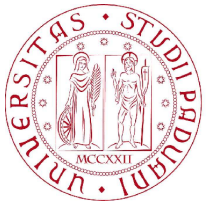
UNIVERSITÀ DI PADOVA  FACOLTÀ DI INGEGNERIA

Dipartimento di Ingegneria dell'Informazione

Indirizzo: Ingegneria informatica

Rilevamento delle salienze visuali per identificazione di oggetti con robot mobili autonomi

Relatore: Ch.mo Prof. Emanuele Menegatti

Corelatore: Dott. Alberto Pretto

Studente: Toffolatti Matteo

Sommario

L'*object recognition* è un task particolarmente importante nella robotica autonoma, vista la potenzialità delle videocamere come sensori, sopratutto in relazione all'interazione con l'ambiente esterno. In questo contesto la *saliency detection* rappresenta il primo passo per direzionare l'attenzione su una porzione dell'immagine analizzata, dove dovrebbe trovarsi un candidato oggetto, prima che ci sia un tentativo di identificazione. E' quindi comprensibile quanto questo passaggio sia rilevante per i tempi di esecuzione di un algoritmo di object recognition, in quanto una cattiva fase di object detection comporta un inutile lavoro nel tentativo di riconoscimento di oggetti laddove non ve ne sia alcuno.
Storicamente il modello seguito per l'ideazione di un algoritmo di saliency detection prende le mosse dalla visione umana, dallo studio dell'interazione occhio-cervello e dei meccanismi fisiologici ad essa collegati. Una volta chiarito il funzionamento della visione umana, e formalizzato nella *FIT* (Features Integration Theory), tale tecnica è stata implementata tramite filtri lineari, nel dominio dello spazio, e si basa sul concetto di *features*.
Successivamente si è passati ad un approccio basato sulla teoria dei segnali, tramite trasformata di Fourier bidimensionale, metodo conosciuto *come spectral* residual. Questo approccio in frequenza, oltre a dare risultati paragonabili a quelli nel dominio dello spazio, risulta enormemente più veloce.
Un ulteriore miglioramento delle performance di questa seconda tecnica si è avuto constatando che uguali saliency map si ottengono utilizzando soltanto lo spettro di fase, rendendo superfluo il calcolo delle ampiezze. Tale evidenza sperimentale è avvalorata da considerazioni sul contenuto spettrale dei punti salienti di una qualsiasi forma d'onda.
Dal punto di vista dell'affidabilità dei risultati si è fatto un ulteriore passo in avan-

ti tramite l'utilizzo dei *quaternioni*, e la definizione una trasformata di Fourier per tale campo. I quaternioni sono un tipo di numero, evoluzione dei complessi, ma con quattro componenti anzichè due.
Tramite i quaternioni si è quindi sviluppata una terza tecnica, operante in frequenza, ma integrata con la FIT, che pur essendo ancora in fase di sperimentazione ha grandissime potenzialità. Infatti, il quaternione, è composto da quattro features, portando quindi molta più informazione di una semplice immagine, a fronte di tempi computazionali competitivi.
Si è quindi implementato un algoritmo di saliency detection tramite quaternioni e trasformata di fase, distinguendosi dalla letteratura sulla scelta di una delle features utilizzate. L'approccio sviluppato, infatti, in vece di una componente legata al movimento riscontrato fra due frame ravvicinati, utilizza una *depth map*. Per la realizzazione di tale mappa delle distanze si è utilizzata una *stereocamera*, che per ogni frame acquisisce immagine destra e sinistra. Dalla differenza tra tali immagini, previa conoscenza dei parametri di calibrazione, si calcola la distanza di ogni pixel. Tale immagine risultante è pregna di significato per la nostra applicazione, in quanto, oltre a differenziare le parti dell'immagine aventi distanza relativa, da un'indicazione precisa sul volume di eventuali oggetti in primo piano. I risultati così ottenuti sono notevoli, tanto per la qualità dei risultati, quanto per i tempi di esecuzione.

Indice

Sommario **I**

Indice **III**

Introduzione **1**

1 Saliency detection: il meccanismo di attenzione dall'uomo alla macchina **7**

1.1 Approccio nel dominio dello spazio 8

1.1.1 La visione umana 8

1.1.2 Feature integration theory 9

1.1.3 Modello Walther-Koch 10

1.2 Approccio in frequenza 14

1.2.1 Spectral Residual 14

2 Approccio tramite Phase Quaternion Fourier Trasform **17**

2.1 Spettro di fase 18

2.1.1 Saliency detection tramite Trasformata di fase 19

2.2 I Quaternioni 20

2.2.1 Definizione 20

2.2.2 Quaternion Fourier Transform 22

2.3 Implementazione di Guo-Ma-Zhang (PQFT) 23

3 Implementazione **27**

3.1 Strumenti 27

3.1.1 Hardware 27

3.1.2 Piattaforma software 28
3.1.3 Depth map 29
3.2 Saliency detection tramite PQFT e depth map 30
3.2.1 Problematiche e dettagli implementativi 33

4 Test e risultati **35**
4.1 Movimento *vs.* depth map 35
4.1.1 Acquisizione immagini 37
4.2 Qualità e tempi di esecuzione 39
4.2.1 Valutazione comparativa 40

5 Conclusioni **43**
5.1 Metodologia 43
5.2 Problematiche e sviluppi futuri 44
5.2.1 Considerazioni soggettive 45

Appendice **47**

Bibliografia **51**

Introduzione

La ricerca nell'ambito della robotica autonoma ha fatto, in questi ultimi anni, rilevanti passi in avanti, ed è a tuttoggi in grande sviluppo, tanto in ambito industriale che in quello accademico. In particolare la "robot perception" è un campo profondamente studiato, in quanto implicato strettamente nell'interazione uomo-macchina, fondamentale per una grossa fetta dei robot in progettazione.
Nel contesto della percezione riveste un ruolo importante il riconoscimento di oggetti (object recognition), in virtù del fatto che la vista è per l'uomo il più stimolato dei sensi, e quello che maggiormente gli fa interpretare l'ambiente. Una grande spinta alla ricerca in questa direzione è data dal *Semantic Robot Vision Challenge* [3] (SRVC), competizione in cui i robot devono identificare una serie di oggetti muovendosi in un ambiente sconosciuto, utilizzando solo immagini di tali oggetti scaricate dal web. I risultati raggiunti dai vincitori di questa competizione sono molto interessanti, pur considerando la non assoluta generalità del contesto, si veda ad esempio [1].

Robot utilizzato

Il progetto (Fig: 1) portato avanti dal gruppo di ricerca di Robotica Autonoma dell'Università di Padova, diretto dal Prof. E. Menegatti, si propone la realizzazione di un robot dalle funzionalità molto simili a quelle dei partecipanti all'SRVC, con l'intento futuro di prendervi parte.
Il robot in realizzazione è autonomo, munito di ruote, dispone di una stereo-camera, e range finder per la misura delle distanze. L'ambiente in cui andrà ad operare è una stanza in cui sono stati distribuiti alcuni oggetti di interesse, di

cui il robot possiede un database di immagini, più alcuni oggetti di disturbo che rendano maggiormente veritiero l'ambiente.

Il robot si sposta all'interno della stanza evitando di urtare mobilio e oggetti tramite un algoritmo di path-planner/obstacle-avoidance basato sulla visione ([2]), e durante tale navigazione attraverso un algoritmo di SLAM ([4]) crea una mappa dell'ambiente, nella quale andrà in seguito ad aggiungere gli oggetti via via riconosciuti. Durante il movimento nella stanza alle immagini raccolte viene applicata la *saliency detection*, operazione finalizzata al ritrovamento degli oggetti (object detection), cui segue una nuova cattura di immagini "centrate" sui candidati oggetti. Questa operazione è necessaria in quanto il database di immagini è una raccolta di "primi piani" degli oggetti, e quindi l'eventuale riconoscimento necessita di immagini in cui l'oggetto ne occupi una buona percentuale. Una volta catturata l'immagine di un candidato oggetto la si da in pasto ad un algoritmo di matching, il quale è realizzato seguendo il metodo del "Bag-of-Words" (BoW). Tale metodo si basa su un vocabolario visuale, creato offline, di features caratteristiche, che vengono confrontate con le features dell'immagine in questione, per un excursus su tale metodo si veda [5], [6], [7].

Figura 1: Robot utilizzato.

L'obbiettivo di tale robot sarà, una volta munitolo di un manipolatore, quello di riordinare una stanza, dopo aver esplorato e mappato l'ambiente. Il robot dovrebbe essere in grado, una volta che gli oggetti siano stati spostati, di ri-

conoscerli e riportarli alla loro posizione iniziale.

Object recognition e saliency detection

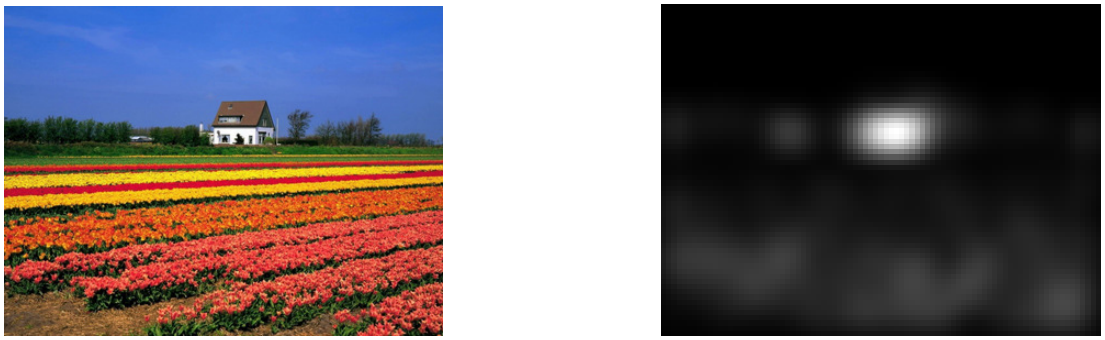
Nel contesto della robotica autonoma l'utilizzo delle videocamere offre grandissime potenzialità, oltre ad un costo contenuto rispetto ad altri sensori,ed è ciò che più si avvicina alla visione umana. Al contempo il loro utilizzo impone una massiva analisi delle immagini per il riconoscimento dell'ambiente di navigazione, il che nasconde criticità molto variegate, alcune delle quali ancora ben lungi dall'essere risolte.
L'**object recognition** è una delle applicazioni che concerne la visione robot, di maggior attualità, e dalle enormi potenzialità. Concettualmente il riconoscimento di oggetti si divide in tre sottoproblemi:

1. Fase di apprendimento;
2. Fase di esplorazione;
3. Fase di riconoscimento.

Nella fase di apprendimento il robot si munisce di un database di immagini, un set per ogni oggetto che gli potrebbe capitare di incontrare. Tali immagini possono essere passate al robot, o più autonomamente, cercate dallo stesso sul web.
Nella fase di esplorazione il robot si muove nell'ambiente in cui sono presenti gli oggetti, ambiente che gli può essere completamente sconosciuto, e raccoglie una serie di immagini.
Nell'ultima fase di riconoscimento le immagini raccolte sono confrontate con quelle presenti nel database, al fine di riconoscervi gli eventuali oggetti presenti.
La visione umana, che rappresenta la fonte di ispirazione e l'obbiettivo ideale per ogni algoritmo di visione robot, si divide in due fasi:

1. Fase di pre-attenzione;
2. Fase di attenzione.

Nella fase di pre-attenzione, rapida e parallela, saltano all'occhio le caratteristiche di basso livello dell'immagine, ovvero linee, colori, gradienti di luminosità, senza tentare ancora di riconoscere ciò che si ha nel campo visivo, ma andando a focalizzarsi su una determinata porzione di esso.
Nella fase di attenzione invece, l'occhio si concentra su ciò che nella precedente ha maggiormente attirato l'attenzione, eseguendo operazioni di più alto livello, come appunto il riconoscimento di oggetti.
La **saliency detection** è l'operazione atta a riprodurre la fase di pre-attenzione, la cui funzione è quella di trovare gli oggetti all'interno dell'immagine di ingresso. Il risultato dell'algoritmo di saliency detection, una mappa dell'immagine in scala di grigi, indica al robot dove focalizzare la propria attenzione, ovvero direzione e livello di zoom delle successive acquisizioni con la fotocamera. Questa operazione è necessaria in quanto le immagini nel database sono dei primi piani degli oggetti, e quindi l'algoritmo di confronto necessita di qualcosa di analogo.



(a) (b)

Figura 2: Esempio di saliency detection: (a) Immagine in ingresso, (b) Saliency map.

Evoluzione storica

Nello studio della letteratura riguardante la saliency detection, il lavoro maggiormente citato è di certo quello di Treisman & Gelade del 1980 ([8]). La parte citata di questo testo riguarda il meccanismo di attenzione per quanto concerne

la visione dei primati, argomento che parrebbe esulare dal nostro contesto, ne è invece il naturale predecessore. I primi ad interessarsi all'argomento forono Tsotsos et al. ([9]), proponendo un modello al calcolatore che emulasse la fase di attenzione nella visione dei primati. I lavori fin qui analizzati, sono stati ripresi da Rensink, che nell'anno 2000 pubblica due articoli([13], [14]) sul meccanismo di attenzione nella visione, coniando il termine di *"proto-object"*, in riferimento ai candidati oggetti, diventato da allora di uso comune.

Contemporaneamente allo studio della visione, e dei meccanismi di attenzione, fine a se stessi, si comincia anche a concepire la possibilità di sfruttare le conoscenze acquisite in tale campo per utilizzarle nella robot/computer vision. L. Itti & C. Koch sono tra i più attivi e propositivi in questo campo, a cavallo tra la fine degli anni novanta e l'inizio del terzo millennio, dando alle stampe tre opere miliari sull'argomento: [10], [11] e [12]. Interessati a tali argomenti ci sono sempre un maggior numero di scienziati e gruppi di ricerca delle principali università mondiali, e molti sono i ricercatori che si dedicano allo sviluppo della computer vision, e di conseguenza della saliency detection. Si è così andato sviluppando e raffinando il modello proposto da Itti-Koch, fino a giungere a una versione definitiva nel 2006, grazie a Walther & Koch ([15]), metodo che lavora nel dominio dello spazio.

Nel 2007 esce un articolo innovativo ([16]), di Hou e Zhang, che va a creare la saliency map andando a lavorare nel dominio delle frequenze. Il fatto è molto rilevante, in quanto tale strategia abbatte sensibilmente i tempi di esecuzione dell'algoritmo, rendendolo molto più appetibile per un utilizzo real-time, quindi anche per un robot autonomo.

Il metodo proposto in [16] è abbastanza grezzo, ma ha subito svariati raffinamenti nel corso di questi ultimi anni, anche andando ad integrarlo con le features caratteristiche dell'approccio originale nello spazio. Tali raffinamenti sfociano nel 2008 in una tecnica di saliency detection basata sullo studio in frequenza, che sfrutta inoltre la potenza dei quaternioni per realizzare una maggior precisione dei risultati unitamente ad ottime velocità di esecuzione: [17].

Presentazione tesi

Nel primo capitolo verranno presentati in maniera approfondita i due approcci, quello nel dominio dello spazio e quello in frequenza. Si comincierà partendo da considerazioni biologiche sul funzionamento della visione nei primati, passando per la formalizzazione matematica di tali concetti, ed andando infine ad accennare alle problematiche implementative collegate.
Nel secondo capitolo, invece, si presenterà l'approccio con quaternioni proposto da Guo, Ma, Zhang, le cui performance sono, allo stato dei fatti, le migliori. Verrà dettagliatamente descritto,dalla formalizzazione matematica, ai dettagli implementativi.
Il terzo capitolo sarà dedicato alla descrizione dell'implementazione da noi sviluppata, la quale prende le mosse dal metodo ibrido proposto nel capitolo 2, integrato con l'utilizzo di una depth map. Si danno inoltre i dettagli sugli strumenti utilizzati, e una descrizione del robot in uso.
Nel quarto capitolo sono descritti tutti i test realizzati, la loro fase di preparazione, di svolgimento, ed alcuni risultati derivanti da tali operazioni. Si porrà l'attenzione sulla difficoltà riscontrata nell'utilizzo della stereocamera, sfruttata per la realizzazione della depth map, che ha occupato una quantità di tempo ingente e inattesa.
Nel quinto ed ultimo capitolo si riportano le conclusioni, più alcune riflessioni sull'argomento, sull'approccio implementato, e su possibili futuri lavori atti a raffinare tale tecnica.
Infine in appendice verrà presentato il codice prodotto.

Capitolo 1

Saliency detection: il meccanismo di attenzione dall'uomo alla macchina

Il problema legato all'individuazione delle salienze visuali all'interno di un'immagine è un task fondamentale nella realizzazione di algoritmi efficienti in svariati settori della computer/robot vision, tanto in ambito commerciale che in quello della ricerca. L'object recognition per robot mobili autonomi necessita di una saliency detection particolarmente precisa e veloce, nel sottoproblema di object detection, essendo stringenti i suoi vincoli di real-time. E sono proprio l'attualità e l'importanza del problema dell'object recognition che hanno dato, in questi ultimi anni, grande spinta alla ricerca in tale direzione.
I primi studi sull'argomento niente avevano a che spartire con le attuali problematiche legate all'object recognition, erano infatti ricerche volte a spiegare il meccanismo dell'attenzione nella visione dei primati: il "*Focus of Attention*" (FoA). Solo in un secondo momento i risultati ottenuti in tali ricerche furono applicati alla computer vision, per simulare il FoA, e utilizzarne i risultati per le applicazioni più svariate.
Partendo quindi dall'analisi delle *features* elementari stimolanti gli apparati neurovisivi animali, come linee o colori, si è sviluppata una tecnica di ripetuti filtraggi dell'immagine in analisi, atti ad isolare queste stesse features, che una volta elaborate nel loro insieme permettono di creare la saliency map. La tecnica è stata via via raffinata, raggiungendo risultati ragguardevoli dal punto di vista della precisione, tuttavia migliorabili dal punto di vista dei tempi computazionali.

L'abbattimento, di un fattore dieci, del tempo di esecuzione dell'algoritmo di saliency detection si è avuto con l'abbandono del dominio dello spazio, ed una nuova tecnica operante in frequenza. Tale tecnica, detta Spectral Residual (SR), sfrutta la Trasformata di Fourier bidimensionale dell'immagine in questione, unitamente ad una evidenza sperimentale sulla regolarità nello spettro medio di un insieme di immagini. Come detto tale algoritmo risulta un ordine di grandezza più rapido di quello nel dominio dello spazio, sebbene i risultati siano talvolta inferiori qualitativamente.
Le tecniche odierne hanno in parte superato questa difficolta, mantenendo l'approccio in frequenza, si è però integrato con il concetto di features, inserite utilizzando il supporto matematico dei *quaternioni*, e lo sfruttamento della Trasformata di fase. I quaternioni sono un'estensione dei numeri complessi, che consentono di comprimere più informazioni all'interno di un solo numero, il che rende la saliency map risultante di miglior qualità.
Tale tecnica ibrida sfrutta la *Phase Quaternion Fourier Trasform*, chiamata in seguito PQFT.

1.1 Approccio nel dominio dello spazio

1.1.1 La visione umana

La visione umana ha per la robot/computer vision un doppio ruolo, da un lato offre i maggiori spunti per lo sviluppo dei suoi algoritmi, dall'altra rappresenta il naturale obbiettivo da raggiungere per tali discipline. Sarebbe infatti utilissimo creare un algoritmo che potesse permettere ad un robot di estrarre da un'immagine tutte le informazioni che dalla stessa immagine un'uomo assimila.
Ovviamente tale obbiettivo non è alla portata delle attuali tecniche di visione artificiale, sebbene, come detto, sia un ambito di grande interesse economico e scientifico.
L'apparato visivo umano ha nel cervello il suo organo principale, che rielabora, integra e interpreta le informazioni che gli giungono dall'ambiente esterno. La luce riflessa dagli oggetti che ci circondano raggiunge le pupille, da qui, tramite

i mezzi diottrici oculari viene riflessa sulla superficie bidimensionale sferica della retina, dove viene inizialmente interpretata.
Il meccanismo che permette questa prima interpretazione, è l'azione dei coni e dei bastoncelli, che convertono l'energia dei fotoni incidenti, in un segnale, che tramite il nervo ottico giunge al cervello. Occhio, coni, bastoncelli, nervo ottico sono l'"hardware" che trasforma le onde elettromagnetiche che arrivano al nostro occhio nell'immagine che ci troviamo di fronte, e che risiede nel cervello, organo che funge da software per l'elaborazione di tale immagine.
Il meccanismo dell'attenzione è la parte di elaborazione svolta dal cervello che ci interessa, in quanto il meccanismo di saliency detection ha come obbiettivo la sua riproduzione. Le discipline scientifiche che maggiormente hanno contribuito alla conoscenza di tale meccanismo sono la psicofisica e la neurofisiologia, la prima ci ha dato risultati interessanti studiando i tempi di reazione a stimoli visivi, la seconda invece ci ha permesso di comprendere i meccanismi neuronali e le zone del cervello implicate nel meccanismo di attenzione.

1.1.2 Feature integration theory

La *Feature Integration Theory* (FIT) è una pietra miliare nello studio del meccanismo di attenzione dei primati, e quindi umano, sviluppata da Treisman & Gelade nel 1980 in un celebre articolo [8]. Uno degli aspetti più interessanti e innovativi di tale teoria fu che, oltre a sfruttare le conoscenze psicofisiche e neurofisiologiche dell'epoca, cominciò ad integrarle tramite le conoscenze sul calcolo parallelo al calcolatore, in gran voga sul finire degli anni Settanta.
E' preliminarmente importante introdurre la metafora dello *Spotlight* (riflettore), che sintetizza come la messa a fuoco dell'attenzione si comporti appunto come un riflettore che illumina parte del cielo notturno, rendendo chiaro come si possa essere attratti da porzioni di campo visivo diverse dal suo centro. Quindi, partendo dalla ben conosciuta metafora dello spotlight, la FIT pone le basi per lo studio del problema della saliency detection,rimanendo al centro di tale studio per più di vent'anni.
La nozione di "features" si inquadra nel contesto del meccanismo di *pre-attenzione*

nella visione umana, e sta ad indicare quelle caratteristiche di basso livello di un'immagine che vengono processate rapidamente ed in parallelo all'intero del campo visivo. Esempi di features sono i colori, le orientazioni, i terminatori di linea, la luminosità, ma anche concetti un po' meno basici come il movimento, i volumi o la disparità stereo. L'articolo prende in considerazione anche alcune caratteristiche nell'elaborazione di tali features, come la ricerca in parallelo, che asserisce a livello teorico, che il tempo di ricerca di una di esse è indipendente dal numero di oggetti presenti nel campo visivo, essendo appunto elaborati in parallelo.
I postulati della FIT si possono riassumere come segue:

1. Per ogni feature viene creata una mappa, topologicamente organizzata, che prende il nome di "*features map*",

2. il contributo di tali mappe porta alla creazione di una "*master map*", dalla quale è determinato il fuoco dell'attenzione,

3. il contenuto delle features map è aggiornato nel processo di pre-attenzione, e non è direttamente accessibile in modo cosciente,

4. l'accesso cosciente si può avere soltanto alla master map, si può cioè sovvertire volontariamente la priorità dell'attenzione.

L'articolo argomenta in maniera approfondita tali teorie dal punto di vista biologico e comportamentale, il che esula però dai nostri scopi, e pertanto non approfondiremo l'argomento. Dal punto di vista computazionale è invece un po' vago, non spiega ad esempio il legame tra features e master maps, ma rimane un testo fondamentale sul cui modello si basa tutta la trattazione "nello spazio" dell'argomento.

1.1.3 Modello Walther-Koch

La FIT rappresenta quindi un fondamentale punto di partenza nello studio del problema di saliency detection, sul cui modello si è basata la ricerca in tale campo per più di vent'anni, fino a che si portò il problema nel dominio della frequenza.

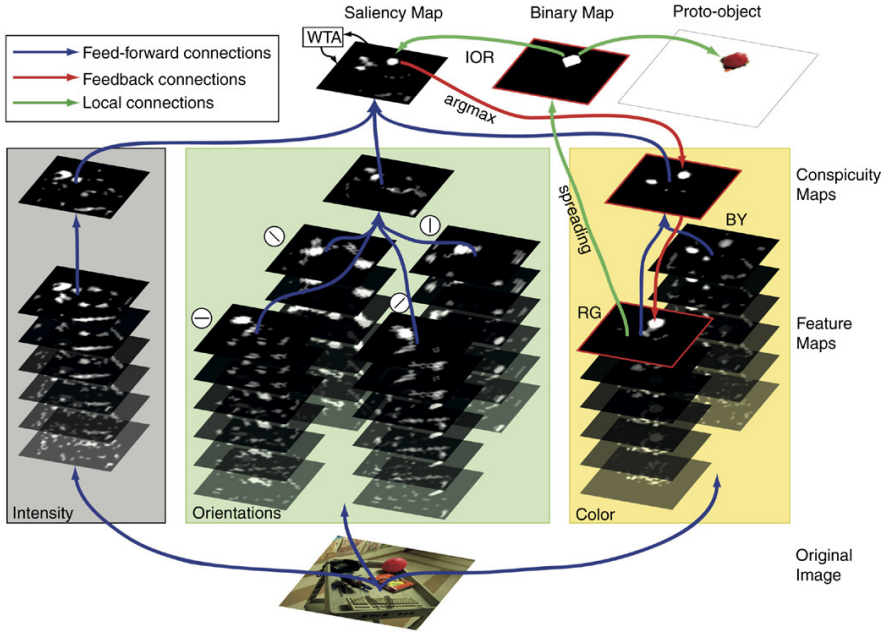


Figura 1.1: Schema logico di funzionamento del modello Walther-Koch.

L'ultima versione di un modello computazionale di saliency detection che operasse esclusivamente nel dominio dello spazio è quella proposta da Walther & Koch nel 2006 ([15]), modello che si può prendere come sunto di tutto ciò che dagli anni Ottanta si è fatto in tale ambito.

Il modello Walther-Koch (Fig.:1.1) è essenzialmente basato su quello proposto da Itti, Koch & Niebur nel 1998 ([10]), il quale si ispira a sua volta dalla formalizzazione della FIT proposta da Koch & Ulmann già nel 1985: [19]. In tale articolo viene implementata, per la prima volta in un modello di focalizzazione di attenzione, la rete neurale detta *Winner-Take-All* (WTA), introdotta a sua volta da Feldman [18], e da allora universalmente accettata come criterio di selezione di uno stimolo. L'immagine di input I viene sotto campionata in una piramide Gaussiana tramite convoluzione con un filtro Gaussiano lineare, e decimazione di un fattore due, operazioni eseguite nell'ordine, prima sull'asse delle x e poi su quello delle y. Vengono in questa maniera creati otto livelli ($\sigma = [1, ..., 8]$), dove la risoluzione dell'immagine al livello σ è $1/2^{\sigma}$, ed ha $(1/2^{\sigma})^2$ pixel, rispetto all'originale.

Detti r, g e b i valori di rosso, verde e blu dell'immagine I, viene creata una

mappa delle intensità per ogni livello della piramide ($M_I(\sigma)$)

$$M_I = \frac{r+g+b}{3} \quad . \tag{1.1}$$

Ogni livello della piramide è inoltre decomposto in mappe per l'opposizione rosso-verde (R-G) e blu-giallo (B-Y)

$$M_{RG} = \frac{r-g}{max(r,g,b)} \quad , \tag{1.2}$$

$$M_{BY} = \frac{b-min(r,g)}{max(r,g,b)} \quad . \tag{1.3}$$

Le mappe delle orientazioni ($M_\theta(\sigma)$) sono calcolate tramite convoluzione dei livelli di intensità di ogni livello della piramide, con filtri di Gabor

$$M_\theta(\sigma) = \|M_I(\sigma) * G_0(\theta)\| + \|M_I(\sigma) * G_{\frac{\pi}{2}}(\theta)\| \quad , \tag{1.4}$$

dove

$$G_\psi(x,y,\theta) = \exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\delta^2}\right)\cos\left(2\pi\frac{x'}{\lambda}+\psi\right) \quad , \tag{1.5}$$

è un filtro di Gabor, con γ che rappresenta l'ellitticità del supporto della funzione di Gabor, δ, λ e ψ rispettivamente la deviazione standar, la lunghezza d'onda e la fase, e le coordinate (x', y'), orientate secondo θ

$$x' = x\cos(\theta) + y\sin(\theta) \quad , \tag{1.6}$$

$$y' = -x\sin(\theta) + y\cos(\theta) \quad . \tag{1.7}$$

I parametri utilizzati nell'articolo [15] sono: $\gamma = 1$, $\delta = 7/3$ pixel e $\psi \in \{0, \pi/2\}$. I filtri sono troncati a 19 x 19 pixel.
La differenza recettiva fra centro e contorno è simulata attraverso la sottrazione tra scale $\ominus$ tra due mappe, una al centro (c) e una di contorno (s) sui livelli della piramide, producendo le "features maps":

$$F_{l,c,s} = N(|M_l(c) \ominus M_l(s)|) \quad , \qquad \forall l \in L = L_I \cup L_C \cup L_O \quad , \tag{1.8}$$

con

$$L_I = \{I\} \quad , \tag{1.9}$$

$$L_C = \{RG, BY\} \quad , \tag{1.10}$$

$$L_O = \{0°, 45°, 90°, 135°\} \quad , \tag{1.11}$$

dove $N(\cdot)$ è un operatore di normalizzazione, iterativo e non lineare, proposto in [12, Itti & Koch 2001].
Le features maps vengono sommate calcolando le differenze centro-contorno, attraverso l'addizione tra scale, e nuovamente normalizzate

$$\bar{F}_l = N\Big(\bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} F_{l,c,s}\Big) \quad \forall l \in L \quad . \tag{1.12}$$

Fatto questo si formano le "*conspiquity maps*" (fig.:1.1) tramite somma e normalizzazione

$$C_I = \bar{F}_l \quad , \tag{1.13}$$

$$C_C = N\Big(\sum_{l \in L_C} \bar{F}_l\Big) \quad , \tag{1.14}$$

$$C_O = N\Big(\sum_{l \in L_O} \bar{F}_l\Big) \quad . \tag{1.15}$$

Infine, dalle tre conspiquity maps si ottiene la saliency map:

$$S = \frac{1}{3} \sum_{k \in \{I,C,O\}} C_k \quad . \tag{1.16}$$

A questo punto la rete neurale WTA esplora la saliency map e sequenzialmente segnala i candidati oggetti in ordine di importanza, tramite la seguente procedura: sceglie il punto più saliente, lo segnala, lo inibisce tramite il meccanismo di *inibition of return* (IOR). Inoltre, per avere una più fedele ricerca dei punti di salienza, si agisce anche sulle conspiquity maps, ed in cascata sulle features, al fine di eliminare da esse il contributo maggiore alla "vittoria" del proto-object in esame.

1.2 Approccio in frequenza

L'analisi nel dominio della frequenza del problema di saliency detection ha un inizio ben determinato, coincide infatti con la pubblicazione dell'articolo "Saliency detection: a spectral residual approach" di Xiaodi Hou & Liqing Zhang [16]. Edito nel 2007, l'articolo rappresenta una svolta nella trattazione dell'argomento, deviando radicalmente da tutta la letteratura fin qui considerata, rivoltando completamente l'approccio al problema.
Se fino ad allora il problema consisteva nell'emulare, nel minor tempo computazionale possibile, il meccanismo di attenzione dell'uomo, questo articolo porta il problema alla sua vera natura, affrontandolo da un punto di vista del tutto nuovo. Si estrapola così il problema stesso dal meccanismo di soluzione noto (quello umano), e lo si risolve in maniera duale. Il metodo di saliency detection finora studiato sfrutta alcune caratteristiche degli oggetti cercati (le features), per la loro stessa identificazione, il metodo *Spectral Residual* (SR) sfrutta invece caratteristiche dello sfondo, nel dominio della frequenza, duale rispetto all'approccio nello spazio.

1.2.1 Spectral Residual

La tecnica di spectral residual prende spunto da uno dei postulati fondamentali della teoria dell'informazione: ciò che porta informazione è solamente ciò che è inatteso, ciò che non si può prevedere a priori. Quindi, considerando il campo di interesse in questione si può riassumere il concetto applicato alla computer vision:

$$H(\text{Immagine}) = H(\text{Innovazione}) + H(\text{Conoscenscenza Pregressa}) \ ,$$

dove H rappresenta l'informazione. Questa semplice formula presuppone che sia possibile scindere l'immagine nella sua parte inattesa e in quella prevedibile, ed è questo il nocciolo del problema, la cui soluzione sta nell'analisi in frequenza. L'argomento è stato ampiamente trattato in letteratura, già Rudemant nel 1994 ([21]) faceva notare la ridondanza nello spettro di immagini naturali, e ne prevedeva il possibile utilizzo nello studio della visione. Infatti, facendo la media degli

spettri di un buon numero di immagini, si è notato come le frequenze seguano la legge $1/f^2$, già solo con un centinaio di immagini tale andamento è molto rispettato. Questa evidenza sperimentale è molto legata alla teoria dell'informazione, e ci fa predire con buona approssimazione come sia strutturato lo spettro di una qualsiasi immagine. Il rapporto fra ridondanza e innovazione in un'immagine è trattato con dovizia da due olandesi, van der Schaaf & van Hateren in [20], e ripreso fra gli altri da Gluckman, in [22], dove si evidenzia come la "pulizia" dello spettro dalla sua parte ridondante sia un'operazione invertibile, con tutte le implicazioni che ciò comporta.

Detta $A(f)$ l'ampiezza dello spettro medio di un insieme di immagini naturali, essa obbedisce alla distribuzione

$$E\{A(f)\} \propto 1/f \quad . \tag{1.17}$$

Un grafico con valori logaritmici sia di f che di $A(f)$, è quasi una linea retta, con coefficiente angolare 1/f, mentre, nel proseguio della nostra trattazione, consideriamo grafici con solo ampiezza logaritmica. Questo ci da un grafico risultante molto simile a quello di $1/f^2$ dove

$$L(f) = \log\big(A(f)\big) \quad . \tag{1.18}$$

Questa dipendenza vale però solo per lo spettro medio di numerose immagini, sulla singola immagine, invece, l'andamento si discosta dall'andamento previsto, e questo ha suggerito agli autori l'idea che la parte singolare dello spettro sia in relazione con i proto objects.
L'immagine in ingresso viene sottocampionata, in modo che l'altezza diventi di 64 pixel, e se ne calcola lo spettro logaritmico $L(f)$. A questo punto è immediato notare come l'informazione portata da $L(f)$ sia:

$$H(R(f)) = H(L(f)|A(f)) \quad , \tag{1.19}$$

dove $A(f)$ rappresenta l'andamento dello spettro medio precedentemente analizzato.

$R(f)$ è detto spectral residual, rappresenta la parte inattesa di un'immagine, ed è centrale per il proseguio dell'analisi:

$$R(f) = L(f) - A(f) \quad . \tag{1.20}$$

$A(f)$ invece è approssimato tramite filtraggio di $L(f)$, utilizzando un filtro di media locale $h_n(f)$

$$A(f) = h_n(f) * L(f) \quad , \tag{1.21}$$

dove $h_n(f)$ è una matrice nxn

$$h_n(f) = \frac{1}{n^2} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} \quad , \tag{1.22}$$

dove, nel nostro caso $n = 3$, anche se l'aumentare di n non altera i risultati in maniera visibile.
La saliency map è ottenuta semplicemente antitrasformando $R(f)$ e convolvendolo con un filtro gaussiano $g(f)$ con $\sigma = 8$.

I risultati ottenuti con tale tecnica sono confrontabili con quelli nel dominio dello spazio, il tempo di esecuzione dell'algoritmo è invece sensibilmente più breve, circa di un fattore dieci, caratteristica fondamentale se considerate le necessità di real-time di un robot autonomo.

Capitolo 2

Approccio tramite Phase Quaternion Fourier Trasform

L'articolo [16] di Hou & Zhang ha spalancato la porta dell'analisi in frequenza per la soluzione del problema di saliency detection, e tale metodo si è subito imposto, per gli indubbi vantaggi di efficienza computazionale. I tempi di esecuzione sono infatti radicalmente abbattuti, circa di un fattore dieci, e, al contempo, la qualità non sembra sembra risentirne, anzi, in alcuni casi migliora. Questo abbattimento dei tempi computazionali è ottimo per tutte le applicazioni di impiego, soprattutto nel nostro caso, dove le esigenze di real time la fanno da padrone.
In questo capitolo andiamo ad analizzare nel dettaglio l'implementazione proposta da Guo, Ma & Zhang in [17], estensione di [16], ma con notevoli miglioramenti e brillanti intuizioni. Il metodo realizzato nell'articolo, è un ibrido delle tecniche in frequenza e nello spazio, ma con l'utilizzo di concetti e strumenti innovativi, non di per se stessi, ma per il campo di applicazione. Tali strumenti sono essenzialmente due: lo sfruttamento della trasformata di fase in vece dello SR, e il supporto matematico dei *quaternioni*. Inoltre, tra le features considerate c'è anche il movimento, che per l'applicazione considerata nell'articolo è utilissimo, volendo analizzare con videocamera fissa una scena in movimento.

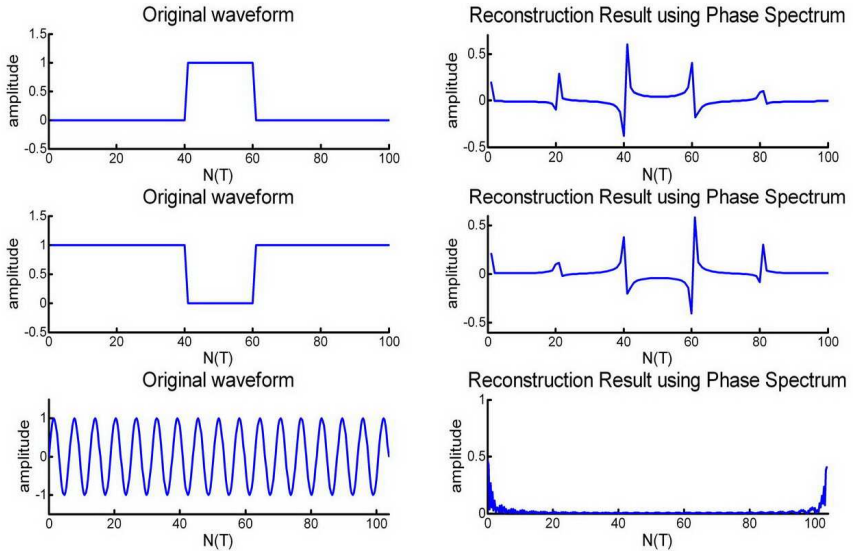


Figura 2.1: Ricostruzione di tre forme d'onda monodimensionali, tramite Trasformata di fase.

2.1 Spettro di fase

Questo algoritmo di saliency detection prende le mosse dalla tecnica di spectral residual ([16]), ma con una fondamentale innovazione: l'utilizzo del solo spettro di fase per il calcolo della saliency map. E' stato dimostrato che la parte innovativa di un'immagine, quella cioè che porta l'informazione, è rappresentata, in frequenza, dallo SR, la cui antitrasformazione indica dove all'interno dell'immagine sono distribuite tali singolarità.
Cionondimeno un attento studio della Trasformata di Fourier, nella sua forma esponenziale, ne mette in risalto una caratteristica utile al nostro scopo. Infatti, presa una qualsiasi forma d'onda, se ne viene fatta una ricostruzione partendo solo dall'antitrasformazione del suo spettro di fase, si evince una caratteristica fondamentale. In tale ricostruzione infatti, i picchi principali appaiono in corrispondenza delle maggiori variazioni della funzione originale, in particolare se si tratta di impulsi. Tale caratteristica la possiamo apprezzare in Fig: 2.1, dove troviamo tre forme d'onda monodimensionali, e le loro ricostruzioni tramite Trasformata di fase. Tale caratteristica è dovuta al fatto che un impulso è dato dalla somma di un ingente numero di sinusoidi, e ciò è messo in risalto dalla Trasformata di fase,

mentre, se una funzione è una singola sinusoide tale ricostruzione non presenta discontinuità (Fig: 2.1). Passando al nostro campo di interesse, ciò significa che l'ampiezza della Trasformata di Fourier indica la quantità di ogni sinusoide presente, mentre la fase ne indica la collocazione all'interno dell'immagine, come si evince dall'opera di Castleman: [23].
Questa caratteristica della Trasformata di fase ci permette il calcolo della saliency map senza dover computare anche la sua ampiezza, metodo che permette un risparmio di 1/3 dei calcoli rispetto allo SR. Un piccolo problema nell'utilizzo della Trasformata di fase è rappresentato dalle discontinuità ai bordi dell'immagine (si veda la terza forma d'onda di Fig: 2.1, che è viziata dal medesimo problema), vedremo in fase di implementazione come è stato facilmente risolto.

2.1.1 Saliency detection tramite Trasformata di fase

Analizziamo ora nel dettaglio il modello matematico che permette di evidenziare i proto object attraverso la Trasformata di fase dell'immagine in analisi, che, oltre ad essere computazionalmente veloce, è anche molto stringato.
Data l'immagine in ingresso $I(x, y)$ si va a computare:

$$f(x, y) = F(I(x, y)) \quad , \tag{2.1}$$

$$p(x, y) = P(f(x, y)) \quad , \tag{2.2}$$

$$sM(x, y) = g(x, y) * ||F^{-1}[e^{i \cdot p(x,y)}]||^2 \quad , \tag{2.3}$$

dove F e F^{-1} rappresentano la Trasformata di Fourier e la Trasformata Inversa di Fourier. $P(\cdot)$ invece è l'operatore che calcola lo spettro di fase della Trasformata, $g(x, y)$ è un filtro gaussiano ($\sigma = 8$), lo stesso utilizzato in SR [16]. Il calcolo della saliency map, per ogni posizione (x, y), è eseguito tramite l'equazione 2.3, mentre con SR andrebbe aggiunto lo spectral residual alla parentesi quadra della stessa.

2.2 I Quaternioni

I *Quaternioni* sono un'estensione dei numeri complessi, e formano uno spazio vettoriale a quattro dimensioni. Essi rappresentano quindi una struttura particolarmente compatta, che consente di "trasportare" quattro differenti informazioni all'interno di un solo numero. E' proprio questa la caratteristica che rende i quaternioni particolarmente adatti al nostro scopo, in quanto permettono di affidare ad uno di essi quattro features differenti, che vengono quindi processate in parallelo.
Una volta introdotti i Quaternioni e le loro proprietà varrà esplicitamente definita la Trasformata di Fourier su tale campo, e illustrata la metodologia di saliency detection tramite tali strumenti matematici.

2.2.1 Definizione

In matematica, i quaternioni sono entità introdotte da William Rowan Hamilton nel 1843 in *"The outlines of Quaternions"*: [24]. L'insieme $\mathbb{H}$ dei quaternioni è un corpo non commutativo, esso soddisfa quindi tutte le proprietà usuali dei campi, come i numeri reali o complessi, tranne la proprietà commutativa del prodotto, e, sul campo reale, sono anche uno spazio vettoriale a quattro dimensioni.
Formalizzando, si definisce un quaternione come un elemento

$$q = a + bi + cj + dk \quad , \tag{2.4}$$

con a, b, c e d numeri reali, ed i, j e k simboli letterali.
Tutte le operazioni fra quaternioni sono definite tenendo conto delle relazioni:

$$i^2 = j^2 = k^2 = ijk = -1 \quad . \tag{2.5}$$

Tali relazioni mettono in risalto la caratteristica fondamentale di tale campo, cioè la non commutatività del prodotto:

$$k = ij \qquad \neq \qquad ji = -k \ ,$$

$$
\begin{aligned}
i = jk \quad &\neq \quad kj = -i \ , \\
j = ki \quad &\neq \quad ik = -j \ .
\end{aligned}
$$

I quaternioni hanno molte caratteristiche proprie ai numeri complessi: anche per i quaternioni, in analogia con i complessi, possono essere definiti concetti come norma e coniugato; ogni quaternione, se diverso da zero, possiede un inverso rispetto al prodotto.
Il coniugato di un quaternione $q = a + bi + cj + dk$ è definito come

$$
\overline{q} = a - bi - cj - dk \quad , \tag{2.6}
$$

con le seguenti proprietà:

$$
\begin{aligned}
\overline{\overline{q}} &= q \quad , \\
\overline{q_1 + q_2} &= \overline{q_1} + \overline{q_2} \quad , \\
\overline{q_1 q_2} &= \overline{q_2} \cdot \overline{q_1} \quad .
\end{aligned}
$$

La norma è invece definita come:

$$
|q| = \sqrt{q\overline{q}} = \sqrt{a^2 + b^2 + c^2 + d^2} \quad . \tag{2.7}
$$

Essa è sempre positiva, tranne nel caso in cui $q = 0$, e soddisfa a

$$
\begin{aligned}
|q|^2 &= q\overline{q} \quad , \\
|q_1 q_2| &= |q_1||q_2| \quad .
\end{aligned}
$$

L'inverso di un quaternione q, diverso da zero, esiste sempre ed è dato da

$$
q^{-1} = \frac{\overline{q}}{|q|^2} \quad , \tag{2.8}
$$

con le proprietà:

$$
\begin{aligned}
|q^{-1}| &= \frac{1}{|q|} \quad , \\
\overline{q^{-1}} &= \overline{q}^{-1} \quad , \\
(q_1 q_2)^{-1} &= q_2^{-1} q_2^{-1} \quad .
\end{aligned}
$$

Andiamo ancora ad analizzare due rappresentazioni alternative di un quaternione che ci torneranno utili nel seguito: *forma polare* e scomposizione di *Cayley–Dickson* (CD).
La formula di Eulero per esponenziali complessi si generalizza ai quaternioni,

$$e^{\mu\theta} = \cos\theta + \mu \sin\theta \quad , \tag{2.9}$$

dove μ è un quaternione unitario puro. Quindi possiamo scrivere la forma polare come

$$q = |q|e^{\mu\theta} \quad , \tag{2.10}$$

dove μ e θ sono detti *eigenaxis* ed *eigenangle* del quaternione. μ da la direzione degli assi della parte vettoriale di q, mentre θ è l'analogo dell'argomento di un numero complesso, che risulta però unico solo nel range $[0, \pi]$.
Un quaternione può essere rappresentato anche come un numero complesso, dove parte reale e immaginaria sono anch'esse numeri complessi, i quali hanno operatori complessi diversi, ed ortogonali fra loro. Questa è conosciuta come forma di Cayley–Dickson:

$$q = A + Bj = (a + bi) + (c + di)j \quad , \tag{2.11}$$

che, per le regole viste, restituisce

$$q = a + bi + cj + dk \quad .$$

2.2.2 Quaternion Fourier Transform

La prima apparizione in letteratura dell'applicazione ai quaternioni della Trasformata di Fourier si deve a T. A. Ell, che nel 1992 presenta l'argomento come tesi per il suo Ph.D. [26]. Lo stesso Ell, in collaborazione con S. Sangwin, è il primo a proporre in [25] l'applicazione dell'argomento a immagini a colori.

La prima definizione di Trasformata di Fourier per quaternioni, definita in [26], è la seguente:

$$H[j\omega, kv] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-j\omega t} h(t,\tau) e^{-kv\tau} dt d\tau \quad , \tag{2.12}$$

e la sua inversa

$$h(t,\tau) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{j\omega t} H[j\omega, kv] e^{kv\tau} dv d\omega \quad . \tag{2.13}$$

Tuttavia, il tentativo di generalizzare le operazioni di convoluzione e correlazione per numeri complessi ai quaternioni fallì, e così venne definita una Trasformata generalizzata, che permette tali operazioni

$$F[u,v] = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} e^{-\mu 2\pi((mv/M)+(nu/N))} f(n,m) \quad , \tag{2.14}$$

dove l'inversa è data cambiando il segno dell'esponenziale, e sommando su u e v invece che su m e n. Tale definizione, con la relativa dimostrazione fu data alle stampe da Ell & Sangwin nel 2000 in [27].

2.3 Implementazione di Guo-Ma-Zhang (PQFT)

Sia dunque $F(t)$ l'immagine aquisita agli istanti $t = 1, 2, ..., N$, dove N è il numero totale di frame acquisiti. $r(t)$, $g(t)$ e $b(t)$ sono i canali di rosso, verde e blu di $F(t)$, da questi definiamo quattro ulteriori canali:

$$R(t) = r(t) - (g(t) + b(t))/2 \quad , \tag{2.15}$$

$$G(t) = g(t) - (r(t) + b(t))/2 \quad , \tag{2.16}$$

$$B(t) = b(t) - (r(t) + g(t))/2 \quad , \tag{2.17}$$

$$Y(t) = (r(t) + g(t))/2 - |r(t) - g(t)|/2 - b(t) \quad . \tag{2.18}$$

Chiaramente $Y(t)$ rappresenta il giallo; questi quattro colori servono per riprodurre il sistema di opposizione di colori presente nel nostro apparato visivo. Infatti

nel campo recettivo del cervello umano i neuroni eccitati da una lunghezza d'onda nel campo del rosso sono inibiti dal verde, e viceversa, lo stesso succede con il blu e il giallo. Quindi si vanno a creare tali opposizioni di colori, che serviranno come prime due features, andando a diventare due componenti del quaternione

$$RG(t) = R(t) - G(t) \quad , \tag{2.19}$$

$$BY(t) = B(t) - Y(t) \quad . \tag{2.20}$$

Le altre due componenti del quaternione sono l'intensità luminosa $I(t)$, e il movimento $M(t)$, definiti come:

$$I(t) = (r(t) + g(t) + b(t))/3 \quad , \tag{2.21}$$

$$M(t) = |I(t) - I(t + \tau)| \quad , \tag{2.22}$$

dove $\tau = 3$ è il coefficiente di latenza.
A questo punto viene creato il quaternione, con le quattro features ottenute:

$$q(t) = M(t) + RG(t)i + BY(t)j + I(t)k \quad , \tag{2.23}$$

dove i, j e k sono gli operatori complessi definiti in Eq: 2.5. Ora, tramite la scomposizione di Caley-Dickson (Eq: 2.11) viene riscritto $q(t)$ in forma *simplettica*:

$$q(t) = f_1(t) + f_2(t)j \quad , \tag{2.24}$$

con

$$f_1(t) = M(t) + RG(t)i \quad , \tag{2.25}$$

$$f_2(t) = BY(t) + I(t)i \quad . \tag{2.26}$$

Quindi, si esegue la QFT di $q(n, m, t)$ come in [25]:

$$Q[u, v] = F_1[u, v] + F_2[u, v]j \quad , \tag{2.27}$$

con

$$F_i[u,v] = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} e^{-i2\pi((mv/M)+(nu/N))} f_i(n,m) \quad , \tag{2.28}$$

dove (n,m) e (u,v) sono le posizioni dei pixel nel dominio dello spazio e della frequenza, rispettivamente, e M e N sono altezza e larghezza dell'immagine.
Ora viene messo $Q(t)$ in forma polare:

$$Q(t) = \|Q(t)\| e^{\mu\Phi(t)} \quad , \tag{2.29}$$

dove μ è un quaternione unitario puro, e $\Phi(t)$ è lo spettro di fase di $Q(t)$. Quindi, per le proprietà viste della Trasformata di fase, si pone $\|Q(t)\| = 1$, cosicchè si ottiene

$$Q'(t) = e^{\mu\Phi(t)} \quad , \tag{2.30}$$

che si va ad antitrasformare tramite

$$f_i(n,m) = \frac{1}{\sqrt{MN}} \sum_{v=0}^{M-1} \sum_{u=0}^{N-1} e^{i2\pi((mv/M)+(nu/N))} F_i[u,v] \quad , \tag{2.31}$$

che dà

$$q'(t) = a(t) + b(t)i + c(t)j + d(t)k \quad . \tag{2.32}$$

Ora non resta che esplicitare la saliency map:

$$sM(t) = g * \|q'(t)\|^2 \quad , \tag{2.33}$$

dove, al solito, g rappresenta un filtro gaussiano bidimensionale, con $\sigma = 8$.
Il metodo descritto produce dei risultati davvero notevoli, è testato molto bene, e oggettivamente confrontato con i vari algoritmi in letteratura. Abbiamo già anticipato i risultati di tali test, rispetto alla trattazione in frequenza i tempi computazionali vengono abbattuti, restando nell'ordine di grandezza di SR, e rispetto a quest'ultimo c'è un apprezzabile miglioramento di qualità, in talune immagini.

Capitolo 3

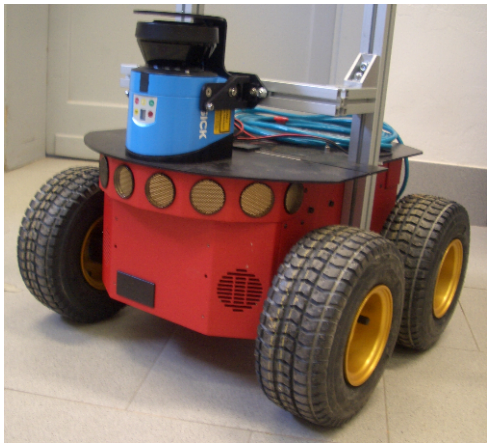
Implementazione

Il metodo implementato sfrutta il supporto matematico dei *quaternioni*, un'estensione dei numeri complessi, e la Trasformata di Fourier di fase definita su tale campo. L'approccio si basa su quello proposto da Guo, Ma e Zhang ([17]), che va a sfruttare la velocità dell'operare in frequenza, unitamente al concetto di features che offrono una visione dettagliata di elementi concettuali presenti nell'imagine.
L'elemento di novità, rispetto a [17], è la soppressione della componente descrivente il movimento, e sua sostituzione con una depth map, che meglio si adatta alle nostre necessità.
I risultati ottenuti sono molto interessanti, tanto dal punto di vista qualitativo, quanto da quello della velocità di esecuzione, il che rende il nostro algoritmo appetibile per un utilizzo real time, per il quale è destinato.

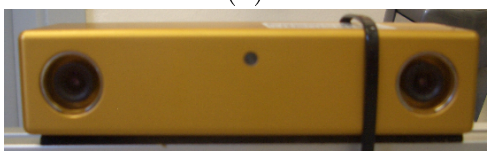
3.1 Strumenti

3.1.1 Hardware

La piattaforma robot utilizzata è la PIONEER 3-AT (P3-AT), commercializzata dalla *MobileRobot* ([29]), P3-AT è munito di 4 ruote motrici, azionate da 4 motori indipendenti e dotati di encoder , la velocità massima di navigazione si aggira sugli 8 metri al secondo, e possiede una capacità di carico di 12 Kg (Fig: 3.1 (a)).



(a)



(b)

Figura 3.1: Hardware:
(a)PIONEER 3-AT, (b)Bumblebee 2.

Inoltre è munito di 8 sensori sonar di distanza posteriori e altrettanti anteriori, con range dai 15 *cm* ai 7 *m*, ciò nonostante il nostro obbiettivo è di utilizzare solamente la visione come strumento di controllo della navigazione.
Inizialmente non era stato concepito l'utilizzo di una stereo camera, si pensava di equipaggiare il nostro robot con una normale fotocamera, ma si è scelta questa via in seguito all'impossibilità di utilizzare la feature legata al movimento. La stereocamera in questione è una *Bumblebee 2*, commercializzata dalla POINT GREY, equipaggiata con *1/3" progressive scan CCD* della Sony, che permette una risoluzione massima di 1032x776 pixels (Fig: 3.1 (b)).

3.1.2 Piattaforma software

Per sviluppare il nostro software si è scelto di lavorare con *MATLAB*, versione *R2010a*. MATLAB (abbreviazione di Matrix Laboratory) è un ambiente per il calcolo numerico e l'analisi statistica che comprende anche l'omonimo linguaggio di programmazione creato dalla *MathWorks*. MATLAB consente di manipolare matrici, visualizzare funzioni e dati, implementare algoritmi, creare interfacce utente, e interfacciarsi con altri programmi. Esso è usato da milioni di persone nell'industria e nelle università per via dei suoi numerosi tool a supporto dei più disparati campi di studio applicati, ed è questa una delle motivazioni che lo hanno reso appetibile anche ai nostri occhi. Infatti, dovendo lavorare con i quaternioni,

si è sfruttato un toolbox realizzato proprio a tale scopo, che ci ha notevolmente semplificato le cose. Inoltre, dovendo operare su immagini, che, dal punto di vista della macchina, altro non sono che matrici, la nostra scelta si è dimostrata felice. Il *Quaternion toolbox* estende MATLAB per consentire le operazioni con i quaternioni, esso sovrascrive numerose funzioni standard, al fine di trattare i quaternioni come fossero numeri reali o complessi. Il toolbox è molto completo, le funzioni implementate sono le più svariate, tra le quali tutte quelle riguardanti la Trasformata di Fourier, da noi utilizzate.
Il toolbox è stato sviluppato da Sangwine & Le Bihan al *Département Images et Signal, GIPSA-Lab* di Grenoble in Francia, nel 2005 (pagina web del progetto:[28]).

3.1.3 Depth map

Il ricorso alla depth map scaturisce da considerazioni sall'ambiente di utilizzo rispetto a [17], dove si analizza con telecamera fissa un ambiente in movimento; nel nostro caso la situazione è opposta, fotocamera in movimento e ambiente fisso. Nelle nostre condizioni si è quindi provato ad utilizzare il movimento come feature, ma fin dai primi test ci si è resi conto che nella maggior parte dei casi la saliency map risultante peggiorava rispetto al non utilizzo della quarta dimensione.
Si è quindi cominciato a pensare a come sostituire tale mancanza, e, avendo a disposizione una stereocamera, si è deciso di utilizzare una depth map, realizzata tramite disparità fra immagine destra e sinistra. Via software, dalle immagini destra e sinistra, vengono ricavate due ulteriori immagini: "rectified", ovvero l'immagine (destra o sinistra) cui è stata rimossa la distorsione radiale introdotta dall'ottica della fotocamera, e "disparity", ovvero la matrice che contiene le distanze dei punti 3D proiettati nell'immagine (depth map).
La realizzazione di queste immagini ha creato alcuni problemi, in quanto il software fornito con la stereocamera falliva nel loro calcolo. Il problema risiedeva nella non perfetta calibrazione dello strumento, si è così provveduto alla ricalibrazione, e alla realizzazione in proprio di rectify e disparity.
L'algoritmo implementato si basa sulla tecnica proposta da Konolige in [30], che

utilizza il paradigma della *correlazione fra aree*, che si articola in 5 punti:

1. Correzione Geometrica. In questo primo step le distorsioni delle immagini in ingresso vengono corrette mettendo tale immagine in "forma standard".

2. Trasformazione dell'Immagine. Ogni pixel dell'immagine viene normalizzato, tramite un operatore locale, secondo la media locale di intensità.

3. Correlazione d'Area. Ogni piccola area viena confrontata con le altre aree presenti nella sua finestra di ricerca.

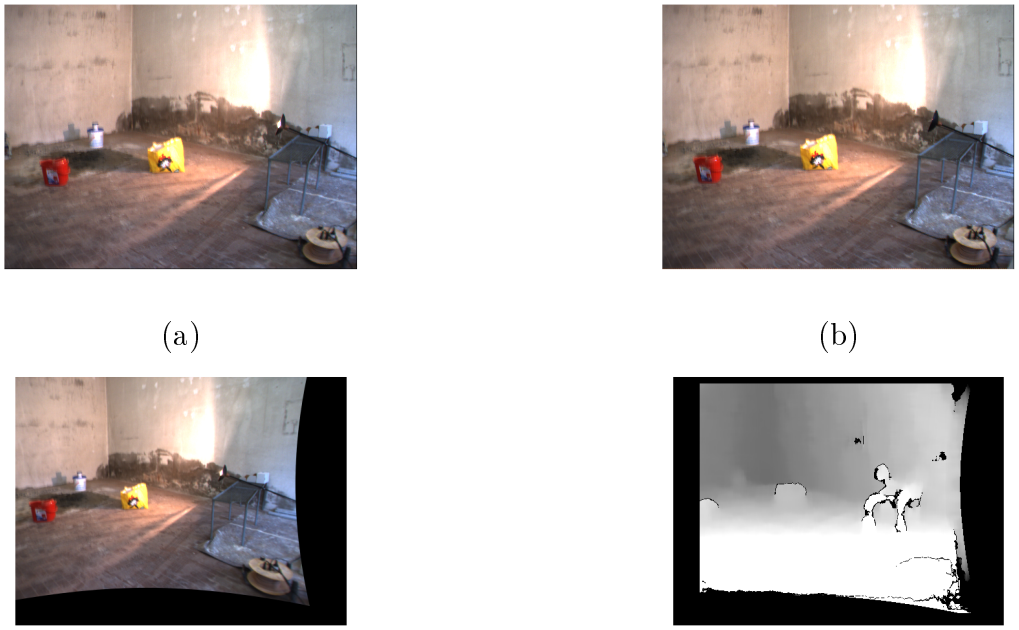
4. Estrazione degli Estremi. Attraverso il massimo valore di correlazione per ogni pixel è determinata la disparity map, dove i valori sono dati dalla differenza fra immagine destra e sinistra.

5. Post Filtraggio. Si passa attraverso un banco di filtri per ridurre il rumore.

La depth map (o disparity map) così ricavata va ad occupare la parte reale del quaternione dell'immagine, completandolo, e rendendolo acor più pregno di significato. La maggior potenza di questa immagine rispetto alle altre features risiede nel fatto che, oltre a non esser influenzata pesantemente dal colore racchiude un'informazione sul volume dei solidi e sulle distanze relative, che concettualmente è fondamentale per discriminare un oggetto.

3.2 Saliency detection tramite PQFT e depth map

La funzione implementata (*saliency_ depth*) prende in ingresso un'immagine, nel nostro caso è la "rectified", e la depth map corrispondente, la "disparity". Le due immagini hanno dimensioni differenti, e perciò vengono ridimensionate in modo da avere 64 pixel nella loro dimensione maggiore.
A questo punto, onde evitare i problemi di discontinuità della trasformata di Fourier ai bordi delle immagini, quest'ultime vengono filtrate in modo da uniformare i colori delle parti superiore e inferiore, come delle parti destra e sinistra.



(a) (b)

(c) (d)

Figura 3.2: (a) Immagine sinistra, (b) Immagine destra, (c) Rectified, (d) Disparity.

Ora, detta $F(t)$ l'immagine in ingresso, e $DM(t)$ la depth map corrispondente, andiamo a creare il nostro quaternione:

$$q(t) = DM(t) + RG(t)i + BY(t)j + I(t)k \quad . \tag{3.1}$$

Definendo $r(t)$, $g(t)$ e $b(t)$ come i canali di rosso, verde e blu di $F(t)$, andiamo a esplicitare i termini del quaternione come visto in 2.19 - 2.21:

$$\begin{aligned} RG(t) &= R(t) - G(t) \quad , \\ BY(t) &= B(t) - Y(t) \quad . \\ I(t) &= (r(t) + g(t) + b(t))/3 \quad . \end{aligned}$$

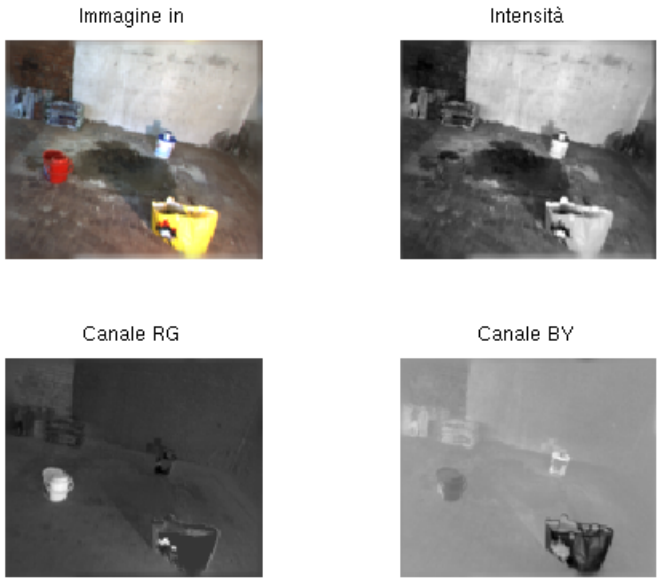


Figura 3.3: Immagine in ingresso con i canali di Intensità, e i due canali di opposizione R/G e B/Y.

Applichiamo ora la QFT come in [17], esplicitata in forma polare:

$$Q(t) = \|Q(t)\|e^{\mu\Phi(t)} \quad . \tag{3.2}$$

Ponendo $\|Q(t)\| = 1$, onde isolare la sola informazione di fase, otteniamo,

$$Q'(t) = e^{\mu\Phi(t)} \quad , \tag{3.3}$$

dalla quale per antitrasformazione:

$$q'(t) = a(t) + b(t)i + c(t)j + d(t)k \quad . \tag{3.4}$$

Si calcola a questo punto la saliency map:

$$sM(t) = g * \|q'(t)\|^2 \quad , \tag{3.5}$$

dove g è il consueto filtro gaussiano bidimensionale, con $\sigma = 8$.

Eseguendo la binarizzazione della saliency map andiamo visivamente ad indicare le parti dell'immagine originale di maggior salienza per il nostro algoritmo.

3.2.1 Problematiche e dettagli implementativi

Data quindi la formulazione matematica della strategia, si è passati all'effettiva implementazione in Matlab. Si è, inanzitutto, implementato lo stato dell'arte, nel nostro caso si è scelto PQFT, essendo, a nostro avviso, il giusto compromesso fra tempi d'esecuzione, e qualità dei risultati ottenuti.
L'implementazione non si è rivelata particolarmente insidiosa, si è solo avuto un piccolo, e noto problema, sulla discontinuità della trasformata di Fourier bidimensionale ai bordi delle immagini. Infatti, l'operazione di trasfomata è "circolare", e quindi bordo destro e sinistro (come superiore e inferiore) sono contigui nell'-operazione, e quindi la loro discontinuità risulta molto rilevante, e fa riscontrare grandi erronee salienze lungo i bordi dell'immagine. Il problema è stato risolto andando a creare una fascia di pochi pixel lungo tutto il perimetro dell'immagine, che creassero una scala graduale fra gli opposti bordi.
Un altro problema si è avuto con la feature del movimento dell'algoritmo originale, risultava infatti fuorviante ai fini dell'obbiettivo, ed è stata eliminata, e sostituita dalla depth map.
Una volta sistemati questi problemi si è lavorato sulla risoluzione, e sui parametri del filtro di Gabor, per trovare la combinazione che desse i migliori risultati nella saliency map, e nei tempi di esecuzione.

Capitolo 4

Test e risultati

La fase di test si è svolta parallelamente alla creazione dell'algoritmo, e ne ha fortemente indirizzato lo sviluppo in tutta la sua evoluzione.
Inizialmente, infatti, si sono utilizzati alcuni toolbox a disposizione on-line([31], [32]), e testati direttamente al fine di validare i dati dichiarati dai vari autori, e verificare qualitativamente i risultati su delle immagini a nostra disposizione. Dai risultati di questi primi test si è decisa la strategia da seguire, si è deciso per l'approccio basato su trasformata di fase e quaternioni, proposto in [17]. In questo articolo infatti, la fase di test è descritta con dovizia, e confronta i propri risultati con quelli di tutti i principali articoli presenti in letteratura, fornendo anche le immagini di tutte le saliency maps (Fig: 4.1). I risultati ottenuti ci sono sembrati oggettivamente rilevanti per l'applicazione, sebbene il nostro ambito applicativo differisse abbastanza dalle immagini utilizzate nei test. Infatti le immagini in questione, pur essendo strutturate come servono a noi, ovvero con alcuni oggetti isolati su un background abbastanza omogeneo, sono tutti scenari esterni, mentre noi opereremo all'interno di una stanza.

4.1 Movimento *vs.* depth map

Una volta implementato il nostro algoritmo si è subito passati al controllo della sua qualità, i primi risultati sono stati sconfortanti. Necessitando di immagini consecutive di poco differenti si sono estratti frame molto vicini da una sequenza

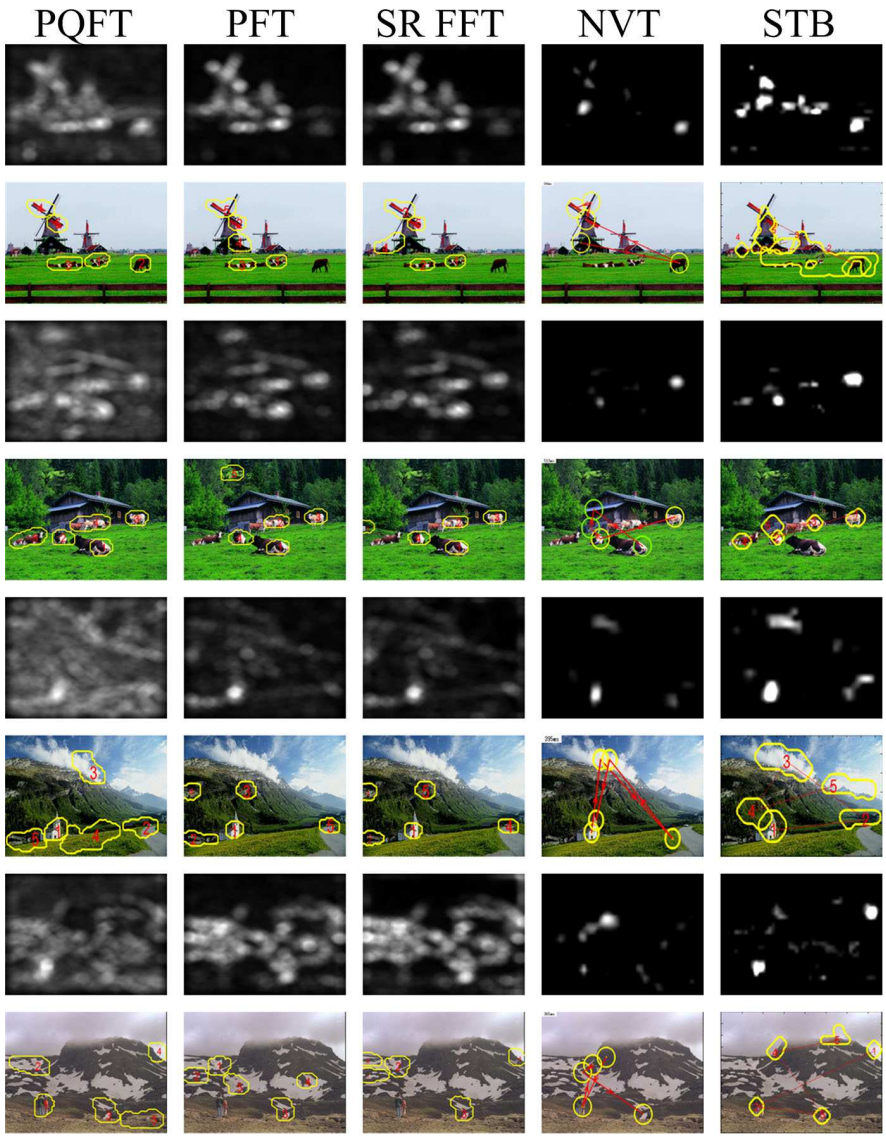


Figura 4.1: Confronto fra diverse tecniche di saliency detection. PQFT=metodo di Guo-Ma-Zhang; PFT=metodo in frequenza tramite trasformata di fase; SR=spectral residual; NVT=metodo di Walther-Kock; STB=Saliency Tool Box (Itti lab).

video, per compiere i primi test. Tuttavia, mentre in [17] si utilizza una telecamera fissa, e un ambiente in movimento, nel nostro caso è esattamente l'opposto: fotocamera su robot, in movimento, e contesto immobile. In questo contesto lo stesso concetto di movimento cambia significato, mentre un oggetto che si

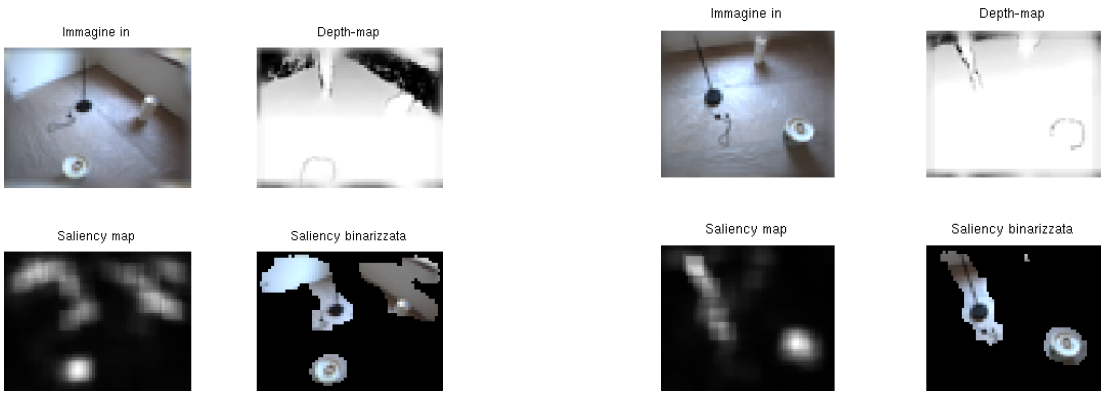
muove rispetto a un osservatore fisso può attrarre l'attenzione, se è l'osservatore a muoversi il movimento degli oggetti è solo apparente e non porta informazione. Infatti, si è notato come i risultati migliorassero senza utilizzare tale feature.
Dopo un'attenta analisi delle possibili features alternative da utilizzare al posto del movimento, si è scelto di utilizzare una stereocamera e creare le depth map. La stereocamera in questione (descritta in 3.1.1) è fornita con il software per il calcolo della depth map, ci si è quindi accinti alla creazione di un dataset di immagini con mappe stereo per la verifica. Infatti, l'utilizzo di una depth map nel campo della saliency detection non era ancora stato tentato, e non era quindi ipotizzabile quali sarebbero stati i risultati.

4.1.1 Acquisizione immagini

Creare un dataset di immagini con la stereocamera è stato uno degli step più dispendiosi dal punto di vista dei tempi, abbiamo avuto diversi problemi, avendo delle necessità ben precise, e non comunemente considerate.
E' stato creato un primo dataset, realizzato all'interno del laboratorio di robotica autonoma dell'Università di Padova, con la Bumbelbee2. Si sono create diverse scene, tutte strutturate in modo da contenere diversi oggetti, in un ambiente il più disadorno possibile, il che è stato piuttosto difficoltoso, vista la caoticità di una stanza normalmente utilizzata.
Purtroppo il software fornito con la stereocamera ha creato delle depth map inesatte, causate presumibilmente da una cattiva calibrazione. Quindi si è utilizzato un software che ci ha fonito i parametri di calibrazione, attraverso i quali è stata implementata una funzione che, prese immagini destra e sinistra, ricalcolasse la disparity map.
Ottenute quindi le depth map, si è finalmente potuto sperimentare il nostro algoritmo, ma, ancora una volta i risultati sono stati deludenti, a causa dei molti pixel in cui la disparity non può essere calcolata, e quindi colorati di nero. Questo fatto è dovuto all'impossibilità di prendere dei riferimenti all'interno dell'immagine, vuoi per la presenza di estesi blob di colore omogeneo, vuoi per la scarsa luminosità dell'ambiente. L'impatto di queste "zone nere" è, per l'algoritmo, molto

negativo, come si evince in Fig.: 4.2, poichè tali zone risultano, erroneamente, le più salienti.

Inoltre, non avendo uno spazio vuoto sufficientemente ampio in laboratorio, gli



(a) (b)

Figura 4.2: Applicazione di *saliency_stereo* a due immagini simili. (a) Immagine con depth map molto viziata dalla presenza di pixel neri (per i quali non si è potuta calcolare la distanza) (b) Immagine priva di tale problema. L'immagine ci permette di osservare come una depth map rumorosa introduca parecchie zone erroneamente salienti, anche in ambiente molto strutturato.

oggetti presenti avevano distanze relative piuttosto limitate, e questo fatto ha prodotto delle depth map dallo scarsissimo significato.

A questo punto si è necessariamente dovuto riaquisire un dataset maggiormente ragionato, non avendo un'idea precisa di come elaborare le nostre saliency maps così rumorose. E' stato così trovato un sito adatto, e ricreato un dataset maggiormente strutturato e molto meno rumoroso, dal quale si sono estratte alcune immagini nelle quali la quasi totalità dei pixel presentava un corretto calcolo della distanza. Fatto ciò si è provveduto all'eliminazione di un ultimo, noto, inconveniente, la leggera traslazione tra disparity e rectify.

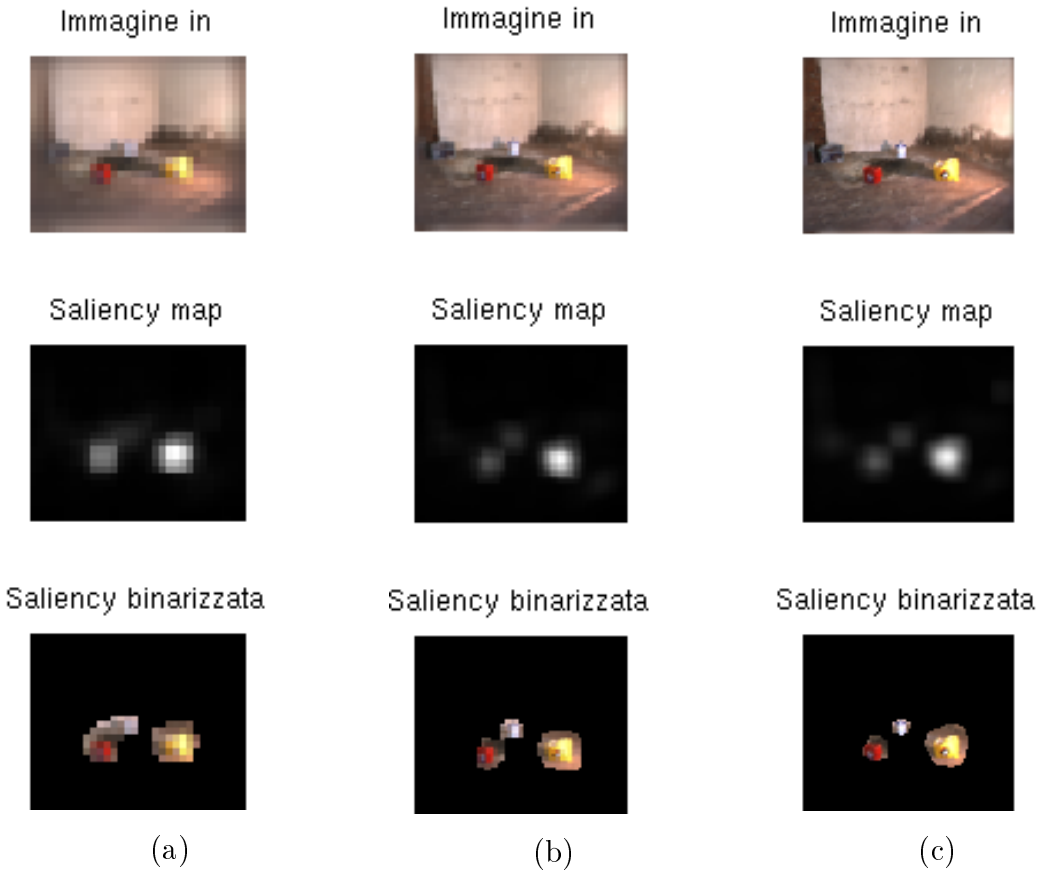


Figura 4.3: Scelta della risoluzione: (a)Risoluzione a 32 pixel orizzontali, (b)64 pixel, (c)128 pixel.

4.2 Qualità e tempi di esecuzione

Finalmente si è potuto passare all'effettivo test sull'efficienza del nostro algoritmo. I parametri da analizzare, nel nostro caso, sono due, la qualità del riconoscimento realizzato, e il tempo di esecuzione, determinante per utilizzo real time. Purtroppo, avendo introdotto la depth map come feature non ci siamo potuti confrontare con la letteratura sulla qualità del riconoscimento, ma possiamo solamente testare secondo quella che è l'evidenza delle salienze di un'immagine per l'uomo.
I parametri sui quali si è giocato per valutare differenti modalità di utilizzo sono essenzialmente, la risoluzione dell'immagine e i parametri del filtro gaussiano tramite il quale la saliency map è realizzata, e che vanno adattati alla risoluzione. Come si può notare in Fig.: 4.3 la diminuzione di risoluzione, entro un certo

limite, non inficia i risultati, che restano molto buoni.
Ovviamente i tempi di esecuzione calano al calare della risoluzione, così, avendo risultati consistenti anche con soli 64 pixel sull'asse maggiore, la prendiamo come risoluzione di riferimento. Tali tempi si aggirano sui 50 millisecondi, confrontabili con quelli di [17], ampiamente utilizzabili per i nostri intenti. Il calcolo di disparity e di rectify invece è maggiormente dispendioso, circa 200 millisecondi, mantenendo comunque il tempo di esecuzione entro i limiti richiesti.

4.2.1 Valutazione comparativa

Nel complesso l'algoritmo presentato mostra una buona qualità nel rilevamento delle salienze (Es. Fig:4.4), mentre i tempi di esecuzione sono competitivi. Rispetto agli altri metodi presenti in letteratura, i tempi di esecuzione si collocano all'incirca a metà tra quelli, elevati, dell'approccio nel dominio dello spazio, e quelli, molto più contenuti, nel dominio della frequenza. Il maggior dispendio temporale, a fronte di un calcolo della trasformata del quaternione molto veloce (50 millisecondi), si ha nella computazione della depth map, che impiega circa 200 millisecondi. Tuttavia anche se c'è un sostanziale aumento dei tempi computazionali tra il nostro algoritmo e PQFT, che è quello che maggiormente somiglia al nostro, essi restano comunque accettabili per i fini del progetto.

Rispetto agli approcci nello spazio, si ha una diminuzione consistente dei tempi, come si è già avuto modo di sottolineare, che ci permettono di affermare che tale metodo sia, allo stato dei fatti, ampiamente superato.
Se invece ci andiamo a misurare con un approccio puramente in frequenza (si è scelto SR), i tempi computazionali del nostro metodo non reggono il confronto, essendo nell'ordine di 4-5 volte superiori. Tuttavia, se si vanno ad analizzare con dovizia le saliency maps prodotte dai due algoritmi (Fig: 4.5) si notano alcune differenze qualitative non trascurabili. Infatti il nostro approccio sembra più preciso nel discriminare due oggetti vicini, mentre in SR se ne nota uno dominante che "mette in ombra" l'altro. Inoltre, in generale, l'algoritmo di spectral residual, tende ad essere più dispersivo nell'assegnazione delle salienze, questo perchè mette in risalto ogni singolarità dell'immagine, senza mediare tra features differ-

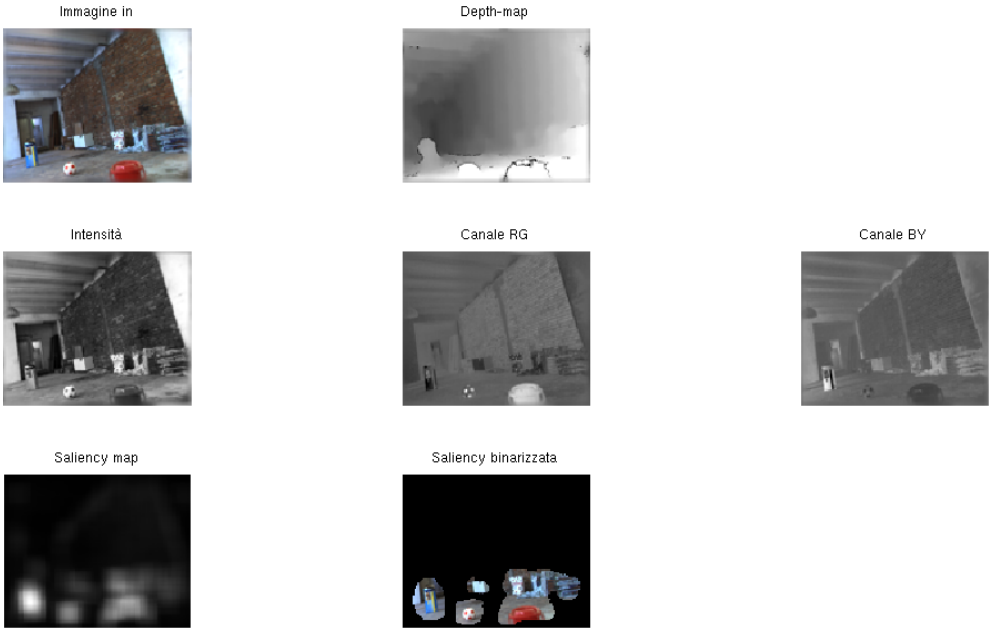


Figura 4.4: Immagine in ingresso, componenti del quaternione e risultati.

enti, come fa la nostra funzione, andando così a concentrarsi su quei punti che sono rilevanti per tutte le features componenti. Questo fatto fa si che il nostro algoritmo risulti più efficiente, con minor falsi allarmi, e maggior numero oggetti ritrovati.

Possiamo quindi dirci soddisfatti del lavoro svolto, che unisce una precisione computazionale oggettiva, a tempi di esecuzione utilizzabili per l'applicazione real time di interesse.

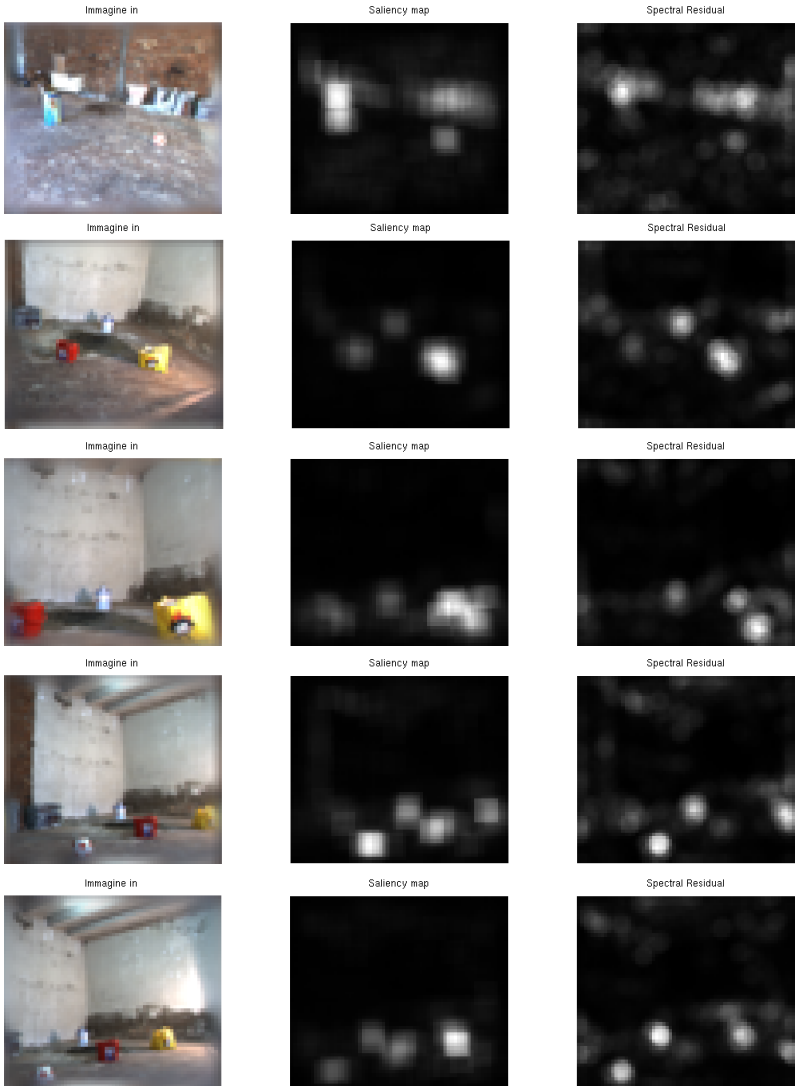


Figura 4.5: Confronto fra la saliency map prodotta dal nostro algoritmo (colonna centrale) e quella prodotta dal SR.

Capitolo 5

Conclusioni

5.1 Metodologia

In questa tesi di laurea si è presentato un approccio al problema di saliency detection tramite un algoritmo innovativo.
In [17] viene sviluppata una tecnica di saliency detection basata sulla trasformata di fase e sul supporto matematico dei quaternioni, ed è proprio da questo articolo che si è presa l'ispirazione per sviluppare la nostra funzione. I quaternioni sono una derivazione dei numeri com plessi, con la differenza che essi rappresentano uno spazio vettoriale a quattro dimensioni, anzichè due. Quindi è possibile formare un quaternione con quattro differenti features come componenti, tre delle quali sono classiche: il canale dell'intensità luminosa dell'imagine da processare, il canale di opposizione dei colori Rosso/Verde e Blu/Giallo. Queste features sono utilizzate fin dai primi modelli, nel dominio dello spazio, di saliency detection ([10], [15]), ma riprese pure dai più recenti lavori a riguardo.
In aggiunta a tali features, in [17], viene completato il quaternione con il movimento, dato dalla differenza fra due frame ravvicinati, e, per l'applicazione considerata ha dato ottimi risultati.
Il nostro approccio invece, si è differenziato per l'abbandono della feature del movimento, in quanto risultava inutile, se non addirittura fuorviante per l'applicazione. In sua vece si è aggiunta una depth map come feature a completamento del quaternione, e i risultati riscontrati si sono dimostrati consistenti. Tale depth map è stata realizzata tramite la stereocamera Bumbleebee 2, che acquisisce ad

ogni frame una coppia di immagini (destra e sinistra), che sono poi processate da una funzione che restituisce la nostra depth map.
Nei nostri test, le salienze visuali sono state identificate con grande precisione, evidenziando alcuni piccoli, ma importanti miglioramenti rispetto alla letteratura. In particolare, gli oggetti vengono messi in luce con grande precisione, senza tralasciarne alcuno, e il problema dei falsi ritrovamenti è quasi assente. Inoltre, anche in presenza di oggetti che occupino porzioni considerevoli dell'immagine, l'informazione volumetrica data dalla depth map consente di vederli come un tutto, e non come punti di salienza vicini, ma scollegati, come talvolta capita con gli algoritmi in letteratura. Inoltre, sempre grazie a tale informazione volumetrica, viene meno una delle maggiori "distrazioni" per la saliency detection, le superfici di diversi colori, che risultando alla stessa distanza vanno a perdere parte della loro rilevanza. Lo stesso discorso vale per oggetti che abbiano un colore simile al background, che vengono evidenziati dalla differente distanza.
Dal punto di vista dei tempi di esecuzione il nostro algoritmo si va a collocare a cavallo tra i due approcci classici, essendo il calcolo della depth map abbastanza dispendioso. Ciononostante un tempo medio di computazione sui 200 millisecondi ci mantiene tranquillamente nei limiti imposti dalla nostra applicazione.
L'algoritmo è ottimizzato per Matlab, sfruttando le sue grandi performance nelle operazioni tra matrici, ed eliminando per quanto possibile i cicli, che ne rallentano notevolmente la velocità di esecuzione.

5.2 Problematiche e sviluppi futuri

Tuttavia esistono diverse problematiche che vanno affrontate ed esposte, e che dovranno per forza essere risolte se si vuole ottimizzare l'algoritmo presentato.
Il principale problema riscontrato, a cui già si è accennato, è la presenza di pixel "neri" nella depth map. Il fatto è dovuto alla non presenza nella zona interessata di punti distinguibili tra loro (per esempio un muro bianco), e l'impossibilità di eseguire un confronto, in quel punto, fra immagine destra e sinistra, non riuscendo quindi a calcolare la distanza. Queste zone di impossibilità di calcolo della depth map sono oltremodo deleterie per il nostro obbiettivo, andando a fuorviarlo in maniera massiva, rendendolo in tal caso inservibile. Questa circostanza è

imputabile al fatto che la saliency map mette in risalto le discontinuità all'interno dell'immagine, e queste macchie nere all'interno di quest'ultima, sono delle discontinuità esageratamente rilevanti. Certamente, se si ha intenzione di utilizzare il metodo esposto, è questo il primo, e più importante, problema da risolvere. In alternativa, qualora il problema non fosse superabile, bisognerebbe eseguire un controllo sulla quantità di pixel di cui non sia riscontrabile la distanza, e trovare una soglia sopra la quale escludere tale feature dal quaternione.
Inoltre, i tempi di computazione, pur restando in limiti accettabili, sono fortemente condizionati dal calcolo della depth map, ed è quindi necessario pensare ed implementare una funzione ottimizzata delegata allo scopo.

5.2.1 Considerazioni soggettive

Il metodo implementato è certamente robusto, i quaternioni si sono rivelati un supporto adeguato all'applicazione, e la qualità dei risultati è buona. Tuttavia l'analisi e la comprensione di ciò che dalla trasformata di fase è messo in evidenza, deve andare più a fondo, in tal modo potremo arrivare alla definizione di features che siano più influenti sul risultato finale, e indipendenti l'una dall'altra.
Infatti, pur riproducendo i meccanismi della visione umana, i canali di intensità, e di opposizione RG e BY, dal punto di vista della teoria dei segnali, sono portatori di informazioni simili tra loro, quando non addirittura ridondanti. A tal proposito la depth map si distingue nettamente da ogni altra feature considerata, andando a condensare in se stessa una doppia informazione, quella cioè delle immagini destra e sinistra. Inoltre va a sviluppare il concetto di profondità, che è concettualmente superiore alle informazioni contenute in ogni altra possibile immagine fino ad ora considerata.

Appendice

In questa appendice è riportato il codice sviluppato, scritto in linguaggio Matlab, con i relativi commenti. L'algoritmo riceve in ingresso un'immagine, con la relativa depth map calcolata off-line, e computa la saliency map, e la sua versione binarizzata.
La risoluzione scelta è di 64 pixel (sul lato maggiore dell'immagine), anche per la depth map viene eseguito lo stesso ridimensionamento, in quanto le componenti del quaternione devono avere le medesime dimensioni.

```
% saliency detection con quaternioni e depth map
function saliency_stereo_64(img_name, dept)

        % leggo immagini
img_in = imread(img_name);
depth = imread(dept);

        % ridimensiono immagini
i_size = 64;
w = size(img_in,2);
h = size(img_in,1);
if( w > h)
    ratio = w/h;
    w = i_size;
    h = round(i_size/ratio);
else
    ratio = h/w;
    h = i_size;
    w = round(i_size/ratio);
```

```
end
img = imresize(img_in, [h w],'bicubic');
depth = imresize(depth, [h w],'bicubic');

        % smusso i bordi dell'immagine per non dare
        % discontinuit  alla trasformata
PSF = fspecial('gaussian',6,15);
img = edgetaper(img,PSF);
depth = edgetaper(depth,PSF);

        % dimensioni matrice dell'immagine
[M,N,O] = size(img);

        % porto valori a double
img = double(img)/255;
depth = double(depth)/255;

        % creo canali di R, G, B, Y, e intensit

img_R = img(:, :, 1) - (img(:, :, 2) + img(:, :, 3))/2;
img_G = img(:, :, 2) - (img(:, :, 1) + img(:, :, 3))/2;
img_B = img(:, :, 3) - (img(:, :, 1) + img(:, :, 2))/2;
img_Y = (img(:, :, 1) + img(:, :, 2))/2 -
          (abs(img(:, :, 1) - img(:, :, 2)))/2 - img(:, :, 3);


img_I = (img(:, :, 1) + img(:, :, 2) + img(:, :, 3))/3;


        % creo canali RG, BY, GR e YB
img_RG = img_R - img_G;
img_BY = img_B - img_Y;
img_GR = img_G - img_R;
img_YB = img_Y - img_B;
```

```
        % formo il quaternione
q = quaternion(depth, img_RG, img_BY, img_I);

        % trasformata del quaternione
Q = fft2(q);

        % Q_primo     Q in forma polare con modulo settato a 1
Q_primo = exp(axis(Q) .* angle(Q));

        % antitrasformo Q_primo
q_primo = ifft2(Q_primo);

        % modulo di q_primo
mod = abs(q_primo);

        % creiamo saliency-map
sal = mod .^ 2;
g = fspecial('gaussian', 7 , 8);
sal_map = imfilter(sal, g,'replicate');

        % normalizzo per visualizzare

min_s = min(min(sal_map));
max_s = max(max(sal_map));
range = max_s - min_s;
sal_map_n = uint8(255 * (double(sal_map) - min_s) ./ range);

min_s = min(min(img_RG));
max_s = max(max(img_RG));
range = max_s - min_s;
img_RG = uint8(255 * (double(img_RG) - min_s) ./ range);

min_s = min(min(img_BY));
max_s = max(max(img_BY));
range = max_s - min_s;
```

```
img_BY = uint8(255 * (double(img_BY) - min_s) ./ range);

min_s = min(min(img_I));
max_s = max(max(img_I));
range = max_s - min_s;
img_I = uint8(255 * (double(img_I) - min_s) ./ range);

min_s = min(min(depth));
max_s = max(max(depth));
range = max_s - min_s;
depth = uint8(255 * (double(depth) - min_s) ./ range);

        % saliency bin show
sal_map_b = img;
for m=1:M
        for n=1:N

                if(sal_map_n(m, n)<40)
                        sal_map_b(m, n, 1) = 0;
                        sal_map_b(m, n, 2) = 0;
                        sal_map_b(m, n, 3) = 0;
                end
        end
end

        % Visualization
subplot(2,2,1); imshow(img); title('Immagine in');
subplot(2,2,2); imshow(depth); title('Depth-map');
subplot(2,2,3); imshow(sal_map_n); title('Saliency map');
subplot(2,2,4); imshow(sal_map_b); title('Saliency binarizzata');

end
```

Bibliografia

[1] Merger D., Forss P., Lai K., Helmer S., McCann S., Southey T., Baumann M., Little J., Lowe D., Dow B.; Curious George: An Attentive Semantic Robot; 2007

[2] Roland Siegwart, Illah Reza Nourbakhsh; Introduction to autonomous mobile robots MIT press; 2004

[3] Website: http://www.semantic-robot-vision-challenge.org/

[4] Pretto Alberto; Visual-SLAM for Humanoid Robot; PHD Thesis; 2009

[5] Sivic J. and Zisserman A.; Video google: A text retrieval approach to object matching in videos; Proceedings of the International Conference on Computer Vision; 2003

[6] Nister D. and Stewenius H.; Scalable Recognition with a Vocabulary Tree; Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2006

[7] Philbin J., Chum O., Isard M., Sivic J., Zisserman A.; Object retrieval with large vocabularies and fast spatial matching; Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2007

[8] Treisman A. M., Gelade G; A feature-integration theory of attention; Cognitive Psychology, 12(1), 97–136; 1980

[9] Tsotsos J. K., Culhane S. M., Wai W. Y. K., Lai Y. H., Davis N., Nuflo F.; Modeling visual-attention via selective tuning; Artificial Intelligence, 78, 507–545; 1995

[10] L. Itti, C. Koch, E. Niebur; A Model of Saliency-Based Visual Attention for Rapid Scene Analysis; IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(11):1254–1259; 1998

[11] L. Itti and C. Koch; Saliency-Based Search Mechanism for Overt and Covert Shifts of Visual Attention; A Vision Research, 40(10-12):1489–1506; 2000

[12] L. Itti and C. Koch; Computational Modelling of Visual Attention; Nature Reviews Neuroscience, 2(3):194–203; 2001

[13] Rensink R. A.; Seeing, sensing, and scrutinizing; Vision Research, 40(10–12), 1469–1487; 2000

[14] Rensink R. A.; The dynamic representation of scenes; Visual Cognition, 7(1/2/3), 17–42; 2000

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

[16] X. Hou and L. Zhang; Saliency detection: A spectral residual approach; IEEE Conference on Computer Vision and Pattern Recognition (CVPR07); 2007

[17] Guo C., Ma Q., Zhang L.; Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform; IEEE Conf. on Computer Vision and Pattern Recognition; 2008

[18] Feldman J.A.; Dinamic connections in neural networks; Biol Cybern 46:27-39; 1982

[19] Koch C., Ullman S.; Shifts in selective visual-attention — towards the underlying neural circuitry; Human Neurobiology, 4, 219–227; 1985

[20] A. van der Schaaf and J. van Hateren; Modelling the Power Spectra of Natural Images: Statistics and Information; Vision Research, 36(17):2759–2770; 1996

[21] D. Ruderman; The Statistics of Natural Images; Network: Computation in Neural Systems, 5(4):517–548; 1994

[22] J. Gluckman; Order Whitening of Natural Images; Proc. CVPR, 2; 2005

[23] K. Castleman; Digital Image Processing; Prentice-Hall, New York; 1996

[24] W. R. Hamilton; The outlines of Quaternions; London Longmans, Green; 1843

[25] T. Ell and S. Sangwin; Hypercomplex Fourier Transforms of Color Images; IEEE Transactions on Image Processing, 16(1):22-35; 2007

[26] T. A. Ell; Hypercomplex Spectral Transforms; Ph.D. dissertation, Univ. Minnesota, Minneapolis; 1992

[27] S. J. Sangwine and T. A. Ell; The discrete Fourier transform of a colour image; Proc. Image Processing II Mathematical Methods, Algorithms and Applications, J. M. Blackledge and M. J. Turner, Eds., Chichester, U.K., pp. 430–441; 2000

[28] http://sourceforge.net/projects/qtfm/

[29] http://www.mobilerobots.com/PDFs/P3ATDX%20Datasheet.pdf

[30] K. Konolige; Small Vision Systems: hardware and implementation; Eighth International Symposium on Robotics Research, Hayama, Japan; 1997

[31] http://ilab.usc.edu/toolkit/home.shtml

[32] http://www.saliencytoolbox.net/