Università degli Studi dell'Insubria
Facoltà di Scienze MM.FF.NN.
Corso di Laurea Specialistica in Informatica

Studio e sviluppo di un nuovo algoritmo di stereo matching basato su reti neurali autorganizzanti

Relatore: Dott. **IGNAZIO GALLO**
Correlatore: Prof. **ELISABETTA BINAGHI**

Tesi di Laurea Specialistica di
MARCO VANETTI
Matricola: 705145

Anno Accademico 2008-2009

A Laura e a me,
per la reciproca sopportazione durante la stesura delle tesi.

Ai miei genitori, Alessandro e Antonietta,
per avermi supportato e motivato durante gli studi.

A mio fratello Luca,
per le svariate pizze e i cinema tra un esame e l'altro.

Studio e sviluppo di un nuovo algoritmo di stereo matching basato su reti neurali autorganizzanti

Marco Vanetti

marco.vanetti@gmail.com

Elaborato finale per il corso di
Laurea Specialistica in Informatica
Sessione di Laurea Autunnale
(14 ottobre 2009)

Sommario

Nell'ambito dell'analisi di immagini stereoscopiche, questo lavoro mira a definire una importante e innovativa estensione di un metodo di stereo matching basato su reti neurali autorganizzanti. Alla base del modello originale, proposto nel 2007 da *Venkatesh, Y.V. et all* in [40], vi è una *Self-Organizing Map*, opportunamente modificata al fine di risolvere il problema delle corrispondenze stereo e produrre mappe di disparità dense. Seppur affascinante e innovativo dal punto di vista metodologico, il modello originale non permetteva di raggiungere risultati di interesse operativo. Le estensioni e le modifiche al modello proposto in [40], ispirate a recenti ed efficienti tecniche di stereo matching, migliorano notevolmente la qualità dei risultati pur mantenendo intatta la struttura di base, caratterizzata da un approccio neurale autorganizzante. Analizzando la letteratura sullo stereo matching si scopre come i metodi basati su analisi locali richiedano generalmente minori tempi di elaborazione a scapito di una qualità dei risultati minore, se confrontata con quella ottenuta da algoritmi che trovano le corrispondenze analizzando globalmente le immagini. Pur collocandosi tra gli algoritmi di corrispondenza locali, con notevoli vantaggi dal punto di vista della semplicità e dell'efficienza, il modello descritto fornisce ottimi risultati, che superano in qualità molti algoritmi di stereo matching globali proposti negli ultimi anni.

Parole chiave:

Visione stereo, stereo matching, corrispondenza stereo,
Self-Organizing Map, mappe di disparità, gestione delle occlusioni

Prefazione

In questo elaborato viene proposta ed analizzata una innovativa estensione di un modello neurale basato su *Self-Organizing Maps* che si presta a risolvere una questione fondamentale della visione stereo: il problema delle corrispondenze. Dal punto di vista operativo il modello esteso, che chiameremo *StereoSOM*, rientra nella categoria degli algoritmi di *stereo matching denso*[1].

È noto come il cervello umano sfrutti la visione stereoscopica per darci maggiori informazioni sulla profondità degli oggetti che guardiamo. Questo, unitamente al fatto che esistono all'interno del cervello umano alcune aree che adottano comportamenti simili alle reti *SOM*[2], rende *StereoSOM* affascinante e, almeno dal punto di vista concettuale, biologicamente plausibile.

Ritornando alla classe dei metodi di stereo matching è possibile specificare due sottoinsiemi di algoritmi, a seconda di come il metodo opera sull'immagine.

Si dicono *locali* quei metodi che, ad ogni passo dell'elaborazione, per trovare una corrispondenza operano solo su una piccola area delle due immagini. Come il metodo operi localmente varia molto da algoritmo a algoritmo, alcuni metodi utilizzano una funzione di similarità o dissimilarità applicata ad una piccola finestra di punti, altri considerano il singolo punto, altri ancora un insieme di punti dipendente dal contesto locale.

Si dicono invece *globali* i metodi che considerano il problema della ricerca delle corrispondenze come un problema di minimizzazione di una funzione energia globale, basata sull'intera immagine. Solitamente gli algoritmi globali considerano

[1] Gli algoritmi di stereo matching sono anche detti algoritmi di corrispondenza stereo; la dicitura *denso* indica che l'algoritmo dovrà trovare le corrispondenze per ogni punto della coppia di immagini stereo e non, come accade per un'altra categoria di algoritmi detti *basati sulle feature*, solo per alcuni punti specifici.

[2] In realtà fu Teuvo Kohonen, uno dei primi ad occuparsi di reti *SOM*, ad ispirarsi al cervello umano nella definizione del suo modello di rete neurale artificiale.

l'intera immagine come un grafo e usano strategie approssimate per minimizzare la funzione energia, poiché una ricerca esaustiva del minimo sarebbe troppo onerosa dal punto di vista computazionale. La qualità dei risultati ottenibili con metodi globali è solitamente superiore a quella che gli algoritmi di tipo locale possono fornire, anche se questi ultimi sono decisamente più semplici dal punto di vista algoritmico ed efficienti in termini di tempi di esecuzione. Inoltre i metodi locali, per definizione stessa, possono essere facilmente ottimizzati per l'elaborazione parallela, diventando ancora più efficienti su alcune architetture hardware.

StereoSOM è un metodo di tipo locale ma, sorprendentemente, si vedrà come la qualità dei suoi risultati superi in accuratezza quella di molti algoritmi globali presentati negli ultimi anni. Questo rende di fatto il metodo proposto in questo elaborato uno dei più accurati algoritmi locali di corrispondenza stereo.

Un ulteriore merito di *StereoSOM* è quello di non impiegare, a differenza della maggior parte degli algoritmi di stereo matching denso, nessuna fase esplicita di segmentazione, inoltre non è prevista alcuna fase di *estrazione delle feature*: le immagini in input vengono utilizzate direttamente dall'algoritmo e, alla fine dell'elaborazione, le mappe di disparità dense vengono estratte direttamente dai pesi neurali della rete autorganizzante.

Il contributo finale di questa tesi di laurea è senza dubbio rappresentato dalla parte strutturale e algoritmica di *StereoSOM* e dall'analisi dei risultati ottenuti, tuttavia è stata redatta nei primi due capitoli una parte introduttiva mirata a presentare gli aspetti più generali dei sistemi stereoscopici insieme ad alcuni passi fondamentali nella letteratura corrente sullo stereo matching. Nella parte introduttiva riguardante il problema della corrispondenza stereo sarà data più enfasi a quei punti che con *StereoSOM* trovano maggior contatto, con l'intenzione di portare il lettore alla descrizione del modello finale con una chiara e forte consapevolezza delle problematiche che si dovranno affrontare.

Nei capitoli successivi sarà poi discusso l'algoritmo vero e proprio, partendo dalla descrizione iniziale su cui è basato l'intero lavoro, passando per la descrizione delle singole estensioni e arrivando infine al modello *StereoSOM*.

Gli ultimi capitoli raccolgono un considerevole numero di prove pratiche e considerazioni mirate a valutare, qualitativamente e quantitativamente, le caratteristiche del modello *StereoSOM* e i miglioramenti introdotti dalle varie estensioni del modello originale. Sono stati effettuati inoltre particolari test di robustezza del-

l'algoritmo in condizioni estreme, con stereocoppie deteriorate artificialmente o catturate con strumenti poco sofisticati.

Per quanto riguarda i test quantitativi, le metodologie impiegate sono quelle maggiormente usate nella letteratura sullo stereo matching e si basano su un noto insieme di dataset forniti dal *Middlebury College* [41], dove è attivo un importante gruppo di ricerca nell'ambito della computer vision, multi-vision e stereo matching.

Marco Vanetti

Indice

Capitolo 1

Introduzione allo stereo matching

Nell'ambito della visione artificiale ha sempre riscosso grande interesse lo studio di sensori e tecniche in grado di acquisire informazioni tridimensionali, specialmente per via delle molteplici applicazioni che dipendono fortemente da questo tipo di dati ambientali (applicazioni di controllo industriale, guida automatica, ricostruzione 3D, topografia satellitare, ecc.)

Le tecniche più comuni in grado di fornire questi dati sono dette *di tipo attivo*, in quanto impiegano particolari dispositivi in grado di generare una interazione fisica con l'ambiente (laser, ultrasuoni, ecc.) acquisendo poi, tramite appositi rilevatori, un segnale utilizzato per calcolare il dato tridimensionale. Le tecniche di tipo attivo vantano quasi sempre grande precisione ed affidabilità ma non rappresentano sempre la scelta più opportuna, infatti i dispositivi impiegati sono spesso costosi e, a causa della interazione diretta con l'ambiente, risultano inevitabilmente invasivi.

Il metodo analizzato in questo elaborato si basa sulla *visione stereoscopica* e fa parte delle tecniche di rilevamento dette *di tipo passivo*, sfrutta infatti come sensori delle tradizionali fotocamere, consentendo costi ridotti e una totale non invasività ambientale.

In letteratura esistono diverse tecniche di visione artificiale mirate alla ricostruzione tridimensionale di una scena osservata da una o più fotocamere, basta citare per esempio *shape from motion*, *shape from shading* o *shape from texture* [1], ma la visione stereoscopica è sicuramente quella che ha riscosso maggiore attenzione poiché non impone alcun vincolo sulle caratteristiche degli oggetti presenti nella scena (e.g. presenza di particolari condizioni di illuminazione o di oggetti in movimento).

Il principio base della visione stereoscopica consiste nella triangolazione geo-

metrica, usata per mettere in relazione la proiezione di un punto della scena sui due[1] o più piani immagine delle fotocamere che compongono il sistema di visione stereoscopico.

L'individuazione dei punti corrispondenti (detti punti omologhi) è un problema largamente affrontato in letteratura col nome di "problema delle corrispondenze" o "stereo matching" e consente di ottenere un importante dato: la disparità. Conoscendo la disparità dei punti che compongono i piani immagine e il valore di alcuni parametri tipici del sistema stereoscopico è possibile calcolare la posizione tridimensionale degli oggetti nella scena.

Il problema delle corrispondenze resta tuttora una sfida aperta, nonostante siano stati proposti da oltre trent'anni molteplici algoritmi.

Si introdurranno ora i concetti fondamentali alla base della geometria di un sistema stereoscopico, passando poi alle fasi operative di acquisizione ed elaborazione delle immagini. Si parlerà poi in generale del problema della ricerca delle corrispondenze, affrontato in dettaglio nel Capitolo 2, e della ricostruzione tridimensionale della scena tramite la triangolazione geometrica. Il capitolo terminerà con una breve discussione sulle molteplici applicazioni dello stereo matching.

Gran parte di questo capitolo introduttivo è stata ispirata da [3], tutorial introduttivo alla visione stereo del dipartimento di elettronica informatica e sistemistica dell'Università di Bologna.

1.1 Sistemi stereoscopici

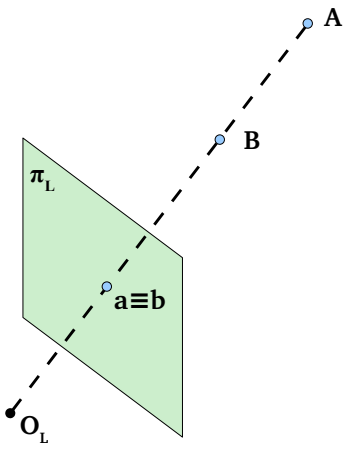
La perdita dell'informazione relativa alla distanza degli oggetti in una scena è dovuta alla trasformazione prospettica che mappa i punti dello spazio nel piano immagine della fotocamera. In Figura 1.1 a fronte è evidente come due distinti punti nello spazio, A e B, intersecati dalla stessa linea che parte dal centro ottico della fotocamera O_L, corrispondano allo stesso punto $a \equiv b$ nel piano immagine.

Un metodo per poter discriminare a quale dei punti nello spazio corrisponda la proiezione di un punto sul piano immagine si basa sull'utilizzo di due o più fotocamere, che ritraggono la stessa scena da due o più punti di vista differenti.

Come mostrato in Figura 1.2 nella pagina 4, che ritrae un sistema stereo com-

[1] Nel caso di due fotocamere e quindi due immagini, si parla di visione binoculare. Un tipico esempio di visione binoculare è il sistema visivo umano.

Figura 1.1: Due punti distinti nello spazio sono proiettati sul piano immagine π_L.



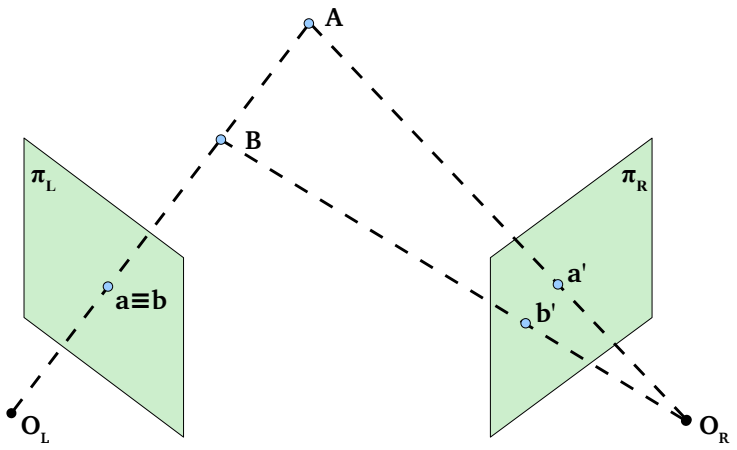
posto da due fotocamere, tra tutti i punti nello spazio che giacciono sulla linea che passa per il centro ottico O_L e il punto a, proiezione di A sul piano immagine π_L, al più un solo punto (detto punto omologo) può trovarsi anche sul piano immagine π_R.

L'individuazione dei punti omologhi permette di mettere in relazione le proiezioni dello stesso punto sui due piani immagine e di risalire, mediante triangolazione geometrica, alle coordinate dei punti nello spazio.

Sia a la proiezione del punto A nell'immagine π_L. Per ottenere le coordinate tridimensionali del punto nello spazio è necessario prima trovare il punto omologo a' nel piano immagine π_R. Tale problema, detto problema delle corrispondenze, è risolvibile mediante una ricerca bidimensionale del punto omologo a' all'interno del piano immagine π_R.

È possibile sfruttare una particolare caratteristica della geometria dei sistemi stereoscopici in modo da ridurre di una dimensione lo spazio di ricerca del punto omologo. Come mostrato in Figura 1.3 nella pagina 6, gli omologhi di tutti i punti dello spazio che risultano potenziali proiezioni dello stesso punto $a \equiv b$ nel piano immagine π_L, si trovano sulla retta ottenuta intersecando π_R con il piano, chiamato *piano epipolare*, che passa per la retta $\overline{O_L AB}$ e i due centri ottici O_L e O_R. Quello appena descritto è a tutti gli effetti un vincolo sulla ricerca delle corrispondenze (chia-

Figura 1.2: Corrispondenze tra due piani immagine π_L e π_R.



mato *vincolo epipolare*) e consente di limitare lo spazio di ricerca dei punti omologhi ad un segmento di retta, rendendo il problema delle corrispondenze sensibilmente più veloce e preciso.

È importante considerare che il problema della ricerca delle corrispondenze non ammette necessariamente una soluzione per ogni punto del piano immagine infatti, a causa della diversa posizione delle fotocamere che compongono il sistema di visione stereoscopico, è possibile che un punto dello spazio non risulti proiettato su tutti i piani immagini. In questo caso il problema delle corrispondenze non è risolvibile e non è quindi possibile determinare in modo preciso la distanza del punto in questione che, date le particolari circostanze, viene detto *punto di occlusione* (o *punto occluso*).

Esiste un ulteriore vincolo che caratterizza i sistemi stereoscopici: l'*intervallo di disparità* (*disparity range*) che corrisponde all'estensione di pixel entro la quale si effettua la ricerca dei punti omologhi. La scelta dell'intervallo di disparità determina la regione dello spazio, all'interno del campo visivo delle fotocamere, entro la quale è possibile calcolare la profondità degli oggetti.

In Figura 1.4 nella pagina 6 è evidenziato il concetto di intervallo di disparità in un sistema stereoscopico, dove d_{min} e d_{max} rappresentano rispettivamente il minimo ed il massimo valore di disparità ammessi. Con le linee tratteggiate in rosso

sono indicate le distanze corrispondenti ai valori di disparità dell'intervallo specificato. Come si vedrà in 1.5, essendo la distanza Z dipendente dal reciproco della disparità, i piani corrispondenti ai vari valori di disparità non sono equidistanti l'uno dall'altro.

La distanza Z_0 dopo la quale inizia il campo visivo del sistema stereo può essere determinata sfruttando la seguente relazione geometrica:

$$\frac{f}{S_x} = \frac{Z_0}{b} \;\Rightarrow\; Z_0 = \frac{bf}{S_x} \tag{1.1}$$

dove S_x è la larghezza del sensore immagine, f è la *lunghezza focale* delle ottiche impiegate e b è il *valore di baseline,* la distanza tra i centri ottici delle due fotocamere.

Sempre attraverso semplici relazioni geometriche è possibile, a patto di conoscere Z, determinare $W(Z)$:

$$\frac{W}{b} = \frac{(Z - Z_0)}{Z_0} \;\Rightarrow\; W(Z) = \frac{b(Z - Z_0)}{Z_0} \tag{1.2}$$

Un sistema di visione stereoscopico è totalmente caratterizzabile mediante particolari parametri detti *intrinseci* ed *estrinseci*.

I parametri intrinseci permettono di definire la trasformazione che mappa un punto dello spazio tridimensionale nelle coordinate del piano immagine di ogni fotocamera e sono le coordinate relative al piano immagine del *principal point*[2], la lunghezza focale e altri parametri che descrivono caratteristiche tipiche dei sensori impiegati, come la distorsione delle lenti, la forma dei pixel, etc.

I parametri estrinseci includono invece le posizioni fisiche delle telecamere rispetto ad un sistema di riferimento noto.

La determinazione dei parametri intrinseci ed estrinseci, ottenuta grazie ad una procedura detta *calibrazione*, consente di descrivere accuratamente il sistema stereoscopico, rendendo possibile l'estrazione di informazioni tridimensionali dalla scena mediante la triangolazione di punti omologhi.

La conoscenza dei parametri intrinseci ed estrinseci permette di applicare una trasformazione di fondamentale importanza nella visione stereo: la *rettificazione* delle stereocoppie. Si pensi ad un sistema virtuale nel quale i piani immagine delle telecamere giacciono sullo stesso piano e nel quale la ricerca dei punti omologhi

[2]Il *principal point* è il punto di intersezione tra il piano immagine e la retta ortogonale al piano immagine stesso, passante per il centro ottico.

Figura 1.3: Vincolo epipolare in un sistema stereoscopico.

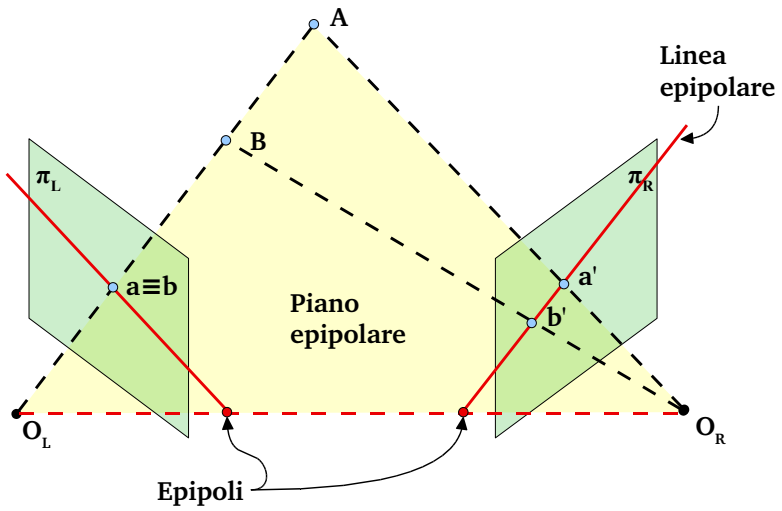


Figura 1.4: Sistema stereoscopico con intervallo di disparità compreso tra d_{min} e d_{max}.

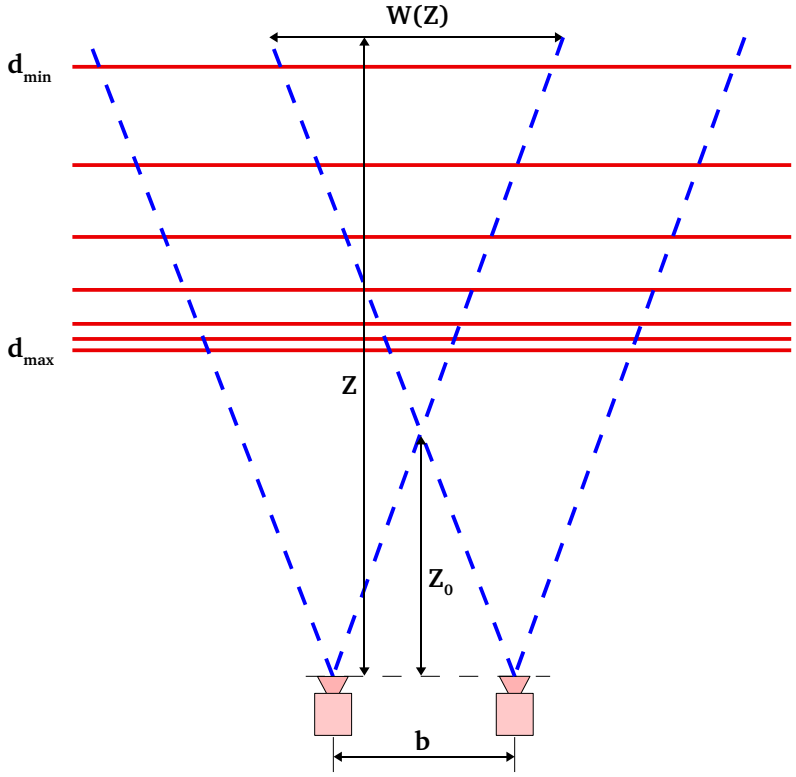
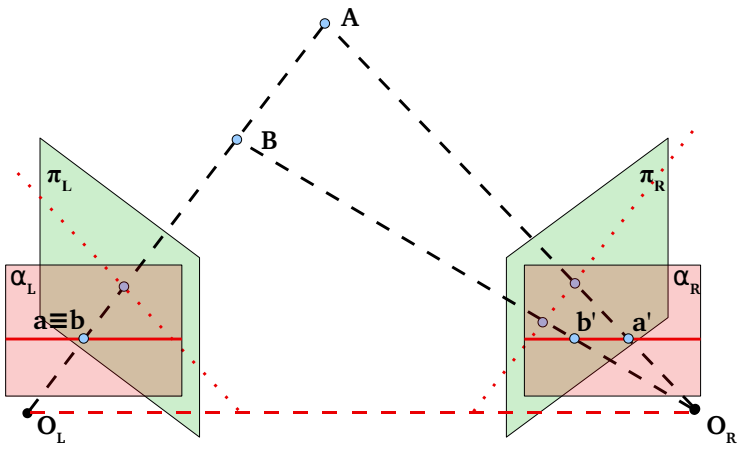


Figura 1.5: Stereocoppie in forma standard, risultato del processo di rettificazione delle immagini.



avviene esaminando le medesime righe nei diversi piani immagine. Questa configurazione del sistema stereoscopico, ottenuta grazie alla rettificazione, è mostrata in Figura 1.5.

A seguito della procedura di rettificazione, il sistema stereo è composto da due piani immagine virtuali complanari: α_L e α_R. Le immagini stereoscopiche ottenute dalla procedura di rettificazione sono dette immagini in *forma standard* e rappresentano l'input per la maggior parte degli algoritmi di corrispondenza stereo.

Fino ad ora sono stati descritti due fondamentali vincoli che caratterizzano la maggior parte dei sistemi stereoscopici, in realtà esistono altri vincoli che possono agevolare gli algoritmi di stereo matching riducendo il numero delle potenziali corrispondenze:

Vincolo di continuità (*smoothness constraint*) che suppone una variazione continua e regolare della disparità su tutte le superfici della scena. Ovviamente questo vincolo non può applicarsi alle aree vicine ai bordi degli oggetti, dette *zone di discontinuità di profondità*, dove la disparità può variare in modo brusco.

Vincolo di ordinamento (*ordering constraint*) secondo il quale dati due punti a e b nell'immagine di sinistra e due punti a' e b' nell'immagine di destra, se

valgono gli omologhi $a \Leftrightarrow a'$ e $b \Leftrightarrow b'$ allora si può dedurre che se a è alla sinistra di b nella prima immagine allora anche a' sarà alla sinistra di b' nella seconda. Il vincolo funziona purché non ci si trovi nelle particolari condizioni in cui uno dei due punti si trova nella cosiddetta *zona proibita* dell'altro punto. Per approfondire le nozioni sul vincolo di ordinamento si consiglia la lettura di [4].

Vincolo di unicità (*uniqueness constraint*) secondo cui ciascun punto in un'immagine può avere al più un omologo nell'altra immagine. Questo non vale sempre, ad esempio per oggetti trasparenti o riflettenti l'unicità del punto omologo non è sempre rispettata.

Gli ultimi due vincoli descritti, pur riuscendo a trovare impiego in molti sistemi di visione stereo, non sono facilmente integrabili in algoritmi di corrispondenza come quello proposto in questo elaborato che, da parte sua, riesce a sfruttare efficacemente il vincolo epipolare, l'intervallo di disparità ed il vincolo di continuità.

1.2 Calibrazione ed acquisizione

Il primo passo per l'acquisizione di stereocoppie è la calibrazione del sistema stereoscopico. La procedura di calibrazione è eseguita *offline* e serve per individuare i parametri intrinseci ed estrinseci del sistema, usati poi dalla procedura di rettificazione (per generare le immagini in forma standard) e nel processo di triangolazione geometrica (per ottenere le coordinate tridimensionali dei punti nella scena).

Esistono in letteratura diverse tecniche per effettuare la calibrazione di un sistema stereoscopico, si veda ad esempio [1] per una trattazione completa. Tipicamente la calibrazione viene eseguita con tecniche basate sull'utilizzo di pattern geometrici standard, dei quali sono note con precisione le caratteristiche. Esempi di pattern geometrici sono delle simil-scacchiere, di cui si conosce con precisione la distanza, la dimensione e l'alternarsi del colore nelle celle.

Mediante l'acquisizione in diverse posizioni di tali pattern e utilizzando procedure standard ampiamente consolidate[3] è possibile calcolare i parametri intrinseci ed estrinseci che caratterizzano il sistema stereoscopico. In Figura 1.6 sono conte-

[3]Si consideri per esempio il *Camera Calibration Toolbox for Matlab*, disponibile gratuitamente sul World Wide Web all'indirizzo [2].

nute alcune immagini che documentano l'operazione di calibrazione di un sistema stereoscopico; sono stati impiegati allo scopo dei pattern planari simili a scacchiere.

La procedura di calibrazione determina per ogni fotocamera la lunghezza focale, le coordinate del *principal point* ed altri parametri, come la deformazione dei pixel rispetto alla forma ideale quadrata (*skew coefficient*) ed i parametri che caratterizzano le deformazioni provocate dalle ottiche (*distortion coefficient*). La calibrazione determina inoltre i parametri estrinseci che, attraverso un vettore di traslazione ed uno di rotazione, descrivono la trasformazione che mappa un piano immagine sull'altro rispetto ad un sistema di riferimento prefissato.

La calibrazione consente inoltre di determinare il valore di baseline, parametro fondamentale per calcolare le profondità mediante la procedura di triangolazione.

Il processo di calibrazione del sistema stereoscopico è direttamente seguito dalla fase di acquisizione delle due (o più) immagini utilizzando le fotocamere. L'acquisizione delle immagini avviene quasi sempre in modo simultaneo [4], facendo uso di tecnologie analogiche o digitali.

In questo elaborato si farà sempre riferimento a *sistemi stereo laterali* in cui gli assi ottici sono paralleli, la lunghezza focale è la stessa per tutte le immagini, i piani immagine risultano coplanari e i sistemi di riferimento (solidali con le due fotocamere) differiscono solo per una traslazione orizzontale (il valore di baseline).

1.3 Rettificazione

La rettificazione è un processo che, basandosi sui parametri intrinseci ed estrinseci ottenuti dalla calibrazione del sistema stereoscopico, trasforma le immagini stereoscopiche provenienti dal dispositivo di acquisizione in modo che soddisfino il vincolo epipolare.

Rettificando un'immagine la si converte in forma standard e, nel caso di sistemi binoculari, questo assicura che dato un punto non occluso dell'immagine il suo omologo possa essere trovato sulla stessa riga dell'altra immagine. Avere immagini in forma standard consente una notevole riduzione dei tempi di calcolo ed una maggiore affidabilità nella soluzione del problema delle corrispondenze.

La rettificazione riduce inoltre alcuni problemi legati alla distorsione provocata dalle ottiche delle fotocamere e permette di ottenere immagini con la stes-

[4] Con l'eccezione di alcuni casi dove la scena è comunque perfettamente statica.

Figura 1.6: Calibrazione di un sistema stereoscopico mediante l'utilizzo di pattern planari. ***(a)***: mediante un dispositivo di puntamento si selezionano alcuni punti di interesse all'interno del pattern acquisito. ***(b)***: pattern correttamente riconosciuto. ***(c)***: l'acquisizione di diversi pattern in varie posizioni della scena permette di calcolare i parametri intrinseci ed estrinseci.

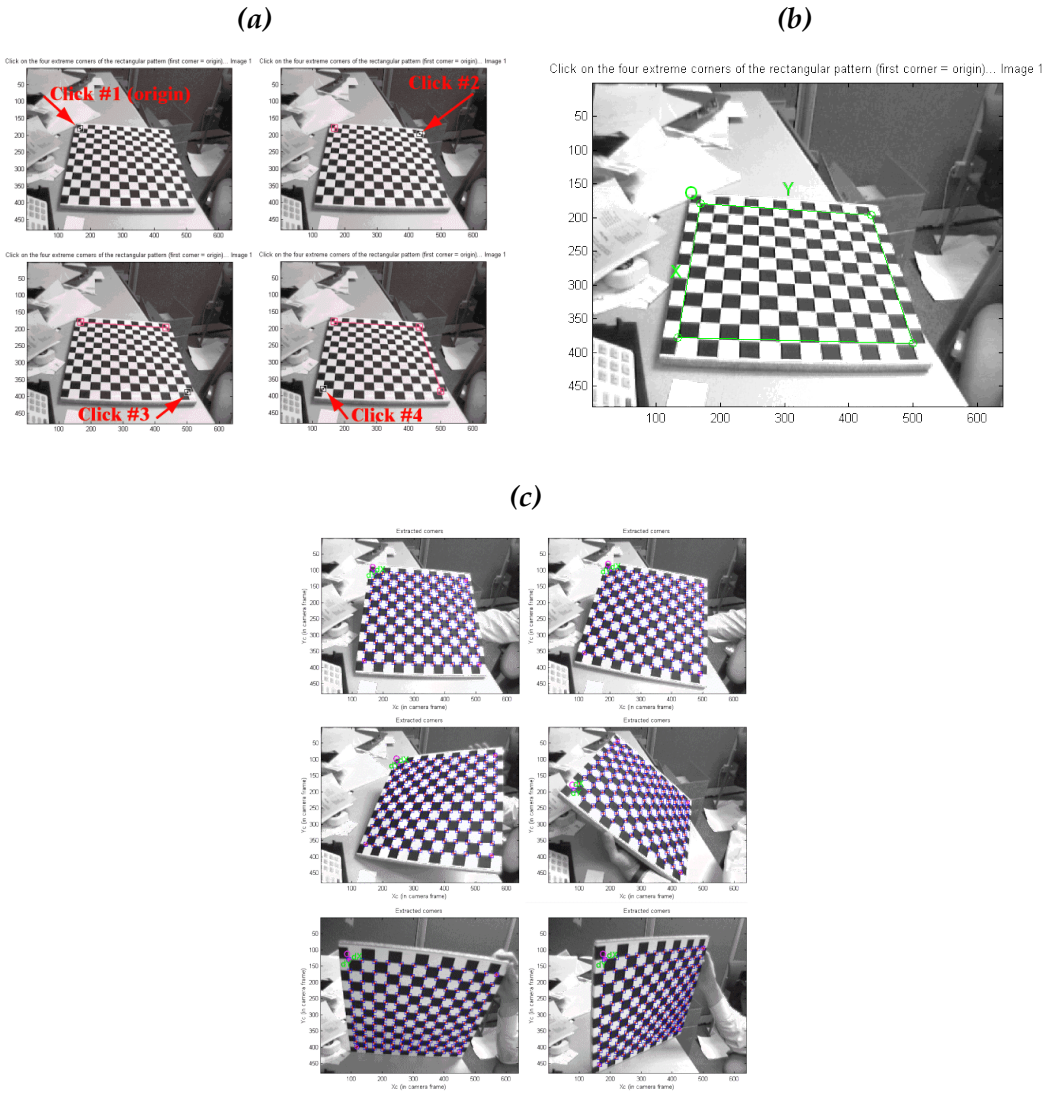


Figura 1.7: Esempio di rettificazione di una coppia stereo laterale acquisita da due fotocamere (sopra). I punti omologhi nella coppia stereo in forma standard (sotto) si trovano tutti lungo linee orizzontali che attraversano entrambe le immagini.

sa distanza focale. Un esempio di rettificazione di una stereocoppia è visibile in Figura 1.7.

1.4 Ricerca delle corrispondenze

Il risultato della ricerca delle corrispondenze tra due (o più) immagini stereo è esprimibile mediante le cosiddette *mappe di disparità*. Una mappa di disparità è sempre accoppiata ad una immagine di riferimento e associa, ad ogni pixel di questa, il suo valore di disparità. Spesso le mappe di disparità sono rappresentate come immagini in scala di grigi dove i valori di grigio, opportunamente normalizzati moltiplicandoli per un fattore costante, esprimono il valore di disparità di ogni pixel dell'immagine di riferimento. Un esempio di stereocoppia con relative mappe di disparità è riportato in Figura 1.8 nella pagina 13.

Come è stato detto in 1.2, in questo elaborato si farà sempre riferimento al sistema stereoscopico laterale. Se si considerano inoltre solo immagini in forma standard è possibile agevolare il problema della ricerca dei punti omologhi, limitandosi ad una ricerca "laterale" lungo una linea orizzontale. Disponendo inoltre dei para-

metri d_{min} e d_{max} (intervallo di disparità) è possibile ridurre ulteriormente il campo di ricerca. Si noti che, il fatto che il sistema stereoscopico sia laterale, verticale o altro, non cambia il concetto alla base della ricerca delle corrispondenze: è possibile adattare gli algoritmi di stereo matching a qualsiasi tipo di allineamento delle fotocamere.

Con l'unica eccezione delle zone occluse, una mappa di disparità dovrebbe stabilire una corrispondenza $1:1$ tra i pixel della prima immagine e quelli della seconda[5].

Sia I una immagine rappresentata come matrice di pixel $(H \times W)$ dove H è il numero di righe della matrice e W è il numero di colonne. Un generico pixel all'interno dell'immagine è rappresentato dalle sue coordinate $I(r, c)$, dove r e c rappresentano rispettivamente il numero di riga e colonna dell'elemento nella matrice. Ogni elemento della matrice conterrà i valori di intensità dei canali associati ad ogni pixel dell'immagine; se si prende come esempio il sistema RGB, vi sono per ogni pixel tre canali (R, G, B) con valori di intensità compresi nell'intervallo $[0, 255]$.

Limitiamoci in questa descrizione al caso di due immagini, sia quindi I_L e I_R la coppia stereo in forma standard presa da un sistema stereo laterale. Per quanto riguarda la ricerca delle corrispondenze, le immagini in forma standard godono della seguente proprietà formale: per ogni corrispondenza $I_L(r_L, c_L) \Leftrightarrow I_R(r_R, c_R)$, tra i pixel delle due immagini si ha che $r_L = r_R$.

La mappa delle disparità associata all'immagine di sinistra d_L può essere rappresentata mediante una funzione $\mathbb{R}^2 \rightarrow \mathbb{R}$ tale che, se il punto $I_L(r, c_L)$ corrisponde al punto $I_R(r, c_R)$, allora:

$$d_L\,(r, c_L) = c_L - c_R \tag{1.3}$$

Allo stesso modo, per l'immagine di destra, se il punto $I_R(r, c_R)$ corrisponde al punto $I_L(r, c_L)$ si ha che:

$$d_R\,(r, c_R) = c_L - c_R \tag{1.4}$$

Se le immagini stereo sono state acquisite correttamente, rispettando le specifiche di posizionamento delle fotocamere, le funzioni d sono sempre positive.

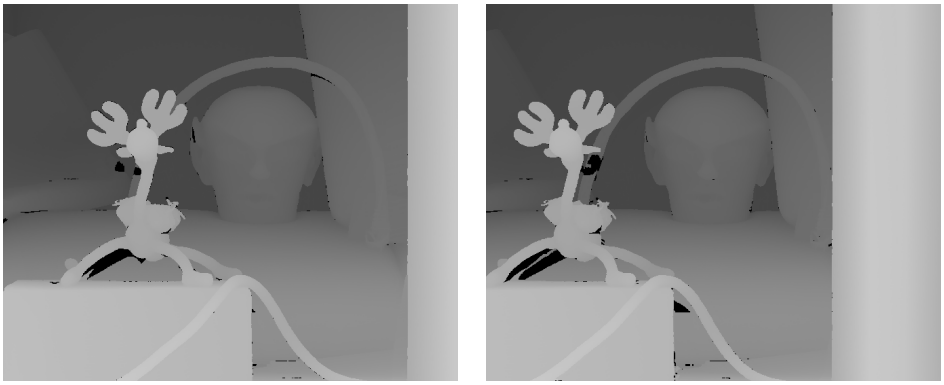
Guardando la mappa delle disparità come una immagine in scala di grigi, gli

[5]Come si vedrà nel Capitolo 2 questo tipo di stereo matching è detto *denso*.

Figura 1.8: Dataset *"Reindeer"*, esempio di stereocoppia in forma standard, completa di mappe di disparità. Per le referenze vedere l'Appendice A. ***(a)***: coppia di immagini stereo in forma standard. ***(b)***: mappe di disparità reale associate alle due immagini.

(a)

(b)



oggetti in essa presenti sono qualitativamente posizionati come nell'immagine di riferimento, ma presentano un'intensità di colore che dipende da quanto questi si "spostano" da I_L a I_R: maggiore sarà questo spostamento, più chiaro sarà il colore (e quindi maggiore sarà il valore di disparità).

Spesso la mappa di disparità è chiamata *mappa di profondità* (*depth map*) poiché, sempre qualitativamente parlando, se ci si posiziona in prossimità degli obbiettivi e si guarda verso la scena ripresa, i punti della mappa che hanno maggiore disparità sono quelli fisicamente più vicini agli obbiettivi delle fotocamere.

Potendo disporre di una funzione d ottimale è possibile stabilire la validità di una corrispondenza tra due potenziali punti omologhi. È facile verificare che le corrispondenze valide sono solo quelle nella forma:

$$I_L(r, c_L) \Leftrightarrow I_R(r, c_L - d_L(r, c_L)) \tag{1.5}$$

o analogamente:

$$I_R(r, c_R) \Leftrightarrow I_L(r, c_R + d_R(r, c_R)) \tag{1.6}$$

Come è mostrato in Figura 1.8 nella pagina precedente la maggior parte dei dataset usati per le analisi quantitative degli algoritmi di corrispondenza stereo dispongono di una *mappa di disparità reale* (*ground truth disparity map*), dato di verità utilizzabile per valutare la qualità delle corrispondenze con il procedimento appena descritto.

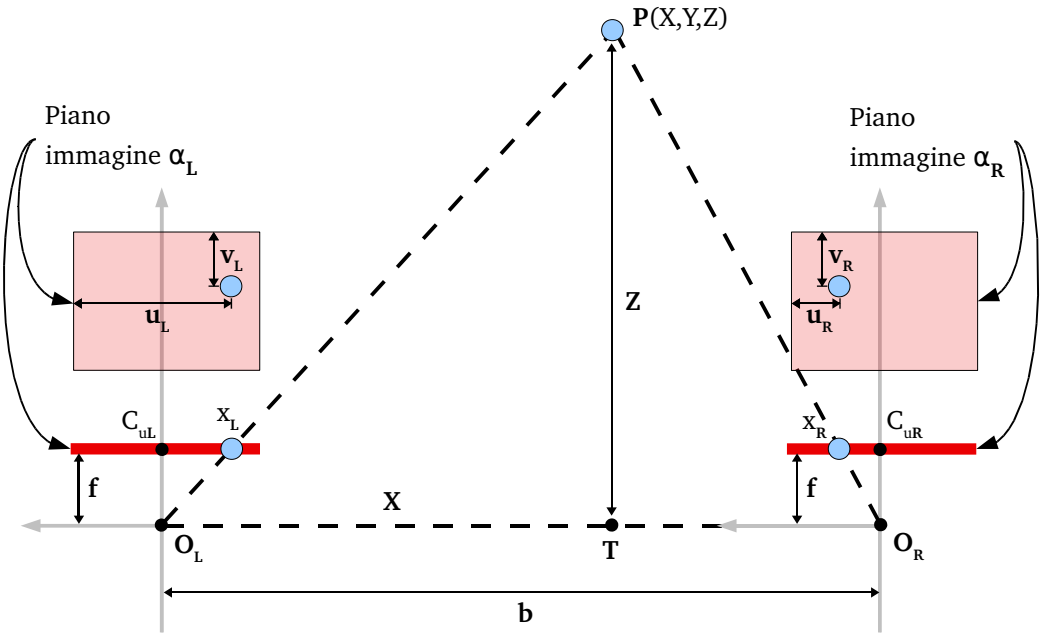
1.5 Ricostruzione 3D

Risolto il problema delle corrispondenze e noti i parametri del sistema stereoscopico si è in grado, per tutti quei i punti per cui è stato possibile determinare l'omologo, di calcolare la distanza dalle fotocamere.

Si consideri la Figura 1.9 a fronte, nella quale è rappresentato un sistema stereoscopico ideale con immagini in forma standard[6]. Siano (x_L, y_L) e (x_R, y_R) le coordinate delle proiezioni del punto P sui due piani immagini α_L e α_R rispetto ai sistemi di riferimento con origine nei centri ottici delle due fotocamere. Questi pun-

[6]Si suppone quindi il sistema come già calibrato e dotato di fotocamere che applicano automaticamente la procedura di rettificazione, generando immagini in forma standard.

Figura 1.9: Triangolazione in un sistema stereoscopico con immagini in forma standard.



ti hanno coordinate (u_L, v_L) e (u_R, v_R), rispetto ai sistemi di riferimento con origine nel punto in alto a sinistra di ogni immagine.

Per ottenere le coordinate (x_L, y_L) e (x_R, y_R) si fa coincidere l'origine delle coordinate con il *principal point* di ogni fotocamera, si moltiplica poi tale valore per la dimensione orizzontale del pixel, nel caso della coordinata x, e per la dimensione verticale del pixel, nel caso della coordinata y.

La forma e le dimensioni del pixel dipendono dal tipo di sensore; nel caso in cui i pixel non siano quadrati, bisogna conoscere le loro dimensioni orizzontali (Dim_x) e verticali (Dim_y).

Siano (C_{u_L}, C_{v_L}) e (C_{u_R}, C_{v_R}) le coordinate dei *principal point* rispetto al sistema di riferimento centrato nel punti in alto a sinistra di ogni immagine. Per i punti sulle immagini α_L e α_R si ha che:

$$\begin{aligned} x_L &= (u_L - C_{u_L}) \cdot Dim_x \quad , \quad x_R = (u_R - C_{u_R}) \cdot Dim_x \\ y_L &= (v_L - C_{v_L}) \cdot Dim_y \quad , \quad y_R = (v_R - C_{v_R}) \cdot Dim_y \end{aligned} \tag{1.7}$$

Assumiamo ora come sistema di riferimento quello con origine nel centro ottico O_L, con assi x e y paralleli ai piani immagine e avente asse z passante per O_L e per il centro dell'immagine. Vediamo quindi come, a partire dalle coordinate immagine x_L e y_L, sia possibile scrivere le relazioni che individuano le coordinate X, Y e Z, posizione del punto P nello spazio.

Indicando con con f la lunghezza focale, con b la distanza di baseline e sfruttando la similitudine tra i triangoli $O_L P O_R$ e $x_L P x_R$ è possibile scrivere la seguente relazione che individua la distanza Z del punto P dalla congiungente i centri ottici O_L e O_R delle due telecamere:

$$\frac{b + x_L - x_R}{Z - f} = \frac{b}{Z} \tag{1.8}$$

e visto che la disparità è ottenuta con $d = \frac{x_L - x_R}{Dim_x}$, risulta che:

$$Z = f \frac{b}{d \cdot Dim_x} \tag{1.9}$$

Si noti come la distanza fisica Z tra l'asse congiungente le fotocamere ed un oggetto nella scena sia proporzionale al reciproco del valore di disparità d; si dimostra quindi formalmente perché oggetti con disparità piccola si trovano lontani dagli obbiettivi, mentre oggetti con grande disparità sono prossimi alle fotocamere.

La coordinata X si ricava dalla similitudine tra i triangoli $O_L P T$ e $O_L C_{u_L} x_L$, si ha quindi che $\frac{Z}{X} = \frac{f}{x_L}$, da cui:

$$X = Z \frac{x_L}{f} \tag{1.10}$$

Analogamente per Y risulta che:

$$Y = Z \frac{y_L}{f} \tag{1.11}$$

Si ottiene quindi la tripla (X, Y, Z), coordinate del punto P rispetto al sistema di riferimento con origine nel centro ottico O_L. Un ragionamento analogo si può fare per ottenere le coordinate rispetto al sistema di riferimento con origine nel centro ottico O_R.

1.6 Applicazioni

L'utilizzo della visione stereo spazia in numerosissimi campi, partendo dal mondo scientifico (e.g. *microscopia ottica*), passando per le creazioni artistiche (e.g. *fotografia*, *autostereogrammi*, *anaglifi*) fino ad arrivare all'intrattenimento (e.g. *cinema 3D*, *videogiochi con visualizzazione stereoscopica*).

Tuttavia gli algoritmi di corrispondenza stereo sono impiegati solo in quei casi dove è necessario ottenere un dato quantitativo sulla profondità dei punti che compongono la scena. Di seguito sono elencate alcune applicazioni dello stereo matching in contesti di diverso genere:

- *Conteggio delle persone* entranti e uscenti da un edificio o da un mezzo pubblico (e.g. in Figura 1.10).

  – Mediante l'analisi dei dati 3D estratti dalla scena inquadrata è possibile contare le persone transitanti attraverso un varco. Questi sistemi hanno particolare interesse in quei contesti dove è importante tenere il conto delle persone all'interno di un ambiente chiuso. Esempi sono mezzi di trasporto (treni, autobus, ecc.), stazioni, edifici con particolari problematiche di sicurezza.

Figura 1.10: Dispositivo *People Sensor CPS-1000* per il conteggio di persone basato sullo stereo matching. È sviluppato dalla Cognex Corporation [7].

- *Riconoscimento biometrico* del volto, basato sulla ricostruzione del profilo facciale (e.g. [8]).

  – Tramite l'analisi dei dati tridimensionali della superficie del volto è possibile realizzare o potenziare un processo di riconoscimento facciale. Que-

sto metodo è spesso utilizzato congiuntamente ad altre tecniche basate su dati 2D.

- *Mappatura delle montagne* basata su immagini satellitari o aeree (e.g. [9]).
  - Tramite l'analisi stereoscopica di immagini stereo prese da satelliti o aerei è possibile determinare con buona precisione la forma del suolo terrestre.
- *People tracking*, controllo spostamenti delle persone nell'ambiente (e.g. [10]).
  - In alcuni ambienti può essere necessario mantenere la posizione delle persone e seguirle negli spostamenti. Per svolgere questo compito i sistemi tradizionali di videosorveglianza richiedono l'intervento umano e non risulta sempre facile seguire più persone contemporaneamente. Sistemi che sfruttano la visione stereoscopica permettono di realizzare, a costi decisamente contenuti, soluzioni di people tracking automatiche e molto robuste.
- *Guida automatica* di veicoli civili o sonde robotizzate (e.g. [5, 6]).
  - Sono attivi numerosi progetti che impiegano i dati provenienti da algoritmi di stereo matching per permettere ad un veicolo di guidare autonomamente. Le informazioni sulla profondità e sulla posizione degli oggetti sono di fondamentale importanza per evitare collisioni durante la marcia.
- *Ricostruzione tridimensionale di superfici microscopiche*, acquisite con il microscopio a scansione elettronica (*SEM*) o altri dispositivi di microscopia ottica (e.g. [11]).
  - Il *Centro di Ricerca in Analisi di Immagini ed Informatica Medica* (*CRAIIM*) dell'*Università degli Studi dell'Insubria* sta sperimentando diversi algoritmi di stereo matching per ricostruire informazioni tridimensionali da superfici acquisite con il microscopio a scansione elettronica (*SEM*). Il microscopio in questione non fornisce dati tridimensionali sul campione, ma immagini 2D della sua superficie. Esiste un altro tipo di microscopio, detto a *forza atomica* (*AFM*), in grado di fornire dati 3D sui campioni

analizzati, tuttavia questo strumento non offre la stessa comodità operativa e lo stesso intervallo di ingrandimento fornito dal *SEM*. Per questo motivo l'analisi stereoscopica delle immagini *SEM* costituisce una applicazione molto interessante degli algoritmi di stereo matching.

- *Interfacce utente* alternative, basate sul riconoscimento dei movimenti di parti del corpo.
- *Robotica industriale*.

È facile rendersi conto che le potenziali applicazioni dello stereo matching sono davvero innumerevoli, tuttavia è importante sottolineare che in molti casi si possono sostituire i metodi basati sulla corrispondenza stereo con sensori tridimensionali di tipo attivo. Lo stereo matching, in quanto tecnica passiva, è quasi sempre però una soluzione più economica, meno invasiva e, in alcuni casi, necessaria: basti pensare alla ricostruzione 3D di superfici dal microscopio *SEM* o alla mappatura delle montagne da satellite, casi in cui l'applicazione di metodi attivi risulterebbe tecnicamente impraticabile.

Capitolo 2

Problema della ricerca delle corrispondenze

La bibliografia relativa agli algoritmi di corrispondenza stereo è estremamente vasta. È possibile tuttavia raggruppare gli algoritmi di stereo matching in due fondamentali insiemi: il primo racchiude i metodi *basati sulle feature*, mentre il secondo include i metodi *basati sull'area*.

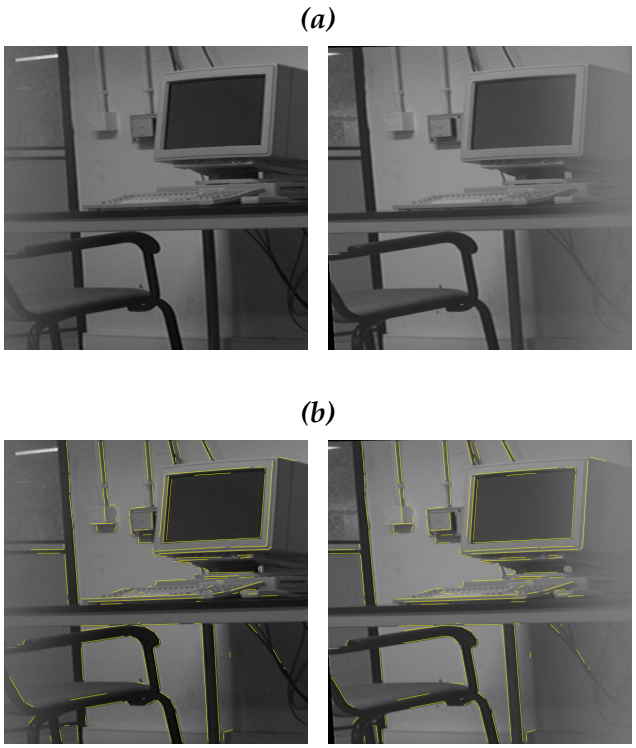
Come si vedrà, la principale differenza tra le due categorie di algoritmi sta nella diversa natura dei risultati: i metodi basati sulle feature non producono infatti mappe di disparità dense, ma possono fornire ottimi risultati in termini di velocità di elaborazione e affidabilità.

2.1 Stereo matching basato sulle feature

Gli algoritmi di corrispondenza stereo basati sulle feature (*feature based*) consentono di ottenere informazioni di disparità relative ad un ristretto sottoinsieme di punti, corrispondenti solitamente a particolari feature come linee, segmenti e angoli. Questi algoritmi sono molto efficienti dal punto di vista computazionale, si pensi al ridotto numero di feature identificabili nelle immagini rispetto al numero totale di punti. Un ulteriore vantaggio di questi metodi è l'elevata affidabilità, visto che le feature estratte dalle immagini stereo sono fortemente distintive e non pongono quasi mai problemi di ambiguità nella soluzione del problema delle corrispondenze.

Il principale motivo per cui gli algoritmi feature based risultano tuttavia poco

Figura 2.1: Applicazione di un algoritmo per la ricerca delle corrispondenze basato sulle feature (in particolare sui segmenti). ***(a)***: coppia stereo di esempio. ***(b)***: Corrispondenze trovate tra i segmenti estratti dalle due immagini.



utilizzati è da ricondursi alle mappe di disparità sparse, limitate cioè ai soli punti che presentano feature distintive.

In un'ottica di schematizzazione, gli algoritmi feature based operano in due principali fasi: il *pre-processing* e il *matching*.

Nella prima fase vengono estratte dalle immagini le feature, rappresentazioni numeriche di alcune caratteristiche particolari dell'immagine. Le feature più utilizzate sono quelle relative a bordi (*edge elements*), angoli (*corners*), segmenti (*line segments*), segmenti curvi (*curve segments*), cerchi (*circles*), ellissi (*ellipses*), descrittori di regioni (*regions*), ecc.

Nella seconda fase si trovano le corrispondenze tra le feature appartenenti alle due (o più) viste della scena.

Un esempio di applicazione dello stereo matching feature based si trova in Figura 2.1, dove sono state impiegate feature relative ai segmenti.

Il fatto di utilizzare poche informazioni di tipo complesso semplifica notevol-

mente il compito della ricerca delle corrispondenze, fornendo ottimi tempi di esecuzione ed eccellenti risultati in termini di "matching esatti".

Come è stato anticipato all'inizio del paragrafo, gli algoritmi feature-based soffrono di alcune limitazioni intrinseche: non forniscono infatti mappe di disparità dense ma si limitano a misurare la disparità dei punti associati alle feature estratte. Si ricorda che, per ottenere una mappa di disparità densa con metodi basati sulle feature, è necessario ricorrere a metodi approssimati basati sull'interpolazione [53].

È molto importante scegliere adeguatamente le feature da impegnare nella ricerca delle corrispondenze, anche se una scelta mirata è possibile solo dopo una analisi qualitativa delle immagini da elaborare. Nonostante la ricerca scientifica abbia compiuto notevoli sforzi a riguardo, non si è ancora giunti alla definizione di feature sufficientemente generali da poter trattare ogni situazione e contesto applicativo.

Approcci basati sulle feature trovano un discreto successo in applicazioni dove le immagini stereo contengono una grande quantità di superfici rettilinee, facilmente descrivibili mediante feature basate sui segmenti; tuttavia questi impieghi sono davvero remoti.

È utile ricordare che sono stati proposti algoritmi feature based che permettono di sfruttare più feature allo stesso tempo, ad esempio in [12] è stata utilizzata una gerarchia di feature che spaziava da *edges* e *corners* a superfici e corpi.

2.2 Stereo matching basato sull'area

Contrariamente agli algoritmi feature based, i metodi di stereo matching basati sull'area possono offrire risultati decisamente più completi. Si generano infatti mappe di disparità dense (e.g. in Figura 1.8 nella pagina 13), che forniscono valori di disparità per ogni punto dell'immagine.

Dal punto di vista operativo, gli algoritmi che seguono un approccio basato sull'area non prevedono una fase di estrazione delle feature, ma trattano direttamente i valori di intensità dei pixel che compongono la coppia stereo. In molti algoritmi area based, le corrispondenze vengono cercate confrontando finestre di pixel prese dalle due immagini, impiegando metriche basate sulla correlazione o su altre misure. È proprio l'utilizzo di finestre di pixel (e quindi di aree circoscritte) il principale motivo del nome *area based*.

In accordo con una recente tassonomia [37] è possibile classificare tali algoritmi

in due principali categorie: *algoritmi di tipo locale* (*local stereo matching*) e *algoritmi di tipo globale* (*global stereo matching*).

2.2.1 Ricerca locale delle corrispondenze

Gli algoritmi di tipo locale cercano le corrispondenze tra i punti omologhi singolarmente ed indipendentemente gli uni dagli altri. Viene sfruttato generalmente un supporto locale (*local support*) nell'intorno di ogni punto esaminato[1], allo scopo di incrementare il *rapporto segnale/rumore* e migliorare l'affidabilità della corrispondenza.

Utilizzando un supporto locale [2], spesso coincidente con una finestra quadrata di pixel, gli algoritmi assumono implicitamente che all'interno del supporto la disparità resti costante. Purtroppo questa ipotesi piuttosto forte non è sempre verificata e conduce spesso ad errori in prossimità delle zone di discontinuità di profondità (zone di confine tra gli oggetti che compongono la scena).

Esistono algoritmi locali (tra cui [39] e quello proposto in questo elaborato), che adattano il proprio supporto in base alle caratteristiche locali delle immagini. Con questi metodi, detti *a supporto adattivo* o *variabile*, è possibile ottenere ottimi risultati, ad un costo computazionale tipicamente più alto rispetto agli algoritmi locali semplici.

Nella maggior parte dei casi gli algoritmi di tipo locale vantano implementazioni con una struttura molto regolare e possono essere realizzati con architetture *hardware* dedicate o utilizzando istruzioni vettoriali di tipo *SIMD* (*Single Instruction Multiple Data*), disponibili su quasi tutte le architetture impiegate nei comuni personal computer. Non è nuovo in questi anni realizzare ottimizzazioni di algoritmi basate su acceleratori grafici di ultima generazione, il cui utilizzo è stato fortemente incentivato dal mercato videoludico. Questi motivi suggeriscono come gli algoritmi di tipo locale ben si prestino ad essere utilizzati in applicazioni di tipo real-time, dove i tempi di risposta e la velocità complessiva di esecuzione sono di fondamentale importanza.

L'assunzione sulla costanza della disparità all'interno della finestra di correla-

[1] L'area nell'intorno di ogni punto è denominata spesso *finestra di correlazione* (*correlation window*). Questo non vuol dire tuttavia che la misura di similarità impiegata nella ricerca delle corrispondenze sia basata sulla correlazione.

[2] Un semplice e veloce algoritmo basato su supporto locale è descritto in [38].

zione non è comunque l'unico problema che affligge gli algoritmi di tipo locale: in molti casi questi metodi non sono in grado di produrre mappe di disparità completamente dense, principalmente a causa della presenza di regioni uniformi prive di texture (dove la corrispondenza tra punti omologhi è spesso ambigua) e di fenomeni di occlusione.

2.2.2 Ricerca globale delle corrispondenze

Diversamente dagli algoritmi basati sulla ricerca locale delle corrispondenze, gli algoritmi di tipo globale non sfruttano direttamente alcun supporto locale. Per arrivare al dato di disparità questi algoritmi ipotizzano particolari proprietà sulla natura degli oggetti che compongono la scena, (e.g. il vincolo di continuità) e ricercano un assegnamento di disparità tale che venga minimizzata una funzione costo globale, estesa a tutti i pixel delle immagini (*global cost function*).

All'interno della classe degli algoritmi globali, le principali differenze consistono spesso nel metodo con cui viene minimizzata la funzione costo; si usano quasi sempre metodi di inferenza approssimata basati su:

- *Belief propagation* [13]
- *Graph-cuts* [14, 15]
- *Programmazione dinamica (Dynamic programming)* [16]
- *Simulated annealing* [17, 18]
- *Maximum flow* [19]
- *Non-linear diffusion* [20]
- *Junction Tree*, *Markov Chain Monte Carlo* (*MCMC*), etc.

Tali algoritmi consentono di ottenere mappe di disparità decisamente migliori rispetto alla media degli algoritmi di tipo locale, ma ad un costo computazionale sostanzialmente più elevato.

Si vedano in Figura 2.2 nella pagina seguente i risultati di due algoritmi proposti a titolo di esempio in [37]: il primo è basato su un approccio locale a finestra di correlazione mentre il secondo sfrutta la minimizzazione di una funziona globale mediante il metodo *Graph-cuts.* È a prima vista evidente come la qualità complessiva dell'algoritmo globale risulti superiore, in particolare si noti come l'algoritmo

Figura 2.2: Immagini riferite al dataset "*Tsukuba*", per le referenze vedere l'Appendice A. ***(a)***: risultato di un algoritmo di stereo matching locale basato su supporto fisso (*SSD*). ***(b)***: risultato di un algoritmo di corrispondenza globale, basato su Graph-cuts. ***(c)***: mappa di disparità reale.

(a) ***(b)*** ***(c)***

locale abbia risultati deludenti sui bordi degli oggetti, in corrispondenza delle discontinuità di profondità; questo è dovuto al supporto fisso, che suppone l'intera area all'interno della finestra come zona a disparità costante.

2.2.3 Principali problematiche

Se è possibile porre un rimedio ad alcune problematiche di natura geometrica (descritte nel Capitolo 1 nella pagina 1) generando immagini in forma standard, non è altrettanto semplice gestire altri tipi di non idealità, legati a fattori di diversa natura. Come sarà ormai chiaro, il principale problema che affligge gli algoritmi di corrispondenza stereo è la gestione delle occlusioni: punti che appaiono in una sola delle immagini stereo e che non ammettono quindi omologhi, rendendo impossibile la definizione di un valore di disparità.

Una possibile soluzione al problema delle occlusioni sarà trattata tra poco nella descrizione della tassonomia degli algoritmi di stereo matching (*disparity refinement*), ma le occlusioni non sono l'unica problematica; si elencano ora alcune delle principali cause di errori di corrispondenza:

Distorsioni fotometriche le superfici reali non sono perfettamente *lambertiane*[3]: per due punti corrispondenti si avranno molto spesso due luminosità differenti a causa del diverso punto di vista. È interessante sapere che

[3]Una superficie si dice *lambertiana* se riflette l'energia incidente in modo uguale in tutte le direzioni. Qualitativamente si ha che variando il punto di vista di una superficie lambertiana la sua apparenza non cambia. Il nome "lambertiana" è legato alla *legge di Lambert sulla riflessione diffusa*.

Figura 2.3: In questa stereocoppia sono state cerchiate con colori diversi le zone più problematiche. *(rosso):* regione occlusa. *(verde):* superficie non lambertiana. *(azzurro):* superficie non fronto-parallela ai piani immagine, affetta da una evidente distorsione prospettica.

la consistenza fotometrica fra due viste è tanto maggiore quanto più è piccolo il valore di baseline rispetto alla distanza dagli oggetti.

Distorsioni prospettiche a causa dei diversi punti di vista, le proiezioni prospettiche degli oggetti sui piani immagini sono in generale diverse. Regioni corrispondenti (specialmente le zone non fronto-parallele ai piani immagine) possono apparire di dimensioni differenti nelle due immagini.

Differenze nei parametri di risposta dei sensori i sensori utilizzati nelle fotocamere e più in generale l'intero processo di acquisizione delle due immagini portano a distorsioni lineari di intensità del tipo $I' = I \cdot g + o$, dove il coefficiente g è detto *gain* e o *offset*.

Rumore come tutti i dati provenienti da sensori reali, anche le immagini nelle stereocoppie sono affette da rumore.

In Figura 2.3 sono evidenziate su una stereocoppia di esempio le principali problematiche appena descritte.

Altri problemi sono legati a regioni con poca texture (*low-texture*), prive di texture (*textureless*) o con pattern periodici (*periodic patterns*); in queste zone è facile commettere errori di disparità per via della elevata ambiguità che contraddistingue la fase di ricerca delle corrispondenze.

2.2.4 Tassonomia di Scharstein e Szeliski

Uno dei passi fondamentali nella recente letteratura sullo stereo matching[4] [37] è stato pubblicato da *Daniel Scharstein*, del *dipartimento di matematica e informatica* del *Middlebury College*, e *Richard Szeliski*, del Microsoft Research. L'importanza di questo articolo è duplice: il primo merito è stato quello di creare una tassonomia in grado di schematizzare a livello operativo le varie componenti degli algoritmi di stereo matching, in secondo luogo è stato creato un potente framework di valutazione (completo di dataset) in grado di avviare una vera e propria competizione tra gli algoritmi di corrispondenza stereo.

Come è ben noto nei gruppi di ricerca in pattern recognition, l'esistenza di dataset comuni e sistemi di valutazione standard è di fondamentale importanza al fine di valutare quantitativamente un algoritmo e confrontarlo seriamente con altri metodi simili.

Per quanto riguarda la tassonomia di Scharstein e Szelizky, gli autori hanno cercato di riassumere le componenti fondamentali di un algoritmo di stereo matching nel modo seguente:

1. *Matching cost computation*

2. *Cost (support) aggregation*

3. *Disparity computation / optimization*

4. *Disparity refinement*

La complessità delle varie fasi dipende molto dal tipo specifico dell'algoritmo di corrispondenza; per gli algoritmi di corrispondenza locali, le prime due fasi ricoprono un ruolo fondamentale mentre la terza è piuttosto banale. Per gli algoritmi di tipo globale la seconda fase è quasi sempre assente mentre la terza è fondamentale e rappresenta il cuore dell'algoritmo.

[4] La tassonomia di Scharstein e Szeliski si limita agli algoritmi di corrispondenza *area based* che forniscono mappe di disparità dense. È sottinteso che questo sottoinsieme di metodi rappresenta tuttora la classe di maggiore interesse scientifico.

Matching cost computation

La fase di *matching cost computation* consiste nel calcolare i costi, intesi come differenza di intensità luminosa, tra i punti delle immagini nella stereocoppia. A tale scopo esistono differenti metodi, tra i più diffusi troviamo:

- Squared intensity Differences (*SD*) [21]
- Absolute intensity Differences (*AD*) [22]
- Cross-Correlazione Normalizzata (*NCC*) [21]
- Binary Matching Costs [23]
- Sign of the Laplacian [24]

Cost (support) aggregation

La fase di aggregation ha come obbiettivo quello di accorpare, sulla base di un supporto (come ad esempio una finestra quadrata), i costi calcolati durante la fase di *matching cost computation*.

La maggior parte dei metodi locali, basati su finestre di pixel, aggregano i costi mediante una somma o una media estesa ai punti appartenenti ad un determinato supporto. Il supporto può essere di tipo bidimensionale o tridimensionale, il primo viene tipicamente implementato con finestre quadrate (*squared windows*), convoluzioni Gaussiane (*Gaussian convolution*) o finestre adattive (*adaptive windows*) [25], mentre i supporti tridimensionali (più complessi) si basano su metodi come *limited disparity difference* [26], *limited disparity gradient* [27] o *Prazdny's coherence principle* [28].

Il metodo descritto in questo elaborato, basandosi su un modello neurale poco convenzionale nella letteratura sullo stereo matching, non è collocabile facilmente all'interno della tassonomia di Scharstein e Szeliski; tuttavia si può trovare un principio di matching cost computation basato su una misura di distanza euclidea tra le intensità dei pixel, mentre per quanto riguarda la cost aggregation viene utilizzato un supporto adattivo a finestra quadrata basato sulla similarità di colore rispetto al punto centrale.

Disparity computation / optimization

Per la fase di disparity computation è necessario effettuare una distinzione tra gli approcci locali e quelli globali.

Per gli algoritmi locali il calcolo della disparità è banalmente computabile e si riduce quasi sempre alla selezione, per ogni punto, del costo con il minimo valore. Il metodo per eccellenza è quindi il *Winner Take All* (*WTA*).

Per gli approcci globali la fase di aggregation è spesso assente, in favore di una fase di disparity computation molto più complessa. Si può dire che per gli algoritmi globali questa fase rappresenti la parte principale della computazione, nonché la responsabile dei tempi di esecuzione spesso proibitivi in contesti *real-time*. L'obiettivo è quello di trovare una mappa di disparità che minimizzi una funzione di energia globale del tipo:

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d) \tag{2.1}$$

dove il termine E_{data} cresce quanto più la funzione di disparità d è in disaccordo con la stereocoppia in input. Il termine E_{smooth} si basa invece sul vincolo di continuità e cresce quanto più la funzione d presenta cambiamenti bruschi nei valori di disparità.

Una volta definita la funzione di energia globale, una grande varietà di algoritmi possono essere usati per minimizzarla rispetto a d. Come indicato precedentemente in 2.2.2 possono essere impiegati metodi di inferenza approssimata, in quanto una ricerca del minimo assoluto è spesso computazionalmente impraticabile.

Disparity refinement

L'ultima fase consiste nell'effettuare dei raffinamenti sulle mappe di disparità calcolate nei passi precedenti. Esistono diverse strategie per migliorare la qualità dei risultati: una delle principali consiste nel limitare la quantizzazione dei valori di disparità con approcci basati sulla valutazione sub-pixel [29].

Un'altra strategia consiste nel controllo di consistenza stereo [30,31], attuato tra l'altro dal metodo proposto in questo elaborato[5]. Disponendo di due mappe di disparità d_L e d_R, relative alle due immagini della stereocoppia I_L e I_R, è possibile

[5] Vedere la sezione 4.3.2 nella pagina 63.

verificare che per ogni corrispondenza:

$$I_L(r,c) \Leftrightarrow I_R(r, c - d_L(r,c)) \tag{2.2}$$

ne esisti una:

$$I_R(r, c - d_L(r,c)) \Leftrightarrow [I_L(r, c - d_L(r,c) + d_R(r, c - d_L(r,c))) = I_L(r,c)] \tag{2.3}$$

e quindi che valga $\forall r, c$ la relazione:

$$d_L(r,c) = d_R(r, c - d_L(r,c)) \tag{2.4}$$

In caso contrario, con alta probabilità, si è in presenza di un punto di occlusione o più in generale di un errore di corrispondenza e bisognerebbe agire di conseguenza utilizzando metodi di interpolazione o etichettando il punto come occluso.

2.3 Metodo di valutazione quantitativa

Per poter valutare un algoritmo di corrispondenza stereo è necessario disporre di una tecnica per stimare quantitativamente la qualità dei risultati ottenuti. Il metodo di valutazione dei risultati impiegato è chiamato *Percentage of Bad pixels in Disparity* (*PBD*), lo stesso utilizzato in [37], e si riferisce alla percentuale di "pixel errati" nella mappa di disparità. Per pixel errati si intendono quei punti a cui è stata assegnata dall'algoritmo una disparità che differisce da quella reale superando un valore di soglia costante.

Esistono molti altri metodi di valutazione, come il noto *Root Mean Squared error* (*RMS*), ma il parere comune è che per ora il *PBD* riesca meglio di tutti a catturare le vere potenzialità di un algoritmo di corrispondenza stereo.

Formalmente, si definisce la funzione:

$$PBD = \frac{1}{N} \sum_{(x,y)} (|d_C(x,y) - d_T(x,y)| > \delta_d) \tag{2.5}$$

dove N è il numero di pixel per cui si dispone del dato di verità, d_C è la mappa di disparità computata, d_T è la mappa di disparità reale e δ_d è la soglia di errore.

L'utilizzo del solo *PBD* non basta tuttavia a catturare tutte le qualità di un algoritmo di corrispondenza; per questo, insieme ai dataset di valutazione, sono incluse

spesso delle mappe binarie che selezionano alcune zone particolarmente sensibili delle mappe di disparità. Tuttora il sistema di valutazione Web del Middlebury College [41] prevede tre zone di valutazione:

Non occluded regions (PBD_{nocc}) si applica l'errore *PBD* solo alle zone non occluse della mappa di disparità.

All regions (PBD_{all}) si applica il *PBD* a tutta la mappa di disparità.

Depth discontinuity regions (PBD_{ddr}) l'errore è valutato solo nelle zone in prossimità dei bordi degli oggetti, dove i valori di disparità variano in modo discontinuo. Non sono incluse in questa valutazione le regioni occluse.

I dataset impiegati prevedono che in alcune aree non vengano valutati gli errori *PBD*. Queste zone sono segnalate nelle mappe di disparità reali con il colore nero (valore di disparità nullo) e corrispondono ad aree di bordo o a piccole zone in cui non è stato possibile ottenere, con metodi attivi, un dato di verità sufficientemente accurato.

È doveroso ricordare che, a partire da queste tre zone, è comunque possibile ottenere mediante operazioni logiche tra mappe binarie la percentuale di errore nelle aree occluse o nelle zone non discontinue.

In Figura 2.4 a fronte è riportato un esempio di dataset completo di mappe binarie corrispondenti alle varie zone di valutazione, mentre in Figura 2.5 nella pagina 34 è stata catturata una schermata dal sistema di valutazione Web del Middlebury College.

Figura 2.4: Immagini riferite al dataset "*Tsukuba*", per le referenze vedere l'Appendice A. ***(a)***: immagine di riferimento. ***(b)***: mappa di disparità reale. ***(c)***: zone non occluse (errore valutato solo nelle regioni bianche). ***(d)***: zone di discontinuità di profondità (errore valutato solo nelle regioni bianche).

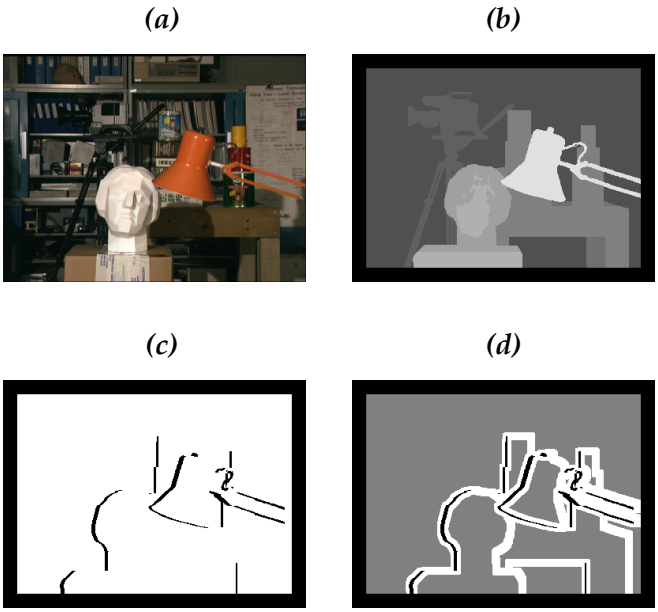


Figura 2.5: Sistema di valutazione Web del Middlebury College. Nel database online sono tuttora inseriti e confrontati più di 70 algoritmi di stereo matching. È possibile impostare diverse chiavi di ordinamento, basate sulla qualità dei risultati in specifici dataset, specifiche zone o per diversi valori di soglia δ_d. L'algoritmo proposto in questo elaborato è visibile nella 21^a posizione, con la sigla *StereoSONN* (Self Organizing Neural Network).

Error Threshold = 0.5 — Error Threshold...

Sort by nonocc ▼ | Sort by all ▼ | Sort by disc ▼

Algorithm	Avg. Rank	Tsukuba ground truth nonocc	Tsukuba all	Tsukuba disc	Venus ground truth nonocc	Venus all	Venus disc	Teddy ground truth nonocc	Teddy all	Teddy disc	Cones ground truth nonocc	Cones all	Cones disc	Average percent of bad pixels (explanation)
GC+SegmBorder [57]	5.5	6.87 6	7.30 4	15.3 8	**0.20** 1	**0.31** 1	**2.44** 1	**7.59** 1	**9.14** 1	**17.5** 1	10.5 29	11.2 3	14.4 10	8.56
SubPixDoubleBP [30]	8.8	8.78 13	9.45 11	14.9 6	0.72 3	1.12 3	5.24 3	10.1 4	16.4 4	21.3 2	8.49 21	14.7 21	16.5 15	10.7
BP+DirectedDiff [61]	9.7	6.21 4	7.96 7	24.5 56	1.59 6	2.34 7	6.94 5	7.78 2	17.3 6	22.5 3	**4.73** 1	14.6 18	**10.7** 1	10.6
Undr+OvrSeg [48]	10.6	**4.84** 1	**5.22** 1	9.39 2	2.00 7	2.15 6	5.44 4	12.9 15	17.9 9	27.6 18	9.16 23	14.6 17	17.6 24	10.7
C-SemiGlob [19]	11.4	13.9 26	14.7 25	18.9 25	3.30 11	3.82 10	10.9 12	9.82 3	17.4 7	22.8 5	5.37 5	11.7 4	12.8 4	12.1
VarMSOH [54]	12.0	7.18 8	8.56 10	20.1 33	1.46 5	2.12 5	7.87 7	12.9 16	19.4 15	27.5 16	6.22 8	12.6 8	15.8 13	11.8
CoopRegion [41]	12.1	18.9 34	19.5 34	21.2 38	1.38 4	1.77 4	9.39 8	10.6 6	15.3 3	22.6 4	5.36 4	**10.6** 1	13.0 5	12.5
ImproveSubPix [25]	12.8	8.96 14	9.66 12	16.2 13	4.62 21	5.41 21	16.9 42	11.0 9	17.8 8	24.1 7	4.90 2	10.8 2	12.2 2	11.9
AdaptOvrSegBP [33]	13.4	5.98 2	6.56 2	**9.09** 1	3.66 13	3.96 11	13.2 25	13.0 17	18.9 12	26.4 12	9.48 25	14.9 22	17.2 19	11.9
OverSegmBP [26]	14.2	7.75 11	8.17 8	13.8 4	4.33 17	4.73 15	16.8 40	13.2 18	19.3 14	27.5 15	6.53 11	12.6 9	14.0 8	12.4
AdaptingBP [17]	16.3	19.1 36	19.3 33	17.4 16	4.84 22	5.08 19	7.84 6	12.8 14	16.7 5	26.3 11	7.02 14	13.2 13	14.0 7	13.6
SemiGlob [6]	16.8	13.4 24	14.3 23	20.3 35	4.55 19	5.38 20	15.7 37	11.0 8	18.5 10	26.1 10	4.93 3	12.5 7	13.5 6	13.3
GradAdaptWgt [60]	17.2	7.67 9	8.25 9	15.0 7	7.51 35	8.05 35	12.3 18	13.5 19	19.6 16	28.5 21	7.34 16	13.0 10	14.8 11	13.0
2OP+occ [37]	19.3	7.10 7	7.70 6	11.9 3	0.56 2	0.83 2	4.21 2	17.5 40	22.7 35	31.5 36	11.6 34	17.1 30	20.4 35	12.8
CostAggr+occ [39]	19.8	18.7 33	19.2 32	20.1 32	4.17 16	4.93 17	12.6 19	13.6 23	19.8 18	29.1 23	6.23 9	12.3 6	14.2 9	14.6
RTCensus [50]	20.3	12.9 21	14.1 22	28.1 63	3.67 14	4.63 14	17.8 45	11.4 10	18.6 11	27.7 19	5.54 6	11.8 5	15.9 14	14.4
PUTv3 [63]	21.1	16.4 28	18.4 29	20.4 36	6.31 29	7.01 28	12.7 22	12.2 11	20.5 26	27.3 14	7.01 13	14.0 14	12.7 3	14.6
Segm+visib [4]	21.4	12.7 19	12.9 15	15.8 11	10.4 50	11.0 51	19.5 47	11.0 7	13.2 2	23.7 6	8.12 17	13.1 12	17.3 20	14.1
ConvexTV [46]	24.1	11.1 15	13.3 16	27.2 62	5.99 26	7.40 31	22.3 55	10.5 5	19.9 19	25.8 9	5.99 7	16.5 28	16.7 16	15.2
InteriorPtLP [34]	24.8	19.2 38	19.6 36	18.1 20	3.00 10	3.71 9	18.8 46	13.5 20	19.2 13	29.7 26	9.58 26	15.9 26	18.9 28	15.8
StereoSONN [71]	25.5	16.7 29	17.5 28	32.5 67	2.78 8	3.10 8	12.9 23	13.6 22	19.9 21	30.0 28	8.28 19	14.7 20	20.3 33	16.0
SAD-IGMCT [52]	25.9	12.1 18	13.4 18	28.2 64	4.06 15	4.86 16	25.9 60	12.3 13	19.7 17	31.3 32	6.91 12	14.0 15	19.2 31	16.0
DoubleBP [35]	26.2	18.7 32	19.1 31	15.8 10	7.82 38	8.22 36	11.3 14	14.4 25	19.9 20	24.4 8	11.8 35	17.6 33	19.7 32	15.7
GeoSup [64]	26.8	22.9 50	23.1 49	20.4 37	6.81 31	7.11 29	11.5 15	13.5 21	20.4 24	26.8 13	8.17 18	15.0 23	15.5 12	15.9

...

ReliabilityDP [15]	54.6	19.0 35	20.7 41	17.5 19	12.7 59	14.0 60	26.1 61	26.3 63	32.5 63	36.8 59	23.7 67	29.9 67	31.5 61	24.2
RealTimeGPU [14]	56.1	24.2 54	26.0 63	24.9 59	10.9 54	12.1 55	27.6 62	19.6 51	27.0 54	33.0 47	16.5 57	23.7 57	29.5 60	22.9
PhaseBased [31]	57.6	11.2 16	13.4 17	23.9 53	15.0 63	16.5 64	35.3 68	33.4 69	40.3 69	44.3 65	27.6 70	35.7 70	38.6 67	27.9
TreeDP [8]	59.1	22.4 49	23.1 48	22.3 45	12.1 57	12.9 57	21.7 53	32.4 68	38.9 68	45.6 66	23.7 68	30.8 68	31.7 62	26.5
DP [1b]	59.8	19.6 39	20.6 40	22.8 48	23.5 71	24.3 70	32.8 65	30.0 66	36.3 66	36.1 57	22.0 66	29.6 66	33.7 64	27.6
PhaseDiff [23]	59.9	11.5 17	13.6 21	24.5 57	16.5 68	17.9 68	34.8 67	36.5 71	43.0 71	45.8 67	36.1 71	43.3 71	43.3 70	30.6
SO [1c]	61.0	17.9 30	19.8 37	23.4 52	23.1 70	24.4 71	34.6 66	36.3 70	42.9 70	41.4 63	23.7 69	32.4 69	34.4 65	29.5
DPVI [67]	61.2	29.7 71	30.4 71	28.3 65	13.1 61	13.9 59	29.3 63	19.9 54	25.9 53	36.1 56	17.8 61	23.8 58	33.5 63	25.2
Infection [10]	61.6	22.1 48	23.5 50	37.6 69	11.1 56	12.3 56	37.3 69	24.3 61	31.7 61	56.4 71	20.2 63	27.8 64	48.2 71	29.4
IMCT [62]	61.7	24.5 58	25.5 60	29.6 66	10.9 53	11.7 54	30.9 64	26.3 62	32.0 62	46.0 68	20.7 64	26.4 63	37.2 66	26.8
STICA [16]	63.2	24.3 56	26.1 64	44.8 71	15.2 64	16.6 65	46.3 71	24.0 59	31.3 60	50.4 70	15.9 53	23.7 56	39.5 69	29.8
RegionalSup [38]	64.2	25.7 66	27.3 67	25.6 60	16.8 69	18.2 69	40.9 70	27.1 64	34.7 64	42.6 64	16.5 56	25.9 62	28.6 59	27.5
SSD+MF [1a]	65.3	28.5 70	30.0 70	38.1 70	12.8 60	14.2 61	23.4 56	28.2 65	35.4 65	48.7 69	21.1 65	29.4 65	38.6 68	29.0

Capitolo 3

Stereo matching "MSOM"

Sarà ora descritto e analizzato il modello *MSOM*, riportato in [40], sul quale è stato sviluppato l'algoritmo proposto in questo elaborato. Per prima cosa è necessario specificare come è strutturato *MSOM* e come utilizzarlo al fine di ottenere, da una coppia di immagini stereo, la mappa di disparità associata.

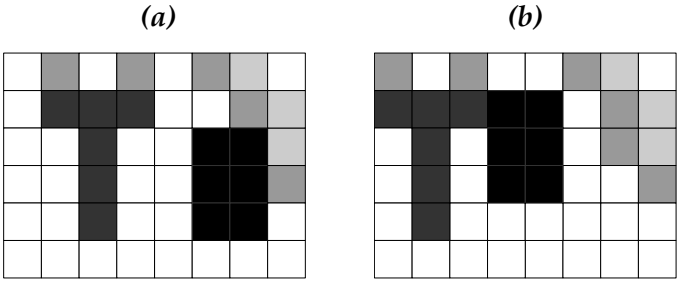
Come suggerito dal nome *MSOM,* è stata presa una rete autorganizzante *SOM* e sono state introdotte alcune modifiche al fine di poter usare la rete per calcolare una mappa di disparità densa e non, come nella maggior parte delle applicazioni di reti *SOM*, per effettuare un clustering di dati. Riferendosi alla descrizione generale del modello *SOM*[1], le principali modifiche riguardano lo step di aggiornamento dei pesi neurali; per gli altri aspetti e comportamenti la rete è paragonabile ad una mappa di Kohonen opportunamente inizializzata come si vedrà in 3.2.

Si ricorda che *MSOM* è un modello generale e non presuppone alcun vincolo epipolare per la coppia di immagini stereo; per questo, oltre alla mappa delle disparità orizzontali, è anche previsto il calcolo di una mappa delle disparità verticali. Una delle prime modifiche che si vedranno nel Capitolo 4, con la definizione del modello *StereoSOM,* è appunto l'introduzione del vincolo epipolare.

Il capitolo terminerà con una valutazione del costo computazionale richiesto da *MSOM*.

[1] Per la descrizione del modello *SOM* vedere l'Appendice B.

Figura 3.1: Semplicissima coppia stereo usata come esempio rappresentativo. ***(a)***: pixel che compongono l'immagine di sinistra I_L. ***(b)***: pixel che compongono l'immagine di destra I_R.



3.1 Esempio rappresentativo

Viene ora mostrato un esempio di coppia stereo che sarà impiegato nelle prossime sezioni per descrivere il modello *MSOM*. È stato scelto appositamente un esempio semplice e di dimensioni ridotte[2] al fine di rendere maggiormente comprensibile il meccanismo alla base di *MSOM*.

La coppia stereo in questione è mostrata in Figura 3.1 mentre in Figura 3.2 nella pagina successiva sono riportate le mappe di disparità reale associate alla coppia stereo.

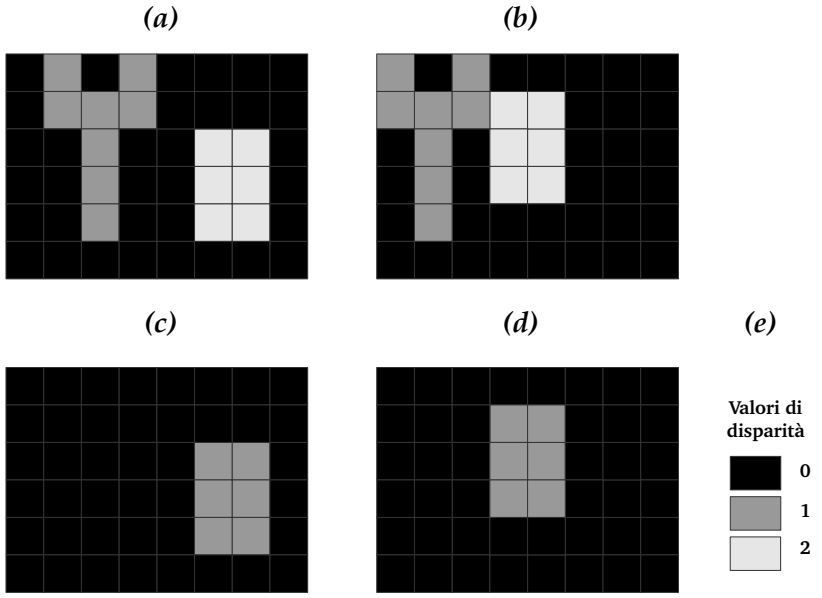
Nelle immagini sono rappresentati due semplici oggetti con disparità diverse, mentre sullo sfondo bianco a disparità nulla è visibile un motivo nell'angolo in alto a sinistra. Da notare come siano presenti per ogni immagine che compone la coppia stereo due mappe di disparità, una orizzontale e una verticale; solitamente è di uso comune calcolare solo la mappa delle disparità orizzontali in quanto i dataset sono spesso rettificati e basati su un sistema stereo laterale, tuttavia in questo primo modello si vuole trattare il problema delle corrispondenze in un modo molto generale.

3.2 Modello

L'algoritmo *MSOM* si compone di quattro principali step che, opportunamente combinati, permettono di risolvere il problema delle corrispondenze stereo fornen-

[2]Con un numero limitato di pixel.

Figura 3.2: ***(a)*** e ***(b)***: mappe (esatte) delle disparità orizzontali per la coppia stereo in Figura 3.1 nella pagina precedente. ***(c)*** e ***(d)***: mappe (esatte) delle disparità verticali. ***(e)***: valori di disparità associati ai colori.



do mappe di disparità dense. Al fine di descrivere al meglio il modello *MSOM* è stata destinata una sezione per ogni step che compone l'algoritmo. Nelle sezioni saranno descritti ed analizzati gli aspetti formali di ogni step, insieme ad alcuni esempi intuitivi che meglio chiariscono il meccanismo alla base del funzionamento di *MSOM*.

Riassumendo, nel primo step (3.2.1) vengono inizializzati i pesi neurali nella mappa di Kohonen mentre nel secondo e terzo step (3.2.2 e 3.2.3), in pieno stile *SOM*, viene eletto un nodo vincente e conseguentemente aggiornati i pesi nel suo vicinato spaziale di neuroni. Il quarto ed ultimo step (3.2.4) viene eseguito una sola volta al termine dell'algoritmo e serve per estrarre la mappa di disparità dai pesi neurali. In 3.2.5 è presentato lo pseudocodice che riassume come devono essere adeguatamente combinati ed alternati gli step descritti.

3.2.1 Step 1: inizializzazione della SOM

Siano H e W l'altezza e la larghezza in pixel delle due immagini I_L e I_R. Per ogni pixel dell'immagine di sinistra $I_L(r, c)$ si inizializza un neurone della rete *SOM* w^{rc}.

Ogni nodo della rete è rappresentato da un vettore di pesi $w^{rc} = (w_1^{rc}, w_2^{rc}, w_3^{rc})$, le prime due componenti corrispondono inizialmente alle coordinate del neurone nella mappa di Kohonen (quindi $w_1^{rc} = r$ e $w_2^{rc} = c$) mentre la terza componente corrisponde al valore di intensità luminosa[3] del pixel ($w_3^{rc} = I_L(r, c)$). In Figura 3.3 a fronte è riportato uno schema che evidenzia quanto è stato appena descritto.

Il significato delle componenti w_1 e w_2 e da cercarsi nelle corrispondenze tra le due immagini che formano la coppia stereo. Si supponga ad esempio che vi sia una corrispondenza tra i pixel $I_L(a, b)$ e $I_R(a', b')$, alla fine del processo di addestramento, se non ci sono errori, il neurone alle coordinate (a, b) avrà i primi due pesi neurali pari a $w_1^{ab} = a'$ e $w_2^{ab} = b'$; questo concetto è illustrato in Figura 3.4 nella pagina successiva. Per questo motivo la rete *MSOM*, durante l'apprendimento non supervisionato, raccoglie al suo interno una mappa delle corrispondenze più che una mappa di disparità.

L'immagine di destra I_R contiene invece i dati da presentare in input alla rete *SOM*, in particolare si considerano vettori di input nella forma $(i_1^{mn}, i_2^{mn}, i_3^{mn})$, dove le prime due componenti sono le coordinate del pixel in I_R ($i_1^{mn} = m$, $i_2^{mn} = n$) mentre la terza componente è il valore di intensità del pixel ($i_3^{mn} = I_R(m, n)$).

Secondo questa descrizione tra i neuroni della rete *SOM* e i pixel dell'immagine di sinistra I_L esiste un rapporto di corrispondenza *1:1*; lo stesso vale per l'insieme dei vettori di input e i pixel dell'immagine di destra I_R. Per garantire una certa intuitività alla spiegazione, in seguito si parlerà di confronti "spaziali" o "di intensità" tra i pixel dell'immagine I_R e i neuroni della rete *SOM*. Formalmente, per quanto riguarda lo spazio, bisognerebbe parlare di confronti tra le coppie di componenti (i_1^{mn}, i_2^{mn}) e (w_1^{rc}, w_2^{rc}), mentre per quanto riguarda le intensità luminose bisognerebbe parlare di confronti tra le componenti i_3^{mn} e w_3^{rc}.

3.2.2 Step 2: elezione del neurone vincente

Secondo il principio generale del modello *SOM*, presentando un certo input alla rete un solo neurone sarà eletto vincitore (*Winner Take All*), definendo successivamente le modalità per l'aggiornamento dei pesi nel suo vicinato spaziale. La scelta del nodo vincente si basa solitamente su una metrica, ad esempio la distanza euclidea tra i pesi del vincitore e il vettore in input.

[3] Nelle realizzazioni tecniche, l'intensità luminosa si riduce ad un valore numerico che rappresenta un livello di grigio.

Figura 3.3: Semplice esempio che evidenzia la corrispondenza *1:1* tra i pixel dell'immagine di sinistra I_L e i neuroni artificiali della rete *SOM*.

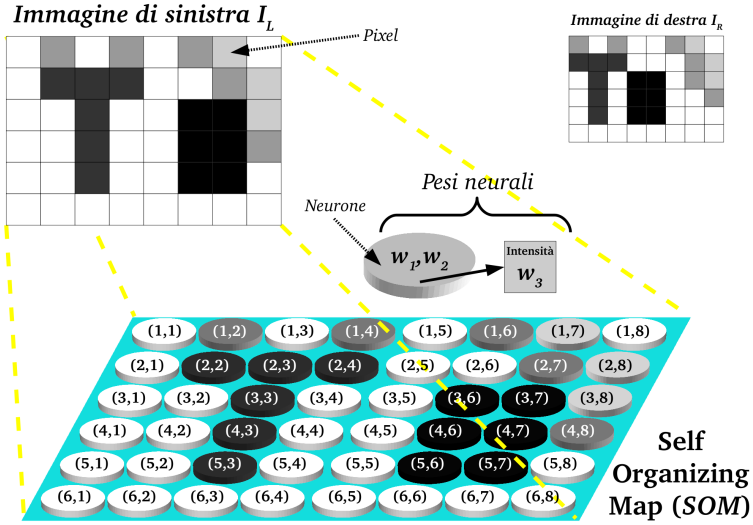
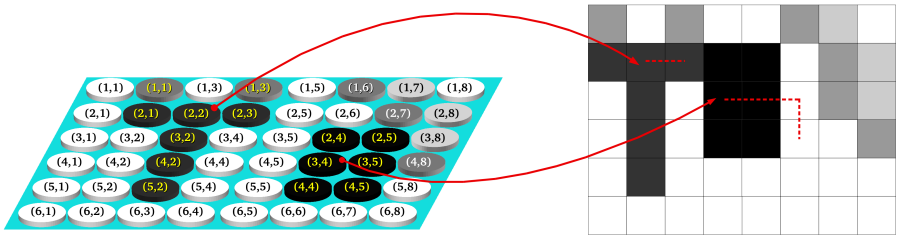


Figura 3.4: Stato della rete *SOM* dopo un ideale apprendimento non supervisionato, sono evidenziati in giallo i pesi neurali corrispondenti a punti con disparità non nulla. Le frecce rosse evidenziano due esempi di corrispondenza, le linee tratteggiate rappresentano invece il percorso da fare per trovare il corrispondente punto omologo. Ovviamente esiste una corrispondenza per ogni neurone della rete.



Dato il neurone vincente w^{φ_r,φ_c} e il suo vicinato spaziale[4] $neigh\left(w^{\varphi_r,\varphi_c}\right)$, l'aggiornamento comporterà per ciascuno dei pesi $neigh\left(w^{\varphi_r,\varphi_c}\right)$ un avvicinamento al valore di input che ha permesso l'elezione del neurone w^{φ_r,φ_c}. L'elezione del nodo vincente e il successivo aggiornamento dei pesi neurali nel suo vicinato, se ripetuti un numero abbastanza grande di volte, portano alla creazione di bolle di neuroni vicini che si attivano per valori di input simili. Questo principio è utilizzato in molte applicazioni come fase di pre-processing, per effettuare il clustering dei dati e ridurne la dimensionalità.

Le due fasi appena descritte sono presenti in *MSOM*, ma seguono una logica piuttosto diversa in quanto non si vuole eseguire una clusterizzazione di dati ma si vuole trovare una mappa di disparità densa.

Vediamo come avviene l'elezione del neurone vincente nel modello *MSOM*. Si estraggano casualmente gli interi $m \in \{1,\dots,H\}$ e $n \in \{1,\dots,W\}$. Sia quindi $(i_1^{mn}, i_2^{mn}, i_3^{mn})$ il vettore di input corrispondente al pixel dell'immagine $I_R\left(m,n\right)$, il calcolo delle coordinate (φ_r,φ_c) del neurone vincente è definito nel modo seguente:

$$(\varphi_r,\varphi_c) = \underset{r\in\{1,\dots,H\},c\in\{1,\dots,W\}}{\arg\min} \sqrt{\sum_{k=1}^{3}\left(w_k^{rc} - i_k^{mn}\right)^2} \tag{3.1}$$

Si ricerca perciò il punto di minimo globale per una funzione *distanza euclidea* tra i vettori di input presentati alla rete e i pesi dei neuroni.

L'idea che sta alla base del modello *MSOM* è la seguente: il matching tra i pixel delle due immagini I_L e I_R è espresso dalle vittorie dei neuroni nella rete MSOM. In particolare, se le coordinate del nodo vincente sono (φ_r,φ_c), allora ci si aspetta un matching tra i pixel $I_L\left(\varphi_r,\varphi_c\right)$ e $I_R\left(m,n\right)$.

Questa idea è banale nel caso della terza componente del vettore dei pesi neurali, è infatti ragionevole pensare che valori di intensità simili tra un pixel dell'immagine di sinistra e un pixel dell'immagine di destra siano indizio di una possibile corrispondenza. Per quanto riguarda invece le prime due componenti il discorso non è altrettanto intuitivo ed è necessario ben comprendere il meccanismo di aggiornamento dei pesi nel vicinato del neurone vincente, descritto in 3.2.3. Concettualmente, inserendo le prime due componenti nella ricerca, si garantisce una certa

[4]Per *vicinato spaziale* si intende l'insieme dei neuroni ad una certa distanza dal nodo eletto $w^{r'c'}$. Ad esempio, usando la semplice distanza euclidea, i neuroni del vicinato sono $neigh\left(w^{\varphi_r,\varphi_c}\right) = \{w^{r,c}\ t.c.\ |\varphi_r - r| \leq d, |\varphi_c - c| \leq d\}$ dove d è il valore di distanza massima.

continuità spaziale alla disparità e si aiuta la corretta stima della disparità in quelle aree dove il solo confronto della componente intensità genera delle inevitabili ambiguità.

Nella prefazione si è parlato di *StereoSOM* come un metodo di stero matching locale, ma dall'equazione 3.1 è evidente come la ricerca coinvolga tutti i neuroni nella rete *SOM*, questo deriva dal fatto che *MSOM* è un modello generale che si applica a qualunque coppia di immagini stereo. È facile tuttavia limitare questa ricerca ad un'area circoscritta di neuroni mediante alcuni vincoli tipicamente impiegati nell'analisi di stereocoppie. Elenchiamo quelli direttamente utilizzabili in *MSOM*[5]:

1. Vincolo epipolare
2. Intervallo di disparità

Il vincolo epipolare "nasce" dal processo di rettificazione delle immagini stereo ed introduce un notevole vantaggio in termini di efficienza, permettendo di eseguire la ricerca su una sola linea di neuroni, solitamente la linea orizzontale con coordinata $r = i_1^{mn} = m$ nei sistemi stereo laterali.

L'intervallo di disparità permette di ridurre ulteriormente il campo di ricerca, limitandolo di fatto ad un piccolissimo sottoinsieme di neuroni.

StereoSOM si distingue da *MSOM* anche per l'utilizzo dei vincoli sopra descritti. Questa limitazione di generalità non vuole rappresentare un merito o una innovazione, ma una scelta dovuta al fatto che il sistema di valutazione utilizzato [41] fornisce dataset composti da immagini in forma standard e dotate di valori di disparità minima e massima, inoltre la maggior parte degli algoritmi con cui *StereoSOM* compete sfruttano a pieno il vincolo epipolare e l'intervallo di disparità.

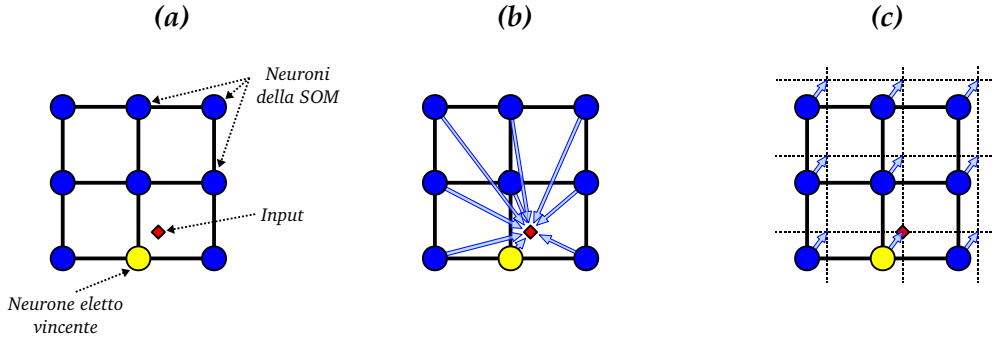
Si ricorda che, se fosse mai richiesto, è comunque una facile impresa rimuovere i vincoli e ricondurre *StereoSOM* ad un modello generale come *MSOM*.

3.2.3 Step 3: aggiornamento dei pesi neurali

In 3.2.2 è stato brevemente descritto il meccanismo alla base del modello *SOM*: dopo la fase di elezione del neurone vincente w^{φ_r,φ_c}, si considera un suo vicinato

[5]Per una descrizione più dettagliata dei vincoli sulle stereocoppie fare riferimento a 1.2 nella pagina 8.

Figura 3.5: ***(a)***: Sezione di rete SOM e neurone eletto vincitore allo step 2. ***(b)***: meccanismo di aggiornamento nelle reti *SOM*, i pesi dei neuroni nel vicinato del nodo vincente si avvicinano al valore in input. ***(c)***: aggiornamento neurale in *MSOM*, i pesi dei neuroni si avvicinano al valore di disparità del nodo eletto.



spaziale $neigh\left(w^{\varphi_r,\varphi_c}\right)$ e si effettua un aggiornamento al fine di avvicinare i pesi dei neuroni in $neigh\left(w^{\varphi_r,\varphi_c}\right)$ ai pesi dell'input che hanno permesso l'elezione di w^{φ_r,φ_c} come nodo vincente.

Iterando gli step di elezione e di aggiornamento neurale, tendono a crearsi delle bolle di attivazione che riconoscono input simili, rendendo la rete autorganizzante uno strumento per ridurre dimensionalmente i dati ed effettuare operazioni di clustering.

È già stato spiegato come in *MSOM* le reti autorganizzanti non siano utilizzate per la clusterizzazione dei dati ma per il calcolo delle mappe di disparità, per questo motivo l'aggiornamento dei nodi nel vicinato spaziale in *MSOM* è da pensare come un metodo per fare "avvicinare" il valore di disparità dei neuroni in $neigh\left(w^{\varphi_r,\varphi_c}\right)$ a quello del neurone eletto w^{φ_r,φ_c}, valore risultante dalla distanza tra la posizione del nodo eletto e la posizione dell'input. Una efficace rappresentazione del concetto appena descritto è visibile in Figura 3.5.

È facile dedurre che i risultati migliori si avranno quando il vicinato di neuroni corrisponde ad un insieme di pixel a disparità costante, infatti se all'interno del vicinato dovessero trovarsi uno o più nodi corrispondenti a pixel con disparità diversa da quella del neurone eletto, si arriverebbe certo ben lontani da una mappa di disparità corretta.

MSOM sfrutta una importante assunzione per cercare di risolvere il problema appena descritto: è ragionevole pensare che, all'interno di una immagine, una piccola zona circoscritta di pixel con valori di intensità simili corrispondano con alta

probabilità ad un singolo oggetto. Se si assume inoltre che una piccola zona di un oggetto abbia disparità pressoché costante[6] è facile trovare una soluzione che permetta di favorire l'aggiornamento neurale per neuroni corrispondenti a pixel con disparità simile. Sorprendentemente, il principio appena descritto è entrato in uso negli algoritmi di stereo matching solo recentemente, si veda per esempio [39].

In *MSOM* il vicinato si definisce usando una particolare funzione che riduce l'entità dell'aggiornamento tanto più quanto il neurone ha una intensità luminosa diversa da quella del nodo eletto; questa funzione è illustrata in Figura 3.6 nella pagina seguente.

Lo step di aggiornamento dei pesi neurali riguarda soltanto le prime due componenti di w, w_1 e w_2, dalle quali sarà poi possibile estrarre le mappe di disparità[7]. Il terzo peso w_3 non mantiene informazioni legate alla disparità ma contiene l'intensità luminosa del pixel; per questa ragione non sarà interessato dallo step di aggiornamento neurale e il suo valore rimarrà costante durante l'esecuzione dell'algoritmo.

Formalmente, l'aggiornamento dei pesi in *MSOM* è specificato nella seguente regola di apprendimento:

$$w_k^{rc} \leftarrow w_k^{rc} + h_k(r,c)\, g_k(r,c) \left(i_k^{(r-(\varphi_r - m))(c-(\varphi_c - n))} - w_k^{rc} \right), \tag{3.2}$$

dove

$$h_k(r,c) = \eta \exp\left(-\frac{(r-\varphi_r)^2 + (c-\varphi_c)^2}{2\sigma_h^2} \right) \tag{3.3}$$

e

$$g_k(r,c) = \exp\left(-\frac{\left(w_3^{rc} - w_3^{\varphi_r \varphi_c}\right)^2}{2\sigma_g^2} \right) \tag{3.4}$$

per $k = 1,2$, $\forall r, m \in \{1,2,\ldots,H\}$ e $\forall c, n \in \{1,2,\ldots,W\}$.

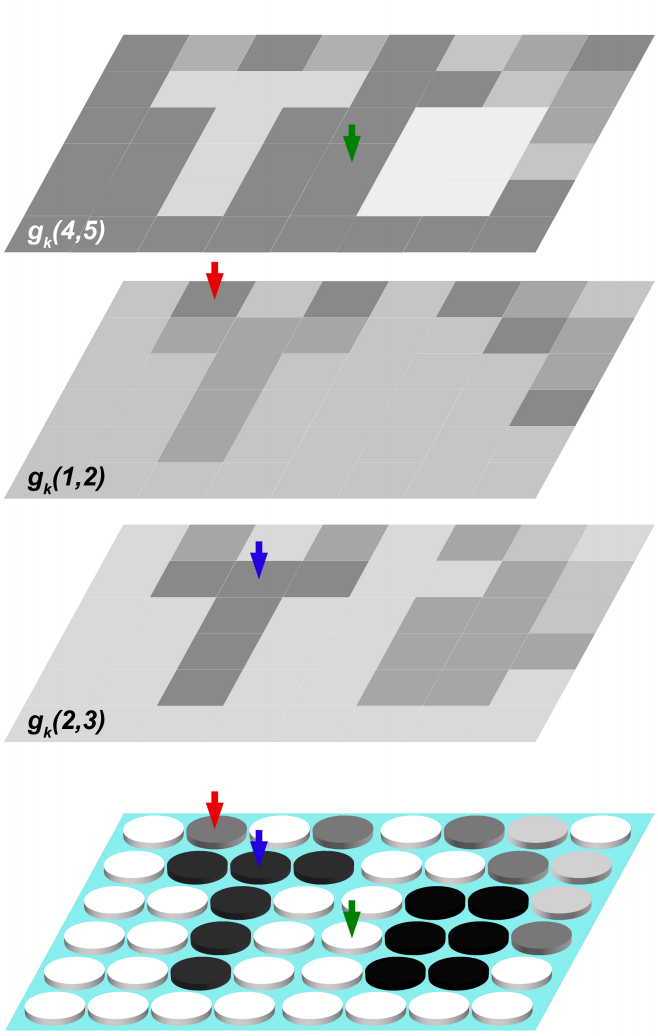
Per fissare le idee si analizzeranno ora alcuni termini fondamentali riportati nelle equazioni 3.2, 3.3 e 3.4:

$(\varphi_r - m)$ è un valore di disparità verticale, calcolato come distanza tra la coordinata verticale del neurone vincitore e quella del pixel scelto casualmente da I_R.

[6] Questa è una conseguenza della più generale ipotesi di continuità della disparità.

[7] La mappa delle disparità verticali si estrae da w_1, mentre da w_2 si ottiene la mappa delle disparità orizzontali.

Figura 3.6: Rappresentazione grafica della funzione 3.4 applicata in tre diversi punti. Valori alti della funzione, indicati nell'immagine con colori scuri, indicano alta somiglianza con l'intensità del neurone vincente.



$(\varphi_c - n)$ è un valore di disparità orizzontale, calcolato come distanza tra la coordinata orizzontale del neurone vincitore e quella del pixel scelto casualmente da I_R.

$i_k^{(r-(\varphi_r-m))(c-(\varphi_c-n))}$, $k = 1, 2$ sono le coordinate del pixel scelto dall'immagine di destra I_R che, secondo la disparità attualmente calcolata nella fase di elezione del neurone vincente, dovrebbe corrispondere al pixel (r, c) nell'immagine di sinistra I_L.

$h_k(r, c)$ è una funzione a due variabili di tipo gaussiano, con varianza σ_h^2. Presenta valori prossimi a 1 per coordinate (r, c) vicine a quelle del neurone vincente (φ_r, φ_c). Per coordinate lontane da quelle del nodo vincente, il valore della funzione sarà prossimo allo 0, rendendo nulla l'entità dell'aggiornamento dei pesi neurali. η è il *learning rate* della *SOM* e può essere pensato come la velocità di apprendimento non supervisionato della rete. In molti problemi risolti da reti autorganizzanti, valori elevati di η permettono un più rapido addestramento della *SOM*, ma è importante tenere conto che la qualità dei risultati decresce con l'aumentare di η. Il valore del learning rate deve comunque restare nell'intervallo $0 \leq \eta \leq 1$.

$g_k(r, c)$ è una funzione ad una variabile[8] di tipo gaussiano, con varianza σ_g^2. La funzione presenta valori prossimi a 1 se il valore di intensità del pixel $I_R(r, c)$ è simile a quello del neurone vincente. Per valori intensità che differiscono sostanzialmente, il valore della funzione sarà molto vicino allo 0, rendendo nulla l'entità dell'aggiornamento ai pesi neurali.

Da notare come le funzioni h e g regolino l'entità dell'aggiornamento neurale: la funzione h è responsabile del fenomeno delle bolle di attivazione locali, tipico delle reti *SOM*, mentre la funzione g realizza l'assunzione, descritta in precedenza, che a pixel con intensità simili corrispondono con alta probabilità oggetti con disparità costante.

[8] La variabile in questo caso è una differenza di intensità luminosa.

3.2.4 Step 4: estrazione delle mappe di disparità

Alla fine del processo di apprendimento non supervisionato è necessario ricavare le mappe di disparità dai pesi neurali, questo processo è molto semplice e visto che *MSOM* contiene al suo interno una mappa delle corrispondenze, fatto visibile in Figura 3.4 nella pagina 39.

Le mappe delle disparità orizzontali e verticali sono estraibili dalle prime due componenti (w_1, w_2) dei pesi neurali:

$$\begin{cases} d^{ver}(r,c) = i_1^{rc} - w_1^{rc} = r - w_1^{rc} \\ d^{hor}(r,c) = i_2^{rc} - w_2^{rc} = c - w_2^{rc} \end{cases} \tag{3.5}$$

3.2.5 Algoritmo MSOM

Algoritmo 3.1 Pseudo codice dell'algoritmo *MSOM*

Input: una coppia di immagini stereo I_L e I_R, con dimensioni $H \times W$;

Output: due mappe di disparità dense (disparità orizzontali e verticali) riferite all'immagine I_L: $d^{hor}(r,c)$ e $d^{ver}(r,c)$;

Require: inizializzazione della rete *SOM* e dei vettori di input (3.2.1);

1: **for** $i = 1$ to $Iterations$ **do**

2: elezione del neurone vincente (φ_r, φ_c) (3.2.2);

3: aggiornamento dei pesi neurali per i neuroni in $neigh\left(w^{\varphi_r,\varphi_c}\right)$ (3.2.3);

4: **end for**

5: estrazione delle mappe di disparità d^{hor} e d^{ver} (3.2.4);

Il numero di iterazioni dipende dalla dimensione delle immagini e dal valore dei parametri σ_g^2 e soprattutto σ_h^2: maggiori saranno i valori delle varianze, maggiore sarà l'entità dell'aggiornamento neurale per ogni iterazione. Come si vedrà nel Capitolo 5, il metodo non è poi così sensibile alle variazioni di σ_g^2 e σ_h^2 da rendere necessaria una accurata taratura iniziale dei parametri tuttavia, se si vuole usare

MSOM come algoritmo di stereo matching in applicazioni reali, una fase di tuning iniziale è preferibile.

Come indicato in [40] il numero di iterazioni *Iterations* è da scegliere dell'ordine di 100 $(H \times W)$, dove H è l'altezza in pixel dell'immagine e W è la larghezza.

3.3 Complessità computazionale

L'analisi della complessità computazionale per l'algoritmo appena descritto è semplice, tuttavia ci si limiterà alla sola complessità in tempo, riferita ad una realizzazione non ottimizzata. Si analizzerà inoltre soltanto il caso del modello generale *MSOM*, senza alcuna ottimizzazione che sfrutti la località degli aggiornamenti neurali; comunque, nella stessa trattazione riferita al modello *StereoSOM* (4.5), si vedrà come una semplice ridefinizione della funzione di aggiornamento possa ridurre sostanzialmente i tempi di esecuzione.

La complessità in spazio dipende molto dalle scelte progettuali e implementative, una trattazione specifica richiederebbe l'introduzione di numerosi concetti algoritmici e di progettazione che vanno al di là delle intenzioni di questo elaborato.

Si prenda come dimensione dell'input n la quantità di pixel di una delle due immagini della coppia stereo[9], sia quindi $n = H \times W$, con H e W rispettivamente l'altezza e la larghezza dell'immagine.

L'inizializzazione della *SOM* richiede un tempo lineare nella dimensione dell'input in quanto si copiano i valori delle intensità dei pixel dell'immagine di sinistra nei corrispondenti pesi neurali, gli altri pesi vengono semplicemente impostati al valore delle coordinate dei rispettivi neuroni. Il tempo richiesto dal primo step è quindi n. La ricerca del neurone vincente (Equazione 3.1) interessa tutti i nodi della rete *SOM*; trascurando l'estrazione casuale delle coordinate iniziali si ha un costo lineare pari a n. Anche l'aggiornamento dei pesi interessa tutti i nodi della rete *SOM* e richiede quindi un tempo n, tuttavia sarebbe buona pratica tenere in conto anche il tempo richiesto per il calcolo dell'equazione 3.2, contenente diversi termini esponenziali. Sempre in riferimento alla complessità dello step di aggiornamento, si ricorda che la funzione h (Equazione 3.3) può essere pre calco-

[9] Le immagini di una coppia stereo devono comunque avere le stesse dimensioni.

lata e conseguentemente utilizzata efficientemente mediante un semplice principio di programmazione dinamica.

Dall'algoritmo 3.2.5 si può ora facilmente calcolare il costo totale, dato dal numero di iterazioni ($100n$) e dai costi degli step, che risulta essere complessivamente $100n \cdot (n + n) = 200n^2$. Un fattore 200 risulta decisamente troppo grande e inaccettabile in contesti applicativi dove sono richiesti tempi ridotti di esecuzione. Il costo asintotico $O\left(n^2\right)$ risulta comunque polinomiale nella dimensione dell'input.

Un metodo per ridurre i tempi di esecuzione, senza tuttavia ridurre la complessità asintotica, consiste nel limitare l'area di ricerca impiegata nel secondo step. Precisamente, date le coordinate del pixel estratto casualmente dall'immagine di destra (m, n), la ricerca del neurone vincente può essere limitata ad una finestra di dimensioni (w, h) centrata sul nodo w^{mn} . Utilizzando questa ottimizzazione il costo complessivo dell'algoritmo diventa $100n \cdot ((w \cdot h) + n) = 100\left(n \cdot w \cdot h + n^2\right)$.

Si consideri per esempio una coppia stereo con dimensioni 200×200 pixel, il tempo richiesto dal metodo con ricerca non ottimizzata è pari a 200^5. Scegliendo una finestra con $w = 20$ e $h = 10$, il tempo di ricerca si riduce a $100\left(200^3 + 200^4\right)$ con uno *speedup* approssimativo di 2, se poi la coppia stereo è stata pre-elaborata con un algoritmo di rettificazione è possibile ignorare la disparità verticale, riducendo lo spazio di ricerca di un fattore 10 e riducendo ulteriormente i tempi.

Capitolo 4

Il metodo "StereoSOM"

Il capitolo conterrà la descrizione e l'analisi del modello *StereoSOM*, proposto in questo elaborato. Sarà mantenuto lo stile utilizzato nel Capitolo 3 ma, data la natura applicativa di *StereoSOM*, si darà più enfasi a concetti di ottimizzazione e di progettazione.

Il nome *StereoSOM* può sembrare sintomo di una mancanza di originalità da parte degli autori, in realtà è stata anteposta a *SOM* la parola *Stereo* per sottolineare la natura cooperativa a doppia rete autorganizzante del modello. Vengono utilizzate infatti due reti *SOM*, una per ogni immagine della coppia stereo, che cooperano al fine di rilevare e gestire al meglio le zone occluse. Il risultato interessante, come si vedrà nel Capitolo 5, è che questa cooperazione porta a una notevole riduzione degli errori di corrispondenza, specialmente in prossimità di zone occluse.

Ma le novità non si limitano alla gestione delle occlusioni; al fine di ridurre notevolmente l'ambiguità delle corrispondenze è stata introdotta una funzione di matching basata su un particolare supporto che riesce, in modo completamente non supervisionato, ad adattarsi ai profili degli oggetti evitando i classici errori di corrispondenza nelle zone di discontinuità di profondità.

A differenza di *MSOM*, *StereoSOM* non è un modello generale, viene infatti sfruttato il vincolo epipolare sulla coppia di immagini stereo per ottimizzare il processo di apprendimento non supervisionato. Questa perdita di generalità è dovuta ad un aspetto applicativo: *StereoSOM* vuole infatti competere con recenti ed efficienti algoritmi di stereo matching che sfruttano il vincolo epipolare[1]. Ovviamente

[1]Questa tipologia di algoritmi è generalmente la più utilizzata in applicazioni reali.

i risultati di *StereoSOM* saranno delle mappe di disparità prettamente orizzontali. Non è comunque difficile rendere *StereoSOM* un modello generale, è stata infatti mantenuta una forma di base molto simile a *MSOM* e le modifiche introdotte possono facilmente estendersi ad un caso privo del vincolo epipolare.

Il capitolo terminerà con la valutazione del costo computazionale richiesto da *StereoSOM* e con una analisi comparativa tra il modello originale e l'estensione proposta.

4.1 Esempio rappresentativo

L'esempio di coppia stereo impiegato in questo capitolo differisce da quello utilizzato per descrivere il modello *MSOM* per due motivi, il primo è l'impiego dei colori, il secondo è l'assenza di disparità verticale. *StereoSOM* non si limita quindi ad una sola intensità luminosa, ma fa uso di più canali colore[2] per trovare le corrispondenze. Inoltre, come è stato anticipato nell'introduzione del capitolo, *StereoSOM* sfrutta il vincolo epipolare e si applica ad immagini in forma standard; è quindi assente il concetto di disparità verticale.

La coppia stereo utilizzata, completa di mappe di disparità, è mostrata in Figura 4.1 a fronte.

4.2 Specializzazione del modello *MSOM*

Nei prossimi paragrafi sono descritte le modifiche apportate al modello *MSOM* al fine di poter sfruttare proficuamente stereocoppie in forma standard e dotate di minimo e massimo valore di disparità.

Sono stati introdotti inoltre i canali colore al posto della sola intensità luminosa, questo migliorerà i risultati con l'utilizzo di stereocoppie a colori.

Infine, alcune modifiche di carattere tecnico porteranno all'ottimizzazione computazionale della fase di aggiornamento neurale.

Figura 4.1: Esempio di coppia stereo a colori e priva di disparità verticale. ***(a)***: pixel che compongono l'immagine di sinistra I_L. ***(b)***: pixel che compongono l'immagine di destra I_R. ***(c)*** e ***(d)***: Mappe di disparità reale (orizzontale) associate. ***(e)***: legenda valori di disparità.

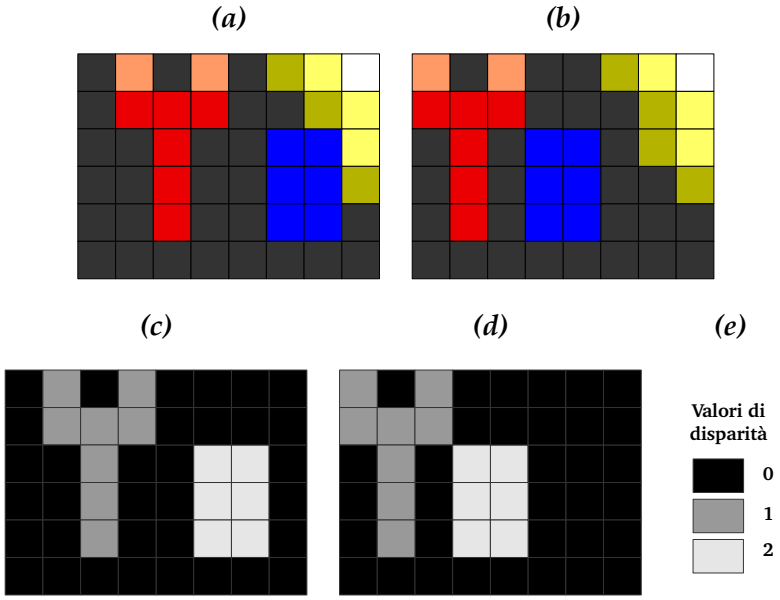
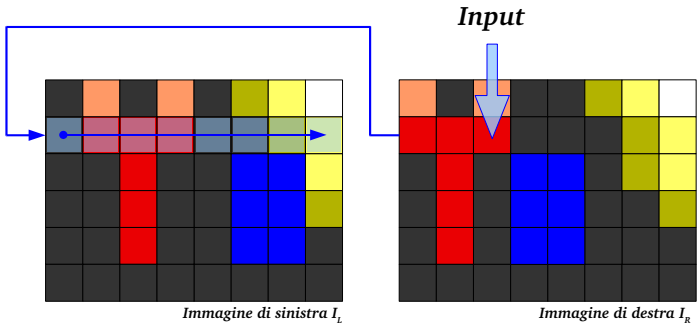


Figura 4.2: Applicazione del vincolo epipolare alla stereocoppia di esempio in Figura 4.1. La ricerca della corrispondenza avviene sulla sola linea orizzontale che interseca il pixel in input.



4.2.1 Vincolo epipolare

Introdurre il vincolo epipolare significa specializzare l'algoritmo per immagini rettificate rispetto ad un sistema stero laterale. Schematizzando, questa modifica implica le seguenti assunzioni:

- Presentando un input alla rete *SOM*, la ricerca del neurone vincente non si estenderà a tutta la mappa, ma solo alla linea di neuroni orizzontale con coordinata i_1^{mn}. Alla fine della ricerca, per il neurone vincente si avrà che $\varphi_r = i_1^{mn} = m\,,\ \forall r$.

- Il valore di disparità verticale $(\varphi_r - m)$ sarà sempre nullo.

- Il valore del peso w_1^{rc} nella *SOM* non cambierà durante l'esecuzione dell'algoritmo.

- Alla terminazione dell'algoritmo, la mappa delle disparità verticali sarà una matrice di valori nulli, ossia $d^{ver}\,(r,c) = 0\,,\ \forall r,c$.

Alla luce di questi fatti si possono attuare delle importanti semplificazioni all'algoritmo *MSOM*, ad esempio si può rimuovere completamente il peso w_1^{rc}, così come il risultato della ricerca del neurone vincente può essere rappresentato dalla sola coordinata φ_r, visto che $\varphi_r = m\,,\ \forall r$.

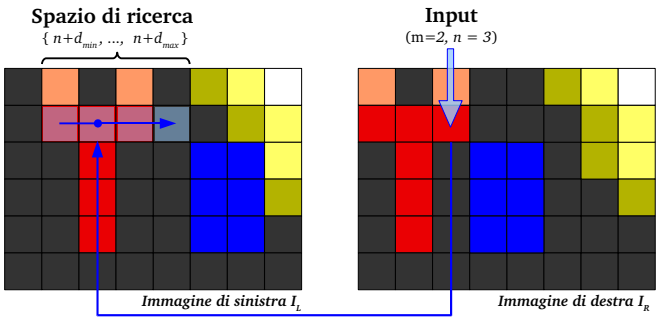
Ogni nodo della rete sarà quindi rappresentato dal vettore di pesi $w^{rc} = (w_1^{rc}, w_2^{rc})$, dove la prima componente w_1^{rc} gestisce le corrispondenze orizzontali e w_2^{rc} l'intensità luminosa, analogamente i dati da presentare in input alla rete saranno vettori nella forma (i_1^{mn}, i_2^{mn}).

È possibile vedere un esempio di applicazione del vincolo epipolare in Figura 4.2 nella pagina precedente.

Si ricorda che con immagini non in forma standard è sconsigliato utilizzare l'algoritmo specializzato, in quanto verrebbe trascurata la *disparità verticale* nella ricerca del neurone vincente, con ovvie conseguenze negative sulla qualità dei risultati. Test a riguardo saranno effettuati nel Capitolo 5 dove si introdurrà una eccezione al vincolo epipolare sviluppata ad hoc per trattare immagini non in forma standard.

[2]Precisamente, negli esperimenti riportati nel Capitolo 5, è stato impiegato il modello *RGB*, a tre componenti cromatiche.

Figura 4.3: Applicazione del vincolo di disparità minima/massima alla coppia di esempio in Figura 4.1 nella pagina 51. L'intervallo di disparità utilizzato è $[d_{min} = -1, d_{max} = 2]$.



4.2.2 Vincolo di disparità minima e massima

In sistemi di stereo matching reali si dispone spesso di un valore di disparità minimo e massimo derivabili, come si è visto nel Capitolo 1, dalla particolare geometria del sistema stereoscopico. L'introduzione di questo vincolo può portare a risultati di qualità migliore ma, il vantaggio più grande, è soprattutto un incremento di efficienza dell'algoritmo nella fase di elezione del neurone vincente.

Siano d_{min} e d_{max} i valori di disparità orizzontale minima e massima rispettivamente, allora nell'equazione 3.1 c non sarà più cercato nell'insieme di valori $\{1, \ldots, W\}$, ma nel suo sottoinsieme $\{n + d_{min}, \ldots, n + d_{max}\}$.

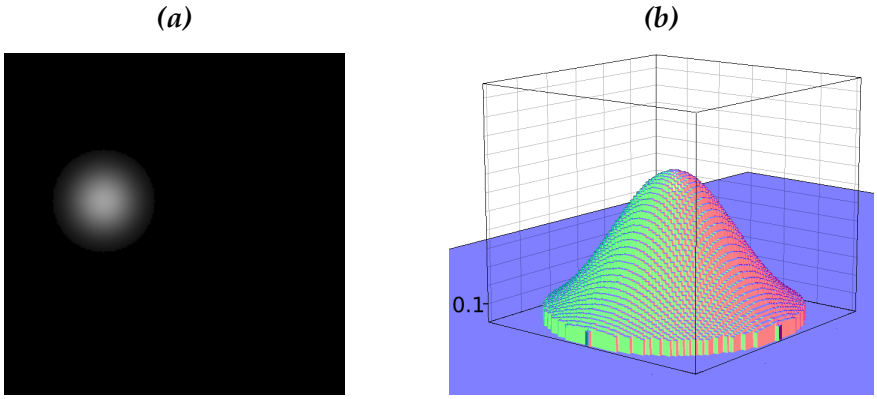
I dataset comprendenti immagini reali più utilizzati in letteratura, tra cui quelli impiegati per i test riportati nel Capitolo 5 nella pagina 75, forniscono i valori d_{min} e d_{max} insieme alla coppia di immagini stereo in forma standard.

Il concetto appena discusso è illustrato il Figura 4.3, dove viene applicato ad un semplice esempio.

4.2.3 Aggiornamento locale dei pesi

È ben noto che per le reti *SOM* gli aggiornamenti ai pesi neurali avvengono, in maggior misura, nelle vicinanze del neurone vincente, per questo motivo è possibile limitare l'aggiornamento dei pesi ad un'area locale al neurone eletto. Se questa soluzione porta a notevoli vantaggi in termini di velocità di esecuzione dell'algoritmo, la definizione di un area neurale di aggiornamento locale non è un'operazione banale. È necessario fare adeguate considerazioni a riguardo, in quanto tutti i neuroni esterni a questa area non saranno interessati dall'aggiornamento e questo

Figura 4.4: Ottimizzazione della funzione gaussiana $h(r,c)$ per $\varepsilon = 0.1$. ***(a)***: rappresentazione bidimensionale della funzione h ottimizzata, i valori puntuali della funzione sono rappresentati mediante valori di intensità luminosa. ***(b)***: rappresentazione tridimensionale della funzione h: l'aggiornamento neurale interesserà solo i neuroni nell'area racchiusa dal cubo.



dovrebbe accadere solo quando l'entità di questo aggiornamento è così ridotta da potersi definire trascurabile.

È utile ricordare che, inoltre, è possibile parallelizzare due o più processi di aggiornamento dei pesi se questi interessano aree neurali distinte e quindi indipendenti. Per una dettagliata trattazione su tecniche di ottimizzazione per reti *SOM* si veda [36].

Nel caso del modello *MSOM* si può definire l'area neurale di aggiornamento A come l'insieme delle coordinate di neuroni per cui la funzione gaussiana $h_k(r,c)$ non ha valori trascurabili. Formalmente

$$A = \{(r,c) : h_k(r,c) > \varepsilon\,,\ r \in \{1,2,\ldots,H\}\ ,\ c \in \{1,2,\ldots,W\}\} \tag{4.1}$$

dove ε è la soglia dei valori trascurabili, solitamente $\varepsilon \leq 0.1$.

Il dominio della funzione di aggiornamento dei pesi (3.2) può quindi essere ristretto $\forall\,(r,c) \in A$, con ovvi guadagni in termini di risorse computazionali richieste dall'algoritmo.

In Figura 4.4 è riportato un esempio di applicazione dell'ottimizzazione appena descritta.

4.2.4 Introduzione delle feature colore

L'ultima tappa nella specializzazione di *MSOM* è l'introduzione dei canali colore, in sostituzione della sola intensità luminosa rappresentata dal peso w_3. È noto nella letteratura sullo stereo matching come l'utilizzo dei colori, in sostituzione alla scala dei grigi, possa portare ad un incremento dei successi di ricerca delle corrispondenze dal 20% al 25% [42][3].

Siano K i canali colore forniti dalle stereocoppie, formalmente i dati da presentare in input alla rete *SOM* saranno vettori di input nella forma $\left(i_1^{mn}, i_{C_1}^{mn}, i_{C_2}^{mn}, \ldots, i_{C_K}^{mn}\right)$ e i pesi neurali saranno nella forma $w^{rc} = \left(w_1^{rc}, w_{C_1}^{rc}, w_{C_2}^{rc}, \ldots, w_{C_K}^{rc}\right)$.

La realizzazione del modello *StereoSOM* utilizza i canali "hardware" *RGB* per rappresentare i colori, si avrà quindi che $w_{C_1} = R, w_{C_2} = G, w_{C_3} = B$, con $K = 3$. Questa scelta tecnica è dovuta principalmente al fatto che *RGB* è direttamente supportato dal software di calcolo utilizzato per la rappresentazione delle immagini. Non sono stati eseguiti test approfonditi su altri modelli di colore, in quanto *RGB* ha offerto già un miglioramento apprezzabile dei risultati.

4.2.5 Algoritmo specializzato

Sono riassunte di seguito le modifiche apportate ai quattro step dell'algoritmo *MSOM*, descritte in 4.2.1, 4.2.2, 4.2.3 e 4.2.4.

STEP 1: INIZIALIZZAZIONE SOM. Ogni nodo della rete è rappresentato da quattro pesi $\left(w_1^{rc}, w_{C_1}^{rc}, w_{C_2}^{rc}, \ldots, w_{C_K}^{rc}\right)$, dove $w_1^{rc} = c$ e le altre componenti corrispondono ai canali colore dei pixel nell'immagine I_L, formalmente $w_{C_k}^{rc} = I_L\left(r, c, C_k\right)$. Analogamente i vettori di input presi dall'immagine di destra I_R sono nella forma $\left(i_1^{mn}, i_{C_1}^{mn}, i_{C_2}^{mn}, \ldots, i_{C_K}^{mn}\right)$ con $i_1^{mn} = n$ e $i_{C_k}^{mn} = I_L\left(r, c, C_k\right)$.
Si noti come non esista più alcun peso in rifermento alle coordinate verticali; questo perché il vincolo epipolare riduce la soluzione del problema alla sola mappa di disparità orizzontale.

STEP 2: ELEZIONE DEL NEURONE VINCENTE. Si estraggano casualmente gli interi $m \in \{1, \ldots, H\}$ e $n \in \{1, \ldots, W\}$. Sia quindi $\left(i_1^{mn}, i_{C_1}^{mn}, i_{C_2}^{mn}, \ldots, i_{C_K}^{mn}\right)$ l'input corrispondente al pixel dell'immagine $I_R\left(m, n, C_k\right)$. Le coordinate (φ_r, φ_c) del

[3] Si veda inoltre [43] per un metodo basato sul colore e su reti autorganizzanti.

neurone vincente si trovano nel modo seguente:

$$(\varphi_r, \varphi_c) = \left(m, \underset{c \in \{n+d_{min},\ldots,n+d_{max}\}}{\arg\min} \sqrt{\left(w_1^{mc} - i_1^{mn}\right)^2 + \sum_{k=1}^{K} \left(w_{C_k}^{mc} - i_{C_k}^{mn}\right)^2} \right) \quad (4.2)$$

STEP 3: AGGIORNAMENTO DEI PESI NEURALI. Come conseguenza dell'introduzione del vincolo epipolare, l'aggiornamento dei pesi riguarda soltanto la prima componente neurale w_1:

$$w_1^{rc} \leftarrow w_1^{rc} + h\,(r,c)\, g\,(r,c) \left(i_1^{r(c-(\varphi_c - n))} - w_1^{rc}\right) \quad (4.3)$$

dove

$$h\,(r,c) = \eta \exp\left(-\frac{(r-\varphi_r)^2 + (c-\varphi_c)^2}{2\sigma_h^2}\right) \quad (4.4)$$

$$g\,(r,c) = \exp\left(-\frac{\sum_{k=1}^{K}\left(w_{C_k}^{rc} - w_{C_k}^{\varphi_r \varphi_c}\right)^2}{2\sigma_g^2}\right) \quad (4.5)$$

$\forall\,(r,c) \in A,\ ,\ \forall\,(m,n) \in A$
$A = \{(r,c) : h_k\,(r,c) > \varepsilon\ ,\ r \in \{1,2,\ldots,H\}\ ,\ c \in \{1,2,\ldots,W\}\}.$

STEP 4: ESTRAZIONE DELLE DISPARITÀ. La mappa delle disparità orizzontali è facilmente estraibile dalla prima componente (w) dei pesi neurali

$$d^{hor}\,(r,c) = i_1^{rc} - w_1^{rc} = c - w_1^{rc} \quad (4.6)$$

ALGORITMO: La specializzazione di *MSOM* riguarda internamente i singoli step ma non tocca l'algoritmo complessivo. Il procedimento riportato in Algoritmo 3.1 nella pagina 46 resta perciò ancora valido.

4.3 Modello *StereoSOM*

StereoSOM introduce sostanziali modifiche rispetto al modello *MSOM*. Innanzitutto il metodo utilizza due reti autorganizzanti in modalità cooperativa per meglio gestire le aree occluse, in secondo luogo l'elezione del neurone vincente è affidata ad un più robusto sistema basato su finestra mobile, utilizzato nella maggior parte degli algoritmi di stereo matching locali.

Il modello *StereoSOM* è caratterizzato da un'architettura neurale simmetrica: è destinata una *SOM* per ognuna delle immagini che compongono la coppia stereo. Una eventuale futura estensione a sistemi *multi-view stereo* si tradurrebbe nell'utilizzo di *SOM* multiple.

La prima delle due *SOM* è detta di *reference* ed è la rete in cui si svolge l'elezione del neurone vincente e l'aggiornamento neurale, l'altra *SOM* è detta di *matching* e contiene i pesi neurali usati come input per la *reference SOM*. La *reference SOM* colleziona, in uno dei suoi pesi neurali, le corrispondenze con la *matching SOM* (da qui il nome *matching*).

Si riporta ora, in modo molto generale, la sequenza di operazioni dell'algoritmo *StereoSOM*:

1. Si creano due reti *SOM*, una per ogni immagine della coppia stereo.

2. Si stabilisce quale delle due *SOM* assume il ruolo di *reference SOM* e quale di *matching SOM*.

3. Si estrae un neurone dalla *matching SOM* e si usano i suoi pesi come input per la *reference SOM.*

4. Nella *reference SOM* si elegge il neurone vincente, stabilendo così la corrispondenza.

5. Sempre nella reference SOM si esegue l'aggiornamento dei pesi neurali, "propagando" la corrispondenza/disparità trovata al passo 5 ai neuroni nel vicinato dell'eletto.

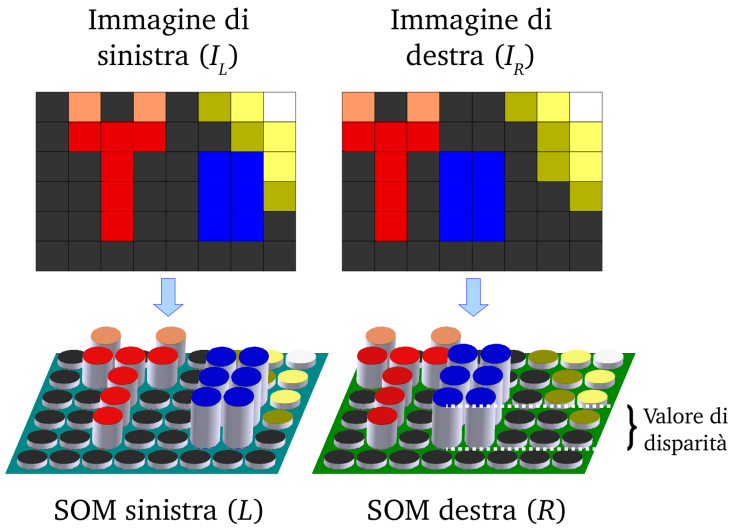
6. Si invertono i ruoli di *reference SOM* e *matching SOM* e si ripetono i passi dal 3 al 5.

7. Si ripetono i passi dal 2 al 6 un numero di volte pari al numero di iterazioni.

Il ruolo di *reference SOM* o *matching SOM* non è quindi vincolato ad una immagine in particolare ma "ruota" in funzione dei passi dell'algoritmo. Risulta evidente che, alla fine del processo, saranno disponibili due mappe di disparità: una per ogni rete *SOM*.

Vediamo ora la definizione formale dei quattro step principali che compongono l'algoritmo *StereoSOM*.

Figura 4.5: Semplice esempio che evidenzia la corrispondenza tra i pixel della coppia di immagini stereo e i neuroni artificiali delle reti *SOM*.



4.3.1 Step 1: inizializzazione delle SOM

Sia data la coppia di immagini stereo I_L e I_R, con dimensioni $H \times W$ pixel e K canali.

Si creano due *SOM* (matrici di neuroni) $L_{rc} = \left[w_1^{rc}, w_{C_1}^{rc}, \ldots, w_{C_K}^{rc}\right]$ e $R_{rc} = \left[v_1^{rc}, v_{C_1}^{rc}, \ldots, v_{C_K}^{rc}\right]$, dove $r = 1 \ldots H$ e $c = 1 \ldots W$. I neuroni delle due *SOM* si inizializzano nel modo seguente:

$$\text{per } L_{rc} \qquad w_1^{rc} = c,\ w_{C_k}^{rc} = I_L\left(r, c, C_k\right),\ \forall r \text{ e } \forall c$$

$$\text{per } R_{rc} \qquad v_1^{rc} = c,\ v_{C_k}^{rc} = I_R\left(r, c, C_k\right),\ \forall r \text{ e } \forall c$$

Il primo peso neurale w_1^{rc} e v_1^{rc} rappresenterà, esattamente come avviene per *MSOM*, le corrispondenze trovate dall'algoritmo. In Figura 4.5 è riportato un esempio di inizializzazione di *StereoSOM* con la coppia stereo di Figura 4.1; notare come è stata rappresentata la prima componente dei pesi neurali. In *StereoSOM* infatti le corrispondenze si trovano, grazie al vincolo epipolare, solo lungo una coordinata: questo permette di rappresentare la prima componente dei pesi neurali come un valore di disparità, proporzionale all'altezza del cilindretto associato ad ogni neurone.

4.3.2 Step 2: elezione del neurone vincente

Ricerca ponderata

Dall'equazione (4.2) è evidente come, ai fini della ricerca del neurone vincente, vengano considerati con uguale rilevanza tutti i pesi neurali. Si è visto[4] come, moltiplicando per opportuni pesi i termini in (4.2), è possibile migliorare il processo di ricerca. Il principale termine da limitare è w_1, componente dei pesi neurali che garantisce la continuità del valore di disparità (*fattore di smoothing*) all'interno di aree con colore omogeneo. Di seguito è riportata l'equazione di ricerca del neurone vincente con i termini opportunemente ponderati:

$$(\varphi_r, \varphi_c) = \left(m, \underset{c \in \{n+d_{min},\ldots,n+d_{max}\}}{\arg\min} \sqrt{\rho_1 \left(w_1^{mc} - i_1^{mn}\right)^2 + \sum_{k=1}^{K} \rho_{C_k} \left(w_{C_k}^{mc} - i_{C_k}^{mn}\right)^2} \right) \quad (4.7)$$

dove una possibile configurazione di pesi è la seguente: $\rho_1 = 1$, $\rho_{C_k} = 20$, $\forall k$ (o analogamente $\rho_1 = 0.05$ e $\rho_{C_k} = 1$, $\forall k$).

Supporto di ricerca adattivo

L'elezione del neurone vincente di *MSOM* è una delle principali cause di errore, infatti, fatta eccezione per il termine w_1 che garantisce indirettamente una certa omogeneità dei valori di disparità, l'approccio pixel-based non tiene conto dei punti in prossimità di quello da ricercare.

Sono state proposte in letteratura molte tecniche che vanno a selezionare per ogni pixel dell'immagine una finestra adattiva che rappresenti al meglio i pixel con valori di disparità uguali a quello del punto centrale. Si rimanda per esempio a [44, 45, 46, 47, 48, 49] che propongono metodi simili; in [50] e [51] è riportata una analisi più completa della letteratura sui metodi a finestra adattiva.

Le tecniche citate in precedenza, pur essendo ad approccio locale, offrono risultati competitivi; è stato questo il principale motivo che ha spinto all'utilizzo di una tecnica a finestra adattiva nel modello *StereoSOM.*

Per giustificare il meccanismo di aggiornamento dei pesi neurali, in (3.2.3) è stata fatta l'assunzione che, all'interno di una immagine, una piccola zona circoscritta di pixel con colori simili corrispondano con alta probabilità ad una piccola zona di

[4] Per un riscontro vedere i test quantitativi nel Capitolo 5.

un oggetto con disparità pressoché costante. Lo stesso ragionamento è applicabile all'elezione del neurone vincente.

Utilizzando per la ricerca delle corrispondenze una piccola finestra mobile è facile accorgersi che, per meglio gestire le zone di discontinuità di profondità, i pixel contenuti nella finestra dovrebbero corrispondere a punti con disparità costante. Utilizzando l'assunzione descritta prima si può costruire una funzione che assegna ad ogni pixel della finestra di ricerca una certa rilevanza, direttamente proporzionale alla similarità di colore con il pixel al centro della finestra (il pixel per cui si vuole trovare la corrispondenza).

La funzione appena descritta è integrabile all'interno dell'equazione (4.7) nel modo seguente:

$$(\varphi_r, \varphi_c) = \left(m, \underset{c \in \{n+d_{min},\ldots,n+d_{max}\}}{\arg\min} \sum_{\Delta r=-\xi}^{\xi} \sum_{\Delta c=-\xi}^{\xi} \sqrt{\rho_1 \left(w_1^{mc} - v_1^{mn}\right)^2 + \rho_{C_k} S(c, \Delta r, \Delta c)} \right) \tag{4.8}$$

con

$$S(c, \Delta r, \Delta c) = \sum_{k=1}^{K} \left[g_s\left(\Delta r, \Delta c\right) \left(w_{C_k}^{(m+\Delta r)(c+\Delta c)} - v_{C_k}^{(m+\Delta r)(n+\Delta c)} \right)^2 \right] \tag{4.9}$$

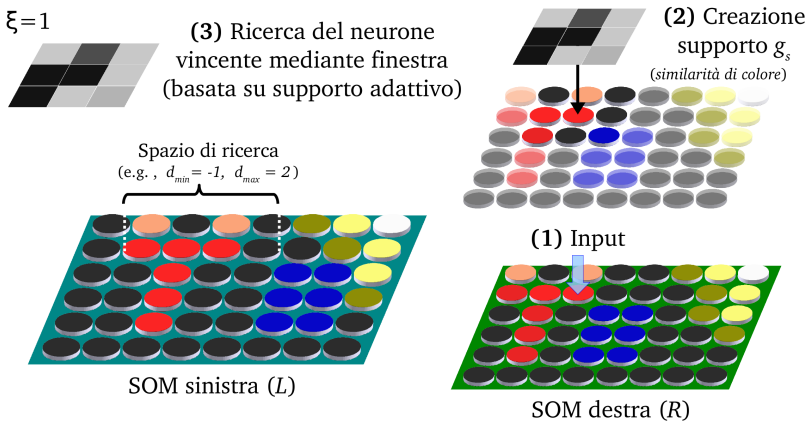
e

$$g_s\left(r, c\right) = \exp\left(- \frac{\sum_{k=1}^{K} \left(v_{C_k}^{(m+r)(n+c)} - v_{C_k}^{mn} \right)^2}{2\sigma_s^2} \right) \tag{4.10}$$

La finestra di ricerca mobile è sempre centrata sul neurone della *reference SOM* (in questo caso la *SOM* di *sinistra L*) con coordinate (m, c), ha dimensioni $(2\xi + 1)$ e viene confrontata con una finestra di uguali dimensioni, centrata sul neurone (m, n) della *Matching SOM* (in questo caso la *SOM* di destra *R*). La funzione g_S, molto simile alla (4.5), definisce un supporto adattivo per la finestra di ricerca, attribuendo ad ogni pixel di questa un peso dipendente dalla similarità di colore con il pixel (m, n) della *Matching SOM*.

Un esempio di funzionamento della funzione g_S per una finestra di dimensioni $3x3$ ($\xi = 1$) si può trovare in Figura 4.6 nella pagina successiva, dove viene descritto graficamente il processo di elezione del neurone vincente di *StereoSOM*.

Figura 4.6: In questo esempio è riassunto il processo di elezione del neurone vincente di *StereoSOM*. Si noti come la creazione del supporto adattivo avvenga interamente nella *Matching SOM*.



Ricerca invariante rispetto a distorsioni di tipo offset

Come riportato in [52], una vantaggiosa trasformazione applicabile alle funzioni di matching per aumentarne la robustezza in presenza di distorsioni fotometriche consiste nel sottrarre a ciascun pixel il valore medio calcolato sulla finestra mobile. Con questo metodo, chiamato *zero-mean*, si ottiene la compensazione di un noto tipo di distorsione fotometrica detto *offset* e schematizzato di seguito:

$$I' = I + \beta$$

dove I rappresenta l'immagine originale (per semplicità si consideri una immagine a singolo canale), I' l'immagine dotata di offset e β il valore dell'offset.

È semplice integrare il metodo *zero-mean* nell'equazione 4.9:

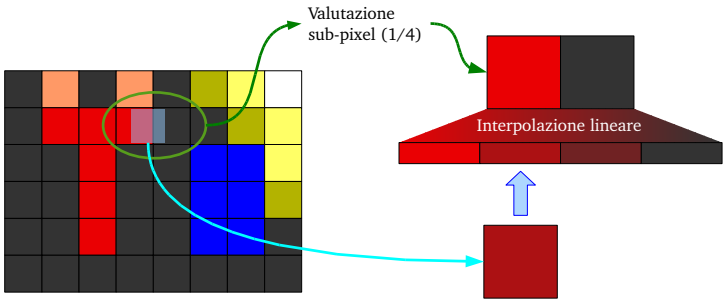
$$S(c, \Delta r, \Delta c) =$$

$$\sum_{k=1}^{K} \left[g_s(\Delta r, \Delta c) \left(\left(w_{C_k}^{(m+\Delta r)(c+\Delta c)} - \mu_L(m, c, k) \right) - \left(v_{C_k}^{(m+\Delta r)(n+\Delta c)} - \mu_R(m, n, k) \right) \right)^2 \right] \tag{4.11}$$

dove

$$\mu_R(r, c, k) = \frac{\sum_{\Delta r=-\xi}^{\xi} \sum_{\Delta c=-\xi}^{\xi} \left(g_s(r + \Delta r, c + \Delta c)\, w_{C_k}^{(r+\Delta r)(c+\Delta c)} \right)}{\sum_{\Delta r=-\xi}^{\xi} \sum_{\Delta c=-\xi}^{\xi} \left(g_s(r + \Delta r, c + \Delta c) \right)} \tag{4.12}$$

Figura 4.7: Esempio di interpolazione lineare finalizzata alla valutazione sub-pixel.



e μ_L si trova in modo analogo, prestando attenzione ad applicare anche la funzione $g_s(r, c)$ alla *SOM* di sinistra.

Si noti che le funzioni μ_R e μ_L non eseguono delle semplici medie aritmetiche sui pixel della finestra mobile, ma delle medie ponderate utilizzando come pesi i valori del supporto variabile definiti dalla funzione g_s.

Si vedrà nel Capitolo 5 come l'utilizzo del nuovo metodo di matching *"zero-mean"*, fornisca migliori risultati in immagini affette da trasformazioni di tipo offset.

Valutazione sub-pixel

In *StereoSOM* è stato implementato un semplice metodo di valutazione *sub-pixel*, in grado di estendere la ricerca delle corrispondenze a valori di disparità reali. La tecnica, ottimizzata in modo da non appesantire computazionalmente l'intero algoritmo, si basa su due fasi:

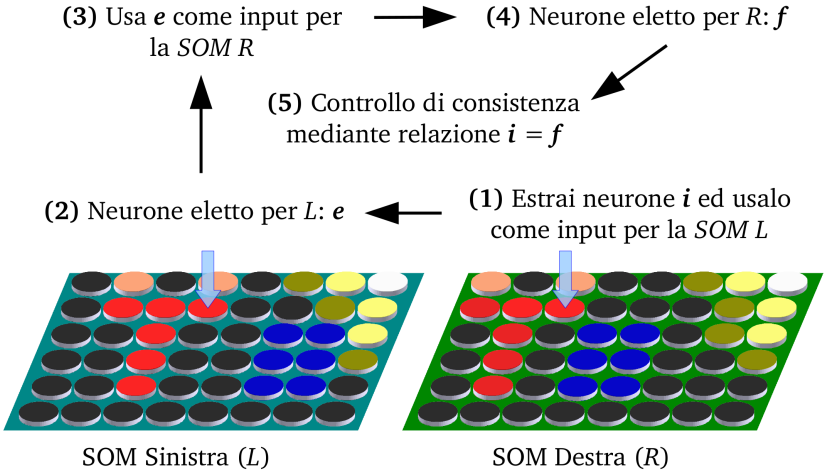
1. Si cerca la corrispondenza a valori di disparità interi, utilizzando la 4.11.

2. Si prosegue la ricerca valutando frazioni di pixel in un intorno del punto trovato al *passo 1*.

Solitamente la valutazione sub-pixel avviene per frazioni di pixel pari a $1/4$, ma può essere estesa anche a frazioni più piccole. Il metodo utilizzato per valutare le frazioni di pixel è l'interpolazione lineare [53].

In Figura 4.7 è illustrato il funzionamento dell'algoritmo appena descritto.

Figura 4.8: Funzionamento del controllo cooperativo di consistenza stereo.



Controllo cooperativo di consistenza stereo

Effettuando analisi specifiche[5] si è visto come il modello *MSOM* commetta un grande numero di errori di corrispondenza in prossimità delle zone di occlusione, ossia le zone i cui pixel, per ovvi motivi legati alla geometria del sistema stereoscopico, compaiono in una sola delle due immagini.

Una soluzione a questo inconveniente è stata trovata nell'uso di un criterio che possa stabilire se il risultato della ricerca del neurone vincente sia o no potenzialmente errato. La tecnica, descritta dettagliatamente in 2.2.4 nella pagina 30, si basa su un controllo di consistenza esteso su entrambe le mappe di disparità e in grado di rilevare potenziali occlusioni o errori di corrispondenza. Nel caso in cui l'algoritmo rilevi una potenziale occlusione o errore di matching, la gestione dell'errore consiste nell'annullamento dell'aggiornamento neurale, proseguendo con l'iterazione successiva.

In Figura 4.8 è illustrato graficamente il funzionamento del controllo di consistenza applicato all'algoritmo *StereoSOM*. È fondamentalmente da questo procedimento che dipende la natura cooperativa del metodo neurale, è infatti evidente come le due reti eleggano autonomamente il neurone vincente, collaborando poi per stabilire se i risultati ottenuti siano coerenti o meno.

[5] Vedere Capitolo 5.

4.3.3 Step 3: aggiornamento dei pesi neurali

La modalità di aggiornamento dei pesi neurali è la stessa definita in 4.2.5-*Step 3*. Cambia invece la funzione h, responsabile dell'entità dell'aggiornamento nel vicinato del neurone vincente, che è stata opportunamente estesa per integrare il *vincolo di continuità* descritto in 1.1.

Ridefinizione dell'area di apprendimento

È stato spiegato nel Capitolo 1 come la disparità tenda a distribuirsi in modo uniforme nelle mappe di disparità, con l'eccezione delle zone di bordo degli oggetti. Questa assunzione deriva dal fatto che le superfici degli oggetti stessi presentano solitamente poche discontinuità di profondità.

La funzione h utilizzata nel modello *MSOM* è a campana gaussiana, una funzione di vicinato locale tipicamente impiegata nelle regole di apprendimento dei modelli *SOM*. Ma il metodo da noi sviluppato si occupa di stereo matching, non di clustering dei dati, per questo è necessario progettare la fase di apprendimento tenendo conto dei vincoli utilizzati negli algoritmi di corrispondenza stereo. Seguendo le linee guida del vincolo di continuità, l'area di neuroni vicini a quello eletto dovrebbe ottenere un valore di disparità pressoché costante, per questo motivo è stata modificata la funzione h in modo da creare una sorta di "altipiano" sopra la collina formata dalla campana gaussiana.

Formalmente, si definisce la funzione h in questo modo:

$$h(r,c) = \begin{cases} \theta(r,c) & se\ \beta < \theta(r,c) < 1 \\ 1 & se\ \theta(r,c) \geq 1 \\ 0 & se\ \theta(r,c) \leq \beta \end{cases} \tag{4.13}$$

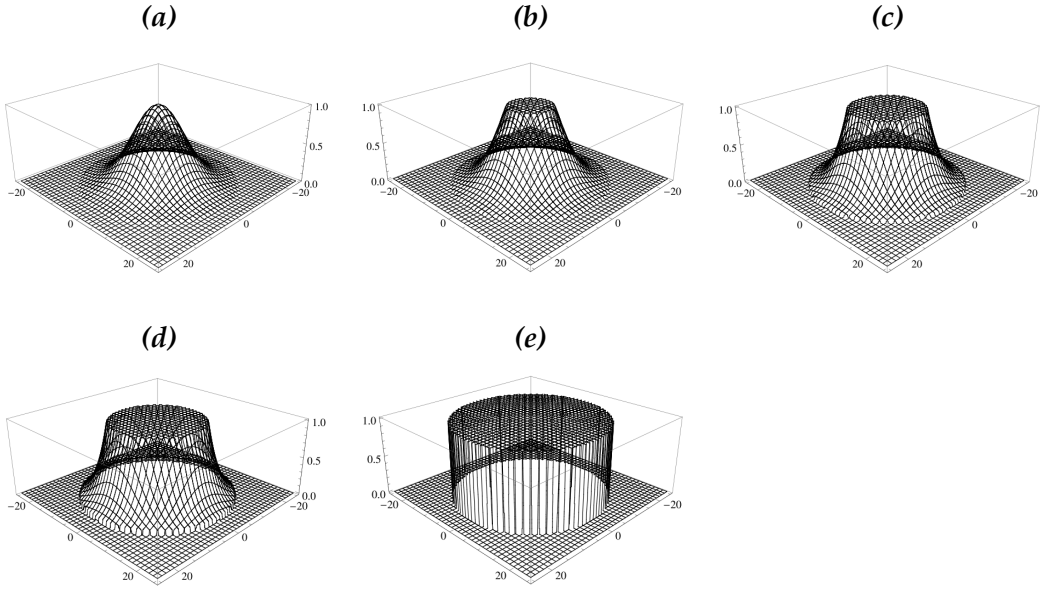
dove

$$\theta(r,c) = \alpha \exp\left(-\frac{(r-\varphi_r)^2 + (c-\varphi_c)^2}{2\sigma_h^2} \right) \tag{4.14}$$

e

$$\sigma_h^2 = \frac{n_{size}^2}{-2\ln\left(\frac{\beta}{\alpha}\right)} \tag{4.15}$$

Figura 4.9: Plot della funzione h al variare dei parametri α, β, per $n_{size} = 20$. ***(a)***: [$\alpha = 1.0$, $\beta = 0.010$] ***(b)***: [$\alpha = 1.5$, $\beta = 0.015$] ***(c)***: [$\alpha = 3.0$, $\beta = 0.030$] ***(d)***: [$\alpha = 6.0$, $\beta = 0.060$] ***(e)***: [$\alpha = 1.0$, $\beta = 1.000$]



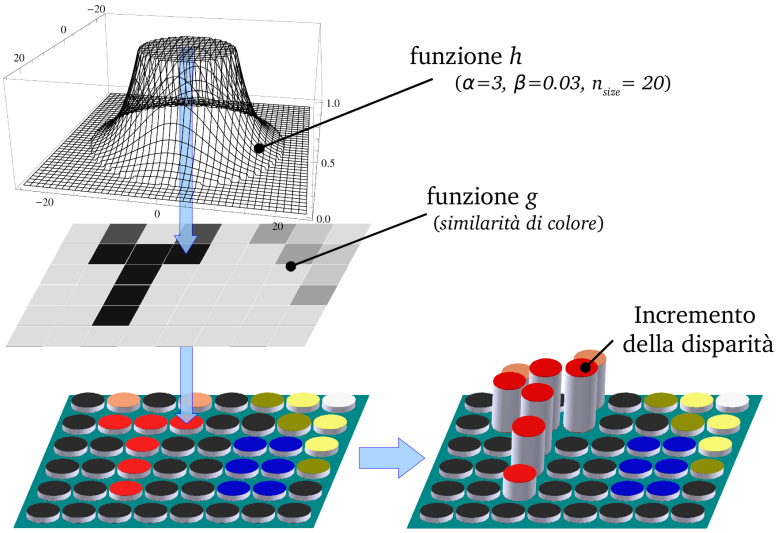
La funzione descritta è una funzione gaussiana le cui code si "arrestano", per $\sqrt{r^2 + c^2} = n_{size}$, ad un valore pari a β e la cui ampiezza è limitata ad un valore massimo pari ad 1. Per valori di α maggiori di 1, la funzione presenta il classico profilo ad "altipiano" discusso in precedenza, questo si tradurrà in un aggiornamento dei pesi neurali con entità costante in un piccolo vicinato del neurone vincente.

Un ulteriore vantaggio di questa definizione è la possibilità di definire esplicitamente le dimensioni del vicinato (n_{size}) da aggiornare.

Si noti come il learning rate η nella funzione 4.4 sia stato rimosso, il parametro σ_h^2 non sia più definibile e siano stati aggiunti tre nuovi parametri α, β e n_{size}.

In Figura 4.9 sono riportati alcuni plot della funzione h, al variare di α, β e per $n_{size} = 20$, mentre in Figura 4.10 è riportato un esempio che mostra graficamente il processo di aggiornamento neurale di *StereoSOM*; da notare come nelle zone dove il vincolo di continuità non vale più (bordi degli oggetti), la funzione g intervenga, limitando l'aggiornamento e garantendo buoni risultati anche nelle zone di discontinuità di profondità.

Figura 4.10: Rappresentazione grafica del processo di aggiornamento neurale in *StereoSOM*.



4.3.4 Step 4: estrazione delle mappe di disparità

Il processo di estrazione delle mappe di disparità dalle reti *SOM* segue il principio impiegato nel modello *MSOM*, con la differenza che, in questo caso, si può disporre di due mappe di disparità, riferite ad entrambe le immagini nella coppia stereo. A causa dell'introduzione del vincolo epipolare si perde ovviamente la mappa delle disparità verticali.

Formalmente:

$$\begin{cases} d_L(r,c) = c - w_1^{rc} \\ d_R(r,c) = v_1^{rc} - c \end{cases} \tag{4.16}$$

4.3.5 Algoritmo StereoSOM

Il metodo *StereoSOM* descritto in questo capitolo è riassunto nell'Algoritmo 4.1 nella pagina successiva.

4.4 Configurazione e ottimizzazione

Saranno ora elencati e descritti i parametri di esecuzione dell'algoritmo, ottenuti dopo una fase di "messa a punto" caratterizzata da test e valutazione degli errori.

Algoritmo 4.1 Pseudo codice dell'algoritmo *StereoSOM*

Input: una coppia di immagini stereo I_L e I_R, con dimensioni $H \times W$ e K canali;

Output: le corrispondenti mappe di disparità (disparità orizzontali) d_L e d_R;

Require: creare due reti *SOM* (matrici di neuroni) $L_{rc} = \left[w_1^{rc}, w_{C_1}^{rc}, \ldots, w_{C_K}^{rc}\right]$ e $R_{rc} = \left[v_1^{rc}, v_{C_1}^{rc}, \ldots, v_{C_K}^{rc}\right]$ con $r = 1 \ldots H$ e $c = 1 \ldots W$;

Require: $\forall r$ e $\forall c$ inizializzare i neuroni per entrambe le *SOM* $L_{rc} : w_1^{rc} = c$, $w_{C_k}^{rc} = I_L(r, c, C_k)$ e $R_{rc} : v_1^{rc} = c$, $v_{C_k}^{rc} = I_R(r, c, C_k)$;

1: selezionare la *reference SOM*, in questo esempio L, e la corrispondente *matching SOM*, R;

2: **for** $i = 1$ to *Iterations* **do**

3: estrarre casualmente un neurone R_{mn} dalla *matching SOM*;

4: usare i pesi di $R_{mn} : \left(v_1^{mn}, v_{C_1}^{mn}, \ldots, v_{C_K}^{mn}\right)$ come input per la *reference SOM*;

5: calcolare le coordinate (φ_r, φ_c) del neurone vincente secondo l'equazione 4.8 (ricordandosi di impiegare le estensioni descritte in 4.3.2);

6: **ripetere** i passi da *3* a *5* usando R come *reference SOM*, L come *matching SOM* e L_{φ_r, φ_c} come input per la *reference SOM*; (Siano quindi (φ'_r, φ'_c) le nuove coordinate del neurone vincente in R)

7: **if** $(m, n) = (\varphi'_r, \varphi'_c)$ **then**

nella *reference SOM* L, aggiornare i pesi dei neuroni in un'area di dimensioni $(2n_{size} + 1)^2$ intorno al neurone vincente come segue:

8: **for** $r = \varphi_r - n_{size}$ to $\varphi_r + n_{size}$ e $c = \varphi_c - n_{size}$ to $\varphi_c + n_{size}$ **do**

9: apprendimento secondo l'equazione 4.3 utilizzando come funzione h la 4.13;

10: **end for**

11: **ripetere** i passi di aggiornamento (da *8* a *10*) questa volta sulla *SOM* R, usando (φ'_r, φ'_c) come neurone vincente;

12: **else**

13: non eseguire l'aggiornamento neurale nelle due reti *SOM*;

14: **end if**

15: **end for**

16: estrarre le mappe di disparità $d_L(r, c)$ e $d_R(r, c)$ secondo le equazioni 4.16;

Al fine di ridurre i tempi di elaborazione, l'algoritmo *StereoSOM* prevede due fasi di esecuzione, dette rispettivamente di *ordering* e di *tuning* e caratterizzate da una diversa configurazione dei parametri.

La fase di tuning è il momento in cui l'algoritmo ottiene dei risultati validi ed utilizzabili e potrebbe essere eseguita singolarmente. L'unico problema è che la fase di *tuning* applica aggiornamenti di piccola entità ai valori dei pesi neurali: supponendo di partire con la computazione da una mappa di pesi nulli, sarebbero richieste troppe iterazioni prima di ottenere dei risultati ragionevoli. La fase di *ordering* serve proprio a risolvere questo inconveniente e portare velocemente i pesi neurali a valori grossolani ma pur sempre non nulli.

L'esecuzione dell'algoritmo sarà quindi caratterizzata da una prima fase di *ordering*, seguita poi dalla fase di *tuning* che perfezionerà il primo calcolo mediante aggiornamenti di piccola entità.

Saranno ora elencati i parametri utilizzati nell'implementazione dell'algoritmo *StereoSOM*.

Parametri generali del dataset

I parametri del dataset, oltre al nome e alle informazioni per recuperare i file necessari, sono:

minDisp valore minimo di disparità per la stereocoppia;

maxDisp valore massimo di disparità per la stereocoppia;

normFactor questo fattore di normalizzazione sarà moltiplicato per i valori di disparità nella mappa di disparità ottenuta come risultato. Lo scopo è di convertire la mappa di disparità (con valori di intensità nell'intervallo $[minDisp, maxDisp]$) in una immagine in scala di grigi (con valori di intensità nell'intervallo $[0, 255]$).

Questi parametri non sono "modificabili" e dipendono dal processo di calibrazione (1.2 nella pagina 8) effettuato durante la creazione del dataset.

Parametri generali dell'algoritmo

In questa sezione sono elencati i valori tipici dei parametri usati in *StereoSOM*. Questi valori valgono per stereocoppie paragonabili, per dimensioni e parametri di calibrazione, a quelle dei dataset Middlebury [41].

Questi parametri sono costanti nelle due fasi di *ordering* e *tuning*.

"*searchEye*" (ξ) valore del "raggio" in pixel del supporto adattivo. Il supporto utilizzato in *StereoSOM* è una finestra quadrata con lato pari a $2\xi + 1$ pixel.
Il valore tipico per questo parametro è $\xi = 3$.
Incrementare ξ significa aumentare le dimensioni del supporto: una finestra troppo grande, sebbene abbia caratteristiche adattive, incrementerà comunque gli errori in presenza di discontinuità di profondità, una finestra troppo piccola ridurrà invece la quantità di informazioni da confrontare aumentando il numero di corrispondenze errate o ambigue.

"*oSearch*" (σ_s^2) valore di varianza usato nel supporto variabile.
Tipicamente $\sigma_s^2 = 300$.
Incrementare questo valore significa ridurre l'adattività del supporto; per assurdo un valore $\sigma_s^2 = \infty$ annullerebbe completamente l'adattività del supporto, rendendolo una normale finestra di correlazione.

"*searchMethod*" parametro che specifica se attivare il controllo cooperativo di consistenza stereo o meno.
$searchMethod = 1$: utilizzo del controllo di consistenza (consigliato).
$searchMethod = 0$: controllo di consistenza disabilitato.

"*matchingFunction*" seleziona la funzione di corrispondenza impiegata nell'elezione del neurone vincente.
$matchingFunction = 1$: utilizzo della funzione con compensazione dell'offset descritta in 4.3.2 nella pagina 61 (consigliato).
$matchingFunction = 0$: utilizzo della funzione originale.

"*subPixGran*" parametro che specifica la precisione fornita dalla valutazione *subpixel* (vedere 4.3.2 nella pagina 62).
Il valore impiegato tipicamente è $subPixGran = \frac{1}{4} = 0.25$.
Se non si è interessati ad ottenere mappe di disparità a valori reali è consigliabile impostare questo parametro all'unità.

Parametri nelle fasi di ordering e tuning

I seguenti parametri dipendono dalla particolare fase dell'algoritmo.

"*enGTerm*" Abilitazione della funzione g (Equazione 4.5 nella pagina 56).

ORDERING $enGTerm = 0$: funzione g disabilitata ($g_s = 1$). Gli aggiornamenti ai pesi neurali sono controllati dalla sola funzione h.

TUNING $enGTerm = 1$: funzione g abilitata.

"*ogk*" (σ_g^2) Con la funzione g abilitata, questo parametro corrisponde alla varianza σ_g^2.

ORDERING (la funzione g è disabilitata: questo parametro non è impostato).

TUNING valore tipico $\sigma_g^2 = 80$.

"*dispWeight*" / "*colorWeight*" (ρ_1 / ρ_{C_k}) Peso delle feature che concorrono nella formula di ricerca delle corrispondenze (Equazione 4.8 nella pagina 60).
E' molto importante che nella fase di ordering il valore di ρ_{C_k} sia molto più grande di ρ_1: questo, oltre all'importante funzione di velocizzare la convergenza, evita gravi errori legati all'affidabilità del fattore di smoothing ρ_1 e causati dalla disattivazione della funzione g.

ORDERING valori tipici $\rho_1 = 1$ / $\rho_{C_k} = 1000$.

TUNING valori tipici $\rho_1 = 1$ / $\rho_{C_k} = 20$.

"*neighSize*" (n_{size}) valore del "raggio" in pixel dell'area di aggiornamento dei pesi (Equazione 4.9 nella pagina 65). L'area interessata dall'aggiornamento sarà una finestra quadrata con lato pari a $2n_{size} + 1$ neuroni, dove il neurone centrale è quello eletto vincitore.

ORDERING n_{size} viene fatto variare, linearmente con le iterazioni eseguite, da un valore iniziale $n_{size,i} = 80$ ad un valore finale $n_{size,f} = 10$.

TUNING valore tipico $n_{size} = 20$.

"*alpha*" (α) valore massimo della funzione di vicinato h (Equazione 4.9 nella pagina 65)

ORDERING valore tipico $\alpha = 1$.

TUNING α viene fatto variare, linearmente con le iterazioni eseguite, da un valore iniziale $\alpha_i = 6$ ad un valore finale $\alpha_f = 1$.

"*gaussBorder*" (β) Valore minimo della funzione di vicinato h, rilevabile nelle code della gaussiana (Equazione 4.9 nella pagina 65)

ORDERING valore tipico $\beta = 1$.

TUNING β viene fatto variare, linearmente con le iterazioni eseguite, da un valore iniziale $\beta_i = \frac{\alpha_i}{100} = 0.06$ ad un valore finale $\beta_f = \frac{\alpha_f}{100} = 0.01$. Si avranno quindi delle code di gaussiana che si "arrestano" ad un centesimo del valore massimo della funzione.

"*iterations*" Numero totale di iterazioni

ORDERING Il numero tipico di iterazioni nella fase di ordering è $\frac{H \times W}{15}$, ossia un quindicesimo del numero totale di pixel.

TUNING La fase di tuning richiede tipicamente un numero di iterazioni pari a $\frac{H \times W}{2}$. Notare come, grazie all'introduzione della fase di ordering, sia stato possibile ridurre di un fattore 200 il numero di iterazioni rispetto al modello *MSOM* (Vedere 3.2.5 nella pagina 46). Aumentare ulteriormente il numero di iterazioni può portare ad un incremento molto blando della qualità dei risultati, non giustificato comunque dalla crescita dei tempi di esecuzione.

4.5 Complessità computazionale

StereoSOM è una estensione del modello *MSOM*, per questo motivo ne eredita direttamente la complessità computazionale. Saranno ora analizzate le differenze nella complessità dovute alle estensioni introdotte con *StereoSOM*. Come è stato fatto per il modello *MSOM* nel Capitolo 3, considereremo la sola complessità in tempo.

Le estensioni al modello *MSOM* che influiscono maggiormente sulla complessità computazionale sono:

- *Introduzione del supporto di ricerca adattivo* (aumento della complessità).
- *Utilizzo del vincolo epipolare* (riduzione della complessità).
- *Utilizzo dell'intervallo di disparità* (riduzione della complessità).
- *Aggiornamento neurale locale* (riduzione della complessità).

StereoSOM gode appieno dei vantaggi legati all'introduzione del vincolo epipolare: per un immagine di dimensioni $H \times W$ lo spazio di ricerca di riduce di un fattore H.

Dalla Figura 4.3 nella pagina 53 potrebbe forse sembrare che l'introduzione del vincolo di disparità minima e massima non porti ad una sostanziale riduzione dello spazio di ricerca. In realtà le immagini comunemente impiegate hanno larghezze di $300 - 500$ pixel, con una disparità minima/massima di $20 - 50$ pixel; si può ridurre quindi lo spazio di ricerca di oltre 10 volte, un fattore non di poco conto.

Formalmente, si avrà un tempo di ricerca del neurone vincente pari a:

$$t_{ricerca}(n) = (d_{max} - d_{min}) * (2\xi + 1)^2 \tag{4.17}$$

dove d_{min} e d_{max} rappresentano l'intervallo di disparità e ξ il "raggio" in pixel del supporto variabile.

Si noti come, utilizzando l'ottimizzazione proposta per la valutazione sub-pixel, il tempo richiesto per la ricerca del neurone vincente viene incrementato solo di un termine costante f, dove $1/f$ è la frazione di pixel impiegata. Senza l'ottimizzazione proposta, il tempo di ricerca verrebbe moltiplicato per f (utilizzando una valutazione a $1/4$ di pixel si quadruplicherebbero i tempi richiesti dalla fase di ricerca delle corrispondenze).

Infine, grazie all'ottimizzazione descritta in 4.2.3 nella pagina 53 e per come è stata ridefinita l'area di apprendimento con la funzione h, la complessità nella fase di modifica dei pesi neurali risulta dipendente dal solo parametro n_{size}. In particolare saranno interessati dall'aggiornamento soltanto $(2n_{size} + 1)^2$ neuroni.

Considerando la dimensione dell'input $n = H \times W$ e il numero di iterazioni pari a circa $\frac{H \times W}{2} = \frac{n}{2}$, la complessità computazionale complessiva dell'algoritmo *StereoSOM* risulta essere:

$$t_{tot}(n) = \frac{n}{2} \cdot \left(t_{ricerca}(n) + (2n_{size} + 1)^2 \right) \tag{4.18}$$

Se si considerano i parametri d_{max}, d_{min}, ξ e n_{size} indipendenti da n, il costo asintotico dell'algoritmo *StereoSOM* risulta lineare nella dimensione dell'input ($O(n)$). In realtà, per come sono definiti i sistemi stereoscopici reali, vi è spesso una relazione di dipendenza tra le dimensioni delle coppie stereo (n) e l'intervallo di disparità ($d_{max} - d_{min}$).

Per ottimizzare il tempo di calcolo delle funzioni esponenziali (come h o le funzioni di similarità di colore g e g_s) è stata impiegata una versione approssimata della funzione $\exp(x)$ documentata in [54]; da alcune analisi appositamente condotte è emerso come questa ottimizzazione riesca a ridurre notevolmente i tempi di esecuzione senza influire significativamente sulla qualità dei risultati.

Dal punto di vista delle "performance reali" ottenute su dataset di medie dimensioni, su un sistema laptop *Intel* 2.53Ghz, 4$GB\ RAM$ e S.O. *GNU/Linux* (*kernel 2.6.28*) si sono registrati tempi di esecuzione dell'ordine di *100 secondi*. Il framework utilizzato per l'implementazione è *GNU Octave* [56], con parte del codice realizzata in *C/C++* sfruttando la tecnologia *Octave Dynamically Linked Functions.*

È da tenere in considerazione per sviluppi futuri dell'algoritmo una eventuale ottimizzazione multithread, in grado di sfruttare più core nelle moderne CPU multicore. Altre ottimizzazioni attuabili sono l'impiego di istruzioni vettoriali di tipo *SIMD* o acceleratori grafici di ultima generazione (*GPGPU - General Purpose computation using GPU*).

4.6 Analisi comparativa dei modelli studiati

Il modello iniziale *MSOM* offre sicuramente importanti vantaggi rispetto ad altre soluzioni di stereo matching, il primo tra tutti è la totale assenza di fasi di *pre* e *post-processing* o *estrazione delle feature*: le immagini in input vengono mappate direttamente sui pesi neurali e l'elaborazione può partire immediatamente. Inoltre la fase finale di estrazione della mappa di disparità è un processo computazionalmente molto semplice, con un tempo di esecuzione lineare rispetto alle dimensioni delle immagini.

È immediato dedurre che il modello ben si presta anche in contesti *real-time*, dove il principio fondamentale è la puntualità temporale dei risultati, è infatti possibile interrompere la fase di apprendimento non supervisionato in ogni momento ed estrarre una mappa di disparità densa senza dover applicare alcun processo di interpolazione.

Altra caratteristica interessante è l'assenza di una fase esplicita di segmentazione [6], molti algoritmi di stereo matching sono infatti fortemente dipendenti, sia

[6] Per essere precisi, la funzione g segue implicitamente e localmente un semplice principio usato per segmentare le immagini basandosi sull'intensità luminosa.

come tempi di esecuzione che come qualità dei risultati, dalla segmentazione delle immagini, un fatto che rende poco significativi molti risultati vista la ampia gamma di metodi di segmentazione esistenti.

Bisogna poi riconoscere a *MSOM* il merito di utilizzare direttamente una rete *SOM* per il calcolo delle disparità e non come fase iniziale di pre-processing, cosa molto interessante anche dal punto di vista della plausibilità biologica[7]. Solo con *StereoSOM* tuttavia è stata proposta una interessante architettura neurale simmetrica, motivata dal controllo cooperativo di consistenza stereo e capace di fornire direttamente mappe di disparità riferite ad entrambe le immagini della stereocoppia.

Una notevole qualità di *StereoSOM,* ereditata da *MSOM* e facilmente riscontrabile guardando i risultati sperimentali, è l'ottima resa visiva in corrispondenza delle discontinuità di profondità. Questo pregio è dovuto principalmente alla particolare funzione di aggiornamento g, basata su una misura di similarità di colore. L'utilizzo della funzione g nella fase di aggiornamento neurale permette di preservare la forma degli oggetti nella mappa di disparità, una qualità rara negli algoritmi di stereo matching locale che non sfruttano la segmentazione.

Uno dei principali problemi di *MSOM* è il metodo di ricerca delle corrispondenze *pixel-based*, principale causa degli scarsi risultati in termini di errori di matching. Non impiegare una finestra di ricerca, se porta a ovvi vantaggi in prossimità di discontinuità di profondità e occlusioni, in media genera molte più ambiguità rispetto ai metodi *window-based*. *StereoSOM*, sfruttando una ricerca delle corrispondenze basata su supporto variabile, possiede ottime capacità di matching e, allo stesso tempo, gestisce correttamente le zone di discontinuità di profondità vicine ai bordi degli oggetti.

Un altro problema riscontrabile in *MSOM* è l'incapacità di gestire a dovere le aree occluse, funzionalità raggiunta in *StereoSOM* grazie alla particolare architettura che implementa il controllo cooperativo di consistenza stereo.

Riassumendo, da una analisi qualitativa dei modelli è emerso come, con *StereoSOM*, si siano riusciti a conservare i pregi di *MSOM* superando gli svantaggi dovuti all'impiego di una funzione di matching *pixel-based* e ad una mancata gestione diretta delle occlusioni.

[7] Si veda [33] per un approfondimento sulla plausibilità biologica delle reti *SOM*.

Capitolo 5

Valutazione dei risultati

Saranno ora mostrati e commentati i risultati sperimentali ottenuti dall'applicazione del metodo *StereoSOM* ai dataset di valutazione quantitativa (vedere Appendice A) forniti da [41]. I primi quattro dataset sono quelli impiegati dal sistema di valutazione online del Middlebury College. Su queste immagini l'algoritmo *StereoSOM* è stato accuratamente configurato con una attenta fase di tuning. Il motivo di questa scrupolosa configurazione è dato dalla forte competitività del sistema di valutazione utilizzato, dove molti algoritmi vengono confrontati e in cui ogni minimo errore può influire drasticamente sulla classifica finale. La taratura dei parametri utilizzata per i sistemi stereo dei primi quattro dataset è quella riportata in 4.4 nella pagina 66. È comunque doveroso ricordare che, seppur *StereoSOM* ha avuto una configurazione mirata ai primi quattro dataset, per ottenere queste stereocoppie sono stati impiegati ben tre tipi diversi di sistemi stereoscopici: una situazione tutt'altro che ideale.

Per completezza sono stati inseriti anche dataset acquisiti con sistemi stereoscopici differenti da quelli utilizzati nelle prime quattro immagini. Si potrà notare un modesto peggioramento nella qualità complessiva dei risultati, imputabile anche alla maggiore difficoltà di queste stereocoppie. Tuttavia si ricorda che ogni algoritmo di stereo matching andrebbe configurato in funzione del sistema stereoscopico sul quale opera e quindi questi risultati sono da considerarsi molto positivi e incoraggianti.

Per quanto riguarda i risultati dell'algoritmo di partenza *MSOM*, non è stato possibile effettuare test rigorosi poiché non si disponeva di una sua implementazione; tuttavia sono stati condotti degli esperimenti con *StereoSOM* mirati a valutare il

contributo di ciascuna estensione al miglioramento dei risultati.

Per concludere sono stati eseguiti alcuni test di robustezza, modificando o deteriorando sinteticamente i primi quattro dataset e utilizzando immagini non in forma standard.

5.1 Dataset Middlebury

I risultati ottenuti nei dataset forniti con il sistema di valutazione online del Middlebury College sono sicuramente quelli più buoni. Prima di eseguire i test è stata infatti effettuata una accurata fase di tuning dell'algoritmo, mediante test e valutazione degli errori, in modo da adattare *StereoSOM* ai sistemi stereoscopici utilizzati in questi quattro dataset.

Questi test sono da considerarsi i più significativi per valutare contesti applicativi in cui è possibile eseguire una calibrazione del sistema stereoscopico e un successivo tuning dei parametri dell'algoritmo di stereo matching.

Sono state utilizzate due soglie δ_d nel calcolo degli errori di disparità (vedere 2.3 nella pagina 31): $\delta_d = 0.5$ e $\delta_d = 1$. Il primo valore di soglia, più significativo, è molto stringente e permette di apprezzare al meglio le capacità di matching subpixel di *StereoSOM*[1].

Lo schema utilizzato per visualizzare i risultati ottenuti è il seguente:

Mappa di disparità	Mappa dei punti inconsistenti
Mappa binaria degli errori di disparità	Mappa degli errori di matching

Si riporta di seguito la descrizione delle quattro mappe:

Mappa di disparità è la mappa di disparità ottenuta utilizzando l'algoritmo *StereoSOM*, normalizzata per il fattore indicato in ciascun dataset (Appendice A).

[1] Molti algoritmi di stereo matching danno buoni risultati con valori di disparità interi, ma appena si passa ad una soglia di errore pari a $\delta_d = 0.5$ le performance crollano drasticamente; questo fatto è inaccettabile visto che è possibile disporre di mappe di verità a valori reali. Gli unici contesti dove gli algoritmi a valori di disparità interi trovano ancora applicazioni sono quelli *real-time*.

Mappa dei punti inconsistenti contiene i punti in cui il modello *StereoSOM* ha annullato l'aggiornamento neurale grazie al controllo cooperativo di consistenza stereo.

Mappa binaria degli errori di disparità evidenzia i punti nella mappa di disparità che si discostano dai dati reali di un valore di disparità maggiore o uguale a $\delta_d = 0.5$.

Mappa degli errori di matching riporta i punti in cui *StereoSOM* ha eseguito l'aggiornamento neurale basandosi su una corrispondenza errata (calcolabile grazie alle mappe di disparità reali).

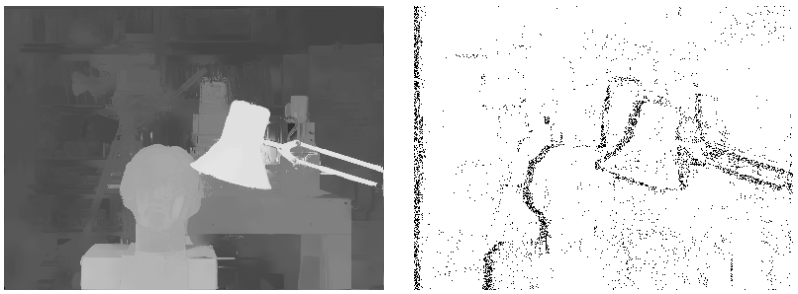
Per alcuni dataset non si dispone delle mappe di disparità reale, in questi casi la seconda riga della tabella non sarà riportata. Per i dataset completi dei dati di verità saranno riportati e commentati i seguenti valori di errore percentuale:

1. *Non occluded regions* (PBD_{nocc})
2. *All regions* (PBD_{all})
3. *Depth discontinuity regions* (PBD_{ddr})
4. *Percentage of Matching Errors*(PME)

Le prime tre misure di errore sono state discusse in 2.3 nella pagina 31, il valore *Percentage of Matching Errors* si riferisce invece al numero di iterazioni in cui *StereoSOM* aggiorna i pesi neurali basandosi su una corrispondenza errata, normalizzato poi per il numero di iterazioni totali (*fase di ordering* + *fase di tuning*).

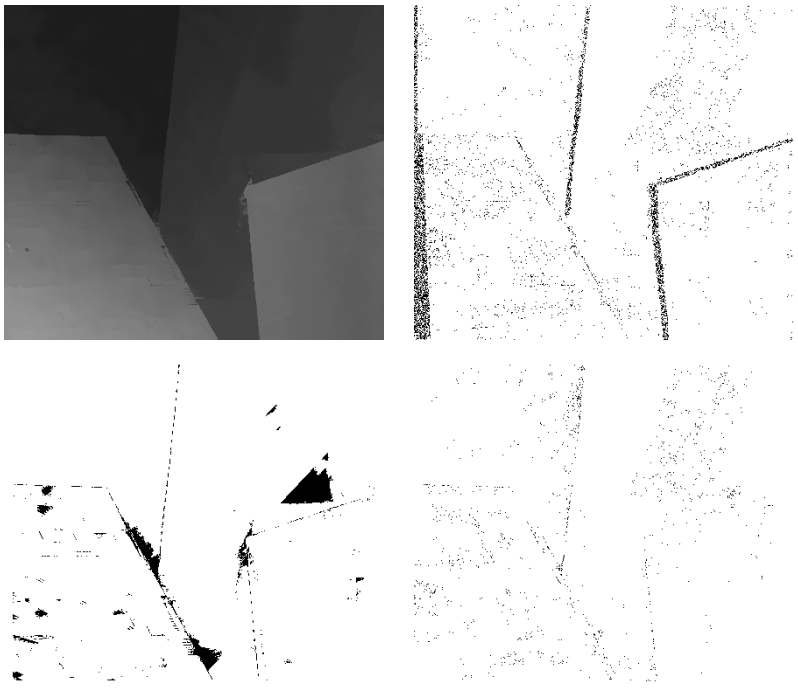
Per una questione di semplicità e compattezza i risultati riportati nelle mappe sono tutti riferiti all'immagine di sinistra I_L, si ricorda che comunque *StereoSOM* fornisce risultati riferiti ad entrambe le immagini I_L e I_R.

5.1.1 "Tsukuba"



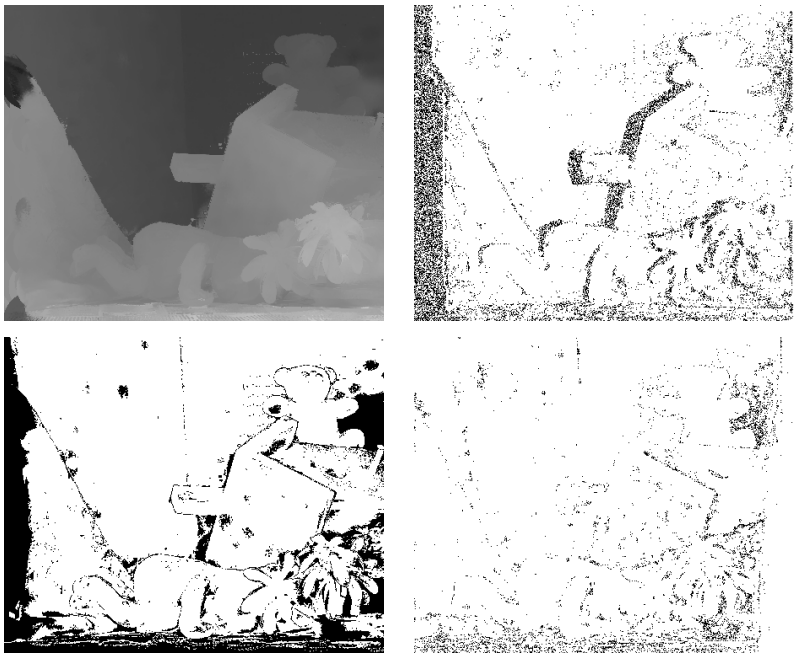
δ_d	PBD_{nocc}	PBD_{all}	PBD_{disc}	PME
0.5	16.70	17.47	32.47	4.09
1	4.05	4.74	18.10	

5.1.2 "Venus"



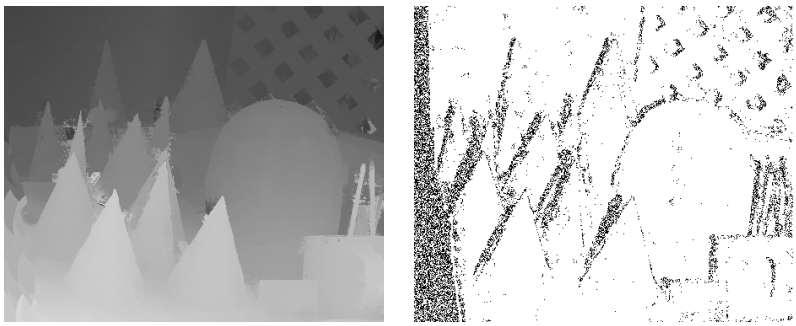
δ_d	PBD_{nocc}	PBD_{all}	PBD_{disc}	PME
0.5	2.78	3.10	12.86	1.26
1	0.53	0.75	6.21	

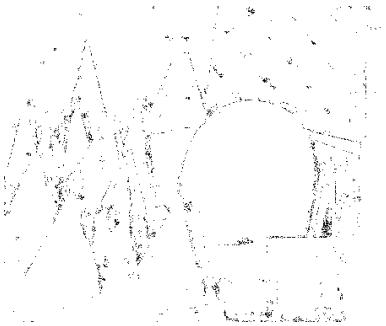
5.1.3 "Teddy"



δ_d	PBD_{nocc}	PBD_{all}	PBD_{disc}	PME
0.5	13.55	19.92	30.03	6.96
1	8.53	13.69	20.21	

5.1.4 "Cones"



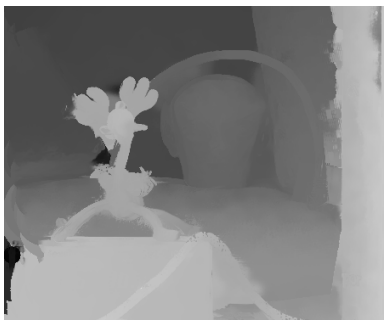
δ_d	PBD_{nocc}	PBD_{all}	PBD_{disc}	PME
0.5	8.28	14.70	20.33	3.01
1	5.07	10.82	14.00	

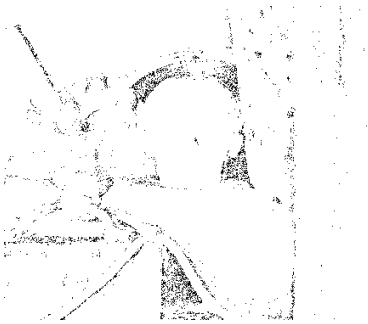
5.1.5 Altre stereocoppie

Le stereocoppie considerate in questa sezione sono sempre tratte da [41], ma in questo caso non è stata effettuata alcuna fase di tuning dei parametri mirata a migliorare la qualità dei risultati su un particolare sistema stereoscopico. Durante l'esecuzione dell'algoritmo sono stati mantenuti gli stessi parametri utilizzati per valutare i primi quattro dataset.

I risultati numerici sono privi dei dati PBD_{nocc} e PBD_{disc} poiché i dataset sono stati forniti sprovvisti delle mappe binarie di zona.

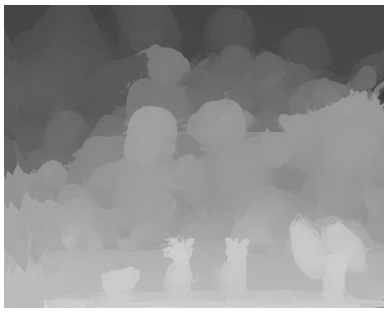
"Reindeer"

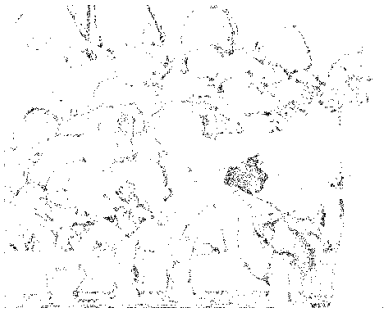


δ_d	PBD_{all}	PME
0.5	32.75	4.75
1	16.93	

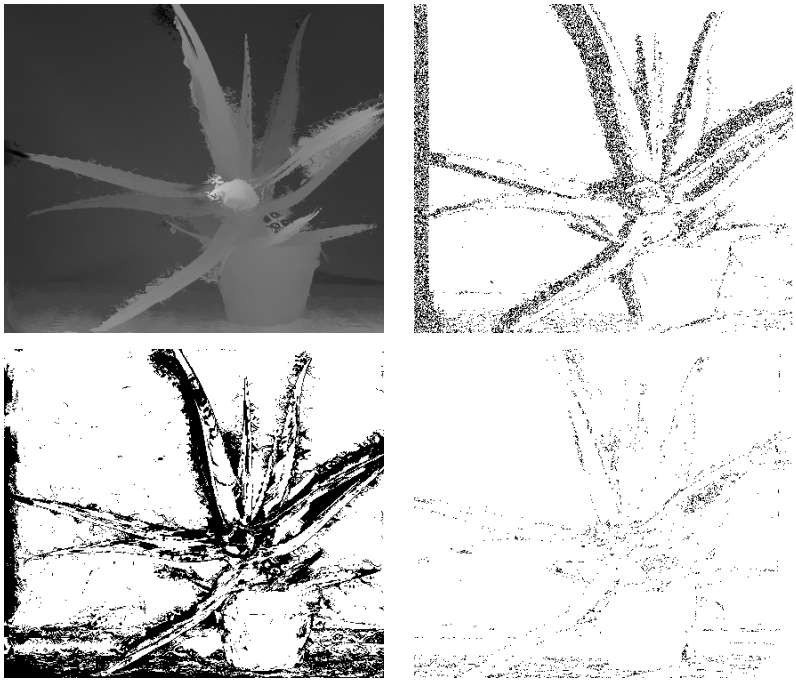
“Dolls”





δ_d	PBD_{all}	PME
0.5	32.97	5.35
1	14.56	

"Aloe"



δ_d	PBD_{all}	PME
0.5	26.46	3.01
1	15.2	

5.1.6 Valutazione dei risultati ottenuti

Come è constatabile nei primi risultati, *StereoSOM* si comporta molto bene con le stereocoppie sulle quali è stato configurato ad hoc. Da notare come vengano preservati nelle mappe di disparità dettagli complessi come i bracci della lampada in *"Tsukuba"* e le bacchette di legno nella tazza di *"Cones"*.

Il dataset "*Venus*" viene praticamente risolto dall'algoritmo, che fornisce risultati corretti al 97% utilizzando la soglia di errore più restrittiva; degna di nota è l'ottima gestione delle aree *low-texture*, frequenti e molto estese in questo dataset.

Nei test con *"Aloe"*, *StereoSOM* riconosce correttamente l'area sullo sfondo contenente pattern periodici, zona molto difficoltosa per via dell'ambiguità nell'assegnamento dei valori di disparità.

Confrontando le *mappe dei punti inconsistenti* con le *mappe binarie delle zone di occlusione* (vedere Appendice A) si deduce come *StereoSOM* riesca a rilevare correttamente le occlusioni. Altro discorso va fatto su come l'algoritmo riesce a gestire

queste aree occluse: nei primi quattro dataset, seppur con qualche problema, la gestione avviene mediamente bene, ma negli ultimi tre dataset le aree occluse sono mal gestite. Si osservino per esempio le zone di occlusione generate dalle foglie della pianta in "*Aloe*", è evidente come l'algoritmo rilevi correttamente le zone ed intervenga fermando l'aggiornamento neurale, così facendo tuttavia alla fine del processo queste aree manterranno i valori di disparità approssimativa calcolati nella fase di ordering (risultati poi errati come si può vedere nella *mappa binaria degli errori di disparità*). Sembrerebbe quindi che la qualità della gestione delle occlusioni in *StereoSOM* sia fortemente legata alla configurazione dell'algoritmo sul sistema stereoscopico.

Sorprendentemente l'errore di matching PME resta sempre a ottimi livelli, anche negli ultimi dataset, totalizzando un picco del 6.96% per il dataset "*Teddy*", dove comunque vengono raggiunti buoni risultati finali. Evidentemente l'impiego del supporto adattivo da ottimi risultati, visibili chiaramente nelle *mappe degli errori di matching* che evidenziano una distribuzione degli errori di corrispondenza quasi nulla nelle aree fronto parallele e contenuta nelle zone di discontinuità di profondità.

Un grave problema di *StereoSOM* è l'assegnamento della disparità laddove, agli estremi di una discontinuità di profondità, vi sono aree con colori simili. Si consideri sempre il dataset "*Aloe*": vicino alle foglie è possibile notare come l'algoritmo assegni la disparità calcolata per la pianta ad alcune aree verdi del pattern sullo sfondo. Questo problema è principalmente legato all'assenza di una fase di segmentazione delle immagini, *StereoSOM* infatti utilizza la semplice similarità di colore (funzione g) per disciplinare l'assegnamento dei valori di disparità nel vicinato del neurone eletto. I problemi con le discontinuità di profondità sono comunque chiaramente riscontrabili nell'errore cumulativo PBD_{disc}, che in quasi tutti i casi risulta più del doppio di PBD_{nocc}.

In generale, tutte le mappe di disparità generate da *StereoSOM* sono affette da un lieve rumore dipendente dalla distribuzione delle intensità nelle stereocoppie, anche questo problema dipende dalla funzione di similarità di colore g e potrebbe essere risolto in futuro con una fase di segmentazione delle immagini.

I risultati ottenuti dalle ultime tre stereocoppie, seppure non brillanti, hanno comunque mostrato una buona tolleranza dell'algoritmo verso sistemi stereoscopici nuovi: basti considerare che il PBD_{all} con soglia $\delta_d = 1$ resta sempre intorno al

15%, un dato accettabile in contesti di applicazione "*general purpose*". Infine non va dimenticato che le stereocoppie impiegate in questi test sono davvero molto complesse ed elaborate, degli esempi piuttosto lontani dai casi tipici di utilizzo di un algoritmo stereo.

Ritornando al sistema di valutazione Web del Middlebury College (descritto in 2.3 nella pagina 31), *StereoSOM* si classifica tuttora al 21^o posto su più di 70 algoritmi di stereo matching (considerando la soglia di valutazione più stringente $\delta_d = 0.5$, vedere Figura 2.5 nella pagina 34). Questo risultato è alquanto positivo visto che oltre metà degli algoritmi nel database online, tra cui tutti quelli in posizioni superiori rispetto a *StereoSOM*, sono ad approccio globale e quindi computazionalmente esigenti e molto meno ottimizzabili.

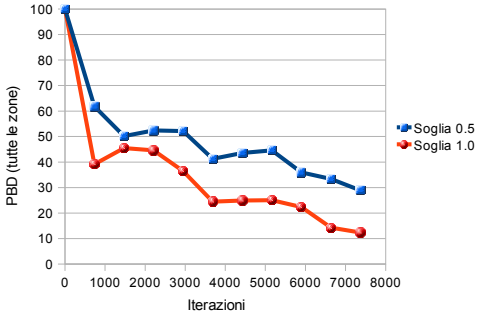
5.2 Convergenza e numero di iterazioni

In tutti gli algoritmi basati su reti *SOM* è necessario effettuare alcune analisi riguardanti la convergenza dei risultati in funzione delle iterazioni. La notevole riduzione del numero totale di iterazioni richieste rispetto all'algoritmo *MSOM* è da ricondursi principalmente all'introduzione della fase di ordering, che inizializza i pesi neurali a valori "ragionevoli" di disparità. La successiva fase di tuning, che occuperà comunque la maggior parte del tempo di esecuzione, andrà a raffinare i risultati.

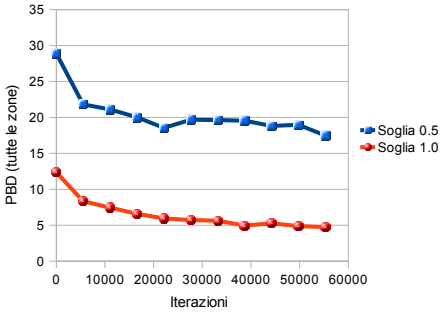
Di seguito sono riportate alcune coppie di grafici, riferiti ai dataset di esempio, dove è mostrato l'errore PBD_{all} commesso dall'algoritmo in funzione delle iterazioni. I dati riportati nei grafici di sinistra si riferiscono a dieci valori di errore, rilevati periodicamente dall'inizio alla fine della fase di ordering. Nei grafici di destra sono riportati i medesimi dati di errore riferiti però alla fase di tuning.

⇓ Dataset *"Tsukuba"*

Fase di ordering

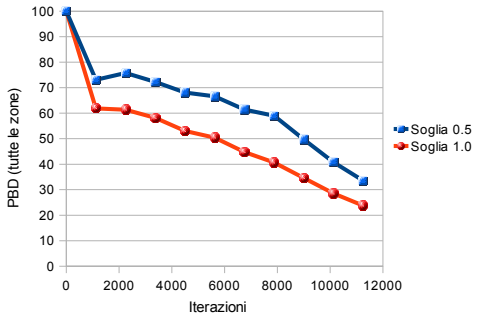


Fase di tuning

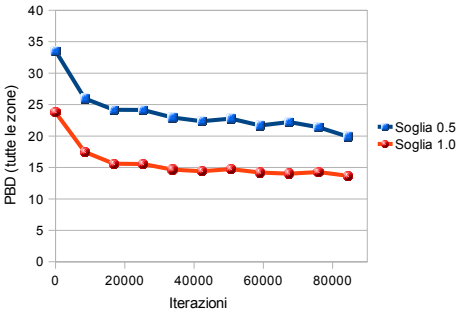


⇓ Dataset *"Teddy"*

Fase di ordering

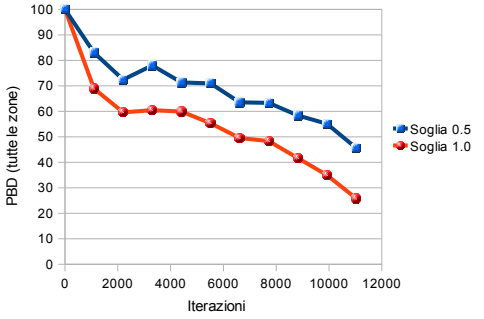


Fase di tuning

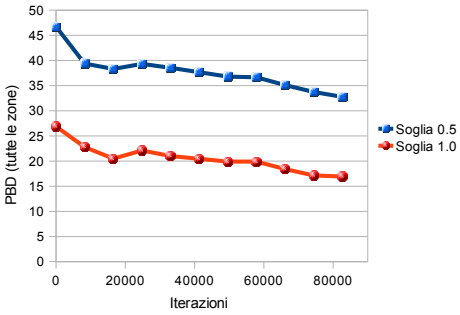


⇓ Dataset *"Reindeer"*

Fase di ordering

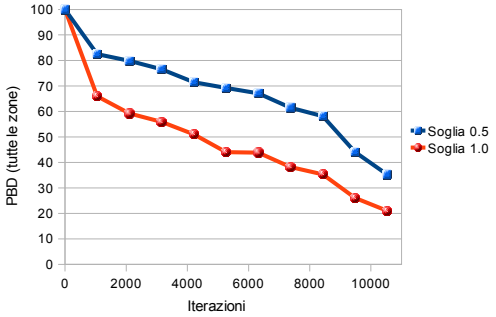


Fase di tuning

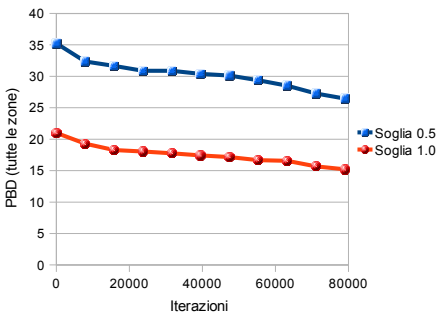


⇓ Dataset *"Aloe"*

Fase di ordering



Fase di tuning



Si noti come gran parte del lavoro venga svolta in poco tempo dalla fase di ordering e dalle prime iterazioni della fase di tuning. Tuttavia, per raggiungere risultati apprezzabili e comparabili con i migliori algoritmi di stereo matching, è necessario raffinare le mappe di disparità attraverso una lunga fase di tuning[2].

Per i dataset "*Reindeer*" e "*Aloe*", casi in cui l'algoritmo non è stato configurato ad hoc, non è possibile attribuire la causa degli scarsi risultati a nessuna delle due fasi. In "*Reindeer*" l'insufficienza dei risultati sembrerebbe essere imputabile alla fase di ordering, che non riesce ad "abbattere" l'errore PBD quanto avviene per gli altri dataset. Per "*Aloe*" avviene invece l'esatto contrario: la fase di ordering si comporta in modo atteso mentre la fase di tuning non riesce ad essere abbastanza efficace.

5.3 Analisi delle estensioni apportate

In questa parte saranno analizzate e valutate le varie estensioni apportate all'algoritmo *MSOM*; come si vedrà, ogni estensione riesce ad introdurre notevoli miglioramenti nei risultati finali. Nei seguenti test è stata impiegata la soglia di errore PBD pari a $\delta_d = 0.5$.

[2] È da intendersi un concetto relativo di lunghezza: la fase di tuning impiega approssimativamente il 90% del tempo richiesto dall'esecuzione dell'algoritmo.

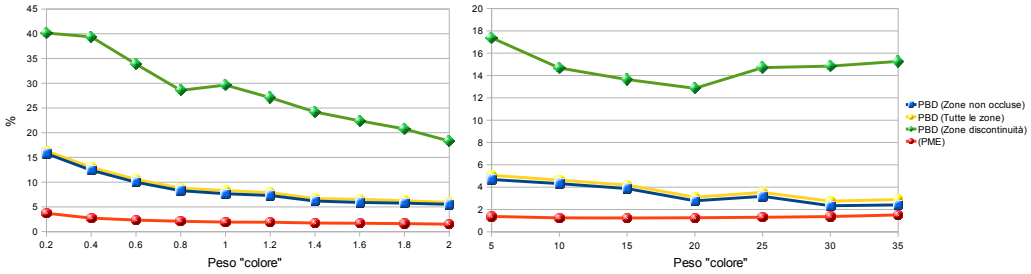
5.3.1 Ricerca ponderata

In queste prove viene fatto variare il peso della feature colore ρ_{C_k} nella fase di tuning con queste modalità:

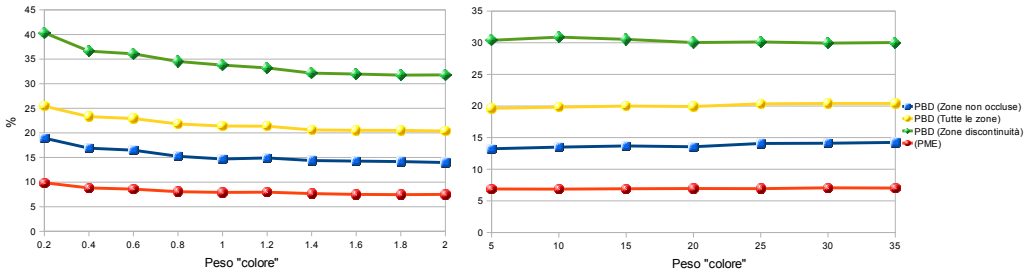
- un intorno del valore $\rho_{C_k} = 1$, peso utilizzato nel modello iniziale *MSOM*
- un intorno del valore $\rho_{C_k} = 20$, peso proposto per il modello *StereoSOM*

Sono stati condotti esperimenti solo sulla fase di tuning poiché nella fase di ordering il valore preciso di ρ_{C_k} non è molto importante, la cosa fondamentale è che sia molto grande rispetto a ρ_1 (vedere 4.3.2 nella pagina 59); il principale motivo è la velocizzazione della convergenza e la necessità di limitare il fattore di smoothing ρ_1, molto dannoso con la funzione di similarità di colore g disabilitata.

⇓ Dataset *"Venus"*



⇓ Dataset *"Teddy"*



Da una analisi dei dati si intuisce come per dataset "semplici" (e.g. *"Venus"*) la riduzione dell'azione del fattore di smoothing ρ_1 dia un notevole contributo alla velocità del processo. Questo è deducibile notando come i valori dell'errore *PME* risultino quasi indipendenti dall'entità dei pesi di ricerca: se la percentuale di errori di matching resta costante, il miglioramento dei risultati finali è da attribuirsi

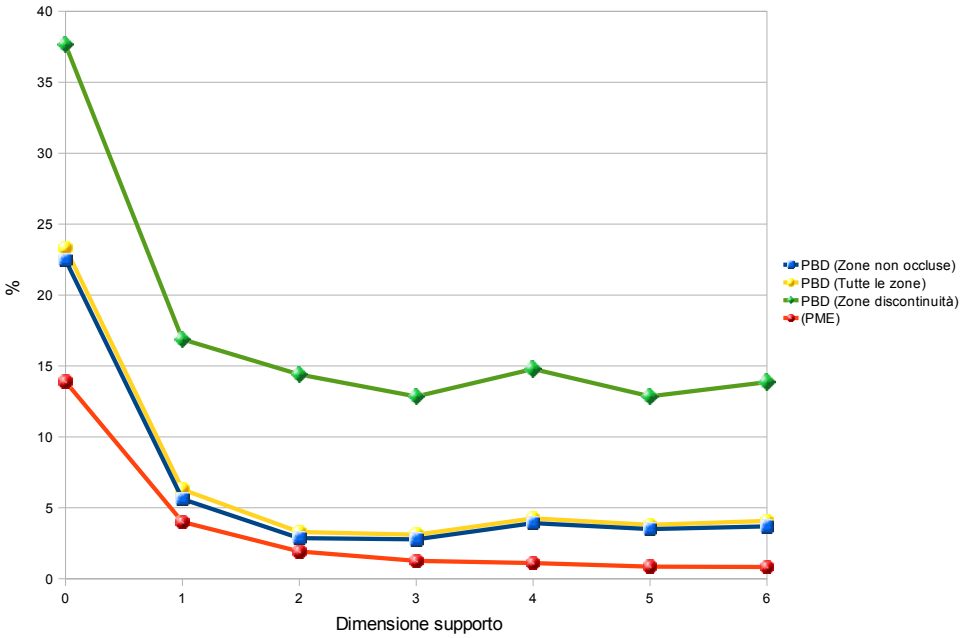
alla maggiore velocità di convergenza dell'algoritmo, dovuta evidentemente alla riduzione degli annullamenti causati dal controllo di consistenza stereo.

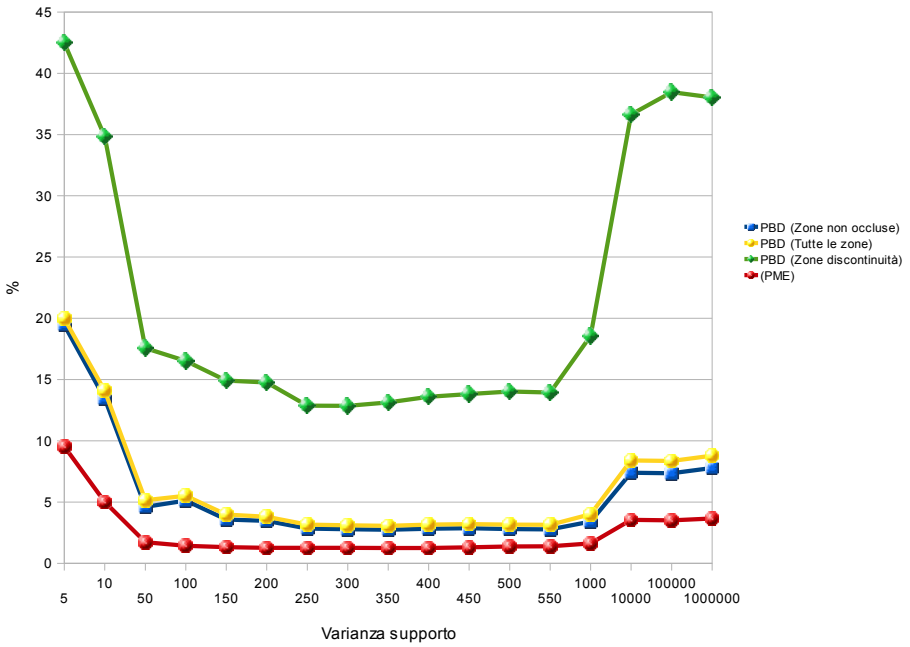
Su dataset complessi (e.g. "*Teddy*") i contributi dei pesi sono meno apprezzabili, anche in questo caso si può comunque notare un lieve miglioramento dei risultati dovuto alla riduzione dell'azione del fattore di smoothing.

5.3.2 Supporto di ricerca adattivo

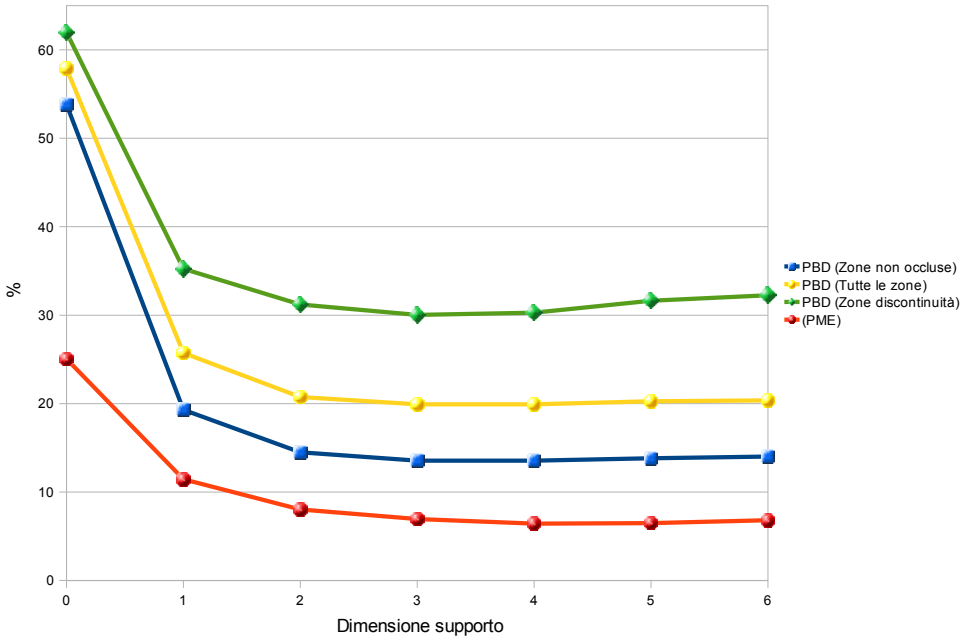
Sono stati eseguiti degli esperimenti sul supporto di ricerca adattivo (4.3.2 nella pagina 59) e, in particolare, è stato analizzato il comportamento di *StereoSOM* al variare della dimensione del supporto di ricerca adattivo (ξ) e del valore di varianza che regola l'adattività del supporto (σ_s^2). Per le dimensioni del supporto, sono stati utilizzati valori da $\xi = 0$ a $\xi = 6$, il primo estremo riconduce l'algoritmo ad un metodo *pixel-based*, il secondo estremo corrisponde ad una finestra di ricerca di dimensioni 13×13. Di seguito riportiamo alcuni risultati:

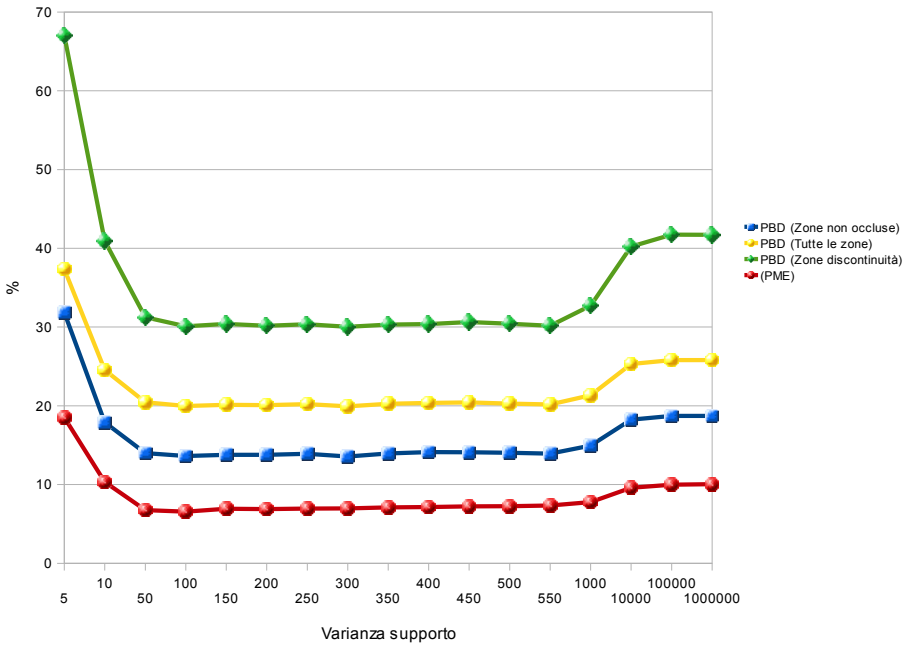
⇓ Dataset "*Venus*"





⇓ Dataset *"Teddy"*





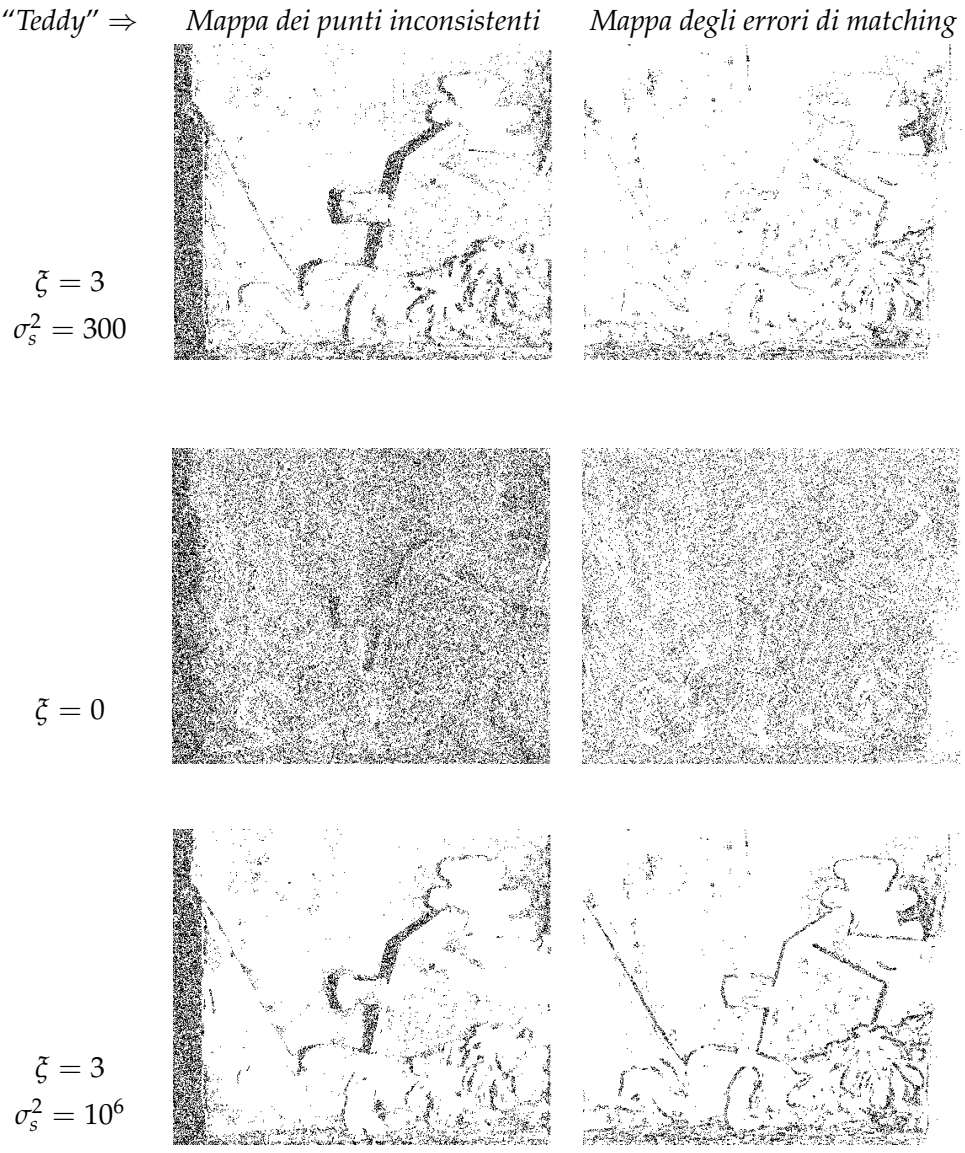
Dai dati ottenuti in funzione della dimensione del supporto, si può notare come l'approccio *pixel-based* ($\xi = 0$) dia pessimi risultati finali. Forse un maggiore numero di iterazioni potrebbe ridurre l'errore complessivo, ma difficilmente ridurrà il valore PME, che è normalizzato per il numero di iterazioni. È quindi evidente come i metodi basati sull'utilizzo del singolo pixel siano caratterizzati da un rapporto segnale/rumore molto basso, causa poi di una impennata degli errori di matching. I metodi che utilizzano un supporto, possono confidare invece in un rapporto segnale/rumore decisamente più alto.

Un'altra caratteristica che salta all'occhio è l'effetto dell'utilizzo di un supporto non adattivo (ottenuto impostando $\sigma_s^2 = 10^6$), come ci si aspetterebbe vengono generati numerosi errori nelle zone di discontinuità di profondità. È infatti proprio questo il problema che affligge gli algoritmi di stereo matching basati su supporto fisso: nelle aree di discontinuità di profondità la finestra di correlazione tende ad includere superfici con disparità diverse, generando ovvi problemi di corrispondenza. In *StereoSOM* invece il supporto adattivo interviene dinamicamente, scartando i punti con disparità potenzialmente diversa da quello centrale e permettendo stime di disparità molto più affidabili.

Come ultima considerazione, si noti l'effetto di una riduzione estrema della varianza del supporto adattivo ($\sigma_s^2 = 0.1$); il supporto tenderà a considerare come

punti validi solo quelli con colori strettamente simili a quello centrale, comportandosi, nella migliore delle ipotesi, come un metodo *pixel-based*.

Al fine di apprezzare al meglio l'importanza del supporto adattivo di *Stereo-SOM*, sono riportati di seguito tre casi significativi estratti dai risultati sul dataset "*Teddy*":



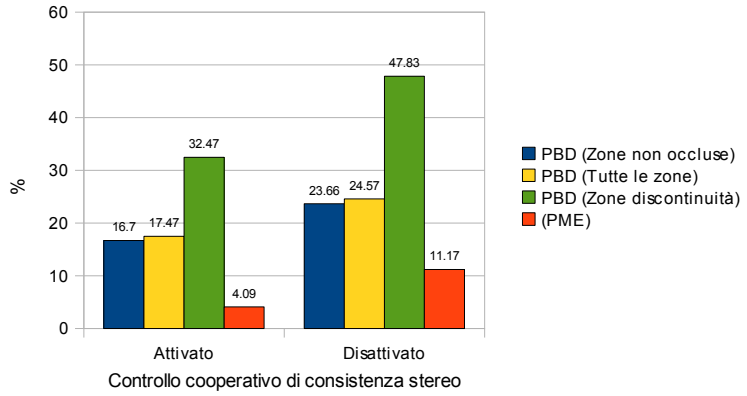
Nel primo caso il supporto adattivo è stato disattivato ($\xi = 0$) e l'algoritmo si comporta usando l'approccio *pixel-based*; di conseguenza i punti di inconsistenza e gli errori di matching si distribuiscono omogeneamente su tutta la mappa. Nel

secondo caso il supporto adattivo diventa fisso ($\sigma_s^2 = 10^6$) e, come è stato rilevato in precedenza con i dati numerici, gli errori di matching risultano più concentrati nelle zone di discontinuità di profondità.

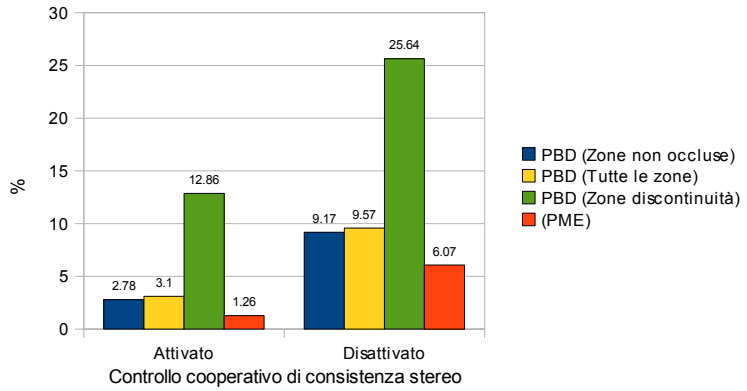
5.3.3 Controllo cooperativo di consistenza stereo

Nel seguente test viene valutato il contributo del controllo cooperativo di consistenza stereo (4.3.2 nella pagina 63):

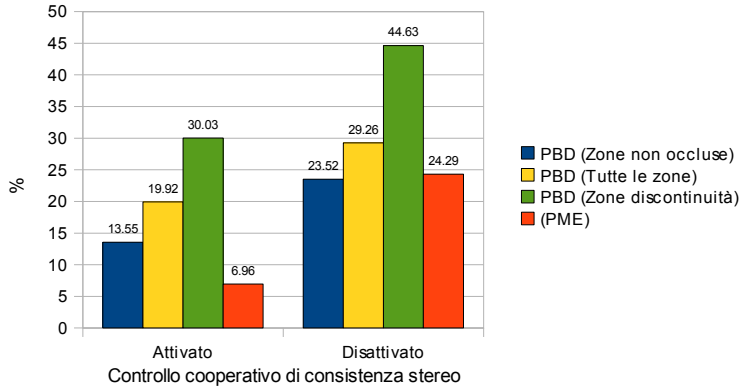
⇓ Dataset *"Tsukuba"*



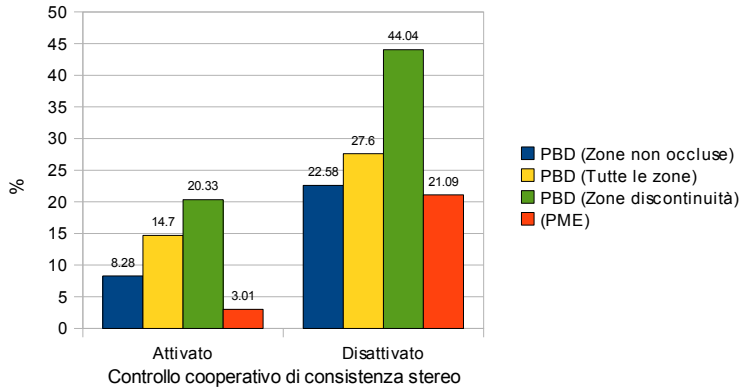
⇓ Dataset *"Venus"*



⇓ Dataset *"Teddy"*

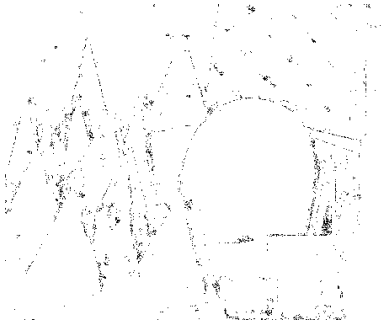
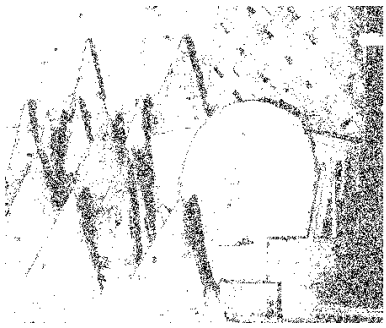


⇓ Dataset *"Cones"*



L'importanza di questa estensione è davvero notevole: si possono infatti apprezzare netti miglioramenti dei risultati con l'introduzione del controllo di consistenza, estesi in particolar modo all'errore di matching *PME*.

Per visualizzare qualitativamente il contributo di questa estensione, si considerino le seguenti mappe degli errori di matching riferite al dataset "*Cones*":

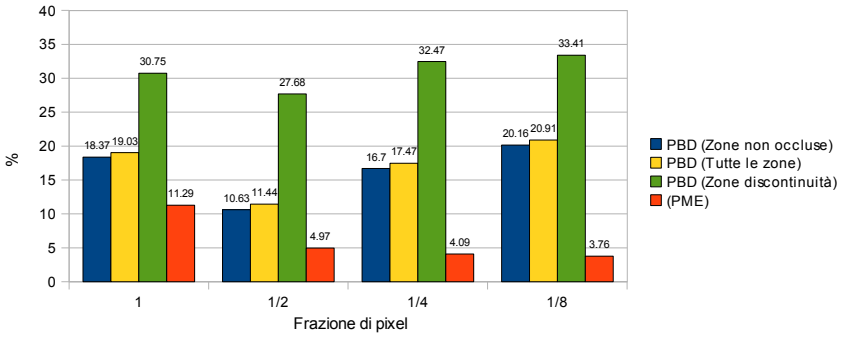
"*Cones*" ⇒	*Mappa degli errori di matching*
Controllo cooperativo di consistenza **attivato**	
Controllo cooperativo di consistenza **disattivato**	

Con l'utilizzo del controllo si riducono drasticamente gli errori di matching in tutte le zone occluse. Da notare inoltre come i punti nel bordo destro dell'immagine I_R, che risultano esclusi dal campo visivo dell'immagine I_L, vengano correttamente scartati dall'algoritmo grazie al controllo di consistenza.

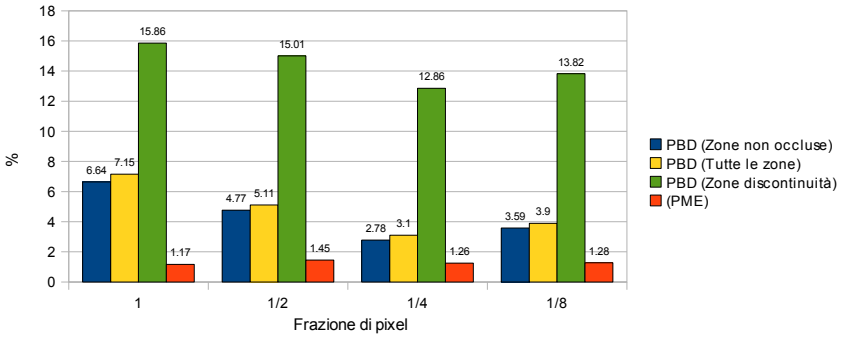
5.3.4 Valutazione sub-pixel

La valutazione *sub-pixel* di *StereoSOM* (4.3.2 nella pagina 62) si avvale dell'interpolazione lineare per stimare i valori di intensità a cavallo di due pixel. Questa estensione è stata introdotta principalmente al fine di favorire la generazione di mappe di disparità a valori reali. I seguenti test mostrano quantitativamente i risultati ottenuti al variare della frazione di pixel considerata.

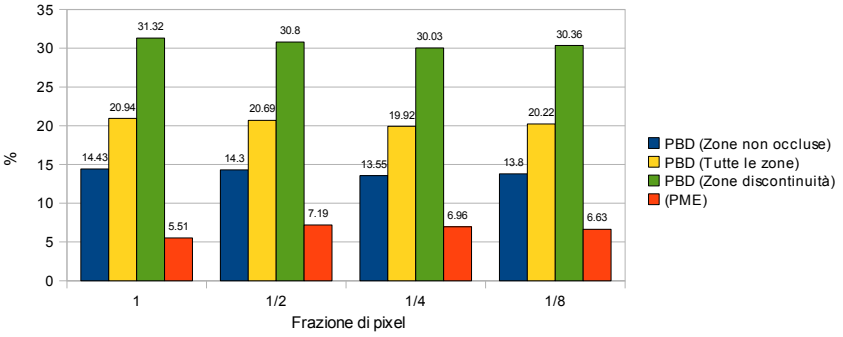
⇓ Dataset *"Tsukuba"*



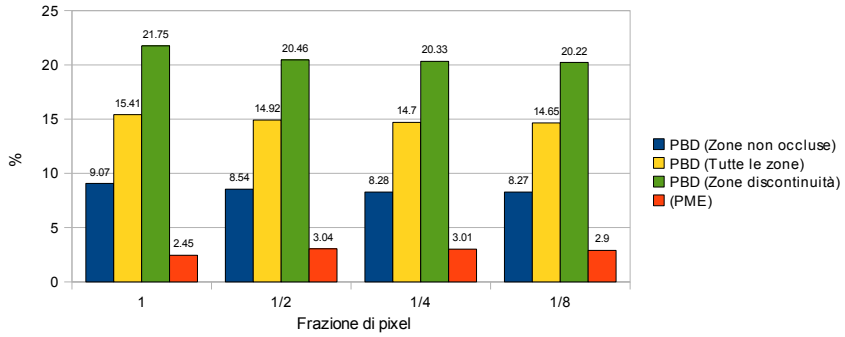
⇓ Dataset *"Venus"*



⇓ Dataset *"Teddy"*

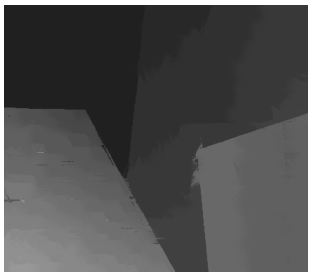
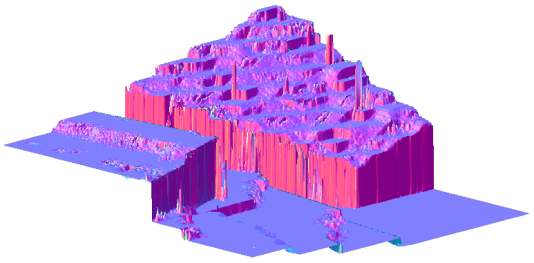
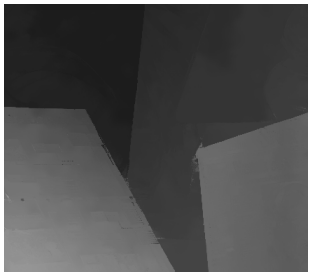
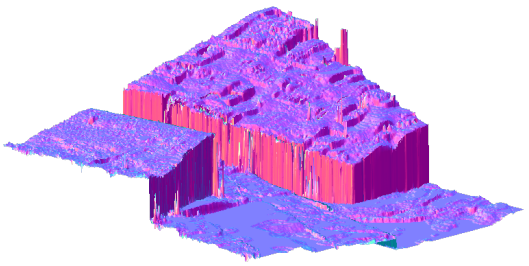


⇓ Dataset *"Cones"*



Non è facile interpretare i risultati appena mostrati. Ridurre la frazione di pixel nella valutazione *sub-pixel* ha dei pro e dei contro: può essere utile in termini di precisione, ma tende ad aumentare lo spazio delle possibili corrispondenze, incrementando la frequenza di detezione degli errori di consistenza. Un buon compromesso è stato comunque trovato con $\frac{1}{4}$ di pixel.

Un modo per apprezzare a pieno la qualità della valutazione *sub-pixel* è guardare le mappe di disparità e le corrispondenti ricostruzioni 3D delle profondità:

"Venus" ⇒	*Mappa di disparità*	*Ricostruzione 3D profondità*
Frazione di pixel: **1**		
Frazione di pixel: $\frac{1}{4}$		

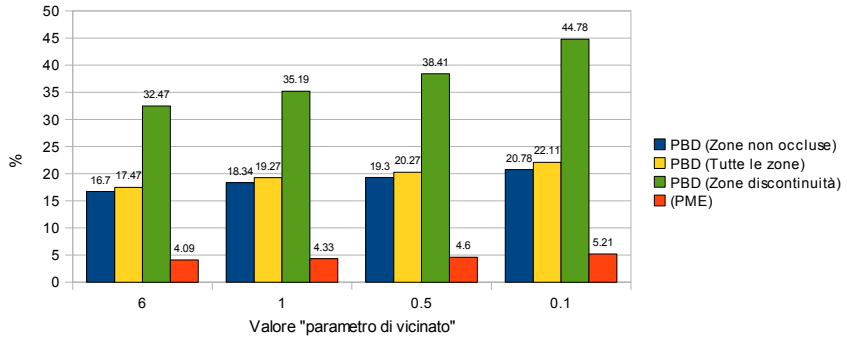
Nelle ricostruzioni delle profondità, si nota come il fenomeno della "scalettatura" sia molto più marcato senza l'utilizzo della valutazione *sub-pixel*. Grazie all'estensione appena analizzata, *StereoSOM* genera mappe di disparità a valori reali migliori rispetto a quelle ottenibili senza la valutazione *sub-pixel*.

5.3.5 Aggiornamento neurale

Come è stato discusso in 4.3.3 nella pagina 64, con *StereoSOM* è stata definita una nuova funzione di apprendimento h, basata sul vincolo di continuità. Per valutarne le qualità sono stati condotti due tipi di test: il primo mostra come si comporta l'algoritmo al variare dei parametri che controllano l'ampiezza (α,β) della funzione gaussiana h, la seconda prova misura i risultati al variare delle dimensioni del vicinato (n_{size}) in cui la funzione h opera.

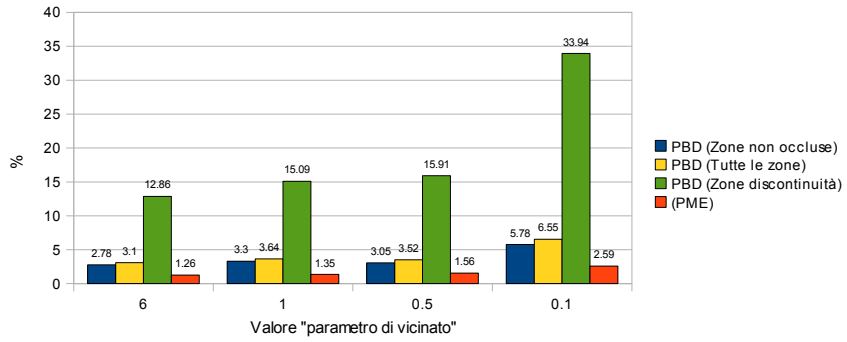
Iniziamo con i parametri α,β: chiameremo "parametro di vicinato" il valore di α quando $\beta = \frac{\alpha}{100}$. Il seguente test mostra, nella prima serie di dati, i risultati ottenuti con la configurazione nuova dell'algoritmo ($\alpha_i = 6$, $\beta_i = \frac{6}{100}$, $\alpha_f = 1$, $\beta_f = \frac{1}{100}$), mentre nelle altre serie sono riportati i risultati ottenuti con la configurazione "classica" impiegata da *MSOM*[3]. È necessario ricordare che *MSOM* non prevedeva una variazione nel tempo della funzione di apprendimento, quindi per $\alpha \neq 6$ si avrà che $\alpha_i = \alpha_f$ e $\beta_i = \beta_f$.

⇓ Dataset *"Tsukuba"*

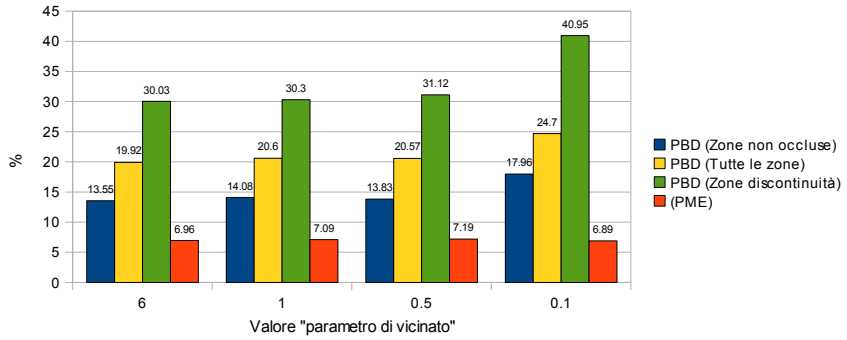


[3]Si ricorda che per valori $\alpha \leq 1$, la nuova funzione h è una normale gaussiana, paragonabile a quella utilizzata nell'algoritmo *MSOM*.

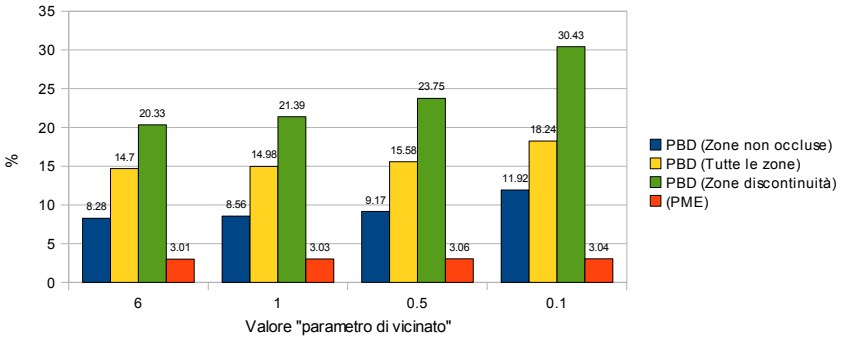
⇓ Dataset *"Venus"*



⇓ Dataset *"Teddy"*



⇓ Dataset *"Cones"*



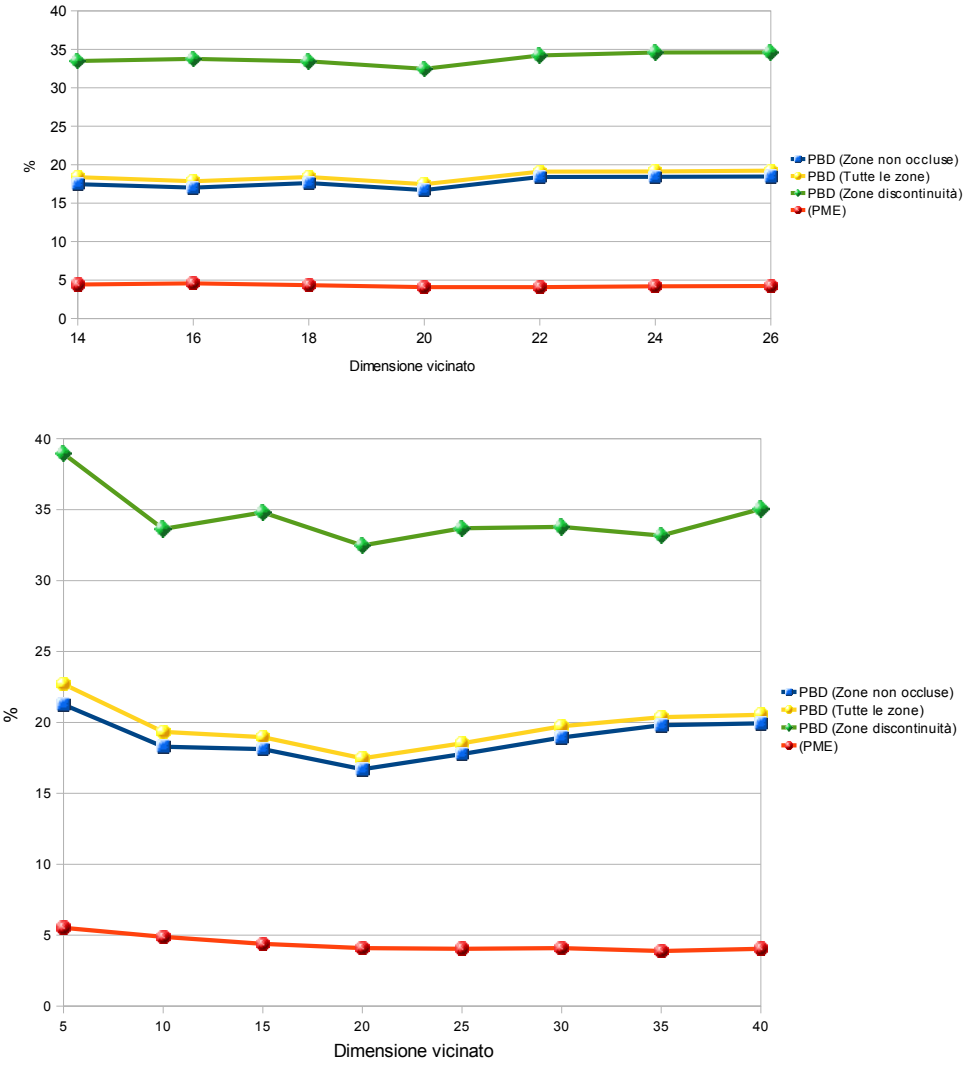
L'utilizzo della nuova funzione *h* porta ad un discreto miglioramento dei risultati finali, apprezzabile soprattutto nei primi dataset dove l'ipotesi di continuità trova maggiore validità, essendo la scena composta da molti oggetti con superfici fronto-paralleli.

Per quanto riguarda il parametro che regola l'estensione del vicinato si considerino i test seguenti, dove *StereoSOM* è stato analizzato con queste modalità:

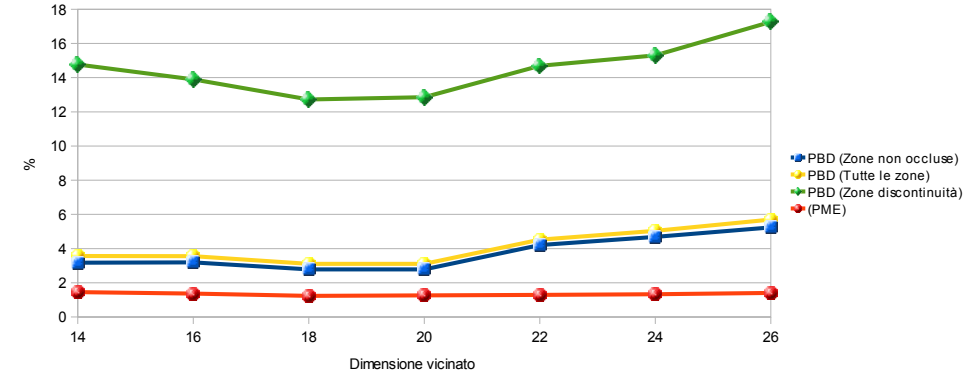
- per valori di n_{size} che vanno da $n_{size} = 14$ (29×29 neuroni vicini al nodo eletto interessati dall'aggiornamento) a $n_{size} = 26$ (53×53 neuroni coinvolti)

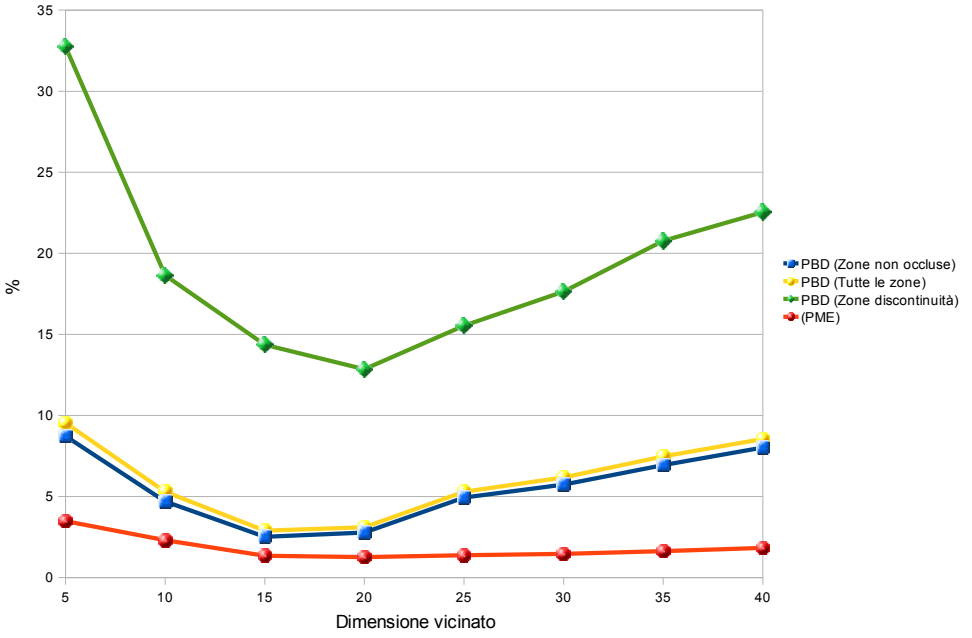
- per valori di n_{size} che vanno da $n_{size} = 5$ (11×11 neuroni vicini al nodo eletto interessati dall'aggiornamento) a $n_{size} = 40$ (81×81 neuroni coinvolti)

⇓ Dataset *"Tsukuba"*

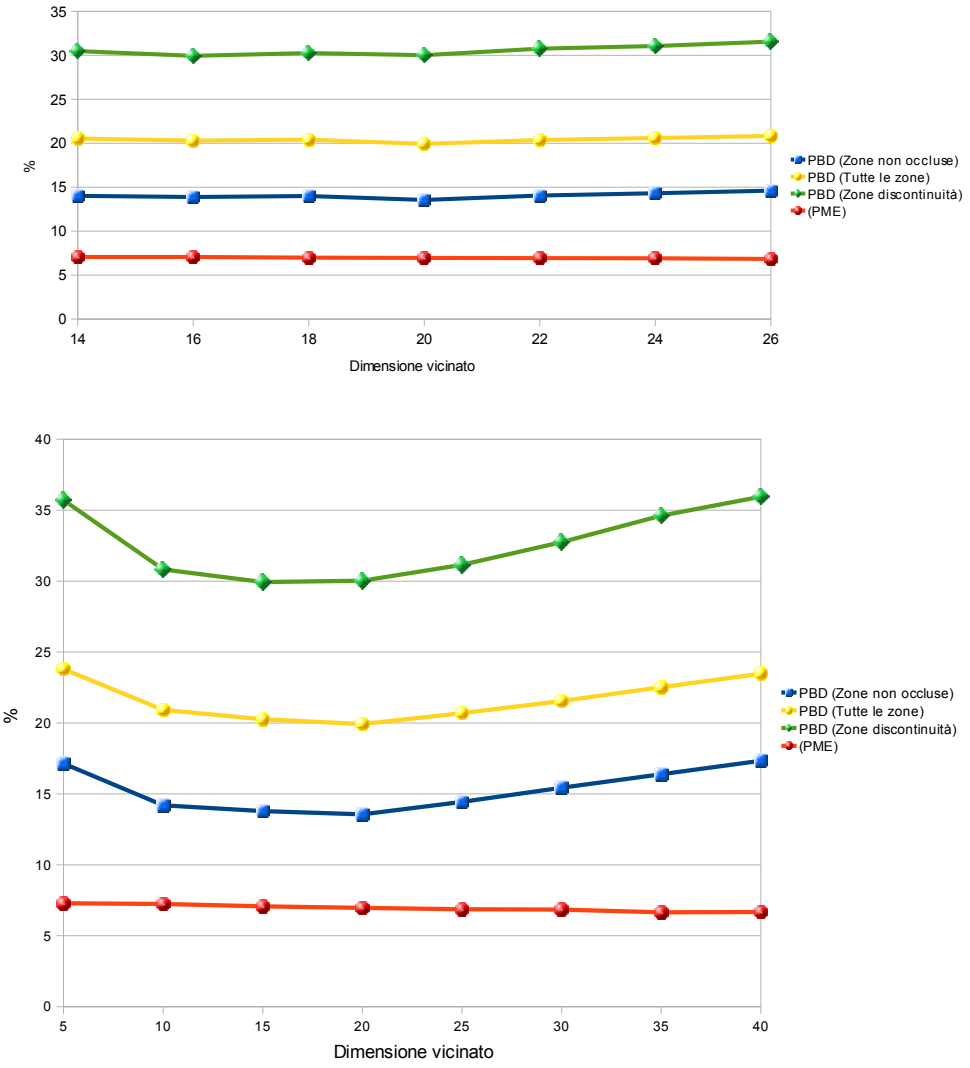


⇓ Dataset *"Venus"*

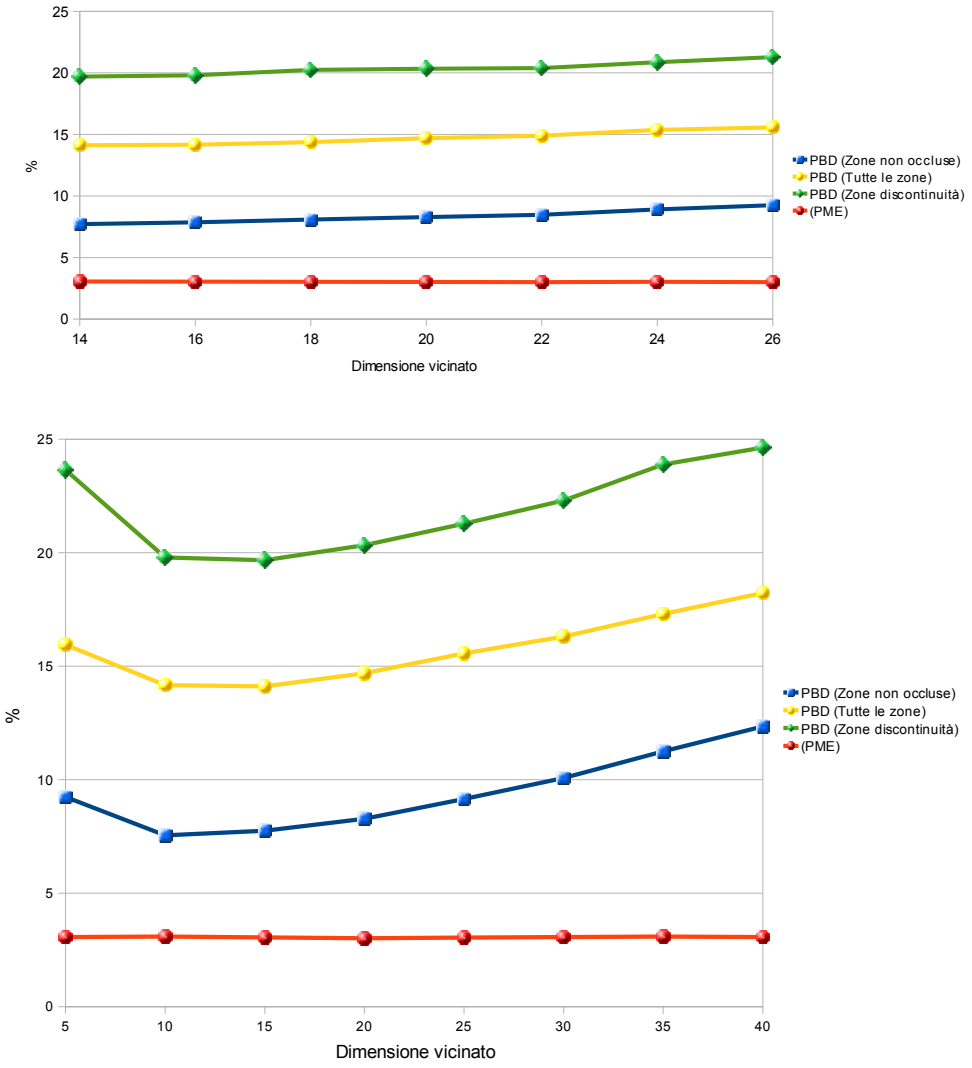




⇓ Dataset *"Teddy"*



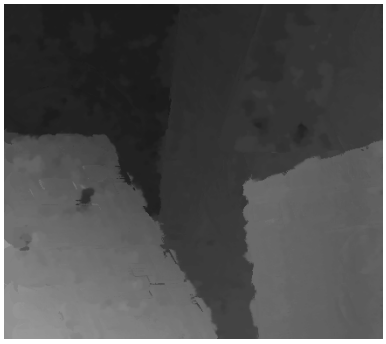
⇓ Dataset *"Cones"*



Dalle prove sembrerebbe che piccoli cambiamenti al parametro n_{size} non influiscano in modo significativo sulla qualità dei risultati. Si può comunque notare una crescita lineare dell'errore per valori di n_{size} maggiori di 15/20; evidentemente estendere l'aggiornamento ad un vicinato troppo grande da problemi, soprattutto nei dataset con mappe di disparità a valori reali, dove l'ipotesi di continuità ha meno validità.

Come si può vedere nella seguente immagine, dimensioni di vicinato troppo ridotte aumentano sostanzialmente i tempi richiesti per la convergenza dell'algoritmo, vanificando l'effetto della fase di tuning:

"*Venus*" ⇒ *Mappa di disparità*



$n_{size} = 5$

Si può comunque constatare come gli errori di matching PME risultino poco dipendenti dal parametro n_{size}.

5.4 Analisi della robustezza

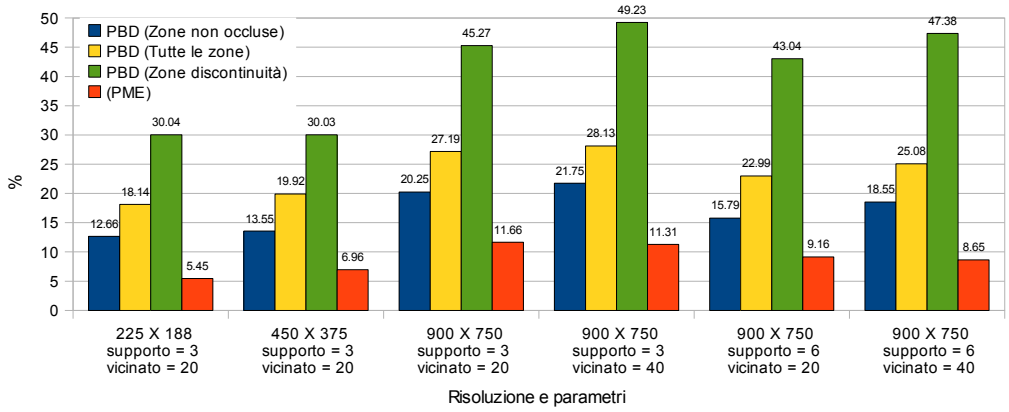
Al fine di valutare la robustezza dell'algoritmo a input non ideali, sono stati eseguiti test utilizzando dataset corrotti sinteticamente o immagini non in forma standard. In tutte le prove è stata impiegata la soglia di errore PBD pari a $\delta_d = 0.5$.

Per applicare le funzioni di rumore o di trasformazione lineare alle immagini, è stato impiegato il software *ImageJ* [57].

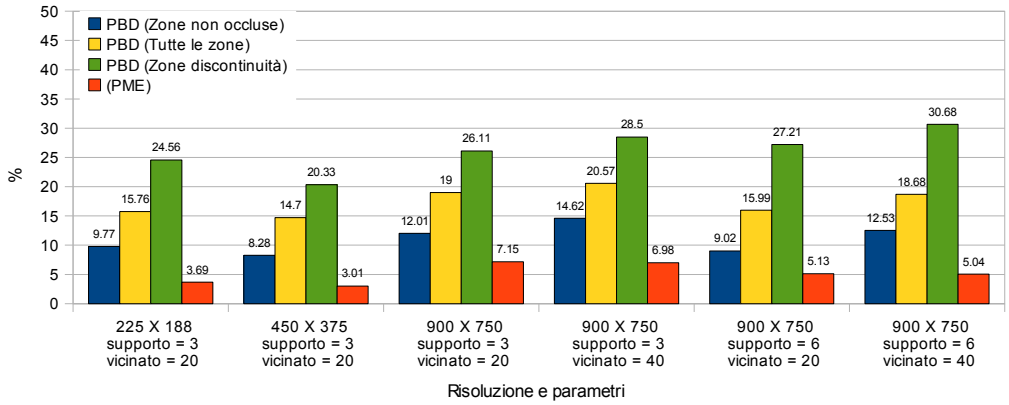
5.4.1 Cambiamenti di risoluzione

In alcuni contesti può essere necessario utilizzare delle stereocoppie alla loro risoluzione originale. I dataset analizzati fino ad ora hanno risoluzioni (e conseguentemente dimensioni) simili, ma non ci sono per ora dati che dimostrano il comportamento dell'algoritmo al variare della risoluzione. Nei seguenti test sono stati impiegati due dataset, "*Teddy*" e "*Cones*", in tre risoluzioni differenti: la prima è di $225 \times 188\, pixel$, la seconda $450 \times 375\, pixel$ (risoluzione originale) e la terza $900 \times 750\, pixel$. Si fa notare che, per aumentare la risoluzione delle immagini, non sono stati impiegati metodi basati sull'interpolazione, ma sono state utilizzate versioni dei dataset a risoluzioni "di scatto" maggiori.

⇓ Dataset *"Teddy"*



⇓ Dataset *"Cones"*

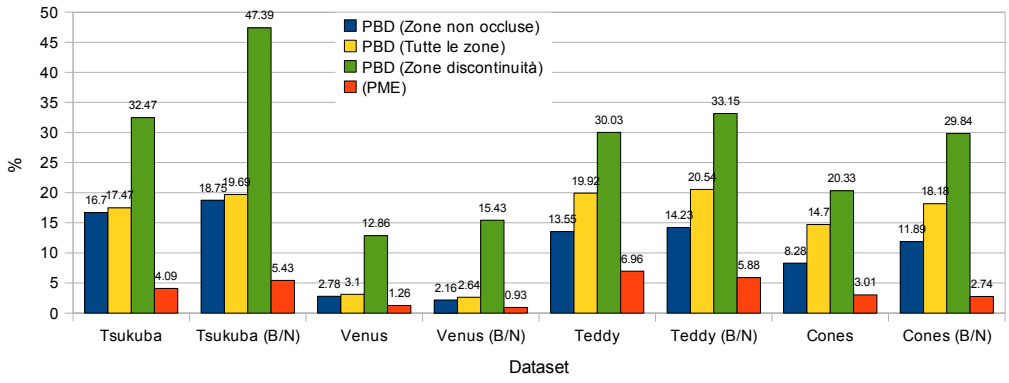


Con le prime due risoluzioni non si evidenziano sostanziali differenze, mentre con l'alta risoluzione l'algoritmo non si comporta in modo corretto. Come è intuibile guardando le ultime tre serie dell'istogramma, per tentare di risolvere il problema sono state prese due strade diverse: la prima prevede il raddoppio della dimensione dell'area del vicinato neurale interessato dall'aggiornamento (n_{size}), la seconda il raddoppio delle dimensioni del supporto (ξ). Infatti questi parametri regolano le dimensioni di alcune aree e, intuitivamente, potrebbero essere dipendenti dalla risoluzione delle stereocoppie.

Raddoppiare la dimensione dell'area di aggiornamento non sembra portare a vantaggi, invece il raddoppio della dimensione del supporto adattivo porta ad un concreto miglioramento dei risultati. Le zone di discontinuità di profondità sembrano il problema principale alle alte risoluzioni.

5.4.2 Immagini in scala di grigi

Il metodo *MSOM* operava utilizzando solo un canale intensità delle stereocoppie. *StereoSOM* utilizza i canali colore per aumentare il numero di informazioni utilizzabili per i confronti e migliorare così il matching. Di seguito si riportano alcuni test in cui è stato impiegato *StereoSOM* su immagini in bianco e nero (B/N), prive dei canali colore.



Come da premessa, l'utilizzo dei colori porta a buoni miglioramenti della qualità dei risultati. Da notare come, con l'utilizzo di immagini in bianco e nero, gli errori *PBD* salgono mentre gli errori di matching *PME* generalmente si riducono. Questo non va interpretato come un miglioramento della fase di matching: utilizzando immagini senza colori vengono infatti generate più ambiguità e il controllo di consistenza stereo interviene più spesso, in questo modo si limitano gli errori totali di matching, ma si rallenta anche il processo di convergenza dell'algoritmo.

5.4.3 Trasformazioni lineari di intensità

In questo esperimento sono state modificate le stereocoppie di destra (I_R) utilizzando, per ogni canale, una trasformazione lineare del tipo $I'_R = I_R \cdot g + o$, dove il coefficiente g è detto *gain* e o *offset*. I test sono stati eseguiti con valori $o = 50$ e $g = 1.5$.

⇓ Dataset *"Venus"* (o = 50)

I_L I_R

⇓ Dataset *"Venus"* (g = 1.5)

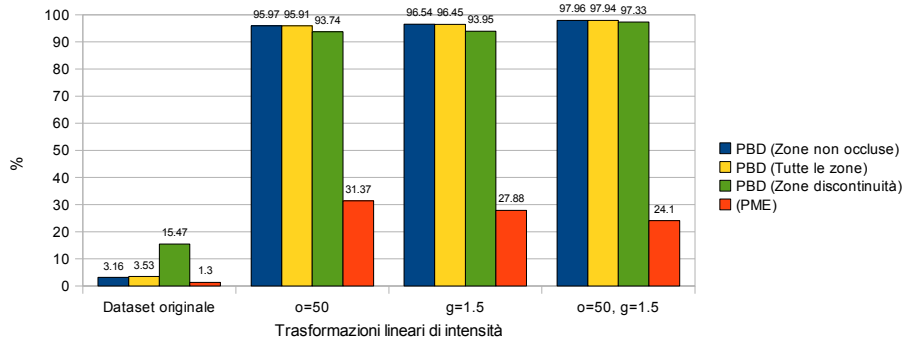
I_L I_R

⇓ Dataset *"Venus"* (o = 50, g = 1.5)

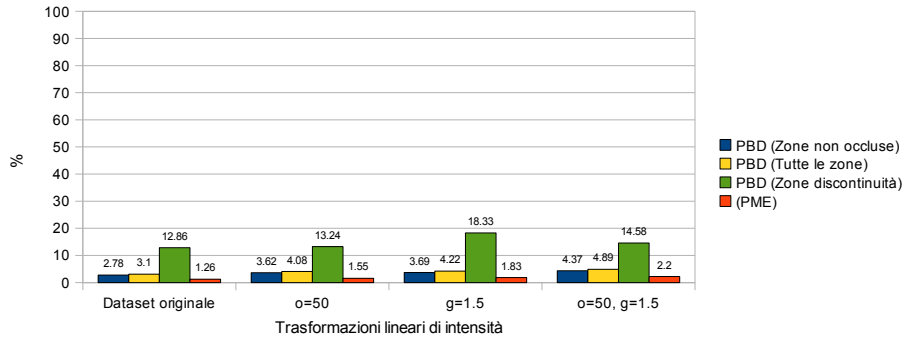
I_L I_R

Di seguito sono riportati i risultati ottenuti:

⇓ Dataset *"Venus"* (Metodo di matching originale)



⇓ Dataset *"Venus"* (Nuovo metodo di matching)



Grazie all'introduzione della nuova funzione di matching (4.3.2 nella pagina 61), invariante rispetto alle trasformazioni di tipo offset, è possibile apprezzare dei perfetti risultati con stereocoppie affette da questo problema. La cosa sorprendente è come l'invarianza sembra estendersi, più in generale, a tutte le trasformazioni lineari. Utilizzando la funzione di matching originale, i risultati sono in tutti i casi pessimi, con errori di matching PME molto elevati e mappe di disparità completamente errate.

5.4.4 Rumore

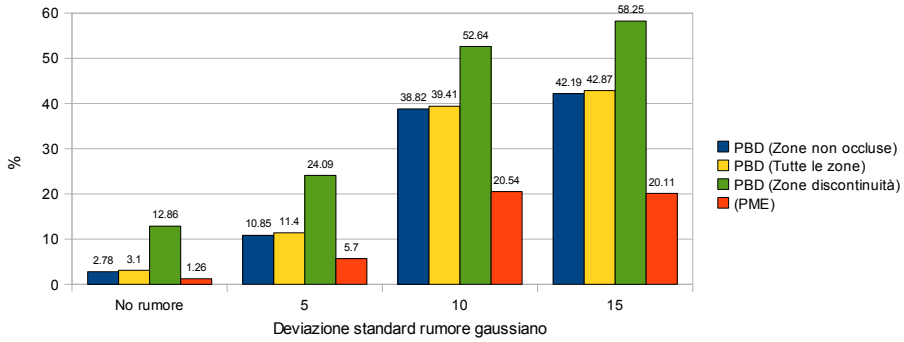
Entrambe le immagini dei dataset analizzati in precedenza (I_L e I_R) sono state corrotte con rumore di tipo Gaussiano. Questo tipo di rumore ha come funzione densità di probabilità una distribuzione normale, con media $\mu = 0$ e deviazione standard σ. In particolare sono stati condotti test con $\sigma = 5$, $\sigma = 10$ e $\sigma = 15$.

⇓ Dataset *"Venus"*

Particolare da I_L	Particolare da I_L
Senza rumore	Rumore $\sigma = 15$

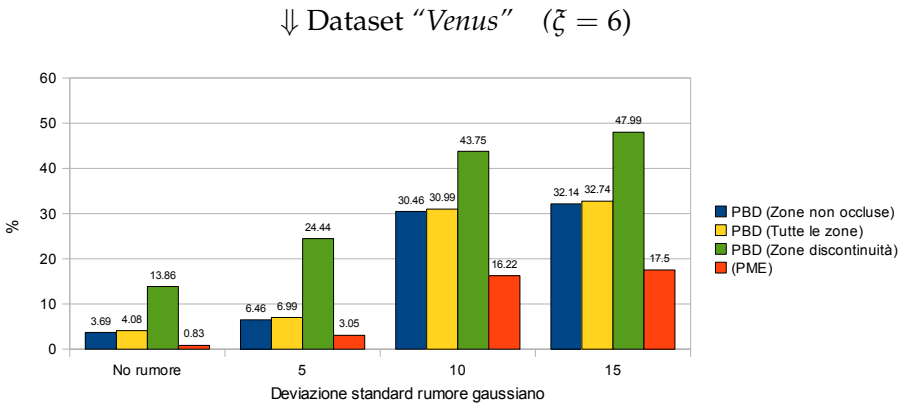
Ecco i risultati ottenuti:

⇓ Dataset *"Venus"* $(\xi = 3)$



Un importante considerazione riguarda la dimensione del supporto di ricerca adattivo; in precedenza è stato detto (5.3.2 nella pagina 88) che, aumentando la dimensione del supporto, aumenta di conseguenza il rapporto segnale/rumore.

Di seguito si riporta lo stesso test eseguito in precedenza, utilizzando però un supporto variabile di dimensioni doppie:



Grazie all'utilizzo di un modello basato su *Self-Organizing Maps*, *StereoSOM* risulta piuttosto robusto al rumore di fondo di entità contenuta ($\sigma = 5$), riuscendo a fornire risultati molto buoni in termini di errori di disparità. Con rumori di grande entità ($\sigma = 10$ e $\sigma = 15$) si hanno delle difficoltà nella fase di match ma, a patto di aumentare le dimensioni del supporto, si possono comunque apprezzare risultati esatti su più di due terzi della mappa di disparità.

5.4.5 Immagini non in forma standard

Un ultimo test riguarda le stereocoppie non in forma standard. È stata creata al proposito una coppia stereo[4] senza fare uso né di sistemi professionali né di post-elaborazione delle immagini: l'unico strumento utilizzato è una fotocamera digitale, spostata orizzontalmente a mano per simulare la distanza di baseline.

[4]Si ringrazia l'Ing. Laura Di Sieno per il supporto tecnico nella creazione del dataset "*Cuccioli*".

Riportiamo di seguito la coppia stereo in questione:

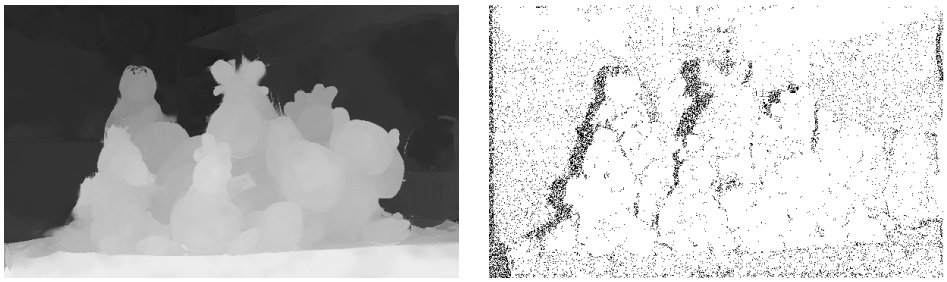
⇓ Dataset *"Cuccioli"*

I_L I_R

Per elaborare queste immagini è stato allargato il campo di ricerca delle corrispondenze da una linea orizzontale e (vincolo epipolare) a un insieme di linee ad una distanza verticale di ± 3 *pixel* da e. I risultati ottenuti sono i seguenti:

⇓ Dataset *"Cuccioli"*

Mappa di disparità *Mappa dei punti inconsistenti*



Si riportano tre ricostruzioni delle profondità, ottenute utilizzando tre differenti prospettive:

⇓ *Ricostruzione 3D profondità*

Pur non disponendo di dati reali, si nota subito la buona fattura della mappa di disparità fornita da *StereoSOM*. La mappa dei punti inconsistenti mostra una corretta detezione dei punti occlusi e di alcuni punti sullo sfondo, zona caratterizzata da un forte effetto *blur* (problema di messa a fuoco).

Concludendo, anche con immagini non in forma standard, scattate a istanti differenti e con mezzi di fortuna, *StereoSOM* si è dimostrato molto robusto: una qualità molto rara negli algoritmi di stereo matching denso.

Capitolo 6

Conclusioni

In questo elaborato è stata descritta e analizzata l'estensione di un algoritmo di corrispondenza stereo basato su reti neurali autorganizzanti. Il metodo finale, ribattezzato con il nome di *StereoSOM*, riesce a conciliare le caratteristiche dei moderni algoritmi di stereo matching locale con quelle dei modelli neurali basati su *Self-Organizing Maps*. La qualità del metodo è stata confermata dal sistema di valutazione Web del gruppo di ricerca in visione artificiale al Middelbury College, standard di fatto per le analisi quantitative degli algoritmi di stereo matching: *StereoSOM* è stato collocato al 21^o posto in classifica, su più di 70 dei migliori algoritmi sviluppati negli ultimi cinque anni.

La grande potenzialità di *StereoSOM*, tecnica impiegata solo recentemente in letteratura, è l'utilizzo di un supporto adattivo per la ricerca delle corrispondenze; in questo modo la finestra di correlazione riesce ad adattarsi ai bordi degli oggetti nella scena, evitando di includere superfici con disparità diverse.

Il secondo punto di forza è il controllo di consistenza stereo, una importante funzionalità che permette di rilevare in modo ottimale le occlusioni dovute alla geometria del sistema stereoscopico. Una grande sfida è stata quella di integrare il controllo di consistenza in un modello basato su *Self-Organizing M*aps; alla fine si è optato per una architettura composta da due reti neurali cooperanti che, oltre alla rilevazione e gestione delle occlusioni, riescono a fornire mappe di disparità per entrambe le immagini della coppia stereo.

I vantaggi legati all'utilizzo di un modello neurale sono sicuramente da ricercarsi nella robustezza al rumore e nella tolleranza a immagini in forma non standard, doti riscontrate poi negli esperimenti eseguiti.

Ma le potenzialità di *StereoSOM* non finiscono qui. Il metodo proposto ricerca le corrispondenze usando un approccio locale, lasciando quindi ampio spazio a future ottimizzazioni dei tempi di esecuzione, già comunque molto contenuti. Non è inoltre richiesta alcuna fase di estrazione delle feature o pre/post-processing dei dati.

Come tutti i metodi, anche quello proposto non è perfetto e presenta alcuni problemi. In primo luogo *StereoSOM* non utilizza la segmentazione delle immagini: questo fa si che le mappe di disparità siano caratterizzate da un lieve rumore dipendente dalle intensità luminose nelle immagini delle coppie stereo. Un altro problema legato alla non utilizzazione della segmentazione si rileva in prossimità delle discontinuità di profondità, dove i risultati quantitativi sono più scarsi se confrontati con le aree non discontinue.

In futuro sarebbe interessante valutare l'introduzione della segmentazione delle immagini all'interno dell'algoritmo, magari impiegando in modo alternativo le stesse reti *SOM* che ricercano le corrispondenze.

Concludendo, si ricorda che il lavoro proposto è stato presentato alla 15^a conferenza internazionale sull'analisi e l'elaborazione delle immagini *ICIAP 2009*, tenutasi nei giorni 8-11 settembre 2009 [55].

Appendice A.
Dataset utilizzati

In questa appendice sono contenuti i dataset in forma standard utilizzati per la valutazione quantitativa di *StereoSOM*. Tutte le immagini riportate di seguito, rilasciate pubblicamente sul Web, appartengono al *dipartimento di matematica e informatica* del *Middlebury College* [41] e sono state incluse in questo elaborato per questioni di completezza.

Lo schema utilizzato per visualizzare i dataset è il seguente:

	Immagine di sinistra I_L	Immagine di destra I_R
Mappa di disparità reale	Mappa binaria zone non occluse	Mappa binaria zone di discontinuità di profondità

Per alcuni dataset non sono disponibili le mappe binarie di zona, in questo caso lo schema seguito sarà il seguente:

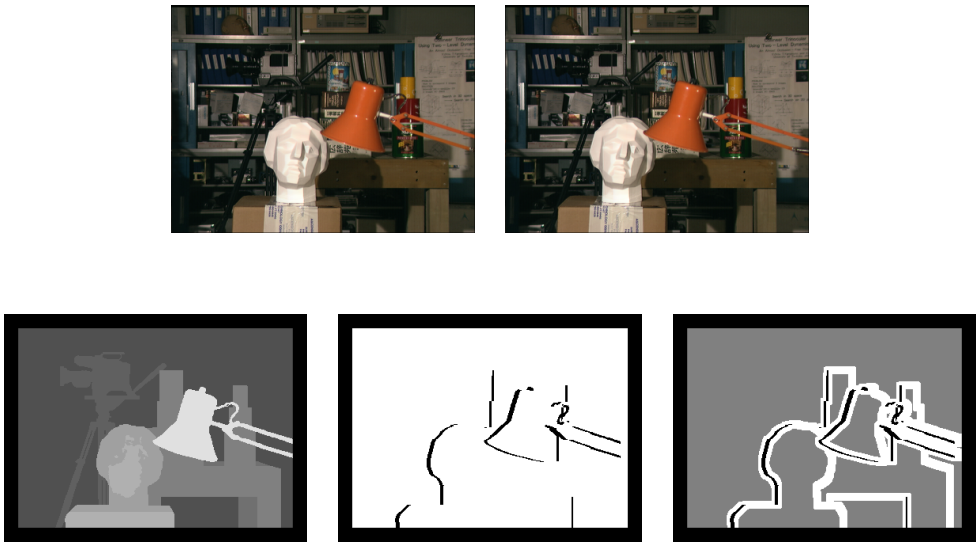
Immagine di sinistra I_L	Immagine di destra I_R	Mappa di disparità reale

In entrambi i casi le mappe di disparità reale sono riferite all'immagine di sinistra I_L.

Per ciascun dataset saranno riportati la risoluzione in pixel, l'intervallo di disparità utilizzato nei test e il fattore di normalizzazione impiegato per la produzio-

ne della mappa di disparità in scala di grigi [1]. Un breve commento seguirà ogni dataset e conterrà alcune indicazioni sulle caratteristiche delle immagini.

Dataset "*Tsukuba*"

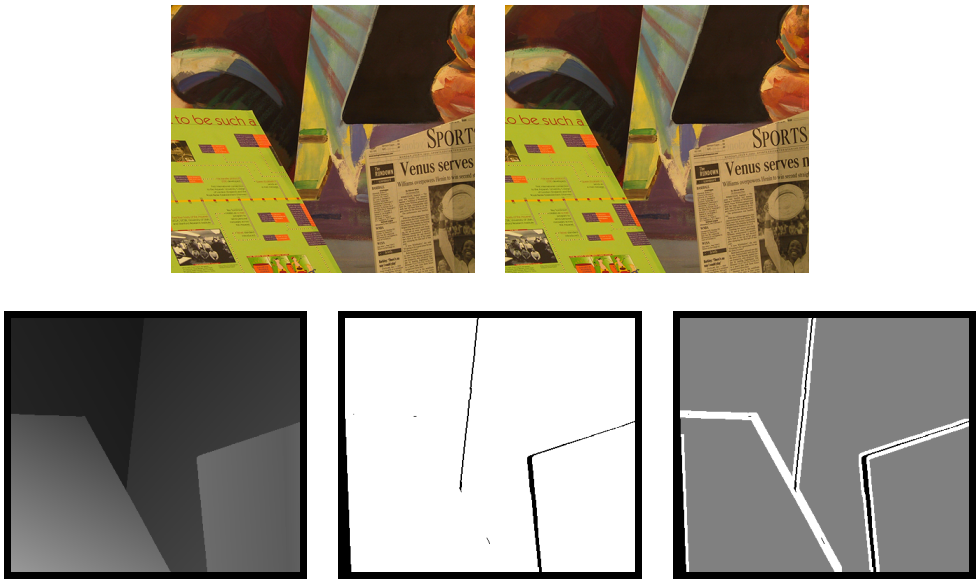


RISOLUZIONE: 384×288.
INTERVALLO DI DISPARITÀ: $[0, 15]$.
FATTORE DI NORMALIZZAZIONE: 16.

Il primo dataset, che prende il nome dall'Università di Tsukuba in Giappone, è stato in passato uno dei più noti ed impiegati nei lavori di stereo matching. Si può subito notare come la mappa delle disparità sia caratterizzata da valori di disparità interi. Questo test è adatto a valutare la qualità di algoritmi che producono mappe di disparità intere. Sono presenti zone di discontinuità di profondità, occlusioni e dettagli fini (si notino i bracci della lampada e le gambe della videocamera). Nelle mappe si può notare una zona di bordo nera, quest'area dovrà essere esclusa dalla valutazione degli errori.

[1] Le mappe di disparità sono infatti rappresentate come immagini in scala di grigi, dove i valori di grigio sono ottenuti moltiplicando i valori di disparità per un valore costante, detto appunto fattore di normalizzazione.

Dataset "*Venus*"



RISOLUZIONE: 434×383.
INTERVALLO DI DISPARITÀ: $[0, 19]$.
FATTORE DI NORMALIZZAZIONE: 8.

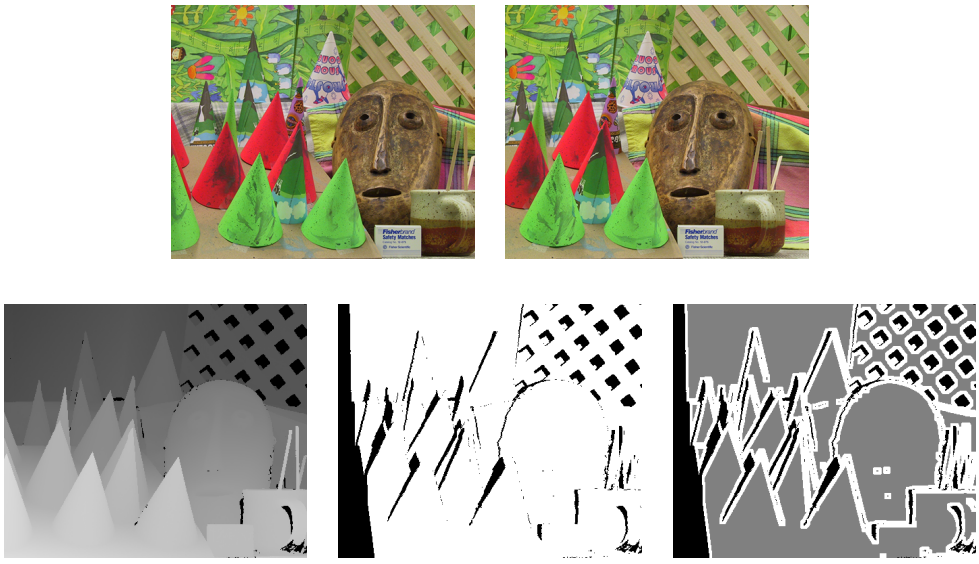
Il dataset "*Venus*" rappresenta una composizione di giornali. La mappa di disparità è a valori reali. Data la sua semplicità il dataset può essere utilizzato per valutare le capacità *sub-pixel* degli algoritmi di stereo matching. Le zone di discontinuità di profondità ed occlusione sono limitate ai bordi delle pagine di giornale e tutti gli oggetti nella scena sono approssimativamente fronto-paralleli al piano immagine. Sono tuttavia presenti ampie aree *low-texture*, dove molti algoritmi potrebbero generare ambiguità nell'assegnamento dei punti omologhi. Nelle mappe si può notare una zona di bordo nera, quest'area dovrà essere esclusa dalla valutazione degli errori.

Dataset "*Teddy*"

RISOLUZIONE: 450×375.
INTERVALLO DI DISPARITÀ: $[0, 59]$.
FATTORE DI NORMALIZZAZIONE: 4.

Il dataset "*Teddy*" ritrae una composizione di oggetti abbastanza complessa. La mappa di disparità reale, ottenuta con metodi attivi a luce strutturata, è a valori reali. Zone di discontinuità di profondità ed occlusione sono distribuite su gran parte delle immagini e si possono notare moltissimi oggetti fini. Sono presenti zone *low-texture* (alla destra dell'orsetto), pattern periodici (alla sinistra dell'orsetto) e molte aree occluse. Le zone di bordo escono dal campo visivo ma sono comunque incluse nella valutazione, per questo motivo gli algoritmi dovranno trovare un modo per approssimare i valori di disparità in queste aree.

Dataset "*Cones*"



RISOLUZIONE: 450×375.
INTERVALLO DI DISPARITÀ: $[0, 59]$.

FATTORE DI NORMALIZZAZIONE: 4.

Le stesse considerazioni fatte per il dataset "*Teddy*" sono applicabili al dataset "*Cones*", i due dataset sono infatti molto simili dal punto di vista degli ostacoli alla fase di matching.

Dataset "*Reindeer*"

RISOLUZIONE: 447×370.
INTERVALLO DI DISPARITÀ: $[0, 77]$.
FATTORE DI NORMALIZZAZIONE: 3.

Questo dataset è stato inserito in aggiunta a quelli utilizzati nel sistema di valutazione Web del Middlebury College. Si possono fare considerazioni analoghe a quelle fatte per i dataset "*Teddy*" e "*Cones*". Non si dispone per questo dataset delle mappe binarie per le zone occluse e di discontinuità di profondità.

Dataset "*Dolls*"

RISOLUZIONE: 463×370.
INTERVALLO DI DISPARITÀ: $[0, 77]$.
FATTORE DI NORMALIZZAZIONE: 3.

Questo dataset risulta più complesso dei precedenti, oltre alle problematiche già discusse vi è una estrema variabilità nelle superfici ed una massiccia presenza di discontinuità di profondità, occlusioni e distorsioni fotometriche.

Dataset "*Aloe*"

RISOLUZIONE: 427×370.
INTERVALLO DI DISPARITÀ: $[0, 77]$.
FATTORE DI NORMALIZZAZIONE: 3.

Il dataset "*Aloe*" sembra a prima vista semplice, in realtà oltre a presentare le problematiche dei dataset precedenti è visibile sullo sfondo un'ampia superficie caratterizzata da pattern periodici.

Appendice B.
Self-Organizing Maps[2]

Nell'ambito dell'apprendimento automatico, un importante sottoinsieme di metodi sono basati su *reti neurali artificiali* (d'ora in poi chiamate solo *reti neurali*), particolari modelli matematici composti di neuroni artificiali e ispirati all'architettura neurale dei cervelli biologici.

Come per quasi tutti gli algoritmi di apprendimento, anche tra i modelli di reti neurali è possibile definire una ulteriore distinzione tra tecniche di apprendimento *supervisionato* e *non supervisionato*[3].

Le prime prevedono una fase di addestramento mediante presentazione di esempi da parte di un "esperto" esterno e, al fine di un corretto funzionamento, necessitano che gli esempi siano completi di dati di verità. Queste reti neurali sono state per lo più impiegate in contesti di *classificazione automatica*, dove è semplice disporre di esempi dotati di dati di verità e quello che si vuole ottenere è la capacità di classificare correttamente nuovi dati.

Tuttavia non è possibile risolvere tutti i problemi con modelli di apprendimento supervisionato, esistono casi in cui è necessario agire diversamente; ne elenchiamo alcuni:

Clustering in alcuni contesti è necessario analizzare i dati al fine di selezionare e raggruppare elementi omogenei. Nel caso in cui non si conosca a priori la struttura dei dati, il modello di apprendimento dovrà riuscire a comprenderla in modo autonomo.

[2] Tratto da "*An introduction to Neural Networks*" [32].

[3] In realtà esistono algoritmi che sfruttano entrambe le tipologie di apprendimento descritte. Questi metodi sono detti di apprendimento *semi-supervisionato* (*SSL*) ma la loro trattazione va oltre gli scopi di questo elaborato.

Vector quantization problema legato alla quantizzazione di dati rappresentati su spazi continui. L'input è un vettore n-dimensionale e l'output voluto è una sua ottimale rappresentazione discreta.

Riduzione della dimensionalità (Dimensionality reduction) l'input di alcuni sistemi può richiedere dimensionalità minore rispetto ai dati disponibili. In questi casi il modello di apprendimento dovrà imparare un mapping ottimale che preservi l'informazione nei dati pur riducendone la dimensionalità.

Estrazione delle feature (*feature extraction*) non conoscendo a priori il tipo dei dati, può essere necessario svolgere autonomamente il processo di estrazione delle feature. Questo coincide spesso con il problema di riduzione della dimensionalità descritto in precedenza.

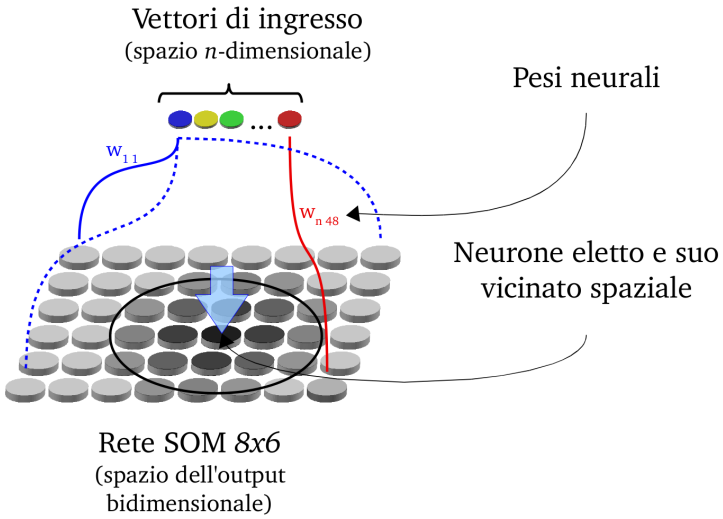
Nei casi sopra citati è necessario impiegare metodi che non presuppongono la presenza di un "esperto" esterno, ma che siano in grado di apprendere autonomamente dagli esempi presentati.

I metodi di apprendimento non supervisionato basati su reti neurali, pur sfruttando spesso il concetto di competizione globale tra neuroni, sono stati al centro di studi scientifici per molti anni (si studiano tutt'oggi in alcuni contesti). Per questo motivo esiste una grande varietà di modelli neurali autorganizzanti; per una trattazione generale e una analisi della bibliografia essenziale si consiglia la lettura di [32].

Le reti neurali brevemente descritte in questa appendice portano il nome di *Self-Organizing Maps* (dette anche *SOM* o *mappe di Kohonen*) e sono da considerarsi una estensione delle più generali *reti neurali competitive*. Come riferimento storico è interessante ricordare che fu Kohonen [34, 35], ispirato dal funzionamento di alcune aree della corteccia celebrale umana, il primo a proporre un modello neurale di questo tipo; solo successivamente le sue reti furono ampiamente studiate e generalizzate nella classe delle reti neurali competitive.

Descrizione del modello Le *SOM* sono reti neurali *feedforward* composte da un singolo strato di neuroni (detti anche *nodi* o *unità*) di output, ordinati spesso in una griglia bidimensionale. La particolare disposizione dei neuroni nella griglia è dipendente dal problema specifico che la rete si presterà a risolvere ed è la principale caratteristica dalla quale dipenderà il concetto di *vicinato neurale*.

Figura 1: Esempio di Self-Organizing Map 8x6 con input n-dimensionale.



Quando i dati di addestramento vengono presentati alla rete, i pesi dei neuroni si adattano in modo che pattern "vicini" tra di loro nello spazio dell'input (spesso con dimensionalità maggiore di 2) vengano mappati su neuroni "vicini" nello spazio bidimensionale dell'output. Notare come la rete effettui una riduzione di dimensionalità mappando dati da uno spazio n-dimensionale ad uno spazio euclideo a due dimensioni ma preservando le proprietà topologiche dello spazio degli ingressi[4].

Come mostrato in Figura 1, rappresenteremo la rete come una mappa di neuroni di output o connessi mediante pesi neurali w_{io} agli ingressi i. Il valore iniziale dei pesi neurali dipende dal particolare compito svolto dalla rete, in alcuni casi i pesi sono inizialmente nulli, in altri casi possono essere inizializzati in modo random o utilizzando tecniche che permettono di approssimare dall'inizio i valori ottenibili dopo una lunga fase di convergenza della rete, ottimizzando così i tempi di elaborazione.

Si supponga che al tempo t venga presentato alla rete il vettore di input $\mathbf{x}(t)$. Mediante l'equazione:

[4]Esistono anche reti di Kohonen che prevedono neuroni disposti in uno spazio con più di due dimensioni o spazi non euclidei. Tuttavia questi modelli sono utilizzati in contesti specifici mentre l'utilizzo di spazi euclidei a due dimensioni resta ancora la scelta più generale.

$$k : \|\mathbf{w}_k(t) - \mathbf{x}(t)\| \leq \|\mathbf{w}_o(t) - \mathbf{x}(t)\| \ , \forall o$$

si seleziona il peso neurale con la minore distanza euclidea[5] dall'input $\mathbf{x}(t)$ e, conseguentemente, si elegge il neurone vincente k (detto anche *Best Matching Unit* o *BMU*). Successivamente si procede all'aggiornamento dei pesi neurali, processo esteso al neurone eletto e a tutto il suo vicinato spaziale. Allo scopo viene utilizzata la seguente regola di apprendimento:

$$\mathbf{w}_o(t+1) \leftarrow \boldsymbol{w}_o(t) + \eta(t)\, h(o,k)\, (\mathbf{x}(t) - \mathbf{w}_o(t)) \ , \forall o$$

dove

$$h(o,k) = \exp\left(-\frac{(o_x - k_x)^2 + (o_y - k_y)^2}{2\sigma^2}\right)$$

è una funzione gaussiana che decresce con la distanza dal neurone eletto k e tale che $h(k,k) = 1$ (esempio in Figura 2 a fronte). $\eta(t)$ è detto *learning rate* e regola l'entità dell'aggiornamento neurale, solitamente è una funzione monotona decrescente con il tempo t del tipo:

$$\eta(t) = \alpha\frac{1}{t} \quad \text{o} \quad \eta(t) = \beta - \alpha t$$

Il processo appena descritto verrà ripetuto, per ogni vettore in input, un numero di cicli variabile[6].

Come la gran parte dei modelli di apprendimento basati su reti neurali, le *SOM* prevedono due principali modalità di funzionamento:

Apprendimento Si costruisce la mappa dei pesi neurali mediante un processo competitivo. In questa fase si presentano in ingresso alla rete un numero elevato di vettori. La scelta ottimale consiste in un insieme di vettori che copra omogeneamente tutte le tipologie di input: in questo modo

[5] La metrica basata su distanza euclidea è una delle più generali, ma non l'unica impiegata nel processo competitivo di elezione del neurone vincente. La distanza euclidea risulta comunque una metrica robusta nei confronti di input non normalizzati.

[6] Il numero di cicli dipende spesso dall'applicazione specifica e dalla cardinalità dell'insieme dei vettori in ingresso.

Figura 2: Esempio di funzione gaussiana che decresce con l'aumentare della distanza dal neurone eletto.

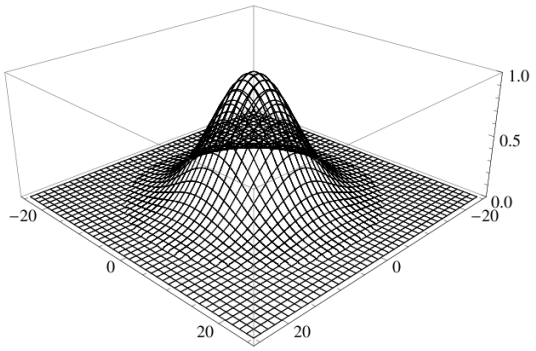
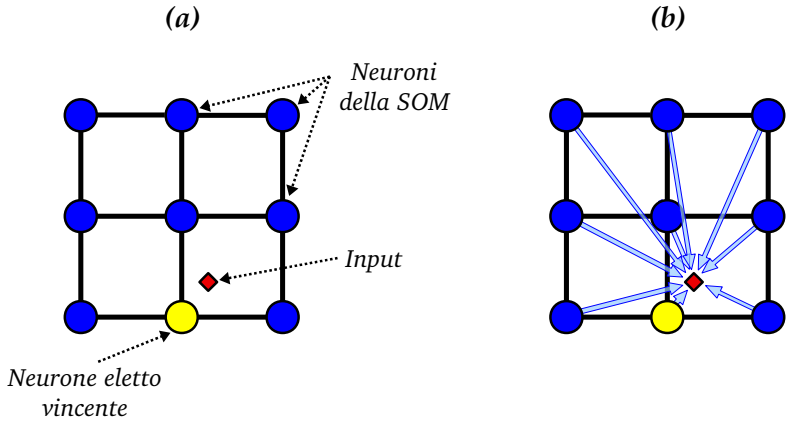


Figura 3: ***(a)***: Sezione di rete *SOM* e neurone eletto vincitore. ***(b)***: Meccanismo di aggiornamento nelle reti *SOM*, i pesi dei neuroni nel vicinato del nodo vincente si avvicinano al valore in input.



la rete potrà "carpire" al meglio il significato dei dati e sarà in grado di ridurne la dimensionalità senza causare troppe perdite di informazione.

Mapping Sono presentati alla rete nuovi vettori d'ingresso. Durante questa fase non avvengono generalmente aggiornamenti neurali, ma si eleggerà di volta in volta il neurone vincente utilizzando la metrica basata su distanza euclidea.

Dalla seconda fase di mapping è evidente come le reti *SOM* possano essere impiegate per la visualizzazione e il clustering dei dati ad alta dimensionalità, applicazione estremamente comoda in contesti in cui si dispone di molti dati ma si hanno poche informazioni sul loro significato.

È curioso discutere su come si possa dimostrare qualitativamente il funzionamento di una *SOM*, infatti durante la fase di apprendimento i pesi di tutto il vicinato vengono spostati nella stessa direzione dell'input che ha attivato il neurone vincente (Figura 3 nella pagina precedente): elementi simili tendono quindi ad attivare neuroni vicini. Da questo fatto si può ipotizzare che la rete crei una mappa semantica ordinata, dove campioni simili vengono mappati in specifiche aree della rete.

Indice analitico

Riferimenti bibliografici

[1] TRUCCO E., VERRI A., ***Introductory Techniques for 3-D Computer Vision***, Prentice Hall, (1998).

[2] ***Camera Calibration Toolbox for Matlab***: `www.vision.caltech.edu/bouguetj/calib_doc/`

[3] MATTOCCIA S., ***Introduzione alla Visione Stereo***, Università di Bologna, Dipartimento di Elettronica Informatica e Sistemistica (DEIS), (2007).

[4] Faugeras O., ***Three-Dimensional Computer Vision: A Geometric Viewpoint***, The MIT Press, (1993).

[5] BERTOZZI M., BROGGI A., ***GOLD: a Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection***, IEEE Trans. on Image Processing, 7(1):62-81, (1998).

[6] ZANIN M., ***Localization of ahead vehicles with on-board stereo cameras***, In Proceedings of the 14th International Conference on Image Analysis and Processing, ICIAP 2007, Modena, Italy, (2007).

[7] ***Cognex Corporation***: `www.cognex.com`

[8] CHANG K., BOWYER K., FLYNN P., ***Face recognition using 2D and 3D facial data***, Proc. Multimodal User Authentication Workshop, Santa Barbara, (2003).

[9] LEE H.Y., KIM, T., PARK W. AND LEE H. K., ***Extraction of Digital Elevation Models from satellite stereo images through stereo matching based on Epipolarity and Scene Geometry***, Image and Vision Computing, 21(9):789-796, (2003).

[10] MUÑOZ-SALINAS R., AGUIRRE E., SILVENTE M. G., ***People detection and tracking using stereo vision and color***. Image Vision Comput. 25(6): 995-1007, (2007).

[11] RASPANTI M., BINAGHI E., GALLO I., MANELLI A., ***A vision-based, 3D reconstruction technique for scanning electron microscopy: direct comparison with atomic force microscopy***, Microsc Res Tech., 67(1):1-7, (2005).

[12] LIM H.S., BINFORD T.O., ***Stereo Correspondence: A Hierarchical Approach***, Proc. Image Understanding Workshop, Vol. 1, pp. 234-241, (1987).

[13] KLAUS A., SORMANN M., KARNER K., ***Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure***, Int. Conf. Pattern Recognition, (2006).

[14] BOYKOV Y., VEKSLER O., ZABIH R., ***Fast approximate energy minimization via graph cuts***, IEEE Trans. on PAMI, (2001).

[15] KOLMOGOROV V., ZABIH R., ***Computing visual correspondence with occlusions using graph cuts***, In Eighth Intern. Conf. on Computer Vision, (2001).

[16] BOBICK A. F., INTILLE S. S., ***Large occlusion stereo***, Int Journ. of Computer Vision, (1999).

[17] GEMAN S., GEMAN D., ***Stochastic relaxation, gibbs distributions, and the bayesian restoration of images***. IEEE Trans. on PAMI, (1984).

[18] BARNARD S. T., ***Stochastic stereo matching over scale***, Int Journ. of Computer Vision, (1989).

[19] ROY S., COX I., ***A maximum-flow formulation of the n-camera stereo correspondence problem***, Proc. Int. Conf. Computer Vision, pp. 492–499, (1998).

[20] SCHARSTEIN D., SZELISKI R., ***Stereo matching with non linear diffusion***, Int. Jorn. of Computer Vision, (1998).

[21] HANNAH M. J., ***Computer Matching of Areas in Stereo Images***. PhD thesis, Stanford University, (1974).

[22] KANADE T., ***Development of a video-rate stereo machine***, In Image UnderstandingWorkshop, pages 549–557, Monterey, CA, Morgan Kaufmann Publishers, (1994).

[23] MARR D., POGGIO T., ***Cooperative computation of stereo disparity***, Science, 194:283–287, (1976).

[24] NISHIHARA H. K., ***Practical real-time imaging stereo matcher***, Optical Engineering, 23(5):536–545, (1984).

[25] OKUTOMI M., KANADE T., ***A locally adaptive window for signal matching***, IJCV, 7(2):143–162, (1992).

[26] GRIMSON W. E. L., ***Computational experiments with a feature based stereo algorithm***, IEEE TPAMI, 7(1):17–34, (1985).

[27] POLLARD S. B., MAYHEW J. E. W., FRISBY J. P., ***PMF: A stereo correspondence algorithm using a disparity gradient limit***, Perception, 14:449–470, (1985).

[28] PRAZDNY K., ***Detection of binocular disparities***, Biological Cybernetics, 52(2):93–99, (1985).

[29] TIAN Q., HUHNS M. N., ***Algorithms for subpixel registration***. CVGIP, 35:220–233, (1986).

[30] COCHRAN S. D., MEDIONI G., ***3-D surface description from binocular stereo***, IEEE TPAMI, 14(10):981–994, (1992).

[31] FUA P., ***A parallel stereo algorithm that produces dense depth maps and preserves image features***, Machine Vision and Applications, 6:35–49, (1993).

[32] KROSE B., VAN DER SMAGT P., ***An Introduction to Neural Networks***, (1996).

[33] KOHONEN T., ***Self-Organizing Maps***, Springer Series in Information Sciences. Springer. Terza ed, (2001).

[34] KOHONEN, T., ***Self-organized formation of topologically correct feature maps***, Biological Cybernetics, 43, 59-69, (1982).

[35] KOHONEN, T., ***Self-Organization and Associative Memory***, Berlin: Springer-Verlag, (1984).

[36] KOHONEN T., SIMULA O. AND KANGAS, J., ***Self-organizing maps: optimization approaches***, Artificial neural networks: proceedings of the 1991 International Conference on Artificial Neural Networks (ICANN-91), Espoo, Finland, 24-28, North-Holland, (1991).

[37] SCHARSTEIN D., SZELISKI R., ***A taxonomy and evaluation of dense two-frame stereo correspondence algorithms***, International Journal of Computer Vision, 47(1/2/3):7-42, (2002).

[38] DI STEFANO L., MARCHIONNI M., MATTOCCIA S., ***A Fast Area-Based Stereo Matching Algorithm***, Image and Vision Computing, Vol. 22, No. 12, pp 983-1005, Elsevier , (2004).

[39] TOMBARI F., MATTOCCIA S., DI STEFANO L., ***Segmentation-based adaptive support for accurate stereo correspondence***, D. Mery, L. Rueda (eds.) PSIVT 2007. LNCS, vol. 4872, pag. 427–438. Springer, Heidelberg, (2007).

[40] VENKATESH Y. V., RAJA S. KUMAR, AND KUMAR A. JAYA, ***On the Application of a Modified Self-Organizing Neural Network to Estimate Stereo Disparity***, IEEE transactions on image processing, vol. 16, no. 11, (2007).

[41] ***Middlebury Stereo Vision Page***:
`http://vision.middlebury.edu/stereo/`

[42] KOSCHAN A., RODEHORST V., SPILLER K., ***Color stereo vision using hierarchical block matching and active color illumination***, Proc. of 13^{TH} international conference on pattern recognition, ICPR '96, Vienna, Austria, pp. 835-839, (1996).

[43] HUA X., YOKOMICHI M., KONO M., ***Color stereo matching based on self-organization neural networks***, MWSCAS '04, the 2004 47th Midwest Symposium on. 25/08/2004; On pages: I- 213-16 vol.1, IEEE Transactions on Circuits and Systems, (2004).

[44] KANADE T., OKUTOMI M., ***Stereo matching algorithm with an adaptive window: theory and experiment***, IEEE Trans. PAMI 16(9), 920–932, (1994).

[45] BOYKOV Y., VEKSLER O., ZABIH R., ***A variable window approach to early vision***, IEEE Trans. PAMI 20(12), 1283–1294, (1998).

[46] HIRSCHMULLER H., INNOCENT P., GARIBALDI J., ***Real-time correlation-based stereo vision with reduced border errors***, Int. Jour. Computer Vision (IJCV) 47(1-3), (2002).

[47] XU Y., WANG D., FENG T., SHUM H., ***Stereo computation using radial adaptive windows***, Proc. Int. Conf. on Pattern Recognition (ICPR 2002), vol. 3, pp. 595–598, (2002).

[48] VEKSLER O., ***Fast variable window for stereo correspondence using integral images***, Proc. Conf. on Computer Vision and Pattern Recognition (CVPR 2003), pp. 556–561, (2003).

[49] GONG M., YANG R., ***Image-gradient-guided real-time stereo on graphics hardware***, Proc. 3D Dig. Imaging and modeling (3DIM), Ottawa, Canada, pp. 548–555, (2005).

[50] YOON K.J., KWEON I.S., ***Adaptive support-weight approach for correspondence search***, IEEE Trans. PAMI 28(4), 650–656, (2006).

[51] WANG L., GONG M.W., GONG M.L., YANG R.G., ***How far can we go with local optimization in real-time stereo matching***, Proc. Third Int. Symp. on 3D Data Processing, Visualization, and Transmission (3DPVT 2006), pp. 129–136, (2006).

[52] WU Q. X., MCNEILL S. J., , PAIRMAN D., ***Fast algorithms for correlation-relaxation technique to determine cloud motion fields***, Digital Image Computing: Techniques and Applications, (Brisbane, Australia), pp. 330–335, (1995).

[53] MEIJERING E., ***A chronology of interpolation: from ancient astronomy to modern signal and image processing***, proceedings of the IEEE 90 (3): 319–342, (2002).

[54] SCHRAUDOLPH N. N., ***A fast, compact approximation of the exponential function***, Neural Computation, v.11 n.4, p.853-862, (1999).

[55] VANETTI M., GALLO I., BINAGHI E., ***Dense Two-Frame Stereo Correspondence by Self-organizing Neural Network***, Image Analysis and Processing – ICIAP 2009, pag. 1035-1042, Springer Berlin / Heidelberg, (2009).

[56] ***GNU Octave:*** `www.octave.org`

[57] ***ImageJ:*** `http://rsbweb.nih.gov/ij/`