

Università degli Studi di Padova – Dipartimento di Ingegneria Industriale

Corso di Laurea in Ingegneria Aerospaziale

***Relazione per la prova finale
«Implementazione di Sistemi di
Rilevamento Ostacoli e Navigazione
Autonoma su UAV»***

Tutor universitario: Prof. Carlo Bettanini Fecia Di Cossato

Laureando: *Matteo Faggian*

Padova, 22/09/25

OBIETTIVI DELLA TESI

- Varie tipologie di UAV.
- Capire il sistema di rilevamento ostacoli (Varie tecniche possibili).
- Analizzare le tecniche di mappatura e localizzazione, con particolare attenzione alla tecnologia VSLAM.
- Capire algoritmi avanzati per la pianificazione del percorso, come A*.

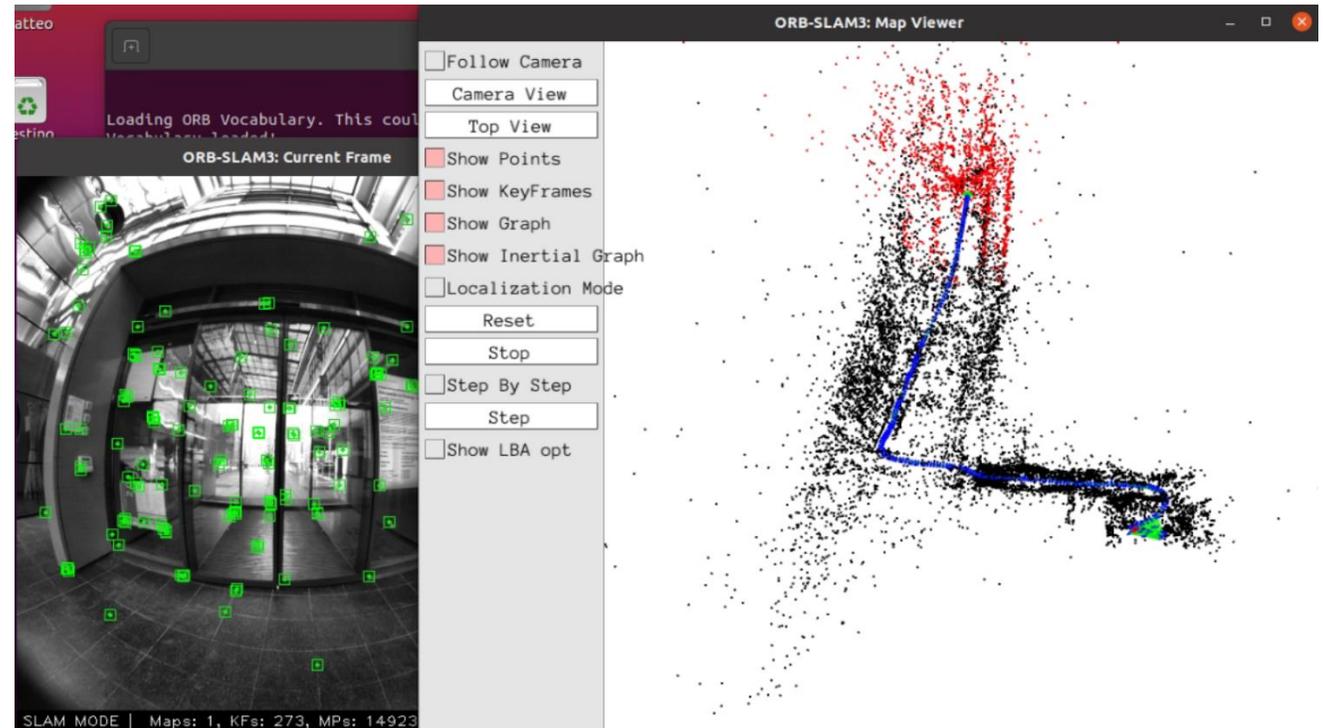


Figura 0. Simulazione ORB SLAM

Tipologie

- **Multirotori:** Elevata manovrabilità e volo stazionario, ideali per fotografia, ricerca e monitoraggio.
- **Monorotori:** Maggiore autonomia per missioni a lunga durata.
- **VTOL:** Uniscono i vantaggi del decollo/atterraggio verticale con il volo orizzontale.



Figura 1. drone Predator [01].



Figura 2. Esacottero [02].



Figura 3. Drone a decollo verticale [03].

Militare

- Sorveglianza e Ricognizione
- Operazioni di attacco
- Supporto logistico



Figura 4. Drone MQ-9 Reaper [04].

Civile

- Riprese aeree
- Monitoraggio urbano
- Controllo ambientale
- Agricoltura di precisione
- Consegne di oggetti
- Emergenze (ricerca e soccorso)



Figura 5. Esacoptero con telecamera termica [05].

Industriale

- Ispezione di infrastrutture
- Monitoraggio delle risorse naturali
- Gestione ambientale
- Controllo qualità nei processi produttivi



Figura 6. Drone per trasporto Industriale [06].

Benefici:

- Costi operativi ridotti e maggiore flessibilità
- Consentono operazioni in ambienti difficili



Figura 7. Drone anti incendio [26].

Limitazioni e Sfide:

- Presentano autonomia limitata
- Complessità normative
- Vulnerabilità alle interferenze elettromagnetiche.



Figura 8. Drone per trasporto Industriale [06].

➤ Piattaforme hardware:

NVIDIA Jetson, Raspberry Pi, Qualcomm Flight , Pixhawk.

➤ Sensori:

Telecamere, LIDAR, radar, ultrasuoni, GPS.



Figura 9. Nvidia Jetson [09].



Figura 10. Lidar a 360. [10].



Figura 11. Camere per la Vslam [11].

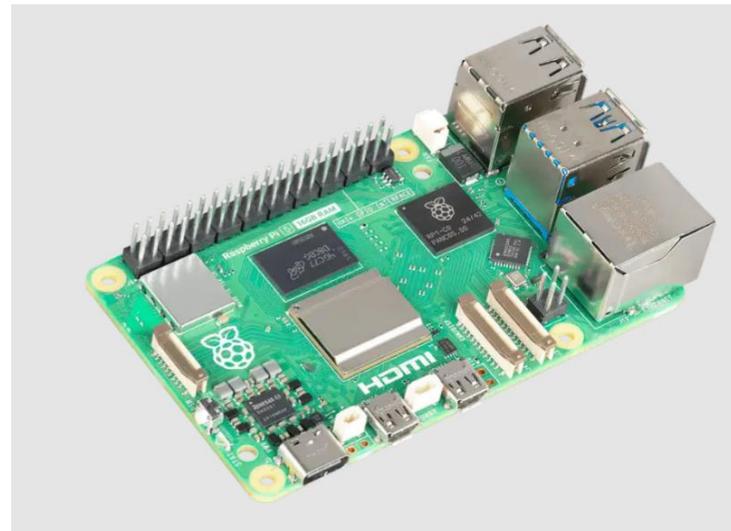


Figura 12. Raspberry [12].

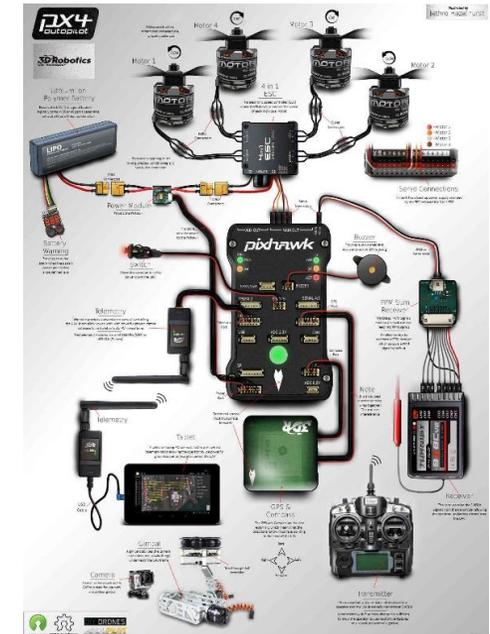


Figura 13. Pixhawk [13].

Definizione e Importanza del CAS

- Il Collision Avoidance System (CAS) utilizza sensori e algoritmi per rilevare ostacoli in tempo reale e prevenire collisioni tra UAV e altri oggetti.
- Operando in autonomia, il CAS permette manovre correttive rapide, migliorando la sicurezza e l'affidabilità operativa in ambienti complessi e affollati.



Figura 14. LIDAR [07].

Stato dell'arte nei sistemi CAS:

- Telecamere
- LIDAR
- Radar e ultrasuoni alternative in ambienti avversi
- Fusione di sensori



Figura 15. Camere [08].

Principi di Base della Localizzazione e Mappatura Simultanea.

Localizzazione ↔ Mappatura → Rettifica dell'immagine
(Solo stereo camere)



Immagine
Rettificata

$$z = \frac{Bf}{X_p - X_{p'}}$$



$$e = M \begin{bmatrix} O' \\ 1 \end{bmatrix} = M \begin{bmatrix} -R^T T \\ 1 \end{bmatrix} = K[I|0] \begin{bmatrix} -R^T T \\ 1 \end{bmatrix} = -KR^T T$$

$$e' = M' \begin{bmatrix} O \\ 1 \end{bmatrix} = M' \begin{bmatrix} 0 \\ 1 \end{bmatrix} = K'[R T] \begin{bmatrix} 0 \\ 1 \end{bmatrix} = K' T$$

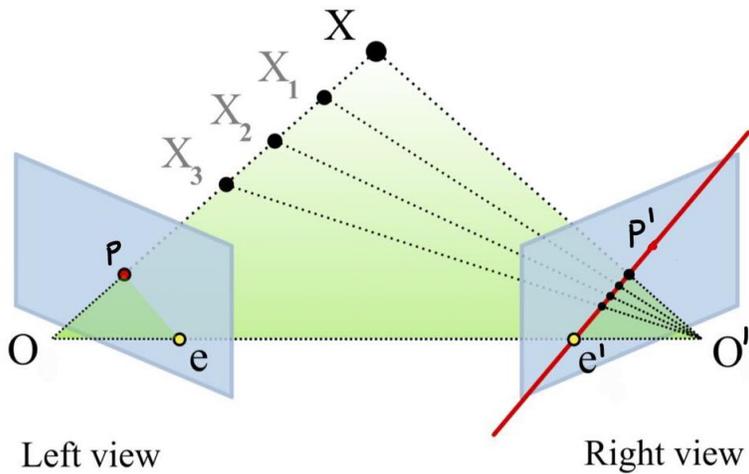


Figura 18. Rettificazione Immagine.

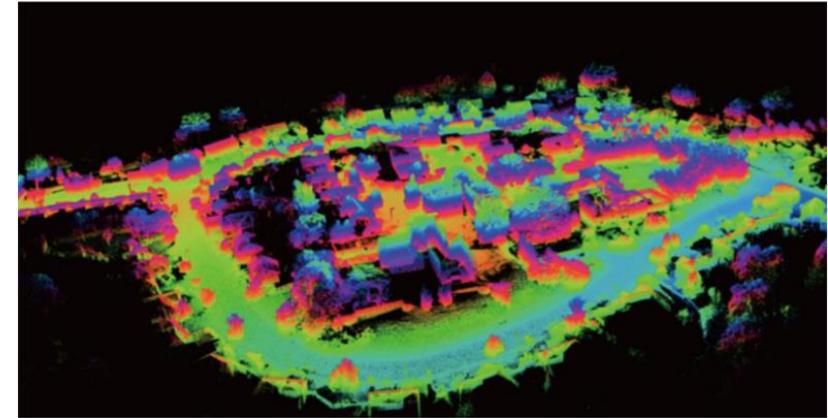
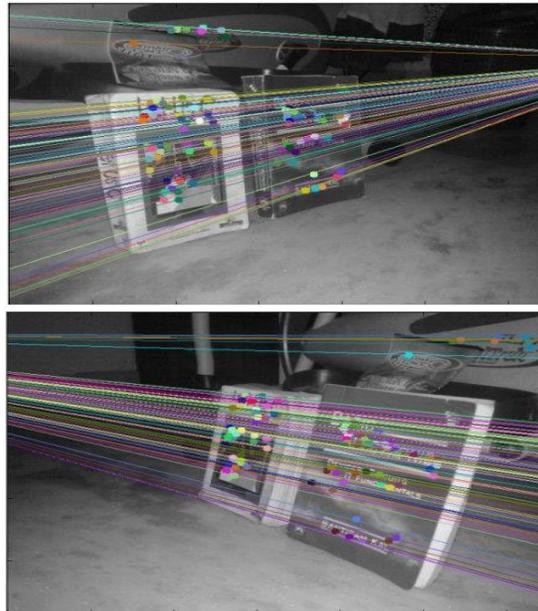


Figura 16. Vslam [14].

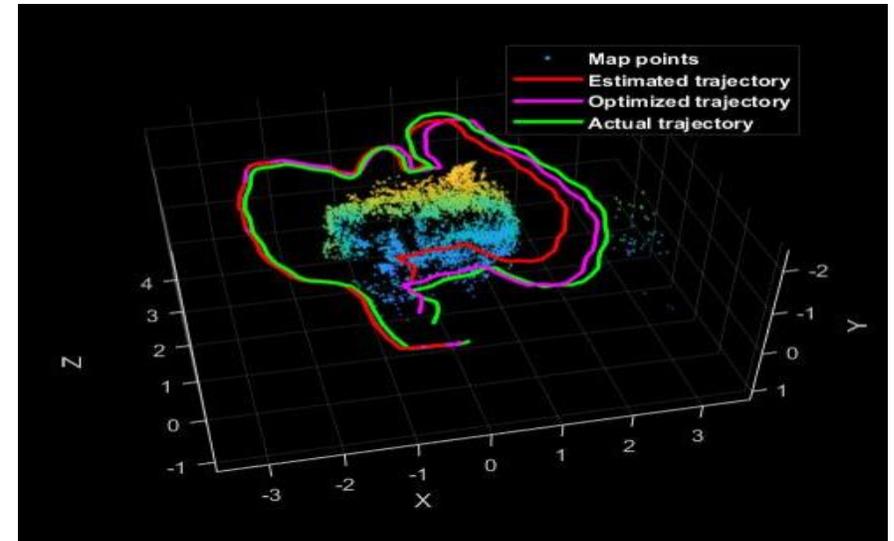


Figura 17. Vslam monoculare [16].

Rilevamento Feature.

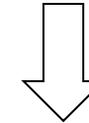
- Vengono utilizzati gli algoritmi elencati
- Tramite telecamere monoculari o binoculari

Algoritmi:

- SIFT (Scale-Invariant Feature Transform)
- SURF (Speeded-Up Robust Features)
- ORB (Oriented FAST and Rotated BRIEF)
- LSD (Large Scale Direct)
- RTAB (Real Time Appearance Based)

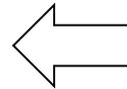
Descrizioni delle feature.

- Viene creato un descrittore che salva le informazioni



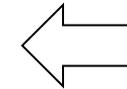
Localizzazione.

- Stima della posizione.
- Problema PnP(perspective-n-point)



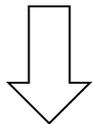
Triangolazione.

- Se monoculare confronta un immagine dopo l'altra
- Binoculare confronta le due immagini contemporaneamente



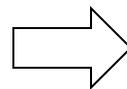
Traking delle feature.

- Confronta le immagini successive per calcolare il movimento della camera



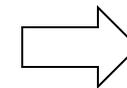
Costruzione mappa.

- Bundle Adjustment
- Costruzione della mappa in tempo reale



Loop closing.

- Riconoscimento di luoghi già visti.
- Ottimizza e coregge errori accumulati



Creazione mappa.

- Il sistema va a creare una mappa con le feature trovate.

ORB-SLAM2 (oriented FAST rotated BRIEFF)

ORB-SLAM2 Costituito da tre thread principali:

- **Tracking:** Utilizza algoritmi oFAST e rBRIEFF e stima la posa della telecamera minimizzando l'errore di proiezione(PnP problem).
- **Local Mapping:** Costruisce in tempo reale una mappa locale utilizzando una struttura ad Octree e keyframes.

Errore di proiezione per la stima della posa.

$$\arg \min_{R_{cw}, t_{cw}} \sum_{i \in \mathcal{X}} \rho (\| \mathbf{x}_i - \pi(R_{cw} \mathbf{X}_i + \mathbf{t}_{cw}) \|_{\Sigma_i}^2)$$

- **Loop Closing:** Identifica posizioni già visitate per correggere l'errore.

$$\mathbf{e}_{i,j} = \log(\mathbf{T}_{i,j}^{-1} \mathbf{T}_i^{-1} \mathbf{T}_j)^\vee,$$

errore di
trasformazione
tra due pose.

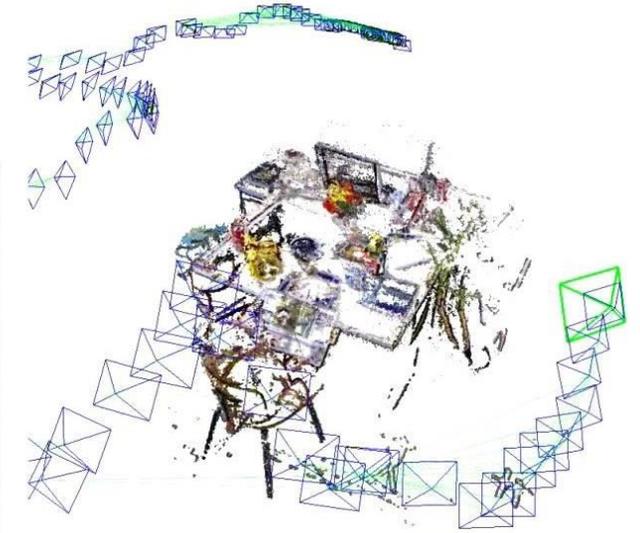
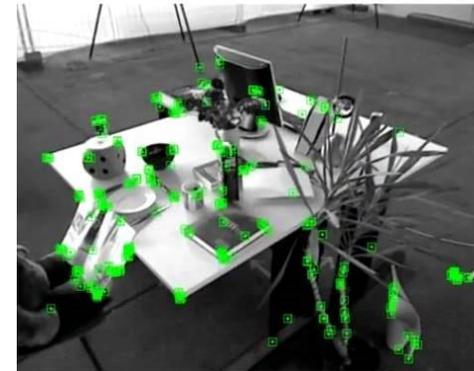


Figura 19. ORB-Slam [17].

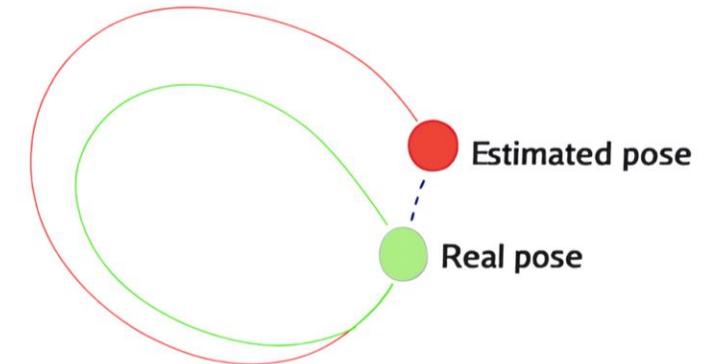


Figura 20. Loop closing.

Dettagli su FAST e BRIEF:

➤ FAST (Feature from Accelerated Segment Test):

Rileva i punti di interesse (feature).

Presenta molti errori.

➤ Oriented FAST

Indipendente dall'orientamento ovvero adotta l'invarianza di scala e rotazione.

$$m_{pq} = \sum_{x,y \in B} x^p y^q I(x,y), \quad p, q = \{0, 1\}$$

Momenti D'immagine
I = "massa luminosa"

$$C = \begin{pmatrix} m_{10} & m_{01} \\ m_{00} & m_{00} \end{pmatrix}$$

$$\theta = \arctan\left(\frac{m_{01}}{m_{10}}\right)$$

Patch 31x31

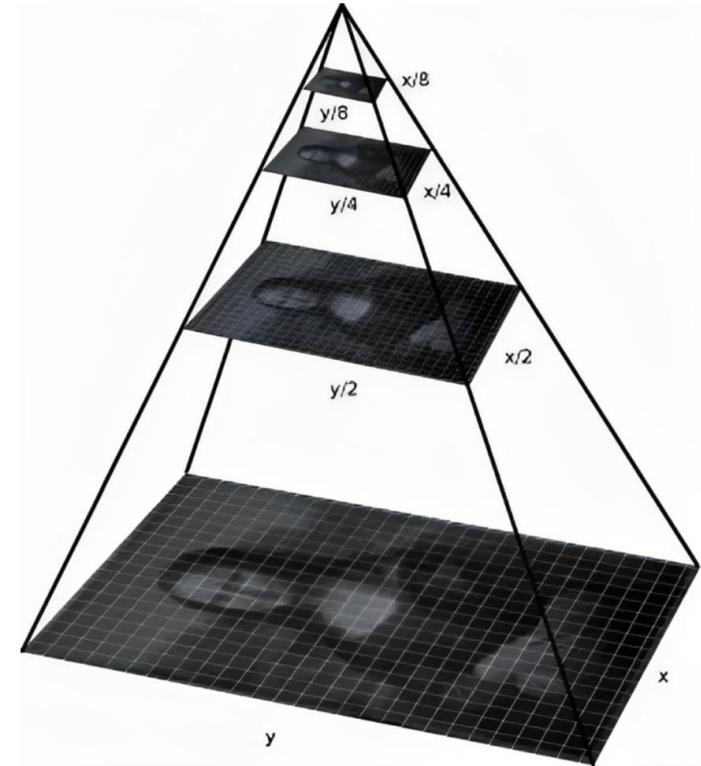
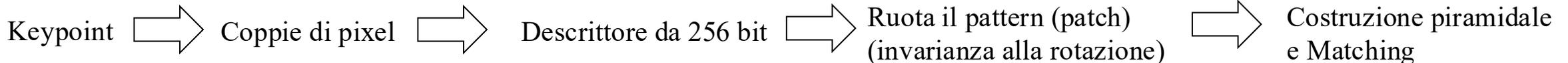


Figura 21. Esempio di costruzione piramidale [21]

➤ BRIEF (Binary Robust Independent Elementary Features):

➤ Rotated BRIEF (rBRIEF) invariabile rispetto alla rotazione, migliorando la robustezza per immagini che potrebbero essere ruotate.



LSD-SLAM (Large-Scale Direct Monocular SLAM)(Mono)

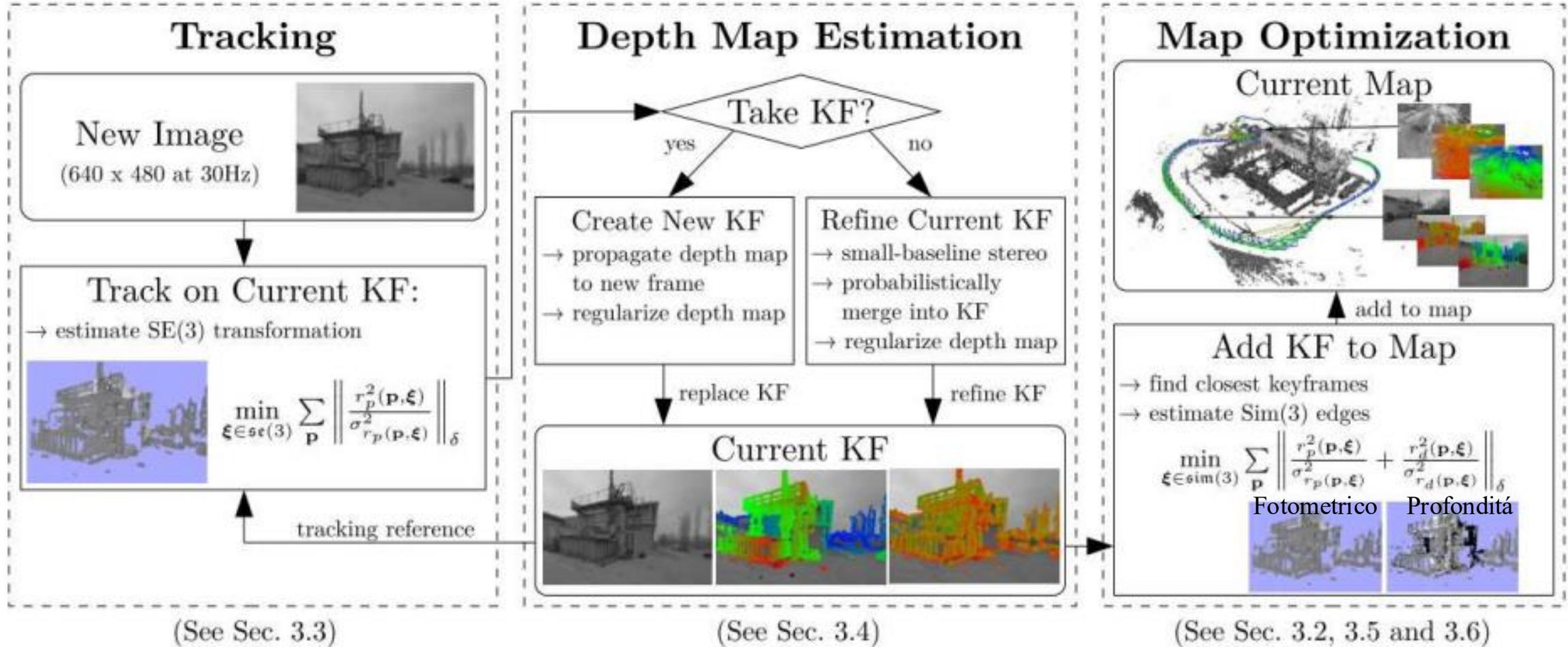


Figura 22. Mappa funzionamento LSD-Slam[25]

LSD-SLAM

LSD-SLAM (Large-Scale Direct Monocular SLAM)

- Utilizza un metodo diretto
- Lavora direttamente con i pixel dell'immagine senza estrarre feature.

Tracking

$$E_p(\xi_{ji}) = \sum_{\mathbf{p} \in \Omega_{D_i}} \left\| \frac{r_p(\mathbf{p}, \xi_{ji})}{\sigma_{r_p(\mathbf{p}, \xi_{ji})}^2} \right\|_{\delta}^2$$

Energia fotometrica

$$r_p(\mathbf{p}, \xi_{ji}) := I_i(\mathbf{p}) - I_j(\omega(\mathbf{p}, D_i(\mathbf{p}), \xi_{ji}))$$

Differenza di Intensità tra Reference frame e keyframe (residuo)

$$\sigma_{r_p(\mathbf{p}, \xi_{ji})}^2 := 2\sigma_I^2 + \left(\frac{\partial r_p(\mathbf{p}, \xi_{ji})}{\partial D_i(\mathbf{p})} \right)^2 V_i(\mathbf{p})$$

varianza di rumore per pixel

➤ Map Optimization

Ora la mappa è costruita, dobbiamo solo ottimizzarla.

$$E(\xi_{W_1} \dots \xi_{W_n}) := \sum_{(\xi_{ji}, \Sigma_{ji}) \in \mathcal{E}} (\xi_{ji} \circ \xi_{W_i}^{-1} \circ \xi_{W_j})^T \Sigma_{ji}^{-1} (\xi_{ji} \circ \xi_{W_i}^{-1} \circ \xi_{W_j})$$

➤ Depth Map Estimation

Generazione mappa semi-densa.
Scelta dei frame ottimali.

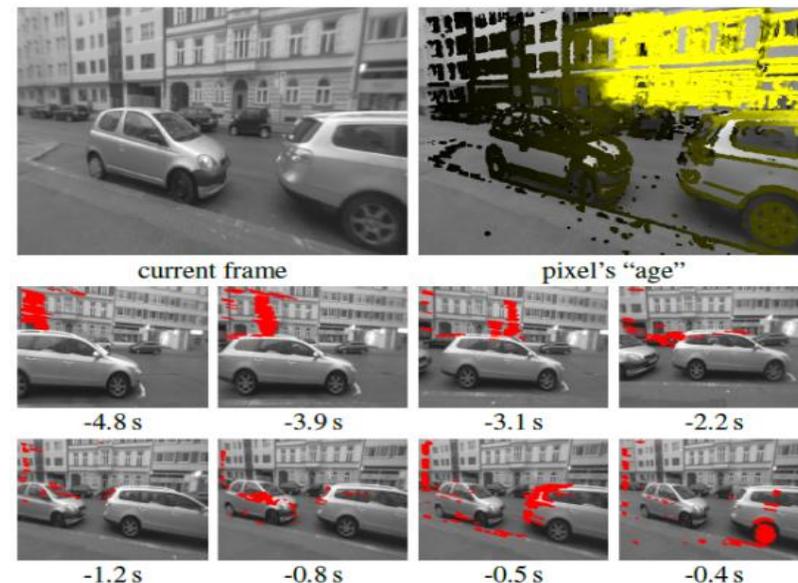
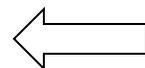


Figura 23. LSD-Slam [25].

Matching dei frame.

Errore tra due immagini consecutive

$$E_{SSD}(u) = \sum_i [I_1(x_i + u) - I_0(x_i)]^2$$



LSD-SLAM

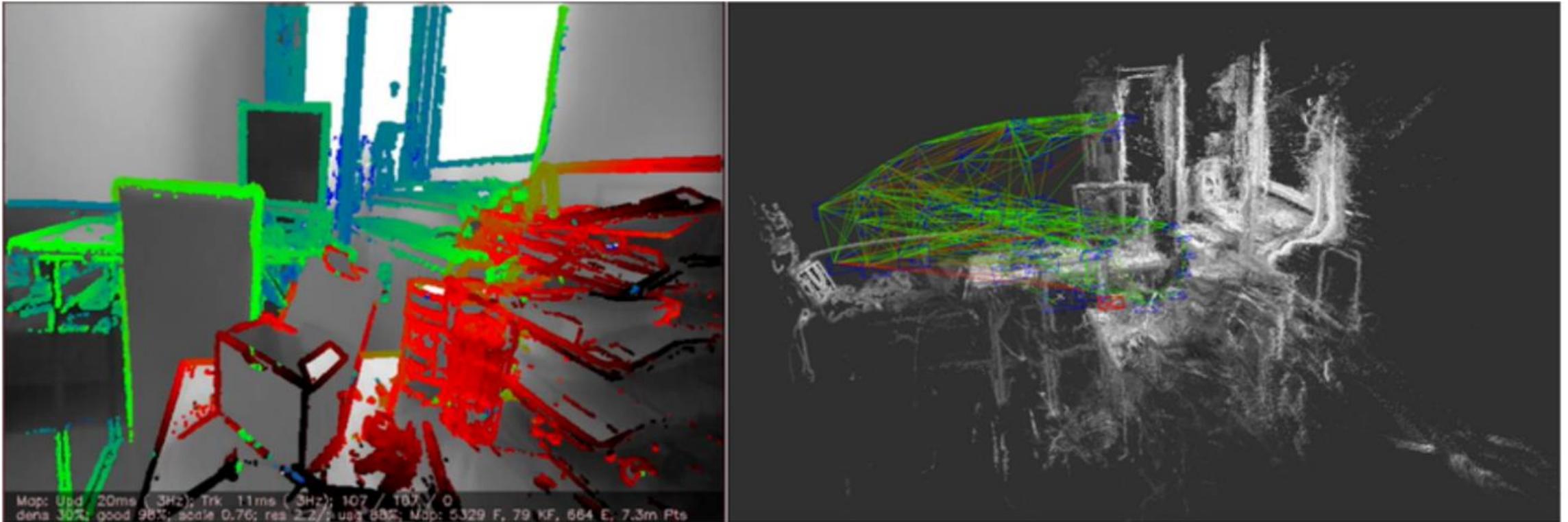
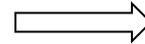


Figura 24. LSD-Slam [18].

RTAB-Map

RTAB-Map (Real-Time Appearance-Based Mapping)



camera stereo, RGB-D e o Lidar.

➤ Funzionamento

- Algoritmi: GFTT o FAST e BRIEF.
- Corrispondenze : nearest neighbor + outlier rejection.
- Stima di posa : PnP + RANSAC (match) e, se hai depth/LiDAR, anche ICP (*Iterative Closest Point*).
- Loop Closure : Bundle adjustment.
- Strategie : F2F (Leggera) o F2M (Robusta).

➤ GFTT

Shi-Tomasi

$$T = \int \int_W \begin{bmatrix} x^2 g_x^2 & x^2 g_x g_y & x y g_x^2 & x y g_x g_y & x g_x^2 & x g_x g_y \\ x^2 g_x g_y & x^2 g_y^2 & x y g_x g_y & x y g_y^2 & x g_x g_y & x g_y^2 \\ x y g_x^2 & x y g_x g_y & y^2 g_x^2 & y^2 g_x g_y & y g_x^2 & y g_x g_y \\ x y g_x g_y & x y g_y^2 & y^2 g_x g_y & y^2 g_y^2 & y g_x g_y & y g_y^2 \\ x g_x^2 & x g_x g_y & y g_x^2 & y g_x g_y & g_x^2 & g_x g_y \\ x g_x g_y & x g_y^2 & y g_x g_y & y g_y^2 & g_x g_y & g_y^2 \end{bmatrix} w dx$$

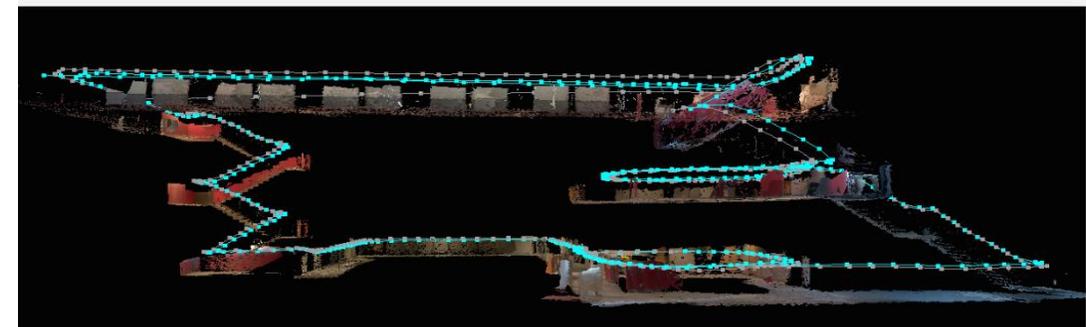
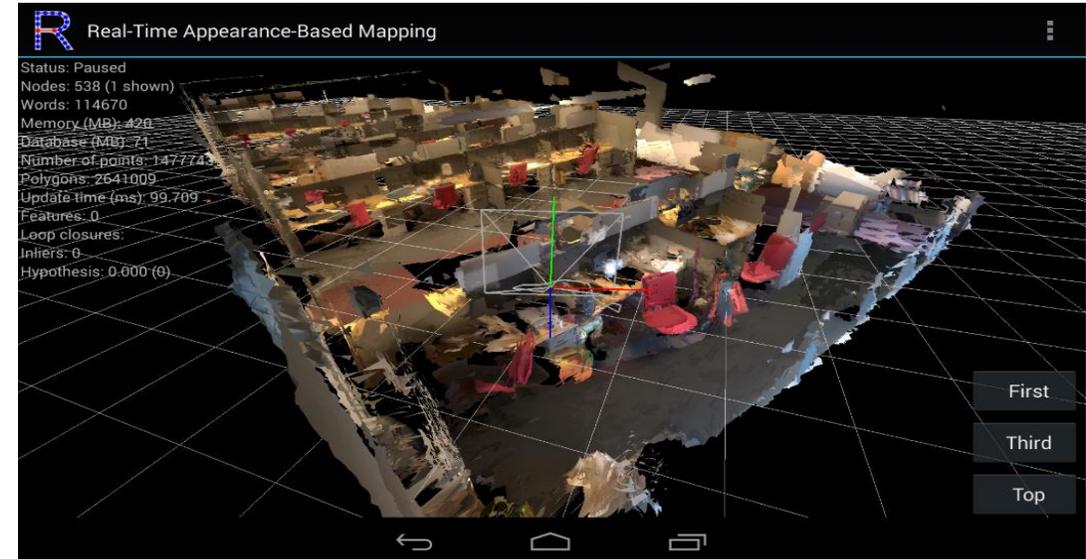


Figura 25. RTAB-Map-Slam [19].

RTAB-MAP $\xrightarrow{\text{OUTPUT}}$ Occupancy grid (2D).

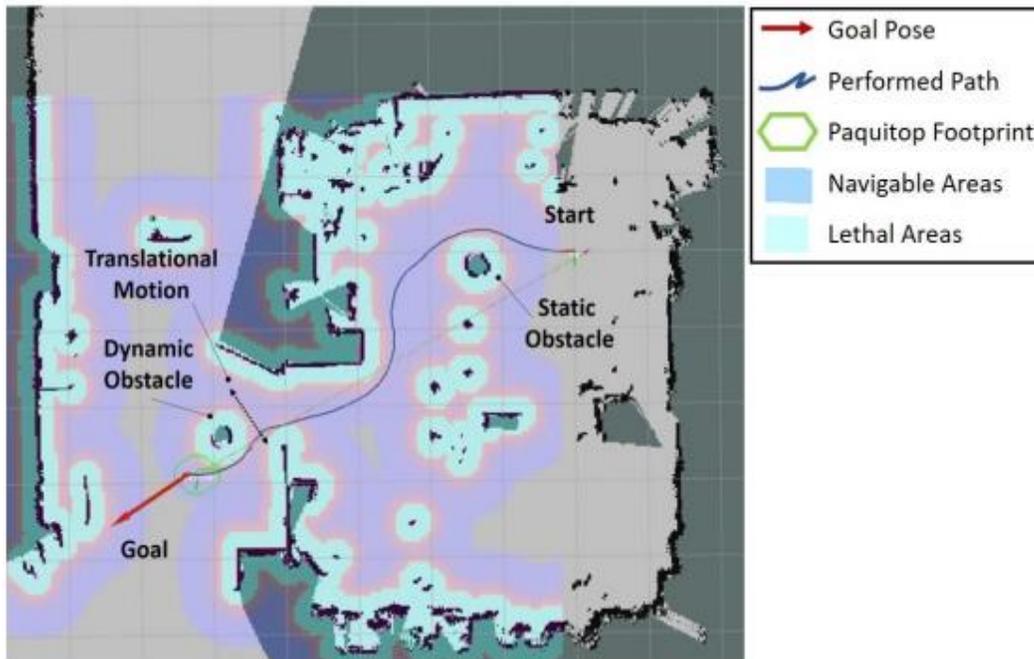


Figura 26. Occupancy grid map [24].

\Rightarrow Algoritmo A* (2D).

\Rightarrow Dynamic Window Approach (DWA) \ Local Planner.

➤ Local Planner (Evitamento Ostacoli)

Cost map, valori 0(cella vuota), 1 (occupata), -1(sconosciuta).

➤ DWA (Brevi distanza)

Basata esclusivamente sulla dinamica del robot

$$G(v, w) = \sigma(\alpha \cdot \text{angle}(v, w) + \beta \cdot \text{dist}(v, w) + \gamma \cdot \text{vel}(v, w))$$

Dove:

- $\text{vel}(v, w)$: definisce la velocità del robot.
- $\text{angle}(v, w)$: rappresenta la direzione attuale del robot rispetto alla direzione dell'obiettivo.
- $\text{dist}(v, w)$: indica la distanza dall'ostacolo più vicino sulla traiettoria.

Si basa su un approccio di ricerca in un grafo, in cui esplora i nodi al fine di trovare il percorso più breve da un nodo iniziale a uno obiettivo.

- Ogni nodo viene valutato tramite una funzione di costo totale:

$$f(n) = g(n) + h(n)$$

- $g(n)$ = costo effettivo dal punto iniziale a n .
- $h(n)$ = stima euristica del costo da n alla destinazione.

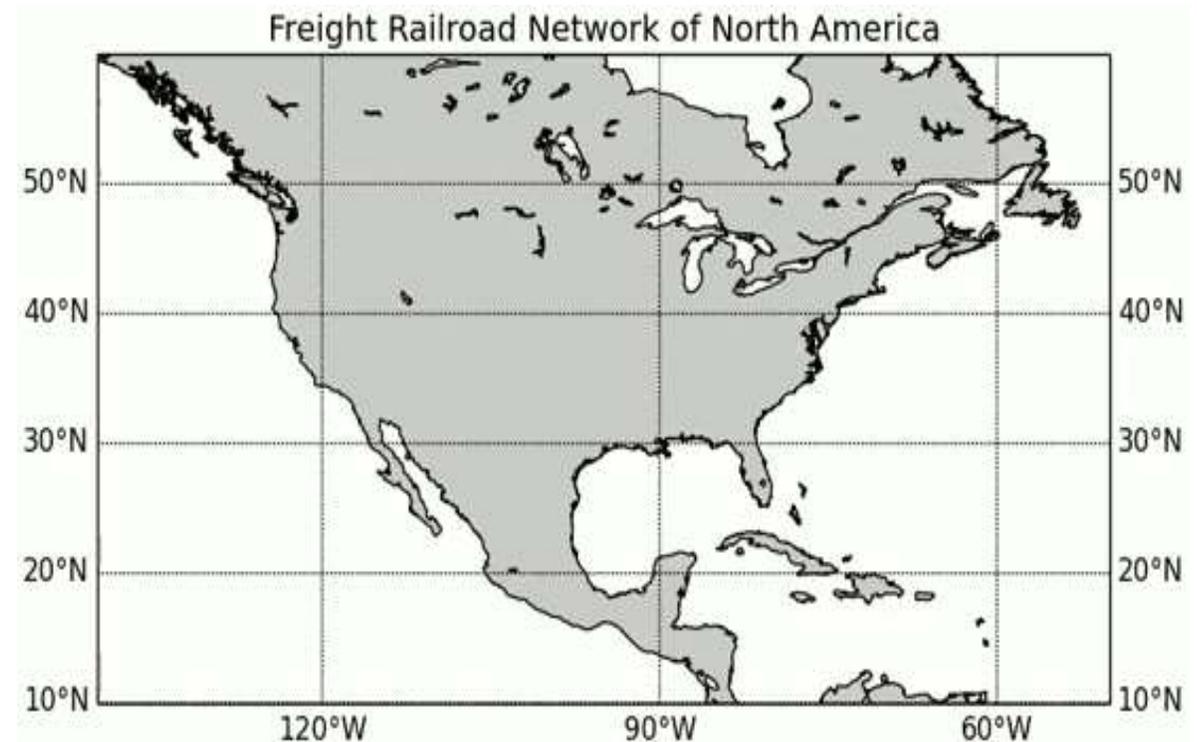


Figura 27. Animazione dell'algoritmo A* che esplora il Nord America cercando un percorso tra Washington D.C. e Los Angeles. [20].

Conclusioni

L'implementazione di un sistema di rilevamento ostacoli e navigazione autonoma su UAV ha dimostrato l'efficacia di tecniche avanzate come VSLAM e algoritmi A* nel migliorare l'autonomia e la sicurezza dei droni.



Figura 28. Drone uav [27].

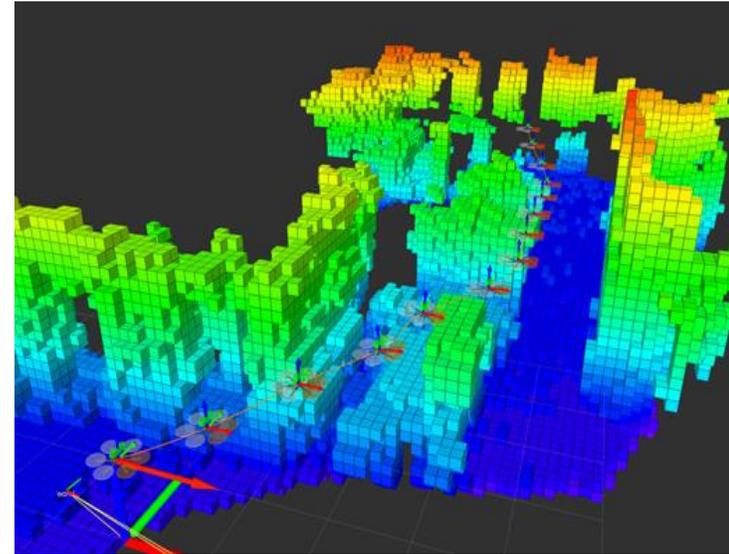


Figura 29. Occupancy grid map [28].

Prospettive Future

I progressi nella tecnologia dei sensori (ad esempio, LIDAR e telecamere avanzate) e l'ottimizzazione degli algoritmi offriranno ulteriori miglioramenti nelle capacità dei droni autonomi. In futuro, si prevede una maggiore integrazione con applicazioni industriali, ambientali e di sicurezza, ampliando il potenziale di utilizzo degli UAV in contesti reali.

Desidero esprimere la mia sincera gratitudine al mio relatore, **Prof. Carlo Bettanini Fecia di Cossato e l'ingegner Sebastiano Chiodini** per la loro guida, il supporto e i preziosi consigli durante lo sviluppo di questa tesi.

1. <https://www.dronezine.it/wp-content/uploads/2014/11/drone-predator-italia.jpg>
2. https://media.licdn.com/dms/image/v2/D5622AQFSGYeFVp728g/feedshare-shrink_2048_1536/B56ZXMtinoGoAo-/0/1742896249917?e=2147483647&v=beta&t=u1H0_-bRvcOMBpV8UuJXtSS9sKtnzE6z8c_nDXaVgal
3. <https://hbmagazineonline.it/wp-content/uploads/2022/10/uav4-1024x705.jpg>
4. https://upload.wikimedia.org/wikipedia/commons/thumb/b/b0/MQ-9_Reaper_in_flight_%282007%29.jpg/500px-MQ-9_Reaper_in_flight_%282007%29.jpg
5. <https://www.osservatorelibero.it/wp-content/uploads/2022/06/Drone-VVFF-Mugello-3-625x344.jpeg>
6. https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEh5eOOkNYpQP9_6GQsliBK2iM4WsXvFfmMR7bVkiNLR4IHZOekkopc2kJJwfa-XlzuMRPXVY05PyeC2DN-74LWZUZLJ9Z4NlcZTf5lafAb1UB954k6DsGiMj03V1wjPEyOe0qh_vdxGHD9go9RCCEFqjewNQYMP_xIupAI_HzmeibPVAm4poSiwuXH6VI/s16000/DJI%20Flycart%2030_2_3.jpg
7. <https://flytodiscover.it/wp-content/uploads/2022/03/dji-zenmuse-l1-dji.jpg>
8. <https://cdn.shopify.com/s/files/1/0699/6927/products/Zed2ifacedawn.png?v=1742567417&width=856&crop=center>
9. <https://www.nvidia.com/it-it/autonomous-machines/embedded-systems/jetson-orin/nano-super-developer-kit/>
10. https://www.generationrobots.com/20326-product_cover/telemetre-laser-360-ydlidar-x4pro.jpg
11. https://www.industrial-production.de/upload_weka/nwo/003/498/9723833_big_3498416.jpg
12. <https://assets.raspberrypi.com/static/8f6eca535dbb23a21eb41a748050e3a0/33b96/16gb.webp>
13. https://ardupilot.org/copter/_images/Pixhawk-Infographic2.jpg
14. <https://levelfivesupplies.com/wp-content/uploads/2022/10/kudan-lidar-slam-1024x532.jpg>
15. https://it.mathworks.com/discovery/slam/_jcr_content/mainParsys/band_1231704498_copy/mainParsys/lockedsubnav/mainParsys/columns_39110516/6046ff86-c275-45cd-87bc-214e8abacb7c/columns_463008322/7b029c5b-9826-4f96-b230-9a6ec96cb4ab/columns_1472205090/bd4f8030-3567-47ae-b28f-3d4ac2e33131/image_copy_222030960.adapt.full.medium.jpg/1753110538263.jpg
16. https://it.mathworks.com/discovery/slam/_jcr_content/mainParsys/band_1231704498_copy/mainParsys/lockedsubnav/mainParsys/columns_39110516/6046ff86-c275-45cd-87bc-214e8abacb7c/columns_463008322/7b029c5b-9826-4f96-b230-9a6ec96cb4ab/columns_1472205090/bd4f8030-3567-47ae-b28f-3d4ac2e33131/columns_copy_173866084/0b779ec8-011c-4f78-9b15-b200d6b0bffa/image_copy_188209747.adapt.full.medium.jpg/1753110537644.jpg
17. https://eric-yyjau.github.io/misc/talks/20200212_progress_report/20200212_progress_report.pdf
18. <https://www.researchgate.net/publication/369380319/figure/fig8/AS:11431281128419921@1679370342575/Result-for-LSD-SLAM-on-room-sequence-from-TUM-data-is-shown-Image-on-the-right-shows-the.ppm>
19. https://raw.githubusercontent.com/wiki/introlab/rtabmap/doc/video_screenshots/rtabmap-tango.png
20. https://upload.wikimedia.org/wikipedia/commons/6/60/A%2A_Search_Example_on_North_American_Freight_Train_Network.gif
21. <https://www.codespeedy.com/orb-feature-detection-in-python-opencv/>
22. <https://webthesis.biblio.polito.it/9516/1/tesi.pdf>
23. <https://www.ftoremotefly.com/portfolio/quando-droni-fanno-sul-serio-emergenze>
24. <https://webthesis.biblio.polito.it/secure/24667/1/tesi.pdf>
25. <https://webthesis.biblio.polito.it/9516/1/tesi.pdf>
26. <https://www.bbc.com/future/article/20230609-can-we-use-firefighting-drones-put-out-wildfires>
27. <https://dronexl.co/de/2025/06/24/vslam-drone-gps-denied-navigation/>
28. <https://www.suasnews.com/2015/10/a-drone-with-a-sense-of-direction/>