

Un sistema multiagente per la navigazione in ambiente esterno di un robot guidato da un sistema di visione omnidirezionale

Michele Folgheraiter, Giuseppina Gini

*DEI, Politecnico di Milano
Piazza L. da Vinci 32
20133 Milano
E-mail gini@elet.polimi.it*

Sommario

Il sistema robotico di navigazione sviluppato usa immagini a 360gradi prese per mantenere il robot su un percorso sicuro e per farlo navigare in ambiente esterno.

Il riconoscimento della strada e dei landmark si compone sia di una analisi cromatica, per estrarre le texture, che di una analisi dei contorni, per estrarre le direzioni libere. Oltre a questi due agenti che trovano quindi la direzione libera e l'area libera, altri agenti funzionano in parallelo per la costruzione della mappa e per il riconoscimento della attuale 'pose' del robot. Gli agenti visivi sviluppati saranno illustrati in dettaglio nel compito di circumnavigare un edificio.

1 Introduzione

Uno dei problemi tuttora aperti nel mondo della robotica mobile è quello della navigazione autonoma in ambiente esterno. Infatti la maggior parte dei metodi sviluppati usa ipotesi sul mondo, quali la presenza di un piano su cui muoversi e la buona qualità dei dati sensoriali, che sono difficili da ottenere in ambiente naturale esterno.

Scopo di questa ricerca è la realizzazione di un sistema software ad agenti che permetta ad un robot mobile di muoversi in ambiente esterno servendosi dei dati ottenuti attraverso un sistema di visione omnidirezionale. Il sistema di visione sul robot prevede una telecamera montata sulla parte superiore del robot e puntata verso l'alto, in direzione di uno specchio omnidirezionale a forma di tronco di cono. Le immagini deformate dallo specchio contengono informazioni sia sull'intorno del robot che sull'ambiente, e costituiscono la sola fonte di informazione sensoriale per il sistema di navigazione.

L'immagine può essere elaborata con svariate tecniche, ma in particolare le due tecniche duali di analisi partendo dai contorni o partendo dall'individuazione di regioni sembrano utili in diverse situazioni. Un sistema che estrae i contorni e li segmenta è utile per scoprire il confine delle aree di

interesse ma può essere troppo lento per il real-time e soprattutto può generare falsi risultati individuando contorni che non esistono. D'altra parte il solo sistema basato sulla analisi di regioni con la stessa texture o con lo stesso colore è debole nelle situazioni in cui tale caratteristica sia simile anche per oggetti confinanti e diversi.

Se ipotizziamo un sistema di controllo multiagente, possiamo sfruttare entrambi i metodi in parallelo: il primo agente ci dice la direzione della strada da seguire, il secondo ci dice quali zone adiacenti al robot sono di strada libera e quali no. Il sistema di arbitraggio fra i due potrà quindi fornire gli input necessari alle politiche ad alto livello (del tipo "manteniti nel centro dei passaggi" oppure "non andare sul prato se c'è una strada", etc.) . In un'altra precedente applicazione, in ambienti interni, gli agenti di visione sono stati usati per costruire una mappa dell'ambiente e per consentire al robot un controllo nel movimento nello spazio cartesiano [1, 2].

Nelle sezioni che seguono illustriamo le caratteristiche più interessanti delle immagini riprese dal sistema omnidirezionale, schematizziamo gli algoritmi usati dai due agenti, ed illustriamo i risultati ottenuti su immagini riprese ad un sito ENEA Casaccia.

2 Il sistema di visione: specchio e caratteristiche delle immagini

2.1 Caratteristiche dello specchio

Al fine di rendere possibile una visione omnidirezionale viene utilizzato un particolare specchio, posto sopra alla telecamera del robot; tale specchio è stato progettato da [3] per una visione accurata sia in termini di distanza che di direzione per la zona circostante il robot. In particolare, si ha una visione accurata in termini di direzione per lo spazio posto ad una certa distanza dal robot; ed una visione molto chiara e precisa delle immediate vicinanze del robot.

Lo specchio è composto da tre parti, ognuna delle quali in grado di produrre una zona dell'immagine che soddisfa uno dei requisiti appena descritti.

- *Parte isometrica* - I sistemi di visione omnidirezionale, ottenuti attraverso l'impiego di specchi conici standard, sono caratterizzati da un legame non lineare tra le distanze misurate tra i punti nell'immagine e quelle misurate tra i corrispondenti punti nella scena inquadrata. Questa non linearità si traduce in una distorsione nell'immagine ottenuta, distorsione che cresce velocemente al crescere della distanza fra l'osservatore e l'oggetto considerato. E' ovviamente possibile correggere le immagini una volta nota la funzione che definisce il profilo dello specchio utilizzato, tuttavia l'accuratezza delle misure e' pesantemente influenzata dall'effetto congiunto della distorsione appena descritta e del campionamento delle immagini. Lo specchio usato fu progettato con il preciso scopo di compensare questo problema agendo direttamente dal punto di vista ottico, di modo che l'immagine ottenuta fosse caratterizzata dalla stessa accuratezza delle misure indipendentemente dalla distanza degli oggetti. Questo obiettivo e' stato raggiunto mediante la parte isometrica dello specchio, che conserva la metrica dell'immagine non distorcendo gli angoli e scalando le distanze di un fattore costante. In Figura 1 e' indicata con la lettera A la zona dell'immagine prodotta da questa parte dello specchio.

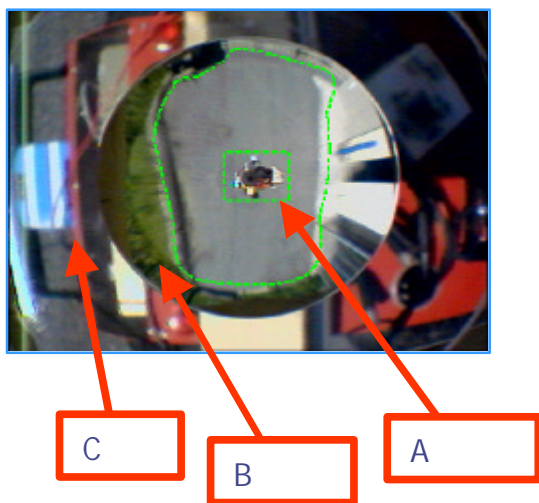


Figura 1. Analisi delle tre diverse zone generate dallo specchio

- *Parte a curvatura costante* - La seconda parte dello specchio, a curvatura costante, e' stata progettata per permettere la visione di aree abbastanza distanti dal robot mantenendo una buona accuratezza dal punto di vista della direzione e tralasciando invece quella riguardante le distanze. In Figura 1 e' indicata con la lettera B la zona dell'immagine prodotta da questa parte dello specchio.
- *Parte planare* - La terza e ultima parte dello specchio, detta planare, non introduce nessun tipo

di distorsione dell'immagine e viene utilizzata per la zona dello spazio immediatamente circostante il robot.. In Figura 1 e' indicata con la lettera C la zona dell'immagine prodotta da questa parte dello specchio. I bordi in verde sono una indicazione dell'area occupata dal robot stesso (quadrato piccolo) e del piazzale in cui può muoversi.

2.2 Le immagini

Per permettere al robot di muoversi in sicurezza nello spazio circostante (evitando gli ostacoli) e di raggiungere determinati target ricercando un percorso adatto, e' necessario procedere ad una accurata analisi delle immagini acquisite dalla telecamera montata a bordo del robot stesso. Come osservabile da Figura 1 le immagini sono composte da due zone principali:

- una zona centrale circolare in cui sono mostrati il robot e l'ambiente che lo circonda, in tutte le direzioni. In questa zona le linee rette vengono deformate il meno possibile;
- una zona esterna in cui sono mostrati i dettagli dell'ambiente nelle immediate vicinanze del robot. A causa delle dimensioni piuttosto importanti del robot, questa zona risulta parzialmente occupata dal lato superiore del robot e dalle sue ruote. Questa parte non verrà quindi usata per l'elaborazione.

L'analisi delle immagini e' stata affrontata da due punti di vista differenti. Il primo approccio e' stato quello di basarsi sulla ricerca dei contorni delle immagini al fine di individuare i bordi della strada e dunque del terreno percorribile dal robot. Il secondo e' stato quello di procedere all'analisi della texture di ogni immagine attraverso la valutazione delle componenti colore oppure di caratteristiche statistiche quali la media o i momenti di ordine superiore. Ognuno dei due approcci funziona concettualmente in parallelo con l'altro e produce una risposta al problema di dove andare.

3 L'agente di analisi dei contorni per trovare la direzione

La scelta di analizzare le immagini con tecniche di rilevazione dei contorni permette di individuare i bordi salienti dell'immagine, come quelli della strada oppure dei marciapiedi, al fine di delimitare, almeno grossolanamente, il terreno percorribile dal robot. L'estrazione di contorni (edge detection) e' un'operazione fondamentale nel processo di elaborazione delle immagini, perché proprio nei contorni si concentra la maggiore quantità di informazioni semantiche. Un contorno può essere definito come il luogo dei punti in corrispondenza del quale una certa caratteristica della immagine come la luminanza, la crominanza, o la statistica locale della tessitura subisce un cambiamento significativo. I criteri comuni che si seguono nelle operazioni di edge detection sono:

- Basso tasso di errore: e' importante che un contorno non venga perso o che, viceversa se ne identifichino di falsi;

- Buona localizzazione: si deve minimizzare lo scarto tra la posizione reale del contorno e quella stimata;
- Risposta unica: nel caso di riscontri molteplici si deve procedere all'eliminazione di quelli che, presumibilmente, non sono corretti.

Fondamentalmente tutte le tecniche di edge detection sfruttano le caratteristiche del gradiente o del laplaciano della luminosità dell'immagine considerata ed in particolare ricercano i punti di stazionarietà nel primo caso, gli zeri nel secondo.

3.1 Metodo di Canny

Il metodo di Canny soddisfa nel modo migliore i criteri esposti sopra. L'algoritmo lavora su immagini in scala di grigio, pertanto è stata realizzata una funzione di utilità che permette di convertire un'immagine a colori in formato ppm in un'immagine in formato pgm a 8 bit.

Vediamo ora le fasi dell'analisi.

A - Filtraggio gaussiano

Preliminarmente viene applicato all'immagine, al fine di ridurre la rilevanza del rumore, un filtro gaussiano separabile. In linea teorica sarebbe necessario generare i campioni (2D) della risposta all'impulso da usare poi per la convoluzione con l'immagine per ottenere quella filtrata; tuttavia avendo scelto un filtro separabile si può procedere nel seguente modo:

- si genera una gaussiana monodimensionale;
- viene fornito il valore di σ (deviazione standard) da cui dipende il numero di campioni prima che le code della gaussiana vadano a zero;
- si effettua la convoluzione fra i campioni della gaussiana e l'immagine secondo la direzione x;
- si ripete l'operazione precedente lavorando però nella direzione y.

L'immagine così ottenuta risulta smussata in modo più o meno accentuato a seconda del valore di σ , che deve essere scelto con cura. Infatti il tempo di esecuzione del filtraggio cresce linearmente al crescere di questo parametro ma la qualità dell'immagine ottenuta degrada al diminuire dello stesso parametro. Il giusto trade-off tra velocità di esecuzione e precisione nella rilevazione è stato individuato nell'intervallo 1.0 – 1.5.

B - Calcolo del gradiente dell'immagine

Il gradiente dell'intensità luminosa dell'immagine ottenuta dopo il filtraggio gaussiano è calcolato in modo separabile calcolando la derivata parziale rispetto ad x e ad y. L'operatore di calcolo delle derivate parziali usa la differenza finita della luminanza di due pixel vicini lungo due direzioni ortogonali. I due filtri differenziali sono: $[-1 \ 0 \ 1]$ e $[1 \ 0 \ -1]^T$. Il passo successivo consiste nel calcolo del modulo del gradiente stesso, come radice della somma dei quadrati.

C - Analisi dei massimi del gradiente

Vengono estratti dall'immagine i punti in cui il modulo del gradiente è massimo lungo la direzione del gradiente stesso, tali punti sono candidati ad appartenere al contorno dell'immagine.

D - Isteresi

I punti così ottenuti vanno ulteriormente selezionati. Un modo piuttosto semplice consiste nel fissare una soglia sul modulo e nello scartare i pixel in cui il modulo del gradiente è inferiore alla soglia. Tale possibilità però tratta allo stesso modo zone diverse dell'immagine, sottoposte magari a diverso contrasto. Si fissano due soglie, alta e bassa. i punti (x,y) tali che il modulo del gradiente

- supera l'alta vengono classificati come sicuramente appartenenti al contorno,
- è inferiore a bassa vengono classificati come sicuramente non appartenenti al contorno.
- è in una zona compresa sono considerati appartenenti al contorno solamente se possono essere connessi, direttamente o indirettamente, a pixel del contorno mediante un cammino che non passi per nessun punto non appartenente.

In Figura 1 abbiamo visto l'immagine di partenza. Per l'applicazione del metodo di Canny è necessario convertire tale immagine a colori nella corrispondente in scala di grigio. Il risultato ottenuto utilizzando il valore 1.0 per la deviazione standard del filtraggio gaussiano, 0.5 per la soglia bassa dell'isteresi e 0.8 per la soglia alta è mostrato nella Figura 2. Come si può vedere i risultati ottenuti possono essere sicuramente definiti buoni, il tempo di esecuzione si attesta attorno a 0.3 secondi.

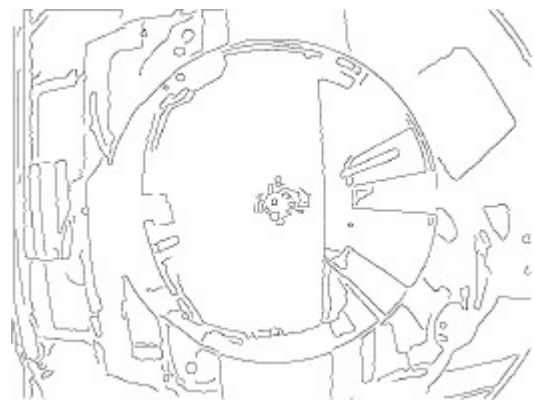


Figura 2. Risultato del metodo di Canny

3.2 Segmentazione dei contorni

L'immagine ottenuta con il metodo di Canny rappresenta solamente una serie di punti appartenenti ad un contorno dell'immagine che non hanno però nessun legame tra di loro. È quindi necessario, al fine di individuare la zona di immagine rappresentante la strada, segmentare l'immagine, per evidenziare delle rette o dei segmenti di retta caratterizzanti appunto tale zona libera.

La trasformata di Hough permette di realizzare il collegamento dei punti dei contorni, determinando se essi giacciono o meno su una curva di forma determinata. Il nostro problema è dunque il seguente:

dati n punti che giacciono sul piano xy , vogliamo determinare dei sottoinsiemi che giacciono su linee rette, che possiamo definire come le nostre primitive. Consideriamo la rappresentazione polare della retta:

$$x\cos\theta + y\sin\theta = \rho$$

con cui un punto nello spazio dei parametri θ e ρ rappresenta una retta nello spazio cartesiano (x,y) , ed un punto nel piano cartesiano (x,y) è rappresentato con una sinusoidale, la cui pulsazione non varia al variare del punto, nello spazio dei parametri. M punti giacenti su una stessa retta

$$x\cos\theta_i + y\sin\theta_i = \rho_i,$$

daranno luogo, nello spazio dei parametri, ad M curve sinusoidali che si intersecano nel punto θ_i, ρ_i . Nella realtà le sinusoidi non si intersecano esattamente nello stesso punto a causa del rumore. Effettuando una quadrettatura dello spazio dei parametri si ottiene una matrice di celle contatori. Ad ogni punto del contorno associamo il fascio di rette centrato in quel punto e lo rappresentiamo attraverso la corrispondente sinusoidale. Per ogni cella attraversata dalla sinusoidale incrementiamo di 1 il contatore associato. Ogni cella di questa matrice rappresenta una retta ed ha associate le seguenti informazioni:

- Il numero di voti ottenuti dalla retta rappresentata;
- Le coordinate del primo e dell'ultimo punto trovati per la retta;
- Le coordinate del primo e dell'ultimo punto appartenenti al più lungo segmento contiguo nella retta stessa;
- Le coordinate del primo e dell'ultimo punto del segmento correntemente trovato;
- La lunghezza del segmento di lunghezza massima trovato.

Dopo aver costruito la matrice di Hough, essa deve essere analizzata cella per cella, per scartare le rette associate a celle che non raggiungono il numero minimo di voti o i cui segmenti non raggiungono la lunghezza minima. Le rette che superano i test sono poi filtrate per eliminare quelle troppo vicine. Nell'immagine di output si riportano le rette che hanno superato tutti i test, disegnate congiungendo il primo e l'ultimo punto trovato, oppure i segmenti di dimensione massima appartenenti alle rette trovate.

Il principale problema riscontrato nell'applicazione di questo metodo è derivato dalla complessità delle immagini ottenute dopo l'elaborazione tramite l'algoritmo di Canny, i contorni sono infatti molti e seppur ben definiti certamente non troppo allineati. Questo ha spesso reso difficile ottenere un'immagine finale in cui fosse individuata solamente la strada.

In Figura 3, si osserva che un bordo della strada è ben visibile, l'altro si trova nella parte deformata. Il risultato è stato ottenuto impostando il parametro indicante il numero di voti a 100, suddividendo lo spazio dei parametri con una granularità pari ad 1 per ρ ed a 2 per θ . I valori minimi per differenziare due rette sono stati settati rispettivamente a 10 e a 20. Infine la lunghezza minima del più lungo segmento appartenente

ad ogni retta è stata posta a 20 e la distanza massima tra due punti successivi appartenenti allo stesso segmento a 50. Il tempo di esecuzione è di 1.18 secondi.



Figura 3. Risultato della segmentazione attraverso trasformata di Hough

4 L'agente di analisi della texture per trovare lo spazio libero

Lo scopo dei metodi di analisi della texture è quello di ricavare informazioni sull'ambiente vicino al robot con particolare interesse, ovviamente, per la posizione dello spazio libero. L'obiettivo principale è dunque quello di produrre immagini binarie in cui siano accesi solo i pixel corrispondenti alla zona in cui il robot può muoversi. Vediamo preliminarmente una breve trattazione sui formati di colore utilizzati, ossia RGB e HSV.

- RGB (red, green, blue) ossia i tre canali utilizzati per descrivere ogni singolo pixel dell'immagine, è un modello additivo poiché le intensità luminose dei tre canali vengono sommate per formare l'effettivo colore del pixel considerato; ogni canale può assumere valori da 0 a 255 (8bit per canale) a seconda dell'intensità luminosa del colore. Pixel completamente neri hanno tutti i valori pari a 0, mentre pixel completamente bianchi hanno tutti i valori pari a 255.
- HSV (hue, saturation, value), è un modello di rappresentazione del valore di un pixel basato su tinta, saturazione e intensità luminosa.
 - hue indica un angolo. I valori di tinta, infatti, sono organizzati in cerchio a partire dal rosso (posizionato a 0 gradi);
 - saturation indica la purezza, l'opacità di un colore. I valori di saturazione vanno da 0 (indice di immagine in scala di grigio) a 100;
 - value esprime la luminosità di un pixel; un valore prossimo allo 0 indica oscurità, mentre 100 è la massima luminosità. Il valore massimo di luminosità non significa bianco, a meno che la saturazione non sia nulla.

4.1 Metodo basato sull'analisi dei momenti statistici

Il primo metodo provato per la ricerca dello spazio libero attorno al robot si basa sull'analisi dei momenti statistici dell'immagine. Questo algoritmo si basa sull'ipotesi che lo spazio libero (la strada su cui si muove il robot) abbia una colorazione abbastanza uniforme e diversa da quella degli ostacoli. Purtroppo nessuna di queste due ipotesi è completamente soddisfatta e questo rende i risultati ottenuti poco attendibili. L'algoritmo si basa sulla valutazione di grandezze di tipo statistico dell'immagine, nello specifico media e momento quarto ordine. Per ogni pixel vengono calcolati la media e il momento del quarto ordine delle singole componenti rgb dei pixel di un intorno quadrato [Gini, Marchi]. Il momento del quarto ordine è un buon indice della diversità dei pixel compresi nell'intorno considerato, dunque assume valori bassi in zone di colore uniforme e alti in prossimità di discontinuità nell'immagine (indici della separazione fra strada e ostacoli).

Il procedimento è molto pesante dal punto di vista tempo; si prevede dunque di calcolare media e momento solo in alcuni punti dell'immagine fra loro equidistanti e di ricavare il valore dei due indici statistici nei rimanenti punti dell'immagine attraverso un'interpolazione bilineare. Una volta calcolati media e momento di tutti i pixel dell'immagine, è possibile servirsi di tali valori per modificare l'immagine stessa al fine di renderla adatta alla successiva fase di analisi e ricerca dello spazio libero. Per fare ciò l'algoritmo prevede l'utilizzo del momento di quarto ordine normalizzato fra 0 e 1. Si procede poi a convertire l'immagine in HSV, ed a costruire il riferimento per individuare la strada. Per questo vengono letti i valori medi di hue e value di una zona di immagine sicuramente corrispondente allo spazio libero che si sta cercando, nel nostro caso la strada.

Una volta disponibili i valori di riferimento è possibile accendere i pixel con valori di hue e value simili a quelli di riferimento (che dunque dovrebbero appartenere alla strada) e spegnere quelli con valori molto diversi. Questa fase è piuttosto delicata; vengono considerate due soglie dinamiche per allargare le maglie del filtro quando si è lontani da discontinuità dell'immagine e restringerle quando ci si avvicina ad esse (e quindi a degli ostacoli). Si osservano questi problemi:

- A causa della bassa saturazione delle immagini disponibili, i valori di hue dei pixel appartenenti alla strada sono spesso molto simili a quelli di pixel che si trovano al di fuori, ad esempio nel prato. Questo fatto rende arduo tarare correttamente i parametri delle soglie;
- È molto difficile stabilire da dove attingere le informazioni sulla zona di strada campione; il sistema visivo omnidirezionale fa sì che il robot stesso sia presente nelle immagini.
- Gli ostacoli (palazzi in genere) proiettano ombre che coprono una parte della strada a seconda delle

condizioni atmosferiche. In condizioni di abbondante illuminazione la discontinuità fra zone in ombra e zone al sole è più marcata di quella fra strada e ostacoli. L'algoritmo pertanto interpreta le zone d'ombra come ostacoli nel caso ci si trovi alla luce e viceversa..

È stata aggiunta un'ulteriore condizione perché un pixel sia catalogato come appartenente allo spazio libero, ossia che la sua componente di colore verde in RGB sia inferiore ad una certa soglia, circa 160-165. Per quanto riguarda i tempi di elaborazione per ottenere i risultati di Fig 4, ci sono voluti circa 1.15 secondi



Figura 4. Risultato metodo dei momenti + soglia verde

4.2 Metodo basato sulle componenti di colore

Il secondo metodo provato si basa sull'analisi delle singole componenti di colore rgb dei pixel; l'idea è nata da programma di fotoritocco Gimp. Le ipotesi per l'applicabilità del metodo sono essenzialmente coincidenti con quelle del metodo basato sui momenti statistici.

L'immagine da analizzare necessita sempre di un filtraggio per ridurre l'incidenza del rumore. Anche la lettura dei dati di riferimento sulla strada è analoga a quanto precedentemente detto. Per ogni pixel dell'immagine sono stati calcolati gli scostamenti delle tre componenti rgb dai valori ricavati dalla zona di riferimento. Per effettuare la sogliaatura si è scelto di utilizzare tre e soglie, una per ogni componente di colore.

In Figura 5 vediamo l'immagine con lo spazio libero ottenuta dalla consueta immagine di test. La qualità del risultato è buona, sicuramente migliore di quella ottenibile dal precedente metodo.

Il tempo di esecuzione è pressoché identico a quello del metodo dei momenti e si attesta attorno a 1.10 secondi; del resto la parte veramente pesante dal punto di vista computazionale è la procedura di filtraggio dell'immagine (immutata rispetto al metodo precedente).

Un filtraggio meno pesante: il filtro gaussiano separabile, fornisce risultati di qualità identica a quella ottenuta con l'altro filtro, ma in un tempo di 0.34

secondi, valore decisamente inferiore a quello precedentemente registrato.

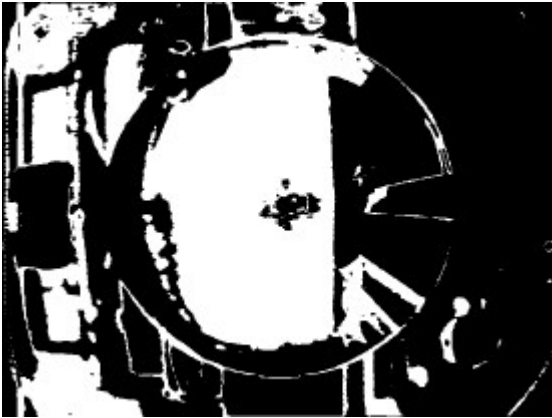


Figura 5. Risultato del metodo delle componenti di colore RGB

Tutti i valori riportati a proposito dei tempi di esecuzione dei vari algoritmi si riferiscono a risultati ottenuti utilizzando un computer portatile dotato di processore AMD Duron 800Mhz e 256MB di memoria; le prove sono state svolte in ambiente Linux Mandrake 8.0 e il software e' stato compilato con l'ottimizzazione massima del codice. In Figura 6 sono riportati i tempi.

Estrazione contorni (Canny)	0.32 s
Segmentazione (trasformata di Hough)	1.20 s
Analisi Texture (Metodo Componenti di colore)	0.34 s
Analisi Texture (Metodo dei Momenti)	1.15 s

Figura 6. Tempi di elaborazione

5 Integrazione degli agenti e conclusioni

In un sistema parallelo in grado di eseguire concorrentemente le due elaborazioni, si potrebbe usare l'analisi della texture come metodo principale da usare nel feedback di controllo del movimento. Esperimenti condotti in ambiente interno hanno dimostrato una buona capacità del robot di evitare ostacoli e di costruire una mappa con solo tale metodo [2]. Il problema è che da sola l'analisi della texture non sfrutta la regolarità della strada e può essere troppo conservativa in aree a colore misto, tendendo a vedere ostacoli dove essi non ci sono. Il metodo della trasformata di Hough, che richiede un tempo superiore di circa 4 volte, potrebbe essere invece usato ogni 4 cicli di controllo per verificare l'allineamento del robot con la direttrice.

Il metodo è stato testato preliminarmente nell'ambito del progetto MURST-ENEA, cui si riferiscono queste immagini. L'idea dei due agenti è stata ispirata dal progetto stesso, che mette a disposizione una architettura di calcolo parallela, ideale per l'esecuzione concorrente dei due agenti di visione sviluppati. Il sito

del test, un edificio circondato da parcheggio, strada, e prato ha fornito l'idea di inseguire profili rettilinei, che sono tali solo in un'area vicina al robot, quindi nel costeggiare un marciapiede o un edificio. Per quanto riguarda l'integrazione con il sistema di navigazione, si rimanda a quanto descritto in [2] per quanto riguarda la costruzione della mappa, ancora non realizzata per il sito di queste immagini.

L'estensione verso il movimento in ambiente esterno ha reso evidente la difficoltà di uso sia del metodo di calibrazione che del sistema odometrico per calcolare le posizioni dei landmark naturali. Ci aspettiamo di continuare lo sviluppo del sistema di navigazione facendo ricorso a dati di localizzazione assoluta ottenuti da GPS, per ovviare agli errori odometrici in ambiente esterno. Come evidenziato in [4], la visione per la navigazione dei robot ha ottenuto molti risultati importanti ma a prezzo di lunghi tempi di calcolo o di lunghi tempi di addestramento. Il nostro progetto cerca di ridurre i tempi di calcolo usando appunto immagini a 360°, così come cerca di evitare lunghi tempi di addestramento usando una finestra stessa dell'immagine per estrarre i parametri di confronto. Questo è compatibile con una modalità mista di controllo di supervisione e di autonomia, come comune nelle applicazioni esterne.

Ringraziamenti

Si ringrazia il contratto MURST-ENEA, progetto PRASSI, sia per il finanziamento che per la cooperazione resa possibile in esso.

Riferimenti bibliografici

- [1] G. Gini, V. Caglioti, "Robotica", ed Zanchelli, Bologna, 2003
- [2] G. Gini, A. Marchi, "Indoor robot navigation with single camera vision..", Pattern Recognition in Information Systems, ICEIS press, Portugal, 2002, pp 67-76.
- [3] P. Lima, A. Bonarini, C. Machado, F. Marchese, C. Marques, F. Ribeiro, D. Sorrenti. Omni-directional Catadioptric Vision for Soccer Robots. 2001.
- [4] G. N. DeSouza, A. C. Kak, "Vision for Mobile Robot Navigation: a survey", IEEE Trans PA&MI, 24, 2, febbraio 2002.