

Università degli studi di Padova
Corso di laurea in Ingegneria Informatica
Dipartimento di Elettronica ed Informatica



Elaborato

Sistema di visione distribuito per la localizzazione di un robot mobile

Relatore: Chiar.mo Prof. *Enrico Pagello*
Correlatore: Dott. *Emanuele Menegatti*

Laureando: Gatto Giorgio

Anno Accademico 2003/2004

Ringraziamenti

Desidero ringraziare sentitamente il Dott. Emanuele Menegatti per il fondamentale contributo dato nello sviluppo di questo elaborato e per avermi seguito in questi mesi di lavoro.

Un grazie va al Prof. Enrico Pagello per la disponibilità sempre dimostrata.

Un ringraziamento va anche al Prof. Hiroshi Ishiguro per le fruttuose discussioni sugli argomenti trattati, al Prof. Takashi Minato e agli studenti Takuya Murakita, Daisuke Matsui e Kazuya Denmyouji, dell'Università di Osaka, per aver effettuato l'acquisizione di parte delle immagini necessarie alle sperimentazioni.

Indice

Sommario	- 1 -
Introduzione.....	- 2 -
Struttura dell'elaborato.....	- 5 -
Capitolo 2	
Il problema della localizzazione.....	- 6 -
2.1 Sensore Omnidirezionale.....	- 9 -
2.2 Fourier signature.....	- 12 -
2.3 Calcolo della similarità tra immagini.....	- 16 -
2.4 Localizzazione dinamica.....	- 18 -
Capitolo 3	
Il Sistema di Visione Distribuito.....	- 23 -
3.1 L'ambiente.....	- 23 -
3.2 Organizzazione delle immagini.....	- 26 -
3.3 Calcolo della similarità.....	- 27 -
3.3.1 Conversione da immagine omnidirezionale a cilindro panoramico.....	- 29 -
Capitolo 4	
Individuazione dei Requisiti e Risultati Sperimentali.....	- 34 -
4.1 Problemi incontrati.....	- 35 -
4.2 Analisi delle soluzioni e dei requisiti.....	- 37 -
4.3 Immagini ottimali.....	- 41 -
4.4 Confronto tra sistema statico e dinamico.....	- 43 -
4.5 Invarianza prospettica.....	- 48 -
4.6 Conclusioni sui risultati sperimentali.....	- 50 -
Capitolo 5	
Future Works - Clustering delle immagini.....	- 51 -
5.1 Color matching.....	- 53 -
5.1.1 Calcolo della signature.....	- 55 -
5.1.2 Tuple matching.....	- 57 -
Conclusioni	- 59 -
Elenco delle Figure	- 61 -
Bibliografia	- 64 -

Sommaro

La realizzazione di robot autonomi in grado di funzionare nel mondo reale basandosi su informazioni visive è uno degli obiettivi principali nel campo della robotica e dell'intelligenza artificiale.

In passato sono già stati sviluppati vari tipi di robot autonomi basati su sensori di visione capaci di lavorare in ambienti statici, ossia non soggetti a cambiamenti. Generalmente si utilizzano sistemi nei quali il solo robot è dotato di sensore. In questi approcci il robot riconosce la propria posizione utilizzando come riferimento modelli precedentemente acquisiti: questo significa che il sistema funziona finché l'ambiente rimane invariato o cambia di poco (ad esempio per la presenza di una persona o di un oggetto), mentre la situazione diventa problematica nel momento in cui esso è soggetto a forti cambiamenti, quali ad esempio variazioni di luminosità (come luci accese o spente), presenza di molti oggetti dinamici (ad esempio una folla di persone) o entrambi. In questo caso, infatti, il robot non è più in grado di riconoscere la propria posizione confrontando l'immagine acquisita con quelle di riferimento.

E' ancora difficile realizzare robot autonomi in grado di muoversi in ambienti soggetti a frequenti cambiamenti. L'idea che affronteremo e che punta a risolvere proprio questo problema prevede l'integrazione nell'ambiente di più agenti visivi (OVA, Omnidirectional Vision Agents). Ogni OVA osserva gli eventi nell'ambiente e invia le informazioni visive acquisite ad un centro di elaborazione, il quale riceve anche l'immagine del robot e le confronta. Questo approccio permette al sistema di funzionare, oltre che in ambienti statici, anche in situazioni caratterizzate da un elevato tasso di dinamicità.

Il sistema non si basa sulla localizzazione esplicita del robot nelle immagini catturate dagli OVA, come nei sistemi finora proposti da altri autori, bensì sul confronto tra ciò che vede l'ambiente e ciò che vede il robot.

Questo permette al sistema di funzionare con tipi diversi di robot, come possono essere robot androidi non distinguibili dalle persone, robot talmente piccoli da non essere visti dagli OVA o robot in grado di cambiare aspetto o vestiario.

Dato che gli OVA non rappresentano una forzatura in un meccanismo quale quello dei robot autonomi, possiamo montare un numero di OVA sufficiente in accordo alle necessità: questi, oltre che a permettere la localizzazione, possono anche svolgere altri compiti, come fornire al robot informazioni utili riguardo all'intero ambiente.

Introduzione

L'elaborato affronta un problema di fondamentale importanza nel campo della robotica mobile: la localizzazione basata su informazioni sensoriali.

Si pensi ad un robot mobile, dotato di un sistema di sensori, che deve muoversi autonomamente in un ambiente ed eseguire dei compiti. Per essere in grado di portare a termine i compiti che gli sono assegnati, ha bisogno di conoscere con sufficiente grado di sicurezza la propria posizione nell'ambiente. Deve essere quindi in grado di acquisire, elaborare ed interpretare le informazioni che ottiene dai sensori, al fine di ottenere la rappresentazione del proprio stato nel modello del mondo che ha a disposizione. Questo modello può essere una mappa metrica dell'ambiente, oppure una rappresentazione topologica che descrive solo le connessioni tra le zone dell'ambiente, senza informazioni dettagliate sulle distanze.

Quello della localizzazione non è certo un problema nuovo: sono, infatti, diversi i sistemi già sperimentati in grado di permettere al robot di conoscere la propria posizione.

Una limitazione che accomuna gli approcci classici di localizzazione consiste nella difficoltà di funzionamento in ambienti soggetti a frequenti cambiamenti. Infatti il robot deve essere in grado di conoscere il proprio stato nell'ambiente, anche se questo cambia in continuazione a causa della presenza di persone, oggetti che vengono spostati, veicoli, ecc..., degli esempi possono essere un aeroporto, o addirittura una città.

Questo lavoro si riferisce proprio al problema della localizzazione per un robot mobile che deve navigare in un ambiente dinamico; la soluzione proposta prevede di utilizzare un sistema distribuito (DVS, Distributed Vision System): il sensore non viene montato solo sul robot, ma vengono usati anche sensori visivi disposti in diversi punti dell'ambiente.

Dato che il robot deve poter lavorare in ambienti popolati, ha senso basare le capacità sensoriali funzionali alla localizzazione sulla visione, come accade per l'uomo e per molti animali. Molti ricercatori hanno utilizzato con successo dei sensori di prossimità come "laser range finder" o sonar: in questo caso il sistema di localizzazione deve cercare la posizione che meglio si adatta ai dati letti dal sensore, in una mappa metrica nota a priori. Questi sensori sono però di tipo attivo: interagiscono con l'ambiente, tramite l'emissione di raggi luminosi o di onde sonore e quindi risultano poco robusti ai disturbi a cui è soggetto un ambiente dinamico.

L'uso di sensori visivi permette di acquisire un numero consistente di informazioni; inoltre questi, essendo di tipo passivo, non sono soggetti ad interferenze, spesso causa di problemi in sistemi basati su sensori di prossimità, e risultano particolarmente adatti per la localizzazione in un ambiente dinamico.

I sensori di visione maggiormente utilizzati sono, classicamente, di tipo prospettico: questi forniscono però una vista dell'ambiente attorno al robot limitata alla direzione puntata dalla telecamera.

Il vantaggio portato dall'utilizzo di telecamere omnidirezionali sta nella loro capacità di descrivere l'intera scena attorno al sensore con una sola immagine, e quindi di fornire una descrizione più completa e accurata dello stato dell'ambiente da una certa posizione.

Ogni sensore disposto nell'ambiente non si limita alla semplice acquisizione: si può parlare infatti di OVA (Omnidirectional Vision Agent). Ogni agente visivo cattura l'immagine in un determinato istante e provvede ad elaborarla in modo da rendere efficiente la trasmissione dei dati e la fase di confronto tra le immagini.

La localizzazione in questione, di tipo topologico (non è cioè metricamente precisa ma indica solo l'area nella quale si trova il robot), si basa sul presupposto che, se il robot si trova in un punto particolare dell'ambiente, l'OVA ad esso più vicino riesce a vedere una scena molto simile: il calcolo della posizione avviene tramite il confronto tra le immagini catturate dai sensori disposti nell'ambiente (dette immagini di riferimento) e quella acquisita dal robot (immagine di input). Inoltre, la rappresentazione dell'ambiente a disposizione del robot per la navigazione consiste semplicemente in un grafo rappresentante le connessioni tra gli OVA: ogni sensore costituisce una posizione di riferimento e il robot si riferisce a queste per lo spostamento.

Il confronto tra le immagini omnidirezionali di riferimento e quella di input avviene con la tecnica già utilizzata in [1]. Le immagini vengono elaborate, prima di essere confrontate e trasmesse, tramite l'applicazione della trasformata di Fourier, in modo da ottenere una rappresentazione compatta dell'immagine, detta "Fourier signature"

L'uso della Fourier signature, combinata all'utilizzo di telecamere omnidirezionali, permette al sistema di fruire di due importanti proprietà: l'invarianza rotazionale e quella prospettica.

La prima permette di non dover tenere in considerazione, nella fase di riconoscimento, l'orientazione del robot: due immagini acquisite dalla stessa posizione, ma con orientazioni diverse, vengono comunque riconosciute come identiche.

La seconda permette al robot di localizzarsi correttamente anche in presenza di persone o di oggetti disposti in modo tale che, a causa della differenza di prospettiva, i sensori del robot e dell'ambiente li vedano collocati diversamente nello spazio, come spiegato nel Capitolo 3.

In questo lavoro abbiamo fatto delle prove sul funzionamento del DVS, effettuate in un ambiente non ottimale, sono orientate all'individuazione di eventuali problemi o limiti del sistema. Per risolvere alcuni problemi è possibile modificare la fase di elaborazione in modo da ottenere risultati più soddisfacenti, mentre per altri questo non si può fare: si sono individuati quindi i limiti di flessibilità del sistema e si sono trovati i requisiti per un buon funzionamento.

Il DVS verrà comparato con un sistema di localizzazione visiva che usa solo un sensore, montato sul robot, e confronta l'immagine corrente con un insieme di immagini di riferimento immagazzinate in memoria, con la stessa tecnica di confronto delle immagini presentata in [2]. Il paragone tra due sistemi basati sullo stesso algoritmo di localizzazione permette di verificare l'efficacia del sistema di localizzazione basato su DVS proposto in questo lavoro.

La tecnica di confronto delle immagini utilizzata, qualora non siano soddisfatti appieno i requisiti, può andare incontro a problemi di robustezza. Allo scopo di aumentare l'efficienza nella localizzazione, verrà quindi presentato un metodo di clustering delle immagini. L'obiettivo è quello di basarsi su caratteristiche dell'ambiente che non vengono considerate con il metodo di confronto basato sulla Trasformata di Fourier, per raggruppare le immagini prese da tutti gli OVA in sottoinsiemi (cluster). Effettuata la suddivisione, è possibile individuare, in base alle caratteristiche, a che gruppo appartiene l'immagine di input (quella del robot) ed applicare solo su questo insieme il confronto tra le Fourier signature. La tecnica presentata utilizza i colori per distinguere le immagini: questi infatti non vengono considerati nella Fourier signature e sono un elemento importante per la descrizione di un'ambiente. Verranno proposti esperimenti preliminari volti a confermare l'efficacia di questa tecnica e verrà infine presentato un algoritmo per il calcolo della similarità tra immagini basato sul confronto tra i colori.

Struttura dell'elaborato

L'elaborato è organizzato nel modo seguente:

Capitolo 1 - Il problema della localizzazione

Verranno descritti la localizzazione basata sulle immagini, il sistema di visione basato su sensore omnidirezionale, la Fourier signature e il sistema di confronto delle immagini. Verrà comparato l'approccio distribuito con altre tecniche di localizzazione comuni.

Capitolo 2 - Il Sistema di Visione Distribuito

Saranno esaminati nel dettaglio il sistema di visione utilizzato, l'ambiente in cui gli esperimenti si sono svolti e i passi seguiti per ottenere la localizzazione.

Capitolo 3 - Individuazione dei Requisiti e Risultati Sperimentali

Verranno presentati i set di immagini utilizzati per i test sul sistema di localizzazione. Saranno analizzati i risultati, i problemi incontrati e le soluzioni trovate agli stessi, arrivando a dimostrare la maggiore efficacia del DVS, rispetto ai sistemi statici, nella localizzazione in ambienti dinamici. Verrà provata la proprietà di invarianza prospettica.

Capitolo 4 - Clustering delle Immagini

Saranno presentate le prospettive di sviluppo future per il DVS, proponendo una nuova tecnica di clustering delle immagini basata sui colori. Verrà descritto un algoritmo per il calcolo della similarità, rispetto ai colori, tra le immagini.

Capitolo 1

Il problema della localizzazione

Una delle caratteristiche fondamentali di cui deve essere dotato un robot mobile autonomo è la capacità di pianificare il proprio percorso nell'ambiente. Questo implica la decisione dei movimenti da compiere, a partire dalla posizione iniziale fino a quella di arrivo, scegliendo tra le strade possibili ed evitando, se presenti, ostacoli fissi o mobili che si possono incontrare durante la fase di spostamento. Il primo passo da compiere, per dotare un robot di queste capacità, sta nel permettergli di conoscere la posizione in cui si trova all'interno dell'ambiente in cui lavora.

Gli approcci possibili orientati alla soluzione di questo problema sono molteplici: generalmente al robot viene fornita una descrizione dettagliata dell'ambiente, di norma utilizzando una mappa geometrica (ad esempio una planimetria CAD), oppure una mappa topologica che descriva solo le connessioni tra le zone dell'ambiente senza precise indicazioni metriche. Il robot viene inoltre dotato di uno o più tipi di sensore, come ad esempio una telecamera prospettica od omnidirezionale, un sensore di prossimità o un sensore sonar: la localizzazione avviene cercando la posizione della mappa che coincide maggiormente con i dati acquisiti dal sensore.

Purtroppo, i sensori sono rumorosi e per questo vengono ingannati facilmente dalla complessità e dagli aspetti dinamici dell'ambiente. Per superare questo problema vengono usati dei sensori laser molto precisi, in combinazione con un sistema in grado di gestire l'incertezza (come in [9]).

Un altro approccio, comune nelle applicazioni reali, è quello di modificare l'ambiente. Vengono inseriti dei punti di riferimento artificiali, chiamati "landmark": si usano strisce e punti riflettenti che sono facilmente riconoscibili e che il robot usa per localizzarsi.

Il robot può essere inoltre in grado di costruire una propria rappresentazione dell'ambiente (come ad esempio in [1] e [5]): questo avviene in una fase, detta di "learning", in cui acquisisce i dati dal sensore in varie posizioni, li associa ad una mappa topologica e li immagazzina in database in memoria. Durante il normale funzionamento, per localizzarsi esegue un confronto tra i dati acquisiti nell'istante corrente e quelli contenuti nel database: da questo ricaverà come posizione corretta quella associata al valore memorizzato più simile a quello attuale.

Questi due approcci hanno lo svantaggio di essere “statici”. Anche se si conosce l’ambiente nel dettaglio e si dispone di una mappa precisa, essa è inutile in ambienti altamente dinamici dove la disposizione degli oggetti cambia frequentemente (ad esempio per la presenza di persone, di altri robot che si muovono, oppure perché l’ambiente stesso cambia).

Un terzo approccio, orientato a risolvere proprio questo tipo di problema, consiste nell’utilizzare un sistema distribuito, ossia posizionare i sensori nell’ambiente: in questo caso è l’ambiente stesso a fornire al robot la sua posizione. Questa tecnica non è però esente da problemi: è necessario, infatti, permettere ai sensori disposti lungo il percorso di distinguere il robot dagli altri elementi presenti nell’ambiente. Una possibile soluzione consiste nel dotare il robot di caratteristiche distintive facilmente riconoscibili che possano essere di riferimento per il riconoscimento da parte dei sensori.

Un’altra soluzione, su cui verte il nostro lavoro, consiste nel localizzare il robot confrontando i dati acquisiti tramite i suoi sensori con quelli raccolti da parte dei sensori dell’ambiente. Questo permette al sistema di funzionare, oltre che in situazioni statiche, in presenza di fattori dinamici e con robot di diverso tipo.

Ogni sensore collocato nell’ambiente costituisce una posizione di riferimento, così denominata perché il robot si riferisce ad essa per localizzarsi; perciò i corrispondenti dati acquisiti sono detti dati di riferimento.

In particolare, i sensori utilizzati sono di tipo visivo: ci occuperemo quindi di navigazione, o meglio di localizzazione, basata sulle immagini. Determinando l’immagine di riferimento più simile a quella corrente, il robot può desumere la sua posizione.

Facciamo riferimento, inoltre, non a semplici telecamere, ma ad agenti visivi: essi non si limitano all’acquisizione delle immagini, ma provvedono anche ad elaborarle e trasmetterle. Poiché, come vedremo, i sensori visivi utilizzati sono di tipo omnidirezionale, possiamo parlare di OVA: Omnidirectional Vision Agent.

Parliamo, quindi, di un sistema di visione distribuito (DVS) orientato alla localizzazione di un robot in grado di muoversi in un ambiente dinamico. Utilizzando questa impostazione il problema di localizzazione viene ricondotto al problema di determinare quale tra le immagini di riferimento è più simile a ciò che vede il robot.

Esiste una corrispondenza diretta e predefinita tra un’immagine di input e la corrispondente localizzazione. Questo può costituire una limitazione: infatti, se più di una posizione di riferimento corrisponde ad un’immagine di input, allora il robot non è in grado di decidere tra le possibili ipotesi. Questa situazione si verifica in ambienti con struttura periodica, ad esempio con lo stesso tipo di porte, pareti uniformi, oggetti e illuminazione disposti in modo regolare: è il caso di uffici di uno stesso palazzo che spesso sono arredati in modo simile o di corridoi di ospedali che si ripetono tutti uguali.

Osserviamo, inoltre, che il sistema effettua solo una localizzazione topologica: si può affermare che il robot è più vicino alla corrispondente posizione di riferimento (almeno con un certo grado di confidenza o sicurezza, legato al sistema di similarità) più che ad ogni altra, mentre non è in generale possibile stabilire una posizione metricamente precisa.

A seconda dell'ambiente in cui il robot deve eseguire i propri compiti possiamo considerare diversi modi di strutturare il sistema:

- possiamo servirci di un sistema centralizzato per eseguire il confronto tra i dati acquisiti dal robot e dagli OVA: ognuno invierà le proprie immagini, dopo una fase di pre-elaborazione, ad un server, che provvederà ad effettuare la localizzazione. Questa tecnica è semplice ma poco efficiente: in grandi ambienti in cui serve un numero elevato di OVA è necessario inviare molte informazioni al server, questo può portare a problemi di traffico sulla rete e richiede una elevata capacità di calcolo
- nel caso di ambienti particolarmente estesi e in presenza perciò di molti OVA, si può affidare il compito di effettuare i confronti direttamente al robot. Quest'ultimo riceverà i dati acquisiti dagli OVA e li confronterà con i propri. In questo caso il vantaggio è dato dalla possibilità di eseguire il confronto solo su un sottoinsieme di OVA legato alla posizione in cui il robot si trova. Possiamo infatti pensare di effettuare il trasferimento dei dati mediante l'utilizzo di una connessione wireless tra il robot e ciascun OVA: saranno considerati solo quelli abbastanza prossimi da poter ricevere ed inviare il segnale, tra questi ci sarà sicuramente quello più vicino al robot. Con questa soluzione, però, il robot deve sostenere un elevato carico computazionale
- una terza soluzione prevede di effettuare i confronti in modo distribuito: il robot, sempre tramite una connessione wireless con gli OVA, può inviare in broadcast ciò che ha acquisito. Ogni agente visivo provvederà ad effettuare il confronto tra la propria immagine e quella di input, restituendo il corrispondente valore di somiglianza. Al robot basterà considerare come posizione corrente quella dell'OVA che ha restituito il valore più elevato. Questa è certamente la tecnica più efficiente, sia per ambienti estesi che ristretti, richiede però agenti visivi più sofisticati rispetto alle prime due.

In tutti i casi sopraelencati, comunque, un requisito essenziale, dato che le applicazioni per le quali si utilizzano robot mobili sono nella quasi totalità dei casi di tipo Real-Time, è quella di avere tempi rapidi sia di trasmissione sia di elaborazione dei dati. Gli obiettivi che ci poniamo sono perciò:

- minimizzare la quantità di dati necessaria a descrivere le immagini acquisite;
- massimizzare la velocità di confronto tra le immagini;
- avere un sistema che descriva la scena in modo completo e che permetta di determinare le immagini simili in modo attendibile anche in un ambiente dinamico.

Dobbiamo tenere conto di questo già a partire dalla fase di acquisizione: una soluzione che ci permette di ottenere buoni risultati, soddisfacendo tutti e tre gli obiettivi, consiste infatti nell'utilizzare il minor numero possibile di sensori, in modo da ridurre sia i dati da trasmettere sia il numero di confronti da effettuare, nonché i costi.

La scelta ottimale sta nell'utilizzo di sensori omnidirezionali, la cui proprietà principale è quella di fornire una vista a 360° in una sola volta.

In questo modo servono meno immagini per descrivere l'intero ambiente e inoltre la localizzazione topologica è più accurata, in quanto si ha a disposizione tutta la scena in una volta sola.

Per ridurre la quantità di spazio occupato e conseguentemente il tempo di trasmissione, non vengono trasmesse direttamente le immagini di riferimento, ma viene utilizzata una loro rappresentazione compatta, basata sull'espansione nella serie di Fourier e che chiamiamo "Fourier signature": ogni OVA pre-elabora quanto acquisito e trasmette solamente i coefficienti corrispondenti alle frequenze più basse della serie.

Infatti, come verrà spiegato più avanti, la parte predominante dello spettro di potenza della trasformata di Fourier dell'immagine è contenuta alle frequenze inferiori alla 15^a. Questo permette di trascurare i coefficienti a frequenze maggiori, ottenendo una rappresentazione molto compatta dell'immagine omnidirezionale.

Nel seguito del capitolo vedremo il tipo di sensore utilizzato e come si calcola la "Fourier signature" di un'immagine omnidirezionale, come da questa possiamo ricavare una misura di similarità tra due immagini e come possiamo sfruttare questo sistema per effettuare la localizzazione. Confronteremo infine il DVS con un sistema di tipo statico che utilizza la stessa tecnica di confronto per il riconoscimento della posizione.

1.1 Sensore Omnidirezionale

Come abbiamo visto, allo scopo di ridurre il numero di immagini necessarie abbiamo scelto di utilizzare un sensore di tipo omnidirezionale: esso è costituito da una telecamera prospettica montata verticalmente sotto uno specchio convesso, con l'asse ottico della prima e l'asse geometrico del secondo allineati.

Un sistema di questo tipo è detto catadiottrico: la diottrica riguarda lo studio degli elementi rifrangenti, mentre la catottrica è la parte dell'ottica che studia la riflessione della luce.

Ogni punto dell'immagine catturata corrisponderà alla luce, riflessa sullo specchio, di un punto dell'ambiente, in modo da essere impressa sul sensore della telecamera, come mostrato in Figura 1.1.

Un sistema strutturato in questo modo permette di memorizzare, da un solo punto e in una singola immagine, l'intera vista a 360° dell'ambiente circostante.

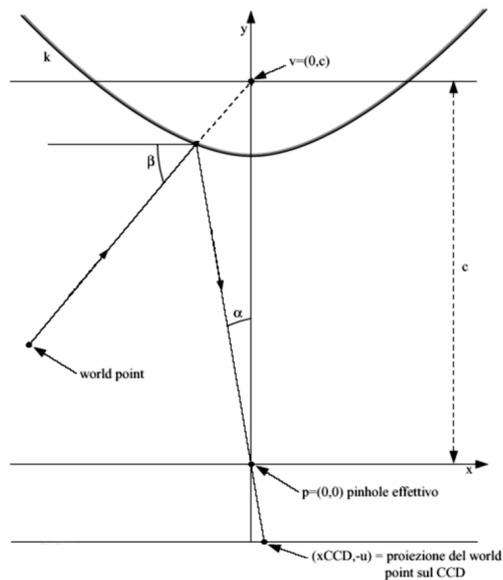


Figura 1.1: Vista in sezione del sistema catadiottrico utilizzato

Molti sistemi di localizzazione per robot mobili utilizzano, invece della telecamera omnidirezionale, una normale telecamera prospettica, che però fornisce un campo visivo ridotto, legato alla direzione puntata: in questo modo sono necessarie più immagini per avere la visione completa dell'ambiente da una certa posizione.

Dato che abbiamo bisogno di collocare nell'ambiente un numero di sensori sufficiente a permettere una corretta localizzazione, l'utilizzo di telecamere con raggio visivo ampio è certamente preferibile rispetto a quelle prospettiche tradizionali.



Figura 1.2: Sensori omnidirezionali

Un aspetto che dobbiamo tenere in considerazione consiste nel fatto che il robot può assumere orientazioni diverse in una stessa posizione: utilizzando telecamere comuni dovremmo essere in grado di garantire che i sensori posizionati lungo l'ambiente abbiano la stessa orientazione di quello montato sul robot, in modo che un'immagine di riferimento rappresenti la stessa scena vista dal robot. Potremmo superare il problema collocando per ogni posizione più sensori orientati diversamente in modo da coprire tutto il raggio visivo possibile, dovendo sostenere però costi molto più elevati, oppure costringere il robot a mantenere un'orientazione fissa in modo che la sua telecamera punti sempre nella stessa direzione dei sensori ambientali: ma questo costituisce una limitazione eccessiva al moto del robot.

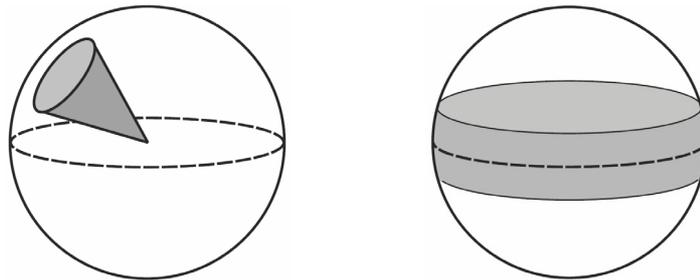


Figura 1.3: confronto tra le superfici visive ottenibili con l'utilizzo di una normale telecamera prospettica e con l'uso di un sensore omnidirezionale.

Inoltre, impiegando più telecamere prospettiche per ogni posizione, il numero di immagini da confrontare aumenta notevolmente: conviene quindi utilizzare un sensore in grado di rappresentare l'ambiente che lo circonda in un'unica immagine, in modo da ridurre costi, tempi di elaborazione e di trasmissione.



Figura 1.4: Esempio di immagine omnidirezionale

Una limitazione che potremmo avere utilizzando immagini omnidirezionali sta nella loro bassa risoluzione rispetto a quelle acquisite con telecamere prospettiche. Nel caso della navigazione, però, la risoluzione non rappresenta un fattore così determinante, anzi l'utilizzo di immagini a bassa risoluzione costituisce un vantaggio, in quanto diminuisce il numero di pixel da elaborare.

Quindi per ogni OVA acquisiremo una sola immagine omnidirezionale.

1.2 Fourier signature

Il secondo passo è la scelta di come effettuare il confronto tra le immagini. L'approccio più semplice potrebbe essere il confronto pixel per pixel, ma questa tecnica, oltre ad essere poco robusta, è inefficiente in quanto richiede di memorizzare l'intera immagine.

Un approccio molto comune per ridurre lo spazio necessario alla memorizzazione consiste nell'effettuare un pre-processing: una tecnica spesso utilizzata è quella di estrarre un insieme di "eigenimages" dalle immagini di riferimento e descrivere queste ultime come combinazione lineare delle "eigenimages".

Ciò che vogliamo è però che la localizzazione riesca indipendentemente dall'orientazione del robot, abbiamo cioè bisogno di una tecnica di confronto che permetta di ottenere l'"invarianza rotazionale". Purtroppo, il metodo appena citato non porta a questo risultato: le immagini appaiono diverse anche se prese nella stessa posizione, ma con orientazioni differenti. Così si deve fissare l'orientazione del robot rispetto ai sensori ambientali, oppure elaborare ulteriormente le immagini per ottenere l'invarianza voluta, aumentando il tempo totale di calcolo.

Utilizzando invece, come abbiamo scelto di fare, una rappresentazione basata sul calcolo della Trasformata di Fourier dell'immagine, riusciamo ad ottenere la proprietà cercata. Non applichiamo la trasformata in entrambe le direzioni all'immagine originale: viene calcolata solamente la trasformata 1-D per ogni riga del cilindro panoramico corrispondente all'immagine omnidirezionale. Il cilindro panoramico è l'immagine ottenuta con una operazione di "unwarping" applicata all'immagine originale. In Figura 1.4 si vede un esempio di immagine omnidirezionale mentre in Figura 1.7 è mostrato il corrispondente cilindro panoramico.

Ottenute le immagini panoramiche, una per ogni immagine omnidirezionale, la seconda fase consiste nel ricavare i coefficienti di Fourier associati ad ogni riga di ogni immagine: questi si ottengono applicando la trasformata, cioè esprimendo il segnale come sommatoria di più componenti, ognuna legata ad una frequenza di variazione spaziale di luminosità.

Come vedremo, i valori di ampiezza e fase dei coefficienti vengono memorizzati separatamente, in quanto descrivono caratteristiche diverse dell'immagine: l'ampiezza viene utilizzata per il calcolo di similarità, mentre la fase consente di stimare l'orientazione del robot.

Riportiamo il grafico dell'ampiezza dei coefficienti di Fourier di una di queste immagini in Figura 1.6. Osserviamo che la potenza dello spettro è contenuta in modo dominante alle basse frequenze, minori della 15^a . Quindi, per ciascuna delle righe del cilindro

panoramico memorizziamo solamente i primi 15 coefficienti di ampiezza e fase della trasformata

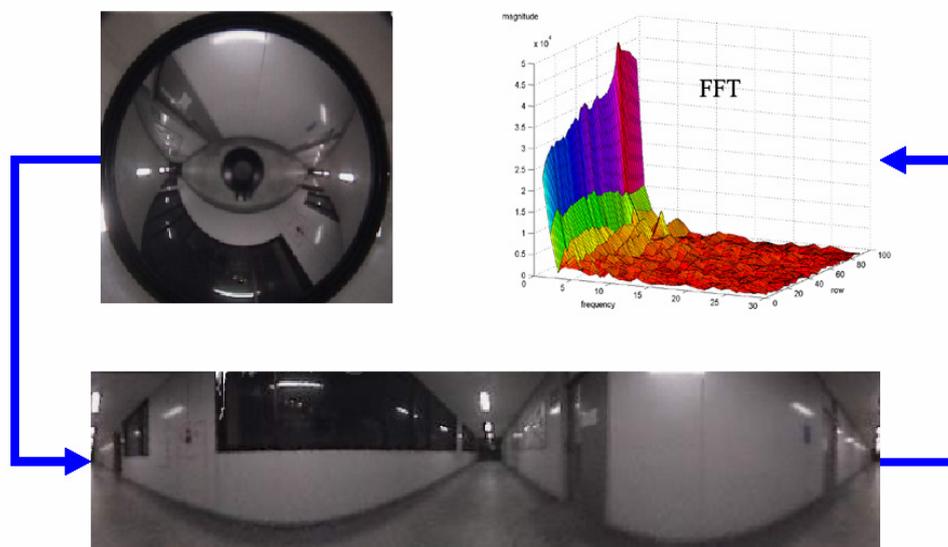


Figura 1.5: Processo di generazione della "Fourier signature" di un'immagine. Dall'immagine omnidirezionale (a) otteniamo il cilindro panoramico (b) tramite un'operazione di "unwarping". Infine calcoliamo i coefficienti della trasformata di Fourier di ogni riga del cilindro panoramico: il grafico è visualizzato in (c).

Immagine	Memoria richiesta (in bit)	Memoria richiesta
Omnidirezionale	320 x 240 x 24	≈ 230 Kb
Cilindro panoramico	512 x 80 x 24	≈ 120 Kb
Fourier signature	80 x 15 x 2 x 8	≈ 2.4 Kb

Tabella 1.1: Riduzione della memoria necessaria per la "signature" dell'immagine omnidirezionale, che otteniamo utilizzando la trasformata di Fourier

La riduzione di memoria che si ottiene con il sistema utilizzato si può vedere in Tabella 1.1: la Fourier signature occupa uno spazio pari a quasi un decimo rispetto all'immagine corrispondente.

L'immagine omnidirezionale originale è di tipo "true color" di 320 x 240 pixel: viene memorizzata usando 8 bit per ciascuno dei 3 canali in formato PPM. Lo spazio occupato corrisponde approssimativamente a 1.8 Mbit, cioè circa 230 Kbyte di memoria. Tramite l'operazione di "unwarping" otteniamo un cilindro panoramico di 512x80 pixel ancora "true color". Il cilindro panoramico occupa circa 120 Kbyte di memoria.

Come abbiamo detto in precedenza, possiamo rappresentare la "signature" dell'immagine omnidirezionale con i soli valori delle prime 15 componenti della trasformata di Fourier. Dato che ogni coefficiente occupa 8 bit, l'intera "Fourier signature" richiede circa 19.2 Kbit, cioè 2.4 Kbyte.

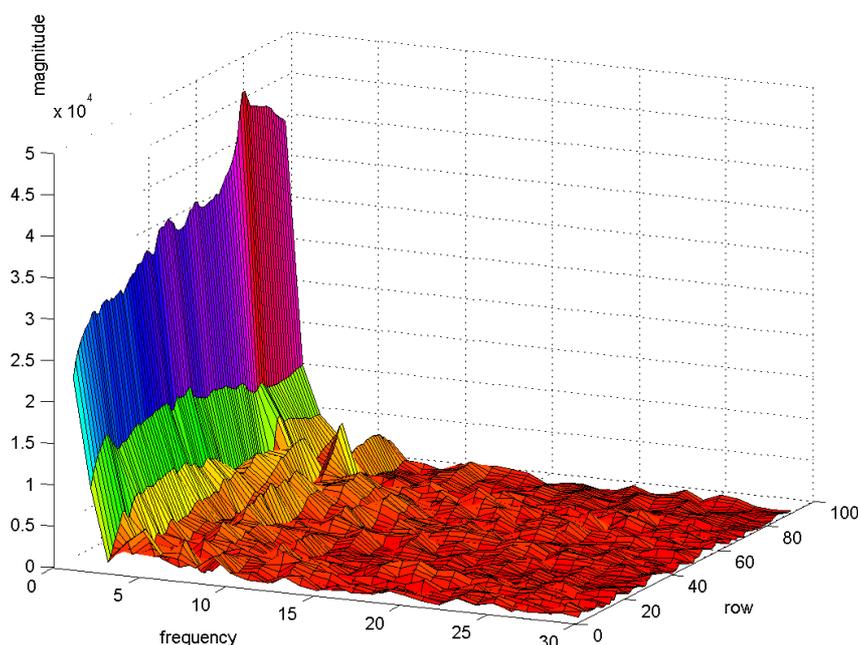


Figura 1.6: Ampiezza dei coefficienti di Fourier di un'immagine omnidirezionale. Sono mostrate le prime 30 componenti. Si nota che le componenti dopo la 15^a assumono valori molto bassi: per questo possiamo utilizzare le sole basse frequenze, fino alla 15^a, come “signature” dell'immagine e trascurare le altre nel calcolo della similarità

Volendo confrontare con il metodo presentato in [5] in cui 936 immagini vengono memorizzate in circa 4 Mbyte, con l'utilizzo della Fourier signature un numero così elevato di immagini richiederebbe circa 2.2 Mbyte.

Possiamo considerare, allo scopo di ridurre il tempo di trasmissione dei dati, la possibilità di affidare la fase di unwarping e quella di generazione della signature all'OVA che ha acquisito l'immagine, dopodiché potrà provvedere a trasmettere il risultato dell'elaborazione, perché possa essere effettuato il confronto con i dati ottenuti dagli altri OVA e dal robot.



Figura 1.7: Cilindro panoramico ottenuto dall'immagine omnidirezionale di Figura 1.3

Il cilindro panoramico è una funzione periodica lungo l'asse x (la lunghezza): questo semplifica il calcolo della trasformata ed è la rappresentazione naturale di una invarianza rotazionale.

Come abbiamo sottolineato, vogliamo essere in grado di determinare l'immagine di riferimento più simile alla vista corrente, indipendentemente dalla sua orientazione. Per questo è necessario introdurre una invarianza rotazionale nel calcolo della similarità: utilizzando i coefficienti di Fourier come “signature” dell'immagine, questo si ottiene automaticamente. Infatti, se il robot acquisisce due immagini omnidirezionali nella stessa posizione ma con orientazioni diverse, allora le due immagini sono in realtà la stessa immagine ruotata rispetto al centro e tale rotazione coincide con la differenza in gradi tra le due orientazioni del robot. Allora i due cilindri panoramici sono la stessa immagine, solo con una traslazione lungo l'asse delle x . Lo stesso vale perciò per l'immagine di riferimento più simile a quella corrente del robot.

Se indichiamo con $f(x)$ il primo cilindro panoramico e con $f(x-a)$ il cilindro panoramico traslato di una quantità pari ad a lungo l'asse x , allora, applicando il Teorema di Traslazione della trasformata, otteniamo:

$$F\{f(x-a)\} = e^{-j2\pi as} F\{f(x)\} \quad (1.1)$$

Quindi la trasformata del segnale traslato è uguale alla trasformata del segnale originale moltiplicato per il numero complesso $e^{-j2\pi as}$, avente modulo unitario. Questo significa che il modulo della trasformata di Fourier dell'immagine traslata non è cambiato e tra le due trasformate c'è solo una variazione di fase proporzionale all'ampiezza della traslazione a .

Sfruttando la proprietà descritta, associamo l'ampiezza della trasformata all'aspetto (vista) dell'ambiente da una certa posizione mentre associamo la fase della trasformata all'orientazione del robot. In questo modo se il robot ruota su se stesso, sul posto la vista dell'ambiente e la relativa ampiezza della trasformata non cambiano. Quello che cambia è la fase della trasformata, che varia proporzionalmente alla variazione di orientazione.

Con il metodo descritto abbiamo ottenuto l'invarianza rotazionale cercata. Inoltre abbiamo anche un modo per calcolare l'orientazione del robot: infatti, dato che gli OVA sono fissi nell'ambiente e ne conosciamo a priori l'orientazione, dopo aver determinato a quale OVA è più vicino il robot, possiamo eseguire il confronto tra la fase ricavata dalle due immagini per stimare l'orientazione del robot [1].

Un'importante proprietà derivante dall'utilizzo della Fourier signature, oltre all'invarianza rotazionale, consiste nell'invarianza prospettica: se in prossimità sia del robot sia dell'OVA più vicino ad esso sono presenti uno o più oggetti o persone è possibile che, per differenza di prospettiva, il robot li veda in una certa posizione rispetto all'ambiente circostante, mentre per l'OVA siano collocati diversamente (Capitolo 3).

La situazione descritta sopra non influisce sul riconoscimento: anche se nelle due immagini sono presenti le stesse occlusioni disposte in modo diverso, l'applicazione della trasformata avviene solo sulle righe, questo rende ininfluente la distribuzione spaziale degli oggetti in senso orizzontale (mentre non è così per la distribuzione spaziale in senso verticale).

1.3 Calcolo della similarità tra immagini

Per il calcolo della similarità di due immagini omnidirezionali definiamo innanzitutto una funzione di dissimilarità che riceve come parametri le “signature” delle immagini, cioè due vettori di m valori, ciascuno rappresentante l'ampiezza dei primi m coefficienti di Fourier associati alle due immagini.

La funzione di dissimilarità tra due immagini omnidirezionali O_i, O_j è calcolata come la norma $L1$ dei coefficienti di Fourier dei cilindri panoramici corrispondenti. Ciascun cilindro è composto da l righe. Inoltre consideriamo solo i primi m coefficienti della serie di Fourier poiché le componenti alle alte frequenze hanno potenza trascurabile:

$$Dis(O_i, O_j) = \sum_{y=0}^{l-1} \sum_{k=0}^{m-1} |F_{iy}(k) - F_{jy}(k)| \quad (1.2)$$

dove $F_{iy}(k)$ e $F_{jy}(k)$ sono i coefficienti di Fourier alla frequenza k della y -esima riga di O_i e O_j rispettivamente.

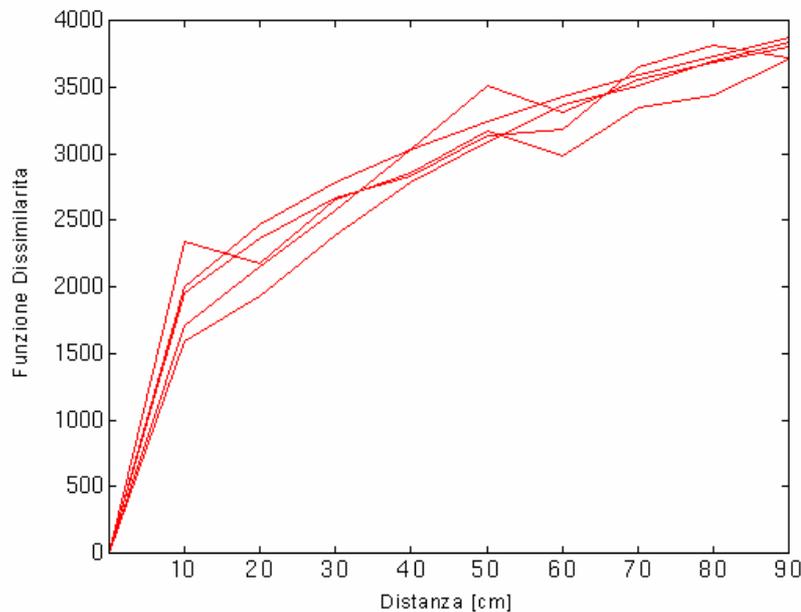


Figura 1.8: Grafico del valore della funzione di dissimilarità in relazione alla distanza tra le immagini. Ciascuna linea rappresenta una coppia diversa di immagini. Si vede che la relazione è lineare per piccole distanze, mentre tende a saturare per distanze elevate.

Si vede dalla Figura 1.8 che il valore della funzione $Dis(O_i, O_j)$ cresce linearmente con la distanza tra l'immagine corrente e quella di riferimento; questo vale nel breve raggio, mentre per distanze maggiori la funzione satura ad una costante che corrisponde ad un

valore di “no-matching” restituito quando il calcolo della sommatoria dell'equazione 1.2 supera una soglia scelta per velocizzare l'algoritmo. Questa scelta è giustificata dal fatto che in tal caso le immagini sono state prese in posizioni distanti e non c'è più nessuna correlazione tra esse.

Per convenienza definiamo anche una funzione di similarità ottenuta riscaldando i valori di dissimilarità tra 0 e 1000:

$$Sim(O_i, O_j) = 1000 - 1000 \frac{Dis(O_i, O_j) - \min_{i,j} Dis(O_i, O_j)}{\max_{i,j} Dis(O_i, O_j) - \min_{i,j} Dis(O_i, O_j)} \quad (1.3)$$

Il valore di $Sim(O_i, O_j)$ è decrescente con la distanza ed esprime correttamente il significato di una stima di similarità: maggiore per immagini più vicine e via via decrescente per immagini più lontane.

Osserviamo che il valore assoluto della funzione di similarità non è importante e dipende dalle caratteristiche dell'ambiente. Sono rilevanti solo i valori relativi della similarità dell'immagine corrente rispetto a tutte le immagini di riferimento. L'immagine di riferimento più simile all'immagine corrente è quella a cui corrisponde il valore più basso della funzione di dissimilarità e quindi il valore più alto, pari a 1000, della funzione di similarità.

Quindi, data l'immagine corrente, possiamo ottenere la sua “Fourier signature” con il processo descritto nella sezione precedente. A questo punto possiamo calcolare i valori di similarità tra quest'immagine e quelle di riferimento secondo l'equazione (1.3) che calcola la norma $L1$, scalata tra 0 e 1000, tra le “Fourier signature”. Le immagini di riferimento con valori più alti della funzione di similarità sono quelle considerate più simili a quella corrente. In particolare l'immagine di riferimento con valore di similarità pari a 1000 fornisce la localizzazione topologica del robot, poiché possiamo affermare che il robot è più vicino alla posizione di questa immagine di riferimento rispetto a tutte le altre.

Per un sistema di localizzazione classico, quanto detto sopra è vero solo se non sono presenti delle notevoli occlusioni o disturbi nelle immagini, che le alterano in modo così significativo da impedire la corretta individuazione dell'immagine più simile. Come vedremo nei capitoli successivi, il DVS consente di superare queste limitazioni consentendo il riconoscimento anche in presenza di occlusioni o di variazioni dell'immagine in generale.

Ricordiamo infine che la localizzazione basata sulle immagini, e in particolare il sistema appena descritto, fallisce in ambienti con struttura periodica come nel caso dei labirinti, o più comunemente di ambienti con corridoi uguali. In questi casi le viste dell'ambiente sono le stesse in posti diversi e la vista corrente può coincidere con più di una vista di riferimento: il sistema descritto non è in grado di decidere tra le varie posizioni.

Questo è un problema di “perceptual aliasing” ed è la situazione in cui si può trovare un uomo che si perde in un ambiente in cui tutti i posti sembrano simili e non esistono dei punti di riferimento distinti. Per orientarsi una persona utilizza altri indizi, come i cartelli

stradali, oppure cerca di memorizzare le direzioni delle strade percorse. Un robot può utilizzare dei sensori più sofisticati della visione, come i sensori GPS (“Global Positioning System”). Ma spesso questi sensori non sono utilizzabili, ad esempio in ambienti indoor oppure all'esterno ma coperti da alberi, dove il segnale GPS non può arrivare; oppure risultano troppo costosi da installare sul robot rispetto all'applicazione a cui è dedicato. Per questo è necessario utilizzare delle tecniche che siano in grado di gestire l'incertezza delle situazioni in cui il robot ha evidenza di trovarsi in due o più posizioni distinte: un esempio è la tecnica illustrata in [2], basata sul metodo di Monte Carlo.

1.4 Localizzazione dinamica

La localizzazione con il DVS avviene in più fasi, come mostrato in Figura 1.9: ad ogni ciclo sia il robot che tutti gli OVA acquisiscono l'immagine tramite il sensore omnidirezionale. Fatto questo, provvedono ad elaborarla in modo da ottenere una rappresentazione che consenta di sfruttare le proprietà di invarianza rotazionale e invarianza prospettica tramite la conversione in cilindro panoramico. Ottenuta la rappresentazione panoramica, calcolano poi una rappresentazione compatta dell'immagine, in modo da incrementare le prestazioni di trasmissione e calcolo della similarità, utilizzando la trasformata di Fourier. Ogni OVA trasmette poi la Fourier signature ottenuta, in modo che possa essere confrontata, insieme a quelle degli altri OVA, con quella del robot.

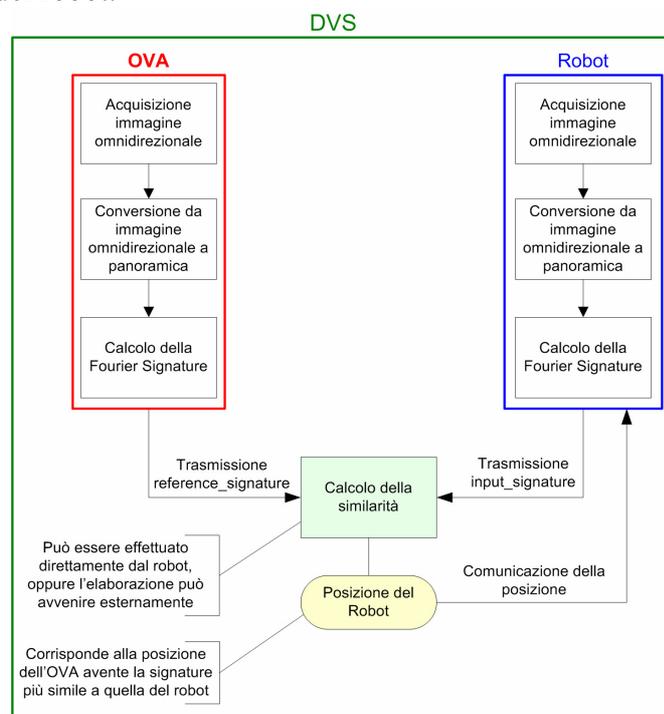


Figura 1.9: Fasi per il calcolo della posizione del robot con l'utilizzo del DVS

Dal confronto tra le immagini di riferimento e quella del robot otteniamo un valore di dissimilarità per ogni coppia (signature_riferimento, signature_robot) secondo la formula 1.2; questi valori vengono poi riscaldati tramite la formula 1.3 in modo da essere compresi tra 0 e 1000: il valore massimo corrisponde alla coppia avente il più alto grado di similarità. La posizione in cui si trova il robot corrisponderà a quella dell'OVA caratterizzato dalla signature relativa alla coppia (signature_riferimento, signature_robot) avente similarità massima.

Come abbiamo detto, questo approccio consente al robot di localizzarsi in ambienti con un grado di dinamicità molto elevato, mentre spesso sistemi di altro tipo falliscono in presenza di troppi cambiamenti.

Per verificare l'efficacia del DVS, possiamo confrontarne le prestazioni rispetto ad un sistema statico. Un approccio che particolarmente si presta ad un confronto è quello descritto in [1]: in questo infatti il calcolo della similarità viene effettuato con lo stesso metodo utilizzato dal DVS. Come si vede dalla Figura 1.10, la differenza sta nel fatto che questo sistema non usa sensori diversi da quello del robot.

Le immagini di riferimento non vengono, come nel DVS, acquisite nello stesso istante sia dal robot che da agenti visivi disposti nell'ambiente, sono immagazzinate (in forma di Fourier signature) nella memoria del robot in una prima fase in cui il robot esplora l'ambiente e memorizza le Fourier signature corrispondenti alle viste dell'ambiente nelle diverse posizioni.

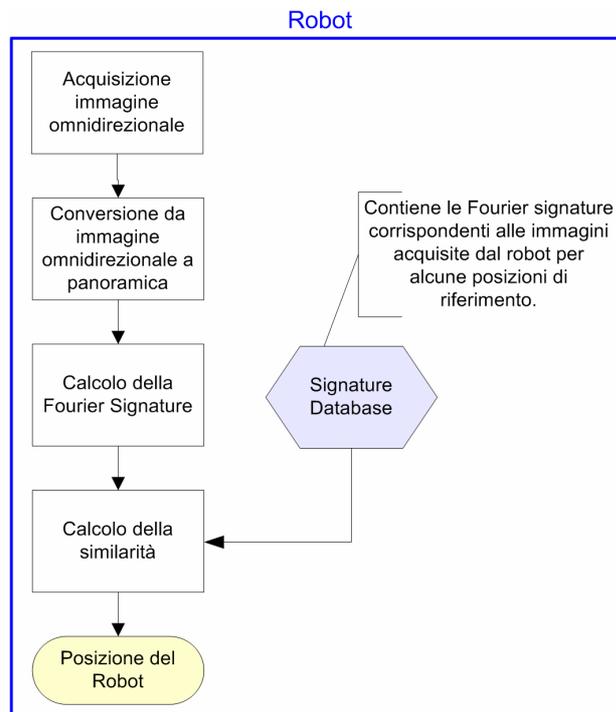


Figura 1.10: Fasi per il calcolo della posizione del robot con il sistema descritto in [1].

Il sistema appena citato può fallire nel caso l'ambiente subisca variazioni: se infatti il robot acquisisce l'immagine in una posizione dove ci sono state variazioni ambientali consistenti, la signature di riferimento (immagazzinata nel database) corrispondente all'immagine catturata, nella stessa posizione ma in tempi diversi, sarà difficilmente simile a quella corrente. Questa situazione è rappresentata in Figura 1.12: nella prima situazione l'immagine di riferimento e quella corrente sono praticamente identiche, in quanto l'ambiente non è cambiato, e la localizzazione riesce senza problemi. Nel secondo caso ci sono delle persone attorno al robot: l'immagine da esso acquisita è diversa da quella di riferimento presa nella stessa posizione, perciò il riconoscimento fallisce.

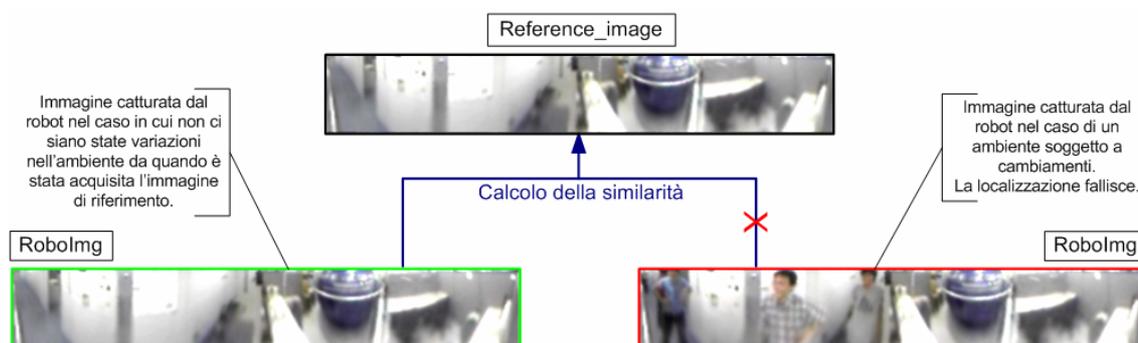


Figura 1.11: Localizzazione con l'utilizzo di un sistema statico in 2 condizioni: la prima senza variazioni ambientali, la seconda con persone che si muovono nell'ambiente.

In Figura 1.11 si vede invece come utilizzando il DVS la localizzazione riesca anche in presenza di contenuti dinamici: la stessa situazione vista dal robot viene catturata anche dagli OVA nell'ambiente, perciò il riconoscimento va a buon fine.

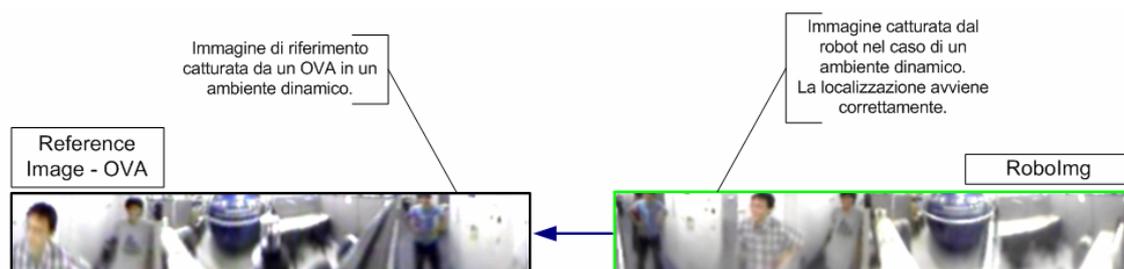


Figura 1.12: Localizzazione con l'utilizzo del DVS con persone che si muovono nell'ambiente.

La localizzazione che otteniamo con l'utilizzo del DVS è esclusivamente topologica: non è possibile stabilire con precisione la posizione del robot, possiamo solo sapere che esso si trova in prossimità dell'OVA che ha fornito la signature di riferimento più simile. Osservando il grafico di Figura 1.8, vediamo che la funzione di dissimilarità (1.2) satura per distanze superiori a circa un metro: possiamo quindi supporre che se un OVA è indicato come il più vicino al robot, quest'ultimo si trovi ad una distanza massima di un metro dal sensore ambientale. Questo implica anche che, se vogliamo ottenere una localizzazione affidabile, è necessario disporre gli agenti visivi nel campo d'azione del

robot in modo che ci sia sempre un sensore in grado di fornire un'immagine di riferimento che sia simile a quella del robot: in questo caso quindi avremo bisogno di disporre un sensore ogni due metri o meno.

Verificata quest'ultima condizione, possiamo fornire al robot la rappresentazione topologica dell'ambiente: una rappresentazione che potremmo utilizzare è costituita da un grafo orientato, dove i nodi rappresentano gli OVA e gli archi le connessioni tra gli agenti visivi: ad ogni arco potremmo inoltre associare la distanza e l'orientazione tra gli OVA (come in Figura 1.14)

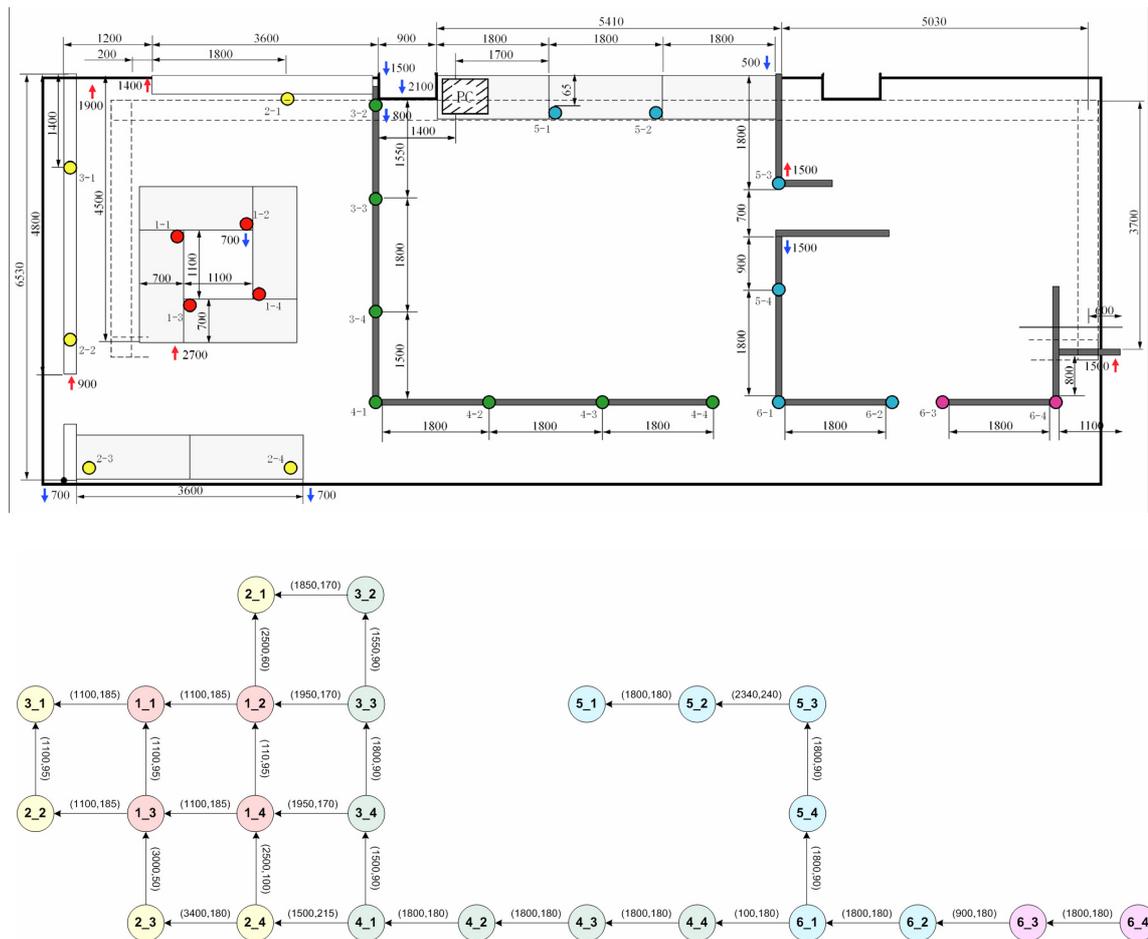


Figure 1.13 e 1.14: Mappa metrica dell'ambiente dove si sono svolti i test sul DVS e un possibile grafo rappresentante le connessioni tra gli OVA.

Durante la navigazione, per raggiungere un punto nell'ambiente, il robot dovrà prima di tutto conoscere la propria posizione, utilizzando la tecnica vista finora. Una volta determinato l'OVA più vicino (e quindi il nodo del grafo corrispondente), il robot dovrà calcolare il percorso più breve (il cammino di costo minimo tra due nodi) per giungere a destinazione, e quindi spostarsi in direzione del prossimo OVA (il nodo successivo del cammino sul grafo) nella direzione corrispondente. Per fare questo dovrà comparare la

propria orientazione con quella dell'OVA corrente tramite il confronto tra le Fourier signature: conoscendo questa potrà dirigersi nella direzione indicata dal valore associato all'arco che unisce i nodi corrispondenti uno alla posizione corrente e l'altro alla prossima posizione.

Quindi ogni OVA costituisce una posizione di riferimento per il robot: questo implica che più fitta sarà la rete di sensori, più accurato sarà lo spostamento del robot. In relazione alla zona in cui il robot deve spostarsi è possibile dunque disporre gli OVA con distanze diverse: ad esempio in uno spazio ampio e privo di ostacoli fissi, tale cioè da permettere un movimento libero, sarà possibile collocare i sensori alla distanza strettamente necessaria al riconoscimento, mentre per zone dove il robot deve spostarsi con maggiore accuratezza, ad esempio in prossimità di una porta, si possono montare sensori più ravvicinati.

Capitolo 2

Il Sistema di Visione Distribuito

2.1 L’ambiente



Figure 2.1 e 2.2: Il robot utilizzato nell’acquisizione delle immagini. Nella figura di destra si possono notare gli OVA collocati sui muri lungo il corridoio e il sensore posizionato sul robot.

Per i test sul funzionamento del DVS abbiamo fatto riferimento ad un robot autonomo, l’Eveliee P1, mostrato in Figura 2.1. Sul robot è stato montato un sistema di visione di tipo omnidirezionale formato da una telecamera montata in verticale in modo da puntare uno specchio iperbolico. Questo sistema consente di avere una visione completa, a 360°, dell’ambiente circostante con l’acquisizione di una sola immagine.

I test sono stati svolti in un ambiente indoor, presso il laboratorio di intelligenza robotica del Dipartimento di “Adaptive Machine Systems” della Facoltà di Ingegneria dell’Università di Osaka (mappa in Figura 2.5).



Figura 2.3: Un color Quad Processor



Figura 2.4: I sei Color Quad Processor utilizzati

L'ambiente è costituito da un lungo corridoio, al termine del quale si trova una stanza con un tavolo al centro. Nell'ambiente sono stati montati 24 OVA, con sensori simili a quello collocato sul robot (Figura 2.5).

Le immagini raccolte dagli OVA vengono inviate a dei Color Quad Processor (Figura 2.3): questi permettono di gestire contemporaneamente più telecamere, inoltre effettuano anche un'elaborazione delle immagini. In particolare, vengono utilizzati dei Quad Processor prodotti dalla Av Tech[®], modello AVC704H.

Nelle prove svolte abbiamo utilizzato sei Color Quad Processor, uno ogni quattro OVA. Come si vede in Figura 2.6, ogni Quad Processor fornisce in output un'immagine bitmap ad una risoluzione di 640x480 pixel contenente le quattro immagini riprese dagli OVA ai quali fa riferimento, ed è identificato con un numero da 1 a 6. Ogni OVA associato ad un Quad Processor è identificato con un numero da 1 a 4: ogni OVA è perciò identificato univocamente dalla coppia $[numeroQuad \ numeroOVA]$.

Gli OVA sono distribuiti nell'ambiente nel modo seguente (facciamo riferimento, come posizione di osservazione, alla Figura 2.5):

- 8 OVA, il gruppo 6 e il gruppo 4, montati sui muri lungo il corridoio
- 4 OVA, il gruppo 5, nella stanza adiacente alla parte finale del corridoio
- 3 OVA, il 3_1, 3_2, 3_3, sul muro a destra del tavolo
- 1 OVA, il 3_1, a ridosso del muro dietro al tavolo
- 2 OVA, il 2_1 e il 2_2, a ridosso del muro alla sinistra del tavolo
- 2 OVA, il 2_3 e 2_4, a ridosso del muro davanti al tavolo
- 4 OVA, il gruppo 1, montati sopra il tavolo

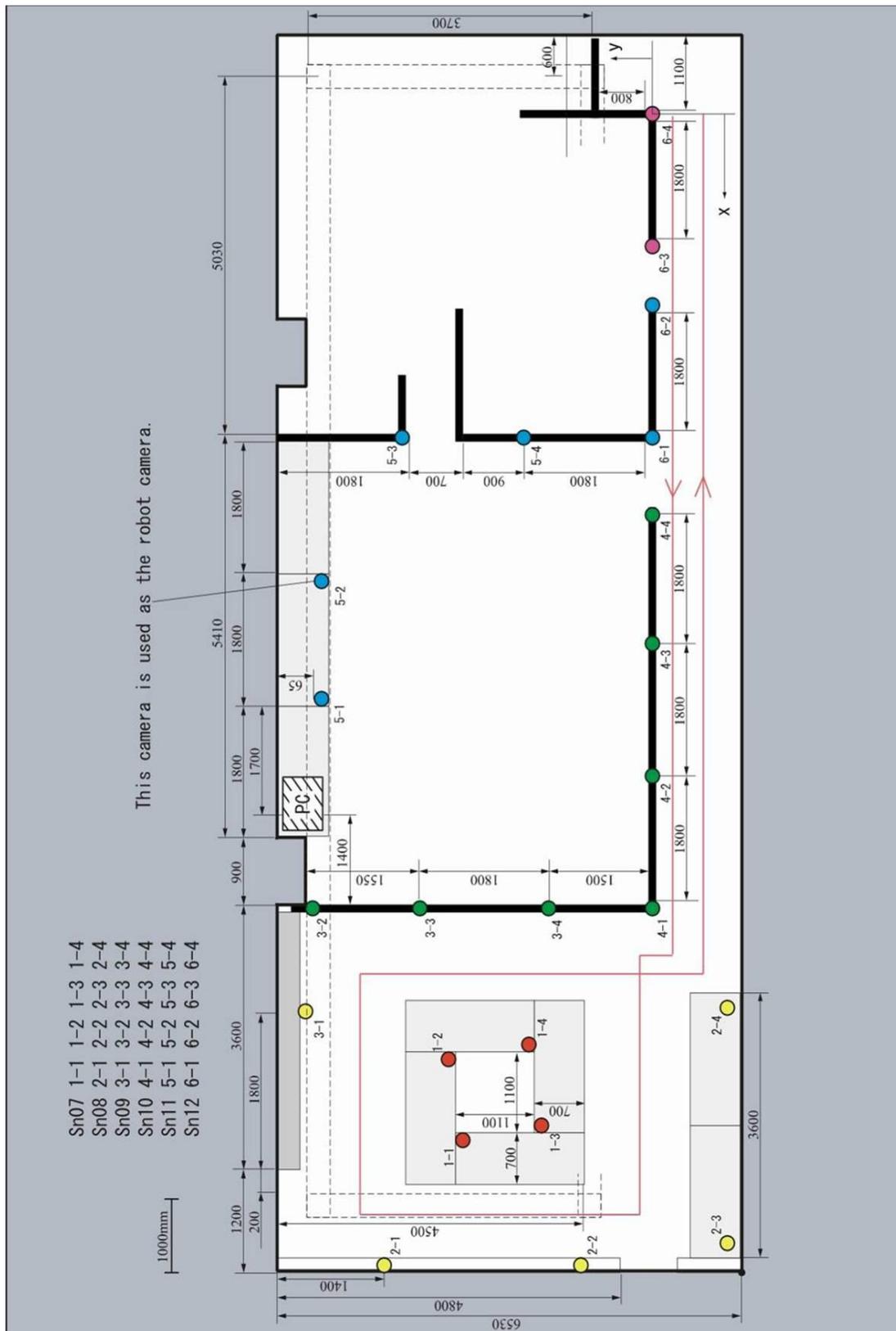


Figura 2.5: Mappa del laboratorio dove sono state acquisite le immagini. Sono segnate le posizioni degli OVA ed il percorso del robot.

Come vedremo nel capitolo successivo, gli OVA non sono collocati in modo da favorire il riconoscimento con l'immagine presa dal robot: alcuni infatti, quelli appartenenti ai gruppi 2, 3 e 5, sono troppo lontani rispetto al percorso del robot e forniscono immagini troppo diverse. Saranno comunque utili in quanto rappresentano un elemento di disturbo nella fase di riconoscimento. Gli OVA del gruppo 1 sono invece collocati ad un'altezza minore rispetto al sensore del robot, mentre gli OVA dei gruppi 4 e 6 sono posizionati sopra a dei muri separatori che costituiscono un ostacolo visivo per il robot.

Le immagini a cui faremo maggior riferimento saranno quelle prese dagli OVA ai quali il robot passa più vicino durante il suo tragitto, ossia quelli posti lungo il corridoio (gruppi 4 e 6) e quelli montati sul tavolo (gruppo 1).

Il percorso compiuto dal robot è mostrato in Figura 2.5: esso parte dall'inizio del corridoio, in prossimità dell'OVA 6_4, percorrendolo incontra nell'ordine gli OVA 6_3, 6_2, 6_1, 4_4, 4_3, 4_2, 4_1: quest'ultimo si trova al termine del corridoio. Il robot prosegue poi facendo il giro del tavolo, incontrando gli OVA 1_4, 1_3, 1_1, 1_2 e di nuovo 1_4. Infine ripercorre il corridoio a ritroso, incontrando gli stessi OVA dell'andata, ma ovviamente in ordine inverso.

2.2 Organizzazione delle immagini

Come abbiamo visto, i Color Quad Processor che si occupano dell'elaborazione delle immagini forniscono in output un'unica immagine ciascuno costituita dalla fusione delle quattro immagini prese dagli OVA ai quali il Quad fa riferimento.

Per ogni posizione del robot avremo perciò 7 immagini: una è quella presa dal robot stesso, mentre le restanti 6 ci vengono fornite dai Quad Processor (Figura 2.6).

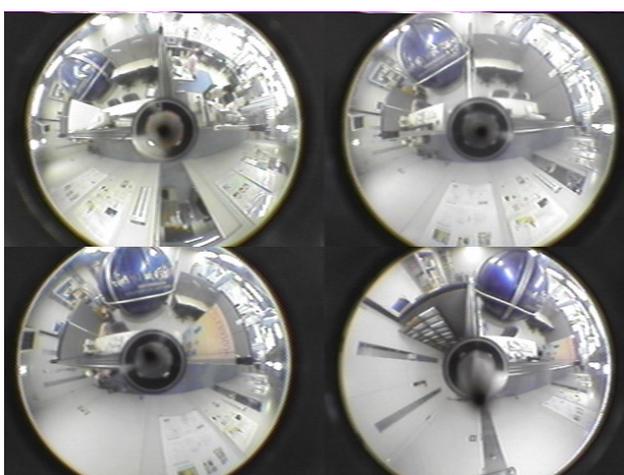


Figura 2.6: risultato dell'elaborazione del Quad Processor 6: le immagini risultanti dalla divisione saranno quelle corrispondenti agli OVA 6_1 , 6_2 , 6_3 , 6_4.

Dato che ciò che vogliamo è confrontare l'immagine del robot presa in un determinato istante con le immagini prese da tutti gli OVA nello stesso istante, avremo bisogno di dividere ogni immagine in quattro, in modo da ottenere 25 immagini: una per ogni OVA più quella del robot.

Più in dettaglio, divideremo l'output del Color Quad Processor n in 4 immagini in modo da ottenere le rappresentazioni dell'ambiente corrispondenti agli OVA n_1, n_2, n_3, n_4, tutte alla risoluzione di 320 x 240 pixel.

In realtà, nonostante le immagini da noi considerate siano ad una risoluzione di 320 x 240, la massima area di interesse è minore, di 240 x 240 pixel.

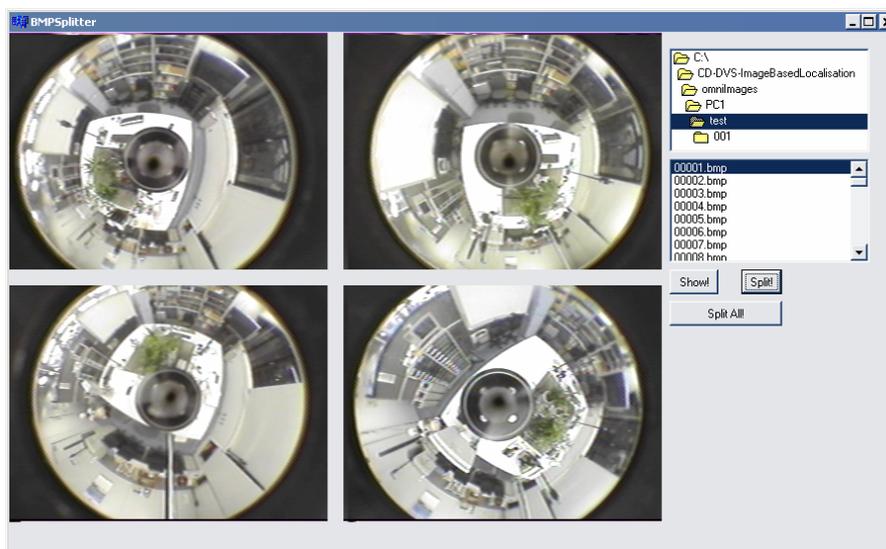


Figura 2.7: Programma realizzato per la divisione delle immagini di partenza

2.3 Calcolo della similarità

Per il calcolo della similarità tra immagini abbiamo utilizzato la procedura spiegata nel Capitolo 1.

Vi è una importante differenza tra l'approccio statico, in cui il solo robot è dotato di un sistema di visione, e l'utilizzo di un sistema distribuito: in quest'ultimo infatti vi sono più sensori ottici aventi caratteristiche diverse. E' quindi necessario elaborare le immagini tenendo conto della diversità dei parametri di ogni sensore, variabili a seconda del tipo di specchio utilizzato, della distanza di questo dalla telecamera e del tipo di telecamera.

La prima fase consiste nel ricavare per ogni immagine omnidirezionale il corrispondente cilindro panoramico.

L'algoritmo di unwarping di cui disponevamo rendeva molto complessa la fase di setting, per cui abbiamo deciso di reimplementarlo in modo che potesse tenere conto di parametri quali:

- la costante di curvatura dello specchio iperbolico
- la distanza tra il fuoco dello specchio iperbolico e la telecamera
- le coordinate del centro dell'immagine omnidirezionale

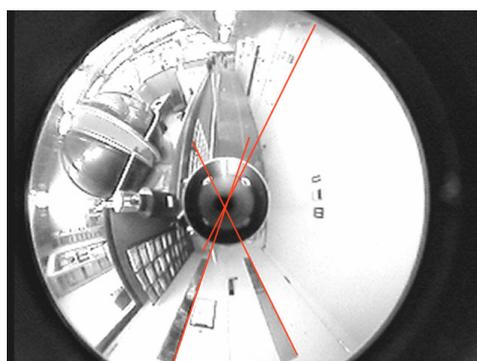


Figura 2.8: Determinazione del centro dell'immagine omnidirezionale

La determinazione del centro esatto dell'immagine omnidirezionale è importante per poter ottenere una rappresentazione panoramica corretta. Infatti nell'immagine omnidirezionale i punti dell'ambiente vengono mappati in una rappresentazione radiale, questo implica che le linee che nell'ambiente sono verticali nell'immagine convergono esattamente verso il centro: bisogna conoscere perciò la posizione di quest'ultimo per poter applicare il processo inverso e ottenere un cilindro panoramico che rappresenti in modo esatto l'ambiente.

Inoltre il centro della rappresentazione omnidirezionale raramente corrisponde con il centro esatto dell'immagine: questo perché spesso specchio e telecamera non sono perfettamente allineati, o perché il sensore è leggermente inclinato.

Per determinare il centro dell'immagine omnidirezionale si procede quindi come mostrato in Figura 2.8: si individuano prima gli oggetti dell'ambiente disposti in modo da essere il più verticali possibili, come ad esempio porte o finestre, e si prolungano le linee corrispondenti nell'immagine: queste si intersecheranno nel centro.

Un secondo motivo, spiegato nel capitolo successivo, che ha portato alla decisione di riscrivere l'algoritmo di conversione da immagine omnidirezionale a panoramica, è l'esigenza di poter effettuare l'unwarp solo su un'area delimitata dell'immagine originale.

Data la bassa risoluzione delle immagini omnidirezionali (240 x 240 pixel), i cilindri panoramici ottenuti risultano sgranati, una terza esigenza è quindi quella di applicare dei filtri di blur Gaussiano all'immagine: infatti un'immagine troppo "granulare" potrebbe falsare i valori delle componenti ad alte frequenze della FFT nel calcolo della similarità.

Ottenute le immagini panoramiche, una per ogni immagine omnidirezionale, ad una risoluzione di 512 x 80 pixel (è necessario che il numero di pixel appartenenti alle colonne dell'immagine sia una potenza di due per poter applicare la FFT), dobbiamo ricavare i coefficienti di Fourier associati ad ogni riga di ogni immagine. Da questa fase si ottengono in output due file di testo per ogni cilindro panoramico, contenenti la signature dell'immagine, entrambi formati da n righe e m colonne, dove n è il numero di righe dell'immagine e m il numero di pixel associato ad ogni pixel dell'immagine panoramica. Il primo file contiene il modulo di ogni coefficiente, utile al confronto per la similarità, mentre il secondo contiene la fase, per permettere di comparare l'orientazione del robot rispetto a quella dell'OVA.

Il terzo e ultimo passo per il calcolo della similarità consiste nel comparare ogni signature relativa agli OVA con la signature corrispondente all'immagine acquisita dal robot. Per fare ciò ci serviamo della formula 1.2. Otteniamo così, per ogni posizione del robot, 24 valori che esprimono il grado di dissimilarità tra le 24 immagini di riferimento degli OVA e quella di input. Questi valori vengono riscaldati usando la formula 1.3: a quello che esprime il massimo grado di somiglianza corrisponderà il valore 1000 mentre a quello corrispondente alla similarità minima sarà associato il valore 0.

2.3.1 Conversione da immagine omnidirezionale a cilindro panoramico

Il nostro scopo è quello di ricostruire, a partire dalla rappresentazione omnidirezionale dell'ambiente, la corrispondente rappresentazione panoramica.

Per poter fare ciò dobbiamo conoscere, dato ogni pixel dell'immagine, a quale punto dell'ambiente corrisponde. Vogliamo poter selezionare due circonferenze che delimitino internamente ed esternamente l'immagine omnidirezionale e, a partire dalla corona circolare formata da queste (ROI, Region Of Interest), trovare a che zona dell'ambiente corrisponde questa regione, per poi dividerla in tante parti quante sono le righe che vogliamo ottenere nel cilindro panoramico. In questo modo possiamo trovare, per ogni punto dell'ambiente, quale pixel dell'immagine omnidirezionale lo rappresenta e mapparlo nell'immagine panoramica corrispondente. Per fare ciò dobbiamo considerare la geometria del sensore omnidirezionale.

Per generare immagini geometricamente prospettiche a partire da immagini acquisite con un sensore catadiottrico omnidirezionale, questo deve essere dotato di un unico punto di vista effettivo (SVP, single view point). Il vincolo di SVP implica che ogni punto dell'ambiente venga riflesso in uno e un solo punto del sensore CCD della telecamera prospettica, in modo che sia possibile mappare tutti i punti del piano dell'immagine in punti del CCD. Gli unici sensori praticamente utili che soddisfano esattamente questa proprietà sono lo specchio iperbolico con una telecamera prospettica e lo specchio parabolico con telecamera ortogonale. Noi faremo riferimento solo al primo tipo, in quanto tutti i sensori utilizzati nei test sono di questa specie. Prenderemo in

2.3.1 – Conversione da immagine omnidirezionale a panoramica

considerazione inoltre solo gli specchi ottenuti per tornitura, perciò dotati di simmetria rotazionale attorno ad un asse.

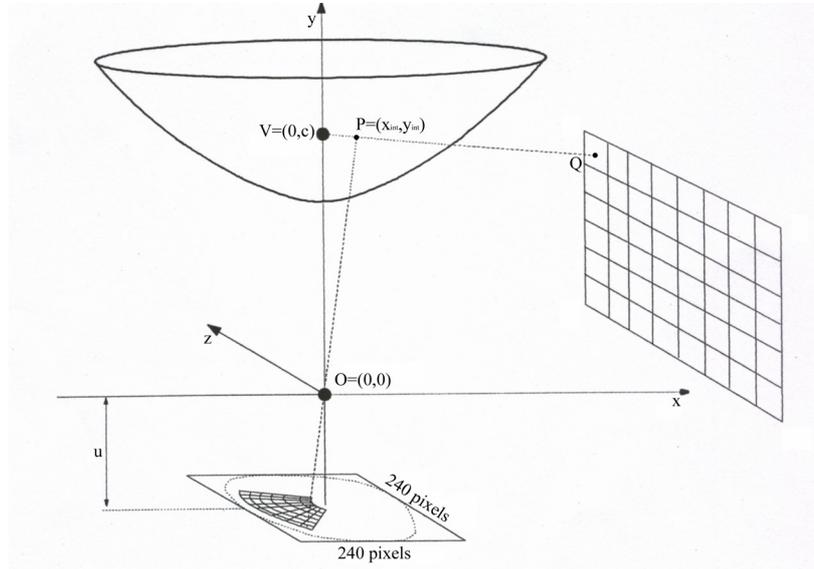


Figura 2.9: Corrispondenza tra i punti dell'immagine omnidirezionale (i pixel del sensore CCD) e quelli dell'ambiente (Immagine tratta da [18]).

In [12] è derivata la classe completa di tutti i possibili profili che risolvono esattamente il vincolo del SVP. La soluzione generale dell'equazione differenziale che si ottiene imponendo il vincolo del SVP è:

$$\left(y - \frac{c}{2}\right)^2 - x^2 \left(\frac{k}{2} - 1\right) = \frac{c^2}{4} \left(\frac{k-2}{k}\right) \quad \text{per } k \geq 2$$

$$\left(y - \frac{c}{2}\right)^2 + x^2 \left(1 + \frac{c^2}{2k}\right) = \left(\frac{2k + c^2}{4}\right) \quad \text{per } k > 0$$

c rappresenta la distanza tra i due punti focali del sensore catadiottrico, cioè la distanza tra il SVP e il pinhole della telecamera; k è la costante di integrazione che determina tutte le soluzioni particolari dell'equazione differenziale.

Per valori di $k > 2$ e $c > 0$ si ottiene l'equazione di un'iperbole in forma canonica:

$$\frac{1}{a_h^2} \left(y - \frac{c}{2}\right)^2 - \frac{1}{b_h^2} x^2 = 1$$

$$\text{dove } a_h = \frac{c}{2} \sqrt{\frac{k-2}{2}} \quad \text{e } b_h = \frac{c}{2} \sqrt{\frac{2}{k}}$$

2.3.1 – Conversione da immagine omnidirezionale a panoramica

Esplicitando in funzione di z , mettendo sotto radice ambo i membri si ottengono le equazioni dei due rami dell'iperbole. Il profilo dello specchio coinciderà con il ramo positivo:

$$y_{profilo} = \frac{c}{2} + a_h \sqrt{1 + \frac{x^2}{b_h^2}}$$

I due fuochi dell'iperbole sono collocati nell'origine $O=(0,0)$ e nel punto $V=(0,c)$. L'origine coinciderà anche con il pinhole della telecamera prospettica utilizzata. V è il SVP del sistema catadiottrico, c è pertanto la distanza tra il pinhole della telecamera e il SVP.

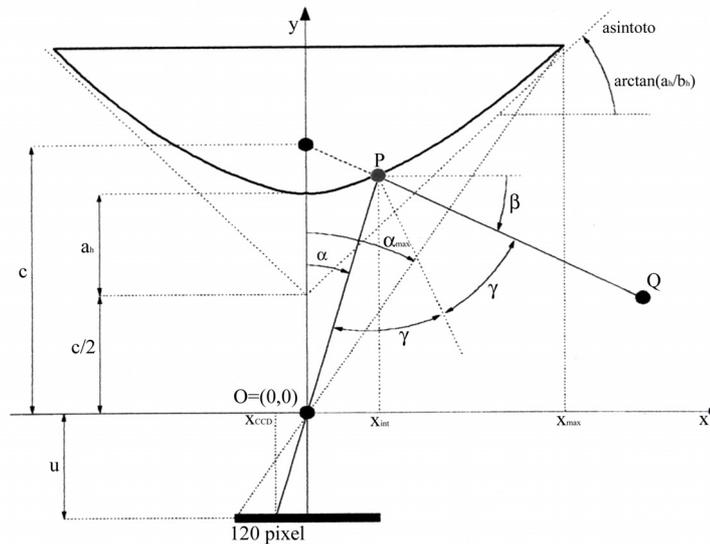


Figura 2.10: Riflessione di un punto dell'ambiente sul sensore CCD. (Immagine tratta da [18]).

Ogni circonferenza dell'immagine omnidirezionale corrisponde ad una riga nel cilindro panoramico. Abbiamo bisogno, a partire dalle due circonferenze delimitanti la ROI selezionate, di trovare a che punti nel piano dell'ambiente (Figura 2.10) corrispondono, e quindi mapparle, punto per punto, nel cilindro panoramico corrispondente.

A questo scopo considereremo il sistema in sezione, lavorando in due dimensioni: trovato un punto del CCD da mappare in un punto dell'immagine panoramica, potremo fare lo stesso per tutti i punti appartenenti alla circonferenza passante per quel punto e avente come centro quello dell'immagine omnidirezionale (e quindi del sensore CCD).

Consideriamo le rette r_1 ed r_2 passanti per l'origine (quindi per un fuoco dell'iperbole) e per i punti $(x_{CCD1}, -u)$, $(x_{CCD2}, -u)$, dove x_{CCD1} e x_{CCD2} sono i raggi delle circonferenze che delimitano, rispettivamente esternamente ed internamente, la ROI nell'immagine omnidirezionale, mentre u corrisponde alla distanza tra il pinhole della telecamera e il sensore CCD.

2.3.1 – Conversione da immagine omnidirezionale a panoramica

Quello che dobbiamo fare è:

- trovare l'intersezione tra r_1 ed r_2 e l'iperbole: i due punti trovati, $P_1=(x_{int1},y_{int1})$ e $P_2=(x_{int2},y_{int2})$ delimiteranno la superficie dello specchio in cui vengono riflessi i punti dell'ambiente che si devono considerare
- trovare i punti del piano dell'ambiente Q_1 e Q_2 che vengono riflessi nello specchio nelle posizioni corrispondenti a P_1 e P_2 : Q_1 e Q_2 delimiteranno inferiormente e superiormente il FOV del cilindro panoramico.
- dividere la parte selezionata del piano dell'ambiente in tante parti quante sono le righe che vogliamo ottenere nell'immagine corrispondente alla rappresentazione panoramica (Figura 2.11)

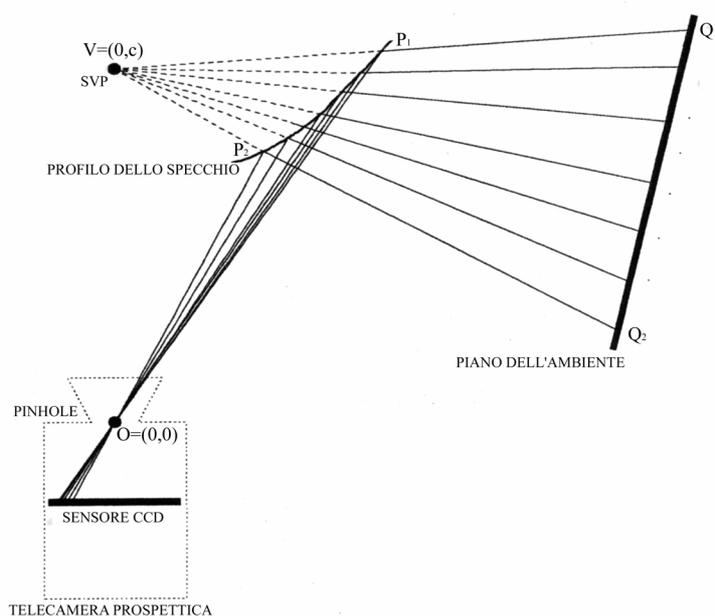


Figura 2.11: Riflessione dei punti dell'ambiente sul sensore CCD. (Immagine tratta da [18])

- a partire dai punti selezionati nel piano dell'ambiente, trovare a quali punti dello specchio iperbolico corrispondono e dove vengono riflessi sul sensore CCD (e quindi a che pixel dell'immagine omnidirezionale corrispondono). Ogni punto x_{CCD} trovato sarà compreso tra i punti x_{CCD1} e x_{CCD2} corrispondenti alle circonferenze che delimitano la regione di interesse. Per ogni valore di x_{CCD} considereremo perciò la circonferenza di raggio x_{CCD} nell'immagine omnidirezionale e mapperemo i punti che la compongono in una riga del cilindro panoramico.

Tutti i parametri di cui avremo bisogno per effettuare la conversione sono:

- la distanza tra il punto focale dello specchio e la telecamera ($c, >0$)
- la costante di curvatura dello specchio ($k, >2$)
- la distanza tra il pinhole della telecamera e il sensore CCD (u)

2.3.1 – Conversione da immagine omnidirezionale a panoramica

- il centro dell'immagine omnidirezionale ($x_{\text{centro}}, y_{\text{centro}}$)
- i raggi delle circonferenze che delimitano la regione di interesse ($x_{\text{CCD1}}; x_{\text{CCD2}}$).

I primi tre parametri dipendono dallo specchio e dalla telecamera utilizzati. Il centro dell'immagine omnidirezionale si ottiene con la procedura spiegata nel paragrafo 2.3, mentre la regione di interesse dipende dal FOV che si vuole ottenere.

Capitolo 3

Individuazione dei requisiti e Risultati sperimentali

Nel capitolo precedente abbiamo descritto la struttura del nostro DVS, la distribuzione degli OVA e il percorso del robot. In questo capitolo vedremo i risultati dei test eseguiti sugli insiemi di immagini a disposizione, analizzeremo i problemi incontrati e le soluzioni trovate, valutandone l'efficacia. Compareremo inoltre le prestazioni del DVS con quelle di un sistema statico che utilizza lo stesso metodo di confronto delle immagini, dimostrando che l'approccio distribuito fornisce risultati migliori in ambienti dinamici.

Abbiamo iniziato la sperimentazione con un solo set di immagini: dato che queste avevano caratteristiche tali da rendere problematico il confronto, abbiamo cercato delle soluzioni a livello software che permettessero rendere efficiente la localizzazione anche in presenza dei problemi riscontrati, come la differenza di luminosità o la presenza di ostacoli visivi. Nonostante si siano ottenuti dei miglioramenti, i risultati non sono stati abbastanza soddisfacenti: in seguito abbiamo perciò avuto l'esigenza di ripetere la fase di acquisizione delle immagini con alcuni accorgimenti che ci permettessero di rendere più efficace il riconoscimento.

Nei prossimi due paragrafi analizzeremo i problemi affrontati con i primi due set di immagini e i miglioramenti ottenuti con le soluzioni proposte. Ricaveremo i requisiti per l'acquisizione delle immagini, individuati tramite le prove, tali da permettere un confronto efficiente. Procederemo poi valutando i risultati con il set di immagini ottimali, il terzo, che utilizzeremo anche nella parte finale per il confronto tra DVS e sistema statico. Dimostreremo infine sperimentalmente la proprietà di invarianza prospettica.

3.1 *Problemi incontrati*

Il primo insieme di immagini acquisite nell'ambiente di prova hanno permesso di mettere in luce i problemi che possono affliggere il DVS.

Il primo set è composto da:

- 86 immagini prese dal robot in 68 posizioni diverse; per alcune posizioni sono state prese più immagini, variando l'ambiente circostante in modo da verificare l'efficacia del DVS in un ambiente dinamico
- 86x24 immagini dell'ambiente, ossia le immagini prese da tutti gli OVA per ogni posizione del robot

Una prima caratteristica che si può notare osservando le immagini appartenenti a questo set sta nella diversità tra quelle prese dal robot e quelle degli OVA:

- lo specchio montato sul robot è diverso da quelli montati sugli OVA: ha una superficie diversa, le immagini appaiono più scure e tendono verso il giallo, sono diversi i parametri di curvatura e distanza tra i fuochi.
- vi è una notevole diversità prospettica tra le immagini acquisite dagli OVA e quelle del robot: il primo, anche in vicinanza ad un sensore dell'ambiente, riesce a vedere parti diverse della scena a causa di ostacoli visivi.

Nel primo caso il problema consiste nel fatto che, dato che l'immagine del robot è caratterizzata da una luminosità minore rispetto alle altre, la prima componente ricavata dal calcolo della Fourier signature avrà un valore di molto diverso da quello calcolato sulle immagini prese dagli OVA, anche per quelli collocati in posizioni molto vicine.

Infatti, quando calcoliamo la trasformata di Fourier di un segnale di luminosità, come una riga del cilindro panoramico, decomponiamo il segnale nelle sue componenti rispetto ad una base di funzioni; queste funzioni sono legate alla variazione spaziale di luminosità, cioè ai "pattern di luminosità".

La prima funzione base che corrisponde alla frequenza zero è il segnale di luminosità costante e il relativo coefficiente esprime il livello medio di luminosità del segnale sull'intera immagine. Le stesse considerazioni valgono per frequenze più elevate.

Come si vede dalle equazioni di (1.2) e (1.3), quando calcoliamo la similarità tra due immagini sommiamo i contributi a diverse frequenze, ma osserviamo che il contributo maggiore viene dato dalle basse frequenze (come si può vedere dalla Figura 1.6). Vi è il rischio perciò che il confronto vada ad indicare come immagine di riferimento corrispondente alla posizione del robot non quella con le caratteristiche più simili all'immagine di input, ma quella a luminosità minore tra tutte quelle acquisite dagli OVA. Inoltre, la differenza di contrasto porta ad una variazione delle componenti a frequenze più alte: essendo le linee dell'ambiente più marcate nell'immagine del robot rispetto a quelle degli OVA, per la prima le variazioni di luminosità, e quindi le

componenti della Trasformata di Fourier, saranno maggiori rispetto a quelle delle immagini di riferimento.

Anche il secondo punto comporta non pochi problemi: consideriamo ad esempio, nelle immagini dei gruppi 4 e 6 corrispondenti al tratto del corridoio: in questa parte dell'ambiente gli OVA sono montati sopra dei muri che fungono da separatori tra le varie zone del laboratorio (come si vede in Figura 3.1) e che hanno un'altezza di poco inferiore a quella a cui è collocato il sensore del robot. In questo modo il robot, che si trova a circa un metro dall'OVA, riesce a vedere il muro, a differenza dell'OVA più vicino che vede invece ciò che sta al di là del muro. Questo crea notevoli problemi nel riconoscimento delle immagini con l'approccio da noi scelto: infatti, calcolando la trasformata di Fourier per ogni riga dell'immagine e poi sottraendo i valori dei coefficienti ottenuti tra un'immagine e l'altra, in presenza di grandi diversità (come può essere proprio un muro), sarà difficile che il match vada a buon fine.



Figura 3.1: Cilindro panoramico corrispondente all'immagine acquisita dal robot in prossimità dell'OVA 6_2 (all'interno del cerchio rosso).



Figura 3.2: Cilindro panoramico corrispondente all'immagine acquisita dall'OVA 6_2.

Nelle Figure 3.1 e 3.2 si vedono i cilindri panoramici ricavati dall'immagine del robot al centro del corridoio e da quella dell'OVA più vicino (il 6_2). Come si può osservare, le differenze tra i due sono notevoli: oltre alla differenza di luminosità e contrasto, il robot dalla sua posizione è in grado di vedere parti diverse dell'ambiente (in Figura 3.1 si vede il muro su cui è posto l'OVA, mentre in Figura 3.2 questo non è presente, si vede invece la sfera blu presente nel laboratorio, che nell'immagine del robot si nota appena).

Da un primo confronto otteniamo delle conferme ai problemi appena citati: il riconoscimento fallisce per quasi tutte le posizioni del robot, eccezione fatta per quelle in cui si trova vicino all'OVA 6_3. E' da notare però che, anche per immagini acquisite in altre posizioni, spesso il calcolo della similarità indica come più vicino sempre l'OVA 6_3: un motivo può essere cercato nel fatto che le caratteristiche (luminosità e variazioni di luminosità) dell'ambiente attorno a questo sensore fanno sì che le immagini risultanti dal 6_3 siano più simili a quella del robot rispetto alle altre immagini di riferimento, indipendentemente da ciò che rappresentano.

3.2 *Analisi delle soluzioni e dei requisiti*

Per tentare di risolvere il primo punto, abbiamo modificato l'algoritmo di calcolo della similarità escludendo la componente a frequenza zero della FFT: quest'ultima infatti corrisponde al segnale di luminosità costante.

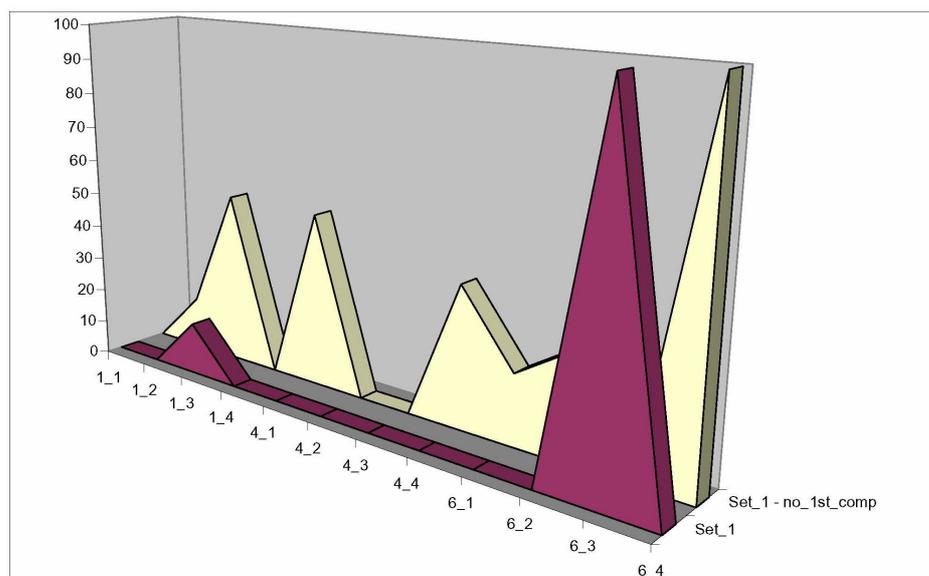


Figura 3.3: Percentuale di match corretto per ogni OVA di interesse

Il grafico in Figura 3.3 mostra la percentuale di immagini riconosciute correttamente per alcuni (i più rappresentativi) OVA nel caso in cui venga utilizzata la prima componente della FFT e nel caso in cui non sia considerata: vediamo che, a parte l'OVA 6_3 (ulteriore dimostrazione dell'ipotesi esaminata nel paragrafo precedente), l'eliminazione della prima componente porta risultati migliori, ma non ancora abbastanza soddisfacenti.

Per quanto riguarda i problemi di diversità di parametri dello specchio e di diversità prospettica, abbiamo modificato l'algoritmo di conversione da immagine omnidirezionale a panoramica, in modo da poter utilizzare per ogni OVA dei parametri diversi e poter selezionare diverse aree di visione (FOV, Field Of View) (Paragrafo 2.3.1).

Purtroppo con l'insieme di immagini considerato non è possibile trovare un FOV comune a tutti gli OVA per vari motivi, quali la distanza variabile tra il robot e gli OVA, la differenza di altezza tra la videocamera del robot e gli OVA, la differenza di altezza tra gli OVA stessi.

Abbiamo perciò deciso di ripetere la fase di acquisizione delle immagini, prendendo degli accorgimenti che ci fornissero un data set meno patologico.

3.2 – Analisi delle soluzioni e dei requisiti

Abbiamo considerato due soluzioni ai problemi incontrati con il primo set di immagini:

- dato che l'ambiente scelto non è ottimale per dei test su questo tipo di sistema a causa degli spazi ristretti, una soluzione è l'acquisizione di immagini in un ambiente diverso, con gli OVA montati su supporti che non ostacolino la visione del robot.
- la seconda soluzione, più rapida ma meno efficiente, consiste nel ripetere l'acquisizione nello stesso ambiente, sostituendo il sensore del robot con uno più simile a quelli degli OVA e prendendo le immagini con il robot più vicino agli OVA, in modo da non avere ostacoli visivi.

La soluzione per cui abbiamo optato è la seconda, in quanto anche se la prima sarebbe stata più efficace, il tempo e i mezzi a disposizione non erano sufficienti.

Abbiamo sostituito lo specchio del robot con quello dell'OVA 5_2, in modo da avere immagini simili tra robot e OVA.

Il secondo set è composto da:

- 82 immagini prese dal robot in 77 posizioni diverse; per alcune posizioni sono state prese più immagini variando l'ambiente circostante in modo da verificare l'efficacia del DVS in un ambiente dinamico
- 82x24 immagini dell'ambiente, ossia le immagini prese da tutti gli OVA per ogni posizione del robot

Le immagini del nuovo set consentono di considerare un FOV comune tra il robot e gran parte degli OVA di interesse, eccezione fatta per gli 1_1, 1_2, 1_3, 1_4 che non abbiamo potuto spostare, essendo fissati in modo permanente al tavolo: questi ultimi si trovano ad un'altezza considerevolmente diversa (più in basso) rispetto al sensore del robot, per cui dobbiamo aspettarci risultati peggiori, nel riconoscimento, rispetto agli altri OVA.

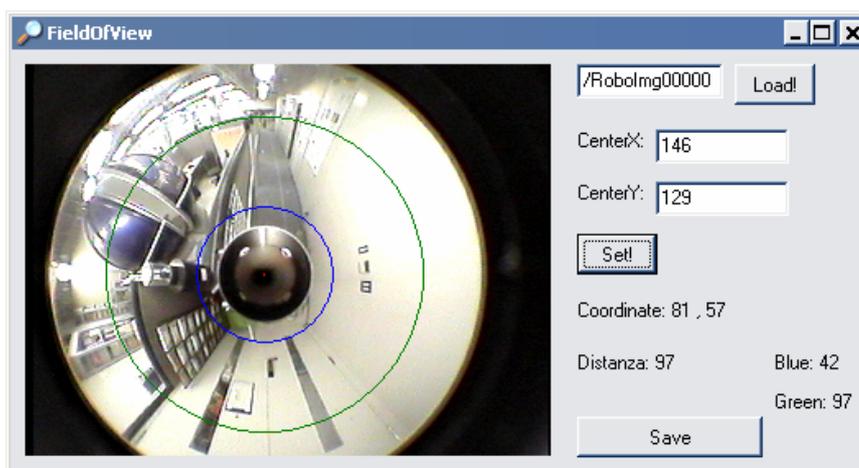


Figura 3.4: Programma realizzato per la determinazione del FOV ottimale

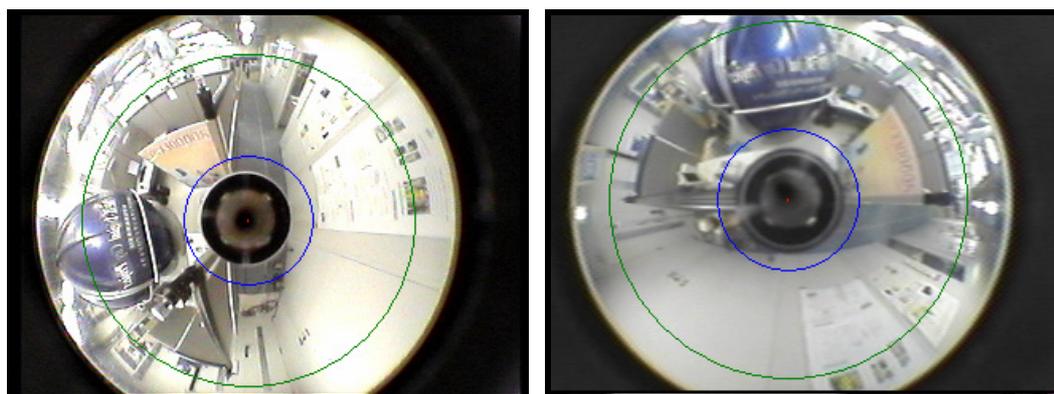


Figure 3.5 e 3.6: Immagini omnidirezionali rispettivamente del robot e dell'OVA 6_3. Le circonferenze segnate in verde e blu delimitano l'area considerata per la fase di riconoscimento.

Nelle Figure possiamo vedere le immagini omnidirezionali catturate dal sensore del Robot e dell'OVA 6_3 e i rispettivi FOV. La circonferenza di colore blu delimita internamente l'area considerata per il confronto, mentre la circonferenza di colore verde la delimita esternamente.

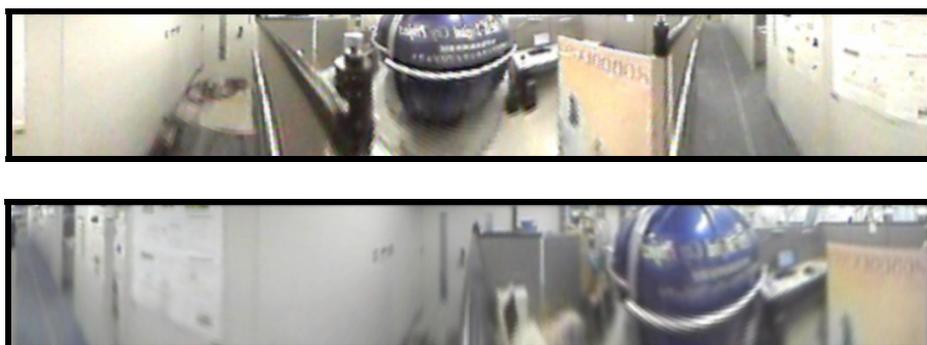


Figure 3.7 e 3.8: Cilindri panoramici corrispondenti alle immagini acquisite dal robot e dall'OVA 6_3 rispettivamente.

Nelle Figure 3.7 e 3.8 sono presentati i cilindri panoramici ottenuti dalle immagini omnidirezionali precedenti. Da una prima analisi possiamo notare alcune differenze:

- vi è ancora una differenza di prospettiva tra l'immagine del robot e quella dell'OVA; questa è comunque molto ridotta rispetto a quella caratteristica delle immagini del primo set.
- l'immagine del robot è più nitida e i colori sono ad una temperatura di colore minore rispetto a quelle degli OVA: questo perché, mentre sulle immagini acquisite dai sensori nell'ambiente vengono applicati dei filtri da parte dei Color Quad Processors, quella del robot non subisce alcuna elaborazione.

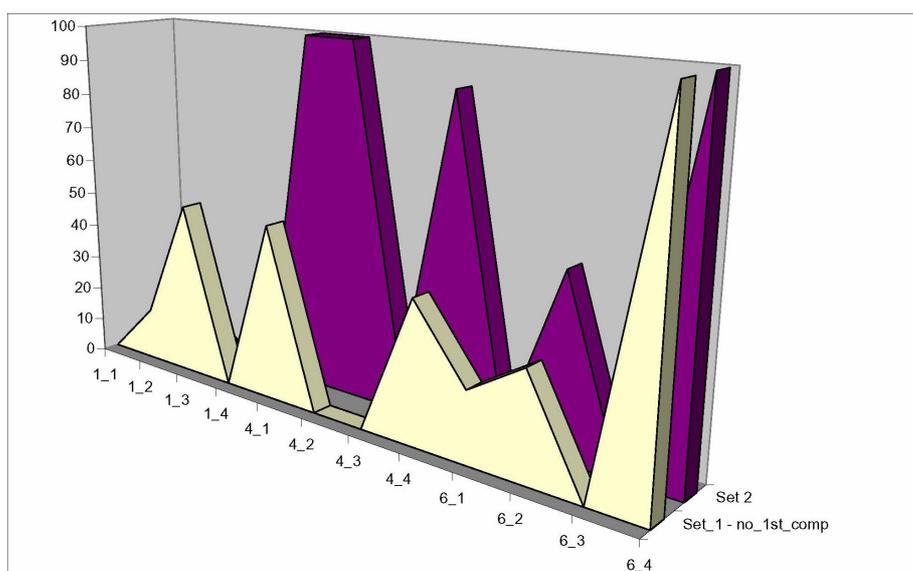


Figura 3.9: Confronto tra le percentuali di match per ogni OVA di interesse del primo e del secondo set di immagini

In Figura 3.9 possiamo vedere il risultato del calcolo di similarità sulle immagini del nuovo set, a confronto con i migliori risultati ottenuti in precedenza: notiamo un netto miglioramento per gli OVA posti lungo il corridoio, a parte l’OVA 6_1, per il quale si ha un peggioramento nel riconoscimento, e i 6_3 e 4_3, per i quali la situazione resta invariata.

I risultati non sono comunque ancora sufficienti: nonostante il match avvenga correttamente quando il robot si trova in posizioni molto vicine agli OVA, non abbiamo ancora risultati buoni per le posizioni intermedie (tra un OVA e l’altro) del robot.

Un motivo del fallimento nel calcolo esatto della similarità può essere ancora dovuto alla diversità tra le immagini del robot e quelle degli OVA: abbiamo perciò deciso di ripetere una terza volta la fase di acquisizione, elaborando anche l’immagine del robot tramite i Color Quad Processor. A questo scopo abbiamo rimosso l’OVA 5_2, che non forniva informazioni utili in quanto molto lontano dal percorso del robot, potendo così liberare una porta di ingresso del Quad Processor numero 5: a quest’ultimo abbiamo affidato l’elaborazione dell’immagine acquisita dal robot.

Inoltre, per alcune posizioni (come vicino agli OVA 4_3, 6_3 e 6_1) il sensore del robot non si trova abbastanza in prossimità dei sensori nell’ambiente: questo causa il peggioramento o comunque il mancato incremento di prestazioni nel riconoscimento per certe posizioni.

Nel prossimo paragrafo effettueremo perciò le prove sulla localizzazione su un terzo set di immagini, acquisite tenendo conto, quando possibile, dei requisiti fino ad ora emersi:

- il robot deve trovarsi il più vicino possibile agli OVA;
- gli OVA non devono essere collocati sopra a possibili ostacoli alla visuale del robot;
- i sensori del robot e dell’ambiente devono essere dello stesso tipo;

- le immagini acquisite dai sensori, se elaborate, devono tutte essere soggette alla stessa elaborazione;
- i sensori devono essere collocati ad altezze simili.

3.3 Immagini ottimali

Il terzo set è composto da:

- 102 immagini prese dal robot in 70 posizioni diverse: per alcune posizioni sono state prese più immagini, variando l'ambiente circostante in modo da verificare l'efficacia del DVS in un ambiente dinamico
- 102x24 immagini dell'ambiente, ossia le immagini prese da tutti gli OVA per ogni posizione del robot

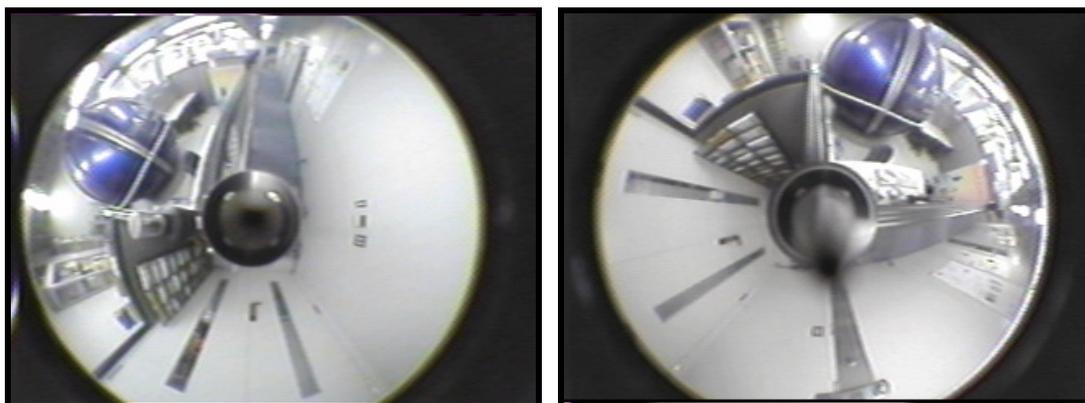


Figure 3.10 e 3.11: Immagini omnidirezionali rispettivamente del robot e dell'OVA 6_1.

Questa volta, come si può vedere nelle Figure 3.10 e 3.11, le immagini catturate dal robot e dall'OVA corrispondente sono molto simili: questo favorisce notevolmente un calcolo corretto di similarità, come si nota dal grafico in Figura 3.12, dove vengono confrontate le percentuali di successo fra i tre set utilizzati.

Data la somiglianza tra le immagini, non abbiamo più bisogno di eliminare la prima componente della FFT, come avevamo fatto in precedenza a causa della diversità di luminosità: in questo caso tale accorgimento, oltre a non fornire risultati migliori, porta in alcuni casi ad un peggioramento nelle prestazioni.

Se le immagini sono acquisite correttamente, la luminosità media dell'immagine assume un'importanza rilevante: dato che per ottenere una localizzazione affidabile il robot deve trovarsi molto vicino agli OVA, è molto difficile che in poco spazio la quantità di luce vari considerevolmente. Infatti, la luminosità media delle immagini varia molto

lentamente all'aumentare della distanza tra le stesse; invece la distribuzione e la presenza dei pattern di luminosità, che rappresentano gli oggetti dell'ambiente, variano più velocemente. Perciò prendere in considerazione le componenti alle basse frequenze delle trasformate ci permette di avere una distinzione maggiore tra zone dell'ambiente a luminosità diversa.

Nel caso del terzo insieme di immagini vi è un notevole miglioramento di prestazioni nella localizzazione rispetto ai primi due: per gli OVA lungo il corridoio la percentuale di match è prossima al 100% (a parte per l'OVA 6_3), mentre per quelli attorno al tavolo solo per l'1_3 abbiamo un 70% di casi in cui la posizione del robot viene riconosciuta. Per gli altri non abbiamo buoni risultati, ma questo è dovuto alla loro disposizione non ottimale. Comunque i risultati ottenuti nel primo settore ci permettono di verificare l'efficacia del DVS a confronto con il sistema statico.

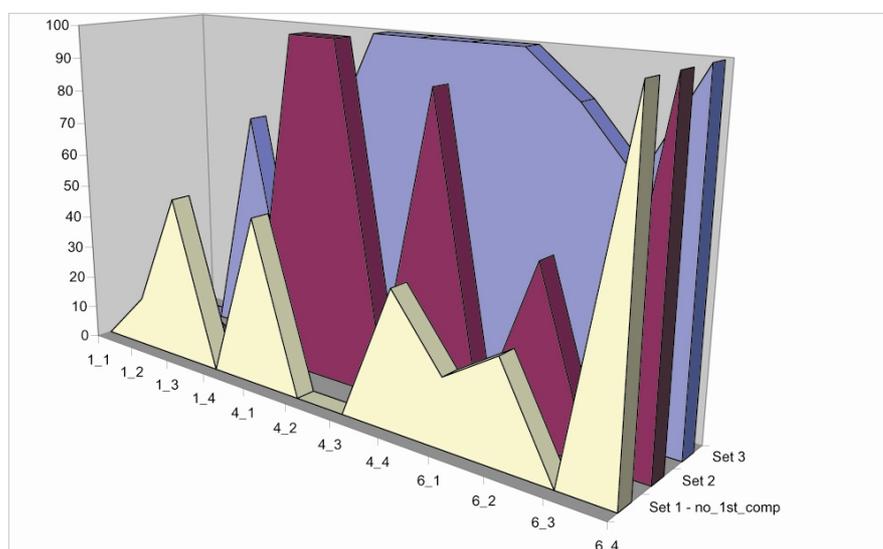


Figura 3.12: Confronto tra le percentuali di match per ogni OVA di interesse del primo, del secondo e del terzo set di immagini

Durante l'acquisizione del terzo set abbiamo, per alcune posizioni del robot, preso più di un'immagine in condizioni ambientali diverse da quelle normali.

Le posizioni interessate sono:

- vicino all'OVA 6_2, una con persone attorno al robot e tre con luci spente
- vicino all'OVA 4_4, una con persone attorno al robot e tre con luci spente
- vicino all'OVA 4_1, una con persone attorno al robot e tre con luci spente
- vicino agli OVA 1_3, 1_2, 1_1, 1_4: scarteremo queste posizioni in quanto per esse la localizzazione fallisce, a causa della differenza di altezza degli OVA rispetto al sensore del robot.

Come vedremo nel paragrafo successivo, queste immagini ci sono utili per verificare le prestazioni del DVS.

3.4 Confronto tra sistema statico e dinamico

In questa sezione compareremo le prestazioni del DVS con un sistema di localizzazione statico, del tipo descritto in [1]. In quest'ultimo le immagini di riferimento non sono acquisite da sensori distribuiti nell'ambiente: vengono utilizzate, invece, immagini precedentemente acquisite solamente dal robot ed immagazzinate nella sua memoria. Il confronto avviene quindi tra ciò che il robot vede nella posizione corrente e ciò che ha già visto precedentemente.

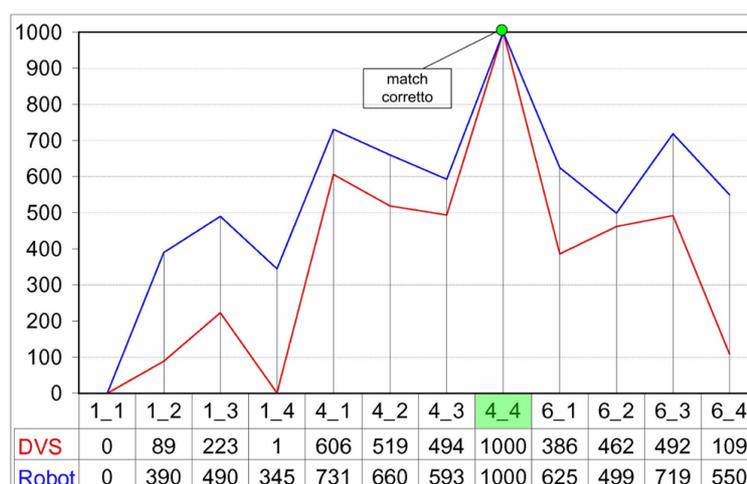


Figura 3.13: Confronto di risultati con l'utilizzo di un sistema statico (in blu) e del DVS (in rosso) in condizioni statiche, senza variazioni ambientali, con il robot vicino all'OVA 4_4.

Nel primo grafico in Figura 3.13 vediamo il risultato della localizzazione, effettuata in condizioni statiche (luminosità costante e nessun elemento dinamico di disturbo) con il robot posizionato vicino all'OVA 4_4. Sulle colonne ci sono, nel caso dell'approccio con DVS (in rosso), i numeri corrispondenti agli OVA di riferimento utilizzati, mentre le righe corrispondono al valore di similarità corrispondente al confronto tra l'immagine acquisita dal robot e gli OVA.

Nel caso dell'approccio statico (in blu), invece, al posto delle immagini di riferimento degli OVA ci sono quelle acquisite dal robot posizionato nelle vicinanze degli OVA, sempre in condizioni statiche: il risultato del confronto ci fornirà comunque la posizione del robot in relazione agli OVA, permettendo un confronto obiettivo.

Nei due casi il valore di similarità maggiore ci fornirà perciò la posizione del robot.

Notiamo che il riconoscimento avviene correttamente con entrambi gli approcci.

3.4 – Confronto tra sistema statico e dinamico

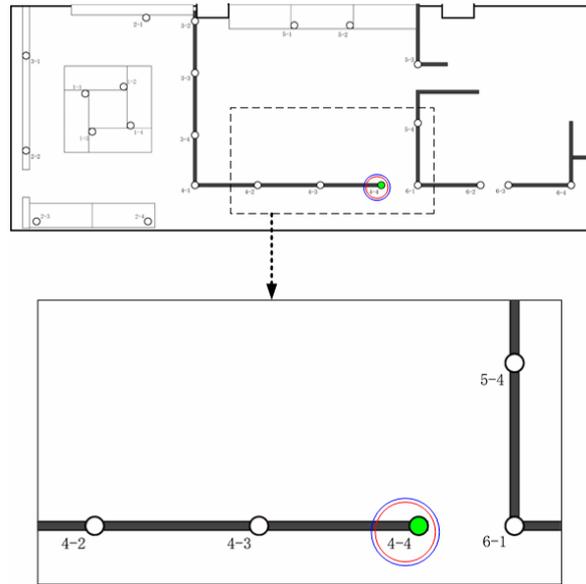


Figura 3.14: Localizzazione corrispondente ai risultati di similarità del grafico in Figura 3.13

Per l'approccio statico la situazione cambia se confrontiamo una diversa immagine del robot, acquisita nella stessa posizione ma questa volta a luci spente, con le stesse immagini prese in precedenza.

In questo caso il riconoscimento fallisce (Figura 3.15), in quanto la localizzazione indica come posizione corretta quella corrispondente alle immagini prese precedentemente dal robot vicino all'OVA 4_3: dovrebbe essere invece, come prima, il 4_4, che in questa situazione va a corrispondere al sesto valore.

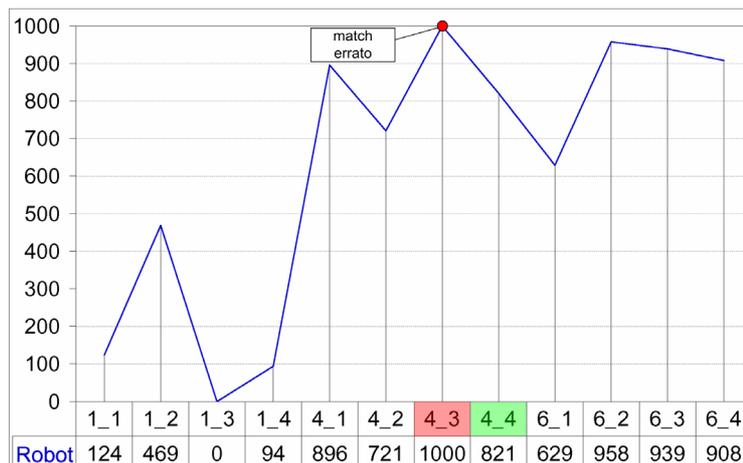


Figura 3.15: Risultati del confronto tra le immagini immagazzinate dal robot (per posizioni vicine agli OVA) in un sistema statico e quella corrente catturata con le luci spente.

Con l'approccio dinamico permesso dal DVS, invece, la localizzazione avviene correttamente anche in caso di variazioni ambientali, come possiamo vedere nel grafico

3.4 – Confronto tra sistema statico e dinamico

in Figura 3.16: qui il confronto viene effettuato con immagini dell'ambiente prese tutte nello stesso istante e quindi nelle stesse condizioni. In questo caso, ad una minore luminosità nell'immagine catturata dal robot corrisponde una minore luminosità anche in quella acquisita dal sensore più vicino.

La posizione riconosciuta corrisponde all' OVA 4_4, come ci aspettavamo.

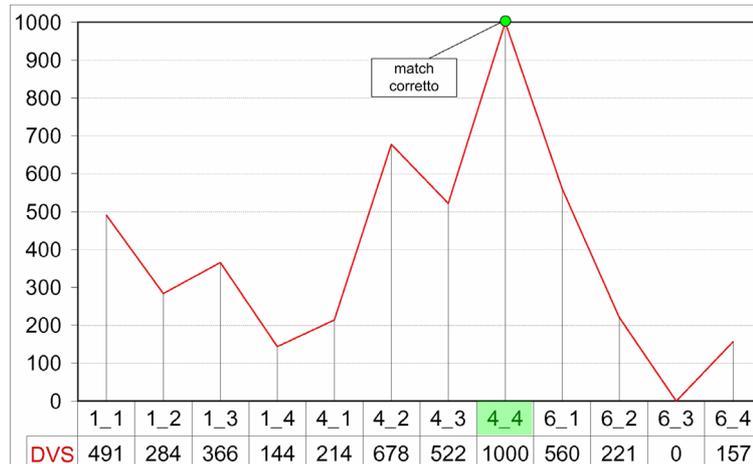


Figura 3.16: Risultati del confronto tra le immagini acquisite dagli OVA e quella del Robot vicino all'OVA 4_4 in condizioni di scarsa luminosità.

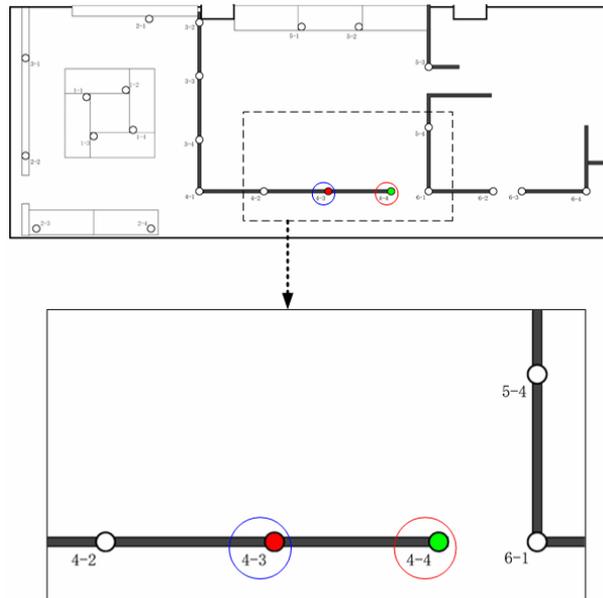


Figura 3.17: Localizzazione corrispondente ai risultati di similarità dei grafici nelle Figure 3.15 e 3.16.

In Figura 3.18 presentiamo un altro esempio di come il match possa fallire nel caso di un confronto con immagini di riferimento statiche: in questo caso il robot si trova in prossimità dell'OVA 6_2. Come nell'esempio precedente, prima eseguiamo il confronto

3.4 – Confronto tra sistema statico e dinamico

con il sistema statico: se l'immagine del robot viene catturata nelle stesse condizioni in cui si era acquisita l'immagine di riferimento, la localizzazione riesce, mentre se l'immagine corrente cambia, la posizione del robot non viene riconosciuta correttamente. In seguito ripetiamo il confronto tra l'immagine del robot, presa nella stessa posizione ma con l'introduzione di persone nell'ambiente, e le immagini dell'ambiente prese dal DVS: a differenza del sistema statico, il DVS riesce ad effettuare la localizzazione.

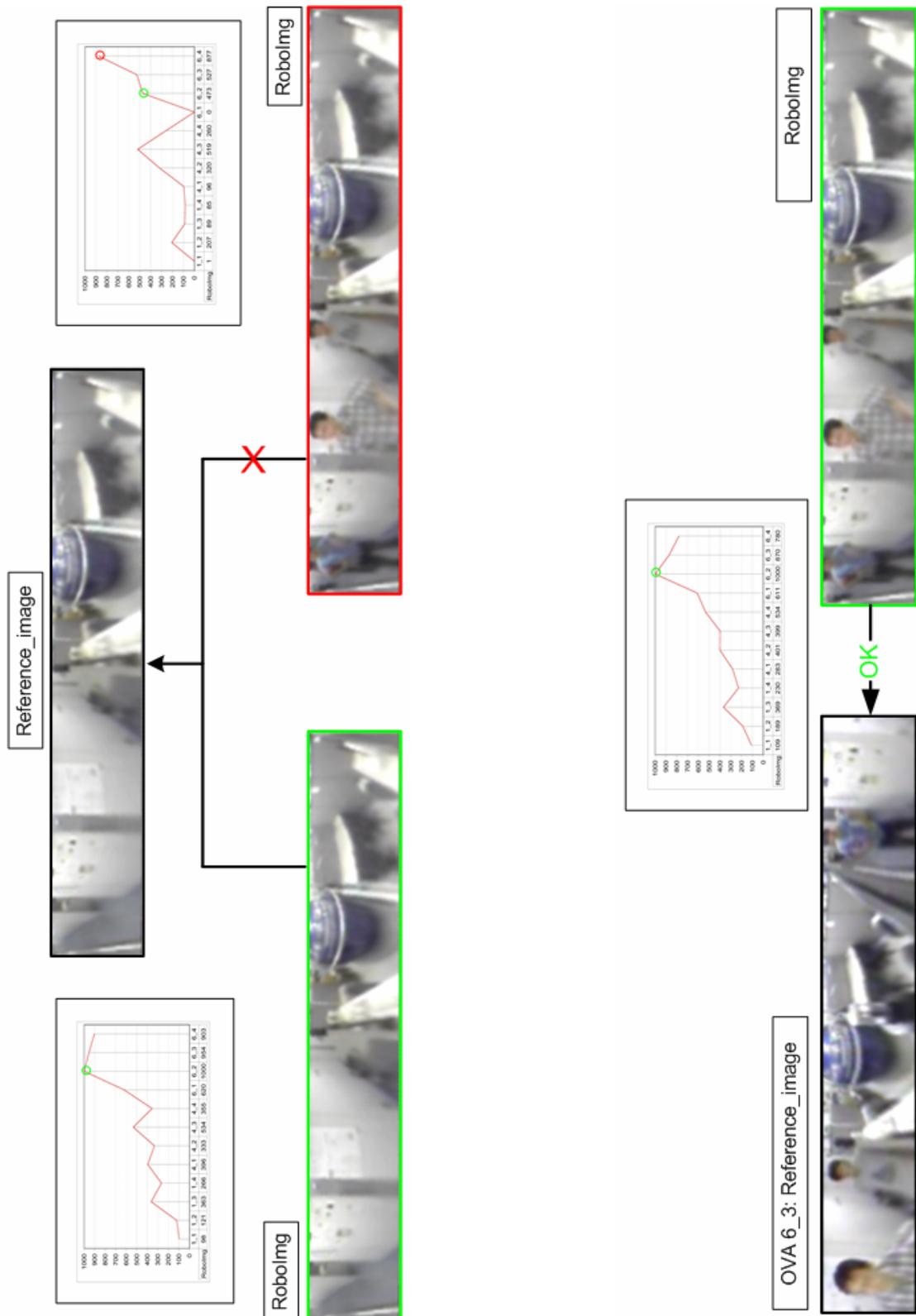


Figure 3.18 e 3.19: Localizzazione con l'uso di un sistema statico e con il DVS.

3.5 Invarianza prospettica

Vi è un'interessante proprietà che caratterizza il nostro DVS: immaginiamo che, nella fase di acquisizione, un elemento dinamico, supponiamo ad esempio una persona, si trovi tra il sensore dell'ambiente e quello del robot. In una situazione del genere il robot vedrà la persona in una certa posizione, mentre l'OVA più vicino la vedrà collocata diversamente a causa della differenza di prospettiva nella visione. Questa situazione è riportata in Figura 3.20.

Nel caso in cui l'elemento dinamico non vada a coprire zone in cui vi siano variazioni orizzontali particolarmente marcate, il sistema è comunque in grado di riconoscere le due immagini: questa proprietà prende il nome di invarianza prospettica.

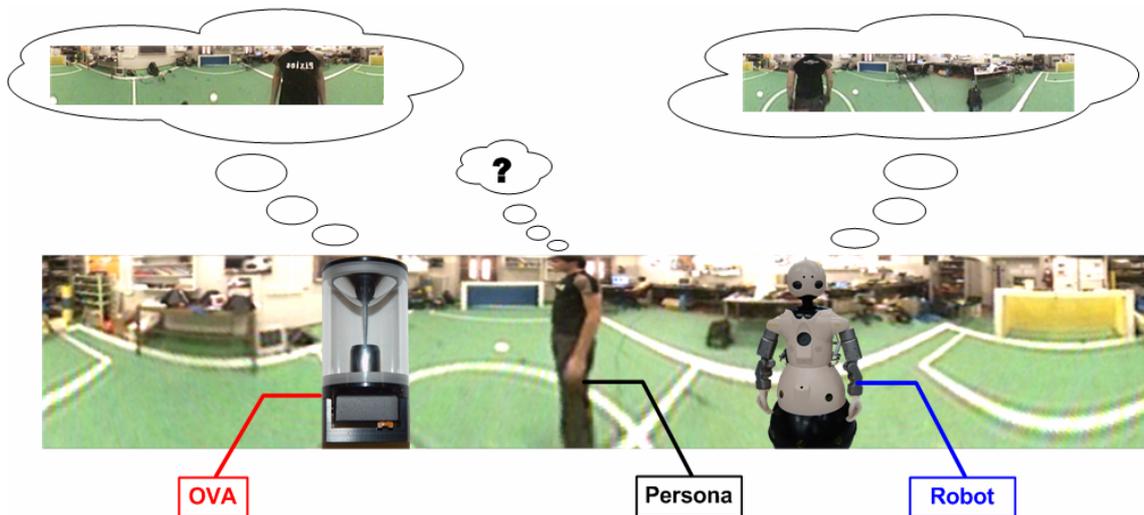


Figura 3.20: Situazione in cui si verifica una differenza di prospettiva tra l'immagine dell'OVA e quella del Robot

Come sappiamo, la trasformata di Fourier applicata ad un'immagine è sensibile alle variazioni di luminosità. Dato che per il calcolo della Fourier signature calcoliamo la trasformata solo alle righe del cilindro panoramico, la distribuzione delle occlusioni in senso orizzontale non viene considerata: se una persona od un oggetto vengono traslati rispetto allo sfondo potrà cambiare la fase delle componenti della Trasformata corrispondenti ad alcuni pattern di luminosità, ma non l'ampiezza, a patto che non vengano coperte parti che danno un forte contributo ai valori dei coefficienti della signature.

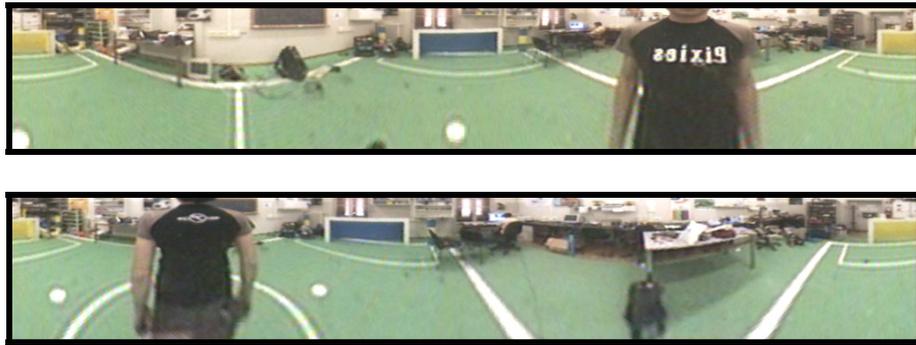


Figure 3.21 e 3.22: Viste dell'OVA e del robot rispettivamente, nella situazione di Figura 3.20

Per poter verificare questa proprietà, abbiamo acquisito una serie di immagini nella situazione appena descritta. Abbiamo disposto dei sensori nell'ambiente e interposto una persona tra un OVA e il sensore del robot, come si vede dalle figure 3.21 e 3.22: la prima corrisponde alla vista dell'OVA, mentre la seconda a quella del robot.

La differenza di prospettiva è evidente: mentre l'OVA vede la persona di fronte e a destra a coprire una certa zona dell'ambiente (un tavolo, come si vede dalla Figura 3.23), per il robot questa è di spalle. Inoltre il robot si vede coprire una zona diversa da quella non visibile nel caso del punto di vista dell'OVA.

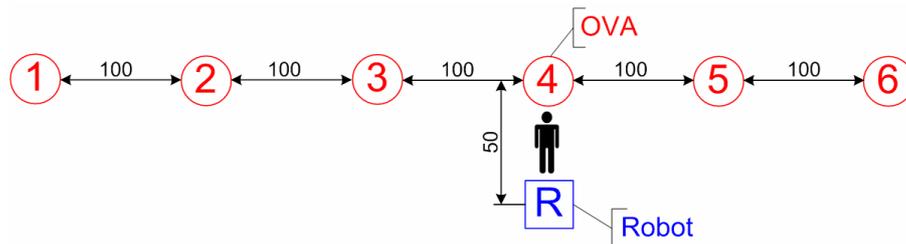


Figura 3.23: posizione dei sensori dell'ambiente, del robot e della persona nella prova sull'invarianza prospettica.

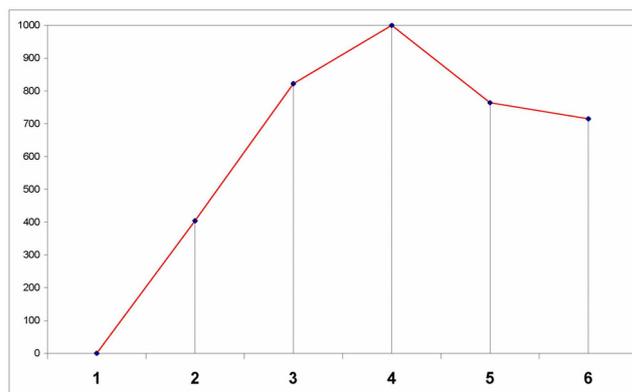


Figura 3.24: valori di similarità corrispondenti al confronto tra le immagini acquisite dai sensori ambientali e il robot con la disposizione di Figura 3.23.

In Figura 3.24 si vede il risultato del calcolo di similarità tra l'immagine acquisita dal robot e quella dei sei sensori disposti lungo l'ambiente (la disposizione è mostrata in Figura 3.23): la localizzazione avviene correttamente, nonostante le diversità appena descritte tra l'immagine del robot e quella di riferimento corrispondente (acquisita dall'OVA 4).

3.6 Conclusioni sui risultati sperimentali

Abbiamo visto in questo capitolo come, rispettando gli opportuni requisiti di posizionamento degli OVA e di elaborazione delle immagini, sia possibile, con l'utilizzo del DVS, ottenere una corretta localizzazione sia in condizioni statiche, sia con elementi dinamici di disturbo nell'ambiente. Inoltre abbiamo presentato delle soluzioni, rivelatesi efficaci, per superare le difficoltà che si possono presentare nel caso in cui vi siano diversità tra le immagini acquisite dai sensori.

Tramite il paragone con un sistema di tipo classico, basato su un confronto con immagini di riferimento statiche, abbiamo sottolineato l'efficacia dell'approccio proposto.

Va preso in considerazione, però, il fatto che il sistema fallisce in alcune condizioni, legate al posizionamento degli OVA. Questo problema può essere risolto con una disposizione ottimale dei sensori, ma va tenuto conto che in un ambiente vi possono essere zone tali da non permettere una soluzione di questo tipo. Un esempio è proprio quello del laboratorio dove gli esperimenti sono stati svolti. I problemi maggiori si sono verificati con gli OVA del gruppo 1, che non abbiamo potuto collocare ad un'altezza adeguata rispetto al sensore del robot: in questo caso il sistema non si è dimostrato abbastanza robusto e raramente abbiamo ottenuto un riconoscimento corretto.

Per superare questo tipo di difficoltà abbiamo cercato un modo che permettesse di restringere il numero di immagini di riferimento da confrontare con l'immagine del robot, effettuando una selezione tra queste. Abbiamo, in altre parole, deciso di effettuare un clustering ("raggruppamento") su tutte le immagini acquisite dagli OVA, in modo che queste, catturate in ogni istante da tutti i sensori, vengano distinte in base a determinate caratteristiche, tali da non essere prese in considerazione dal sistema fino a qui illustrato. Individuando a quale gruppo di immagini, aventi proprietà simili, corrisponde quella del robot presa in una certa posizione, è poi possibile applicare l'algoritmo di similarità basato sulla Fourier signature, utilizzando come immagini di riferimento solo quelle appartenenti al sottoinsieme selezionato. La caratteristica sulla quale abbiamo deciso di basarci è il colore: questo infatti non viene considerato nella rappresentazione tramite Fourier signature ed è un efficace segno distintivo per zone diverse di un ambiente. Purtroppo il tempo a disposizione non è stato sufficiente ad eseguire le prove necessarie e a portare a termine il lavoro, presenteremo comunque quanto finora svolto a riguardo nel Capitolo successivo.

Capitolo 4

Future Works - Clustering delle immagini

In alcuni casi, ad esempio quando il robot non è sufficientemente vicino ad un OVA, il DVS non riesce a riconoscere correttamente la posizione del robot. Dato che la tecnica di confronto che utilizziamo fornisce sempre un risultato, nel caso in cui non venga riconosciuta la posizione corretta, il sistema fornirà una posizione sbagliata per il robot, credendo che si trovi in una locazione anche totalmente diversa da quella effettiva.

Spesso il motivo del fallimento nella localizzazione sta nel fatto che esistono nell'ambiente immagini di riferimento che, anche se apparentemente del tutto diverse da quella di input, hanno caratteristiche tali che il calcolo di similarità, come presentato nel capitolo 1, fornisce un valore maggiore per queste rispetto ad immagini più vicine. Infatti l'applicazione della FFT solo sulle righe dell'immagine, sebbene ci permetta di ottenere invarianza rotazionale, rappresenta in alcuni casi una limitazione: quando la prospettiva di visione è troppo diversa o l'immagine del robot non è acquisita abbastanza in prossimità dell'OVA, i coefficienti della FFT delle stesse righe delle due immagini possono non essere simili. Questo è vero in particolare quando si hanno nell'immagine oggetti con forme particolari o caratterizzati da poca altezza e molta larghezza.

Una tecnica che può permetterci di superare questo limite consiste nel considerare attributi dell'immagine diversi da quelli già utilizzati (diversi cioè dalla variazione di luminosità) per irrobustire il metodo di localizzazione.

Quello che vogliamo è poter applicare l'algoritmo di similarità non su tutte le immagini del DVS, ma su un sottoinsieme selezionato in base a determinate caratteristiche. Abbiamo perciò bisogno di un metodo che ci consenta, a partire dalle immagini a disposizione, di raggruppare tutti gli OVA del DVS in sottoinsiemi. Tale metodo deve inoltre tenere conto del fatto che stiamo lavorando in condizioni di invarianza rotazionale: non possiamo dunque considerare la distribuzione spaziale degli elementi nell'immagine.

Osserviamo la Figura 5.1, in cui sono rappresentati i colori caratteristici, nel piano HS (Hue – Saturation) per quattro immagini prese dagli OVA disposti in parti diverse dell'ambiente: notiamo che per gli OVA 1_2 e 1_3, situati entrambi attorno al tavolo, vi è una maggiore quantità di verde e una minore quantità di blu rispetto agli OVA 6_2 e

6_4, collocati all'inizio del corridoio. Il verde è dato dalla presenza di una pianta al centro del tavolo, mentre il blu caratterizza la sfera presente lungo il corridoio. Uno degli elementi che caratterizzano un ambiente è perciò il colore: abbiamo deciso di basarci su questo per raggruppare le immagini in clusters.

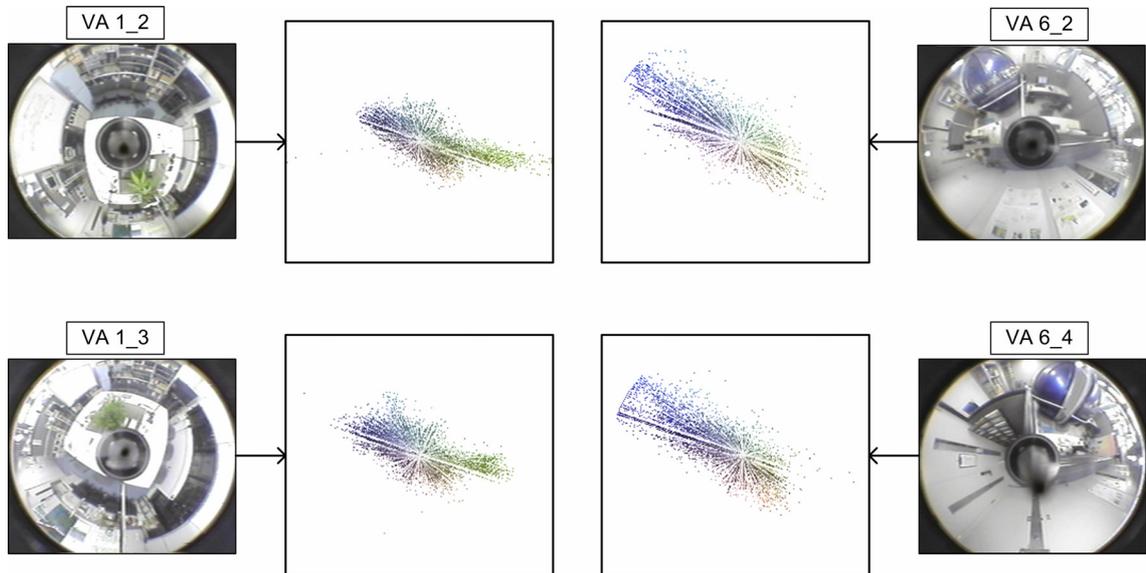


Figura 4.1: distribuzioni di colori nel piano HS (Hue-Saturation) per alcune immagini catturate dagli OVA.

Una volta scelti i gruppi di OVA in base ai colori presenti nelle immagini acquisite, sarà sufficiente determinare a quale cluster appartiene l'immagine catturata dal robot. Applicando l'algoritmo di similarità solo al cluster considerato avremo notevoli probabilità in più di ottenere una localizzazione corretta.

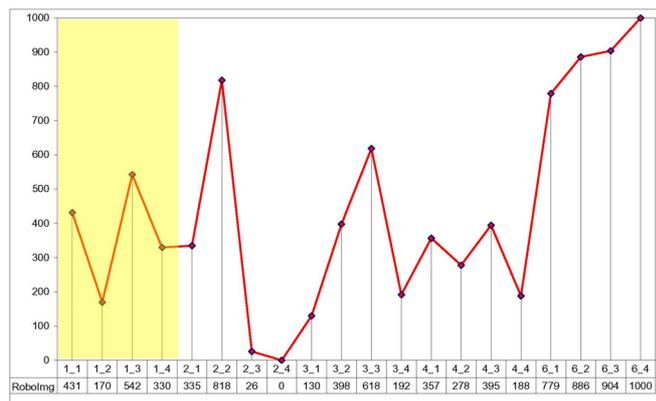


Figura 4.2: Valori di similarità fra le immagini di riferimento degli OVA e quella del robot presa in una posizione prossima all'OVA 1_3

In Figura 4.2 è riportato il grafico che rappresenta il valore di similarità calcolato per una posizione del robot prossima all'OVA 1_3: come possiamo osservare il valore più alto

corrisponde all' OVA 6_4, mentre il valore corrispondente all'OVA corretto è solo il sesto, dopo 6_3, 6_2, 2_2, 6_1, 3_3. Questo perché gli OVA del gruppo 1 sono collocati troppo in basso rispetto al sensore del robot. Notiamo che se avessimo effettuato un clustering delle immagini prima di procedere con la localizzazione e avessimo ottenuto un cluster contenente le immagini prese dagli OVA attorno al tavolo (caratterizzate da una maggiore presenza di bianco, colore del tavolo, e verde, colore delle piante), la localizzazione sarebbe avvenuta correttamente, indicando appunto l'OVA 1_3 come il più vicino al robot.

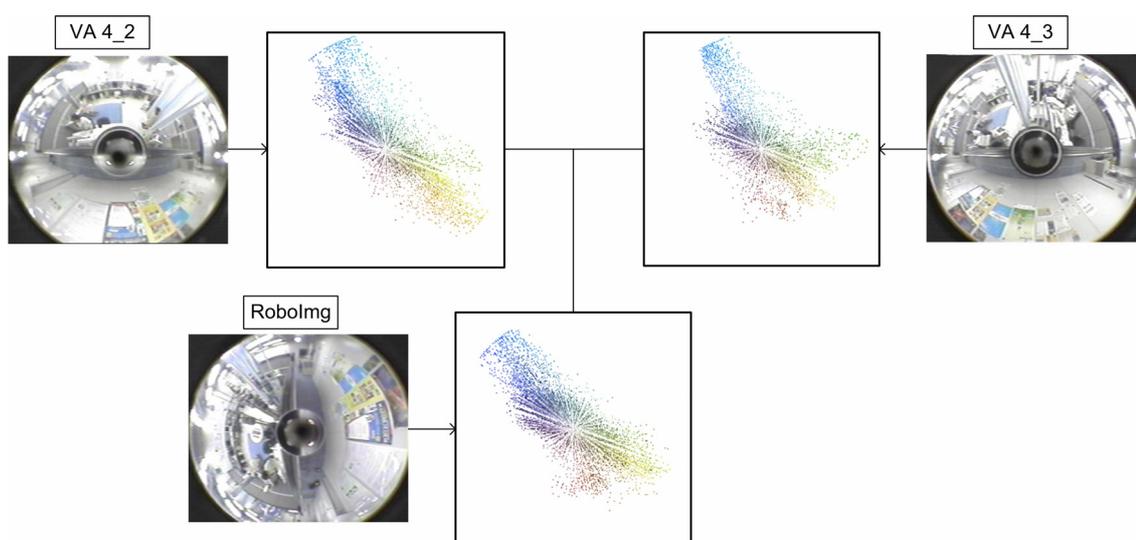


Figura 4.3: distribuzioni di colore per tre immagini catturate in posizioni vicine.

In Figura 4.3 sono riportate le distribuzioni di colore per le immagini catturate da due OVA vicini e dal robot in prossimità di questi: data la somiglianza delle distribuzioni, ci aspettiamo che il nostro algoritmo di clustering assegni allo stesso insieme le tre immagini, sulle quali poi andremo ad effettuare la localizzazione utilizzando la solita procedura.

4.1 Color Matching

Per poter effettuare il clustering sulle immagini di riferimento abbiamo prima bisogno di conoscere quali colori sono presenti in ogni immagine. Il nostro scopo è quello di ottenere per ogni immagine delle classi di equivalenza rappresentanti i colori e, in base a queste, raggruppare le immagini aventi i colori più simili.

L'algoritmo che utilizzeremo è una variante di quello descritto in [15], modificato in modo da fornire risultati migliori. A differenza di questo, che usa solamente lo spazio di

colore bidimensionale HS (Hue Saturation), considereremo anche la luminosità del colore, spostandoci nello spazio HSV (Hue Saturation Value). Inoltre introdurremo delle modifiche nella parte riguardante il calcolo della similarità, in modo da ottenere maggiore robustezza.

In una prima fase ad ogni immagine viene associata una signature contenente i valori corrispondenti alle “regioni di colore” che la caratterizzano: ogni signature è composta da coppie, ognuna contenente un identificativo di una classe di colore (un insieme di colori aventi caratteristiche simili) e il numero di pixel dell’immagine che appartengono alla classe.

Nella seconda fase si effettua il clustering vero e proprio: in base alle signature ottenute si raggruppano in sottoinsiemi tutte le immagini, sia quelle acquisite dagli OVA sia quella appartenente al Robot. Per poter fare ciò abbiamo bisogno di un metodo per calcolare quanto simile sia un’immagine rispetto ad un’altra: utilizzeremo a questo proposito un algoritmo di “tuple matching”, descritto nel paragrafo 4.1.2.

Ci sono più tecniche possibili per associare ogni immagine ad un cluster:

- possiamo confrontare tutte le immagini tra di loro, considerando ogni volta un’immagine diversa come immagine di riferimento: ci aspettiamo che il confronto con gli OVA più vicini a quest’ultima fornisca un valore di similarità maggiore rispetto a quello con OVA lontani. Perciò possiamo stabilire una soglia di similarità tale che, se il confronto tra due immagini fornisce un valore superiore alla soglia, queste appartengono allo stesso cluster.
- possiamo utilizzare delle immagini di riferimento aventi caratteristiche di colore precise (ad esempio un insieme di immagini composte ciascuna da un solo, determinato colore) e raggruppare le immagini degli OVA in base al valore di similarità ottenuto dal confronto con le prime. In questo modo otteniamo tanti sottoinsiemi quante sono le immagini di riferimento: ogni immagine dell’OVA appartenente al sottoinsieme conterrà colori simili a quelli dell’immagine di riferimento.
- un’ultima possibilità consiste nel ricavare direttamente delle signature di riferimento osservando le caratteristiche comuni alle signature appartenenti ad immagini prese da OVA vicini: si tratta di una sorta di “taratura” del sistema. In pratica, si creano delle signature di riferimento partendo da un set di immagini già presente: si divide prima l’ambiente in zone, poi si raggruppano le immagini degli OVA in base a queste zone. Per ogni gruppo ottenuto si analizzano le signature e si ricavano le coppie comuni, che andranno a far parte di una signature di riferimento, corrispondente a quel gruppo. Ci aspettiamo, in questo modo, che i confronti successivi tra le immagini acquisite dagli OVA e le signature di riferimento permettano di ottenere sottoinsiemi contenenti i gruppi selezionati nella prima fase, però associando anche l’immagine del robot al cluster corretto.

Nel seguito del capitolo vedremo in dettaglio l’algoritmo proposto per il calcolo della similarità tra le immagini.

4.1.1 Calcolo della signature

La signature è calcolata in due fasi. Per prima cosa viene calcolato un istogramma tridimensionale nello spazio HSV: in questa rappresentazione i colori presenti nell'immagine sono collocati in punti spazialmente vicini.

Sull'istogramma ottenuto viene poi applicata una tecnica, detta di GT-Clustering, tale da permettere di raggruppare le aree di colore ottenute in regioni.

Per passare dallo spazio dei colori RGB a quello HSV, definiamo prima di tutto la variabile

$$r = \max(R, G, B) - \min(R, G, B).$$

Calcoliamo ora i valori da S (saturation), H (hue) e V (value):

$$S = \begin{cases} 0 & \text{se } r = 0 \\ \frac{r}{\max(R, G, B)} & \text{altrimenti} \end{cases}$$

$$H = \begin{cases} \text{non definito se } r = 0 \\ \frac{G - B}{r} \times 60 & \text{se } R = \max(R, G, B) \\ \left(2 + \frac{G - B}{r}\right) \times 60 & \text{se } G = \max(R, G, B) \\ \left(4 + \frac{G - B}{r}\right) \times 60 & \text{altrimenti} \end{cases}$$

$$V = \max(R, G, B)$$

Le coordinate di colore sono definite come:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} S \cdot \sin(H) \\ S \cdot \cos(H) \\ V \end{pmatrix} \quad \text{se } H \text{ è definito}$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ V \end{pmatrix} \quad \text{se } H \text{ non è definito.}$$

Ogni istogramma è una partizione dello spazio dei colori in classi di equivalenza. Questo processo è detto binning. Prima di calcolare l'istogramma dobbiamo definire il binning.

Selezioniamo un binning che suddivida lo spazio dei colori in parti uguali. Ogni bin contiene colori aventi coordinate di colore (H,S,V) simili. Dato che abbiamo scelto di utilizzare lo spazio H-S-V, bin vicini rappresentano colori simili.

Utilizzando solo pochi bin, l'istogramma risultante sarà poco accurato. Se usiamo molti bin, l'istogramma risultante sarà accurato ma popolato sparsamente. Saranno necessari inoltre più memoria e più tempo di computazione per effettuare il GT-Clustering.

Sia $Bin(R,G,B) \rightarrow (x,y,z)$ una funzione che mappa ogni pixel R,G,B in un bin(x,y,z).

L'istogramma tridimensionale è definito come una matrice contenente in ogni elemento il numero di pixel dell'immagine I aventi coordinate di colore tali da appartenere allo stesso bin:

$$H : N^3 \rightarrow N$$

$$H(b_x, b_y, b_z) = \left| \left\{ (x, y) \in Z^2 \mid (b_x, b_y, b_z) = Bin((x, y)) \right\} \right|$$

Dopo che l'istogramma è stato calcolato utilizziamo una tecnica di GT-clustering allo scopo di trovare i punti di massimo locale nell'istogramma. Il GT-clustering ricerca i massimi locali nell'istogramma collegando ogni posizione con quella adiacente avente il valore più alto. Se nessuna posizione ha valore maggiore a quella corrente, allora quest'ultima è un massimo locale. Tutti i vicini collegati transitivamente con un massimo locale dipendono da questo e formano una regione R, contenente tutte le coordinate dei bin collegati al massimo.

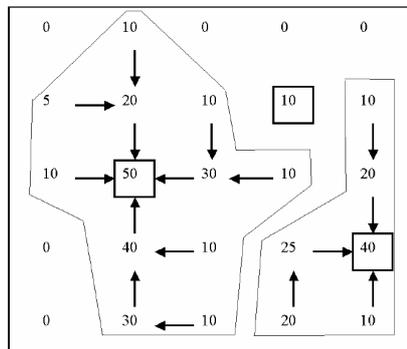


Figura 4.4: Esempio di funzionamento del GT-Clustering su una matrice bidimensionale

Il numero di pixel collegati ad una regione R è calcolato come la sommatoria su tutti i pixel in tutti i bin appartenenti alla regione:

$$Pix(R) := \sum_{(x_i, y_i, z_i) \in R} H(x_i, y_i, z_i)$$

Ogni regione R corrisponde ad una tupla nella signature. Le coordinate di colore della tupla sono calcolate come il baricentro di tutti i bin appartenenti alla regione:

$$Col(R) = \left(\frac{\sum_{(x_i, y_i, z_i) \in R} x_i H(x_i, y_i, z_i)}{Pix(R)}, \frac{\sum_{(x_i, y_i, z_i) \in R} y_i H(x_i, y_i, z_i)}{Pix(R)}, \frac{\sum_{(x_i, y_i, z_i) \in R} z_i H(x_i, y_i, z_i)}{Pix(R)} \right)$$

Mettendo insieme il numero di pixel appartenenti ad una regione R e le coordinate di colore caratteristiche di tale regione, otteniamo la definizione di coppia:

$$Tuple(R) = (Col(R), Pix(R))$$

Se il GT-Clustering trova le regioni R_1, \dots, R_i nell'istogramma, la signature S è definita come l'insieme delle coppie:

$$S := \{Tuple(R_j) | j = 1 \dots i\}$$

Ad ogni immagine corrisponderà perciò una signature composta da coppie, una per ogni regione di colore, contenenti ciascuna le coordinate di colore della regione ed il numero di pixel appartenenti alla regione.

4.1.2 Tuple matching

Per effettuare la segmentazione sull'insieme delle immagini, avremo ora bisogno di un metodo per confrontare tra loro le signature.

L'algoritmo di confronto si basa sulla distanza euclidea tra le diverse regioni di colore appartenenti a due signature:

$$\Delta((x_1, y_1, z_1), (x_2, y_2, z_2)) = \frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}}{\sqrt{(x_{\max} - x_{\min})^2 + (y_{\max} - y_{\min})^2 + (z_{\max} - z_{\min})^2}}$$

dove x_{\max} , x_{\min} , y_{\max} , y_{\min} , z_{\max} , z_{\min} corrispondono i valori che delimitano il range delle coordinate di colore.

Quindi dalla funzione di distanza si ottengono valori compresi tra 0 e 1, dove 1 corrisponde alla distanza massima.

L'algoritmo di confronto procede cercando tra due signature S_1 e S_2 le regioni caratterizzate dalle coordinate di colore più simili, ottenendo la coppia di tuple $t_1 = (c_1, p_1) \in S_1$ e $t_2 = (c_2, p_2) \in S_2$, dove c esprime le coordinate del colore corrispondenti alla regione e p il numero di pixel ad essa appartenenti.

Il valore di similarità M' tra queste due tuple sarà dipendente dalla loro distanza e dal numero di pixel da cui le regioni corrispondenti sono composte:

$$M' = \min(p_1, p_2) \cdot (1 - \Delta(c_1, c_2))$$

Quindi, più pixel hanno in comune le due tuple, maggiore sarà la similarità risultante: al contrario, più distanti saranno i colori, minore sarà il contributo al valore di similarità totale.

Eseguito il confronto, procediamo in questo modo:

- se le due tuple caratterizzano regioni aventi lo stesso numero di pixel, allora entrambe possono essere eliminate dalla signature (il confronto è stato già eseguito)
- se una tupla t_1 corrisponde ad una regione caratterizzata da meno pixel rispetto alla tupla t_2 dell'altra signature, questa può essere rimossa, mentre il valore p_2 di t_2 viene decrementato di un valore pari a p_1 .

Dopo aver aggiornato le due signature, possiamo continuare il confronto cercando nuovamente le due tuple aventi le coordinate più simili, e sommando la similarità tra queste con quella ottenuta nei passi precedenti:

$$M = M + M' = M + \min(p_1, p_2) \cdot (1 - \Delta(c_1, c_2))$$

Il confronto termina quando almeno una delle due signature non contiene più nessuna tupla.

Conclusioni

In questo lavoro abbiamo proposto un nuovo approccio per la localizzazione basata sulle immagini.

Il nostro sistema di visione si basa su un sensore di visione montato sul robot e su diversi OVA (Omnidirectional Vision Agent) disposti lungo il percorso che il robot deve affrontare. Le immagini acquisite vengono elaborate, tramite l'applicazione della trasformata di Fourier, ottenendo una "Fourier signature", che rappresenta l'immagine in modo compatto ed efficiente, basandosi sulle variazioni di luminosità che la caratterizzano.

Il confronto tra le immagini avviene calcolando una funzione di similarità come la norma tra le Fourier signature.

Si è dimostrato che questo sistema di calcolo della similarità permette di ottenere l'invarianza rotazionale e quella prospettica.

Il sistema è stato sperimentato in un ambiente non ottimale in modo da poter evidenziare le possibili difficoltà: i problemi emersi da una prima acquisizione delle immagini consistono nella differenza di luminosità tra l'immagine del robot e quelle degli OVA e nella differenza di prospettiva nella visione causata dalla disposizione degli OVA. A partire da questi, abbiamo cercato delle soluzioni che permettessero un buon funzionamento anche in condizioni critiche: per risolvere il primo problema è stato modificato l'algoritmo di calcolo della similarità, in modo da non tenere conto della prima componente della Trasformata di Fourier. Per il secondo problema è stato riscritto l'algoritmo di conversione da immagine omnidirezionale a panoramica, in modo da permettere di scegliere un'area (FOV) diversa per ogni immagine acquisita, così da poter considerare parti dell'ambiente visibili allo stesso modo da posizioni diverse. Per il primo set di immagini questo non è stato comunque possibile, e ha indotto a ripetere perciò la fase di acquisizione, in modo da avere delle immagini sulle quali trovare un FOV comune tra gli OVA e il robot, ottenendo risultati migliori.

Sono stati poi ricavati i requisiti per un corretto posizionamento dei sensori e per l'elaborazione delle immagini, al fine di ottenere una localizzazione efficiente.

Acquisito un terzo set di immagini, ottenute considerando i criteri ricavati in precedenza, sono stati comparati il sistema distribuito e un sistema statico che utilizza la stessa tecnica di confronto delle immagini. Abbiamo verificato la maggiore efficienza, nel caso di ambienti con contenuti dinamici, del primo rispetto al secondo. Il paragone tra due sistemi basati sullo stesso metodo di calcolo della similarità ha, inoltre, permesso di mettere in evidenza non tanto l'efficacia del sistema in generale, quanto quella dell'approccio distribuito rispetto a quelli classici.

Abbiamo inoltre acquisito altre immagini in un ambiente diverso per provare la proprietà di invarianza prospettica, raggiungendo i risultati aspettati.

Per ottenere migliori prestazioni in situazioni nelle quali i requisiti di acquisizione non possono essere pienamente soddisfatti, e comunque per rendere il sistema più robusto ed efficiente, è stata proposta una tecnica che permette di ridurre il numero di confronti sulle immagini. L'idea è quella di effettuare una selezione, basata sui colori presenti nell'ambiente, delle immagini di riferimento sulle quali applicare il metodo di localizzazione. Sempre a questo scopo è stato presentato un algoritmo di color matching, in grado di fornire un valore di similarità dal confronto tra i colori caratterizzanti due immagini.

Elenco delle figure

Capitolo 1

1.1	Vista in sezione del sistema catadiottrico.....	- 10 -
1.2	Sensori omnidirezionali.....	- 10 -
1.3	FOV di una telecamera prospettica e di un sensore omnidirezionale.....	- 11 -
1.4	Esempio di immagine omnidirezionale.....	- 11 -
1.5	Processo di generazione della “Fourier signature”.....	- 13 -
1.6	Ampiezza dei coefficienti di Fourier di un’immagine omnidirezionale.....	- 14 -
1.7	Esempio di cilindro panoramico.....	- 14 -
1.8	Grafico del valore di dissimilarità in relazione alla distanza tra immagini...	- 16 -
1.9	Fasi del processo di localizzazione con l’utilizzo del DVS.....	- 18 -
1.10	Fasi del processo di localizzazione con l’utilizzo di un sistema di tipo statico.....	- 19 -
1.11	Fallimento della localizzazione con immagini acquisite da un sistema statico.....	- 20 -
1.12	Localizzazione con immagini acquisite dal DVS.....	- 20 -
1.13	Mappa metrica dell’ambiente utilizzato per i test.....	- 21 -
1.14	Grafo rappresentante le connessioni topologiche tra OVA.....	- 21 -

Capitolo 2

2.1	Il Robot – Eveliee P1.....	- 23 -
2.2	Il Robot utilizzato all’interno dell’ambiente.....	- 23 -
2.3	Un Color Quad Processor.....	- 24 -
2.4	I Color Quad Processor utilizzati nei test.....	- 24 -
2.5	Mappa del laboratorio dove si sono svolti i test.....	- 25 -
2.6	Risultato dell’elaborazione di un Color Quad Processor.....	- 26 -
2.7	Programma per la divisione delle immagini.....	- 27 -
2.8	Determinazione del centro dell’immagine omnidirezionale.....	- 28 -
2.9	Corrispondenza tra i punti dell’immagine omnidirezionale e quelli del cilindro panoramico.....	- 30 -
2.10	Riflessione di un punto dell’ambiente sul sensore CCD.....	- 31 -
2.11	Riflessione dei punti dell’ambiente sul sensore CCD.....	- 32 -

Capitolo 3

3.1	Cilindro panoramico corrispondente all'immagine acquisita dal robot, relativa al primo set di immagini.....	- 36 -
3.2	Cilindro panoramico corrispondente all'immagine acquisita dall'OVA 6_2, relativa al primo set di immagini.....	- 36 -
3.3	Percentuale di match corretto per ogni OVA di interesse con il primo set di immagini.....	- 37 -
3.4	Programma utilizzato per la determinazione del FOV ottimale.....	- 38 -
3.5	Immagine omnidirezionale acquisita dal robot, relativa al secondo set di immagini.....	- 39 -
3.6	Immagine omnidirezionale acquisita dall'OVA 6_3, relativa al secondo set di immagini.....	- 39 -
3.7	Cilindro panoramico corrispondente all'immagine acquisita dal robot, relativa al secondo set di immagini.....	- 39 -
3.8	Cilindro panoramico corrispondente all'immagine acquisita dall'OVA 6_3, relativa al secondo set di immagini.....	- 39 -
3.9	Confronto tra le percentuali di match per ogni OVA di interesse del primo e del secondo set.....	- 40 -
3.10	Immagine omnidirezionale acquisita dal robot, relativa al terzo set di immagini.....	- 41 -
3.11	Immagine omnidirezionale acquisita dall'OVA 6_1, relativa al secondo set di immagini.....	- 41 -
3.12	Confronto tra le percentuali di match per ogni OVA di interesse del primo, del secondo e del terzo set.....	- 42 -
3.13	Confronto di risultati con l'utilizzo di un sistema di localizzazione statico e del DVS in una situazione statica.....	- 43 -
3.14	Localizzazione corrispondente al confronto di Figura 4.13.....	- 44 -
3.15	Risultati della localizzazione con un sistema statico in una situazione dinamica.....	- 44 -
3.16	Risultati della localizzazione con il DVS in una situazione dinamica.....	- 45 -
3.17	Localizzazione corrispondente ai confronti di Figura 4.15 e 4.16.....	- 45 -
3.18	Localizzazione con immagini acquisite con un sistema statico.....	- 47 -
3.19	Localizzazione con immagini acquisite dal DVS.....	- 47 -
3.20	Situazione in cui si verifica differenza di prospettiva.....	- 48 -
3.21	Cilindro panoramico corrispondente all'immagine acquisita dal robot, relativa alla situazione di Figura 4.20.....	- 49 -
3.22	Cilindro panoramico corrispondente all'immagine acquisita dall'OVA 4, relativa alla situazione di Figura 4.20.....	- 49 -
3.23	Posizione dei sensori e del robot nell'ambiente per la prova sull'invarianza prospettica.....	- 49 -
3.24	Grafico di similarità del confronto tra le immagini acquisite per la prova sull'invarianza prospettica.....	- 49 -

Capitolo 4

4.1	Distribuzioni di colore per alcune immagini catturate dagli OVA.....	- 52 -
4.2	Grafico dei valori di similarità con l'applicazione del GT-Clustering.....	- 52 -
4.3	Distribuzioni di colore per immagini catturate in posizioni vicine.....	- 53 -
4.4	Esempio di funzionamento del GT-Clustering.....	- 56 -

Bibliografia

- [1] E. Menegatti E., T. Maeda , H. Ishiguro. *Image-based Memory for Robot Navigation Using Properties of the Omnidirectional Images*. 2002
- [2] E. Menegatti , M. Zoccarato , E. Pagello , H. Ishiguro. *Image-based Monte-Carlo Localisation without a Map*. 2004
- [3] E. Menegatti, E.Pagello. *Omnidirectional Distributed Vision for Multi-Robot Mapping*.
- [4] H. Ishiguro. *Distributed Vision System: A Perceptual Information Infrastructure for Robot Navigation*. 1997
- [5] J.Wolf, W.Burgard, H.Burkhardt. *Robust Vision-based localization for Mobile Robots using an image retrieval system based on invariant features*. 2002
- [6] J.Wolf, W.Burgard, H.Burkhardt. *Robust Vision-based Localization by Combining an Image Retrieval System with Monte Carlo Localization*.
- [7] H. Ishiguro , T.Sogo , T. Ispida. *Mobile Robot Navigation by a Distributed Vision System*. 1997
- [8] S.K. Nayar. *Omnidirectional Vision*. 1997
- [9] W.Burgard, A. Cremers, D. Fox, D. Hhnel, G. Lakemeyer, D. Schulz, W. Steiner, S.Thrun. *Experiences with an interactive museum tour-guide robot*. 1999
- [10] A.Kropp, N.Master, S.Teller. *Acquiring and Rendering High-Resolution Spherical Mosaics*.
- [11] V.N. Peri , S.K. Nayar. *Generation of Perspective and Panoramic Video from Omnidirectional Video*. 1997
- [12] S.Baker , S.K. Nayar. *A Theory of Single-Viewpoint Catadioptric Image Formation*.
- [13] H. Bakstein , T. Pajdla. *Rendering novel views from a set of omnidirectional mosaic images*. 2003

- [14] T. Kawamura , T.Ueoka , Y.Kiuchi , Y.Kono , M.Kidode. *Image Similarity Measurement by an Integrated Probabilistic Histogram*.
- [15] D. Balthasar. *Color Matching Using Tuple Matching*.
- [16] E. Menegatti E. , F. Nori , E. Pagello , C. Pellizzari , D. Spagnoli. *Designing an omnidirectional vision system for a goalkeeper robot In RoboCup-2001: Robot Soccer World Cup V*. A cura di Birk A., Coradeschi S., Tadokoro S., pp. 78–87. Springer.
- [17] H. Ishiguro. *Development of Low-cost Compact Omnidirectional Vision Sensors In Panoramic Vision*.
- [18] S.M. Aghito. *Sensori catadiottrici omnidirezionali con singolo centro di proiezione*. 2001
- [19] I. Ulrich, I.Nourbakhsh. *Appearance-Based Place Recognition for Topological Localization*.
- [20] J. Matas, J. Kittler. *Spatial and Feature Space Clustering: Applications in Image Analysis*. 1995
- [21] G. Gordon, T. Darrel, M. Harville, J.Woodfill. *Background estimation and removal based on range and color*. 1999
- [22] J.F. Campapum, M.H. Fisher. *3-D Colour-based Segmentation and Recognition for Environmental Monitoring*. 1998
- [23] J.F. Campapum, M.H. Fisher. *Segmentation using Spatial-Feature Clustering from Image Sequences*.
- [24] H. Ishiguro , K. Kimoto. *Toward social interaction technologies of robot systems*.
- [25] E.Menegatti, E.Pagello. *Designing omnidirectional vision systems and inserting them in a distributed vision system for mobile robots*. 2001
- [26] G. Adorni, S.Cagnoni, M.Carletti, M.Mordonini. *Omnidirectional vision algorithms in robotics*.
- [27] Intel[®] Corporation. *Intel[®] Integrated Performance Primitives for Intel[®] Architecture, Reference Manual*. 2003