**Homework 1– Manipulation**

Obiettivo

L'obiettivo del Homework è quello di sviluppare un algoritmo ROS in grado di localizzare un insieme di oggetti posti su un’area di lavoro, utilizzando un kinect RGBD posta ad una certa distanza al di sopra dell’area di lavoro.

Utilizzando un manipolatore UR5, che sarà implementato tramite Movelt, che navigherà nello spazio al fine di raggiungere la posizione degli oggetti e raccoglierli con un magnete (end-effector) e depositarli in un'area di destinazione, evitando collisioni con gli altri oggetti posti sul tavolo.

Posizioni Tag

Gli oggetti target che il manipolatore dovrà raccogliere saranno specificati in un nodo presente nell’Homework 1 che continua a pubblicare le Pose rilevate su un topic chiamato hw\_target\_objecs,il quale viene aggiornato costantemente ( void Task::\_updateTargets(const hw1::poseArray::ConstPtr &msg) ) per eventuali rilevazioni errate, se il manipolatore durante gli spostamenti possa trovarsi al di sopra dell’oggetto oppure se l’oggetto viene inserito successivamente al rilevamento iniziale, infine le pose possano essere utilizzate dal manipolatore che si appresterà a muoversi.

Al fine di evitare collisioni tra manipolatore ed oggetti abbiamo inizializzato la Collision Detection ( \_collisionObjects) inserendo le meshes utilizzate nel Homework 1, che ci permetterà tramite i rilevamenti della kinect di posizionare le mesh, ed avere una maggiore accuratezza nel collision detection.

Posizione Manipulatore

Il manipulatore che andremo ad utilizzare è UR5, che implementeremo con la libreria Movelt, che ci permetterà di spostare il manipolatore nel piano di lavoro.

Nella programmazione del manipolatore abbiamo scelto di spostare il robot in due modi: Tramite dei spostamenti con i giunti oppure in cartesiano.

Questa combianazione di movimenti ci permette di rendere il robot piu stabile ed evita improvvisi movimenti che potrebbero portare a collisioni inaspettate.

Questo tipo di manipolatore presenta un numero di 5 giunti, inizialmente impostiamo una posizione al di sopra dellarea di lavoro, con un valore per ogni giunto(\_referencePosition).

1 Stima della posa con i marcatori Apriltag

Per il primo compito, un approccio simile a quello utilizzato nel laboratorio st, nel compito

1, è stato utilizzato.

Per nd la posizione degli oggetti target è stato utilizzato il nodo apriltag, che

pubblica, sul tema /rilevamenti dei tag, un elenco di tutti i marcatori apriltag rilevati".

posizioni e orientamenti.

Conoscendo il codice del marker, siamo in grado di abbinare i diversi tipi di oggetti su

la tabella con l'id del frame corrispondente, e abbinarla ad una lista di oggetti target

speciale tramite riga di comando.

La posa estratta viene poi riferita al quadro di riferimento mondiale utilizzando l'opzione

tf buffer.lookup transform(), che sarà usato per calcolare il tar-

prendi il manipolatore.

2 Negazione della posizione del bersaglio

Il nodo apriltag emette un elenco delle pose dierenti del tag, ma, per una buona comprensione,

la pinza deve chiudersi il più vicino possibile al baricentro dell'oggetto, che fa

non corrispondono alla posizione dell'oggetto trasformato. Inoltre, il successo della presa

il funzionamento dipende dal corretto orientamento del settore finale, che può essere dedotto

dall'orientamento del marker di aprile.

In particolare, per evitare problemi con i dierenti punti finali di rotazione del ma-

nipulatore fine etere, la posizione dell'oggetto è impostata meno la periodicità degli oggetti

se stessa. Cioè

oggetto

:=

a:tag

mod

2

n

+

Con caratteristica oset dovuto all'orientamento apriltag sulla faccia superiore e 2

n

periodicità nell'asse di imbardata della forma considerata.

Dato il requisito del manipolatore di manipolare oggetti di forme dierenti,

un set caratteristico, sia nella posizione che nell'orientamento, deve essere applicato ad ogni

uno basato sul tipo di oggetto. Come per il primo incarico di laboratorio, il dierente

osets possono essere conguidati dinamicamente dall'utente per ottenere il miglior punto possibile per un

afferrare con successo, attraverso la riconfigurazione rqt.

3

3 Negazione degli ostacoli alle collisioni

Dopo il calcolo del baricentro per tutti gli oggetti rilevati sul tavolo, il

vengono generati i corrispondenti ostacoli di collisione, per evitare di colpire oggetti secondari in

il percorso verso l'obiettivo. Questi sono generati come segue:

Per il cubo, una versione in scala del 120% del cubo viene posizionata nella stessa posizione

come il baricentro del cubo

Per il prisma esagonale, un cilindro con raggio 110% del lato dell'esagono

si usa la faccia superiore, al centro del baricentro dell'oggetto

Per il prisma tringolare, un cubo ruotato di 45 lungo l'asse del midollo, con una scala del 120% è

utilizzato, centrato al centro della faccia inferiore del prisma (il baricentro di

il solido "specchiato")

4 Calcolo del piano

Quando la procedura rst inizia, viene chiamato il metodo init arm. Questo nega il

quadro di riferimento per il manipolatore, imposta le tolleranze di posizione e di orientamento per un

condizione di "obiettivo raggiunto" e mette il manipolatore in una posizione di partenza standard.

Questa posizione è posizionata in modo tale che il braccio non copra porzioni di

il tavolo, dove la telecamera kinect non sarebbe in grado di individuare i marcatori apriltag.

4

Quando un oggetto che corrisponde ad uno dei bersagli specificati da riga di comando

si trova, dopo aver controllato la validità della posizione per garantire che l'oggetto possa essere

raggiunto, viene calcolata la posizione obiettivo per la presa dell'oggetto, come specificato in

la sezione precedente.

Dopo aver fatto questo, il braccio viene portato in una posizione standerd hovering, e poi

una posizione di meta intermedia viene impostata direttamente sopra l'oggetto.

Utilizzando il metodo di calcolo del percorso cartesiano, viene calcolata una sequenza di waypoint.

Dopo aver impostato la meta intermedia direttamente sull'oggetto, la navigazione è meno critica,

dato che tutti i possibili oggetti in collisione hanno un'altezza inferiore rispetto al bersaglio

del manipolatore, e un percorso lineare è il modo più veloce per raggiungere l'obiettivo intermedio.

L'etere finale viene poi portato ad un'altezza calcolata come:

zeef = zb + zoffs + heef GD

Il significato dei parametri dierenti è spiegato nel disegno sottostante

5 Scarico di oggetti

Dopo che l'oggetto target è stato afferrato, viene chiamato il metodo backoff(), che

utilizza ancora una volta un percorso lineare cartesinico per alzare il gruppo di oggetti fine etere + afferrato sopra

potenziali oggetti secondari.

Per lo scarico, un nuovo percorso cartesiano, che porta il braccio sopra uno dei due

stazioni selezionate, viene calcolato ed eseguito, e agin il braccio è l'abbassato al

punto che dovrebbe simulare il piano di carico del robot mobile.

6 Riconfigurazione dinamica

Questo nodo è stato implementato per essere riconfigurabile dinamicamente per consentire l'aggiustamento del

molti parametri utilizzati. I parametri diversi utilizzati dalla procedura sviluppata possono

essere regolato dinamicamente utilizzando il ricognitore rqt.

Questo ci permette di selezionare, tra gli altri, i parametri che si riferiscono alle osets,

dimensioni, e la profondità di presa (la quantità di cui la fripper si sovrapporrà, nella z

direzione, l'oggetto).

7 Visualizzazione Rviz

Utilizzando Rviz, è possibile visualizzare molte informazioni utili, come ad esempio:

Gli ostacoli di collisione costruiti

Il waypoint utilizzato per la pianificazione del movimento del manipolatore

Il dierente oggetto calcolato baricentro degli oggetti