



UNIVERSITÀ DEGLI STUDI DI MILANO - BICOCCA  
**Dipartimento di Informatica, Sistemistica e  
Comunicazione**  
Corso di Laurea in Informatica

# **Percezione dell'Ambiente e Costruzione della Base di Conoscenza per Robot Umanoidi**

**Relatore:** Prof. Dimitri Ognibene

**Tesi di Laurea di:**  
Luca Brini  
Matricola 879459

**Anno Accademico 2023-2024**

# Indice

<b>Introduzione</b>	<b>3</b>
<b>1 Stato dell'Arte</b>	<b>4</b>
<b>2 Mappe semantiche</b>	<b>5</b>
2.1 Definizione . . . . .	5
2.1.1 Nodi . . . . .	6
2.1.2 Archi . . . . .	8
2.1.3 Grafo stanze . . . . .	8
2.1.4 Grafo Oggetti . . . . .	8
2.1.5 Esempio di Mappa Semantica . . . . .	9
2.2 Scena semantica . . . . .	9
<b>3 Grafo di Scena</b>	<b>11</b>
3.1 Generazione del Grafo di Scena . . . . .	11
3.1.1 Lettura del frame RGB-D . . . . .	11
3.1.2 Inferenza . . . . .	12
3.1.3 Costruzione del grafo . . . . .	13
3.2 Aggiornamento del Mappa Semantica . . . . .	14
3.2.1 Stanza corrente robot . . . . .	14
3.2.2 Proiezione del Camera Frustum . . . . .	14
3.2.3 Controllo della posizione degli oggetti . . . . .	14
3.2.4 Aggiornamento e salvataggio a DB . . . . .	14
3.3 Conclusioni . . . . .	14
<b>4 Riconoscimento di Stanze</b>	<b>15</b>
4.1 Analisi e risultati . . . . .	15
4.2 Conclusioni . . . . .	15
<b>5 Analisi e Risultati</b>	<b>16</b>
5.1 Errore della posizione degli oggetti . . . . .	16
5.2 Punti di forza e svantaggi . . . . .	16
5.2.1 Inferenza efficiente . . . . .	16
5.2.2 Merging efficiente . . . . .	16
<b>6 Conclusioni e Sviluppi Futuri</b>	<b>17</b>
6.1 Miglioramenti . . . . .	17
6.1.1 Database a grafo . . . . .	17
6.1.2 Finetuning OpenPSG . . . . .	17

6.1.3	Object tracking . . . . .	17
6.1.4	Utilizzo di OpenPVSG e Open4PSG . . . . .	17
<b>A</b>	<b>Appendice</b>	<b>18</b>
A.1	RoBee System . . . . .	18
A.1.1	Dashboard and Console . . . . .	18
A.1.2	Infrastructure architecture, microservices and MQTT . . .	18
A.1.3	Maps, navigation and LiDaRs . . . . .	18
A.1.4	Joints and transformations . . . . .	18
A.1.5	Cameras and point cloud . . . . .	18
A.2	Codice . . . . .	18
A.2.1	Costruzione Grafo di Scena . . . . .	18
A.2.2	Aggiornamento Mappa Semantica . . . . .	18

# Introduzione

Negli ultimi anni il campo della robotica ha vissuto un significativo incremento di applicazioni e innovazioni. Lo sviluppo di nuove tecnologie e la disponibilità di nuovi strumenti hanno reso possibile la creazione di robot in grado di svolgere compiti sempre più complessi. La **pianificazione automatica delle missioni** è sempre stata una delle attività di sviluppo in questo campo più affascinanti, pur essendo una delle più tediosa. Con l'avvento di ChatGPT e modelli simili, si è iniziato a pensare di integrare i **Large Language Models**, come alternativa ai classici planner, all'interno del sistema robot, con l'obiettivo di pianificare missioni autonome sulla base della descrizione in linguaggio naturale di ciò che si vuole far eseguire al robot.

La percezione dell'ambiente circostante è dunque una delle attività più importanti per un robot, soprattutto nell'ambito del **Mission Planning**. La capacità di riconoscere gli oggetti e di calcolarne la posizione è fondamentale per poterci interagire. Inoltre, è essenziale potersi localizzare nella mappa, sia in modo geometrico che topologico, in modo da poter pianificare anche eventuali movimenti verso gli oggetti desiderati che si trovano in punti non raggiungibili al momento dal robot.

In questo documento definiremo il significato di **Mappa Semantica**, le ragioni alla base della sua esistenza, la struttura e come viene utilizzata per pianificare le missioni del robot. Successivamente entreremo nel dettaglio del **Grafo di Scena**, come viene generato e tenuto aggiornato con i cambiamenti dell'ambiente. Infine analizzeremo il **Riconoscimento delle Stanze** a partire dalla mappa SLAM generata attraverso i sensori LiDaR del Robot, essenziale per suddividere l'insieme degli oggetti nelle loro stanze e gestire le missioni che necessitano lo spostamento in altre stanze.

## Capitolo 1

# Stato dell'Arte

## Capitolo 2

# Mappe semantiche

Gli esseri umani, talvolta senza rendersene conto, riescono a integrare continuamente nuove informazioni riguardo l'ambiente che li circonda, sia esso una casa, un edificio pubblico o un parco. Questa capacità, conscia e inconscia, è essenziale per la successiva pianificazione di quei obiettivi o movimenti basati sulle informazioni appena apprese.

Così come per gli esseri umani, anche i robot hanno bisogno di informazioni per poter essere considerati "cognitivi" e pianificare rispetto alla propria base di conoscenza. In particolar modo quando l'obiettivo è pianificare missioni data la descrizione in linguaggio in naturale di ciò che il robot deve fare, come in questo caso.

**Esempio** Consideriamo una persona che entra per la prima volta in una biblioteca. Egli osserva scaffali pieni di libri, tavoli per lo studio e un'area dedicata ai computer. Queste informazioni vengono immagazzinate e utilizzate successivamente per trovare un libro specifico o un luogo tranquillo per studiare.

Allo stesso modo, immaginiamo un robot progettato per operare in una casa intelligente. Riceve l'istruzione: "Prendi il libro dal tavolo del soggiorno e portalo in cucina." Per svolgere questo compito, il robot deve comprendere la struttura della casa, identificare il tavolo corretto e navigare verso la cucina.

### 2.1 Definizione

La Mappa Semantica è un grafo orientato  $G_m = (V_m, E_m)$  che rappresenta questa base di conoscenza dove:

- Un nodo può essere un:
  - Nodo stanza;
  - Nodo oggetto;
  - Nodo tag.
- Un arco può rappresentare:
  - La relazione tra due oggetti;
  - Il collegamento tra due stanze;

- L'appartenenza di un oggetto o un tag ad una ed una sola stanza.

Di conseguenza, per trovare gli oggetti o i nodi appartenenti ad una stanza  $s$  è sufficiente considerare il sottografo del nodo stanza  $s$

### 2.1.1 Nodi

#### Nodi oggetto

I nodi oggetto rappresentano gli oggetti riconosciuti all'interno dell'ambiente attraverso la segmentazione panoptica dei frame video proveniente dalle camere del robot. Ogni nodo oggetto ha i seguenti attributi:

- Identificativo: utilizzato per identificare un oggetto all'interno della Mappa Semantica;
- Nome: label inferita dal modello di segmentazione panoptica;
- Posizione: terna  $(x, y, z)$  rappresentante la posizione dell'oggetto all'interno dell'ambiente rispetto alla Reference Posizione;
- Reference Posizione: origine del sistema di riferimento delle posizioni degli oggetti. Può essere l'origine del sistema Mappa o l'origine del sistema Robot;
- Tipo: rappresenta la tipologia dell'oggetto che può essere scelta tra:
  - Pickable: qualora l'oggetto possa essere preso attraverso gli end effectors del robot;
  - Non Pickable: qualora l'oggetto non possa essere preso attraverso gli end effectors del robot;
  - Asset: in tutti gli altri casi (*Per esempio un tavolo*).

I nodi oggetto vengono aggiornati con le inferenze di nuovi frame video: possono dunque essere eliminati dalla mappa semantica qualora un oggetto non si presenti più all'interno della scena oppure aggiornati, per esempio a livello di posizione, qualora l'oggetto venga spostato.

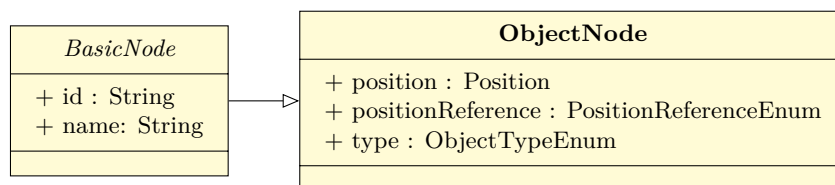


Figura 2.1: Diagramma delle classi - Nodo Oggetto

#### Nodi tag

I nodi tag rappresentano tutti quei riferimenti che vengono utilizzati dal robot per localizzarsi o localizzare oggetti di particolare rilevanza (come la stazione di ricarica o il tavolo di lavoro). Ogni nodo tag ha i seguenti attributi:

- Identificativo: utilizzato per identificare il tag all'interno della Mappa Semantica;
- Nome: assegnato dall'utente;
- Posizione: terna  $(x, y, z)$  rappresentante la posizione del tag all'interno dell'ambiente rispetto all'origine della mappa;
- Dimensione: dimensione del tag in millimetri;
- Di Navigazione: flag che indica se il tag è utilizzato per la navigazione del robot;
- Per Picking: flag che indica se il tag è utilizzato per identificare un oggetto di cui fare il picking con gli end effectors.

I nodi tag sono permanenti all'interno della mappa semantica perché si assume che questi non vengano mai spostati o rimossi dall'ambiente del robot

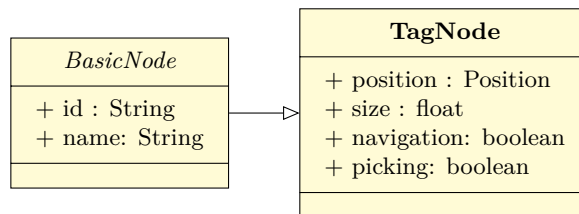


Figura 2.2: Diagramma delle classi - Nodo Tag

### Nodi stanza

I nodi stanza rappresentano un'area semantica all'interno della mappa slam generata attraverso i sensori LiDaR del robot che, data in input ad un algoritmo, questo individua le stanze e ne genera il poligono. Ogni nodo stanza ha i seguenti attributi:

- Identificativo: utilizzato per identificare la stanza all'interno della Mappa Semantica;
- Nome: assegnato dall'utente;
- Segmenti: Lista di segmenti che delimitano il poligono della stanza. Viene usato per verificare se un oggetto appartiene ad una stanza o no;
- Oggetti: Sottografo degli oggetti appartenenti alla stanza.

La presenza di queste aree nella mappa è fondamentale per diverse ragioni:

- Permette di suddividere gli oggetti rispetto alla stanza di appartenenza, facilitando la discriminazione degli omonimi in base alla stanza di appartenenza e aggiungendo **keypoint** per la descrizione in linguaggio naturale di una missione;
- Consentirà l'utilizzo di algoritmi di ricerca su grafo per la pianificazione del percorso per raggiungere gli oggetti;
- Permetterà di creare percorsi pianificati che evitano determinate stanze.



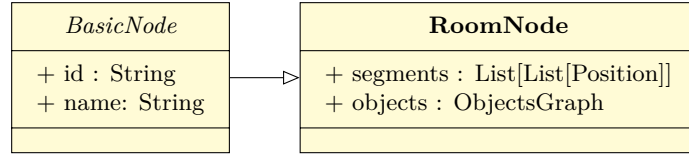


Figura 2.3: Diagramma delle classi - Nodo Stanza

## 2.1.2 Archi

### Archi tra stanze

Gli archi tra i nodi di tipo stanza rappresentano il collegamento diretto tra due stanze.

### Archi tra oggetti

Gli archi orientati tra i nodi di tipo oggetto rappresentano la relazione tra due oggetti. L'etichetta associata ad ogni arco appartiene all'insieme delle relazioni che possono essere inferite dal modello PSGTr. Queste informazioni sono importanti per poter pianificare task all'interno della missione che supportano il raggiungimento dell'obiettivo.

Per esempio, immaginiamo la missione "Prendi la bottiglia". Se è presente un ostacolo davanti alla bottiglia, rispetto alla posizione di presa del robot, questo deve prima pianificare lo spostamento dell'ostacolo. Ecco il motivo per il quali vi è la necessità di rappresentare queste relazioni tra oggetti.

## 2.1.3 Grafo stanze

Definiamo il grafo delle stanze come il sottografo  $(V_s, E_s)$  tale che:

- $V_s = \{v \in V_m \mid v \text{ è un nodo stanza}\}$
- $E_s = \{(v, u) \in E_m \mid v \in V_s \wedge u \in V_s\}$

dove  $V_m$  è l'insieme dei vertici del grafo della Mappa Semantica e  $E_m$  è l'insieme degli archi del grafo della Mappa Semantica.

Questo grafo, viene generato a partire dall'algoritmo di riconoscimento delle stanze, illustrato nel Capitolo 4, ed è la prima parte di Mappa Semantica creata, in concomitanza con la generazione della mappa slam. Solo successivamente sarà possibile la costruzione del grafo degli oggetti.

## 2.1.4 Grafo Oggetti

Definiamo il grafo degli oggetti (o grafo di scena) come il sottografo  $(V_o, E_o)$  tale che:

- $V_o = \{v \in V_m \mid v \text{ è un nodo oggetto o tag}\}$
- $E_o = \{(v, u) \in E_m \mid v \in V_o \wedge u \in V_o\}$

dove  $V_m$  è l'insieme dei vertici del grafo della Mappa Semantica e  $E_m$  è l'insieme degli archi del grafo della Mappa Semantica.

Una versione grezza del grafo degli oggetti viene generata dal modello PSG-Tr di [9]. Successivamente, come verrà mostrato nel Capitolo 3, questa verrà fusa con la Mappa Semantica rendendo dunque la base di conoscenza coerente rispetto lo stato dell'ambiente attuale.

### 2.1.5 Esempio di Mappa Semantica

Di seguito un esempio di mappa semantica e i vari sottografi.

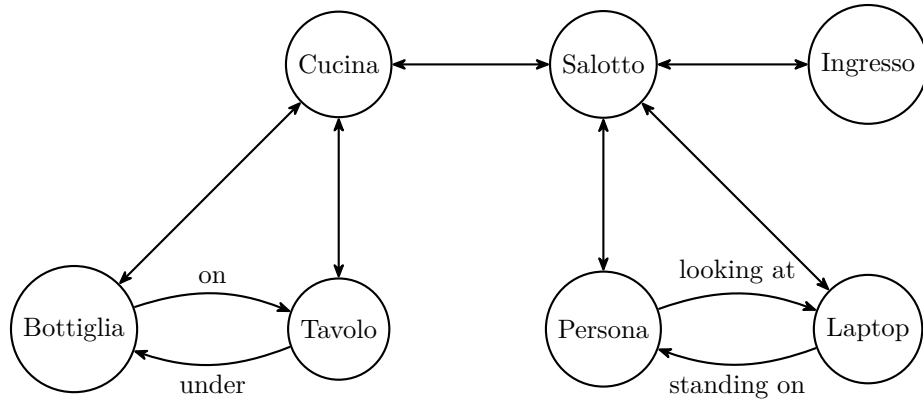
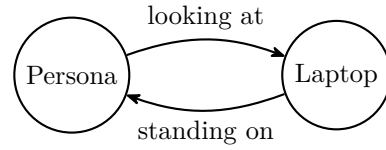
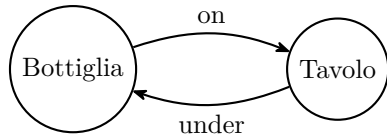


Figura 2.4: Mappa Semantica



(a) Grafo delle stanze



(b) Grafo degli oggetti della stanza Cucina (c) Grafo degli oggetti della stanza Salotto

## 2.2 Scena semantica

La Scena Semantica non è altro che un sottografo  $G_{ss} = (V_{ss}, E_{ss})$  della mappa semantica che viene costantemente aggiornato con le ultime inferenze, senza considerare la suddivisione tra stanze e i precedenti oggetti individuati, dove:

- $V_{ss} = \{v \in V_m \mid v \text{ è un nodo oggetto}\}$

- $E_{ss} = \{(v, u) \in E_m \mid v \in V_{ss} \wedge u \in V_{ss}\}$

In poche parole è il grafo della scena direttamente inferito dal modello seppur con alcune differenze:

- I nodi oggetto presenti sono i soli rilevati entro un certo range dalla camera del robot. Questo range è impostato dalle configurazioni del servizio
- I nodi oggetto hanno la posizione rispetto alla terna del robot e non della mappa

## Capitolo 3

# Grafo di Scena

In questo capitolo verrà affrontata la generazione del grafo di scena dato un frame e l'aggiornamento della Mappa Semantica con queste nuove informazioni per mantenerla aggiornata rispetto all'ambiente.

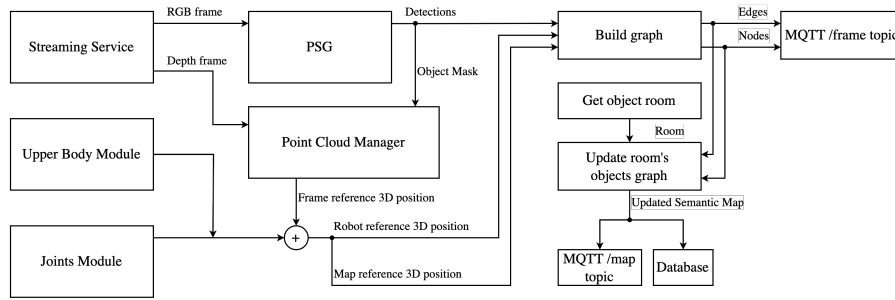


Figura 3.1: Schema dei flussi dati per la generazione del grafo di scena e aggiornamento della mappa semantica

### 3.1 Generazione del Grafo di Scena

La generazione del grafo di scena è un passo fondamentale per il mantenimento della coerenza tra Mappa Semantica e Ambiente reale.

Il grafo di scena è una struttura dati che rappresenta gli oggetti presenti nell'ambiente e le relazioni tra loro composto da nodi e archi. I nodi rappresentano gli oggetti, mentre gli archi rappresentano le relazioni tra gli oggetti.

#### 3.1.1 Lettura del frame RGB-D

All'interno dell'architettura cloud-native di Robee vi è la presenza di un pod chiamato "Streaming Module" il cui compito è streammare il feed video delle camera sul pod redis del robot, in modo che gli altri servizi o moduli possano accedere a questi dati tramite l'utilizzo di librerie wrapper, rendendo il tutto agnostico rispetto alla tipologia e modello di videocamera montati.

### 3.1.2 Inferenza

Ogni frame ricevuto dal feed video viene successivamente dato in input al modello PSGTr [9] che restituisce un oggetto di tipo Detections il quale contiene i seguenti dati:

- labels: lista con lunghezza pari al numero di oggetti rilevati. Ogni valore indica la label corrispondente all'  $i$ -esimo oggetto. Per esempio, l'oggetto  $i$ -esimo ha label  $labels[i]$ ;
- masks: lista contenente le maschere di ogni oggetto rilevato;
- bboxes: lista contenente le bounding boxes di ogni oggetto rilevato;
- rel\_pair\_idxes: lista con lunghezza pari al numero di relazioni tra oggetti rilevate. Ogni valore è a sua volta un array di dimensione due dove il primo elemento è l'oggetto target della relazione e il secondo è l'oggetto sorgente della relazione;
- rel\_labels: lista con lunghezza pari al numero di relazioni tra oggetti rilevate. Ogni valore indica la label della  $i$ -esima relazione
- rel\_dists: lista con lunghezza pari al numero di relazione tra oggetti rilevate. Ogni valore indica la probabilità associata alla  $i$ -esima relazione.

Questi dati vengono successivamente utilizzati per la costruzione del grafo di scena mediante l'algoritmo seguente.

#### Panoptic Scene Graph - Transformer

Il modello PSGTr [9] è un modello di deep learning a singolo stato basato su architettura Transformer [3] il cui obiettivo è quello di generare una rappresentazione a grafo della scena data la segmentazione panottica piuttosto che le bounding box degli oggetti rilevati.

**Training** Il modello, per quanto riguarda gli oggetti, è stato addestrato su un dataset composto da 49mila immagini annotate basato su COCO [1] e Visual Genome [2]. Per le relazioni hanno estratto e costruito un dataset di 56 predicati a partire da dataset come VG-150 [4], VrR-VG [6] and GQA [5].

**Segmentazione Panoptica** La segmentazione panoptica individua gli oggetti e assegna a ogni pixel la label della classe dell'oggetto a cui appartengono. L'utilizzo di questa rispetto alle bounding da notevoli vantaggi:

- Garantisce una localizzazione più precisa degli oggetti, segmentandoli a livello di pixel e riducendo la presenza di pixel rumorosi o ambigui tipici delle bounding box, che spesso includono porzioni di altre categorie o oggetti;
- Copre l'intera scena di un'immagine, inclusi gli sfondi, offrendo una comprensione più completa del contesto rispetto alle bounding box, che tendono a trascurare importanti informazioni di sfondo;
- Riduce anche le informazioni ridondanti o irrilevanti presenti nei dataset basati su bounding box, focalizzandosi sulla segmentazione degli oggetti piuttosto che sulle loro parti.

**Funzionamento di PSGTr** L'architettura di PSGTr è basata su DETR [7] e HOI [8]. Il modello predice triple (*soggetto, predicato, verbo*) e la localizzazione degli oggetti simultaneamente.

**Pipeline PSGTr** Attraverso una backbone CNN, PSGTr estrae le features dell'immagine e i positional encodings che, insieme alle triplet queries, vengono dati in input al transformer encoder-decoder. In questo processo, l'obiettivo è che le query apprendano la rappresentazione del grafo di scena a triple in modo che per ognuna di esse, le predictions di (*soggetto, predicato, verbo*) possano successivamente essere estratte da tre Feed Forward Network. Infine, il task di segmentazione viene eseguito da due head panottiche, una per il soggetto e una per l'oggetto della relazione.

### 3.1.3 Costruzione del grafo

L'obiettivo di questo step è la costruzione del grafo di scena rispetto all'ultimo frame. Per farlo, è necessario estrarre i dati dai risultati dell'inferenza di PSGTr e calcolare quei valori che dipendono dal sistema robot, come la posizione. L'algoritmo di costruzione del grafo è costituito da 2 fasi principali:

- Costruzione dei nodi per la scena semantica e per la mappa semantica:
  - Calcolo della posizione dell'oggetto
- Costruzione degli archi per la scena semantica e per la mappa semantica

#### Costruzione dei nodi

**Estrazione dati dai risultati** L'oggetto MMDetResult ritornato dalla funzione di inferenza del modello, come detto precedentemente, possiede un attributo *labels* che è una lista con lunghezza pari al numero di oggetti rilevati dove il valore *i*-esimo, indica l'indice della classe di appartenenza dell'oggetto *i*. Lo stesso meccanismo vale anche per le maschere.

---

**Algorithm 1** Estrazione classi e maschere degli oggetti individuati

---

```

1: obj_classes  $\leftarrow$  []
2: obj_masks  $\leftarrow$  []
3: obj_labels_ids  $\leftarrow$  detectionResults.labels
4: for i = 0 to obj_labels_ids.length do
5:   obj_classes.append(PSG_CLASSES[obj_labels_ids[i]])
6:   obj_masks.append(detectionResults.masks[i])
7: end for

```

---

**Calcolo posizioni 3D** Per ogni oggetto, si estrae la posizione 3D nella mappa del robot in modo che questo possa successivamente localizzarlo e raggiungerlo.

**Calcolo della  $z$  di un oggetto** Le maschere generate dal modello consentono di calcolare il centroide  $(x_i, y_i)$  dell'oggetto  $i$ -esimo. Tuttavia, queste maschere forniscono solo un valore in due dimensioni. Per il calcolo del valore  $z_i$  si utilizza il Point Cloud che, combinando il frame RGB con il frame Depth, permette di ottenere una rappresentazione 3D della scena. Per ogni oggetto  $i$ , si maschera il Point Cloud con la maschera  $i$ -esima e si calcola  $z_i$  come valore mediano tra le  $z_s$  di tutti i punti mascherati.

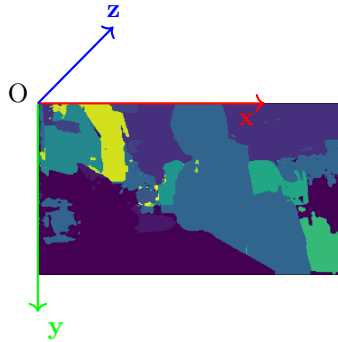


Figura 3.2: Sistema di riferimento del risultato  $(x, y, z)$

Calcolo posizione 3D rispetto alla camera

Calcolo posizione 3D rispetto alla mappa

Costruzione degli archi

## 3.2 Aggiornamento del Mappa Semantica

### 3.2.1 Stanza corrente robot

### 3.2.2 Proiezione del Camera Frustum

### 3.2.3 Controllo della posizione degli oggetti

### 3.2.4 Aggiornamento e salvataggio a DB

## 3.3 Conclusioni

## Capitolo 4

# Riconoscimento di Stanze

### 4.1 Analisi e risultati

### 4.2 Conclusioni



## Capitolo 5

# Analisi e Risultati

### 5.1 Errore della posizione degli oggetti

### 5.2 Punti di forza e svantaggi

#### 5.2.1 Inferenza efficiente

#### 5.2.2 Merging efficiente

## Capitolo 6

# Conclusioni e Sviluppi Futuri

### 6.1 Miglioramenti

#### 6.1.1 Database a grafo

#### 6.1.2 Finetuning OpenPSG

#### 6.1.3 Object tracking

#### 6.1.4 Utilizzo di OpenPVSG e Open4PSG

# Appendice A

## Appendice

### A.1 RoBee System

A.1.1 Dashboard and Console

A.1.2 Infrastructure architecture, microservices and MQTT

A.1.3 Maps, navigation and LiDaRs

A.1.4 Joints and transformations

A.1.5 Cameras and point cloud

### A.2 Codice

A.2.1 Costruzione Grafo di Scena

Costruzione Nodi

Costruzione Archi

A.2.2 Aggiornamento Mappa Semantica

Proiezione Camera Frustum

Controllo se l'oggetto appartiene al frustum

Recupero Stanza corrente Robot

# Glossario

**backbone** Architettura di rete neurale pre-addestrata che funge da base per ulteriori sviluppi e adattamenti specifici di una particolare applicazione. Grazie al transfer learning è possibile costruire architetture per task complessi sopra a questo modello. Utilizzata molto spesso nella visione artificiale come supporto. . 13

# Bibliografia

- [1] T.-Y. Lin, M. Maire, S. Belongie et al., *Microsoft COCO: Common Objects in Context*, 2014. arXiv: 1405.0312 [cs.CV].
- [2] R. Krishna, Y. Zhu, O. Groth et al., «Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations,» *International Journal of Computer Vision*, vol. 123, mag. 2017. DOI: 10.1007/s11263-016-0981-7.
- [3] A. Vaswani, N. Shazeer, N. Parmar et al., «Attention Is All You Need,» *CoRR*, vol. abs/1706.03762, 2017. arXiv: 1706.03762. indirizzo: <http://arxiv.org/abs/1706.03762>.
- [4] D. Xu, Y. Zhu, C. B. Choy e L. Fei-Fei, «Scene Graph Generation by Iterative Message Passing,» in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [5] D. A. Hudson e C. D. Manning, «GQA: a new dataset for compositional question answering over real-world images,» *CoRR*, vol. abs/1902.09506, 2019. arXiv: 1902.09506. indirizzo: <http://arxiv.org/abs/1902.09506>.
- [6] Y. Liang, Y. Bai, W. Zhang, X. Qian, L. Zhu e T. Mei, «Rethinking Visual Relationships for High-level Image Understanding,» *CoRR*, vol. abs/1902.00313, 2019. arXiv: 1902.00313. indirizzo: <http://arxiv.org/abs/1902.00313>.
- [7] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov e S. Zagoruyko, «End-to-End Object Detection with Transformers,» *CoRR*, vol. abs/2005.12872, 2020. arXiv: 2005.12872. indirizzo: <https://arxiv.org/abs/2005.12872>.
- [8] C. Zou, B. Wang, Y. Hu et al., «End-to-End Human Object Interaction Detection with HOI Transformer,» *CoRR*, vol. abs/2103.04503, 2021. arXiv: 2103.04503. indirizzo: <https://arxiv.org/abs/2103.04503>.
- [9] J. Yang, Y. Z. Ang, Z. Guo, K. Zhou, W. Zhang e Z. Liu, «Panoptic Scene Graph Generation,» in *ECCV*, 2022.