

# Rekonstrukcija položaja 3D objekta iz kontinuiranog izvora 2D fotografija

---

Škopelja, Ena

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:859204>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-13**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



**SVEUČILIŠTE U ZAGREBU**  
**PRIRODOSLOVNO–MATEMATIČKI FAKULTET**  
**MATEMATIČKI ODSJEK**

Ena Škopelja

**REKONSTRUKCIJA POLOŽAJA 3D  
OBJEKTA IZ KONTINUIRANOG  
IZVORA 2D FOTOGRAFIJA**

Diplomski rad

Voditelj rada:  
doc. dr. sc. Goranka Nogo

Zagreb, 2022

Ovaj diplomski rad obranjen je dana \_\_\_\_\_ pred ispitnim povjerenstvom u sastavu:

1. \_\_\_\_\_, predsjednik
2. \_\_\_\_\_, član
3. \_\_\_\_\_, član

Povjerenstvo je rad ocijenilo ocjenom \_\_\_\_\_.

Potpisi članova povjerenstva:

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_

# Sadržaj

<b>Sadržaj</b>	<b>iii</b>
<b>Uvod</b>	<b>1</b>
<b>1 Osnovni koncepti</b>	<b>2</b>
1.1 Homogene koordinate i projekcijske transformacije . . . . .	2
1.2 Model kamere . . . . .	3
1.3 Problem najmanjih kvadrata . . . . .	5
<b>2 Opis problema</b>	<b>12</b>
<b>3 Pregled metoda estimacije 3D položaja iz projekcije u 2D prostoru</b>	<b>14</b>
3.1 Iz jednog pogleda . . . . .	14
3.2 Iz više pogleda . . . . .	14
<b>4 Implementacijski detalji</b>	<b>18</b>
4.1 Biblioteka Tensorflow . . . . .	20
<b>5 Rezultati</b>	<b>22</b>
5.1 Testni podaci . . . . .	22
5.2 Metrike . . . . .	23
5.3 Odabir optimizatora gradijentnog spusta . . . . .	24
5.4 Otpornost na pogrešku u mjerenju . . . . .	26
<b>Bibliografija</b>	<b>31</b>

# Uvod

U posljednje vrijeme obrada fotografija i posebno lokalizacija objekata na fotografijama predmet su intenzivnih istraživanja. Pojavom dubokog učenja u tom polju napravljeni su veliki pomaci, a danas se industrijski proizvodi sve više oslanjaju na takva rješenja, primjerice u medicini, automobilske industriji ili velikim industrijskim postrojenjima. S nedavnim razvojem tehnologija virtualne i proširene stvarnosti lokalizacija na fotografijama pokazuje se nedostatna zbog manjka prostorne, tj. dubinske informacije. Bez dubinske percepcije nije moguće interagirati s objektima od interesa, primjerice nije moguće postaviti niti zadržati virtualne objekte relativno s njima. Jedan od načina na koji se nastoji doskočiti tom problemu je korištenjem uređaja LiDAR. Njime se omogućuje to da se uz fotografije, kao prostorna informacija, uzima i oblak točaka (eng. *point cloud*, skup točaka u trodimenzionalnom prostoru) promatrane scene. Primjer takvog skupa podataka može se pronaći u [6].

S druge strane, ukoliko je cilj lokalizirati statičan objekt koji se nalazi u ravnini u prostoru, pokazat ćemo kako je njegov položaj u trodimenzionalnom prostoru moguće rekonstruirati koristeći samo niz njegovih fotografija iz različitih kuteva. Za svaku od fotografija standardnim metodama dubokog učenja dobivaju se ključne točke objekta na fotografiji. Iz njih je, koristeći znanje o pozicijama fotografiranja, moguće rekonstruirati položaj traženog objekta u trodimenzionalnom prostoru. Takav pristup baza je daljnje diskusije.

U uvodnom poglavlju (1) razmatra se projektivno geometrijska pozadina problema i standardne optimizacijske tehnike kojima se nastoji ublažiti efekt nepreciznosti u procjenama lokacije objekata na fotografijama. Zatim se, u poglavlju 2, daje detaljniji opis problema i pregled standardnih metoda za 3D rekonstrukciju zajedno s prijedlogom nove metode kojom se osigurava da je rekonstruirani objekt zaista pravokutnog oblika u poglavlju 3.1. Nakon kratkog opisa prijedloga implementacije sustava u poglavlju 4 spomenute metode se evaluiraju na umjetno generiranim podacima u ovisnosti o pogreškama procjene položaja u dvodimenzionalnom prostoru i broju razmatranih fotografija objekta. Rezultate navodimo u poglavlju 5.

# Poglavlje 1

## Osnovni koncepti

U ovom poglavlju navodimo nekoliko osnovnih koncepata na kojima se baziraju promatrane metode. Na početku uvodimo homogene koordinate koje olakšavaju opis prostora i projektivnih transformacija. Homogene koordinate zatim koristimo u modeliranju kamere i detaljnije opisujemo korišteni model kamere. Naposljetku, kako se u kontekstu rješavanja problema rekonstrukcije u trodimenzionalnom prostoru pojavljuje potreba za rješavanjem problema najmanjih kvadrata, ukratko prezentiramo problem i metode njegovog rješavanja.

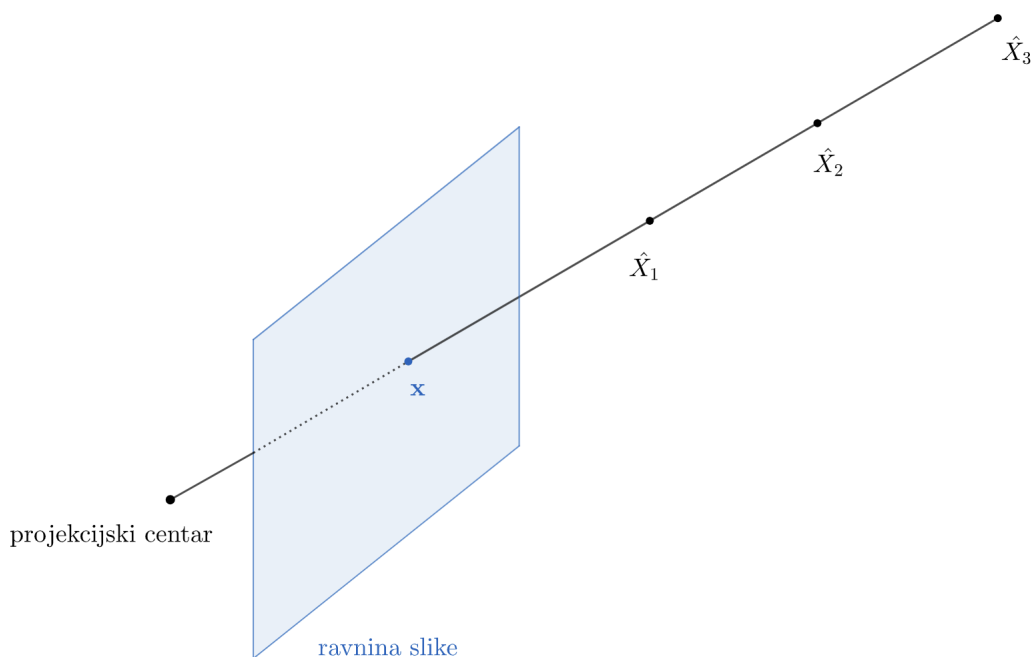
### 1.1 Homogene koordinate i projekcijske transformacije

Točku u Euklidskoj ravnini obično zapisujemo kao uređeni par realnih brojeva  $(x, y) \in \mathbb{R}^2$ . U projektivnoj geometriji pokazalo se korisnim dodati još i treću koordinatu,  $(x', y', z')$ . Takvom zapisu pridružujemo relaciju:

$$(x_1, y_1, z_1) \sim (x_2, y_2, z_2) \iff \left[ (z_1 = 0 \wedge z_2 = 0) \vee \left( \frac{x_1}{z_1}, \frac{y_1}{z_1} \right) = \left( \frac{x_2}{z_2}, \frac{y_2}{z_2} \right) \right]. \quad (1.1)$$

Lako je provjeriti kako je gornja relacija relacija ekvivalencije. Njene klase ekvivalencije identificiramo s točkama u  $\mathbb{R}^2$  na način da točki  $(x, y)$  odgovara klasa  $[(x, y, 1)]$ . Posebno, točke oblika  $(x, y, 0)$  nazivamo točkama u beskonačnosti, a za njih se kaže kako tvore tzv. pravac u beskonačnosti.

Prethodno opisani postupak lako se proširuje na  $\mathbb{R}^n$ , a rezultatni prostor naziva se **projektivni prostor**, ozn.  $\mathbb{P}^n$ . Nadalje, linearna transformacija Euklidskog prostora ostvaruje se djelovanjem linearnog operatora na vektor  $x \in \mathbb{R}^n$ , analogno tome, **projekcijska transformacija** projektivnog prostora ostvaruje se djelovanjem regularnog operatora na vektor u homogenim koordinatama. Takva transformacija dozvoljava mapiranje točaka u beskonačnosti u proizvoljne točke u projektivnom prostoru. [5]



Slika 1.1: Projekcija triju projektivno ekvivalentnih točaka ( $\hat{X}_1, \hat{X}_2, \hat{X}_3 \in \mathbb{P}^3$ ) u istu točku na ravnini slike ( $\mathbf{x} \in \mathbb{P}^2$ ).

## 1.2 Model kamere

Razumijevanje 3D strukture iz niza 2D fotografija zahtjeva formalniju definiciju procesa formiranja fotografija, točnije procesa formiranja dvodimenzionalne reprezentacije trodimenzionalnog prostora. Praktično, redukcija dimenzionalnosti ostvaruje se projekcijom, a uobičajeni način modeliranja tog procesa je centralna projekcija.

Kod centralne projekcije promatraju se fiksna **ravnina slike** te zraka u prostoru koja povezuje  $\hat{X}_i \in \mathbb{P}^3$ ,  $i \in \{1, 2, 3, 4\}$  s fiksnom točkom u prostoru koju nazivamo **centar projekcije**. Na sjecištu ravnine slike sa spomenutom zrakom nalazi se slika  $x$  promatrane točke  $\hat{X}_i$  (slika 1.2).

Najjednostavniji model kamere promatra zraku svjetla koja iz točke u prostoru prolazi kroz leću kamere te se registrira na senzoru kamere. Ovaj pristup pretpostavlja kako sve zrake svjetla prolaze točno kroz centar leće kamere, dakle ignorira fokus i debljinu leće [7]. Dodatno, u opisanom modelu kamere (eng. *pinhole model*) često se uzima još i kako se ravnina slike nalazi ispred centra projekcije što olakšava modeliranje projekcije jer izbjegava invertiranje konačne slike [11].

Iz aspekta projektivne geometrije, svijet se obično modelira kao 3D projektivni prostor

( $\mathbb{P}^3$  zajedno s točkama u beskonačnosti). Analogno, model slike je 2D projektivna ravnina  $\mathbb{P}^2$ . U tom kontekstu, centralna projekcija je mapiranje iz  $\mathbb{P}^3$  u  $\mathbb{P}^2$ . Odaberemo li za projekcijski centar točku  $(0, 0, 0, 1)^\top \in \mathbb{P}^3$  tada vrijedi:

$$(\forall x_1, x_2, x_3 \in \mathbb{R}) (\forall t \in \mathbb{R}) \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ t \end{bmatrix} \in \mathbb{P}^3 \longrightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathbb{P}^2 \quad (1.2)$$

tj. zadnja homogena koordinata ne utječe na lokaciju projekcije originalne slike. U tom slučaju djelovanje kamere možemo reprezentirati matricom  $M_{3 \times 4} \ni P = [I_{3 \times 3} \mid \mathbf{0}_3]$ .

Općenito (uz različite centre projekcije i relativne koordinatne sustave u prostoru slike) pokazuje se kako se kamera može prikazati kao proizvoljna matrica  $P \in M_{3 \times 4}$  ranga 3 [7]. Matricu  $P$  nazivamo još i **intrinzičnom matricom kamere**, ona opisuje komponentu projekcije koja ovisi o intrinzičnim parametrima kamere (npr. fokalnoj duljini).

Kod modeliranja kamere u realnom svijetu važno je uzeti u obzir i međusoban odnos centra kamere i centra koordinatnog sustava, taj problem je ilustriran slikom 1.2. Takav odnos modelira se **ekstrinzičnom matricom**, a produkt spomenutih dviju matrica nazivamo **(projekcijska) matrica kamere**.

**Ekstrinzična matrica kamere** u osnovi predstavlja promjenu baze koordinatnog sustava i kao takva sastoji se od rotacijske ( $E_R$ , 1.4) i translacijske ( $E_T$ , 1.3) komponente [11]:

$$E_T = \begin{bmatrix} 1 & 0 & 0 & x_T \\ 0 & 1 & 0 & y_T \\ 0 & 0 & 1 & z_T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.3)$$

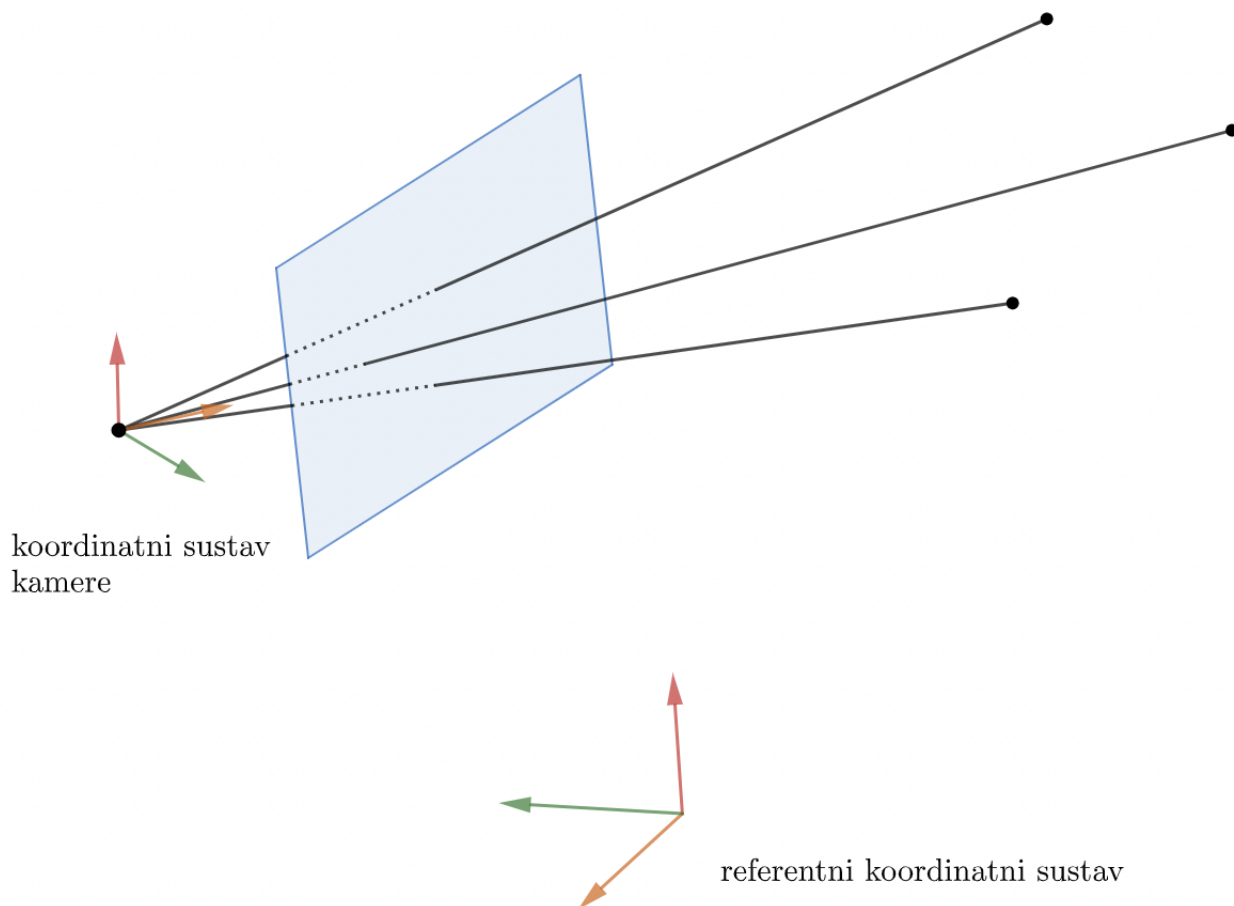
$$E_R = E_R^x(\phi_x) \times E_R^y(\phi_y) \times E_R^z(\phi_z) \quad (1.4)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi_x) & -\sin(\phi_x) & 0 \\ 0 & \sin(\phi_x) & \cos(\phi_x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.5)$$

$$\cdot \begin{bmatrix} \cos(\phi_y) & 0 & \sin(\phi_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\phi_y) & 0 & \cos(\phi_y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.6)$$

$$\cdot \begin{bmatrix} \cos(\phi_z) & -\sin(\phi_z) & 0 & 0 \\ \sin(\phi_z) & \cos(\phi_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \quad (1.7)$$





Slika 1.2: Pinhole model kamere.

### 1.3 Problem najmanjih kvadrata

Konačno, zbog nepreciznosti procjena položaja objekta u dvodimenzionalnom prostoru, pojavljuje se potreba za aproksimacijom trodimenzionalnog položaja iz niza nepreciznih procjena položaja u dvodimenzionalnom prostoru. Tom se problemu pristupa kao problemu najmanjih kvadrata, stoga u nastavku opisujemo problem i neka od mogućih pristupa njegovom rješavanju.

Pretpostavimo da imamo skup mjerenih podataka  $(x_i, y_i) \in \mathbb{R}^{n+1}$ ,  $i = 1, \dots, m$  koji želimo aproksimirati funkcijom oblika  $f(x)$ . Problem najmanjih kvadrata rješenje traži u parametrima  $p$  koji minimiziraju kvadratnu pogrešku  $E$ :

$$E = \sum_i \|r_i\|^2 = \sum_i \|f(x_i; \mathbf{p}) - y_i\|^2$$

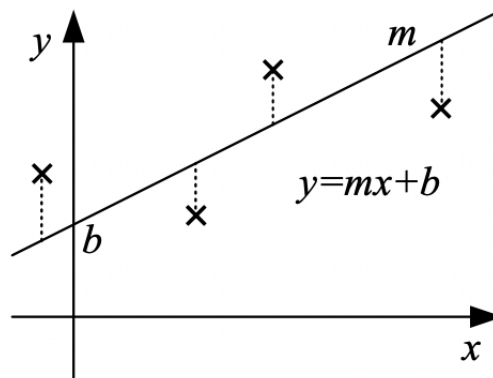
gdje se s

$$r_i = f(x_i; \mathbf{p}) - y_i = \hat{y}_i - y_i$$

označava **rezidual** između mjerenih podataka  $y_i$  i vrijednosti predviđene trenutnim modelom  $f(x_i; \mathbf{p})$  [11].

### Linearni problem najmanjih kvadrata

Kažemo da je problem najmanjih kvadrata linearan ukoliko je funkcija  $f(\mathbf{x}, \mathbf{p})$  linearna u  $\mathbf{p}$ . Primjerice ukoliko modeliramo uzorkovane  $(x_i, y_i) \in \mathbb{R}^2$  pravcem,  $f(\mathbf{x}, \mathbf{p})$  se svodi na  $mx + b$  (slika 1.3). Tako dobivenu grešku označavamo s  $E_{VLS}$  (eng. *vertical least squares*):



Slika 1.3: Problem linearnih najmanjih kvadrata na primjeru linearne funkcije jedne varijable [11]

$$E_{VLS} = \sum_i |y_i - (mx_i + b)|^2. \quad (1.8)$$

Općenitije, ako za model podataka uzmemo polinom stupnja  $p$  dobivamo grešku oblika  $E_{PLS}$  (eng. *polynomial least squares*) [11]:

$$E_{PLS} = \sum_i \left| y_i - \left( \sum_{k=0}^p a_j x_i^k \right) \right|^2. \quad (1.9)$$

### Rješenje problema linearnih najmanjih kvadrata

Ukoliko je sustav  $i$ .  $f(x_i; \mathbf{p}) = y_i$  rješiv za linearnu  $f$ , rješenje se dobiva klasičnim metodama linearne algebre. Taj je slučaj u praksi izuzetno rijedak zbog greški u mjerenjima pa nadalje pretpostavimo da je sustav nerješiv, ali da su sva mjerenja linearno nezavisna. U tom slučaju postavljaju se tzv. **normalne jednadžbe** čijim rješavanjem se dolazi do parametara  $\mathbf{p}$  koji minimiziraju grešku  $E$  [11].

### Nelinearni problem najmanjih kvadrata - iterativne metode

Ukoliko  $f(x; \mathbf{p})$  ne ovisi linearno o  $\mathbf{p}$  govorimo o nelinearnom problemu najmanjih kvadrata. Taj se problem obično rješava iterativno, linearizacijom oko trenutne estimacije parametra  $\mathbf{p}$  koristeći Jakobijan  $\mathbf{J} = \partial f / \partial \mathbf{p}$  i računanjem inkrementalnog ažuriranja  $\Delta \mathbf{p}$ :

$$\begin{aligned} E(\Delta \mathbf{p}) &= \sum_i \|f(x_i; \mathbf{p} + \Delta \mathbf{p}) - y_i\|^2 \\ &\approx \sum_i \|\mathbf{J}(x_i; \mathbf{p}) \Delta \mathbf{p} - \mathbf{r}_i\|^2. \end{aligned} \quad (1.10)$$

Pošto gornja aproksimacija vrijedi jedino oko lokalnog minimuma ili za male vrijednosti  $\Delta \mathbf{p}$ , ažuriranje  $\mathbf{p} \leftarrow \mathbf{p} + \Delta \mathbf{p}$  ne mora uvijek smanjiti grešku  $E$ . Zbog toga se ažuriranje parametara radi u manjim koracima [11]:

$$\mathbf{p} \leftarrow \mathbf{p} + \alpha \Delta \mathbf{p}, \quad 0 < \alpha \leq 1.$$

### Newtonova metoda

Neka je  $e(\mathbf{p})$  skalarna funkcija više varijabli koju nastojimo minimizirati po svim mogućim vrijednostima parametra  $\mathbf{p}$ . Uz dvije pretpostavke:  $e(\mathbf{p})$  ima dobro definiran minimum i poznata je vrijednost  $\mathbf{p}_0$  razumno blizu  $\min_{\mathbf{p}} e(\mathbf{p})$ , možemo razviti  $e(\mathbf{p})$  u Taylorov red:

$$e(\mathbf{p}_0 + \Delta \mathbf{p}) = e(\mathbf{p}_0) + J_e(\mathbf{p}_0) \Delta \mathbf{p} + \frac{1}{2} \Delta \mathbf{p}^\top H_e \Delta \mathbf{p} + \dots \approx e(\mathbf{p}_0) + J_e(\mathbf{p}_0) \Delta \mathbf{p} + \frac{1}{2} \Delta \mathbf{p}^\top H_e \Delta \mathbf{p}.$$

Uz pretpostavku da prva tri člana aproksimacije dovoljno dobro opisuju  $f$ , možemo minimizirati prethodni izraz u ovisnosti o  $\Delta \mathbf{p}$  tako da za iduću aproksimaciju  $\mathbf{p}$  uzmemo onu vrijednost u kojoj derivacija po  $\Delta \mathbf{p}$  poprima vrijednost 0. Tj. tražimo rješenje:

$$J_e(\mathbf{p}_0) + H_e \Delta \mathbf{p} = 0. \quad (1.11)$$

Koristeći  $\Delta \mathbf{p}$  koji rješava jednadžbu 1.11 iterativno ažuriramo parametar  $\mathbf{p}$  do konvergencije.

Nadalje promatramo specifičan oblik funkcije greške na koju nailazimo u problemu najmanjih kvadrata (kao u 1.10), preciznije slučaj u kojem je  $e(\mathbf{p})$  kvadrat norme:

$$e(\mathbf{p}) = \frac{1}{2} \|\epsilon(\mathbf{p})\|^2 = \frac{1}{2} \epsilon(\mathbf{p})^\top \epsilon(\mathbf{p})$$

gdje je:

$$\epsilon(\mathbf{p}) = f(\mathbf{p}) - \mathbf{y}$$

a faktor  $\frac{1}{2}$  dodan zbog jednostavnosti daljnjeg računa.

U tom slučaju vrijedi:

$$\frac{\partial e}{\partial \mathbf{p}} = \left( \frac{\partial \epsilon}{\partial \mathbf{p}} \right)^\top \epsilon. \quad (1.12)$$

Nadalje:

$$\frac{\partial \epsilon}{\partial \mathbf{p}} = \frac{\partial f}{\partial \mathbf{p}} =: \mathbf{J}_f \quad (1.13)$$

pa iz 1.12 i 1.13:

$$\mathbf{J}_e = \frac{\partial e}{\partial \mathbf{p}} = \mathbf{J}_f^\top \epsilon. \quad (1.14)$$

Ponovnom derivacijom 1.12 po  $\mathbf{p}$  dobivamo:

$$\frac{\partial^2 e}{\partial \mathbf{p}^2} = \left( \frac{\partial \epsilon}{\partial \mathbf{p}} \right)^\top \frac{\partial \epsilon}{\partial \mathbf{p}} + \frac{\partial^2 \epsilon}{\partial \mathbf{p}^2} \epsilon \quad (1.15)$$

$$\stackrel{1.14}{=} \mathbf{J}_f^\top \mathbf{J}_f + \frac{\partial^2 f}{\partial \mathbf{p}^2} \epsilon. \quad (1.16)$$

Uvrštavanjem u 1.11 dobivamo izraz za ažuriranje parametara  $\mathbf{p}$ :

$$\mathbf{J}_f^\top \epsilon + \left( \mathbf{J}_f^\top \mathbf{J}_f + \frac{\partial^2 f}{\partial \mathbf{p}^2} \epsilon \right) \Delta \mathbf{p} = 0.$$

### Gauss-Newtonova metoda

Pod pretpostavkom da je  $f(\mathbf{P})$  lokalno linearna vrijedi  $\frac{\partial^2 f}{\partial \mathbf{p}^2} = 0$  uvrštavanjem u 1.16 dobivamo jednostavniju aproksimaciju Hessiana:

$$\frac{\partial^2 e}{\partial \mathbf{p}^2} = \mathbf{J}_f^\top \mathbf{J}_f$$

čijim uvrštavanjem u 1.11 dobivamo normalnu jednadžbu za  $\Delta \mathbf{p}$ :

$$\mathbf{J}_e + \mathbf{J}_f^\top \mathbf{J}_f \Delta \mathbf{p} \stackrel{1.14}{=} \mathbf{J}_f^\top \epsilon + \mathbf{J}_f^\top \mathbf{J}_f \Delta \mathbf{p} = 0. \quad (1.17)$$

### Gradijentni spust

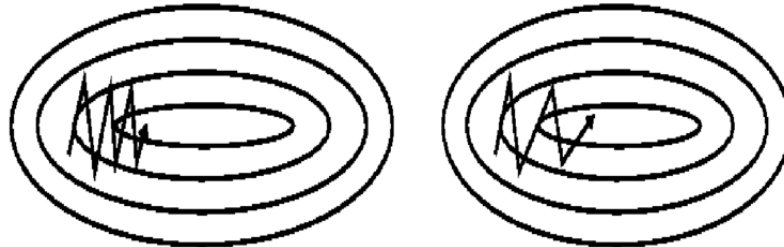
Metoda gradijentnog spusta polazi od činjenice da je smjer najbržeg pada funkcije suprotan vektoru gradijenta:  $-\frac{\partial e}{\partial \mathbf{p}} = -\left(\frac{\partial \epsilon}{\partial \mathbf{p}}\right)^\top \epsilon$ . Strategija minimizacije je stoga iterativno ažurirati parametre u smjeru najbržeg pada. Veličina koraka  $\lambda$  (eng. *learning rate*) predstavlja hiperparametar metode i određuje se nekom formom pretrage (npr. linijska pretraga).

$$\Delta \mathbf{p} = -\lambda \frac{\partial e}{\partial \mathbf{p}} = -\lambda \mathbf{J}_e. \quad (1.18)$$

1.18 je oblika 1.11 s (donekle proizvoljnom) aproksimacijom Hesseove matrice s  $\frac{1}{\lambda} \mathbf{I}$ .

Gradijentni spust je sam po sebi karakteriziran sporom konvergencijom (zbog tzv. *zig-zag* uzorka) i ne smatra se povoljnom strategijom minimizacije. Zbog toga se kombinira s Gauss-Newtonovom iteracijom ili dodatnim optimizacijskim algoritmima.

**Gradijentni spust s momentom.** Jedan od glavnih problema kod gradijentnog spusta su regije gdje je zakrivljenost značajno izraženija u jednom smjeru tj. omjer najveće i najmanje svojstvene vrijednosti Hesseove matrice je velik (tj. Hesseova matrica je loše uvjetovana). Tu karakteristiku često pronalazimo kod lokalnih optimuma. U tim regijama gradijentni spust sporo napreduje oscilirajući duž obronaka plohe greške kao što je vidljivo na slici 1.4 lijevo.



Slika 1.4: Gradijentni spust sa (desno) i bez (lijevo) momenta. [10]

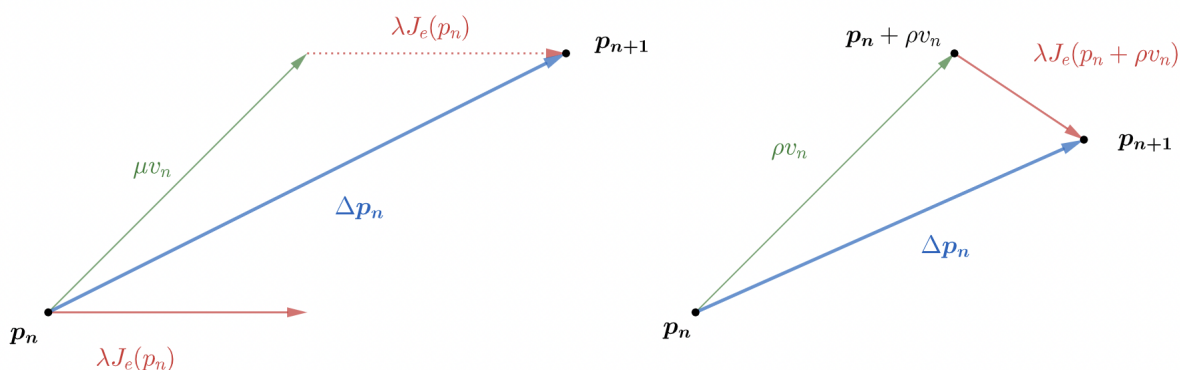
Metoda momenta motivirana je fizikalnom interpretacijom minimizacije kao postavljanja kuglice brzine  $v = 0$  na proizvoljnoj poziciji na brdovitom terenu. Sila na kuglicu se u tom slučaju opisuje negativnim gradijentom potencijalne energije ( $F = -\nabla U$ ). S druge strane  $F = ma$  iz čega slijedi kako je gradijent obrnuto proporcionalan akceleraciji kuglice. Iz toga slijedi kako gradijent (za razliku od klasičnog gradijentnog spusta) ne utječe direktno na poziciju kuglice, već se promjena pozicije uvjetovana gradijentom odvija posredno preko promjene brzine [1]:

$$\Delta \mathbf{p} = v_{n+1} = \mu v_n - \lambda \mathbf{J}_e. \quad (1.19)$$

Metoda momenta pomaže ubrzati napredovanje gradijentnog spusta u relevantnom smjeru i smanjuje oscilacije (slika 1.4 desno). Na taj način ubrzavaju se promjene koje nastavljaju istim smjerom i usporavaju oscilacije. [10]

**Nesterov moment** dodaje opisanoj kuglici ideju o smjeru kojim se kreće s idejom kako bi trebala usporiti pred sljedeće brdo. U praksi često pokazuje bolje rezultate od standardnog momenta. Za razliku od klasičnog momenta, Nesterov moment kod računanja gradijenta uzima *pogled unaprijed* (aproksimaciju buduće pozicije)  $\mathbf{p}_n + \mu v_n$ :

$$\Delta \mathbf{p} = v_{n+1} = \rho v_n - \lambda \mathbf{J}_e(\mathbf{p}_n + \rho v_n). \quad (1.20)$$



Slika 1.5: Usporedba ažuriranja parametara kod klasičnog (lijevo) i Nesterovog momenta (desno). [1]

**Adagrad** (eng. *Adaptive Gradient Algorithm*) problemu konvergencije gradijentnog spusta pristupa iz nešto drugačijeg kuta. Parametar  $\lambda$  često nije idealan za svaku komponentu vektora  $\mathbf{p}$  (zbog razlika u skali), zbog toga Adagrad adaptira magnitudu promjene svakog parametra posebno, tj. adaptira  $\lambda$  za svaki parametar. Modifikacije se izvode na temelju prethodnih gradijenata  $\frac{\partial e}{\partial p_i}$ :

$$\Delta \mathbf{p}_n = -\frac{\lambda}{\sqrt{C_n + \varepsilon}} \mathbf{J}_{e,n-1}(\mathbf{p}_{n-1}) \quad (1.21)$$

$$C_n = \sum_{i=1}^{n-1} \mathbf{J}_{e,i}^2(\mathbf{p}_i) \quad (1.22)$$

gdje je  $\varepsilon$  korekcijski termin koji sprječava dijeljenje s 0.

Adagrad osigurava da se komponente  $\mathbf{p}$  koje se značajnije mijenjaju ažuriraju sporije i obrnuto za manje izražene smjerove. Glavni nedostatak Adagrada je što kroz iteracije  $C_n$  monotono raste i uzrokuje prebrzu konvergenciju  $\Delta\mathbf{p}_n$  u 0. [1]

**RMSProp** je modifikacija Adagrada koja nastoji smanjiti problem nestajućih ažuriranja tako da akumulira gradijente na sljedeći način [1]:

$$C_n = 0.9 \cdot C_{n-1} + 0.1 \cdot \mathbf{J}_{e,n-1}^2(p_i). \quad (1.23)$$

**Adam** (*Adaptive moment estimation*) je novija optimizacijska metoda koja inkorporira estimaciju prvog i drugog momenta u RMSProp:

$$m_n = \beta_1 m_{n-1} + (1 - \beta_1) \mathbf{J}_{e,n-1} \quad (1.24)$$

$$v_n = \beta_2 v_{n-1} + (1 - \beta_2) \mathbf{J}_{e,n-1}^2. \quad (1.25)$$

Pošto su  $m_0, v_0$  jednaki  $\vec{0}$ , autori Adama dodatno uvode prigušujuće korekcije kako bi izmakli vrijednosti  $m_n$  i  $v_n$  od  $\vec{0}$  [8]:

$$\hat{m}_n = \frac{m_n}{1 - \beta_1^n} \quad (1.26)$$

$$\hat{v}_n = \frac{v_n}{1 - \beta_2^n}. \quad (1.27)$$

Konačno,  $m_n$  i  $v_n$  se inkorporiraju u pravilo ažuriranja  $\Delta\mathbf{p}$ :

$$\Delta\mathbf{p}_n = -\frac{\lambda}{\sqrt{\hat{v}_n} + \varepsilon} \hat{m}_n. \quad (1.28)$$

### Löwenberg-Marquardtova metoda

Löwenberg-Marquardtova metoda je proširenje Gauss-Newtonove metode u kojem se promatraju augmentirane normalne jednadžbe po uzoru na 1.17. Često se smatraju kombinacijom gradijentnog spusta i Gauss-Newtonove metode zbog komponente  $\frac{1}{\lambda}I$  spomenute u diskusiji gradijentnog spusta (1.3):

$$\mathbf{J}_f^\top \boldsymbol{\epsilon} + (\mathbf{J}_f^\top \mathbf{J}_f + \frac{1}{\lambda}I) \Delta\mathbf{p} = 0. \quad (1.29)$$

Ako vrijednost  $\Delta\mathbf{p}$  dobivena rješavanjem normalnih jednadžbi smanjuje  $\boldsymbol{\epsilon}$ , inkrement se prihvaća i  $\lambda$  smanjuje za faktor  $\approx 10$ , inače se povećava za isti faktor i ponovno se pristupa rješavanju augmentirane normalne jednadžbe.

## Poglavlje 2

### Opis problema

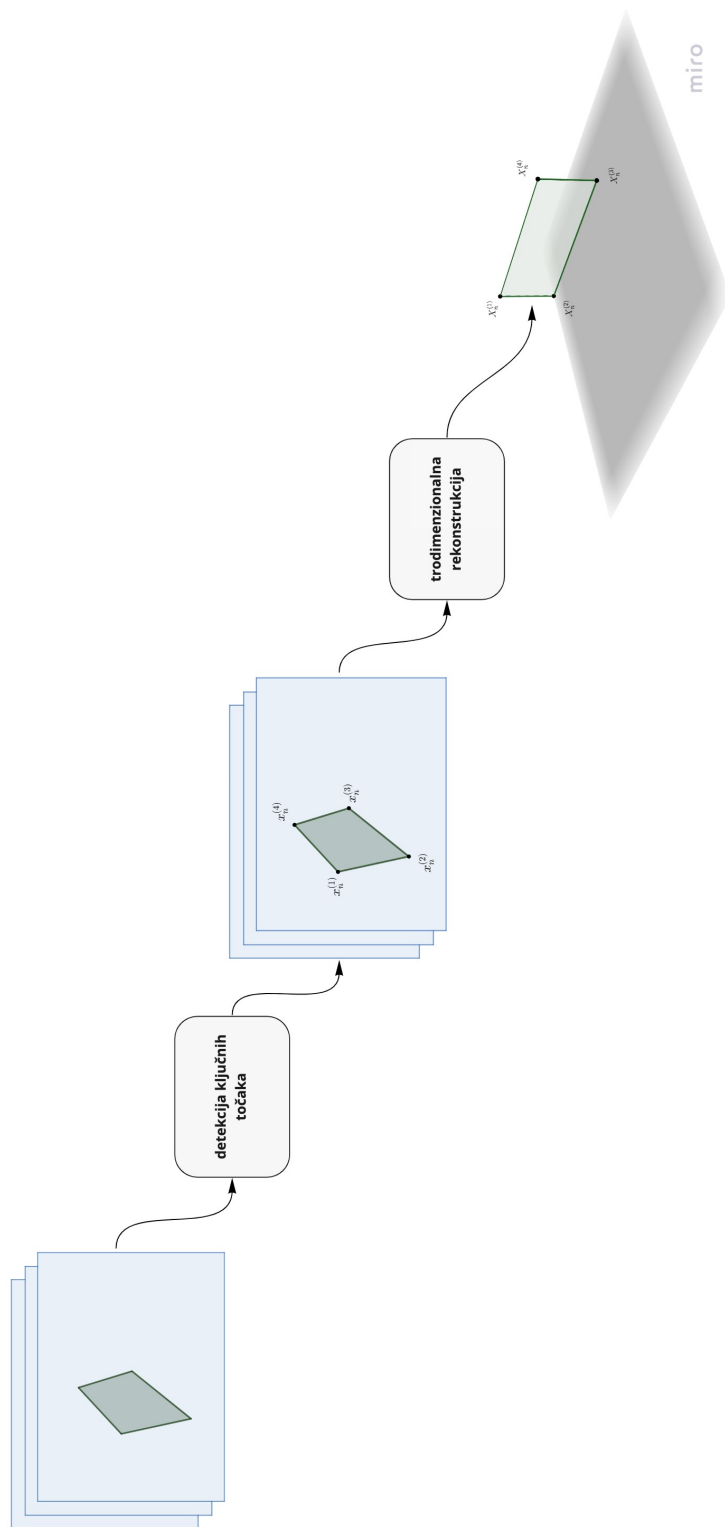
U nastavku rada promatra se problem rekonstrukcije položaja objekta u trodimenzionalnom prostoru na temelju njegovih fotografija. Fotografije objekta nadalje nazivamo i projekcijama referirajući se na projekcije nastale djelovanjem kamere opisane u 1.2. Pretpostavljamo da je objekt statičan i pravokutnog oblika, projekcije sva četiri vrha objekta su vidljive u konačnoj fotografiji (dakle u  $\mathbb{P}^2$  tvore konveksan četverokut) i poznati su intrinzični i ekstrinzični parametri kamere.

Problem se pristupa u dvije etape. Prvo se na fotografijama, dubokim učenjem identificiraju vrhovi projekcije traženog objekta. Taj problem je modeliran kao detekcija ključnih točaka, a producira četiri točke u  $\mathbb{P}^2$  i njega uzimamo kao gotovu komponentu. Zatim se pristupa rekonstrukciji položaja trodimenzionalnog objekta iz niza projekcija. Rekonstruirani objekt čini oblak točaka koji se sastoji od četiriju vrhova traženog objekta. Skica postupka vidljiva je na slici 2.1.

Pretpostavlja se kako je greška algoritma za procjenu pozicije projekcije objekta normalno distribuirana, tj. da su procjene projekcija traženih točaka (u  $\mathbb{R}^2$ ) normalno distribuirane oko stvarnih projekcija. Dodatno, ovim je radom cilj procijeniti maksimalnu dozvoljenu pogrešku u procjenama položaja dvodimenzionalnih projekcija tako da rekonstrukcije ostanu u dozvoljenoj okolini oko traženog predmeta. Uzima se kako je dovoljno da svaki od vrhova reprodukcije bude unutar sfere radijusa 10% kraće stranice stvarnog objekta (slika 5.2).

Dodatan zahtjev na finalan sustav je da predstavlja cjelovito rješenje, to jest da sustav za procjenu 3D pozicije objekta na ulazu ima fotografiju objekta i pripadne matrice kamere, a na izlazu treba dati procjenu položaja objekta (kroz četiri vrha u  $\mathbb{R}^3$ ). Uz to, finalan sustav bi se trebao izvršavati na mobilnim uređajima i uređajima proširene stvarnosti.





Slika 2.1: Gruba skica sustava za rekonstrukciju položaja objekta u trodimenzionalnom prostoru na temelju njegovih fotografija.

## **Poglavlje 3**

# **Pregled metoda estimacije trodimenzionalnog položaja iz projekcije u dvodimenzionalnom prostoru**

### **3.1 Iz jednog pogleda**

Problem estimacije 3D položaja iz projekcije u 2D prostoru iz samo jedne fotografije obrađen je u [9]. U radu se detaljnije promatra problem rekonstrukcije položaja pravokutnog objekta poznavajući njegove stvarne dimenzije (visinu i širinu). Postupak je egzaktan i relativno jednostavan, a temeljen je na vrlo osnovnim geometrijskim odnosima. Bez obzira na to, nedostaju mu par ključnih stvari kako bi se pokazao koristan na ovom konkretnom problemu:

1. Bez poznavanja visine i širine objekta od interesa ovaj pristup nije praktično koristan jer procjena visine i širine objekta ponovno vodi na sličan problem.
2. Zbog grešaka u 2D procjenama, nije garantirano da će izračunate 3D pozicije zaista tvoriti pravokutan objekt, a nije pronađen jednostavan i objašnjiv način kako to osigurati.

Iz spomenutih razloga ocijenjeno je kako metoda u ovom slučaju nije od praktične koristi, stoga nije dalje evaluirana.

### **3.2 Iz više pogleda**

U nastavku opisujemo tri metode koje u konačnici evaluiramo i međusobno uspoređujemo (poglavlje 5). Nehomogenom metodom dobiva se inicijalna procjena pozicije objekta u tro-

dimenzionalnom prostoru, ta se procjena dalje rafinira koristeći Löwenberg-Marquardtovu metodu i metodu gradijentnog spusta. Nehomogena i Löwenberg-Marquardtova metoda opisane su u [7] i koriste se u praksi za slične probleme, dok se metoda gradijentnog spusta uz parametrizacije finalnog objekta, prema autorovom saznanju, do sada nije koristila u ovoj domeni.

### Nehomogena metoda

Ova metoda opisana je u [7], a radi na neovisnim točkama, dakle ni ona u osnovnom obliku ne koristi geometrijski *prior* (tj. da četiri točke koje pokušava rekonstruirati tvore pravokutnik u 3D prostoru).

Neka je  $X = \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \in \mathbb{P}^3$  jedan od vrhova pravokutnika koje nastojimo rekonstruirati. Iz

modela kamere opisanog u 1.2 imamo da vrijedi:

$$P_i X = \lambda \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} \in \mathbb{P}^2$$

za proizvoljan  $\lambda \in \mathbb{R}$ . Parametar  $\lambda$  možemo ukloniti ukoliko prethodnu jednadžbu zapišemo na sljedeći (analogan) način:

$$\begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} \times P_i X = 0.$$

Takav zapis vodi na tri jednadžbe s četiri nepoznanice. Dodatno, kako je  $X$  točka u trodimenzionalnom Euklidskom prostoru, dakle ne pripada ravnini u beskonačnosti, možemo bez smanjenja općenitosti uzeti  $w = 1$ . Tako dobivamo sustav tri jednadžbe s tri nepoznanice, ranga 2 (pošto je rješenje, zadano u homogenim koordinatama, definirano do na  $\lambda \in \mathbb{R}$ ). Iz toga slijedi kako za rješavanje ovog problema trebamo barem dvije fotografije istog objekta iz različitih pozicija.

Pošto su mjerenja u dvodimenzionalnom prostoru neprecizna, s dvije ili više fotografija, rješenje se traži linearnom metodom najmanjih kvadrata.

### Dodatna pretpostavka

Kako bismo pristupili rješavanju problema koristeći više pogleda (tj. više projekcija istog objekta) bilo je potrebno na neki način identificirati istovrsne projicirane točke i povezati

ih između različitih pogleda. Uz dodatnu pretpostavku kako su fotografije objekta napravljene na način da čuvaju orijentaciju prostora (ne rotiraju objekt oko niti jedne od osi), spomenuti algoritam (detekcija ključnih točaka) može jednostavno razlikovati svaki od vrhova projekcija.

### Minimizacija reprojekcijske pogreške

Nehomogena metoda postiže dobre rezultate na podacima s relativno malo šuma, međutim pokazuje se kako performanse metode značajno opadaju s povećanjem greške u ulaznim podacima. Dodatno, greška koju minimizira nema jasno geometrijsko značenje u ovom kontekstu. Takvo razmišljanje navodi na novi pristup u vidu minimizacije **reprojekcijske pogreške**:

$$E_{rp}^2(\mathbf{X}) = \sum_i d^2(\mathbf{x}_i, P_i \mathbf{X}) \quad (3.1)$$

$$=: \sum_i d^2(\mathbf{x}_i, \mathbf{x}_i') \quad (3.2)$$

$$=: \sum_i \delta_i^2. \quad (3.3)$$

gdje je  $P_i$  projekcijska matrica pridružena pogledu  $i$ , a  $\mathbf{X} \in \mathbb{P}^3$  trenutna procjena pripadajuće 3D točke.

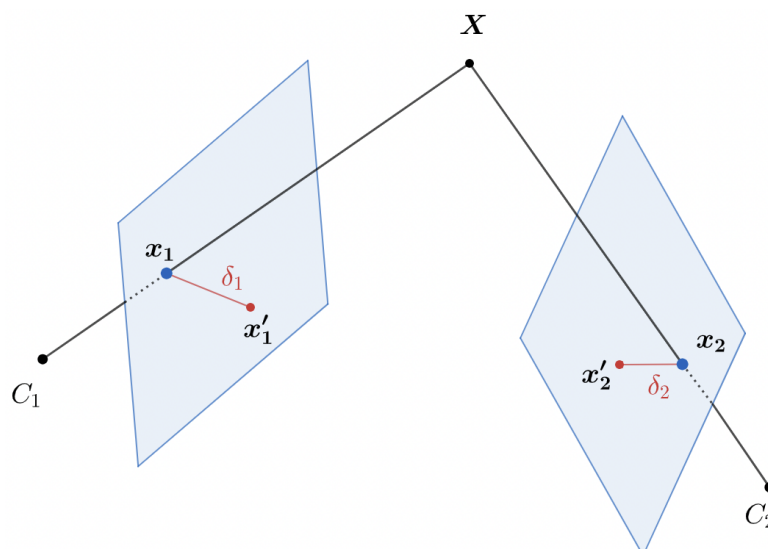
Intuitivno, nastoji se pronaći  $\mathbf{X} \in \mathbb{R}^3$  čije su projekcije (istim projekcijskim matricama  $P_i$  kojima su dobivene i procjene projekcije  $x_i$ ) najbliže, u smislu kvadratne udaljenosti, procijenjenim projekcijama  $x_i$ .

### Löwenberg-Marquardtovom metodom

Klasičan pristup ovom problemu opisan u [7] reprojekcijsku pogrešku minimizira Löwenberg-Marquardtovom metodom opisanom u 1.3. Löwenberg - Marquardtova metoda osjetljiva je na inicijalnu procjenu  $\hat{\mathbf{X}}$ , zbog toga se algoritam inicijalizira s rezultatom dobivenim nehomogenom metodom opisanom u 3.2.

### Gradijentnim spustom

Kao alternativa prethodnom pristupu i jednostavan način za osigurati da je konačan objekt zaista pravokutnog oblika, razvijena je još jedna metoda koja se također zasniva na minimizaciji reprojekcijske pogreške.

Slika 3.1: Reprojekcijska greška (ozn.  $\delta_i = d(x_i, P_i X)$ )

U sklopu ove metode, umjesto da posebno promatramo svaki od četiriju vrhova projekcije pravokutnika, definira se parametrizacija pravokutnika i zatim traže optimalne vrijednosti zadanih parametara. U sklopu ovog rada promatrane su sljedeće parametrizacije:

1. **8 parametarska** zadaje dva vrha pravokutnika i dvije komponente vektora koji opisuje treću (u  $\mathbb{R}^3$ ). Na taj način je pravokutnik potpuno definiran, a ostatak nepoznatih vrijednosti može se dobiti jednostavnim geometrijskim računom.
2. **9 parametarska** potpuno zadaje tri vrha pravokutnika.
3. **parametrizacija u obliku krutog tijela** za pravokutnik se zadaju translacija, os i kut rotacije te širina i visina.
4. **parametrizacija u obliku kvaterniona** za pravokutnik se zadaju translacija, rotacija opisana kvaternionom te širina i visina.

Optimalan  $X$  traži se metodom gradijentnog spusta koristeći napredne optimizacijske algoritme (vidi 1.3), a hiperparametre optimizatora definira se ovisno o uspješnosti minimizacijske procedure na skupu podataka za treniranje. Ova metoda jedina direktno pretražuje prostor pravokutnika u trodimenzionalnom prostoru.

## Poglavlje 4

### Implementacijski detalji

Na slici 4.2 vidi se apstraktna skica implementacije sustava za procjenu 3D položaja točaka iz 2D projekcija koristeći više pogleda. Ulaze u algoritam čine nova fotografija i ključne točke detektirane u prethodnih  $n - 1$  koraka (ozn.  $S_{n-1}^x$ ) zajedno s projekcijskim matricama koje definiraju parametre kamere u trenutku fotografiranja za trenutni i sve prethodne korake algoritma (ozn.  $S_n^P$ ). Oznaka  $S$  stoji za stanje (eng. *state*), a  $n$  označava duljinu stanja koje se uzima u obzir prilikom izvođenja tog koraka algoritma.

$$S_{n-1}^x = \left( \begin{array}{c} \left[ \begin{array}{c} \mathbf{x}_0^{(1)} \\ \mathbf{x}_0^{(2)} \\ \mathbf{x}_0^{(3)} \\ \mathbf{x}_0^{(4)} \end{array} \right], \left[ \begin{array}{c} \mathbf{x}_1^{(1)} \\ \mathbf{x}_1^{(2)} \\ \mathbf{x}_1^{(3)} \\ \mathbf{x}_1^{(4)} \end{array} \right], \dots, \left[ \begin{array}{c} \mathbf{x}_{n-1}^{(1)} \\ \mathbf{x}_{n-1}^{(2)} \\ \mathbf{x}_{n-1}^{(3)} \\ \mathbf{x}_{n-1}^{(4)} \end{array} \right] \end{array} \right) \quad (4.1)$$

$$S_n^P = (P_0, P_1, \dots, P_n) \quad (4.2)$$

Na stanja  $S_{n-1}^x$  i  $S_{n-1}^P$  može se gledati i kao niz uređenih parova ključnih točaka i parametara kamere u trenutku fotografiranja:

$$\left( \left[ \begin{array}{c} \mathbf{X}_0^{(1)} \\ \dots \\ \mathbf{X}_0^{(4)} \end{array} \right], P_0 \right), \left( \left[ \begin{array}{c} \mathbf{X}_1^{(1)} \\ \dots \\ \mathbf{X}_1^{(4)} \end{array} \right], P_1 \right), \dots, \left( \left[ \begin{array}{c} \mathbf{X}_{n-1}^{(1)} \\ \dots \\ \mathbf{X}_{n-1}^{(4)} \end{array} \right], P_{n-1} \right).$$

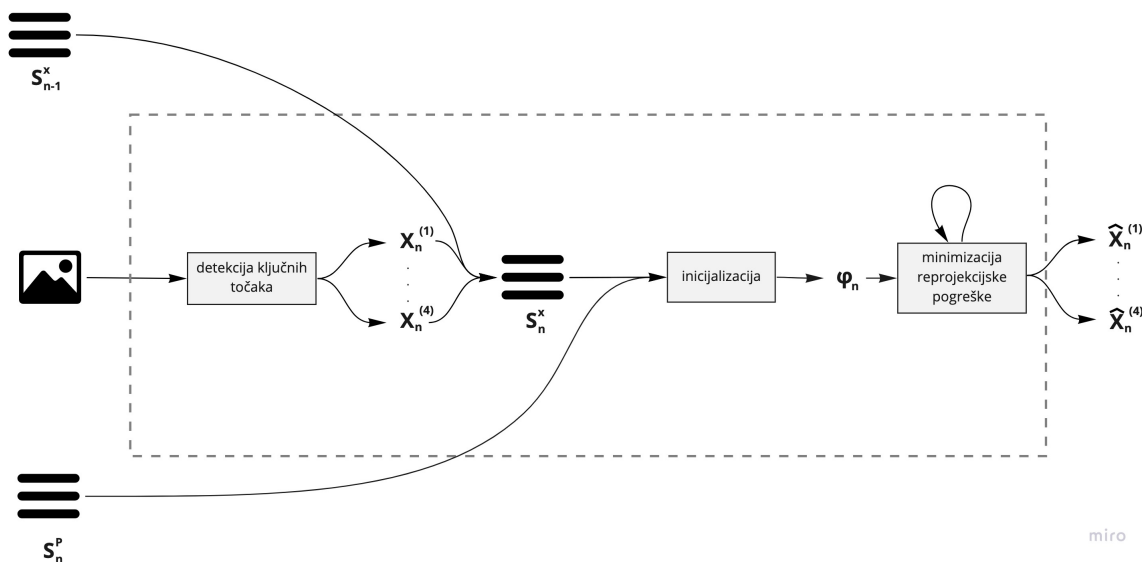
U tom slučaju, ulaz u algoritam za  $i = n$  dodatno sadrži i parametre kamere u  $i = n$  (ozn.  $P_n$ ), a ključne točke za tu iteraciju algoritma (ozn.  $(\mathbf{x}_n^{(1)}, \dots, \mathbf{x}_n^{(4)})$ ) odredit će se iz fotografije tijekom izvođenja algoritma.

Dulja stanja su komputacijski zahtjevnija ali rezultiraju kvalitetnijim procjenama (vidi 5). Način implementacije algoritma dozvoljava promjenu vrijednosti parametra  $n$  svakim pozivom algoritma, što ostavlja prostor za optimizaciju u tom smjeru. Jedna mogućnost bi

bila da se procjene generiraju koristeći minimalno dvije (zbog prirode algoritma) i maksimalno  $N$  (parametar algoritma) fotografija. Tako se rezultati mogu brzo generirati (iz manjeg stanja) i vremenom ispravljati dodavajući nove informacije. Dodatna optimizacija u tom smjeru može se dobiti metodom RANSAC ili jednostavnim vremenskim usrednjavanjem. Spomenuti optimizacijski pristupi nisu obrađeni ovim radom.

Detekcija ključnih točaka provodi se algoritmom dubokog učenja i producira novu procjenu projekcije vrhova objekta (ozn.  $\hat{X}^{(1)}, \hat{X}^{(2)}, \hat{X}^{(3)}, \hat{X}^{(4)} \xrightarrow{P} \mathbf{x}_n^{(1)}, \mathbf{x}_n^{(2)}, \mathbf{x}_n^{(3)}, \mathbf{x}_n^{(4)}$ ). Nova se procjena potom dodaje u  $S_{n-1}^x$  i ulazi inicijalizacijski algoritam zajedno s  $S_n^P$ . Ovisno o metodi estimacije, inicijalizacija može biti realizirana provedbom nehomogene metode opisane u 3.2 ili nasumičnim odabirom, a rezultat inicijalizacije predstavlja ulaz u algoritam minimizacije reprojekcijske pogreške. U slučaju optimizacije gradijentnim spustom rezultat inicijalizacije se zapisuje kao parametrizacija pravokutnika, tj. pretpostavlja se da je nehomogenom metodom zaista dobiven pravokutnik te se njegovi parametri (tj. parametri nekog sličnog pravokutnika) koriste za inicijalizaciju metode minimizacije reprojekcijske pogreške.

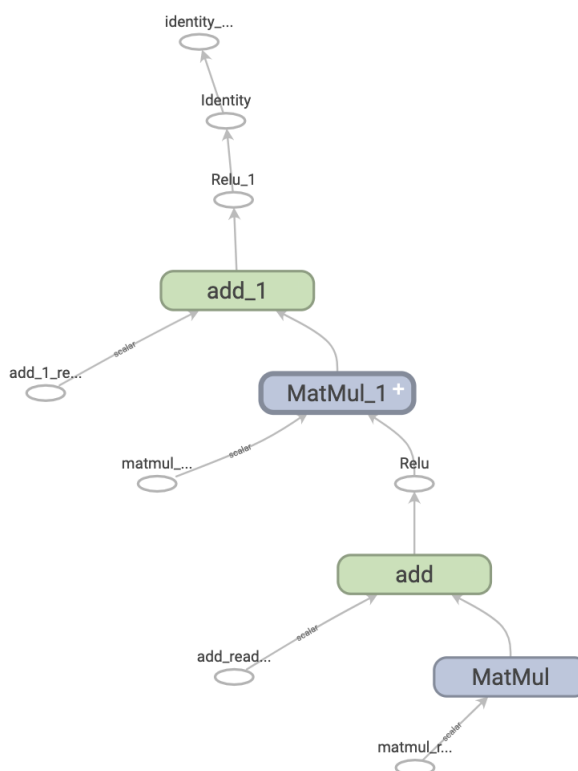
Inicijalni parametri  $\varphi_n$ , zajedno sa  $S_n^x$  i  $S_n^P$  (nije prikazano na dijagramu zbog preglednosti) predstavljaju ulaz u iterativni algoritam minimizacije reprojekcijske pogreške kako je opisano u 3.2. Zbog vremenske učinkovitosti izračunavanja, broj iteracija tog algoritma ograničen je na 20.



Slika 4.1: Skica implementacije sustava za procjenu 3D položaja pravokutnog objekta (definiranog točkama  $\hat{X}^{(1)}, \dots, \hat{X}^{(4)}$ ) iz niza 2D projekcija.

## 4.1 Biblioteka Tensorflow

Tensorflow je biblioteka otvorenog koda (eng. *open source*) za strojno učenje. Glavni građevni blokovi u Tensorflow-u su čvor (eng. *node*) i tenzor (eng. *tensor*), čvorovi predstavljaju operacije (jedinice izračunavanja, *tf.Operation*), a tenzori jedinice podataka (*tf.Tensor*). Zajedno se povezuju u graf izvršavanja (eng. *execution graph*) [2]. Takav zapis omogućava da se operacije optimiziraju na razini prevoditelja, među ostalim pojednostavljaju se nizovi aritmetičkih operacija, distribuiraju se operacije između dretvi (eng. *thread*) i uređaja, uklanjaju se nepotrebne operacije itd. [3]. Konačan izračun vrši se tek nakon predaje stvarnih podataka (tipa *tf.Tensor*) kao ulaza u graf. Dodatno, Tensorflowove grafove se može spremirati i izvršavati neovisno o izvornom kodu [2].



Slika 4.2: Primjer grafa u biblioteci tensorflow iz [2].

Jedan od praktičnih izazova bilo je to što se algoritam u konačnici morao izvršavati na više različitih platformi (mobilnim uređajima, uređajima za proširenu stvarnost, u browserima, ...). Kako unutar sebe sadrži podalgoritam detekcije ključnih točaka baziran na dubokom učenju i implementiran u biblioteci Tensorflow, cilj je bio i ostatak algoritma napisati



u formi Tensorflow grafa kako bi u konačnici imali jednu izvršnu datoteku za finalan algoritam koja bi se mogla izvesti u formatu u kojem se može pokretati na mobitelima, mikrokontrolerima, u browseru itd. (npr. TensorflowLite, TensorflowJS).

# Poglavlje 5

## Rezultati

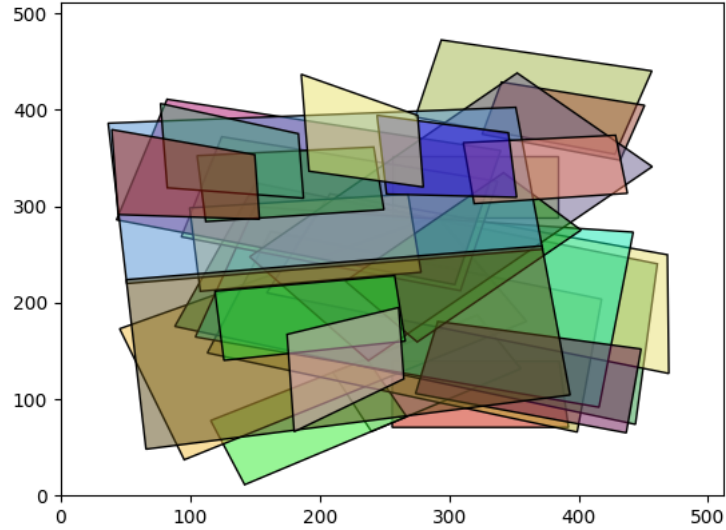
### 5.1 Testni podaci

Zbog prirode problema, pokazalo se relativno jednostavno generirati umjetne parove pozicije pravokutnika u trodimenzionalnom prostoru, nizove projekcijskih matrica i pridruženih projekcija pravokutnika na ravninu slike. Na taj način dobivamo proizvoljno mnogo, proizvoljno dugačkih nizova:  $S_N^x$  i  $S_N^p$ , za svaki od virtualnih pravokutnika ( $\hat{X}^{(1)}, \hat{X}^{(2)}, \hat{X}^{(3)}, \hat{X}^{(4)}$ ), a takvih se također može generirati proizvoljno mnogo. Takav se način prikupljanja podataka pokazao idealnim zbog toga što daje savršeno točnu informaciju o poziciji traženog objekta i omogućava generiranje proizvoljno mnogo testnih podataka.

Podaci su generirani uz pomoć skripte napisane u programskom jeziku Python koja omogućuje okidanje fotografije, vizualizaciju trenutnog pogleda i podešavanje parametara kamere. Skripta u konačnici producira poziciju virtualnog pravokutnika ( $\hat{X}^{(1)}, \hat{X}^{(2)}, \hat{X}^{(3)}, \hat{X}^{(4)}$ ) te liste ključnih točaka objekata na fotografiji ( $S_N^x$ ) i parametara kamere koje su ih producirale ( $S_N^p$ ).

Za jedan niz fotografija jednog objekta dopušteno je podešavanje isključivo ekstrinzičnih parametara kamere zbog pretpostavke da se u stvarnim primjenama kamera ne mijenja unutar jedne sesije (rekonstrukcije jednog specifičnog objekta). Intrinzični parametri kamere mogu se mijenjati između sesija. Dio prikupljenog skupa podataka za testiranje prikazan je kao preklapajući skup projekcija istog objekta (uz različite ekstrinzične parametre kamere) na slici 5.1.

Kako bi se približili stvarnim podacima, prilikom evaluacije i adaptiranja hiperparametara optimizacijskih metoda, umjetnim podacima je dodavana normalno distribuirana pogreška kako je opisano u 2. Stupanj pogreške je mjereno relativno s visinom projekciji opisanog pravokutnika (vidi 5.2).



Slika 5.1: Prikaz dijela testnih podataka. Svaki od četverokuta reprezentira jednu projekciju traženog objekta.

## 5.2 Metrike

U sklopu evaluacije uspješnosti predloženih rješenja definirane su dvije metrike. Jedna od njih, reprojekcijska pogreška, spomenuta je u diskusiji o zadanom problemu 3.2 te ju od sada označavamo s  $m_1$ , dok je druga zadana produktno i definirana kao postotak slučajeva u kojima su sva četiri vrha pravokutnika rekonstruirana unutar sfere određenog radijusa oko stvarne vrijednosti. Radijus je ovdje definiran kao 10% kraće stranice promatranog objekta. Detaljnije:

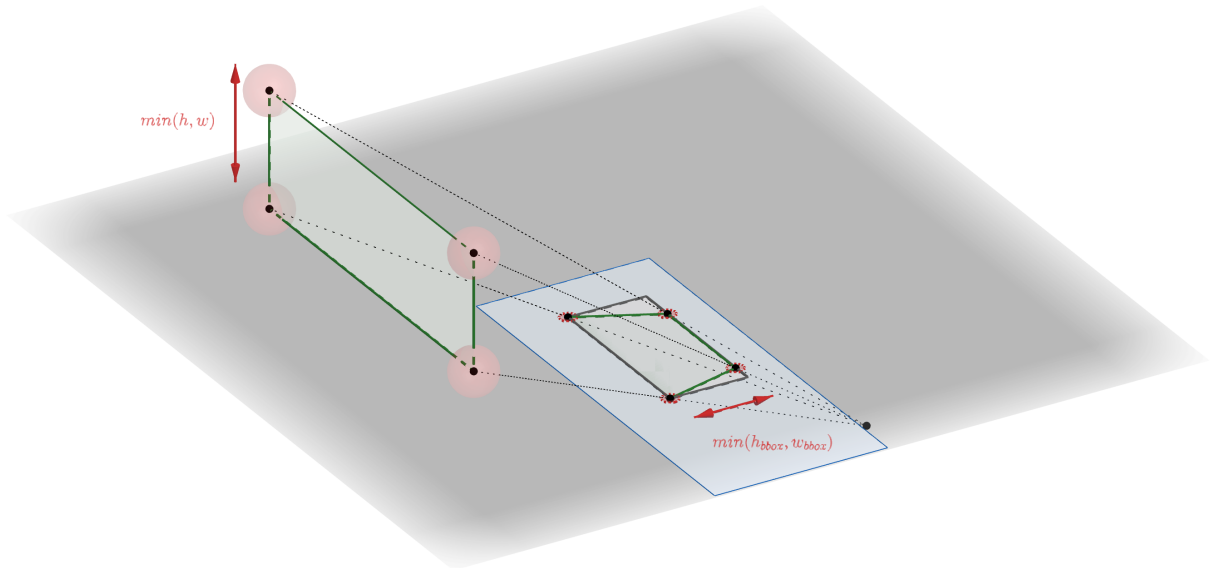
$$m_2 = \frac{1}{n} \sum_{i=0}^n \mathbb{1}_{\mathcal{I}(X_i)} \quad (5.1)$$

gdje je  $\mathcal{I}$  relacija definirana s:

$$\mathcal{I}(X_i) \iff X_i \in \mathcal{K}(\hat{X}, r) \quad (5.2)$$

a  $\mathcal{K}(\hat{X}, r)$  označava kuglu radijusa  $r$  oko  $\hat{X}$ .

Metrika  $m_2$  uvedena je zbog toga što daje jasniju sliku o performansama finalnog produkta od reprojekcijske greške.



Slika 5.2: Dopuštena greška u trodimenzionalnoj reprodukciji i procijenjena (a zatim i inducirana, kako je opisano u 5.1) greška u dvodimenzionalnim mjerenjima definirane su u terminima postotka minimuma visine i širine stvarnog objekta u 3D slučaju, tj. minimuma visine i širine projekciji opisanog pravokutnika (ozn. *bbox*, eng. *bounding box*) u 2D slučaju).

### 5.3 Odabir optimizatora gradijentnog spusta

Kako bismo pristupili evaluaciji metode gradijentnog spusta, bilo je potrebno prvo odabrati optimizator i definirati vrijednosti njegovih hiperparametara. U tu svrhu, izvršena je pretraga korištenjem biblioteke Optuna, a razmatrani su optimizatori opisani su u 1.3.

#### Biblioteka Optuna

Optuna je popularna Python biblioteka za pretragu hiperparametara. U svom opsegu definira cilj (eng. *objective*) pretrage, u smislu da kolekciji odabranih hiperparametara (eng. *trial*) pridružuje bročanu vrijednost koja kvantificira njegovu povoljnost i koju se u konačnici

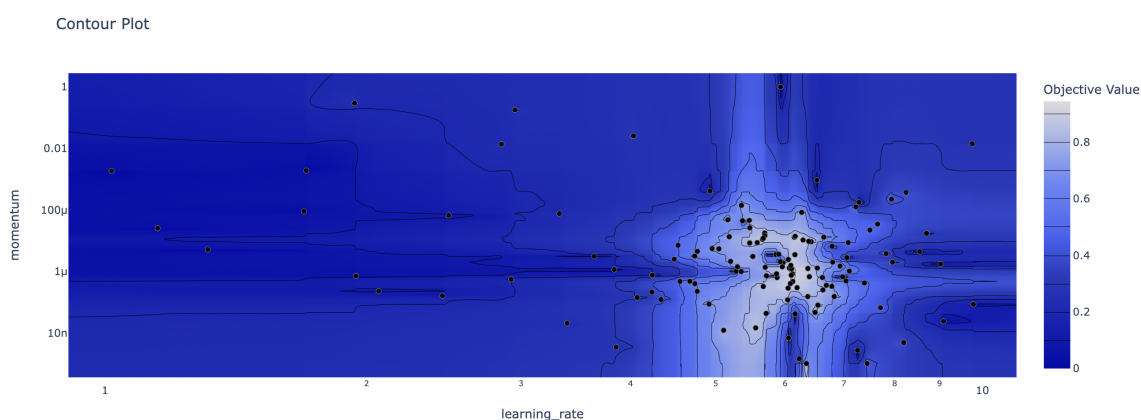
nastoji maksimizirati (ili minimizirati). Odabir idućeg *trial*-a vrši se na temelju Bayesove optimizacije ili evolucijskim algoritmima [4].

### Primjena na pretragu zadanog prostora optimizatora

Pretraga je provedena na testnom skupu podataka, a kao cilj pretrage uzeta je maksimizacija metrike  $m_2$  opisane u 5.2. Jedan *trial* sastojao se od odabira optimizatora i njemu pridruženih hiperparametara. Pretraga je provedena u 100 *trial*-a a rezultati su prikazani u tablici 5.1. Jedan prikaz postupka optimizacije vidljiv je na slici 5.3.

parametrizacija	optimizator	$\lambda$
8 parametarska	Adam ( $\beta_1 = 0.9, \beta_2 = 0.999$ )	0.3
9 parametarska	Adam ( $\beta_1 = 0.9, \beta_2 = 0.999$ )	0.3
kruto tijelo	gradijentni spust s Nesterovljevim momentom ( $\nu = 0.6$ )	$10^{-3}$
kvaternion	gradijentni spust s klasičnim momentom ( $\nu = 0.16$ )	2.8

Tablica 5.1: Najbolji pronađeni hiperparametri optimizacijskih metoda ovisno o parametrizaciji pravokutnika



Slika 5.3: Primjer contour plota vrijednosti optimizacijskog cilja u ovisnosti o parametrima optimizatora (moment eng. *momentum* i stopa učenja eng. *learning rate* tj. parametar  $\lambda$  gradijentnog spusta). Točke označavaju izvedene eksperimente.

## 5.4 Otpornost na pogrešku u procjeni pozicije dvodimenzionalnih, ključnih točaka

Konačno, iznosimo rezultate testa otpornosti spomenutih metoda u ovisnosti o stupnju pogreške u procjeni pozicije dvodimenzionalnih ključnih točaka. Nadalje se na tu mjeru referiramo kao otpornost na pogrešku u mjerenju. Dodatno, promatramo i ovisnost uspješnosti metode o kardinalitetu stanja ( $n$  u  $S_n^x$  i  $S_n^P$ ).

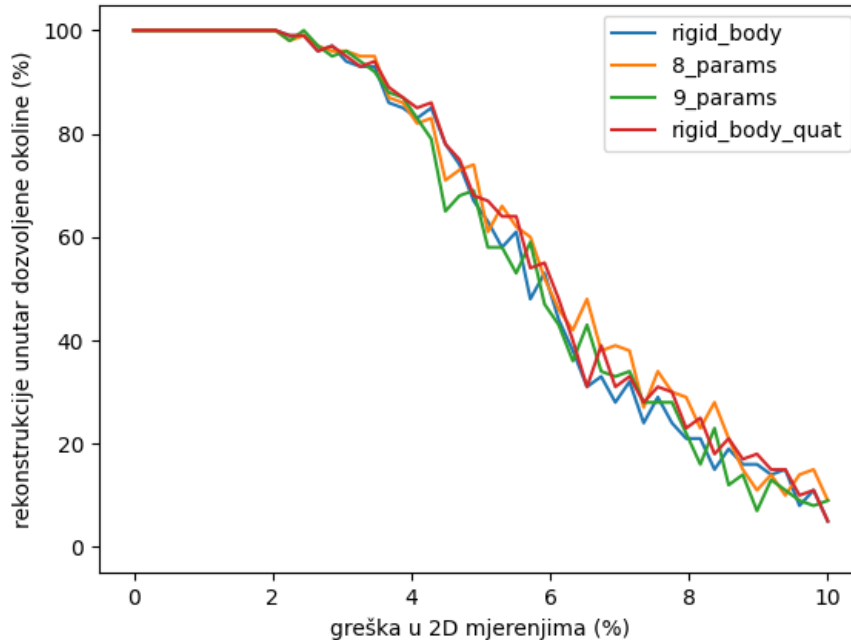
Na početku promatramo relativnu uspješnost varijanti metoda gradijentnog spusta u ovisnosti o odabiru parametrizacije pravokutnika (vidi 1.3), a zatim uspoređujemo uspješnost svih triju opisanih metoda (nehomogena, Löwenberg-Marquardtova i metoda gradijentnog spusta).

Test otpornosti metoda na pogrešku u mjerenju izveden je na normalno distribuiranim pogreškama unutar 10% najmanje stranice projekciji opisanog pravokutnika (vidi sliku 5.2). Zbog elementa nasumičnosti (kod generiranja pogreške), za svaku od razine pogreške izvedeno je po 100 eksperimenata po metodi, a rezultati su uprosječeni.

### Evaluacija i izbor parametrizacije pravokutnika

Svaka od parametrizacija pravokutnika evaluirana je sa stupnjevima pogreške mjerenja do 10% i stanjima duljine 5, 10 ili 20. Rezultati mjerenja za stanje duljine 10 prikazani su na slici 5.4. Sve su metode inicijalizirane rezultatom nehomogene metode iz razloga što nasumična inicijalizacija u velikoj većini slučajeva vodi do divergencije.

Pokazuje se kako među parametrizacijama pravokutnika nema neke značajne razlike u uspješnosti, što je relativno očekivano s obzirom na to da konceptualno ne predstavljaju značajno različite pristupe. Nadalje, odnos uspješnosti metoda nije se mijenjao u ovisnosti o duljini stanja. Zbog toga, u svrhu preglednosti, daljnje usporedbe vršimo samo s parametrizacijom kvaternionima.



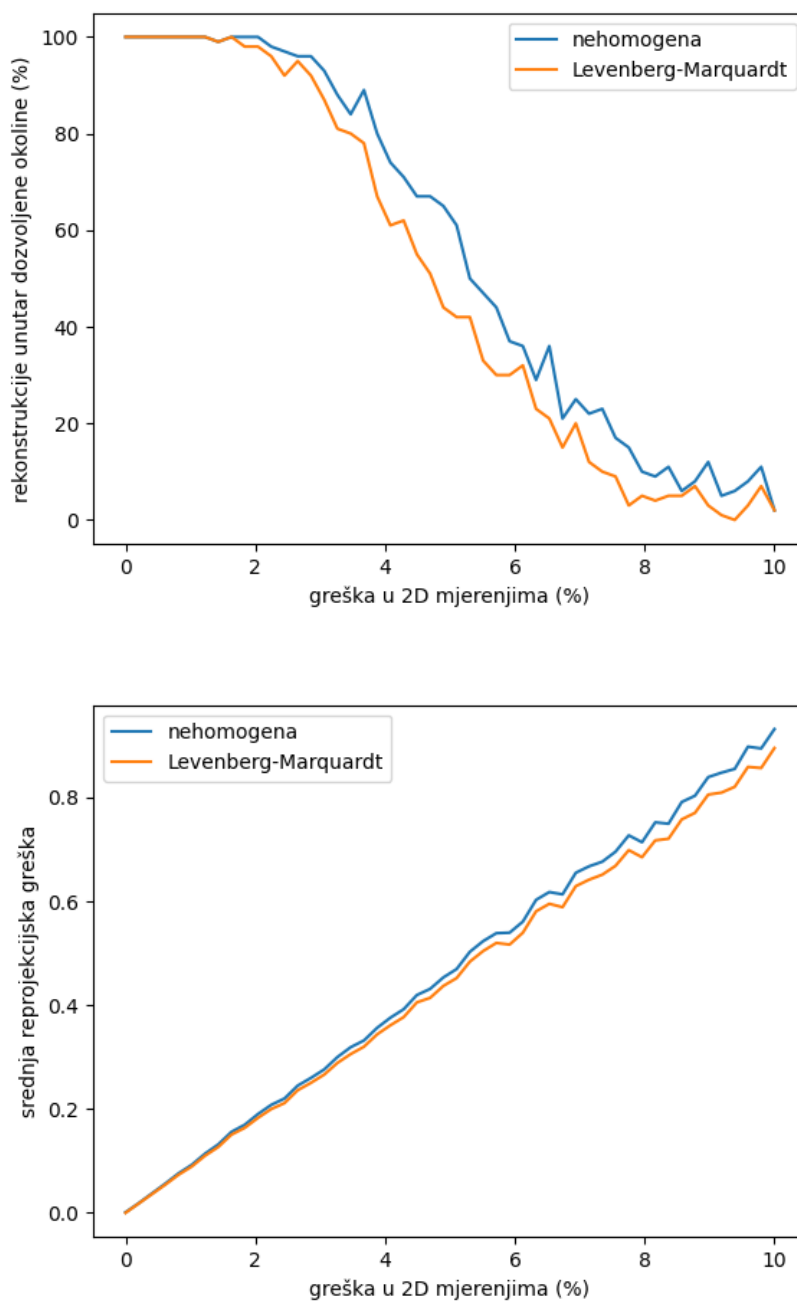
Slika 5.4: Usporedba otpornosti metoda rekonstrukcije temeljenih na gradijentnom spustu na pogrešku, u ovisnosti o parametrizacijama pravokutnika.

## Evaluacija metoda u ovisnosti o pogreški u mjerenju i kardinalitetu stanja

### Usporedba nehomogene i Löwenberg-Marquardtove metode

Pokazuje se kako Löwenberg-Marquardtova metoda u zadanih 10 koraka ne uspijeva pronaći bolje rješenje od inicijalnog. Zbog toga vraća identične rezultate kao i nehomogena metoda (razlog iza toga jasan je ukoliko se sjetimo načina rada metode opisanog u 1.3).

Nastavno na to, proveden je test Löwenberg-Marquardtove metode u kojem joj je dozvoljeno 100 koraka. Pokazuje se kako taj postupak čak pogoršava rezultate dobivene nehomogenom metodom (slika 5.5 lijevo). Jedna od mogućih interpretacija tog ishoda je da reprojekcijska pogreška ne opisuje adekvatno metriku  $m_2$  pa se Löwenberg-Marquardtovom metodom minimizacijom reprojekcijske pogreške zapravo odmičemo od minimuma u  $m_2$ . To je vidljivo na grafikonu 5.5 desno. Alternativno, moguće je da se Löwenberg-Marquardtovom metodom, uz sto iteracija, previše adaptiramo unesenim greškama na dvodimenzionalnim projekcijama.



Slika 5.5: Otpornost nehomogene i Löwenberg-Marquardtove metode sa 100 koraka na pogrešku u procjeni lokacije ključnih točaka u dvodimenzionalnom prostoru (lijevo) i srednja reprojekcijska pogreška rekonstruiranih objekata ovisno o metodi i pogrešci u dvodimenzionalnom prostoru (desno).

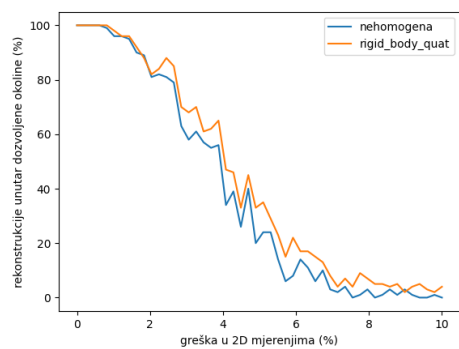


**Usporedba nehomogene i metode gradijentnog spusta**

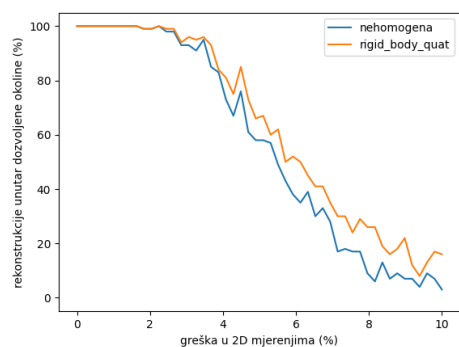
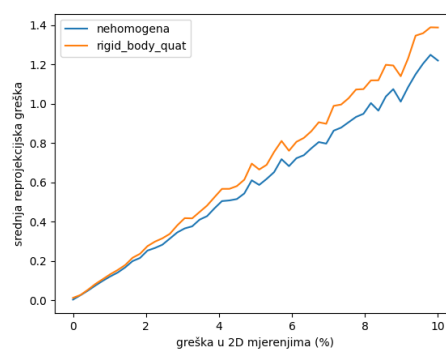
Na stanjima duljine 5, 10 i 20 proveden je test otpornosti metoda na pogrešku u mjerenju. Rezultati testa vidljivi su na grafikonu 5.6. Kako je i očekivano, metoda gradijentnog spusta poboljšava rezultat dobiven nehomogenom metodom, a veći kardinalitet stanja poboljšava rezultat na metrici  $m_2$ . Dodatno, može se primijetiti kako rast u performansama u metrici  $m_2$  ne prati pad srednje reprojekcijske pogreške, što dodatno potvrđuje slutnju kako te dvije metrike ne opisuju isti optimizacijski cilj.

Bitno je naglasiti kako uspješnost metode gradijentnog spusta u maksimiziranju rezultata metrike  $m_2$  gotovo sigurno ovisi i o adaptiranju parametara optimizatora koje u obzir uzima upravo tu metriku.

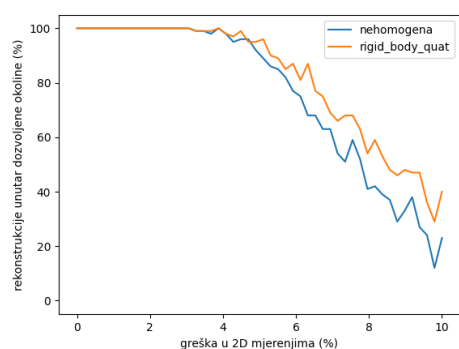
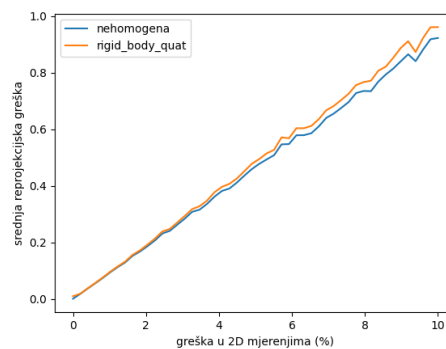
Dodatna prednost metode gradijentnog spusta je da garantira da rezultatni oblak točaka zaista definira pravokutnik u trodimenzionalnom prostoru.



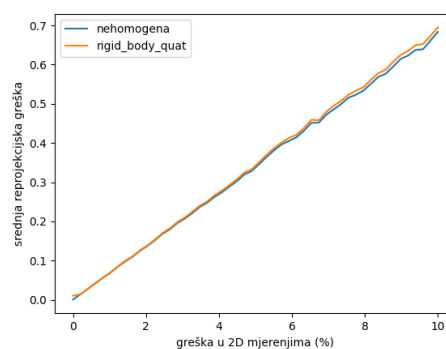
(a) Stanje kardinaliteta 5.



(b) Stanje kardinaliteta 10.



(c) Stanje kardinaliteta 20.



Slika 5.6: Rezultati testa otpornosti metode gradijentnog spusta s parametrizacijom kvaternionima na pogrešku u mjerenju s kardinalitetima stanja 5, 10 i 20. Metrika  $m_2$  prikazana je u lijevom stupcu, a srednja reprodukcijaska greška u desnom.

# Bibliografija

- [1] *CS231n Stanford*, 2022, <https://cs231n.github.io/neural-networks-3/>.
- [2] *Introduction to graphs and tf.function*, 2022, [https://www.tensorflow.org/guide/intro\\_to\\_graphs](https://www.tensorflow.org/guide/intro_to_graphs).
- [3] *TensorFlow graph optimization with Grappler*, 2022, [https://www.tensorflow.org/guide/graph\\_optimization](https://www.tensorflow.org/guide/graph_optimization).
- [4] T. Akiba, S. Sano, T. Yanase, T. Ohta i M. Koyama, *Optuna: A Next-generation Hyperparameter Optimization Framework*, Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2019).
- [5] B. Artmann, *Projective geometry*, <https://www.britannica.com/science/projective-geometry>.
- [6] A. Geiger, P. Lenz i R. Urtasun, *Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite*, Conference on Computer Vision and Pattern Recognition (CVPR), 2012.
- [7] R. Hartley i A. Zisserman, *Multiple view geometry in computer vision*, Cambridge university press, 2003.
- [8] D. P. Kingma i J. Ba, *Adam: A Method for Stochastic Optimization*, 2014, <https://arxiv.org/abs/1412.6980>.
- [9] H. L. Prematunga i A. T. Dharmaratne, *Finding 3d positions from 2d images feasibility analysis*, In Proc. of The Seventh International Conference on Systems (ICONS 2012), 2012.
- [10] N. Qian, *On the momentum term in gradient descent learning algorithms*, Neural Networks **12** (1999), br. 1, 145–151, ISSN 0893-6080, <https://www.sciencedirect.com/science/article/pii/S0893608098001166>.

- [11] R. Szeliski, *Computer vision: algorithms and applications*, Springer Science & Business Media, 2010.

# Sažetak

U ovom radu analiziran je problem rekonstrukcije položaja statičnog, pravokutnog objekta u trodimenzionalnom prostoru na temelju niza njegovih fotografija. Pravokutnik je opisan svojim vrhovima u 3D prostoru, a položaj njegovih vrhova na fotografiji (u 2D prostoru) se procjenjuje pomoću modela dubokog učenja. Opisani algoritam kao ulaz uzima skup procjena položaja objekta na fotografijama zajedno s parametrima koji definiraju kameru koja je proizvela odgovarajuće fotografije i vraća procjenu položaja objekta od interesa u trodimenzionalnom prostoru.

Prezentiran je pregled teorijske pozadine i predstavljene su dvije standardne metode rješavanja zadanog problema, nehomogena metoda i Löwenberg-Marquardtova metoda. Dodatno, predložena je nova metoda koja se temelji na gradijentnom spustu i kojom je osigurano da je rekonstruirani objekt zaista pravokutnog oblika. To je ostvareno na način da se pretraga vrši u prostoru parametara jedne od četiriju odabranih parametrizacija pravokutnika u trodimenzionalnom prostoru.

Rezultat nehomogene metode uzet je kao početna procjena za preostale dvije metode, kojima je rađena daljnja optimizacija. Promatrane su dvije metrike, reprojekcijska pogreška ( $m_1$ ) i poslovna metrika ( $m_2$ ). Pošto je za poznavanje rezultata poslovne metrike nužno poznavati i stvarnu poziciju traženog objekta u trodimenzionalnom prostoru, u algoritmu se promatra  $m_1$ . Međutim, pokazalo se da rezultati ove dvije metrike ne slijede nužno isti trend. Bez obzira na to, kako je i očekivano, pokazalo se da predložena metoda u ovom području zaista unaprijeđuje dosadašnje. Cjeloviti sustav implementiran je u biblioteci Tensorflow.

# Summary

This thesis analyzes the problem of reconstructing the position of a static, rectangular object in three-dimensional space based on a series of its photographs. The rectangle is described by its vertices in 3D space and the position of its vertices in the photograph (in 2D space) is estimated using a deep learning model. The algorithm takes in a set of estimates along with parameters defining the camera that produced the corresponding images and produces an estimate of the 3D position of the object of interest.

An overview of the theoretical background is presented and two standard methods for solving the given problem, the inhomogeneous method and the Levenberg-Marquardt method, are introduced. Additionally, a new method, based on gradient descent, is proposed. The method ensures that the reconstructed object is indeed rectangular in shape by searching the parameter space of one of the four selected rectangle parameterizations in three-dimensional space.

The result of the inhomogeneous method was taken as the initial estimate for the remaining two methods, with which a further optimization was performed. Two metrics were monitored, the reprojection error ( $m_1$ ) and a business metric ( $m_2$ ). Since, to calculate the results of the business metrics, one has to know the true position of the desired object in three-dimensional space, the algorithm monitors  $m_1$ . However, it was shown that the results of these two metrics do not necessarily follow the same trend. Regardless of that, as expected, it has been shown that the proposed method, in this domain, improves the previous ones. The entire system was implemented in the Tensorflow library.

# Životopis

Rođena sam 22. 12. 1996. godine u Splitu. Osnovnoškolsko obrazovanje sam završila 2011. godine u Osnovnoj školi Manuš u Splitu. Iste godine upisujem Treću gimnaziju u Splitu (MIOC) koju završavam 2015. godine. Fakultetsko obrazovanje započinjem na biološkom odsjeku Prirodoslovno-matematičkog fakulteta, smjer Molekularna biologija. Tijekom druge godine studija radila sam kao pripravnik u Laboratoriju za staničnu biologiju i prijenos signala na institutu Ruđer Bošković. 2017. godine upisujem preddiplomski sveučilišni studij Matematika na Matematičkom odsjeku istog fakulteta kojeg završavam 2020. godine kada upisujem diplomski sveučilišni studij Računarstvo i matematika na istom fakultetu.

Tijekom preddiplomskog studija na Matematičkom odsjeku pohađala sam ljetne programe: *Hilbert-Bernays Summer School on Logic and Computation* (2019.), *The Cornell, Maryland, Max-Planck Pre-doctoral Research School* (2020.) i *Lviv Data Science Summer School* (2020.). Za uspjeh na preddiplomskom i diplomskom studiju 2020. i 2022. godine uručena mi je *Nagrada za najuspješnije studente završnih godina svih preddiplomskih, diplomskih i integriranog studija*. Od siječnja 2021. godine radim u statup kompaniji PhishAR kao inženjer strojnog učenja.