

# Model prostornog odnosa i interakcije industrijske kamere i linijskog lasera

---

Šijak, Igor

Undergraduate thesis / Završni rad

2023

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:268399>

*Rights / Prava:* [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

*Download date / Datum preuzimanja:* **2024-07-22**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

**Igor Šijak**

Zagreb, 2023.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

Mentori:

Doc. dr. sc. Marko Švaco, mag.ing.mech.

Doc. dr. sc. Filip Šuligoj, mag.ing.mech.

Student:

Igor Šijak

Zagreb, 2023.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se doc. dr. sc. Marku Švaci na zadavanju teme završnog rada i mentoru doc. dr. sc. Filipu Šuligoju na stručnim savjetima te pruženoj pomoći tijekom izrade ovog rada.

Zahvaljujem se obitelji na svojoj podršci i strpljenju te prijateljima i kolegama na uzajamnoj pomoći i motivaciji tijekom studiranja.

Igor Šijak



SVEUČILIŠTE U ZAGREBU  
**FAKULTET STROJARSTVA I BRODOGRADNJE**



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za završne i diplomske ispite studija strojarstva za smjerove:  
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo  
materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 – 04 / 23 – 6 / 1	
Ur.broj: 15 - 1703 - 23 -	

## ZAVRŠNI ZADATAK

Student: **Igor Šijak** JMBAG: **0035225167**

Naslov rada na hrvatskom jeziku: **Model prostornog odnosa i interakcije industrijske kamere i linijskog lasera**

Naslov rada na engleskom jeziku: **A model of the spatial relationship and interaction of an industrial camera and a line laser**

Opis zadatka:

Danas postoji više načina i tehnologija za dovođenje objekta iz stvarnog svijeta u digitalni 3D prostor. Lasersko skeniranje je proces snimanja preciznih, trodimenzionalnih informacija stvarnog objekta, grupe objekata ili okoline, korištenjem lasera kao izvora svjetlosti. Projiciranjem laserskog svjetla na objekt i snimanjem istog s kamerom, vizijski sustav stvara oblake izmjerenih 3D točaka koje definiraju položaj objekta u prostoru. Zadatak završnog rada je izraditi matematičko-prostorni model odnosa kamere s objektivom i linijskog lasera. Razvijeni model namijenjen je brzom testiranju parametara vezanih uz specifične karakteristike individualnih komponenti u odnosu na željeni radni prostor takvog vizijskog sustava. Prema tome u sklopu rješenja potrebno je:

- Model programirati u Python pomoću knjižnice za linearnu algebru (matrice homogene transformacije) i vizualizacijskog modula,
- izraditi program koji omogućava inicijalni unos parametara poput prostorne pozicije i orijentacije kamere i lasera te karakteristika kamere (rezolucija, veličina senzora), objektiva (žarišna dužina) i lasera (kut širenja snopa svjetlosti),
- omogućiti vizualizaciju svih komponenata, presjeke njihovih radnih područja kao i sliku koju kamera vidi,
- izradeni model eksperimentalno testirati pomoću sustava dostupnog u sklopu laboratorija.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

30. 11. 2022.

Datum predaje rada:

1. rok: 20. 2. 2023.  
2. rok (izvanredni): 10. 7. 2023.  
3. rok: 18. 9. 2023.

Predvideni datumi obrane:

1. rok: 27. 2. – 3. 3. 2023.  
2. rok (izvanredni): 14. 7. 2023.  
3. rok: 25. 9. – 29. 9. 2023.

Zadatak zadao:

Doc. dr. sc. Marko Švaco

Dr. sc. Filip Šuligoj

Predsjednik Povjerenstva:

Prof. dr. sc. Branko Bauer

**SADRŽAJ**

SADRŽAJ .....	I
POPIS SLIKA .....	III
POPIS TABLICA.....	V
POPIS OZNAKA .....	VI
SAŽETAK.....	VIII
SUMMARY .....	IX
1. UVOD .....	1
2. PINHOLE MODEL KAMERE.....	4
2.1. Matematički model <i>pinhole</i> kamere .....	5
2.2 Matematički model <i>pinhole</i> kamere koristeći diskretnu ravninu slike .....	9
2.3. Matrica rotacije.....	11
2.4. Matematički model kamere s objektivom .....	12
3. MATEMATIČKE FUNKCIJE POTREBNE ZA VIZUALIZACIJU KOMPONENATA.....	15
3.1. Plückerova matrica .....	15
3.2. Jednadžba ravnine u matričnom obliku.....	15
3.3. Jednadžba pravca u prostoru .....	16
4. MATEMATIČKI MODEL LINIJSKOG LASERA .....	17
5. PROGRAM .....	19
5.1. Početni zaslon i zaslon s uputstvima .....	20
5.2. Unos podataka .....	22
5.3. Vizualizacija.....	24
5.4. <i>Camera matrix</i> kartica.....	25
5.5. Najbitniji dijelovi programskog koda .....	27
5.5.1. Kalibracijska matrica i matrica projekcije.....	27

---

5.5.2. Matrica rotacije.....	28
5.5.3. Plückerova matrica i matrica ravnine.....	29
5.5.4. Jednadžba za generiranje seta laserskih točaka.....	30
5.5.5. Jednadžbe za računanje centralne projekcije.....	30
6. PRAKTIČNA ANALIZA IMPLEMENTIRANOG RJEŠENJA.....	33
6.1. Kalibracija kamere .....	35
6.2. Usporedba realnog sustava i programa .....	39
7. ZAKLJUČAK .....	44
LITERATURA.....	45
PRILOG .....	46

## POPIS SLIKA

Slika 1.	Optički 3D skener ATOS Core .....	1
Slika 2.	Laserski skener FARO Focus S 150 .....	2
Slika 3.	<i>Pinhole</i> model kamere [5].....	4
Slika 4	Efekt različitih veličina otvora blende na sliku [5] .....	5
Slika 5.	Matematičko-prostorni model <i>pinhole</i> kamere [6] .....	6
Slika 6.	Ravnina slike [7] .....	7
Slika 7.	Translacija i rotacija koordinatnog sustava kamere u odnosu na prostorni koordinatni sustav [7]	8
Slika 8.	Centralna projekcija na diskretnu ravninu slike [6] .....	10
Slika 9.	Eulerovi kutovi i njihove pripadne koordinatne osi [8] .....	11
Slika 10.	Bačvasta i jastučasta distorzija.....	13
Slika 11.	Tangencijalna distorzija [9] .....	13
Slika 12.	Pravac u prostoru [10].....	16
Slika 13.	Prostorni model linijskog lasera.....	18
Slika 14.	Početni zaslon programa .....	20
Slika 15.	Zaslon s uputstvima .....	21
Slika 16.	Set okvira za unos podataka.....	22
Slika 17.	Vizualizacija kamera-laser sustava .....	24
Slika 18.	<i>Camera matrix</i> kartica .....	25
Slika 19.	.txt datoteka kalibracijske matrice kamere.....	26
Slika 20.	Slika kamere koristeći kalibracijsku matricu iz .txt datoteke .....	27
Slika 21.	Kamera-laser sustav DRASER .....	33
Slika 22.	Nacrt kamera-laser sustava DRASER.....	34
Slika 23.	Šahovnica za kalibraciju kamere.....	35
Slika 24.	Primjer snimljenih slika šahovnice za kalibraciju .....	36
Slika 25.	Distorzijski parametri i kalibracijska matrica dobiveni kalibracijom .....	38
Slika 26.	Iznos idealne kalibracijske matrice .....	39
Slika 27.	.txt datoteka kalibracijom dobivenih distorzijskih parametara .....	39
Slika 28.	Okvir sa unesenim parametrima .....	40



Slika 29. Grafovi generirani na temelju idealnih parametara kamera-laser sustava .....41

## POPIS TABLICA

Tablica 1.	Tehničke karakteristike kamere potrebne za unos u program .....	34
Tablica 2.	Usporedba rezultata programa i realnog sustava.....	42

## POPIS OZNAKA

Oznaka	Jedinica	Opis
$\Theta_x, \Theta_y, \Theta_z$	°	Eulerovi kutovi, kutovi valjanja, posrtanja i skretanja
$\pi$	/	Jednadžba ravnine u prostoru u matricnom zapisu
$\rho_w, \rho_h$	mm	Širina i visina piksela
$\varphi$	°	Kut širenja snopa laserske svjetlosti
$C$	/	Ishodište koordinatnog sustava kamere
$dy_1, dy_2$	/	Jedinični vektori $\vec{j}$ zarotirani za kut $\frac{\varphi}{2}$ , odnosno $-\frac{\varphi}{2}$ za $dy_2$
$f$	mm	Žarišna duljina kamere
$I$	/	Jedinična matrica
$K$	/	Kalibracijska matrica kamere ili matrica kamere
$k_1, k_2, k_3$	/	Koeficijenti radijalne distorzije
$l$	mm	Udaljenost ishodišta lasera od ravne površine
$L$	/	Plückerova matrica
$L_0$	/	Ishodište koordinatnog sustava lasera
$M_p$	/	Matrica projekcije
$n$	/	Broj točaka na kraju laserske linije
$p$	/	Točka na 2D slici
$P$	/	Točka u prostoru
$\tilde{p}$	/	Homogena točka slike
$\tilde{P}$	/	Homogena točka u prostoru
$p_x, p_y$	/	Koordinate glavne točke u x, y ravnini
$P_{cam}$	/	Točka u prostornom koordinatnom sustavu kamere
$p_1, p_2$	/	Koeficijenti tangencijalne distorzije
$p$	/	Pravac u prostoru
$R$	/	Matrica rotacije
$R_x, R_y, R_z$	/	Matrice valjanja, posrtanja i skretanja
$r$	/	Oznaka za vektorsku jednadžbu pravca u prostoru
$r$	/	Udaljenost točke na slici od centra zakrivljenosti
$T_1, T_2$	/	Točke u prostoru, krajnje točke laserske linije
$u, v$	/	Koordinate točke u pikselima
$u_0, v_0$	/	Koordinate glavne točke u pikselima
$u_n, v_n$	/	Normalizirane piksel koordinate
$u_d, v_d$	/	Distorzijske normalizirane piksel koordinate
$u', v'$	/	Distorzijske koordinate

$\tilde{u}, \tilde{v}, \tilde{w}$	/	Homogene koordinate u pikselima
$x, y$	/	Koordinate točke u ravnini
$\tilde{x}, \tilde{y}, \tilde{z}$	/	Homogene koordinate točke slike
$X, Y, Z$	/	Koordinate točke u prostoru

## SAŽETAK

U ovom završnom radu razrađen je model prostornog odnosa i interakcije industrijske kamere i linijskog lasera. Napravljen je pregled teorije potrebne za izradu programa koji će omogućiti vizualizaciju kamera-laser sustava te unos svih potrebnih parametara za dovođenje kamere i lasera u željeni odnos. Kod teorijskog pregleda razrađeni su matematički modeli kamere i lasera te su definirane matematičke funkcije koje će se koristiti za vizualizaciju navedenih modela i projekcije laserske linije na samu kameru. Najbitniji pojmovi vezani uz matematički model kamere koji su razrađeni u teorijskom dijelu rada su *pinhole* model kamere i centralna projekcija. U drugom dijelu rada je opisan izrađeni program, prikazano je njegovo korisničko sučelje, navedene su korištene biblioteke potrebne za funkcionalnost programa i objašnjeni su najbitniji dijelovi programskog koda. Program je pisan u *Python* programskom jeziku zajedno s njegovim grafičkim korisničkim sučeljem. Nadalje, radi testiranja funkcionalnosti programa i njegove validacije, napravljena je eksperimentalna provjera gdje su uspoređene slike snimljene pomoću kamere realnog kamera-laser sustava i slike generirane pomoću programa.

Ključne riječi: industrijska kamera, linijski laser, *pinhole* model kamere, centralna projekcija, *Python*, kamera-laser sustav

## **SUMMARY**

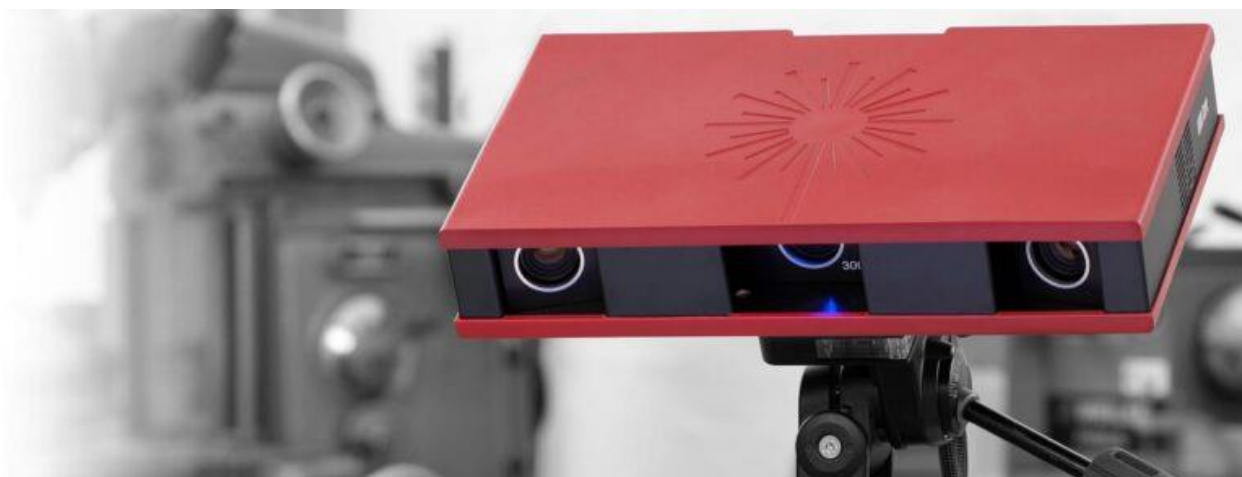
In this bachelor's thesis, a model of the spatial relationship and interaction between an industrial camera and a line laser is elaborated. An overview of the theory necessary for the development of a program that will enable visualization of the camera-laser system along with the input of all required parameters to establish the desired relationship between the camera and the laser is provided. In the theoretical review, mathematical models of the camera and laser are developed. Also, the mathematical functions for visualizing the mentioned models and the projected laser line onto the camera itself are defined. The most important concepts related to the mathematical camera model that are elaborated in the theoretical part of the paper are the pinhole camera model and central projection. The second part of the thesis describes the developed program, presents its user interface, lists the utilized libraries necessary for the program's functionality and explains the most crucial parts of the program code. The program is written in the Python programming language, along with its graphical user interface. Furthermore, to test the functionality of the program and to validate it, an experimental verification is conducted by comparing images captured by the camera of a real camera-laser system and images generated by the program.

Keywords: industrial camera, line laser, pinhole camera model, perspective projection, Python, camera-laser system

## 1. UVOD

Lasersko skeniranje je proces kojim se laserska svjetlost projicira na željeni objekt u prostoru i snima kamerom kako bi se stvorio digitalni oblak 3D točaka koje definiraju položaj tog objekta. Korištenjem laserskog skeniranja, inženjeri danas mogu dobiti precizne digitalne rekonstrukcije 3D objekata koje dalje mogu koristiti u različite svrhe. Prema [1] i [2] lasersko skeniranje se koristi u građevinarstvu, medicini (npr. skeniranje zubne šupljine u stomatologiji), povratnom inženjerstvu, strojarstvu (npr. mjerenje deformacija) i u brojnim drugim područjima.

Kao primjer na Slika 1 je prikazan industrijski optički 3D skener zvan ATOS tvrtke Topomatika. Uređaj se prvenstveno koristi za kontrolu kvalitete, povratno inženjerstvo, brzu izradu prototipova, brzo glodanje i brzu izradu alata te digitalno modeliranje i montažu. [3]



**Slika 1. Optički 3D skener ATOS Core**

Drugi primjer industrijskih laserskih skenera je Focus serija tvrtke FARO. Skeneri omogućuju detaljno skeniranje objekata i okoline, a koriste se u građevinskoj industriji (brzo i precizno dokumentiranje građevinskih projekata, uključujući snimanje zgrada, fasada i unutarnjih prostora), arheologiji (digitalizacija i dokumentiranje arheoloških ostataka), proizvodnji (provjera dimenzija i kvalitete proizvoda) i ostalim područjima. Na Slika 2 je prikazan jedan takav skener iz navedene serije. [4]



**Slika 2. Laserski skener FARO Focus S 150**

Kod projektiranja kamera-laser sustava bitno je uzeti u obzir brojne karakteristike individualnih komponenti kao što su kut širenja snopa svjetlosti lasera, žarišna duljina kamere, njena veličina senzora i rezolucija, međusobni položaj kamere i lasera itd. Ručno projektiranje takvog sustava bez mogućnosti vizualizacije odnosa individualnih komponenti te testiranja njihovih različitih parametara može biti vrlo zamoran proces.

Cilj ovog završnog rada je izraditi program matematičko-prostornog odnosa kamere s objektivom i linijskog lasera koji će omogućiti prethodno navedenu vizualizaciju i brzo testiranje parametara vezanih uz specifične karakteristike individualnih komponenti. Isto tako, omogućiti će pronalaženje optimalnih postavki sustava čime se minimizira potreba za isprobavanjem različitih postavka na stvarnom sustavu. Također, program će biti moguće koristiti u obrazovne svrhe za analizu teoretskih osnova koje su prikazane u nastavku rada. Uz to, program može služiti kao alat za potrebe drugih istraživačkih i znanstvenih radova. Primjer toga je da će poslužiti za generiranje trening seta prostornih i 2D točaka u znanstvenom radu na temu



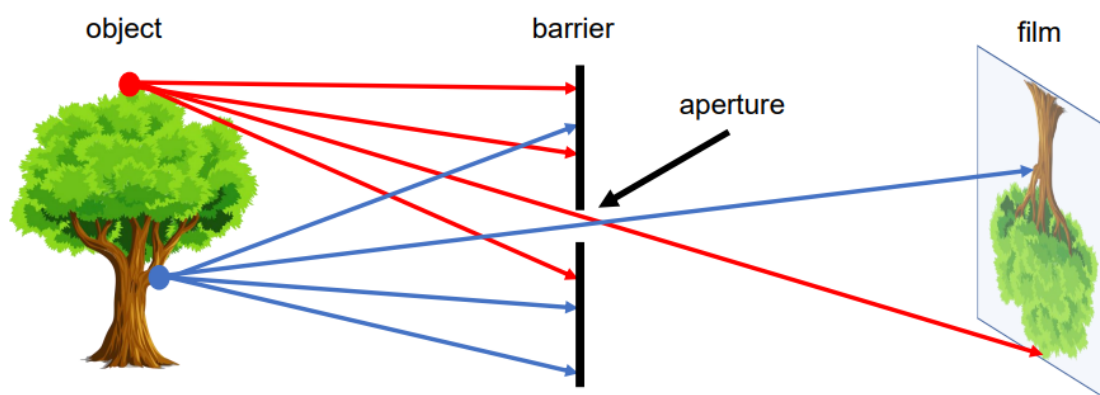
kalibracije kamera-laser sustava pomoću neuronskih mreža. Osim samog treniranja, program će se koristiti i za evaluaciju različitih arhitektura neuronskih mreža.

U nastavku je objašnjena teorija potrebna za shvaćanje matematičko-prostornog odnosa kamere i lasera, nakon toga je opisana funkcionalnost samog programa, a naposljetku je prikazana usporedba idealnih i realnih rezultata kako bi se testirala valjanost programa.

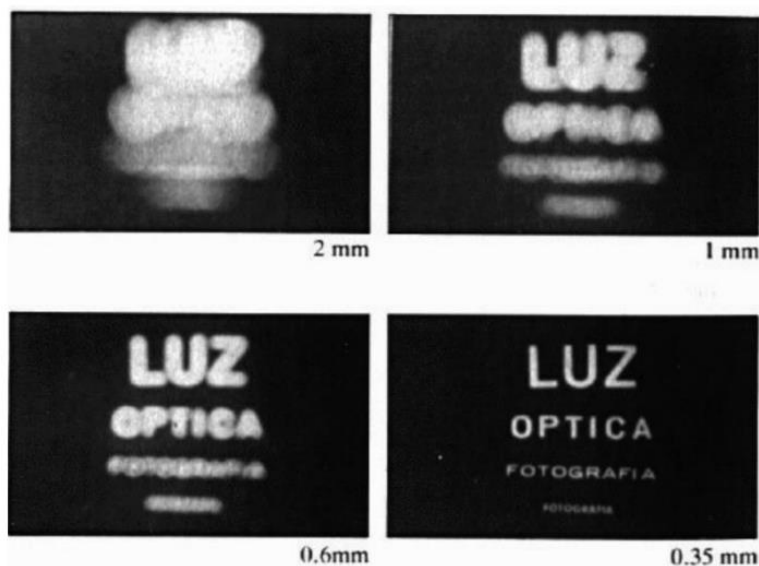
## 2. PINHOLE MODEL KAMERE

Temelj matematičkog modela kamere s objektivom je *pinhole* model kamere. Poznato je da se uz pomoć male rupice, tj. *pinholea*, može dobiti slika na zidu zamračene sobe. Ako se neki objekt stavi ispred te rupice, zrake svjetlosti koje pogađaju objekt preslikavaju svaku točku tog objekta na zid prolazeći pritom kroz rupicu. Svaka od tih zraka svjetlosti predstavlja pravac koji spaja jednu točku na objektu i *pinhole*, a sjecište pravca i ravnine zida predstavlja preslikanu točku. Iz toga slijedi da će slika na zidu biti obrnuta. Što je *pinhole* manji to će slika biti oštija i jasnija, no proći će manje svjetlosti, što je *pinhole* veći slika će biti mutna, no proći će više svjetlosti. [5]

Slika 3 prikazuje *pinhole* model kamere dok Slika 4 prikazuje učinak manjeg i većeg *pinholea* tj. otvora blende kod stvarnih kamera.



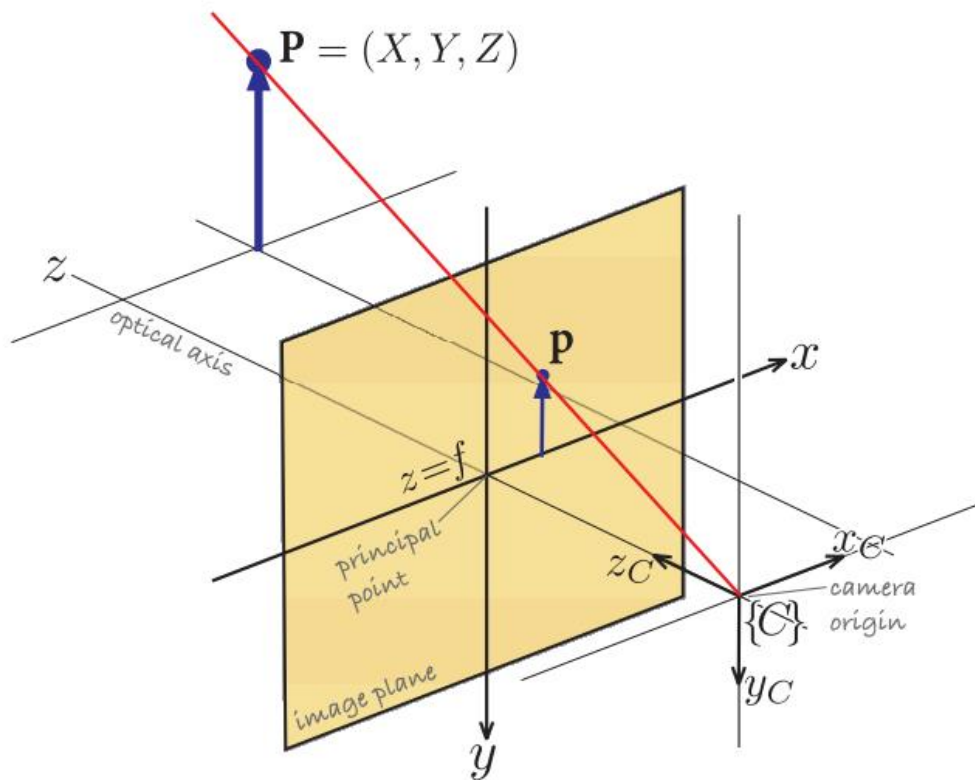
Slika 3. *Pinhole* model kamere [5]



Slika 4 Efekt različitih veličina otvora blende na sliku [5]

### 2.1. Matematički model *pinhole* kamere

Kod matematičkog modela *pinhole* kamere, *pinhole* predstavlja optičko središte kamere, tj. ishodište njezinog 3D koordinatnog sustava. U navedenom koordinatnom sustavu, z-os predstavlja optičku os kamere i usmjerena je prema željenom objektu. Zid zamračene sobe predstavlja ravninu slike koja se nalazi na udaljenosti  $f$  od ishodišta koordinatnog sustava kamere (u stvarnoj kameri ravninu slike predstavlja senzorski čip kamere). Oznaka  $f$  predstavlja žarišnu duljinu kamere i ima pozitivnu vrijednost te zbog toga dobivena slika neće biti obrnuta (u slučaju Slika 3 to bi značilo da bi se slika drveta nalazila na lijevoj strani *pinholea* i ne bi bila obrnuta). Sjecište optičke osi i ravnine slike naziva se glavna točka. Navedeni pojmovi jasno su vidljivi na Slika 5. Izvodi u nastavku proizlaze iz knjiga [6] i [7].



Slika 5. Matematičko-prostorni model *pinhole* kamere [6]

Koristeći svojstvo sličnih trokuta vrijedi da se projiciranjem točke  $\mathbf{P} = (X, Y, Z)$  u prostoru na ravninu slike dobije njena točka slike  $\mathbf{p} = (x, y)$  uz sljedeću jednakost:

$$x = f \frac{X}{Z}, y = f \frac{Y}{Z} \quad (1)$$

Navedena geometrijska transformacija naziva se centralna projekcija.

Nadalje, točka slike se može zapisati u homogenom obliku  $\tilde{\mathbf{p}} = (\tilde{x}, \tilde{y}, \tilde{z})$  gdje vrijedi:

$$\tilde{x} = fX, \tilde{y} = fY, \tilde{z} = Z \quad (2)$$

U matričnom obliku prethodni izraz glasi:

$$\tilde{\mathbf{p}} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3)$$

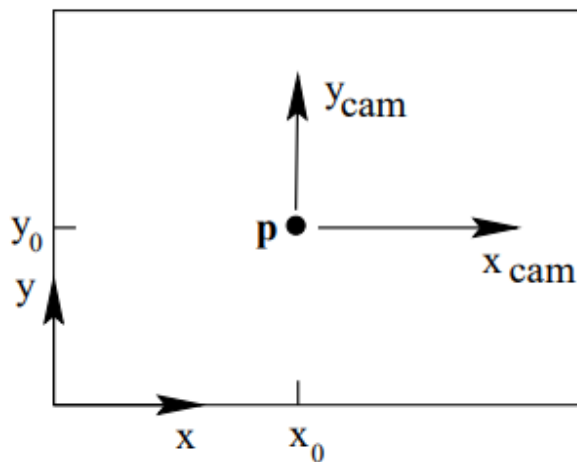
Nehomogene koordinate točke slike tada iznose:

$$x = \frac{\tilde{x}}{\tilde{z}}, y = \frac{\tilde{y}}{\tilde{z}} \quad (4)$$

Ako se točka  $\mathbf{P}$  također zapiše u homogenom obliku, jednačba (3) poprima sljedeći oblik:

$$\tilde{\mathbf{p}} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (5)$$

U prethodnim izrazima pretpostavljeno je da se ishodište koordinatnog sustava u ravnini slike poklapa s glavnom točkom. Kao što je prikazano na Slika 6, to najčešće nije slučaj, već se ishodište koordinatnog sustava nalazi u jednom od kutova slike.



Slika 6. Ravnina slike [7]

Jednačba (5) tada glasi:

$$\tilde{\mathbf{p}} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (6)$$

gdje su  $p_x$  i  $p_y$  koordinate glavne točke u ravnini slike. Daljnjim faktoriziranjem dobije se:

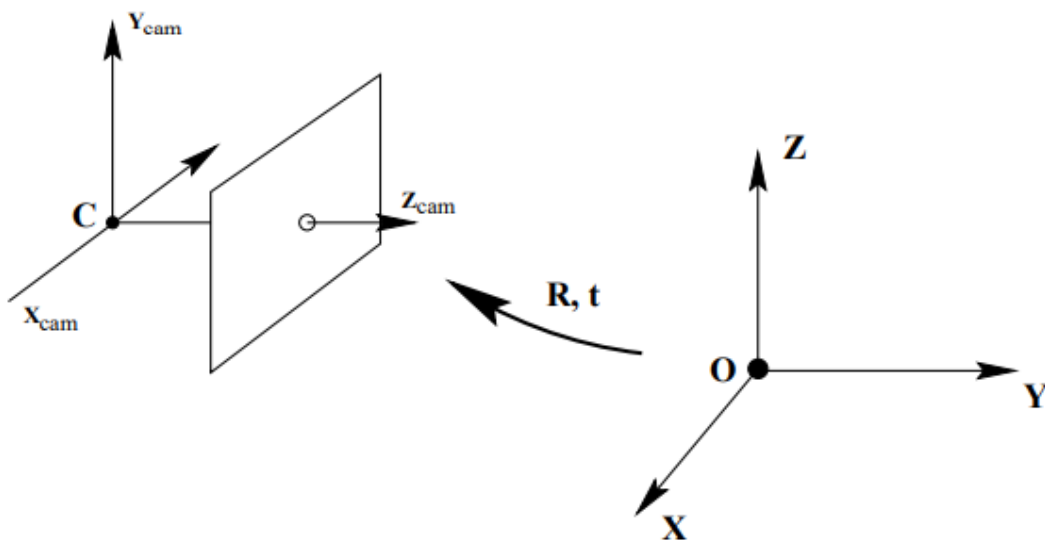
$$\tilde{\mathbf{p}} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (7)$$

što omogućuje sljedeći matrični zapis:

$$\tilde{\mathbf{p}} = \mathbf{K}[\mathbf{I} \mid \mathbf{0}]\tilde{\mathbf{P}} \quad (8)$$

Matrica  $\mathbf{K}$  naziva se kalibracijskom matricom kamere ili samo matrica kamere (engl. *camera matrix*).

Svi prethodni izrazi vrijede uz pretpostavku da se ishodište koordinatnog sustava kamere poklapa s ishodištem prostornog koordinatnog sustava (engl. *world coordinate frame*). U praksi je najčešće koordinatni sustav kamere translatican i zarotiran u odnosu na prostorni koordinatni sustav kao što je slučaj na Slika 7. U slučaju ovog rada, bit će potrebno translaticati i zarotirati koordinatni sustav kamere u odnosu na koordinatni sustav lasera.



Slika 7. Translacija i rotacija koordinatnog sustava kamere u odnosu na prostorni koordinatni sustav [7]

Ako se ishodište koordinatnog sustava kamere označi kao  $\mathbf{C}$ , točka u njenom koordinatnom sustavu kao  $\mathbf{P}_{\text{cam}}$ , a ista ta točka u prostornom koordinatnom sustavu zadrži oznaku  $\mathbf{P}$ , vrijedi sljedeća jednakost:

$$\mathbf{P}_{\text{cam}} = \mathbf{P} - \mathbf{C} \quad (9)$$

Uzimanjem u obzir rotaciju, dobije se:

$$\mathbf{P}_{\text{cam}} = \mathbf{R}(\mathbf{P} - \mathbf{C}), \quad (10)$$

gdje  $\mathbf{R}$  predstavlja  $3 \times 3$  matricu rotacije koja će biti detaljnije obrađena u nastavku rada. Važno je napomenuti da je translacija apsolutna što znači da se koordinatni sustav kamere translacija u odnosu na prostorni koordinatni sustav dok je rotacija relativna, koordinatni sustav kamere rotira oko vlastitih koordinatnih osi. Jednadžbu (10) moguće je zapisati u homogenom obliku:

$$\mathbf{P}_{\text{cam}} = \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{P} \quad (11)$$

Konačna jednadžba matematičkog modela *pinhole* kamere tada glasi:

$$\tilde{\mathbf{p}} = \mathbf{KR}[\mathbf{I} \mid -\mathbf{C}]\tilde{\mathbf{P}}, \quad (12)$$

odnosno skraćeno

$$\tilde{\mathbf{p}} = \mathbf{M}_p \tilde{\mathbf{P}}, \quad (13)$$

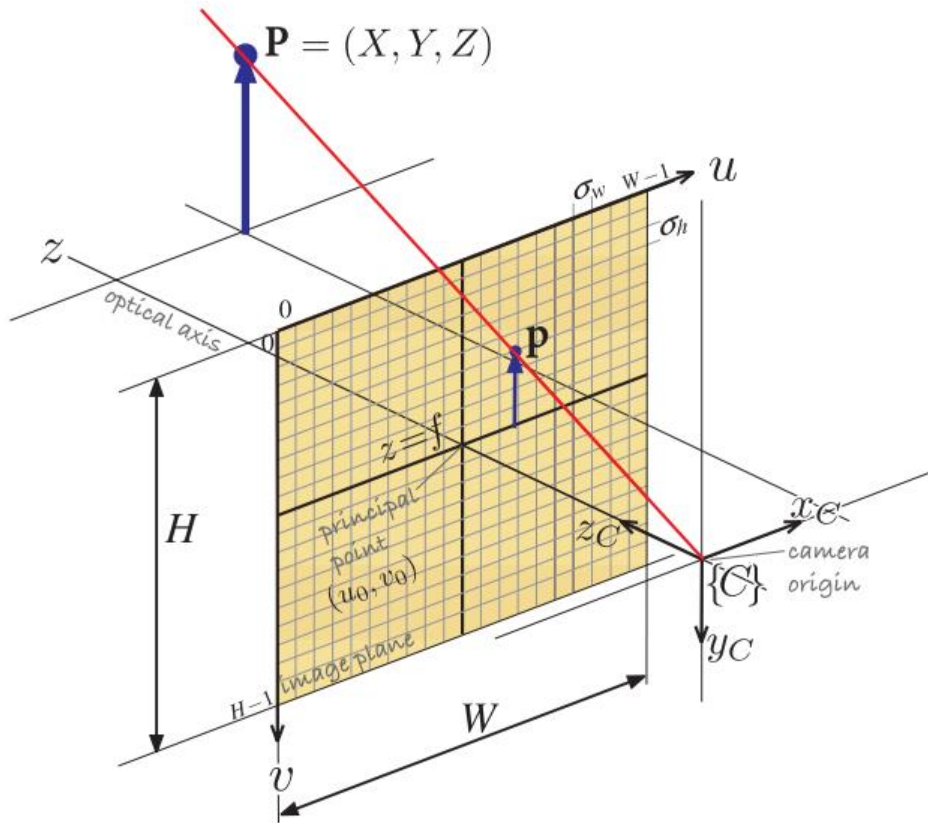
gdje  $\mathbf{M}_p$  predstavlja matricu projekcije (eng. *projection matrix*, u literaturi se najčešće označuje slovima  $\mathbf{P}$  ili  $\mathbf{C}$ ). Matrica se sastoji od intrinzičnih parametara kamere (parametri matrice kamere  $\mathbf{K}$ ) i ekstrinzičnih parametara kamere (parametri matrice rotacije  $\mathbf{R}$  i ishodišta koordinatnog sustava kamere  $\mathbf{C}$ ).

Naposlijetku, potrebno je pretvoriti homogene koordinate točke  $\tilde{\mathbf{p}}$  u nehomogene prema jednadžbi (4).

## 2.2 Matematički model *pinhole* kamere koristeći diskretnu ravninu slike

Da bi se prethodno izvedeni matematički model mogao primijeniti u praksi, potrebno je uzeti u obzir da u stvarnoj digitalnoj kameri ravnina slike odgovara senzorskom čipu kamere (kao što je spomenuto na početku prethodnog poglavlja). To znači da ravnina slike predstavlja koordinatnu mrežu malih senzorskih elemenata koji služe za prikupljanje svjetlosti (engl. *photosites*) koji zajedno čine prethodno spomenuti senzorski čip. Koordinatna mreža ima određenu širinu i visinu koje odgovaraju širini i visini senzorskog čipa. Senzorski elementi na čipu odgovaraju pikselima na digitalnoj slici.

Zbog toga je potrebno proširiti matematički model tako da se svi potrebni parametri mogu izraziti u adekvatnim mjernim jedinicama. Slika 8 prikazuje centralnu projekciju točke na diskretnu ravninu slike.



Slika 8. Centralna projekcija na diskretnu ravninu slike [6]

Kao što se može vidjeti, ravnina slike podijeljena je na piksele koordinata  $(u, v)$  za koje vrijedi sljedeća jednakost:

$$u = \frac{x}{\rho_w} + u_0, v = \frac{y}{\rho_h} + v_0, \quad (14)$$

gdje  $\rho_w$  i  $\rho_h$  predstavljaju širinu i visinu svakog piksela, a  $(u_0, v_0)$  predstavlja koordinate glavne točke u pikselima:

$$u_0 = \frac{p_x}{\rho_w}, v_0 = \frac{p_y}{\rho_h} \quad (15)$$

Uzimajući u obzir prethodne jednakosti, potrebno je modificirati matricu kamere:

$$\mathbf{K} = \begin{bmatrix} f/\rho_w & 0 & u_0 \\ 0 & f/\rho_h & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (16)$$



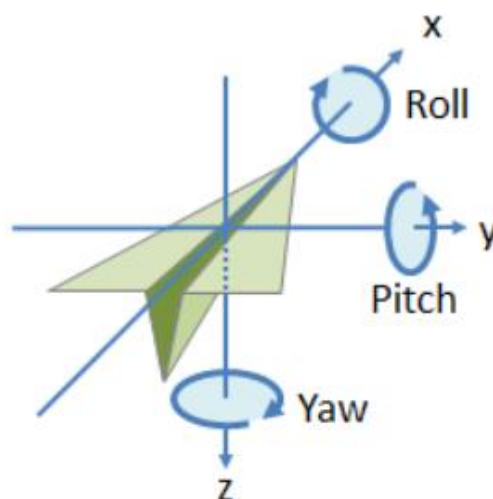
Ostatak jednadžbe (12) ostaje isti te se pomoću nje dobije točka  $\tilde{\mathbf{p}} = (\tilde{u}, \tilde{v}, \tilde{w})$ . Nadalje, potrebno je pretvoriti homogene koordinate točke  $\tilde{\mathbf{p}}$  u nehomogene:

$$u = \frac{\tilde{u}}{\tilde{w}}, v = \frac{\tilde{v}}{\tilde{w}} \quad (17)$$

### 2.3. Matrica rotacije

Matricu rotacije  $\mathbf{R}$  moguće je dobiti pomoću Eulerovih kutova. To su kutovi koji određuju orijentaciju jednog koordinatnog sustava u odnosu na drugi. Najčešće se označuju grčkim slovima  $\alpha, \beta$  i  $\gamma$  ili  $\varphi, \psi$  i  $\vartheta$ . U ovom radu koristit će se oznake  $\Theta_x, \Theta_y$  i  $\Theta_z$  -  $\Theta_x$  je oznaka za kut rotacije oko osi x,  $\Theta_y$  oko osi y, a  $\Theta_z$  oko osi z. Često se navedeni kutovi nazivaju kutovima valjanja, posrtanja i skretanja (eng. *roll, yaw* i *pitch*). Redoslijed rotacija koji će se koristiti u radu je z-y-x (konvencionalan redoslijed u robotici je z-y-z). Važno je napomenuti da se ovdje zapravo koriste Tait-Bryan Eulerovi kutovi (kod normalnih Eulerovih kutova, rotacija se opisuje oko dvije koordinatne osi dok je kod Tait-Bryan kutova moguća rotacija oko sve tri koordinatne osi). Pomoću navedenih kutova, moguće je dobiti matricu valjanja  $\mathbf{R}_x$  (eng. *roll matrix*), matricu posrtanja  $\mathbf{R}_y$  (eng. *pitch matrix*) i matricu skretanja  $\mathbf{R}_z$  (eng. *yaw matrix*). Matrica valjanja opisuje rotaciju oko osi x, matrica posrtanja oko osi y, a matrica skretanja oko osi z. [8]

Slika 9 prikazuje Eulerove kutove i njihove pripadne koordinatne osi. Kutovi su pozitivni u smjeru strelica prikazanih na slici (pravilo desne ruke).



Slika 9. Eulerovi kutovi i njihove pripadne koordinatne osi [8]

Jednadžbe matrica  $\mathbf{R}_x$ ,  $\mathbf{R}_y$  i  $\mathbf{R}_z$  su sljedeće:

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \Theta_x & -\sin \Theta_x \\ 0 & \sin \Theta_x & \cos \Theta_x \end{bmatrix} \quad (18)$$

$$\mathbf{R}_y = \begin{bmatrix} \cos \Theta_y & 0 & \sin \Theta_y \\ 0 & 1 & 0 \\ -\sin \Theta_y & 0 & \cos \Theta_y \end{bmatrix} \quad (19)$$

$$\mathbf{R}_z = \begin{bmatrix} \cos \Theta_z & -\sin \Theta_z & 0 \\ \sin \Theta_z & \cos \Theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (20)$$

Množenjem navedenih matrica dobije se matrica rotacije  $\mathbf{R}$ :

$$\mathbf{R} = \mathbf{R}_z \mathbf{R}_y \mathbf{R}_x = \begin{bmatrix} \cos \Theta_y \cos \Theta_z & a & b \\ \cos \Theta_y \sin \Theta_z & c & d \\ -\sin \Theta_y & \sin \Theta_x \cos \Theta_y & \cos \Theta_x \cos \Theta_y \end{bmatrix}, \quad (21)$$

gdje su:

$$\begin{aligned} a &= \sin \Theta_x \sin \Theta_y \cos \Theta_z - \cos \Theta_x \sin \Theta_z \\ b &= \sin \Theta_x \sin \Theta_z + \cos \Theta_x \sin \Theta_y \cos \Theta_z \\ c &= \cos \Theta_x \cos \Theta_z + \sin \Theta_x \sin \Theta_y \sin \Theta_z \\ d &= \cos \Theta_x \sin \Theta_y \sin \Theta_z - \sin \Theta_x \cos \Theta_z \end{aligned} \quad (22)$$

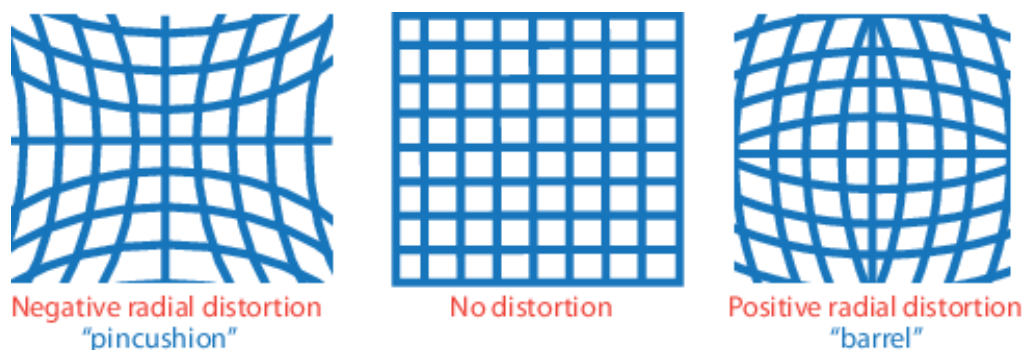
#### 2.4. Matematički model kamere s objektivom

Matematički model obrađen u poglavlju 2.4. predstavlja idealan slučaj projekcije točke na ravninu slike kamere. Kod stvarne kamere s objektivom, potrebno je uzeti u obzir utjecaj leće i nesavršenosti koje ona uzrokuje. Geometrijska distorzija je najbitniji učinak korištenja leće i potrebno ju je matematički opisati kako bi model kamere bio valjan. Može se podijeliti na radijalnu i tangencijalnu distorziju.

Radijalna distorzija se pojavljuje zbog nesavršenosti u konstrukciji leće. Njezin učinak je da ravne linije u stvarnom svijetu izgledaju zakrivljeno na snimljenoj slici. Može se podijeliti na bačvastu (eng. *barrel*) i jastučastu (eng. *pincushion*) distorziju. Kod bačvaste distorzije, povećanje slike smanjuje se što je veća udaljenost od glavne točke. Uzrokuje zakrivljenje ravnih linija blizu ruba slike prema van. S druge strane, kod jastučaste distorzije, povećanje slike je

veće što je veća udaljenost od glavne točke. Uzrokuje zakrivljenje ravnih linija blizu ruba slike prema unutra. [6]

Na Slika 10 su prikazane bačvasta i jastučasta distorzija.

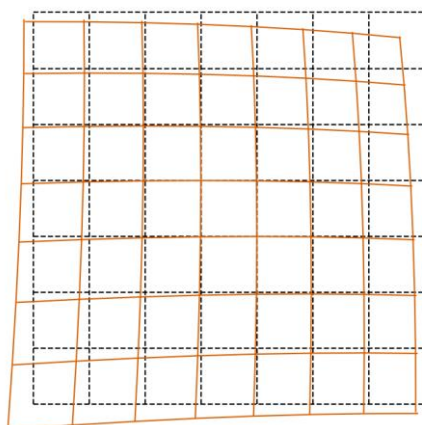


**Slika 10. Bačvasta i jastučasta distorzija**

Tangencijalna distorzija manje je uobičajena vrsta izobličenja leće koje se događa kada elementi leće nisu savršeno poravnati sa senzorom slike kamere. Zbog toga se ravne linije na slici pojavljuju nakošene. Tangencijalna distorzija općenito je uzrokovana malim neusklađenostima ili nepravilnostima u montaži i konstrukciji leće. [9]

Na Slika 11 je prikazana tangencijalna distorzija.

Tangential (Decentering) Distortion



**Slika 11. Tangencijalna distorzija [9]**

Uzimajući u obzir utjecaj radijalne i tangencijalne distorzije, potrebno je modificirati jednadžbu (17). Kako bi se distorzijski model mogao primijeniti, najprije je potrebno piksel koordinate  $(u, v)$  točke  $\mathbf{p}$  pretvoriti u normalizirane koordinate  $(u_n, v_n)$ :

$$\begin{aligned} u_n &= \frac{u - u_0}{\frac{f}{\rho_w}} \\ v_n &= \frac{v - v_0}{\frac{f}{\rho_h}} \end{aligned} \quad (23)$$

Također, potrebno je definirati kvadrat udaljenosti točke  $\mathbf{p}$  od glavne točke, tj. centra zakrivljenosti slike (također u normaliziranim koordinatama):

$$r^2 = u_n^2 + v_n^2 \quad (24)$$

Nakon toga, moguće je primijeniti distorzijski model:

$$\begin{aligned} u_d &= u_n(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + (2p_1 u_n v_n + p_2(r^2 + 2u_n^2)) \\ v_d &= \underbrace{v_n(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)}_{\text{radijalna distorzija}} + \underbrace{(p_1(r^2 + 2v_n^2) + 2p_2 u_n v_n)}_{\text{tangencijalna distorzija}}, \end{aligned} \quad (25)$$

gdje  $k_1, k_2, k_3, p_1$  i  $p_2$  predstavljaju distorzijske koeficijente. Naposljetku, potrebno je normalizirane koordinate natrag pretvoriti u piksel koordinate:

$$\begin{aligned} u' &= u_d \frac{f}{\rho_w} + u_0 \\ v' &= v_d \frac{f}{\rho_h} + v_0 \end{aligned} \quad (26)$$

### 3. MATEMATIČKE FUNKCIJE POTREBNE ZA VIZUALIZACIJU KOMPONENATA

U ovom poglavlju dat će se pregled preostalih potrebnih matematičkih funkcija za funkcionalnost programa. Funkcije će poslužiti za dio programa vezan uz vizualizaciju komponenata. Omogućit će vizualizaciju projekcije točaka na ravninu slike u prostoru, prikaz vidnog polja kamere i prikaz točaka koje predstavljaju lasersku liniju.

#### 3.1. Plückerova matrica

Plückerova matrica (eng. *Plücker matrix*) je homogena matrica dimenzija  $4 \times 4$  koja predstavlja pravac u prostoru koji se može dobiti pomoću dvije točke poznatih koordinata. Njezina jednadžba je sljedeća: [7]

$$\mathbf{L} = \mathbf{AB}^T - \mathbf{BA}^T \quad (27)$$

Ona će poslužiti za računanje koordinata točke sjecišta pravca i ravnine. U programu ovog završnog rada, koristit će se za vizualizaciju projekcije točaka koje predstavljaju lasersku liniju na ravninu slike kamere u prostoru te za prikaz vidnog polja kamere.

#### 3.2. Jednadžba ravnine u matričnom obliku

Jednadžba ravnine u matričnom obliku može se dobiti pomoću poznatih koordinata triju točaka u prostoru. Navedene točke označit će se sa  $\mathbf{X}_1$ ,  $\mathbf{X}_2$  i  $\mathbf{X}_3$ . Jednadžba ravnine je:

$$\boldsymbol{\pi} = \begin{bmatrix} (\mathbf{X}_1 - \mathbf{X}_3) \times (\mathbf{X}_2 - \mathbf{X}_3) \\ -\mathbf{X}_3(\mathbf{X}_1 \times \mathbf{X}_2) \end{bmatrix} \quad (28)$$

U ovom kontekstu, oznaka  $\times$  predstavlja vektorski produkt. Važno je napomenuti da u jednadžbi (28) prvi redak predstavlja vektor normale matrice, a sama matrica je homogena, dimenzija  $4 \times 1$ . Za detaljna svojstva i izvod navedene matrice kao i svojstva Plückerove matrice, čitatelja se upućuje na literaturu [7].

Konačno, korištenjem jednadžba (27) i (28), moguće je zapisati jednadžbu za sjecište pravca i ravnine:

$$\mathbf{x} = \mathbf{L}\boldsymbol{\pi} \quad (29)$$

Nadalje, točku  $\mathbf{x}$  je potrebno pretvoriti iz homogenih koordinata u nehomogene.

### 3.3. Jednadžba pravca u prostoru

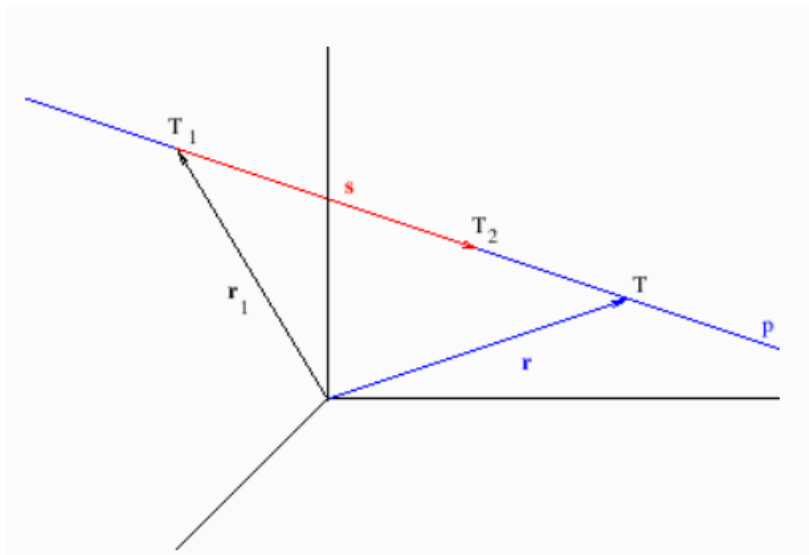
Jednadžba pravca u prostoru koristit će za generiranje i vizualizaciju diskretiziranog seta točaka koje predstavljaju lasersku liniju. Pravac  $\mathbf{p}$  je u prostoru određen s dvije različite točke  $T_1$  i  $T_2$ . Za svaku točku  $T$  koja leži na tom pravcu, vektori  $\overrightarrow{T_1T_2}$  i  $\overrightarrow{T_1T}$  su kolinearni. Vektorska jednadžba pravca prema [10] glasi:

$$\mathbf{r} = \mathbf{r}_1 + t\mathbf{s}, \quad t \in \mathbb{R} \quad (30)$$

gdje su:

$$\begin{aligned} \mathbf{s} &= \overrightarrow{T_1T_2} \\ \mathbf{r}_1 &= \overrightarrow{OT_1} \\ \mathbf{r} &= \overrightarrow{OT} \end{aligned} \quad (31)$$

Točka  $O$  predstavlja ishodište prostornog koordinatnog sustava. Na Slika 12 prikazan je pravac u prostoru.



Slika 12. Pravac u prostoru [10]

## 4. MATEMATIČKI MODEL LINIJSKOG LASERA

Za potrebe ovoga završnog rada, uvedena su sljedeća pojednostavljena kod matematičkog modela linijskog lasera:

- Koordinatni sustav lasera smješten je u ishodište Kartezijevog trodimenzionalnog koordinatnog sustava.
- Koordinatni sustav lasera je fiksiran, tj. rotacija i translacija nisu moguće.
- Laserska svjetlost pogađa ravnu površinu tako da se uvijek dobije ravna, neisprekidana linija.
- Za potpunu definiciju laserskog sustava potrebno je poznavati njegov kut širenja snopa svjetlosti i udaljenost od ravne površine.
- Snop laserske svjetlosti gleda u pozitivnom smjeru osi  $y$ .

Ako se sa  $\varphi$  označi kut širenja snopa svjetlosti, sa  $\mathbf{L}_0 = (0, 0, 0)$  ishodište koordinatnog sustava lasera, a sa  $l$  udaljenost ishodišta od ravne površine, moguće je dobiti krajnje točke laserske linije na sljedeći način:

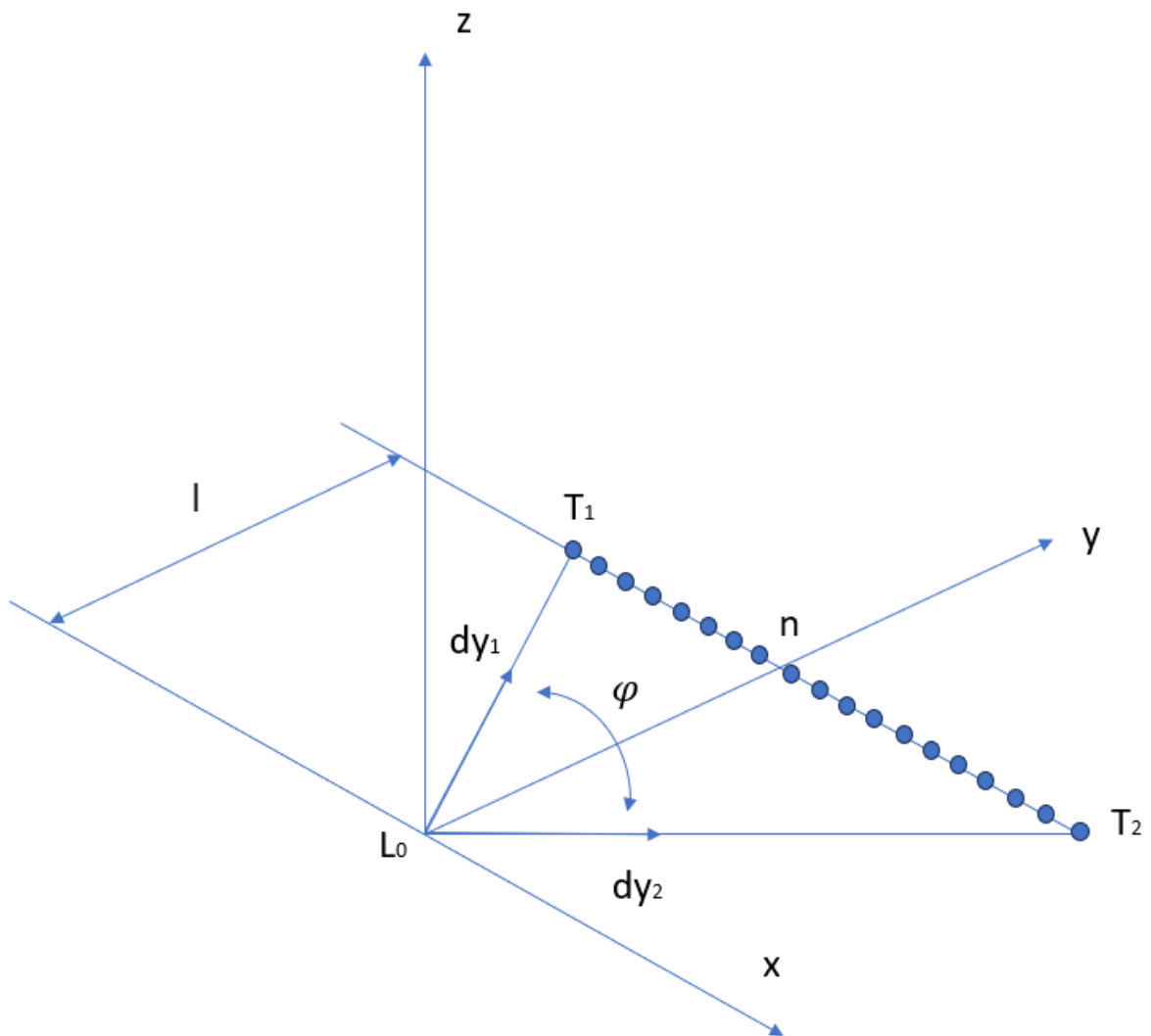
$$T_1 = \mathbf{L}_0 + \left( \frac{l}{\cos\left(\frac{\varphi}{2}\right)} \right) * \mathbf{dy}_1$$

$$T_2 = \mathbf{L}_0 + \left( \frac{l}{\cos\left(-\frac{\varphi}{2}\right)} \right) * \mathbf{dy}_2$$
(32)

Sa  $\mathbf{dy}_1$  je označen jedinični vektor  $\vec{j}$  zarotiran oko osi  $z$  za kut  $\frac{\varphi}{2}$ , a sa  $\mathbf{dy}_2$  isti jedinični vektor zarotiran oko osi  $z$  za kut  $-\frac{\varphi}{2}$ . Preostale točke na laserskoj liniji (točke između  $T_1$  i  $T_2$ ) dobiju se pomoću jednadžba (30) i (31) gdje, u ovom slučaju,  $t$  iznosi:

$$t = \frac{k}{n - 1},$$
(33)

gdje je  $k$  jednak  $1, 2, 3 \dots n - 1$  (u programskom kodu se dobije iteracijom u *for* petlji), a  $n$  predstavlja broj točaka na laserskoj liniji. Slika 13 prikazuje prostorni model linijskog lasera.



Slika 13. Prostorni model linijskog lasera



## 5. PROGRAM

Matematičko-prostorni model odnosa kamere s objektivom i linijskog lasera programiran je u *Python* programskom jeziku. U programu su korištene sljedeće *Pythonove* biblioteke:

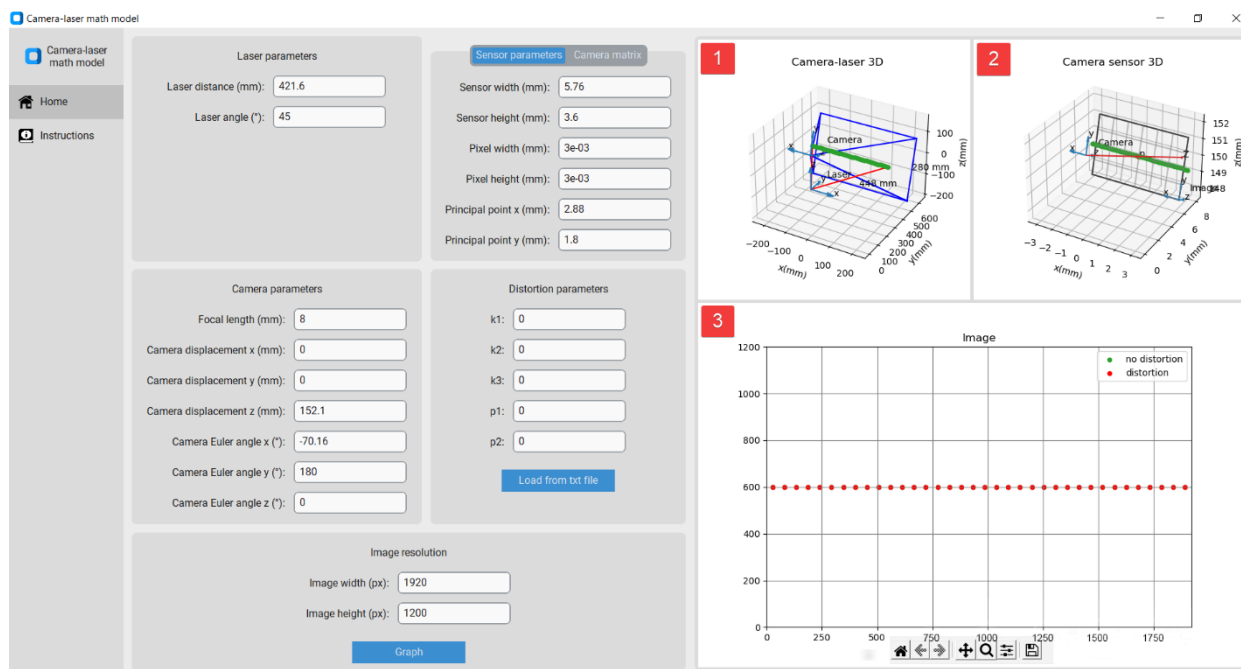
- *Numpy* – biblioteka potrebna za linearnu algebru (omogućuje stvaranje n-dimenzionalnih polja, sadrži veliki broj matematičkih operacija koje se mogu primijeniti na navedena polja te se tzv. vektorizacijom može ubrzati izvršavanje *Python* programa).
- *Matplotlib* – biblioteka koja služi za vizualizaciju podataka i stvaranje 2D i 3D grafova koji mogu biti statični, animirani i interaktivni.
- *Tkinter* – biblioteka koja omogućuje stvaranje grafičkog korisničkog sučelja.
- *CustomTkinter* – biblioteka koja je nadogradnja na standardni *Tkinter* i omogućuje moderniji izgled korisničkog sučelja.

Programski kod je pisan objektno orijentiranim pristupom programiranju. Kod koji opisuje temeljni matematički model *pinhole* kamere preuzet je s [11]. Model je nadograđen i prilagođen zahtjevima ovog završnog rada. Uz prethodno navedene *Pythonove* biblioteke, napravljeno je nekoliko korisničko definiranih modula koji omogućuju bolju preglednost koda i jasniju strukturu programa. Popis navedenih modula i njihov kratak opis je sljedeći:

- *\_frames* – modul koji služi za stvaranje 3D koordinatnih sustava.
- *\_homogeneous* – modul koji omogućuje pretvorbu homogenih koordinata u nehomogene i obrnuto.
- *\_image* – modul koji služi za generiranje ravnine slike.
- *\_laser* – modul koji služi za vizualizaciju linijskog lasera i generiranje seta 3D točaka koje odgovaraju laserskoj liniji.
- *\_matrices* – modul koji omogućuje stvaranje potrebnih matrica: Plückerovu matricu, matricu rotacije, kalibracijsku matricu, matricu projekcije i matricu ravnine.
- *\_points* – modul pomoću kojeg se dobiju koordinate projiciranih 3D točaka na ravninu slike kamere. Uz to, omogućuje vizualizaciju navedenih 3D i 2D točaka.
- *\_principal axis* – modul koji omogućuje vizualizaciju glavne osi kamere.
- *\_utils* – modul koji služi za crtanje 3D vektora

Na Sliku 14 prikazan je početni zaslon programa.

## 5.1. Početni zaslon i zaslon s uputstvima

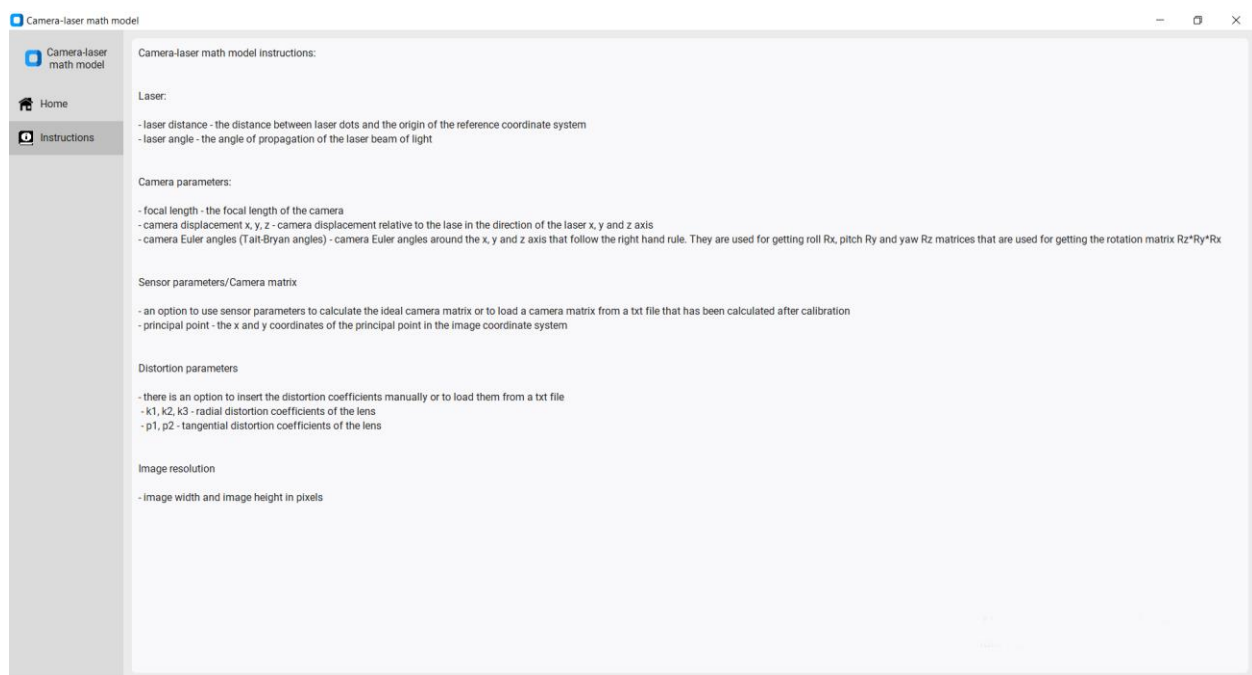


Slika 14. Početni zaslon programa

Program je podijeljen u tri segmenta. Lijevo se nalazi navigacijski okvir koji omogućuje odabir između početnog zaslona i zaslona s kratkim uputstvima vezanih uz glavne značajke programa. Desno od navigacijskog okvira nalazi se set okvira koji služi za unos podataka potrebnih za vizualizaciju kamera-laser sustava. Sama vizualizacija jasno je prikazana u desnom dijelu zaslona. Sastoji se od tri grafa koji su na Slika 14 označeni brojevima 1-3:

1. kamera-laser sustav u 3D-u
2. senzor kamere u 3D-u
3. slika koju kamera vidi u 2D-u.

Na Slika 15 prikazan je zaslon s uputstvima.



**Slika 15. Zaslonski prikaz s uputstvima**

## 5.2. Unos podataka

The form is organized into five main sections:

- Laser parameters:** Laser distance (mm): 421.6, Laser angle (°): 45.
- Sensor parameters:** Sensor width (mm): 5.76, Sensor height (mm): 3.6, Pixel width (mm): 3e-03, Pixel height (mm): 3e-03, Principal point x (mm): 2.88, Principal point y (mm): 1.8.
- Camera parameters:** Focal length (mm): 8, Camera displacement x (mm): 0, Camera displacement y (mm): 0, Camera displacement z (mm): 152.1, Camera Euler angle x (°): -70.16, Camera Euler angle y (°): 180, Camera Euler angle z (°): 0.
- Distortion parameters:** k1: 0, k2: 0, k3: 0, p1: 0, p2: 0. Includes a 'Load from txt file' button.
- Image resolution:** Image width (px): 1920, Image height (px): 1200. Includes a 'Graph' button.

Slika 16. Set okvira za unos podataka

Kao što Slika 16 prikazuje, parametri koje je moguće unijeti su sljedeći:

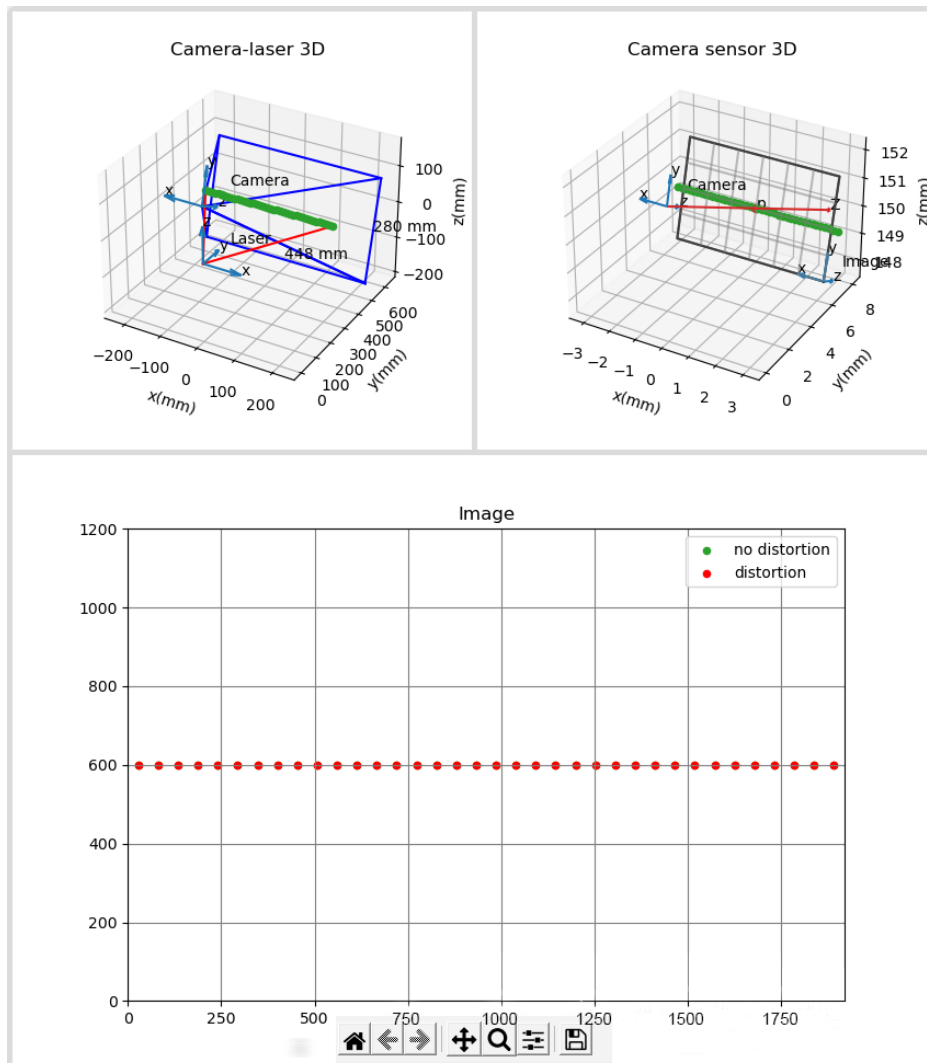
- Laserski parametri – udaljenost lasera od ravne površine u mm (*laser length*) i kut širenja snopa svjetlosti u stupnjevima (*laser angle*).
- Senzorski parametri – širina i visina senzora u mm (*sensor width, sensor height*), tj. širina i visina ravnine slike u 3D-u, širina i visina piksela (*pixel width, pixel height*) te koordinate

glavne točke (*principal point x*, *principal point y*), sve u mm. Kod ovog okvira moguć je i odabir kartice *Camera matrix* koja će kasnije biti detaljno objašnjena.

- Parametri kamere – žarišna duljina kamere u mm (*focal length*), translacija kamere u smjeru  $x$ ,  $y$  i  $z$  osi prostornog koordinatnog sustava u mm (*camera displacement x, y, z*) te rotacije kamere oko osi vlastitog koordinatnog sustava u stupnjevima, koristeći pravilno desne ruke (*camera Euler angle x, y, z*).
- Distorzijski parametri – distorzijski parametri, tj. parametri radijalne distorzije  $k_1, k_2$  i  $k_3$  te tangencijalne distorzije  $p_1$  i  $p_2$ . Navedene parametre je moguće unijeti ručno ili učitati iz .txt datoteke.
- Rezolucija slike – širina i visina slike u pikselima (*image width, image height*). U ovom okviru se nalazi i gumb za vizualizaciju kamera laser sustava koja se dobije pomoću prethodno navedenih parametara.

Kao što se može vidjeti, program nakon pokretanja sadrži već zadane vrijednosti parametara koje su dobivene na temelju stvarnog kamera-laser sustava dostupnog u laboratoriju. Način na koji su parametri dobiveni objasniti će se u poglavlju vezanim uz eksperimentalnu provjeru programa.

### 5.3. Vizualizacija



Slika 17. Vizualizacija kamera-laser sustava

Slika 17 prikazuje vizualizaciju kamera-laser sustava u 3D-u i onoga što kamera vidi u 2D-u. Vizualizacija je dobivena pomoću prethodno navedenih zadanih vrijednosti parametara.

Na grafu kamera-laser sustava u 3D-u može se vidjeti koordinatni sustav lasera i kamere, set točaka laserske linije te iznos širine i visine vidnog polja kamere. Naime, radna udaljenost kamere je za potrebe ovog završnog rada zadana kao udaljenost lasera od ravne površine uvećana za 200 mm. Ako bi udaljenost kamere i lasera bila velika, a udaljenost lasera od ravne površine mala, vizualni prikaz dometa kamere bio bi premalen te bi bilo potrebno povećati radnu udaljenost kamere u programskom kodu ili staviti mogućnost njenog ručnog unosa u

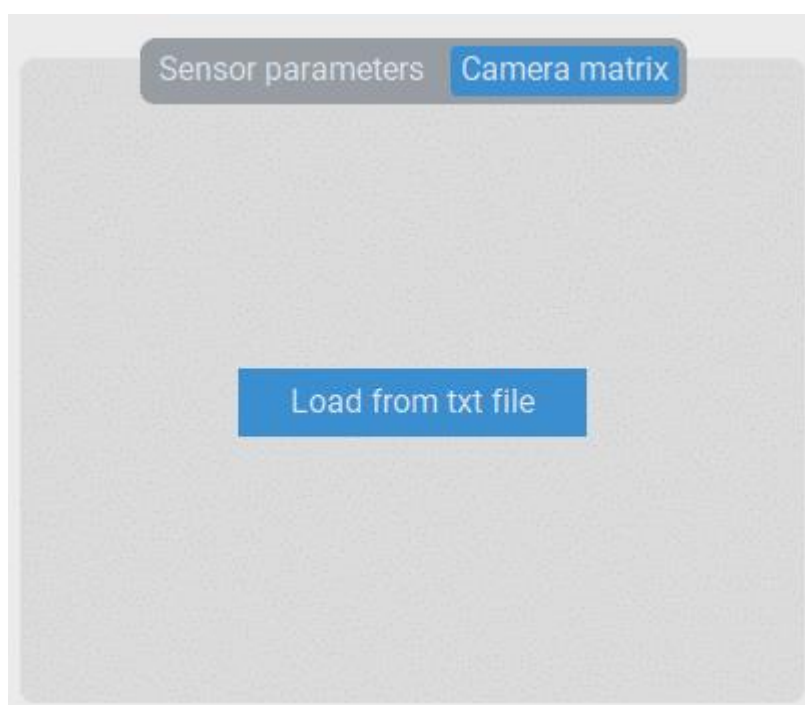
korisničkom sučelju. Budući da je udaljenost kamere i lasera u kamera-laser sustavu dostupnom u laboratoriju mala, mogućnost ručnog unošenja radne udaljenosti kamere nije stavljena u korisničko sučelje programa.

Na grafu koji prikazuje senzor kamere je vidljiva projekcija točaka laserske linije. Također jasno je vidljiva optička os kamere kao i glavna točka. Ishodište koordinatnog sustava senzora (ravnine slike) nalazi se u njegovom donjem desnom kutu.

Posljednji graf prikazuje sliku koju kamera vidi. Na slici su prikazane distorzirane i ne distorzirane točke laserske linije. Budući da su zadane vrijednosti distorzijskih parametara nula, oba seta točaka se poklapaju.

#### 5.4. Camera matrix kartica

Na Sliku 18 prikazana je *Camera matrix* kartica

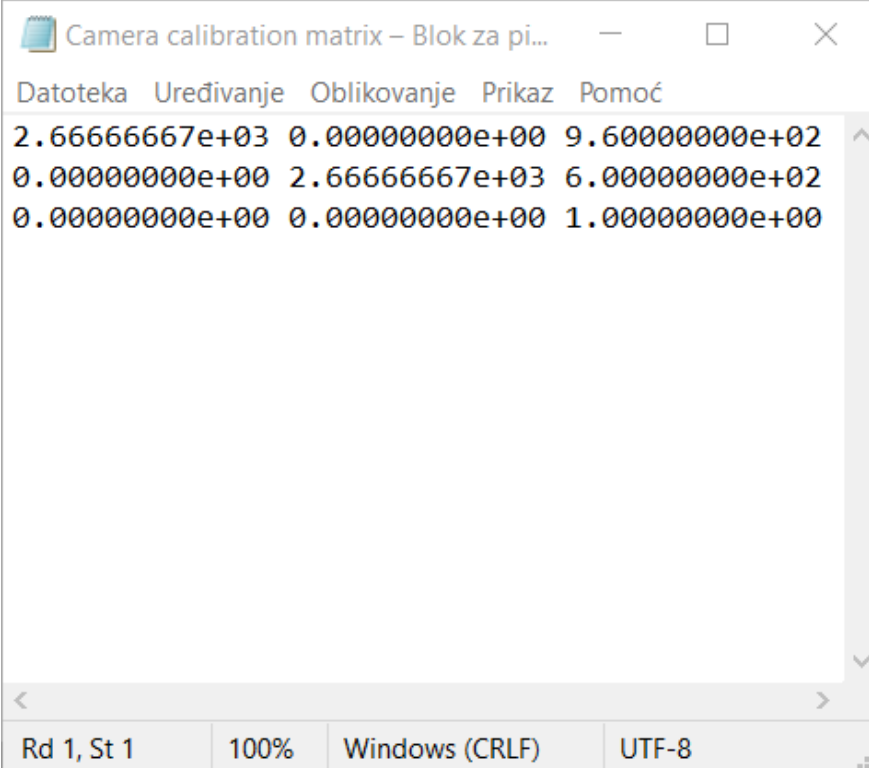


Slika 18. *Camera matrix* kartica

Ova kartica omogućuje učitavanje kalibracijske matrice kamere  $\mathbf{K}$  iz .txt datoteke. Naime, na početnom zaslonu moguće je unos svih parametara potrebnih za proračun idealne kalibracijske matrice. U slučaju stvarnih kamera, uz iste parametre, kalibracijska matrica će za određeni

postotak uvijek odstupati od idealne kalibracijske matrice. Učitavanjem stvarne kalibracijske matrice kamere i klikom na gumb *Graph*, program generira graf slike koju kamera vidi u novom prozoru što omogućuje usporedbu s grafom dobivenim koristeći idealnu kalibracijsku matricu.

Slika 19 prikazuje .txt datoteku kalibracijske kamere dobivene pomoću zadanih parametara na početnom zaslonu programa.



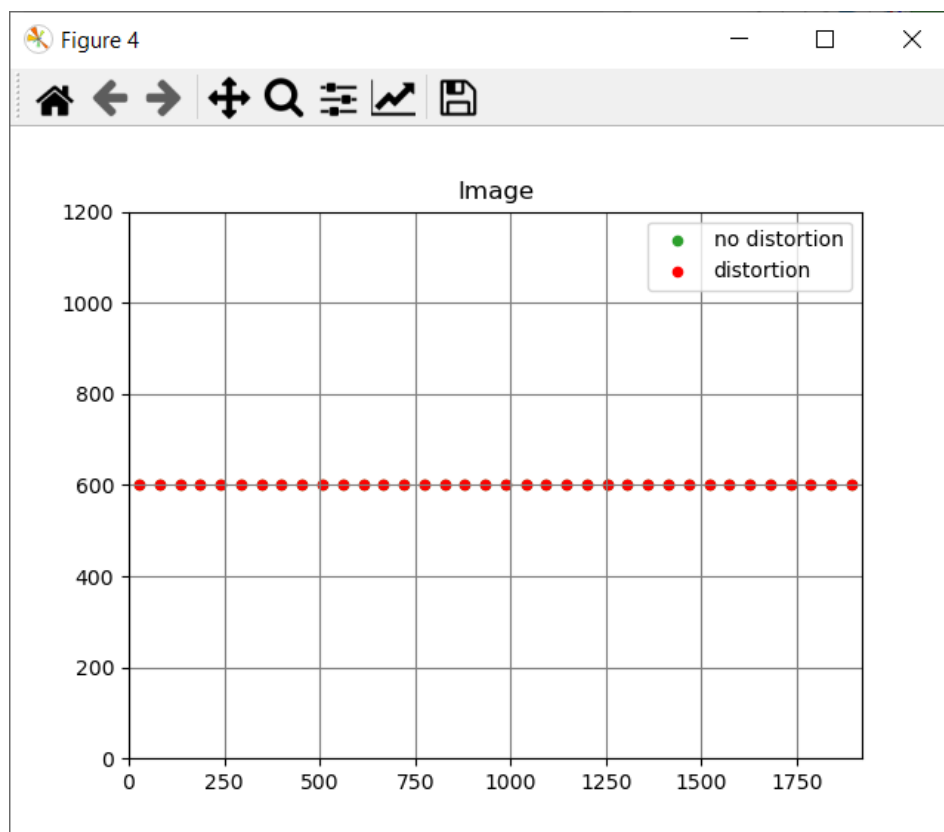
The image shows a text editor window titled "Camera calibration matrix - Blok za pi...". The window contains a 3x3 matrix of floating-point numbers in scientific notation. The status bar at the bottom indicates the current position is "Rd 1, St 1", the zoom is "100%", the line ending is "Windows (CRLF)", and the encoding is "UTF-8".

```
Datoteka Uređivanje Oblikovanje Prikaz Pomoć
2.66666667e+03 0.00000000e+00 9.60000000e+02
0.00000000e+00 2.66666667e+03 6.00000000e+02
0.00000000e+00 0.00000000e+00 1.00000000e+00
```

**Slika 19.** .txt datoteka kalibracijske matrice kamere

Slika 20 prikazuje sliku koju kamera vidi učitavanjem kalibracijske matrice iz .txt datoteke.





Slika 20. Slika kamere koristeći kalibracijsku matricu iz .txt datoteke

## 5.5. Najbitniji dijelovi programskog koda

U ovom poglavlju opisat će se najbitniji dijelovi koda vezani uz centralnu projekciju i vizualizaciju koje su opisane u teoretskom dijelu ovog završnog rada.

### 5.5.1. Kalibracijska matrica i matrica projekcije

Programski kod pomoću kojeg se računaju kalibracijska matrica i matrica projekcije je sljedeći:

```
# Calibration matrix
def get_calibration_matrix(
    f,
    px=0.0,
    py=0.0,
    pw=1.0,
    ph=1.0
):
    K = np.array([[f/pw, 0.0, px/pw], [0.0, f/ph, py/ph], [0.0, 0.0, 1.0]])
    return K

# Projection matrix - ideal
```

```

def get_projection_matrix(
    f,
    px=0.0,
    py=0.0,
    pw=1.0,
    ph=1.0,
    C=(0.0, 0.0, 0.0),
    theta_x=0.0,
    theta_y=0.0,
    theta_z=0.0
):
    K = get_calibration_matrix(f=f, px=px, py=py, pw=pw, ph=ph)
    R = get_rotation_matrix(theta_x=theta_x, theta_y=theta_y, theta_z=theta_z).T
    P = K @ R @ np.c_[np.eye(3), -np.asarray(C)]
    return P

# Projection matrix with real calibration matrix
def get_projection_matrix_2(
    K,
    C=(0.0, 0.0, 0.0),
    theta_x=0.0,
    theta_y=0.0,
    theta_z=0.0,
):
    R = get_rotation_matrix(theta_x=theta_x, theta_y=theta_y, theta_z=theta_z).T
    P = K @ R @ np.c_[np.eye(3), -np.asarray(C)]
    return P

```

Kod kalibracijske matrice,  $f$  označuje žarišnu duljinu kamere,  $px$  i  $py$  koordinate glavne točke na senzoru kamere u mm, a  $pw$  i  $ph$  širinu i visinu piksela. Matrica projekcije dobije se zadavanjem svih intrinzičnih i ekstrinzičnih parametara ili učitavanjem gotove kalibracijske matrice uz zadavanje ekstrinzičnih parametara. Intrinzični parametri su prethodno navedeni parametri kod kalibracijske matrice dok su ekstrinzični parametri koordinate ishodišta kamere  $C$  i Eulerovi kutovi  $\theta_x$ ,  $\theta_y$  i  $\theta_z$ .

### 5.5.2. Matrica rotacije

```

# Roll matrix Rx
def _get_roll_matrix(theta_x):
    Rx = np.array(
        [
            [1.0, 0.0, 0.0],
            [0.0, np.cos(theta_x), -np.sin(theta_x)],
            [0.0, np.sin(theta_x), np.cos(theta_x)]
        ]
    )
    return Rx

# Pitch matrix Ry
def _get_pitch_matrix(theta_y):
    Ry = np.array(

```

```

        [
            [np.cos(theta_y), 0.0, np.sin(theta_y)],
            [0.0, 1.0, 0.0],
            [-np.sin(theta_y), 0.0, np.cos(theta_y)]
        ]
    )
    return Ry

# Yaw matrix Rz
def _get_yaw_matrix(theta_z):
    Rz = np.array(
        [
            [np.cos(theta_z), -np.sin(theta_z), 0.0],
            [np.sin(theta_z), np.cos(theta_z), 0.0],
            [0.0, 0.0, 1.0]
        ]
    )
    return Rz

# Rotation matrix
def get_rotation_matrix(theta_x=0.0, theta_y=0.0, theta_z=0.0):
    # Roll matrix
    Rx = _get_roll_matrix(theta_x)
    # Pitch matrix
    Ry = _get_pitch_matrix(theta_y)
    # Yaw matrix
    Rz = _get_yaw_matrix(theta_z)
    return Rz @ Ry @ Rx

```

Pomoću programskog koda iznad dobiju se matrice valjanja, posrtanja i skretanja te pomoću njih matrica rotacije.

### 5.5.3. Plückerova matrica i matrica ravnine

```

# Plucker matrix
def get_plucker_matrix(A, B):
    A = to_homogeneous(A)
    B = to_homogeneous(B)
    L = A.reshape(-1, 1) * B.reshape(1, -1) - B.reshape(-1, 1) * A.reshape(1, -1)
    return L

# Calculate the plane equation from three points
def get_plane_from_three_points(X1, X2, X3):
    pi = np.hstack([np.cross(X1 - X3, X2 - X3), -X3 @ np.cross(X1, X2)])
    return pi

```

Zadavanjem dviju točaka u prostoru A i B (točka na laserskoj liniji i ishodište koordinatnog sustava kamere), dobije se Plückerova matrica koja, kao što je objašnjeno u teoretskom dijelu, predstavlja pravac u prostoru. Zadavanjem točaka X1, X2, X3 dobije se matrica ravnine u

prostoru. Pomoću ovih dviju matrica, kao što je već spomenuto, moguće je vizualizirati sjecište pravca s ravninom, tj. prikazati centralnu projekciju na 3D grafu.

#### 5.5.4. Jednadžba za generiranje seta laserskih točaka

```
# Function to get the laser points at the end of the laser plane
def get_points(self, n):
    list_points = []
    T1 = self.origin + (self.length/np.cos(self.fi/2)) * self.dy1
    T2 = self.origin + (self.length/np.cos(-self.fi/2)) * self.dy2
    list_points.append(T1)
    s = T2 - T1 # T1T2
    r1 = T1 # OT1
    ds = 1 / (n - 1)

    for i in range(0, n - 1):
        r = r1 + ds * s
        list_points.append(r)
        ds = ds + 1 / (n - 1)

    points = np.asarray(list_points)

    return points
```

Navedeni kod služi za generiranje seta točaka koje predstavljaju lasersku liniju. Točke se dobiju pomoću jednadžbe pravca u prostoru objašnjenom u teoretskom dijelu završnog rada. Oznaka  $f_i$  predstavlja kut širenja snopa svjetlosti lasera,  $\text{self.origin}$  ishodište lasera, a  $\text{dy1}$  i  $\text{dy2}$  su vektori pomoću kojih se dobiju krajnje točke na laserskoj liniji (između njih je kut  $f_i$ ).

#### 5.5.5. Jednadžbe za računanje centralne projekcije

```
# Calculate 2D laser points in pixels from 3D points in mm
def calculate_2D(
    point,
    f,
    px=0.0,
    py=0.0,
    pw=1.0,
    ph=1.0,
    C=(0.0, 0.0, 0.0),
    theta_x=0.0,
    theta_y=0.0,
    theta_z=0.0
):
    P = get_projection_matrix(
        f,
        px=px,
        py=py,
        pw=pw,
        ph=ph,
        theta_x=theta_x,
        theta_y=theta_y,
        theta_z=theta_z,
```

```

        C=C,
    )

    x = to_inhomogeneous(P @ to_homogeneous(point))

    return x

# Calculate 2D laser points in pixels from 3D points in mm with camera lens
# distortion
def calculate_2D_with_distortion(
    point,
    k1,
    k2,
    k3,
    p1,
    p2,
    px,
    py,
    f,
    pw,
    ph
):
    x_n = (point[0] - px)/(f/pw)
    y_n = (point[1] - py)/(f/ph)
    r2 = x_n**2 + y_n**2

    x_dist = x_n*(1 + k1*r2 + k2*(r2**2) + k3*(r2**3)) + (2*p1*x_n*y_n + p2*(r2 +
2*(x_n**2)))
    y_dist = y_n*(1 + k1*r2 + k2*(r2**2) + k3*(r2**3)) + (p1*(r2 + 2*(y_n**2)) +
2*p2*x_n*y_n)

    x_d = (x_dist * (f/pw)) + px
    y_d = (y_dist * (f/ph)) + py

    point_dist = np.array([x_d, y_d])

    return point_dist

# Calculate 2D laser points in pixels from 3D points in mm with a camera matrix
# loaded from a txt file
def calculate_2D_w_camera_matrix(
    point,
    K,
    C=(0.0, 0.0, 0.0),
    theta_x=0.0,
    theta_y=0.0,
    theta_z=0.0,
):
    P = get_projection_matrix_2(
        K,
        theta_x=theta_x,
        theta_y=theta_y,
        theta_z=theta_z,
        C=C,
    )

```

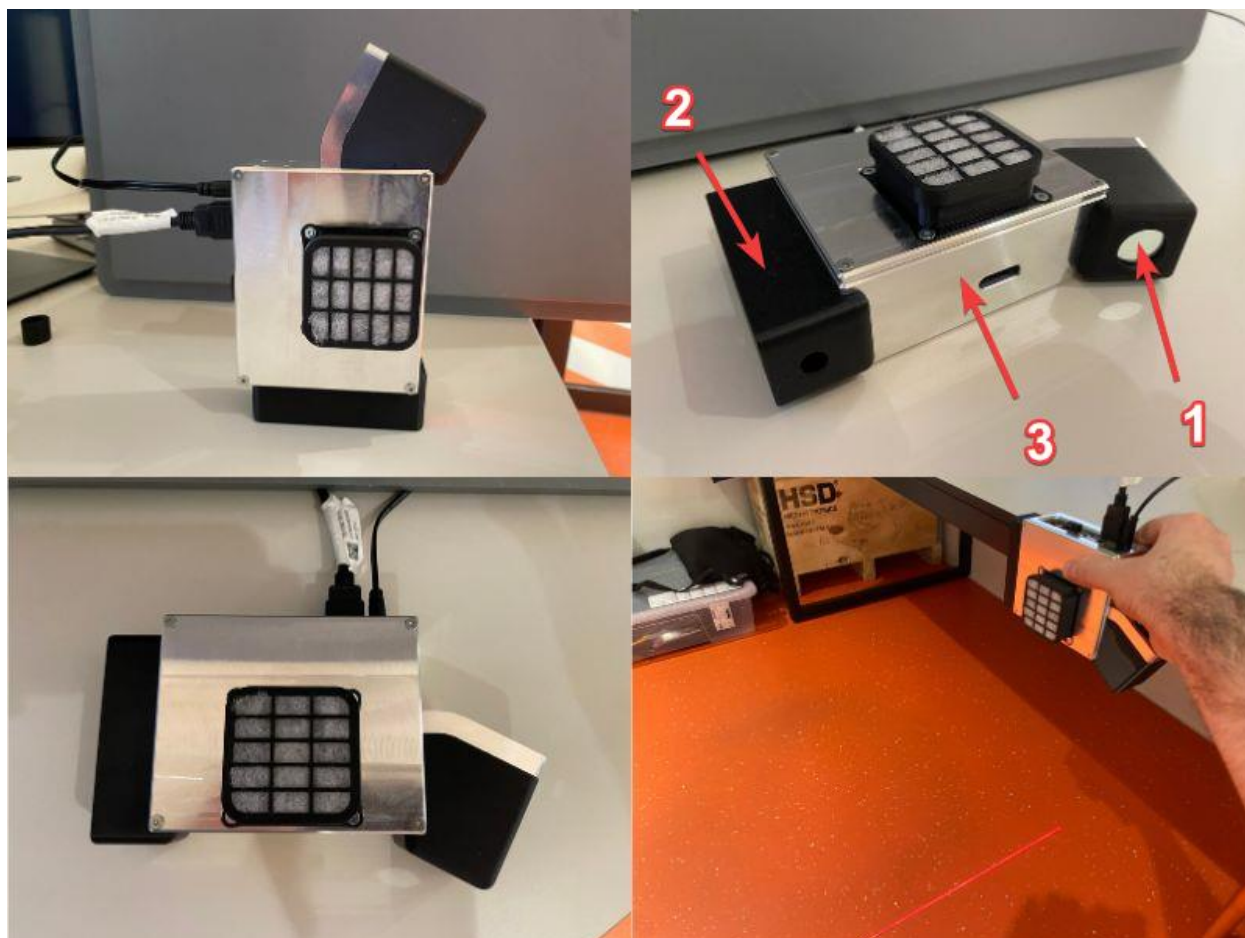
```
x = to_inhomogeneous(P @ to_homogeneous(point))  
return x
```

Navedenim programskim kodom dobiju se projicirane točke na slici kamere. Radi usporedbe, programirane su dvije funkcije: jedna za dobivanje idealnih točaka, a jedna za dobivanje točaka uz distorziju na temelju distorzijskog modela opisanog u teorijskom dijelu. Koeficijenti radijalne distorzije su označeni s  $k_1$ ,  $k_2$  i  $k_3$ , a koeficijenti tangencijalne distorzije s  $p_1$  i  $p_2$ . Zadnja funkcija služi za računanje točaka pomoću gotove kalibracijske matrice koju je moguće učitati iz .txt datoteke.

## 6. PRAKTIČNA ANALIZA IMPLEMENTIRANOG RJEŠENJA

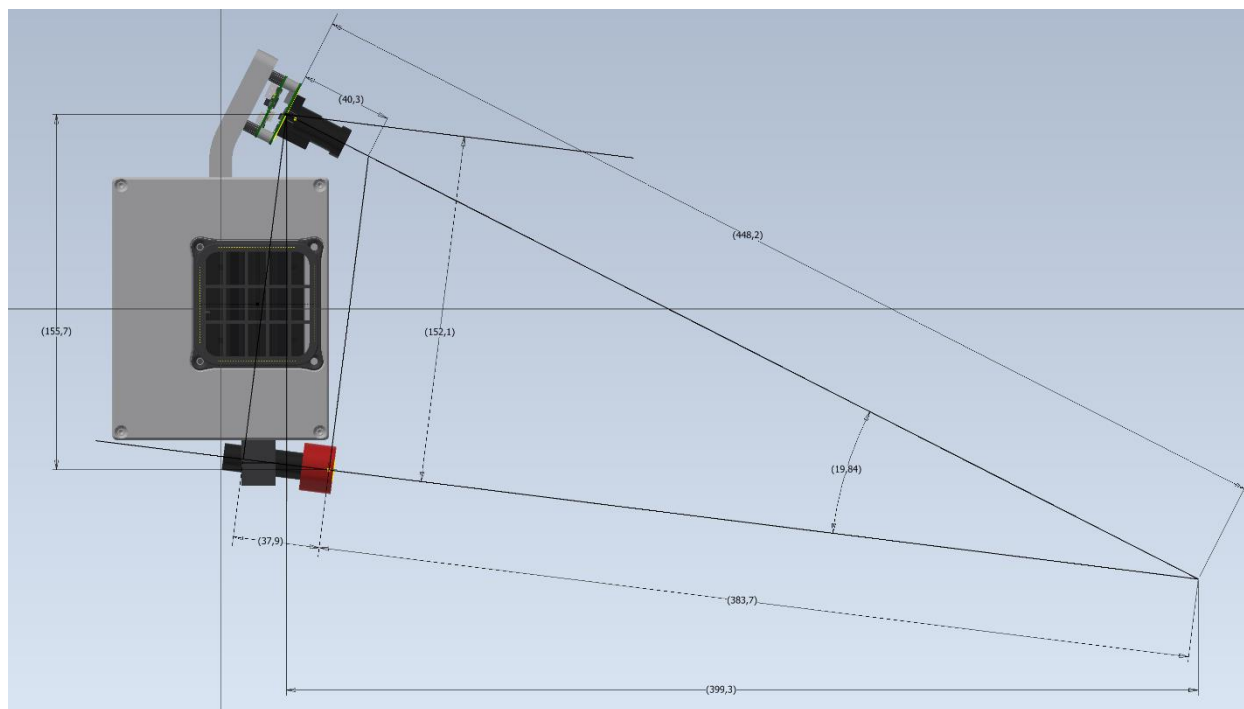
Zadnje poglavlje ovog završnog rada odnosi se na eksperimentalnu provjeru izrađenog programa, tj. usporedbu programa i realnog kamera-laser sustava. Pomoću realnog sustava snimit će se određeni broj slika laserske linije projicirane na ravnu površinu na različitim udaljenostima. Navedene slike usporedit će se sa slikama dobivenim pomoću programa koristeći iste parametre kao i realan sustav. Sustav koji će se koristiti nazvan je DRASER i prikazan je na Slika 21. Također, na slici su brojevima označene glavne komponente kamera-laser sustava:

1. kamera
2. laser
3. SBC (*single-board computer*)



Slika 21. Kamera-laser sustav DRASER

Na Sliku 22 prikazan je nacrt navedenog sustava.



**Slika 22. Nacrt kamera-laser sustava DRASER**

Iz navedenog nacrtu moguće je očitati potrebne podatke za prostorni odnos kamere i lasera, tj. ekstrinzične parametre potrebne za izračun matrice projekcije. Vertikalna udaljenost kamere u odnosu na laser iznosi 152.1 mm, a kut između njih iznosi 19.84°. To znači da je kameru potrebno zarotirati za 70.16° stupnjeva oko njene poprečne koordinatne osi. Da bi se laserska linija prikazala točno na sredini slike kamere, udaljenost sustava od ravne površine mora biti 421.6 mm, kao što je vidljivo iz nacrtu.

Tablica 1 prikazuje najbitnije tehničke karakteristike kamere koje je potrebno unijeti u program.

**Tablica 1. Tehničke karakteristike kamere potrebne za unos u program**

Žarišna duljina	8 mm
Rezolucija	1920×1200
Format senzora	1/2.6 inča
Veličina piksela	3.0 μm



Navedeni podaci koristit će za izračun kalibracijske matrice te ujedno time i matrice projekcije. Format senzora prikazan je u inčima, a u mm njegova širina iznosi 5.76, a visina 3.6.

Zadnji bitan podatak koji nije dan u tehničkim karakteristikama sustava je kut širenja snopa svjetlosti lasera koji iznosi  $45^\circ$ .

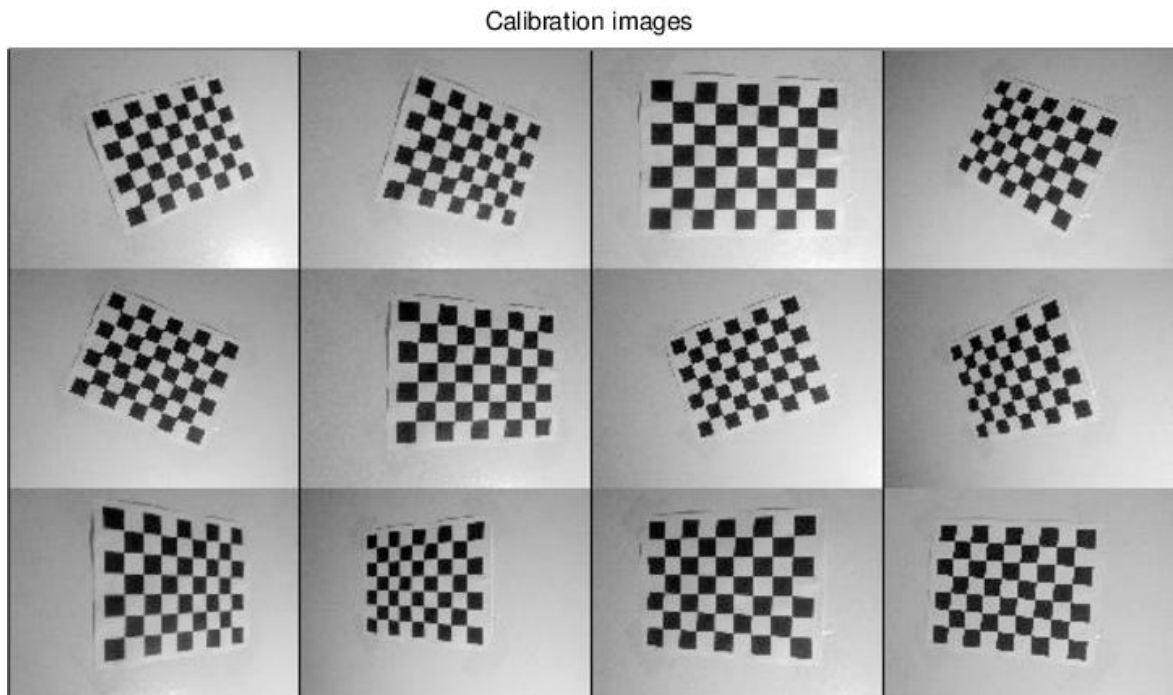
### 6.1. Kalibracija kamere

Posljednji podaci koji su potrebni za usporedbu stvarnog modela s idealnim su distorzijski parametri i stvarna kalibracijska matrica kamere. Njih je moguće dobiti kalibracijom kamere, postupkom kojim se određuju intrinzični i/ili ekstrinzični parametri kamere te distorzijski parametri definirani u teorijskom poglavlju rada. U ovom radu koristit će se kalibracija kamere pomoću kalibracijskog uzorka šahovnice poznatih dimenzija. Dimenzija šahovnice koja će se koristiti je  $7 \times 9$  presjeka kvadratića, a svaki kvadratić ima dimenziju  $20 \times 20$  mm. Šahovnica je prikazana na Slika 23.



Slika 23. Šahovnica za kalibraciju kamere

Potrebno je snimiti između 10 i 20 slika različitih orijentacija šahovnice kao što je prikazano na Slika 24.



Slika 24. Primjer snimljenih slika šahovnice za kalibraciju

Kalibracija će se provesti u *Pythonu* koristeći *OpenCV* biblioteku za računalni vid koja sadrži potrebne funkcije za provedbu navedene kalibracije. Programski kod korišten za kalibraciju je sljedeći:

```
import numpy as np
import cv2
import glob

# Define the size of the checkerboard corners
CHECKERBOARD_SIZE = (7, 9)

# Define the criteria for subpixel corner detection
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 30, 0.001)

# Prepare object points
objp = np.zeros((CHECKERBOARD_SIZE[0] * CHECKERBOARD_SIZE[1], 3), np.float32)
objp[:, :2] = np.mgrid[0:CHECKERBOARD_SIZE[0], 0:CHECKERBOARD_SIZE[1]].T.reshape(-1, 2)

# Create empty arrays for object points and image points from all images
objpoints = [] # 3D points in real world space
imgpoints = [] # 2D points in image plane
```

```
# Load calibration images
images = glob.glob('calibration_images/*.jpg')

# Loop through images and find checkerboard corners
for fname in images:
    # Load image and convert to grayscale
    img = cv2.imread(fname)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Find checkerboard corners
    ret, corners = cv2.findChessboardCorners(gray, CHECKERBOARD_SIZE, None)

    # If corners found, refine the corner location
    if ret == True:
        objpoints.append(objp)
        corners = cv2.cornerSubPix(gray, corners, (11, 11), (-1, -1), criteria)
        imgpoints.append(corners)

        # Draw and display the corners
        cv2.drawChessboardCorners(img, CHECKERBOARD_SIZE, corners, ret)
        cv2.imshow('img', img)
        cv2.waitKey(500)

# Perform camera calibration
ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoints,
gray.shape[::-1], None, None)

# Print calibration parameters
print("Camera matrix:")
print(mtx)
print("Distortion coefficients:")
print(dist)

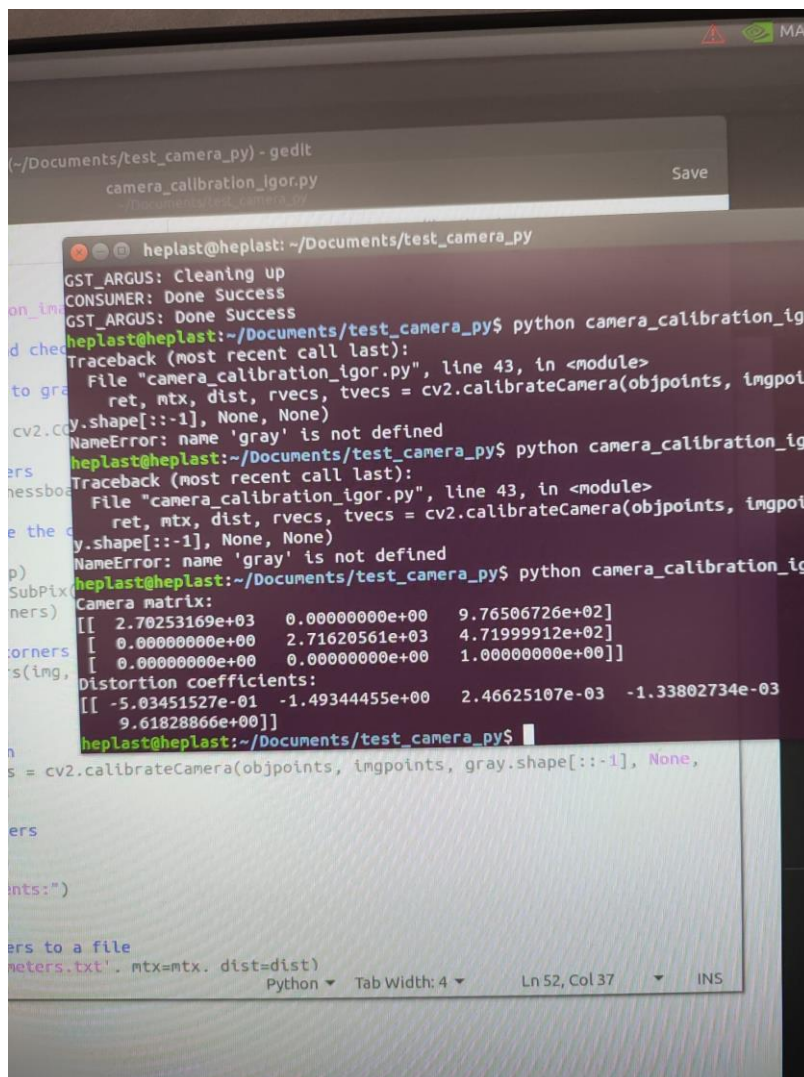
# Save calibration parameters to a file
np.savez('calibration_parameters.npz', mtx=mtx, dist=dist)
```

Opis programskog koda je sljedeći:

1. Na početku programa uvode se potrebne biblioteke *Numpy*, *cv2* (*OpenCV*) i *glob*.
2. Definišu se veličina šahovske ploče (*CHECKERBOARD\_SIZE*) i kriteriji za detekciju podpikselnih kutova (*criteria*).
3. Kreira se matrica *objp* koja predstavlja 3D koordinate kutova unutar šahovske ploče koje će se koristiti kao referentne točke tijekom kalibracije kamere.
4. Učitavaju se slike iz direktorija *calibration\_images*. Za svaku sliku se pokušavaju detektirati kutovi šahovske ploče. Ako su kutovi uspješno detektirani, dodaju se u *objpoints* i *imgpoints* liste.
5. Poziva se funkcija *cv2.calibrateCamera* s objektima i slikama kako bi se izračunali parametri kamere (*mtx* – kalibracijska matrica, *dist* – distorzijski koeficijenti) te rotacijski i translacijski vektori.

6. Ispisuju se matrica kamere i koeficijenti distorzije.
7. Parametri kamere se spremaju u datoteku calibration\_parameters.npz.

Na Slika 25 su prikazani rezultati dobiveni kalibracijom.



```
heplast@heplast: ~/Documents/test_camera_py
GST_ARGUS: Cleaning up
CONSUMER: Done Success
GST_ARGUS: Done Success
heplast@heplast:~/Documents/test_camera_py$ python camera_calibration_ig
Traceback (most recent call last):
  File "camera_calibration_igor.py", line 43, in <module>
    ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoint
    y.shape[:-1], None, None)
NameError: name 'gray' is not defined
heplast@heplast:~/Documents/test_camera_py$ python camera_calibration_ig
Traceback (most recent call last):
  File "camera_calibration_igor.py", line 43, in <module>
    ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoint
    y.shape[:-1], None, None)
NameError: name 'gray' is not defined
heplast@heplast:~/Documents/test_camera_py$ python camera_calibration_ig
Camera matrix:
[[ 2.70253169e+03  0.00000000e+00  9.76506726e+02]
 [ 0.00000000e+00  2.71620561e+03  4.71999912e+02]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Distortion coefficients:
[[ -5.03451527e-01  -1.49344455e+00  2.46625107e-03  -1.33802734e-03
   9.61828866e+00]]
heplast@heplast:~/Documents/test_camera_py$
```

Slika 25. Distorzijski parametri i kalibracijska matrica dobiveni kalibracijom

Iznos idealne kalibracijske matrice za kamera-laser sustav DRASER prikazan je na Slika 26, a dobiven je pomoću programa.

```
[ [2.66666667e+03 0.00000000e+00 9.60000000e+02 ]  
 [0.00000000e+00 2.66666667e+03 6.00000000e+02 ]  
 [0.00000000e+00 0.00000000e+00 1.00000000e+00 ] ]
```

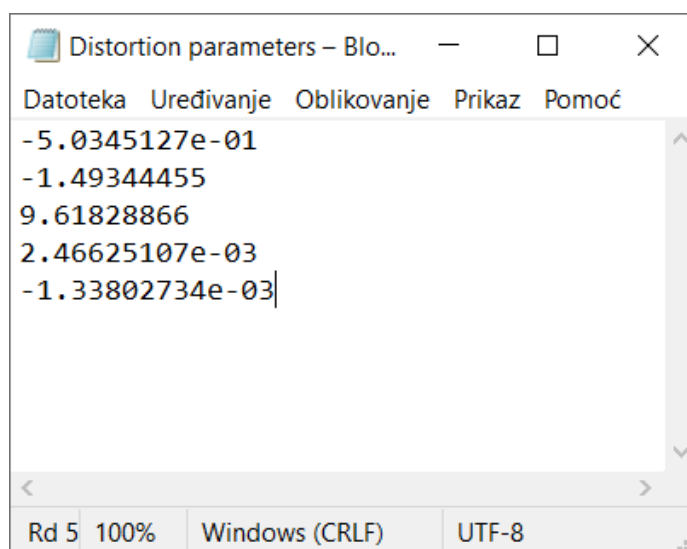
Slika 26. Iznos idealne kalibracijske matrice

Vidljivo je da odstupanja realne matrice od idealne nisu velika. Najveće odstupanje iznosi 21% što je i dalje prihvatljiv iznos.

Pomoću dobivenih distorzijskih parametara i kalibracijske matrice dalje je moguće napraviti usporedbu realnog sustava i programa.

## 6.2. Usporedba realnog sustava i programa

Usporedba će se provesti na način da će se snimiti pet slika laserske linije na različitim udaljenostima od ravne površine. Zatim će se te slike usporediti sa slikama koje su generirane pomoću programa koristeći iste parametre kao i realni sustav. Snimit će se slike na udaljenostima 680, 550, 350, 280 i 250 mm od ravne površine. Kalibracijom dobiveni distorzijski parametri spremljeni su u .txt datoteku pod nazivom *Distortion parameters* koja je prikazana na Slika 27.



Slika 27. .txt datoteka kalibracijom dobivenih distorzijskih parametara

Navedenu datoteku potrebno je učitati u program (ili ručno unijeti). Okvir sa svim unesenim parametrima prikazan je na Slika 28.

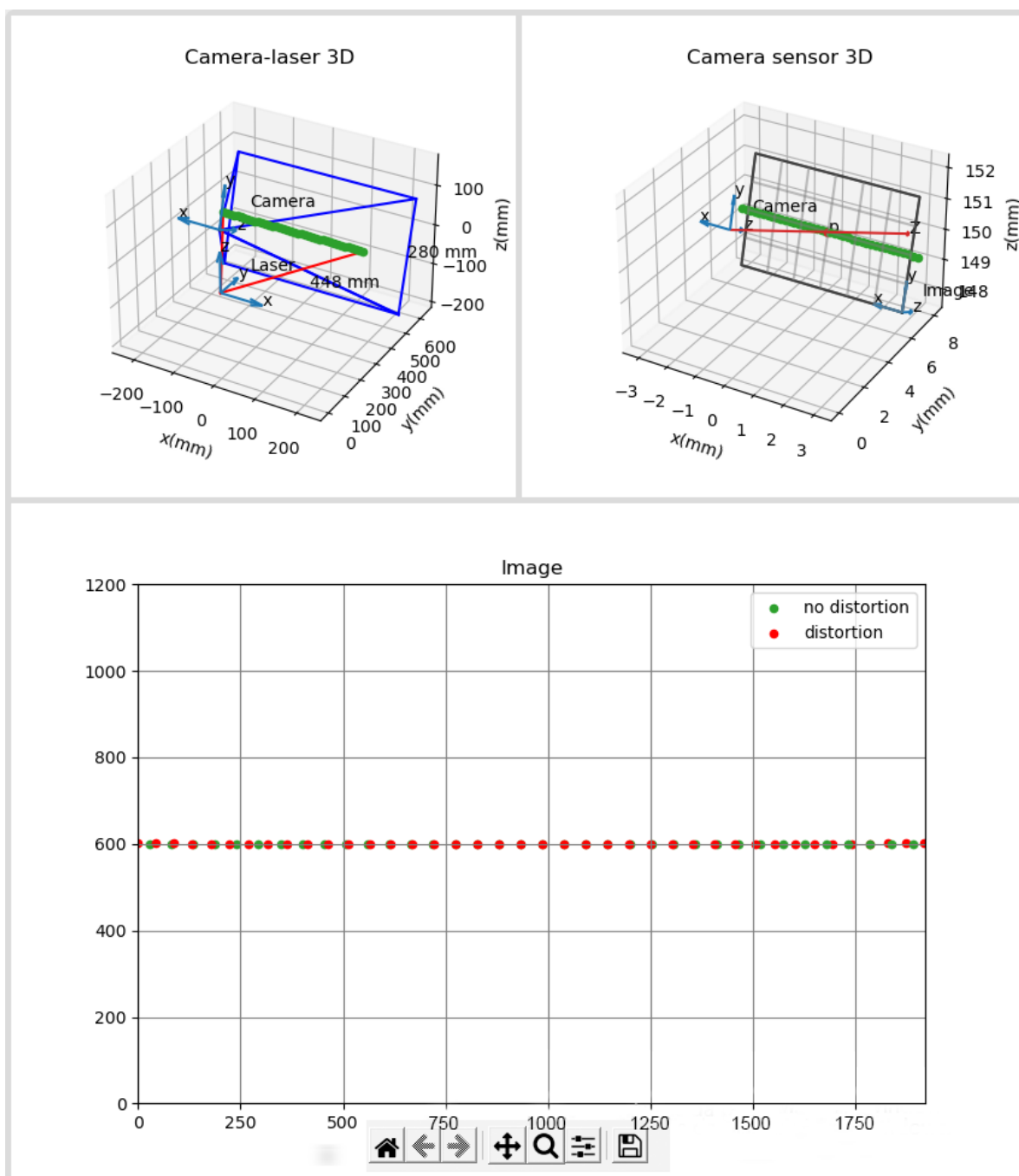
The image shows a software interface with several panels for parameter input:

- Laser parameters:** Laser distance (mm): 421.6, Laser angle (°): 45.
- Sensor parameters:** Sensor width (mm): 5.76, Sensor height (mm): 3.6, Pixel width (mm): 3e-03, Pixel height (mm): 3e-03, Principal point x (mm): 2.88, Principal point y (mm): 1.8.
- Camera matrix:** (Label above the sensor parameters panel).
- Camera parameters:** Focal length (mm): 8, Camera displacement x (mm): 0, Camera displacement y (mm): 0, Camera displacement z (mm): 152.1, Camera Euler angle x (°): -70.16, Camera Euler angle y (°): 180, Camera Euler angle z (°): 0.
- Distortion parameters:** k1: -5.0345127e-01, k2: -1.49344455, k3: 9.61828866, p1: 2.46625107e-03, p2: -1.33802734e-03. Includes a "Load from txt file" button.
- Image resolution:** Image width (px): 1920, Image height (px): 1200. Includes a "Graph" button.

Slika 28. Okvir sa unesenim parametrima

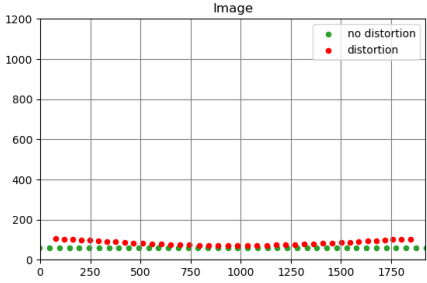
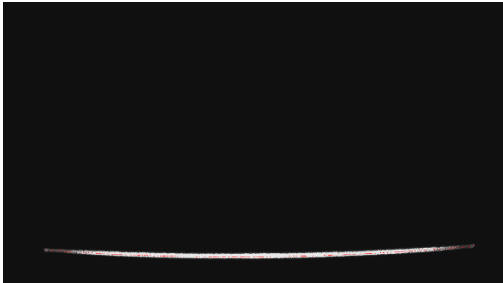
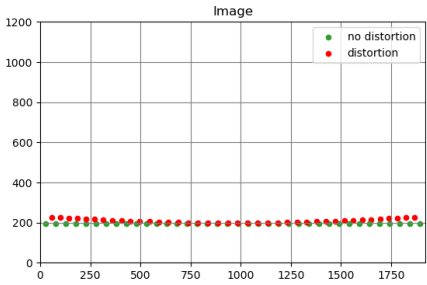
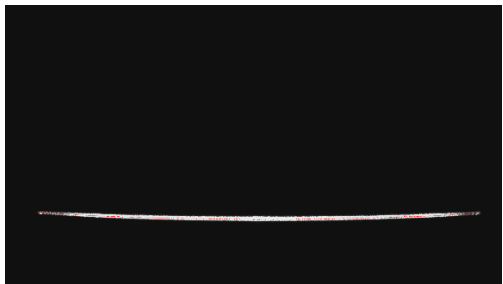
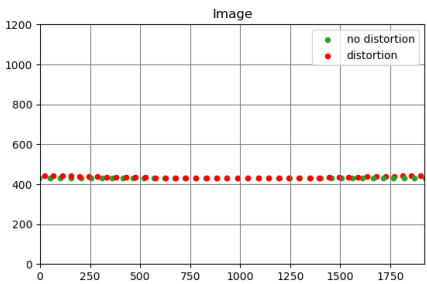
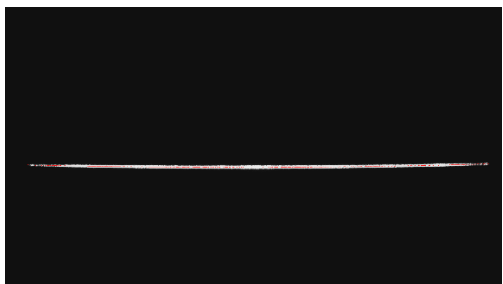
Svi parametri su dobiveni na temelju tehničkih podataka predstavljenim u prethodnim poglavljima. Jedino što je preostalo napomenuti je da je kameru potrebno zarotirati za 180 stupnjeva oko njene y-osi kako bi 2D slika koju kamera vidi bila u skladu s njenim prikazom u 3D-u radi lakšeg pregleda. Parametar koji će se mijenjati je *laser distance*.

Slika 29 prikazuje grafove generirane pomoću navedenih parametara, a Tablica 2 usporedbu slika dobivenih pomoću realnog sustava i slika dobivenih programom za različite udaljenosti lasera od ravne površine.

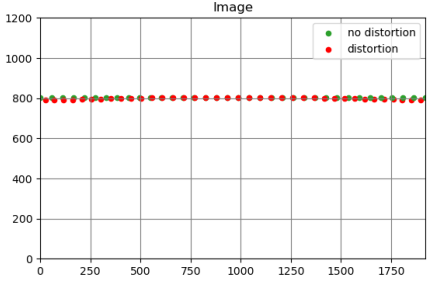
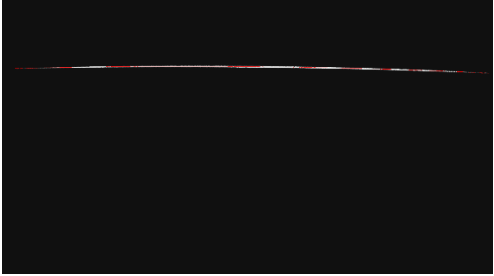
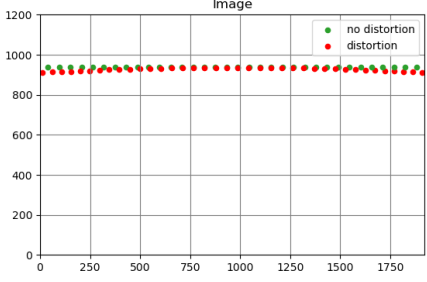
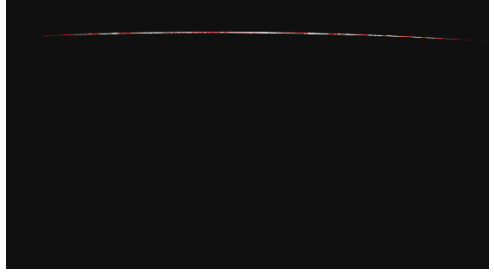


Slika 29. Grafovi generirani na temelju idealnih parametara kamera-laser sustava

Tablica 2. Usporedba rezultata programa i realnog sustava

Laser distance (mm)	Program	Realni sustav
250		
280		
350		



550		
680		

Kao što je vidljivo, idealni model vjerno predstavlja realni. Bitno je napomenuti da kod uzimanja slika realnim sustavom, udaljenosti nisu u potpunosti precizne, već su aproksimirane.

## 7. ZAKLJUČAK

U ovom završnom radu obrađena je teorija potrebna za izrađivanje modela kamera-laser sustava koji omogućuje brzo testiranje različitih parametara navedenog sustava. Objasnjeni su pojmovi poput kalibracijske matrice, matrice projekcije, matrice rotacije i centralne projekcije potrebni za shvaćanje kako se objekti iz stvarnog svijeta projiciraju na sliku koju kamera vidi. Utvrđeno je da je za vjerodostojan model kamere potrebno definirati intrinzične i ekstrinzične parametre poput žarišne duljine, translacije i rotacije kamere u odnosu na fiksni koordinatni sustav, širine i visine piksela kao i koordinate glavne točke. Uz to, za kameru je potrebno definirati njenu rezoluciju, širinu i visinu senzora te distorzijske parametre koji određuju zakrivljenost slike koju kamera vidi. Kod modela laserske linije potrebno je poznavati njezin kut širenja snopa svjetlosti kao i udaljenost od površine koja je u ovom radu pretpostavljena ravnom. Model je u završnom poglavlju rada uspješno eksperimentalno provjeren čime je dokazana njegova valjanost te ispravno programiranje.

---

## LITERATURA

- [1] Edl M, Mizerák M, Trojan J. 3D LASER SCANNERS: HISTORY AND APPLICATIONS. Acta Simulatio 2018;4:1–5. <https://doi.org/10.22306/asim.v4i4.54>.
- [2] Tunjić B. 3D skeniranje-Teorijska i praktična primjena.
- [3] 3D Skeneri | TOPOMATIKA trodimenzionalno skeniranje, optički mjerni sustavi i računalna obrada d.o.o. n.d. <https://topomatika.hr/proizvodi/3d-skeneri/> (29.8.2023.).
- [4] FARO Focus Laser Scanner| Hardware | FARO n.d. <https://www.faro.com/en/Products/Hardware/Focus-Laser-Scanners> (30.8.2023.).
- [5] Hata K, Savarese S. CS231A Course Notes 1: Camera Models.
- [6] Corke P. Robotics, Vision and Control. vol. 118. Cham: Springer International Publishing; 2017. <https://doi.org/10.1007/978-3-319-54413-7>.
- [7] Hartley R, Zisserman A. Multiple view geometry in computer vision.
- [8] 3D Rotations-Part 1  
[https://danceswithcode.net/engineeringnotes/rotations\\_in\\_3d/rotations\\_in\\_3d\\_part1.html](https://danceswithcode.net/engineeringnotes/rotations_in_3d/rotations_in_3d_part1.html)  
(21.7.2023.).
- [9] Camera Modeling: Exploring Distortion and Distortion Models, Part I  
<https://www.tangramvision.com/blog/camera-modeling-exploring-distortion-and-distortion-models-part-i#tangential-de-centering-distortions> (27.7.2023.).
- [10] Pravac <http://www.mathematics.digital/matematika1/predavanja/node63.html>  
(26.7.2023.).
- [11] Simple Camera Models with NumPy and Matplotlib | by Mario Namtao Shianti Larcher | Analytics Vidhya | Medium <https://medium.com/analytics-vidhya/simple-camera-models-with-numpy-and-matplotlib-92281f15f9b2> (21.7.2023.).

## **PRILOG**

Programski kod:

<https://github.com/IgorShijak/Camera-laser-Python-math-model-.git>