Politecnico di Milano
Scuola di Ingegneria Industriale e dell'Informazione
Corso di Laurea Magistrale in Ingegneria Informatica

Cognitive SLAM: Knowledge-Based Simultaneous Localization and Mapping

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

Relatore:
Prof. Andrea Bonarini

Tesi di Laurea di:
Ärdil Su ErdenliÄ§ Matricola n. 852772

Anno Accademico 2017-2018

Indice

Elenco delle figure

Capitolo 1

State of the Art

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

1.1 Hierarchical Policy Search Algorithms

$\mathcal{C}, \mathcal{M}$

The high-dimensional, continuous state and action spaces is one of the main challenges in robot learning. Value function based methods require filling the complete action-state space with data 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.

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 spesso utilizzato un filtro di Kalman esteso.

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 prestazioni in tempo reale. Per risolvere questo problema è stato introdotto in [**?**] 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 il numero di particelle usate dal Particle Filter. Questo approccio rende il problema trattabile nella maggior parte dei casi, pur essendo pesante computazionalmente, dato che è 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 [**?**], [**?**]. 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. Il bundle adjustment consiste in un processo iterativo di raffinamento della stima dei punti 3D ricostruiti e della posa della videocamera. 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 approccio alternativo consiste nell'usare tutti i dati dell'immagine per eseguire la localizzazione, questo approccio è alla base, ad esempio, di DTAM, Dense Tracking and Mapping [**?**]. Questo algoritmo crea un modello denso dell'ambiente e usa l'allineamento della videocamera

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 ne 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 [**?**], [**?**]. 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 [**?**].

1.2 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 [**?**]. 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 [**?**], 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.

Tra le mappe più promettenti esistono le mappe basate sugli octree, che permettono un'efficiente rappresentazione in memoria sia dello spazio occupato sia 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 una precisione differente. Un'efficiente implementazione di questo tipo di mappa può essere trovata in [**?**].

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 [**?**]. Sono stati sviluppati anche linguaggi specifici di dominio (DSL) per poter interagire ad alto livello, ad esempio generando istanze di oggetti a partire da un modello generico, come ad esempio in [**?**].

1.3 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 [**?**]. 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 [**?**] ed è stato ampliato in [**?**], 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 dà una risposta binaria; per riconoscere effettivamente l'oggetto deve essere applicato tramite una finestra mobile su tutta l'immagine.
Un algoritmo simile è descritto in [**?**], dove, però, invece che feature Haar-like, vengono utilizzati direttamente un insieme di pixel selezionati nell'immagine.

Un'altra importante classe di algoritmi è basata 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 [**?**]. 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 permette un training più efficiente. Un esempio di applicazione si può trovare in [**?**].

Metodi recenti, sfruttano sensori RGB-D, come ad esempio Kinect, per riconoscere oggetti nella scena, come ad esempio in [**?**]. La maggior parte delle tecniche usate sono un'estensione delle tecniche già descritte, spesso estendendo i descrittori delle feature grazie alle informazioni sulla profondità.

1.4 Reasoning

I sistemi di ragionamento sono stati alla base dell'Intelligenza Artificiale fin dalla nascita ed hanno trovato sbocco in diverse applicazioni tra cui i sistemi esperti, per lungo tempo una delle branche dell'intelligenza artificiale più attive [**?**] [**?**] [**?**].

Con la maturazione della tecnologia per lo sviluppo dei sistemi esperti, e l'avvento di Internet, l'attenzione della ricerca collegata ai sistemi di ragionamento formale si è spostata dai sistemi esperti al web semantico. Il linguaggio più diffuso per i sistemi di inferenza è il linguaggio OWL 2 [**?**], [**?**], fortemente basato sulle logiche descrittive che permettono un buon compromesso tra espressività, e complessità computazionale e mantengono la logica utilizzata decidibile.

Fin dai primi sistemi esperti si è riconosciuta l'importanza del trattamento dell'incertezza in sistemi di ragionamento che avessero a che fare con il mondo reale [**?**], ed in particolare si è avuta una larga diffusione in moltissimi settori della logica fuzzy [**?**], adottata anche nel nostro lavoro.

Un esempio di reasoning applicato al riconoscimento delle immagini può essere trovato in [**?**], 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. Un altro esempio di utilizzo di sistemi di inferenza nel riconoscimento di oggetti si può trovare in [**?**], dove viene utilizzato il linguaggio OWL per specificare una base di conoscenza e riconoscere tramite la conoscenza di dominio gli oggetti nell'ambiente.

Bibliografia