

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 3667

**Mjerenje udaljenosti u ravnini ceste
kamerom postavljenom na vozilu**

Josip Milić

Zagreb, lipanj 2014.

Umjesto ove stranice umetnite priložen izvornik rada

Zahvaljujem se mentoru Izv.prof.dr.sc. Siniši Šegviću na mentorstvu i stručnim savjetima kojima je pridonio ostvarenju ovog rada. Zahvaljujem se i mr. sc. Marku Ševroviću na ustupljenoj kameri.

SADRŽAJ

1. Uvod	1
2. Model stvaranja slike	2
2.1. Model stvaranja slike kamerom s točkastim otvorom	2
2.2. Preslikavanje točaka 3D prostora u 2D sliku	3
3. Kalibracija kamere	7
3.1. Radijalno i tangencijalno izobličenje slike	7
3.2. Intrinzični parametri kamere	9
3.3. Ispravljanje izobličenja slike	10
4. Određivanje udaljenosti u ravnini ceste	13
4.1. Određivanje homografije	13
4.2. Određivanje udaljenosti u ravnini ceste pomoću referentnog koordinatnog sustava	16
5. Programska podrška i implementacija	18
5.1. Programska podrška	18
5.2. Programska implementacija	19
6. Eksperimentalni rezultati i problemi	23
6.1. Korištena kamera	23
6.2. Korišten kalibracijski uzorak	24
6.3. Eksperimentalni rezultati	26
6.4. Eksperimentalni problemi	30
7. Zaključak	31
8. Literatura	32

1. Uvod

Računalni vid (eng. *Computer Vision*), područje umjetne inteligencije (eng. *Artificial Intelligence*), pokušava uz pomoć računala i slika oponašati i unaprijediti ljudsku sposobnost percepcije svoje okoline. Glavni zadatak računalnog vida je automatska ekstrakcija i analiza korisne informacije iz digitalne slike. Njegov razvoj započeo je u 60-im godinama prošlog stoljeća, a intenzivirao se u posljednjim desetljećima zbog značajnog povećanja računalne moći. Rješenja koje nudi računalni vid primjenjuju se u mnogim granama društva, od zabavnog sadržaja do značajnog olakšanja i ubrzanja mnogih poslova.

U ovom radu potrebno je izračunati geometrijske odnose odnosno udaljenost u ravnini ceste uz pomoć slike nastale kamerom na vozilu. Krajnji cilj je omogućavanje korisniku odabiranje više točaka na slici nakon čega će mu se automatski prikazati stvarna međusobna udaljenost odabranih točaka i stvarna površina lika u ravnini ceste čiji su vrhovi odabrane točke. U sljedećim poglavljima pokazat će se koji problemi nastaju u ovom naizgled jednostavnom zadatku i kako ih riješiti dobro poznatim metodama koje nudi računalni vid.

Poglavlje 2. 'Model stvaranja slike' opisuje preslikavanje točaka iz prostora u ravninu slike odnosno princip snimanja točaka. Sljedeće, u poglavlju 3. 'Kalibracija kamere' opisuje se nužnost i postupak kalibracije kamere pri ispravljanju izobličenja slike koje nastaju kamerom. Poglavlje 4. 'Određivanje udaljenosti u ravnini ceste' opisuje pronalazak ravninske transformacije pomoću koje se točke slike preslikaju u ravninu ceste i nakon čega im se odrede geometrijski odnosi. Zatim se u poglavlju 5. 'Programska implementacija' opisuje kako se problemski zadatak programski riješio. Poglavlje 6. 'Eksperimentalni rezultati i problemi' opisuje preciznost rezultata i uočene probleme. U poglavlju 6. 'Zaključak' donosi se zaključak rada, opisuje primjena rješenja i mogućnost poboljšanja. Konačno, poglavlje 7. 'Literatura' navodi popis literature koja se koristila u ovom radu. Nakon poglavlja slijedi sažetak rada, a na zadnjoj stranici se nalazi privitak s DVD-ROM na kojem se nalazi izvorni programski kod zajedno s brojnim primjerima koji se mogu koristiti.

2. Model stvaranja slike

Prije opisa rješenja problemskog zadatka potrebno je razumjeti kako nastaje slika uz pomoć kamere odnosno kako se točka 3D (trodimenzionalnog) prostora preslikava u 2D sliku. Navedeno je opisano u ovom poglavlju. Model stvaranja slike izravno je važan za kalibraciju kamere čiji je postupak opisan u sljedećem poglavlju.

2.1. Model stvaranja slike kamerom s točkastim otvorom

Riječ kamera potječe od starogrčke riječi *καμάρα* (lat. *camera*) što u jednom od tumačenja znači nadstvođena soba (komora). Ibn al-Haytham (poznatiji pod imenom Alhazen¹) u 11. stoljeću je opisao pojavu prikaza invertirane slike koja nastaje prolaskom svjetlosti kroz sitnu pukotinu. Uočio je da se smanjivanjem pukotine slika izoštrava, ali i da postaje tamnija zbog manje svjetlosti. Time je zapravo opisao kasnije vrlo poznatu napravu pod imenom *camera obscura* čiji je prijevod s latinskog tamna komora. Nakon godina razvoja koncepta kamere, što uključuje i dodavanje sustava leća, početkom 19. st. Nicéphore Niépce² uspijeva s njom uspješno fotografirati prvu fotografiju. Za fotografiranje se koristila kamera s točkastim otvorom (eng. *Pinhole camera*, doslovni prijevod bi bio kamera ušica igle što odgovara konceptu sitne pukotine). Kamera s točkastim otvorom je preteča današnjih fotoaparata i videokamera i zato se model stvaranja slike kamerom s točkastim otvorom (u nastavku teksta model stvaranja slike) može primijeniti u ovom radu.

¹ Alhazen (oko 965. – oko 1040.), arapski znanstvenik, matematičar, astronom i filozof, smatra se ocem moderne optike, oftamologije, eksperimentalne fizike i znanstvene metodologije.

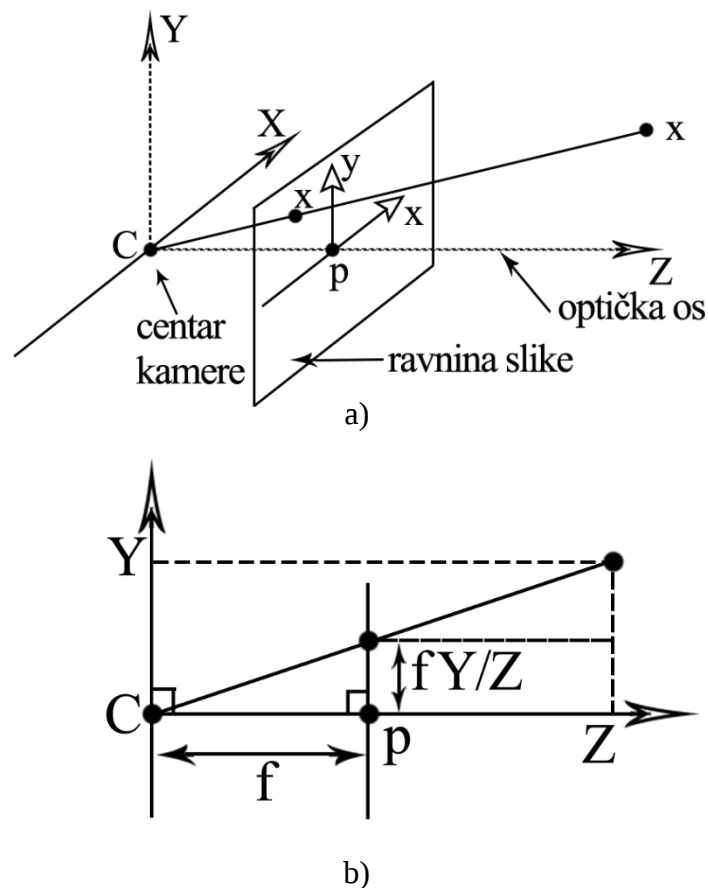
² Nicéphore Niépce (1765. – 1833.), francuski izumitelj, pionir i izumitelj fotografije. Njegov suradnik Louis Daguerre (1787. – 1851.), francuski umjetnik i fotograf, 1839. razvija dagerotipiju, prvi poznati fotografski postupak. Henry Fox Talbot (1800. – 1877.), engleski znanstvenik, pionir fotografije, ubrzo predstavlja bolji i jeftiniji postupak – kalotipski postupak.

2.2. Preslikavanje točaka 3D prostora u 2D sliku

Preslikavanje koje svakoj točki $T \neq O$ pridružuje točku T_c koja je probodište pravca OT i ravnine π naziva se centralnom projekcijom na ravninu π . Neka je centar projekcije ishodište Euklidskog koordinatnog sustava. Na udaljenosti $Z = f$ nalazi se ravnina slike (još poznata kao fokalna ravnina). U modelu se neka koordinata $X = (X, Y, Z)^T$ preslikava u točku na ravnini slike. Uz pomoć sličnosti kutova ustanovi se da se točka X preslikava u točku $(fX/Z, fY/Z, f)^T$ na ravnini slike. Odbacivanjem zadnje koordinate točke dobiva se preslikavanje centralne projekcije iz prostora u koordinatu slike

$$(X, Y, Z)^T \mapsto (fX/Z, fY/Z)^T \quad (2.1)$$

Ovime je zapravo opisano preslikavanje iz Euklidskog 3D prostora $\mathbb{R}^3$ u Euklidski 2D prostor $\mathbb{R}^2$. [1]



Slika 1. – u a) dijelu slike prikazuje se preslikavanje točke u ravninu sliku, u b) dijelu slike prikazuje se sličnost trokuta.

Prikazivanjem stvarne točke u prostoru i točke na slici u obliku homogenih vektora³ navedeno se preslikavanje može prikazati kao linearno preslikavanje između homogenih koordinata

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2.2)$$

Matrica se može sažeto zapisati kao $\text{diag}(f, f, 1)[I \mid 0]$ gdje $\text{diag}(f, f, 1)$ predstavlja dijagonalnu matricu, a $[I \mid 0]$ jediničnu matricu kojoj je dodan stupac, u ovom slučaju 0-vektor. Kompaktno se zajedno može zapisati u obliku

$$x = PX \quad (2.3)$$

gdje je x točka slike prikazana homogenim vektorom, a točka X točka prostora prikazana homogenim vektorom $(X, Y, Z, 1)^T$.

$P = \text{diag}(f, f, 1)[I \mid 0]$ je homogena matrica projekcije kamere dimenzija 3×4 .

Pretpostavljeno je da se točka u kojoj se sijeku optička os kamere i ravnina slike nalazi u ishodištu ravnine slike. U praksi to nije slučaj što treba uzeti u obzir. Zato se izraz (2.1) proširuje

$$(X, Y, Z)^T \mapsto (fX/Z + c_x, fY/Z + c_y)^T \quad (2.4)$$

Koordinata $(c_x, c_y)^T$ je koordinata probodišta p ravnine slike. Potrebno je proširiti i izraz (2.2)

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX + Zc_x \\ fY + Zc_y \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2.5)$$

³ Kod homogenih vektora bitan je samo međusoban omjer.

Iz izraza (2.5) može se izlučiti matrica

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

Matrica K se zove matrica kamere. Ona sadrži vrijednosti sljedećih fokalnih duljina i vrijednosti koordinata optičkog centra na slici:

f_x – fokalna duljina u pikselima dobivena dijeljenjem stvarne fokalne duljine F izražene u milimetrima s horizontalnom duljinom piksela s_x u optičkom senzoru kamere izraženom u milimetrima po pikselu.

$$f_x = F [mm] / s_x [mm/pix] \quad (2.7)$$

f_y – fokalna duljina u pikselima dobivena dijeljenjem stvarne fokalne duljine F izražene u milimetrima s vertikalnom duljinom piksela s_y u optičkom senzoru kamere izraženom u milimetrima po pikselu.

$$f_y = F [mm] / s_y [mm/pix] \quad (2.8)$$

(c_x, c_y) – koordinata točke u kojoj se sijeku ravnina slike i optička os kamere izražena u pikselima. Idealni položaj je na sjecištu vrijednosti polovice dužine i polovice visine slike.

Izraz dobivanja točke x može se kompaktno zapisati u obliku

$$x = K [I | 0] X_{cam} \quad (2.9)$$

Točka X posebno se označava s X_{cam} kako bi se naglasilo da se kamera nalazi u ishodištu Euklidskog koordinatnog sustava i da optička os kamere leži na Z - osi tog sustava. Takav koordinatni sustav obično se naziva koordinatni podsustav (eng. *coordinate frame*, u nastavku teksta podsustav) kamere.

Dva podsustava međusobno se odnose kroz rotaciju i translaciju. Točka X može se prikazati nehomogenim vektorom $\tilde{X}$ koji prikazuje točku u podsustavu stvarnog prostora. Točka X_{cam} može se prikazati s nehomogenim vektorom $\tilde{X}_{cam}$ koji prikazuje istu točku u svojem podsustavu kamere. U tim oblicima zapisuje se sljedeće

$$\tilde{X}_{cam} = R (\tilde{X} - \tilde{C}) \quad (2.10)$$

gdje $\tilde{C}$ predstavlja koordinatu centra kamere u podsustavu stvarnog prostora, a R rotacijsku matricu koja predstavlja orijentaciju podsustava kamere.

Može se zapisati sljedeće

$$X_{cam} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} X \quad (2.11)$$

Uvrštavajući izraz (2.11) u izraz (2.9) dobiva se

$$x = KR [I \mid -\tilde{C}] X \quad (2.12)$$

Ovime je postignuto općenito preslikavanje točaka u ravninu slike odnosno opisan model stvaranja slike kamere s točkastim otvorom.

Takav model

$$P = KR [I \mid -\tilde{C}] \quad (2.13)$$

ima ukupno 9 stupnjeva slobode:

- 3 za matricu kamere K
- 3 za rotacijsku matricu R
- 3 za koordinate $\tilde{C}$.

Vrijednosti matrice kamere intrinzični su parametri kamere. U sljedećem poglavlju objašnjena njihova važnost.

Vrijednosti rotacijske matrice R i koordinate $\tilde{C}$ ekstrinzični su parametri kamere (ekstrinzični jer pokazuju odnos orijentacije i položaja kamere s koordinatnim sustavom prostora).

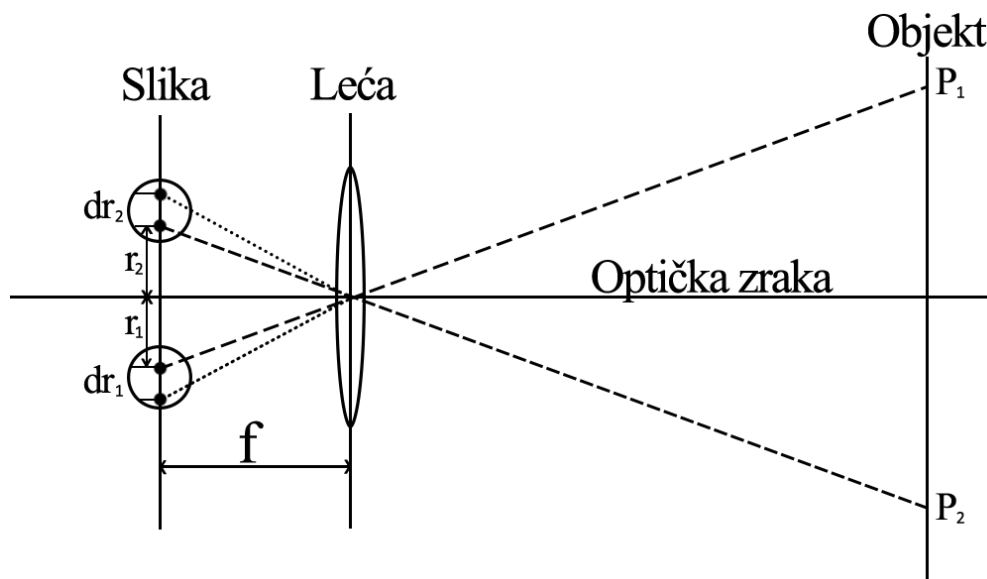
3. Kalibracija kamere

Od polovice 18. st. do danas, tehnologija snimanja neprestano se razvijala, a njegov postupak se olakšavao i postajao dostupniji širem broju ljudi. Dostupnost se najviše postiže manjom cijenom izrade pa tako i proizvoda što uzrokuje neke gubitke u kvaliteti koje velika većina korisnika kompromisno prihvaća ili ih ni ne primjećuje. U ovom poglavlju je objašnjeno kako modelirati i ispraviti izobličenja slike uz pomoć kalibracije kamere.

3.1. Radijalno i tangencijalno izobličenje slike

Problem koji je usko vezan za temu rada je nelinearno izobličenje slike zbog kojeg točke odnosno omjer njihovih međusobnih udaljenosti na slici ne odgovara onom stvarnom. Najistaknutija izobličenja su radijalno i tangencijalno izobličenje slike.

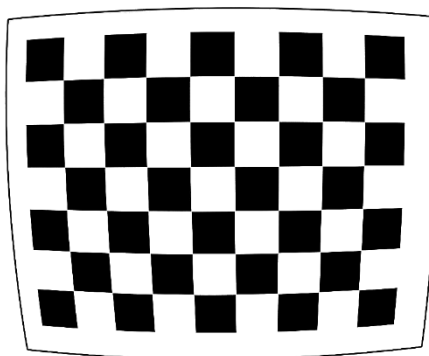
Radijalno izobličenje nastaje zbog korištenja sfernih leća u kameri koje su jeftinije i jednostavnije ih je izraditi za razliku od paraboličnih leća koje se npr. često koriste u astronomiji gdje se zahtijeva velika oštrina slike uz što manje smetnji. Takvo izobličenje najviše se primjećuje u rubovima slike. Zraka svjetlosti, prikazana isprekidanom linijom na *Slici 2.*, trebala bi se kretati potpuno ravno kroz centar leće no umjesto toga savija se nakon prolaska kroz leću. Posljedica toga je pogrešan položaj projekcije točaka u slici koje se ne nalaze na udaljenosti r_i od optičke zrake već se nalaze na udaljenosti $r_i + dr_i$ (prema eng. *distortion* - izobličenje) gdje je pogreška jednaka dr_i . Pogreška se povećava udaljavanjem točke P_i od optičke zrake, a smanjuje u suprotnom gdje se minimum pogreške ($dr_i = 0$) postiže ako se točka P_i nalazi na optičkoj zraci. Kamere koje snimaju s malom fokalnom duljinom f , imaju veći kut snimanja što omogućuje više sadržaja na slici no i veliko radijalno izobličenje u rubovima.



Slika 2. – ilustracija radijalnog izobličenja projekcije točaka P_1 i P_2 . Crtkanim linijama su prikazane idealne, a točkastim linijama prikazane stvarne putanje svjetlosti nakon prolaska kroz centar leće.

Tangencijalno izobličenje, za razliku od radijalnog, ne nastaje zbog samih leća već zbog njihovog položaja u odnosu na ravninu u kojoj se nalazi sklop za pretvorbu svjetlosnog u digitalni signal. Zbog nesavršenosti, koja je posljedica obično grubog procesa izrade, leća se ne nalazi potpuno paralelno u odnosu na ravninu sklopa.

Radijalno i tangencijalno izobličenje najviše se uočava kad se snimani objekt nalazi vrlo blizu kamere i sadrži ravne linije. Slika 4. prikazuje takav primjer u kojem se kamerom snimala šahovnica (pod nekim kutom) prikazana preko cijelog zaslona monitora.



Slika 3. – primjer izobličenja slike. Valja obratiti pozornost na oblik kvadrata u sredini i u rubovima slike.

Izobličenja slika moraju se uzeti u obzir prilikom rekonstrukcije geometrijskih odnosa u sceni uz pomoć slike. Zbog prije navedenih razloga, izobličenja ovise o korištenoj kameri odnosno izvedbi leća koje se nalaze u njoj. To također znači da dva ista modela kamere ne moraju stvarati jednaka izobličenja.

3.2. Intrinzični parametri kamere

Izobličenje je gotovo neizbježno, ali postoji način kako ga matematički ispraviti. Kako bi se to moglo postići potrebni su intrinzični parametri kamere koji su obično prikazani u obliku matrica.

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad D = \begin{bmatrix} k_1 \\ k_2 \\ p_1 \\ p_2 \\ k_3 \end{bmatrix} \quad (3.1)$$

Matrica K, još poznata kao matrica kamere, opisana je u prethodnom poglavlju. Matrica D se zove matrica koeficijenata izobličenja. Obično se uzimaju po dva koeficijenta oba izobličenja (k_1, k_2 za radijalno i p_1, p_2 za tangencijalno izobličenje) dok je moguće koristiti i više koeficijenata za radijalno izobličenje. Vrijednosti matrice koeficijenata izobličenja vrijede za bilo koju rezoluciju slike snimljenom istom kamerom. U slučaju mijenjanja rezolucije, vrijednosti matrice kamere moraju se sukladno skalirati.

Kalibracija kamere [2] zapravo je umjeravanje matrice K i matrice D. U prethodnom poglavlju opisan je postupak dobivanja matrice P, modela stvaranja slike. Matrica P se još zove potpuna kalibracijska matrica kamere. Uz pomoć nje mogu se odrediti intrinzični i ekstrinzični parametri kamere. Potpuna kalibracijska matrica kamere je dimenzija 3×4 . Sadrži dakle ukupno 12 vrijednosti no zapravo su nepoznate 11 vrijednosti matrice jer vrijednost elementa zadnjeg retka i stupca matrice iznosi 1. Izraz (2.13) može se preoblikovati na sljedeći način

$$P = [KR] - \tilde{C}KR = [M] - \tilde{C}M \quad (3.2)$$

gdje je matrica $M=KR$.

QR dekompozicijom matrica M može se rastaviti na dvije 3×3 matrice, $M=AB$. Matrica A je gornja trokutasta, a matrica B ortogonalna matrica. Vidljivo je da matrica A korespondira matrici kamere K , a matrica B matrici rotacije R .

Iz izraza (3.2) može se izlučiti četvrti stupac matrice potpune kalibracijske matrice

$$p_4 = -\tilde{C}M \rightarrow -\tilde{C} = M^{-1}p_4 \quad (3.3)$$

Kako bi se pronašla matrica P potrebne su korespondencije između stvarnih 3D točaka i njihovih projekcija na 2D slici. Koristeći potpunu kalibracijsku matricu u obliku

$$P = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \quad (3.4)$$

gdje r_i predstavlja i -ti redak matrice P i poznavanjem korespondencije stvarne točke X i njene projekcije na ravninu slike x mogu se zapisati izrazi za koordinatu projekcije točke (x, y)

$$x' = x/z = (r_1X)/(r_3X) \quad (3.5)$$

$$y' = y/z = (r_2X)/(r_3X) \quad (3.6)$$

pomoću kojih se dobiju dvije linearne jednadžbe

$$x'(r_3X) - r_1X = 0 \quad (3.7)$$

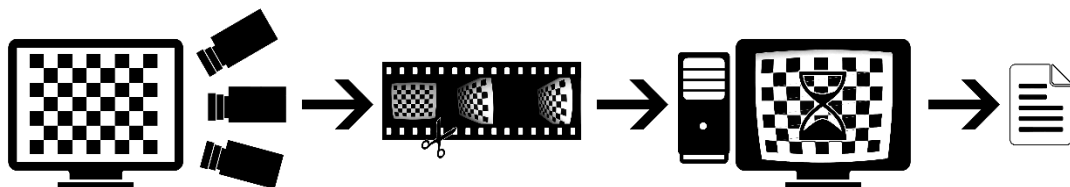
$$y'(r_3X) - r_2X = 0 \quad (3.8)$$

gdje su r_1, r_2 i r_3 nepoznanice. Kako za svaku točku odnosno korespondenciju između stvarne točke i njezine projekcije na ravninu slike postoje dvije linearne jednadžbe, potrebno je ukupno najmanje 6 korespondencija kako bi se izračunalo svih 12 matrice P . Zbog bolje preciznosti, koristi se mnogo više od tog broja nakon čega se metodom singularne dekompozicije (eng. *Singular Value Decomposition, SVD*) izračunaju sve vrijednosti potpune kalibracijske matrice

$$A = U\Sigma V^T \quad (3.9)$$

Intrinsični parametri kamere programski se navedenim postupkom umjeravaju uz pomoć slika nastale fotografiranjem ili vađenjem iz video snimke. Na slikama je potrebno imati kalibracijski uzorak s precizno poznatim dimenzijama. Kalibracijski uzorak može biti šahovnica s crnim i bijelim kvadratima dok se često koriste i crni krugovi na bijeloj podlozi. U svrhu kalibracije, postoje dva načina snimanja. U jednom se položaj kamere ne mijenja, već se kalibracijski uzorak pomiče u razne položaje. U drugom načinu, koji se češće koristi, se radi obrnuto. Bez obzira na odabran način snimanja, kalibracijski uzorak bi trebao biti prikazan potpuno u kadru i biti sniman u što više različitih položaja s naglaskom na ona u kojima se izobličenje najviše primjećuje.

Programski postupak na ulazu, osim navedenih slika, zahtijeva, u slučaju s pravilnim poljem značajki, broj unutarnjih kutova u uzorku i stvarnu udaljenost između dva susjedna vrha u jednom retku (duljina stranice kvadrata). Postupak svakom od unutarnjih kutova daje slijedno koordinatu u koordinatnom sustavu. Zatim za svaku sliku traži uzorak s danim brojem unutarnjih kutova. U slučaju kad je broj pronađenih kutova manji od ukupnog broja stvarnih kutova slika se odbacuje, a u suprotnom se rezultati prihvaćaju i uspoređuju s koordinatama.



Slika 4. – ilustracija pojednostavljenog procesa dobivanja intrinzičnih parametara kamere. Kalibracijski uzorak se snima iz različitih perspektiva, zatim se iz snimke izvade slike koje se programski obrađuju i na kraju kao rezultat programa dobivaju se intrinzični parametri u obliku datoteke koja se može koristiti za ispravljanje izobličenja točaka u slikama nastalim istom kamerom.

3.3. Ispravljanje izobličenja slike

Vrijednosti matrice kamere K i matrice koeficijenata izobličenja D dobivenih iz intrinzičnih parametara kamere mogu se koristiti za ispravljanje izobličenja neke točke na slici odnosno pogreška položaja točke može se matematički eliminirati [3].

Za ispravljanje radijalnog izobličenja koriste se sljedeće jednačbe:

$$x_p = x(1 + \sum_{i=1}^n k_i r^{2i}) \quad (3.10)$$

$$y_p = y(1 + \sum_{i=1}^n k_i r^{2i}) \quad (3.11)$$

Gdje je (x, y) originalna, a (x_p, y_p) ispravljena koordinata točke na slici.

Radijus r predstavlja udaljenost koordinate (x, y) od optičkog centra slike (c_x, c_y) :

$$r = \sqrt{(x - c_x)^2 + (y - c_y)^2} \quad (3.12)$$

Koeficijenti k_i koeficijenti radijalnog su izobličenja.

Za tri koeficijenta ($n = 3$), broj koji se koristio u ovom radu, jednačbe su sljedeće:

$$x_p = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (3.13)$$

$$y_p = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (3.14)$$

Za ispravljanje tangencijalnog izobličenja koriste se sljedeće jednačbe:

$$x_p = x + [2p_1 xy + p_2(r^2 + 2x^2)] \quad (3.15)$$

$$y_p = y + [2p_2 xy + p_1(r^2 + 2y^2)] \quad (3.16)$$

Gdje su koeficijenti p_1 i p_2 koeficijenti tangencijalnog izobličenja.

4. Određivanje udaljenosti u ravnini ceste

Problemski zadatak ovog rada je pronaći geometrijske odnose u sceni uz pomoć slike. U 2. poglavlju opisan je postupak preslikavanja stvarnih točaka iz scene u ravninu slike. Te preslikane točke mogu poslužiti za mjerenje stvarne međusobne udaljenosti točaka ako se preslikaju u ravninu slike koja predstavlja ravninu ceste. Ovo poglavlje objašnjava kako to postići.

4.1. Pronalazak homografije

Točka na slici ima dvije dimenzije, svoju dužinu i širinu koje su prikazane u obliku koordinate (x, y) Kartezijevog koordinatnog sustava. Ona se može prikazati kao 3D vektor $\mathbf{x} = (X, Y, Z)^T$ gdje su $x = \frac{X}{Z}$ i $y = \frac{Y}{Z}$. To je homogen prikaz točke na ravnini π . Točke te ravnine mogu se preslikati u istu ravninu, ali pod uvjetom da ako su na istom pravcu onda i njihove projicirane točke moraju također biti na istom pravcu. Homografija je naziv za ravninsku projekcijsku transformaciju kojom se omogućava navedeno preslikavanje.

Prikazuje se transformacijskom matricom H i vrijedi sljedeće:

$$\mathbf{x}' \sim H \cdot \mathbf{x} \quad (4.1)$$

gdje su:

$$\mathbf{x}' = [X' \quad Y' \quad Z']^T \quad (4.2)$$

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (4.3)$$

Transformacijska matrica H homogena je matrica jer u njoj je bitan samo omjer elemenata matrice. Ukupno je 8 međusobno neovisnih omjera u homografiji. Skaliranjem proizvoljnim faktorom $F \neq 0$ homografija se ne mijenja.

Koordinata točke (x, y) uz pomoć navedenog se transformira u novu koordinatu (x', y') :

$$x' = \frac{h_{11}x_1 + h_{12}y_1 + h_{13}}{h_{31}x_1 + h_{32}y_1 + h_{33}} \quad (4.4)$$

$$y' = \frac{h_{21}x_1 + h_{22}y_1 + h_{23}}{h_{31}x_1 + h_{32}y_1 + h_{33}} \quad (4.5)$$

Kako bi se izračunala međusobna udaljenost stvarnih točaka uz pomoć točaka slike potrebno je preslikati točke slike u ravninu ceste. Točke se snimaju obično pod nekim kutom α između normale ravnine i kamere. Kad bi vrijedilo $\alpha = 0$ (kamera snima ravninu ceste pod okomitim kutom), omjer međusobnih udaljenosti između točaka na slici bi odgovarao omjeru stvarnih točaka. To je idealan slučaj no u općem slučaju problem se može riješiti matematički.

Poznavanjem udaljenosti stvarnih točaka i gdje se projekcije tih točaka nalaze na slici moguće je izračunati homografiju H [4].

Redak j matrice H može se zapisati kao $h^j{}^T$ uz pomoć čega se izraz (4.1) može zapisati i na ovaj način:

$$Hx_i = \begin{bmatrix} h^1{}^T x_i \\ h^2{}^T x_i \\ h^3{}^T x_i \end{bmatrix} \quad (4.6)$$

x_i prikazuje točke, a x_i' točke koje su nastale preslikavanjem x .

x' i Hx_i ne moraju biti numerički jednaki već se razlikuju za određen faktor F no imaju jednaki smjer i zato vrijedi

$$x_i' \times Hx_i = 0 \quad (4.7)$$

Zapisujući $x_i'^T = [x_i' \ y_i' \ z_i']^T$ vrijedi

$$x_i' \times Hx_i = \begin{bmatrix} y_i' h^3 x_i - z_i' h^2 x_i \\ z_i' h^1 x_i - x_i' h^3 x_i \\ x_i' h^2 x_i - y_i' h^1 x_i \end{bmatrix} \quad (4.8)$$

Vrijedi $h^{jT} x_i = x_i^T h^j$ čime se sustav jednadžbi $x_i' \times Hx_i = 0$ raspisuje u obliku

$$\begin{bmatrix} 0^T & -z_i' x_i^T & y_i' x_i^T \\ z_i' x_i^T & 0^T & -x_i' x_i^T \\ -y_i' x_i^T & x_i' x_i^T & 0^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0 \quad (4.9)$$

U izrazu broj prva matrica A_i je dimenzija 3×9 , a druga matrica h je zapravo vektor dimenzija 9×1 (9 vrijednosti matrice H). Matrica A_i je ranga 2 što znači da za svaki par (x, x') postoje dvije jednadžbe.

Popunjavajući sustav za npr. četiri točke ($i = 1, 2, 3, 4$) dobiva se izraz $Ah = 0$ gdje matrica A je dimenzije 12×9 , a rang joj iznosi 8. Konačno, metodom singularne dekompozicije, izraz (3.9), moguće je izračunati matricu H jer zadnji stupac matrice V predstavlja vektor h .

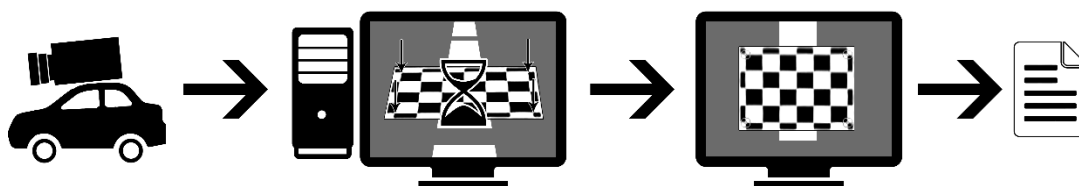
Objašnjeno je kako preslikati točke slike u ravninu slike koja predstavlja ravninu ceste. Izračun homografije ravnine ceste zahtijeva poznavanje odnosa projekcija određenih točaka ravnine na slici. Kao i u slučaju kalibracije kamere u tu svrhu se koristi pravilan kalibracijski uzorak precizno poznatih dimenzija.

Uzorak se postavlja u ravninu ceste i mora biti u kadru snimanja. Poželjno je koristiti uzorak većih dimenzija čime se postiže precizniji rezultat. U slučaju korištenja

šahovnice kao kalibracijskog uzorka značajke predstavljaju unutarnji kutovi odnosno vrhovi kvadrata šahovnice. Međusobna udaljenost dvije susjedne značajke je poznata duljina stranice kvadrata.

Projekcije značajki na slici ručno se odabiru ili programski pronalaze uz pomoć automatske detekcije kutova na slici. Naknadno se može odrediti podpikselsna preciznost značajki na slici kako bi rezultat bio što precizniji.

Pomoću poznatih geometrijskih odnosa značajki na kalibracijskom uzorku i njihovih projekcija na slici izračuna se matrica homografije H . Ona se može spremiti i koristiti za preslikavanje točaka sve dok se ne promjeni položaj kamere. Ovo je korisna mogućnost jer nije potrebno imati u svakoj slici snimke kalibracijski uzorak u ravnini ceste, već samo u jednoj od njih. Promjenom položaja kamere potrebno je ponoviti postupak određivanje homografije.



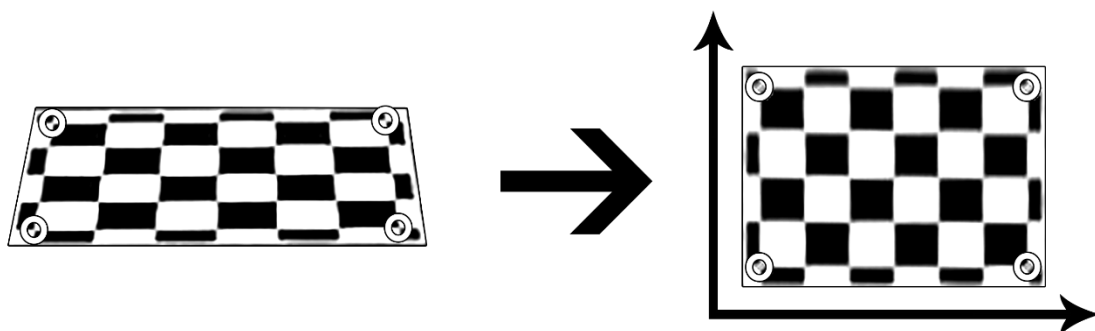
Slika 5. – ilustracija pojednostavljenog procesa dobivanja homografije. Iz snimke dobivene kamerom postavljenoj na vozilu izvadi se slika koja sadrži kalibracijski uzorak. Na njoj se automatski ili ručno pronađu značajke nakon čega se izračuna matrica homografije koja se spremi u obliku datoteke. Datoteka se može koristiti za ravninsko preslikavanje točaka slike ako kamera nije mijenjala položaj.

4.2. Određivanje udaljenosti u ravnini ceste pomoću referentnog koordinatnog sustava

Razumijevajući napisano u 2. poglavlju mogu se koristiti postupci iz 3. i 4. poglavlja kako bi se konačno mogla odrediti udaljenost u ravnini ceste. U prethodnom dijelu poglavlja opisana je potreba preslikavanja točaka slike u ravninu ceste.

Matrica homografije H je homogena matrica što znači da su bitni samo međusobni omjeri vrijednosti. Homogenost olakšava postupak preslikavanja jer su potrebni samo položaji projiciranih značajki kalibracijskog uzorka. Položaji točaka ravnine ceste u kojoj se prethodno navedene točke moraju preslikati mogu se proizvoljno odrediti uz uvjet da se sačuva pravilan omjer.

Primjer je korištenje šahovnice dimenzije 6×4 kvadrata kao kalibracijskog uzorka gdje su značajke unutarnjih kutovi. Koriste se npr. četiri značajke: prvi unutarnji kut u prvom retku kutova, zadnji unutarnji kut u prvom retku kutova, prvi unutarnji kut zadnjeg retka kutova, zadnji unutarnji kut zadnjeg retka kutova. Prije utvrđivanja položaja njihovih projekcija na slici potrebno je ispraviti izobličenje slike, postupkom objašnjenom u odjeljku 3.3., kako bi konačan rezultat bio precizan. Točke ravnine ceste u koje se značajke preslikavaju mogu biti vrhovi pravokutnika koji ima omjer stranica jednak omjeru retka i stupca unutarnjih kutova kalibracijskog uzorka. Preslikane značajke mogu se promatrati kao koordinate Kartezijevog koordinatnog sustava. Poznate su stvarne međusobne udaljenosti značajki izražene u milimetrima i međusobne udaljenosti projekcije značajki na ravninu ceste izražene u pikselima. Zbog toga se mogu izračunati stvarne međusobne udaljenosti između bilo koje dvije preslikane točke jer je poznato koliko jedan piksel na slici iznosi milimetara u stvarnosti. Shodno se može izračunati površina nekog lika čiji su vrhovi odabrane točke.



Slika 6. – ilustracija preslikavanja značajki u ravninu ceste. Ravnina ceste može se promatrati kao koordinatni sustav koji ima poznato mjerilo i uz pomoć kojeg se mogu izračunati međusobne udaljenosti između bilo kojih preslikanih točaka.

5. Programska podrška i implementacija

Ovo poglavlje opisuje ključne dijelove programske implementacije opisanih postupaka iz prethodnih poglavlja. Najprije je opisana programska podrška uz pomoć koje se izvela implementacija.

5.1. Programska podrška

Python je interpretirani programski jezik visoke razine. Dopušta korištenje nekoliko stilova programiranja (objektno orijentirano, strukturno i aspektno orijentirano programiranje). Navedena fleksibilnost, jednostavna čitljivost i intuitivnost učinili su ga veoma popularnim jezikom. Python je skriptni jezik i kao takav obično se koristi za brzo programiranje. Podržava brojne pakete i biblioteke među kojima su *NumPy* paket i *OpenCV* biblioteka koji su se koristili u ovom radu. Njegova najveća mana u usporedbi s drugim popularnim programskim jezicima je brzina izvođenja. [5].

NumPy (*Numerical Python*) je paket namijenjen Python jeziku koji sadrži mnoge implementirane znanstvene metode. Uključivanjem u Python skriptu omogućava korištenje sofisticiranih funkcija, metoda linearne algebre, Fourierove transformacije i mnogih drugih. Najčešće se koristi, pa tako i u ovom radu, u svrhu stvaranja i korištenja matrica različitih dimenzija. [6].

OpenCV (*Open Source Computer Vision Library*) je biblioteka koju mogu koristiti popularni jezici C++, C, Java i Python. Sadrži brojne metode vezane uz računalni vid. Napisana je i optimizirana u C/C++ jeziku čime postiže veliku brzinu izvođenja, jedan od glavnih razloga zašto se koristi u ovom radu. Velika zajednica ljudi (preko nekoliko desetaka tisuća) uključena je u razvijanje *OpenCV* biblioteke zbog čega ima vrlo široku primjenu [7].

C++ je kompajlirani programski jezik niže razine. Prednost pred drugim sličnim popularnim programskim jezicima mu je brzina jer se prije izvođenja programski kod prevodi potpuno u strojni kod. Sadrži veliki broj biblioteka i nudi mogućnost dodavanja mnogih drugih.

Visual Studio 2013 je programsko okruženje (IDE, *Interactive Development Enviroment*) uz pomoć kojeg se mogu pisati programski kodovi različitih programskih jezika. Najveća prednost nad sličnim rješenjima su mu stabilnost, kvalitetan sustav pomoći pri uklanjanju pogrešaka (eng. *Debugging*) i kvalitetan sustav pomoći pri pisanju koda (eng. *Intellisense*).

PTVS (*Python Tools for Visual Studio*) je dodatak namijenjen Visual Studio IDE-u. Omogućuje pisanje Python izvornog koda u VS-u [8].

5.2. Programska implementacija

Programska implementacija kalibracije kamere napisana je izvorno u C++-. Izdvojena je metoda *runCalibration*

```
static bool runCalibration(vector<vector<Point2f> > imagePoints,
    Size imageSize, Size boardSize, Pattern patternType,
    float squareSize, float aspectRatio,
    int flags, Mat& cameraMatrix, Mat& distCoeffs,
    vector<Mat>& rvecs, vector<Mat>& tvecs,
    vector<float>& reprojErrs,
    double& totalAvgErr)
{
    cameraMatrix = Mat::eye(3, 3, CV_64F);
    if (flags & CV_CALIB_FIX_ASPECT_RATIO)
        cameraMatrix.at<double>(0, 0) = aspectRatio;
    distCoeffs = Mat::zeros(8, 1, CV_64F);
    vector<vector<Point3f> > objectPoints(1);
    calcChessboardCorners(boardSize, squareSize, objectPoints[0],
        patternType);

    objectPoints.resize(imagePoints.size(), objectPoints[0]);
    double rms = calibrateCamera(objectPoints, imagePoints, imageSize,
        cameraMatrix, distCoeffs, rvecs, tvecs, flags | CV_CALIB_FIX_K4 |
        CV_CALIB_FIX_K5);
    printf("RMS error reported by calibrateCamera: %g\n", rms);
    bool ok = checkRange(cameraMatrix) && checkRange(distCoeffs);

    totalAvgErr = computeReprojectionErrors(objectPoints, imagePoints,
        rvecs, tvecs, cameraMatrix, distCoeffs, reprojErrs);
    return ok;
}
```

Metoda *runCalibration* na ulazu prima položaje projekcije točaka na slici, veličinu slike, dimenzije unutarnjih kutova kalibracijskog uzorka, vrstu kalibracijskog uzorka, duljinu stranice kvadrata. Stvara matricu kamere i matricu koeficijenta izobličenja. Uz pomoć *OpenCV* metode *calibrateCamera* vrši postupak opisan u odjeljku 3.2.

Programska implementacija određivanja položaja projekcija značajki kalibracijskog uzorka na slici, izračuna homografije i konačno izračuna udaljenosti u ravnini ceste napisana je u Pythonu. Izdvojeno je nekoliko metoda. Metoda *detekcijaUzorka*

```
def detekcijaUzorka() :
    global slika
    global slikaOriginal
    global detekcijaUkljucena
    sys.stdout.write('Pokrenuta je automatska detekcija uzorka...\n')
    brojUnutarnjihKutovaDetekcija = brojUnutarnjihKutova
    slikaBezIzoblicenja = makniIzoblicenje(0, slikaOriginal)
    while (brojUnutarnjihKutovaDetekcija[0] >= 3 and
    brojUnutarnjihKutovaDetekcija[1] >= 3) :
        podaciUzorkaDetekcija = cv2.findChessboardCorners(slikaBezIzoblicenja,
        tuple(brojUnutarnjihKutovaDetekcija), flags = cv2.CALIB_CB_ADAPTIVE_THRESH
        + cv2.CALIB_CB_NORMALIZE_IMAGE)
        if (podaciUzorkaDetekcija[0]) :
            sys.stdout.write('Kalibracijski predlozak je automatski pronaden!\n')
            podaciUzorkaDetekcija = podaciUzorkaDetekcija[1]
            slikaSiva = cv2.cvtColor(slikaBezIzoblicenja, cv2.COLOR_BGR2GRAY)
            cv2.cornerSubPix(slikaSiva, podaciUzorkaDetekcija, (detekcijaPr,
            detekcijaPr), (-1, -1), (cv2.TERM_CRITERIA_MAX_ITER +
            cv2.TERM_CRITERIA_EPS, 30, 0.1))
            vrhoviUzorkaDetekcija = [
            [podaciUzorkaDetekcija[0][0][0], podaciUzorkaDetekcija[0][0][1]],
            [podaciUzorkaDetekcija[brojUnutarnjihKutovaDetekcija[0] - 1][0][0],
            [podaciUzorkaDetekcija[brojUnutarnjihKutovaDetekcija[0] - 1][0][1]],
            [podaciUzorkaDetekcija[brojUnutarnjihKutovaDetekcija[0] *
            brojUnutarnjihKutovaDetekcija[1] - 1][0][0],
            podaciUzorkaDetekcija[brojUnutarnjihKutovaDetekcija[0] *
            brojUnutarnjihKutovaDetekcija[1] - 1][0][1]],
            [podaciUzorkaDetekcija[(brojUnutarnjihKutovaDetekcija[1] -
            1)*brojUnutarnjihKutovaDetekcija[0]][0][0],
            podaciUzorkaDetekcija[(brojUnutarnjihKutovaDetekcija[1] -
            1)*brojUnutarnjihKutovaDetekcija[0]][0][1]]
            ]
```

Metoda najprije poziva metodu *makniIzoblicenje* pomoću koje se mogu, uz pomoć matrice kamere i matrice koeficijenata izobličenja, ispraviti izobličenje bilo koje točke slike. U ovom slučaju ispravlja se izobličenje svih točaka slike kako bi se ispravno

detektirale značajke kalibracijskog uzorka (unutarnji kutovi šahovnice). Pomoću *OpenCV* metode *findChessboardCorners* pokreće se automatska detekcija značajki za određene dimenzije unutarnjih kutova. Ukoliko se uzorak automatski detektira, uz pomoć *OpenCV* metode *cornerSubPix* traži se podpikslna preciznost svakog položaja značajke. Podaci se spremaju u listu *vrhoviUzorkaDetekcija*. U slučaju da uzorak nije automatski detektiran metoda smanjuje tražene dimenzije odnosno broj unutarnjih kutova i pokreće ponovno postupak. Ako se konačno nije detektirao uzorak, metoda 'obavještava' glavni program nakon čega se pokreće metoda *odabirTocaka* pomoću koje korisnik ručno odabere značajke.

Metoda *odabirTocaka* se osim za odabir značajki koristi i za odabir točaka slike između kojih se želi izračunati međusobna udaljenost ili površina. Također služi i za moguću naknadnu korekciju položaja. Ona poziva metodu *prikaziUvecano* pomoću koje se u posebnom prozoru otvara dio slike oko odabrane točke slike povećati omogućavajući precizniji odabir točke.

Homografija i preslikavanje odabranih točaka računa se pomoću metode *transformiraj*. Prikazan je dio te metode

```
pomakPiksela = 2500
brojPikselaKvadrat = 20

vrhoviUzorkaKS = podaciUzorka[:]
vrhoviUzorkaKS[0] = [pomakPiksela, pomakPiksela]
vrhoviUzorkaKS[1] = [pomakPiksela + (brojUnutarnjihKutova[0] -
1)*brojPikselaKvadrat, pomakPiksela]
vrhoviUzorkaKS[2] = [pomakPiksela + (brojUnutarnjihKutova[0] -
1)*brojPikselaKvadrat, pomakPiksela + (brojUnutarnjihKutova[1] -
1)*brojPikselaKvadrat]
vrhoviUzorkaKS[3] = [pomakPiksela, pomakPiksela + (brojUnutarnjihKutova[1]
- 1)*brojPikselaKvadrat]
vrhoviUzorkaMatrica = np.array(podaciUzorka, dtype = np.float32)
vrhoviUzorkaKSMatrica = np.array(vrhoviUzorkaKS, dtype = np.float32)
transfMatrica = cv2.findHomography(vrhoviUzorkaKSMatrica,
vrhoviUzorkaMatrica)[0]
transfMatrica = np.linalg.inv(transfMatrica)
listaKoordinataMatrica = np.array(listaKoordinata, dtype = np.float32)
listaKoordinataMatrica = np.array([listaKoordinataMatrica])
podaciUzorkaMatrica = np.array(podaciUzorka, dtype = np.float32)
podaciUzorkaMatrica = np.array([podaciUzorkaMatrica])
listaKoordinataTransf = cv2.perspectiveTransform(listaKoordinataMatrica,
transfMatrica)
podaciUzorkaTransf = cv2.perspectiveTransform(podaciUzorkaMatrica,
transfMatrica)
```

Vrijednosti varijabli *pomakPiksela* i *brojPikselaKvadrat* ne utječe na rezultat jer je matrica homografije homogena. Koriste se kako bi se prikazala pregledna transformirana slika (prikazan primjer u sljedećem poglavlju). Matrica *vrhoviUzorkaMatrica* predstavlja matricu točaka u koje se značajke kalibracijskog uzorka trebaju preslikati. Njene vrijednosti su određene s prethodne dvije varijable. U oblik matrice su pretvorene uz pomoć *NumPy* metode *array*. Matrica homografije dobiva se uz pomoć *OpenCV* metode *findHomography*. Ta se matrica koristi za transformaciju odabranih točaka u ravninu ceste uz pomoć *OpenCV* metode *perspectiveTransform*.

Navedeni su neki od najbitnijih isječaka izvornog programskog koda. Potpuni izvorni nalazi se u prilogu rada.

6. Eksperimentalni rezultati i problemi

U prethodnim poglavljima opisana je teorijska podloga ovog rada i kako je ona programski implementirana. Ovo poglavlje opisuje korištene uređaje, eksperimentalne rezultate odnosno njihovu preciznost i uočene probleme.

6.1. Korištena kamera

U ovom radu koristila se kamera GoPro Hero3+ BE. Kamere proizvođača GoPro poznate su kao minijaturene kamere koje se zbog svojih dimenzija često koriste u sportskim aktivnostima gdje se obično stavljaju na kacigu vozača ili na vozilo. Takve kamere omogućavaju široko vidno polje (do čak 170°) čime se omogućava snimanje više sadržaja u istom kadru, ali i uzrokuje lako vidljivo radijalno izobličenje slike.



Slika 7. – Korištena GoPro Hero3+ BE kamera. Odlična kvaliteta snimaka u odnosu na male dimenzije kamere. Jednostavna za korištenje s bitnom manom – nedostatkom ekranskog prikaza trenutnog kadra snimanja.

Kapacitet baterije ograničen je zbog dimenzija kamere. Kako bi kamera postigla autonomiju baterije od ukupno deklariranih sat i pol pri visokim postavkama, kamera ne sadrži ekranski prikaz trenutnog kadra snimanja (na slici se može vidjeti monokromatski ekran koji prikazuje osnovne informacije i preko kojeg se mogu mijenjati postavke).

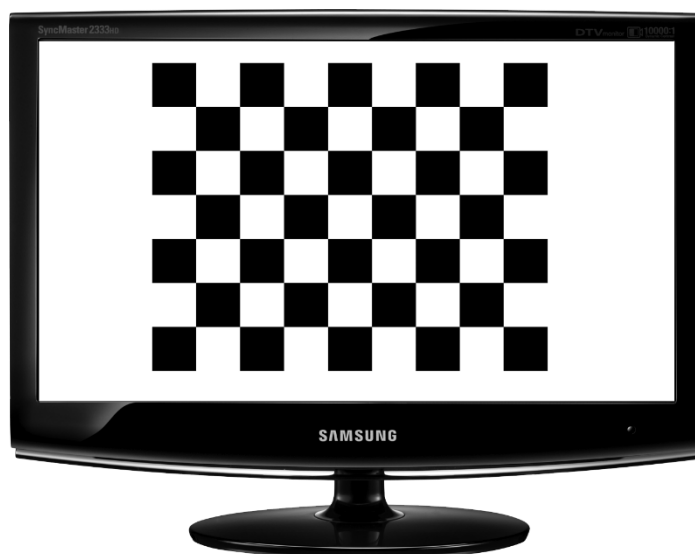
Zbog dimenzija, nedostaje i optičko tražilo. Navedeno je stvaralo problem pri kadriranju kalibracijskog uzorka. U svrhu kalibracije kamere snimalo se iz više kutova i udaljenosti od kalibracijskog uzorka. Kamerom se snimalo u rezoluciji od 1920×1080 piksela pri 50 sličica u sekundi (eng. *frames per second* - FPS).

Najveća rezolucija koja kamera nudi je 3840×2160 no u tom slučaju snimanje je ograničeno na 15 sličica u sekundi. Za potrebe ovog rada broj sličica u sekundi nije važan no potrebno je paziti na upotrebljivost snimaka u neku drugu svrhu. Podsjetimo, rezolucija snimanja prilikom kalibracije i rezolucija snimke moraju biti iste kako bi se ispravno maknulo izobličenje.

Kut vidnog polja kamere (eng. *field of view* - FOV) može se postaviti na 'ultra širok' (170°), 'širok' (127°) i 'uzak' (90°). Manjim kutom snimanja postiže se manje radijalnog izobličenje slike no poželjno je imati što više sadržaja u kadru zbog čega se kamerom snimalo u 'ultra širokom' načinu snimanja.

6.2. Korišten kalibracijski uzorak

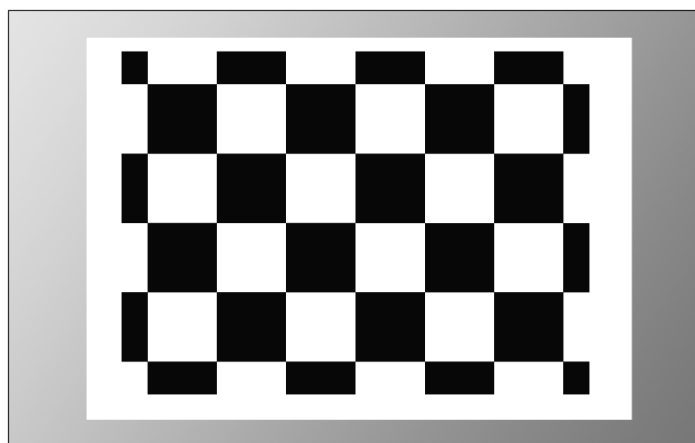
Kalibracijski uzorak koji se koristio u svrhu kalibracije kamere bila je slika šahovnice dimenzija 9×7 crno-bijelih kvadrata prikazana na monitoru Samsung 2333HD dijagonale 58.42 cm (23 inča).



Slika 6. – Korišteni monitor Samsung 2333HD s prikazanim kalibracijskim uzorkom dimenzija 9×6 kvadrata.

Duljina stranice kvadrata izmjerena je u pikselima (130 px). Provjerom specifikacija monitora saznala se duljina piksela na monitoru u milimetrima (0.265 mm). Umnoškom te dvije vrijednosti dobila se stvarna duljina stranice kvadrata izražena u milimetrima ($130 \text{ px} \times 0.265 \text{ mm/px} = 34.45 \text{ mm}$). Uzorak sadrži 8 unutarnjih kutova u jednom retku i 6 unutarnjih kutova u jednom stupcu što čini ukupno 48 unutarnjih kutova.

Kalibracijski uzorak korišten u svrhu izračuna homografije bila je slika šahovnice dimenzija 6×4 crno-bijelih kvadrata s odsječcima kvadrata u rubovima. Slika je bila ispisana na dva foto-papira A4 formata⁴ koji su potom bili zalijepljeni na tvrdu prozirnu plastičnu podlogu debljine 4 mm. Duljina stranice kvadrata iznosila je 6 cm.



Slika 7. – Korišteni predložak s kalibracijskim uzorkom dimenzija 6×4 kvadrata s odsječcima kvadrata u rubovima.

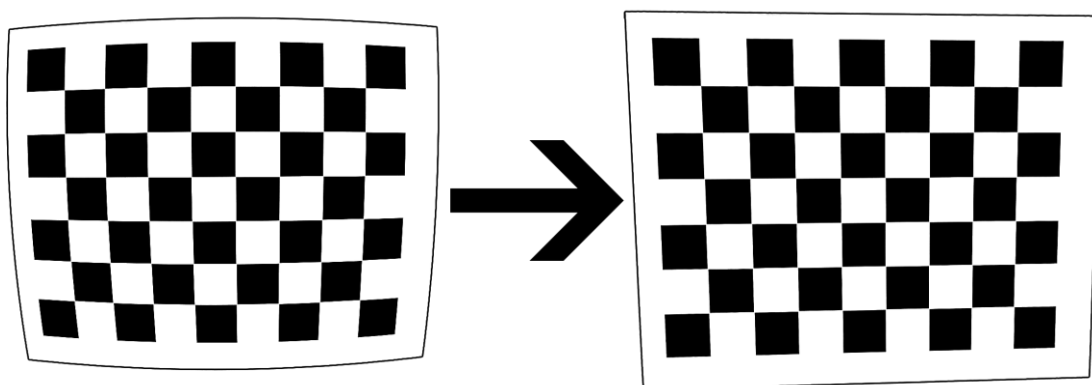
⁴ Dimenzije A4 formata prema ISO standardu iznose 210 mm $\times$ 297 mm.

6.3. Eksperimentalni rezultati

U 3. poglavlju opisan je postupak kalibracije kamere. Napomenula su se dva načina snimanja, jedan u kojem je uzorak miran, položaj kamere se mijenja i drugi u kojem je situacija obrnuta. U ovom radu snimao se nepomičan monitor na kojem je bila prikazana slika kalibracijskog uzorka (odjeljak 6.2.). Kamerom se snimao kalibracijski uzorak iz više kutova i udaljenosti nakon čega se iz snimke izvadilo ukupno 27 slika. Programskom obradom tih slika dobili su se intrinzični parametri kamere (vrijednosti su, zbog preglednosti, zaokružene na 3 decimale):

$$K = \begin{bmatrix} 864.124 & 0 & 946.841 \\ 0 & 881.801 & 540.967 \\ 0 & 0 & 1 \end{bmatrix} \quad D = \begin{bmatrix} -2.778 \cdot 10^{-1} \\ 1.263 \cdot 10^{-1} \\ 6.454 \cdot 10^{-4} \\ -3.273 \cdot 10^{-5} \\ -3.548 \cdot 10^{-2} \end{bmatrix}$$

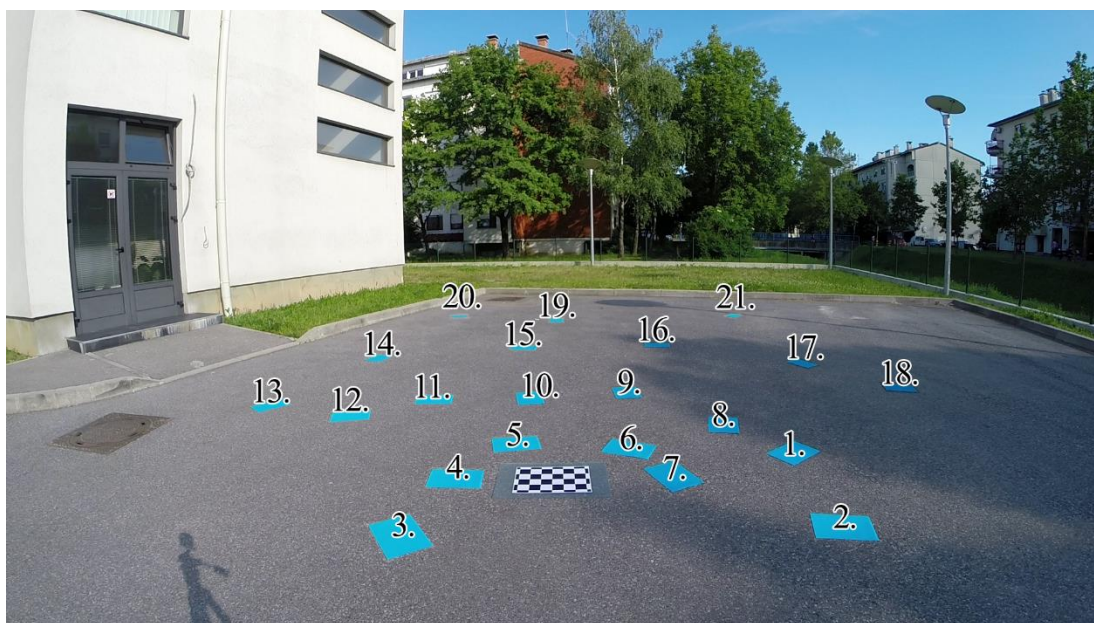
Uz pomoć dobivenih intrinzičnih parametara kamere omogućeno je ispravljanje izobličenja položaja točaka na slici. *Slika 8.* prikazuje primjer u kojem se svakoj točki slike ispravio položaj čime se ispravilo izobličenje slike.



Slika 8. – primjer ispravljanja izobličenja slike. Linije nisu pod pravim kutom jer se kalibracijski uzorak snimao pod nepravim kutom no može se primijetiti jasna razlika u pravilnosti kvadrata prije i poslije ispravljanja izobličenja slike.

Ispravnost odnosno preciznost izračuna udaljenosti u ravni ceste ispitivala se uz pomoć objekta poznatih dimenzija koji je bio postavljen u različite položaje. U ovom primjeru to je bio papir tirkizno plave boje A4 formata.

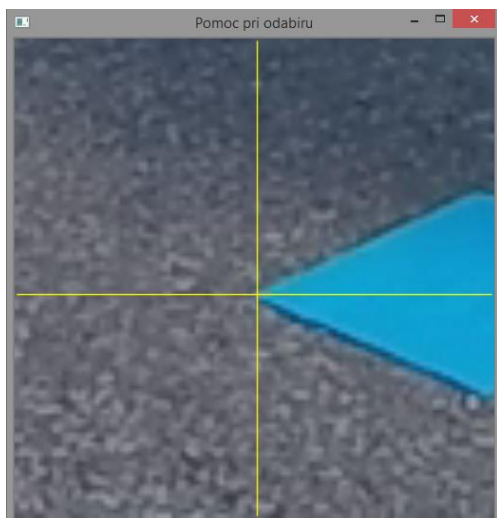
Objekt se postavio u ukupno 21 različitih položaja. Kalibracijski uzorak se nalazio na otprilike 240 cm udaljenosti od podnožja stativa visokog 120 cm na kojem je bila kamera. *Slika 9.* prikazuje kombiniranu sliku svih položaja objekta.



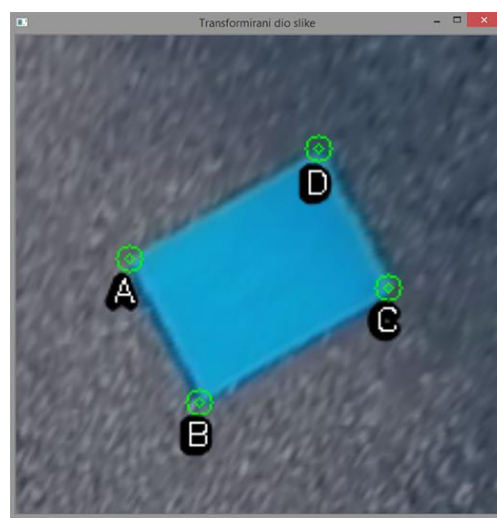
Slika 9. – kombinirana slika sastavljena od ukupno 21 slika od kojih svaka sadrži objekt u različitom položaju.

Odabir točaka vrši se lijevom tipkom miša uz pomoć grafičkog sučelja programa. Odabirom bilo koje točke slike otvara se novi prozor u kojem se nalazi uvećani dio slike s odabranom točkom u centru. Pritiskom tipke + i – dio slike se može uvećavati i smanjivati. Kao pomoć na slici nalaze se dvije okomite crte koje se mogu pomicati pomoću standardnih tipaka za kretanje. Njihov presjek čini položaj odabrane točke. Pritiskom tipke *Enter* potvrđuje se odabir točke. Za odabir proizvoljnog broja točaka postupak se ponavlja. *Slika 10. a)* prikazuje primjer odabira točke na slici.

Nakon što se konačno odaberu sve željene točke, koje npr. predstavljaju vrhove objekta, pritiskom tipke *Enter* pokreće se postupak ispravljanja izobličenja točaka nakon čega se uz pomoć transformacijske matrice točke preslikaju u ravninu ceste. Dio slike ravnine ceste, relevantan za odabrane točke, prikazuje se u novom prozoru koji se također može povećavati i smanjivati. Time se može vidjeti koliko se precizno odabrao vrh. *Slika 10. b)* prikazuje preslikane odabrane točke.



a)



b)

Slika 10. – primjer prozora za odabir točke a) i prozora za pregled preslikavanja odabranih točaka b).

U komandnoj liniji programa ispisuju se međusobne udaljenosti između odabranih točaka i površina lika čiji su vrhovi odabrane točke. Dobivene vrijednosti uspoređuju se sa stvarnim vrijednostima. U ovom primjeru (*Položaj 1.* u *Slici 9.*) odabrane točke vrhovi su papira A4 formata. Dužina papira iznosi 29.7 cm, a visina 21 cm.

```
Izmjerena površina objekta : 615.313363234 cm ^ 2
Stvarna površina objekta : 29.7 cm * 21.0 cm = 623.7 cm ^ 2
Razlika površina objekta: -8.386636766 cm^2
Pogreska izracuna površine : -1.34465877284 %
(tocnost : 98.6553412272 %)
Stvarne udaljenosti između oznacenih vrhova :
Duzina = 29.7 cm
Visina = 21.0 cm
Izmjerene udaljenosti između oznacenih vrhova :
  | AB | = 21.307 cm    V Razlika : 0.307 cm(tocnost : 98.538 %)
  | AC | = 34.85 cm
  | AD | = 29.152 cm    D Razlika : -0.548 cm(tocnost : 98.155 %)
  | BC | = 29.483 cm    D Razlika : -0.217 cm(tocnost : 99.269 %)
  | BD | = 37.281 cm
  | CD | = 20.778 cm    V Razlika : -0.222 cm(tocnost : 98.943 %)
```

Sljedeća tablica prikazuje izračune za sve korištene položaje i njihove pogreške. Vrhovi objekta su odabrani redoslijedom prikazanim na *Slici 10. b)*. Sve prikazane vrijednosti zaokružene su na dvije decimale zbog preglednosti. Mjerna jedinica za udaljenost je centimetar [cm], a za površinu kvadratni centimetar [cm²].

	AB	AD	BC	CD	$\Delta AB $	$\Delta AD $	$\Delta BC $	$\Delta CD $	P	ΔP
R	21.00	29.7	29.7	21.0	0.00	0.00	0.00	0.00	623.7	0.00
1.	21.31	29.15	29.48	37.28	0.31	-0.55	-0.22	-0.22	615.31	-8.39
2.	20.89	28.99	28.90	20.75	-0.11	-0.71	-0.80	-0.25	602.54	-21.16
3.	20.49	29.71	30.04	20.46	-0.51	0.01	0.34	-0.54	611.78	-11.92
4.	21.09	29.51	29.63	20.96	0.09	-0.19	-0.07	-0.04	620.1	-3.6
5.	21.02	29.31	29.45	20.90	0.02	-0.39	-0.25	-0.10	615.58	-8.12
6.	21.33	29.41	29.39	21.30	0.33	-0.29	-0.31	0.30	625.99	2.29
7.	20.74	29.60	30.20	20.77	-0.26	-0.10	0.50	-0.23	620.00	-3.70
8.	20.46	29.49	29.36	20.76	-0.54	-0.21	-0.34	-0.24	606.23	-17.47
9.	20.17	30.05	30.0	20.84	-0.83	0.35	0.30	-0.16	615.59	-8.11
10.	20.41	29.99	30.09	20.9	-0.59	0.29	0.39	-0.10	617.3	-6.4
11.	20.63	29.85	30.00	21.48	-0.37	0.15	0.30	0.48	628.66	4.96
12.	20.75	29.99	29.52	20.99	-0.25	0.29	-0.18	-0.01	620.69	-3.01
13.	22.20	29.54	29.18	21.22	1.20	-0.16	-0.52	0.22	635.82	12.12
14.	21.01	30.06	27.24	22.57	0.01	0.36	-2.46	1.57	601.27	-22.43
15.	20.48	30.20	30.05	20.24	-0.52	0.50	0.35	-0.76	605.68	-18.02
16.	21.96	29.36	29.23	22.51	0.96	-0.34	-0.47	1.51	651.13	27.43
17.	21.01	29.79	30.2	20.14	0.01	0.09	0.50	-0.86	607.33	-16.37
18.	21.05	28.11	28.17	20.10	0.05	-1.59	-1.53	-0.90	577.88	-45.82
19.	20.27	32.33	28.93	22.27	-0.73	2.63	-0.77	1.27	648.27	24.57
20.	25.11	31.34	30.78	23.26	4.11	1.64	1.08	2.26	684.55	60.85
21.	19.08	29.68	28.44	23.31	-1.92	-0.02	-1.26	2.31	565.31	-58.39
Σ	21.02	29.78	29.44	22.05	0.02	0.08	-0.26	1.05	617.95	-5.75

R – referentna udaljenost (točne dimenzije objekta).

1. – položaj objekta na slici (*Slika 9.*).

|AB| / P- udaljenost između dva vrha objekta / izmjerena površina.

$\Delta|AD|$ / ΔP - pogreška izmjerene udaljenosti između dva vrha objekta / izmjerene površine.

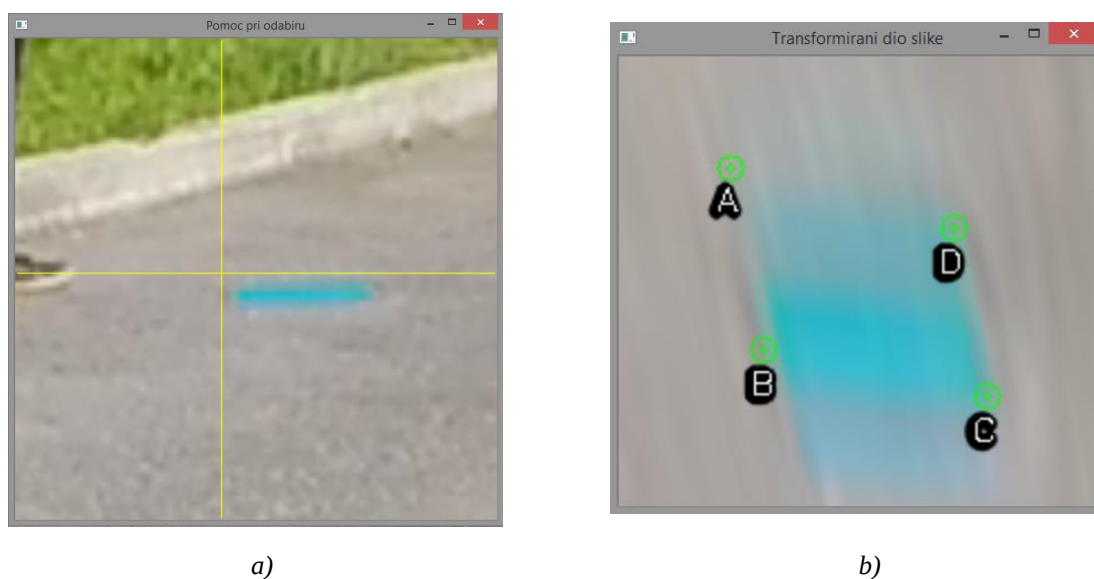
Σ - prosječne izmjerene udaljenosti i pogreške.

6.3. Eksperimentalni problemi

Tablica u prethodnom odjeljku prikazuje eksperimentalne rezultate. Ovisno o položaju objekta u ravnini ceste, razlikuje se preciznost izmjerenih dimenzija objekta. Prosječne vrijednosti pokazuju zadovoljavajuću preciznost izračuna. Podebljanim granicama ćelije tablice naglašene su vrijednosti pogrešaka koje su najviše odstupale od drugih i broj položaja kojima pripadaju.

Pomoću *Slike 9.* mogu se vidjeti 'sporni' položaji. Uočava se da najgore rezultate postižu položaji objekta kojima je udaljenost od kamere bila relativno velika. Pomnijim promatranjem slike i tablice može se zaključiti da najpreciznije rezultate postižu položaji koji su blizu kamere i kalibracijskog uzorka.

Moderne kamere, pa tako i kamera koja se koristila u eksperimentu, koriste funkciju automatskog fokusiranja (eng. *Autofocus*) čime se korisniku olakšava snimanje, ali uzrokuje i određene probleme vezane uz ovaj rad. Najprije, kalibracija kamere se otežava. Potrebno je održati stalnu fokalnu duljinu f kako bi se kalibracija uspješno izvršila. Zatim, konkretno u primjeru, otežava se precizan odabir vrhova objekta kojem se žele izmjeriti dimenzije i koji se nalazi relativno daleko od kamere. Primjer je odabir vrhova objekta u *Položaju 20.* što prikazuje *Slika 11. a)*. Pregled odabranih vrhova na slici ravnine ceste je također vrlo otežan što prikazuje *Slika 11. b)*.



Slika 11. – primjer otežanog odabira točaka vrhova objekta i njihova transformacija u ravninu ceste.

7. Zaključak

U ovom radu kroz poglavlja 2. Model stvaranja slike, 3. Kalibracija kamere i 4. Određivanje udaljenosti u ravnini ceste opisana je teorija zajedno s postupcima koji su bili potrebni kako bi se izračunala udaljenost između točaka u ravnini ceste. Poglavlje 5. Programska podrška i implementacija opisalo je implementaciju navedenih postupaka. Kolika se postiže preciznost izmjerenih udaljenosti točaka u različitim položajima prikazano je u poglavlju 6. Eksperimentalni rezultati i problemi.

Računalni vid zahtijeva razumijevanje relativno mnogo teorije usko vezane uz višu matematiku no zato zauzvrat nudi mnogo praktičnih metoda koji se, kao u ovom radu, mogu koristiti. Objašnjeno je kroz ovaj rad kako relativno jednostavno samo uz pomoć kamere i kalibracijskog uzorka, koji se čak ne mora nalaziti u svakoj slici snimke, ako se položaj kamere ne mijenja, precizno odrediti međusobnu udaljenost između bilo kojih snimanih točaka u ravnini ceste.

Praktična primjena ovog postupka je određivanje dimenzija objekata koji se nalaze na cesti ili u ravnini ceste. Koristi li se kalibrirana kamera koja ima stalan položaj i poznaje li se transformacijska matrica moguće je iz bilo koje snimke kamere izvaditi bilo koju sliku i na njoj odrediti udaljenosti u ravnini ceste. Primjer je određivanje površine zakrpe ceste ili dijela kojem je potrebna zakrpa. Zakrpa zahtijeva određenu površinu asfalta. Postupak mjerenja obično se izvršava neprecizno jer se promet ne može zaustaviti za mjerenje svake zakrpe/rupe na cesti pa se vrši procjena. Tu je također mogući faktor namjerno krive i skuplje procjene zbog vlastite koristi. Postupkom navedenim u ovom radu može se snimiti cesta i naknadno vrlo precizno odrediti površina svake zakrpe/rupe. Time se lako dobiva ukupno potrebna količina asfalta čime se eliminiraju navedeni problemi. Trošak postupka je samo inicijalan zbog nabave kamere i kalibracijskog uzorka, a i u tom slučaju je relativno nizak pogotovo ako se uzme u obzir ušteda koja se ostvaruje.

Postupak se može olakšati ako se koristi stereoskopska kamera koja ne zahtijeva snimanje kalibracijskog uzorka u ravnini ceste. Time se postupak još više pojednostavljuje no početni trošak je veći.

8. Literatura

- [1] R. I. Hartley i A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, 2004.
- [2] A. Majumber. CS211A: Visual Computing (predavanja, URL <http://www.ics.uci.edu/~majumder/VC/vc.htm>). University of California, Irvine
- [3] Z. Zhang. A Flexible New Technique for Camera Calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000.
- [4] S. Banerjee. Projective geometry, camera models and calibration (predavanja, URL <http://www.cse.iitd.ernet.in/~suban/vision/geometry/>). Indian Institute of Technology Delhi, New Delhi
- [5] Python dokumentacija. URL <https://www.python.org/doc/>
- [6] OpenCV dokumentacija. URL <http://docs.opencv.org/modules/refman.html>
- [7] NumPy dokumentacija. URL <http://docs.scipy.org/doc/>

Mjerenje udaljenosti u ravnini ceste kamerom postavljenom na vozilu

Sažetak

Ovim radom opisani su postupci dobivanja modela stvaranja slike, kalibracije kamere i preslikavanja točaka slike u ravninu ceste. Tim postupcima ostvarilo se mjerenje udaljenosti u ravnini ceste uz pomoć slike dobivene kamerom postavljenom na vozilu. Za mjerenje potrebna je relativno jeftina kamera i tanka podloga s kalibracijskim uzorkom. Postupak kalibracije kamere implementiran je uz pomoć C++-a, a ravninsko preslikavanje uz pomoć Pythona. Metode računalnog vida, potrebne za ovaj rad, implementirane su uz pomoć *OpenCV* biblioteke. Korišteni *NumPy* paket omogućio je stvaranje i korištenje matrica. Eksperimentalni rezultati pokazali su relativno vrlo precizno mjerenje udaljenosti u ravnini ceste korištenjem opisanih postupaka. Veće pogreške mjerenja uočile su se kod točaka u ravnini ceste koje su se nalazile na relativno velikoj udaljenosti od kamere.

Ključne riječi: računalni vid, model stvaranja slike, kamera, kalibracija, uzorak, homografija, ravninsko preslikavanje, mjerenje, udaljenost u cesti, C++, Python, OpenCV, NumPy

Performing measurements in the ground plane by a camera mounted on a vehicle

Abstract

This work explains obtaining image creation model, camera calibration and perspective mapping of image points in the ground plane. By that, measurements in the ground plane can be made using images created with camera mounted on a vehicle. Measurements require inexpensive camera and thin base with calibration pattern. Camera calibration procedure was implemented in C++ and planar projection was implemented in Python. Computer vision methods, needed for this work, were implemented using *OpenCV* library. Imported *NumPy* package allowed creation and usage of matrices. Experimental results showed measurements in the ground plane with relatively high precision using procedures described in this work. Larger errors were noticed when measurements were performed between points in the ground plane located relatively far from camera.

Keywords: computer vision, image creation model, camera, calibration, pattern, homography, planar projection, measurements, measurements in the ground plane, C++, Python, OpenCV, NumPy

PRIVITAK

Uz rad je priložen DVD-ROM na kojem se nalaze izvorni kod programskog rješenja rada, *OpenCV* biblioteka, *NumPy* paket, instalacija *Pythona* i testni primjeri. U datoteci *Intstalacija.txt* opisani su postupci instalacije i upute korištenja programa.