Politecnico di Milano
Dipartimento di Elettronica, Informazione e Bioingegneria
Corso di Laurea Magistrale in Ingegneria Informatica

Cognitive SLAM: knowledge-based self-localization and mapping

AI & R Lab
Laboratorio di Intelligenza Artificiale
e Robotica del Politecnico di Milano

Relatore:
Prof. Andrea Bonarini

Tesi di Laurea di:
Davide Tateo Matricola n. 799311

Anno Accademico 2013-2014

A qualcuno...

Sommario

Lo SLAM è uno dei principali problemi nello sviluppo di robot autonomi. Gli approcci correnti sono afflitti da un pesante costo computazionale e si comportano male in abienti dinamici e disordinati; le performance di questi sistemi diventano ancora peggiori se si utilizzano sensori a basso costo, necessari per le applicazioni commerciali. Questa tesi affronta il problema da un punto di vista nuovo e originale, usando feature ad alto livello come punti chiave e sfruttando la conoscenza di un esperto e un linguaggio fuzzy per riconoscerli, per tenere traccia di punti chiave forti e stabili e permettere mappe più intelligenti e una localizzazione robusta in ambienti complessi. L'idea fondamentale è quella di mantenere il tasso di errore della localizzazione limitato e ridurre il costo necessario a far navigare con successo un robot autonomo in un ambiente interno, usando solo i dati provenienti da una webcam e una unità di misura inerziale a basso costo. Il principale problema è il riconoscimento di feature ad alto livello, come porte e scaffali, affrontato tramite un classificatore fuzzy ad albero, definito da un esperto, in modo da evitare fasi di allenamento e migliorare la generalità del riconoscimento.

Abstract

SLAM is one of the key issues in autonomous robots development. Current approaches are affected by heavy computational load and misbehave in cluttered and dynamic environment; their performance get even worse with low cost sensors, needed for market applications. This thesis faces the problem in a new and original way, working with high level features as key points and using expert knowledge and a fuzzy language to detect them, in order to track strong and stable key points and allow smarter maps and robust localization in complex environments. The key idea is to keep the error rate of the localization process limited, and to reduce the cost of an autonomous robot to successfully navigate into an indoor environment using only the data from a webcam and a low cost Inertial measurement unit. The main issue is the high level feature recognition, like doors and shelves, done by an expert-defined fuzzy tree classifier, in order to avoid training and improve generalization of recognition.

Indice

Ringraziamenti

Ringrazio .....

Capitolo 1

Introduzione

"Qualunque cosa che accade, accade"
"Qualunque cosa che, accadendo, ne fa accadere un'altra, ne fa accadere un'altra."
"Qualunque cosa che, accadendo, induce se stessa a riaccadere, riaccade."
"Però non è detto che lo faccia in ordine cronologico."

Douglas Adams, Mostly Harmless

Uno dei problemi chiave della robotica è la localizzazione di un robot in un ambiente sconosciuto. Questo problema è noto come "SLAM", Simultaneus localization and Mapping, ed è attualmente una delle aree più importanti della ricerca riguardante i robot autonomi. In particolare, questo problema risulta ancora più difficile se applicato a robot dotati di sensori a basso costo, quali webcam e unità di misura inerziale economiche, restrizione fondamentale per la diffusione di applicazioni di robotica autonoma nel mondo reale.
Attualmente questo problema è affrontato attraverso tecniche probabilistiche, che sono ottime per analizzare ambienti statici e per ottenere una localizzazione precisa. Le mappe create sono utilizzabili principalmente per la navigazione. I sensori più usati sono gli scanner laser, che tuttavia sono molto costosi, i sensori RGB-D, sensori economici che nascono per applicazioni videoludiche commerciali e hanno alcune limitazioni se applicati alla robotica, le telecamere stereo.
Lo scopo della tesi è di sviluppare un framework per risolvere il problema della localizzazione di robot autonomi dotati di sensori a basso costo, quali ad esempio i quadricotteri. L'idea è di sviluppare una metodologia che non solo permetta al robot di localizzarsi nella mappa in maniera efficiente, ma anche di interagire con l'ambiente in maniera avanzata, di compiere ragionamenti, di adattarsi a eventuali cambiamenti ed agli eventi che possono

occorrere, quali la presenza di persone o altri agenti.
Basandosi principalmente sulle informazioni provenienti da una videocamera monoculare, è stato sviluppato un sistema che è in grado di processare informazioni provenienti da un esperto, espresse in un linguaggio formale, di estrarre feature di basso livello e aggregarle, grazie alla base di conoscenza, per riconoscere e tracciare feature di alto livello, quali porte e armadi, e di basare conseguentemente su di essi il processo di localizzazione. Per permettere all'esperto di trasmettere la sua conoscenza all'agente, è stato sviluppato un linguaggio formale, basato sulla logica fuzzy, che permetta di esprimere sia regole fuzzy linguistiche, sia di definire un classificatore fuzzy ad albero in grado di definire una gerarchia di modelli e le relazioni tra di essi.

La tesi è strutturata nel modo seguente:
Nel capitolo 2 si illustra lo stato dell'arte.
Nel capitolo 3 si illustrano le basi teoriche necessarie.
Nel capitolo 4 si descrive l'architettura generale del sistema.
Nel capitolo 5 si parla del reasoner e del linguaggio formale utilizzato.
Nel capitolo 6 si descrive l'estrazione delle feature di basso livello.
Nel capitolo 7 si mostra come gli oggetti riconosciuti sono tracciati.
Nel capitolo 8 si illustra la creazione della mappa e la localizzazione.
Nel capitolo 9 si analizzano i risultati sperimentali del sistema proposto.
Nel capitolo 10 si riassumono gli scopi, le valutazioni di questi e le prospettive future.

Capitolo 2

Stato dell'arte

Doc: Ecco perché non ha funzionato: c'è scritto "Made in Japan".
Marty: E che vuol dire Doc? Tutta la roba migliore è fatta in Giappone.
Doc: Incredibile!

Ritorno al Futuro, parte III

2.1 SLAM

Il problema della localizzazione di un robot in un ambiente sconosciuto è stato affrontato fin dagli anni 90' a partire da [15], articolo nel quale per la prima volta si delineava un framework per localizzare un robot costruendo contemporaneamente la mappa dell'ambiente. Tramite l'utilizzo di sonar, venivano estratte feature geometriche con cui veniva costruita la mappa, nella quale il robot si localizzava. Il problema principale della localizzazione è il "problema della correlazione": se la posizione della feature rispetto alla quale ci si localizza è affetta da incertezza, la conseguente stima della posizione effettuata rispetto a tale feature sarà affetta da un errore che dipende dall'errore della posizione della feature stessa. Questo problema diventa tanto più grave se si pensa che la posizione del robot in ogni istante non è nota a priori, ma deve essere stimata sulla base delle osservazioni precedenti. Come è facile vedere, è necessario risolvere questo problema per evitare che l'errore della generazione della mappa e l'errore della stima della posizione divergano nel tempo. Per risolverlo, gli autori hanno utilizzato un filtro di Kalman esteso.

Come è noto, il filtro di Kalman è uno stimatore Bayesiano ricorsivo, che, supposto noto il modello lineare che regola la generazione dei dati e la

loro osservazione, supposto che l'errore di misura e di modello siano gaussiani, restituisce la densità di probabilità del sistema osservato. Il filtro di Kalman, se utilizzato secondo le ipotesi, è uno stimatore ottimo dello stato del sistema osservato, secondo i minimi quadrati. Tuttavia, nell'ambito della robotica, e in particolare nel problema della localizzazione, il modello di generazione e osservazione dei dati non può essere considerato lineare. E' quindi necessario utilizzare un'estensione del filtro di Kalman al caso non lineare: il filtro di Kalman esteso (EKF) è una delle possibili soluzioni al problema. L'idea alla base del filtro di Kalman esteso è quella di lavorare sul modello linearizzato, stimato ricorsivamente dal modello non lineare sulla base della stima corrente.

Per avere una buona stima della posizione è necessario utilizzare un gran numero di feature, numero che cresce molto rapidamente con l'aumentare della dimensione dell'ambiente. La complessità computazionale dell'approccio tradizionale basato sul filtro di Kalman esteso è $\mathcal{O}(N^3)$, con N numero di feature, e quindi il tempo di calcolo diventa ben presto inaccettabile. Per risolvere questo problema è stato introdotto in [18] un nuovo algoritmo detto FastSLAM, che consiste nell'utilizzo del Particle Filter, e del filtro di Kalman esteso in combinazione. L'algoritmo associa ad ogni feature considerata, un filtro di Kalman esteso; la densità di probabilità congiunta, invece, viene calcolata sfruttando il Particle filter. Il Particle Filter è un altro stimatore Bayesiano ricorsivo, che, invece di un modello e dell'assunzione di rumore gaussiano, sfrutta metodi di tipo Monte Carlo per stimare la densità di probabilità del sistema che genera i dati. Il risultato è un algoritmo che ha complessità computazionale $\mathcal{O}(N \log M)$, con M numero di feature e N è il numero di particelle usate dal Particle Filter. Questo approccio rende il problema trattabile nella maggior parte dei casi, pur essendo pesante computazionalmente, essendo necessario un elevato numero di particelle per avere una buona localizzazione.

Vista la particolarità del problema quando il sensore utilizzato è una videocamera monoculare, sono stati sviluppati algoritmi ad hoc. Uno degli algoritmi più usati è PTAM, descritto in [11] e [12]. L'idea alla base di questo algoritmo è dividere in due thread separati il tracking e la creazione della mappa: un thread si occupa del tracking robusto di feature a basso livello, mentre l'altro thread si occupa della creazione della mappa. Per rendere efficiente il processo di mapping, solo i keyframe, ossia i frame che contengono maggiore informazione rispetto a quella già presente, vengono considerati. Per rendere il processo di mapping robusto, vengono utilizzate tecniche batch per costruire la mappa, come ad esempio il bundle adjustment. PTAM, tuttavia, nasce per applicazioni di realtà aumentata, e quindi

necessita di una inizializzazione, per risolvere i problemi dell'acquisizione del primo keyframe e per gestire la scala della mappa.

Un altro approccio alternativo è quello di usare tutti i dati dell'immagine per eseguire la localizzazione, questo approccio è alla base ad esempio di DTAM, Dense Tracking and Mapping, descritto in [19].

Si sono dimostrati efficaci anche i metodi semi-diretti, come SVO, Semi Direct Visual Odometry, un algoritmo che riesce a ottenere altissime prestazioni limitando l'estrazione delle feature ad alto livello ai soli keyframe, operando direttamente sulle intensità dei pixel nei frame successivi, eliminando le fasi computazionalmente più onerose, che sono l'estrazione e l'abbinamento delle feature. SVO si basa sulle idee di PTAM, ma migliora sia le prestazioni, riuscendo a essere computazionalmente più leggero, sia la precisione della mappa e della localizzazione riducendo di molto gli outlier.

Recentemente stanno avendo molto successo i sistemi basati su sensori RGB-D [9] [5]. Questi sensori vengono utilizzati come scanner laser a basso costo per creare una mappa dell'ambiente. Tuttavia, sono soggetti a molte limitazioni, non essendo stati progettati per questo scopo, e soffrono tra l'altro di un raggio d'azione limitato. Nonostante queste limitazioni i sistemi riescono comunque a ottenere buone prestazioni in ambienti indoor [23].

2.2 Rappresentazione del Mondo

Sono noti diversi modi per rappresentare un ambiente tridimensionale. Il più semplice possibile è quello di usare delle nuvole di punti, direttamente estratte dai sensori. Un'altra rappresentazione comune è quella di filtrare le nuvole di punti ottenute tramite una griglia di voxel, come ad esempio in [22]. Metodi più avanzati permettono una rappresentazione geometrica dell'ambiente con un minor uso di memoria, come ad esempio le mappe di quota, in cui una mappa a due dimensioni è estesa con il calcolo del valore medio di altezza di ciascun punto 2D come in [6], le mappe di quota estese, che tengono conto di possibili aperture attraversabili dai robot oppure le mappe di quota multi livello, che riescono a descrivere complesse geometrie multi livello [24].

Tra le mappe più promettenti esistono le mappe basate sugli octree, che permettono una efficiente rappresentazione in memoria sia dello spazio occupato che dello spazio libero, pur potendo descrivere geometrie complesse. Inoltre questo tipo di rappresentazione permette di scalare facilmente la risoluzione della mappa, permettendo di utilizzare la stessa mappa per compiti che richiedono un sensibilità differente. Una efficiente implementazione di questo tipo di mappa può essere trovata in [8].

Recentemente si sta sviluppando l'idea di rappresentare il mondo ad alto livello, incorporando informazioni semantiche e geometriche che possano essere utilizzate non solo dagli algoritmi di navigazione, ma anche per svolgere compiti ad alto livello e ragionamenti. Una possibile soluzione al problema è l'approccio basato su Scene Graph, come descritto in [2]. Sono stati sviluppati anche DSL per poter interagire ad alto livello, ad esempio generando istanze di oggetti a partire da un modello generico, come ad esempio in [3].

2.3 Riconoscimento di Oggetti

La quasi totalità degli algoritmi di riconoscimento di oggetti nell'immagine è basata su tecniche di machine learning. Una delle classi di algoritmi più usati sono quelli basati sulle Haar-like feature, descritte in [21]. Queste feature sono calcolate in aree rettangolari, all'interno delle quali vengono sommati i valori dell'intensità dei pixel. Le somme sono in seguito usate per calcolare differenze tra aree di interesse, per poter riconoscere feature geometriche quali angoli, linee o bordi. Un efficiente algoritmo per il riconoscimento di oggetti è descritto in [25] ed è stato ampliato in [16], e consiste nell'utilizzare in cascata una serie di classificatori via via più restrittivi; ognuno dei classificatori di ogni stadio è costruito attraverso una tecnica di boosting, ossia sono formati da un insieme di classificatori deboli che creano un classificatore complessivo più restrittivo. I classificatori di base sono solitamente alberi di decisione. Il classificatore complessivo da una risposta binaria, per riconoscere effettivamente l'oggetto deve essere applicato tramite una finestra mobile su tutta l'immagine.
Un algoritmo simile può essere trovato in [1], dove però invece che feature Haar-like, vengono utilizzati direttamente un insieme di pixel nell'immagine.

Un'altra importante classe di algoritmi è basato sulle feature HOG (Histogram of Oriented Gradients). Queste feature sono ricavate contando le occorrenze dell'orientamento dei gradienti in sotto blocchi dell'immagine. Su queste feature si basa l'algoritmo descritto in [4]. Questo algoritmo è in grado di definire un oggetto a partire dalle sue parti, e quindi è robusto anche a occlusioni parziali dell'oggetto. Tuttavia è computazionalmente molto più pesante dell'approccio basato su feature Haar-like.

Più recenti sono i metodi basati sul deep learning. Tra questi, i più promettenti nel riconoscimento di oggetti sono basati sulle reti neurali convoluzionali. La particolarità di queste reti è di essere basata su kernel di nodi collegati in maniera fissa, e ripetuti per coprire tutta l'immagine. La struttura rigida sfrutta la località dell'informazione nell'immagine, e per-

mette un training più efficiente. Un esempio di applicazione si può trovare in [13].

Metodi recenti, sfruttano sensori RGB-D, come ad esempio Kinect, per riconoscere oggetti nella scena, come ad esempio in [14].

2.4 Reasoning

I sistemi esperti sono stati per lungo tempo una delle branche dell'intelligenza artificiale più attive. In particolare si può citare il noto strumento per la creazione di sistemi esperti, CLIPS, e la sua estensione alla logica fuzzy, FuzzyCLIPS. Entrambi permettono di descrivere delle classi di oggetti e di definire predicati tra di essi, in modo da poter compiere inferenza sui dati in ingresso. Una risorsa sui sistemi esperti in generale, e CLIPS in particolare può essere trovata in [10]. La ricerca sui sistemi di deduzione formale tuttavia si è spostata da i sistemi esperti al web semantico. L'attuale standard per i sistemi di inferenza è il linguaggio OWL 2, descritto in [20] e [7], linguaggio fortemente basato sulle logiche descrittive che permettono un buon compromesso tra espressività, complessità computazionale e mantengono la logica utilizzata decidibile. Un esempio di reasoning applicato al riconoscimento delle immagini può essere trovato in [17], in cui feature a basso livello vengono estratte da un meccanismo di machine learning, e utilizzate da un sistema di inferenza che sfrutta una ontologia per l'analisi ad alto livello dell'immagine.

Capitolo 3

Concetti Fondamentali

Capitolo 4

Architettura del Sistema

Capitolo 5

Reasoning

Marty: Aspetta un momento Doc. Se vado dritto verso lo schermo, andrò a sbattere contro quegli indiani!
Doc: Marty, non stai pensando quadridimensionalmente!

Ritorno al Futuro, parte III

Capitolo 6

Riconoscimento degli Oggetti

Capitolo 7

Tracking

Capitolo 8

Mapping

Capitolo 9

Risultati sperimentali

Capitolo 10

Conclusioni e Sviluppi Futuri

Bibliografia

[1] Yotam Abramson, Bruno Steux, and Hicham Ghorayeb. Yet even faster (yef) real-time object detection. *IJISTA*, 2(2/3):102–112, 2007.

[2] S. Blumenthal, H. Bruyninckx, W. Nowak, and E. Prassler. A scene graph based shared 3d world model for robotic applications. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 453–460, May 2013.

[3] Sebastian Blumenthal and Herman Bruyninckx. Towards a domain specific language for a scene graph based robotic world model. *arXiv preprint arXiv:1408.0200*, 2014.

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[5] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *The International Journal of Robotics Research*, 31(5):647–663, 2012.

[6] M Herbert, C Caillas, Eric Krotkov, In So Kweon, and Takeo Kanade. Terrain mapping for a roving planetary explorer. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 997–1002. IEEE, 1989.

[7] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph, editors. *OWL 2 Web Ontology Language: Primer*. W3C Recommendation, 27 October 2009. Available at `http://www.w3.org/TR/owl2-primer/`.

[8] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D map-

ping framework based on octrees. *Autonomous Robots*, 2013. Software available at `http://octomap.github.com`.

[9] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[10] Gary D. Riley Joseph C. Giarratano. *Expert Systems Principles and Programming*. 1994.

[11] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

[12] Georg Klein and David Murray. Improving the agility of keyframe-based SLAM. In *Proc. 10th European Conference on Computer Vision (ECCV'08)*, pages 802–815, Marseille, October 2008.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[14] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Detection-based object labeling in 3d scenes. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1330–1337. IEEE, 2012.

[15] J.J. Leonard and H.F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on*, pages 1442–1447 vol.3, Nov 1991.

[16] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In *IEEE ICIP 2002*, pages 900–903, 2002.

[17] Nicolas Maillot and Monique Thonnat. Ontology based complex object recognition. *Image Vision Comput.*, 26(1):102–113, 2008.

[18] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.

[19] Richard A. Newcombe, Steven Lovegrove, and Andrew J. Davison. Dtam: Dense tracking and mapping in real-time. In Dimitris N. Metaxas, Long Quan, Alberto Sanfeliu, and Luc J. Van Gool, editors, *ICCV*, pages 2320–2327. IEEE, 2011.

[20] W3C OWL Working Group. *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation, 27 October 2009. Available at `http://www.w3.org/TR/owl2-overview/`.

[21] C.P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Computer Vision, 1998. Sixth International Conference on*, pages 555–562, Jan 1998.

[22] Y. Roth-Tabak and R. Jain. Building an environment model using depth information. *Computer*, 22(6):85–90, June 1989.

[23] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 573–580. IEEE, 2012.

[24] R. Triebel, P. Pfaff, and W. Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2276–2282, Oct 2006.

[25] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. pages 511–518, 2001.

Appendice A

Il manuale utente

Manuale utente per l'utilizzo del sistema

Appendice B

Esempio di impiego

Un esempio di impiego del sistema realizzato.