 1: Sestavljena globinska slika zobatega obroča.



Slika 2: Detekcija vtiskov na ozobju in boku zobatega obroča.



Slika 3: Detekcija razpok na površini izdelka.



Slika 4: Meritev širine, višine in kota struženega dela.

IZLOČANJE INFORMACIJ IZ 3D SLIK POVRŠJA

Domen Mongus

Laboratorij za geometrijsko modeliranje in algoritme multimedijev
Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v
Mariboru

E-pošta: domen.mongus@um.si

URL: <http://gemma.feri.um.si/>

POVZETEK: *V letu 2015 smo izvedli nacionalni projekt zajema površja Slovenije. Pridobljene 3D slike zelo visoke ločljivosti so dosegljive na spletnih straneh Agencije Republike Slovenije za Okolje. Brezplačen dostop do teh podatkov pa odpira številne nove možnosti pri izvajanju prostorskih študij in predstavlja edinstveno priložnost za razvoj naprednih storitev temelječih na lokaciji. V tem članku predstavljamo nekatere temeljne metode, ki smo jih razvili in uporabili za obdelavo 3D podatkov površja Slovenije kot del celovite programske infrastrukture gLiDAR. Ta vključuje razpoznavo terena, vegetacije ter stavb in omogoča izgradnjo digitalnih modelov reliefa in površja. Predstavimo tudi naprednejše pristope, ki jih med drugim uporabljamo za razpoznavo posameznih dreves.*

1. UVOD

V zadnjem desetletju smo priča silovitemu napredku optičnih tehnologij daljinskega zaznavanja. Te omogočajo detajlno opazovanje površja Zemlje ter odkrivanje vzorcev in karakteristik na velikih geografskih področjih. Zaradi svoje zanesljivosti, visoke gostote zajema in natančnosti, je vodilno vlogo na tem področju danes prevzela tehnologija laserskega zajema LiDAR (ang. Light Detection and Ranging). Laserske merilnike v ta namen namestimo na vozila, letala ali satelite, od koder opravljajo natančne meritve karakteristik laserskega žarkov, ki se odbijejo od oddaljenih objektov. Merilniki določijo razdaljo do oddaljenega objekta glede na meritve časovne razlike med oddajo laserskega impulza in zaznavo njegovega odboja. Položaj merilnika je pri tem določen s pomočjo sistema referenčnega GPS (ang. global positioning system), podporni inercialni merilni sistem (ang. inertial measurement unit, IMU) pa meri njegov zasuk, naklon in usmerjenost. Glede na te informacije lahko natančno določimo tudi globalen položaja izmerjenih točk ter jih tako georeferenciramo. Ker so merilniki zmožni zaznati več hkratnih odbojev posameznega laserskega impulza, lahko prodrejo skozi reže v vegetaciji in zaznavo teren pod njimi [1]. Na ta način so merilniki zmožni izvesti več kot 200.000 meritev v sekundi ter zajeti več kot 40 točk iz kvadratnega metra zemeljskega površja [2]. V letu 2015 se je končal nacionalni projekt zajema 3D površja Slovenije s pomočjo tehnologije LiDAR. Gostota pridobljenega oblaka 3D točk se giblje med 5 in 10 točk na kvadratni meter, njegova točnost pa sega pod 15 centimetrov, tako v

vodoravni kot tudi navpični smeri. Podatki, ki so brezplačno dostopni na spletnih straneh Agencije Republike Slovenije za Okolje, odpirajo številne nove možnosti pri izvajanju prostorskih študij in predstavlja resnično priložnost za razvoj naprednih storitev temelječih na lokaciji.

V tem članku predstavljamo nekatere napredne pristope obdelave podatkov LiDAR in pridobivanja znanja iz njih. V poglavju 2 predstavimo matematično ogrodje, ki nam služi za robustno in natančno razpoznavo objektov. Slednje podrobneje opišemo v poglavju 3. Poglavje 4 predstavlja sklep tega dela.

2. PRINCIPI OBDELAVE PODATKOV LIDAR

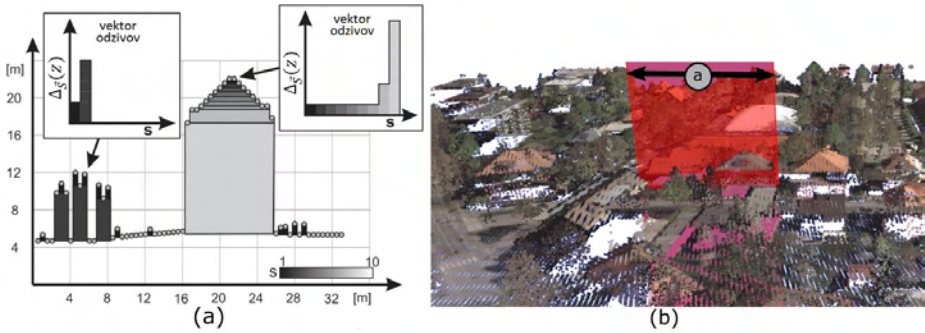
Obdelava podatkov LiDAR se nekoliko razlikuje od tradicionalnih postopkov obdelave slik. Glavni razlog za to izhaja iz dejstva, da tako zajeti oblaki točk niso topološko strukturirani. Vzpostavitev topologije je zato običajno prvi korak procesa razpoznavanja vzorcev, ki hkrati pogojuje tudi implementacijo vseh nadaljnjih metod. Danes v ta namen najpogosteje uporabljamo:

- trikotniško mrežo [3,4,5,6], ustvarjeno z Delaunayevo triangulacijo [7] točk LiDAR, ki jih v ta namen projiciramo na xy-ravnino,
- graf k-najbližjih sosedov [7,8], kjer vsako točko povežemo z njej najbližjih k točk,
- mrežo enakovelikih celic (ang. grid), ki jo izdelamo s podvzorčenjem točk [9,10,11,12] ter
- vokselni prostor, pri čemer vsaki vokselni celici pripišemo seznam vsebujočih se točk [13,14]

Ker želimo v tem članku predstaviti splošen pristop k obdelavi podatkov LiDAR, se bomo izognili implementacijskim detajlom. Algoritme bomo predstavili v kontekstu matematične morfologije [15], pri čemer bomo poskušali razviti definicije metod in operatorjev na način, ki bo neodvisen od dejansko uporabljene topologije. Prevzeli bomo, da je slednja podana v obliki neusmerjenega grafa $G = (V, R)$, ki ga določa množica vozlišč $V = \{v_i\}$ in množica robov $R = \{r_{i,j}\}$. Vsak rob je pri tem definiran s parom vozlišč $r_{i,j} = (v_i, v_j)$. V primeru trikotniške mreže in grafa najbližjih sosedov tako vsako vozlišče predstavlja dejansko točko LiDAR, v primeru mreže in vokselnega prostora pa vozlišče predstavlja sredino celice. V prvih dveh primerih je množica robov definirana z Delaunayevo triangulacijo, oziroma z razdaljo med točkami, v drugih dveh pa jo določajo tradicionalne 4 ali 8 oziroma 6 ali 24 sosednosti. Kadar se sklicujemo na poljubno atributno vrednost vozlišč (na primer skalarne vrednosti, pridobljene z meritvami LiDAR), to definiramo kot diskretno funkcijo $a: V \rightarrow \mathbb{R}$, katere definicijsko območje določa množica vozlišč V . Koordinate vozlišča v_i pa predstavimo z $x[v_i], y[v_i]$ in $z[v_i]$. Ta notacija zadošča za podajo dveh ključnih konceptov obdelave podatkov LiDAR, ki ju opišemo v nadaljevanju tega poglavja. Njuno uporabo pa demonstriramo v poglavju 3.

2.1 Diferencialni morfološki profili

Diferencialni profili v kontekstu matematične morfologije izhajajo iz tako imenovane granulometrije. To je postopek hierarhične podatkovne razgradnje, kjer s postopnim odstranjevanjem detajlov ocenjujemo geometrijske karakteristike vsebovanih objektov [16,17]. S ciljem obdelave podatkov LiDAR je bila granulometrija najprej uporabljena za izgradnjo tako imenovanih diferencialnih atributnih profilov v [10]. V [11] je bila podana še formalizacija diferencialnih morfoloških profilov, ki jo povzemam v nadaljevanju. V ta namen pa označimo morfološko odpiranje z okroglim strukturnim elementom w^s velikosti s , kot $\gamma^s(a)$. V skladu z notacijo teorije mrež označimo še infimum (minimum) kot \wedge , supremum (maksimum) pa kot \vee .



Slika 1: Definicija diferencialnih morfoloških profilov kjer (a) pridobimo dva različna vektorja odzivov v primeru vegetacije (levo) in stavb (desno) iz (b) vhodnega oblaka točk LiDAR.

Granulometrijo s poljubnim morfološkim operatorjem izvedemo s postopnim povečevanjem uporabljenega strukturnega elementa. Množico velikosti strukturnega elementa podamo s $\vec{S} = \{s_i\}$, kjer $s_{i-1} < s_i$. Kadar izvedemo granulometrijo z porabo morfološkega odpiranja, ustvarimo tako imenovani pozitivni vektor odzivov $\Delta_{\vec{S}}(a) = \{\gamma^{s_{i-1}}(a) - \gamma^{s_i}(a)\}$, kjer $i \in [1, I]$. Pri tem $\gamma^{s_i}(a)$ predstavlja ostanek podatkovne razgradnje, $\Delta_{\vec{S}}(a)$ pa nam omogoča izločanje naslednjih pomembnih karakteristik (glej sliko 1):

- funkcija $a' : V \rightarrow \mathbb{R}$ beleži največjo vrednost v $\Delta_{\vec{S}}(a)$, ki predstavlja atributno razliko med objektom in njegovo okolico ter
- funkcija $a^* : V \rightarrow \mathbb{R}$, ki določa velikost strukturnega elementa s_i , pri katerem je zabeležena najvišja vrednost v $\Delta_{\vec{S}}(a)$.

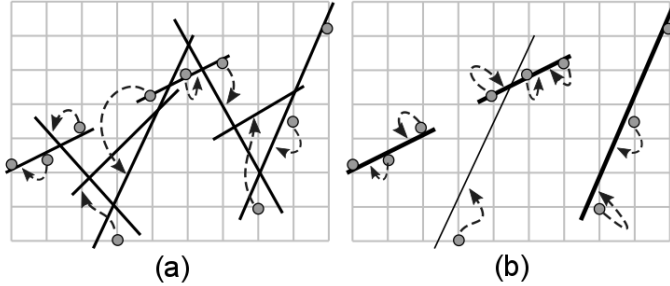
Formalno, diferencialne morfološke profile definiramo s preslikavo $\Theta : a \rightarrow (a', a^*)$, ki je v danem vozlišču v_j definirana kot:

$$a'[v_j] = \vee_{i \in [1, I]} \Delta_{\vec{S}}(a)[v_j][i], \quad (1)$$

$$a^*[v_j] = \wedge_{i \in [1, I]} s_i : \Delta_{\vec{S}}(a)[v_j][i] = a'[v_j].$$

2.2 Lokalno prilaganje točk

Zgolj ocenjevanje geometrijskih karakteristik objektov pogosto ni dovolj za njihovo uspešno razpoznavo, saj so si lahko v tem pogledu številni med seboj podobni. Metoda lokalnega prilaganja ploskev nam omogoča analizo površja objekta tako, da v poljubnem oblaku točk poišče tisto podmnožico, ki sledi poljubni vnaprej definirani porazdelitvi (na primer linearni ali polinomski). To se izvede v dveh korakih:



Slika 2: Lokalno prilaganje ploskev, kjer (a) v okolici vsake točke določimo najbolj prilagajočo se ploskev ter (b) vsako točko povežemo z njej najbližjo ploskvijo iz njene okolice (debelina ploskve je sorazmerna s številom povezanih točk).

- V koraku prilaganja se v okolici w_i^s vsake dane točke $v_i \in V$ izračuna najbolj prilagajoča se ploskev $\hat{\pi}^s[v_i]$ po metodi najmanjših kvadratov. Ker je postopek dobro poznan, njegove razlage na tem mestu ne podajamo. Detajlen opis metode je predstavljen v [18], v našem primeru pa uporabljamo implementacijo, podano v [19]. Očitno je, da takšno prilaganje pripelje do številnih napak, predvsem v primerih šuma in ostrih robov, kjer se srečajo točke, ki sledijo različnim porazdelitvam (glej sliko 2a). Vseeno pa se v postopku prilaganja okolice sosednjih vozlišč med seboj prekrivajo, vsako vozlišče v_i pa je tako vključeno v izračun vseh $\hat{\pi}^s[v_j]$ v okolici $v_j \in w_i^s$. Dejansko najbolj prilagajočo ploskev danemu v_i tako izberemo izmed teh (glej sliko 2c in 2d). Na ta način samodejno odstranimo slabo prilagajoče se ploskve. Formalno, najbolj prilagajočo ploskev $\pi^s[v_i]$ poljubni atributni funkciji a v okolico w_i^s točke v_i definiramo kot

$$\pi^s[v_i] = \hat{\pi}^s[v_i]: \forall v_k \in w_i^s, \hat{\pi}^s[v_k] - a[v_i] \geq \hat{\pi}^s[v_j] - a[v_i]. \quad (2)$$

- Povezovanje vozlišč nato izvedemo glede na množico ploskev π^s . V ta namen definiramo utežno funkcijo $u: R \rightarrow \mathbb{R}$, kot največjo izmed absolutnih napak para točk do para njima pripisanih ploskev. Formalno, utežno funkcijo $u[r_{i,j}]$ za rob $r_{i,j} \in R$ določimo kot

$$u[r_{i,j}] = \max\{|\pi^s[v_i] - a[v_j]|, |\pi^s[v_j] - a[v_i]|\}. \quad (3)$$

S preprosto pragovno funkcijo lahko nato izločimo tiste robove, katerih napaka je večja od največje dovoljene variacije znotraj točk t^L . Pridobljene povezane

komponente pa združujejo tista vozlišča, katerih višinske vrednosti sledijo enaki porazdelitvi (glej sliko 2b). Formalno lahko graf lokalno prilagojenih ploskev $L = (V^L, R^L)$, pri čemer $L \subseteq G$, definiramo kot

$$V^L = V, \quad (4)$$

$$R^L = \{r_{i,j} \in R: u[r_{i,j}] \leq t^L\}.$$

3. METODE RAZPOZNAVANJA OBJEKTOV V PODATKIH LIDAR

V tem poglavju predstavljamo več metod za razpoznavo objektov v podatkih LiDAR, ki temeljijo na diferencialnih morfoloških profilih in lokalno prilagajajočih se ploskvah iz prejšnjega poglavja. Konkretnije, predstavili bomo metodi za izgradnjo digitalnih modelov reliefa in zaznavo stavb (predlagani v [11]) ter postopek za delineacijo posameznih krošenj dreves (predlagano v [14]).

3.1 Izgradnja digitalnega modela reliefa

Izgradnja digitalnega modela reliefa (DMR) je običajno prvi korak obdelave podatkov LiDAR, saj predstavlja enega izmed ključnih podatkov za izvajanje številnih prostorskih analiz. Hkrati pa lahko razumemo teren kot ozadje pri obdelavi 3D podatkov zemeljskega površja in ga kot takega vključimo v razpoznavo ostalih objektov.

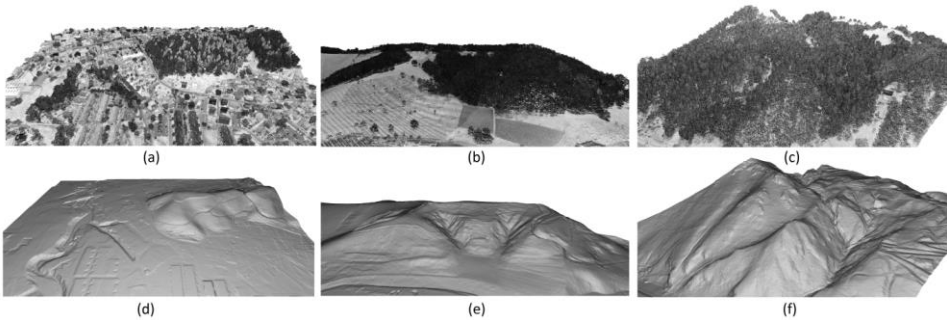
Ker obravnavamo točke terena kot spodnjo mejo oblaka točk LiDAR, izvedemo diferencialne morfološke profile nad višinskimi vrednostmi $z[v_i]$ točk $v_i \in V$. Pri tem je V lahko definiran z dejanskim oblakom točk LiDAR ali pa z njihovim podvzorčenjem v mrežo enakovelikih celic (glej poglavje 2). V obeh primerih pa temu pravimo digitalni model površja (DMP). V praksi se pogosto izkaže, da je v primeru razgibanega terena smiselno zmanjšati vpliv njegovih višinskih razlik na višinske razlike v DMP. Slednjega zato najprej zgladimo in izvedemo diferencialne morfološke profile nad višinskimi razlikami med dejanskim DMP in zglajenim DMP. Formalno definiramo zglajeni DMP s funkcijo $z^\uparrow: V \rightarrow \mathbb{R}$ v v_i kot

$$z^\uparrow[v_i] = z[v_i] - g(z)[v_i], \quad (5)$$

kjer funkcija g označuje konvolucijo z Gausovim jedrom. Velikost jedra je določena z uporabniškim parameterom in je odvisna od največjega vsebovanega objekta. Naj s_l določa velikost največjega objekta v oblaku točk LiDAR. Diferencialne morfološke profile nato definiramo s pozitivnim vektorjem odzivov $\Delta_{\vec{s}}(z^\uparrow)$, kjer $\vec{s} = \{0, 0, m, 0, 5, m, \dots, s_l\}$, pridobljeni funkciji z' in z^* (glej enačbo 1) pa uporabimo za definicijo terena. Vozlišča terena V^G definiramo kot:

$$V^G = \{v_i \in V: z'[v_i] \leq t^k z^*[v_i] + t^n\}, \quad (6)$$

kjer sta t^k in t^n uporabniško določeni vrednosti in določata največje razmerje med velikostjo objekta in njegovo višinsko razliko do okolice, ki jo lahko še pripišemo razgibanosti terena. Na mestih, kjer točk terena nismo razpoznali, izvedemo interpolacijo. Rezultati enačbe 6 so prikazani na sliki 3.



Slika 3: Digitalni površja in reliefa, ustvarjeni v (a in d) urbanem okolju, (b in e) gozdnem področju in (c in f) na zelo strmem terenu.

3.2 Razpoznavnost stavb

V tem poglavju dopolnimo enačbo 6 z metodo lokalno prilegajočih se ploskev, kar nam omogoča uspešno razpoznavnost stavb. V ta namen uporabimo linearno definicijo ploskev, saj lahko stavbe razlikujemo od ostalih naravnih objektov predvsem po ravninskem (linearnem) površju. Pri tem uporabimo zgolj tiste točke, ki ne predstavljajo terena $\overline{V^G} = V \setminus V^G$. Z uporabo enačbe 2 na ta način pridobimo množico ploskev $\pi^s[v_i]$, kjer $v_i \in \overline{V^G}$. Z enačbama 3 in 4 nato definiramo graf $L = (V^L, R^L)$, ki povezuje množice vozlišč, katerih višinske vrednosti sledijo podobnim linearnim porazdelitvam. Pragovno vrednost v enačbi 4 določimo glede na natančnost merilnika LiDAR, katerega naključna višinska napaka običajno zadošča definiciji $t^L = 0,15 m$. Graf $L = \{C_i\}$ v tem kontekstu obravnavamo kot množico povezanih komponent C_i . Vsako izmed njih pa naslovimo s poljubnim vsebovanim vozlišče $v_i \in C_i$. To nam omogoča, da lahko izvedemo filtriranje povezanih komponent glede na njihove geometrijske lastnosti. V našem primeru za ta namen uporabimo funkcijo površine $p(C_i)$, ki ocenjuje velikost geografskega področja, ki ga prekriva C_i . Definicijo vozlišč stavb V^B lahko tako zapišemo kot

$$V^B = \{v_i \in \overline{V^G} : p(C_i) \geq t^p\}, \quad (7)$$

kjer je t^p uporabniško določena pragovna vrednost in predstavlja najmanjšo površino, ki jo lahko zavzema stavba (na primer $t^p = 10 m^2$). Podrobna analiza učinkovitosti metode je predstavljena v [10], v splošnem pa lahko natančnost metode ocenimo z več kot 90%.

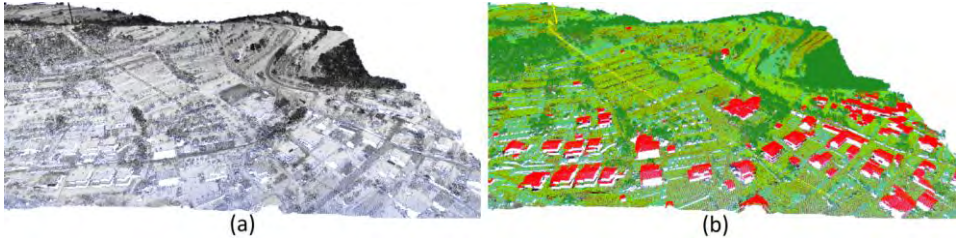
3.3 Razpoznavnost vegetacije in posameznih krošenj dreves

Razpoznavnost vegetacije lahko razumemo kot inverzni problem razpoznavnosti stavb. Če so stavbe določene z množicami vozlišč, katerih višinske vrednosti sledijo linearni porazdelitvi, želimo v primeru vegetacije zaznati tista vozlišča, ki temu pogoju ne zadoščajo. Postopek razpoznavnosti je zato podoben postopku razpoznavnosti stavb. Pomembnejša razlika je le to, da v tem primeru spremenimo definicijo enačbe 4 tako, da povežemo tiste točke, ki ne sledijo podobni linearni razporeditvi. Tako definiramo graf $L = (V^L, R^L)$ kot

$$V^L = V/V^G, \quad (8)$$

$$R^L = \{r_{i,j} \in R: \forall v_i, v_j \in V^L, u[r_{i,j}] \geq t^L\}.$$

Tudi v tem primeru izvedemo filtriranje povezanih komponent $C_i \in V^L$ glede na njihovo ocenjeno površino z enačbo 7. Množico vozlišč lahko nato razdelimo še na nizko, srednjo in visoko vegetacijo, glede na njihovo višinsko razliko do terena. Rezultat tega postopka, skupaj s prej opisanim postopkom razpoznave stavb, pa prikazuje slika 4.



Slika 4: Klasifikacija (a) vhodnega oblaka točk LiDAR v (b) točke terena (obarvane rjavo), točke stavb (obarvane rdeče), ter točke srednje, nizke in visoke vegetacije (obarvane zeleno).

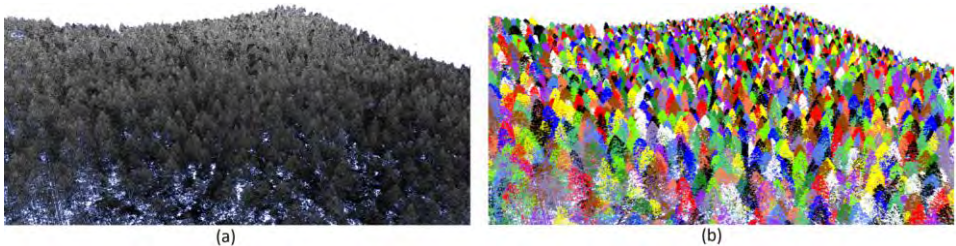
Nad tako določenimi vozlišči visoke vegetacije V^V nato izvedemo delinacijo posameznih drevesnih krošenj z algoritmom razvodja (ang. watershed). V ta namen najprej identificiramo vrhove dreves z uporabo lokalnega prilagajanja ploskev (enačba 2). Namesto linearne funkcije pa v tem primeru uporabimo polinome druge stopnje, določene z enačbo:

$$\pi^s[v_i] = c_5x[v_i]^2 + c_4x[v_i]y[v_i] + c_3y[v_i]^2 + c_2x[v_i] + c_1y[v_i] + c_0. \quad (9)$$

Glede na dobljene koeficiente lahko okolico danega vozlišča $v_i \in V^V$ ocenimo kot:

- prevoj, če $c_4^2 - 4c_5c_3 \geq 0$, sicer
- konveksno, če $c_5 \geq 0$ in
- konkavno, če $c_5 < 0$.

Intuitivno lahko sklepamo, da vozlišča s konkavno okolico določajo vrhove dreves in jih tako uporabimo kot označevalce regij za postopek razvodja, ki ga izvedemo nad DMP. Celoten prostor tako razdelimo v množice posameznih dreves, kot to prikazuje slika 5.



Slika 5: Delinacija posameznih drevesnih krošenj v (a) vhodnem oblaku točk, kjer je (b) vsaka razpoznanjena krošnja pobarvana z naključno določeno barvo.

Na ta način nam uspe pravilno zaznati več kot 93% vseh najpomembnejših (najvišjih) dreves. Metoda, skupaj z analizo rezultatov, je podrobneje predstavljena v [14].

5. ZAKLJUČEK

V tem članku smo predstavili matematično ogrodje za obdelavo podatkov LiDAR, ki nam omogoča razpoznavo terena, stavb in nizke, srednje ter visoke vegetacije hkrati z delineacijo slednje v posamezne drevesne krošnje. Glede na predstavljene rezultate lahko trdimo, da predlagano matematično orodje omogoča natančno obdelavo podatkov LiDAR. Uporaba teh metod je bila demonstrirana tudi v nacionalnem projektu zajema 3D površja Slovenije, s čimer smo pokazali robustnost predlaganega pristopa. Vrednost tako pridobljenih informacij o geometriji objektov na površju Zemlje pa potrjujejo tudi številne študije, ki temeljijo na podatkih LiDAR. Kot zanimivejše pa lahko izpostavimo oceno fotovoltaičnega potenciala stavb [20], oceno na ta način pridobljene električne energije [21] in optimizacijo postavitve stavb in njihove oblike za dosego povečane učinkovitosti fotovoltaičnih sistemov [22].

ZAHVALA

Predstavljene raziskave je podprla Agencija za raziskovanje Republike Slovenije s projekti L2-3650, J2-5479 in J2-6764.

LITERATURA

1. S. Popescu (2011), Advances in Environmental Remote Sensing: Sensors, Algorithms, and Applications. In: Weng, Q. (Ed.), *Advances in Environmental Remote Sensing: Sensors, Algorithms, and Applications*. Boca Raton: CRC Press, str. 57–84.
2. <http://www.riegl.com/>
Riegel Specifications 2010, DataSheet Q560 20-09-2010.
3. G. Vosselman (2000), Slope based filtering of laser altimetry data. *International Archives of Photogrammetry and Remote Sensing*, vol. 33 (Part B3), str. 935–942.
4. G. Sithole (2001), Filtering of laser altimetry data using a slope adaptive filter. *International Archives of Photogrammetry and Remote Sensing*, vol. 34 (Part 3/W4), str. 203–210.
5. J. Shan, A. Sampath (2005), Urban DEM generation from raw LiDAR data: a labeling algorithm and its performance. *Photogrammetric Engineering and Remote Sensing*, vol. 71 (2), str. 217–222.
6. C.K. Wang, Y.H. Tseng (2010), Dem generation from airborne lidar data by an adaptive dual-directional slope filter. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38 (Part 7B), str. 628–632.

7. J. Wang, J. Shan (2009), Segmentation of LiDAR point clouds for building extraction, *Proceedings ASPRS Annual Conference*, Baltimore: American Society for Photogrammetry and Remote Sensing, str. 9-13.
8. A. T. Hudak, N. L. Crookston, J. S. Evans, D. E. Hall, M. J. Falkowski (2008), Nearest neighbor imputation of species-level, plot-scale forest structure attributes from LiDAR data. *Remote Sensing of Environment*, vol. 112(5), str. 2232-2245.
9. T.J. Pingel, K.C. Clarke, W. A. McBride, (2013), An improved simple morphological filter for the terrain classification of airborne LiDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 77, str. 21–30.
10. D. Mongus, Domen, B. Žalik (2014), Computationally efficient method for the generation of a digital terrain model from airborne LiDAR data using connected operators, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol 7(1), str. 340-351.
11. D. Mongus, N. Lukač, B. Žalik. (2014), Ground and building extraction from LiDAR data based on differential morphological profiles and locally fitted surfaces, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 93, str. 145-156.
12. D. Mongus, N. Lukač, D. Obrul, B. Žalik (2013), Detection of planar points for building extraction from LiDAR data based on differential morphological and attribute profiles. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 1 (II-3), str. 21-26.
13. F. Hosoi, Y. N. Fumiki, O. Kenji (2013), 3-D voxel-based solid modeling of a broad-leaved tree for accurate volume estimation using portable scanning lidar, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 82, str. 41-48.
14. D. Mongus, B. Žalik (2015), An efficient approach to 3D single tree-crown delineation in LiDAR data, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 108, str. 219-233.
15. L. Najman. H. Talbot (2010), *Mathematical Morphology: From Theory to Applications*, London, U.K.: Wiley-ISTE.
16. J. A. Benediktsson, M. Pesaresi, K. Amason (2003), Classification and feature extraction for remote sensing images from urban areas based on morphological transformations, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41(9), str. 1940-1949.

17. G. K. Ouzounis, M. Pesaresi, P. Soille (2012), Differential area profiles: Decomposition properties and efficient computation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34(8), str. 1533-1548.
18. P. Schneider, D. H. Eberly (2002), *Geometric Tools for Computer Graphics*. San Francisco: Morgan Kaufman.
19. <http://www.geometrictools.com/Documentation/>
D. Eberly (1999), Least Squares Fitting of Data.
20. N. Lukač, D. Žlaus, S. Seme, B. Žalik, G. Štumberger (2013), Rating of roofs' surfaces regarding their solar potential and suitability for PV systems, based on LiDAR data. *Applied energy*, vol. 102, str. 803-812
21. N. Lukač, S. Seme, D. Žlaus, G. Štumberger, B. Žalik (2014), Buildings roofs photovoltaic potential assessment based on LiDAR (light detection and ranging) data. *Energy*, vol. 66, str. 598-609.
22. M. Bizjak, B. Žalik, N. Lukač (2015), Evolutionary-driven search for solar building models using LiDAR data. *Energy and buildings*, vol. 92, str. 195-203.

IZBOLJŠANJE UPORABNIŠKE IZKUŠNJE SPLETNIH KAZINOJEV S PREPOZNAVO OBRAZNIH IZRAZOV

Rok Kreslin, Rok Irt, Luka Kumelj, Mitja Pugelj

Comland d.o.o.

E-pošta: rok.kreslin@comland.si

URL: <http://www.comland.si>

POVZETEK: *Zaradi enosmerne vizualne interakcije krupjeji v spletnem kazinoju velikokrat ne posvečajo dovolj pozornosti svojemu vedenju. Spremljanje vedenja z nadzorniki je časovno potratno in neobjektivno. V članku predstavljen sistem omogoča samodejno analizo vedenja krupjejev z analizo obraznih izrazov in smerjo pogleda. O nezaželenem vedenju v realnem času obvešča tako nadzornike kot krupjeje. Spletnemu kazinoju zniža stroške nadzornikov, hkrati pa poskrbi za boljšo uporabniško izkušnjo in s tem dodano vrednost igralcev.*

1. UVOD

Uporabniška izkušnja (UX) pomembno vpliva na uspeh storitev in produktov. Pri storitvah, pri katerih prihaja do vizualne interakcije uporabnika s ponudnikom storitev, je celovito spremljanje uporabniške izkušnje uporabnika težavno. Poleg same vsebine storitve namreč na uporabniško izkušnjo pomembno vpliva tudi vedenje ponudnika storitve.

Segment, ki se srečuje z omenjenim problemom, so spletni kazinoji. Preden nadaljujemo definirajmo pojem spletnega kazinoja. Spletni kazino je kazino, kjer krupjeji v fizičnem svetu izvajajo igre, igralci pa te igre spremljajo preko videa na njihovih elektronskih napravah. V tej konfiguraciji je vizualna interakcija enosmerna: igralci vidijo krupjeje, krupjeji igralcev ne.

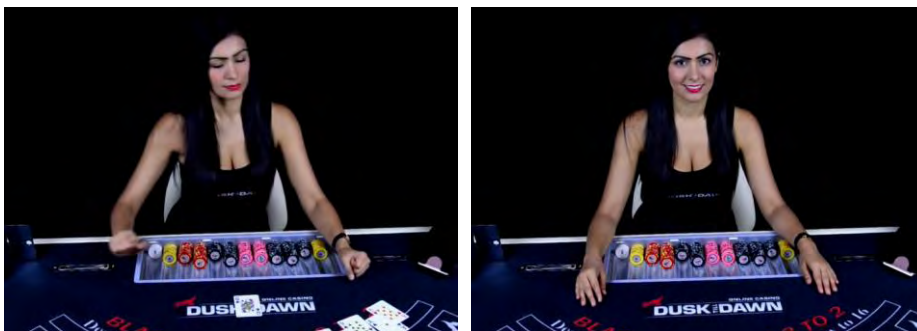
Da bi spletni kazinoji privabili in obdržali igralce, je pomembno, da je vedenje krupjejev pozitivno. Večina spletnih kazinojev obratuje 24 ur na dan. Zaradi utrujenosti in pomanjkanja dvosmerne vizualne interakcije krupjeji pozabijo, da jih preko videa spremlja sto in več igralcev. Če ti igralci začutijo, da vizualna interakcija ni pozitivna, zamenjajo spletni kazino, kar povzroči izgubo v prihodkih kazinoja.

Spletni kazinoji problem tradicionalno rešujejo z nadzorniki. To so zaposleni, ki spremljajo potek iger in vedenje krupjejev. Tak pristop je stroškovno neučinkovit, hkrati pa je težko zagotoviti objektivno spremljanje krupjejev.

V članku je predstavljena rešitev, ki omogoča samodejno spremljanje vedenja krupjejev. Rešitev modelira vedenje s prepoznavo obraznih izrazov in smerjo pogleda. V primeru, da zazna nezaželeno vedenje, o tem obvesti nadzornika in krupjeja. Ta svoje vedenje prilagodi instantno in s tem pripomore k pozitivni izkušnji igralcev za njegovo mizo.

2. PROBLEM

V spletnih kazinojih imamo opravka z enosmerno vizualno komunikacijo: igralci vidijo krupjeje, krupjeji igralcev ne. Vizualna informacija, ki jo igralci dobijo o krupjejih, je lahko pozitivna ali negativna in pomembno vpliva na uporabniško izkušnjo igralcev. Slika 1 prikazuje primer pozitivnega in negativnega vedenja, kot ga vidi igralec.



Slika 1: negativno vedenje krupjeja (levo): pogled vstran, negativni obrazni izraz; pozitivno vedenje krupjeja (desno): pogled v kameru, pozitiven obrazni izraz.

Vzrokov za negativno vedenje krupjejev je več:

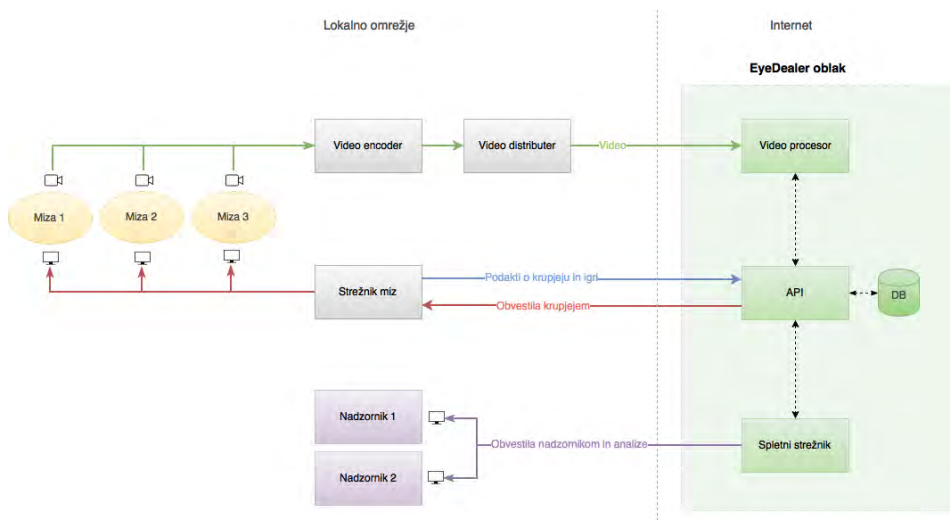
- Neizkušnost - krupjeji, ki šele začenjajo svojo poklicno pot, se osredotočajo zgolj na pravilni potek igre in ne posvečajo pozornosti interakciji z igralci.
- Utrujenost - predvsem v nočnih izmenah so krupjeji utrujeni in ne posvečajo dovolj pozornosti igralcem.
- Pomanjkanje dvosmerne vizualne interakcije - krupjeji izgubijo občutek, da igralci na drugi strani imajo vizualno informacijo o njih.

Tradicionalno spletni kazinoji spremljajo vedenje krupjejev z nadzorniki. Takšna rešitev je stroškovno neučinkovita. Ker imajo nekateri spletni kazinoji tudi več kot 100 igralnih miz, potrebujemo veliko nadzornikov. Če bi te razbremenili kontinuiranega spremljanja krupjejev, bi lahko racionalizirali število nadzornikov. Prav tako težko zagotovimo, da so nadzorniki objektivni in da imajo enaka merila za vse igralce.

V nadaljevanju je predstavljena rešitev, ki odpravi potrebo po ročnem spremljanju vedenja krupjejev. Sistem s pomočjo algoritmov za prepoznavo obraznih izrazov in smeri pogleda vedenje spremlja samodejno in o nepravilnostih obvešča krupjeje ter nadzornike.

3. REŠITEV

Visoko nivojska shema predlaganega rešitve je predstavljena na sliki 2. Rešitev je sestavljena iz video procesorjev, ki analizirajo vedenje krupjejev in spletnega orodja, ki skrbi za pošiljanje obvestil ter pregled zgodovine vedenja krupjejev. Celoten sistem je implementiran v oblaku v obliki Kubernetes sistema [1]. Video procesor analizira videje iz obstoječega sistema za distribucijo videjev, preko katerega spletni kazino ponuja video igralcem. Spletni kazino tako nima dodatnih stroškov z infrastrukturo.



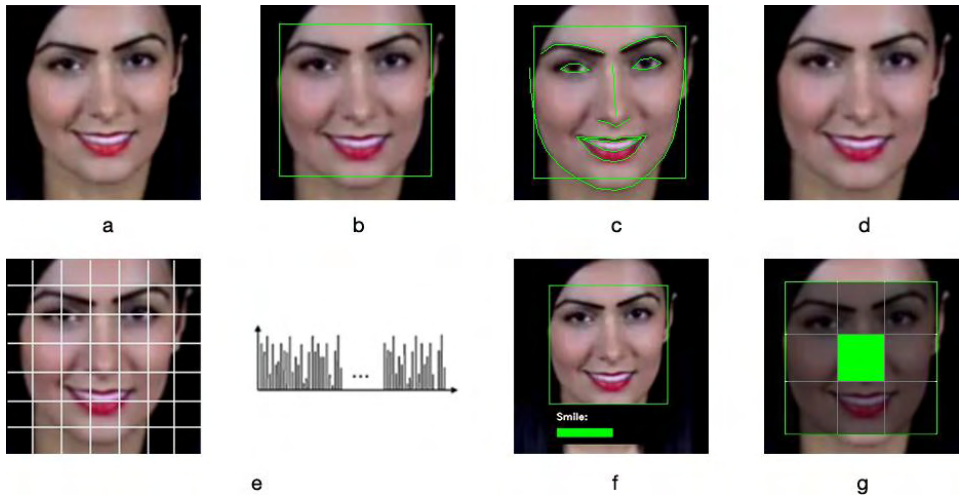
Slika 2: Shema in integracija sistema za samodejno spremljanje vedenja krupjejev.

3.1 Algoritem za analizo obraznih izrazov in smeri pogleda

Vedenje krupjejev parametriziramo z valenco (pozitivnostjo/negativnostjo) obraznega izraza in smerjo pogleda. Pomembno je, da algoritem deluje v realnem času. S tem zagotovimo, da lahko iz videa izluščimo vso informacijo, ki je na voljo. Prav tako mora algoritem delovati pri nižjih resolucijah, saj je velikost obraza krupjeja v videih spletnih kazinojev velikokrat majhna.

Slika 3 prikazuje vizualne korake v izvajanju algoritma. Na vhodni sliki najprej poiščemo obraz. Na najdenem obrazu poiščemo ključne obrazne točke in obraz geometrijsko normaliziramo. S tem omejimo vpliv poze obraza na natančnost prepoznavne valence. Za geometrijsko normalizacijo sledi svetlobna normalizacija, s katero omejimo vpliv osvetlitve na prepoznavo valence. Iz svetlobno normaliziranega opisnika slike izluščimo valenco, smer pogleda pa določimo iz konfiguracije ključnih obraznih točk.

Posamezni koraki so podrobneje opisani v podpoglavjih. Na tem mestu naj omenimo, da se vsi koraki algoritma testirajo z navzkrižnim preverjanjem, vendar so uporabljene množice slik last podjetja in niso javno dostopne. Zato rezultatov testiranja v članku ne navajamo.



Slika 3: Koraki v algoritmu za analizo obraznih izrazov in smeri pogleda. Vhodna slika (a), detektiran obraz (b), ključne točke na obrazu(c), geometrijsko normaliziran obraz (d), svetlobno normalizirani opisnik slike (e), regresor valence (f), smer pogleda(g).

3.1.1 Detekcija in sledenje obraza (b)

Naloga modula za sledenje obraza je, da na sliki iz videa najde obraz in mu sledi. Osnovo modula predstavlja detektor obraza, ki temelji na HOG značilkah [2] in na pristopu strukturiranega iskanja objektov [3]. Detektor obraza vsebuje pet podsistemov. Vsak od njih je zadolžen za detekcijo obraza ob naslednjih rotacijah glave: pogled naravnost, pogled levo, pogled desno, pogled navzgor in pogled navzdol. Na tak način pokrijemo večje število rotacij glave in izboljšamo rezultat sledenja.

3.1.2 Iskanje ključnih obraznih točk (c)

Iskanje ključnih obraznih točk je pomembno iz dveh razlogov. S pomočjo konfiguracije ključnih točk izvedemo geometrijsko normalizacijo slike in določimo 3D rotacijo glave, ki je ključna za ugotavljanje smeri pogleda. Za iskanje ključnih točk uporabimo gručo regresijskih dreves [4].

3.1.3 Geometrijska normalizacija obraza (d)

Ob najdeni konfiguraciji ključnih obraznih točk moramo poiskati transformacijo, ki te točke poravnava s ključnimi obraznimi točkami na kanoničnem obrazu. Kanonični obraz je standardni obraz v frontalni pozi, brez neželenih rotacij, v zeleni velikosti. S transformacijo omejimo vpliv poze obraza na nadaljnje procesiranje. Uporabljamo odsekoma linearno 2D transformacijo.

3.1.4 Svetlobna normalizacija obraza (e)

Standardni problem algoritmov računalniškega vida je osvetlitev okolja. Dve sliki identičnega objekta/subjekta se namreč lahko ob različnih iluminacijah močno razlikujeta. Ob neupoštevanju zgornjega dejstva se kaj hitro lahko pripeti, da zgradimo modele, ki v določenih svetlobnih pogojih delujejo, v drugih popolnoma odpovejo. Zato je svetlobna normalizacija obraza nujen korak v procesu razpoznavne obraznih izrazov. Za opis svetlobno-invariantne slike smo uporabili značilke, ki temeljijo na bločnih histogramih LBP [5].

3.1.5 Regresija valence obraznega izraza (f)

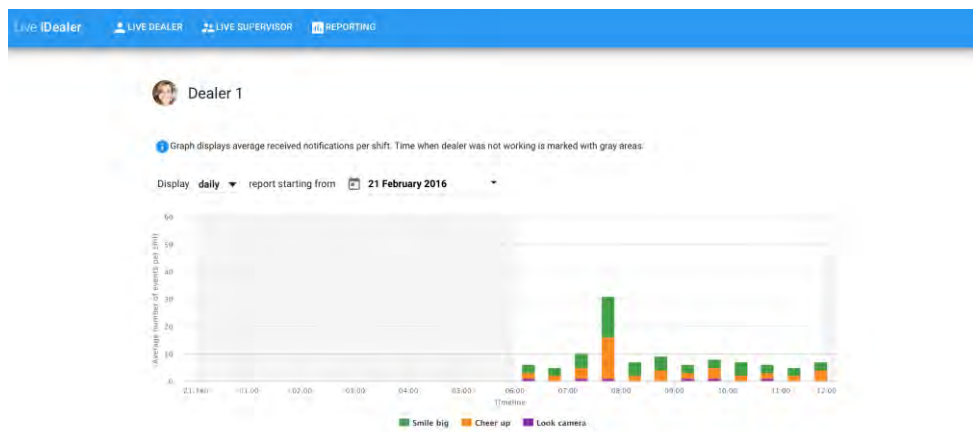
Naloga regresorja valence je, da se na osnovi opisa slik (v našem sistemu so to bločni histogrami LBP), odloči, kako pozitiven oziroma negativen je opisan obrazni izraz. Za predikcijo valence uporabljamo SVR regresorje [6].

3.1.6 Regresija 3D rotacije glave in smeri pogleda (g)

Najdeno konfiguracijo ključnih točk primerjamo s kanoničnimi ključnimi točkami in poiščemo 3D rotacijo glave [7]. Rotacijo glave diskretiziramo v 9 razredov, s katerimi opišemo smer pogleda krupjeja.

3.2 Spletno orodje za analizo krupjejev

Algoritem za analizo obraznih izrazov in smeri pogleda generira ogromno količino podatkov. Da bi bili ti podatki uporabni, jih je potrebno analizirati in interpretirati. Za to poskrbi spletno orodje. S pomočjo vnaprej definiranih pravil se iz podatkov v realnem času izluščijo dogodki, ki opozarjajo na nezaželeno vedenje krupjeja. Ti dogodki se posredujejo tako nadzornikom (preko spletnega orodja), kot krupjejem (integracija s sistemom spletnih kazinojev). Hkrati se dogodki shranijo v podatkovno bazo za poznejšo analizo in spremljanje trendov v obnašanju krupjejev. Slika 4 prikazuje zasloni posnetek spletnega orodja za analizo zgodovine obvestil krupjejev.



Slika 4: zasloni posnetek spletnega orodja za analizo zgodovine obvestil.

4. ZAKLJUČEK

Krupjeji v spletnih kazinojih velikokrat posvečajo premalo pozornosti pozitivnemu vedenju, kar negativno vpliva na uporabniško izkušnjo igralcev. Kontinuirano spremljanje krupjejev z nadzorniki je potratno in neučinkovito. Z uporabo samodejne rešitve za analizo obraznih izrazov in smeri pogleda lahko spremljanje krupjejev avtomatiziramo. O nezaželenem vedenju v realnem času obvestimo tako nadzornike kot krupjeje. Ti lahko ukrepajo takoj in s tem pripomorejo k boljši uporabniški izkušnji igralcev.

V prihodnosti bomo v sistem vključili še več parametrov kot so analiza govora in mimike telesa. Opisan sistem seveda ni omejen na spletne kazinoje ampak je uporaben v vseh domenah, kjer prihaja do vizualne interakcije med uporabniki in ponudniki storitev.

LITERATURA

1. <http://kubernetes.io>
Informacije o Kubernetes sistemu.
2. N. Dalal, B. Triggs (2005), Histograms of oriented gradients for human detection, *Computer Vision and Pattern Recognition (CVPR)*, vol. 1, str. 886-893
3. M. B. Blaschko, C. H. Lampert (2008), Learning to localize objects with structured output regression, *European Conference on Computer Vision (ECCV)*
4. X. Cao, Y. Wei, F. Wen, J. Sun (2013), Face alignment by explicit shape regression, *International journal of computer vision*, vol. 107, str. 177-190
5. C. Shan, S. Gong, P. W. McOwan (2009), Facial expression recognition based on local binary patterns: a comprehensive study, *Image and vision computing*, vol. 27, str. 803-816
6. A. J. Smola, B. Schölkopf (2004), A tutorial on support vector regression, *Statistics and Computing*, vol. 14, str. 199-222
7. V. Lepetit, F. Moreno-Noguer, P. Fua (2008), EPnP: An Accurate O(n) Solution to the PnP Problem, *International Journal of Computer Vision*, vol. 81, str. 155-166

INDUSTRIJSKE APLIKACIJE

Barvna vztrajnost na osnovi ocenjevanja povprečne reflektivnosti scene

Martin Šavc, Božidar Potočnik
Laboratorij za sistemsko programsko opremo
Fakulteta za elektrotehniko, računalništvo in informatiko
Univerza v Mariboru
E-pošta: martin.savc@um.si

POVZETEK: *V tem prispevku opišemo nov algoritem barvne vztrajnosti. Namesto običajnega pristopa, kjer se osredotočimo na ocenjevanje osvetlitve scene, predlagamo ocenjevanje povprečne reflektivnosti scene. Uporabljene značilnice pripravimo tako, da so invariantne na spremembe osvetlitve scene. Nato pa uporabimo metode strojnega učenja za ocenjevanje povprečne reflektivnosti scene. To oceno nato uporabimo za izračun osvetlitve in popravek barv. Rezultate predlaganega postopka primerjamo s klasičnim pristopom neposrednega ocenjevanja osvetlitve in pokažemo smiselnost takšnega pristopa. Predlagan postopek na podatkovni zbirki Gehler doseže mediano kotne napake do 2,71 stopinje.*

1 Uvod

Barve v barvni sliki pogosto pripisujemo in obravnavamo zgolj kot lastnosti opazovanih predmetov. Vendar so izmerjene barve odvisne od številnih elementov scene. Eden izmed pomembnih elementov, ki ga zaradi lastne percepcije hitro spregledamo, je barva osvetlitve scene. Ta ima lahko bistven vpliv na barve, ki bi ga morali upoštevati pri primerjavi različnih slik.

Človek in nekatere druge živali z razvitim barvnim vidom imajo sposobnost dojemanja barv, ki je v določeni meri neodvisna od barve osvetlitve. Tej lastnosti pravimo barvna vztrajnost in precej truda je bilo usmerjenega v razvijanje algoritmov, ki bi jo bili sposobni posnemati. Takšne algoritme najdemo tudi v barvnih kamerah, kjer služijo natančnejši barvni reprodukciji.

Algoritem barvne vztrajnosti običajno najprej analizira zajeto sliko ter na podlagi izbranih lastnosti poda oceno barve osvetlitve. S to oceno nato popravi sliko na barve, ki bi bile izmerjene z belo osvetlitvijo. Ocena barve osvetlitve je lahko osnovana na fizikalnih pojavih, npr. na osnovi zrcalnega odboja svetlobe [1]. Pogosteje pa je ocena osnovana

na statističnih lastnostih izbrane značilnice [2]. Izbira značilnice igra pomembno vlogo. Odražati mora barvo osvetlitve, hkrati pa mora biti invariantna na spremembe površin.

Pridobitev takšne značilnice se izkaže kot izjemno zahtevna. Večina značilnic je razvitih s povsem obratnim namenom: da bi dobro odražale površine v sceni in bile odporne na spremembe osvetlitve. V tem prispevku zato predlagamo drugačen pristop k barvni vztrajnosti. Uporabili bomo značilnice, ki bodo invariantne na spremembe osvetlitve. Z njimi bomo ocenili barvno lastnost površin, in sicer povprečno barvo reflektivnosti površin v sliki. Reflektivnost je lastnost, ki pove kakšen delež svetlobe se odbije s površine pri izbrani valovni dolžini. Uporabili bomo metode strojnega učenja, ki so se v preteklosti izkazale pri ocenjevanju osvetlitve. S to oceno bomo nato primerjali barve v zajeti sliki in ocenili osvetlitev. Ob pregledu literature opazimo, da obstaja že vrsta algoritmov, ki deluje po tem principu [4]. Naš pristop pa je zelo ekspliciten, kar nam bo dolgoročno omogočilo njegovo izpopolnitev.

Za opis barve bomo uporabili poenostavljen model razpršenega odboja svetlobe, ki ga najdemo v [3]. Barvno vrednost v pikslu \mathbf{x} opišemo z enačbo:

$$I_c(\mathbf{x}) = S_c(\mathbf{x})e_c, \quad (1)$$

kjer je S_c barvna vrednost površine (reflektivnost), e_c je barvna vrednost osvetlitve in $c \in (R, G, B)$ je barvni kanal. Običajno nas zanimajo barve I_c^{wh} izmerjene z belo osvetlitvijo $e_c^{wh} = 1/\sqrt{3}$. Z algoritmi barvne vztrajnosti tipično ocenjujemo barvo osvetlitve e_c . Barve slike nato popravimo po enačbi:

$$I_c^{wh}(\mathbf{x}) = \frac{I_c(\mathbf{x})e_c^{wh}}{e_c} = \frac{I_c(\mathbf{x})}{\sqrt{3}e_c}. \quad (2)$$

Preostanek članka je sestavljen iz naslednjih poglavij. V 2. poglavju bomo predstavili izbrane metode strojnega učenja za oceno osvetlitve v sliki. V tem poglavju bomo opisali tudi uporabljeno značilnico. V 3. poglavju bomo opisali prilagoditev značilnice in metod ocenjevanja osvetlitve za določitev povprečne reflektivnosti scene. V tem poglavju bomo tudi natančneje razložili, kako iz ocene povprečne reflektivnosti pridobimo oceno osvetlitve. V 4. poglavju bomo opisali eksperiment, s katerim smo ovrednotili naš pristop. V 5. poglavju pa bomo povzeli dobljene rezultate in nakazali smernice za nadaljnje delo.

2 Ocenjevanje osvetlitve

Za ocenjevanje osvetlitve s strojnem učenjem najprej izberemo primerno značilnico za opis slike. S tehniko nadzorovanega učenja nato poiščemo oz. naučimo funkcijo, tj. transformacijo, ki to značilnico preslika v oceno osvetlitve. V fazi uporabe za sliko z neznano osvetlitvijo najprej določimo značilnico in z naučeno funkcijo ocenimo osvetlitev.

V nadaljevanju bomo opisali dva pristopa za ocenjevanje osvetlitve e_c z uporabo strojnega učenja, opisana v [5] in [6]. Oba pristopa uporabljata isto značilnico. Nato bomo

predstavili pristop z umetnimi nevronskimi mrežami, zatem pa še pristop s strojem podpornih vektorjev. Oba pristopa bomo uporabili za ocenjevanje povprečne barve scene namesto osvetlitve. To oceno bomo nato uporabili za izračun osvetlitve.

2.1 Binarni histogram

Kot značilnica za oceno osvetlitve je v [5] in [6] uporabljen binarni histogram normaliziranih barv. Binarni histogram bi naj bil bolj odporen na spremembe scene, v primerjavi z običajnim histogramom.

Iz barvne slike najprej pridobimo normalizirane barve. Barvno sliko pretvorimo v sliko svetlosti L in sliki normiranih barvnih kanalov r in g . Vrednost L izračunamo kot vsoto barvnih kanalov po enačbi:

$$I_L(\mathbf{x}) = I_R(\mathbf{x}) + I_G(\mathbf{x}) + I_B(\mathbf{x}). \quad (3)$$

To nato uporabimo za izračun normaliziranih barv r in g kot:

$$I_r(\mathbf{x}) = I_R(\mathbf{x})/I_L(\mathbf{x}), \quad (4)$$

$$I_g(\mathbf{x}) = I_G(\mathbf{x})/I_L(\mathbf{x}). \quad (5)$$

Iz normaliziranih barv izračunamo 2D binarni histogram B_{rg} . Barvni prostor razdelimo na N enako velikih intervalov z mejami r_n in g_n za $n \in [0, N]$. V histogramu B_{rg} nato označimo z 1 tista polja, pri katerih najdemo vsaj en piksel v njihovem intervalu:

$$B_{rg}(n, m) = \begin{cases} 1 : \exists \mathbf{x}, & I_r(\mathbf{x}) \in [r_n, r_{n+1}) \wedge \\ & I_g(\mathbf{x}) \in [g_m, g_{m+1}) \\ 0 & \end{cases} \quad (6)$$

Boljše rezultate dosežemo z uporabo 3D binarnega histograma B_{rgL} [2], kjer je upoštevana tudi svetlost v sliki. Svetlosti razdelimo na N_L enako velikih intervalov z mejami L_n za $n \in [0, N_L]$. Polja histograma nato nastavimo po spodaj zapisanem pravilu:

$$B_{rgL}(n, m, l) = \begin{cases} 1 : \exists \mathbf{x}, & I_r(\mathbf{x}) \in [r_n, r_{n+1}) \wedge \\ & I_g(\mathbf{x}) \in [g_m, g_{m+1}) \wedge \\ & I_L(\mathbf{x}) \in [L_l, L_{l+1}) \\ 0 & \end{cases} \quad (7)$$

2.2 Večslojni perceptron

Prvi pristop k ocenjevanju osvetlitve s strojnim učenjem je povzet po članku [6] in uporablja večplastni perceptron MLP (*ang. multilayer perceptron*). Lastnosti MLP so odvisne

od njegove topologije: števila nevronov na vhodnem sloju, števila skritih slojev in nevronov v teh slojih ter števila nevronov na izhodnem sloju. Pomembno vlogo igra tudi prenosna funkcija, ki karakterizira lastnosti posameznega nevrona. Za učenje MLP je uporabljen algoritem vzvratnega širjenja napake (*ang. back propagation*).

Z MLP naučimo funkcijo za napoved osvetlitve, in sicer:

$$\begin{bmatrix} e_R \\ e_G \end{bmatrix} = \mathbf{e}_{MLP}(B_{rg}). \quad (8)$$

MLP se nauči oceniti le vrednost rdečega in zelenega kanala osvetlitve. Ker je vrednost osvetlitve normirana, lahko modro vrednost izračunamo po enačbi:

$$e_B = \sqrt{1 - e_R^2 - e_G^2}. \quad (9)$$

2.3 Stroj podpornih vektorjev

Drug pristop k ocenjevanju osvetlitve s strojnim učenjem je povzet po članku [5] in uporablja stroj podpornih vektorjev SVM (*ang. support vector machine*). Osnovna metoda SVM omogoča linearno klasifikacijo in regresijo. To lahko razširimo z uporabo nelinearnih jeder. Za dobre rezultate moramo v postopku učenja izbrati tudi primerne hiper-parametre metode. Pri izbiri hiper-parametrov smo se zgledovali po članku [2].

Učenje regresije s SVM privzeto ne omogoča učenje večdimenzionalnih izhodov. Zato s SVM-jem naučimo dve funkciji in te združimo v vektorsko funkcijo kot:

$$\begin{bmatrix} b \\ c \end{bmatrix} = \begin{bmatrix} f_{SVM,b}(a) \\ f_{SVM,c}(a) \end{bmatrix} = \mathbf{f}_{SVM}(a). \quad (10)$$

Zgornja enačba opisuje splošno transformacijo, kjer funkciji $f_{b,SVM}(a)$ in $f_{c,SVM}(a)$ naučeni s SVM iz značilnice a ocenita vrednosti b in c . Funkcija \mathbf{f}_{SVM} pa ti dve funkciji združi za ocenjevanje vektorja. Tako lahko s SVM naučimo funkcijo za napoved osvetlitve iz 2D barvne značilnice kot:

$$\begin{bmatrix} e_R \\ e_G \end{bmatrix} = \mathbf{e}_{SVM}(B_{rg}). \quad (11)$$

Podobno s SVM naučimo tudi funkciji za napoved iz 3D barvne značilnice, in sicer:

$$\begin{bmatrix} e_R \\ e_G \end{bmatrix} = \mathbf{e}_{SVM}(B_{rgL}). \quad (12)$$

Modri kanal izračunamo na enak način kot pri metodi MLP po enačbi (9).

3 Ocenjevanje reflektivnosti

Pri neposrednem ocenjevanju osvetlitve je glavni problem odvisnost uporabljene značilnice od vsebine scene. Idealna značilnica bi naj bila invariantna na vsebino scene $S_c(\mathbf{x})$ in bi naj odražala samo osvetlitev e_c . V praksi so vse uporabljene značilnice močno odvisne od vsebine. Zato v tem prispevku predlagamo, da metode strojnega učenja uporabimo za neposredno ocenjevanje povprečne normirane barve reflektivnosti, kar je enakovredno povprečni barvi v sliki, izmerjeni z belo osvetlitvijo. To izračunamo po enačbi:

$$k_1 \hat{I}_c^{wh} = \bar{I}_c^{wh} = \sum_{\mathbf{x}} I_c^{wh}(\mathbf{x}), \quad (13)$$

$$k_2 \hat{I}_c = \bar{I}_c = \sum_{\mathbf{x}} I_c(\mathbf{x}), \quad (14)$$

kjer sta \hat{I}_c^{wh} in \hat{I}_c povprečna barva pod belo in pod neznano osvetlitvijo. Koeficienta k_1 in k_2 sta normi barvnih vektorjev \bar{I}_c^{wh} in \bar{I}_c . Vrednosti $I_c^{wh}(\mathbf{x})$ v enačbi (13) lahko razširimo z enačbo (2) in dobimo enačbo:

$$k_1 \hat{I}_c^{wh} = \sum_{\mathbf{x}} \frac{I_c(\mathbf{x})}{e_c \sqrt{3}} = \frac{1}{e_c \sqrt{3}} \sum_{\mathbf{x}} I_c(\mathbf{x}). \quad (15)$$

To sedaj združimo z enačbo (14), izpostavimo e_c ter dobimo enačbo:

$$e_c = \frac{k_2 \hat{I}_c}{\sqrt{3} k_1 \hat{I}_c^{wh}}, \quad (16)$$

$$k_3 e_c = \bar{e}_c = \frac{\hat{I}_c}{\hat{I}_c^{wh}}, \quad (17)$$

kjer je $k_3 = \sqrt{3} k_1 / k_2$. Za poljubno sliko lahko izračunamo \hat{I}_c , ocenimo \hat{I}_c^{wh} ter nato izračunamo \bar{e}_c . Končno oceno izračunamo z normiranjem kot $e_c = \bar{e}_c / \sqrt{\bar{e}_R^2 + \bar{e}_G^2 + \bar{e}_B^2}$.

Za ocenjevanje reflektivnosti \hat{I}_c^{wh} želimo, da je značilnica invariantna na osvetlitev. Barve zato najprej popravimo z enostavnejšo metodo barvne vztrajnosti. V našem delu uporabimo algoritem sivega sveta (*ang. gray world assumption*). Osvetlitev s tem algoritmom ocenimo kot normirano povprečno barvno vrednost slike $e_c^{gwa} = \hat{I}_c$. S to oceno nato popravimo barve kot:

$$\dot{I}_c(\mathbf{x}) = \frac{I_c(\mathbf{x})}{\sqrt{3} e_c^{gwa}}. \quad (18)$$

Iz popravljenih vrednosti izračunamo \dot{L} , \dot{r} in \dot{g} . Te nato uporabimo za izračun histogramov \dot{B}_{rg} ter \dot{B}_{rgL} . Ta histograma sta sedaj invariantna na spremembe osvetlitve po poenostavljenem modelu (1). Nato ju uporabimo za učenje funkcij za oceno \hat{I}_c^{wh} slike:

$$\begin{aligned} \begin{bmatrix} \hat{I}_R^{wh} \\ \hat{I}_G^{wh} \end{bmatrix} &= \hat{\mathbf{I}}_{MLP}^{wh}(\dot{B}_{rg}), \\ \begin{bmatrix} \hat{I}_R^{wh} \\ \hat{I}_G^{wh} \end{bmatrix} &= \hat{\mathbf{I}}_{SVM}^{wh}(\dot{B}_{rg}), \\ \begin{bmatrix} \hat{I}_R^{wh} \\ \hat{I}_G^{wh} \end{bmatrix} &= \hat{\mathbf{I}}_{SVM}^{wh}(\dot{B}_{rgL}). \end{aligned} \quad (19)$$

Te funkcije so podobno kot (8), (11) in (12) naučene s pomočjo MLP in SVM za ocenjevanje reflektivnosti \hat{I}_c^{wh} . Parametri metod strojnega učenja ostanejo isti kot pri ocenjevanju osvetlitve. Modri kanal izračunamo po enačbi (9), kjer e_c nadomestimo z \hat{I}_c^{wh} .

4 Eksperiment

Predstavili smo pristop k barvni vztrajnosti, kjer namesto neposrednega ocenjevanja osvetlitve z algoritmi strojnega učenja poskušamo oceniti povprečno reflektivnost. Za ta namen smo modificirali dve obstoječi metodi ocenjevanja osvetlitve s pomočjo strojnega učenja. V nadaljevanju bomo opisali primerjavo izvornih metod in predlaganega pristopa, zatem pa še predstavili rezultate.

Za primerjavo ocene osvetlitve smo uporabili podatkovno zbirko Gehler [3], ki jo je dodatno predobdelal Shi [7]. Zbirka vsebuje nabor 586 slik naravnih scen. Vsaka slika vsebuje tablico GretagMachbeth, s katero je izmerjena dejanska barva osvetlitve v sliki.

Slike so razdeljene v 3 sete. Dva seta uporabljamo za učenje in enega za testiranje. To ponovimo trikrat, tako da testiramo z vsakim setom enkrat. Pri učenju nevronske mreže pridobimo dodatne učne vzorce z izbiro naključnih podslik, kot je opisano v [6].

Parametre in hiper-parametre obeh metod smo povzeli po izvornih člankih. Uporabljene velikosti histogramov so bile $N = 50$ in $N_L = 15$, pri čemer smo obe vrednosti povzeli po članku [2].

Za MLP smo parameter povzeli po [6]. Vhodni sloj MLP je bil sestavljen iz 2500 nevronov. Prvi skriti sloj je bil sestavljen iz 200, drugi pa iz 40 nevronov. Vsi sloji so bili polno povezani s sosednjimi. Na vhodnem in skritih slojih je bila uporabljena sigmoidna, na izhodnem pa linearna prenosna funkcija.

Za SVM smo parametre povzeli po [5]. SVM z 2D binarnim histogramom je uporabljal jedro RBF (*ang. radial basis function*). Hiper-parametri za obe funkciji so bili nastavljeni na $C = 0, 1$ in $\gamma = 0, 025$. SVM s 3D binarnim histogramom je uporabljal linearno jedro s hiper-parametrom $C = 0, 01$ za obe funkciji.

Natančnost ocenjene osvetlitve smo ocenili s kotno napako [8] izračunano po enačbi:

$$E(\mathbf{e}_{est}, \mathbf{e}_{gt}) = \cos^{-1} \left(\sum_{c \in (R,G,B)} e_{c,est} \cdot e_{c,gt} \right), \quad (20)$$

kjer je \mathbf{e}_{est} ocenjena \mathbf{e}_{gt} pa dejanska barva osvetlitve v sliki. Kotna napaka je kot med barvnima vektorjema \mathbf{e}_{gt} in \mathbf{e}_{est} izražena v stopinjah. Napako nato povzamemo s povprečjem in mediano.

4.1 Rezultati

Rezultate eksperimenta smo podali v tabeli 1. V vseh primerih vidimo, da je ocenjevanje povprečne reflektivnosti vodilo k boljši oceni osvetlitve kot pa neposredno ocenjevanje osvetlitve. Razlika je bila večja pri uporabi nevronske mreže, kjer so tudi rezultati v splošnem bili slabši. Tudi pri uporabi SVM smo izboljšali rezultate.

Tabela 1: Rezultati eksperimenta. V 1. stolpcu je navedena metoda ocenjevanja osvetlitve in reflektivnosti. V 2. in 3. stolpcu sta navedeni povprečna kotna napaka in mediana te napake za celotno zbirko slik. V 4. stolpcu so zbrani rezultati objavljeni v [2].

| | metoda | povprečje | mediana | mediana v [2] |
|----------------------------|--|-----------|---------|---------------|
| ocenjevanje osvetlitve | $\mathbf{e}_{MLP}(B_{rg})$ | 7,67 | 5,83 | 3,77 |
| | $\mathbf{e}_{SVM}(B_{rg})$ | 5,28 | 4,18 | 5,15 |
| | $\mathbf{e}_{SVM}(B_{rgL})$ | 4,12 | 3,32 | 3,23 |
| ocenjevanje reflektivnosti | $\hat{\mathbf{I}}_{MLP}^{wh}(B_{rg})$ | 4,24 | 3,17 | |
| | $\hat{\mathbf{I}}_{SVM}^{wh}(B_{rg})$ | 3,63 | 2,71 | |
| | $\hat{\mathbf{I}}_{SVM}^{wh}(B_{rgL})$ | 3,53 | 2,86 | |

Omeniti še moramo, da se naši dobljeni rezultati razlikujejo od tistih, navedenih v [2]. Z MLP smo sami dosegli veliko slabše rezultate, medtem ko smo za SVM z značilnico B_{rg} ocene malce izboljšali. Najboljši rezultat v [2], dosežen s kombiniranjem rezultatov različnih algoritmov je 1,97. Najboljši rezultat samostojnega algoritma za ocenjevanje osvetlitve pa doseže 3,23 [2].

5 Zaključek

V tem članku smo predstavili pristop k barvni vztrajnosti z algoritmi strojnega učenja. Namesto neposrednega ocenjevanja osvetlitve smo predlagali ocenjevanje reflektivnosti. Rezultati so pokazali, da s takšnim pristopom dosežemo boljše rezultate.

Predlagan pristop ima nekaj ključnih prednosti. Večina značilnic je običajno močno odvisnih od predmetov in njihovih površin v sceni. Iz tega sledi, da z njimi lažje napovedujemo njihove lastnosti. Uporabljeni značilnici v naši metodi pa sta tudi invariantni na

spremembe osvetlitve. To nam dodatno olajša ocenjevanje, saj v sliki z neznano osvetlitvijo, le ta ne bo negativno vplivala na oceno.

V učni zbirki slik naj bi zajeli različne scene. Težko pa bomo v učnih slikah zajeli vse možne kombinacije površin. Kljub temu v obstoječih zbirkah opazimo, da je pestrost scen visoka, medtem ko je pestrost različnih virov osvetlitve običajno nizka. V predlaganem pristopu v učni zbirki slik ne potrebujemo različno osvetljenih scen. To deloma olajša sestavo primerne zbirke.

Pristop predstavljen v tem delu je precej naiven. Napovedovanje povprečne barve v kompleksni sceni zagotovo zahteva podrobnejšo analizo. Algoritmi strojnega učenja se morajo namreč zanašati na precej omejeno količino učnih vzorcev. Dve smeri nadaljnjega razvoja sta v tem trenutku očitni. Namesto ocenjevanja zgolj povprečne reflektivnosti bi lahko ocenjevali podrobnejšo porazdelitev. To bi nam kasneje omogočalo podrobnejše sklepanje o osvetlitvi.

Druga možna nadgradnja pa je, da bi ocenjevanje povprečne reflektivnosti izvedli nad manjšimi regijami v sliki. Tako bi dobili natančnejšo barvno informacijo. Z omejitvami nad osvetlitvami pa bi bili sposobni ocenjevati tudi neuniformne vire osvetlitev.

Literatura

- [1] J. Toro, B. Funt (2007). A Multilinear Constraint on Dichromatic Planes for Illumination Estimation. *Image Processing, IEEE Transactions on*, vol. 16, str. 92–97.
- [2] B. Li, W. Xiong, W. Hu, B. Funt (2014). Evaluating Combinational Illumination Estimation Methods on Real-World Images. *Image Processing, IEEE Transactions on*, vol. 23, str. 1194–1209.
- [3] P. V. Gehler, C. Rother, A. Blake, T. Minka, T. Sharp (2008). Bayesian Color Constancy Revisited. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. str. 1–8.
- [4] H. R. V. Joze, M. S. Drew (2014). Exemplar-based color constancy and multiple illumination. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, str. 860–873.
- [5] W. Xiong, B. Funt (2006). Estimating illumination chromaticity via support vector regression. *Journal of Imaging Science and Technology*, vol. 50, str. 341–348.
- [6] V. C. Cardei, B. Funt, K. Barnard (2002). Estimating the scene illumination chromaticity by using a neural network. *J. Opt. Soc. Am. A*, vol. 19, str. 2374–2386.
- [7] L. Shi, F. B. (2016). Re-processed Version of the Gehler Color Constancy Dataset of 568 Images. Zadnji dostop januar 2016.
URL <http://www.cs.sfu.ca/~colour/data/>
- [8] A. Gijsenij, T. Gevers, M. P. Lucassen (2009). Perceptual analysis of distance measures for color constancy algorithms. *J. Opt. Soc. Am. A*, vol. 26, str. 2243–2256.

NOV ALGORITEM GRAFOV VIDLJIVOSTI V 3D PROSTORU IN NJEGOVA APLIKACIJA ZA ANALIZO SLIK

Matej Babič

Inštitut Jožef Stefan

E-pošta: babicster@gmail.com

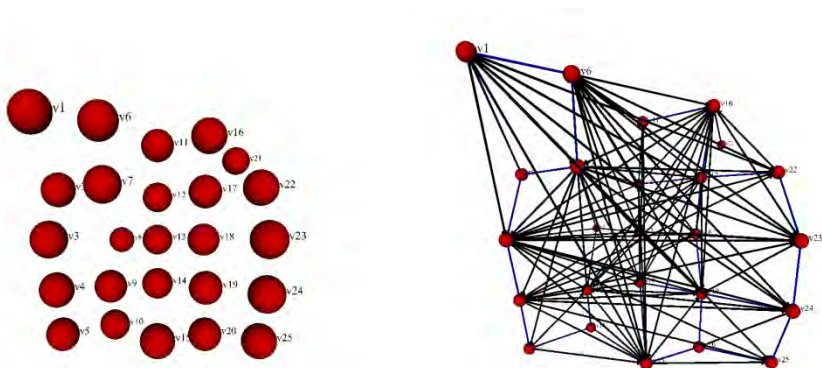
POVZETEK: Tehnike obdelave slik lahko uporabljamo za identifikacijo slikovne funkcije, kot so teksture, gladkost, trdnost, hrapavost in kompleksnost površine. Na površini imamo veliko težav. Eden izmed velikih problemov je, kako analizirati površino in primerjati različne površine na podlagi slik. Karakterizacija površine topografije je pomembna pri aplikacijah, ki vključujejo trenja, grobost in obrabo. Ta dokument opisuje nov algoritem grafov vidljivosti v 3D prostoru. Grafi vidljivost imajo številne aplikacije. Ena izmed njih je analiza trendnih linij grafov na borzi. Novo metodo bomo uporabili za analizo slik topografije površine. Teren je zelo kompleksen. Njegove lastnosti ni mogoče opisati s klasično Evklidsko geometrijo, zato bomo za analizo kompleksnosti terena uporabili metodo fraktalne geometrije.

1. UVOD

V današnjem času je obdelava in analiza slik [1] izjemno pomembna v družbi. Obdelava slik je postala pomemben del sodobne znanosti in tehnologije, saj ima številne pomembne vloge. Najbolj znani metodi za analiziranje slik sta Fourierjeva transformacija in spektralna analiza. Vendar pa so v zadnjih desetletjih številni avtorji izpostavili nove metode, ki so konkurenčne omenjenima metodama. Velik napredek je bil dosežen na področju hidrogeomerfološkega modeliranja kot rezultat razvoja na področju digitalne analize terena, ki ga imenujemo geomorfometrija. Velik vpliv na modeliranje terena je imelo izboljšanje računalniške tehnologije in širok dostop do digitalnih modelov višin reliefa. Cilj analize terena je pojasnilo o razporeditvi zemeljske površine kot tudi njihova klasifikacija, ki temelji na podobnosti površinskega vzorca. Reliefne oblike lahko opredelimo kot posebne geomorfnihi značilnosti na Zemlji, od obsežnimi funkcijami (kot ravninah in gorskih masivov) na manjše možnosti (kot posamezni jarki, okvar in dolin), tako človeške ali iz naravnega geneze, ki so definirane razpon fizičnih in vizualnih lastnosti. Pred uvedbo Digitalnega modela višin, so reliefne oblike le ročno identificirati s pomočjo ankete, ki so bili na voljo preko aeroposnetkov. Od prve svetovne vojne so bile fotografije iz zraka v veliki meri uporabljene za vpogled na sovražnikovo območje. V svojem času je bilo letalo opremljeno s kamerami za

snemanje gibanja sovražnika. Tehnološki napredek v fotogrametriji je povečal polje geomorfoloških raziskav.

V računski geometriji in gibanju robotov, je graf vidljivosti [2] sestavljen iz nizov točk in ovire v evklidski ravnini. Vsako vozlišče v grafu predstavlja točko lokacije in vsak rob predstavlja vidno povezavo med njima. To pomeni, če daljica na dveh lokacijah ne gre skozi oviro, je rob vidljiv v grafu. Vidljivost grafov je bila uporabljena na področju analize časovnih vrst. Grafi 3D vidljivosti so osrednjega pomena za katero koli uporabo v računalniški grafiki. Risanje grafov kot povezava vozlišč v 3D prostoru je vizualno privlačno, vendar računsko težavno. Tako je izgradnja 3D grafov vidljivosti zelo zapletena in zahteva zmogljive računalnike ali superračunalnike. Ta članek opisuje novo metodo za konstruiranje 3D grafov vidljivosti. Nova metoda za gradnjo 3D vidljivosti grafov je zelo uporabna v mnogih primerih, vključno z: osvetlitvijo in upodabljanem, načrtovanjem gibanja, razpoznavanjem vzorcev, računalniško grafiko, računalniško geometrijo in senzorskim omrežjem ter vojaško in avtomobilsko industrijo. Mi bom uporabili to novo metodo za analiziranje topografije terena. Slika 1 prikazuje točke grafa in graf vidljivosti v 3D prostoru.



Slika 1: Graf vidljivosti v 3D prostoru.

Pri analizi naravnih površinskih oblik in gradnji terena zemljevidov za modeliranje imamo veliko nerešenih problemov. Eden od razlogov je, da kompliciranih in nepravilnih oblik, ki jih najdemo v naravi, ne moremo opisati s klasično Evklidsko geometrijo. Zato je Mandelbrot [3] predlagal pojem fraktalov kot družino matematičnih funkcij za opis naravnih pojavov kot so obala, gore, razvejani vzorci dreves in rek, oblaki in potresi. Ugotovili so, da lahko s pomočjo fraktalne geometrije opišemo mnogo drugih kompleksnih sistemov v naravi, tehniki in medicini. Zaradi tega odkritja, so fraktali pritegnili veliko pozornost znanstvenikov. Fraktalna dimenzija je lastnost fraktala, ki se ohranja prek vseh povečav in je zato dobro definirana, poleg tega nam pa tudi pove, kako kompleksen je fraktal. Fraktalno dimenzijo izračunamo po formuli (1)

$$M_D(E) = \lim_{\epsilon \rightarrow 0} \inf_{r_k < \epsilon} \sum r_k^D \quad (1)$$

kjer so r_1, r_2, \dots, r_k radiji krogel (krožnic) s katerimi pokrivamo množico E . M_D je mera množice v D dimenzijah in je ekvivalentna dolžini pri $D=1$, ploščini pri $D=2$ itn.

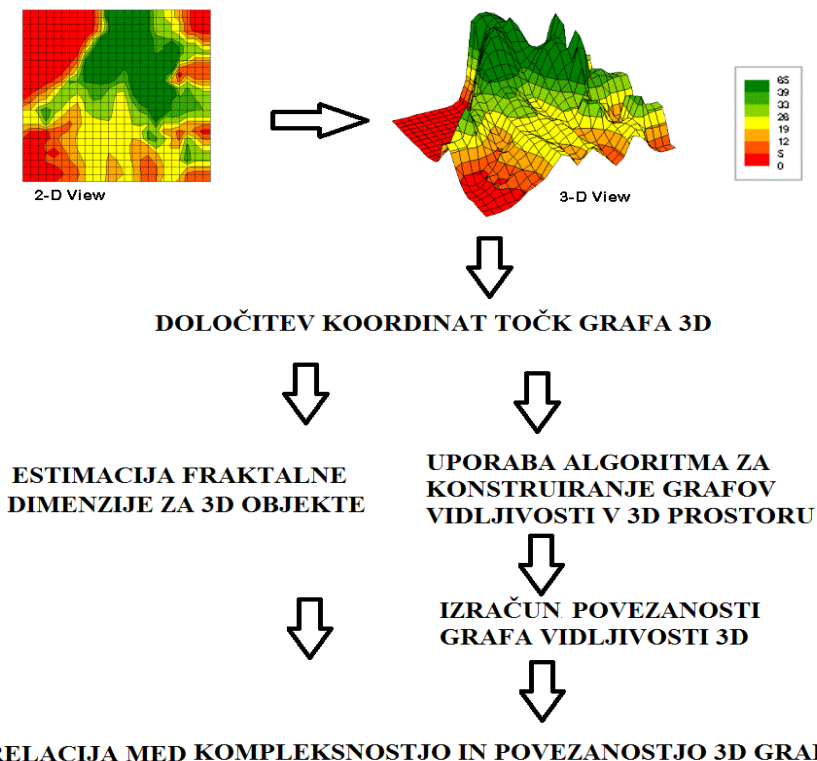
Fraktalne dimenzije se v splošnem ne da računati po zgoraj navedenem postopku, saj je to mogoče le na čistih matematičnih konstruktih, ki pa jih v naravi ni. Kljub veliki natančnosti je zgornja definicija za fiziko bolj ali manj neuporabna, saj ε ne more biti poljubno majhen, pa tudi fraktale opazimo samo v določenem velikostnem območju. V praktične namene se za določanje dimenzije največ uporablja metoda "štetja škatel" (angl. box-counting dimension), kjer preučevani fraktal pokrijemo s kvadratno mrežo, ki jo nato vse bolj manjšamo in opazujemo spreminjanje števila kvadratkov, potrebnih, da pokrijemo celotno množico. Rezultat je seveda približek, ki ga izračunamo na želeno število mest. Fraktal je skupek točk katerih je fraktalna dimenzija večja od topološke dimenzije. Fraktalna analiza je uporabna ravno takrat, ko nam klasična geometrija ne da dovolj uporabnih rezultatov za opis iregularnih objektov. Fundamentalna lastnost fraktalov je fraktalna dimenzija D , ki da pomemben pogled na fizikalne lastnosti različnih materialov. Fraktalna geometrija je nov pristop, ki temelji na karakterizaciji nepravilnih mikrostruktur in služi kot ocenjevalno orodje za določitev strukturnih lastnosti in jo bomo uporabili pri analizi terena. Uvedena je bila kot karakteristična značilnost kompleksnih naravnih in abstraktnih sistemov. Pri analizi terena moramo upoštevati, da je fraktalna struktura terena statistično samo-afina. Slika 2 prikazuje primer statistično samoafine strukture terena.

2. METODOLOGIJA

Slike reliefa, formata (npr. jpeg) smo pretvorili v matriko barv na intervalu [1-256] s programom ImageJ in naredili dvodimenzionalni graf. V vsaki točki (x,y) slike (2D ravnina) določimo vrednost od 1 do 256. Ta vrednost določa tretjo koordinato v 3D koordinatnem sistemu oz. z-koordinato. To pomeni, da točka $T=(x,y)$ z ravnine dobi tretjo komponento z in ima potem obliko $T_{3D}=(x,y,z)$. Na podlagi točk $T_{3D}=(x,y,z)$ smo določili fraktalno dimenzijo reliefa. Kot osnova za izračun fraktalne dimenzije se uporablja Hurstov eksponent H . Ime nosi po britanskem arhitektu in hidrologu H. E. Hurstu, ki je sodeloval pri projektu gradnji jezov na reki Nil v začetku 20. stoletja. Iskal je rešitev tipičnega hidrološkega problema, tj. zahtevano kapaciteto vodnega rezervoarja jeza. Naključni proces je statistično ovrednoten s pomočjo Hurstovega parametra H ali z določanjem porazdelitvene funkcije. Hurstovega parametra H kot merilo samopodobnosti ni mogoče natančno izračunati, ampak samo oceniti. Obstaja več različnih metod za oceno Hurstovega parametra H . Pri tem nimamo merila za določitev, katera metoda daje najboljši rezultat. Uporabili smo grafično metodo prilagojeni razpon R/S , ki temelji na lastnostih Hurstovega pojava. S to metodo smo določili kompleksnost terena po formuli $D=3-H$.

Za dobljen graf smo uporabili novo metodo, tj. algoritem za konstruiranje grafov vidljivosti v 3D prostoru [4]. Za grafe vidljivosti smo izračunali topološko lastnost povezanost grafa. Topološka povezanost grafov pomeni, koliko povezav ima graf glede na vse možne povezave. Slika 2 prikazuje postopek analize slik.

KONVERTIRANJE SLIKE RELIEFA V 3D



Slika 2: Analiza slik.

3. REZULTATI IN DISKUSIJA

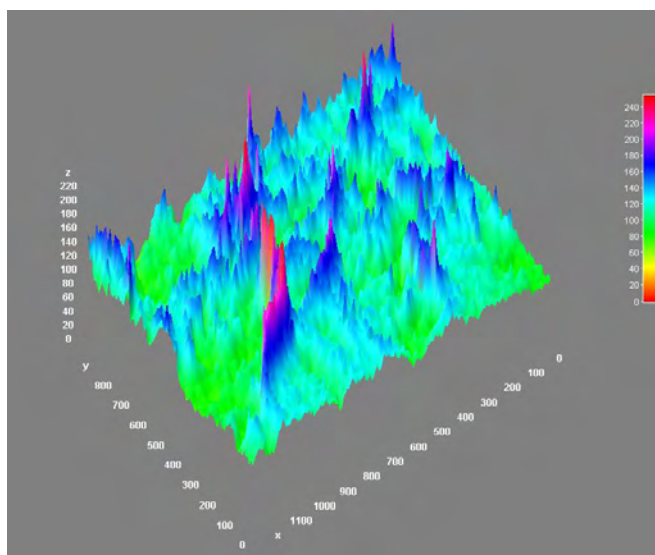
Slike terena smo izbrali naključno s spleta. Prvi stolpec v Tabeli 1 prikazuje vrednost fraktalne dimenzije, drugi stolpec pa topološko lastnost povezanost grafov vidljivosti v 3D prostoru. Najbolj kompleksen vzorec je teren št. 11. Teren št. 11 ima fraktalno dimenzijo 2,485. Teren št. 17 ima najvišjo povezanost med vozlišči 0,2832. S pomočjo teorije grafov, algoritma za konstruiranje grafov vidljivosti v 3D prostoru, smo določili mrežo za 18 različnih terenov. Ker je teren odvisen od koordinat točk dobimo mreže v odvisnosti od teh koordinat. Prav tako je kompleksnost odvisna od koordinat točk terena. Enačba (2) prikazuje linearni model, relacijo med kompleksnostjo in povezanostjo grafov vidljivosti v 3D za različne terene.

$$Y=0.128 \times X - 0.087 \quad (2)$$

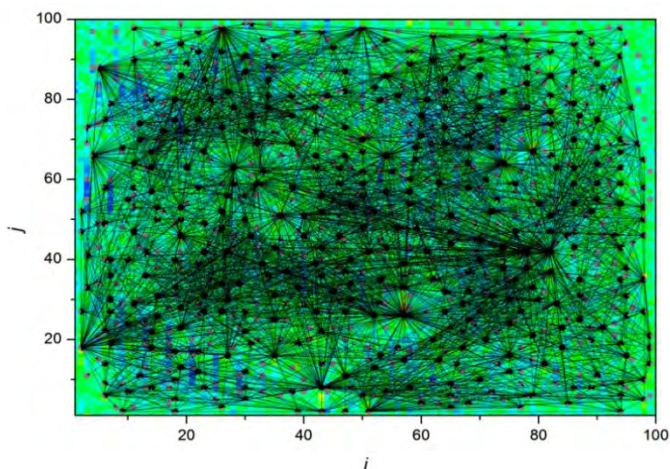
Od dobrega terena pričakujemo, čim manjšo kompleksnost in čim manjšo povezanost grafa vidljivosti v 3D grafa. Najbolj primeren teren v našem primeru je teren št. 16.

Tabela 1: Relacija med kompleksnostjo terena in topološko lastnostjo grafov vidljivosti v 3D.

| <i>TEREN</i> | <i>KOMPLEKSNOST</i> | <i>POVEZANOST</i> |
|--------------|---------------------|-------------------|
| 1 | 2,423 | 0,1936 |
| 2 | 2,424 | 0,2208 |
| 3 | 2,430 | 0,2144 |
| 4 | 2,418 | 0,2256 |
| 5 | 2,431 | 0,2445 |
| 6 | 2,434 | 0,2221 |
| 7 | 2,439 | 0,2036 |
| 8 | 2,460 | 0,2096 |
| 9 | 2,450 | 0,2352 |
| 10 | 2,439 | 0,2288 |
| 11 | 2,485 | 0,2144 |
| 12 | 2,456 | 0,2352 |
| 13 | 2,404 | 0,2208 |
| 14 | 2,327 | 0,2320 |
| 15 | 2,450 | 0,1984 |
| 16 | 2,380 | 0,1904 |
| 17 | 2,441 | 0,2832 |
| 18 | 2,477 | 0,2688 |



Slika 3: Teren št. 5.



Slika 4: Graf vidljivosti za teren št. 5.

3. ZAKLJUČEK

V članku smo uporabili novo metodo grafov vidljivosti v 3D prostoru. Vsak teren ima določene koordinate v 3D prostoru. S pomočjo analize slik smo opisali lastnosti različnih terenov. Struktura 3D terena je zelo kompleksna in jo je nemogoče opisati s klasičnimi metodami matematike. Zato smo za analizo kompleksnosti slik, ki predstavljajo različne terene, uporabili metodo za estimacijo fraktalnih dimenzij 3D objektov. Nov algoritem grafov vidljivosti pa smo uporabili za določitev povezanosti točk terena v 3D prostoru. Predstavili smo tudi relacijo med kompleksnostjo in povezanostjo terena.

LITERATURA

1. Anne MerieSvane (2015). Estimation of Minkowski tensors from digital grey-scale images. *Image Anal Stereol* 34:51-61.
2. Ben-Moshe, O. Hall-Holt, M.J. Katz and J.S.B. Mitchell. (2004). *Computing the Visibility Graph of Points within a Polygon*, Proc. 20th ACM Symp. on Computational Geometry, 27-35.
3. Barton, C. C., 1995, Fractal analysis of scaling and spatial clustering of fractures. In: Barton, C.C. and La Pointe, P.R. (eds.), *Fractals in the earth sciences*. Plenum Press. New York, p. 141–178.
4. Matej Babič. *Analiza kaljenih materialov s pomočjo fraktalne geometrije : doktorska disertacija*. [Maribor: M. Babič], 2014. 164 f., ilustr.

Real Time Pellet Size Distribution using a DSP Embedded Platform

Jagdish Lal Raheja, Arun Rajsekhar Gadula

Digital Systems Group, Central Electronics Engineering Research Institute
(CEERI)/Council of Scientific and Industrial Research (CSIR), Pilani 333031

INDIA

e-mail: jagdish@ceeri.ernet.in

ABSTRACT: *A real time pellet size distribution using a DSP platform has been implemented in this work. The proposed system has been developed using the TI's (Texas Instruments) DM642 EVM (Digital Media 642 Evolution Module) board which is a standalone Digital Media Processing board. Iron ore pellets are used in steel production and to make the production more reliable and energy efficient. It is necessary to check the size distribution during the formation of the pellets. Size measurement of pellet in industry is usually performed by manual sampling and sieving techniques. The manual sampling is performed infrequently and is time-consuming. Fast feedback of pellets sizes is desirable and this research work reports the automatic size distribution measurement in real time using image processing.*

Keywords: Image Preprocessing, Edge Detection, Circular Hough Transform, Pellet Size Analysis, Estimation of Scaling Factor.

1. Introduction

In steel production, using iron ore pellets instead of ordinary ore makes more reliable and energy efficient. Pellet size must be checked during pellet production and in steel production when the blast furnace is loaded. Size measurement of pellet in industry is usually performed by manual sampling and sieving techniques. Manual sampling followed by sieving with a square mesh is used for quality control. The manual sampling is performed infrequently and is time-consuming. Fast feedback of pellets sizes is desirable.

In our approach, Pellet Size Distribution (PSD) was done using image processing algorithm using Circular Hough Transform (CHT) [1] [4] [6] implemented on a DSP embedded board, a Texas Instruments TMS320DM642 EVM [8]. It is used for the real-time implementation. It's operating at 720 MHz. This processor is a fixed-point digital media processor and optimized for video and image applications. . This provides an 8 fold increase in speed over PC implementations. Robust real time performance is thus made possible, and this increases the scope of applications to include ones where high frame rates are necessary.

2. System Architecture

The system consists of a central computer, which is essentially a computer connected with a PC, which, in turn, connected to a DSP Processor. An actual experimental setup is shown in figure 1. System includes the following hardware

1. Bosch Dinion Colour Camera.
2. DM642 EVM Processor.
3. PC

When the Pellets were formed on the pelletizer disk, its image was captured by a high speed camera.

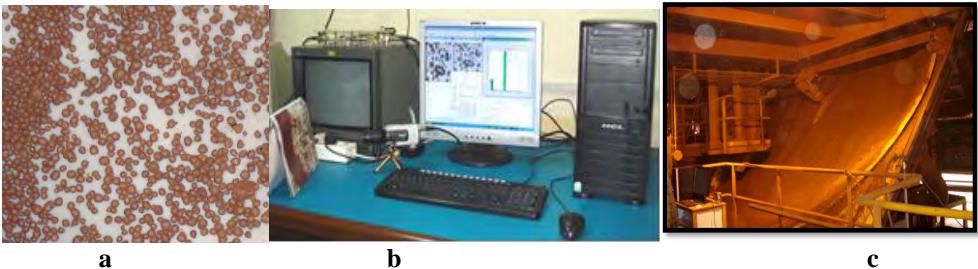


Figure 1(a) Pellets (b) Experimental setup in Lab (c) Pelletizer Disk

Figure 2 shows the block diagram of the whole setup consisting of a high resolution camera and a TI DM642 EVM DSP board as the target which is interfaced with a host PC using an Ethernet connection [13].

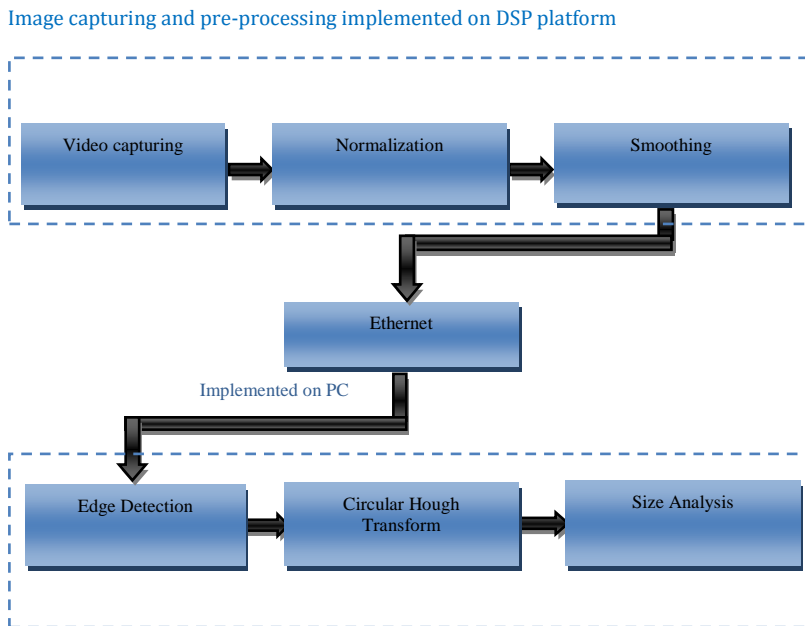


Figure 2: Block Diagram

The procedure starts with image acquisition through the camera and a grayscale image was taken. To prepare the image for further processing, preliminary steps such as contrast enhancement through normalization and noise removal through smoothing were carried out. These were implemented on the DSP board for faster performance speeds. Subsequent implementation of the remaining steps were done on the host PC. It included edge detection of the image, whereby the boundaries of the pellets were extracted. Pellets are assumed to be near spherical objects, and their detection may be done with the aid of the Circular Hough Transform. By applying this technique we obtained a representation of pellets as circles lying within a specified radius range.

3. Methodology

The preprocessing involved four steps which are Image pre-processing, Edge Detection, Circular Hough Transform, Pellet size Analysis, Estimation of Scaling Factor and are discussed briefly in the following paragraphs.

3.1. Image Pre-processing

Image was captured by a high resolution camera and image preprocessing techniques were applied on it. Two important pre-processing techniques i.e. smoothing and normalization reside on DSP processor (**DM642EVM**). After these two operations, image was transferred to PC using UDP target to HOST Ethernet communication. The overall setup consisted of a high resolution smart camera and a TI DM642 EVM DSP board as the target which was interfaced with a host PC using an Ethernet connection [13]. The procedure starts with a grey scale image acquisition through the smart camera.

3.2. Normalizing the gray scale image

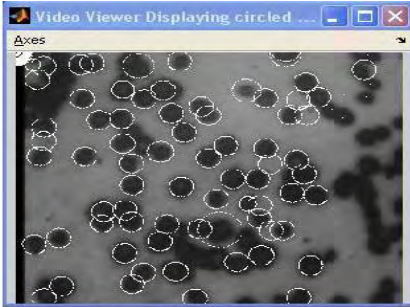


Figure 3 (a): Without Normalization and Smoothing



Figure 3(b): With Normalization and Smoothing.

The above Figure 3(a) is without normalization and without smoothing and Figure 3(b) is with normalization and with smoothing and after detecting the circles.

3.3. Normalization

In order to compensate the effect of different light conditions, light normalization of the image is performed prior to it is inputting to the CHT algorithm. After normalization the intensity of light at each and every point is equally distributed. The normalization was also implemented on the DSP board.

3.4. Smoothing

The smoothing is done to remove the small noise. This is accomplished by convolution, where a Gaussian filter removes ('denoises') noisy pixels to give a smoothed output image. Smoothed image is then transferred to the PC, via Ethernet port. Ethernet port on the DM642

EVM provides the connection between the target TI's DM642 Processor and the host PC. DM642 EVM provides 10/100 Mbps interface.

3.5. Edge Detection

To identify given pellets radius, it is necessary to find the edges of the boundaries of the pellets, including overlapping pellets. These edges are nothing but the significant local changes of intensity in an image and mostly occur on the boundary between two different regions in an image. A canny edge detection technique is used to find the edges of the image.

3.6. Circular Hough Transform

The CHT [1], [2], [4], aims to find circular patterns within an image and was implemented as Simulink block as shown in figure 5. The CHT is used to transform a set of feature points in the image space into a set of accumulated votes in a parameter space. Circular Hough Transform based on the gradient field of an image. A fundamental property of the gradient vector is that it points in the direction of the maximum rate of change of field *at* point coordinates (x, y) .

After finding all edges in the image, a set of circles are defined in the accumulator space [7] an edge point. These circles are defined by all possible values of the radius and they are centered on the coordinates of the edge point this circle is drawn in the parameter space, such that our x axis is the a-value and the y axis is the b-value while the z-axis is the radii. These circles are defined for a given radius value. Each edge point defines circles for the other values of the radius. This implies that the accumulator space is three dimensional (for the three parameters of interest) and that edge point's map to a cone of votes in the accumulator space. After gathering evidence of all the edge points, the maximum in the accumulator space again corresponds to the parameters of the circle in the original image.

4. Pellet Size Analysis

To estimate the size of the iron ore pellets, image of the pellets has taken. The shape of the pellets is assumed spherical. To find the size of each pellet and then the whole distribution of pellets, we have to find the exact circles of the spheres. From those circles the radius and centre has to be determined.

4.1. Estimation of Scaling Factor

Figure a&b are the examples how we estimation of scaling factor. We converted the radius values of the pellets calculated by the system in units of pixels into millimeters by determining the 'scaling factor' of the system in following steps

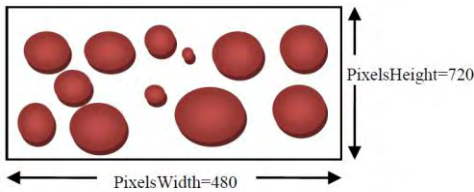


Figure 4(a): Captuted Image

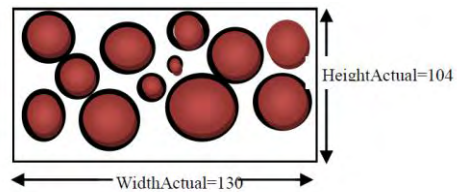


Figure 4(b) Circles Detected

1. The camera resolution denotes the number of pixels used to represent each dimension. For example a 720x480 resolution refers to 720 pixels available along the width and 480 along the height. We denote this as HorzPixels and VertPixels, as can be seen in Figure a.

2. The actual height and width in millimeters of the pan holding the pellets is represented as HeightActual and WidthActual, as can be seen in Figure 4(a) & 4(b).

3. Scaling factors along each direction are calculated as

$$\text{RatioHorz}=\text{WidthActual}/\text{HorzPixels}; \quad \text{RatioVert}=\text{HeightActual}/\text{VertPixels}$$

Either of the two ratios may be used for further calculation.

4. The actual pellet radius may be calculated from the radius in pixels estimated by the system as

$$\text{RadiusActual}=\text{Ratio} \times \text{RadiusPixels}$$

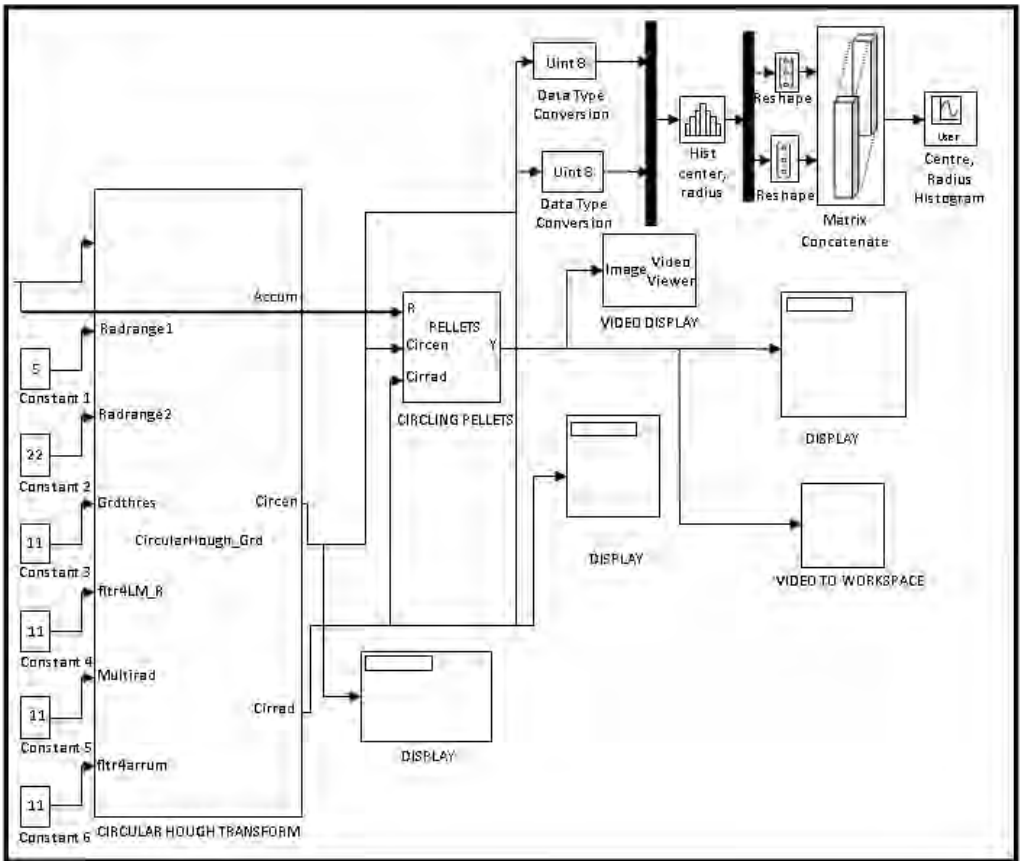


Figure 5: Main Algorithm Simulation Block

5. Experimental Results

The Figure 6 shows the results of the image with and without normalisation with their respective histogram plots. In the histogram, we can see how many circles found with radius in mm.

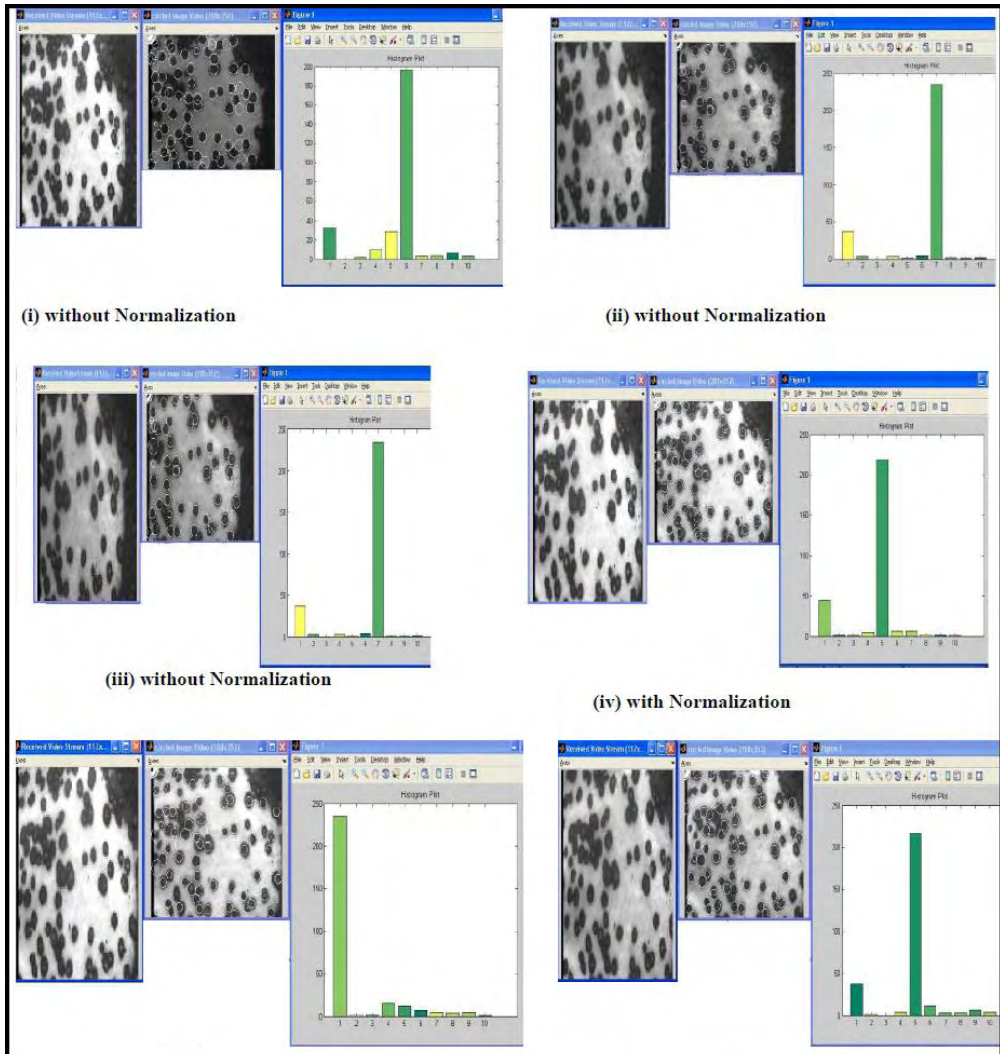


Figure 6: Results

6. Conclusion

An approach based on real time image processing was implemented to find the PSD in real time. Images are acquired using a Bosch Dinion Color Camera, and the algorithms are executed on a TMS320DM642EVM. This provides an 8 fold increase in speed over PC implementations. Robust real time performance is thus made possible, and this increases the scope of applications to include ones where high frame rates are necessary.

Code Composer Studio which is a programming environment used to implement various algorithms and load code onto a Digital Signal Processor (DSP). We have use Code Composer Studio version 3.3.

Acknowledgement

This research has been carried out at CSIR-Central Electronics Engineering Research Institute (CEERI), Pilani, India and Authors would like to thank Director, CSIR-CEERI for his active encouragement and support.

References

1. Atherton T. J. and Kerbyson D. J. (1993). The coherent circle Hough transform, Proceedings of the British Machine Vision Conference, Guildford, UK, pp. 269–278.
2. Atherton T. J. and Kerbyson D. J. (1993). Using phase to represent radius in the coherent circle Hough transform, *IEE* Colloquium on the Hough Transform **5**: 1–4.
3. Chan T. F. and Vese L. A. (2001). Active contours without edges, *IEEE Transactions on Image Processing* **10**(2): 266–277.
4. Duda R. O. and Hart P. E. (1972). Use of the Hough transform to detect lines and curves in pictures, *Communications of the ACM* **15**(1): 11–15.
5. Jack Bresenham, A linear algorithm for incremental digital display of circular arcs. *Communications of the ACM* February, Volume 20, Number 2, February 1977.
6. Chester F. Carlson, Lecture 10: Hough circle transform. Rochester Institute of Technology: Lecture Notes, October 11, 2005.
7. Kimme, C., D. Ballard and J. Sklansky, 1975. Finding circles by an array of accumulators. *Proc. ACM*, **18**: 120-122.

8. Texas Instruments Incorporated. TMS320DM642 Video/Imaging Fixed-Point Digital Signal Processor, SPRS200L - July 2002 - Revised January 2007
9. Texas Instruments Incorporated. TMS320C6000 Technical Brief, February 1999. Literature Number: SPRU197D.
10. S. Qureshi, Embedded Image Processing on the TMS320C6000TM DSP, Springer, New York, NY, USA, 1st edition, 2005.
11. Texas Instruments Incorporated. TMS320C6000 Optimizing Compiler User's Guide, May 2004. Literature Number: SPRU187L.
12. Texas Instruments Incorporated. TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide, June 2005. Literature Number: SPRU732A.
13. Texas Instruments Incorporated. Getting Started With the C6000 Network Development Kit (NDK), SPRAAX4 - July 2008.
14. John Stevenson, Texas Instruments Code Composer Studio IDE v3 White Paper, Application Report, and Literature Number: SPRAA08 - July 2004.
15. Texas Instruments, TMS320C6000 DSP/BIOS Application Programming Interface (API) Reference Guide, Literature Number: SPRU403G, April 2004.

MEDICINSKE IN BIOMEDICINSKE APLIKACIJE

AVTOMATSKA METODA NAČRTOVANJA VELIKOSTI IN TRAJEKTORIJE PEDIKULARNIH VIJAKOV

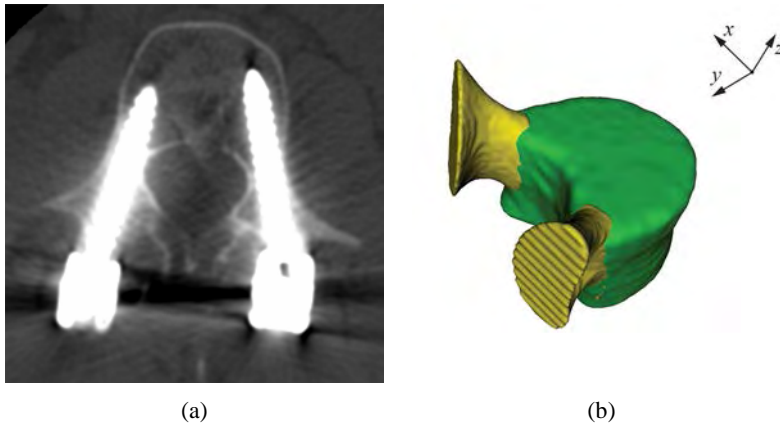
Dejan Knez, Boštjan Likar, Franjo Pernuš, Tomaž Vrtovec

Laboratorij za slikovne tehnologije
Fakulteta za elektrotehniko, Univerza v Ljubljani
E-pošta: dejan.knez@fe.uni-lj.si
URL: <http://lit.fe.uni-lj.si/>

POVZETEK: *Stabilizacija vretenc z vstavitvijo pedikularnih vijakov skozi pedikle ter medsebojno pritrditvijo s povezovalnimi palicami je ena izmed najbolj razširjenih metod za operativno zdravljenje deformacijskih stanj hrbtenice. Zaradi bližine živčevja in organov je natančnost vstavitve pedikularnih vijakov neposredno povezana z uspešnostjo operacije. Ker je ročno predoperativno načrtovanje pedikularnih vijakov časovno zamudno, so bile razvite avtomatske metode za načrtovanje pedikularnih vijakov, vendar brez upoštevanja pritrdilne moči vijaka. Predlagamo novo avtomatsko metodo za optimalno načrtovanje velikosti in trajektorije pedikularnih vijakov, ki upošteva tako geometrijske (t.j. morfometrija) kot tudi anatomske (t.j. kostna mineralna gostota) lastnosti vretenca pri iskanju maksimalne pritrdilne moči. Predlagano avtomatsko metodo smo ovrednotili na računalniško tomografskih slikah devetih pacientov, kjer smo določili velikosti in trajektorije 61 pedikularnih vijakov. Avtomatsko načrtovani vijaki so se od ročno načrtovanih razlikovali za 0,6 mm v premeru, 4,0 mm v dolžini, 1,7 mm v presečiščni točki pedikla ter $6,1^\circ$ v naklonih, medtem ko je bila pritrdilna moč večja pri 50 pedikularnih vijakih (82%).*

1. UVOD

Stabilizacija vretenc s spinalno fuzijo je najbolj razširjena metoda, ki se trenutno uporablja pri operativnem zdravljenju deformacijskih stanj hrbtenice, kot so npr. skolioza, kifoza in zlom vretenca [1]. Spinalna fuzija je dosežena z vstavitvijo pedikularnih vijakov skozi pedikle v telesa vretenc (slika 1(a)) ter medsebojno pritrditvijo s povezovalnimi palicami, kar rezultira v zmanjšani kompresiji na poškodovano vretenca. Natančnost vstavitve pedikularnih vijakov je neposredno povezana z uspešnostjo operacije ter je posledično ključnega pomena, ker lahko napačno vstavljeni pedikularni vijaki vodijo do resnih poškodb živčevja in organov [1]. V izogib temu ima predoperativno načrtovanje pomembno vlogo za varno vstavitve pedikularnih vijakov [2,3], kjer kirurgi med načrtovanjem preučujejo anatomijo hrbtenice pacienta na



Slika 1: (a) Primer uspešno vstavljenih pedikularnih vijakov na postoperativnem sredinskem prečnem prerezu. (b) Primer tridimenzionalnega parametričnega modeliranja vretenčnega telesa ter pripadajočih pediklov.

predoperativnih slikah, v današnjem času običajno na tridimenzionalnih (3D) računalniško tomografskih (CT) slikah.

Ročno načrtovanje velikosti in trajektorije pedikularnih vijakov je časovno zamudno, poleg tega pa je pri ročnem načrtovanju praktično nemogoče upoštevati vse parametre, kot je npr. pritrdilna moč vijaka. Kirurgi pri načrtovanju stremijo k maksimalni vrednosti pritrdilne moči, ki je neposredno povezana s kostno mineralno gostoto (BMD) [2,4]. Ročna (vizualna) določitev BMD je prav tako zahtevno in časovno zamudno opravilo, ker zahteva natančno analizo večjega števila prerezov 3D slike v različnih ravninah pogleda. Takšna analiza mora biti ponovljena za vsako spremembo velikosti in/ali trajektorije pedikularnega vijaka. V ta namen so bila razvita različna vodila in navigacijske tehnike za računalniško podprto kirurgijo [1,5,6]. Nekaj študij je bilo usmerjenih tudi v avtomatsko načrtovanje pedikularnih vijakov, ki temelji na analizi predoperativnih CT slik [3,7], vendar brez upoštevanja pritrdilne moči. Z razvojem računalniško podprte kvantitativne analize 3D slik je lahko pritrdilna moč določena z BMD za vsako velikost in trajektorijo pedikularnega vijaka. Kombinacijo velikosti in trajektorije pedikularnega vijaka, ki doseže maksimalno pritrdilno moč, je mogoče doseči s postopkom optimizacije. Ker se z navedenim zmanjša subjektivno načrtovanje kirurga, se s tem poveča zanesljivost predoperativnega načrtovanja.

V prispevku predlagamo novo metodo predoperativnega avtomatskega načrtovanja optimalne velikosti in trajektorije pedikularnega vijaka. Predlagana metoda stremi k maksimalni vrednosti pritrdilne moči z upoštevanjem tako geometrijskih (t.j. morfometrija) kot tudi anatomskih (t.j. BMD) lastnosti vretenc. Vretenca so izločena iz predhodno nerazgrajenih 3D CT slik pacientov, ki so bile pridobljene predoperativno za

namen ročnega načrtovanja pedikularnih vijakov pri operativnem zdravljenju s sistemom pedikularnih vijakov in povezovalnih palic.

2. METODOLOGIJA

Za ustrezno načrtovanje pedikularnih vijakov je potrebno znanje o 3D anatomiji telesa vretenca ter pedikla. Velikost (t.j. premer, dolžina) in trajektorija (t.j. presečiščna točka pedikla in nakloni) pedikularnega vijaka sta določeni tako, da je pedikularni vijak popolnoma znotraj telesa vretenca ter pedikla (pedikularni vijak ni izven kostnih struktur), ne prebije sprednjega dela telesa vretenca ter se ne seka s pedikularnim vijakom, ki je lahko vstavljen skozi sosednji pedikel (slika 1(a)). Načrtovanje pedikularnih vijakov je tako mogoče z ustreznim modeliranjem vretenčnih struktur in pedikularnih vijakov v 3D na podagi kvantitativne analize 3D slik.

2.1 Modeliranje vretenca v 3D

Pri modeliranju vretenčnih struktur smo se omejili na modeliranje telesa vretenca ter pedikla s parametričnima superkvadrikoma [8]. Parametrična oblika superkvadraka (t.j. generalizirana kvadrična površina v 3D) neposredno določa ali poljubna točka leži znotraj, izven ali na površini superkvadraka, kar daje prednost pri umestitvi pedikularnega vijaka popolnoma znotraj vretenčnega telesa ter pedikla. Poleg tega so parametri superkvadraka neposredno povezani z morfometrijo pripadajoče anatomske strukture.

2.1.1 Parametrično modeliranje vretenčnega telesa

Superkvadrik začetne 3D oblike telesa vretenca $V_i(\mathbf{x})$ je predstavljen z eliptičnim valjem [8]:

$$V_i(\mathbf{x}) = \left(\frac{x^2 + y^2}{r(\theta)^2} \right)^{10} + \left(\frac{z}{h} \right)^{20}, \quad (1)$$

$$r(\theta) = \frac{ab}{\sqrt{(a \sin \theta)^2 + (b \cos \theta)^2}}, \quad (2)$$

kjer je $\mathbf{x} = (x, y, z)$ točka v 3D prostoru, $r(\theta)$ predstavlja sled elipse z veliko in malo polosjo a in b , $\theta = \arctan(y/x)$ je radialni kot ter h polovica višine valja. Bolj natančna predstavitev 3D oblike telesa vretenca $V_d(\mathbf{x}) = T_v(V_i(\mathbf{x}))$ je dosežena z vpeljavo dodatnih 22 parametrov, ki določajo transformacijo T_v ter predstavljajo specifične 3D anatomske deformacije telesa vretenca [8], t.j. (1) oblika eliptičnega valja na mestu levega in desnega pedikla, vretenčnega kanala ter sprednjega dela telesa

vretenca, (2) konkavnost sten vretenčnega telesa na mestu vretenčnega kanala in sprednjega dela telesa vretenca, (3) konkavnost in prečni naklon krovnih plošč telesa vretenca in (4) spreminjanje velikosti in torzija telesa vretenca. Končna predstavitev 3D oblike telesa vretenca $V(\mathbf{x}) = R_V(V_d(\mathbf{x}))$ je dosežena s togo poravnavo R_V na 3D sliko (slika 1(b)). Za določitev T_V in R_V je uporabljen podobnostni kriterij, ki je sestavljen iz dveh komponent, s katerima se upošteva lastnosti CT slike [8]. Komponenta intenzitet C_I temelji na intenzitetah 3D slike in maksimira vrednost kostne strukture znotraj 3D modela in vrednost mehkih tkiv izven 3D modela:

$$C_I = \sqrt{1 - \sum_s \sqrt{p_{in}(s)p_{out}(s)}}, \quad (3)$$

kjer sta p_{in} in p_{out} verjetnostni porazdelitvi slikovnih intenzitet s znotraj 3D modela in volumna, ki obdaja 3D model. Komponenta oblike C_G temelji na gradientih intenzitet 3D CT slike in maksimira njihovo prileganje z normalami na površini 3D modela:

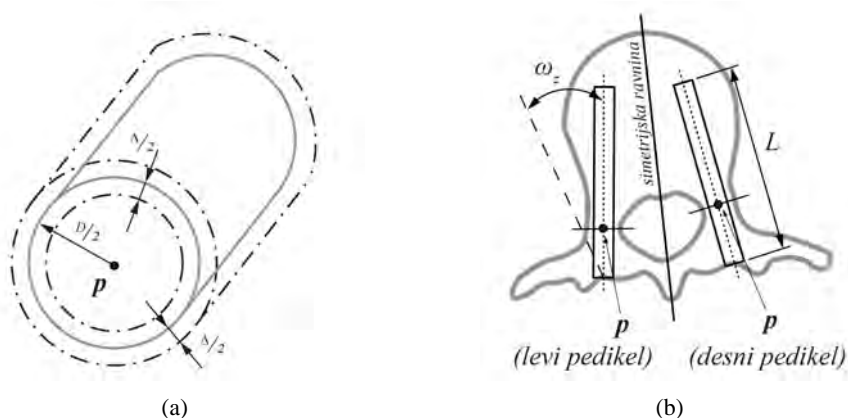
$$C_G = \sum_{\mathbf{x} \in V_{in}} \left(\langle \mathbf{g}(\mathbf{x}), \mathbf{n}_{in}(\mathbf{x}) \rangle \frac{e^{-\frac{d(\mathbf{x})^2}{2}}}{\sqrt{2\pi}} \right) \cdot \sum_{\mathbf{x} \in V_{out}} \left(\langle \mathbf{g}(\mathbf{x}), \mathbf{n}_{out}(\mathbf{x}) \rangle \frac{e^{-\frac{d(\mathbf{x})^2}{2}}}{\sqrt{2\pi}} \right), \quad (4)$$

kjer je $d(\mathbf{x})$ evklidska razdalja med točko \mathbf{x} in površino 3D modela, $\langle \mathbf{g}(\mathbf{x}), \cdot \rangle$ je skalarni produkt normaliziranih gradientov intenzitet 3D slike $\mathbf{g}(\mathbf{x})$ in enotskih normal površine 3D modela \mathbf{n}_{in} ali \mathbf{n}_{out} usmerjenih navzven ali navznoter, V_{in} in V_{out} pa sta volumna, ki se razprostirata navznoter oz. navzven 3D površine modela. Optimalni parametri transformacij T_V in R_V , ki predstavljata anatomske deformacije oz. togo poravnano, so dobljeni s postopkom iskanja maksimuma kombinacije komponente intenzitet in oblike, ki določata kriterij podobnosti:

$$\{T_V, R_V\} = \arg \max_{\{T, R\}} \left((C_I C_G) \Big|_{\{T, R\} \rightarrow V(\mathbf{x})} \right). \quad (5)$$

2.1.1 Parametrično modeliranje pedikla

Superkvadrik začetne 3D oblike pedikla $P_i(\mathbf{x})$ je predstavljen z eliptičnim valjem (enačba 1) in je avtomatsko določen iz parametrov poravnane 3D modela vretenčnega telesa $V(\mathbf{x})$, in sicer na mestu levega (ali desnega) pedikla. Bolj natančna predstavitev 3D oblike pedikla $P_d(\mathbf{x})$ je dosežena z vpeljavo 22 dodatnih parametrov, ki določajo transformacijo T_p ter predstavljajo specifične 3D anatomske deformacije pedikla, t.j. (1) odmik notranje oblike pedikla, (2) konkavnost na zgornjem, spodnjem, levem in desnem delu pedikla, (3) konkavnost prereza pedikla v smeri navzgor-navzdol ter levo-desno in (4) deformacija prereza pedikla v obliki solze. Končna predstavitev 3D oblike pedikla



Slika 2: Model pedikularnega vijaka, prikazan v (a) tridimenzionalnem pogledu in (b) sredinskem prečnem prerezu vretenčnega telesa, je določen s premerom D , odmikom Δ okoli premera, dolžino L , presečiščno točko pedikla \mathbf{p} ter naklonoma ω_x in ω_z v stranski in prečni ravnini.

$P(\mathbf{x}) = R_p(P_d(\mathbf{x}))$ je dosežena s togo poravnavo R_p na 3D sliko (slika 1(b)). Optimalni parametri transformacije T_p in R_p so določeni s postopkom iskanja maksimuma kombinacije komponente intenzitet in oblike, ki določata kriterij podobnosti $(C_I C_G) \Big|_{\{T, R\} \rightarrow P(\mathbf{x})}$ (enačbe (3)-(5)).

2.2 Modeliranje pedikularnega vijaka v 3D

Superkvadrik 3D oblike modela pedikularnega vijaka $S_i(\mathbf{x})$ je predstavljen s krožnim valjem, ki definira velikost pedikularnega vijaka, t.j. premer in dolžina. Končna predstavitev pedikularnega vijaka $S(\mathbf{x}) = R_S(S_i(\mathbf{x}))$ je dosežena s togo poravnavo R_S na 3D sliko, ki definira njegovo trajektorijo, t.j. presečiščno točko pedikla v sredinski čelni ravnini pedikla ter naklon v stranski in prečni ravnini.

2.2.1 Velikost in trajektorija pedikularnega vijaka

3D model pedikularnega vijaka je določen s parametri $\mathbf{s} = \{D, L, \mathbf{p}, \omega_x, \omega_z\}$ v referenčnem koordinatnem sistemu, ki definirajo velikost pedikularnega vijaka s premerom D (t.j. $D = 2a = 2b$ v enačbi 1) in dolžino L (t.j. $L = 2h$ v enačbi 1) ter trajektorijo pedikularnega vijaka s presečiščno točko pedikla $\mathbf{p} = (x_p, y_p, z_p)$ in naklonoma ω_x in ω_z v stranski in prečni ravnini (slika 2).

2.2.1 Načrtovanje pedikularnega vijaka

Pritrdilna moč vijaka je neposredno povezana s CT slikovnimi intenzitetami, ki linearno korelirajo z BMD [2,4,9]. Linte in sod. [2] so definirali pritrtilno moč kot:

$$F = \int_0^L \int_0^{2\pi} \int_0^{D/2} rI(r, \varphi, z) drd\varphi dz, \quad (6)$$

kjer je L dolžina vijaka, $D/2$ radij vijaka in $I(r, \varphi, z)$ so 3D slikovne intenzitete v cilindričnih koordinatah (r, φ, z) . Tako definirana pritrtilna moč upošteva slikovne intenzitete znotraj celotnega volumna vijaka, kar po našem mnenju ni optimalno, ker intenzitete blizu središčne osi vijaka ne prispevajo k pritrtilni moči. V izogib temu smo definirali pritrtilno moč kot:

$$F = \int_0^L \int_0^{2\pi} \int_{(D-\Delta)/2}^{(D+\Delta)/2} rI(r, \varphi, z) drd\varphi dz, \quad (7)$$

kjer Δ je odmik okoli radija vijaka. Pritrdilno moč smo nato še normirali z volumnom $\pi D \Delta L$, ki se razteza med $(D-\Delta)/2$ in $(D+\Delta)/2$ (slika 2(b)):

$$F_n = \frac{F}{\pi D \Delta L}. \quad (8)$$

Poleg dejstva da v primerjavi s predlagano definicijo v delu Linte in sod. [2] slikovne intenzitete blizu središčne osi vijaka ne prispevajo k pritrtilni moči, smo tudi predlagali normalizirano pritrtilno moč F_n , katero maksimiramo pri iskanju optimalnega 3D modela pedikularnega vijaka; če namreč maksimiramo F , potem imata večji premer in dolžina pedikularnega vijaka prevladujoč vpliv. Iskanje maksimuma normalizirane pritrtilne moči je dodatno omejeno z anatomijo vretenca, ki je določena s 3D modelom telesa vretenca $V(\mathbf{x})$ ter pedikla $P(\mathbf{x})$, t.j. 3D model pedikularnega vijaka $S(\mathbf{x})$ mora biti popolnoma znotraj obeh anatomskih struktur $V(\mathbf{x})$ in $P(\mathbf{x})$, presečišče simetrijske ravnine, dobljene iz telesa vretenca $V(\mathbf{x})$ in trajektorije pedikularnega vijaka, mora biti izven telesa vretenca $V(\mathbf{x})$ ter presečišče pedikularnih vijakov levega in desnega pedikla mora biti prav tako izven telesa vretenca $V(\mathbf{x})$ (slika 2(b)). Parametri $\mathbf{s}^* = \{D^*, L^*, \mathbf{p}^*, \omega_x^*, \omega_z^*\}$, ki določajo optimalno velikost in trajektorijo 3D modela pedikularnega vijaka $S(\mathbf{x})$, so tako določeni kot:

$$\mathbf{s}^* = \arg \max_{\mathbf{s}} \left(F_n \Big|_{s \rightarrow S(\mathbf{x}), V(\mathbf{x}), P(\mathbf{x})} \right). \quad (9)$$

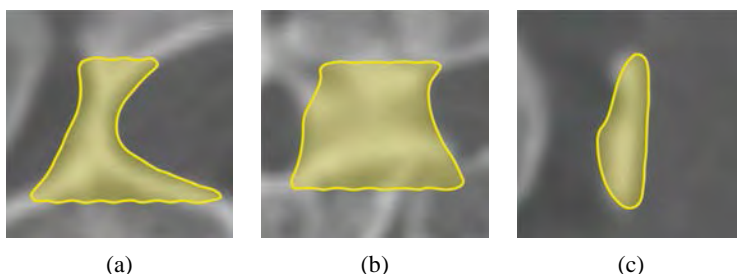
3. EKSPERIMENTI IN REZULTATI

3.1 Pacienti

Ekspirimenti so bili izvedeni na devetih pacientih (7 moških in 2 ženski; povprečna starost 18,7 let; razpon 14-34 let) z adolescentno idiopatsko skoliozo (7 pacientov) in degenerativno spremembo medvretenčne ploščice (2 pacienta) v prsnem delu hrbtenice. Vsi pacienti so bili napoteni na vstavev pedikularnih vijakov, za kar so bile predoperativno zajete CT slike (velikost slikovnega elementa 0,26-0,46 mm; debelina prereza CT slike 0,6 mm). Ortopedski kirurg je ročno načrtoval velikost in trajektorijo 61 pedikularnih vijakov v predoperativno zajetih CT slikah. Za načrtovanje so uporabili namensko aplikacijo, ki omogoča postavitev pedikularnega vijaka in njegovo manipulacijo v 3D. Iz teh ročnih predoperativnih načrtovanih vijakov so načrtovali pacientu prilagojena vodila, jih izdelali s postopkom 3D tiskanja ter medoperativno položili na načrtovano mesto na anatomiji pacienta. Skozi vodilo so nato vstavili pedikularni vijak, katerega velikost in trajektorijo so predoperativno načrtovali [10].

3.2 Implementacija

Predlagano metodo smo implementirali v programskem jeziku C++ brez optimizacije programske kode in izvedli na osebнем računalniku (Intel Core i7 pri 3,2 GHz in 32 GB spomina) s pospeševanjem na grafični procesni enoti (Nvidia GeForce GTX 760). Za vsako vretenca, kjer je kirurg ročno načrtoval pedikularni vijak, sta bila telo vretenca in pedikel razgrajena s parametričnima modeloma telesa vretenca $V(\mathbf{x})$ (vretenčno telo je bilo inicializirano kot eliptični valj, z ročno določenim centrom na CT sliki pacienta) in pedikla $P(\mathbf{x})$ (avtomatsko inicializirano iz parametrov pripadajočega vretenčnega telesa $V(\mathbf{x})$), kjer so bili optimalni parametri (enačba 5) določeni z nelinearno Nelder-Mead optimizacijsko metodo [11]. Predlagan 3D model telesa vretenca je bil uporabljen tako v prsnem kot tudi v ledvenem delu hrbtenice in podobno je bil 3D model pedikla uporabljen tako v prsnem kot tudi v ledvenem delu hrbtenice. Avtomatsko načrtovanje pedikularnega vijaka je bilo izvedeno v začetku z inicializacijo 3D modela pedikularnega vijaka $S_i(\mathbf{x})$ iz parametrov pripadajočega telesa vretenca $V(\mathbf{x})$ in pedikla $P(\mathbf{x})$ ter nato



Slika 3: Neustrezna oblika pedikla je prikazana na izbranem (a) prečnem, (b) stranskem in (c) čelnem prerezu.

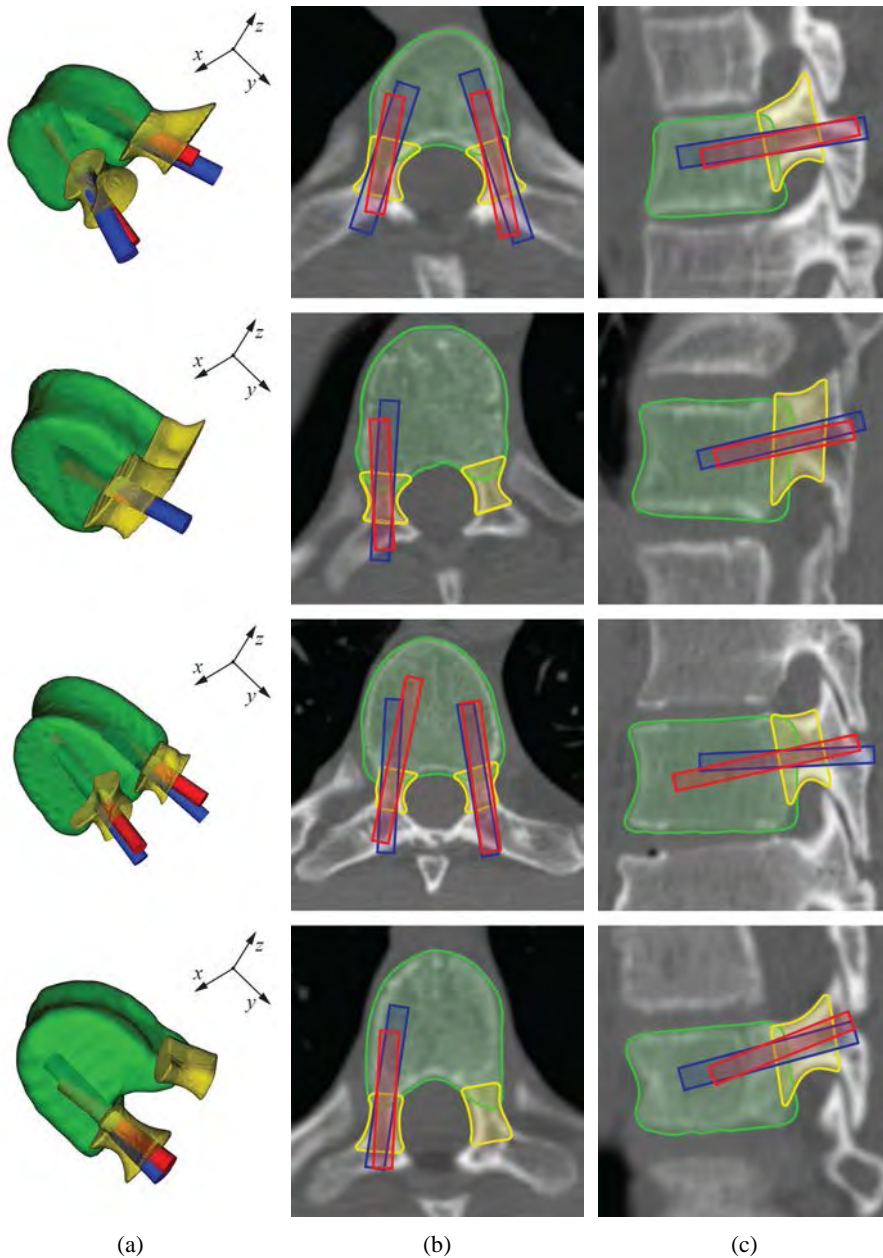
Tabela 1: Primerjava med ročno in avtomatsko načrtovanimi pedikularnimi vijaki z ozirom na povprečni absolutni odklon (MAD) in pripadajoči standardni odklon (SD) parametrov pedikularnega vijaka.

| Vretence | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 | T11 | T12 | Vsi |
|--|-----|-----|------|------|-----|-----|-----|-----|-----|-----|-----|------------|
| Št. Pedikularnih vijakov | 2 | 3 | 1 | 2 | 6 | 10 | 8 | 10 | 10 | 5 | 4 | 61 |
| Velikost pedikularnega vijaka: | | | | | | | | | | | | |
| premer (mm) | | | | | | | | | | | | |
| MAD | 0,7 | 0,8 | 0,6 | 0,3 | 0,7 | 0,6 | 0,8 | 0,5 | 0,6 | 0,1 | 0,4 | 0,6 |
| SD | 0,5 | 0,2 | 0,0 | 0,4 | 0,4 | 0,5 | 0,6 | 0,3 | 0,5 | 0,2 | 0,5 | 0,4 |
| dolžina (mm) | | | | | | | | | | | | |
| MAD | 1,1 | 5,5 | 0,8 | 7,2 | 2,8 | 3,9 | 3,8 | 5,4 | 4,5 | 2,3 | 7,3 | 4,0 |
| SD | 0,1 | 2,9 | 0,0 | 9,4 | 1,5 | 3,3 | 2,1 | 3,1 | 4,2 | 1,4 | 6,0 | 3,5 |
| Trajektorija pedikularnega vijaka: | | | | | | | | | | | | |
| presečiščna točka pedikla (mm) | | | | | | | | | | | | |
| MAD | 1,6 | 1,8 | 1,2 | 1,8 | 2,0 | 1,5 | 1,3 | 1,5 | 2,0 | 2,2 | 1,4 | 1,7 |
| SD | 0,6 | 1,1 | 0,0 | 1,2 | 1,2 | 1,1 | 1,0 | 1,5 | 1,3 | 1,1 | 0,9 | 1,2 |
| naklon v stranski ravnini (°) | | | | | | | | | | | | |
| MAD | 9,5 | 5,8 | 7,7 | 8,1 | 7,2 | 6,0 | 5,5 | 5,6 | 6,6 | 4,8 | 5,4 | 6,5 |
| SD | 8,6 | 4,7 | 0,0 | 8,9 | 8,0 | 3,6 | 4,2 | 3,0 | 4,1 | 2,7 | 7,2 | 4,8 |
| naklon v prečni ravnini (°) | | | | | | | | | | | | |
| MAD | 3,8 | 5,0 | 10,4 | 10,7 | 4,6 | 4,7 | 2,5 | 6,7 | 3,9 | 4,3 | 5,9 | 5,7 |
| SD | 2,7 | 4,8 | 0,0 | 9,9 | 4,4 | 4,5 | 2,9 | 3,1 | 3,1 | 2,5 | 4,8 | 3,9 |
| Načrtovanje pedikularnega vijaka: | | | | | | | | | | | | |
| normalizirana pritrdilna moč (1/mm³) | | | | | | | | | | | | |
| MAD | 44 | 67 | 42 | 191 | 69 | 92 | 65 | 111 | 97 | 68 | 57 | 82 |
| SD | 63 | 42 | 0 | 211 | 61 | 60 | 59 | 87 | 131 | 50 | 72 | 87 |

z iskanjem optimalnega 3D modela pedikularnega vijaka $S(\mathbf{x})$, pri čemer se je maksimirala normalizirana pritrdilna moč F_n (enačbi 8 in 9; $\Delta = 0,1D$); ponovno z uporabo nelinearne Nelder-Mead optimizacijske metode [11].

3.2 Rezultati

Za 73 teles vretenc in 146 pediklov je bil povprečni absolutni odklon (MAD) \pm standardni odklon (SD) pri modeliranju 3D teles vretenca in pediklov, določen z radialno evklidsko razdaljo do ročno postavljenih točk na telesih vretenca in pediklih, enak $1,2 \pm 0,3$ mm za telesa vretenca ter $0,7 \pm 0,5$ mm za pedikle. Dobljene avtomatsko načrtovane pedikularne vijake smo primerjali s pripadajočimi ročno načrtovanimi pedikularnimi vijaki (tabela 1). Kot je razvidno iz tabele 1 je podobnost med avtomatsko in ročno načrtovanimi pedikularnimi vijaki relativno visoka, kar rezultira v MAD \pm SD razliki $0,6 \pm 0,4$ mm za premer D in $4,0 \pm 3,5$ mm za dolžino L v povezavi z velikostjo pedikularnega vijaka ter $1,7 \pm 1,2$ mm za presečiščno točko pedikla p , $6,5 \pm 4,8^\circ$ za naklon v stranski ravnini ω_x in $5,7 \pm 3,0^\circ$ za naklon v prečni ravnini ω_z v povezavi s trajektorijo pedikularnega vijaka.



Slika 4: Vizualna primerjava med avtomatsko načrtovanimi (v rdeči barvi) in ročno načrtovanimi (v modri barvi) pedikularnimi vijaki za izbrana vretenca dveh pacientov z adolescentno idiopatsko skoliozo (prva in druga vrstica) ter dveh pacientov z degenerativno spremembo medvretenčne

V povprečju predlagana avtomatska metoda za vsak pedikularni vijak porabi okoli 6 min (t.j. 3,7 min za modeliranje 3D modela telesa vretenca, 2,2 min za modeliranje 3D modela pedikla in 3 s za načrtovanje pedikularnega vijaka); če je potrebno načrtovati ploščice (tretja in četrta vrstica). Primerjava je prikazana na (a) tridimenzionalnem pogledu ter izbranem (b) prečnem in (c) stranskem prerezu (samo za levi pedikel). pedikularni vijak v levem in desnem pediklu, potem se modeliranje 3D modela telesa vretenca izvede samo enkrat. Pri 50 avtomatsko načrtovanih pedikularnih vijakih (82%) je bila normalizirana pritrdilna moč F_n v primerjavi z ročno načrtovanimi pedikularnimi vijaki večja v povprečju za $96 \pm 87 \text{ mm}^3$ (razlika z ročno načrtovanimi: $0,6 \pm 0,4 \text{ mm}$ za premer D in $4,4 \pm 3,7 \text{ mm}$ za dolžino L v povezavi z velikostjo pedikularnega vijaka ter $1,6 \pm 1,0 \text{ mm}$ za presečiščno točko pedikla p , $6,2 \pm 4,3^\circ$ za naklon v stranski ravnini ω_x in $5,0 \pm 4,1^\circ$ za naklon v prečni ravnini ω_z v povezavi s trajektorijo pedikularnega vijaka). Na drugi strani je bil majhen padec normalizirane pritrdilne moči F_n pri ostalih 11 pedikularnih vijakih za $38 \pm 41 \text{ mm}^3$ (razlika z ročno načrtovanimi: $0,5 \pm 0,5 \text{ mm}$ za premer D in $3,0 \pm 3,1 \text{ mm}$ za dolžino L v povezavi z velikostjo pedikularnega vijaka ter $2,1 \pm 1,7 \text{ mm}$ za presečiščno točko pedikla p , $5,5 \pm 6,3^\circ$ za naklon v stranski ravnini ω_x in $4,7 \pm 4,1^\circ$ za naklon v prečni ravnini ω_z v povezavi s trajektorijo pedikularnega vijaka). Statistično značilna razlika ($p < 0,01$) je bila opažena med ročno in avtomatsko načrtovanimi pedikularnimi vijaki. Primeri avtomatsko načrtovanih in pripadajočih ročno načrtovanih pedikularnih vijakov so prikazani na sliki 4. Predlagana avtomatska metoda je napačno načrtovala pedikularni vijak v enem primeru, kjer ni bilo ustrezne oblike pedikla (slika 3); v praksi se kirurgi izogibajo pediklov, kjer ni mogoče vstaviti primernega pedikularnega vijaka.

4. DISKUSIJA IN ZAKJUČKI

Opisali smo novo avtomatsko metodo za predoperativno načrtovanje pedikularnih vijakov, ki temelji na maksimiranju pritrdilne moči vijakov. Predlagali smo modifikacijo pritrdilne moči (enačba 7), ki upošteva le slikovne intenzitete v okolici pedikularnega vijaka; samo okolica pedikularnega vijaka je v kontaktu z navojem pedikularnega vijaka in zato samo pripadajoča BMD prispeva k dejanski pritrditvi vijaka. Poleg tega smo predlagali še normalizacijo pritrdilne moči z volumnom znotraj katerega se upoštevajo intenzitete, zato da smo odpravili vpliv večjega premera in dolžine pedikularnega vijaka med iskanjem optimalnih parametrov ter omogočili primerjavo z ročno načrtovanimi pedikularnimi vijaki različnih velikosti. Izvedeni eksperimenti in dobljeni rezultati kažejo na skladnost s pedikularnimi vijaki, ki jih je ročno načrtoval izkušen ortopedski kirurg. Pri 82% avtomatsko načrtovanih pedikularnih vijakov je bila normalizirana pritrdilna moč višja kot pri ročno načrtovanih pedikularnih vijakih, kar kaže na potencialno uporabnost predlagane metode v računalniško podprti kirurgiji. Čeprav so časi avtomatskega načrtovanja pedikularnih vijakov primerljivi s časi ročno načrtovanih, je avtomatsko načrtovanje pedikularnih vijakov izvedljivo brez prisotnosti kirurga med samim načrtovanjem. Kirurg po končanem načrtovanju samo preveri ustreznost dobljenih velikosti in trajektorij pedikularnih vijakov, kar se odraža v znatno hitrejšem

predoperativnim načrtovanju. Bodoče raziskovanje bo usmerjeno na razširitev zbirke pacientov, vrednotenje variabilnosti med opazovalci in znotraj opazovalca ročno načrtovanih pedikularnih vijakov ter primerjavo dejanskih postoperativnih trajektorij pedikularnih vijakov z načrtovanimi.

ZAHVALA

To delo je podprla Javna agencija za raziskovalno dejavnost Republike Slovenije (ARRS) v okviru projektov P2-0232, J2-5473, J7-6781 in J2-7118. Avtorji se zahvaljujejo podjetju Ekliptik d.o.o. za predoperativne slike pacientov in ročne načrte za pedikularne vijake.

LITERATURA

1. A. Manbachi, R. Cobbold, H. Ginsberg (2014), Guided pedicle screw insertion: techniques and training, *Spine Journal*, vol. 14, str. 165-179.
2. C. Linte, K. Augustine, J. Camp, R. Robb, D. Holmes III (2015), Toward virtual modeling and templating for enhanced spine surgery planning, *S. Li, J. Yao (ur.), Spinal Imaging and Image Analysis, LNCVB*, vol. 18, str. 441-467, Springer International Publishing AG, Cham, Switzerland.
3. J. Lee, S. Kim, Y. Kim, W. Chung (2012), Optimal surgical planning guidance for lumbar spinal fusion considering operational safety and vertebra-screw interface strength, *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 8, str. 261-272.
4. R. Lehman Jr., D. Polly Jr., T. Kuklo, B. Cunningham, K. Kirk, P. Belmont Jr. (2003), Straight-forward versus anatomic trajectory technique of thoracic pedicle screw fixation: a biomechanical analysis, *Spine* vol. 28, str. 2058-2065.
5. N. F. Tian, Q. S. Huang, P. Zhou, Y. Zhou, R. K. Wu, Y. Lou, H. Z. Xu (2011), Pedicle screw insertion accuracy with different assisted methods: a systematic review and meta-analysis of comparative studies, *European Spine Journal*, vol. 20, str. 846-859.
6. P. Helm, R. Teichman, S. Hartmann, D. Simon (2015), Spinal navigation and imaging: history, trends and future, *IEEE Transactions on Medical Imaging*, vol. 34, str. 1738-1746.
7. J. Lee, S. Kim, Y. Kim, W. Chung (2011), Automated segmentation of the lumbar pedicle in CT images for spinal fusion surgery, *IEEE Transactions on Biomedical Engineering*, vol. 58, str. 2051-2063.

8. D. Štern, B. Likar, F. Pernuš, T. Vrtovec (2011), Parametric modelling and segmentation of vertebral bodies in 3D CT and MR spine images, *Physics in Medicine and Biology*, vol. 56, str. 7505-7522.
9. J. J. Schreiber, P. A. Anderson, H. G. Rosas (2011), Hounsfield units for assessing bone mineral density and strength: a tool for osteoporosis management, *Journal of Bone and Joint Surgery*, vol. 93, str. 1057-1063.
10. U. Tominc, M. Vesel, S. Al Mawed, M. Dobravec, M. Jug, S. Herman, D. Kreuh (2014): Personalized guiding templates for pedicle screw placement, *Proceedings of 37 International Convention on Information and Communication Technology, Electronics and Microelectronics - MIPRO 2014*, Maj 26.-30., Opatija, Hrvatska, IEEE, str. 249-251.
11. W. Press, S. Teukolsky, W. Vetterling, B. Flannery (2007), Numerical recipes: the art of scientific computing, *Cambridge University Press*, Cambridge, UK, 3. izdaja.

DRUGO

DIAGNOSTIKA HITROSTI Z LASERSKIM MERILNIKOM LDM-301

Milan Čoh
Peter Planjšek

Fakulteta za šport, Univerza v Ljubljani
Laboratorij za gibalni nadzor
E –pošta: milan.coh@fsp.uni-lj.si
URL: <http://cobsis.uni-mb.si>

POVZETEK: *Razvoj sodobnega športa je vse bolj povezan z novimi tehnologijami in metodami v procesu treniranja športnikov. Vrhunski rezultati so vedno težje dosegljivi, zato mora biti trening skrajno racionalen, učinkovit in kontroliran. Zato pa potrebujemo kvalitetne in kar se da objektivne informacije. V Laboratoriju ju za gibalni nadzor, Fakultete za šport v Ljubljani smo razvili povsem novo metodologijo in tehnologijo merjenja linearne tekaške hitrosti. Z laserskim merilnikom LDM-301 lahko merimo na najbolj natančen način parametre dinamike sprinterske hitrosti v posameznih fazah.)*

1. UVOD

Rezultati so na današnji stopnji razvoja športa vse bolj produkt programiranega in kontroliranega procesa treninga. To je kompleksen proces, ki ima v naprej definirane cilje, sredstva in metode vadbe. V sodobnem atletskem treningu ima diagnostika, ki temelji na novih tehnologijah in tehnološko- metodoloških rešitvah izjemno pomembno funkcijo. Smisel diagnostičnih postopkov je ugotavljanje relevantnih in čim bolj objektivnih parametrov trenutne pripravljenosti športnika. Brez podatkov o biomotoričnih, morfoloških, fizioloških, biokemičnih in psiholoških značilnosti ni mogoče planirati, programirati in

modelirati sodobnega trenajnega procesa (Latash, 2000; Zatsiorsky, Kraemer, 2006). Na osnovi pridobljenih podatkov je mogoče izbrati najoptimalnejše metode in sredstva, načrtovati ciklizacijo in korekcije športne priprave.

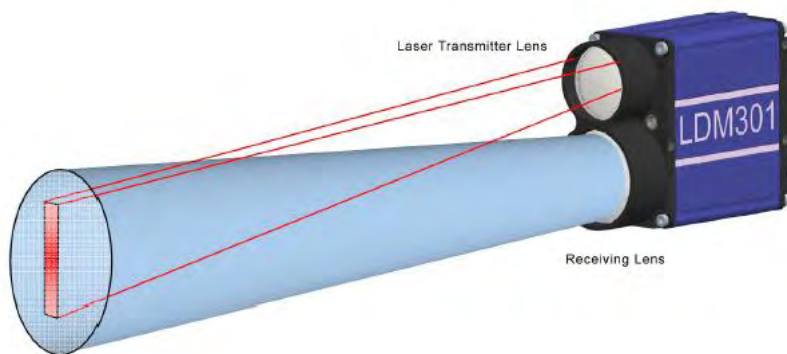
Lokomotorna hitrost je nedvomno ena od najpomembnejših biomotoričnih sposobnosti. Le- ta se pojavlja v različnih atletskih disciplinah kot sprinterska hitrost ali hitrost zaleta pri skokih in metih. Laserski merilnik hitrosti pomeni povsem novo poglavje v razvoju diagnostičnih metod hitrosti. V Laboratoriju ju za gibalni nadzor smo razvili povsem novo metodologijo diagnosticiranja sprinterske hitrosti (avtorji metode: Milan Čoh, Planjšek Peter, Stanko Štuhec).

2.0 PREDSTAVITEV METODE - Laser LDM 301

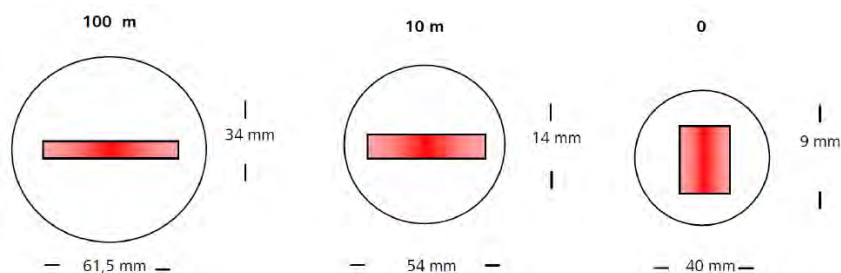
Laserski merilnik razdalje LDM - 301 (Laser distance measuring device, Tabela 1, Slika 1) nam omogoča ne invazivno merjenje hitrosti in razdalj, kar v praksi pomeni da lahko športnik teče v tekmovalnih pogojih, brez senzorjev, ki bi ga lahko ovirali pri samem izvajanju meritve. Laserski žarek LDM 301 je kategoriziran z kategorijo 1, kar pomeni, da je neškodljiv za uporabo v normalnih pogojih. To pomeni, da maksimalna dovoljena izpostavljenost ne more biti presežena z gledanjem laserskega žarka z prostim očesom. Opremljen je z kalibracijskim laserjem (t.i. "pilot laser") za jasno identifikacijo točke merjenja (Slika 2). Povprečni kot pod katerim se širi laserski žarek znaša 1,7mRad (Slika 3). Povezava z osebnim računalnikom je zagotovljena preko serijskega vhoda RS232 ali RS422.



Slika 1: Laserski merilnik hitrosti LDM 301.



Slika 2: Diagram laserskega žarka 1,7mRad.



Slika 3: Velikost laserskega žarka LDM301 na različni oddaljenosti.

Tabela 1: Tehnične karakteristike laserskega merilca hitrosti LDM301 (Astech (online). (Rostock, Nemčija): Laser distance meter 301 User manual ver. 1.5.04.)

Tehnične karakteristike LDM-301:

Merilno območje: 0,5, do 300m

Natančnost: +/- 20mm

Čas merjenja: 0,5 ms z možnostjo 0,1ms

Območje merjenja hitrosti: 0m/s do 100m/s

Merilni laser: 905nm infrardeč, kategorija 1, EN 60825-1:2003-10

Kontrolni laser: 650nm viden rdeč, kategorija 2, 1mW

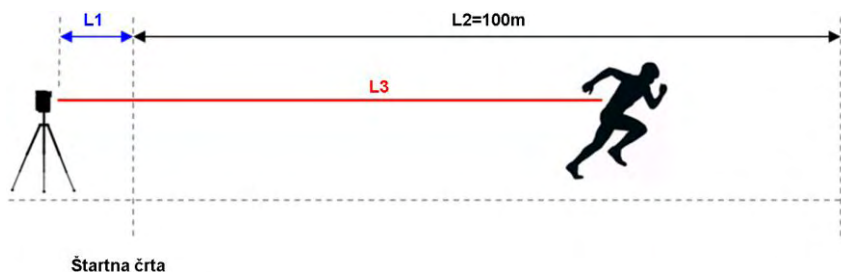
Delovna temperatura: -40 °C do +60 °C

Povezava : preko serijskega vhoda RS232 ali RS422

3. POSTAVITEV IN KALIBRACIJA CONE MERJENJA

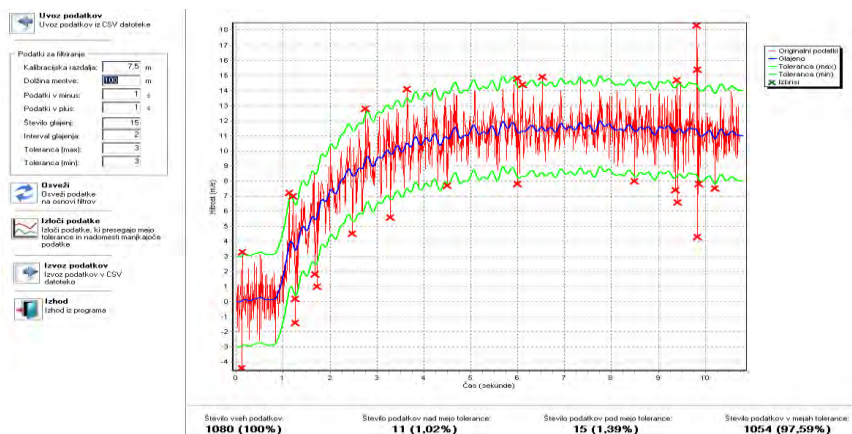
Pred meritvijo je potrebno sistem kalibrirati. Kalibracija je potrebna zaradi določitve cone merjenja. Laser namreč izmeri razdaljo 100 krat v sekundi. Z določitvijo cone merjenja tako določimo območje v katerem bomo merili hitrost šprinterja. Kalibracija se izvede tako, da na štartno linijo postavimo

pravokotno letvico in z laserjem izmerimo razdaljo do letvice (L1). Ta razdalja predstavlja osnovo za meritev. Ko šprinter preide izmerjeno kalibracijsko razdaljo, vstopi v cono meritve (L2). Meritev traja tako dolgo, dokler se šprinter nahaja v coni meritve (L2) – Slika 4.



Slika 4: Shematski prikaz meritve šprinta na 100m. L1– kalibracijska razdalja, L2 – cona merjenja razdalje, L3 – razdalja meritve laserja Dejansko razdaljo šprinterja do laserja izračunamo kot razliko med razdaljo L3 in L1.

V drugi fazi obdelave podatkov določimo stopnje glajenja in določimo tolerančno območje za napako. Na ta način dobimo vse podatke, ki so izven meja tolerance (označene z rdečimi križci) in jih izločimo (Slika 5).



Slika 5: Druga faza izločanja napake. Prikaz napak nad mejo tolerance določene v programu.

4. APLIKACIJA METODE

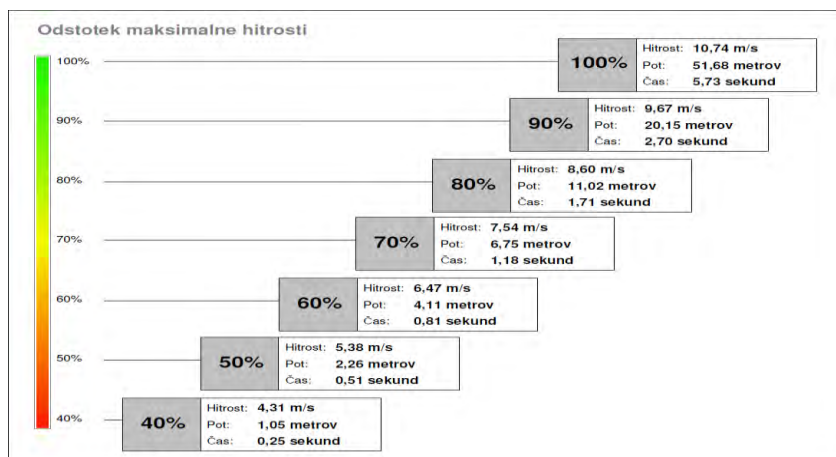
Šprinterski tek je ciklično gibanje odvisno od mnogih biomehanskih dejavnikov in povezanosti le-teh. Šprinterska hitrost se spreminja v posameznih fazah teka, zato vsaka faza zasluži posebno obravnavo tako iz vidika diagnostike, kot tudi iz vidika treninga (Mero, Komi, Gregor, 1992; Delecluse in sod, 1995). Trenutna programska oprema nam omogoča analizo naslednjih faz sprinterskega teka:

1. Analizo startne akceleracije
2. Analizo maksimalne hitrosti
3. Analizo točke maksimalne hitrosti
4. Analizo deceleracije - upadanja hitrosti
5. Prikaz hitrosti v conah (graf po poti, graf v času) - Slika 6
6. Realizacijo hitrosti v startnem pospešku - Slika 7

| | Faza pospeševanja | Faza maksimalne hitrosti | Faza zaviranja |
|--------------------|-------------------|--------------------------|------------------|
| Čas trajanja: | 4,46s 43,13% | 3,32s 32,11% | 2,56s 24,76% |
| od - do: | 0,00s - 4,46s | 4,46s - 7,78s | 7,78s - 10,34s |
| Razdalja trajanja: | 38,18m 38,15% | 35,37m 35,34% | 26,53m 26,51% |
| od - do: | 0,00m - 38,18m | 38,18m - 73,55m | 73,55m - 100,07m |

Slika 6: Faze šprinterskega teka. Tabelarni prikaz parametrov posamezne faze.

Na sliki 7 lahko vidimo kako procentualno narašča hitrost šprinterja glede na maksimalno hitrost. Na ta način lahko ugotovimo na kakšni razdalji in v kakšnem času je šprinter dosegel določeno hitrost.



Slika 7: Shematski prikaz procentualnega doseganja maksimalne hitrosti s podatki o poti in času.

5. ZAKLJUČEK

Iz biomehanskega vidika je šprinterski tek kompleksna disciplina, kjer se hitrost spreminja v posameznih fazah. Laserska metoda merjenja nam omogoča zelo natančno analitiko hitrosti in s tem dobro kontrolo te motorične sposobnosti. Metoda je uporabna pri vseh tistih športnih panogah kjer je absolutna hitrost ali hitrost startnega pospeševanja pomemben dejavnik tekmovalne uspešnosti. Metoda je povsem ne invazivna, ne ovira in ne moti športnika, rezultati pa so dosegljivi v nekaj minutah po meritvi. V sodobnem treningu je ravno pravočasna informacija ključnega pomena za učinkovito programiranje procesa treninga.

LITERATURA

1. Latash, M., L. (2000). Control of Human movement. Human Kinetics Publishers. Champaign, Illinois
2. Mero, A., Komi, P.V., Gregor, R.J. (1992). Biomechanics of sprinting running. Sport medicine 13 (6): 376-392.
3. Zatsiorsky, V. M., Kraemer, W.J (2006). Science and practice of strength training (second edition). Human Kinetics, Champaign, IL.
4. Astech (online). (Rostock, Nemčija): Laser distance meter 301 User manual ver. 1.5.04. Dostopno na naslovu: http://www.astech.de/download/ldm301/manual/english/ldm301_manual_15e.pdf
5. KU Leuven (online). (Leuven, Belgija): KU Leuven team uses laser technology to analyse Usain Bolt's speed at Van Damme Memorial. Dostopno na naslovu: <http://www.kuleuven.be/english/news/k.u.leuven-team-uses-laser-technology-to-analyse-usain-bolt2019s-speed-at-van-damme-memorial>

FACE DE-IDENTIFICATION FOR PRIVACY PROTECTION IN SURVEILLANCE SYSTEMS

Darijan Marčetić, Branko Samaržija, Martin Soldić, Slobodan Ribarić

**Laboratory for Pattern Recognition and Biometric Security Systems
Faculty of Electrical Engineering and Computing, University of Zagreb
E-pošta: slobodan.ribaric@fer.hr
URL: <http://rubiooss.zemris.fer.hr>**

ABSTRACT: *The article presents some of the intermediate results of face de-identification, after the first year of the scientific project "De-identification for Privacy Protection in Surveillance Systems - DePPSS". We present two face de-identification methods. The first face de-identification method, which improves the naturalness and utility of a de-identified video, is based on the active appearance models (AAMs) of a person-specific set of images. The second one is based on a unified model for face detection, pose and landmark estimation. The experimental results for the both methods are given.*

1. INTRODUCTION

Recent advances in technology and signal processing have greatly facilitated the efficacy of video surveillance [1], primarily for the benefit of security and law enforcement. Video surveillance is now widely exploited in a variety of scenarios to capture video recordings of people in public and semi-public environments [2], either for immediate recognition and tracking of people or/and abnormal behaviour recognition, or for storage, and subsequent data analysis and sharing. Whilst it is recognized that there are justified reasons for acquisition and sharing videos in manners such as security, bio-terrorism surveillance applications, law enforcement and forensics, there is also a strong need for protecting the privacy of the guiltless individuals who are inevitably captured in the recordings. There are no doubts that video surveillance is privacy intrusive because it allows the observation of certain information that is considered privacy sensitive. Face has central role in the process of human recognition and identification in videos, thus the special attention has to be devoted to the face de-identification methods for privacy protection. De-identification, in general, is the process of concealing or removing personal identifiers, or replacing them with surrogate personal identifiers in personal information, in order to prevent the disclosure and use of data for purposes unrelated to the purpose for which the information was originally obtained.

Project's research plan defines the following main phases and activities:

- i. Review and analysis of previous approaches to the problem of face de-identification for still images and videos,

- ii. Robust face localization in videos adopted to de-identification process,
- iii. Novel algorithms and methods for automatic concealing of face identifiers with preserving the data utility and naturalness in videos,
- iv. Set up the experimental camera surveillance system with inbuilt face de-identification,
- v. Evaluation of privacy protection solutions in video surveillance.

In this paper we present the intermediate results of face de-identification, after the first year of 4-year scientific project "De-identification for Privacy Protection in Surveillance Systems - DePPSS".

2. FACE DE-IDENTIFICATION METHODS

The early research on face de-identification was focused on face still images, by using the ad-hoc approaches such as "black box", "pixelation" and "blurring" of the image region occupied by the face [3]. In the black-box approach, after the face detection and face localization in the image, the face region is simply substituted by a black (or white) rectangle, elliptical or circular covers. To achieve an improved level of privacy protection, more sophisticated approaches have been proposed: eigenvector-based de-identification method [4], k -Same, k -Same-Select algorithms and Model-based k -Same method [5], morphing- and warping-based [6] methods, cartooning [7] and scrambling-based methods [8]. The basic idea of the k -Same algorithms is illustrated in Figure 1.

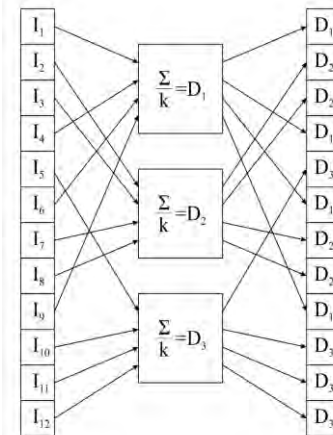


Figure. 1. The basic idea of k -Same algorithms

By applying the k -Same algorithm, to the given original person-specific set of images I , where each person is represented by no more than one image, a set of de-identified images D is computed. Each de-identified image is represented by an average face image of the k closest face images from the person-specific set of images. The k closest face images in the person specific set are replaced by the same k de-identified face images. The k -Same algorithm selects the k closest images based on Euclidean distances in the image space or in the Principal Component Analysis (PCA) coefficient space. The

process of de-identification is irreversible. Figure 2. illustrates the result of the k -Same de-identification.



Figure 2. k -Same de-identification: a) Original image; b) De-identified image for $k = 6$; [9].

Special attention in the field of privacy protection is now being devoted to automatic face de-identification in video surveillance systems because of their privacy-intrusive characteristics [10]. The process of automatic face de-identification in videos includes:

i) face detection; ii) face tracking; and iii) face masking by concealing or removing personal identifiers, or replacing them with surrogate personal identifiers.

i) There are face-detector candidates for use in videos as follows: neural network based detector [11], Schneiderman-Kanade detector [12], Viola-Jones detector [13], local edge orientation histograms based (EOH) [14], and histograms of oriented gradients [15]. Recently, new methods have been proposed for face detection, pose estimation and landmark localization in the wild [16]. It is worth noting that privacy might be compromised in video sequences if the face detection algorithm fails in a single frame, so one of the directions of research is the development of robust and effective algorithms for privacy protection that can efficiently cope with situations when computer vision algorithms fail.

ii) Face tracking is the process of locating a moving human faces in a sequence of frames. Tracking is based on features such as segmented regions, skin-colour models [17], local binary patterns (LBP) [18], a combination of LBP and skin-colour information [19], a combination of shape and texture information [20], and histogram-based Mean-Shift features [21]. The combination of face detection and tracking, i.e. the combination of the spatial and temporal correspondence between frames, can improve the effectiveness of the localization of faces.

iii) Each localized and traced face region in each frame has to be de-identified by some effective means. Approaches to face masking for privacy protection in video-surveillance systems follow techniques that are used in still-face images, such as: ad-hock methods, k -Same-based methods, morphing, warping, cartooning and scrambling. These methods are tested and level of privacy protection was evaluated by PCA-, LDA-, LBP-based face recognition algorithms [22] and crowdsourcing approach [23].

3. TWO FACE DE-IDENTIFICATION METHODS AND RESULTS

In order to improve the naturalness and utility of a de-identified video, we adapted the de-identification methods for still images [24]. Normally, the faces captured in a video

sequence are of varied poses. Such variations may range from a full left profile to a full right profile (yaw angle from -90° to $+90^{\circ}$) and a pitch from -90° to $+90^{\circ}$, while the roll is usually more restricted. Following the idea from *k*-Same-Select [5], where images are grouped before de-identification to preserve the facial expression and the gender, the proposed approach groups the face images into a person-specific set of images according to their poses. Each person-specific set is represented by an active appearance model (AAM); Figure 3. A raw face image is matched with each of the active appearance models of a person-specific set of images. The model with the best matching based on shape and texture is chosen to represent the pose of the raw face image. Then, from the images in the selected person-specific set of images, one image is chosen to replace the texture of the raw image. The shape of the de-identified face image remains the same as that detected during the model fitting, but the texture is changed. Note that in order to enhance the privacy protection, instead of using the most similar appearance for the raw image, the appearance of an image that is far enough (*q*-far based on the Euclidean distance) is used [24]. The proposed de-identification method is irreversible. Figure 4. illustrates the above-described approach.

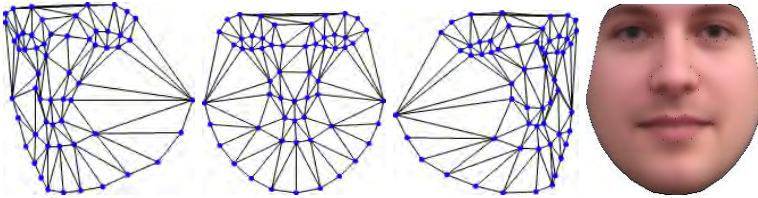


Figure 3: Model shapes (for 3 different poses) that are used for fitting to the raw image; Besides the shape, AAMs use the appearance. The appearance corresponding to the shape on far right.

The second method is based on a unified model for face detection, pose estimation, and landmark estimation described in [16]. This model is based on mixtures of trees with a shared pool of parts; facial landmarks are modelled as parts and global mixtures are used to capture topological changes due to viewpoint. This tree-structured can robustly handle global elastic deformation, and can be optimized with linear programming. This method is suitable for “in the wild” datasets. Figure 5. illustrates results of the second de-identification method.

For both methods the face region (ROI) detected in the original image is replaced with selected swapping face by performing following procedure. By using Delaunay triangulation, each part of the face region detected in the original image bounded by the triangles (Figure 3.) is taken into account (it is a convex region), while the rest of the ROI is neglected. Then affine warps from each triangle of ROI to the corresponding triangle of selected swapping face, are computed. The appearance of this triangle is copied into corresponding ROI triangle of the original image. To avoid possible gaps in the appearance (due to the differences in the triangle area sizes), backward warps are performed by appearance normalisation to the mean shape.

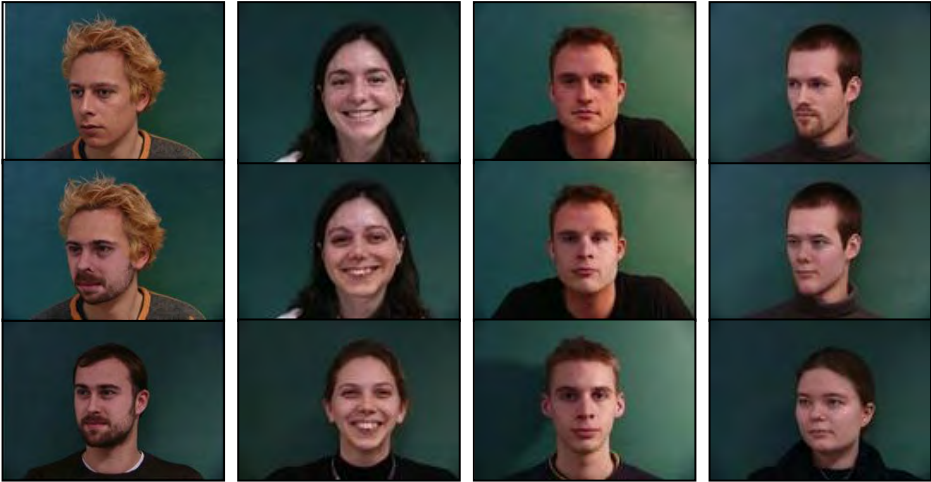


Figure 4. Illustration of the q-far de-identification method [24]: first original image; second row de-identified image $q\text{-far} = 35$; third row image used for the face swapping.



Figure 5. Illustration of the de-identification method: first original image; second row de-identified image; third row image used for the face swapping.

3. CONCLUSION

De-identification of the face in video surveillance systems is far from a complete solution. The problem lies not in the de-identification of ROIs, but in computer vision algorithms for the detection and localization of face(s) in video sequences. Despite recently intensive research in computer vision, numerous problems still remain to be solved in automatic face detection and consequently of face de-identification. These include issues such as the detection of the face under different illumination conditions, bad lighting conditions, different head positions, the presence of structural components (e.g., glasses, sunglasses, beards, moustaches), and occlusions. The unsolved problems are the detection of faces in crowd scenes and real-time de-identification.

ACKNOWLEDGEMENT

This work has been fully supported by Croatian Science Foundation under the project 6733 De-identification for Privacy Protection in Surveillance Systems (DePPSS), Grant no:6733.

LITERATURE

1. D. T. Raty, Survey on Contemporary Remote Surveillance Systems for Public Safety, IEEE Transactions on Systems, Man, and Cybernetics - Part C, vol. 40, no. 5, (2010) 493 - 515.
2. P. Agrawal, P. J. Narayanan, Person De-Identification in Videos, IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, no. 3, (2011) 299 - 310.
3. M. Boyle, C. Edwards, S. Greenberg, The Effects of Filtered Video on Awareness and Privacy, ACM Conference on Computer Supported Cooperative Work, Philadelphia, December 2000, pp. 1-10
4. P. J. Phillips, Privacy operating characteristic for privacy protection in surveillance applications, in: T. Kanade, A. Jain, and N. Ratha (Eds.), Audio- and Video-Based Biometric Person Authentication, Lecture Notes in Computer Science, Springer, 2005, pp. 869 - 878.
5. R. Gross, L. Sweeney, J. Cohn, F. de la Torre, S. Baker, Face De-Identification, in: A. Senior (Ed.), Protecting Privacy in Video Surveillance, Springer, 2009, pp. 129 - 146.
6. P. Korshunov, T. Ebrahimi, Using Face Morphing to Protect Privacy, IEEE Int. Conference on Advanced Video and Signal-based Surveillance, (2013) 208 - 213.
7. A. Erdely, T. Barat, P. Valet, T. Winkler, B. Rinner, Adaptive Cartooning for Privacy Protection in Camera Networks, 11th IEEE Int. Conference on Advanced Video and Signal Based Surveillance (AVSS), (2014) 26 - 29.
8. F. Dufaux, T. Ebrahimi, Scrambling for Privacy Protection in Video Surveillance Systems, IEEE Trans. on Circuits and Systems for Video Technology, vol. 18, no. 8, (2008) 1168 - 1174.

9. S. Ribaric, N. Pavesic, An Overview of Face De-identification in Still Images and Videos, Workshop on De-identification for Privacy Protection in Multimedia, IEEE FG 2015 Ljubljana, (2015) 1 - 6.
10. A. Senior, Privacy Protection in a Video Surveillance System, in: A. Senior (Ed.), Protecting Privacy in Video Surveillance, Springer, Dordrecht, 2009, pp. 35 - 47.
11. H. A. Rowley, S. Baluja, T. Kanade, Neural Network-Based Face Detection, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 20, no. 1, (1998) 23-38.
12. H. Schneiderman, T. Kanade, A Statistical Method for 3D Object Detection Applied to Faces and Cars, Proceedings on Conference Computer Vision and Pattern Recognition, vol. I, (2001) 746 - 751.
13. P. Viola, M. J. Jones, Robust Real-Time Face Detection, International Journal of Computer Vision 57(2), (2004) 137 - 154.
14. K. Levi, Y. Weiss, Learning Object Detection from a Small Number of Examples: the Importance of Good Features, Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04), vol. 2, (2004) II - 53 - II - 60.
15. N. Dalal, B. Triggs, Histograms of Oriented Gradients for Human Detection, IEEE Conference on Computer Vision and Pattern Recognition, 2005, vol. 1, 886 - 893.
16. X. Zhu, D. Ramanan, Face Detection, Pose Estimation, and Landmark Localization in the Wild, Proceedings on Conference Computer Vision and Pattern Recognition, (2012) 2879 - 2886.
17. J. Yang, A. Waibel, A Real-Time Face Tracker, Proceedings 3rd IEEE Workshop on Applications of Computer Vision WACV '96, (1996) 142 - 147.
18. L. Xu, J. Li, K. Wang, Real-time and Multi-View Face Tracking on Mobile Platform, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), (2011) 1485 - 1488.
19. W. Chuan-xu, L. Zuo-yong, A New Face Tracking Algorithm Based on Local Binary Pattern and Skin Color Information, 2008 International Symposium on Computer Science and Computational Technology, (2008) 657 - 660.
20. W-P. Choi, K-M. Lam, An Effective Shape-Texture Weighted Algorithm for Multi-view Face Tracking in Videos, 2008 Congress on Image and Signal Processing, (2008) 156 - 160.
21. D. Comaniciu, V. Ramesh, P. Meer, Real-Time Tracking of Non-Rigid Objects using Mean Shift, Proceedings on IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, (2000) 142 - 149.
22. R. Gross, E. Airoldi, B. Malin, L. Sweeney, Integrating Utility into Face De-identification, in: G. Danezis and D. Martin (Eds.): PET- Privacy Enhancing Technologies 2005, LNCS 3856, 2006, pp. 227 - 242.
23. P. Korshunov, S. Cai, and T. Ebrahimi, Crowdsourcing approach for evaluation of privacy filters in video surveillance, Proceedings of the ACM Multimedia 2012 Workshop on Crowdsourcing for Multimedia, Japan, CrowdMM'12, (2012) 35 - 40.
24. B. Samarzija, S. Ribaric, An Approach to the De-Identification of Faces in Different Poses, Proceedings of Special Session on Biometrics, Forensics, De-identification and Privacy Protection BiForD 2014, (2014) 21 - 26.

ŠTUDENTSKE APLIKACIJE

DOLOČANJE ODZIVA NA ZAUŽITO HRANO Z METODAMI ZA PREPOZNAVNO OBRAZNEGA IZRAZA

Leon Ropoša, Borut Batagelj, Franc Solina

Laboratorij za računalniški vid

Fakulteta za računalništvo in informatiko, Univerza v Ljubljani
leon.roposa@gmail.com, {borut.batagelj, franc.solina}@fri.uni-lj.si

POVZETEK: V članku opisujemo metodo za prepoznavo obraznega izraza da bi določili odziv testirane osebe na okus zaužite hrane. Trenutno obstaja zgolj nekaj podobnih raziskav, kjer so z uporabo obstoječe programske opreme, ki zna na obrazih ločiti med šestimi osnovnimi čustvi, iskali korelacijo med oceno hrane, ki jo je podal preizkuševalec, in njegovo reakcijo izraženo na obrazu. V naši raziskavi smo izdelali svojo metodo za prepoznavo čustev na obrazih in jo preizkusili na lastni bazi posnetkov okušanja hrane. Problem smo definirali kot dvorazredni klasifikacijski, torej, ali lahko s pomočjo prepoznave obraznega izraza ugotovimo, ali je osebi okus zaužite hrane všeč ali ne. Ločeno smo obravnavali odzive na zaužito hrano in zaužito pijačo. Dosegli smo dobre rezultate pri analizi obraznih odzivov na hrano in zelo dobre rezultate pri analizi obraznih odzivov na pijačo.

1. UVOD

Večina raziskav na področju prepoznave obraznih izrazov z računalniškim vidom, poteka na osnovi zaznavanja šestih osnovnih čustev: jeze, strahu, gnusa, presenečenja, veselja in žalosti. Teh šest čustev je v 70.-letih 20. stoletja psiholog Paul Ekman definiral kot univerzalna, kar pomeni, da so ta čustva prisotna v vseh kulturah in skozi celotno človeško zgodovino. V podatkovnih bazah in člankih se včasih pojavijo tudi še kakšna dodatna čustva. Recimo, ko ni zaznano nobeno od naštetih čustev, se včasih doda tudi nevtralni izraz. Poleg šestih osnovnih čustev se pri prepoznavi obraznih izrazov uporablja tudi dimenzija prijetnost/vzbujenje (ang. Valence/Arousal). V tem primeru prevedemo problem iz klasifikacijskega v regresijski, kjer vzbujenje pomeni stopnjo intenzitete čustva. Te metode je možno uporabiti tudi za povsem določene aplikacije, recimo za ugotavljanje bolečine iz obraznega izraza.

V raziskavah [1] so ugotovili da obstaja povezava med obraznim izrazom in okusom zaužite hrane, tako zaradi prijetnega okusa, kot intenzitete okusa. Obrazni izrazi so najbolj izraziti pri visokih koncentracijah neprijetnih okusov, kot sta kislo (izrazi na

ustnicah) in grenko (izrazi na očeh in čelu). Ugotovljeno je bilo tudi, da prijetnejši okusi sprožijo na obrazu najmanj opazne obrazne izraze.

Obstaja nekaj člankov kjer so z namenom ugotavljanja odzivov ljudi na zaužito hrano, uporabili metode za prepoznavo obraznega izraza [2, 3, 4, 5]. Avtorji člankov so sicer aktivni na področju raziskav povezanih s hrano, zato se članki ne osredotočajo na metode računalniška vida, temveč opisujejo kako so z uporabo obstoječe programske opreme analizirali odziv na zaužito hrano. V vseh omenjenih člankih so za prepoznavo obraznih izrazov uporabili program FaceReader [6]. Program napove prisotnost enega izmed šestih osnovnih čustev in nevtralnega izraza. Za vsakega izmed šestih čustev in nevtralnega izraza, nam pove tudi intenziteto čustva na obrazu.

Avtorji so v člankih primerjali kako so se spreminjale napovedi intenzitet čustev, kot jih je napovedal FaceReader. V člankih so uporabili različne vrste hrane in pijače. Pozitivni odziv na okus je imel največjo korelacijo ali z nevtralnimi izrazom [3] ali pa z veseljem [4, 5]. Negativni odziv je imel korelacijo z gnusom, jezo in žalostjo [2, 3, 4, 5]. V [2] je imel negativni odziv na okus veliko večji vpliv na spremembo zaznanih intenzitet čustev. Nekateri izmed avtorjev [4, 5] so mnenja, da bi bile metode za prepoznavo obraznega izraza lahko primerne za določanje odziva na okus hrane. V člankih so tudi ugotovili, da je ob zaporednih okušanjih korelacija med obraznimi izrazi in prijetnostjo okusa vse manj izrazita [3] in da so opazne tudi razlike v intenzitetah med različnimi vrstami hrane [5].

2. METODA ZA PREPOZNAVANJE ČUSTEV NA OBRAZU

Našo metodo smo implementirali v C++, z uporabo knjižnice OpenCV in obsega: detekcijo obraza in lokacije oči s Haarovi značilkami, poravnavo obraza glede na zaznano lokacijo centra oči, lokalne binarne vzorce (ang. local binary pattern, LBP) za izločanje značilk, metodo glavnih komponent (ang. principal component analysis, PCA) za zmanjšanje dimenzionalnosti, pri čemer smo ohranili 95% variance in metodo podpornih vektorjev (ang. support vector machine, SVM) z uporabo RBF jedra in klasifikacijo po strategiji "eden proti ostalim". Poravnava oči je bila izvedena z afino transformacijo tako, da sta centra oči na istih lokacijah. Uporabili smo LBP različico z uniformnimi vzorci, soseščino smo določili z 8 sosednimi elementi v radiju 1. Na sliki obraza smo izračunali histograme v mreži 10×10, tako da je končna dolžina vektorja značilk za sliko obraza znašala 5900. Pri klasifikaciji z metodo SVM smo za iskanje najprimernejših parametrov s prečnim preverjanjem uporabili metodo *auto_train* iz knjižnice OpenCV.

3. PREIZKUS METODE NA STANDARDNIH BAZAH

Predlagano metodo smo najprej preizkusili na dveh standardnih podatkovnih bazah, ki vsebujeta prikaz več različnih čustev: Cohn Kanade CK+ [7] in GEMEP-FERA 2011 [8].

3.1 Rezultati na bazi Cohn Kanade CK+

Baza Cohn Kanade CK+ vsebuje 327 posnetkov v katerih nastopa 118 oseb, od tega 82 ženskega in 36 moškega spola, osebe pripadajo različnim rasnim skupinam. Osebe

izražajo eno izmed sedmih čustev: jeza, žalost, veselje, strah, gnus, presenečenje in prezir. Posnetek je predstavljen kot zaporedje sličic in v njem nastopa zgolj ena oseba. Na začetku vsakega posnetka ima oseba na obrazu nevtralen izraz, ki nato postopoma preide v določeno čustvo, intenziteta izražanja čustva pa je največja na koncu posnetka. Osebe v bazi so posnete frontalno, v kontroliranem okolju z manjšimi premiki glave. Baza je zato za analizo manj zahtevna.

Baza nima točno določene testne in učne množice, zato smo ju določili sami. Za vsak posnetek smo izbrali tri sličice s konca posnetka, ki izražajo čustvo najbolj intenzivno. Tako sestavljene skupine treh slik smo naključno razdelili v učno in testno množico. V učni množici smo uporabili 876 posameznih sličic, v testni množici pa 84 sličic. Sličice iz posameznega posnetka so lahko zgolj v testni ali zgolj v učni množici. Izbrane slike, kjer naša metoda ni mogla zaznati oči, smo ročno zamenjali s slikami z zaznanimi očmi, ki nastopajo prej v posnetku in se na njih še zmeraj izraža čustvo. Težave z zaznavanjem oči smo imeli, ker osebe pri intenzivno izraženih čustvih pogosto imele oči zaprte. Na testni množici je naša metoda dosegla zelo dobre rezultate: 96.42% natančnost. Kar pomeni 81 pravilno napovedanih izrazov, pri 3 slikah pa ni bilo zaznano nobeno čustvo.

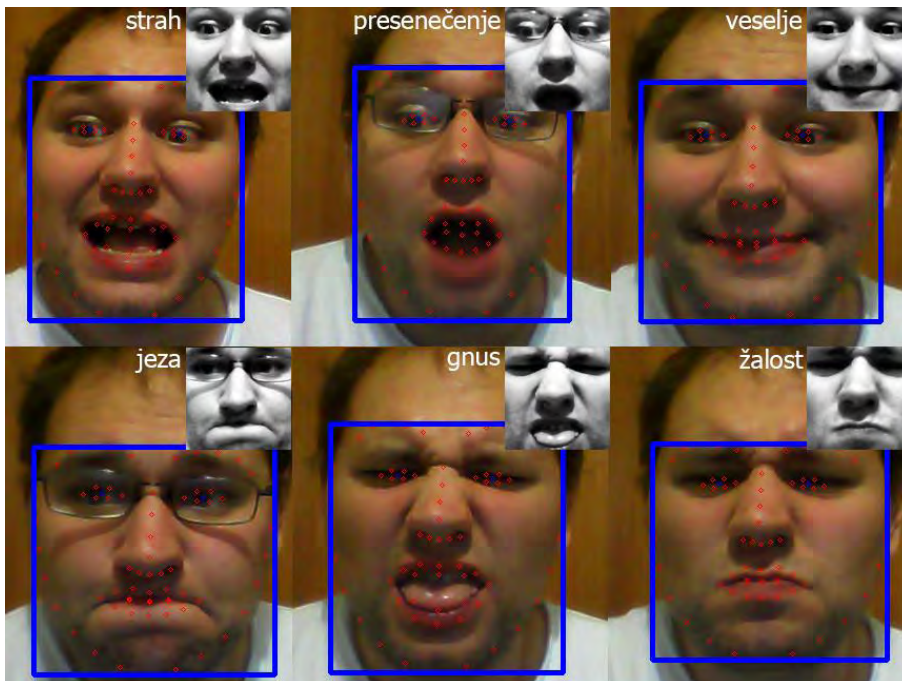
3.2 Rezultati na bazi GEMEP-FERA 2011

GEMEP-FERA je del baze GEMEP, ki se je uporabil na tekmovanju FERA-2011 (Facial Expression Recognition and Analysis challenge 2011) [8]. Baza GEMEP vsebuje posnetke 18 različnih čustev, katere so prispevali poklicni igralci, zato se posnetki smatrajo za spontane. Čustva v posnetkih so izražena z različno intenziteto, osebe v posnetkih se tudi veliko več premikajo, najbolj izrazito je premikanje glave. Zato se baza smatra za zahtevnejšo. V posnetkih vedno nastopa zgolj ena oseba, posnetki so zajeti v kontroliranem okolju. Baza GEMEP-FERA vsebuje 155 posnetkov v učni množici in 134 posnetkov v testni množici, v posnetkih nastopa skupno sedem različnih igralcev, obeh spolov in različnih starosti, v posnetkih je prisotnih 5 čustev (veselje, jeza strah, žalost in olajšanje). Posnetke smo pretvorili v zaporedje sličic, končna napoved za nek posnetek je bila čustvo z največjim številom napovedi po sličicah posnetka. Končni rezultat je torej število pravilno napovedanih posnetkov V 3 posnetkih testne množice ni bil nikoli zaznan obraz z obema očesoma, zato smo jih šteli kot nepravilne, sicer so bile končne slike obrazov dovolj natančne. Rezultat naše predlagane metode na tej bazi je: 79.85% natančnost, od tega 84.81% natančnost na osebah, ki so nastopile v učni množici in 72.72% natančnost na osebah, ki nastopajo samo v testni množici.

3.3 Preizkus s posnetki spletne kamere

Ker smo nameravali metodo določanja čustev na obrazu preizkusiti na lastnih posnetkih s spletno kamero, smo našo metodo preizkusili tudi na tovrstnih posnetkih, da bi ugotovili, ali je metoda dovolj natančna (Slika 1). Ugotovili smo, da je detekcija oči s Haarovi značilkami premalo natančna. Zato smo jo sprva nadomestili z omejenim lokalnim modelom (angl. constrained local model, CLM) [9], kasneje pa z detektorjem obraznih značilk iz knjižnice *dlib*, ki temelji na gradientemu boostingu [10]. Detektor obraznih značilk vsakič posebej zazna lokacije obraznih značilk na zaznani sliki obraza. Detekcija obraza v knjižnici *dlib* temelji na rabi histograma orientiranih gradientov (ang.

histogram of oriented gradients, HOG). Centra oči, ki se uporabita za poravnavo, smo določili kot povprečje napovedi obraznih značilnk okoli očesa.



Slika 1: Prepoznavanje čustev na posnetkih spletne kamere. Za iskanje značilnic na obrazu je uporabljena metoda CLM [9]. Metoda za prepoznavo čustev je bila naučena na učni množici Cohn Kanade CK+. Zgoraj desno so normalizirane slike obrazov namenjene prepoznavi čustev.

4. PRIPRAVA PODATKOVNE BAZE

Ker nismo našli prosto dostopne baze posnetkov okušanja hrane smo jo pripravili sami. Naša baza vsebuje 135 posnetkov s petimi različnimi osebami. V posnetku vedno nastopa samo ena oseba. Tri osebe so moškega spola in dve ženskega, starost oseb v posnetkih se giblje od 28 do 71 let. Za vsako osebo je na voljo od 22 do 30 posnetkov. Snemali smo tako prizore okušanja hrane, kot okušanja pijače, ki smo jih kasneje ločeno testirali. Posnetke smo snemali pri enakomerni osvetlitvi, tako da je bil obraz okuševalcev frontalno dobro viden. Obraz okuševalca je bil v posnetku postavljen na sredino. Izbrano ozadje je bilo monotono brez motečih predmetov. Ker naš sistem ni sposoben avtomatsko ugotoviti, ali je obraz zakrit bodisi z roko, posodo ali hrano, smo zakrivanje obraza med okušanjem poskušali kar se da minimizirati.

Vsak posnetek se začne pred okušanjem hrane oziroma pijače. Pri okušanju pijače so okuševalci pri pitju uporabili slamico zato, da bi bilo pri pitju čim manj prekrivanja obraza. Okuševalci so spili določeno količino pijače, umaknili slamico in pustili pijačo

par sekund v ustih zato, da bi se na obrazu jasno videl odziv na okus. Obraz so okuševalci imeli med snemanjem frontalno obrnjen proti kameri. Pazili smo, da se okuševalci pri vnosu pijače niso sklanjali naprej, temveč so si pijačo prinesli z roko bližje k obrazu in da so po zaužitju čim prej odmaknili slamico iz obraznega dela. Sicer smo se pri snemanju trudili, da so bili izrazi čim bolj spontani. Pri zaužitju hrane so okuševalci vnesli hrano večinoma z rokami. Tudi tukaj smo pazili, da je bilo pri zaužitju, prekrivanje obraza čim krajše. Pazili smo tudi, da so bili obrazi snemani frontalno in da se okuševalci pri zajemu hrane niso sklanjali naprej. Pri okušanju hrane smo okuševalcem dopustili, da so hrano žvečili poljubno dolgo. Posledično so bili posnetki okušanja hrane daljši in s tem množica slik pri testiranju večja. Zato, da posnetki okušanje hrane niso bili predolgi, smo uporabili manjše kose hrane. Pri snemanju smo uporabili različne vrste hrane in pijače. Ob koncu okušanja so okuševalci podali oceno okusa hrane oz. pijače na lestvici od 1 do 7, kjer 1 pomeni zelo neprijeten okus, 7 pa zelo prijeten, 4 pa nevtralen okus. Posnetke z ocenami od 1 do 3 smo potem obravnavali kot negativne odzive, posnetke z ocenami od 4 do 7 pa kot pozitivne odzive. Pri snemanju smo za vsako osebo posneli negativne in pozitivne odzive, tako za pitje pijače, kot tudi za okušanje hrane. V večini primerov smo pri posamezni osebi posneli dva posnetka z določeno vrsto hrane oziroma pijače. Bazo sestavlja 66 posnetkov okušanja pijače in 69 posnetkov okušanja hrane.

5. PREIZKUS NA PRIPRAVLJENI BAZI

Posnetke smo najprej pretvorili v zaporedje sličic velikosti 200×200 slikovnih elementov. Na dobljenih zaporedjih sličic smo potem ročno določili območja po zaužitju hrane, ki smo jih potem uporabili pri klasifikaciji. Bazo smo ovrednotili z večkratnim trikratnim prečnim preverjanjem, ločeno smo obravnavali množici okušanja pijače in hrane. Eno izmed treh skupin smo uporabili kot učno in ostali dve kot testni množici. Tako je bil vsak posnetek dvakrat v testni množici. V vsaki iteraciji prečnega preverjanja smo posnetke naključno razdelili v 3 skupine tako, da je bilo za vsako osebo število pozitivnih in negativnih odzivov v skupinah čimbolj enakomerno razporejeno. Za klasifikacijo smo uporabili en klasifikator SVM z jedrom RBF. Končen odziv posnetka je bil določen glede na prevladujočo napoved po sličicah. Končni rezultati prečnih preverjanj smo zaokrožili na celo število. Pri pijači je povprečna natančnost z desetimi prečnimi preverjanji 92%, najnižja dosežena natančnost je bila 87%, najvišja pa 94%. Pri hrani je povprečna natančnost z desetimi prečnimi preverjanji 81%, najnižja dosežena natančnost je bila 76%, najvišja pa 84%.

Podrobno smo analizirali tudi napake. Tako pri pijači kot pri hrani je velik delež napak predstavljajo zgolj nekaj posnetkov. Pri pijači je 44 posnetkov vedno brez napak, torej točno dve tretjini, medtem, ko je bilo pri hrani takih posnetkov slaba tretjina. Pri hrani prevladujejo narobe napovedani posnetki z negativnim odzivom. Za te posnetke je značilno, da je za razliko s posnetki negativnega odziva, ki so pravilno klasificirani, na teh posnetkih sploh težko zaznati negativen odziv na obrazu (Slika 2).



Slika 2: Posnetki negativnega odziva na hrano; zgornje tri slike prikazujejo izrazit negativni odziv, pri spodnjih treh slikah pa negativni odziv ni izrazil.

Pri pijači in pri hrani smo identificirali dva posnetka z negativnim odzivom, ki sta bila velikokrat narobe klasificirana. Oba posnetka sta bila druga oziroma tretja zaporedna posnetka snemanja okušanja iste pijače oziroma hrane. V primerjavi s predhodnimi posnetki, na teh posnetkih negativni odziv na obrazu ni zelo izrazit, saj se je okuševalec najbrž navadil na neprijetni okus. Tako pri pijači, kot tudi hrani, smo pri posnetkih z negativnim odzivom opazili še eno slabost naše metode. Posnetki so sicer prikazovali izrazito opazen negativni odziv na okus, vendar je ta odziv nastopil šele z zamikom. Zato je bil posnetek napačno označen kot pozitiven odziv. Pri pijači je bil en posnetek sicer zmeraj pravilno napovedan, vendar je bilo kar okoli 46% sličic posnetka zmeraj napačno napovedanih, ravno zaradi zakasnelega negativnega odziva.

6. ZAKLJUČEK

V članku smo prikazali uporabo metod za prepoznavo obraznega izraza za napovedovanje ali je osebi okus zaužite hrane ali pijače všeč. Na lastni bazi posnetkov smo dobili zelo dobre rezultate na množici pitja pijače in dobre rezultate na množici okušanja hrane. Pri pijači je bila povprečna natančnost 92%, pri hrani pa 81%. Podrobno smo tudi analizirali napake. Ugotovili smo, da se večina napak pojavi pri analizi posnetkov okušanja hrane, kadar je na posnetku neprijeten odziv na okus hrane, toda izraz na obrazu okuševalca ni dovolj izrazit.

Možno izboljšavo metode vidimo v avtomatski izbiri sličic po zaužitju hrane s pomočjo metod za prepoznavanje prekrivanja delov obraza, predvsem na območju okoli ust.

Uporabili bi lahko metode za istočasno zaznavanje lokacije in prekrivanja obraznih značilnik [11, 12] ali za zaznavanje prekrivanja glede na spremembe v vektorskem polju slike [13, 14]. Za prepoznavanje prekrivanja bi lahko uporabili tudi naprednejše senzorje kot so Kinect. Bazo posnetkov, bi bilo potrebno dopolniti z večjim številom posnetkov, lahko tudi v zahtevnejših pogojih. Tudi samo metodo za prepoznavo izrazov bi bilo smiselno preveriti z naprednejšim postopkom normalizacije in značilniki kot so LPQ, HOG, PHOG, LBP-TOP in LPQ-TOP.

LITERATURA

1. K. Wendin, B.H. Allesen-Holm, WLP. Bredie (2011), Do facial reactions add new dimensions to measuring sensory responses to basic tastes?, *Food Quality and Preference*, št. 4, zv. 22, str. 346-354.
2. R.A. de Wijk, V. Kooijman, R.H. Verhoevenb, N.T. Holthuysen, C. de Graaf (2012), Autonomic nervous system responses on and facial expressions to the sight, smell, and taste of liked and disliked foods, *Food Quality and Preference*, št. 2, zv. 26, str. 196-203.
3. R.A. de Wijk, W. He, M.G. Mensink, R.H. Verhoeven, C. de Graaf (2014), ANS responses and facial expressions differentiate between the taste of commercial breakfast drinks, *PLoS one*, št. 4, zv. 9.
4. L. Danner, L. Sidorkina, M. Joechl, K. Duerrschmid (2014), Make a face! Implicit and explicit measurement of facial expressions elicited by orange juices using face reading technology, *Food Quality and Preference*, zv. 32, str. 167-172.
5. G. Juodeikiene, L. Basinskiene, D. Vidmantiene, D. Klupsaite, E. Bartkiene (2014), The use of face reading technology to predict consumer acceptance of confectionery products, *V 9th Baltic Conference on Food Science and Technology "Food for Consumer Well-Being" FOODBALT 2014*, str. 276-279.
6. <http://www.noldus.com/human-behavior-research/products/facereader>
Program FaceReader za prepoznavo 6 osnovnih čustev.
7. P. Lucey, J.F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, I. Matthews (2010), The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression, *V Computer Vision and Pattern Recognition Workshops (CVPRW)*, str. 94-101.
8. M.F. Valstar, M. Mehu, B. Jiang, M. Pantic, K. Scherer (2012), Meta-analysis of the first facial expression recognition challenge, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, št. 4, zv. 42, str. 966-979.

9. J.M. Saragih, S. Lucey, J.F. Cohn (2009), Face alignment through subspace constrained mean-shifts, *IEEE 12th International Conference on Computer Vision*, str. 1034-1041.
10. V. Kazemi, J. Sullivan (2014), One millisecond face alignment with an ensemble of regression trees, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, str. 1867-1874.
11. G. Ghiasi, C.C. Fowlkes (2014), Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, str. 1899-1906.
12. X.P. Burgos-Artizzu, P. Perona, P. Dollár (2013), Robust face landmark estimation under occlusion, *IEEE International Conference on Computer Vision (ICCV)*, str. 1513-1520.
13. M. Mahmoud, R. El-Kaliouby, A. Goneid (2009), Towards communicative face occlusions: machine detection of hand-over-face gestures, *Proceedings of the 6th International Conference on Image Analysis and Recognition (ICIAR)*, str. 481-490.
14. J. Xu, X. Zhang (2015), A Real-Time Hand Detection System during Hand over Face Occlusion, *International Journal of Multimedia and Ubiquitous Engineering*, št. 8, zv. 10, str. 287-302.

PREPOZNAVA POSLOVNIH VIZITK IN KLASIFIKACIJA BESEDILA

Jaka Konda, Peter Peer

Fakulteta za računalništvo in informatiko, Univerza v Ljubljani

Večna pot 113, 1000 Ljubljana

<http://www.fri.uni-lj.si>

jaka@konda.si, peter.peer@fri.uni-lj.si

POVZETEK: *Poslovne vizitke v fizični papirnati obliki so še vedno glavni način izmenjave kontaktnih podatkov v 21. stoletju. Članek opisuje zaznavanje poslovnih vizitk in besedila na fotografiji, korake za boljšo prepoznavo s pomočjo optične razpoznavne znakov ter na koncu klasifikacijo besedila. Primerjava naše rešitve s programoma Abby Business Card Reader in CamCard, ki sta vodilna v industriji, pokaže, da so rezultati naše rešitve zelo spodbudni.*

1. UVOD

Kljub temu da imamo v 21. stoletju ogromno elektronskih medijev, ki močno olajšajo izmenjavo kontaktov z ostalimi, še vedno prevladujejo klasične papirnate vizitke z imenom, priimkom, telefonsko številko in podjetjem. Zamudni in odvečni del je pretvorba teh podatkov v digitalno obliko, kjer imajo največjo dodano vrednost. Človek je pri prepoznavi polj izvrsten. Ne glede na jezik vizitke je zmožen zgolj iz metapodatkov in vzorcev pravilno klasificirati vso besedilo.

Cilja našega projekta sta bila zaznava kontaktne vizitke na sliki in klasifikacija vseh polj ter se pri tem po natančnosti ter hitrosti čimbolj približati sorodnim rešitvam, npr. znanemu komercialnemu produktu Abby Business Card Reader [1] in brezplačnemu CamCard [2], ki sta bila uporabljena tudi za evalvacijo naše rešitve. Rešitev je implementirana v programskem jeziku C++ in na voljo na GitHubu [3]. Celoten algoritem se zgleduje po človeškem zaznavanju vizitke in razvrščanju besedila. Kot pomožni knjižnici sta bili uporabljeni OpenCV [4] za obdelavo slike in zaznavo vizitke, v drugem delu pa Tesseract [5], znana odprtokodna knjižnica za prepoznavo besedila na slikah.

2. PREDLAGAN ALGORITEM

Algoritem ima tri osnovne korake: iskanje vizitk na fotografiji, iskanje in prepoznavo besedila, klasifikacija besedila. Celoten algoritem temelji na vrednostih parametrov dobljenih eksperimentalno iz nabora petdeset poslovnih vizitk. Slike vizitk v tem članku so zaradi varstva osebnih podatkov ustrezno zamegljene in obdelane.

2.1 Iskanje vizitke na sliki

Pomemben del za boljši rezultat pri prepoznavi besedila na sliki je predprocesiranje. Prepoznavna se najboljše obnese, če je slika pravokotna na besedilo. Da bi to dosegli, je potrebno vizitko na sliki najprej najti.

Za korak predprocesiranja so bile vse vizitke poenotene na enako velikost tako, da je bila daljša stranica velika 800 slikovnih pik. Sledilo je iskanje robov z metodo Canny [6]. Meji sta bili samodejno določeni z izračunom mediane črno-bele (sivinske) slike in prišteti konstantami za nekoliko boljši rezultat. Pred tem je bil za boljše delovanje metode Canny uporabljen filter *box blur* [7], ki je odstranil šum. Najdeni robovi so bili odebeljeni s pomočjo $3 * 3$ matrike polne enic, za lažje iskanje le-teh v naslednjem koraku.

Označene robove je bilo potrebno pretvoriti v uporabno strukturo. Uporabljena je bila funkcija `HoughLinesP` [8] iz `OpenCV`-ja, ki te robove pretvori v tabelo daljic, začetno in končno točko. Zaradi odebeljenih robov lahko funkcija vrne podvojene rezultate, torej dve daljici, ki imata začetni in končni točki zelo blizu skupaj. Dvojniki so bili poiščani z mejo oddaljenosti in združeni s pomočjo povprečenja točk. Podobno je bilo narejeno z daljicami, ki ležijo na isti premici.

Nad najdenimi daljicami so bila izračunana vsa presečišča. Daljice so bile upoštewane kot premice. Med računanjem se je presečišča sproti zavračalo. Zavržena so bila pod pogoji, če kot sekanja ni bil med $90^\circ \pm 16^\circ$, če je bila točka na sredi ene izmed daljic, ali zelo daleč stran od obeh daljic (slika 1).



Slika 1: Levo najdena presečišča pred filtriranjem, desno po filtriranju.

Med računanjem presečišč je bil sproti zgrajen še graf presečišč. Vsako vozlišče je predstavljalo presečišče, povezave pa pripadajoči premici. Vsak cikel, torej kjer se iz poljubne točke pride nazaj v isto točko v n korakih, predstavlja nek n -kotnik. Ker so vizitke v večini primerov štirikotne, nas zanimajo cikli dolžine štiri. Vsi takšni unikatni cikli so poiščani z zavračanjem, kot pri iskanju presečišč. Pogoji so razmerje stranic, ki ne sme biti več kot 1:4, vzporedni stranici morata biti približno enako dolgi in imeti naklon različen za največ 20° . Največji štirikotnik, ki zadosti pogojem, je izbran kot potencialna vizitka.

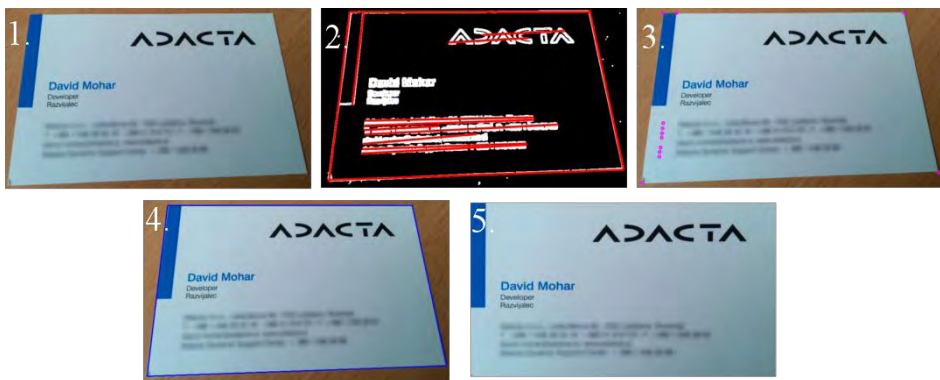
Če takšen štirikotnik ni obstajal, se je naivno štel, da je vizitka zlita skupaj z ozadjem. V tem primeru se je s pomočjo MSER (Maximum Stable Extremal Regions) [9] našlo dele besedila in poiskalo skrajne točke (spodaj, zgoraj, levo in desno), ki zajemajo najdene dele.

Zadnji korak pri iskanju vizitke je funkcija WarpPerspective [10] iz OpenCV, ki poljuben štirikotnik transformira tako, da je pravokoten na pogled uporabnika (slika 2). Celoten algoritem deluje za iskanje kateregakoli štirikotnika na sliki, dokler je le-ta glavni objekt. Z manjšo predelavo bi algoritem deloval tudi za iskanje poljubnega n -kotnika.



Slika 2: Levo prvotna slika, desno slika po iskanju štirikotnika in uporabi funkcije WarpPerspective.

Celoten opisan postopek je prikazan na sliki 3. V nadaljevanju bomo potrebovali tudi primer izhoda celotne naše rešitve, zato je ta podan v tabeli 1.



Slika 3: Postopek iskanja vizitke po korakih. Prva je vhodna slika, sledi iskanje robov, nato presečišč. V četrtem koraku poiščemo največji štirikotnik, na koncu pa ga transformiramo.

Tabela 1: Primer izhoda poslovne vizitke na sliki 3. Nekatera polja so cenzurirana zaradi varstva osebnih podatkov.

| POLJE | IZHOD |
|-----------------|--|
| Ime in priimek | David Mohar |
| Naziv | Razvijalec, Developer |
| Mobilni tel. | +386 31 ____ |
| E-poštni naslov | ____@____ |
| Podjetje | Adacta d.o.o. |
| Naslov | Leskoekova 9d, 1000 Ljubljana, Slovenija |
| Telefon | +386 1 548 ____ |
| Faks | +386 1 548 ____ |
| Spletna stran | www.adacta.si |

2.2 Iskanje in prepoznavna besedila

Posamezni deli besedila so bili obdelani ločeno, kar je pripomoglo k boljši prepoznavi besedila v koraku optične razpoznavne znakov in ohranjanju metapodatkov, kot sta velikost in lokacija besedila.

Iskanje besedila je bilo izvedeno s pristopom LPD (*License Plate Detector*) za zaznavanje registrskih tablic s slik oziroma s prirejenim pristopom LPD [11]. Alternativa, ki zna prinesiti boljše rezultate, je SWT (*Stroke Width Transform*) [12]. Ta dva algoritma sta tudi primerjana v podpoglavju 3.1.

Slika je bila, podobno kot pri iskanju vizitke, najprej poenotena, tokrat je bila daljša stranica dolga 400 slikovnih pik. Sledila je pretvorba slike v črno-bel format in iskanje gradientov. Če sta sosednji točki približno enako svetli, bo gradient majhen. Če je razlika velika, kot je običajno za lažjo berljivost besedila na sliki, bo tudi na sliki razlika velika. S tem dobimo v večini črno sliko in z belo izpostavljene posamezne črke ter še nekaj odvečnega šuma.

Rahle prehode in močne prehode se je ločilo z metodo Otsu [13] upragovljanja. Ta samodejno določi prag, kaj bo črno in kaj belo. Posamezne črke je bilo nato potrebno še smiselno povezati. Uporabljena je bila matrika $9 * 2$ polna enic, ki je bližnje črke povezala med seboj v horizontali. Postopek je viden na sliki 4.



Slika 4: Postopek iskanja regij besedila. Od leve proti desni: prvotna slika, iskanje gradientov, binarizacija in povezovanje črk.

Zadnji korak pri iskanju je bil narejen s pomočjo funkcije Contours [14] iz OpenCV. Najdena so bila vsa bela področja. Ta predstavljajo besedilo na vizitki. Zaradi šuma so bila vsa področja, ki so bila prenizka, prekratka, ali so imela več kot 40% površine črne, odstranjena.

Najdena področja so bila nato pretvorjena nazaj na istoležeča področja na prvotni večji sliki. Ta področja so bila izrezana in binarizirana z gradientno in Otsu metodo ter poslana v prepoznavo s pomočjo knjižnice Tesseract. Za boljši rezultat je bil omejen nabor dovoljenih črk le na črke angleške abecede z dodanimi šumniki, številkami in posebnimi znaki, kot so oklepaji, pomišljaj, &, poševnica in še nekaj drugih. Prepoznano besedilo je bilo popravljeno za najbolj običajne napake, kot so številke sredi besedila, npr. 5 je bila zamenjana s S in obratno. Prepoznano besedilo je bilo shranjeno skupaj z metapodatki, to so pisava, širina, višina, ali je odebeljeno, poševno in podčrtano. Velikost besedila je bila zaradi nekonsistentnosti Tesseracta vzeta kar v slikovnih točkah vhodne slike. Vsi ti metapodatki so uporabljeni pri naslednjem koraku, klasifikaciji besedila.

2.3 Klasifikacija besedila

Besedilo je bilo pred klasifikacijo prečiščeno. Izločilo se je vse, kar je vsebovalo premajhen delež alfanumeričnih znakov in samoglasnikov glede na besedilo, ali pa dolga zaporedja ponavljajočih se znakov. Vsa ta pravila so bila določena zgolj iz opazovanja vzorcev besedila, ki ga po prepoznavi vrne Tesseract. Ko je del besedila klasificiran, je ta popolnoma odstranjen, da olajša klasifikacijo prihodnjim korakom.

V besedilu so se nato poiskali vsi potencialni elektronski naslovi in spletne strani. Elektronski naslov se je iskal z regularnim izrazom, kjer je zaradi napak pri optični prepoznavi znakov zadosten vzorec za ujemanje že niz veljavnih znakov za elektronski naslov, ki vsebuje znak @. Če je rezultatov več, se je naivno štelo daljšega za elektronski naslov osebe. Na podoben način se je iskalo spletno stran, le da se je tu poiskalo začetne nize http, www ali veljavne končnice domen npr. si, com, eu ipd.

Sledila je klasifikacija številčnih polj. V besedilu se je našlo vsa zaporedja števk, ki so bila nato razdeljena v dve skupini. Daljša od šest znakov so predstavljala telefonske številke, krajša pa poštna in hišna številka. Telefonske številke so bile klasificirane glede na besedilo, ki se nahaja pred najdenimi nizi števk. Algoritem poizkuša najti nize, ki se pojavljajo na poslovnih vizitkah, kot so mobile, faks, fax, mob, tel, T, M, F in še nekaj drugih, v vrstnem redu od daljšega proti krajšemu, kar pripomore k pravilnejši klasifikaciji.

V zadnjem koraku so bili klasificirani še ime, priimek in pripadajoč naziv ter podjetje. V primeru, ko je najden elektronski naslov, se ime poizkuša najti iz dela, ki se nahaja pred @. Ujemanje poteka z drsečim oknom in Levenshteinovo razdaljo [15], ki primerja dva niza. Vsaka vrstica besedila se obdela ločeno z drsečim oknom. V okno se dodajajo besede, dokler se Levenshteinova razdalja manjša. Če se ta poveča, se odstrani zadnja beseda. S takšnim načinom pride do ujemanja, kjer je elektronski naslov sestavljen le iz

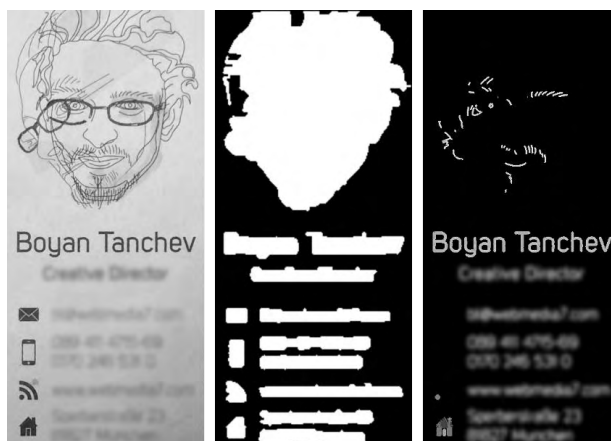
prve črke imena in celim priimkom, obratno, ali pa poljubne kombinacije obojega. Izbran je niz z najmanjšo Levensteinovo razdaljo. Na podoben način se je pridobilo ime podjetja, le da se je upošteval tudi URL spletne strani, če je le ta podan. V nasprotnem primeru se je pregledalo nize za število besed, dolžino le-teh, velikost in število besed, ki se začnejo z veliko začetnico. Algoritem teži k nizom, kjer se vsa besedila začnejo z veliko začetnico, imajo dve do tri besede in so nekoliko večje od ostalega besedila. Podobno velja tudi za ime podjetja, le da je tam glavni poudarek na velikosti, vse ostalo ima zelo majhen vpliv. Za naziv osebe in slogan podjetja so vzeti nizi dovolj blizu pravemu imenu in priimku in imenu podjetja, običajno pod temi podatki.

3. REZULTATI

3.1 Kvalitativna primerjava algoritma LPD s SWT

Oba algoritma sta, čeprav zelo različna, uporabljena za iskanje besedila na sliki. LPD [11] deluje kot opisano v poglavju 2.2, medtem ko SWT [12] zazna črte na sliki s Canny filtrom in nato gleda razmake med njimi. Kjer so razmaki med zaznanimi črtami zelo podobni, je najverjetneje črka. Za primerjavo je bil vzet algoritem, ki je na voljo na GitHubu [16].

Pri zaznavanju regij besedila sta se oba algoritma odrezala približno enako dobro, SWT nekoliko slabše. V splošnem je SWT širše aplikativen, saj je primeren tudi za uporabo v naravnih scenah brez predprocesiranja, vendar zaradi predhodnih korakov ta lastnost ni prišla toliko do izraza. Ta širša aplikativnost je najbolj vidna na vizitkah, ki imajo dodatno slikovno gradivo (slika 5) ali pa bolj eksotično tipografijo.



Slika 5: Levo prvotna slika, na sredini rezultat algoritma LPD in desno SWT.

SWT algoritem je imel največ težav, če slika ni bila enakomerna osvetljena, je bilo besedilo premalo kontrastno ali pa so bile črke pretanke. V tem primeru so bile izpuščene posamezne črke ali pa sploh ni bila nobena zaznana, kot je razvidno na sliki 6.



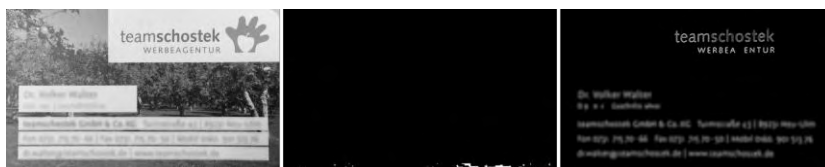
Slika 6: Levo prvotna slika, na sredini rezultat algoritma LDP in desno SWT. SWT da slabši rezultat zaradi slabše osvetlitve in tankih črt črk.

Pri zaznavanju regij se je SWT obnesel bolje v primerih, kjer je bil velik del vizitke slika. LPD je zaznal vse prehode na sliki in posledično je nastal velik povezan element, ki je bil zaradi filtrov odstranjen (sliki 7 in 8).

V pod poglavju 3.2 smo primerjavo delali ob uporabi LPD algoritma.



Slika 7: Zaznana področja teksta pred filtriranjem na vizitki na sliki 8.



Slika 8: Levo prvotna slika, na sredini rezultat algoritma LPD in desno SWT. LPD zaradi ozadja poveže celotno področje, ki je nato odstranjeno zaradi filtrov.

3.2 Kvantitativna primerjava s sorodnimi rešitvami

Našo rešitev smo primerjali s programom Abby Business Card Reader [1] in CamCard [2]. Obe sta znani in zelo priljubljeni rešitvi v različnih trgovinah z aplikacijami.

Pri testiranju uspešnosti se je ocenjevala natančnost zaznane vizitke na sliki (20% celotne ocene), število napak v besedilu in pravilno klasificirana polja (80% celotne ocene). Ocenjevalo se je po kriteriju v tabeli 2 in predstavlja 80% celotne ocene. Vsaka napaka v besedilu je odbila 1% in v telefonskih številkah 2%. Če je na vizitki kakšno polje manjkalo (recimo, da mobilni telefon ni bil zapisan), so bile ostale vrednosti normalizirane tako, da je seštevek vseh dejansko prisotnih polj še vedno 80%.

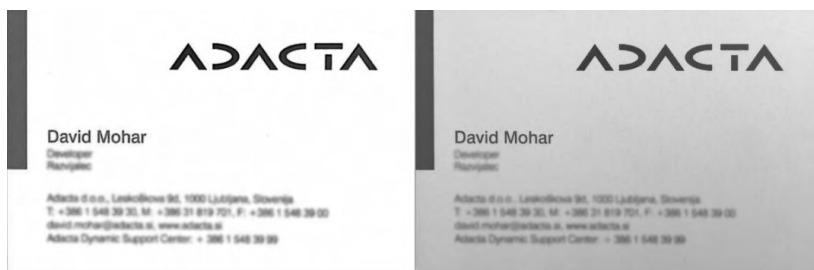
Tabela 2: Kriterij vrednotenja algoritmov pri klasifikaciji polj.

| <i>POLJE</i> | <i>VREDNOST (%)</i> |
|--------------------------------------|---------------------|
| Ime in priimek | 16 (8 + 8) |
| Naziv osebe | 8 |
| Podjetje | 12 |
| Slogan | 8 |
| Naslov | 16 |
| Mobilni, stacionarni telefon in faks | 12 (4 + 4 + 4) |
| E-pošta in naslov spletne strani | 8 (4 + 4) |
| Skupno: | 80 |

Ocena zaznane vizitke je predstavljala preostalih 20% skupne ocene. Vsak pravilno zaznan rob (zgornji, desni, spodnji in levi) predstavljajo četrtno ocene oz. 0%, če na zaznani vizitki manjka besedilo. Na testni množici so vse vizitke bile pravilno zaznane.

V testnem naboru je bilo dvajset vizitk (seveda različnih od učne množice, kjer jih je bilo 50), vsi algoritmi so dobili enako sliko ob dobri svetlobi, ki igra pomembno vlogo.

Vsi omenjeni algoritmi so si po hitrosti in zaznavanju vizitke na sliki skoraj enakovredni. CamCard izmed obeh algoritmov deluje nekoliko hitreje od ostalih dveh, medtem ko naša implementacija bolje najde vizitko na sliki z veliko šuma kot ostala algoritma. Oba sta tudi veliko bolje povečala kontrast po zaznani vizitki, kar zmanjša napake pri prepoznavi besedila (slika 9). V ostalih pogledih so bili med iskanjem vizitke in besedila vsi enakovredni.



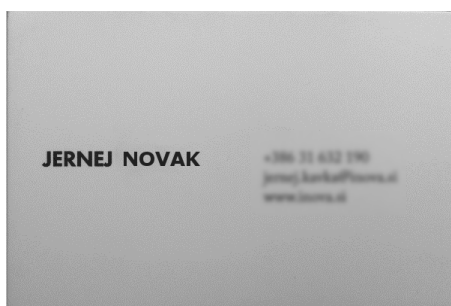
Slika 9: Levo slika pred korakom prepoznave besedila programa Abby in CamCard, desno našega algoritma.

Vsi testi so zaradi programa CamCard, ki je na voljo le za mobilne platforme, tekli na telefonu Nexus 5, kjer je bil nameščen operacijski sistem Android 5.1.1. Na voljo je bilo 2GB pomnilnika, kjer 500MB zasede OS, in procesor s štirimi jedri s frekvenco 2.3GHz. Vsi algoritmi so imeli na voljo zadostno količino pomnilnika in v ozadju ni delovala nobena aplikacija. Vsi programi so za prepoznavo vizitke uporabljali le eno jedro, Abby je pa edini poleg procesorskega jedra uporabljal tudi grafično kartico.

Tabela 3: Primerjava algoritmov po uspešnosti in hitrosti.

| <i>USPEŠNOST</i> | <i>ALGORITEM</i> | <i>POVPREČEN ČAS IZVAJANJA</i> |
|------------------|---------------------------|--------------------------------|
| 89,05% | Abby Business Card Reader | 8,978s |
| 83,90% | naš algoritem | 9,563s |
| 55,25% | CamCard | 6,045s |

Rezultati primerjave algoritmov so zbrani v tabeli 3. Najslabše se je odrezal CamCard. Omenjen program je naredil največ napak pri prepoznavi; pravilno je recimo razločil faks le v primeru, da se je pred številko nahajala beseda fax. Sicer so bile vse telefonske številke klasificirane kot mobilni telefon. Prav tako je imel kar nekaj težav s klasifikacijo vizitk z manj vsebine (slika 10), kjer je program vrnil le napako.



Slika 10: Slika le z nekaj polji, kjer CamCard vrne le napako.

Ostala programa sta si bila v rezultatih bolj podobna. Abby je bil na večini področij nekoliko boljši. Zaznavanje regij besedila je imel malo boljše, kjer je naš program sem tja izpustil kakšno začetno ali končno črko. Največja razlika se je poznala pri prepoznavi zahtevnejših besedil in logotipov, kjer je Abby naredil zelo malo napak v primerjavi z našim in je lahko posledično naredil boljšo klasifikacijo. Naslednja večja razlika je bila pri razvrščanju polj, kjer se je naš algoritem z iskanjem polj, katerih na vizitki ni, npr. vizitka ne vsebuje imena podjetja, izkazal za preveč agresivnega. Naš algoritem je v takšnem primeru izbral najbližje ujemanje, čeprav je bilo besedilo npr. del naslova, ki ni bil uvrščen nikamor. Abby teh težav ni imel in je v nekaj primerih podjetje nadomestil z domeno spletne strani.

4. ZAKLJUČEK

Predstavljen algoritem za razpoznavo besedila na poslovnih vizitkah sledi standardnemu trikorlačnemu pristopu: najprej zaznamo samo vizitko, nato poiščemo besedilo ter ga na koncu razpoznamo. Algoritem se je izkazal za zelo dobrega in konkurenčnega, kljub dejstvu, da je postopek močno odvisen od prejšnjega koraka. Zaradi ene napake v začetnem koraku lahko zatajijo vsi naslednji, kar pa je nemalokrat problem sistemov računalniškega vida.

Izboljšave so možne na vseh področjih. Pri prvem koraku se naš algoritem izkaže slabše od konkurenčnih v primerih, kjer je vizitka zlita z ozadjem. Konkurenčna algoritma sta uspešno zaznala robove, ki se jih težko razloči od ozadja, medtem ko je naš algoritem uporabil MSER za iskanje potencialnih regij na vizitkah. Pri iskanju regij besedila le-tega težje najde na vizitkah z razgibanim ozadjem, kot je bil recimo podan primer na sliki 7. Takšne primere bi se lahko zaznalo in se nato uporabilo algoritem SWT.

Največ napak pride pri zadnjem koraku na koncu verige, nepovezano s trikorlačnim pristopom in odvisnosti od prejšnjega koraka. Napake se pojavljajo predvsem pri razločitvi imen in priimkov od podjetij, kjer se naš algoritem zanaša zgolj na metapodatke. Tako težko loči besedi *Janez* in *Vizitka*, če se obe začneta z veliko začetnico. Možni izboljšavi bi bili dodan slovar imen in končnic podjetij, s katerimi bi algoritem lažje razločil med enim in drugim ter aktivna internetna povezava, kjer bi lahko dodatne informacije, npr. o podjetju, dobil s spleta. S pomočjo slovarjev in spleta bi se prav tako lahko zaznalo in rešilo agresivno klasifikacijo in klasifikacijo manjkajočih polj, saj bi se algoritem bolje zavedal pomena besedila. Slovarje prav tako uporablja Abby.

Pri algoritmu je prav tako na voljo še nekaj hitrostnih izboljšav. Najpočasnejši del je prepoznavna besedila s Tesseractom, ki zavzame več kot 90% vsega časa. Delovanje Tesseracta je bila v našem primeru izvedena le procesorsko, čeprav podpira tudi procesiranje na grafični kartici, kar bi najverjetneje prepolovilo čas izvajanja algoritma. Dodatna pohitritev programa bi se lahko naredila z deljenjem določenih delov algoritma med procesorska jedra. Primerni deli bi bili iskanje štirikotnikov na grafu presečišč, filtriranje besedila in iskanje z ujemanjem z drsečega okna.

5. LITERATURA

1. www.abbyy.com/business-card-reader-for-windows/
Abby Business Card Reader.
2. www.camcard.com
CamCard Business Card Reader.
3. www.github.com/jakakonda/Business-Card-Reader
Izvorna koda rešitve opisane v tem članku.
4. www.opencv.org
Odpriktokodna knjižnica OpenCV za računalniški vid.
5. www.github.com/tesseract-ocr/tesseract
Odpriktokodna knjižnica Tesseract za optično prepoznavo besedila (OCR).
6. J. Canny (1986), A Computational Approach To Edge Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, str. 679-69.
7. L. G. Shapiro, G. Stockman (2001), *Computer Vision, Prentice Hall*, str. 137-150.
8. R. Duda, P Hart (1972), Use of the Hough transformation to detect lines and curves in pictures, *Communications of the ACM*, vol. 15, str. 11-15.
9. J. Matas, O. Chum, M. Urban, T. Pajdla (2004), Robust wide baseline stereo from maximally stable extremal regions, *Image and Vision Computing*, vol. 22, str. 761-767.
10. O. Chum, T. Pajdla, P. Sturm (2005). The Geometric Error for Homographies, *Computer Vision and Image Understanding*, vol. 97, str. 86–102.
11. <http://stackoverflow.com/questions/23506105/extracting-text-opencv>
LPD algoritem uporabljen za zaznavo besedila na slikah.
12. B. Epsthein, E. Ofek, Y. Wexler (2010), Detecting text in natural scenes with stroke width transform, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, str. 2963-2970.
13. N. Otsu (1979), A threshold selection method from gray-level histograms, *Systems, Man and Cybernetics*, vol. 9, str. 62-66.
14. S. Suzuki, K. Abe (1985), Topological Structural Analysis of Digitized Binary Images by Border Following, *Computer Vision, Graphics, and Image Processing*, vol. 30, str. 32-46.
15. V. Levensthein (1966), *Binary codes capable of correcting deletions, insertions, and reversals, Soviet Physics Doklady*, vol. 10, str. 707-710.
16. <https://github.com/aperrau/detecttext>
SWT algoritem uporabljen za primerjavo z LPD.



SLOVENSKI PODJETNIŠKI SKLAD

Slovenski podjetniški sklad (v nadaljevanju Sklad oz. SPS) predstavlja osrednjo finančno podporo slovenskim mikro, malim in srednje velikim podjetjem (MSP) pri njihovem dostopu do finančnih sredstev na finančnem trgu.

KOGA FINANCIRAMO?

Sklad v letu 2016 izvaja dva vsebinska programa finančnih spodbud za **mikro, mala in srednje velika podjetja**, in sicer:

PROGRAM MLADI (podjetja mlajša od 5 let)

Kaj ponujamo njim?

Mladim podjetjem, ki imajo potencial rasti, nudimo finančne spodbude v vseh štirih fazah razvoja podjetja, vse od zagona pa do zrele rasti. Konkretno jim nudimo:

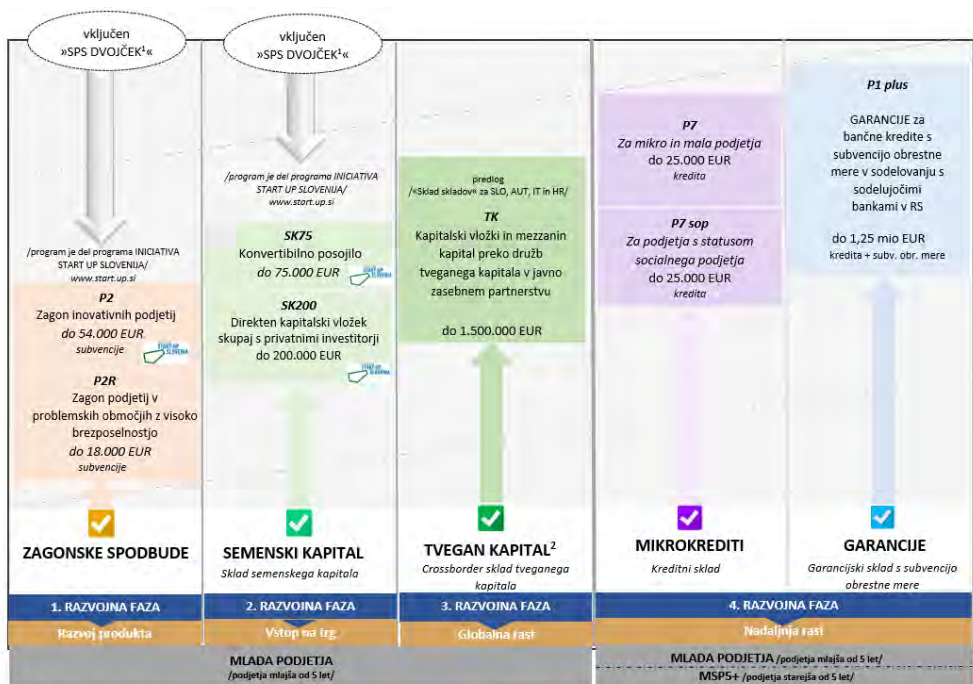
- **ZAGONSKE SPODBUDE** za zagon inovativnih podjetij v prvi razvojni fazi
- **SEMENSKI KAPITAL** za vstop podjetij na trg v drugi razvojni fazi
- **TVEGAN KAPITAL** za hitro globalno rast v tretji razvojni fazi
- **MIKROKREDITE** za tekoče in likvidno poslovanje v četrti razvojni fazi
- **GARANCIJE** za zavarovanje bančnih kreditov za nadaljnjo rast in razvoj v četrti razvojni fazi

PROGRAM MSP5+ (podjetja starejša od 5 let)

Kaj ponujamo njim?

MSP5+ podjetjem nudimo *produkte v četrti razvojni fazi* za tekoče poslovanje in nadaljnjo rast ustaljenih podjetij, ki donosno poslujejo, nimajo pa potenciala hitre rasti. Konkretno jim nudimo:

- **MIKROKREDITE** za tekoče in likvidno poslovanje ter
- **GARANCIJE** za zavarovanje bančnih kreditov za nadaljnjo rast in razvoj podjetij



*»SPS dvoječek« Kombinacija finančne podpore s vsebinsko podporo (mentorstvo, coaching usposabljanje,...)

²TVEGAN KAPITAL – program v pripravi, potencialno razpoložljiv v drugi polovici leta 2016

