

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 934

**SUSTAV ZA GEOMETRIJSKU
TRANSFORMACIJU I POVEZIVANJE
DIJELOVA SLIKE**

Antonio Soldo

Zagreb, lipanj 2015.

Zahvaljujem se svojoj obitelji što su mi uvijek bili podrška i što su uvijek vjerovali u mene, mentoru prof. dr. sc. Damiru Seršiću na savjetu i korisnim idejama, ekipi iz tvrtke divIT, posebice Darianu i Vedranu Škarici na uloženom trudu, stručnoj pomoći, ali i zabavnom društvu koje je izradu ovog rada činilo lakšom. I naposljetku, veliko hvala mojoj Janji jer je uvijek znala uputiti pravu riječ ohrabrenja i smirenja.

Sadržaj

1. Uvod.....	1
2. Oculus Rift i Google Cardboard	3
3. Kalibracija kamere.....	5
3.1. Geometrijsko iskrivljenje slike	5
3.1.1. Osnovne jednačbe.....	7
3.1.2. Određivanje homografije.....	7
3.1.3. Određivanje parametara	10
3.2. Operacije u prostoru boja.....	15
3.2.1. Histogram	15
3.2.2. Izjednačavanje histograma	16
3.2.3. Modeliranje histograma	19
4. Implementacija.....	23
5. Rezultati	27
6. Zaključak.....	32
7. Literatura.....	34
Privitak A	38

Popis slika

Slika 1. Komercijalna verzija Riffa (Michaels, 2015).....	3
Slika 2. Popis potrebnih dijelova (Statt, 2014).....	4
Slika 3. Složeni Cardboard (Statt, 2014)	4
Slika 4. Efekt pojedine distorzije na ravnu liniju (Hugemann, 2010.)	5
Slika 5. Slika sivih tonova i pripadni normalizirani histogram.....	15
Slika 6. Valjana transformacijska funkcija	16
Slika 7. Grafički prikaz modeliranja histograma (Gonzalez, 2002)	21
Slika 8. Slika prosječnog lica (Moses, 2011)	23
Slika 9. Područje interesa na pozadini	24
Slika 10. Histogram pozadine.....	25
Slika 11. Zaglađivanje maske.....	27
Slika 12. Ispravljanje distorzije	27
Slika 13. Izjednačavanje histograma pozadine.....	28
Slika 14. Izjednačavanje histograma kamere	28
Slika 15. Modeliranje histograma	28
Slika 16. Rezultat spajanja (1).....	29
Slika 17. Rezultat spajanja (2).....	30
Slika 18. Rad alata za praćenje kontura lica	31

1. Uvod

Virtualna stvarnost (dalje u tekstu VS) predstavlja trodimenzionalnu računalno generiranu okolinu kojom osoba može istraživati i upravljati. Ta osoba tada postaje dio virtualnog svijeta ili je potpuno uronjena u tu okolinu i, dok je ondje, u mogućnosti je upravljati objektima ili izvoditi niz akcija. Osoba najčešće nosi zaslon montiran na glavi (*engl.* Head-mounted display, HMD) ili naočale koje prikazuju trodimenzionalne slike kao dio njihovog iskustva (vrs. org, 2015). Postoji više metoda po kojima se može ostvariti virtualna stvarnost, a u nastavku slijedi kratki popis. Virtualna stvarnost temeljena na simulacijama (npr. simulatori vožnje auta/motora), VS temeljena na slikama avatara (omogućuje pridruživanje korisnika virtualnoj okolini u obliku računalno generiranog lika ili u obliku pravog videa), VS temeljena na projekcijama (podrazumijeva modeliranje stvarne okoline u svrhu navigacije robota), VS temeljena na radnoj površini (prikaz 3D virtualnog svijeta na običnom zaslonu, mnoge moderne računalne igrice mogu poslužiti kao primjer) te potpuno uranjajuća VS (hipotetska, najvjerojatnije će se ostvariti uporabom mozak-računalo sučelja).

Virtualna stvarnost može rekreirati čulna iskustva, koja uključuju virtualni okus, vid, njuh, zvuk i dodir. Iako ta tehnologija vuče svoje korijene iz 60-ih godina prošlog stoljeća, prvi pravi zamah tehnologija doživljava početkom 1990-ih kada je Howard Rheingold napisao dokumentarnu literaturu *Virtualna Stvarnost* kako bi demistificirao pojam virtualne stvarnosti, čineći ga dostupnijim slabije potkovanim istraživačima i entuzijastima (Rheingold, 1991). Nakon 1997. velik utjecaj na tehnologiju ima zajednica koja okuplja igrače videoigrica koja dodatno povećava interes za razvojem ove tehnologije (Zyda, 2005). Ipak, u 1990-im tehnologija se nije uspjela održati dovoljno dugo da stekne više pažnje, ponajprije zato što su uređaji za glavu bili nezgrapni, spojeni na poseban stalak čime se igračko iskustvo više svodilo na trčanje po višesmjernoj traci za trčanje. Također su takvi uređaji bili cjenovno nepristupačni čime se ograničavala strast razvojnih inženjera za izgradnju boljih aplikacija (Edelson, 2014).

Za uspjeh današnje virtualne stvarnosti najzaslužniji su pad cijene proizvoda, povećanje rezolucije prikaza slika te povećanje brzine razvoja popratne tehnologije. Danas je virtualna stvarnost tržište s najvećim potencijalom razvoja u idućih pet godina, gdje se predviđa da će upravo zbog videoigara do 2020. biti prodano 26,5 milijuna HMD uređaja. Takav razvoj znači da bi tehnologija imala godišnji rast razvoja od skoro 100 %. Drugim riječima, prodaja će u prosjeku svake godine biti dvostruko veća nego prijašnja (Rosoff, 2015). Takvim predikcijama trenutno najviše pridonosi tvrtka Oculus VR koju je 2014. godine kupio Facebook za dvije milijarde američkih dolara zahvaljujući svom proizvodu Rift. Odmah po izlasku Rifta, dolazi to tehnološkog uspona gdje tvrtke poput Samsunga, HTC-a, Sonya, Microsofta i Googlea predstavljaju slične proizvode. Posebno se izdvajaju Samsung i Google koji umjesto zasebnog zaslona koriste pametne telefone kao zaslone. Trenutno HMD uređaji omogućuju praćenje položaja glave korisnika kao i njegovog položaja u prostoru, čime se nudi širok spektar primjena od uporabe u edukaciji, igranju videoigara preko uporabe u vojsci i medicini sve do primjene u arhitekturi i graditeljstvu, mogućnosti su ogromne. No, nijedan uređaj trenutno ne omogućava animiranje lica osobe koja nosi HMD uređaj, i to iz praktičnih razloga – pola lica zauzima sam uređaj. To se može izbjeći ako se na vanjski dio uređaja montira kamera koja bi snimala donji dio lica pa bi se mogla dobiti parcijalna animacija ljudskog lica, ali i to predstavlja problem. Svi postojeći programski alati za praćenje kontura lica ne mogu raditi s pola lica, već isključivo s cijelom slikom lica. Ovaj rad zaobilazi upravo navedene probleme tako da spaja sliku donjeg dijela lica koju snima kamera montirana na vanjskom dijelu uređaja s unaprijed definiranim ljudskim licem. Rješenje koje se dobilo ne ovisi ni o jednom HMD uređaju i trenutno omogućava animaciju donjeg dijela ljudskog lica. U poglavlju 2. bit će objašnjena dva najpoznatija HMD uređaja, u poglavlju 3. bit će objašnjeno što je i zašto je potrebno provoditi operacije nad histogramima te će se objasniti postupak poništenja distorzije. U poglavlju 4. detaljno će se obrazložiti implementacija, u poglavlju 5. će biti predstavljeni rezultati, a u zadnjem poglavlju će se sve sumirati i raspraviti o mogućim poboljšanjima.

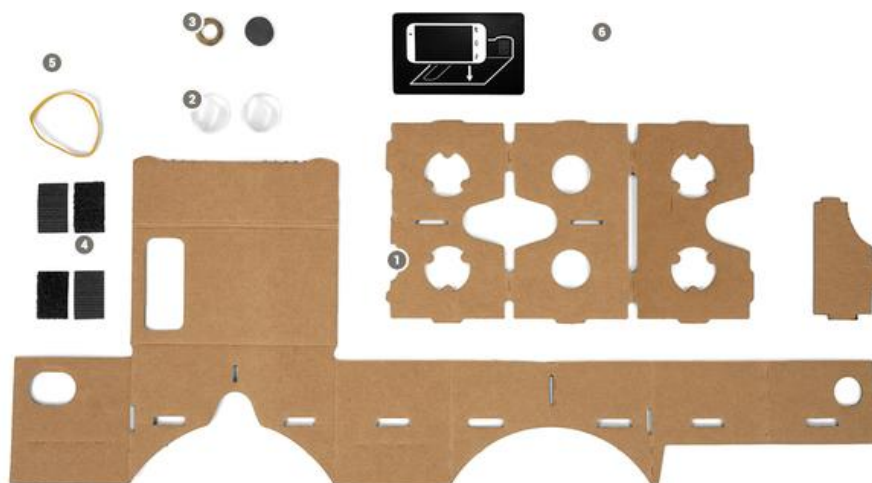
2. Oculus Rift i Google Cardboard

Oculus Rift VR trenutno je najpoznatiji i najnapredniji HMD uređaj koji se nalazi na tržištu. Osnivač i izumitelj Rifta je Palmer Luckey, a nastao je Palmerovom frustriranošću na neadekvatnost postojećih HMD uređaja u svrhu igranja videoigara zbog niskoga kontrasta, visoke latencije i maloga vidnog polja. Prvi prototip napravio je 2011. godine, a tijekom deset mjeseci napravio je još niz prototipova koji su istraživali mogućnosti poput 3D stereoskopije, bežičnosti i ekstremnog vidnog polja od 270°, ujedno smanjujući veličinu i težinu uređaja. Upravo zadnju verziju je primijetio John Carmack, napravio dodatna poboljšanja i prikazao demoverziju Doom 3 BFG Edition igrice na Electronic Entertainment Expo 2012. (Purchase, 2013). U tom trenutku se umjesto 40-50 ljudi zainteresiralo njih tisuće. Objavljena je Kickstarter kampanja s ciljem prikupljanja od \$ 250 000, a taj iznos je skupljen već prvog dana. Do kraja kampanje skupili su skoro deset puta veći iznos – \$ 2 437 429. Do danas su se pojavile dvije verzije uređaja za programere, a kao komercijalno dostupna verzija pojavit će se u prvom kvartalu 2016. godine. Komercijalna verzija prikazana je na slici 1, a koristi OLED zaslon s rezolucijom 2160x1200 (1080x1200 za svako oko), frekvenciju osvježavanja 90 Hz, za praćenje položaja glave ima na raspolaganju 6 stupnjeva slobode te će težiti manje od 440 g.



Slika 1. Komercijalna verzija Rifta (Michaels, 2015)

Potpuno drugačiji pristup virtualnoj stvarnosti napravio je Google razvojem Google Cardboard platforme. Nastao je kao cjenovno jeftin sustav kako bi povećao interes i potakao razvoj virtualne stvarnosti i njenih aplikacija u mobilnom svijetu. Google Cardboard nema posebni zaslon, već za to koristi zaslone pametnih telefona. Specifikaciju uređaja dizajnirao je Google, ali ne postoji službeni proizvođač ili dobavljač. Umjesto toga, Google je na svojim stranicama objavio listu potrebnih dijelova, sheme i upute za sastavljanje te time omogućio da ljudi samostalno slažu opremu. Listu potrebnih dijelova čine dijelovi kartona izrezani u precizne oblike, leće s 45 mm fokalnom duljinom, magnetne ili kapacitivne trake, čičak-traku, gumenu traku i opcionalnu NFC oznaku, a to je sve prikazano na slici 2.



Slika 2. Popis potrebnih dijelova (Statt, 2014)

Upravo je za potrebe ovog rada korišten model Cardboarda, a kako izgleda kada je model potpuno složen prikazano je na slici 3.



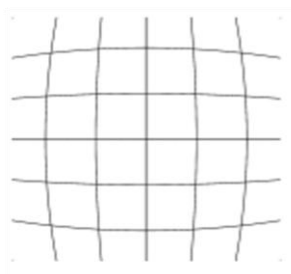
Slika 3. Složeni Cardboard (Statt, 2014)

3. Kalibracija kamere

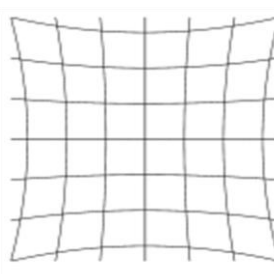
Kamere kao uređaji za akviziciju slike postoje već vrlo dugo, a s razvojem tehnologije postaju sve jeftinije i pristupačnije. Međutim, ta jeftinoća ima svoju cijenu – jeftinije kamere unose znatna iskrivljenja slike. Srećom, te vrijednosti su konstante koje se mogu matematički procijeniti pa se kalibracijom mogu poništiti neželjeni efekti (OpenCV, 2015). Uz kalibraciju parametara kamere, za potrebe ovog rada potrebno je izvršiti i kalibraciju boja. To je potrebno obaviti jer se spajaju dvije slike koje su dobivene na dva različita načina pod različitim osvjetljenjima. U potpoglavlju 3.1. bit će objašnjen postupak poništenja distorzije, dok će u potpoglavlju 3.2. biti objašnjena kalibracija boja.

3.1. Geometrijsko iskrivljenje slike

Pod pojmom distorzija misli se na odstupanje slike od pravocrtne projekcije (projekcija kod koje ravne linije u sceni ostaju ravnim u slici). Distorzije mogu biti nepravilne ili mogu slijediti mnoštvo uzoraka, ali najčešće susretane distorzije su radijalno simetrične. Dva najpoznatija tipa takvih iskrivljenja su „bačvasta distorzija“ (engl. Barrel distortion) i „jastučasta distorzija“ (engl. Pincushion distortion). Kod bačvaste distorzije, povećanje slike opada s udaljenošću od optičke osi. Tim efektom dobiva se slika koja izgleda kao da je omotana oko sfere (ili bačve, po tome je dobila i ime). Ravne linije koje ne prolaze kroz centar savijaju se dalje od centra. S druge strane, potpuno suprotno izgleda slika kod koje je vidljiva jastučasta distorzija. Ovdje povećanje slike raste s udaljavanjem od optičke osi, a linije koje ne prolaze sredinu su udubljene prema centru slike. Kako se manifestira pojedina distorzija prikazano je na slici 4.



a) Bačvasta distorzija



b) Jastučasta distorzija

Slika 4. Efekt pojedine distorzije na ravnu liniju (Hugemann, 2010.)

Radijalne distorzije nastaju zbog same karakteristike leće, ali uz radijalnu distorziju moguća je pojava i tangencijalne. Tangencijalna distorzija nastaje zbog odstupanja položaja ravnine leće od senzora kamere. Oba tipa izobličenja mogu se modelirati polinomijalnim modelom izobličenja, a to se može izraziti matematičkom formulom koja je poznata pod nazivom Brown (Brown, 1966) – Conrady (Conrady, 1919) model izobličenja (De Villiers, 2008):

$$x_u = x_d + (x_d - u_0)(K_1 r^2 + K_2 r^4 + \dots) + (P_1(r^2 + 2(x_d - u_0)^2) + 2P_2(x_d - u_0)(y_d - v_0)(1 + P_3 r^2 + \dots)) \quad (1)$$

$$y_u = y_d + (y_d - v_0)(K_1 r^2 + K_2 r^4 + \dots) + (2P_1(x_d - u_0)(y_d - v_0) + P_2(r^2 + 2(y_d - v_0)^2)(1 + P_3 r^2 + \dots)) \quad (2)$$

gdje vrijede oznake:

- (x_d, y_d) – koordinate iskrivljene točke
- (x_u, y_u) – koordinate ispravljene točke
- (u_0, v_0) – središte distorzije
- K_n - n – ti koeficijent radijalne distorzije
- P_n - n – ti koeficijent tangencijalne distorzije
- $r = \sqrt{(x_d - u_0)^2 + (y_d - v_0)^2}$

Bačvasta radijalna distorzija najčešće se prepoznaje po negativnom članu K_1 , dok će kod jastučaste radijalne distorzije taj član biti pozitivan. Jednom kada imamo definiran oblik distorzije, možemo poništiti njezine efekte. Problem je što tako definiran model nije linearan u pogledu r^2 , zbog čega se moraju koristiti iterativna numerička poboljšanja. Predstavljeni problem nije moguće riješiti odmah zato što sadrži nepoznanice. Parametri u_0 i v_0 nisu unaprijed poznati i moraju se proračunati, a pripadaju intrinzičnim parametrima kamere. Postupak proračuna intrinzičnih parametara bit će opisan u nastavku.

3.1.1. Osnovne jednađbe

Za proračun intrinzičnih parametara kamere u ovom radu koristi se algoritam koji je 2000. godine predstavio Zhengyou Zhang, a donijela je napredak u odnosu na postojeće metode za kalibraciju zato što zahtijeva prikaz planarnog uzorka s nekoliko različitih orijentacija (minimalno dvije). Kao planarni uzorak najčešće se uzima binarizirana slika šahovskog polja. Metoda prvo proračunava rješenje u zatvorenoj formi, a zatim ga nelinearno profinjuje na temelju kriterija najveće izglednosti (Zhang, 2000). Definirajmo prvo korištene oznake u ovom algoritmu.

Označimo 2D točku s $m = [u, v]^T$, a 3D točku s $M = [X, Y, Z]^T$. Ako s $\tilde{x}$ označimo umjetni vektor kojem dodajemo vrijednost 1 kao zadnji element, dobivamo $\tilde{m} = [u, v, 1]^T$ i $\tilde{M} = [X, Y, Z, 1]^T$. Veza između 3D točke M i njene slikovne projekcije m dana je relacijom:

$$s \tilde{m} = A[R \ t] \tilde{M} \quad (3)$$

gdje je s proizvoljni faktor skaliranja, (R, t) poznati pod nazivom ekstrinzični parametri kamere, gdje R predstavlja matricu rotacije, a t matricu translacije. Te matrice povezuju globalni koordinatni sustav s koordinatnim sustavom kamere. Matrica A predstavlja intrinzične parametre kamere i oblika je:

$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

gdje (u_0, v_0) predstavljaju koordinate glavne točke (engl. *principal point*), α i β predstavljaju faktore skaliranja osi u i v te γ predstavlja faktor asimetričnosti osi slike.

3.1.2. Određivanje homografije

Kako bismo odredili intrinzične parametre, potrebno je prvo odrediti matricu homografije koja povezuje ravninu modela s ravninom slike. Bez gubitka generalizacije, pretpostavimo da se ravnina modela nalazi točno u $Z=0$ u globalnom koordinatnom sustavu. Ako s r_i označimo i -ti stupac rotacijske matrice R možemo pisati sljedeće:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Ovdje ćemo i dalje koristiti istu notaciju za M , samo što će nam sada vrijediti $M = [X, Y]^T$ zato što je Z uvijek jednak nuli. Naravno, to se odnosi i na $\tilde{M}$ koji će biti zapisan kao $\tilde{M} = [X, Y, 1]^T$.

Sukladno tome, točka modela M i točka pripadne slike m vezani su preko matrice homografije H :

$$s\tilde{m} = H\tilde{M} \text{ gdje je } H = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \quad (5)$$

Sada još samo treba procijeniti matricu homografije na temelju jednadžbe (5). Postoji više načina kako procijeniti takvu vezu, a ovdje se koristi procjena na temelju kriterija najveće izglednosti. Označimo s M_i točke modela, a s m_i točke slike. Pretpostavimo da je m_i zašumljena Gaussovim šumom sa srednjom vrijednošću nula i kovarijacijskom matricom Σ_{m_i} . Tada se procjenitelj najveće izglednosti H može dobiti minimizacijom izraza:

$$\sum_i (m_i - \hat{m}_i)^T \Sigma_{m_i}^{-1} (m_i - \hat{m}_i)$$

gdje je:

$$\hat{m}_i = \frac{1}{\bar{h}_3^T M_i} \begin{bmatrix} \bar{h}_1^T M_i \\ \bar{h}_2^T M_i \end{bmatrix}, \text{ gdje } \bar{h}_i \text{ predstavlja } i\text{-ti redak matrice } H$$

U praksi se pretpostavlja da je matrica kovarijacije dijagonalna i dijeljena za sve članove, tj oblika je $\Sigma_{m_i} = \sigma^2 I$. Zbog toga, navedeno postaje nelinearan problem najmanjih kvadrata, tj. formula se pretvara u $\min_H \sum_i \|m_i - \hat{m}_i\|^2$. Nelinearna optimizacija se provodi pomoću Levenberg-Marquardt algoritma. Za više detalja o implementaciji algoritma može se pogledati u dodatnoj literaturi [15]. U nastavku slijedi postupak za određivanje matrice homografije prema [14] i [16]. Definirajmo prvo oblik matrice H . Ona se u općenitom obliku može predstaviti kao 3x3 matrica.

$$H = \begin{bmatrix} P_0 & P_1 & P_2 \\ P_3 & P_4 & P_5 \\ P_6 & P_7 & P_8 \end{bmatrix} = \begin{bmatrix} - & - & h_1 & - & - \\ - & - & h_2 & - & - \\ - & - & h_3 & - & - \end{bmatrix} \text{ te neka je}$$

$$x = [\bar{h}_1^T, \bar{h}_2^T, \bar{h}_3^T]^T$$

Uvjet u jednadžbi (5) govori da su matrice $\tilde{m}$ i $H\tilde{M}$ jednake do na konstantu, ali imaju jednak smjer i sukladno tome možemo pisati:

$$\tilde{m} \times x\tilde{M} = 0 \quad (6)$$

Ako sada napišemo desni član ovoga vektorskog umnoška kao

$$x\tilde{M} = \begin{bmatrix} \bar{h}_1^T \tilde{M} \\ \bar{h}_2^T \tilde{M} \\ \bar{h}_3^T \tilde{M} \end{bmatrix}$$

Sada možemo potpuno raspisati jednadžbu (6):

$$\tilde{m} \times x\tilde{M} = \begin{bmatrix} \bar{h}_3^T \tilde{M} * v - \bar{h}_2^T \tilde{M} \\ -\bar{h}_3^T \tilde{M} * u + \bar{h}_1^T \tilde{M} \\ \bar{h}_2^T \tilde{M} * u - \bar{h}_1^T \tilde{M} * v \end{bmatrix}$$

Dodatno vrijedi $\bar{h}_j^T \tilde{M} = \tilde{M}^T \bar{h}_j$ pa umnožak možemo zapisati izražavanjem članova od x kao nepoznanica:

$$\begin{bmatrix} 0^T & -\tilde{M}^T & v * \tilde{M}^T \\ \tilde{M}^T & 0^T & -u * \tilde{M}^T \\ -v * \tilde{M}^T & u * \tilde{M}^T & 0^T \end{bmatrix} \begin{bmatrix} \bar{h}_1 \\ \bar{h}_2 \\ \bar{h}_3 \end{bmatrix} = 0 \quad (7)$$

Time smo dobili homogeni sustav jednadžbi gdje su prva dva retka linearno nezavisna. Izbacivanjem trećeg retka, zamjenom položaja prva dva retka te množenjem prvog retka u (7) s -1 dobivamo sustav:

$$\begin{bmatrix} \tilde{M}^T & 0^T & -v * \tilde{M}^T \\ 0^T & \tilde{M}^T & -u * \tilde{M}^T \end{bmatrix} \begin{bmatrix} \bar{h}_1 \\ \bar{h}_2 \\ \bar{h}_3 \end{bmatrix} = 0 \quad (8)$$

Kada imamo n točaka preko kojih utvrđujemo homografiju, imamo n jednadžbi koje su oblika kao i u (8). Tih n jednadžbi može se zapisati u matricu L čija će dimenzija biti $2n \times 9$. Kako je x definiran do na konstantu, rješenje se nalazi postupkom singularne dekompozicije, a kao rješenje se uzima desni singularni vektor matrice L koji je povezan s najmanjom singularnom vrijednošću (tj. uzima se svojstveni vektor od $L^T L$ s najmanjom svojstvenom vrijednošću). Nakon što je sustav riješen, dobivena rješenja potrebno je još minimizirati korištenjem Levenberg-Marquardt algoritma.

Jednom kada smo odredili matricu homografije, možemo se vratiti na jednakost koju smo uveli u jednadžbi (5). Ako zapišemo $H = [h_1 \ h_2 \ h_3]$, imamo sljedeću jednakost:

$$[h_1 \ h_2 \ h_3] = \lambda A [r_1 \ r_2 \ t] \quad (9)$$

gdje je λ proizvoljni skalar. Pošto znamo da su r_1 i r_2 ortonormalni, imamo sljedeća ograničenja:

$$h_1^T (A^{-1})^T A^{-1} h_2 = 0 \quad (10)$$

$$h_1^T (A^{-1})^T A^{-1} h_1 = h_2^T (A^{-1})^T A^{-1} h_2 \quad (11)$$

Ti izrazi predstavljaju osnovna ograničenja intrinzičnih parametara. Kako homografska matrica ima 8 stupnjeva slobode, a postoji 6 ekstrinzičnih parametara (3 za rotaciju i 3 za translaciju), možemo odrediti samo 2 ograničenja na intrinzične parametre. U nastavku će biti objašnjen postupak pronalaska intrinzičnih i ekstrinzičnih parametara kako bismo uspješno mogli poništiti sve efekte koje kamera unosi u sliku.

3.1.3. Određivanje parametara

Konstruirajmo matricu oblika:

$$B = \lambda (A^{-1})^T A^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}$$

$$= \lambda \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2 \beta} & \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} \\ -\frac{\gamma}{\alpha^2 \beta} & \frac{\gamma^2}{\alpha^2 \beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0 \gamma - u_0 \beta)^2}{\alpha^2 \beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix}$$

Treba primijetiti da je matrica B simetrična i da se može definirati vektorom:

$$b = [B_{11}, B_{12}, B_{13}, B_{22}, B_{23}, B_{33}]^T$$

Označimo s $h_i = [h_{i1}, h_{i2}, h_{i3}]^T$ i-ti stupac matrice H . Zatim definirajmo sljedeći odnos:

$$h_i^T B h_j = v_{ij}^T b$$

gdje je v_{ij}^T definiran kao

$$v_{ij}^T = [h_{i1}h_{j1}, \quad h_{i1}h_{j2} + h_{i2}h_{j1}, \quad h_{i2}h_{j2}, \quad h_{i3}h_{j1} + h_{i1}h_{j3}, \quad h_{i3}h_{j2} + h_{i2}h_{j3}, \quad h_{i3}h_{j3}]$$

Nadalje, dva osnovna ograničenja (10) i (11) za određenu homografiju mogu se još napisati i u obliku dvije homogene jednačbe po nepoznanici b :

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \quad (12)$$

Ako se promatra n slika, slaganjem jednačbi oblika (12) jedne na drugu dobivamo $Vb = 0$, gdje je V matrica dimenzija $2n \times 6$ (za svaki n imamo dvije jednačbe, a svaka jednačba ima 6 članova). Da bismo riješili tu jednačbu, opet provodimo postupak singularne dekompozicije, a rješenje je poznato kao svojstveni vektor sustava $V^T V$ koji je povezan s najmanjom svojstvenom vrijednošću (tj. desni singularni vektor od V koji je povezan s najmanjom singularnom vrijednošću).

Jednom kada procijenimo vektor b , možemo izračunati sve intrinzične parametre kamere. To možemo učiniti na sljedeći način:

$$\begin{aligned} v_0 &= \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2} \\ \lambda &= B_{33} - \frac{B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})}{B_{11}} \\ \alpha &= \sqrt{\frac{\lambda}{B_{11}}} \\ \beta &= \sqrt{\frac{\lambda B_{11}}{B_{11}B_{22} - B_{12}^2}} \\ \gamma &= -\frac{B_{12}\alpha^2\beta}{\lambda} \\ u_0 &= \frac{\gamma v_0}{\beta} - \frac{B_{13}\alpha^2}{\lambda} \end{aligned}$$

Jednom kada su definirani svi intrinzični parametri, vrlo lako se mogu proračunati ekstrinzični parametri iz jednadžbe (9):

$$\begin{aligned}\lambda_{r_1} &= \frac{1}{\|A^{-1}h_1\|} \\ \lambda_{r_2} &= \frac{1}{\|A^{-1}h_2\|} \\ \lambda_{r_3} &= \frac{\lambda_{r_1} + \lambda_{r_2}}{2} \\ r_1 &= \lambda_{r_1}A^{-1}h_1 \\ r_2 &= \lambda_{r_2}A^{-1}h_2 \\ r_3 &= r_1 \times r_2 \\ t &= \lambda_{r_3}A^{-1}h_3\end{aligned}$$

Zbog šuma u podacima, tako izračunata matrica $R = [r_1, r_2, r_3]$ rijetko kada će zadovoljavati svojstva rotacijske matrice. Kako se to može bolje procijeniti, pogledati u literaturi [14].

Opisani postupak dobiva se minimizacijom algebarske udaljenosti koja fizikalno nije značajna. To može usavršiti korištenjem funkcije najveće izglednosti. Pretpostavimo da imamo n slika modela ravnine i m točaka na ravnini modela. Dodatno, pretpostavimo da su točke slike sadrže šum koji je nezavisno i jednoliko distribuiran. Tada se procjenitelj najveće izglednosti može dobiti minimizacijom funkcije:

$$\sum_{i=1}^n \sum_{j=1}^m \|m_{ij} - \hat{m}(A, R_i, t_i, M_j)\|^2 \quad (13)$$

gdje $\hat{m}(A, R_i, t_i, M_j)$ predstavlja projekciju točke M_j u slici i , što je definirano jednadžbom (5). Minimizacija jednadžbe (13) predstavlja nelinearan optimizacijski problem koji se može riješiti Levenberg-Marquardt algoritmom.

Nakon što su proračunati i ekstrinzični parametri, napokon možemo poništiti iskrivljenja koja unose leće. U jednadžbama (1) i (2) dozvoljava se proizvoljan broj faktora koji modeliraju radijalnu i tangencijalnu distorziju, ali poznato je da se model distorzije najčešće dobro procjenjuje već s malo parametara. U nastavku će biti dana procjena za 3 koeficijenta radijalne distorzije i 2 parametra tangencijalne distorzije. Sukladno tome, jednadžbe (1) i (2) zapisujemo kao:

$$x_u - x_d = (x_d - u_0)(K_1 r^2 + K_2 r^4 + K_3 r^6) + (P_1(r^2 + 2(x_d - u_0)^2) + 2P_2(x_d - u_0)(y_d - v_0)) \quad (14)$$

$$y_u - y_d = (y_d - v_0)(K_1 r^2 + K_2 r^4 + K_3 r^6) + (2P_1(x_d - u_0)(y_d - v_0) + P_2(r^2 + 2(y_d - v_0)^2)) \quad (15)$$

Jednadžbe u (14) i (15) možemo promatrati kao sustav jednadžbi, što nam omogućava da ih zapišemo u matričnom obliku s tim da su nepoznanice faktori distorzija.

$$x' = [(x_d - u_0)r^2 \quad (x_d - u_0)r^4 \quad (x_d - u_0)r^6 \quad r^2 + 2(x_d - u_0)^2 \quad 2(x_d - u_0)(y_d - v_0)]$$

$$y' = [(y_d - v_0)r^2 \quad (y_d - v_0)r^4 \quad (y_d - v_0)r^6 \quad 2(x_d - u_0)(y_d - v_0) \quad r^2 + 2(y_d - v_0)^2]$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} x_u - x_d \\ y_u - y_d \end{bmatrix} \quad (16)$$

Ako imamo m točaka u n slika, možemo slagati jednadžbe jedne na drugu kako bismo ukupno imali $2mn$ jednadžbi. Tada se jednadžba (16) može matrično izraziti kao $Dk = d$. Rješenje te jednadžbe može se izračunati kao

$$k = (D^T D)^{-1} D^T d \quad (17)$$

Rješenje jednadžbe (17) predstavlja dobre početne uvjete za daljnju optimizaciju. Naime, ta procjena se može i doraditi tako da se rezultati uvrste u proširenje funkcije (13):

$$\sum_{i=1}^n \sum_{j=1}^m \|m_{ij} - \hat{m}(A, k_1, k_2, R_i, t_i, M_j)\|^2 \quad (18)$$

To je i dalje problem nelinearne optimizacije, koji se može riješiti korištenjem Levenberg-Marquardt algoritma.

Kada su svi koeficijenti procijenjeni, poništavanje utjecaja distorzije obavlja se preko relacija:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} \quad (19)$$

Prema [14], preporučena rutina za kalibraciju kamere sastoji se od sljedećih koraka:

- ispisati sliku binariziranoga šahovskog polja i pričvrstiti je na ravnu površinu
- uzeti nekoliko slika (barem dvije) ravnine modela pod različitim orijentacijama (ili micanjem kamere ili micanjem uzorka)
- detektiraj pozicije značajki u slikama
- procijeni pet intrinzičnih parametara i sve ekstrinzične parametre postupkom opisanim u 3.1.3.
- procijeni koeficijente radijalne i tangencijalne distorzije pomoću formule (17)
- doradi procjenu svih parametara minimizacijom izraza (18)
- konačno, poništi distorziju izrazima definiranim u (19)

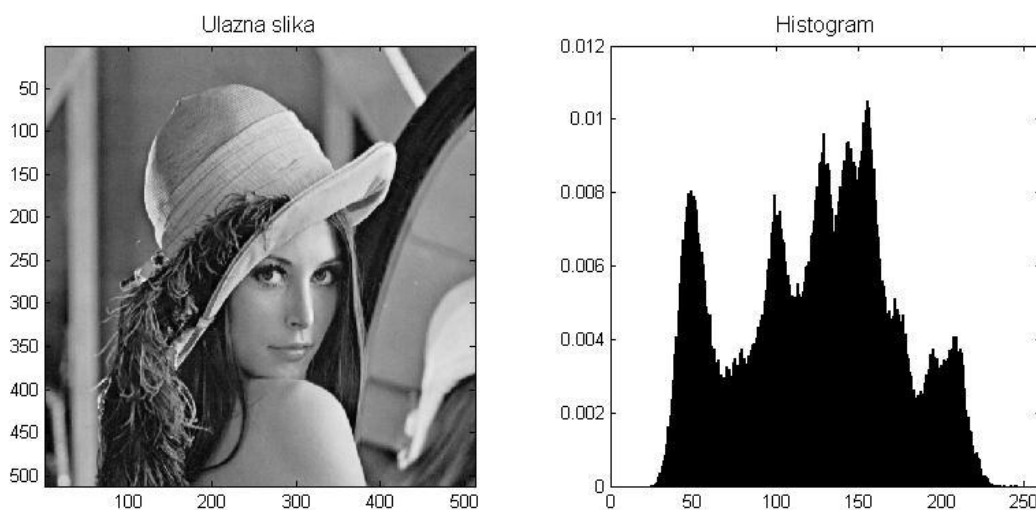
Kako bi se proces poništenja distorzije automatizirao, potrebno je implementirati detekciju i ekstrakciju značajki, gdje značajku predstavlja vrh kvadrata. Fokus ovog rada nije bio na tome, ali generalna ideja postupka je razdvajanje crnih regija šahovske ploče od bijelih (korištenjem morfoloških operacija erozije i dilatacije). Nakon toga se iskorištava činjenica da je svaki vektor kojemu je središte vrh kvadrata, recimo q , ortogonalan na gradijent slike u točki p (koja se nalazi u susjedstvu od q) u ovisnosti o zašumljenosti slike i podataka. Detaljnije se može pronaći u literaturi [17].

3.2. Operacije u prostoru boja

Mapiranje boja je funkcija koja transformira boje jedne slike prema boji druge slike. Na mapiranje boja može se gledati kao na algoritam koji za rezultat daje funkciju mapiranja. U slučaju da se radi mapiranje boja slika u boji, govorimo o *prijenosnoj funkciji boje* (engl. *color transfer function*), a ako radimo sa slikama u sivim tonovima, tada govorimo o *prijenosnoj funkciji svjetline* (engl. *brightness transfer function*). Postoje dva tipa algoritama za mapiranje boja, jedni koji koriste statistiku boja u dvije slike te oni koji se oslanjaju na odnose među slikovnim elementima dviju slika. Za potrebe ovog rada, koristili su se algoritmi izjednačavanja i modeliranja histograma, koji pripadaju u prvu skupinu algoritama za mapiranje boja. U nastavku će biti objašnjen pojam histograma te postupci za izjednačavanje i modeliranje histograma (Gonzalez, 2002). Sve operacije se odvijaju na slikama sivih tonova.

3.2.1. Histogram

Histogram slike koja ima sive tonove u rasponu $[0, L - 1]$ diskretna je funkcija $h(r_k) = n_k$, gdje je r_k k-ta razina sive boje, a n_k je broj slikovnih elemenata koji u slici imaju razinu r_k . Česta je praksa normalizirati histogram, i to dijeljenjem svake vrijednosti n_k s ukupnim brojem slikovnih elemenata u slici, n . Tada je normalizirani histogram definiram izrazom $p(r_k) = n_k/n$, $\forall k, k \in [0, L - 1]$. Slobodno govoreći, normalizirani histogram nam daje procjenu vjerojatnosti pojavljivanja pojedine razine sive boje. Na slici 5 prikazan je jedan takav histogram.



Slika 5. Slika sivih tonova i pripadni normalizirani histogram

Histogrami predstavljaju osnovu za brojne tehnike obrade slike u prostornoj domeni. Operacije nad histogramima mogu se efikasno iskoristiti za poboljšanje slike, a u nastavku će biti поближе opisana dva postupka koja su se koristila u ovome radu.

3.2.2. Izjednačavanje histograma

Zamislmo na trenutak kontinuiranu funkciju te označimo varijablom r razinu sive boje u slici koju želimo poboljšati. Pretpostavljamo da je r normalizirana na raspon $[0, 1]$, gdje nula predstavlja crnu, a jedan bijelu boju. Da bi bilo koji r zadovoljio ranije spomenute uvjete, uzimat ćemo transformacije koje su oblika

$$s = T(r), \quad 0 \leq r \leq 1$$

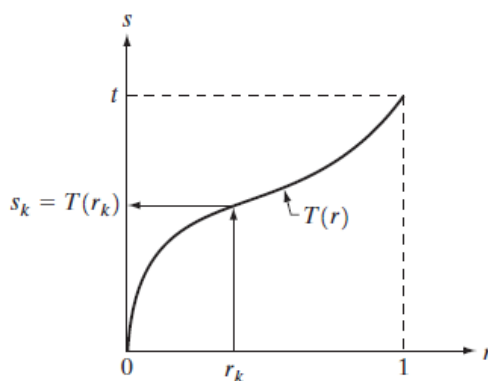
i koje kao rezultat daju razinu s za svaku vrijednost slikovnog elementa r u polaznoj slici.

Funkcija $T(r)$ treba imati sljedeća svojstva:

- a) $T(r)$ treba biti jednoznačna i monotono rastuća funkcija na intervalu $[0, 1]$
- b) $T(r) \in [0, 1]$, za $r \in [0, 1]$.

U prvom uvjetu jednoznačnost mora biti zajamčena kako bi postojao inverz funkcije, dok monotonost osigurava da se i u izlaznoj slici vrijednosti povećavaju od crnog prema bijelom. Drugi uvjet osigurava da će raspon izlazne slike također biti kao i raspon ulazne slike. Na slici 6 prikazana je transformacijska funkcija koja je jednoznačno određena i monotono rastuća. Inverzna transformacija koja omogućava povratak iz s u r označena je formulom:

$$r = T^{-1}(s) \quad 0 \leq s \leq 1$$



Slika 6. Valjana transformacijska funkcija

Razine sive boje u slici mogu se promatrati kao slučajne varijable na intervalu $[0, 1]$. Jedan od najosnovnijih načina za opis slučajne varijable je njezina funkcija gustoće vjerojatnosti (engl. *probability density function* – *PDF*). Neka $p_r(r)$ i $p_s(s)$ označavaju funkcije gustoće vjerojatnosti slučajnih varijabli r i s . Jedno od osnovnih načela osnovne probabilističke teorije nam govori da ako su $p_r(r)$ i $T(r)$ poznati i ako $T^{-1}(s)$ zadovoljava uvjet pod a), tada se funkcija gustoće vjerojatnosti $p_s(s)$ transformirane varijable s može izračunati korištenjem jednostavne formule:

$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right| \quad (20)$$

Prema tome, funkcija gustoće vjerojatnosti transformirane varijable s određena je PDF funkcijom ulazne slike i izabranom transformacijskom funkcijom. Transformacijska funkcija od posebnog značenja u području obrade slike ima oblik

$$s = T(r) = \int_0^r p_r(w)dw \quad (21)$$

gdje je w varijabla po kojoj integriramo izraz. Desna strana jednadžbe (21) predstavlja kumulativnu funkciju distribucije (engl. *cumulative distribution function* – *CDF*) slučajne varijable r . Tako definirana funkcija zadovoljava svojstva a) i b) što je čini pogodnom za daljnje procesiranje.

Za danu transformacijsku funkciju $T(r)$, $p_s(s)$ pronalazimo primjenjujući formulu (20). Iz osnova matematike poznato nam je sljedeće svojstvo (Leibnizovo pravilo):

$$\begin{aligned} \frac{dr}{ds} &= \frac{dT(r)}{ds} \\ &= \frac{d}{ds} \left[\int_0^r p_r(w)dw \right] \\ &= p_r(r) \end{aligned} \quad (22)$$

Zamjenom dobivenog rezultata u jednadžbi (20) i poznavanjem svojstva da su sve vrijednosti vjerojatnosti pozitivne proizlazi svojstvo:

$$\begin{aligned} p_s(s) &= p_r(r) \left| \frac{dr}{ds} \right| \\ &= p_r(r) \left| \frac{1}{p_r(r)} \right| \\ &= 1 \quad 0 \leq s \leq 1 \end{aligned} \quad (23)$$

Zato što je $p_s(s)$ funkcija gustoće vjerojatnosti, mora vrijediti da je jednaka nuli izvan intervala $[0, 1]$ u ovom slučaju zbog toga što integral preko svih vrijednosti od s mora biti jednak 1. Oblik $p_s(s)$ dan u jednadžbi (23) predstavlja poznatu PDF funkciju koja se naziva uniformna funkcija gustoće vjerojatnosti. Jednostavno govoreći, pokazali smo da obavljanjem transformacijske funkcije u jednadžbi (21) proizvodi slučajnu varijablu s koja je karakterizirana uniformnom funkcijom gustoće vjerojatnosti. Važno je napomenuti da prema jednadžbi (21) transformacijska funkcija $T(r)$ ovisi o $p_r(r)$, ali prema jednadžbi (23), resultantni $p_s(s)$ je uvijek uniforman, neovisno o obliku funkcije $p_r(r)$.

U slučaju diskretnih vrijednosti baratamo s vjerojatnostima i sumama umjesto s funkcijom gustoće vjerojatnosti i integralima. Vjerojatnost pojavljivanja razine sive boje r_k u slici aproksimirana je pomoću izraza

$$p_r(r_k) = \frac{n_k}{n} \quad k \in [0, L - 1]$$

Diskretna formulacija jednadžbe (21) glasi:

$$\begin{aligned} s_k = T(r_k) &= \sum_{j=0}^k p_r(r_j) \\ &= \sum_{j=0}^k \frac{n_j}{n} \quad k \in [0, L - 1] \end{aligned} \tag{24}$$

Prema tome, procesirana slika dobiva se mapiranjem svakoga slikovnog elementa s razinom r_k ulazne slike u odgovarajući slikovni element s_k izlazne slike pomoću jednadžbe (24). Mapiranje dano u jednadžbi (24) naziva se izjednačavanje histograma. Za razliku od kontinuirane verzije, ne može se općenito dokazati da će diskretna transformacija rezultirati diskretnim ekvivalentom uniformne funkcije gustoće vjerojatnosti, što bi predstavljalo uniformni histogram. Svejedno, korištenje formule (24) ima generalnu tendenciju raspršenja histograma ulazne slike tako da se razine histogramski izjednačene slike rasprostiru po većem rasponu sive skale. Diskretna inverzna transformacija koja omogućava povratak iz s u r označena je kao

$$r_k = T^{-1}(s_k) \quad k \in [0, L - 1] \tag{25}$$

Može se pokazati da navedena jednadžba zadovoljava uvjete a) i b) samo ako su u ulaznoj slici zastupljene sve razine $r_k, k \in [0, L - 1]$. Iako se inverzna

transformacija ne koristi u procesu izjednačavanja histograma, bitna je za proces modeliranja histograma koji će biti objašnjen u nastavku.

3.2.3. Modeliranje histograma

Ponekad nam poboljšanje koje donosi proces izjednačavanja histograma ne daje dovoljno dobar rezultat. Tada bismo htjeli definirati oblik histograma koji želimo da naša slika ima. Metoda koja se u tom slučaju koristi naziva se modeliranje histograma.

Vratimo se natrag na kontinuirane sive razine r i z te neka r označava razine sivih tonova ulazne slike, a z razine sivih tonova izlazne slike. Neka $p_r(r)$ i $p_z(z)$ označavaju PDF funkcije slučajnih varijabli r i z . Kao što je već spomenuto, $p_r(r)$ možemo procijeniti iz slike, dok $p_z(z)$ predstavlja željenu funkciju gustoće vjerojatnosti koju želimo da naša izlazna slika ima. Neka je s slučajna varijabla s jednakim svojstvima definiranim u jednadžbi (21).

Definirajmo sada slučajnu varijablu z sa sljedećim svojstvom:

$$G(z) = \int_0^z p_z(t) dt = s \quad (26)$$

gdje t predstavlja parametar po kojemu se integrira. Iz jednadžbi (21) i (26) proizlazi jednakost $G(z) = T(r)$, zbog čega z mora zadovoljavati uvjet:

$$z = G^{-1}(s) = G^{-1}[T(r)] \quad (27)$$

Transformacijska funkcija $T(r)$ može se izračunati prema jednadžbi (21) jednom kada se procijeni $p_r(r)$ iz ulazne slike. Slično, transformacijska funkcija $G(z)$ može se proračunati korištenjem jednadžbe (26) zato što je $p_z(z)$ zadana.

Uz pretpostavku da G^{-1} postoji i da zadovoljava svojstva a) i b), jednadžbe (21), (26) i (27) pokazuju da se može dobiti slika sa željenom funkcijom gustoće vjerojatnosti na temelju ulazne slike ako se slijede koraci:

1. izračunaj transformacijsku funkciju $T(r)$ izrazom u (21)
2. izračunaj transformacijsku funkciju $G(z)$ prema (26)
3. izračunaj inverznu transformacijsku funkciju G^{-1}
4. izračunaj izlaznu sliku primjenom (27) na svaki slikovni element ulazne slike

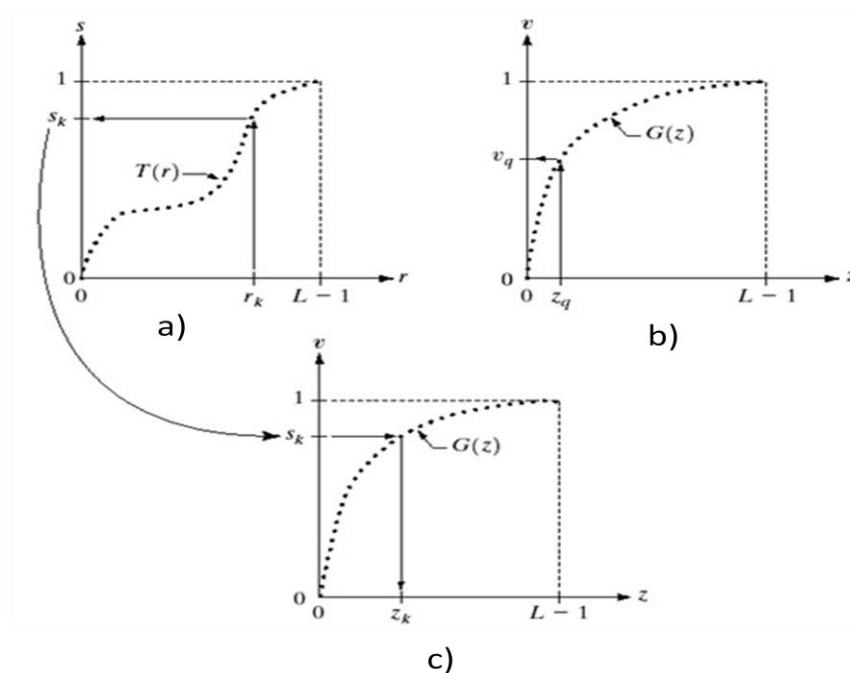
Rezultat tih koraka bit će nova slika čije će se razine sive boje, z , pokoravati željenoj funkciji gustoće razdiobe $p_z(z)$. Iako je opisan postupak veoma jasan u teoriji, u praksi je malokad moguće dobiti analitički izraz za $T(r)$ i za G^{-1} . Srećom, taj problem je znatno pojednostavljen u slučaju diskretnih vrijednosti. Cijena koju u tom slučaju plaćamo je u tome što možemo dobiti samo aproksimaciju željenog histograma. Unatoč tom ograničenju, mogu se dobiti vrlo korisni rezultati i uz ovako grube aproksimacije. Diskretna varijanta formule (26) može se dobiti iz željenog histograma $p_z(z_i)$, $i \in [0, L - 1]$ i oblika je

$$v_k = G(z_k) = \sum_{i=0}^k p_z(z_i) = s_k, \quad k \in [0, L - 1] \quad (28)$$

Kao i s kontinuiranim vrijednostima, ovdje isto tražimo vrijednosti od z koje zadovoljavaju ovu jednadžbu. Još nam preostaje definirati i diskretnu verziju formule (27) koja glasi

$$\begin{aligned} z_k &= G^{-1}[T(r_k)], \quad k \in [0, L - 1] \\ z_k &= G^{-1}(s_k), \quad k \in [0, L - 1] \end{aligned} \quad (29)$$

Jednadžbe (24), (28) i (29) predstavljaju temelj za implementaciju modeliranja histograma. Jednadžba (24) definira mapiranje razina iz originalne slike u odgovarajuće razine s_k na temelju histograma originalne slike, koji proračunavamo na temelju slikovnih elemenata ulazne slike, a to je prikazano na slici 7.a). Jednadžba (28) računa transformacijsku funkciju G iz željenog histograma $p_z(z)$, što je prikazano na slici 7.b). Konačno, jednadžba (29) nam daje aproksimaciju željenih razina sive boje na slici s tim histogramom, prikazano na slici 7.c). Prve dvije jednadžbe je vrlo lako implementirati jer su nam sve vrijednosti poznate, dok je implementacija jednadžbe (29) jasna, ali zahtijeva dodatno objašnjenje koje će biti dano u nastavku.



Slika 7. Grafički prikaz modeliranja histograma (Gonzalez, 2002)

Uvedimo sljedeće oznake:

- svako je polje razina sivih boja $\{r_j\}$, $\{s_j\}$ i $\{z_j\}$, $j \in [0, L - 1]$ 1D polje dimenzija $L \times 1$
- sva mapiranja iz r u s i iz s u z jednostavni su dohvati iz tablice pretraživanja
- svaki element polja, npr. s_k , sadrži dvije bitne informacije – indeks k označava lokaciju elementa, a s označava vrijednost na tom mjestu
- treba razmatrati samo cjelobrojne vrijednosti (to znači da radimo s razinama sive boje na intervalu $[0, L - 1]$ umjesto normaliziranog intervala $[0, 1]$ koji smo koristili kako bismo pojednostavili razvoj histogramskih operacija).

Pogledajmo ponovno sliku 7, na trenutak ignorirajući vezu između grafa a) i c). Graf pod a) prikazuje hipotetsku diskretnu transformacijsku funkciju $s = T(r)$ koju dobivamo na temelju ulazne slike. Na tom se grafu prva razina sive boje, r_1 , mapira u s_1 , r_2 u s_2 , r_k u s_k i tako dalje (bitno je primijetiti uređenu korespondenciju između tih vrijednosti). Svaka vrijednost s_j u polju preračunava se korištenjem formule (24), tako da proces mapiranja jednostavno koristi trenutnu vrijednost slikovnog elementa kao indeks u polju u kojemu se mora odrediti korespondentna vrijednost od s . Graf pod b) predstavlja hipotetsku transformacijsku funkciju G koju proračunavamo na željenom histogramu $p_z(z)$ pomoću jednadžbe (28). Za bilo

koju vrijednost z_q , ta transformacija proizvodi odgovarajuću vrijednost v_q . Obratno, za bilo koju vrijednost v_q našli bismo korespondentnu vrijednost z_q pomoću G^{-1} . Iz jednadžbe (28) znamo da vrijedi $v = s$ za odgovarajuće indekse, a to znači da možemo koristiti točno taj izraz kako bismo pronašli z_k koji odgovara bilo kojoj vrijednosti s_k koje smo maloprije izračunali iz jednadžbe $s = T(r)$. To je prikazano na grafu c).

Kako zapravo nemamo z -ove, moramo se osloniti na neku vrstu iterativnog algoritma kako bismo pronašli z iz s . U osnovi, zato što vrijedi $v_k = s_k$, na temelju jednadžbe (28) mora vrijediti da z -ovi za koje tražimo rješenje moraju zadovoljiti jednadžbu $(G(z_k) - s_k) = 0$. Kako bismo pronašli z_k kojemu je par odgovarajući s_k , moramo iterirati po vrijednostima od z tako da je ova jednadžba zadovoljena za $k \in [0, L - 1]$. To je isti uvjet kao i u jednadžbi (29), osim što ne moramo tražiti inverz od G jer ćemo iterirati po z . Kako radimo s cjelobrojnim vrijednostima, najbliže što se možemo približiti zadovoljenju jednadžbe $(G(z_k) - s_k) = 0$ je da pustimo $z_k = \hat{z}$ za sve vrijednosti od k , gdje je $\hat{z}$ najmanji cijeli broj na intervalu $[0, L - 1]$ takav da vrijedi sljedeće:

$$(G(\hat{z}) - s_k) \geq 0 \quad (30)$$

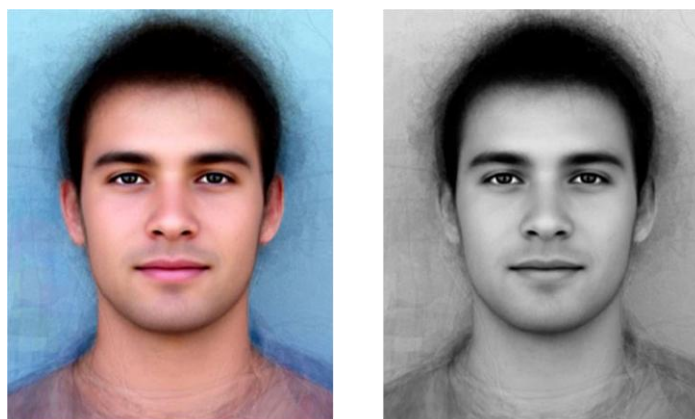
To znači da bismo za neku vrijednost s_k krenuli s vrijednošću $\hat{z} = 0$ i povećavali korak sve dok nije zadovoljena jednadžba (30), kada bismo rekli da je $z_k = \hat{z}$. Ponavljanjem tog postupka za sve vrijednosti od k rezultiralo bi svim potrebnim mapiranjima iz s u z .

4. Implementacija

Programsko rješenje implementirano je unutar razvojnog okruženja Visual Studio 2012 u programskom jeziku C++ uz korištenje biblioteke OpenCV 2.4.11.0 na operacijskom sustavu Windows 8.1. U prilogu A bit će opisan postupak snimanja korištene biblioteke. U nastavku će biti opisani i model pozadine koji se koristi u radu, konkretan način izvedbe procesa poništenja geometrijskih iskrivljenja te implementacija procesa modeliranja histograma.

Kako je već navedeno, u ovom radu korišten je model Google Cardboard uređaja na čiji je prednji dio montirana USB endoskopska kamera. Kamera ima kut prikaza od 67° , promjera je 7 mm, a korištena rezolucija snimanja je 640x480 slikovnih elemenata. Položaj kamere je fiksiran i statičan tako da uvijek snima donji dio lica kada se Cardboard stavi na glavu.

Kao pozadinska slika koristi se uprosječena slika muškog lica koja je nastala na uzorku od sto fotografija ljudi iz Sydney-a (Moses, 2011). Na slici 8.a) prikazano je kako ona izgleda u boji, dok je na slici 8.b) prikazana crno-bijela varijanta jer se sve operacije u ovom radu obavljaju na takvim slikama.

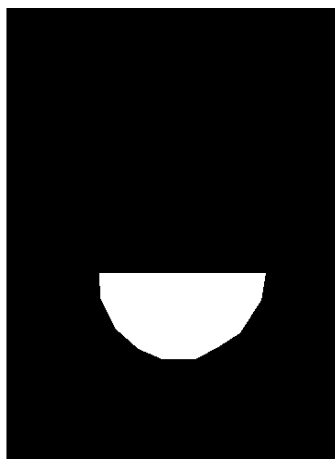


a)

b)

Slika 8. Slika prosječnog lica (Moses, 2011)

Kako je pozadina statična, unaprijed se odredilo područje interesa preko kojega će se dodati snimka s kamere. Tako definirana regija ujedno je i poslužila kao maska u daljnjoj implementaciji, što je prikazano na slici 9.



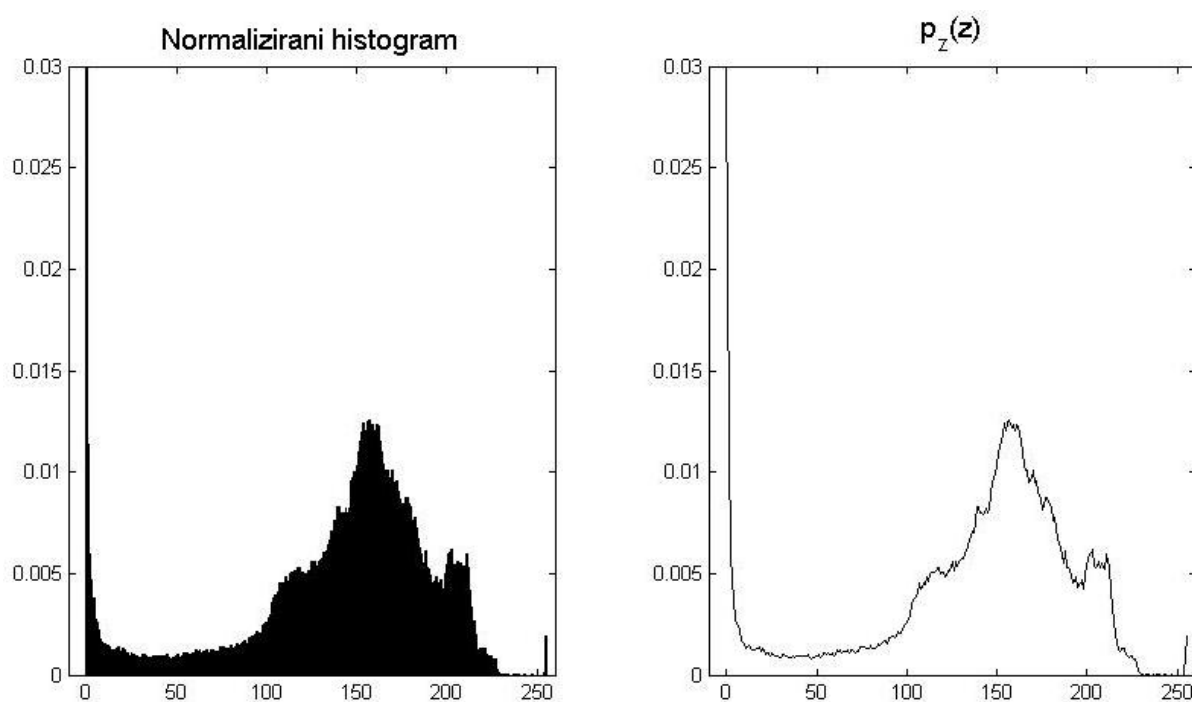
Slika 9. Područje interesa na pozadini

Prije nego što se počnu obrađivati slike kamere, definirana je još jedna statička regija interesa. Naime, zbog fiksiranog položaja kamere, slika koju kamera snima uvijek je ista – donji dio lica. Jedino što varira je postotak koje to lica zauzima u kadru. Za potrebe ovog rada uvedena je jedna pretpostavka koja nije matematički provjeravana. Pretpostavka se odnosi na područje interesa na snimci kamere, a kaže da će kamera uvijek biti dobro pozicionirana tako da snima područje oko usta i brade. Zbog tako definirane pretpostavke smijemo statički definirati regiju koju ćemo kasnije obrađivati. Ako tu regiju interesa obilježimo pravokutnikom koji ćemo definirati dvjema točkama koje su točno nasuprot jedna drugoj, tada je za gornji lijevi kut uzeta koordinata $(0,89)$, a za donji desni $(639,479)$. Rezultati takvog izbora bit će prikazani kasnije, u nastavku, a sada će biti objašnjen proces kalibracije kamere.

Prvo se proračunavaju parametri za poništenje distorzije i svih nepravilnosti koje kamera nosi sa sobom. Kalibracija se obavlja tako da korisnik pokazuje uzorak šahovske ploče koja je zalijepljena na ravnu plohu. U ovom radu korištena je šahovska ploča dimenzija 8×8 . Sustav ne pokreće proračun parametara sve dok nije skupio 10 slika šahovnice. Nakon što je skupljeno 10 slika, odvija se proces koji je detaljno opisan u poglavlju 3.1, a može se slijediti naputak koji je dan u 3.1.3. – određuje se homografska matrica, a na temelju matrice se proračunavaju intrinzični i ekstrinzični parametri kamere. Nakon kalibracije slijede histogramске operacije. Za potrebe ovog rada prvo je bio implementiran postupak izjednačavanja histograma, ali on nije davao zadovoljavajuće rezultate, što će biti pokazano u poglavlju o rezultatima. Umjesto toga, implementiran je proces

modeliranja histograma prema proceduri koja je detaljno opisana u poglavlju 3.2.3., a može se sažeti u sljedeće korake:

1. izračunaj histogram ulazne slike
2. po formuli (24) preračunaj mapirane razine s_k za svaku ulaznu razinu r_k
3. izračunaj transformacijsku funkciju G iz željene $p_z(z)$ korištenjem (28)
4. preračunaj z_k za svaku vrijednost od s_k korištenjem iterativne procedure definirane u (30)
5. za svaki slikovni element originalne slike, ako je njegova vrijednost r_k , mapiraj tu vrijednost u odgovarajuću razinu s_k . Zatim mapiraj s_k u z_k .



Slika 10. Histogram pozadine

Ovdje je potrebno napomenuti da željenu funkciju $p_z(z)$ predstavlja normalizirani histogram pozadinske slike koji je prikazan na slici 10, a da funkciju koju želimo modelirati ($p_r(r)$) predstavlja normalizirani histogram ulazne slike kamere.

Završetkom procesa modeliranja histograma, preostaje nam izvršiti poklapanje slike kamere s pozadinom korištenjem definirane maske. Prije poklapanja, potrebno je uzeti dio slike koji smo statički definirali te ga smanjiti na veličinu jednaku minimalnoj pravokutnoj regiji koja obuhvaća cijelu masku (engl. *bounding rectangle*). Zatim je potrebno obaviti i zaglađivanje maske. Zaglađivanje je potrebno napraviti kako bi se ublažio prijelaz između slike kamere i pozadine. Za

zaglađivanje se koristi Gaussov filter čija je jezgra veličine 11x11 sa standardnom devijacijom u oba smjera od 21,0. Te vrijednosti su proračunate empirijski i pokazale su se optimalnima. Parametri za poništenje distorzije ne moraju se svaki put iznova računati već se mogu spremati u lokalnu varijablu radi efikasnosti programa. Na temelju navedenih stvari, možemo generirati pseudokod koji bi izgledao

Definiraj područje interesa na uprosječenom licu

Definiraj područje interesa na slici kamere

Izračunaj intrinzične i ekstrinzične parametre kamere

Zagladi područje interesa na uprosječenom licu

While (true){

frame = učitaj jednu sliku s kamere

Poništi iskrivljenje i rezultat vrati u frame

Modeliraj histogram kamere prema histogramu pozadine

Prilagodi veličinu frame-a veličini uokvirujućeg pravokutnika maske

Spoji područje interesa kamere s područjem interesa uprosječenog lica

Prikaži rezultat spajanja korisniku

Ako je (pritisnuta_tipka = ESC) {

break;

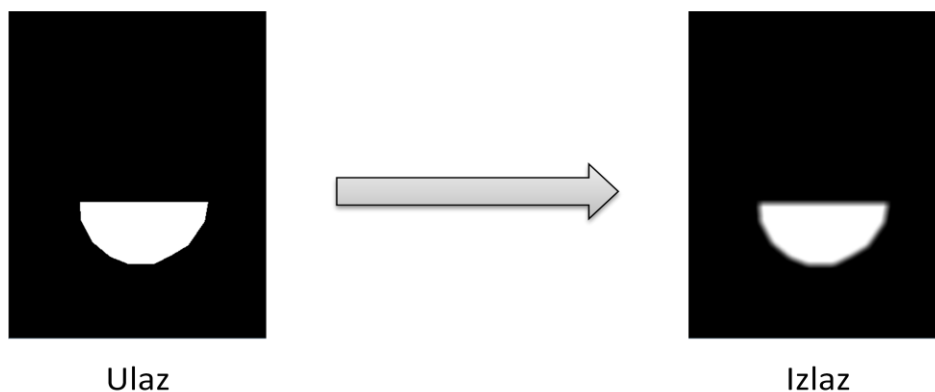
}

}

5. Rezultati

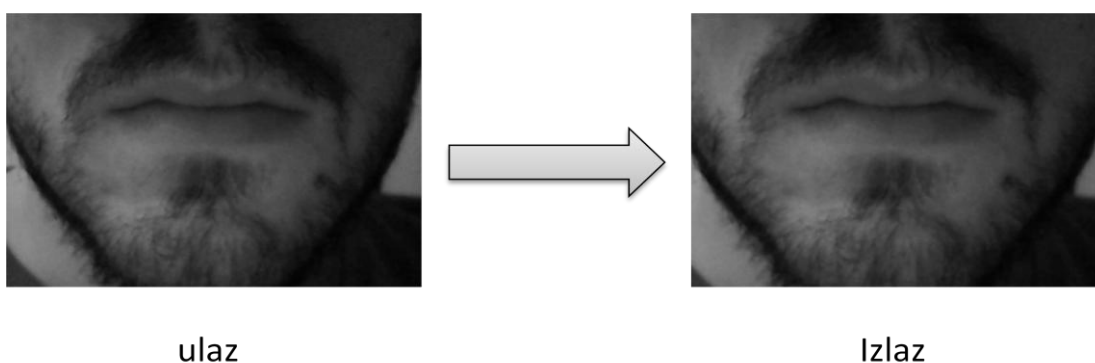
U nastavku će biti prikazani rezultati dobiveni izvođenjem prethodno navedenih koraka. Za svaki od važnijih koraka bit će prikazana ulazna slika te rezultat te operacije.

Prvo što radimo je zaglađivanje maske, a rezultat je prikazan na slici 11.



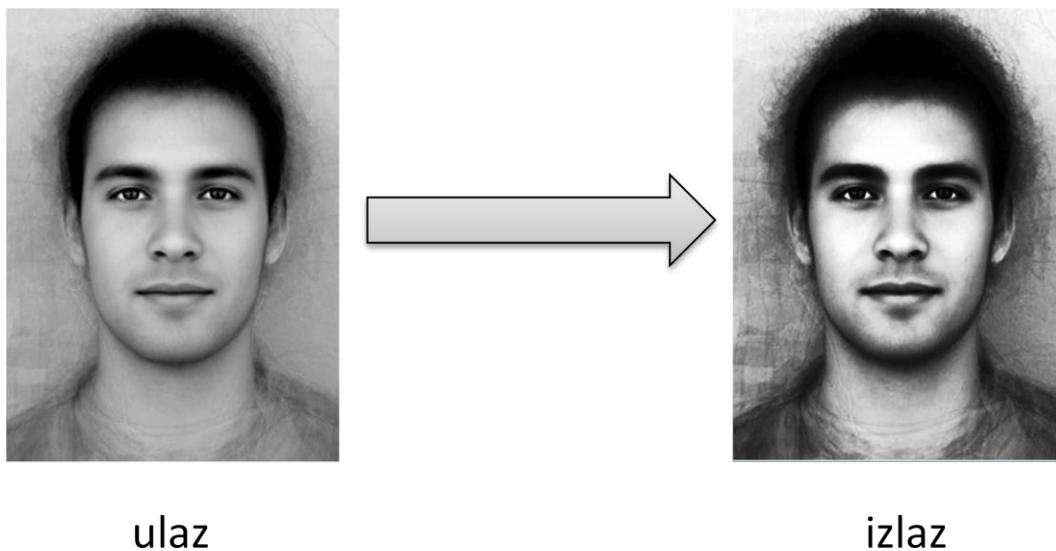
Slika 11. Zaglađivanje maske

Nakon zaglađivanja maske slijedi ispravljanje svakoga novog okvira koji dohvatimo s kamere. Kako je korištena kamera unosila neznatna iskrivljenja, rezultat poništenja je gotovo istovjetan ulaznoj slici (prosječna pogreška ponovne projekcije iznosi 1.812) što je i prikazano na slici 12.

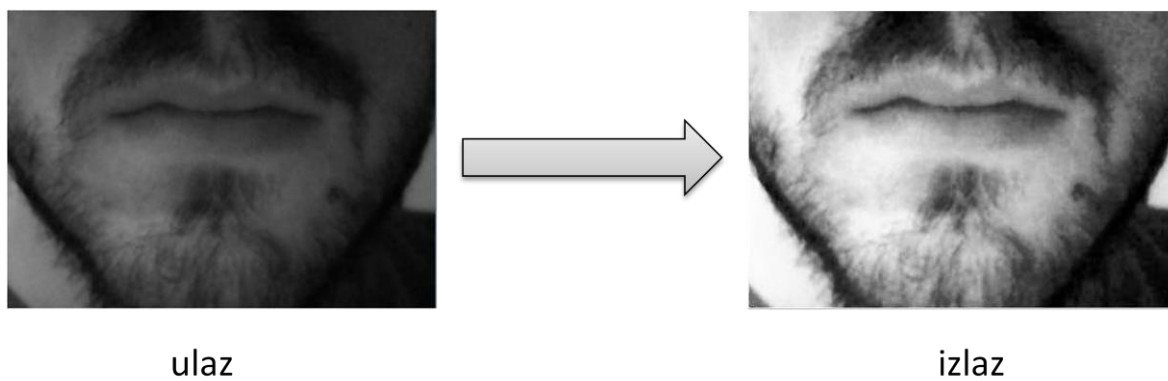


Slika 12. Ispravljanje distorzije

Prvotno je nakon poništavanja distorzije slijedio postupak izjednačavanja histograma, međutim on nije davao zadovoljavajuće rezultate. Problem je predstavljala pozadina, gdje su obrasi modela nakon izjednačavanja histograma postali presvijetli u odnosu na rezultat izjednačavanja histograma kamere. Na slici 13 prikazan je rezultat izjednačavanja histograma modela pozadine, a na slici 14 prikazan je rezultat izjednačavanja histograma kamere.

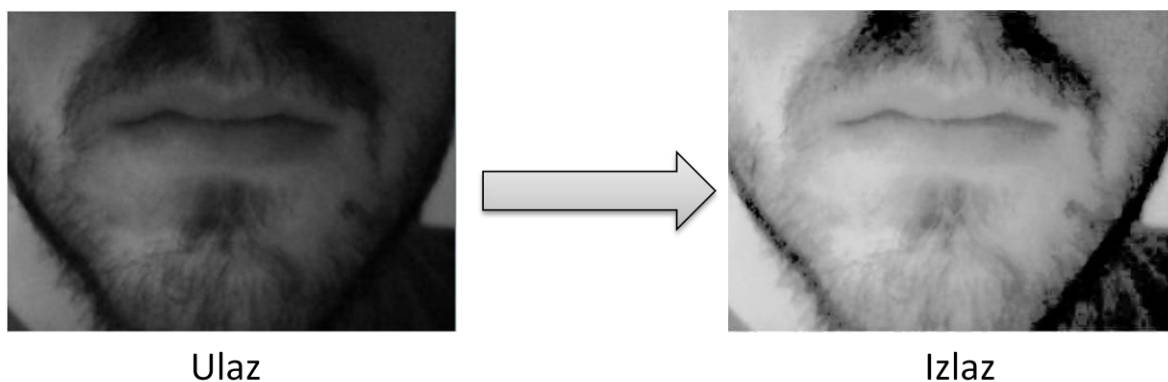


Slika 13. Izjednačavanje histograma pozadine



Slika 14. Izjednačavanje histograma kamere

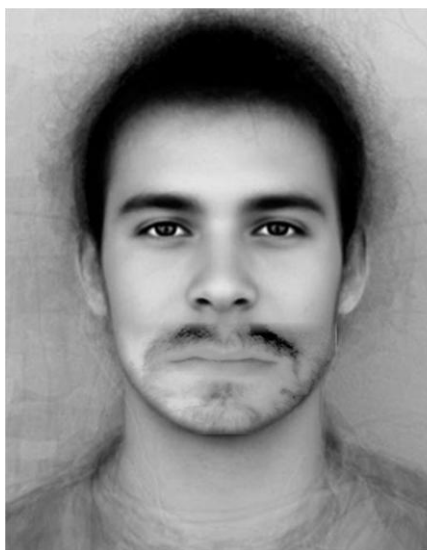
Umjesto izjednačavanja histograma, implementiran je postupak modeliranja histograma, a rezultat te operacije prikazan je na slici 15.



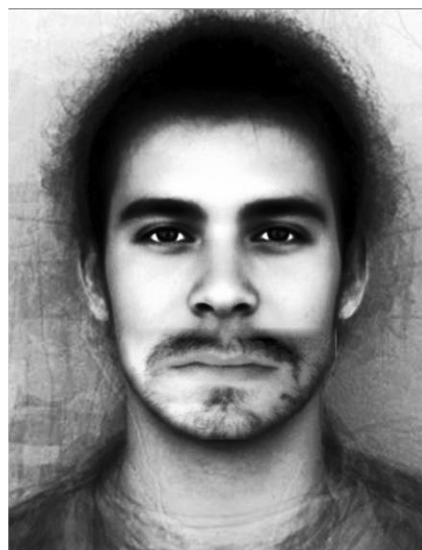
Slika 15. Modeliranje histograma

Iako tako na prvi pogled izgleda da se bolji rezultati dobivaju korištenjem izjednačavanja histograma jer je slika kamere za slučaj modeliranja histograma niskoga kontrasta, ipak, treba primijetiti nekoliko stvari. Uočimo da su obrazi na modelu pozadine drastično svjetliji nego gornji lijevi i desni kutovi na slici kamere. Dodatno, pri izjednačavanju histograma na slici kamere se događa da je jedna strana lica malo svjetlija od druge. Ti problemi pridonose neprirodnosti konačne slike, a to će biti prikazano u nastavku.

Naposljetku, preostaje još samo spojiti rezultat dobiven izjednačavanjem histograma te rezultat dobiven modeliranjem histograma s modelom pozadine. Na slici 16. a) prikazan je rezultat dobiven postupkom modeliranja histograma, dok je na slici 16. b) prikazan rezultat dobiven procedurom izjednačavanja histograma. Iz tih slika se najbolje vide problemi navedeni ranije u tekstu, zbog kojih je očitije da je donji dio lica dodan preko modela pozadine, dok je to puno teže uočiti na slici pod a).



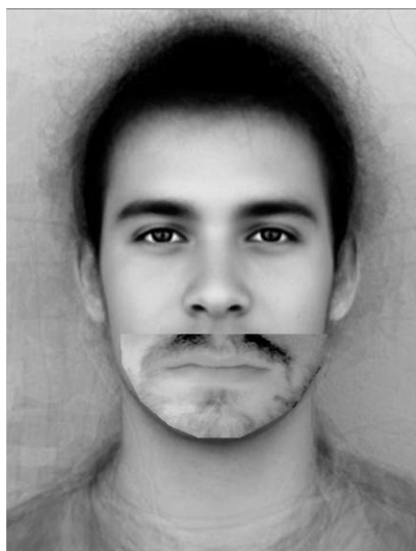
a)



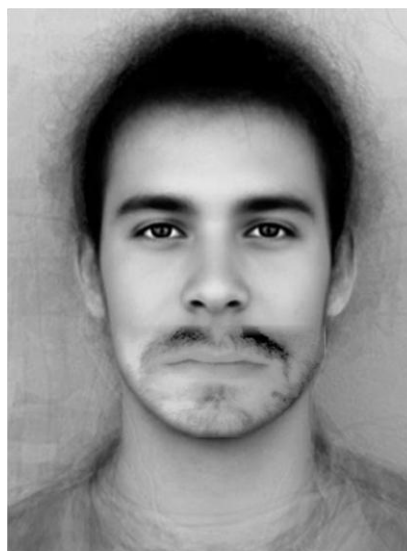
b)

Slika 16. Rezultat spajanja (1)

Za kraj ostaje još samo pokazati značajnost zaglađivanja maske. Na slici 17. a) prikazan je rezultat spajanja bez zaglađivanja maske, dok je na slici pod b) maska zaglađena. Iz dobivenih rezultata se vidi da slika pod b) izgleda prirodnije u odnosu na sliku pod a) te da je dosta teško i uočiti regije dodavanja (jedino je dovoljno očita horizontalna linija preko cijelog lica)



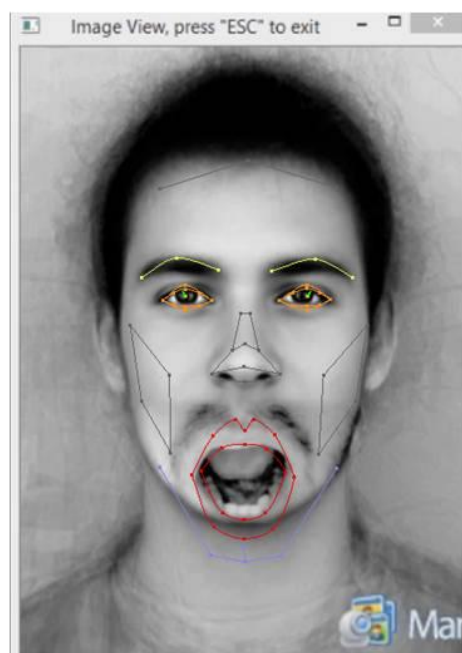
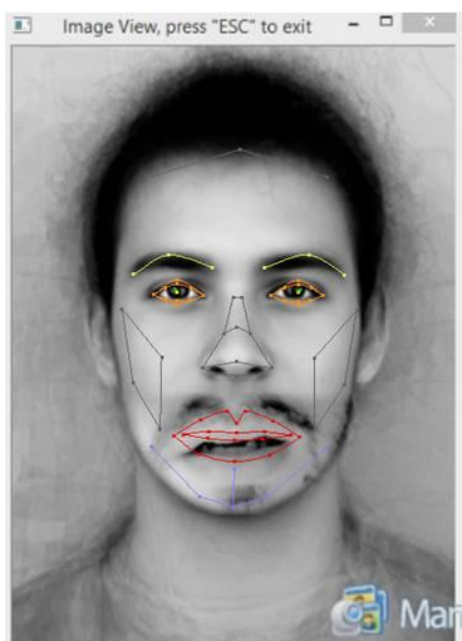
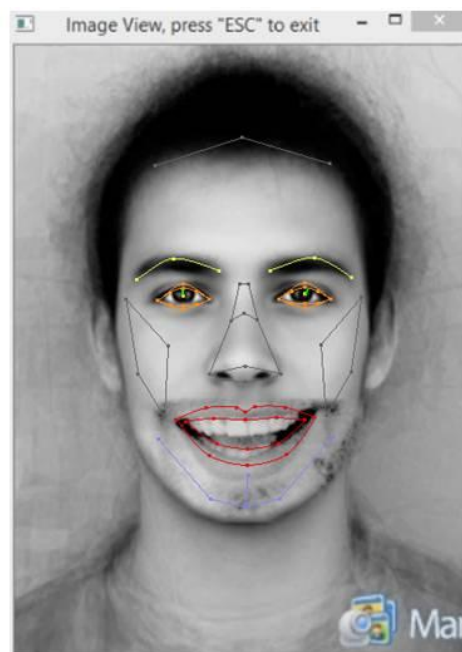
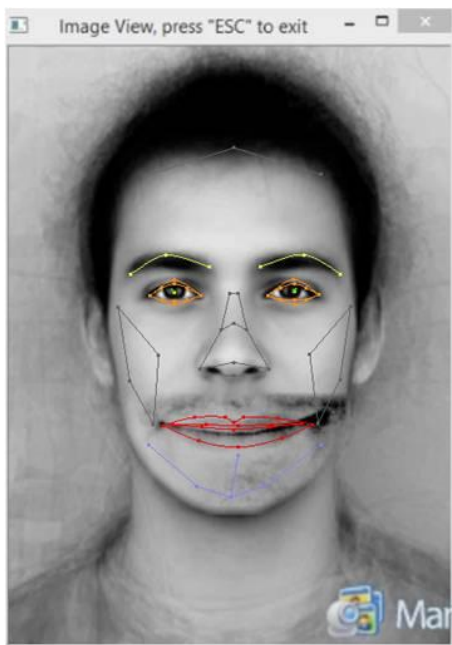
a)



b)

Slika 17. Rezultat spajanja (2)

Dobiveni rezultat je zatim predan programskom alatu za praćenje kontura lica Visage FaceTracker koji je uspješno prepoznao rezultat kao jedno lice te je uredno pratio sve geste koje su mu predane. Na slici 18. bit će prikazano nekoliko fotografija rezultata koje demonstriraju rad ovog alata, a uspješan rad ovog alata je preduvjet za daljnju animaciju.



Slika 18. Rad alata za praćenje kontura lica

6. Zaključak

Cilj ovog rada bio je predstavljanje metode koja spaja sliku kamere montirane na vanjsku stranu HMD uređaja sa slikom nekog unaprijed određenoga ljudskog lica. Kamera se prvo kalibrira kako bi se poništile nepravilnosti koje kamera nosi, a zatim se histogramskim operacijama podešavaju svjetlina i kontrast na slici kamere kako bi dvije slike prilikom spajanja bile što sličnije.

Predložena metoda unosi i nekakve pretpostavke koje su opravdane, ali unosi i nekakva ograničenja koja se mogu svladati korištenjem drugačije opreme. Primjerice, položaj kamere montiran na HMD uređaj potpuno je opravdan. Naime, velika većina ljudskih lica je zrcalno simetrična s obzirom na liniju nosa pa je opravdano postaviti kameru na sredinu. Dodatno, ako bi se napravio nosač za kameru kojim se može regulirati nagib snimanja, osiguralo bi se da svaka kamera uvijek snima onaj dio lica koji je zanimljiv u pogledu animacije. Problematika izbora položaja kamere i njenog nagiba može se svladati i korištenjem drugačijeg tipa kamere. Također, trenutna kamera ima malo vidno polje, zbog čega se kod nekih osoba ne može snimiti cijela brada što za neke geste lica smanjuje efikasnost programa za praćenje kontura lica. Kada bi se koristila kamera koja unosi bačvastu distorziju, kamera bi se mogla postaviti puno bliže, pod nekim fiksnim kutom tako da nema zahtjeva da korisnik sam podešava njen kut snimanja. Zbog svojstva kamere, ona bi imala puno veći kut snimanja pa bi se automatski mogao snimati i veći dio lica, a kako se prije obrade bilo kojeg okvira poništavaju iskrivljenja, uspješno bi se mogla obrađivati takva slika. Nadalje, trenutno rješenje može se nadograditi tako da uz snimanje donjeg dijela lica snimamo i oči korisnika, no tada bi takvo rješenje bilo ovisno o pojedinom HMD uređaju. Svaki HMD uređaj pokriva lice počevši od linije obrva pa sve do linije vrha nosa, a kako uređaji ispred zaslona imaju držače za dvije leće, mogle bi se ugraditi i dvije manje kamere u gornje kutove tih držača koje bi snimale oči i dijelove obrva na način da ih korisnik ni ne primjećuje, čime bi se dodatno poboljšala animacija lika u virtualnom svijetu.

Usprkos navedenim nedostacima, realizirani sustav pokazao se dovoljno dobrim uz zadovoljavajuće performanse obrade pojedinog okvira. To se najbolje očituje

testiranjem postojećega programskog alata za praćenje kontura koji ne prepoznaje razliku između stvarnog lica i lica definiranog na ovaj način, što je bio jedan od konačnih ciljeva.

7.Literatura

- [1] What is virtual reality?. 8. travnja 2015. godine. *Virtual reality - What is virtual reality?*. <http://www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html> 12. lipnja 2015.
- [2] Rheingold, H., Virtual Reality. First Edition. New York, New York, USA. Touchstone, 1992.
- [3] Zyda, M. From visual simulation to virtual reality to games. Computer. Volume 38, Issue 9, listopad 2005., str. 25 – 32
- [4] Edelson, J. Predicting The Impact Of Virtual Reality. 9. listopada 2014. godine. *Predicting The Impact Of Virtual Reality On Mainstream Markets* <http://www.forbes.com/sites/theyec/2014/10/09/predicting-the-impact-of-virtual-reality-on-mainstream-markets/>. 12. lipnja 2015.
- [5] Rosoff, M. Virtual reality is going to grow like crazy over the next five years. 19. ožujka 2015. *Tech – Virtual reality is going to grow like crazy over the next five years*. <http://www.businessinsider.com/chart-of-the-day-virtual-reality-headset-sales-projections-2015-3>, 12. lipnja 2015.
- [6] Johnson, B. Opening the Rift, 07. ožujka 2014. godine. *How the Oculus Rift Works: Opening the Rift*. <http://electronics.howstuffworks.com/oculus-rift1.htm>, 12.06.2015
- [7] Michaels, P. Oculus Reveals Consumer Rift, 11. lipnja 2015. *Tom's Guide – Oculus Reveals Consumer Rift*. <http://www.tomsguide.com/us/oculus-rift-xbox-oculus-touch,news-21092.html>. 13. lipnja 2015.
- [8] Statt, N. Facebook has Oculus, Google has Cardboard, 25. lipnja 2014. *News – Facebook has Oculus, Google has Cardboard*, <http://www.cnet.com/news/facebook-has-oculus-google-has-cardboard/>, 15. travnja 2015.
- [9] OpenCV dev team, Camera calibration With OpenCV, 25. veljače 2015. *Camera calibration With OpenCV*, http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html, 2. ožujka 2015.

- [10] Hugemann, W. Correcting Lens Distortions in Digital Photographs, 19. EVU Conference, Prague, 2010, http://www.imagemagick.org/Usage/lens/correcting_lens_distortions.pdf
- [11] Brown, D. C., Decentering Distortion of lenses, Photogrammetric Eng. Remote Sensing, Vol. 32, No. 3, svibanj 1966, pp. 444-462
- [12] Conrady, A., Decentered lens system, Monthly notices of the Royal Astronomical Society, Vol 79, 1919., pp. 384-390.
- [13] De Villiers, J. P., Leuschner, F. W., & Geldenhuys, R., Centi-pixel accurate real-time inverse distortion correction. Proc. SPIE 7266, Optomechatronic Technologies. 17 studenog 2008.
- [14] Zhang, Z. A flexible new technique for camera calibration., IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI). Volume 22, Issue 11. Studeni, 2000. pp. 1330-1334.
- [15] More, J. J. The Levenberg – Marquardt algorithm, implementation and theory. Numerical analysis. Lecture Notes in Mathematics. Volume 630. Dundee, 1978. pp. 105-116
- [16] Banerjee, S. How to compute a homography, 20. siječnja 2008., *Projective geometry, camera models and calibration: How to compute a homography*. <http://www.cse.iitd.ernet.in/~suban/vision/geometry/node24.html>, 17. ožujka 2015.
- [17] OpenCV dev team, Camera Calibration and 3D Reconstruction. 25. veljače 2015. *Camera Calibration and 3D Reconstruction*. http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html, 2. ožujka 2015
- [18] Gonzalez, R. C., Woods, R. E. Digital Image Processing. Second edition. New Jersey: Prentice Hall, 2002.
- [19] Moses, A. Technology plots the average face of Sydney... and the rest of the world. 11. veljače 2011. *Technology plots the average face of Sydney... and the rest of the world*. <http://www.businessinsider.com/faces-of-tomorrow-2011-2?op=1>, 19. ožujka 2015.

SUSTAV ZA GEOMETRIJSKU TRANSFORMACIJU I POVEZIVANJE DIJELOVA SLIKE

Sažetak

Tržište računalne i mobilne industrije danas je tržište najbržeg rasta i razvoja, a u zadnjih nekoliko godina rapidno raste interes za uređaje koji imaju mogućnost potpunog uranjanja u virtualni svijet. Iako se tehnologija najviše povezuje uz videoigrice, platforma nudi nove mogućnosti za video, internet oglašavanje i još mnogo više. Jedna od mogućnosti koje ta tehnologija može omogućiti je i ostvarivanje virtualnog razgovora, s tim da osoba s kojom razgovaramo bude animirani lik po želji. Trenutno to nije moguće jer je pola lica prekriveno HMD uređajem. Iako se taj problem može vrlo lako riješiti, ostaje još jedan problem. Naime, nijedan današnji program za praćenje lica nema mogućnost praćenja samo donje polovice lica. Ovaj rad zaobilazi navedenu problematiku tako što pretpostavlja jednu kameru na HMD uređaju koja snima donji dio lica, a onda takvu sliku spaja s unaprijed određenim ljudskim licem pazeći da to preklapanje bude što prirodnije. Prije nego što se uopće počnu obrađivati slikovni okviri kamere, kamera se kalibrira uz pretpostavku radijalne i tangencijalne distorzije. Zatim se na svakom okviru odvija poništenje distorzije koeficijentima dobivenim u procesu kalibracije. Nakon toga slijedi operacija modeliranja histograma prema histogramu pozadine kako bi varijacije u kontrastu i osvjetljenju bile što sličnije pozadini. Konačno se takva slika spaja u jednu cjelinu pomoću binarne maske zaglađene Gaussovim filtrom. Predloženo rješenje ne ovisi o specifičnom HMD uređaju niti o nekoj posebnoj kameri.

Ključne riječi: kalibracija kamere, modeliranje histograma, zaglađivanje, Gaussov filter

SYSTEM FOR GEOMETRIC TRANSFORMATION AND IMAGE STITCHING

Abstract

Today, market with one of the highest growth and development is computer and mobile industry market, and in last few years there is rapid interest in devices that have the possibility of total immersion into a virtual world. Although this technology is usually linked with video games, platform provides new possibilities for video, e-advertising and many more. One of the possibilities that this technology offers is also virtual chatting, where the person we are talking to can be animated character as desired. That is not currently possible, because half of the face is covered with HMD. Although that problem can be easily overcome, we still have one problem to solve. Namely, none of today's softwares for facial tracking have the ability to track only lower part of human face. This paper bypasses stated problems by assuming one camera mounted on HMD that records only lower part of human face, and then it joins that picture with predefined image of human face ensuring that given overlay looks as natural as possible. Before any frame is processed, camera is calibrated with assumption of radial and tangential distortion. After that, every frame is undistorted by coefficients calculated in calibration process. Afterwards, histogram of every frame is modelled towards histogram of background so the contrast and brightness would be more similar to the background. Finally, processed image is merged into a whole using binary mask that is filtered with Gaussian kernel. Given solution is independent of specific HMD or special sort of camera.

Keywords: camera calibration, histogram specification, smoothing, Gaussian kernel

Privitak A

Upute za instalaciju i korištenje biblioteke OpenCV 2.4.11.0 na operacijskom sustavu Windows 7/8.0/8.1 uz korištenje razvojne okoline Visual Studio 2012:

1. S web stranice <http://opencv.org/downloads.html> preuzeti verziju 2.4.11
2. Nakon što se datoteka preuzme, pokrenuti je i sve potrebne datoteke raspakirati negdje na disk (pri tome paziti da korisnik ima pravo pristupa tom području diska te da ima pravo čitanja i pisanja po tom dijelu). U slučaju da korisnik ima dvije particije diska, zgodno je sve datoteke smjestiti na nesistemska particiju. Pretpostavimo da je za vršnu točku smještanja odabrana putanja 'D:\opencv\'
3. Odrediti verziju operativnog sustava – Control Panel -> System -> kartica System pa polje System type. Ako je tip operativnog sustava 32-bitni, onda je arhitektura sustava x86, a ako je operativni sustav 64-bitni, onda je arhitektura sustava x64.
4. Visual Studio 2012 i OpenCV 2.4.11 podržavaju vc11 verziju pa će se u ovim uputama koristiti upravo ta verzija.
5. Sada možemo podesiti varijable okoline. Korištenjem grafičkog sučelja to možemo na sljedeći način.
 - a. Control Panel -> System.
 - b. U lijevom gornjem uglu prozora nalazi se 'Advanced system settings' na koji kliknemo jednom
 - c. U novootvorenom prozoru kliknuti na gumb 'Environment Variables'.
 - d. Sada se otvara novi prozor u kojem imamo dva polja. Ispod svakog polja postoje i tri gumba – New..., Edit... i Delete. Ispod drugo polja kliknuti na gumb New...
 - e. Otvara se prozor s dva tekstualna polja za unos. Prvo se naziva 'Variable name', a drugo 'Variable Value'. Za 'Variable name' pišemo 'OPENCV_DIR', a za 'Variable value' pišemo 'D:\opencv\build\x86\vc11'

6. Sada treba doraditi i sistemsku varijablu.
 - a. U prozoru koji se otvara nakon što kliknemo na 'Advanced system settings' pronaći varijablu imena 'Path' ili 'PATH' u donjem polju. Odabrati tu varijablu pa kliknuti na gumb 'Edit...'
 - b. Opet se otvara prozor s dva tekstualna polja – 'Variable name' i 'Variable value'. U polju 'Variable value' na kraj dodajemo ';%OPENCV_DIR%\bin'
7. Ovime je završena instalacija OpenCV-a. Sada treba konfigurirati Visual Studio.
8. Pokrenuti Visual Studio -> File -> New Project...
 - a. Odabrati 'Visual C++' na lijevoj strani prozora, a kao tip projekta odabrati 'Win32 Console Application' te napisati ime projekta. Zatim kliknuti na gumb 'OK'.
9. U novootvorenom prozoru u alatnoj traci otići na PROJECT -> ProjectName Properties. Zatim napraviti sljedeće korake:
 - a. U padajućem izborniku 'Configuration' koji se nalazi lijevo pri vrhu prozora odabrati 'All Configurations'.
 - b. U popisu koji se nalazi s lijeve strane odabrati 'Configuration Parameters' -> C/C++ -> General
 - c. S desne strane postoji polje 'Additional Include Directories' u koje treba upisati '\$(OPENCV_DIR)\..\..\include'
 - d. U popisu koji se nalazi strane pod 'Configuration Parameters' pronaći 'Linker' -> General
 - e. S desne strane postoji polje 'Additional Library Directories' u koje treba upisati '\$(OPENCV_DIR)\lib'
 - f. U popisu koji se nalazi strane pod 'Configuration Parameters' pronaći 'Linker' -> Input pa s desne strane pronaći polje 'Additional Dependencies', kliknuti na malu strelicu prema dolje pa izabrati 'Edit...'
 - g. Dodati sljedeće biblioteke:

opencv_calib3d2411d.lib,	opencv_contrib2411d.lib,
opencv_core2411d.lib,	opencv_features2d2411d.lib,
opencv_flann2411d.lib,	opencv_gpu2411d.lib,
opencv_highgui2411d.lib,	opencv_imgproc2411d.lib,

opencv_legacy2411d.lib,
opencv_nonfree2411d.lib,
opencv_ocl2411d.lib,
opencv_stitching2411d.lib,
opencv_ts2411d.lib,
opencv_videostab2411d.lib

opencv_ml2411d.lib,
opencv_objdetect2411d.lib,
opencv_photo2411d.lib,
opencv_superres2411d.lib,
opencv_video2411d.lib,

h. Kliknuti na 'OK' pa zatim ponovno na 'OK'.

10. Ovime je završeno i konfiguriranje Visual Studio-a