Università degli Studi di Padova

Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingeneria Informatica

tesi di laurea

Algoritmi per il riconoscimento e la localizzazione di oggetti in scene complesse mediante stereo camera.

Relatore: Emanuele Menegatti
Correlatore: Alberto Pretto

Laureando: Antonello Mauro

6 ottobre 2011

Sommario

Il lavoro descritto in questa tesi rappresenta lo studio ed il tentativo di dare un contributo ad uno dei settori più stimolanti e promettenti di questi anni, la *robotica autonoma*. La particolare problematica trattata è quella dell'individuazione degli oggetti e dell'identificazione della loro rotazione spaziale. Dopo aver analizzato e compreso le peculiari caratteristiche della visione artificiale nella robotica autonoma, verranno discusse brevemente le principali metodologie utilizzate in questo settore. Presa confidenza con i concetti chiave, verrà presentata una panoramica dei principali lavori apparsi nella letteratura per la classificazione di oggetti in ambito robotico. Attraverso la comprensione di questi importanti articoli verrà introdotto il contributo che questa tesi cerca di dare al settore, un algoritmo innovativo per la stima della classe e della posizione di oggetti tridimensionali. L'algoritmo in questione sfrutta i dati provenienti da una videocamera stereoscopica per costruire una struttura probabilistica in grado di modellare le caratteristiche tridimensionali degli oggetti ripresi. Verrà quindi discussa sia la creazione che l'interrogazione di tale struttura, fornendo anche dettagli specifici sull'implementazione dell'algoritmo. Tale sistema verrà quindi confrontato con uno tra i più comuni e famosi algoritmi del settore. Un'analisi delle prestazioni dell'algoritmo esposto confermerà che la soluzione ideata è una valida proposta al mondo della robotica autonoma.

Autore: Antonello Mauro

Indice

Sommario

Capitolo 1

Introduzione

Durante gli ultimi decenni, l'evoluzione delle tecnologie informatiche ha visto crescere velocemente il proprio impatto nella società, sia nella produzione di beni che nella quotidianità. Il peso di questo impatto è dovuto in gran parte alla capacità di semplificazione e automatizzazione che l'informatica ha apportato nei processi in cui è stata introdotta. La percezione comune di un buon prodotto informatico è infatti direttamente dalla capacità di quest'ultimo di essere semplice, automatizzato o, in una parola, "intelligente". La miniaturizzazione dei dispositivi di calcolo, ne ha permesso negli ultimi anni l'integrazione con una sempre più vasta gamma di prodotti, rendendo così telefoni, televisori o anche frigoriferi dei prodotti sempre più autonomi ed intelligenti. Se ci si spinge ad osservare il fronte di questa linea di pensiero, ci si rende conto che non sembra poi più tanto fantascientifica l'idea di un futuro in la presenza di robot, mobili ed autonomi, siano a supporto dell'industria, della ricerca o anche della nostra vita quotidiana. Già attualmente la robotica autonoma è oggetto di forti investimenti economici, suggerendo inoltre una forte espansione per gli anni futuri. Di seguito vengono riportati dati riguardanti alcune aree di mercato:

militare Secondo il report *"UAVs, UGVs, UUVs, and Task Robots for Military Applications"* la robotica in ambito militare vedrà una crescita di investimenti dagli 5.8 miliardi di dollari del 2010 a circa 8 miliardi nel 2016.

healthcare secondo lo studio condotto da ABI Research *"Healthcare and Medical Robots"* il mercato della robotica nell'healthcare crescerà da poco meno di 790 milioni di dollari nel 2011 a più di 1.3 miliardi nel 2016.

globale secondo un report[1] condotto direttamente dalla CIA, l'industria della robotica è stata tra i settori più promettenti dell'economia mondiale nel 2010.

[1] `https://www.cia.gov/library/publications/the-world-factbook/geos/countrytemplate_xx.html`

La nascita di nuove opportunità nell'ambito della robotica ha attirato, nel corso degli ultimi due decenni, l'attenzione di molti ricercatori, provenienti da molti settori differenti. Le sfide poste dalla creazione di robot autonomi sono infatti molto eterogenee, legano tra loro ambienti che spaziano dall'informatica alla meccanica fino allo studio dei processi cognitivi e anatomici. Si è creata così una forte cooperazione che ha generato un ambiente di ricerca molto fertile[2].

Una caratteristica peculiare della robotica autonoma è la capacità degli agenti di saper muoversi in un ambiente, interagire con disinvoltura in esso per portare a compimento determinati compiti. Per essere in grado di programmare ed eseguire i propri compiti in maniera affidabile, il robot deve però essere in grado di individuare e riconoscere gli oggetti presenti nella scena. Questa facoltà nell'uomo è così naturale da essere data per scontata, dal punto di vista informatico pone invece notevoli sfide, prima tra tutte la necessità di generalizzare e rendere robusta la percezione del robot in una vastissima serie di condizioni ed eccezioni. Il problema è reso ancor più arduo se si tiene conto delle che, nonostante gli enormi progressi in questo senso, i dispositivi mobili di computazione hanno ancora difficoltà nell'elaborare adeguatamente l'enorme quantità di informazione proveniente dai vari sensori, soprattutto quelli visivi. Al di fuori dell'ambito della robotica, la visione artificiale (*Computer Vision*) riveste da molti anni una posizione di rilievo nella ricerca scientifica, proponendo una vastissima gamma di algoritmi innovativi le cui applicazioni vanno dagli OCR alla videosorveglianza. Sebbene i temi trattati dalla Computer Vision classica e quella in ambito robotico siano estremamente affini, le particolari caratteristiche di quest'ultima hanno mantenuto molto alta la necessità di innovazione.

Proprio in questo clima di forte interesse scientifico ed economico, si colloca il lavoro descritto in questa tesi.

1.1 Obiettivi

Il sistema proposto si colloca come soluzione ad una problematica particolarmente rilevante per la robotica autonoma, l'identificazione e la localizzazione di oggetti nell'ambiente circostante. Per portare a termine i propri compiti l'uomo utilizza pesantemente il senso della vista, da questo deriva la necessita dei robot, il cui scopo è spesso quello di aiutare o sostituire l'uomo in tali compiti, di utilizzare capacità percettive simili.

Ciò che distingue la tecnica proposta in questa tesi dai comuni algoritmi di object recognition, nasce da un'attenzione particolare alle caratteristiche necessità e potenzialità della robotica autonoma. Attraverso un adeguato sviluppo dell'idea originale, il sistema sarà infatti in grado di soddisfare molti tra i principali vincoli posti dalla robotica autonoma per questo ambito:

[2] Il numero di pubblicazioni annue riguardanti la robotica autonoma, individuate utilizzando IEEE Xplore, è passato da ~330 nell'anno 2000 a ~1200 nell'anno 2010.

reazioni veloci è importante ottenere velocemente dei risultati iniziali, anche se imprecisi. Il raffinamento di tali risultati può avvenire successivamente ma il robot deve mantenere reazioni veloci.

robustezza l'algoritmo deve poter operare anche in presenza di dati di bassa qualità (bassa risoluzione, condizioni di luce ecc.).

mobilità un buon algoritmo deve sfruttare la capacità di movimento e l'odometria del robot per poter migliorare i propri risultati.

utilizzo di più sensori per aumentare la confidenza nei risultati è utile sfruttare forme addizionali di informazione, come ad esempio una videocamera stereoscopica o dei sensori di prossimità.

.

1.2 Struttura del documento

Nella prima parte della tesi (capitolo 2) verranno descritte le principali tecnologie utilizzate nel settore, in modo da introdurre gli strumenti necessari ad una migliore comprensione dell'algoritmo proposto.

Nel capitolo successivo (capitolo 3) verrà analizzato in dettaglio il problema trattato, il lettore verrà introdotto alle necessità e alle potenzialità del settore del riconoscimento di oggetti in ambito robotico. Verranno quindi analizzate le attuali risposte della ricerca scientifica a tali problemi, partendo da algoritmi generali fino alla valutazione dello stato dell'arte.

Nel capitolo 4 verrà quindi introdotta la tecnica proposta, analizzandola prima nella sua versione più intuitiva e poi nella sua immediata estensione.

Nel capitolo 6 verranno analizzati e discussi i risultati derivanti dalla sperimentazione pratica dell'algoritmo proposto.

In conclusione nel capitolo 7 verranno analizzate le potenzialità future del metodo introdotto.

Capitolo 2

Elaborazione delle immagini

Le immagini che il robot, o in generale un dispositivo di ingresso video, raccoglie dal mondo reale si presentano all'elaboratore come una matrice di pixel. Ogni pixel rappresenta un singolo punto luminoso, un frammento dell'immagine reale che, per le necessità fisiche degli elaboratori attuali, deve essere discretizzata prima di poter essere processata. Sebbene i dispositivi video installati nei robot siano in genere delle periferiche abbiano una risoluzione (qualità della discretizzazione) relativamente bassa, ognuno dei fotogrammi raccolti porta con se una quantità di dati che difficilmente gli elaboratori attuali potrebbero gestire in tempo reale. Quello che tutti gli algoritmi di riconoscimento degli oggetti devono fare, una volta ricevuta un'immagine da elaborare, è tentare di ridurre al minimo la quantità di dati associati all'immagine, evitando però di ridurne significativamente il carico di informazione utile. Se da un lato è necessario individuare e mantenere solo la porzione di dati più ricchi di informazione, dall'altro spesso è possibile che alcune informazioni siano solo intrinsecamente nelle immagini e che la loro estrazione sia possibile solo un'opportuna elaborazione. In questo capitolo saranno presentate alcune delle tecniche di maggior successo utilizzate per le problematiche descritte.

2.1 Estrazione delle feature

Nell'ambito della visione artificiale una porzione di immagine particolarmente ricca di significato, viene indicata con il termine *feature*. L'individuazione e l'estrazione delle feature stesse è una questione di fondamentale importanza per qualsiasi buon algoritmo di object recognition. Dalla qualità e quantità delle feature che si riescono ad individuare in un immagine dipende la quantità di informazione che sarà disponibile all'algoritmo. Il concetto di "significato" per una porzione di immagine è legato soprattutto alla variazione dell'intensità di colore e luce dei pixel al suo interno, ad esempio il contorno di un oggetto può risultare particolarmente significativo mentre un'area completamente bianca o nera porterà con se ben poca informazione(vedi figura 2.2) . Una volta individuati i

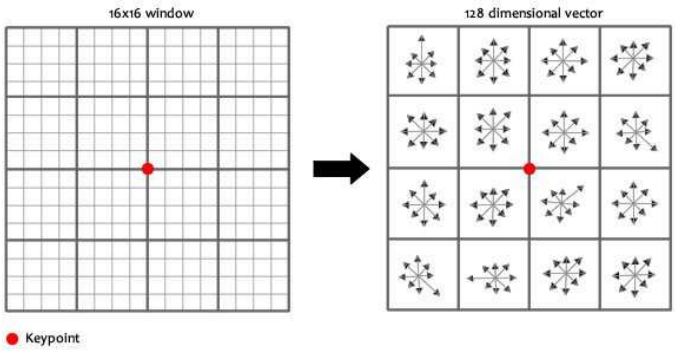

Figura 2.1: costruzione di un descrittore SIFT.

punti nell'immagine (*keypoints*) in cui la quantità di informazione è localmente maggiore, è necessario ricavare un descrittore per tale area. Un descrittore è un punto ad elevata dimensionalità (es. $\Re^{128}$ nel caso delle SIFT) in grado di identificare meglio la feature stessa. I descrittori estratti da porzioni molti simili di un'immagine dovranno inoltre fornire descrittori simili. In figura 2.1 l'orientamento del gradiente di colore attorno ad un *keypoint* viene utilizzato per la costruzione di un descrittore SIFT.

Sebbene esistano vari modi per ottenere delle feature con le caratteristiche descritte, la bontà di un metodo di estrazione va valutata nell'ottica del suo utilizzo. Nel settore del riconoscimento di oggetti i principali punti di forza di una buona feature sono i seguenti:

- il descrittore estratto per una determinata feature, deve essere il più possibile invariante a trasformazioni affini della feature. Due aree dell'immagine rappresentanti la stessa feature con orientamenti differenti, dovrebbero quindi fornire descrittori simili.

- i tempi necessari ad individuare le feature ed estrarne i descrittori devono essere il più possibile brevi.

- la feature dovrebbe essere in grado di rappresentare aree dell'immagine di varie dimensioni, a seconda di come sia distribuito il picco nel carico d'informazione riscontrato nell'area.

- oltre al descrittore, l'estrazione di una feature dovrebbe fornire anche informazioni sul luogo, l'orientamento e la scala a cui la feature è stata individuata.

In altri casi specifici è possibile siano preferibili anche altre caratteristiche, come ad esempio l'invarianza rispetto alla luminosità di una feature.

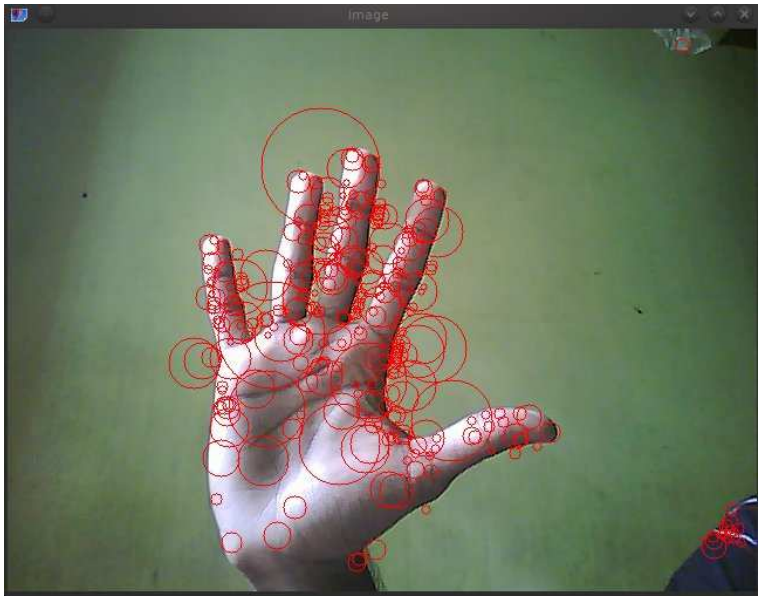

Figura 2.2: insieme di feature individuate.

La selezione del tipo di feature più adatto per un determinato scopo non è un problema scontato[1], ad esempio l'invarianza rispetto ad un determinato tipo di trasformazione potrebbe far si che un algoritmo non sia in grado di osservare alcune informazioni utili ad un problema specifico.

2.2 Elaborazione di immagini stereo

La visione stereoscopica rappresenta uno dei metodi più comuni ed economici in grado di dare ad un robot la percezione della profondità nelle immagini. Il successo della tecnica deriva in primo luogo dalla sua passività e dai suoi bassi costi: diversamente da altri metodi, come ad esempio i laser scanner, nella visione stereoscopica si utilizzano infatti solamente normali videocamere. In figura 2.3 è raffigurato il modello di videocamera stereoscopica (due canali) *Bumblebee2*, utilizzata per l'acquisizione dati nella realizzazione di questa tesi di laurea.

Sebbene esistano diverse tecniche proposte, nella letteratura della visione artificiale, mirate ad ottenere la struttura tridimensionale di una scena osservata da una o più telecamere [2] tra queste la visione stereoscopica è quella che ha riscosso maggior successo soprattutto grazie all'assenza di vincoli sulle caratteristiche degli oggetti ripresi (ad esempio la presenza o meno di oggetti in movimento

[1] per una comparazione esauriente tra vari i vari metodi, consultare la pagina web `http://computer-vision-talks.com/2011/08/feature-descriptor-comparison-report/`

[2] alcune tecniche comuni sono *shape from motion*, *shape from shading*, *shape from texture*

Figura 2.3: videocamera *Bumblebee2*.

Figura 2.4: principio di funzionamento della stereovisione.

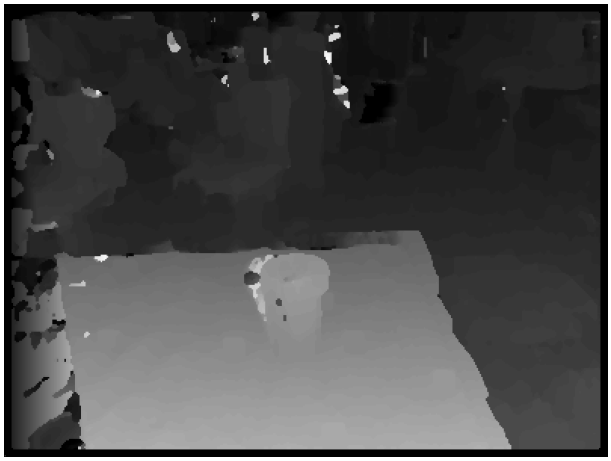

Figura 2.5: disparity map.

o di particolari condizioni di illuminazione). Il principio alla basa della visione stereoscopica, noto sin dal rinascimento, consiste in una triangolazione mirata a mettere in relazione la proiezione di un punto della scena sui due (o più) piani immagine delle telecamere (tali punti sono denominati punti *omologhi*). In figura 2.4 è possibile osservare il principio alla base della stereovisione, utilizzare lo scostamento di punti omologhi (es. sfere dello stesso colore) per triangolarne la profondità.

L'individuazione dei punti omologhi, problema noto in letteratura come il problema della corrispondenza (*correspondence problem* o *matching stereo*), consente di ottenere un valore numerico denominato disparità (*disparity*) mediante il quale, conoscendo opportuni parametri del sistema stereoscopico, è possibile risalire alla posizione tridimensionale del punto esaminato. In figura 2.5 è raffigurata la disparity map estratto da una Bumblebee2. Il problema delle corrispondenze rimane ancora aperto e produce tuttora un'ampia attività di ricerca nonostante siano stati proposti sin dagli anni innumerevoli algoritmi.

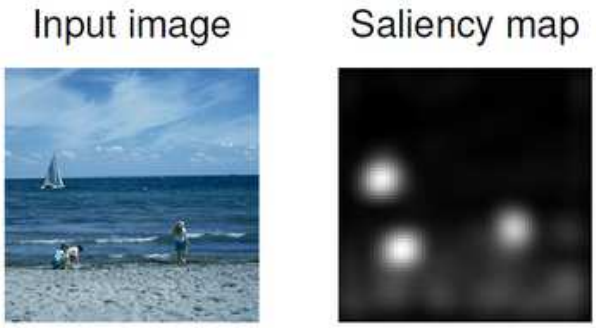

Figura 2.6: saliency map.

2.3 Individuazione di aree salienti

L'estrazione delle feature permette agli algoritmi di ridurre notevolmente la quantità di tempo e le risorse necessarie ad elaborare un'immagine, ma si occupa di estrarre i punti con maggior intensità di informazione a prescindere dal reale valore di tali informazioni per il robot. Frequentemente accade infatti che buona parte dell'area in un'immagine contenga informazioni non rilevanti negli obiettivi del robot, le feature individuate in tale area potrebbero essere scartate senza perdita di precisione. Per ridurre ulteriormente il tempo di elaborazione degli algoritmi di object recognition, spesso una buona scelta è analizzare a priori un immagine ed individuarne, approssimativamente, le aree ritenute più interessanti per lo scopo attuale del robot. Questo processo è denominato individuazione delle aree salienti (*saliency detection*) e, a partire da un'immagine di ingresso, restituisce un'immagine in scala di grigi (*saliency map*) indicante le aree di maggior interesse (figura 2.6).

Nel corso degli anni questo settore ha attirato forte attenzione nella ricerca legata alla visione artificiale, caratterizzata da una particolare alchimia che ha visto affiancate l'analisi matematica e semantica delle scene.

Capitolo 3

Tecniche di object recognition

L'importanza di algoritmi in grado di estrarre efficientemente informazioni da immagini del mondo reale è cruciale in una vasta serie di problemi comuni, basti pensare a quanto l'uomo, e moltissime altre specie, basino le proprie reazioni in risposta agli stimoli visivi. Il problema di estrarre di estrarre informazione dalle immagini risulta quindi essere molto comune nella ricerca informatica ed in particolar modo nella robotica autonoma. Proprio la necessità di eseguire in modo autonomo compiti di interesse pratico per l'uomo, fa si che i robot debbano essere in grado di gestire efficacemente un senso tanto importante quanto lo è la vista per l'uomo.

Dal punto di vista algoritmico, le soluzioni ideate a problemi legati all'analisi di immagini provenienti dal mondo reale sono tutt'altro che semplici o statiche. La quantità di dati in ingresso è molto elevata, come è elevata la loro variabilità ed il grado di intelligenza necessario ad interpretarli. Sebbene grandi sforzi siano stati dedicati al settore, nel campo della *Computer Vision*, non è quindi raro imbattersi in recenti pubblicazioni molto innovative. Proprio all'interno di questa macro area ricade il problema di riconoscimento degli oggetti (*Object Recognition*) e, scendendo ancor più nel particolare caso trattato, il riconoscimento degli oggetti in ambito robotico. L'obiettivo dell'*Object Recognition* è quello di individuare se in un'immagine sono rappresentati uno o più oggetti appartenenti ad una lista determinata a priori, in una versione leggermente meno generale del problema è importante anche fornire la posizione nell'immagine degli oggetti individuati. Proprio a quest'ultimo caso, definito *Object Recognition and Pose Estimation* fa riferimento il riconoscimento di oggetti in ambito robotico, e di conseguenza il lavoro di questa tesi.

3.1 Riconoscimento di oggetti in ambito Robotico

In natura l'importanza delle capacità degli organismi va valutata in funzione delle caratteristiche del loro ambiente nativo, questa regola generale vale anche nel nostro caso: sebbene esistano numerosi algoritmi per il riconoscimento degli

oggetti, ognuno con il proprio punto di forza, l'utilità effettiva di tali algoritmi va valutata secondo le peculiari caratteristiche dell'ambiente in cui dovranno essere utilizzati. La robotica, e in particolare la robotica autonoma, è caratterizzata da una serie di aspetti particolari, che influenzano molto le capacità richieste degli algoritmi vi operino. Di seguito verranno descritti brevemente le principali che ogni buon algoritmo utilizzato nel campo della robotica autonoma dovrebbe soddisfare:

Tempi di reazione veloci I tempi necessari ad ottenere delle stime sulla posizione e la categoria degli oggetti devono essere molto brevi: non è essenziale ottenere subito un'elevata confidenza dei risultati ottenuti, ma è importante che il robot sia in grado di reagire velocemente in base agli stimoli visivi. In questo senso sono da tenere in considerazione i limiti della potenza di calcolo disponibile nel robot.

Bassa qualità delle immagini Le immagini o i video ottenuti dalle periferiche ottiche del robot sono spesso di bassa qualità, inoltre l'acquisizione avviene in ambienti con condizioni di luce molto variabili.

Ambienti mutevoli e complessi Un robot spesso ha la necessità di operare in ambienti mutevoli, deve disporre quindi di un algoritmo di visione sufficientemente dinamico da permettergli di gestire anche forti variazioni della scena. La grande quantità di oggetti che deve essere in grado di gestire, e la necessità di aggiornare tale lista, aggiungendo nuovi oggetti incontrati con il passare del tempo, rende necessaria una buona scalabilità dell'algoritmo. Un altro comune svantaggio derivato dall'operare con immagini del mondo reale, è la necessità di saper riconoscere gli oggetti anche in diversi gradi di occlusione.

Mobilità I robot mobili autonomi, per definizione, sono avvantaggiati dalla possibilità di muoversi nell'ambiente o, se non altro, di direzionare l'ingresso video. Questo permette ai robot di aumentare, se ritenuto necessario, la confidenza dei risultati, focalizzandosi su un particolare oggetto o acquisendo immagini degli oggetti interessanti da diverse angolazioni. Un buon algoritmo dovrebbe essere in grado di basare le proprie decisioni in base alle informazioni ricevute in un certo intervallo di tempo, più che il solo singolo istante presente.

Presenza di diversi sensori Per rendere i robot più percettivi, sono spesso integrati vari tipi di sensori con cui possono analizzare l'ambiente circostante. Sebbene sia spesso difficile fare in modo che un singolo algoritmo si possa avvantaggiare dei vari modi con cui il robot percepisce l'ambiente, la robustezza che ne deriva ripaga spesso lo sforzo dedicato a tale scopo. In particolare, per quanto riguarda la visione artificiale, è opportuno avvantaggiarsi della diffusa presenza di dispositivi che permettono al robot di percepire la profondità in alcune delle aree visive inquadrate.

Alcuni casi più particolari, soprattutto i sistemi multirobot, richiedono inoltre vari gradi di cooperazione tra gli agenti. In questo caso un altro fattore determinante è la capacità di un algoritmo di avvantaggiarsi della distribuzione dell'informazione.

3.2 Tecniche comuni

Come accennato in precedenza, le tecniche utilizzate nel riconoscimento degli oggetti in ambito robotico, trovano le proprie radici nella più generica disciplina madre. La quantità e la diversità degli algoritmi proposti per il problema generico dell'*Objcect Recognition* ne rende impossibile una presente trattazione esauriente, saranno quindi analizzati solo due tra gli algoritmi si adattano meglio al modello di problematiche descritto per la robotica autonoma. Le tecniche citate sono il "*Constellation Method*", proposto da Lowe assieme ad un nuovo tipo di descrittori denominato SIFT [1], e il metodo "*Bag of Features*", proposto da Nistèr e Stewènius [2]. Per non appesantire inutilmente il presente documento, verrà fornita solo una descrizione concettuale di tali algoritmi e si lascia al lettore la possibilità di approfondire tali argomenti leggendo le pubblicazioni citate.

3.2.1 Constellation Method

Il metodo di riconoscimento degli oggetti proposto da Lowe presenta alcune interessanti caratteristiche che si inseriscono bene nel contesto della robotica. In primo luogo, l'algoritmo analizza l'immagine nella sua interezza, individuandone tutti gli oggetti allo stesso tempo ed evitando di doverla segmentare a priori. Un'altra caratteristica importante è l'utilizzo dell'algoritmo *Best-Bin-First* per ottenere una serie di risultati iniziali approssimati, questo riduce i tempi di ricerca di due ordini di grandezza nel caso di un database contenente 100.000 keypoint, con una perdita di precisione inferiore al 5%. La prima parte dell'algoritmo consiste nell'estrazione di tutti i descrittori SIFT[3] delle feature individuate nell'immagine di test, questi descrittori ricercano l'oggetto più vicino (in termini di distanza Euclidea) in un database di descrittori creato durante la fase di addestramento. In figura 3.1 si osserva come i descrittori dell'immagine di test trovino una propria corrispondenza con uno degli oggetti usati nell'addestramento come modelli.

Nel caso in cui la distanza tra il descrittore estratto dall'immagine di test e quello trovato sia superiore ad una certa soglia, il descrittore di partenza viene scartato, in caso contrario verrà aggiunto un voto corrispondente all'oggetto da cui è stato ricavato il secondo descrittore. Una volta ottenuti tutte le varie corrispondenze, viene utilizzata la trasformata generica di Hough per raggruppare le votazioni in base al tipo di oggetto di partenza e alle trasformazioni (rotazioni, scalature e traslazioni) individuate nella coppia di feature relative ai descrittori votanti. Anche se tale processo risulta essere impreciso, a causa di false cor-

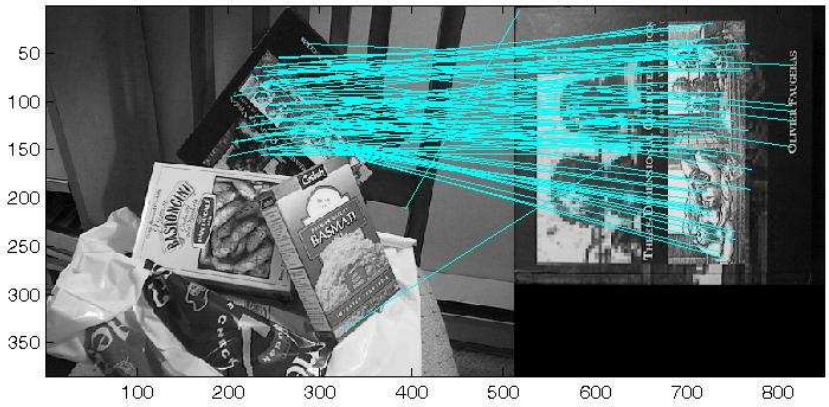

Figura 3.1: Corrispondenze tra descrittori ed i propri vicini nel modello.

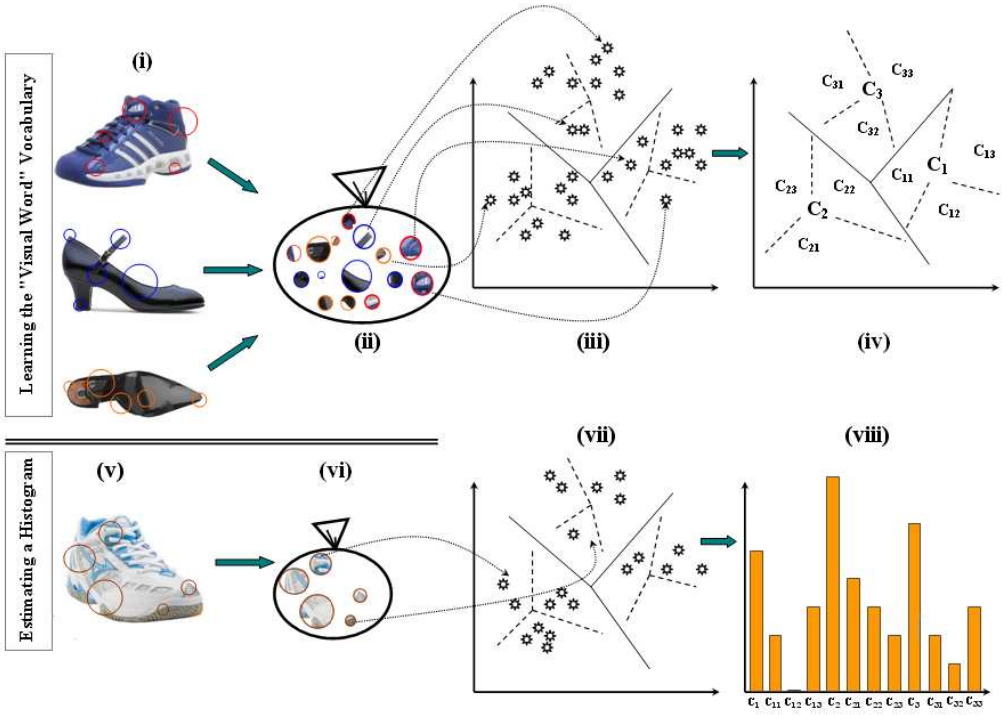

Figura 3.2: Acquisizione modello per BoF.

rispondenze, questo passaggio genere una serie di ipotesi coerenti, rimuovendo buona parte delle feature non significative. A questo punto, tutti i gruppi ottenuti, generati da almeno tre voti, possono essere processati attraverso un metodo più delicato e preciso. La fase finale consiste nell'utilizzo di *Iterative Reweighed Least Squares* per ottenere una stima migliore della trasformazione affine tra l'immagine dell'oggetto originale e quella dell'oggetto individuato nell'immagine do test.

3.2.2 Bag of Features

L'approccio proposto da Nistèr e Stewènius per il riconoscimento degli oggetti, nasce come variante di un algoritmo utilizzato per la categorizzazione di testi, in cui sono monitorate le occorrenze di certe parole all'interno dei testi già categorizzati, utilizzandole per predire il soggetto di nuovi testi. Questa tecnica è stata adattata alla classificazione di oggetti sostituendo alle parole dei descrittori locali (ad esempio le SIFT) individuati nell'immagine(*visual words*).

Durante la prima fase dell'algoritmo, detta di addestramento (*Learning*), l'algoritmo crea un dizionario(ii) con tutte le feature estratte(i). Successivamente utilizzando algoritmi di clustering(iii) crea un dizionario di taglia fissata, i cui elementi rappresentano le *visual words*. Esiste una variante a questo processo il

cui scopo è irrobustire l'algoritmo ad effetti di variazione del punto di vista o di scala. I descrittori locali possono essere utilizzati come indicatori di aree più vaste dell'immagine da utilizzare poi come parole. Alcuni autori puntualizzano, tuttavia, che l'utilizzo di un gran numero di parole casuali "piccole" risulti fornire risultati migliori rispetto ad un numero minore di parole più "grandi"[4].

L'acquisizione del modello di un oggetto, che il sistema dovrà essere in grado di riconoscere successivamente, avviene nel seguente modo. Per prima cosa vengono estratti i descrittori dalle varie immagini dell'oggetto(v). Gli elementi dell'insieme i descrittori estratti (vi) vengono quindi utilizzati per cercare nel dizionario la *visual word* che sia loro più simile(vii). Un istogramma delle occorrenze di ogni parola viene così costruito per descrivere ogni immagine(viii).

La taglia del dizionario risultante è un fattore importante, deve essere sufficientemente ampio da poter distinguere correttamente oggetti differenti e allo stesso tempo abbastanza generico da essere insensibile a piccole variazioni nei descrittori locali. Una volta ottenuti gli istogrammi per le varie immagini di addestramento, un qualsiasi algoritmo di classificazione può utilizzarli come dati in ingresso per predire la categoria dei nuovi istogrammi provenienti dalle immagini di test. Questo metodo è attraente nell'ambito del riconoscimento di oggetti in ambito robotico per la sua flessibilità al numero di oggetti riconosciuti, è infatti facile apportare alcune modifiche al dizionario per includervi nuove parole adatte a descrivere i nuovi oggetti incontrati.

3.3 Stato dell'arte

Durante gli ultimi anni, la ricerca nel settore del riconoscimento di oggetti in ambito robotico ha generato una mole considerevole di nuove idee o perfezionamenti agli algoritmi esistenti. In questo capitolo verrà data un po' di spazio ai lavori il cui impatto è stato più significativo, per quanto sia possibile farlo in modo oggettivo data l'enorme variabilità delle proposte e delle condizioni al contorno.

Curious George: An Attentive Semantic Robot

In questo articolo [5] viene data particolare enfasi al sistema di individuazioni delle aree salienti dell'immagine. Il principale contributo innovativo sta appunto nel particolare metodo con cui i dati provenienti da una stereo camera vengono utilizzati per scartare, a priori, alcune aree dell'immagine considerate poco interessanti.

Per determinare le aree salienti viene fatto uso della misura di salienza spettrale residua definita in [6], tale metrica è stata poi estesa ai colori in maniera simile a [7]. Regioni di varie dimensioni sono così individuate a formare la mappa di salienza (*saliency map*) utilizzando la tecnica *Maximally Stable Extremal Region* (MSER) [8].

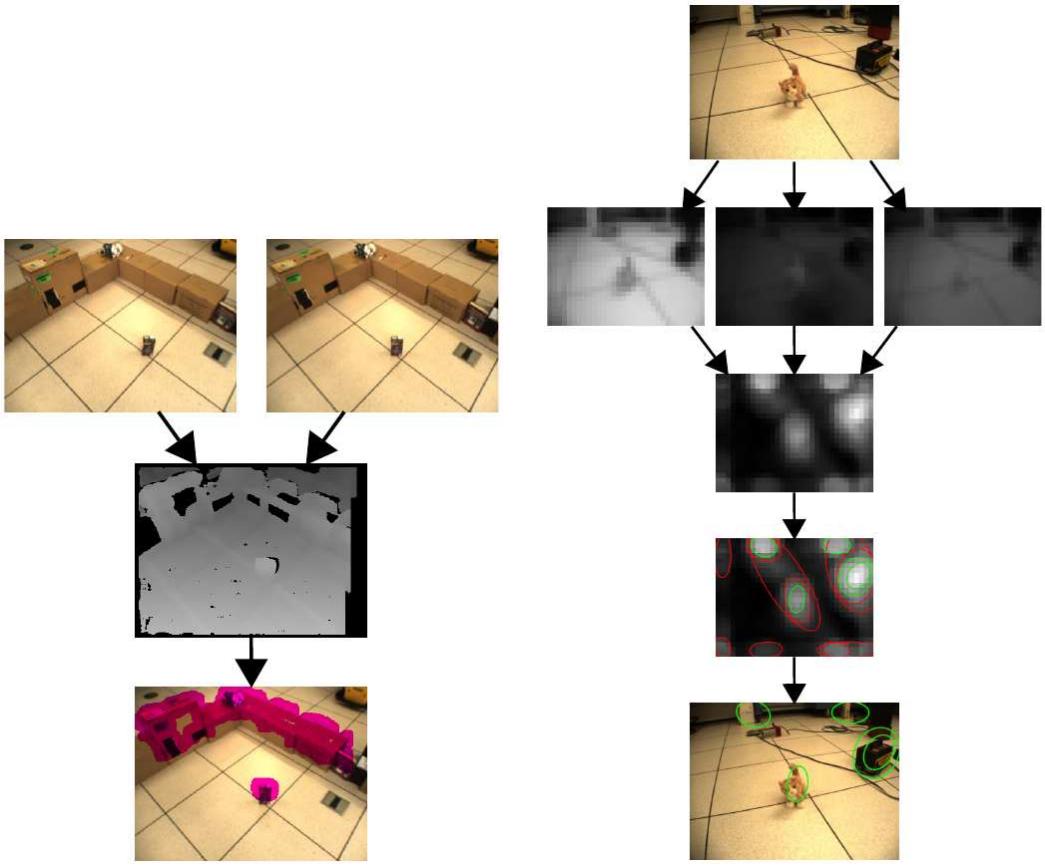

Figura 3.3: Analisi della depth-map e individuazione delle zone salienti.

Una volta ottenute le varie regioni, le meno significative vengono eliminate in base in base ad una soglia proporzionale alla salienza media dell'immagine. Sono gestiti anche i casi di regioni nidificate, in questo caso ogni regione nidificata deve avere essere almeno il 20% più piccola della regione ospite. Attraverso questa tecnica si ottengono con buona confidenza le regioni di un'immagine in cui è più probabile trovare degli oggetti interessanti, riducendo quindi il carico di elaborazione di un qualunque algoritmo di individuazione che debba processare tale immagine. L'articolo afferma che utilizzando questa tecnica l'algoritmo da loro utilizzato durante i test, incrementava di un ordine di grandezza la velocità di elaborazione di un'immagine.

Using Stereo for Object Recognition

Anche in questo articolo [9] viene utilizzata una videocamera stereografica per acquisire una mappa di profondità per la scena analizzata, in questo caso però viene descritto un metodo per utilizzarla migliorando, a posteriori, la confidenza di un risultato.

Normalmente è impossibile calcolare le reali dimensioni metriche solo in base all'area occupata dall'oggetto nell'immagine, dato che un oggetto di grandi dimensioni posto lontano dall'obiettivo può apparire identico una sua versione più piccola e vicina. Qualora si disponga di un metodo per acquisire anche la distanza dell'oggetto dalla telecamera, è possibile utilizzare tali dati per stimare le sue dimensioni reali. Il metodo descritto utilizza i dati sulla profondità forniti dalla mappa di profondità, per ottenere la distanza media nell'area dell'immagine occupata da un oggetto, quindi ne vengono calcolate le dimensioni reali, definendole supporto dell'oggetto. Durante la fase di addestramento l'algoritmo assegna ad ogni classe di oggetti un supporto, successivamente questo supporto viene confrontato con il supporto dei vari risultati ottenuti da un riconoscitore, e la confidenza di tali risultati viene diminuita in base alla loro differenza. Grazie a questa tecnica è possibile eliminare una buona parte di risultati falsi positivi.

Lo studio presentato utilizza per il riconoscimento delle informazioni sulla forma degli oggetti, e non sulla texture come solitamente accade, è tuttavia facile tradurre tale tecnica per renderla operativa anche in altri riconoscitori.

A Multi-View Probabilistic Model for 3D Object Classes

Diversamente dai precedenti, questa pubblicazione [10] focalizza la propria attenzione sulla strategia di memorizzazione e riconoscimento degli oggetti. La metodologia proposta si basa sull'apprendimento delle mutue relazioni spaziali delle principali componenti di un oggetto, a seconda delle diverse visuali. Da ogni angolazione con cui l'oggetto viene fornito nelle immagini di addestramento, vengono estratte le principali feature e viene creato un modello coerente basato sulle diverse parti dell'oggetto. Le parti così ottenute vengono quindi trattate secondo un modello probabilistico, per cercare di individuarne le relazioni spa-

Figura 3.4: Falsi positivi di tazze (sx) e scarpe (dx) individuati.

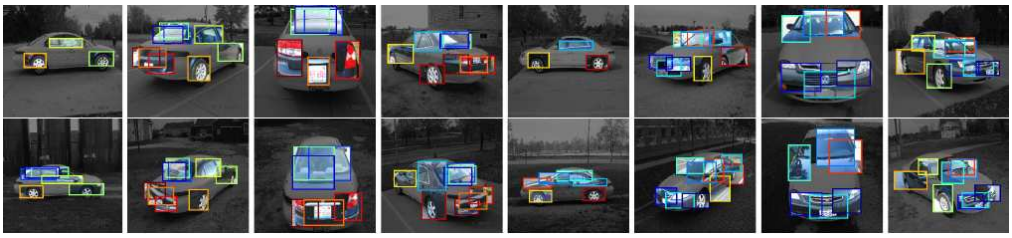

Figura 3.5: Modello costruito utilizzando le parti di un'auto.

ziali e le diverse proporzioni al variare della visuale. Questo tipo di approccio fa parte di una classe di algoritmi per l'object recognition detti generativi, il cui obiettivo è riconoscere un oggetto in base alla sua similarità con un modello costruito a partire da dei "mattoni" fondamentali (Figura 3.5). Questa classe di algoritmi da un lato fornisce un'ottima capacità di generalizzazione, permettendo l'apprendimento di grandi varietà di oggetti, dall'altro presenta prestazioni e velocità leggermente inferiori rispetto ai comuni algoritmi discriminativi.

3.4 Scelta dell'algoritmo adatto

La maggior parte degli algoritmi presenti in letteratura, focalizzano la propria attenzione solo su un sottoinsieme degli aspetti precedentemente descritti a seconda delle esigenze prioritarie del robot per cui è stato sviluppato. Spesso accade inoltre che, al variare delle circostanze, alcuni algoritmi siano più efficaci di altri, ad esempio potrebbero essere più robusti al variare della luminosità ma meno alla presenza di occlusioni. Sebbene la tendenza sia quella di creare algoritmi con buone performance sotto le più diverse condizioni, esistono tecniche semi automatiche[11] per permettere al robot di scegliere quale sia la tecnica migliore da utilizzare in un determinato momento.

Capitolo 4

Recognition in 3D: covisibilità delle feature

Il lavoro descritto in questo capitolo esplora un metodo innovativo per la creazione di una struttura probabilistica in grado di rappresentare informazioni riguardanti la struttura tridimensionale degli oggetti. Tale struttura verrà quindi integrata in un sistema di object recognition destinato all'uso nella robotica autonoma. Sebbene esistano tecniche simili in questo senso, spesso comportano la creazione di un modello tridimensionale complesso (come ad esempio quello visto nel paragrafo 3.3), conducendo in genere a tempi elevati per la generazione dei risultati. L'approccio proposto sacrifica invece parte della precisione iniziale con lo scopo mantenere tempi di risposta più rapidi, riservandosi di aumentare la confidenza dei risultati con l'acquisizione di ulteriori dati nel tempo.

L'idea di base del metodo proposto è quella di monitorare quale sia la probabilità di osservare un certo sottoinsieme delle feature di un oggetto, al variare delle angolazioni da cui viene ripreso. Una volta creata la struttura, durante il processo di riconoscimento è possibile utilizzare tale struttura per stimare, dato un insieme di feature osservate nella scena, la probabilità che un oggetto sia presente in una sua visuale. Approcci simili, basati sulla visibilità di un gruppo di feature, sono stati proposti in letteratura in vari ambiti [10, 12, 13] ma sono poco affini all'ambito robotico, soprattutto per le loro restrittive condizioni di operabilità (ad esempio la necessità a priori di un modello 3D, o di pattern ben conosciuti). L'algoritmo proposto cerca inoltre di estendere il concetto esaminando non tanto singole feature, bensì considerandole a coppie ed esaminandone anche la mutua distanza. Sfruttando i dati provenienti da una mappa di profondità (ottenuta mediante visione stereoscopica) viene dunque individuata la distanza reale tra le feature di una coppia, questa viene poi utilizzata per migliorare la confidenza dei risultati (concetto simile a quanto visto nel paragrafo 3.3).

Il modello proposto è stato inoltre strutturato per permettere un agile rafforzamento delle ipotesi generate introducendo progressivamente nuove immagini

ottenute dal robot negli spostamenti.

4.1 Creazione di una struttura per la ricerca

Il modello probabilistico per l'analisi la visibilità delle feature prevede una fase iniziale di addestramento, in essa avviene la costruzione di una struttura da utilizzare poi nella fase di riconoscimento. I dati necessari alla costruzione del modello sono ricavati da uno studio iniziale dell'oggetto che il robot deve imparare a riconoscere. In questa fase il robot "osserva" l'oggetto in questione, ruotandolo in varie angolazioni di vista ed ottenendone una serie di immagini ad angoli di rotazione conosciuti. L'algoritmo estrae così tutte le feature individuate assieme ad alcune informazioni addizionali, tali informazioni comprendono:

- (ϕ, θ): angoli di rotazione della visuale da cui è stato ripreso l'oggetto.
- (x, y, d): posizione (nell'immagine) della feature estratta, in riferimento all'oggetto. Il valore d è ottenuto dalla mappa di profondità alle coordinate (x, y).
- s: scala della feature estratta, pesata in base alla distanza dell'oggetto (dato ottenuto dalla mappa di profondità).
- α: angolo di rotazione della feature rispetto all'immagine.
- *type*: classe (e sottoclasse) dell'oggetto ripreso.

Prima di proseguire con l'analisi di una delle viste, l'algoritmo ricerca all'interno di un proprio dizionario le feature (descrittori) più simili a quelle trovate nell'immagine analizzata. Il processo di creazione del dizionario come quello di ricerca sono simili a quanto visto nel paragrafo 3.2.2 per la *Bag of Feature*.

Di seguito verrà proposto un modello semplificato per l'analisi della visibilità delle singole feature che verrà poi esteso per monitorarne le coppie.

Singole feature

Data un'immagine per una generica vista dell'oggetto, l'algoritmo utilizzerà le seguenti informazioni nella costruzione della struttura:

- $c_1, ..., c_n$: lista dei descrittori, presenti nel dizionario, che sono risultati essere i più vicini ai descrittori relativi alle feature trovate nell'immagine analizzata.
- (ϕ, θ): angoli di rotazione della visuale da cui è stato ripreso l'oggetto.

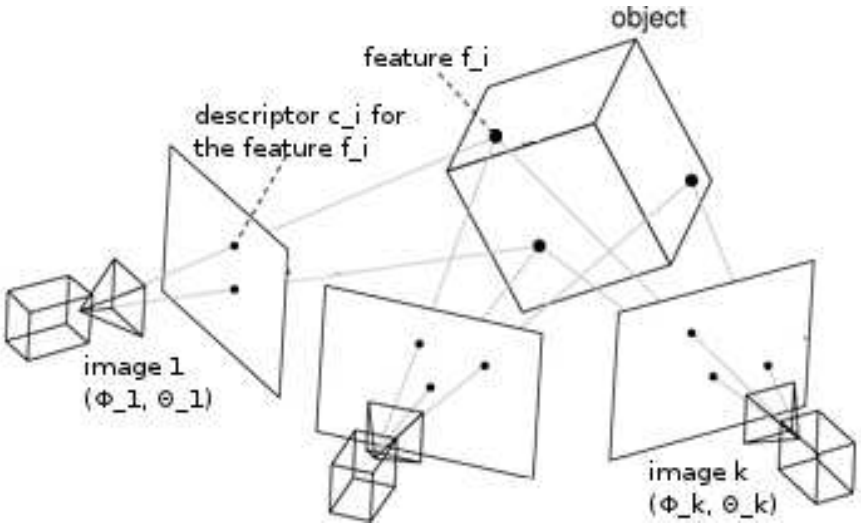

Figura 4.1: visibilità delle feature.

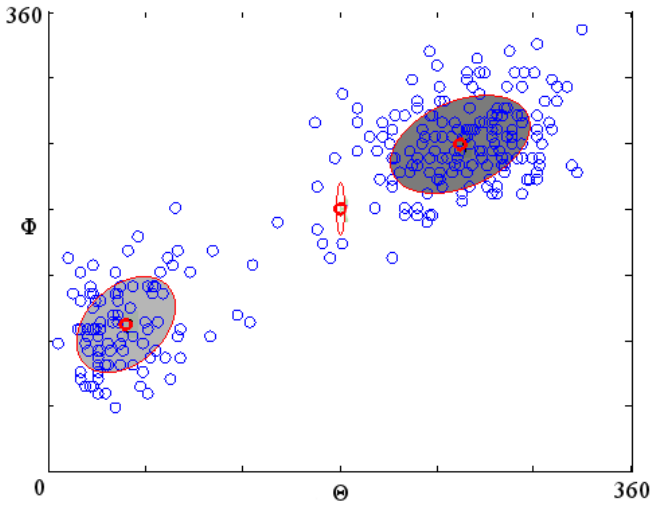

Figura 4.2: Struttura di visibilità per un descrittore c_i.

Utilizzando questi dati l'algoritmo crea una lista dei punti (ϕ, θ), queste coordinate rappresentano gli angoli in cui un generico descrittore c_i risulta visibile per l'oggetto analizzato (vedi figura 4.1).

Una volta completato lo studio dell'oggetto, la lista dei punti ottenuta per il descrittore c_i viene processata utilizzando un algoritmo di *clustering* con lo scopo di individuare le aree nello spazio (ϕ, θ) in cui il descrittore risulta essere più visibile. Siano $V_1, ..., V_k$ gli insiemi (*cluster*) di punti di maggior densità individuati nello spazio di visibilità di c_i (figura 4.2). L'algoritmo memorizza nella propria struttura una gaussiana per ognuno degli insiemi trovati, con media $\boldsymbol{\mu} = [\mu_\phi, \mu_\theta]^T$ e deviazione $\boldsymbol{\sigma^2} = [{\sigma_\phi}^2, {\sigma_\theta}^2]^T$ pari a quelle dei punti nel cluster. Indicando con S la struttura creata e con $G(\boldsymbol{\mu}, \boldsymbol{\sigma^2})$ una generica gaussiana, si avrà che:

$$S(c_i) = \{G_1(\boldsymbol{\mu_1}, \boldsymbol{\sigma_1}^2), ..., G_k(\boldsymbol{\mu_k}, \boldsymbol{\sigma_k}^2)\}$$

Ognuna di queste gaussiane porterà quindi con sé informazioni riguardanti la probabilità di osservare una feature simile a quella che ha generato c_i, al variare della visuale dell'oggetto ripreso. Il modello probabilistico creato è di fatto una mistura gaussiana (*Gaussian Mixture Model*), questa metodo di rappresentazione dell'informazione è spesso usato con successo in vari altri ambiti, come ad esempio il riconoscimento vocale[14].

Coppie di feature

Per modellare la probabilità che due feature siano covisibili, l'algoritmo utilizza sfrutta una quantità maggiore delle informazioni ottenute durante l'osservazione

dell'oggetto, queste informazioni comprendono:

- $c_1, ..., c_n$: lista dei descrittori, presenti nel dizionario, che sono risultati essere i più vicini ai descrittori relativi alle feature trovate nell'immagine analizzata.
- $(x_1, y_1, d_1), ..., (x_n, y_n, d_n)$: lista delle posizioni nell'immagine in cui sono state individuate le feature che hanno generato $c_1, ..., c_n$. d_i rappresenta la profondità individuata nel punto di estrazione (x_i, y_i).
- (ϕ, θ): angoli di rotazione della visuale da cui è stato ripreso l'oggetto.

A questo punto, al posto di studiare la visibilità delle singole feature, l'algoritmo procede selezionandone coppie in modo casuale. Per ogni coppia viene stimata la distanza tra le coordinate di estrazione delle due feature, i dati ottenuti vengono quindi utilizzati per la costruzione dello spazio di visibilità. I dati ottenuti dal passaggio descritto sono i seguenti:

- $(c_{i_1}, c_{j_1}), .., (c_{i_k}, c_{j_k})$: lista delle k coppie di descrittori, selezionate casualmente utilizzando elementi di $c_1, ..., c_n$.
- $l_1, ..., l_k$: lista delle distanze individuate tra le feature generatrici delle varie coppie di descrittori. In generale, la distanza l_i, ottenuta dalla coppia (c_{i_i}, c_{j_i}) è calcolata come segue:

$$l_i = \sqrt{(x_{i_i} - x_{j_i})^2 + (y_{i_i} - y_{j_i})^2} * \frac{d_{i_i} + d_{j_i}}{2}$$

  La distanza euclidea tra i punti di estrazione è pesata in base profondità dell'immagine in quei punti, in modo da ottenere un valore indipendente dalla distanza dell'oggetto dall'obiettivo.

Per ognuna delle coppie (c_i, c_j) l'algoritmo memorizza la lista dei punti (ϕ, θ, l) in cui è stata individuata. Similmente a quanto visto per le singole feature, la lista di punti viene processata mediante clustering, ottenendo una serie di cluster di maggior densità. Questi cluster, con media $\boldsymbol{\mu} = [\mu_\phi, \mu_\theta, \mu_l]^T$ e deviazione $\boldsymbol{\sigma^2} = [{\sigma_\phi}^2, {\sigma_\theta}^2, \sigma_l]^T$, sono quindi utilizzati (come visto in precedenza per c_i) per memorizzare una lista di gaussiane di covisibilità per la coppia (c_i, c_j) (figura 4.2).

$$S(c_i, c_j) = \{G_1(\boldsymbol{\mu_1}, \boldsymbol{{\sigma_1}^2}), ..., G_k(\boldsymbol{\mu_k}, \boldsymbol{{\sigma_k}^2})\}$$

4.2 Analisi della struttura di ricerca

Una volta ultimata, la struttura di covisibilità viene utilizzata durante la fase di riconoscimento, per stimare la confidenza e la rotazione di un possibile oggetto individuato.

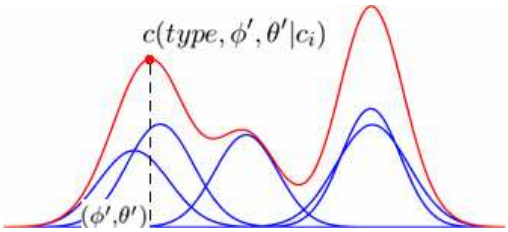

Figura 4.3: valutazione nella mistura del descrittore c_i.

Come fase preliminare la scena viene analizzata con la versione semplificata di uno tra gli algoritmi visti per il riconoscimento di oggetti (es. *Constellation Method*). Da questa analisi vengono ricavate una serie di ipotesi iniziali sulla presenza e la posizione di oggetti nell'immagine. Una volta individuato un possibile risultato l'algoritmo proposto analizza le feature che hanno originato tale risultato, per migliorarne la confidenza e stimarne la rotazione. Di seguito saranno viste più in dettaglio le principali componenti di questa fase, prima per l'approccio con una singola feature ed infine quello per le coppie.

Singole feature

Successivamente all'elaborazione di una scena da parte di un algoritmo di object recognition, vengono prese in esame le singole ipotesi di risultato. Per ognuna di queste viene identificato l'insieme di feature che ha contribuito alla creazione dell'ipotesi. L'algoritmo utilizzerà le seguenti informazioni nell'interrogazione della struttura di visibilità:

- $c_1, ..., c_n$: lista dei descrittori, presenti nel dizionario, che sono risultati essere i più vicini ai descrittori delle feature per il risultato analizzato.

Diversamente dalla fase di creazione della struttura, la coppia di angoli di rotazione (ϕ, θ) del risultato sono ancora incognite. Per ognuno dei descrittori c_i l'algoritmo procede consultando nella struttura di visibilità relativa al tipo ipotizzato S_{type}. Dalla struttura viene esaminata la lista di gaussiane relative alla visibilità di tale descrittore (figura 4.2). La confidenza apportata da ogni descrittore c_i, all'ipotesi che l'oggetto cercato sia del tipo corretto e sia ruotato di (ϕ', θ'), viene indicata con $c(type, \phi', \theta'|c_i)$ ed è proporzionale alla valutazione, nel punto (ϕ', θ'), della somma delle gaussiane individuate (figura 4.3).

$$c(type, \phi', \theta'|c_i) = k * \sum_{G_k \in S_{type}(c_i)} G_k|_{(\phi', \theta')}$$

Dove con $G|_{(\phi,\theta)}$ si intende

$$G(\boldsymbol{\mu}, \boldsymbol{\sigma^2})|_{(\phi,\theta)} = e^{-(\frac{(\phi-\mu_\phi)^2}{2*{\sigma_\phi}^2} + \frac{(\theta-\mu_\theta)^2}{2*{\sigma_\theta}^2})}$$

La confidenza finale per l'ipotesi studiata sarà calcolata nel seguente modo.

$$c(type, \phi', \theta') = \sum_{i=1}^{n} c(type, \phi', \theta' | c_i) * c(c_i)$$

Dove $c(c_i)$ rappresenta la confidenza del descrittore c_i per l'oggetto, proporzionale alla probabilità del descrittore di essere osservato nell'oggetto.

Coppie di feature

Questa versione dell'algoritmo è molto simile alla precedente, ma sfrutta alcune informazioni aggiuntive per migliorare le confidenze ottenute. Per ogni ipotesi viene identificato l'insieme di feature che ha contribuito alla creazione dell'ipotesi. Procederà poi come già visto nel paragrafo riguardante la creazione della struttura per ottenere le seguenti informazioni:

- $(c_{i_1}, c_{j_1}), .., (c_{i_k}, c_{j_k})$: lista delle k coppie di centroidi, selezionate casualmente da $c_1, ..., c_n$.
- $l_1, ..., l_k$: lista delle distanze individuate tra le feature generatrici delle varie coppie di centroidi. In generale, la distanza l_i, ottenuta dalla coppia (c_{i_i}, c_{j_i}) è calcolata come segue:

$$l_i = \sqrt{(x_{i_i} - x_{j_i})^2 + (y_{i_i} - y_{j_i})^2} * \frac{d_{i_i} + d_{j_i}}{2}$$

Dato che ognuna delle coppie (c_i, c_j) può essere individuata più volte, l'algoritmo calcola la media μ_l e la varianza σ_l delle lunghezze l individuate per le varie coppie. Per ognuna delle coppie distinte individuate viene quindi creata una gaussiana G_H con media $\boldsymbol{\mu} = [0, 0, -\mu_l]^T$ e deviazione $\boldsymbol{\sigma^2} = [0, 0, \sigma_l]^T$. In pratica viene costruita una mistura gaussiana molto semplificata (un elemento e con varianza solo sula dimensione l), concettualmente simile alla struttura creata durante la fase di apprendimento. In figura 4.4 viene rappresentata la mistura ottenuta per (c_i, c_j) e il punto (media varianza nulla in θ) relativo a G_H, vengono considerate le sole dimensioni l ed θ per motivi di chiarezza nella visualizzazione.

Ricercando il punto di massima somiglianza tra i modelli degli oggetti conosciuti e il modello corrente, è possibile determinare sia una stima della classe che della posizione per ognuno dei potenziali oggetti osservati. Per rendere estremamente veloce questa ricerca, l'algoritmo proposto utilizza una una nota tecnica di super imposizione, la convoluzione [15].

L'algoritmo procede quindi consultando la struttura di visibilità del tipo ipotizzato S_{type}, ottenendo la lista di gaussiane relative alla visibilità delle varie coppie di descrittori. Per ognuna delle gaussiane G_S trovate nella struttura per coppia (c_i, c_j), l'algoritmo calcola la convoluzione con G_H.

Un'importante proprietà delle gaussiane è la loro chiusura rispetto all'operatore di convoluzione, in particolare $G_1(\mu_1, {\sigma_1}^2) \circledast G_2(\mu_2, {\sigma_2}^2) = G(\mu_1 + \mu_2, {\sigma_1}^2 +$

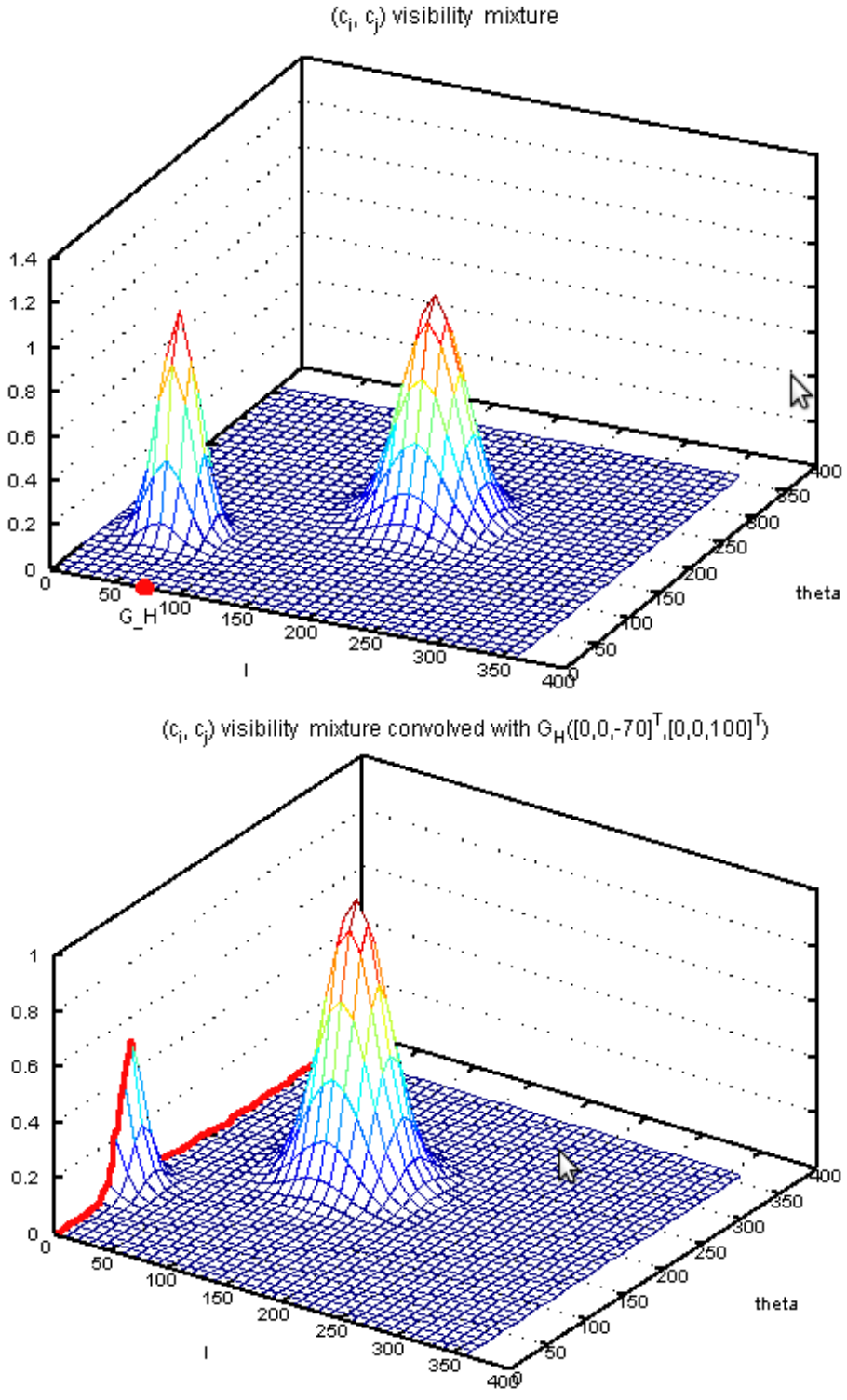

Figura 4.4: calcolo della confidenza per (c_i, c_j).

${\sigma_2}^2$). La chiusura delle gaussiane rispetto alla convoluzione semplifica di molto il calcolo delle gaussiane risultanti, riducendo di fatto il tutto a semplici somme tra i parametri. Data una coppia individuata (c_i, c_j) e la relativa gaussiana costruita G_H, la confidenza apportata all'ipotesi che l'oggetto cercato sia del tipo corretto e sia ruotato di (ϕ', θ'), viene indicata con $c(type, \phi', \theta' | c_i, c_j, G_H)$ ed è proporzionale alla valutazione, nel punto $(\phi', \theta', 0)$, della somma delle gaussiane risultanti dalle convoluzioni. In figura 4.4 è illustrata la curva(in rosso) relativa ai valori di confidenza ottenuti, per motivi di visualizzazione viene considerato solo l'asse θ.

$$c(type, \phi', \theta' | c_i, c_j, G_H) = k * \sum_{G_k \in S_{type}(c_i, c_j)} (G_H \circledast G_k)|_{(\phi', \theta', 0)}$$

Anche in questo caso, la confidenza totale per l'ipotesi viene calcolata come somma pesata delle confidenze di ogni descrittore, analogamente a quanto visto per le singole feature.

4.3 Vantaggi e problematiche

Di seguito verranno descritti alcuni tra i principali vantaggi e svantaggi comportati dall'integrazione dell'algoritmo in un sistema di object recognition.

4.3.1 Vantaggi

Modello probabilistico

Sebbene il metodo proposto possa sembrare laborioso offre il vantaggio di poter sfruttare informazioni sulla struttura di un oggetto interamente dal punto di vista probabilistico. Il vantaggio di un modello probabilistico è che il tempi di analisi possono essere notevolmente minori rispetto ad altri metodi basati sul confronto di modelli tridimensionali.

Dizionario di taglia limitata

Un altro vantaggio rispetto alle problematiche della robotica autonoma (vedi paragrafo 3.1), è rappresentato dal buon funzionamento dell'algoritmo in presenza di dizionari limitati, leggeri e veloci da analizzare. Questa caratteristica è dovuta alla necessità di monitorare la visibilità dei descrittori più "vicini" ad ogni feature al variare della rotazione dell'oggetto: la probabilità che il descrittore più "vicino" ad una feature non cambi durante la rotazione aumenta con il diminuire della taglia del dizionario. L'utilizzo di un dizionario di taglia limitata riduce inoltre il numero delle possibili coppie di descrittori, diminuendo la quantità di informazione che ogni struttura deve memorizzare.

Robustezza delle coppie di feature

L'utilizzo di coppie di feature, intese quasi come fossero una macro-feature, permette i seguenti vantaggi:

- la presenza in un'ipotesi di risultato di una coppia di feature già incontrata in quell'oggetto durante l'addestramento, è un indice più forte rispetto alla presenza delle singole feature. Questo concetto è ulteriormente rafforzato se si tiene conto anche della corrispondenza tra la distanza mutua delle feature.

- tenere conto della distanza tra le feature permette di rendere la macro-feature, rappresentata dalla coppia, non invariante rispetto alla rotazione dell'oggetto. Al variare della visuale infatti, sebbene l'aspetto delle feature cambi relativamente poco, lo fanno le loro distanze mutue. Il vantaggio è di ottenere una precisione maggiore nel riconoscere l'angolo dell'oggetto indicato dall'individuazione dalla coppia di feature[16].

Scalabilità

Nell'ottica di utilizzo su larga scala, dal punto di vista della scalabilità il sistema proposto offre delle buone premesse. In primo luogo la suddivisione delle strutture di in base agli oggetti conosciuti, permette la veloce immissione ed eliminazione di nuovi modelli di oggetti. Questo permetterà non solo ad un robot di imparare a riconoscere velocemente nuovi oggetti, ma di scambiare con altri individui la conoscenza di un determinato oggetto conosciuto. In secondo luogo, l'utilizzo del modello probabilistico descritto permette di evitare la memorizzazione di pesanti modelli tridimensionali, diminuendo l'impatto sulle prestazioni derivanti dall'apprendimento di numerosi oggetti. Un aspetto che questo progetto si pone di studiare, è la creazione di un sistema di analisi gerarchica dell'insieme di modelli conosciuti. Utilizzando l'idea di base proposta in [17] per la suddivisione dello spazio di un'immagine, si valuterà la suddivisione dello spazio di ricerca nelle misture gaussiane in base alle probabilità di ottenere superimposizioni(convoluzioni) migliori nei sottoinsiemi partizionati.

4.4 Problematiche

Scelta della taglia del Dizionario

Sebbene sia vantaggioso per il modello proposto l'utilizzo di un dizionario di taglia limitata, questo rende meno precisi gli algoritmi di object recognition utilizzati per la generazione delle ipotesi di oggetto iniziali. La scelta di un giusto compromesso spesso dipende da oggetto ad oggetto, rendendo difficoltosa l'individuazione di un valore generalmente buono.

Selezione delle coppie di feature

Anche in presenza di dizionari limitati il numero di possibili coppie è quadratico rispetto alla taglia del dizionario, la memorizzazione dello spazio di visibilità di ogni coppia potrebbe in alcuni casi essere troppo oneroso. Sebbene spesso, per ottenere buoni risultati, sia sufficiente monitorare un sottoinsieme delle coppie di feature di un oggetto, la selezione intelligente di tali coppie potrebbe rivelarsi difficoltosa.

Individuazione dei punti di massima confidenza

La valutazione di una mistura gaussiana per individuarne i punti di massimo, senza l'utilizzo di particolari accorgimenti, potrebbe rivelarsi un problema estremamente oneroso dal punto di vista computazionale. Esistono in questo senso numerose tecniche che possono alleggerire il processo, da quelle di intelligenza artificiale (ad esempio la *Swarm Intelligence*) ad altre più analitiche (ad esempio *Expectation Maximization*).

Capitolo 5

Implementazione

Per la realizzazione di un sistema di object recognition basato sull'algoritmo descritto nel capitolo 4 si è scelto di adottare un approccio base piuttosto comune in letteratura, basato sulla tecnica *Constellation Method* (vedi paragrafo 3.2.1). L'unica modifica rilevante riguarda la creazione del database di descrittori, si è scelto di utilizzare un dizionario creato in maniera simile a quanto visto per l'approccio *Bag of Features* (vedi paragrafo 3.2.2). Le immagini da cui sono stati ricavati i descrittori per la creazione del dizionario sono state selezionate tra immagini generiche, non di oggetti, per mantenere generalità ed evitare overfitting. L'implementazione è rivolta soprattutto al confronto di prestazioni rispetto all'algoritmo base, è possibile tuttavia integrare l'algoritmo proposto anche in altri sistemi in modo da sfruttare eventuali potenzialità o migliorie apportate da questi.

5.1 principali fasi

Di seguito saranno riportate in dettaglio le scelte implementative che hanno portato alla realizzazione pratica dell'algoritmo. La sezione verrà suddivisa rispecchiando quelle che sono alcune tra le fasi più comuni negli algoritmi di object recognition:

- **Preprocessing**: creazione del dizionario.
- **Addestramento**: creazione della struttura di ricerca.
- **Riconoscimento**: valutazione di una scena tramite struttura di ricerca.

5.1.1 Creazione del dizionario

Seguendo la traccia indicata dall'approccio *Bag of Features*, viene selezionato un insieme di immagini di scene generiche, da esse vengono estratte tutte le feature

(SIFT) ed i relativi descrittori. Tutti i descrittori così ottenuti vengono partizionati tramite k-means, i centroidi delle varie partizioni sono quindi utilizzati come parole del dizionario.

5.1.2 Creazione della struttura di ricerca

Nell'approccio base *Constellation Method* viene creata un'unica struttura, per la memorizzazione delle informazioni sulle feature estratte, nel nostro caso invece sarà considerata anche la creazione delle strutture di covisibilità. L'idea per la realizzazione e la ricerca sul dizionario, sono state prese direttamente dal lavorodi David Nistèr e Henrik Stewènius in [2].

Struttura di base

In primo luogo viene creata una struttura efficiente per la ricerca nel dizionario. Data la necessità di utilizzare spesso la ricerca tramite KNN *K-Nearest Neighbor*, i descrittori del dizionario sono stati inseriti in un *KD-Tree*.

L'algoritmo analizza, una alla volta, le immagini degli oggetti su cui viene addestrato, ogni immagine viene fornita unitariamente ad informazioni sul tipo dell'oggetto contenuto e alla visuale da cui è stato ripreso. L'algoritmo estrae da ogni immagine tutte le feature (SIFT) e per ognuna ricava il relativo descrittore. Una volta ottenuto il descrittore di una feature, viene ricercata nel dizionario la parola (descrittore centroide) la cui distanza euclidea risulta essere minore (Nearest Neighbor), associandole informazioni sulla feature appena estratta. L'algoritmo mantiene per ogni parola del dizionario una lista lista con le informazioni di tutte le feature che sono risultate essergli più vicine, tali informazioni comprendono:

- (ϕ, θ): angoli di rotazione della visuale da cui è stato ripreso l'oggetto che ha generato la feature.
- (x, y, d): posizione (nell'immagine) della feature estratta in riferimento all'oggetto, d è il valore ottenuto nella mappa di profondità alle coordinate (x, y).
- s: scala della feature estratta, pesata in base alla distanza dell'oggetto (dato ottenuto dalla mappa di profondità).
- α: angolo di rotazione della feature rispetto all'immagine che ha generato la feature.
- *type*: classe (e sottoclasse) dell'oggetto ripreso nell'immagine che ha generato la feature.

  Durante l'estrazione delle feature viene analizzata anche la rilevanza di una feature nel discriminare un certo oggetto. Tanto più una determinata

feature è individuata in un singolo oggetto, tanto più la sua presenza in una scena sarà caratterizzante per la presenza di tale oggetto. Calcolando quindi la presenza totale della feature F_i e la sua presenza nelle immagini relative all'oggetto O_j, si può calcolare la seguente probabilità:

$$P(F_i|O_j) = \frac{P(F_i \cap O_j) * P(F_i)}{P(O_j)}$$

Invertendo la condizione utilizzando il Teorema di Bayes si ottiene:

$$P(O_j|F_i) = \frac{P(F_i|O_j) * P(O_j)}{P(F_i)}$$

Questo valore di probabilità rappresenta la rilevanza della feature F_i per l'oggetto O_j.

Struttura di covisibilità

Viene creata una struttura per ogni oggetto separatamente utilizzando gli insiemi di feature estratti dalle immagini di addestramento. Quando l'immagine di una nuova vista per un oggetto viene analizzata, l'insieme di feature viene passato alla struttura dell'oggetto corrispondente, che provvederà a campionarne le coppie di feature per poi analizzarle come descritto nel capitolo 4. Il clustering dei punti in cui una coppia di feature è risultata visibile è effettuato mediante l'utilizzo di *Agglomerative Mean-Shift*[18], sebbene la tecnica risulti essere meno efficiente di k-means, rispetto a quest'ultima mean-shift non necessita di una conoscenza a priori del numero di cluster da individuare.

5.2 Analisi della struttura di ricerca

Durante la prima parte di questa fase l'algoritmo la tecnica *Constellation Method* per generare delle prime ipotesi di risultati. Ognuna delle feature individuate nella scena vota, utilizzando KNN, per K possibili parole nel dizionario. Ogni parola porta con se una lista degli oggetti in cui è stata rilevata, e per ognuno di essi verrà generato un voto distinto. Ogni votazione ha un peso in base alla distanza della feature dal vicino individuato e alla sua rilevanza per l'oggetto votato. Le votazioni così ottenute sono partizionate a seconda dell'oggetto votato, per ognuno dei voti viene poi ricostruita la posizione nell'immagine dell'oggetto votato. La ricostruzione della stima della posizione di un risultato utilizza le informazioni addizionali associate all'oggetto votato e quelle associate alla feature votante, il procedimento usato è il seguente:

- viene calcolato il rapporto s tra le scale delle feature individuate.
- viene calcolata la rotazione $\triangle$ α, sottraendo la rotazione della feature individuata da quella presente nelle informazioni per l'oggetto votato.

- viene individuato il punto (x, y) votato, traslando, ruotando e scalando opportunamente la posizione (x_v, y_v) indicata nelle informazioni dell'oggetto votato. Sia (x_f, y_f) la posizione della feature votante.

$$x = x_f - (x_v * s * cos(\triangle\ \alpha) - y_v * s * sin(\triangle\ \alpha))$$

$$y = y_f - (x_v * s * sin(\triangle\ \alpha) + y_v * s * cos(\triangle\ \alpha))$$

A questo punto viene utilizzato Agglomerative Mean-Sift per partizionare i voti di un oggetto in base alla posizione (x, y) e alla rotazione α del risultato votato. Dopo questo passaggio si ottengono una lista delle prime bozze di risultati, per completare le feature che hanno originato i vari risultati vengono analizzate utilizzando la struttura di covisibilità (vedi paragrafo 4.2).

5.3 Motivazioni di scelte pratiche

In letteratura esistano numerose tecniche con la capacità di risolvere le problematiche incontrate, la scelta delle tecniche utilizzate è motivata nei paragrafi seguenti.

Utilizzo delle SIFT

La scelta di utilizzare le SIFT è stata dettata soprattutto dalla loro stabilità ed efficienza implementativa. L'algoritmo proposto è stato testato anche utilizzando più recente e performante medoto, le ORB (*Oriented BRiefs*), a causa di problemi implementativi nelle librerie utilizzate, sono risultate essere inadatte allo scopo.

Utilizzo di Agglomerative Mean-Shift

Questo algoritmo di clustering gode del fatto di non necessitare di un numero di cluster noto a priori, un fattore determinante per le necessità dell'algoritmo implementato. Sono state fatti vari test utilizzando anche *Hierarchical K-Means*[1] ma anche in questo caso limitazioni implementative hanno volto a favore ad una versione modificata di Agglomerative Mean-Shift. Le modifiche apportate hanno permesso di considerare, nella variazione del centroide di ogni cluster, anche un peso relativo ai punti da partizionare, questo ha migliorato sensibilmente la qualità dei cluster ottenuti.

[1] una versione particolare di K-Means in cui il numero di cluster può variare dinamicamente, dividendo ripetutamente i cluster ottenuti se l'operazione risulta diminuire la varianza delle partizioni ottenute rispetto a quella base.

Capitolo 6

Risultati

Per confrontare l'algoritmo proposto sono stati realizzati tre differenti sistemi, questi comprendono un sistema di confronto, un sistema che sfrutta la struttura di visibilità delle singole feature ed infine il sistema con la struttura di covisibilità (coppie di feature). La realizzazione finale dell'algoritmo ha richiesto il confronto di numerose diverse varianti implementative, per questioni di equità di condizioni, per l'implementazione di tutti i sistemi si è però preferito appoggiarsi il più possibile allo stesso ambiente e alle stesse librerie.

6.1 Ambiente di testing

Framework

La scelta di utilizzare il C++ come linguaggio di sviluppo, unitamente alla natura delle problematiche incontrate, hanno reso una scelta quasi naturale l'utilizzo OpenCV[19] come framework di base per i tre sistemi. La scelta di utilizzare OpenCV garantisce un'elevata efficienza nei metodi di Computer Vision integrati, sebbene in alcuni casi sia risultato essere poco flessibile e parametrizzabile (in particolare nel modulo di estrazione delle ORB feature).

Acquisizione immagini

Il sistema è stato progettato per l'utilizzo in un robot dotato di videocamera stereoscopica, con questo obiettivo è stata implementato un sistema di acquisizione video ed estrazione della depth-map utilizzando la stereo camera Bumblebee2[1]. Sebbene la telecamera sia stata fornita associata a librerie per acquisire video e depth-map, le loro performance pratiche sono risultate essere piuttosto insoddisfacenti. Per questo motivo è stato implementato anche un metodo di calibrazione e acquisizione indipendente, basato su OpenCV (vedi paragrafo 8.1). La realizzazione di entrambe i sistemi di acquisizione è stata caratterizzata

[1] http://www.ptgrey.com/products/bumblebee2/bumblebee2_stereo_camera.asp

da una serie di difficoltà: nonostante la fervente ricerca nel campo della visione stereoscopica, risulta ancora difficoltoso individuare la giusta configurazione dei parametri per i diversi algoritmi.

Dataset

Per testare gli algoritmi in un ambiente il più possibile indipendente da fattori locali, si è deciso di utilizzare un dataset esterno al posto delle immagini provenienti dal sistema di acquisizione creato. Dopo un attenta ricerca, si è deciso di utilizzare il dataset RGB-D[20], per la qualità della sua struttura e della varietà di immagini contenute. Per la particolare natura dell'algoritmo da testare, si è scelto un dataset comprendente sia le immagini degli oggetti da classificare, che le relative depth-map. Per l'addestramento dell'algoritmo, e la creazione delle strutture di visibilità, è stato utilizzato solamente il 50% delle immagini disponibili per ognuno degli oggetti. Il motivo della scelta è che si intendeva testare il sistema anche in presenza di relativamente poche informazioni.

Hardware

I vari test sono avvenuti utilizzando un normale computer portatile, dotato di processore Intel Centrino2 P8400 e 3GB di memoria DDR2 (bus 1066MHz). Il sistema operativo utilizzato per i test è Linux Ubuntu 11.04, gran parte del sistema implementato è stato tuttavia creato nell'ottica di una facile integrazione con ROS (*Robot Operating System*).

6.2 Risultati

I test, e l'esposizione dei relativi risultati, sono stati strutturati nel seguente modo:

standard risultati relativi all'algoritmo di confronto.

mono risultati relativi all'algoritmo nella sua versione base (singole feature).

couple risultati relativi all'algoritmo finale (coppie di feature).

class rappresenta la classe principale dell'oggetto (es. Tazza).

sub class rappresenta un'istanza della classe particolare (es. Tazza da Caffè).

top 1-5 rappresentano rispettivamente il risultato migliore individuato globalmente, tra i primi cinque risultati.

400 rappresenta la taglia del dizionario utilizzata nell'esperimento.

class 400	classificazione corretta	Errore medio x	y	α	*scale*	ϕ	θ
top 1							
standard	61%	28	25	31	0.31	13.39	30.59
mono	51%	45	49	47	0.40	18.24	47.19
couple	65%	32	26	35	0.34	9.16	23.54
top 5							
standard	86%	19	16	18	0.11	5.17	13.09
mono	71%	27	31	23	0.19	6.73	27.34
couple	90%	16	17	17	0.13	5.01	10.98

sub class 400	classificazione corretta	Errore medio x	y	α	*scale*	ϕ	θ
top 1							
standard	46%	20	21	27	0.25	11.28	26.93
mono	40%	28	35	35	0.33	14.01	38.22
couple	50%	24	19	28	0.26	8.45	19.44
top 5							
standard	67%	15	13	14	0.10	4.93	10.33
mono	54%	23	21	19	0.16	5.12	20.01
couple	66%	12	15	14	0.11	4.81	9.74

Per quanto riguarda i tempi di riconoscimento dei tre algoritmi, la media (in secondi) per l'analisi completa di una scena è stata la seguente:

standard 1.73

mono 0.87

couple 0.91

6.3 Analisi dei risultati

Come è possibile vedere dai risultati il metodo proposto da risultati migliori del metodo di comparazione, fornendo tali risultati in tempi quasi dimezzati. Risulta inoltre particolarmente evidente che il metodo base (visibilità di una singola feature) ottenga prestazioni nettamente inferiori rispetto alla versione completa. Di seguito verranno analizzate le principali cause di errore dell'algoritmo:

posizione Analizzando la maggior parte degli oggetti la cui errore nella stima della posizione risulta maggiore, si è potuto notare che avveniva in oggetti in cui la rotazione influiva poco o nulla sull'aspetto visivo. In questo caso bisogna dire che ben poco si può fare per migliorare la stima, a meno di nuove e migliori tecniche di estrazione delle feature.

classificazione Analizzando gli errori di classificazione, è stato possibile notare una tendenza del classificatore a mantenere ai primi posti dei risultati collegati ad oggetti con una forte presenza di feature (molti contorni, variazioni di colore ecc.). Questo è in parte dovuto ad un sistema non ottimale di assegnazione dei pesi alle feature, in parte è dovuto alla costruzione di migliori modelli per gli oggetti con queste caratteristiche. L'errore è presente sia nel sistema di riferimento che nelle due versioni dell'algoritmo proposto, questo fa pensare che un'implementazione più accurata potrà potrà risolvere facilmente il problema.

Eseguendo test con diverse taglie del dizionario (200 e 800), è stata notata un tasso di crescita nelle prestazioni a favore dell'algoritmo standard nel caso di un aumento dei centroidi. Questa tendenza era prevista ed è dovuta alla minore concentrazione di punti nelle gaussiane della struttura di visibilità. All'aumentare della taglia del dizionario, il numero di coppie di descrittori possibili aumenta con legge quadratica, di conseguenza il numero di punti visti per ogni coppia diminuisce. Questo fatto non è un forte problema, dato che piccoli dizionari sono più veloci e preferibili, certo è che utilizzando dizionari più ampi si guadagna in precisione di riconoscimento. Trovare una buona taglia del dizionario è un problema ancora aperto, molto dipendente dall'implementazione e dal tipo di ambiente in cui deve operare il robot.

6.4 Conclusioni personali

L'analisi dei risultati indica un parziale successo della tecnica proposta, sebbene infatti la comparazione dei dati risulti essere a favore del nuovo metodo, bisogna tener conto della semplicità del metodo di confronto. Ciò che fa supporre che i dati ottenuti siano comunque validi anche per altri sistemi, si basa proprio sul fatto che il metodo proposto può appoggiarsi a qualsiasi altra tecnica, analizzandone i risultati prodotti e migliorandone la confidenza (vedi paragrafo 4.2). Sarà quindi probabile che anche l'utilizzo di un sistema base più evoluto, il sistema proposto migliori le prestazioni in maniera simile ai risultati ottenuti. Per quanto riguarda la percentuale di successo dell'algoritmo, sono fiducioso che un'implementazione più accurata e uno sviluppo teorico più robusto possa portare a fornire risultati decisamente migliori, purtroppo il tempo a disposizione per la tesi ha posto dei limiti in questo senso.

Nel capitolo successivo verranno descritte alcune tra le principali idee e migliorie che per questioni di tempo ho dovuto posporre.

Nel caso questo lavoro possa essere d'ispirazione per il lettore, se necessario sarò felice di discutere per eventuali chiarimenti o proposte.

Capitolo 7

Sviluppi Futuri

L'algoritmo proposto è il risultato di un progetto di tesi laurea e, sebbene vi sia stato investito molto tempo, il lavoro è stato necessariamente focalizzato alla realizzazione dell'idea principale. Durante la progettazione e l'implementazione dell'algoritmo stesso alcune tra le idee più complesse o secondarie sono state messe in secondo piano, nei paragrafi successivi ne verranno descritte brevemente le principali.

Odometria per migliorare la confidenza

Nel presente documento è stata focalizzata l'attenzione al riconoscimento di oggetti in una singola scena, sebbene l'algoritmo proposto abbia dimostrato buone capacità in questa modalità una sua evoluzione naturale è l'analisi di sequenze temporali di scene.

L'idea di base per questo importante sviluppo è basta nell'utilizzo di dati provenienti sistemi odometrici del robot per raffinare la convoluzione tra gaussiane descritta nel paragrafo 4.2. Attualmente nell'algoritmo implementato, da ognuna delle coppie di feature estratte da una scena, viene generata una gaussiana G_H con media $\boldsymbol{\mu} = [0, 0, -\mu_l]^T$ e deviazione $\boldsymbol{\sigma^2} = [0, 0, \sigma_l]^T$. Il motivo per cui le dimensioni relative a (ϕ, θ) hanno media e deviazione nulle sta nel fatto che le coppie provengono da una singola immagine. Di fatto durante la convoluzione questi valori nulli non contribuiscono al raffinamento della gaussiana risultante, lo sviluppo suggerito interviene proprio in questo punto. Mediante l'utilizzo di stime sugli spostamenti fatti da un robot (dati odometrici), è possibile calcolare la variazione della vista di un oggetto rispetto al robot e utilizzarli per migliorare la costruzione delle gaussiane nella fase di riconoscimento.

Questa modifica in primo luogo al robot di migliorare la stima per un oggetto semplicemente cambiando il punto di vista di tale oggetto (ad esempio muovendosi verso di esso o ruotandolo). Un secondo vantaggio è che, attraverso l'analisi della struttura di visibilità, il robot può individuare gli angoli con la presenza di

più feature caratteristiche e quindi decidere quale direzione di spostamento sia la migliore per acquisire più confidenza per l'oggetto.

Selezione delle feature da monitorare

Durante la fase di addestramento descritta nel paragrafo 4.1, vengono selezionate casualmente coppie di feature dall'immagine di una vista e ne viene calcolato lo spazio di visibilità. L'attuale selezione casuale fornisce un elevato numero di coppie nel tentativo di non mancare di monitorare una coppia di feature importanti, questo però grava sia nella creazione della struttura che nella sua analisi (molte gaussiane da confrontare). Come visto nel paragrafo 5.1.2 alcune feature risultano essere più caratterizzanti rispetto ad altre per un determinato oggetto, sfruttando questo concetto è possibile migliorare la selezione delle feature dando maggior probabilità di essere selezionate alle feature con maggior rilevanza. Questa modifica migliorerebbe sia la velocità nella fase di creazione, che la precisione e la velocità nella fase di riconoscimento.

Miglioramento del clustering

Le fasi di maggior peso computazionale dell'algoritmo sono rappresentate dall'individuazione di cluster, questo avviene sia nella fase di creazione delle strutture che nella fase di riconoscimento. Attualmente viene utilizzata una versione pesata di *Agglomerative Mean-Shift*, tale algoritmo risulta però essere piuttosto oneroso rispetto al più veloce *Hierarchical K-Means*. Le librerie utilizzate non hanno permesso lo sfruttamento di pesi nei punti per il secondo algoritmo, rendendo preferibile l'utilizzo di *Mean-Shift*, un'adeguata implementazione di *Hierarchical K-Means* potrebbe però ridurre notevolmente i tempi dell'algoritmo.

Feature più robuste o veloci

Come descritto nel paragrafo 5.3, l'utilizzo delle SIFT è stata motivata soprattutto da questioni implementative, a sfavore di tecniche potenzialmente migliori (ad esempio le ORB). Sebbene le SIFT offrano una buona invarianza rispetto a rotazione e scalatura, sono risultate essere meno robuste rispetto ad altri tipi di trasformazioni affini. L'utilizzo di feature, più robuste rispetto alla variazione della feature durante la rotazione dell'oggetto, migliorerebbe sensibilmente la qualità e l'efficienza della fase di creazione della struttura di covisibilità, con un conseguente miglioramento anche nelle prestazioni del riconoscimento.

Capitolo 8

Appendice

8.1 Calibrazione di videocamere stereo

Durante lo svolgimento della tesi, buona parte della prima fase del lavoro è stata dedicata alla creazione di alcuni strumenti per il corretto utilizzo della videocamera stereoscopica. Per prima cosa è stato creato un tool per la calibrazione della videocamera stereoscopica e, successivamente ne è stato creato uno volto all'estrazione delle depth map delle immagini riprese.

Il secondo strumento è stato scritto dando la possibilità di utilizzare due metodi distinti, il primo metodo utilizza le librerie (Triclops[1]) fornite direttamente con la videocamera stereoscopica, il secondo si è appoggiato alle librerie OpenCV[19].

Tool di Calibrazione

Questo strumento è stato implementato utilizzando le librerie di OpenCV, in queste librerie è presente un modulo sviluppato appositamente per la calibrazione di videocamere, sia normali che stereoscopiche.

Calibrare una telecamera significa stimarne alcuni parametri (matrici), questi parametri descrivono la relazione di trasformazione che proietta la luce proveniente dal mondo reale direttamente nei sensori delle videocamere (quindi nelle immagini). Esistono due principali tipi di parametri da stimare:

intrinseci questi parametri consentono di definire la trasformazione che mappa un punto dello spazio 3D nelle coordinate del piano immagine di ogni telecamera. Essi comprendono le coordinate relative al piano immagine del principal point (punto di intersezione tra il piano immagine e la retta ortogonale al piano immagine stesso passante per il centro ottico), la distanza focale, ed eventualmente altri parametri che descrivono altre caratteristiche del sensore (distorsione delle lenti, forma dei pixels, etc).

[1] `http://www.ptgrey.com/products/triclopsSDK/index.asp`

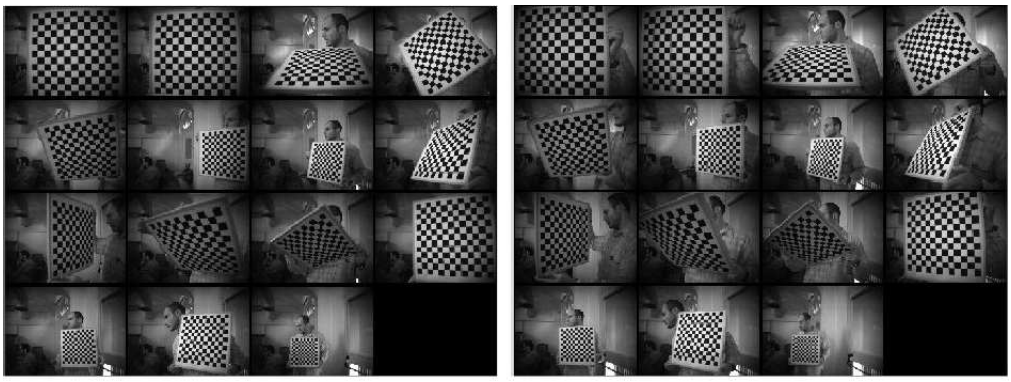

Figura 8.1: Calibrazione tramite acquisizione di un pattern noto in 15 posizioni differenti.

estrinseci i parametri estrinseci rappresentano le posizioni di ogni telecamera rispetto ad una sistema di riferimento noto.

Una volta ottenuti questi parametri è possibile determinare le coordinate 3D di un punto nello spazio, conoscendone le coordinate 2D delle sue proiezioni nelle immagini destra e sinistra.

L'algoritmo utilizzato da OpenCV per ottenere i dati di calibrazione di una videocamera stereoscopica, si basa su un algoritmo[21] di Tsai molto noto nel settore. L'algoritmo consiste nell'analisi della trasformazione subisce che un pattern, fisso e ben caratterizzato, quando viene ripreso dalle telecamere in diverse posizioni (figura 8.1).

Tool per l'estrazione della depth-map

Una volta ottenuti i parametri intrinseci ed estrinseci, è possibile calcolare le coordinate 3D di un punto, per farlo è necessario però conoscerne le coordinate di proiezione nelle immagini. Il problema dell'individuazione delle coordinate nelle due immagini di un punto nel mondo reale è un problema tutt'ora molto studiato. La distanza tra gli obiettivi è, in genere, sufficientemente piccola da far si che le due immagini catturate non differiscano di troppo. Un algoritmo comunemente utilizzato per individuare corrispondenze di punti nelle due immagini, prende il nome di *Block Matching*. Come ne suggerisce il nome l'idea di base dell'algoritmo consiste nell'individuare nelle due immagini delle aree particolarmente simili (ad esempio distanza euclidea dei blocchi di pixel). Una volta ottenute le coordinate dei due blocchi ritenuti simili, l'algoritmo procede nell'estrazione della loro profondità, sfruttando l'informazione sulla differenza di coordinate dei due blocchi(figura 8.4). Il tipo di ricerca descritto potrebbe comportare tempi di esecuzione quadratici rispetto alla taglia dell'immagine, con

Figura 8.2: Immagini acquisite dalla videocamera.

Figura 8.3: Immagini rettificate.

conseguenti tempi di elaborazione non accettabili in contesti real-time. Per ridurre il carico computazionale le due immagini destra e sinistra subiscono una fase di preprocessing detta *Rettificazione*. La fase di rettificazione consiste nello sfruttare i parametri intrinseci ed estrinseci della telecamera per antidistorcere le immagini ed allineare tutte le possibili corrispondenze nella medesima riga di pixel (vedi figure 8.2 e 8.3). In questo modo l'algoritmo *Block Matching* si limiterà a cercare le sole corrispondenze di blocchi giacenti nella medesima riga dell'immagine, riducendo la velocità dell'algoritmo ad un ordine lineare.

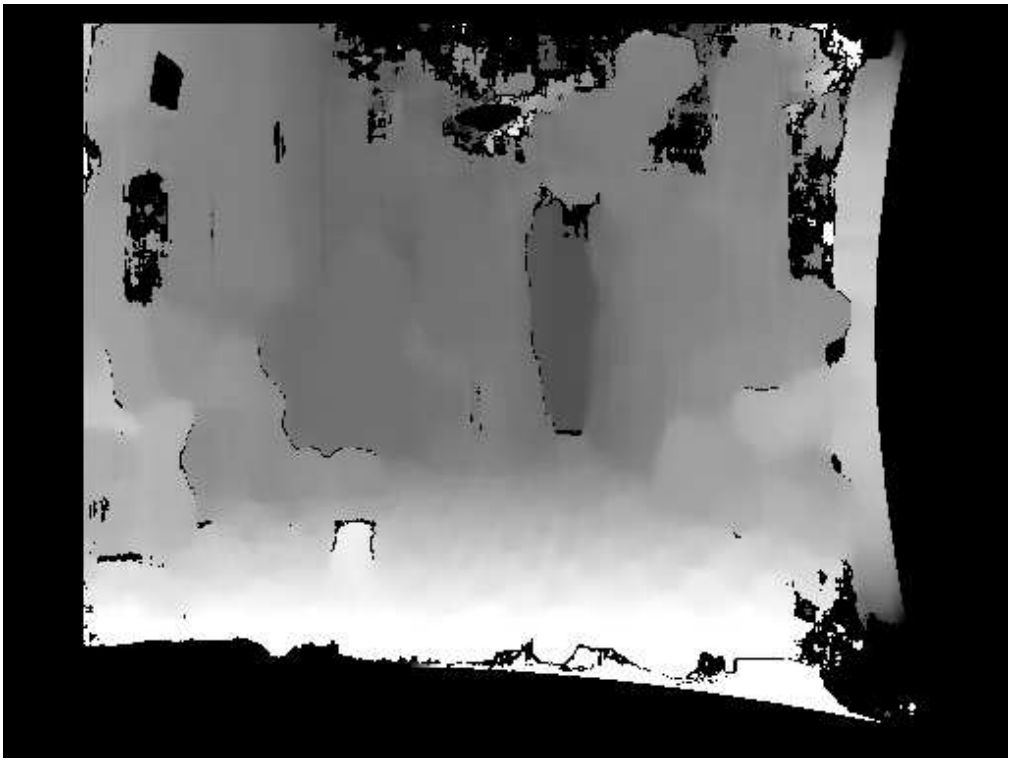

Figura 8.4: Depth map del laboratorio.

Bibliografia

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, November 2004.

[2] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Volume 2 CVPR06*, vol. 2, no. c, pp. 2161–2168, 2006.

[3] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, pp. 1–22, 2004.

[4] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *European Conference on Computer Vision*, Springer, 2006.

[5] D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little, D. G. Lowe, and B. Dow, "Curious george: An attentive semantic robot," in *IROS 2007 Workshop: From sensors to human spatial concepts*, (San Diego, CA, USA), IEEE, November 2007.

[6] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR07). IEEE Computer Society*, pp. 1–8, 2007.

[7] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, no. 9, pp. 1395 – 1407, 2006.

[8] J. Matas, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

[9] S. Helmer and D. Lowe, *Using Stereo for Object Recognition*, pp. 3121–3127. IEEE, 2010.

[10] M. Sun, H. Su, S. Savarese, and L. Fei-Fei, "A multi-view probabilistic model for 3d object classes," in *Proc. Computer Vision and Pattern Recognition*, 2009.

[11] R. Bianchi, A. Ramisa, and R. de MÃ¡ntaras, "Automatic selection of object recognition methods using reinforcement learning," in *Advances in Machine Learning I* (J. Koronacki, Z. Ras, S. Wierzchon, and J. Kacprzyk, eds.), vol. 262 of *Studies in Computational Intelligence*, pp. 421–439, Springer Berlin / Heidelberg, 2010.

[12] S. Romdhani and T. Vetter, "3d probabilistic feature point model for object detection and recognition," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1 –8, june 2007.

[13] K. Pulli and L. Shapiro, "Triplet-based object recognition using synthetic and real probability models," in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, vol. 4, pp. 75 –79 vol.4, aug 1996.

[14] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, A. Rastrow, R. C. Rose, P. Schwarz, and S. Thomas, "The subspace gaussian mixture model-a structured model for speech recognition," *Comput. Speech Lang.*, vol. 25, pp. 404–439, April 2011.

[15] W. K. Pratt, *Superposition and Convolution*, pp. 161–183. John Wiley & Sons, Inc., 2002.

[16] F. Viksten, P.-E. Forssén, B. Johansson, and A. Moe, "Comparison of local image descriptors for full 6 degree-of-freedom pose estimation," in *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, ICRA'09, (Piscataway, NJ, USA), pp. 1139–1146, IEEE Press, 2009.

[17] C. Lampert, M. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1 –8, june 2008.

[18] X. Yuan, B. Hu, and R. He, "Agglomerative mean-shift clustering," *Knowledge and Data Engineering, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2010.

[19] G. Bradski, "The opencv library," *Dr. Dobb's Journal of Software Tools*, 2000.

[20] X. R. Kevin Lai, Liefeng Bo and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.

[21] R. Y. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, (Miami Beach, FL.), pp. 364–374, 1986.

[22] J. W. Goethe-UniversitÃ¤t, "Hierarchical k-means clustering," *Program*, pp. 1–14, 2004.

[23] S. O'Hara and B. A. Draper, "Introduction to the bag of features paradigm for image classification and retrieval," *CoRR*, vol. abs/1101.3354, 2011.

Elenco delle figure