

# Navigazione basata su inseguimento di frecce $Relazione\ di\ progetto$

Progetto del corso Robotica (principi e progetto) Università degli Studi di Bergamo A.A. 2019/2020

> Calegari Andrea - xxxxxxx Paganessi Andrea - 1040658 Piffari Michele - xxxxxxx

> > November 25, 2019

# Contents

1	Stato dell'arte			
	1.1	SAL - Stato avanzamento lavori		
<b>2</b>	Car	nera		
		Scelta della camera		
		Vericale o orizzontale?		
	2.3	Come ottenere le immagini dalla camera?		
3		stione delle maschere		
	3.1	HSV		
	3.2	Frecce o cerchi?		
	3.3	Maschere		
	3.4	Erosione e dilatazione		

iv CONTENTS

# List of Figures

1.1	Base robotica addetta alla movimentazione	2
1.2	Base verticale sulla quale andare ad inserire la camera	2
2.1	Camera utilizzata inizialmente	6

vi LIST OF FIGURES

#### 1

### Stato dell'arte

Obbiettivo: andare a implementare sistema di visual navigation per la base robotica in figura 1.1.

#### 1.1 SAL - Stato avanzamento lavori

- Analizzato codice Out-Of-Box del progetto dello scorso anno
  - Il codice preso non aveva main: creato
  - Compresa struttura pub/sub
  - Analizzati topic/nodes pubblicati
- Cambio camera. Perchè? Prestazioni scarse al variare della luce
- Nuova camera -> ueye cam
- Fatta funzionare nuova camera
  - Demo (programma già fornito con la camera)
  - Ros -> utilizzato file debug-launch (inserire caratteristiche che la camera fornisce quando parte lo script).
- Scelta la posizione della camera: verticale inclinata e non orizzontale
- Progetto A.A. utilizza formule vecchie
- Problema CPU consuming (problema intrinseco della camera)
- Memory problem -> risolto con free
- Doppie maschere: frecce di due colori
- Aggiunte queste features:
  - Gaussian blur
  - Brightness
  - Erosion Dilatation
- Fatta erosione solo sulle frecce vicine (quelle nella metà inferiore del frame), mentre invece, le frecce nella metà superiore non vengono erose ma solo dilatate.
- Problema inizializzazione che mostrava rettangoli bianchi su alcune immagini intermedie nelle maschere
- Aggiunta distanza tra centri con tracciamente linea

1. STATO DELL'ARTE

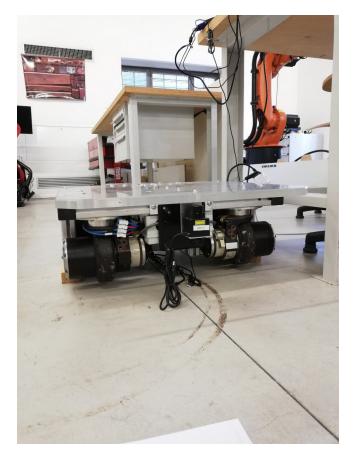


Figure 1.1: Base robotica addetta alla movimentazione



Figure 1.2: Base verticale sulla quale andare ad inserire la camera

• Prendiamo la freccia più vicina e analizziamo i dati relativi solo a questa freccia: supponendo che tutte le frecce siano uguali, è ovvio che l'area maggiore è quella della freccia più vicina (TODO: da mettere come giustificazione del codice che scriveremo)

Per creare grafi della struttura del codice ROS vedi e comando rqt.

1. STATO DELL'ARTE

2

## Camera

#### 2.1 Scelta della camera

Ad inizio del progetto siamo andati a lavorare con una camera *SpotLight Pro Webcam*, fornita dalla casa *Trust* (fig. 2.1): questa camera abbiamo però visto che non era in grado di dare sufficienti garanzie di funzionamento stabile in alcune delle più comuni condizioni luminose.

Si è deciso quindi di passare ad una camera di tipo industriale, in grado di fornire delle prestazioni più stabili e affidabili. La scelta è ricaduta sulla camera della casa produttrice IDS (Imaging Development System): si tratta del modello UI-1221LE-C-HQ equipaggiata con la lente BM2420 della casa Lensagon.

#### 2.2 Vericale o orizzontale?

#### 2.3 Come ottenere le immagini dalla camera?

6 2. CAMERA



Figure 2.1: Camera utilizzata inizialmente

## 3

## Gestione delle maschere

- 3.1 HSV
- 3.2 Frecce o cerchi?
- 3.3 Maschere
- 3.4 Erosione e dilatazione

# **Bibliography**

- $[1] \ \textit{Descrizione della camera} \ \texttt{https://en.ids-imaging.com/store/ui-1221le-rev-2.html}$
- $[2] \ \textit{Manuale della camera} \ \texttt{https://en.ids-imaging.com/IDS/datasheet\_pdf.php?sku=ABO2422}$
- [3] Manuale della lente https://www.lensation.de/product/BM2420/
- [4] Ueye cam e ROS http://wiki.ros.org/ueye