

Università degli studi di Roma
“La Sapienza”
Anno Accademico 2002/2003

Corso di Laurea in Ingegneria Informatica
Nuovo Ordinamento

Prova Finale

Progetto Interno

Tecniche
di
Correlazione
per la
Stereo Visione

relatore Professor Luca Iocchi

autore Domenico Bloisi

<< C'è un modo solo di vedere le cose, finché qualcuno non ci mostra
come guardare con altri occhi >>

PABLO PICASSO

Il presente lavoro nasce da un'idea del Professor Luca Iocchi che ringrazio per l'opportunità, che egli mi ha offerto, di approfondire una affascinante tematica qual è la stereo visione, oggetto dell'odierna ricerca, e per la grande disponibilità che egli ha dimostrato durante tutto il processo di sviluppo di questo progetto.

Indice dei contenuti

Introduzione	pagina 1
Capitolo I Analisi del problema	
I.1 Che cosa si vuole realizzare	pagina 5
I.2 La misura di prestazione	pagina 6
I.3 L'input	pagina 8
I.4 L'output	pagina 9
Capitolo II La Corrispondenza	
II.1 Le tecniche di corrispondenza	pagina 11
II.1.1 I vincoli locali	pagina 11
II.1.2 I vincoli globali	pagina 11
II.2 I metodi locali	pagina 12
II.2.1 Block matching	pagina 12
II.2.1 A La correlazione	pagina 13
II.2.1 B Le metriche legate al rango	pagina 17
II.2.2 Gradient Methods	pagina 18
II.2.3 Feature Matching	pagina 18
II.3 I metodi globali	pagina 19
II.4 Complessità computazionali	pagina 19
Capitolo III Le Tecniche di Correlazione	
III.1 La funzione di correlazione	pagina 21
III.2 La gestione delle oclusioni	pagina 21
III.3 I valori di soglia	pagina 22
III.4 L'intorno di un pixel	pagina 23
III.5 L'unione di più metodi	pagina 24

Capitolo IV Implementazione e risultati sperimentali

IV.1 L'algoritmo di correlazione	pagina 25
IV.2 Il test dell'algoritmo	pagina 27
IV.3 Specifiche tecniche	pagina 29
IV.4 Il metodo SSD puro (immagini campione)	pagina 30
IV.5 Il metodo SSD con soglia interna	pagina 36
IV.6 Il metodo SSD con distanza di picco	pagina 42
IV.7 Il metodo SSD con soglia interna + distanza di picco	pagina 48
IV.8 Il metodo SSD unito con il metodo SAD	pagina 54
IV.9 Risultati per il metodo SSD con D_{MAX} variabile	pagina 60
IV.10 Conclusioni	pagina 66

Appendice A La stereo visione

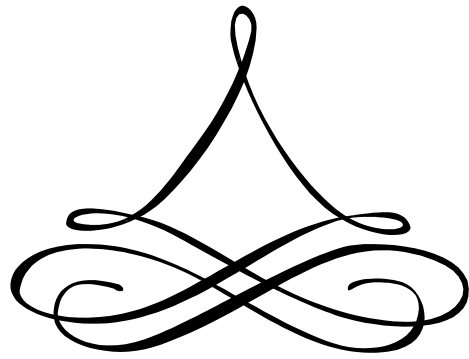
A.1 Che cos'è la stereo visione	pagina 67
A.2 Le occlusioni	pagina 68
A.3 La ricostruzione della profondità	pagina 69
A.3.1 La Calibrazione	pagina 69
A.3.2 La Corrispondenza	pagina 70
A.3.3 La Triangolazione	pagina 70
A.4 La Geometria Epipolare	pagina 71
A.5 Il modello Lambertiano	pagina 72

Appendice B Il Codice

B.1 La classe Correlazione.java	pagina 73
---	-----------

Bibliografia	pagina 85
------------------------	-----------

Introduzione



Introduzione

L'analisi stereo

L'analisi stereo è il processo di misurazione della distanza da un oggetto, basato sul confronto di due o più immagini dell'oggetto stesso ottenute simultaneamente. In altre parole, i sistemi di stereo visione determinano la profondità di una scena attraverso due o più immagini di tale scena ottenute allo stesso tempo, ma da punti di vista lievemente differenti. Questo processo dovrebbe essere ben noto a qualsiasi essere umano. Infatti, l'effetto di profondità, ossia la percezione della terza dimensione, che avvertiamo attraverso i nostri occhi, deriva dal fatto che i due bulbi oculari hanno i loro assi ottici distanti circa 65 mm e forniscono due immagini leggermente diverse degli oggetti che, sommando i loro effetti, procurano il senso della profondità (figura 1).

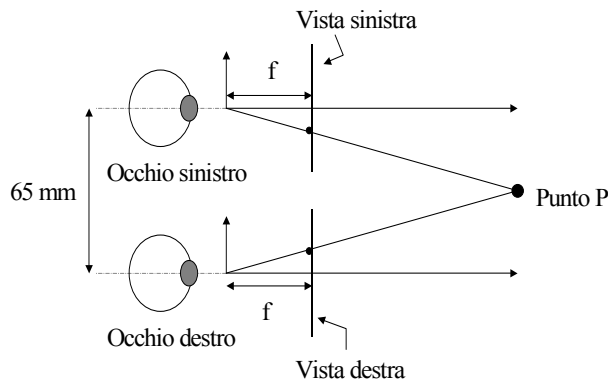


Figura 1 – Modello semplificato della vista umana

Il problema fondamentale nell'analisi stereo è quello di trovare la corrispondenza tra gli elementi delle varie immagini disponibili. Una volta che tale corrispondenza è stata scoperta, la distanza dall'oggetto può essere ottenuta tramite l'ottica geometrica. Se osserviamo con due telecamere lo stesso punto P nello spazio, a tale punto corrisponderanno le coordinate (x, y) sull'immagine della prima camera e le coordinate (x', y') sull'immagine prodotta dalla seconda camera (figura 2).

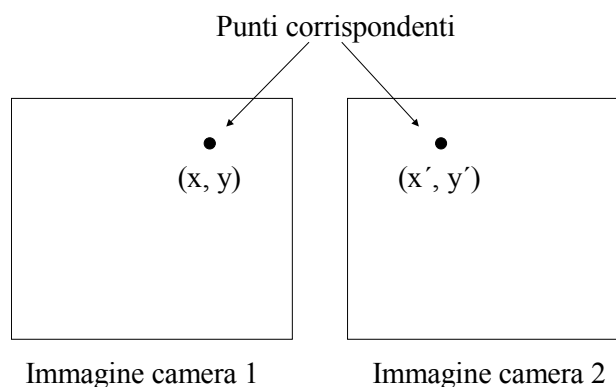


Figura 2 – Punti corrispondenti in una coppia stereo di immagini

La coppia di locazioni (x, y) e (x', y') è unica. Proprio perché tale coppia è unica, se si riescono a trovare le due locazioni che corrispondono allo stesso identico punto nello spazio, allora è possibile risalire alle coordinate tridimensionali di detto punto.

Le tecniche di correlazione sono parte dell'analisi stereo e cercano proprio di trovare punti corrispondenti fra le diverse immagini massimizzando una qualche misura di similarità.

La correlazione area-based

La correlazione area-based è un metodo di correlazione locale basato sul matching di due aree di pixels (una per ogni immagine). Per prima cosa, viene scelto un pixel p di coordinate (i, j) nella prima immagine. Questo pixel sarà il baricentro di una finestra di ricerca di dimensioni $N \times N$ (finestra di riferimento). Trovare il pixel della seconda immagine corrispondente a p significa confrontare la finestra di riferimento con alcune finestre prese dalla seconda immagine (frames di ricerca). Una volta trovato il frame di ricerca che più assomiglia alla finestra di riferimento, il baricentro di tale frame sarà il pixel corrispondente a p .

Visto in termini assolutamente generali, il problema ha una complessità molto elevata ed è intrattabile: esso va, dunque, rilassato. L'introduzione del vincolo epipolare è un'ottima soluzione al fine di rilassare il problema iniziale. Questo vincolo impone che, nelle immagini stereo, punti corrispondenti debbano sempre giacere su linee epipolari. Queste linee corrispondono all'intersezione di un piano epipolare (il piano attraverso un punto della scena ed i punti focali dei due obiettivi) con i piani di immagine sinistro e destro. La ricerca delle corrispondenze, quindi, deve essere effettuata solo su tali linee epipolari e non sull'intera immagine (si veda la figura 3).

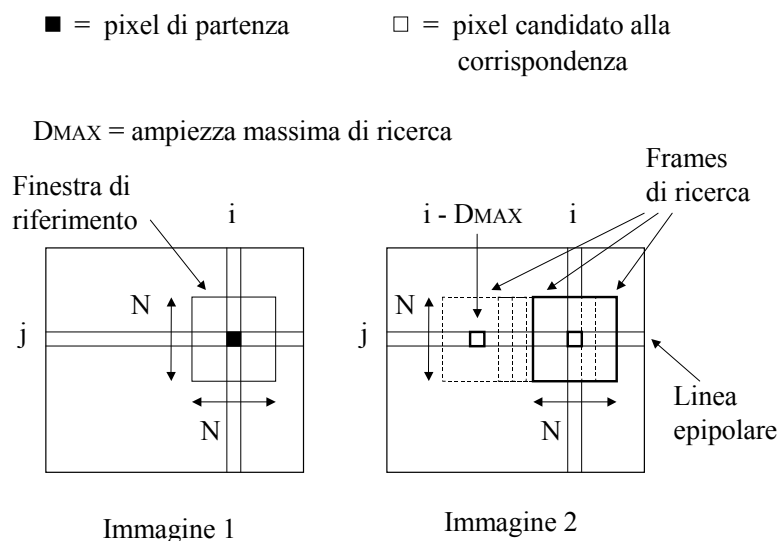


Figura 3 – La correlazione area-based con vincolo epipolare

L'utilizzo del vincolo epipolare riduce il problema, inizialmente bidimensionale, ad uno monodimensionale, poiché il pixel obiettivo nella seconda immagine deve trovarsi sulla linea orizzontale di ordinata uguale a quella del pixel di partenza (linea epipolare).

Un modo semplicistico di trovare la disparità è quello di cercare lungo le linee epipolari alla ricerca del massimo della funzione di correlazione, tuttavia si può fare molto meglio adoperando dei vincoli aggiuntivi:

- 1) Unicità: un punto in una immagine può corrispondere al massimo ad un punto nell'altra immagine.

- 2) Continuità a tratti delle superfici nella scena: il fatto che il mondo solitamente è continuo a tratti implica che punti vicini di una scena abbiano valori di profondità, e quindi di disparità, simili.

Oltre a questi vincoli, si possono trovare funzioni euristiche atte a migliorare l'algoritmo, come ad esempio l'introduzione di valori di soglia o l'analisi dell'andamento dei vari picchi che contraddistinguono la funzione di correlazione (figura 4).

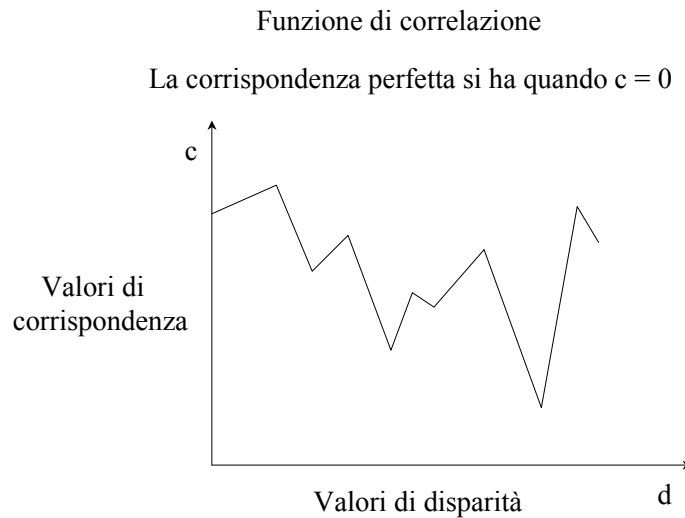


Figura 4 – Grafico della funzione di correlazione

Obiettivi

Il presente lavoro ha per fine l'implementazione e l'analisi di diverse strategie di correlazione di aree di immagini stereo per risolvere il problema delle corrispondenze. L'obiettivo principale è quello di proporre delle strategie innovative e di confrontarle con le strategie di stereo visione note. Ulteriori scopi di questo progetto sono:

- 1) la comparazione tra un algoritmo di ricerca basato su frames quadrati ed uno basato su frames rettangolari
- 2) l'implementazione di un algoritmo con valori $M \times N$ (con M ampiezza ed N altezza) delle finestre di ricerca che siano dinamici, ovvero che varino in funzione del valore di texture dell'immagine
- 3) l'implementazione di un algoritmo con ampiezza di ricerca D_{MAX} e valori $M \times N$ dinamici, che varino in base alla porzione di immagine che si sta analizzando.

La struttura portante del progetto è l'analisi (complessità asintotica, tempo di esecuzione, risorse utilizzate, completezza, ottimalità, etc.) di tali algoritmi, nel tentativo di proporre un metodo innovativo per risolvere il problema delle corrispondenze nella stereo visione.

Il confronto con i metodi di correlazione già esistenti sarà effettuato sfruttando alcune immagini campione, tra le quali una coppia di immagini fornita dall'Università di Tsukuba che rappresenta un classico bench-mark per gli algoritmi di corrispondenza.

Input ed output

Per diminuire la complessità del problema, si sfrutterà il già citato vincolo epipolare, ovvero si supporrà sempre di lavorare su due immagini prese con una coppia di telecamere orizzontali ad asse ottico parallelo, in modo che tale vincolo sia realmente applicabile. Queste due immagini rappresenteranno i dati in ingresso per l'algoritmo di correlazione che si intende realizzare. L'output sarà, invece, rappresentato dalla mappa di disparità puntuale (ovvero, pixel a pixel) delle due immagini. Il metodo utilizzato sarà una correlazione area-based tra l'intorno di un pixel di partenza (ad esempio nell'immagine di sinistra) ed i candidati pixels nell'altra immagine; inoltre si tenterà di sfruttare alcune funzioni euristiche per rendere più proficua la ricerca del miglior match.

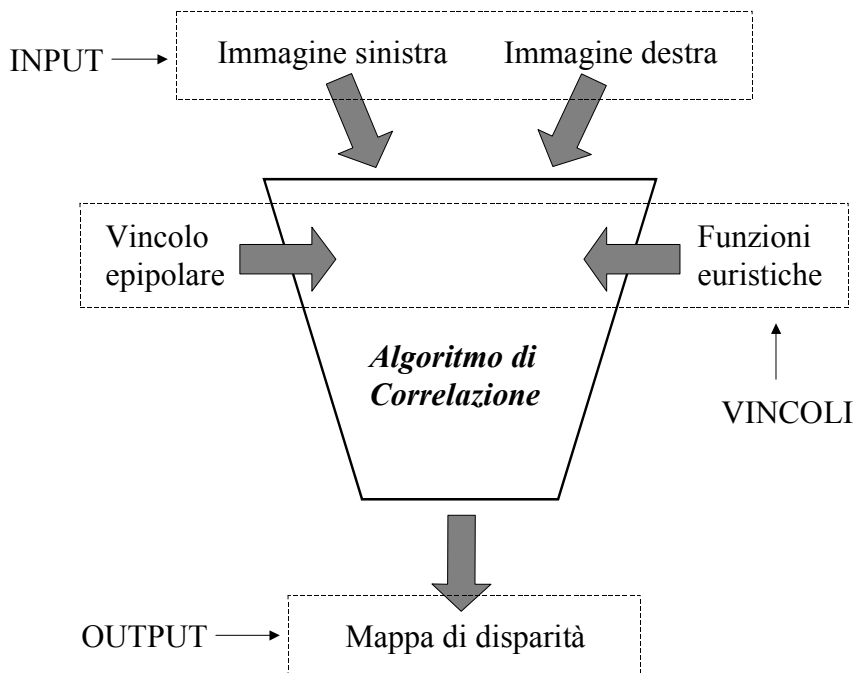
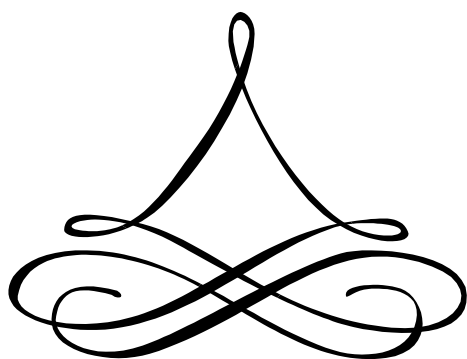


Figura 5 – Input ed output per l'algoritmo di correlazione

Capitolo

I

Analisi del problema



Analisi del problema

I.1 Che cosa si vuole realizzare

Quello che si vuole realizzare è un agente software in grado di prendere in ingresso due immagini e di produrre come output una mappa di disparità per tali immagini. La mappa di disparità può essere pensata come una matrice M di interi di dimensione $W \times H$, dove W ed H sono rispettivamente l'ampiezza e l'altezza dell'immagine. Ogni intero d presente nelle celle di M rappresenta la distanza (in pixel) tra il pixel di riferimento p , preso dalla prima immagine, ed il pixel p' , appartenente alla seconda immagine, che più di ogni altro "assomiglia" al pixel p . Più è alto il valore della disparità d , più il punto tridimensionale, rappresentato univocamente dalla coppia di pixel p e p' , si trova vicino alle due camere. Questo è vero a condizione che le due camere abbiano i propri obiettivi allineati e lievemente distanti come in figura I.1.

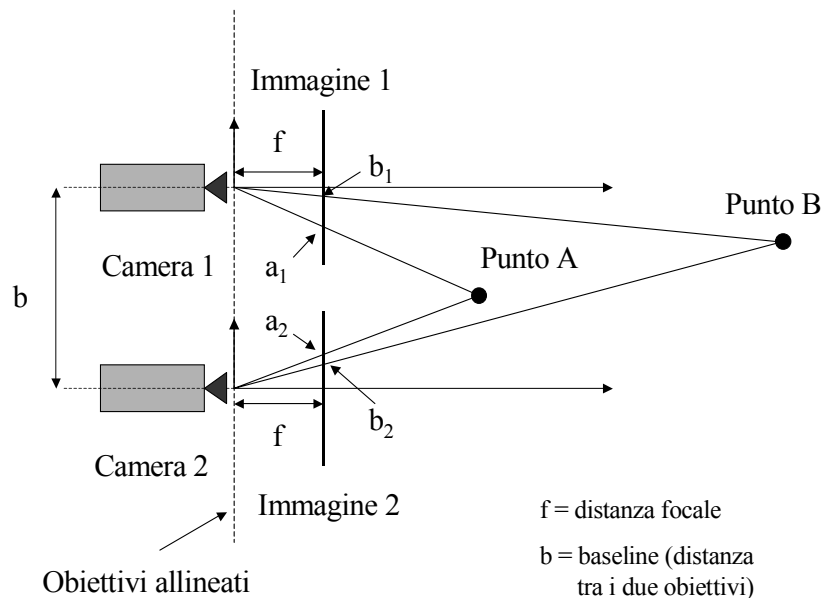


Figura I.1 – Sistema di telecamere con obiettivi allineati

La disparità $d(A)$ del punto A deve essere maggiore rispetto alla disparità $d(B)$ del punto B, poiché evidentemente A si trova più vicino alle due camere. Definiamo a_1 il punto proiezione di A su Immagine 1, a_2 il punto proiezione di A su immagine 2, b_1 il punto proiezione di B su immagine 1 e b_2 il punto proiezione di B su immagine 2. Si assuma, inoltre, che tutti i punti a_1 , a_2 , b_1 e b_2 abbiano lo stesso valore di ordinata. Questa assunzione è lecita se si valuta applicabile il vincolo epipolare [Appendice A, paragrafi 1 e 4]; in tal caso i quattro punti si trovano sulla stessa linea orizzontale. Calcoliamo i due valori di disparità:

$$\text{disparità per il punto A:} \quad d(A) = a_1 - a_2$$

$$\text{disparità per il punto B:} \quad d(B) = b_1 - b_2$$

Troviamo $d(A) > d(B)$, poiché $a_1 > b_1$ e $a_2 < b_2$.

E' bene notare che vanno considerati soltanto i valori di ascissa dei quattro punti, e tali valori possono essere confrontati facendo riferimento alla figura I.2.

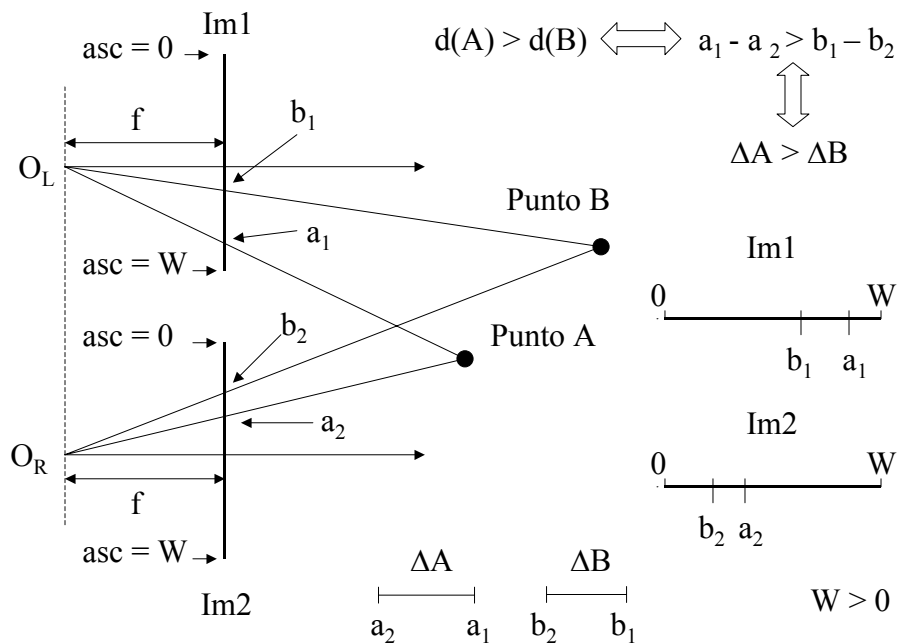


Figura I.2 – Calcolo della disparità

Una volta che la matrice M è stata completamente riempita, cioè una volta che per ogni pixel nella immagine di riferimento è stato trovato il corrispettivo nella seconda immagine, abbiamo una mappa della disparità della scena che stiamo analizzando. Se trasformiamo l'informazione contenuta in M sotto forma di numeri interi (che vanno da zero a D_{MAX} , con D_{MAX} massima disparità ammessa) in una qualche scala di colori, possiamo ottenere una terza immagine che rappresenta gli oggetti della scena con differenti colori, con gradazioni che variano a seconda della distanza di tali oggetti dalle camere. Questa terza immagine verrà, nel seguito, indicata come *immagine di disparità*. Adoperando, ad esempio, una scala di verdi con rapporto crescente tra intensità di verde e valore di disparità, gli oggetti più vicini alle camere appaiono di un verde più acceso, quelli più distanti sono verde più opaco. Una volta prodotta l'immagine di disparità, l'agente software esaurisce il suo compito e resta solo da valutare se il lavoro realizzato è un buon lavoro o meno.

I.2 La misura di prestazione

Fissare una misura di prestazione significa asserire dei criteri oggettivi con cui valutare il lavoro dell'agente. Avendo presente che questo progetto prevede, oltre alla realizzazione dell'immagine di disparità per le due immagini di input, anche il confronto tra i risultati prodotti da più algoritmi, definiremo due tipi di misure di prestazione: una assoluta, l'altra relativa. Infatti, diversi algoritmi di correlazione producono differenti mappe di disparità e di conseguenza diverse immagini di disparità che vanno relazionate tra loro.

Definizione I.1 *Definiamo misura assoluta di prestazione la percentuale di corrispondenze corrette trovate dall'algoritmo di correlazione adoperato dall'agente. Più questa percentuale è elevata, più il nostro agente lavora bene.*

Si ha una corrispondenza corretta quando la disparità trovata per un pixel dall' algoritmo corrisponde alla **reale** disparità del pixel. Per immagini virtuali (figura I.3) è possibile ottenere una mappa di disparità corretta (*ground truth*) con cui confrontare i dati prodotti dall' agente.

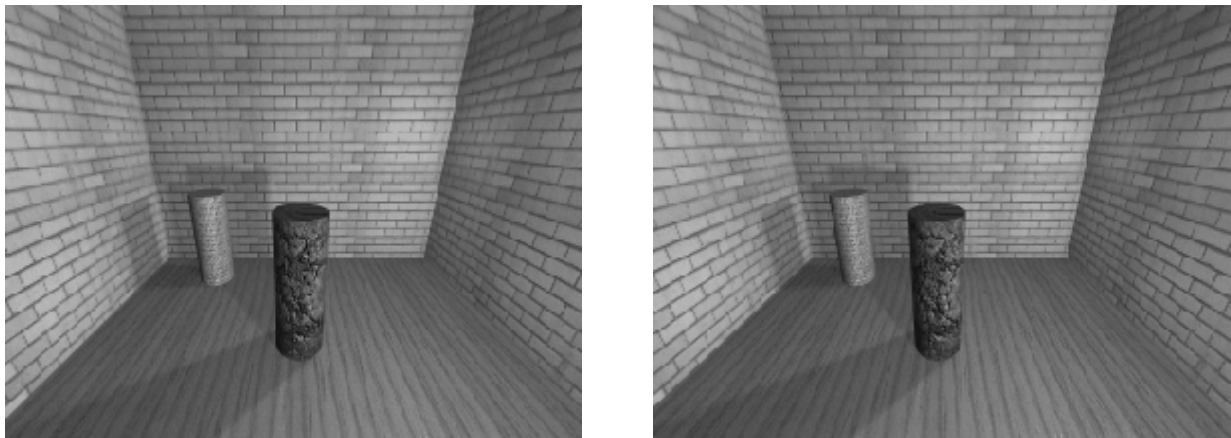


Figura I.3 – Coppia stereo di immagini virtuali

Per immagini reali la mappa ground truth è più complicata da ottenere, anche se non impossibile (la figura I.4 presenta un classico esempio di coppia stereo reale con relativa ground truth).

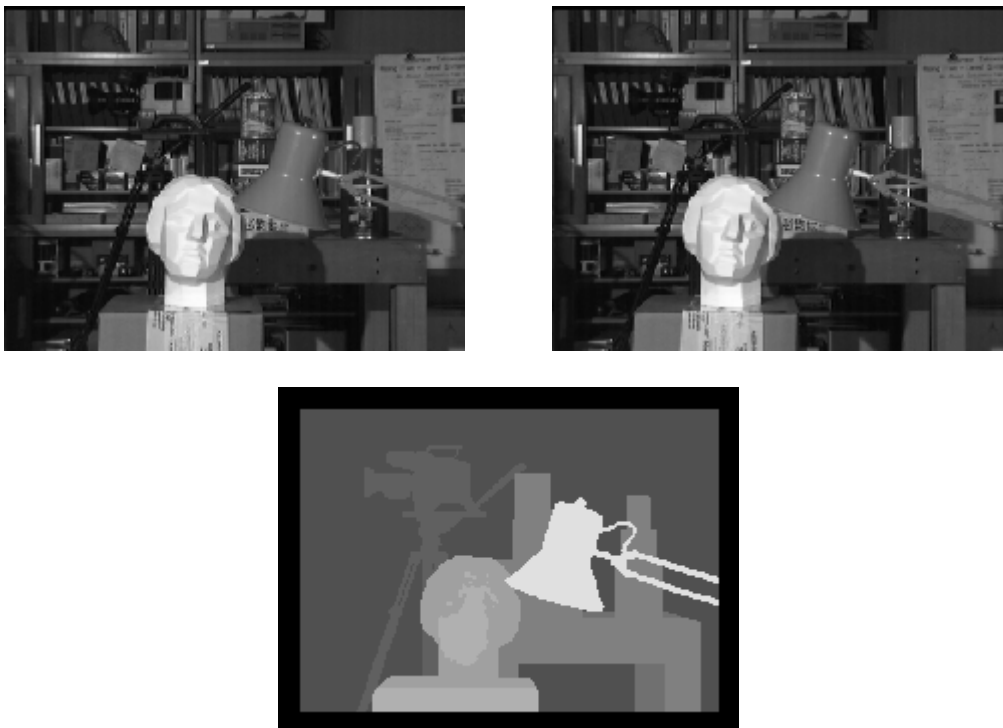


Figura I.4 – Coppia stereo con ground truth (Università di Tsukuba)

Per ovviare alla mancanza della mappa ground truth, introduciamo una seconda misura che non faccia riferimento a tale mappa, ma che consenta, comunque, di effettuare un confronto tra le immagini di disparità prodotte dall' algoritmo:

Definizione I.2 *Definiamo misura relativa di prestazione il confronto visivo tra l'immagine campione, ovvero l'immagine di disparità ottenuta adoperando un algoritmo SSD [Capitolo II], e l'immagine di cui si sta valutando la qualità.*

Oltre al confronto con l'immagine campione, è possibile effettuare un primo, rapido, confronto della qualità dell'immagine di disparità ottenuta semplicemente attraverso un'analisi visiva, ad esempio esaminandone:

- 1) la pulizia. Definiamo pulita una immagine in cui gli oggetti della scena sono facilmente riconoscibili e non sono presenti macchie di colore non riconducibili ad alcun oggetto.
- 2) le regioni intorno ai bordi degli oggetti presenti nella scena. Se non è possibile individuare nettamente i bordi degli oggetti, ovvero se i contorni di due oggetti distinti si confondono, allora l'algoritmo sta producendo false corrispondenze.
- 3) la presenza o meno di buchi. Un buco è una zona a disparità nulla immersa in una zona ad elevata disparità. Evidentemente un buco è una perdita di informazione, per cui una immagine di disparità con molti buchi deve essere migliorata.

La misura di prestazione dovrà tenere in conto anche il tempo necessario per realizzare l'immagine di disparità.

Definizione I.3 *Definiamo tempo di elaborazione T il tempo, in secondi, che occorre all'algoritmo di correlazione per produrre la mappa di disparità.*

Una buona immagine che necessita di un basso tempo di elaborazione è da preferirsi ad un'ottima immagine per la quale bisogna attendere molto tempo. Si deve, come in ogni problema ingegneristico che si rispetti, trovare il giusto compromesso tra qualità dell'immagine di disparità e tempo che l'algoritmo impiega per produrre tale immagine.

I.3 L'input

L'input per l'applicazione è una coppia di immagini. Poiché immagini prodotte da telecamere soffrono inevitabilmente di distorsioni, bisogna che tali immagini originali siano trattate prima di essere date in pasto all'algoritmo.



Figura I.5 – Coppia originale di immagini stereo

Questo trattamento può essere definito *correzione geometrica*. Le distorsioni presenti nelle immagini originali, attraverso un processo di rettificazione (warping), sono corrette e portate in una forma standard.



Figura I.6 – Coppia di immagini rettificate

Il processo di warping non rientra tra gli argomenti di questo scritto, per tale motivo, nel seguito, si darà per scontato che l'input per l'algoritmo di correlazione sia rappresentato da immagini già rettificate. Ciò vuol dire che il calcolo del tempo di elaborazione non terrà conto della quantità di tempo che occorre per rettificare le due immagini che si vogliono correlare.

I.4 L'output

L'output è rappresentato dall'immagine di disparità che scaturisce dal processo di correlazione. Ogni diverso algoritmo di matching produce una differente mappa di disparità e di conseguenza immagini diverse. Si assumerà di rappresentare le informazioni presenti nella mappa con una scala di verdi. Tale scelta è puramente soggettiva: può essere scelta qualsiasi scala di colori, l'importante è che sia possibile distinguere le varie profondità della scena. Viene presentata una serie di immagini di disparità tipo (figura I.7 e seguenti figure I.8 e I.9) per le coppie stereo presentate in precedenza, al fine di mostrare ciò che ci attendiamo come output dall'applicazione.

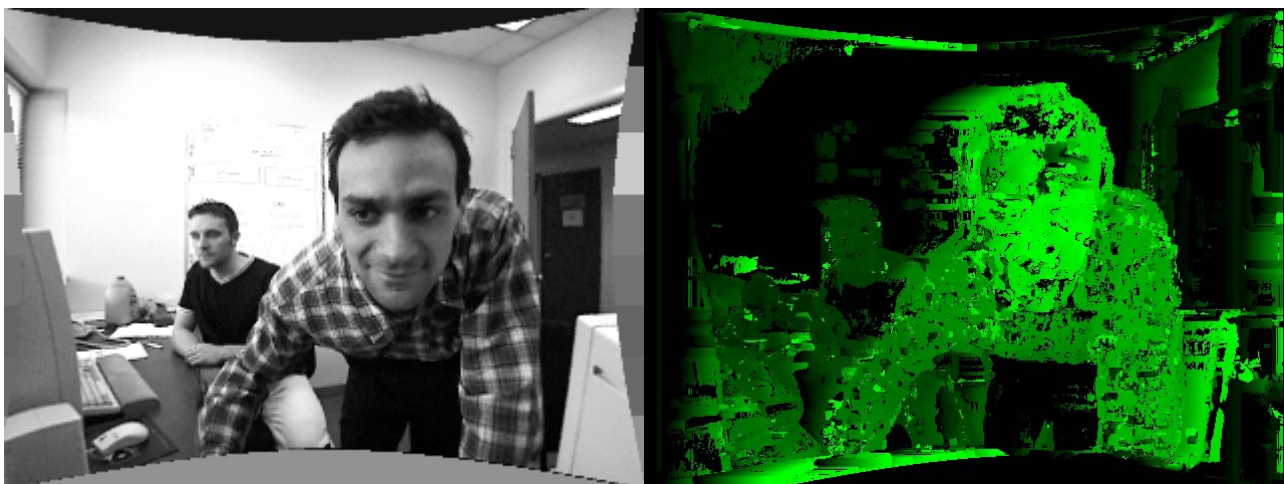


Immagine sinistra

Immagine di disparità

Figura I.7 – L'immagine di disparità è il prodotto dell'algoritmo di correlazione



Immagine sinistra

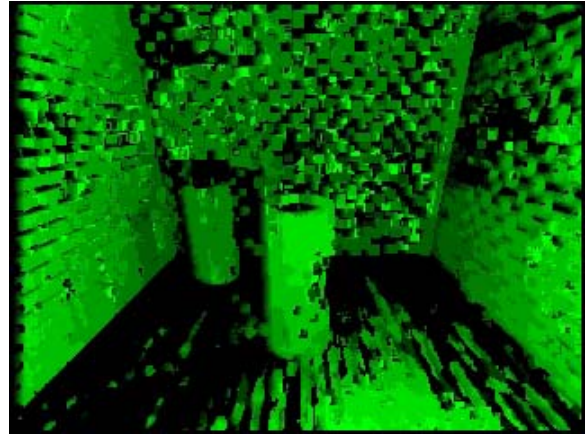


Immagine di disparità

Figura I.8 – Immagine di disparità ottenuta da una coppia virtuale



Immagine sinistra

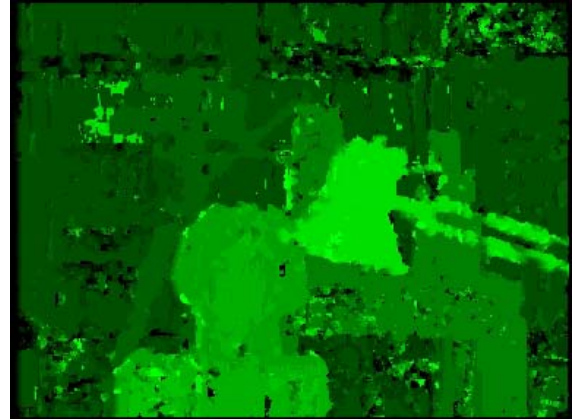


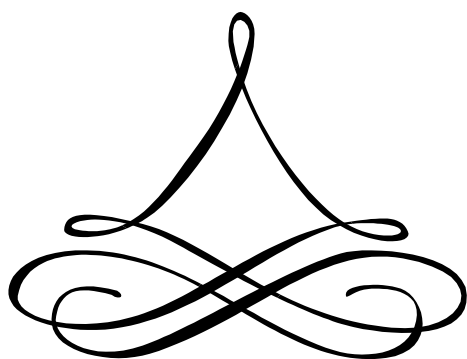
Immagine di disparità

Figura I.8 – Immagine di disparità ottenuta dalla coppia stereo di Tsukuba

Capitolo

II

La Corrispondenza



La Corrispondenza

II.1 Le tecniche di corrispondenza

La disparità di una coppia coniugata di punti può essere determinata adoperando diversi metodi, sfruttando numerosi vincoli. Tutti questi metodi cercano di accoppiare i pixels dell'immagine di riferimento con i corrispondenti pixels nell'altra immagine. I vincoli, così come i vari metodi, possono essere suddivisi, per semplicità, in *locali* e *globali*.

II.1.1 I vincoli locali

I vincoli locali possono essere applicati su un piccolo numero di pixels che circondano il pixel di interesse. Ad esempio, se i pixels che sono immediatamente adiacenti il pixel p di interesse hanno tutti disparità d , molto probabilmente anche p avrà disparità d . Se esprimiamo questo concetto attraverso un vincolo, chiamiamolo *vincolo di continuità a tratti della disparità*, stiamo adottando un vincolo locale.

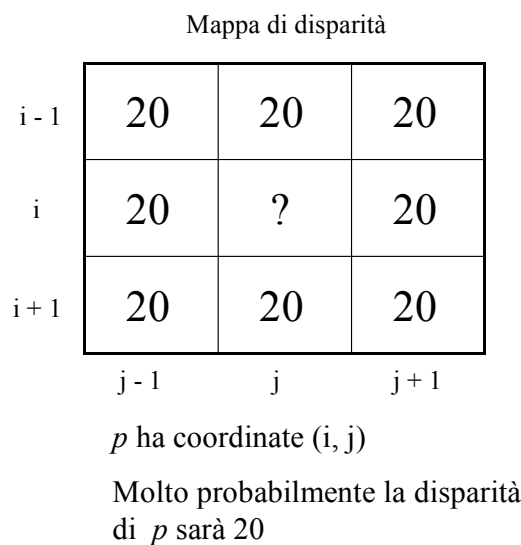


Figura II.1 – Continuità della disparità per un punto dell'immagine

Il vincolo è locale poiché dipende dal valore di disparità che può essere calcolato in un intorno di p . Se cambiassimo punto di interesse, ad esempio p' , quel che era valido per p non sarà più valido per il nuovo punto di interesse p' : ecco il motivo della denominazione "locale".

II.1.2 I vincoli globali

I vincoli globali si riferiscono alla direzione di ricerca dei punti coniugati (scan line) o all'immagine nella sua interezza. Il più noto ed usato dei vincoli sulla direzione di ricerca è il vincolo epipolare. Un vincolo che può essere applicato all'immagine intera è il vincolo di luminosità costante: la luminosità di un punto è la stessa nelle due immagini che lo rappresentano. L'adozione del modello Lambertiano [Appendice A, paragrafo 5] è un ulteriore esempio di vincolo globale.

II.2 I metodi locali

I metodi locali di correlazione possono essere molto efficienti, tuttavia essi sono notevolmente sensibili alla presenza di regioni delle immagini la cui disparità è ambigua (regioni con presenza di occlusioni o regioni con texture uniforme). E' possibile suddividere questi metodi in tre ampie classi: *block matching*, *gradient methods* e *feature matching*.

II.2.1 Block matching

I metodi di block matching, detti anche metodi area-based, cercano di stimare la disparità di un punto p nella prima immagine comparando una piccola regione presa attorno a p (area di riferimento o finestra fissa) con una serie di altre regioni, delle stesse dimensioni dell'area di riferimento, estratte dalla seconda immagine (frames di ricerca). Il valore per la funzione di corrispondenza è ottenuto mettendo a confronto la finestra fissa della prima immagine con una finestra scorrevole nella seconda immagine. La finestra scorrevole viene spostata sulla seconda immagine, attraverso incrementi interi, lungo la linea epipolare di p : ogni singolo incremento genera un nuovo frame di ricerca. In base al vincolo epipolare, i frames di ricerca avranno per centro un punto con coordinate di ascissa variabile, ma con ordinata fissa ed uguale al valore di ordinata per p . Il vincolo epipolare, poiché fissa il valore di ordinata, riduce la complessità della ricerca da bidimensionale a monodimensionale.

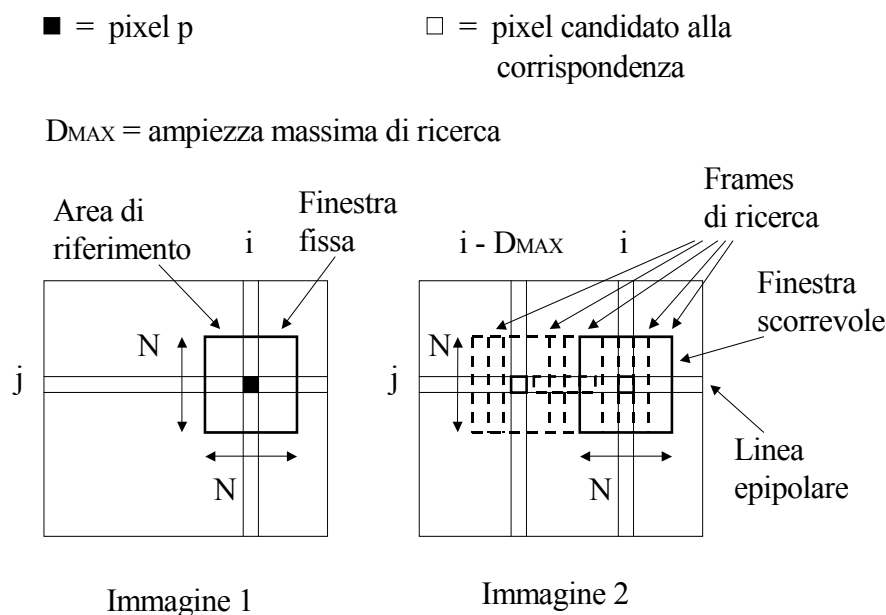


Figura II.2 – Il block matching

Una volta trovato il frame di ricerca che più assomiglia alla finestra di riferimento, il baricentro p' di tale frame sarà il pixel corrispondente a p . E' fondamentale stabilire in che modo si intende attribuire un grado di somiglianza per ogni singolo frame che si processa. In altre parole, deve essere definita una qualche *misura di similarità*, in modo che il confronto tra la finestra di riferimento ed i frames possa essere legato ad una scala di valori con cui produrre la curva di corrispondenza. Così facendo, il confronto tra la finestra di riferimento ed i vari frames genererà una curva dei valori di corrispondenza (funzione di corrispondenza) per ogni punto della seconda immagine che risiede sulla linea epipolare di p . La disparità corretta può essere definita come la differenza (in valore assoluto) tra il valore di ascissa di p ed il valore di ascissa del centro p' del frame di ricerca in relazione al quale la funzione di corrispondenza presenta il picco più alto.

Alcune possibili misure di similarità sono la *correlazione* e le metriche legate al rango (*rank metrics*).

II.2.1 A La correlazione

La correlazione è una tecnica di facile implementazione che genera una mappa di disparità densa, ovvero viene trovata la disparità per ogni punto dell'immagine di riferimento; tuttavia, proprio perché processa tutti i pixels, la correlazione risulta relativamente lenta, lentezza che mal si addice a sistemi real-time. Il match della finestra di riferimento con i vari frames viene effettuato confrontando algebricamente i valori (per immagini in bianco e nero si tratta dei valori GRAYSCALE, per immagini a colori si utilizzano i valori RGB) dei pixels presenti nella finestra e nel frame. Focalizziamo l'attenzione sulla figura sottostante.

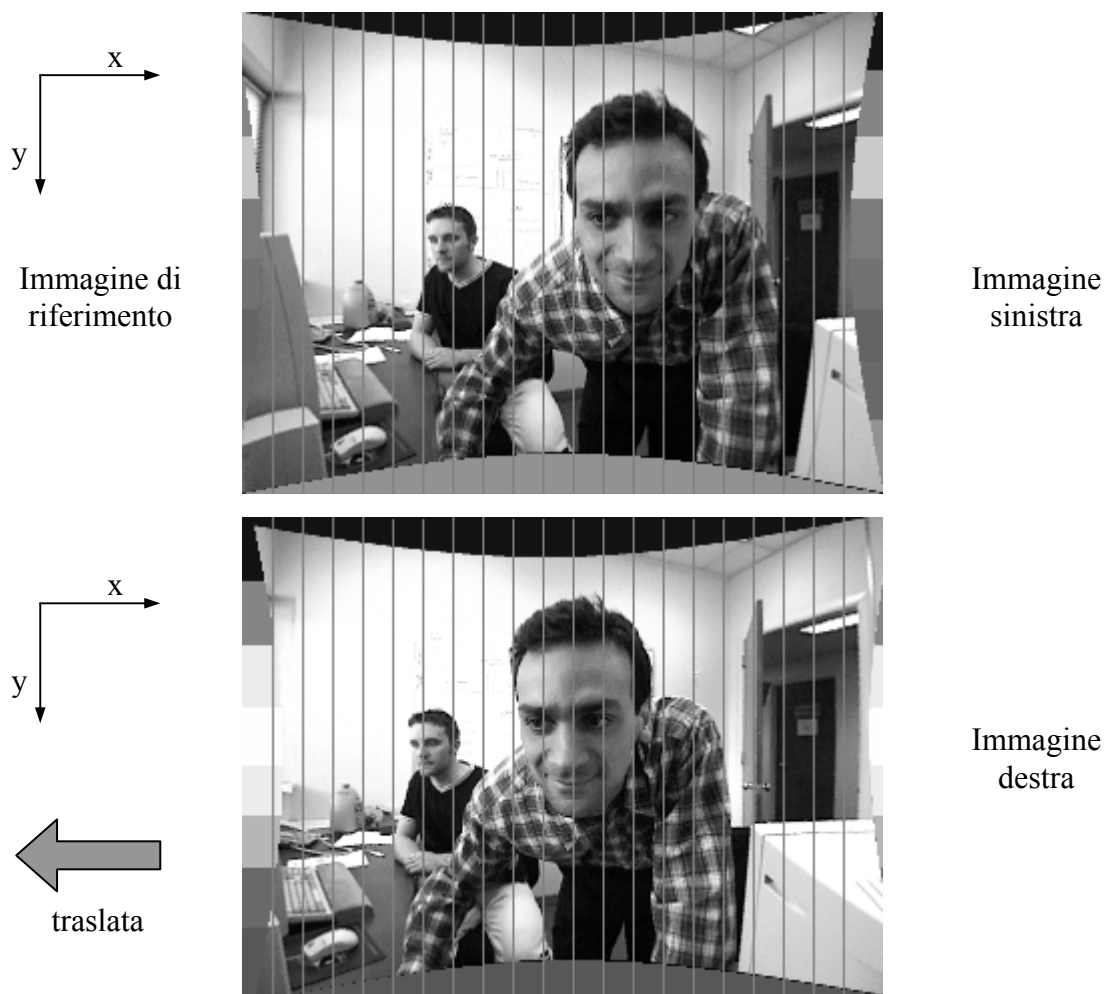


Figura II.3 – Calcolo della disparità per una coppia stereo

L'immagine sinistra è l'immagine di riferimento, mentre sull'immagine destra devono essere scelti i frames di ricerca con cui confrontare la finestra di riferimento. Se si presta un minimo di attenzione all'immagine inferiore (destra) si nota che la scena è traslata verso sinistra in relazione all'immagine di riferimento. La traslazione è dovuta alle posizioni relative degli obiettivi delle due telecamere: l'obiettivo più a destra necessariamente vedrà gli oggetti leggermente shiftati rispetto all'obiettivo sinistro. Questo vuol dire che un punto p di coordinate (x, y) preso sull'immagine sinistra avrà un coniugato sull'immagine destra di coordinate $(x - d, y)$, con d valore di disparità che varia tra 0 e D_{MAX} . La ricerca andrà fatta all'indietro, nel senso che i frames di ricerca avranno di volta in volta un centro con valore di ascissa decrescente. Se si scegliesse di prendere come

immagine di riferimento quella destra, il coniugato del generico punto p (questa volta p sarà sull'immagine destra) avrà coordinate $(x + d, y)$. Una volta individuata la rosa dei possibili candidati ad essere il frame di ricerca che più assomiglia alla finestra di riferimento, bisogna trovare in questo insieme il miglior match possibile. Esistono diversi metodi per farlo, ne vengono presentati alcuni.

SSD Sum of Squared Difference

$$C_{SSD}(x, y, d) = \min \left\{ \sum_{i,j} [I_1(x+i, y+j) - I_2(x-d+i, y+j)]^2 \right\}$$

SAD Sum of Absolute Difference

$$C_{SAD}(x, y, d) = \min \left\{ \sum_{i,j} |I_1(x+i, y+j) - I_2(x-d+i, y+j)| \right\}$$

NCC Normalized Cross Correlation

$$C_{NCC}(x, y, d) = \max \left\{ \frac{\sum_{i,j} [I_1(x+i, y+j) - \bar{I}_1(x, y)] \cdot [I_2(x-d+i, y+j) - \bar{I}_2(x+d, y)]}{\left\{ \sum_{i,j} [I_1(x+i, y+j) - \bar{I}_1(x, y)]^2 \cdot \sum_{i,j} [I_2(x-d+i, y+j) - \bar{I}_2(x+d, y)]^2 \right\}^{1/2}} \right\}$$

L'indice i varia tra $-\lfloor N/2 \rfloor$ e $+\lfloor N/2 \rfloor$ con N ampiezza della finestra, l'indice j varia tra $-\lfloor M/2 \rfloor$ e $+\lfloor M/2 \rfloor$ con M altezza della finestra. $I_1(x, y)$ e $I_2(x, y)$ denotano i valori di intensità del pixel (x, y) per la prima (I_1) e la seconda (I_2) immagine, d è il valore di disparità. $\bar{I}_k(x, y)$ con $k = 1, 2$ è la media, all'interno della finestra di dimensione $N \times M$, dei valori di luminosità dei pixels dell'immagine k .

Procediamo con l'analisi di un esempio che sfrutta il metodo SSD. Si considerino due ipotetiche matrici che devono essere correlate (figura II.3). Ogni cella (i, j) di una matrice contiene un intero che rappresenta il valore di colore per il punto (i, j) dell'immagine a cui si riferisce la matrice. Si supponga di voler trovare per il punto p , di coordinate riga-colonna $(4, 5)$, il proprio coniugato p' . Inoltre, si suppongano le dimensioni della finestra di correlazione pari a 3×3 e la massima ampiezza di ricerca (D_{MAX}) pari a 2.

Matrice 1 immagine sinistra													Matrice 2 immagine destra												
0	1	3	7	6	4	8	9	6	5	4	6	7	0	1	4	4	7	9	0	0	8	5	7	3	2
1	5	8	5	3	5	7	9	8	6	4	3	4	1	2	6	5	3	5	9	8	8	8	8	5	4
2	6	7	9	0	6	6	3	5	6	7	8	9	2	5	6	8	9	4	5	2	3	4	1	1	1
3	7	5	4	3	4	4	7	6	5	4	4	9	3	4	5	6	7	8	9	0	8	5	4	4	1
4	7	3	7	8	9	9	1	2	3	0	0	0	4	1	3	1	1	9	6	9	2	3	5	0	4
5	4	5	6	7	8	4	5	6	7	7	5	1	5	4	5	5	7	8	9	0	8	5	3	6	7
6	9	4	6	3	4	6	7	8	3	4	6	7	6	3	2	6	3	4	3	2	4	5	6	4	3
7	2	3	4	5	6	7	7	5	4	3	2	4	7	4	5	6	5	7	7	5	6	4	7	2	5
8	3	7	4	9	9	9	9	5	5	5	3	4	8	5	7	6	6	5	5	6	7	9	6	3	2
9	5	6	7	5	4	6	8	9	4	5	7	7	9	3	4	5	6	4	4	8	4	6	7	8	7
10	8	5	8	5	8	5	6	5	7	8	5	8	10	2	4	8	4	4	4	4	5	2	3	4	6
11	5	5	3	4	8	8	9	5	3	0	0	0	11	5	5	6	4	6	8	6	5	6	0	6	0
	0	1	2	3	4	5	6	7	8	9	10	11		0	1	2	3	4	5	6	7	8	9	10	11

Figura II.4 – Le due matrici da correlare

La figura II.5 mostra i passi da affrontare per trovare p' a partire da p .

Finestra di riferimento	Primo frame di ricerca	Prima iterazione
$ \begin{array}{c} 4 \quad \begin{array}{ c c c } \hline 4 & 4 & 7 \\ \hline 9 & 9 & 1 \\ \hline 8 & 4 & 5 \\ \hline \end{array} \\ 5 \end{array} $	$ \begin{array}{c} 4 \quad \begin{array}{ c c c } \hline 8 & 9 & 0 \\ \hline 9 & 6 & 9 \\ \hline 8 & 9 & 0 \\ \hline \end{array} \\ 5 \end{array} $	$d = 0$
Finestra di riferimento	Secondo frame di ricerca	Seconda iterazione
$ \begin{array}{c} 4 \quad \begin{array}{ c c c } \hline 4 & 4 & 7 \\ \hline 9 & 9 & 1 \\ \hline 8 & 4 & 5 \\ \hline \end{array} \\ 5 \end{array} $	$ \begin{array}{c} 4 \quad \begin{array}{ c c c } \hline 7 & 8 & 9 \\ \hline 1 & 9 & 6 \\ \hline 7 & 8 & 9 \\ \hline \end{array} \\ 4 \end{array} $	$d = 1$
Finestra di riferimento	Terzo frame di ricerca	Terza iterazione
$ \begin{array}{c} 4 \quad \begin{array}{ c c c } \hline 4 & 4 & 7 \\ \hline 9 & 9 & 1 \\ \hline 8 & 4 & 5 \\ \hline \end{array} \\ 5 \end{array} $	$ \begin{array}{c} 4 \quad \begin{array}{ c c c } \hline 6 & 7 & 8 \\ \hline 1 & 1 & 9 \\ \hline 5 & 7 & 8 \\ \hline \end{array} \\ 3 \end{array} $	$d = D_{MAX} = 2$

Figura II.5 – I passi della correlazione per il punto $p \equiv (4, 5)$

Il vincolo epipolare garantisce che p' abbia come ordinata il valore 4, mentre per il valore di ascissa devono essere prese in considerazione diverse ipotesi. Poiché il valore fissato per l'ampiezza di ricerca (D_{MAX}) è 2, la finestra di riferimento dovrà essere confrontata con tre frames di ricerca.

$$\begin{aligned}
 \text{Prima iterazione: } S_1 &= (4 - 8)^2 + (4 - 9)^2 + (7 - 0)^2 + \\
 &\quad (9 - 9)^2 + (9 - 6)^2 + (1 - 9)^2 + \\
 &\quad (8 - 8)^2 + (4 - 9)^2 + (5 - 0)^2 = 213 \qquad d = 0
 \end{aligned}$$

$$\begin{aligned}
 \text{Seconda iterazione: } S_2 &= (4 - 7)^2 + (4 - 8)^2 + (7 - 9)^2 + \\
 &\quad (9 - 1)^2 + (9 - 9)^2 + (1 - 6)^2 + \\
 &\quad (8 - 7)^2 + (4 - 8)^2 + (5 - 9)^2 = 151 \qquad d = 1
 \end{aligned}$$

$$\begin{aligned}
 \text{Terza iterazione: } S_3 &= (4 - 6)^2 + (4 - 7)^2 + (7 - 8)^2 + \\
 &\quad (9 - 1)^2 + (9 - 1)^2 + (1 - 9)^2 + \\
 &\quad (8 - 5)^2 + (4 - 7)^2 + (5 - 8)^2 = 233 \qquad d = D_{MAX} = 2
 \end{aligned}$$

La disparità per p è 1. Infatti, adoperando il metodo SSD, un valore di somma nullo indica la corrispondenza perfetta, il miglior match è quello con il valore di somma più basso.

Vediamo cosa accade utilizzando il SAD. Questo metodo è da preferirsi se si intende risparmiare risorse per la computazione o se le risorse sono comunque limitate. Infatti, il calcolo dei quadrati nell'SSD è certamente più oneroso sia in termini di risorse adoperate, sia in termini di tempo di elaborazione rispetto al semplice valore assoluto che adopera il SAD.

$$\begin{aligned}
 \text{Prima iterazione: } S_1 &= |4 - 8| + |4 - 9| + |7 - 0| + \\
 &\quad |9 - 9| + |9 - 6| + |1 - 9| + \\
 &\quad |8 - 8| + |4 - 9| + |5 - 0| = 37 \qquad d = 0
 \end{aligned}$$

$$\begin{aligned}
 \text{Seconda iterazione: } S_2 &= |4 - 7| + |4 - 8| + |7 - 9| + \\
 &\quad |9 - 1| + |9 - 9| + |1 - 6| + \\
 &\quad |8 - 7| + |4 - 8| + |5 - 9| = 31 \qquad d = 1
 \end{aligned}$$

Terza iterazione: $S_3 = |4 - 6| + |4 - 7| + |7 - 8| +$
 $|9 - 1| + |9 - 1| + |1 - 9| +$
 $|8 - 5| + |4 - 7| + |5 - 8| = 39$ $d = D_{MAX} = 2$

Anche adoperando il SAD il secondo frames risulta essere il più somigliante alla finestra di riferimento; si può concludere affermando che il coniugato di $p \equiv (4, 5)$ è $p' \equiv (4, 4)$. □

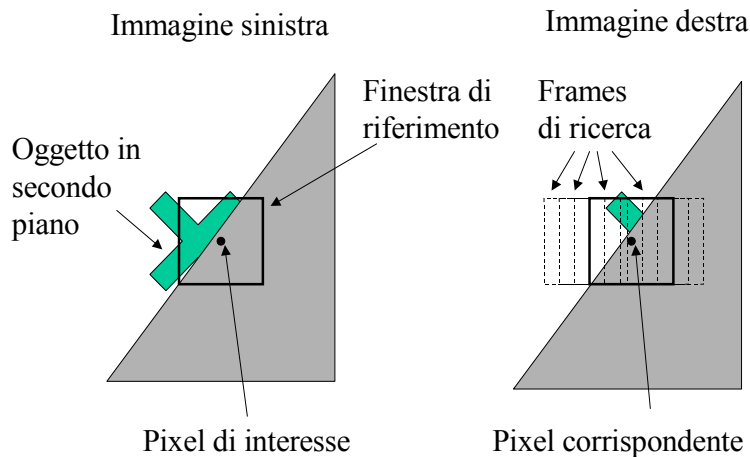


Figura II.6 – La correlazione sul bordo di un oggetto

La probabilità di un match sbagliato decresce in proporzione all'aumentare delle dimensioni della finestra di correlazione [Bibliografia 3]. L'uso di finestre di grosse dimensioni permette di incrementare l'attendibilità dei risultati, poiché è possibile confrontare aree più grandi che contengono maggiori informazioni sulla localizzazione del pixel di interesse e ridurre, in tal modo, l'effetto del rumore che sempre accompagna una immagine. Tuttavia, l'impiego di grandi finestre porta ad una perdita di accuratezza e alla possibile omissione di importanti caratteristiche dell'immagine. Finestre di piccole dimensioni aumentano la risoluzione dell'immagine di disparità, poiché sono in grado di gestire anche piccoli oggetti presenti nella scena, che altrimenti sarebbero trascurati da finestre più ampie. Per finestre più piccole, però, aumentano le probabilità di falsi accoppiamenti ed è necessaria l'introduzione di un valore di soglia [Capitolo III, paragrafo 3] per correggere la correlazione. Tale valore è arbitrario e, di conseguenza, di difficile individuazione.

La correlazione assume che i punti all'interno della finestra di riferimento e nei frames di ricerca abbiano profondità costante; ciò non è vero, invece, nel caso in cui la finestra o un frame coprano una regione dell'immagine in cui siano presenti due oggetti a diversa distanza dagli obiettivi delle telecamere.

Osservazione II.1 *In generale, variazioni repentine di profondità introducono errori nel calcolo della disparità con la correlazione.*

Una particolare situazione si ha se una finestra coincide con una regione che presenta forti differenze di profondità: in questo caso, una parte della finestra influenzerà negativamente il risultato. La figura II.6 mostra l'eventualità in cui la regione sinistra della finestra di riferimento contenga parte di un oggetto in secondo piano (bassa disparità) e parte di un oggetto in primo piano (alta disparità); inoltre, l'oggetto in secondo piano risulta parzialmente occluso per i frames di

ricerca. Conseguentemente, la parte sinistra della finestra di riferimento introduce un errore nei calcoli della disparità. Per di più, si può notare che la dimensione della parte occlusa dipende dalla differenza di disparità e dalle dimensioni della finestra di riferimento. L'uso di una finestra più piccola riduce il problema, poiché una finestra di dimensioni minori riduce la probabilità di contenere al suo interno la discontinuità di profondità.

II.2.1 B Le metriche legate al rango

In alternativa alla correlazione, sono stati proposti metodi per il calcolo della corrispondenza che sfruttano l'applicazione alle immagini, prima del matching, di trasformazioni locali non parametriche. Nel tentativo di eliminare la sensibilità al rumore, una trasformata legata al rango (rank transform) è localmente applicata alle regioni di entrambe le immagini. La rank transform per una regione R che circonda un pixel p è definita come il numero di pixels in R per i quali l'intensità è minore di quella di p . I valori risultanti sono basati sull'ordine relativo dei pixels in R anziché sul valore assoluto delle intensità.

Regione	89	63	72	p = 55
R	67	55	64	Rank Transform = 2
	58	51	49	

Regione	22	24	34	p = 30
R	25	30	35	Rank Transform = 4
	45	39	29	

Figura II.7 – Esempi di rank transform

L'intervallo di valori che può assumere la trasformata è limitato dalla dimensione della finestra di correlazione, per cui si otterrà un insieme di valori molto livellato che è di aiuto nell'eliminare gli errori dovuti alle oclusioni. Infatti, per quanto grande, la finestra di ricerca è di ampiezza limitata. Dopo aver applicato la rank transform, si procede adoperando la correlazione, tipicamente facendo uso della SAD.

Rank Transform

$$\sum_{u,v} [I'_1(u, v) - I'_2(u - d, v)]$$

$$I'_k(u, v) = \sum_{m,n} I_k(m, n) < I_k(u, v)$$

Se è vero che l'uso della trasformata legata al rango riduce la sensibilità ai rumori, è anche vero che essa riduce un po' la potenza computazionale dell'algorithmo di corrispondenza poiché fa perdere parte dell'informazione. L'ordine relativo di tutti i pixels che giacciono intorno al pixel p a cui si applica la trasformata è codificato tramite un unico valore, per questo è possibile conoscere il numero dei pixels che hanno intensità minore di p ma non la loro posizione all'interno della regione. Per limitare la perdita d'informazione, è stata proposta una trasformata che possa fornire dati anche sulla posizione relativa dei pixels nella regione di interesse, una sorta di censimento di tali pixels. La trasformata che effettua questo censimento (detta proprio *census transform*) conserva

l'informazione sulla distribuzione spaziale dei pixels producendo una stringa di numeri binari. La lunghezza L della stringa dipende dall' area A della finestra di correlazione ed è pari ad $L = A - 1$ poiché la posizione del pixel di interesse non deve essere codificata. La procedura di corrispondenza prosegue con il calcolo della distanza di Hamming tra le due stringhe di bit. E' evidente che, aumentando l'informazione da memorizzare, la census transform risulti molto più costosa, in termini di risorse e tempo di esecuzione, rispetto alla rank transform di un fattore proporzionale alla dimensione della finestra di ricerca.

Regione	89	63	72	p = 55
R	67	55	64	Census Transform = 0000011
	58	51	49	

Regione	22	24	34	p = 30
R	25	30	35	Census Transform = 11010001
	45	39	29	

Figura II.8 – Esempi di census transform

Census Transform

$$\sum_{u,v} \text{HAMMING} [I'_1(u, v), I'_2(u - d, v)]$$

$$I'_k(u, v) = \text{BITSTRING}_{m,n} [I_k(m, n) < I_k(u, v)]$$

Alcuni studi indicano che i metodi basati sulle trasformate permettono di ridurre il rumore dovuto alle distorsioni radiometriche ed alle occlusioni [Bibliografia 4].

II.2.2 Gradient Methods

I metodi basati sul calcolo di un gradiente cercano di determinare le disparità tra due immagini formulando una equazione differenziale che collega la disparità alla variazione della luminosità dell'immagine. La luminosità di un punto della scena viene assunta essere costante in tutte e due le immagini, in questo modo si può calcolare la traslazione orizzontale (disparità) di un punto da una immagine all'altra risolvendo l'equazione differenziale

$$(\nabla_x E) v + E_t = 0$$

dove $\nabla_x E$ denota la componente orizzontale del gradiente dell'immagine, E_t la derivata temporale (qui si riferisce alla differenza di intensità tra l'immagine sinistra e destra) e v la traslazione tra le due immagini.

II.2.3 Feature Matching

Il block matching ed i gradient methods sono molto sensibili alle discontinuità di profondità, giacché una finestra di ricerca può contenere punti con diversi valori di profondità. Questi metodi, inoltre, sono molto sensibili alle regioni delle immagini in cui la texture sia molto uniforme.

I metodi feature-based cercano di oltrepassare questi problemi limitando il calcolo della corrispondenza solo ad alcune zone della scena, in cui il calcolo della disparità sia molto attendibile. Queste zone (*features*) sono gli angoli, gli spigoli ed i bordi degli oggetti presenti nella scena. Certamente, questo modo di procedere non permette di realizzare mappe di disparità dense, perché solo una parte dei pixels delle due immagini è coinvolta nel processo di individuazione delle disparità; d'altro canto, il feature matching è estremamente efficiente sotto l'aspetto del tempo di elaborazione.

Negli anni '80 i sistemi di stereo visione basati sul feature matching ricevettero grande attenzione, soprattutto in virtù della loro grande efficienza. Nell'ultimo decennio, invece, sia perché una gran quantità di applicazioni necessita di mappe dense, sia perché i metodi basati sul block matching hanno incrementato la loro efficienza e robustezza, l'interesse per i metodi feature-based è andato scemando. Un tipico approccio per l'implementazione del feature matching è il metodo gerarchico. Questo metodo utilizza quattro tipi di features: linee, vertici, spigoli e superfici. L'algoritmo gerarchico parte analizzando le superfici (livello più alto), fino ad arrivare alle linee (livello più basso).

II.3 I metodi globali

I metodi globali sono molto meno sensibili rispetto ai metodi locali alle regioni che presentano disparità ambigua (occlusioni e regioni a texture uniforme) poiché i vincoli globali che essi sfruttano garantiscono un supporto addizionale per la gestione delle regioni difficilmente trattabili con metodi locali. L'uso di questi vincoli globali, però, fa diventare la complessità computazionale dei metodi globali significativamente più elevata rispetto a quella dei metodi locali. Per tale motivo, sono stati proposti diversi metodi matematici per ridurre la complessità, sfruttando la decomposizione dei problemi di ottimizzazione in sottoproblemi più piccoli e più semplici da risolvere. La trattazione dei metodi globali non rientra tra gli scopi di questo scritto, quindi essi non verranno più presi in considerazione.

II.4 Complessità computazionali

Il block matching ha un costo computazionale di $O(NDn)$ dove N è il numero totale dei pixels che compongono l'immagine, D è il massimo valore di disparità ammesso (D_{MAX}) ed n è l'area della finestra di ricerca. Questa complessità è molto elevata, comunque bisogna aver presente che una ingenua implementazione della correlazione genera un elevato numero di calcoli ridondanti. Queste ridondanze possono essere eliminate gestendo con attenzione i calcoli per ogni blocco che viene processato, ed, adoperando piccoli accorgimenti [Bibliografia 5], è possibile ridurre la complessità fino a $O(ND)$ rendendo il calcolo delle corrispondenze indipendente dalla dimensione della finestra di ricerca.

I gradient methods hanno una complessità di solo $O(N)$, tuttavia soltanto le traslazioni nella direzione del gradiente possono essere stimate con attendibilità. E' necessario inserire un vincolo addizionale che possa aumentare l'accuratezza dei calcoli, tale inserimento porta la complessità computazionale a $O(Nn)$, come per il block matching.

Il feature matching ha una complessità dipendente dal numero di caratteristiche dell'immagine (features) che vengono prese in considerazione: valutando il metodo gerarchico la complessità per ogni livello è $O(N^4)$, con N numero di features esaminate a quel livello [Bibliografia 6].

Tabelle riassuntive delle tecniche di corrispondenza

VINCOLI	LOCALI	Continuità a tratti della disparità
	GLOBALI	Geometria epipolare
		Modello Lambertiano

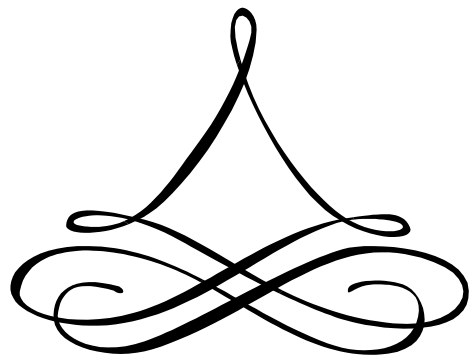
Tabella II.1 – I vincoli che facilitano la corrispondenza

METODI	LOCALI	Block matching	<i>Correlazione</i>
			<i>Rank metrics</i>
		Gradient methods	
	Feature matching	<i>Metodo gerarchico</i>	
	GLOBALI	Decomposizione in sottoproblemi	

Tabella II.2 – I metodi di corrispondenza

Capitolo
III

Le Tecniche di
Correlazione



Le Tecniche di Correlazione

III.1 La funzione di correlazione

La funzione di correlazione è una relazione matematica che mette in comune il valore di disparità dei pixels candidati alla corrispondenza e le connesse misure di similarità trovate per i frames di ricerca di cui tali pixels sono i baricentri.

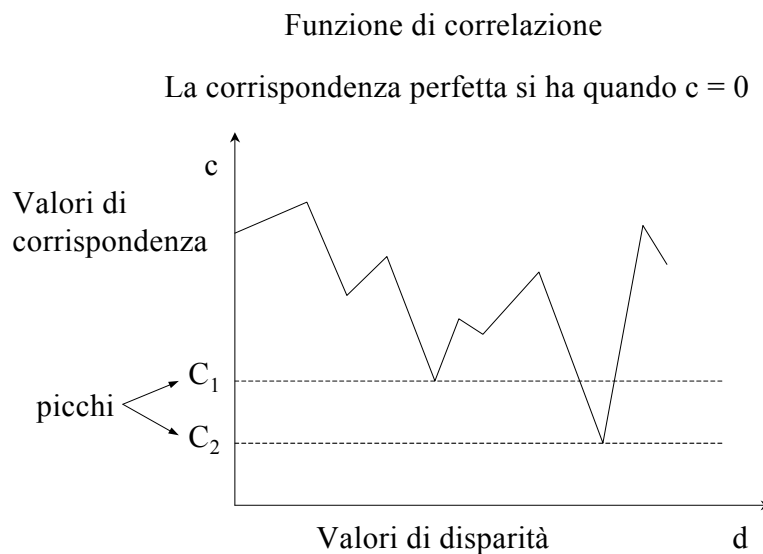


Figura III.1 – Funzione di correlazione

La funzione di correlazione può essere graficata ponendo sulle ascisse i valori di disparità e sulle ordinate i valori ottenuti dal calcolo della corrispondenza. Se si adopera il metodo SSD od il metodo SAD il miglior match sarà il pixel a cui corrisponde il valore di corrispondenza più basso (picco C_2 in figura III.1); se, invece, si adopera il metodo NCC, il pixel da scegliere sarà quello relativo al picco più alto.

Lo studio di tale funzione può risultare utile per correggere, ed in qualche modo indirizzare meglio, l'algoritmo di correlazione. Una funzione quasi piatta corrisponde a regioni dell'immagine in cui la texture è insufficiente, mentre una funzione che presenti troppi picchi (minimi se si sta usando il SAD o l'SSD, massimi se si adopera l'NCC) indica molti buoni match che possono essere causati da una texture ripetitiva. In quest'ultimo caso, il rumore eventualmente presente nell'immagine può portare alla scelta di valori di disparità erronei.

III.2 La gestione delle oclusioni

La gestione delle oclusioni [Appendice A, paragrafo 2] è uno dei punti cruciali di ogni algoritmo di corrispondenza. Le oclusioni sono purtroppo inevitabili, ma possono essere scovate e gestite. Un metodo per scoprire possibili oclusioni è l'analisi dei picchi della funzione di correlazione. Sia C_1 il minimo valore di correlazione, corrispondente ad un pixel γ della seconda immagine, e C_2 il

secondo miglior valore di correlazione, il quale non sia relativo ad un pixel immediatamente vicino a γ (il miglior punto per la correlazione giace di solito tra due pixels, per questo uno scarto di 1 per il valore di disparità deve essere previsto e messo in conto [Bibliografia 7]).

Definizione III.1 Definiamo distanza di picco D_p la differenza $D_p = C_2 - C_1$ tra il valore di correlazione C_1 relativo al picco più basso della funzione di correlazione ed il valore di correlazione C_2 relativo al secondo picco più basso.

La distanza di picco è stata definita assumendo che il metodo di correlazione sia l'SSD. Se si adottano metodi che prevedono una massimizzazione della funzione di corrispondenza (per esempio NCC), la distanza di picco sarà il valore $C_1 - C_2$ con C_1 picco più alto e C_2 secondo picco più alto. Un basso valore di D_p è indice di possibili problemi. Gli errori dovuti alle occlusioni possono essere evitati invalidando i match che presentano D_p inferiore ad una certa soglia S . Deve essere ben chiaro che questa soglia S necessita di un settaggio empirico, che dipende dalle particolari immagini che si stanno analizzando ed, in più, dipende dalla dimensione della finestra di riferimento e dei frames di ricerca. Si può tentare di normalizzare la distanza di picco per renderla indipendente dalla grandezza della finestra di correlazione, in modo da poter adottare un valore di soglia S' più generale. La distanza di picco normalizzata prende il nome di *differenza relativa* [Bibliografia 8]. La differenza relativa C_d può essere calcolata come

$$C_d = \frac{C_2 - C_1}{C_1}$$

Un basso valore di C_d è, come per D_p , spia di possibili match incorretti. Se C_d è inferiore alla soglia S' il risultato del processo di correlazione viene rifiutato. S' necessita ancora di un settaggio empirico, però, ora tale soglia non dipende più né dalle particolari immagini che si stanno analizzando, né dalla dimensione della finestra di ricerca, ma soltanto dal particolare sistema di cattura delle immagini.

III.3 I valori di soglia

L'uso dei valori di soglia può essere esteso anche al di fuori dello studio della funzione di correlazione. Sappiamo [Capitolo II, paragrafo 2.1] che la probabilità di un match sbagliato decresce in proporzione all'aumentare delle dimensioni della finestra di correlazione. Oltre a ciò, è chiaro che finestre di piccole dimensioni aumentano la risoluzione dell'immagine di disparità, poiché sono in grado di gestire anche i più piccoli oggetti presenti nella scena. Vorremmo utilizzare finestre piccole per incrementare la risoluzione, ed allo stesso tempo, non andare incontro ad un numero elevato di falsi match. Si può tentare adoperando un valore di soglia all'interno dell'algoritmo di correlazione. La soglia che si usa per limitare gli errori dovuti ad occlusioni [paragrafo 2 del presente capitolo] è un tipo di soglia *esterno*, nel senso che tale soglia è applicata a valori che si ottengono una volta che l'algoritmo di correlazione ha già analizzato tutti i possibili frames di ricerca e prodotto la funzione di correlazione completa. Il nuovo tipo di soglia che si vuole introdurre è una soglia *interna*. L'algoritmo di correlazione **per ogni** frame verifica che il valore v ottenuto come grado di similarità sia inferiore alla soglia interna S_I . Se $v > S_I$ si passa al prossimo frame, altrimenti si dichiara ottimo il frame per cui il valore di corrispondenza sia minore od uguale alla soglia S_I .

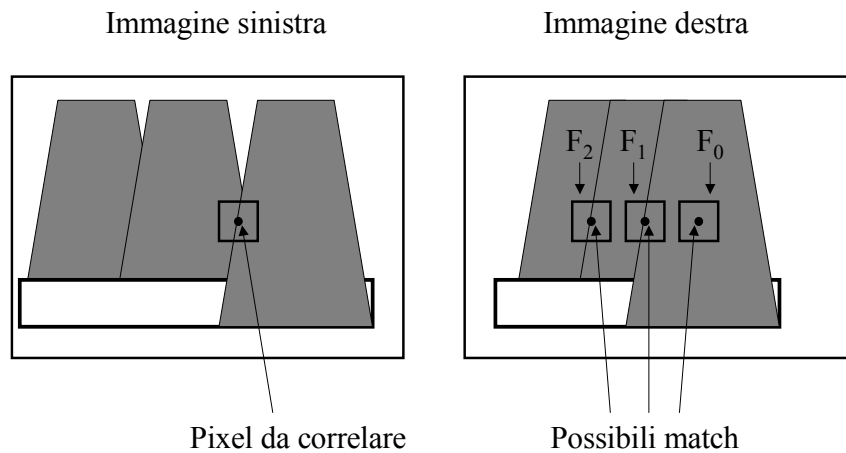
calcolo del valore v_n di similarità per il frame n

if ($v_n < S_I$) return n

else calcolo del valore v_{n+1} di similarità per il frame $n+1$

Questo modo di procedere rende la ricerca dei punti coniugati incompleta, poiché la ricerca può arrestarsi senza aver indagato tutti i possibili match.

L'inserimento di una soglia migliora il rendimento delle finestre più piccole perché limita l'ampiezza di ricerca e riduce la possibilità di incappare in errori dovuti a texture ripetitiva (si veda la figura III.2).



Il match corretto è F_1 , ma potrebbero essere scelti F_2 o F_0 .
Adoperando un valore di soglia l'algoritmo sceglie F_1

Figura III.2 – I valori di soglia migliorano le prestazioni delle finestre più piccole

Il frame F_1 viene controllato prima di F_2 . La soglia, se correttamente settata, dirà all'algoritmo di "accontentarsi" di F_1 senza procedere oltre, evitando possibili errori che si verificherebbero nel processare F_2 . Se il valore di soglia è troppo basso, il valore di corrispondenza per F_1 potrebbe risultare comunque maggiore della soglia e l'algoritmo proseguirebbe verso F_2 . Se il valore di soglia è troppo alto, allora l'algoritmo potrebbe arrestarsi ancor prima di arrivare ad F_1 , scegliendo un frame (F_0) che è molto distante dall'essere il più somigliante, ma che viene processato prima di F_1 . Scegliere il giusto valore di soglia è molto difficile e richiede diversi esperimenti sulle immagini, esperimenti che, però, non sono realizzabili da applicazioni real-time.

III.4 L'intorno di un pixel

L'adozione di alcuni vincoli locali [Capitolo II, paragrafo 1.1] può aiutare nel calcolo di una corretta disparità. Non ha senso considerare ogni pixel di una immagine come una entità isolata, invece bisogna guardare e considerare anche i pixels che si trovano nell'intorno del pixel che si vuole processare, perché la regione attorno ad un punto di cui si vuole trovare il coniugato può fornire importanti informazioni ed indirizzare la ricerca del miglior match. Ad esempio, se il risultato della correlazione per un pixel p non rispetta il vincolo sulla distanza di picco, tale risultato verrà scartato. Scartare un risultato significa assegnare disparità nulla al pixel p in questione. Immaginiamo che p si trovi in una regione a disparità media molto elevata, in tal caso, verrà a crearsi un buco [Capitolo I, paragrafo 2], cosa assolutamente sgradita. Per evitare che ciò si verifichi, si potrebbe calcolare la media M della disparità propria della regione attorno a p ed assegnare il valore M come disparità di p : il buco verrebbe eliminato applicando il vincolo di continuità a tratti della disparità.

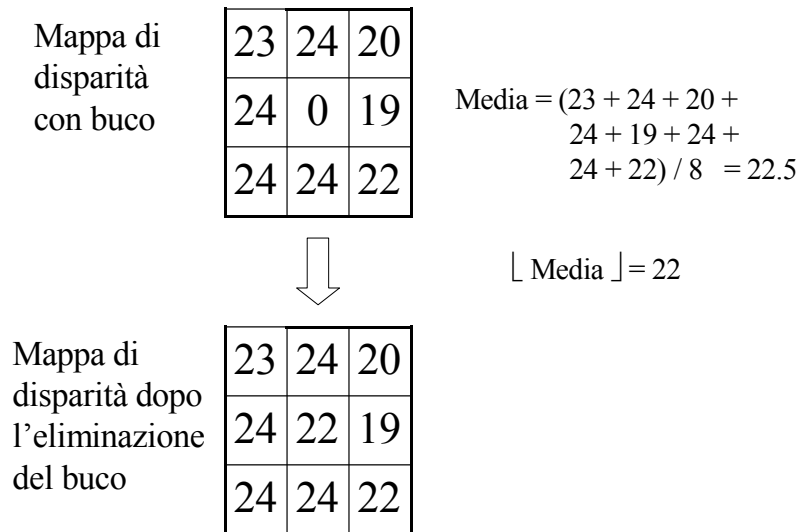


Figura III.3 – Eliminazione di un buco

Oltre che per riempire buchi, la media della disparità per l'intorno di un pixel può essere sfruttata per ridurre il numero di frames da controllare. Se il valore di tale media è d , invece di processare tutti i frames di ricerca le cui disparità variano da 0 a D_{MAX} , si potrebbero esaminare solo i frames a partire da $d - l$ fino a $d + l$, con l intero fissato a piacimento rappresentante il limite di ricerca. Se l non è eccessivamente grande, il numero di iterazioni per l'algoritmo di correlazione viene notevolmente ridotto. Il costo di questa operazione è la mancanza di completezza nella ricerca (non vengono considerati tutti i possibili match) ed una maggiore difficoltà nell'individuare zone della scena in cui la profondità varia repentinamente (la media di disparità di una regione si modifica troppo lentamente rispetto a grosse variazioni di profondità della scena).

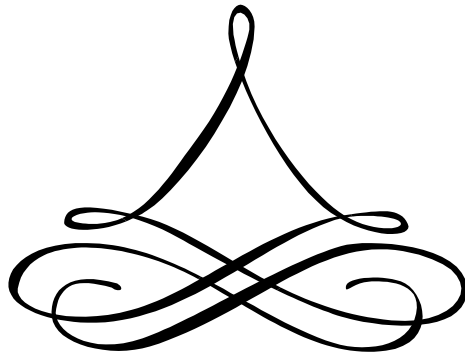
III.5 L'unione di più metodi

L'affidabilità di un match può essere valutata combinando più di un algoritmo di correlazione. Ad esempio, per uno stesso punto p è possibile stabilire la disparità abbinando i risultati prodotti dall'algoritmo SSD e dall'algoritmo SAD. Quando sia l'SSD sia il SAD generano per p la medesima disparità d , possiamo considerare perlomeno attendibile tale valore. Se, in caso contrario, l'SSD genera una disparità d_{SSD} diversa dalla disparità d_{SAD} generata dal SAD, sicuramente una tra d_{SSD} e d_{SAD} non è corretta, infatti p non può avere un valore doppio di disparità per l'unicità dei punti coniugati. In caso le disparità calcolate con metodi diversi non convergano, il match deve essere rigettato, poiché presumibilmente si tratta di una zona di occlusione o, più in generale, di una regione dell'immagine che fornisce una informazione insufficiente (presenza di rumore o texture troppo uniforme). Come avvenuto per la gestione delle distanze di picco insufficienti, ad un pixel che presenti disparità ambigua può essere assegnata la disparità media del proprio intorno. Risultati sperimentali [Bibliografia 9] affermano l'utilità di combinare insieme più metodi di correlazione, al fine di ottenere delle corrispondenze affidabili e il più possibile veritiere.

La combinazione di metodi diversi non permette il riuso massiccio dei calcoli eventualmente già effettuati in precedenza, per tal motivo, se si adopera un unico processore, il tempo di elaborazione risulterà pari alla somma dei singoli valori di costo temporale. Il costo, invece, non aumenta se si adopera una architettura parallela di calcolo: in questo caso, il costo temporale è pari al più alto dei distinti costi di elaborazione dei singoli algoritmi che si vanno a combinare.

Capitolo
IV

Implementazione e
risultati sperimentali



Implementazione e risultati sperimentali

IV.1 L'algoritmo di correlazione

La correlazione è la parte più costosa, sia in termini di tempo di elaborazione sia in termini di risorse necessarie, di tutte le attività legate alla stereo visione [Bibliografia 9 e 11]: si deve cercare, perciò, di ottimizzare il più possibile l'algoritmo di correlazione.

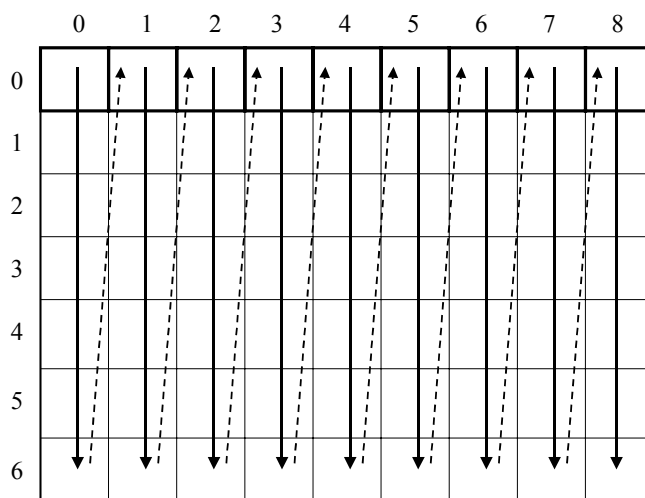


Figura IV.1 – Ordine di processamento per i pixels dell'immagine

In tale algoritmo i pixel vengono processati secondo l'ordine riportato in figura IV.1, dove le frecce indicano il cammino che si compie nell'analizzare l'intera immagine. La scansione avviene per colonne, in modo da poter sfruttare una importante proprietà della matrice di pixels che rappresenta l'immagine [Bibliografia 5]. Al fine di evidenziare tale proprietà, riprendiamo la figura II.4 precedentemente adoperata che, per comodità, viene mostrata di nuovo.

Matrice 1
immagine sinistra

0	1	3	7	6	4	8	9	6	5	4	6	7
1	5	8	5	3	5	7	9	8	6	4	3	4
2	6	7	9	0	6	6	3	5	6	7	8	9
3	7	5	4	3	4	4	7	6	5	4	4	9
4	7	3	7	8	9	9	1	2	3	0	0	0
5	4	5	6	7	8	4	5	6	7	7	5	1
6	9	4	6	3	4	6	7	8	3	4	6	7
7	2	3	4	5	6	7	7	5	4	3	2	4
8	3	7	4	9	9	9	9	5	5	5	3	4
9	5	6	7	5	4	6	8	9	4	5	7	7
10	8	5	8	5	8	5	6	5	7	8	5	8
11	5	5	3	4	8	8	9	5	3	0	0	0
	0	1	2	3	4	5	6	7	8	9	10	11

Matrice 2
immagine destra

0	1	4	4	7	9	0	0	8	5	7	3	2
1	2	6	5	3	5	9	8	8	8	8	5	4
2	5	6	8	9	4	5	2	3	4	1	1	1
3	4	5	6	7	8	9	0	8	5	4	4	1
4	1	3	1	1	9	6	9	2	3	5	0	4
5	4	5	5	7	8	9	0	8	5	3	6	7
6	3	2	6	3	4	3	2	4	5	6	4	3
7	4	5	6	5	7	7	5	6	4	7	2	5
8	5	7	6	6	5	5	6	7	9	6	3	2
9	3	4	5	6	4	4	8	4	6	7	8	7
10	2	4	8	4	4	4	4	5	2	3	4	6
11	5	5	6	4	6	8	6	5	6	0	6	0
	0	1	2	3	4	5	6	7	8	9	10	11

Si facciano le stesse ipotesi dell'esempio riportato al paragrafo II.2.1 sezione A e si assuma di adoperare il metodo SSD. I calcoli necessari per trovare la disparità del pixel p di coordinate riga-colonna (4, 5) sono di nuovo esibiti.

$$\begin{aligned} \text{Prima iterazione: } S_1 &= (4-8)^2 + (4-9)^2 + (7-0)^2 + \\ &\quad (9-9)^2 + (9-6)^2 + (1-9)^2 + \\ &\quad (8-8)^2 + (4-9)^2 + (5-0)^2 = 213 \end{aligned} \quad d = 0$$

$$\begin{aligned} \text{Seconda iterazione: } S_2 &= (4-7)^2 + (4-8)^2 + (7-9)^2 + \\ &\quad (9-1)^2 + (9-9)^2 + (1-6)^2 + \\ &\quad (8-7)^2 + (4-8)^2 + (5-9)^2 = 151 \end{aligned} \quad d = 1$$

$$\begin{aligned} \text{Terza iterazione: } S_3 &= (4-6)^2 + (4-7)^2 + (7-8)^2 + \\ &\quad (9-1)^2 + (9-1)^2 + (1-9)^2 + \\ &\quad (8-5)^2 + (4-7)^2 + (5-8)^2 = 233 \end{aligned} \quad d = D_{\text{MAX}} = 2$$

Si calcoli ora la disparità per il punto p' di coordinate riga-colonna (5, 5), ovvero per il pixel immediatamente sottostante il pixel p di cui sopra.

$$\begin{aligned} \text{Prima iterazione: } S_1' &= (9-9)^2 + (9-6)^2 + (1-9)^2 + \\ &\quad (8-8)^2 + (4-9)^2 + (5-0)^2 + \\ &\quad (4-4)^2 + (6-3)^2 + (7-2)^2 = 157 \end{aligned} \quad d = 0$$

$$\begin{aligned} \text{Seconda iterazione: } S_2' &= (9-1)^2 + (9-9)^2 + (1-6)^2 + \\ &\quad (8-7)^2 + (4-8)^2 + (5-9)^2 + \\ &\quad (4-3)^2 + (6-4)^2 + (7-3)^2 = 143 \end{aligned} \quad d = 1$$

$$\begin{aligned} \text{Terza iterazione: } S_3' &= (9-1)^2 + (9-1)^2 + (1-9)^2 + \\ &\quad (8-5)^2 + (4-7)^2 + (5-8)^2 + \\ &\quad (4-6)^2 + (6-3)^2 + (7-4)^2 = 241 \end{aligned} \quad d = D_{\text{MAX}} = 2$$

Si faccia caso ai calcoli in neretto per p e p' : quando si va a calcolare la disparità per p' , vengono rifatti alcuni calcoli che erano stati già effettuati per p . Si manifesta una proprietà di *ripetitività* nel meccanismo di calcolo delle disparità.

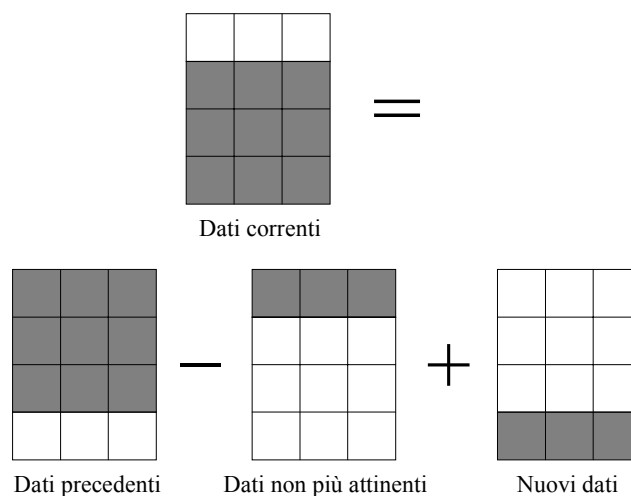


Figura IV.2 – I calcoli già effettuati possono essere riusati

Il salvataggio in memoria dei conteggi già eseguiti permette di abbattere notevolmente la complessità dell'algoritmo, poiché per ogni nuovo pixel che si va ad esaminare, una parte dei calcoli è già pronta in memoria e va soltanto richiamata. Bisogna soltanto riporre grande cura nello scartare la parte di dati precedentemente calcolati che non sono più relativi al pixel corrente e nell'inserire la parte nuova di calcoli (precedente figura IV.2). In questo modo, vengono effettuati calcoli completi per l'intera finestra di correlazione esclusivamente per i pixels che si trovano in cima ad ogni colonna della matrice che rappresenta l'immagine (si tratta dei pixels evidenziati in figura IV.1 all'inizio del presente capitolo).

IV.2 Il test dell'algoritmo

Ogni algoritmo verrà testato prendendo in considerazione tre tipologie di immagini (figura IV.3).



Tipologia I

Dim.

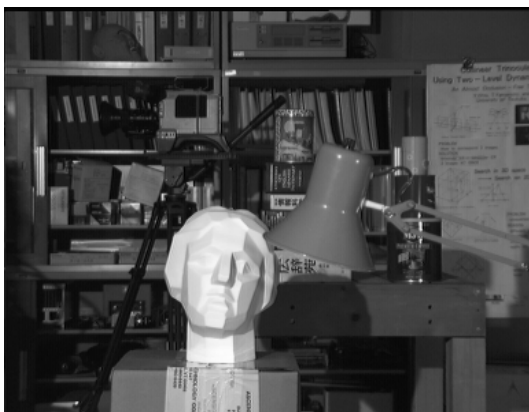
320 x 240



Tipologia II

Dim.

320 x 240



Tipologia III

Dim.

384 x 288

Figura IV.3 – Le tre tipologie di immagini con relative dimensioni in pixel

La prima (I) e la terza (III) tipologia rappresentano esempi di immagini reali, la seconda (II), al contrario, rappresenta una coppia di immagini virtuali realizzata al computer.

Il grado di correttezza sarà valutato in base ai seguenti criteri:

- 1) Confronto visivo con l'immagine campione ottenuta tramite il metodo SSD puro, cioè senza l'uso di alcuna soglia e senza sfruttare le informazioni che provengono dall'analisi dell'intorno di ogni pixel [Capitolo I, paragrafo 2 definizione I.2].
- 2) Analisi visiva con valutazione della pulizia, delle regioni intorno ai bordi degli oggetti presenti nella scena e della presenza o meno di buchi [Capitolo I, paragrafo 2].
- 3) Tempo di elaborazione T con specifiche relative al sistema operativo, alla quantità di RAM ed al processore impiegati nella simulazione [Capitolo I, paragrafo 2 definizione I.3].

Inoltre, si utilizzerà un fattore di qualità che andrà a creare un ordinamento tra le immagini, al fine di produrre una "classifica" oggettiva delle migliori immagini di disparità. Infatti, per valutare la bontà di un algoritmo di correlazione o, semplicemente, gli effetti prodotti dalla variazione di alcuni parametri, è necessario stabilire una misura quantitativa per stimare la qualità delle corrispondenze calcolate. Un generale approccio a tale problema è quello di calcolare gli errori che vengono fatti rispetto all'immagine ground truth [Bibliografia 15].

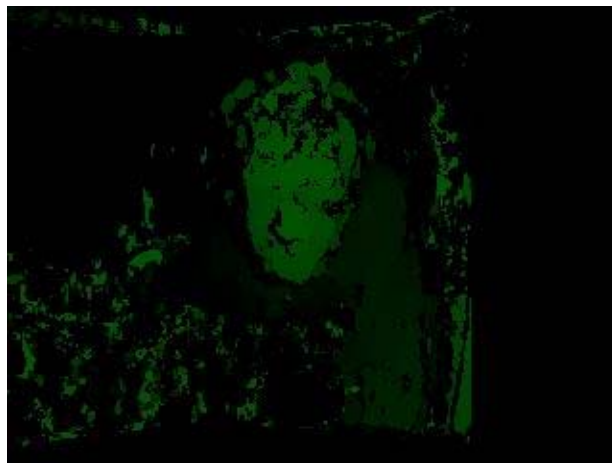
Definizione IV.1 Definiamo errore R di una immagine di disparità rispetto alla relativa mappa ground truth il valore

$$R = \left(\frac{1}{N} \sum_{x,y} |d_C(x,y) - d_T(x,y)|^2 \right)^{1/2}$$

dove N è il numero totale dei pixels, $d_C(x,y)$ è il valore di disparità calcolato, $d_T(x,y)$ è il valore di disparità della mappa ground truth ed R è misurato in unità di disparità.

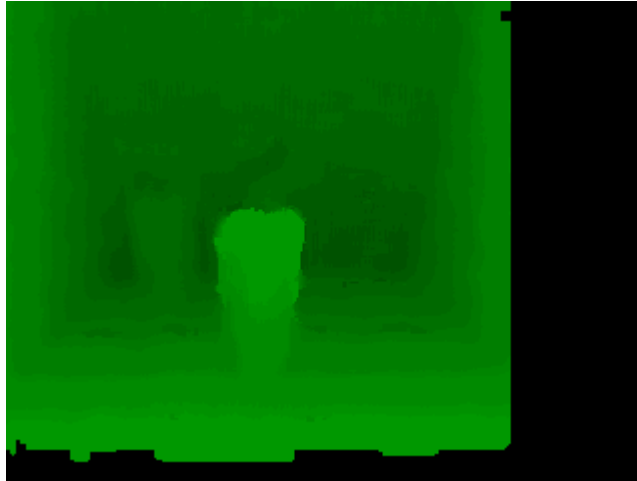
E' chiaro che più R è piccolo, più l'immagine di disparità è corretta, poiché R misura la quantità di corrispondenze errate.

Vengono, di seguito, riportate le mappe ground truth che verranno utilizzate nella valutazione degli algoritmi. Per le prime due mappe è necessario introdurre un *fattore di normalizzazione*, poiché non si tratta di ground truth vere e proprie, ma di immagini di disparità ottenute con un algoritmo che produce risultati molto attendibili [Bibliografia 2 e 17]. La normalizzazione si rende necessaria a causa delle discrepanze presenti nell'assegnare i valori di verde relativi alle varie disparità.



Fattore di
normalizzazione
2.6

Figura IV.4 – Ground truth per tipologia I
(fornita dal Professor Luca Iocchi)



Fattore di
normalizzazione
4.88

Figura IV.5 – Ground truth per tipologia II
(fornita dall'Ingegnere Luigi Scozzafava)

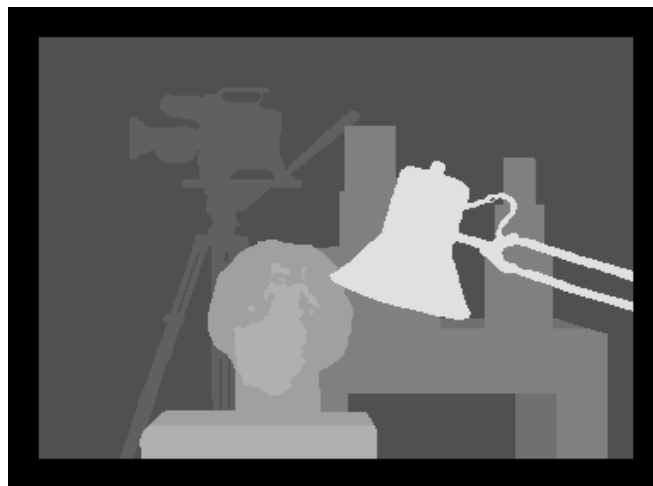


Figura IV.6 – Ground truth per tipologia III
(fornita dall'Università di Tsukuba)

IV.3 Specifiche tecniche

I risultati per tutti i metodi di correlazione che saranno, successivamente, riportati sono ottenuti adoperando un calcolatore con processore Intel Celeron da 800 MHz, memoria RAM da 120 MB e sistema operativo Windows Millennium Edition. Il linguaggio di programmazione in cui sono scritti i vari algoritmi di correlazione è il Java della Sun Microsystems. L'utilizzo di questo linguaggio rende l'agente software in grado di funzionare su qualsiasi sistema operativo, tuttavia il tempo di elaborazione risente di tale scelta, poiché il Java è noto essere abbastanza lento; i tempi di elaborazione saranno, pertanto, elevati.

Poiché il principale scopo del presente lavoro è quello di confrontare i vari algoritmi, non è necessario realizzare codice ottimizzato, né usare linguaggi molto più veloci del Java, come il C o il C++: infatti interessa conoscere e valutare i tempi di elaborazione **relativi**, ovvero si vuole stabilire se, **a parità di condizioni**, un algoritmo *A* sia temporalmente più efficiente di un altro algoritmo *B*, non se *A* sia un algoritmo di per sé efficiente in maniera assoluta.

IV.4 Il metodo SSD puro (immagini campione)

Risultati per Tipologia I

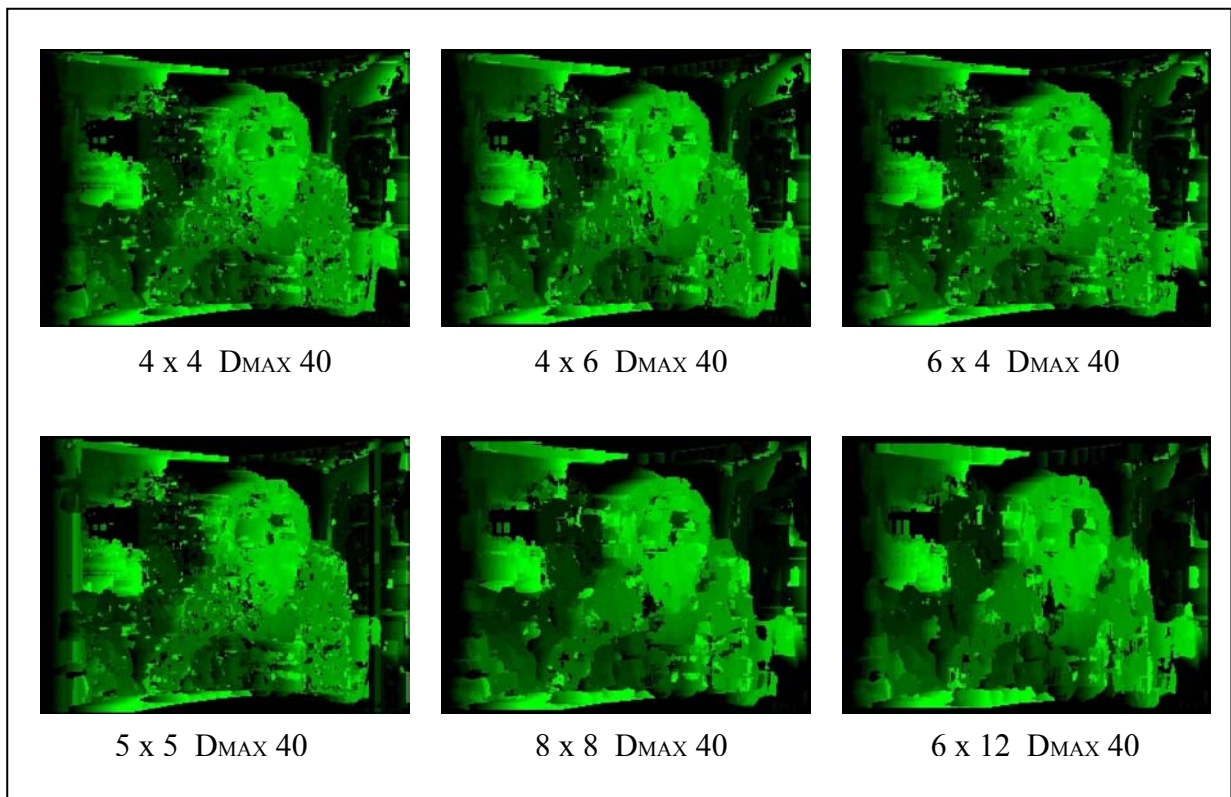


Tabella IV.1 – Risultati per tipologia I

SSD Puro Tipo I

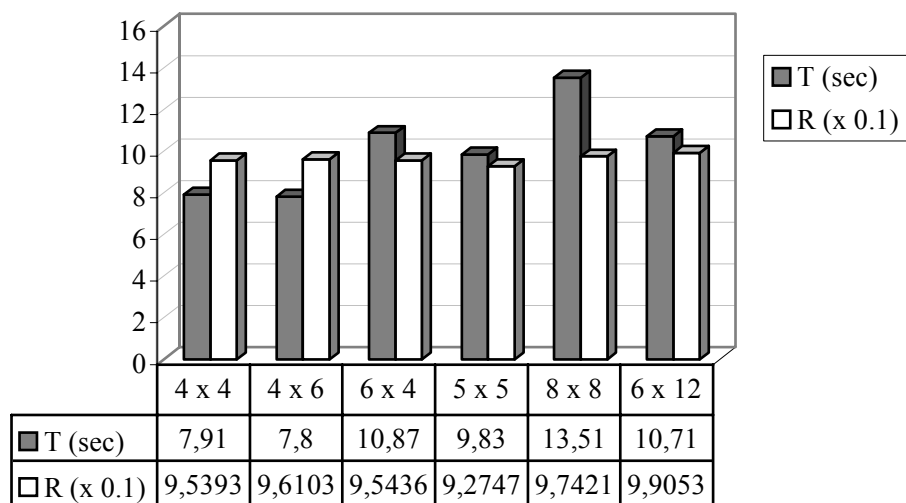


Tabella IV.2 – *T* ed *R* per tipologia I

Valutazione per Tipologia I

Le immagini campione

Le immagini di disparità che sono state ottenute con il metodo SSD puro rappresentano le immagini campione che saranno usate da riferimento per il confronto visivo [Capitolo I, definizione I.2] cui verranno sottoposte le immagini ottenute adoperando algoritmi e metodi diversi dall'SSD puro.

Parametri

Per tutte le sei immagini il valore di D_{MAX} è settato a 40, tale valore è stato trovato sperimentalmente, analizzando preventivamente la coppia stereo di immagini alla ricerca del massimo valore di disparità. Questo procedimento si è reso necessario poiché non sono noti i valori di calibrazione del sistema di telecamere che ha ripreso la scena; se si conoscessero i parametri adottati per la calibrazione delle telecamere [Appendice A, paragrafo 3.1], la massima disparità sarebbe un valore calcolabile con precisione.

Analisi visiva

L'analisi visiva stabilisce chiaramente la presenza di rumore sullo sfondo di tutte le sei immagini, rumore che si concentra prevalentemente sulla parte sinistra delle mappe. Tale rumore è presente indipendentemente dalla dimensione del frame di ricerca che si adopera per trovare la disparità, infatti esso è dovuto alla ripetitività della texture della parete che fa da sfondo all'immagine. Sono presenti diversi buchi in tutte le immagini, in particolare sulla fronte e sulla manica sinistra del soggetto in primo piano. Le figure 8x8 e 6x12 soffrono, inoltre, di un notevole buco intorno al collo di tale soggetto.

I contorni sulla destra della scena sono abbastanza delineati, tranne nella zona tra manica sinistra e monitor, dove è presente una notevole occlusione: il monitor si vede bene nell'immagine stereo destra, mentre è quasi assente nell'immagine sinistra [per una conferma si dia una occhiata alla coppia stereo riportata al paragrafo 2 di questo capitolo].

I bordi sulla parte sinistra della mappa sono meno delineati, a causa della zona di occlusione che affligge la parte sinistra del volto in primo piano. Ciò è dovuto al fatto che la zona sinistra della faccia, che si vede bene nell'immagine sinistra, è completamente occlusa nell'immagine destra, ciò causa l'assenza di un bordo netto tra contorni del viso e background della scena ed, in più, aumenta le dimensioni della faccia.

Tempo di elaborazione T

L'analisi dei vari tempi di calcolo evidenzia la maggiore efficienza temporale degli algoritmi che adoperano frame rettangolari con valori di ampiezza minori dei valori di altezza. Questo risultato è particolarmente evidente se si confrontano i tempi per 4x6 e 6x4, di fatto risulta molto più veloce 4x6 a parità di area analizzata; identico discorso vale per 8x8 e 6x12 le cui aree sono confrontabili.

Il minor costo temporale è 7,8 secondi, fatto registrare usando un frame 4x6, la peggiore prestazione tocca all'algoritmo con frame 8x8.

Errore R

I valori per R sono abbastanza livellati, con la migliore prestazione appannaggio del frame 5x5. La peggiore prestazione si ottiene con il frame 6x12, il che può essere facilmente confermato da una veloce analisi visiva. Infatti, il 6x12 elimina i piccoli errori sulla camicia presenti nei frames ad area minore, ma perde molta informazione sulle caratteristiche del volto, che appare troppo uniforme (il naso quasi scompare).

Risultati per Tipologia II

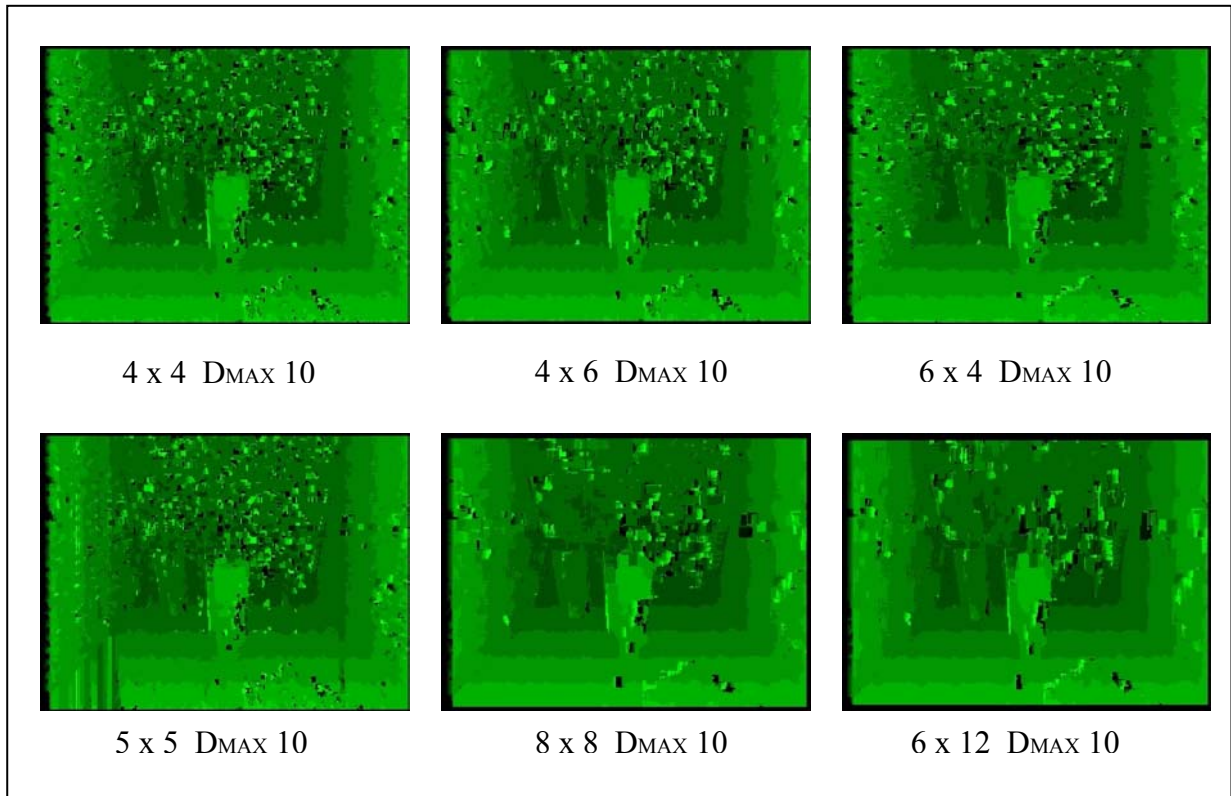


Tabella IV.3 – Risultati per tipologia II

SSD Puro Tipo II

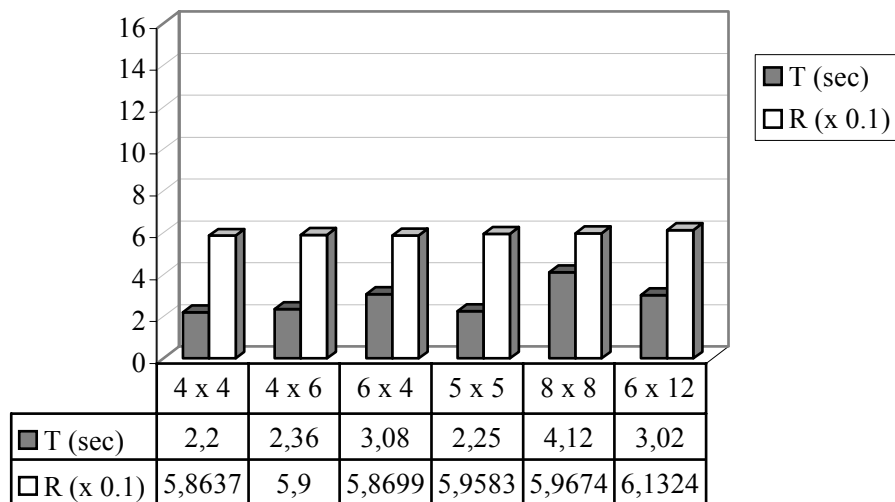


Tabella IV.4 – T ed R per tipologia II

Valutazione per Tipologia II

Parametri

La coppia di immagini che forma la Tipologia II è virtuale, per cui non esistono parametri relativi ad alcuna telecamera. Per tale motivo, come massimo valore di disparità è stato scelto sperimentalmente, sempre attraverso un'analisi della coppia stereo, il valore 10. Questo valore è molto inferiore al D_{MAX} presente in Tipologia I che è 40. La minore ampiezza di ricerca rende molto più veloce la correlazione per Tipologia II rispetto a Tipologia I perché l'algoritmo compie un numero di passi 4 volte inferiore.

Analisi visiva

L'analisi visiva stabilisce, come per Tipologia I, la presenza di rumore sullo sfondo di tutte le sei mappe, rumore che si concentra prevalentemente nella parte centrale delle figure. Tale rumore è dovuto alla ripetitività della texture della parete che fa da fondale all'immagine virtuale. Poiché la parete è a mattoni, la trama dello sfondo risulta molto uniforme e ripetitiva, cosa che rende difficile all'algoritmo calcolare con precisione la giusta disparità. Il rumore si attenua adoperando frames più grandi, poiché essi permettono di considerare un'area più estesa che dà maggiori garanzie su una texture molto uniforme, in quanto aumenta l'informazione a disposizione.

Sono presenti diversi buchi in tutte e sei le mappe, in particolare sulla parete di destra, sul pavimento (sempre a destra) ed al centro del vaso in primo piano. Anche questi buchi sono determinati dalla texture troppo uniforme di pareti e pavimento.

I contorni sono abbastanza delineati per le prime quattro immagini, al contrario, utilizzando i frames 8x8 e 6x12 si perde un po' dei bordi dei due vasi, soprattutto con il 6x12.

L'immagine 5x5 presenta una notevole zona di rumore in basso a sinistra, cosa che non accade per le altre immagini.

Tempo di elaborazione T

L'analisi dei vari tempi di calcolo, così come per Tipologia I, evidenzia la maggiore efficienza temporale degli algoritmi che adoperano frame rettangolari con valori di ampiezza minori dei valori di altezza. Questo risultato è particolarmente evidente se si confrontano i tempi per 4x6 e 6x4. A parità di area analizzata risulta molto più veloce 4x6 di 6x4; identico discorso vale per 8x8 e 6x12 le cui aree sono confrontabili.

Il minor costo temporale è 2,2 secondi, fatto registrare usando un frame 4x4, la peggiore prestazione tocca all'algoritmo con frame 8x8, il quale impiega quasi il doppio del tempo. Come ci si attendeva, i costi temporali sono più o meno quattro volte inferiori a quelli necessari per Tipologia I, il che è dovuto al valore di D_{MAX} , quattro volte inferiore a quello di Tipologia I.

Errore R

I valori per R premiano, anche se di poco, il frame 4x4 che risulta essere il più efficiente ed efficace tra i sei. La peggiore prestazione è per 6x12, che nonostante migliori il rumore sullo sfondo, perde troppa informazione sui particolari del vaso in primo piano. Il frame 5x5 risente della zona di rumore nell'angolo in basso a sinistra e si piazza in penultima posizione rispetto al fattore R .

Risultati per Tipologia III

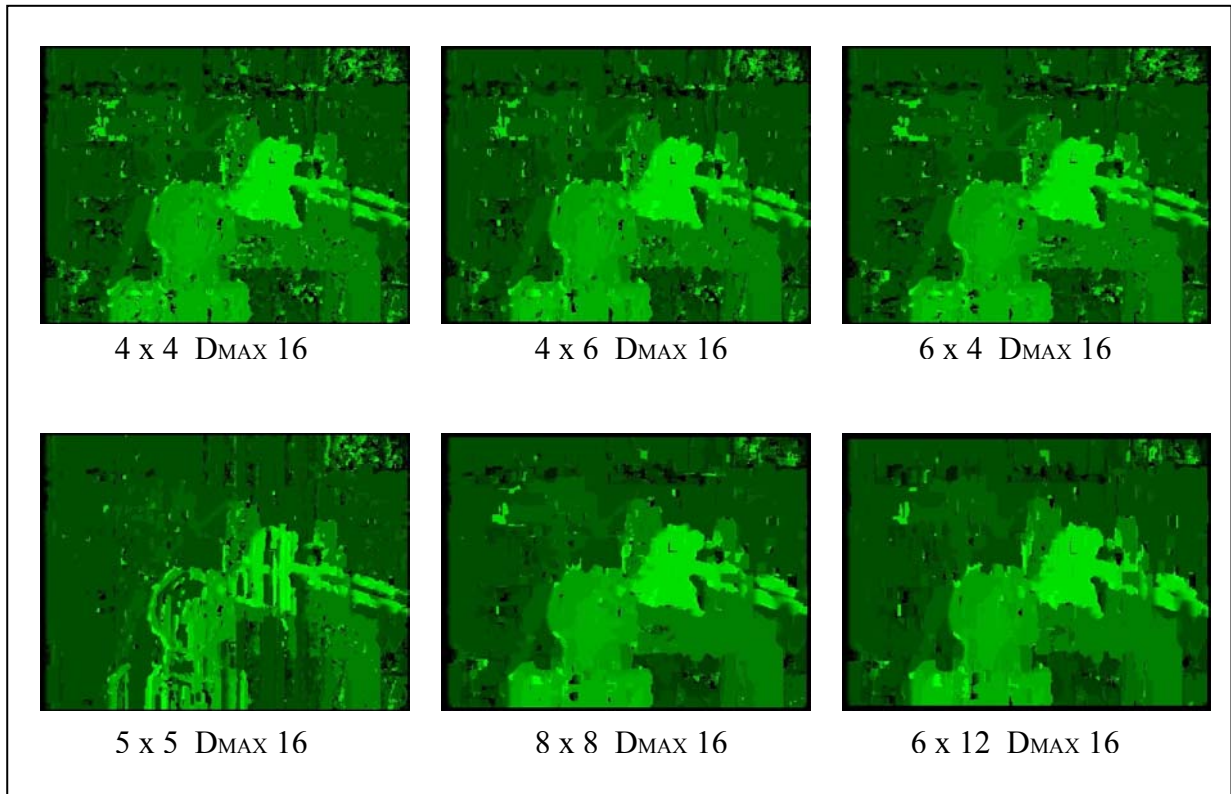


Tabella IV.5 – Risultati per tipologia III

SSD Puro Tipo III

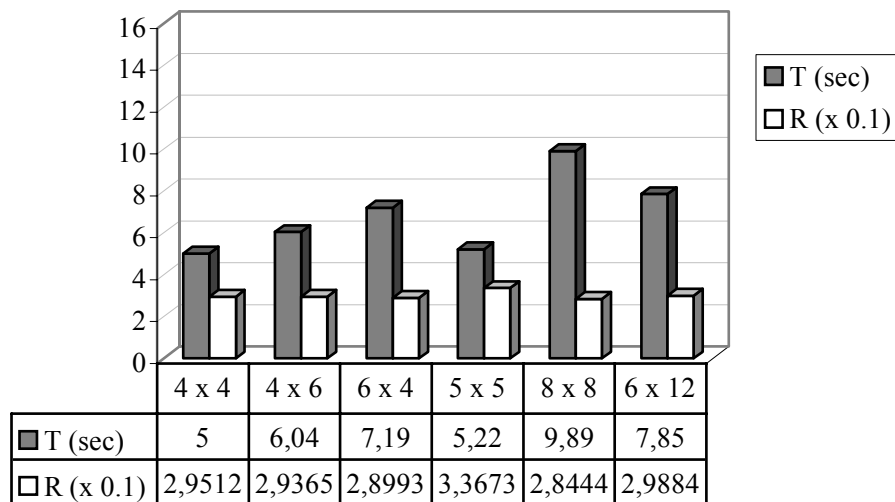


Tabella IV.6 – T ed R per tipologia III

Valutazione per Tipologia III

Parametri

L'analisi della coppia stereo ha evidenziato un valore massimo di disparità pari a 16. Questo valore si colloca all'incirca a metà tra il D_{MAX} di Tipologia I e quello di Tipologia II, il che dovrebbe portare a costi temporali che si collocano nel mezzo di quelli per le precedenti tipologie. Purtroppo, anche per questa coppia di immagini non sono disponibili i parametri estrinseci ed intrinseci del sistema di telecamere, per cui il valore 16 è stato ottenuto, come per le coppie stereo precedenti, "manualmente".

Analisi visiva

Il primo impatto visivo porta immediatamente a considerare la mappa 5x5 come la peggiore, sensazione che è confermata dal fattore R che indica proprio la 5x5 come immagine meno accurata. In tutte e sei le mappe è presente rumore in maniera considerevole vicino all'obiettivo della telecamera posta dietro il busto. Tale rumore è dovuto ad occlusione: infatti, se si osserva attentamente l'immagine destra della coppia stereo, si nota che in prossimità dell'obiettivo è occlusa una parte dei libri nello scaffale che invece è visibile nell'immagine destra. Altro rumore si riscontra sui bordi a sinistra della pila di libri dietro il busto, anche in questo caso, la colpa è dei libri nello scaffale retrostante: parte di un libro, nell'immagine sinistra, è occlusa dalla pila di oggetti sul tavolino.

Le immagini con frames 8x8 e 6x12 risentono in maniera minore del rumore sullo sfondo, al tempo stesso, però, se si fa caso ai bracci della lampada, appare evidente che frames più piccoli, come il 4x4, riescono a mostrare con chiarezza tali bracci che invece sfumano man mano che si aumenta l'area della finestra di correlazione.

I buchi sono raggruppati nella parte superiore della scena, in particolare nell'angolo in alto a destra, dove la parete liscia e di colore uniforme non fornisce elementi all'algoritmo per trovare la giusta disparità. Altri buchi di una certa rilevanza sono presenti sul basamento del busto.

Tolta la 5x5, tutte le mappe forniscono un risultato che può considerarsi più che accettabile, in quanto non è difficile individuare i principali oggetti presenti nella scena.

Tempo di elaborazione T

Ancora una volta, appare evidente che adoperando frames rettangolari, è più efficiente un rettangolo con altezza maggiore dell'ampiezza rispetto ad uno che si sviluppa in ampiezza: si vedano i tempi di elaborazione per le coppie {4x6, 6x4} ed {8x8, 6x12}.

Il minor costo temporale è 5 secondi, fatto registrare usando un frame 4x4, la peggiore prestazione tocca all'algoritmo con frame 8x8, il quale impiega 9,89 secondi. Come ci si attendeva, i costi temporali sono più o meno a metà tra quelli necessari per Tipologia I e quelli di cui necessita Tipologia II.

Errore R

I valori per R indicano 8x8 come miglior dimensione per il frame di correlazione, tale risultato è dovuto alla forte riduzione di rumore che si ottiene adoperando finestre di questa area. Il peggiore risultato si ha con il 5x5, l'immagine risulta davvero molto disturbata e si fa fatica a distinguere gli oggetti.

IV.5 Il metodo SSD con soglia interna

Risultati per Tipologia I

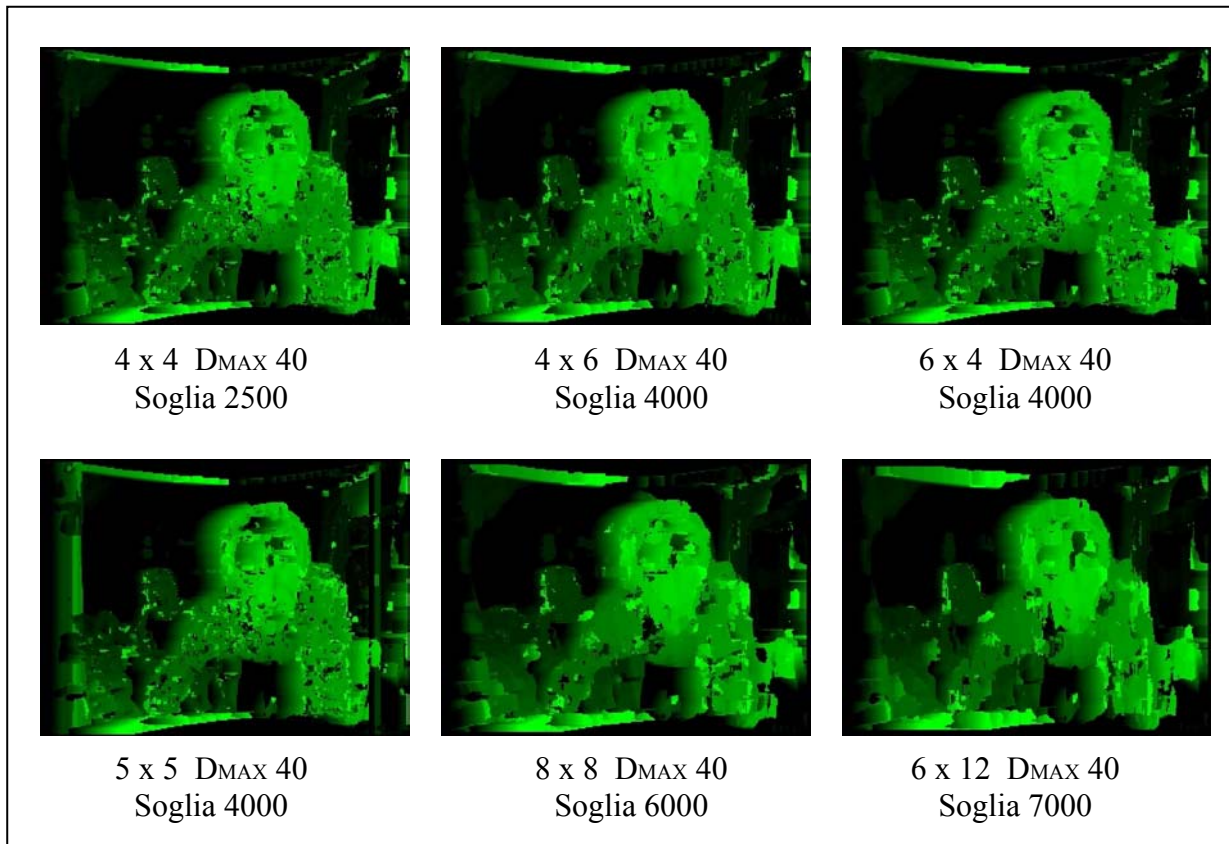


Tabella IV.7 – Risultati per tipologia I

SSD Soglia Tipo I

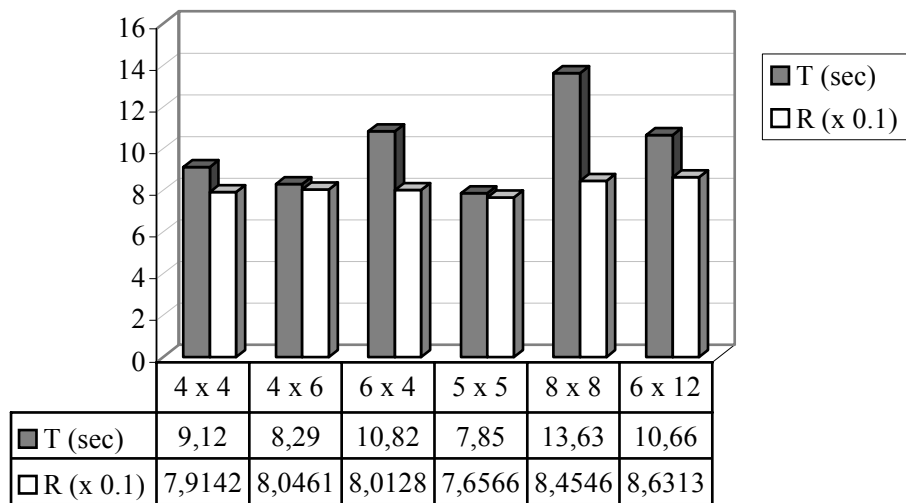


Tabella IV.8 – T ed R per tipologia I

Valutazione per Tipologia I

Parametri

L'introduzione di una soglia interna si rende necessaria se si vuole eliminare il rumore dovuto a texture ripetitiva. Il principale problema della soglia interna è la sua intrinseca mancanza di oggettività, poiché il valore da assegnare alla soglia dipende dall'immagine che si considera e dalla dimensione della finestra di ricerca che si va ad usare. I valori riportati sotto ogni immagine indicano soglie che sono state trovate empiricamente, ripetendo più volte il calcolo della mappa di disparità e selezionando i valori che fornivano i migliori risultati. Sarebbe possibile avvalersi di un valore di disparità normalizzato che vada bene per qualsiasi frame si vada a considerare, tuttavia i risultati ottenuti sperimentalmente non hanno portato all'individuazione di una costante che desse esiti soddisfacenti per tutti e sei i frames.

Analisi visiva

L'analisi visiva mette subito in rilievo la scomparsa del rumore che era presente nella parte sinistra dell'immagine. L'assenza di rumore rende i bordi degli oggetti più nitidi e rende senz'altro più chiara la struttura spaziale della scena. I bordi del viso appaiono più marcati, anche se rimane la tendenza nell'aumentare il volume della testa, a causa dell'inevitabile occlusione di una parte di essa. Unico neo dell'introduzione della soglia interna è la comparsa di un buco nella zona superiore della testa, generato dall'uniformità del colore dei capelli. Nella 5x5 si notano ora molto bene particolari del viso che non erano evidenti precedentemente, in tutte e sei le immagini risaltano maggiormente i bordi dei pantaloni e della camicia.

Confronto con l'immagine campione

Il confronto con le immagini campione [precedente paragrafo IV.4] avvalsa l'utilità dell'inserimento di una soglia quando si è di fronte a immagini che soffrono di grande rumore, in particolare mappe con presenza di zone ad elevata disparità laddove la disparità dovrebbe essere molto contenuta.

I problemi generati da occlusioni non possono essere risolti dalla soglia interna, infatti se si osserva la zona tra manica sinistra e monitor, non si nota alcun miglioramento, l'occlusione continua a provocare disparità elevate confondendo i bordi della camicia con quelli del monitor.

Tempo di elaborazione T

La migliore prestazione temporale, 7,85 secondi, si ottiene per il frame 5x5, mentre come al solito l'8x8 si rivela l'algoritmo più dispendioso. I tempi per i sei frames sono grosso modo identici a quelli ottenuti con l'SSD puro, solo per il 5x5 si nota un netto miglioramento, provocato probabilmente da una soglia molto azzeccata. Di fatto, se la soglia è ben scelta, l'algoritmo che sfrutta la soglia interna può risparmiare un considerevole numero di assegnazioni di variabili, anche se, in ogni caso, vengono confrontati con la finestra di riferimento tutti i frames possibili, poiché altrimenti non potrebbe attuarsi la politica del riutilizzo dei dati già calcolati [paragrafo IV.1].

Errore R

I valori per R indicano 5x5 come miglior dimensione per il frame di correlazione, tale risultato è dovuto alla migliore risoluzione che garantisce questo genere di finestre. Il peggiore risultato si ha con il 6x12, il quale tende a mascherare troppo i particolari del viso, anche se riesce a non perdere informazione sui capelli evitando il verificarsi del buco citato nell'analisi visiva. Il miglioramento rispetto all'SSD puro è netto ed è dimostrato dai valori di R che sono a vantaggio dell'SSD con soglia interna.

Risultati per Tipologia II

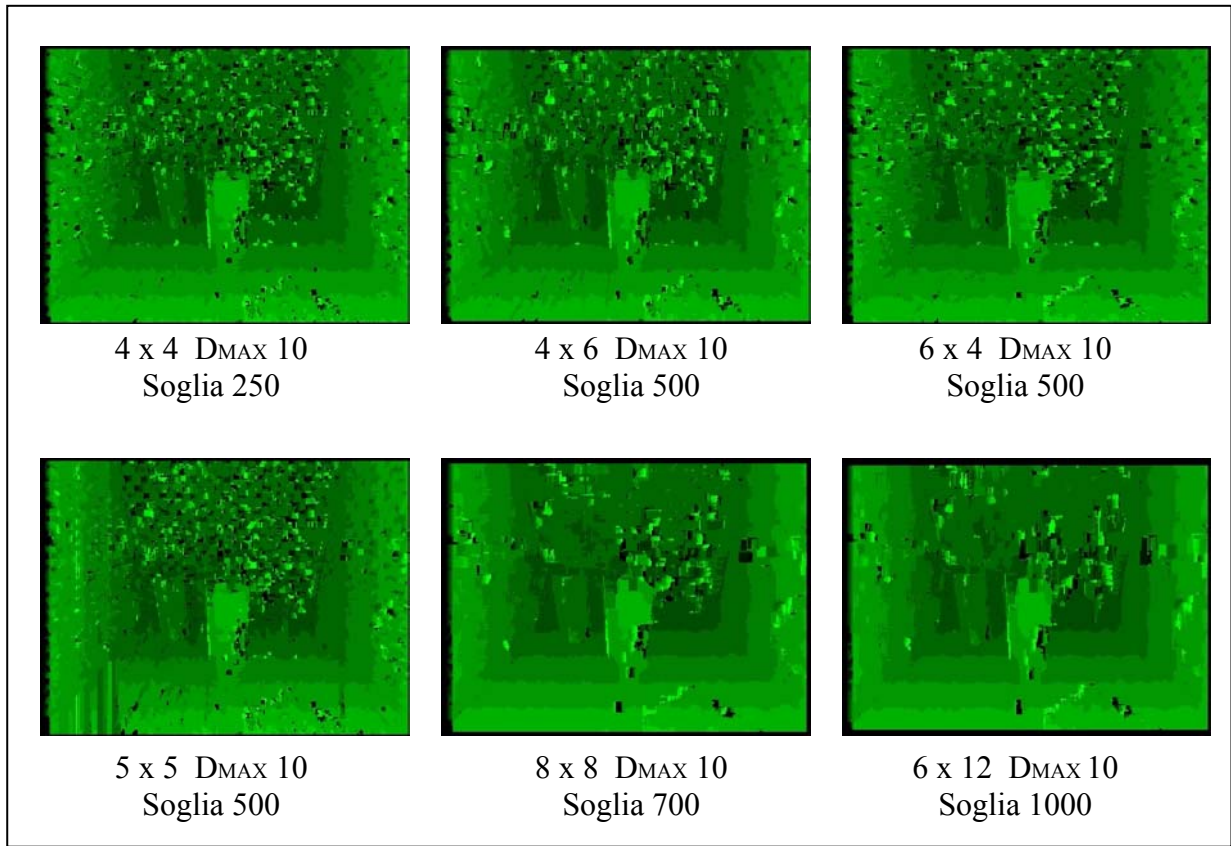


Tabella IV.9 – Risultati per tipologia II

SSD Soglia Tipo II

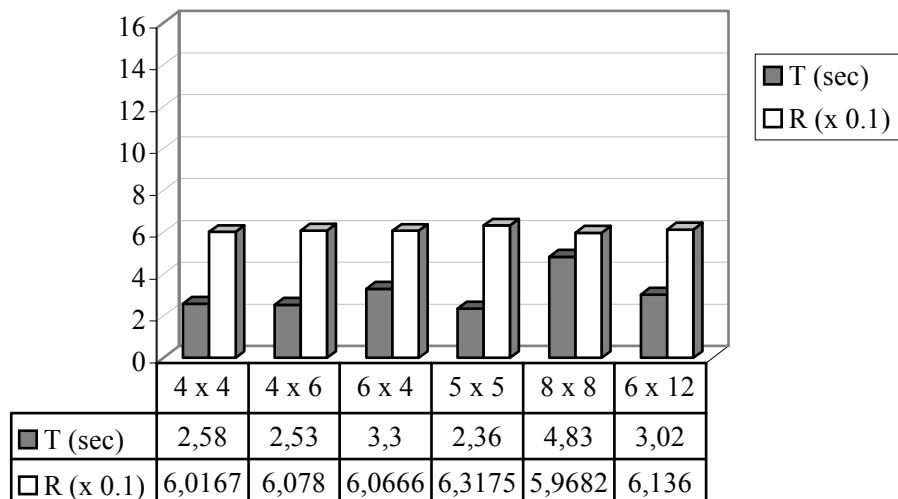


Tabella IV.10 – T ed R per tipologia II

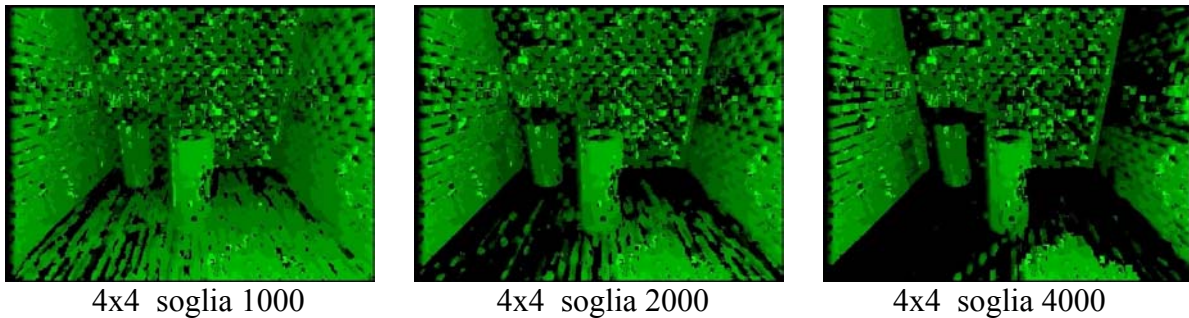
Valutazione per Tipologia II

Parametri

L'introduzione di una soglia interna non giova particolarmente alla Tipologia II, in virtù della particolare natura del pavimento e delle pareti. Lo sfondo, presentando texture estremamente ripetitiva, tende ad assumere disparità nulla man mano che si alza la soglia. Questo accade perché quando si va a cercare la corrispondenza per un pixel dello sfondo, la differenza tra i valori che scaturiscono dalla correlazione è molto sottile (perché lo sfondo è uniforme) ed una soglia può indurre l'algoritmo ad accontentarsi di un frame a disparità troppo bassa rispetto a quella reale. In tutte e sei le immagini è stata scelta una soglia piuttosto bassa, proprio per evitare che per troppi pixel l'algoritmo fosse tratto in inganno.

Analisi visiva

L'analisi visiva depone a sfavore dell'introduzione di un valore di soglia, infatti in generale non si ottiene alcun miglioramento, ma anzi c'è la tendenza a perdere informazione sulle pareti e sul pavimento, senza che il rumore venga eliminato. Più la soglia viene alzata più il pavimento tende a scomparire, senza che vi siano miglioramenti riguardo il rumore presente sullo sfondo. L'unico miglioramento si ha sui bordi dei vasi che divengono sempre più netti (esempio IV.1).



Esempio IV.1 – Mappe ottenute con diversi valori di soglia

Confronto con l'immagine campione

La differenza con le immagini campione è minima come dimostrano i valori di R . Si può affermare che per Tipologia II l'inserimento di una soglia non giova alla correttezza della mappa di disparità.

Tempo di elaborazione T

La migliore prestazione è realizzata dal frame 5x5, così come era accaduto per Tipologia I, mentre il peggiore in termini di costi temporali risulta essere il frame 8x8. Non si notano differenze con i tempi fatti registrare dall'SSD puro, il che avvalorava maggiormente l'ipotesi che non sia necessario inserire un valore di soglia per Tipologia II.

Errore R

I valori per R indicano 8x8 come miglior dimensione per il frame di correlazione, questo perché per 8x8 è stata scelta una soglia piuttosto bassa che non incide significativamente sui valori della mappa. Scegliendo una soglia più elevata, il valore di R per 8x8 si innalzerebbe inevitabilmente. Il peggior frame è 5x5 che risente di notevole rumore nella zona in basso a sinistra.

Risultati per Tipologia III

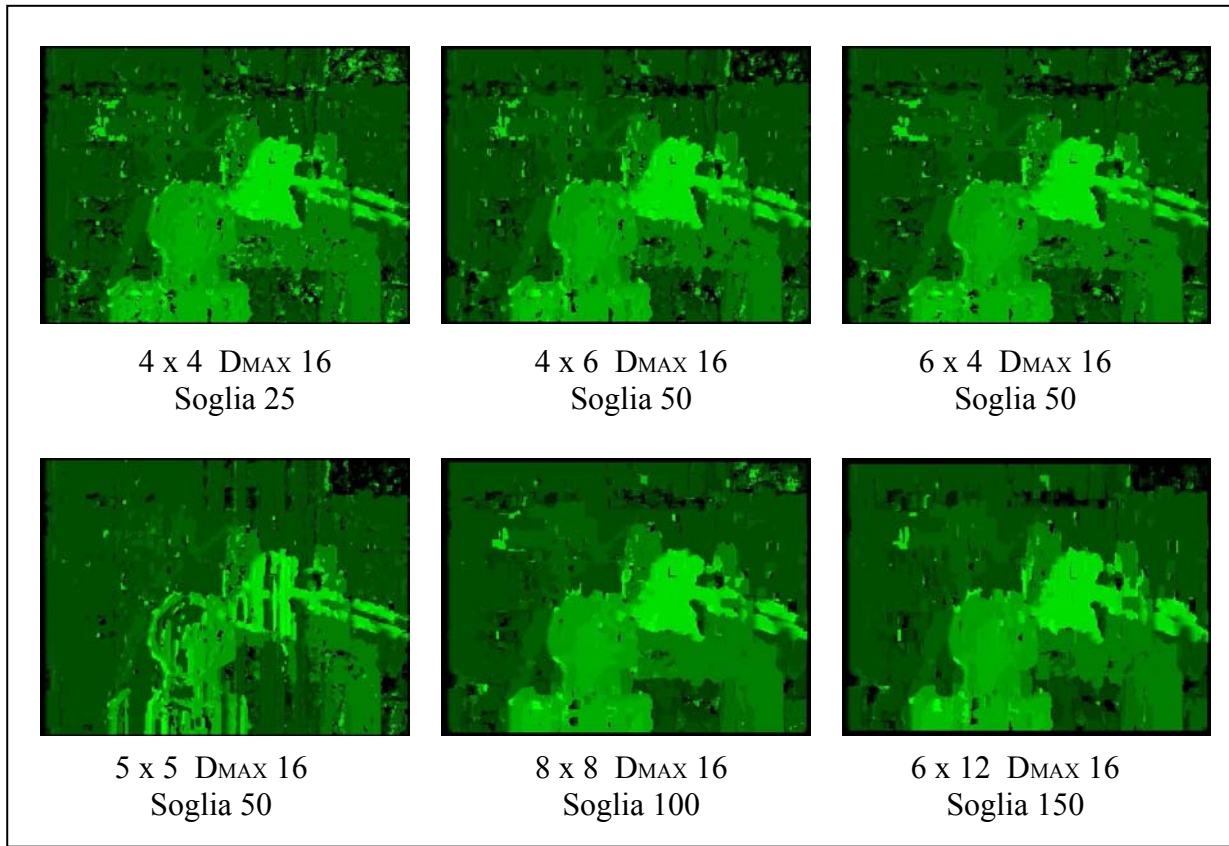


Tabella IV.11 – Risultati per tipologia III

SSD Soglia Tipo III

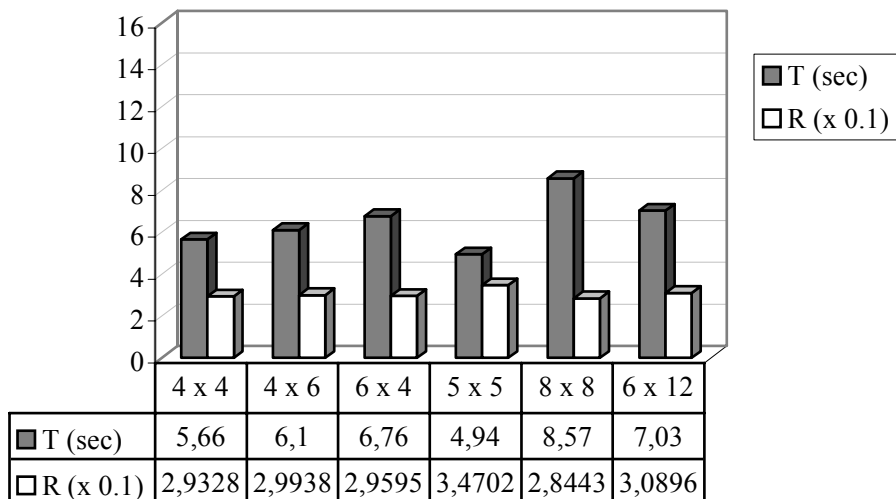


Tabella IV.12 – T ed R per tipologia III

Valutazione per Tipologia III

Parametri

L'introduzione di una soglia interna serve a migliorare leggermente le mappe di disparità nei casi 4x4 ed 8x8, negli altri casi non c'è miglioramento. I valori di soglia sono molto inferiori a quelli usati per Tipologia I e per Tipologia II perché valori simili a quelli usati nei precedenti casi comportano una eccessiva perdita di informazione sul tavolino e sul busto, come si evince con chiarezza dall'esempio sottostante.



4x4 soglia 100



4x4 soglia 300



4x4 soglia 1000

Esempio IV.2 – Mappe ottenute con diversi valori di soglia

Analisi visiva

Un primo sguardo alle immagini fa capire che sono ancora evidenti i problemi di occlusione intorno all'obiettivo della telecamera e persistono i buchi nella zona superiore. Il frame 8x8 soffre, tuttavia, meno di tutti di questo problema (il grande buco in alto a destra è meno evidente che negli altri casi) consentendo di ottenere risultati leggermente migliori rispetto all'SSD puro.

Confronto con l'immagine campione

La differenza con le immagini campione è minima come dimostrano i valori di R . Se si presta molta attenzione alle immagini si può notare che l'introduzione della soglia riesce a ridurre il rumore presente sulla sinistra delle gambe del tavolino e le macchie nell'angolo in basso a sinistra. Tale riduzione riguarda una parte molto piccola dell'intera immagine, ed il miglioramento che ne consegue non riesce a bilanciare la perdita di informazione che si ha nella parte alta.

Tempo di elaborazione T

La migliore prestazione è realizzata dal frame 5x5, però è evidente che non si tratta di una buona mappa, quindi possiamo considerare 4x4 il frame più rapido con 5,66 secondi di costo temporale. Ancora una volta, la peggiore prestazione è quella che si ottiene con il frame 8x8, anche si ha un certo miglioramento rispetto al costo che aveva l'SSD puro, in virtù di una soglia indovinata. Per i restanti frames i tempi sono in linea con quelli conseguiti con l'SSD puro.

Errore R

I valori per R indicano 8x8 come miglior dimensione per il frame di correlazione, seguito a ruota da 4x4. Questi due frame sono anche i soli per i quali l'inserimento della soglia permette di conseguire risultati migliori rispetto alle mappe campione. Il peggior risultato lo ottiene 5x5 che produce una immagine piuttosto scadente.

IV.6 Il metodo SSD con distanza di picco

Risultati per Tipologia I

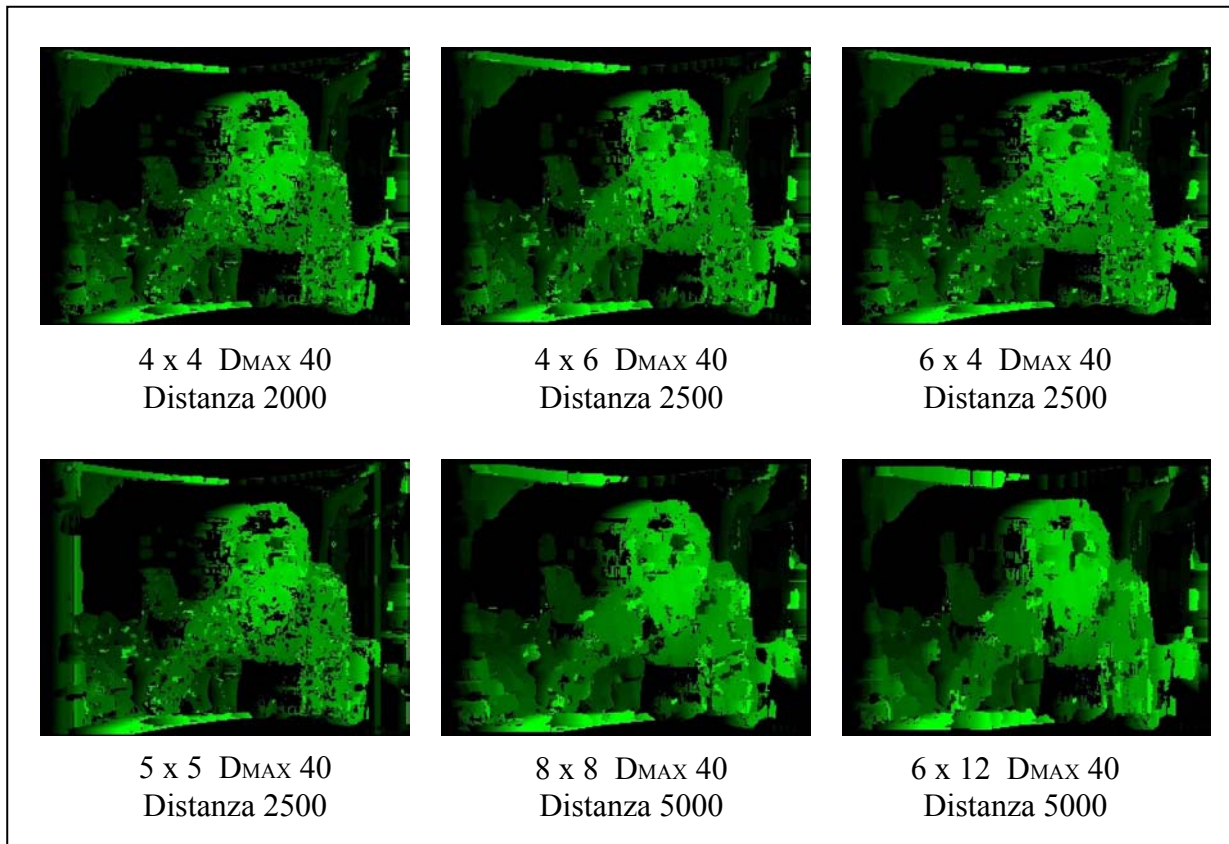


Tabella IV.13 – Risultati per tipologia I

SSD Picchi Tipo I

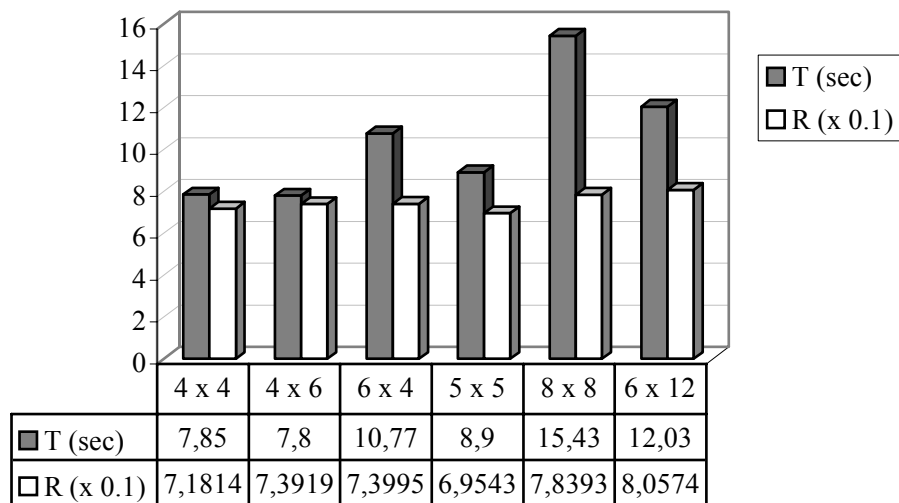
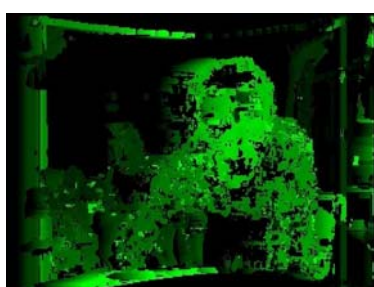


Tabella IV.14 – T ed R per tipologia I

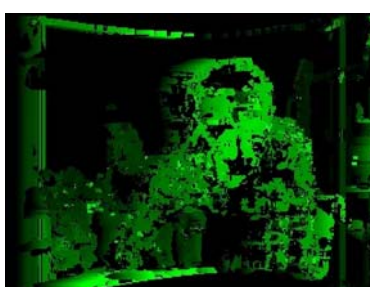
Valutazione per Tipologia I

Parametri

L'introduzione della distanza di picco serve a migliorare il riconoscimento delle oclusioni e ad impedire che l'algoritmo faccia confusione su texture ripetitive. Infatti, in presenza di oclusioni l'algoritmo è come se brancolasse nel buio alla ricerca di qualcosa che non in realtà non c'è. In tal modo, esso produce una funzione di correlazione con una serie di picchi di altezza simile (i quali indicano tutti match potenzialmente ottimi) che possono essere facilmente riconosciuti e scartati adottando la distanza di picco come criterio per gestire le disparità inquisite da oclusioni. Sono state scelte distanze di picco abbastanza elevate nel tentativo di ripulire la zona di occlusione che si trova a sinistra della faccia. Per eliminare l'occlusione che interessa il monitor nell'area in basso a destra dovrebbero essere scelti valori di picco talmente elevati che gran parte dell'informazione per il viso ed il busto andrebbe persa (si veda l'esempio IV.3).



5x5 distanza 5000



5x5 distanza 10000



5x5 distanza 50000

Esempio IV.3 – Mappe ottenute con diversi valori di distanza di picco

Analisi visiva

Effettivamente, la zona a sinistra del volto appare con bordi più netti rispetto alle mappe precedenti e il volume della testa non è più sovradimensionato. La zona tra manica sinistra della camicia e monitor è ancora vittima dell'occlusione anche se si nota un leggero miglioramento rispetto all'SSD con soglia. La zona dei capelli soffre notevolmente dell'inserimento della distanza di picco e, come conseguenza, si va a creare un buco di notevoli dimensioni. Un numero ragguardevole di buchi si viene a creare sulla camicia e sul volto, rendendo meno nitida la mappa di disparità in tutti e sei i casi.

Confronto con l'immagine campione

Rispetto all'immagine campione, utilizzando la distanza di picco viene eliminato il rumore sulla sinistra dello sfondo, si individuano con chiarezza i contorni dell'individuo in secondo piano, la zona tra manica della camicia e monitor si pulisce leggermente ed, infine, si nota nettamente la differenza tra pantaloni e camicia. Come contropartita, si ottiene una grossa perdita di informazione sui capelli ed un aumento dei buchi sul volto e sulla camicia.

Tempo di elaborazione T

La miglior prestazione tocca, con 7,8 secondi, a 4x6 che batte di gran lunga 6x4. La peggiore performance la compie 8x8 con 15,43 secondi.

Errore R

I valori di R indicano che l'SSD con distanza di picco è molto migliore dell'SSD puro ed anche dell'SSD con soglia interna. La migliore mappa è 5x5, la peggiore 6x12.

Risultati per Tipologia II

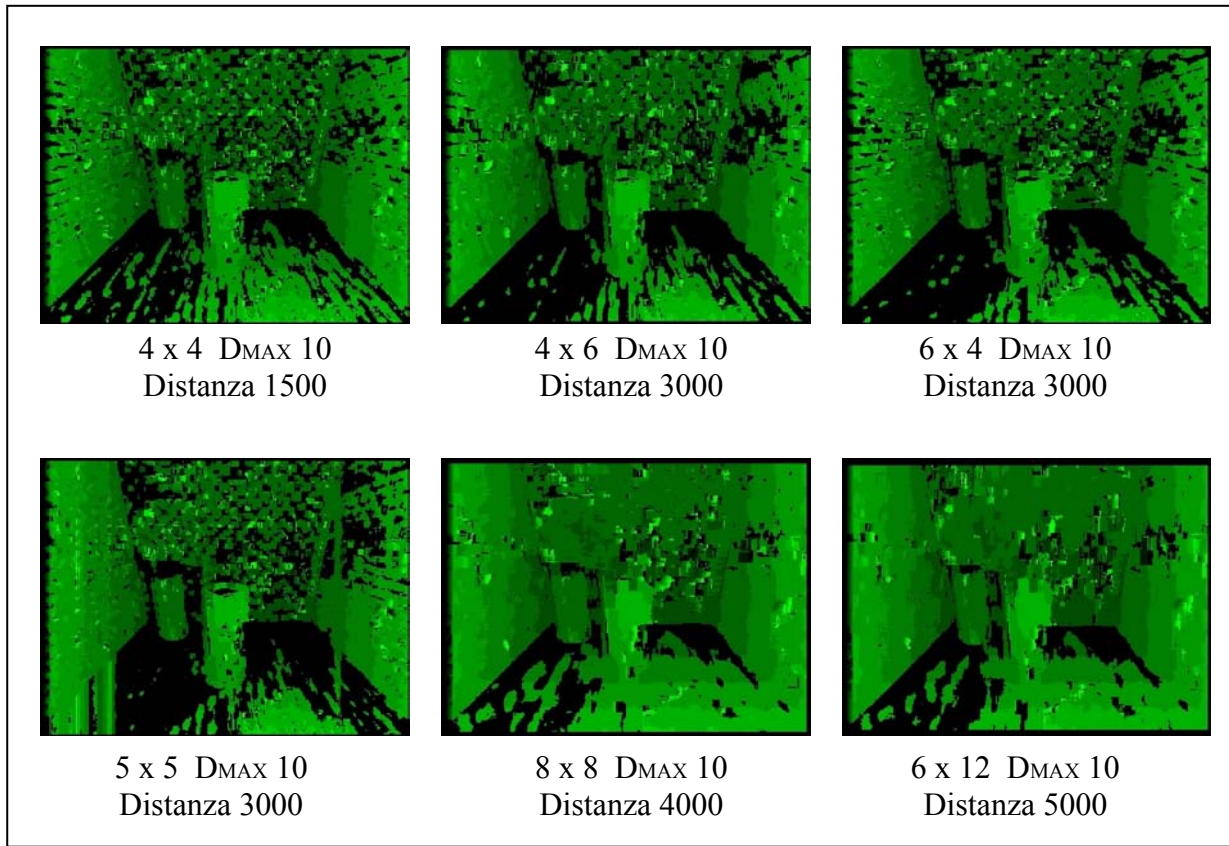


Tabella IV.15 – Risultati per tipologia II

SSD Picchi Tipo II

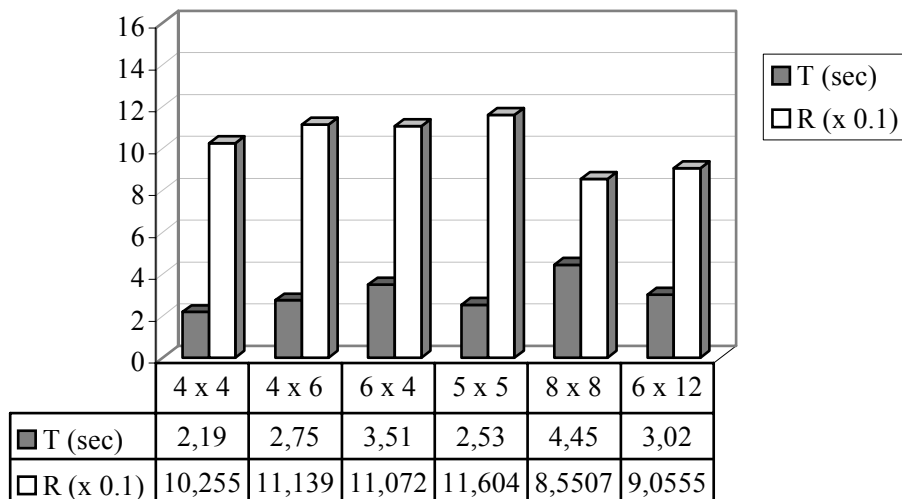


Tabella IV.16 – T ed R per tipologia II

Valutazione per Tipologia II

Parametri

Il grosso problema di cui soffre la mappa di disparità che viene generata dalla coppia stereo virtuale è il rumore sulla parete centrale. Questo rumore non è stato eliminato inserendo un valore di soglia, invece con la distanza di picco esso viene in parte arginato. I valori di soglia che sono stati scelti sono volutamente molto elevati per evidenziare due aspetti molto interessanti:

- 1) il rumore sullo sfondo viene ridotto e la zona di occlusione a sinistra del vaso in primo piano scompare completamente
- 2) si perde tantissima informazione riguardo il pavimento, che tende a scomparire all'aumentare del valore per la distanza di picco.

Analisi visiva

I valori molto elevati per la distanza di picco permettono di isolare i contorni dei due vasi e la scomparsa del pavimento rende più esplicita la profondità della sala dove sono collocati tali vasi. Questo è un aspetto molto interessante per un osservatore umano, mentre un agente software che deve poter ricostruire la struttura tridimensionale della scena è gravemente penalizzato dalla perdita di informazione sulla pavimentazione.

Confronto con l'immagine campione

I miglioramenti rispetto all'immagine campione sono relativi alla scomparsa dell'occlusione sulla sinistra del vaso in primo piano e alla diminuzione del rumore sullo sfondo. Il frame 5x5 conserva il rumore nell'angolo in basso a sinistra.

Tempo di elaborazione T

Il frame 4x4 impiegando 2,19 secondi risulta essere il più veloce, il più lento è 8x8 con 4,45 secondi. Il 4x6 è ancora una volta più rapido del 6x4 di quasi 80 centesimi di secondo.

Errore R

L'errore R non deve essere preso molto in considerazione, poiché si è esagerato nel settaggio dei valori di picco. I risultati ottenuti con queste distanze esagerate sono tutti peggiori di quelli ottenuti con SSD puro e SSD con soglia, ma lo scopo dell'esperimento era quello di mostrare come un errato settaggio della soglia porti benefici molto minori rispetto agli svantaggi (in termini di informazione persa) che ne conseguono. In ogni caso, la migliore mappa è l'8x8, anche se il fattore R che gli compete è molto distante dai migliori risultati dell'SSD puro.

Risultati per Tipologia III

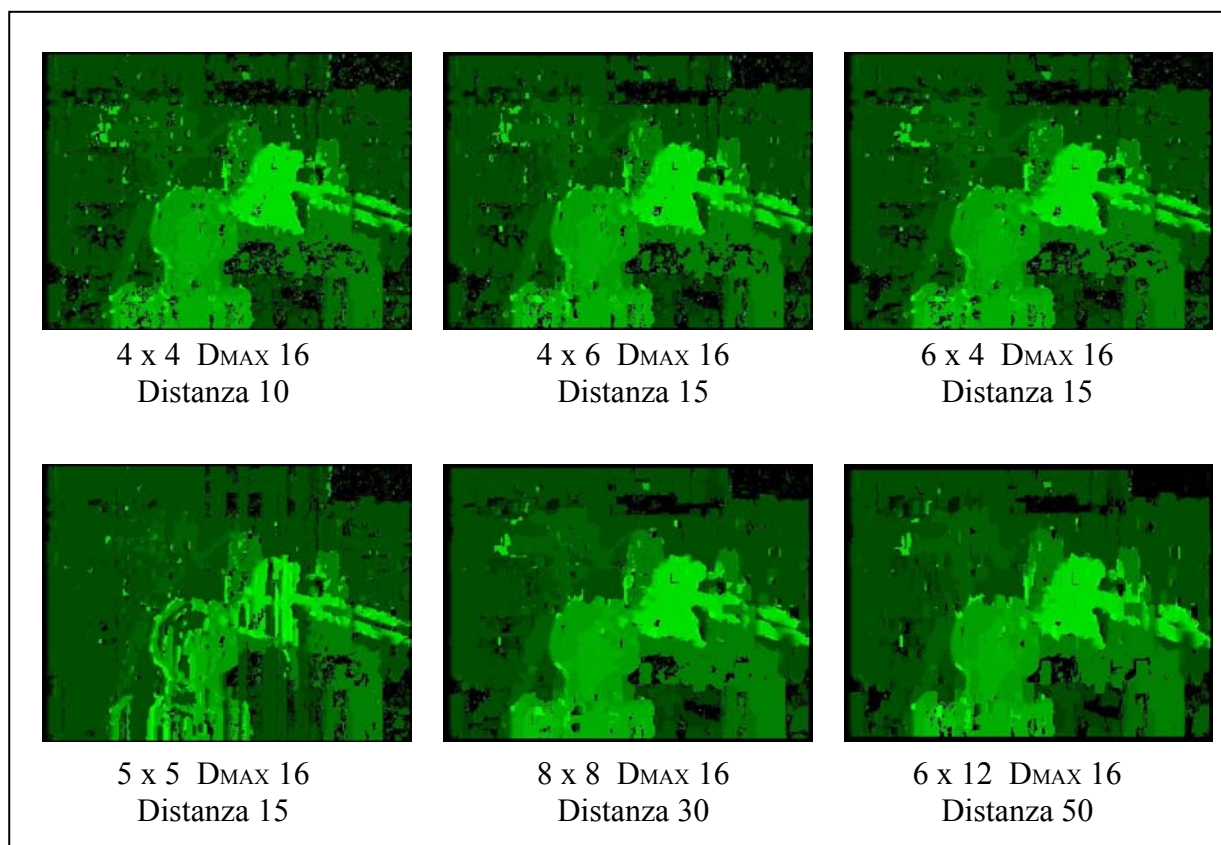


Tabella IV.17 – Risultati per tipologia III

SSD Picchi Tipo III

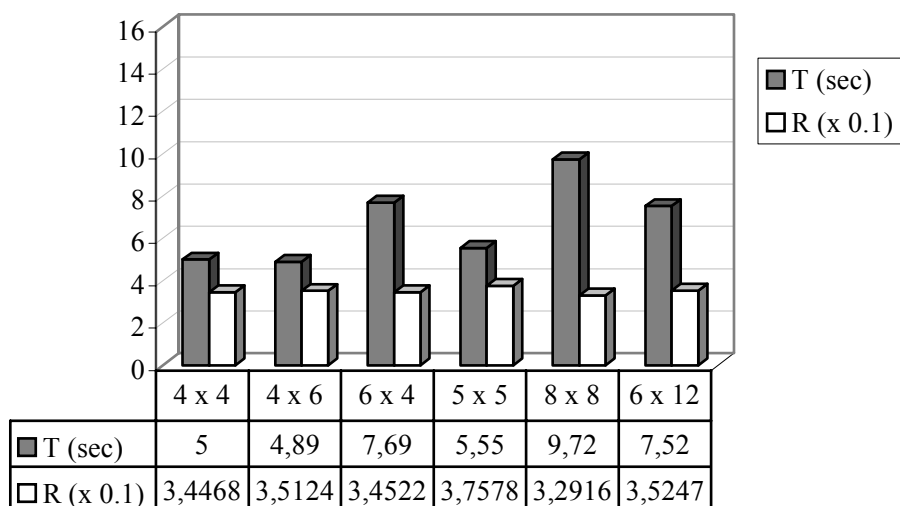


Tabella IV.18 – T ed R per tipologia III

Valutazione per Tipologia III

Parametri

I valori scelti come parametri per le sei distanze di picco sono molto contenuti perché l'immagine presenta un equilibrio molto delicato e, se si eccede nel settare le distanze, si ottiene una perdita di informazione molto maggiore dei benefici che si ricavano dalla diminuzione del rumore.

Analisi visiva

L'analisi visiva mette in evidenza la presenza di una zona di occlusione nei pressi dell'obiettivo della telecamera che si trova dietro il busto. Tale zona tende a diminuire se si alza il valore per la distanza di picco fino a scomparire del tutto se si adopera una distanza dell'ordine di 10^4 , come si vede dall'esempio IV.4. Dall'esempio si nota anche che la perdita di informazione è abbondante e tende a pregiudicare la leggibilità della mappa. In ogni caso, se la distanza di picco viene mantenuta bassa (e si applica l'interpolazione) si consegue un buon miglioramento della pulizia della mappa.



4x4 distanza 100



4x4 distanza 1000

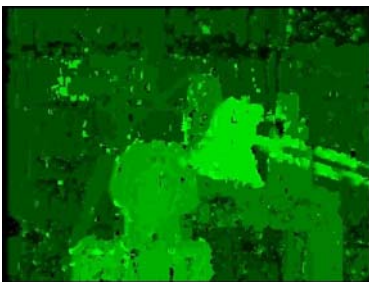


4x4 distanza 10000

Esempio IV.4 – Mappe ottenute con diversi valori di distanza di picco

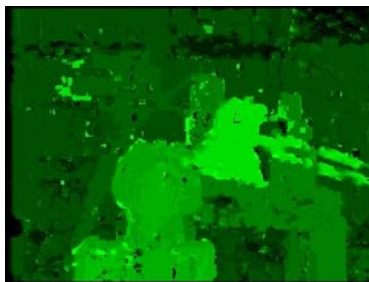
Il metodo SSD con distanza di picco interpolato

Grazie alla presenza della distanza di picco, si riesce a diminuire il rumore che affligge la mappa di disparità in varie zone. Scompaiono le macchie di colore in alto a destra, al centro del tavolino, nell'angolo in basso a sinistra, nell'angolo in basso a destra e alla sinistra delle gambe del tavolino. Queste macchie vengono sostituite, come accaduto per tutte le immagini trattate con distanza di picco (cioè per Tipologia I e Tipologia II presentate in questo paragrafo), da buchi neri. Ciò accade perché non si è voluto sfruttare la proprietà di continuità a tratti della disparità, ovvero non si è voluto adoperare il metodo di interpolazione tramite analisi dell'intorno di un pixel [Capitolo III, paragrafo 4] in modo che risultassero più chiare le zone dove va a lavorare la distanza di picco. Se si applica tale proprietà, i buchi vengono riempiti grazie all'interpolazione e le mappe risultanti sono molto buone, come si capisce dai valori di R riportati nell'esempio sottostante.



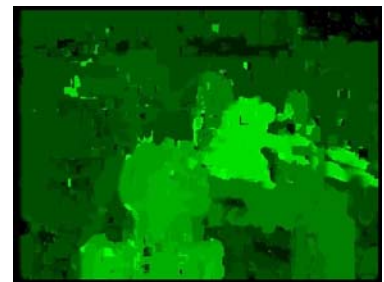
4x4 distanza 10

$T = 5,00$ s $R(x 0.1) = 2,7855$



6x4 distanza 15

$T = 6,76$ s $R(x 0.1) = 2,7985$



8x8 distanza 30

$T = 9,11$ s $R(x 0.1) = 2,9031$

Esempio IV.5 – Il metodo SSD con distanza di picco interpolato

IV.7 Il metodo SSD con soglia interna + distanza di picco

Risultati per Tipologia I

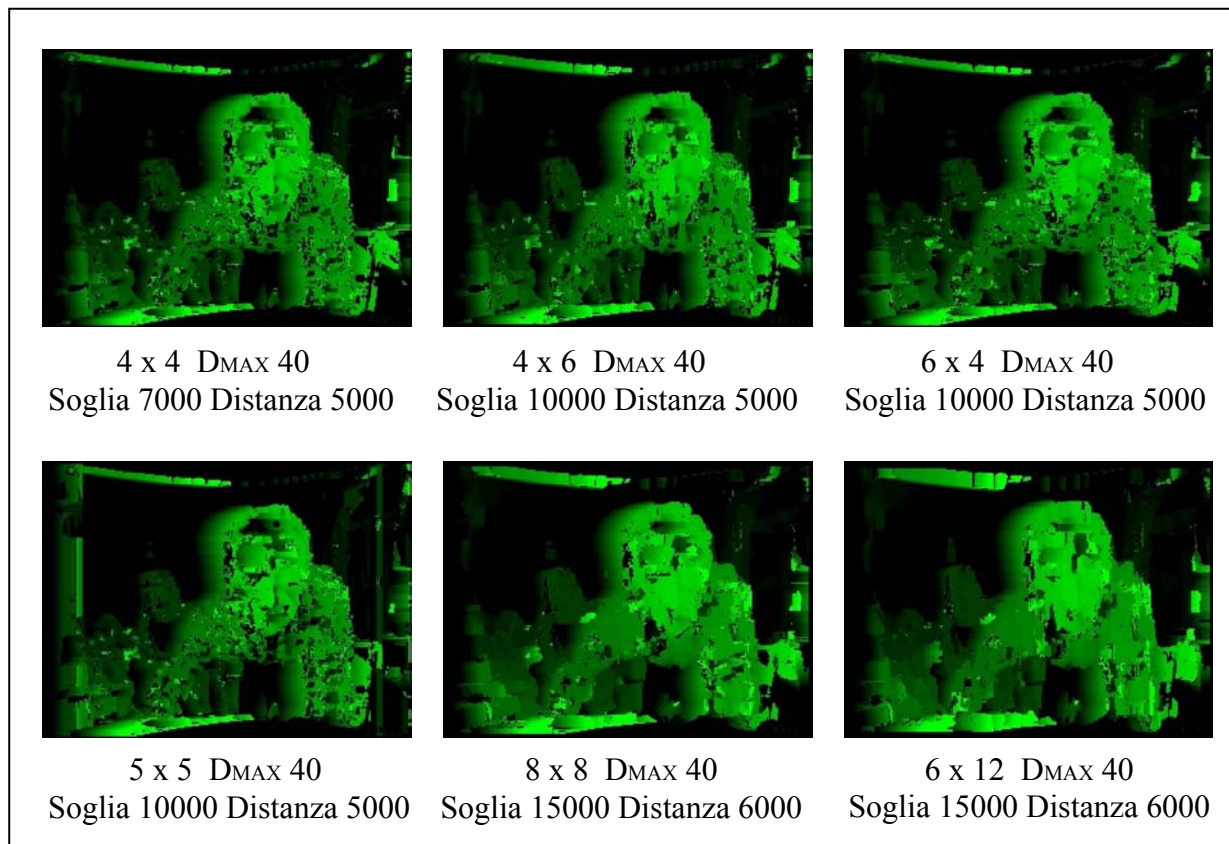


Tabella IV.19 – Risultati per tipologia I

SSD Picchi + Soglia Tipo I

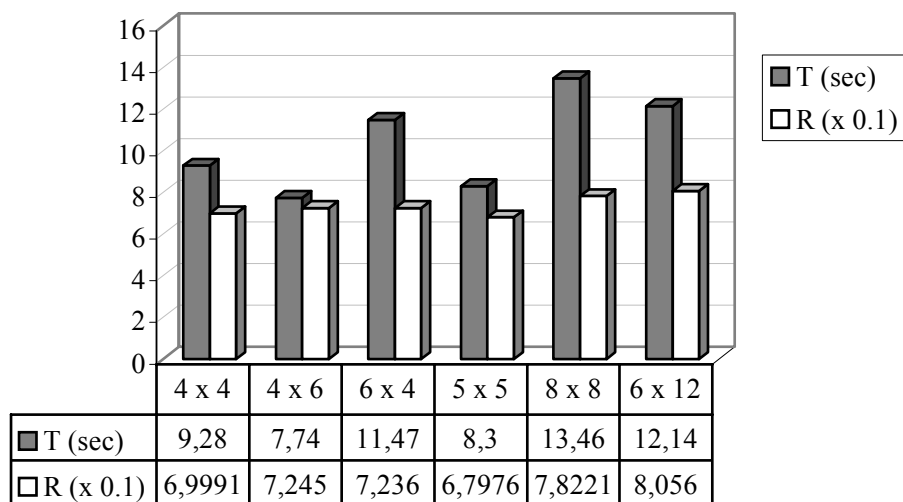


Tabella IV.20 – T ed R per tipologia I

Valutazione per Tipologia I

La combinazione di soglia interna e soglia esterna (distanza di picco) permette di aumentare la qualità della mappa di disparità per Tipologia I poiché è possibile, in virtù della soglia interna, aumentare considerevolmente la distanza di picco senza che ci sia grossa perdita di informazione. Si può pensare che la soglia interna lavori come un primo filtro che viene applicato ad ogni valore di correlazione, mentre la soglia esterna come secondo ed ultimo test di accettazione che viene applicato ai due massimi picchi una volta che tutti i valori di correlazione siano stati analizzati. Solo se il valore di correlazione ottimo è superiore alla soglia interna, viene avviato il confronto con la distanza di picco; in altre parole solo ad una parte (selezionabile tramite la soglia interna) dei pixels dell'immagine viene applicato il secondo filtro.

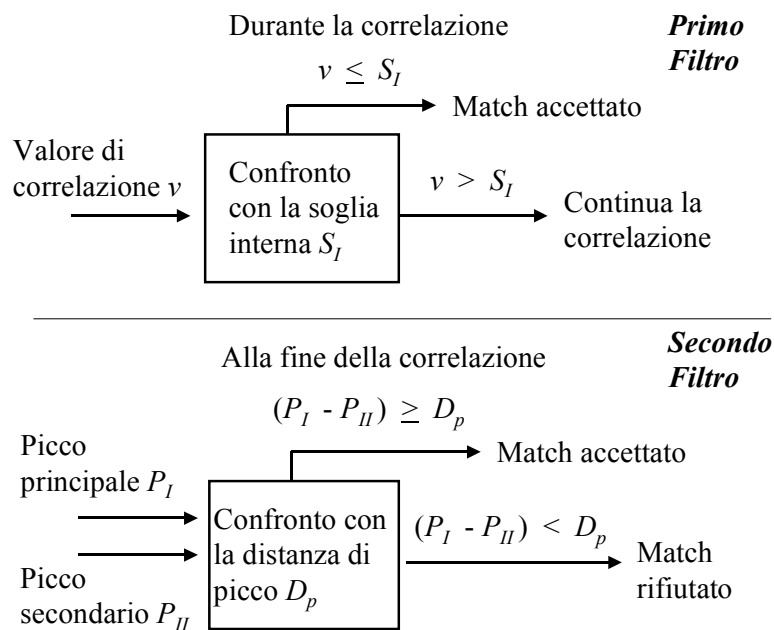


Figura IV.7 – Combinazione di soglia interna ed esterna

Analisi visiva

Rispetto all'SSD con soglia viene ridotto notevolmente il rumore senza che vi sia perdita di informazione, rispetto all'SSD con distanza di picco c'è un buon miglioramento nella zona di occlusione tra manica e monitor ed in più si ha minor perdita di informazione sui capelli e sul viso. Inoltre il rumore nell'angolo in alto a sinistra è quasi completamente scomparso

Tempo di elaborazione T

I tempi di elaborazione sono in linea con quelli fatti registrare per Tipologia I nei casi precedenti. Il miglior frame si dimostra 4x6 con 7,74 secondi, mentre 8x8 è il più lento con 13,46 secondi. Il frame 6x12 fa registrare una performance non eccellente probabilmente a causa dei parametri non settati al meglio.

Errore R

L'errore R risulta per tutti e sei i frames molto inferiore a quello fatto registrare con i metodi precedenti senza che vi sia un aumento del costo di computazione. Il migliore è 5x5 che stacca nettamente i suoi concorrenti, in particolare 6x12 che ottiene il miglioramento minore rispetto ai risultati delle tecniche già esaminate.

Risultati per Tipologia II

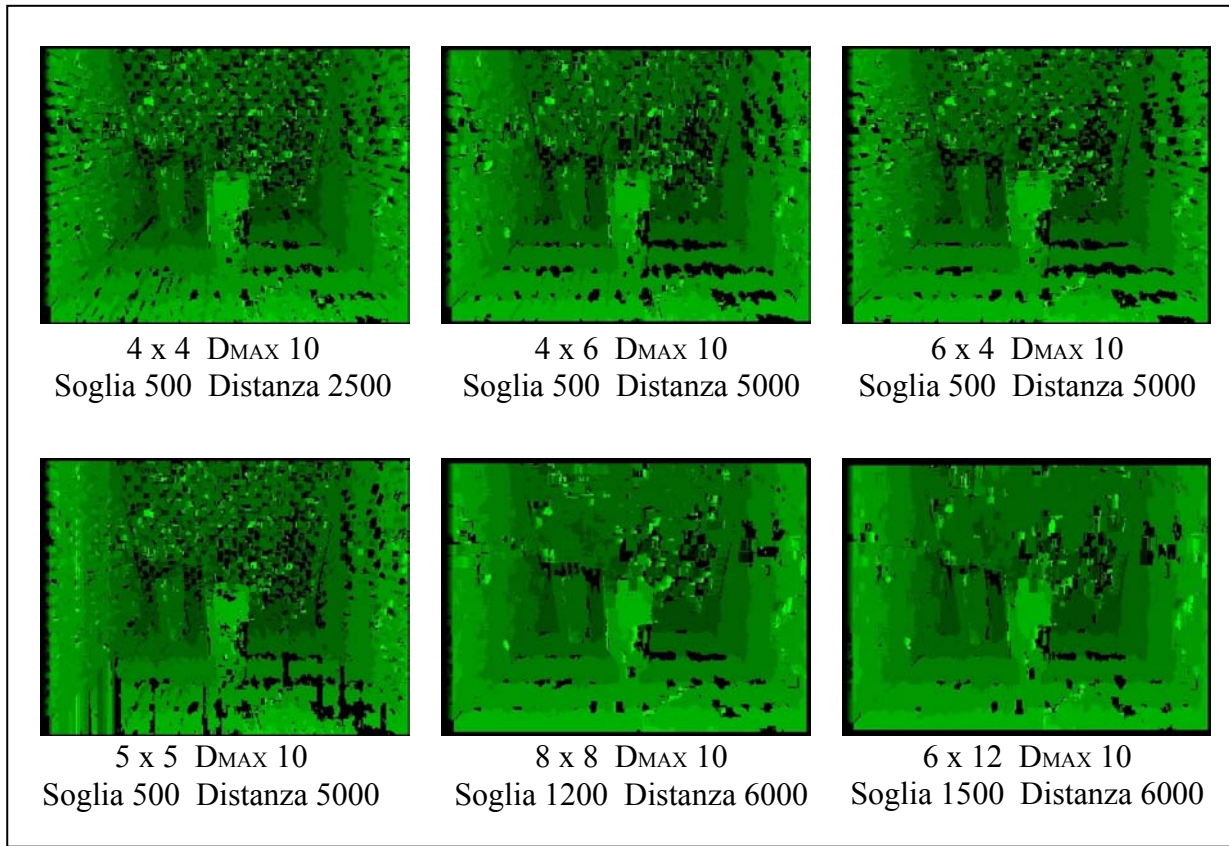


Tabella IV.21 – Risultati per tipologia II

SSD Picchi + Soglia Tipo II

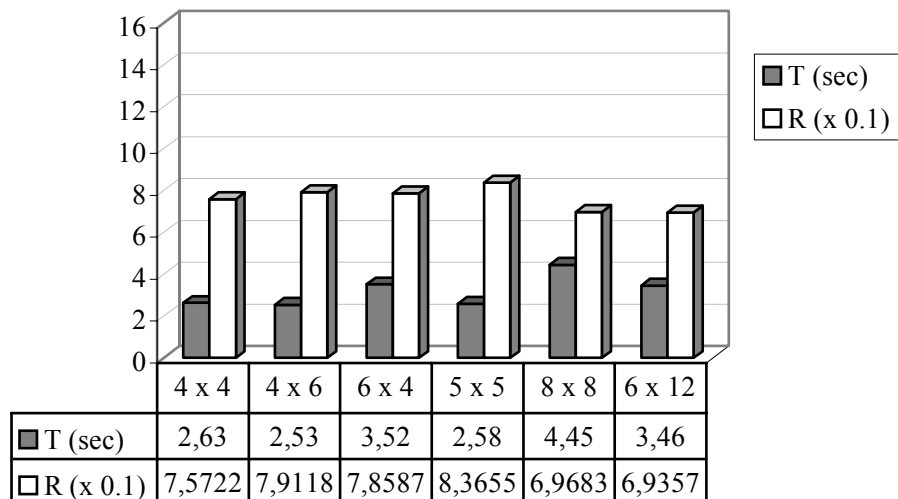


Tabella IV.22 – T ed R per tipologia II

Valutazione per Tipologia II

Parametri

I valori dei parametri scelti per la distanza e per la soglia sono volutamente elevati per evidenziare, ancor meglio, che la combinazione distanza di picco-soglia fa perdere molta meno informazione rispetto alle due tecniche prese singolarmente. Si è deciso di non applicare l'interpolazione dei risultati scartati affinché potesse risultare lampante la presenza di zone in cui agiscono i due filtri (si tratta dei buchi neri) laddove, nelle mappe di disparità ottenute con gli altri metodi visti in precedenza, erano presenti aree di rumore.

Analisi visiva

L'analisi visiva mostra una perdita di informazione sul pavimento che è più accentuata rispetto al metodo SSD con soglia nonostante i valori dei parametri di soglia siano simili. I valori per il parametro distanza sono molto più alti rispetto a quelli visti nel paragrafo precedente senza perdita di informazione così consistente come nell'SSD con distanza di picco. Un inconveniente piuttosto evidente è la minor definizione del bordo destro del vaso in primo piano. Questo vaso soffre di un grande buco laterale che era certamente più contenuto nei casi esaminati in precedenza. La parete destra risente di notevoli buchi in tutte le sei mappe.

Confronto con l'immagine campione

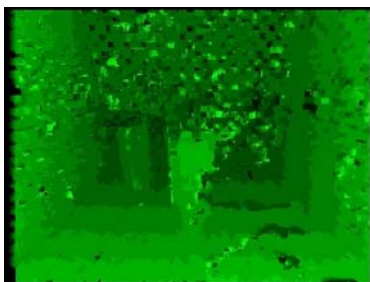
Il confronto con l'immagine campione permette di apprezzare al meglio i progressi che si ottengono riguardo al rumore di fondo sfruttando la combinazione distanza di picco-soglia. In tutte e sei le immagini, le macchie sulla destra della parete di fondo sono praticamente scomparse ed anche nella parte superiore di tale parete si ha una diminuzione del rumore. I frames 8x8 e 6x12 rispondono meno efficacemente nell'eliminare le imperfezioni perché si è tentato di utilizzare dei parametri con valori bassi per non perdere troppa informazione sul pavimento e sulle pareti laterali. Infatti, 6x12 e 8x8 risultano, nell'ordine, le due migliori mappe rispetto al fattore R proprio in virtù della minor perdita di informazione che esse subiscono.

Tempo di elaborazione T

Il frame più rapido è 4x6, il quale impiega 2,53 secondi, seguito a pochissima distanza da 5x5 (con 2,58 secondi) e 4x4 (con 2,63 secondi). Il peggiore è 8x8 con 4,45 secondi. Notevole la prestazione di 6x12 che riesce a battere, anche se di poco, il frame 6x4 nonostante analizzi un'area esattamente 3 volte più grande.

Errore R

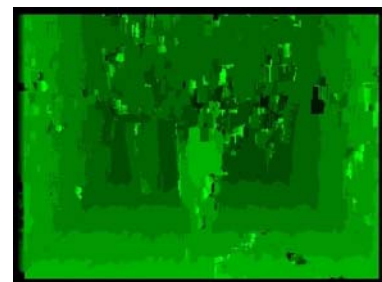
L'errore R è piuttosto elevato in tutte le sei mappe perché non si è usata interpolazione; usandola e mantenendo identici i valori dei parametri l'errore diminuisce immediatamente (esempio IV.6).



6x4 soglia 500 distanza 5000
 $T = 3,46$ s $R(x 0.1) = 6,1444$



8x8 soglia 1200 distanza 6000
 $T = 4,51$ s $R(x 0.1) = 5,892$



6x12 soglia 1500 distanza 6000
 $T = 3,46$ s $R(x 0.1) = 6,078$

Esempio IV.6 – Miglioramenti con l'interpolazione

Risultati per Tipologia III

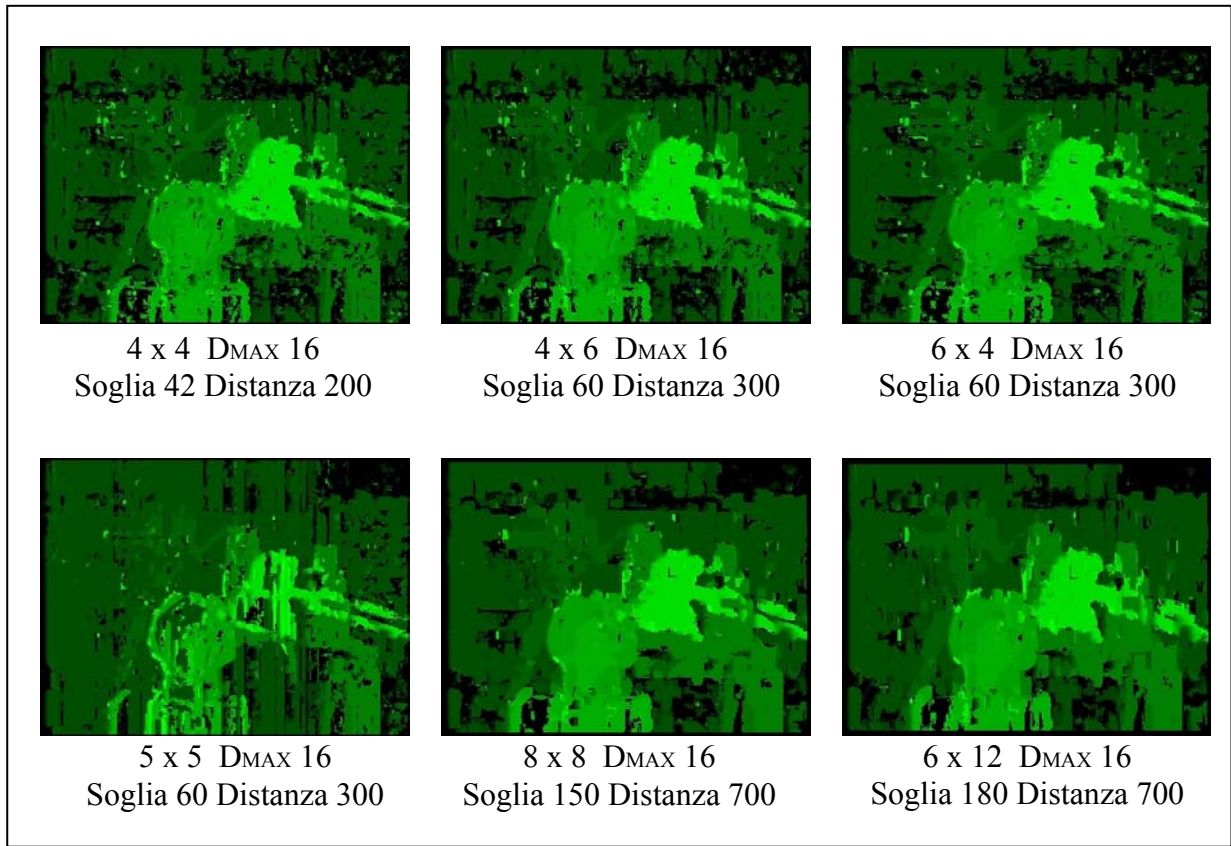


Tabella IV.23 – Risultati per tipologia III

SSD Picchi + Soglia Tipo III

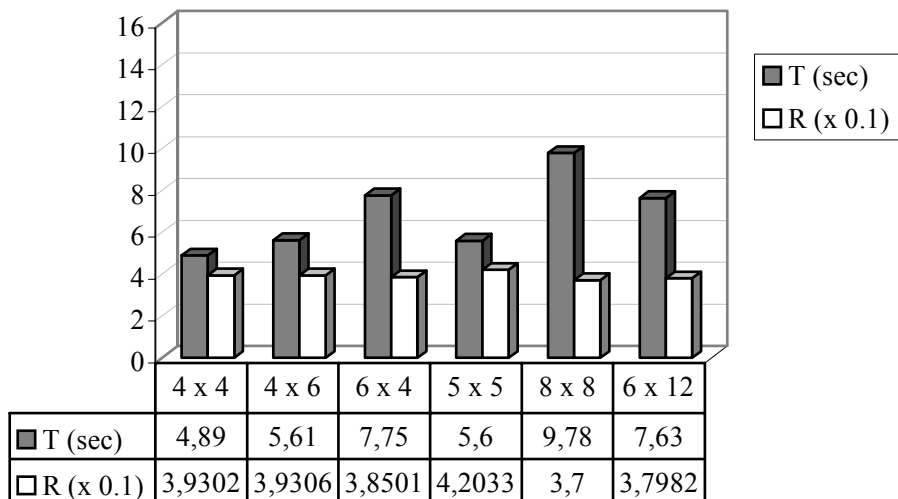
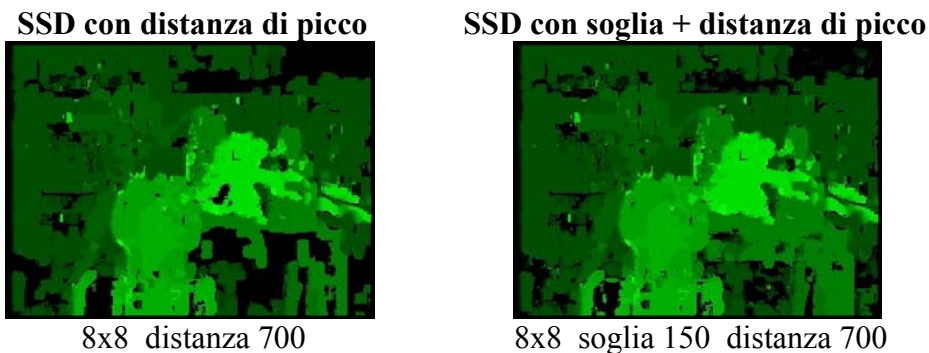


Tabella IV.24 – T ed R per tipologia III

Valutazione per Tipologia III

Parametri

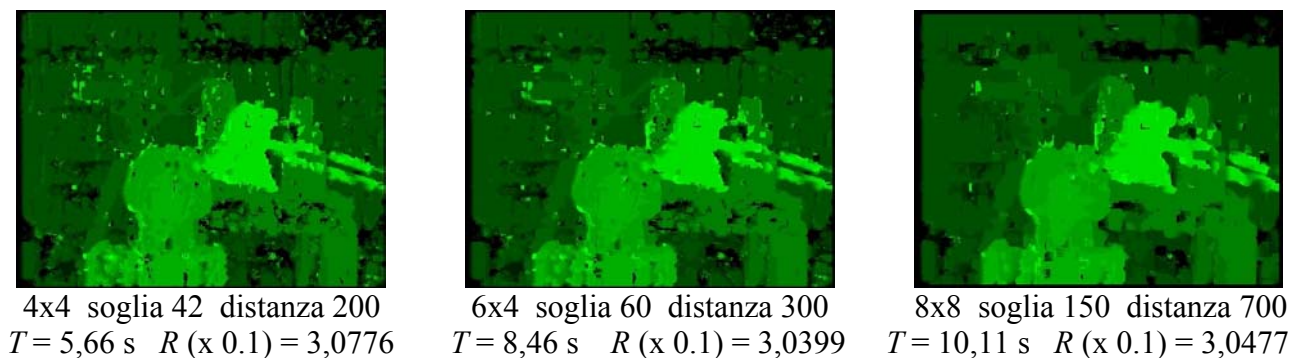
Così come era avvenuto per Tipologia I e Tipologia II in questo stesso paragrafo, è possibile settare dei valori di soglia e dei valori di distanza di picco che sono molto più elevati di quelli finora adoperati. Basti pensare che si era usato, in precedenza, con il metodo SSD con distanza di picco un valore parametrico massimo di 50, mentre ora ci si può spingere fino a 700. Questo permette di intervenire con convinzione sulla zona di occlusione nei pressi dell'obiettivo della telecamera, riducendo sensibilmente il rumore con una perdita di informazione non eccessiva. Nell'esempio IV.7 viene mostrata la differenza tra SSD con distanza di picco e l'SSD con soglia + distanza di picco usando la stessa distanza. Si osservi il tavolino e la lampada: la discrepanza tra le due mappe è netta.



Esempio IV.7 – Confronto tra due metodi diversi a parità di parametri

Uso dell'interpolazione

Le mappe ottenute in questa sezione sono tutte senza interpolazione e, conseguentemente, offrono dei valori di R decisamente insufficienti, a causa dei moltissimi buchi presenti. Se a queste mappe applichiamo l'interpolazione avremo una decisa diminuzione dei buchi ed una conseguente forte trasformazione positiva per il fattore R . Tuttavia, l'interpolazione fatica a coprire tutti i buchi, che sono veramente tanti, e tende a reinserire le macchie attorno all'obiettivo della telecamera dietro il busto che tanto faticosamente erano state eliminate. Solo con il frame 8x8 si riesce a mantenere relativamente pulita la zona di occlusione in questione (si osservi la terza figura da sinistra del sottostante esempio IV.8). Va aggiunto che l'interpolazione comporta un, seppur minimo, aumento del costo temporale di cui va, sicuramente, tenuto conto.



Esempio IV.8 – Il metodo SSD con soglia + distanza di picco interpolato

IV.8 Il metodo SSD unito con il metodo SAD

Risultati per Tipologia I

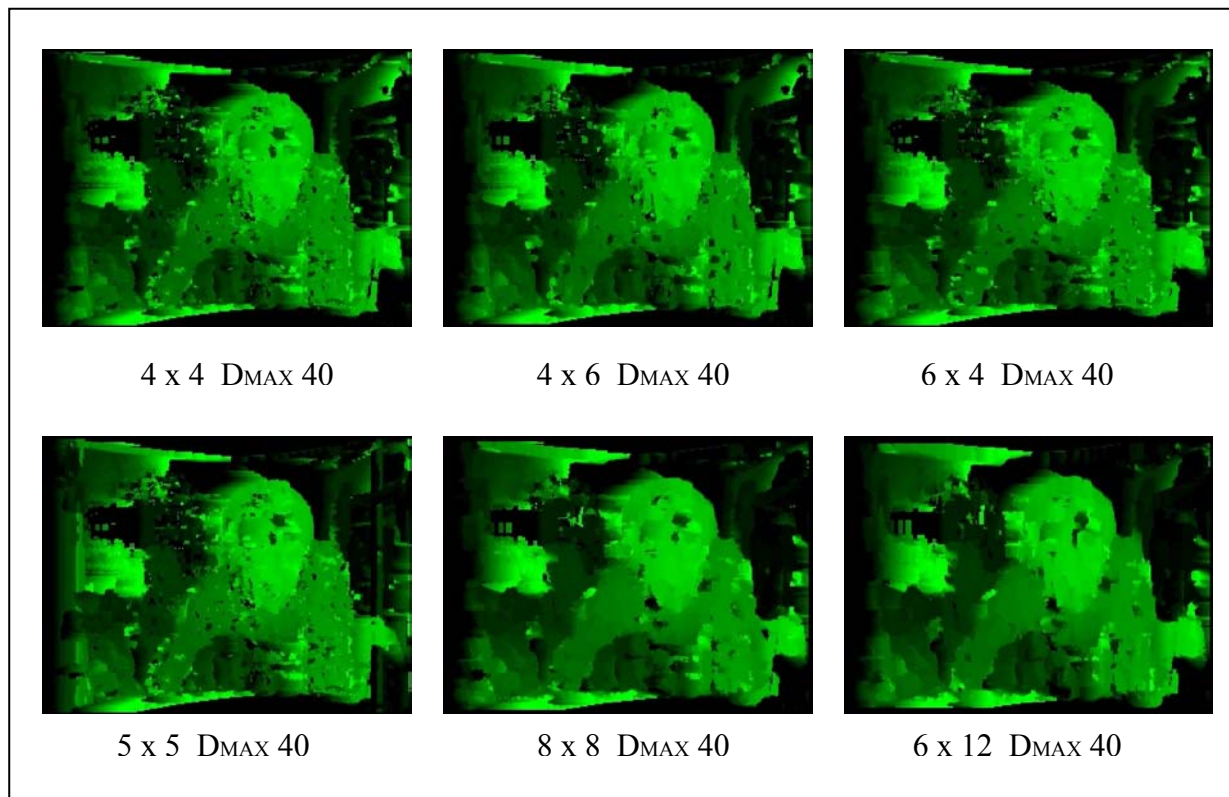


Tabella IV.25 Risultati per tipologia I

SSD + SAD Tipo I

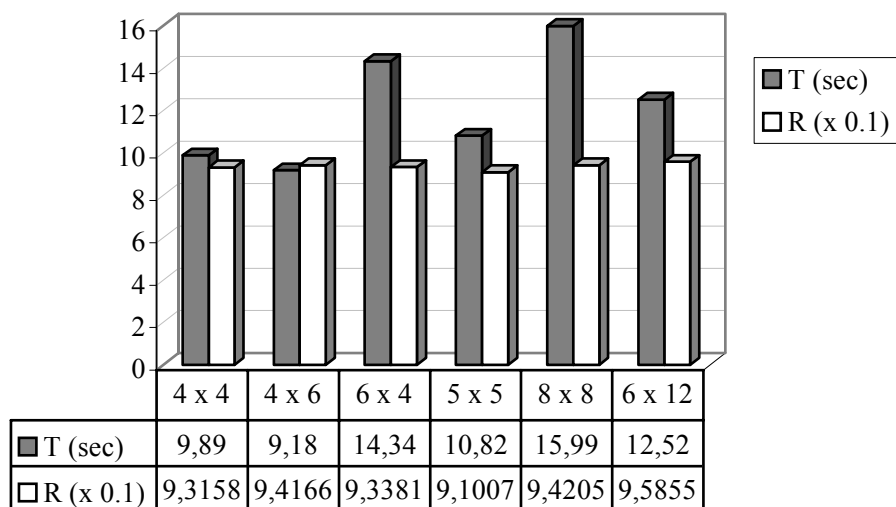


Tabella IV.26 – T ed R per tipologia I

Valutazione per Tipologia I

La combinazione di SSD e SAD

Il metodo di correlazione SAD è stato trattato nel paragrafo II.2.1 sezione A, insieme al metodo SSD, mentre l'unione di SSD e SAD è stata analizzata nel paragrafo III.5. Una semplice osservazione permette di capire perché può essere vantaggiosa la combinazione di più metodi:

Osservazione IV.1 *Date due tecniche di correlazione diverse T_1 e T_2 che si suppongono corrette, se per un pixel p si ottiene, usando tali tecniche, la medesima disparità d_M , mentre per un pixel p' si ottengono le disparità d_1 usando T_1 e d_2 usando T_2 con $d_1 \neq d_2$, allora possiamo considerare d_M più attendibile rispetto a d_1 e d_2 .*

La precedente osservazione va presa con la dovuta cautela perché nulla vieta che ambedue gli algoritmi T_1 e T_2 commettano errori che portano ad avere come output una identica disparità; tuttavia, partendo dal presupposto che le tecniche SSD e SAD sono corrette, se tutte e due raggiungono il medesimo risultato molto probabilmente tale risultato è esatto.

Confronto con l'immagine campione

Il confronto con l'immagine campione mette in luce proprio la necessità di trattare con cautela la precedente osservazione IV.1. A tal proposito, si considerino le sei immagini ottenute con la combinazione di SAD ed SSD: esse appaiono, tutte, contraddistinte da una grossa macchia a disparità elevata sulla sinistra, ad altezza corrispondente al volto della figura in secondo piano. Tale macchia indica chiaramente una serie di match errati, perché si tratta della parete di fondo, la quale dovrebbe avere disparità bassissima essendo molto distante dalle due telecamere che riprendono la scena. Sempre la stessa macchia era presente anche nelle immagini campione (ottenute con SSD puro), dimostrando che SAD sbaglia nel calcolare la disparità per i pixels che fanno parte di tale area allo stesso modo di SSD, ovvero sia SSD sia SAD calcolano le medesime disparità errate, poiché altrimenti tale zona dovrebbe avere disparità nulla. Tutto ciò porta l'algoritmo SSD + SAD a considerare attendibili tali disparità nonostante siano errate.

In figura IV.8 viene riportato uno schema di massima per il metodo SSD + SAD

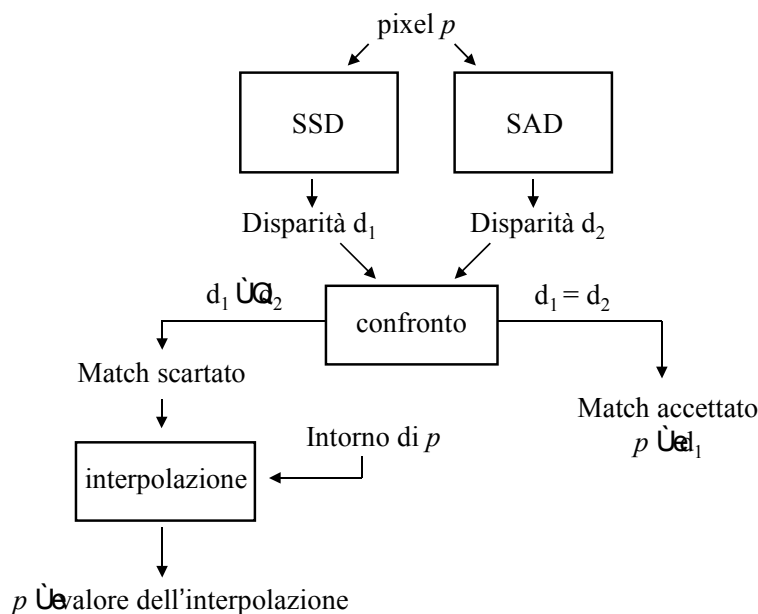


Figura IV.8 – Il metodo SSD + SAD

Risultati per Tipologia II

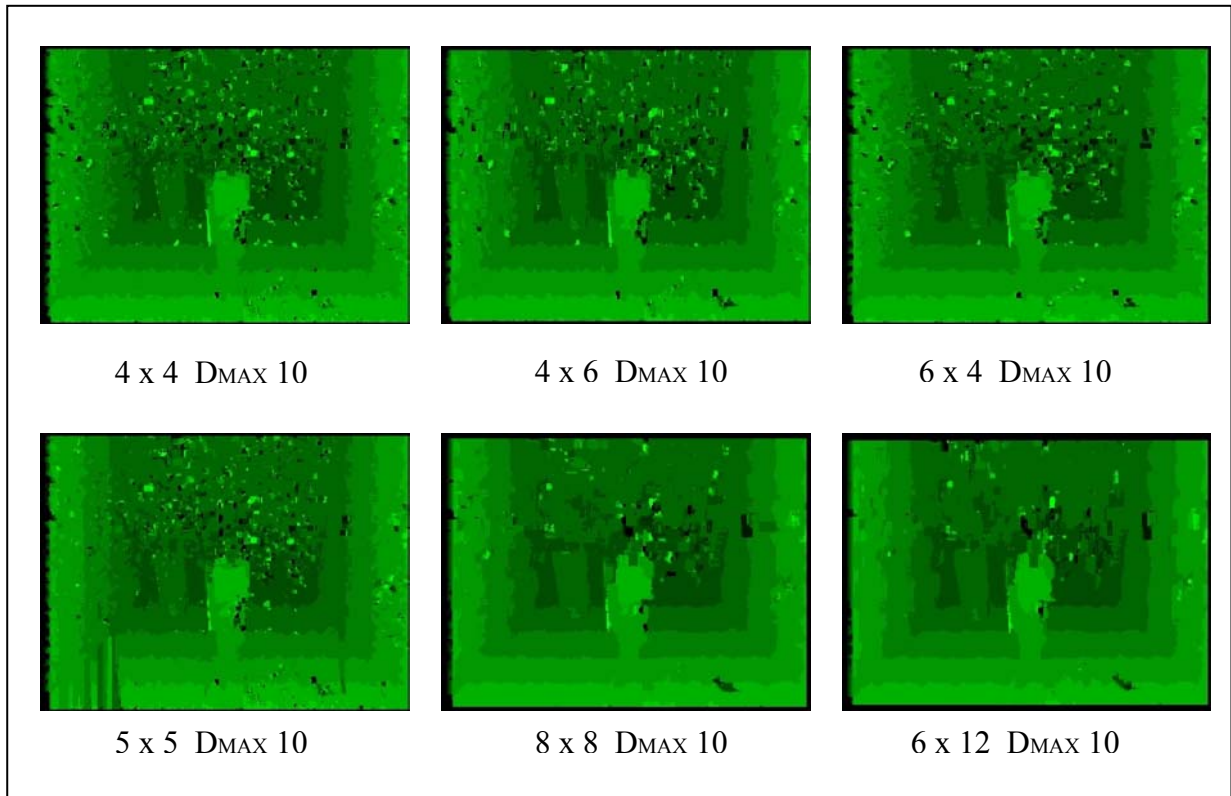


Tabella IV.27 – Risultati per tipologia II

SSD + SAD Tipo II

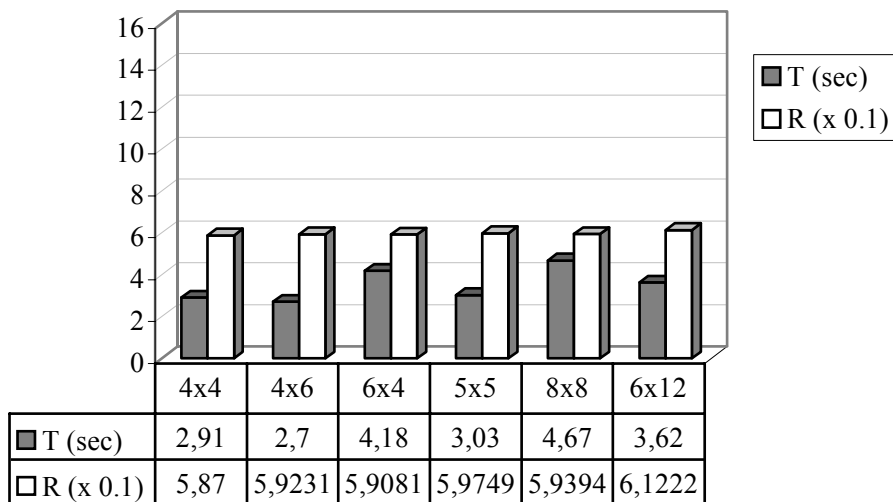


Tabella IV.28 – T ed R per tipologia II

Valutazione per Tipologia II

Analisi visiva

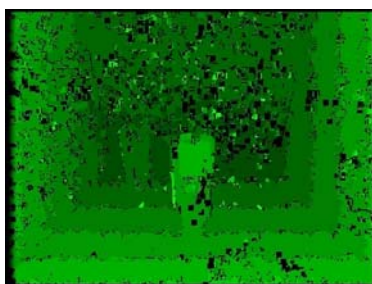
La combinazione di SSD e SAD riduce considerevolmente il rumore sullo sfondo e praticamente azzerava le macchie di colore presenti nell'angolo di pavimento in basso a destra. Le mappe risultano molto più leggibili delle precedenti e per 8x8 e 6x12 la pulizia e la qualità migliorano tantissimo. Unico inconveniente, la presenza di alcuni grossi buchi, sulla parete sinistra, sullo sfondo e sul pavimento, che l'interpolazione non riesce ad arginare.

Confronto con l'immagine campione

Se il confronto con le immagini campione viene basato sulla pulizia dell'immagine, non c'è davvero storia perché la combinazione di SSD e SAD è molto più efficace del solo SSD nell'eliminare il rumore. Se il raffronto si fa sulla base del fattore R , la differenza non è più tanto netta ed, anzi, l'SSD puro risulta di pochissimo migliore per i frames 4x4, 4x6, 6x4 e 5x5. Al contrario, l'8x8 e il 6x12 sono migliori se si utilizza l'SSD + SAD. Questi risultati si spiegano a causa dei buchi che si creano nell'SSD + SAD al posto delle macchie di colore presenti nell'SSD puro e che purtroppo l'interpolazione non può mascherare più di tanto.

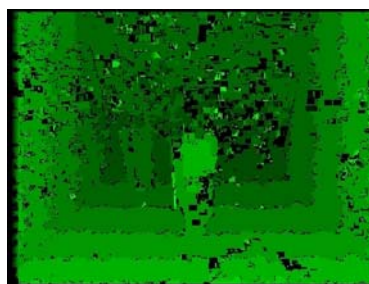
Il metodo SSD + SAD e l'interpolazione

Come è stato riportato nella precedente figura IV.8, lo schema del metodo SSD + SAD prevede l'interpolazione dei risultati, poiché altrimenti si otterrebbero immagini con innumerevoli buchi di piccole dimensioni, quasi come se la mappa fosse un quadro dipinto con la tecnica del *puntinismo*. Infatti, le differenze di valutazione tra SSD e SAD si riscontrano particolarmente sui bordi degli oggetti presenti nella scena ed, in generale, nelle zone di confine tra parti della mappa a disparità distinte. Si osservi l'esempio IV.9 dove i puntini neri si concentrano sui confini delle fasce orizzontali di verde sul pavimento, fasce che indicano zone a diversa disparità, in virtù della profondità della scena. Si notino, inoltre, i contorni del vaso in secondo piano, essi sono composti da piccoli punti neri che stanno a testimoniare la discrepanza di risultati per gli algoritmi SSD e SAD.



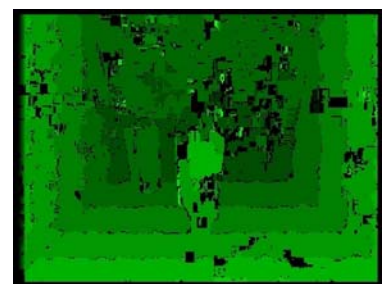
4x4

$T = 2,80 \text{ s}$ $R(x 0.1) = 7,2348$



6x4

$T = 4,23 \text{ s}$ $R(x 0.1) = 7,3369$



8x8

$T = 5,22 \text{ s}$ $R(x 0.1) = 7,3174$

Esempio IV.9 – Risultati senza interpolazione

Tempo di elaborazione T

I tempi di elaborazione sono chiaramente più elevati rispetto a quelli fatti registrare in precedenza perché la mole di calcoli da effettuare è quasi doppia. In realtà, il SAD impiega un tempo inferiore rispetto all'SSD per correlare lo stesso numero di pixels, poiché il calcolo del valore assoluto è meno dispendioso del calcolo delle radici quadrate [Capitolo II, paragrafo 2.1, sezione A]. Il miglior frame è 4x6, mentre è degno di nota il fatto che 6x12 impiega più di un secondo in meno rispetto a 8x8.

Risultati per Tipologia III

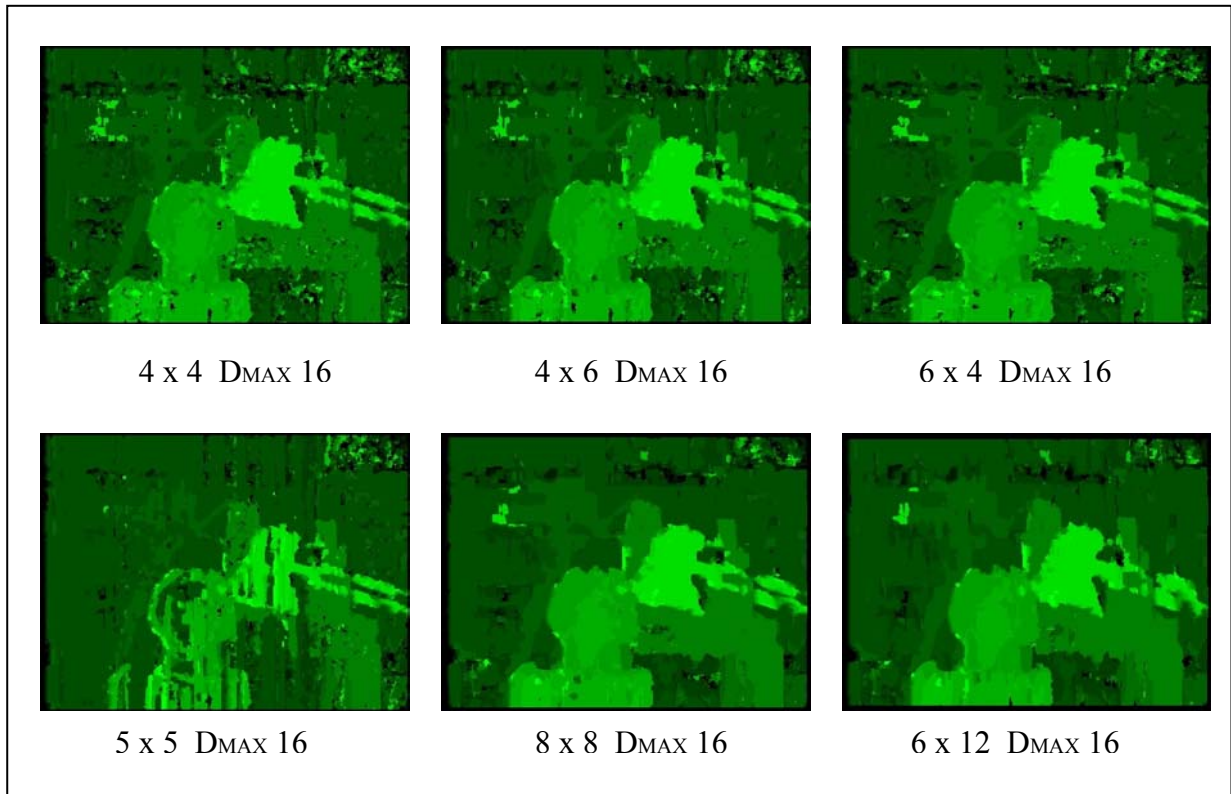


Tabella IV.29 – Risultati per tipologia III

SSD + SAD Tipo III

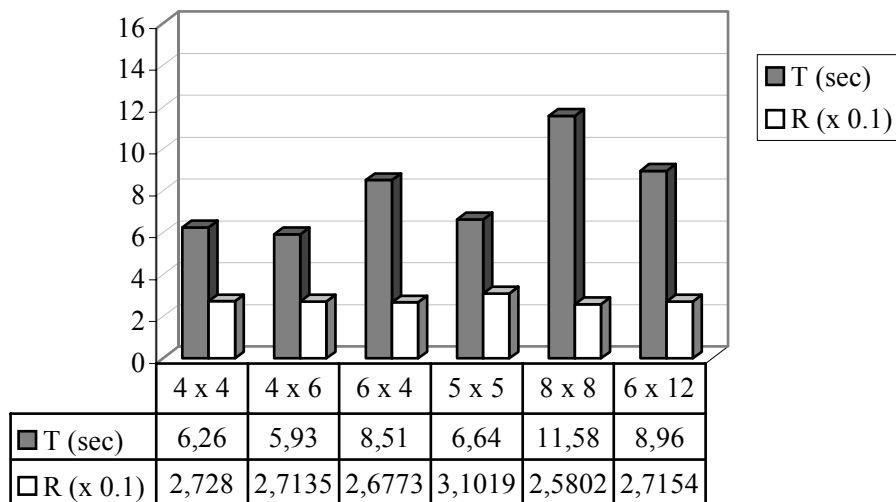
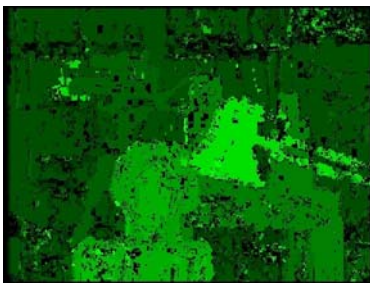


Tabella IV.30 – *T* ed *R* per tipologia III

Valutazione per Tipologia III

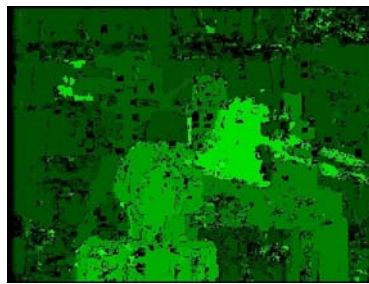
Analisi visiva

L'analisi visiva mette in mostra la riduzione significativa del rumore sull'intera immagine. La zona di occlusione attorno all'obiettivo della telecamera resta, a dimostrazione che entrambi i metodi SSD e SAD, senza l'aiuto di funzioni euristiche come la distanza di picco, non sono in grado, presi singolarmente o in coppia, di fronteggiare il problema dell'occlusione. A ulteriore conferma di ciò, si osservi il seguente esempio IV.10 dove si è applicata la combinazione SSD + SAD senza inserire l'interpolazione dei match scartati, ma considerando nulle le disparità con incongruenze di risultati. La zona di occlusione attorno all'obiettivo della telecamera è ancora presente a mostrare che sia SSD che SAD calcolano lo stesso risultato errato per quei particolari pixels: in caso contrario avremmo dovuto ottenere un buco in tale zona.



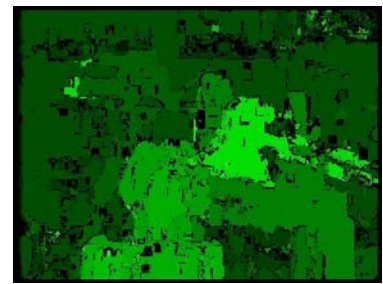
4x4

$T = 5,99 \text{ s}$ $R(x 0.1) = 3,7072$



6x4

$T = 8,07 \text{ s}$ $R(x 0.1) = 3,7395$



8x8

$T = 12,25 \text{ s}$ $R(x 0.1) = 3,8612$

Esempio IV.10 – Risultati senza interpolazione

Confronto con l'immagine campione

Rispetto alle immagini campione, viene eliminato il rumore che era localizzato sul piedistallo del busto, a destra del busto sul tavolino, a sinistra del piede della telecamera ed, infine, sulla telecamera stessa. Il valore di R per la combinazione di SSD e SAD è molto minore di quello ottenuto per l'SSD senza che ci sia un esagerato aumento del tempo di esecuzione.

Tempo di elaborazione T

I tempi di elaborazione sono maggiori di quelli incontrati in precedenza per Tipologia II perché i calcoli da fare sono raddoppiati. Il miglior frame è 4x6 con un buonissimo 5,93, mentre come al solito il peggiore è 8x8 con 11,58 secondi, più di un secondo e mezzo più lento dell'SSD puro, ma al tempo stesso molto migliore sotto l'aspetto dell'errore R .

Errore R

Per quel che riguarda l'errore R , si può affermare che i risultati di SSD e SAD combinati e con interpolazione sono i migliori fin'ora ottenuti, il discorso vale per tutti e sei i frames. Il migliore è 8x8, mentre il peggiore è, come si capisce già dal primo sguardo, 5x5.

IV.9 Risultati per il metodo SSD con D_{MAX} variabile

Risultati per Tipologia I

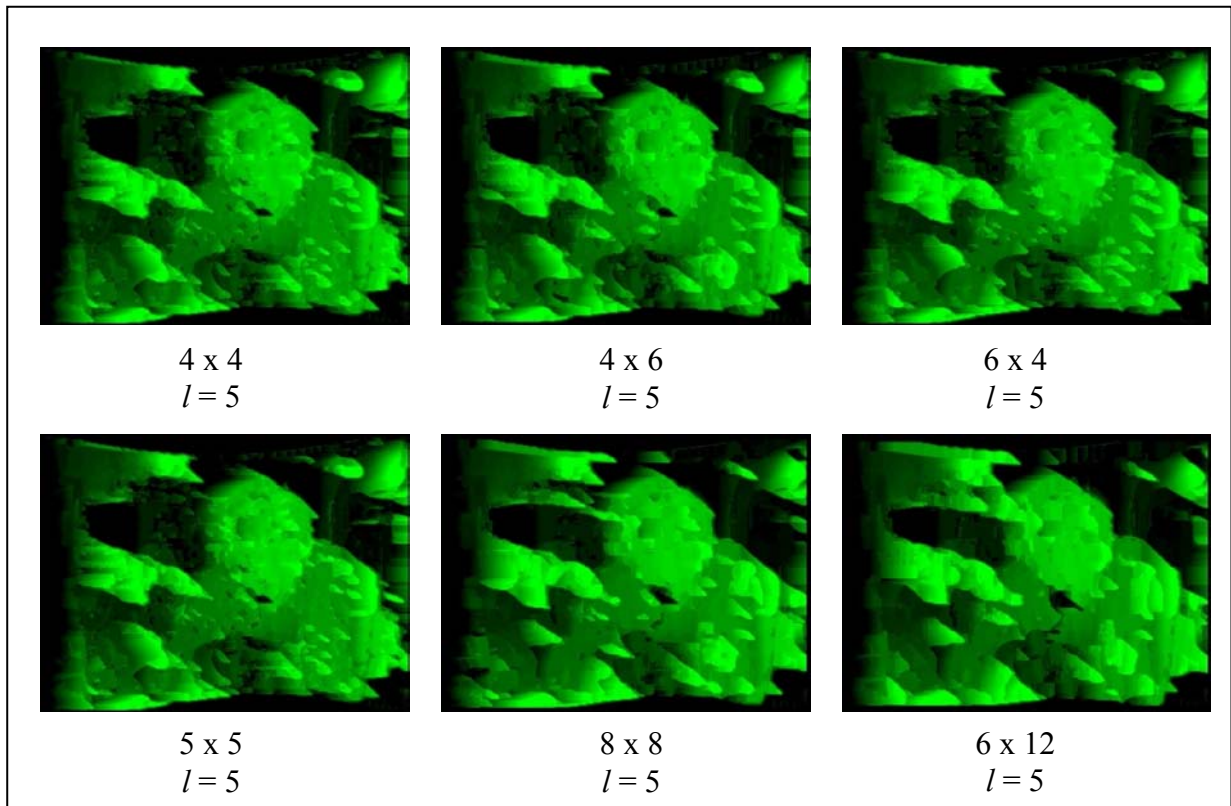


Tabella IV.29 – Risultati per tipologia I

D_{MAX} Variabile Tipo I

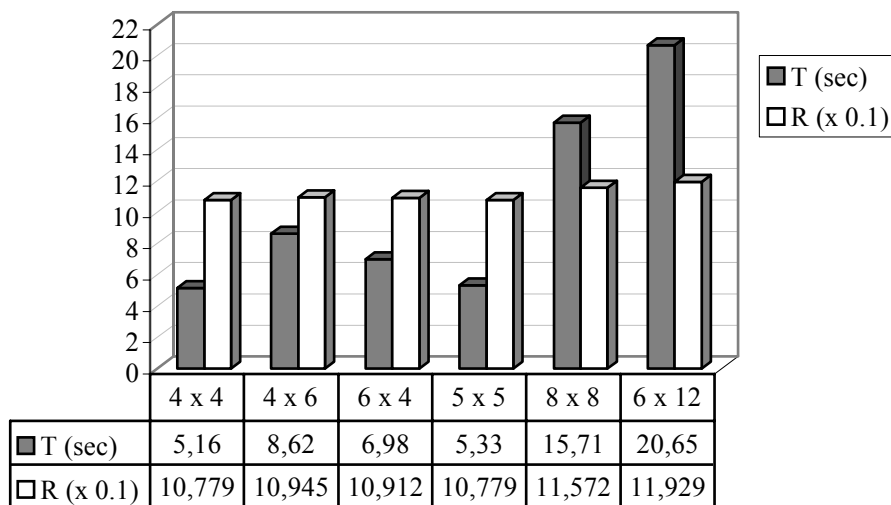


Tabella IV.30 – T ed R per tipologia I

Valutazione per Tipologia I

L'algoritmo con D_{MAX} variabile

Il termine D_{MAX} variabile sta ad indicare che non vengono valutati i valori di correlazione per tutti i frames da 0 a D_{MAX}, ma solo una parte di essi viene analizzata, più precisamente al massimo 11 frames. Come accennato nel paragrafo 4 del terzo capitolo, se si vuole calcolare la disparità per un pixel p , si può calcolare il valore d per l'intorno di p ed assumere d come punto di partenza. A questo punto, vengono esaminati solo i frames a partire da $d - l$ fino a $d + l$, con l intero fissato a 5. I frames sono 11 perché va considerato anche il frame centrale di disparità d : $5 + 1 + 5 = 11$. Bisogna precisare che la ricerca non va in nessun caso oltre D_{MAX} né scende sotto lo zero, per cui 11 è il massimo numero di frames che si analizzano; ad esempio, se D_{MAX} = 40 e $d = 36$ si confrontano 10 frames, da 31 a 40.

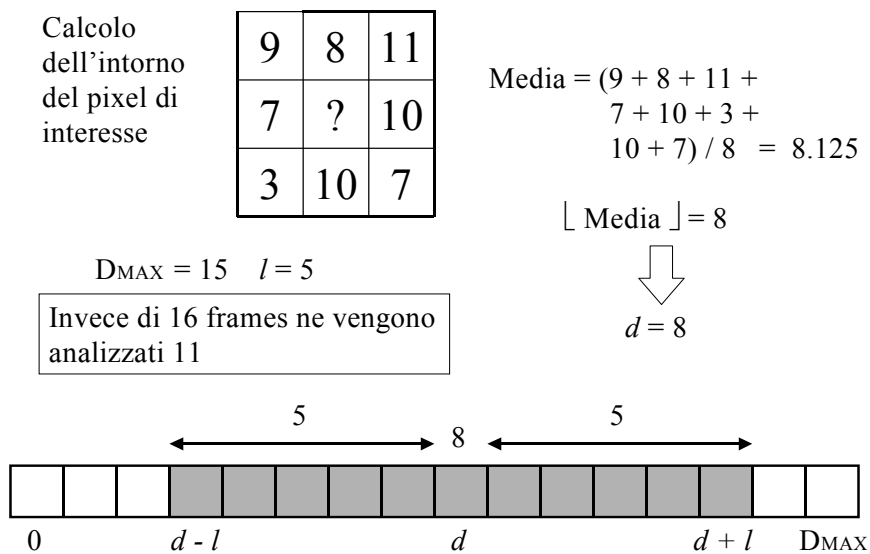
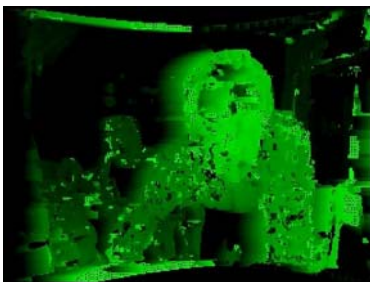


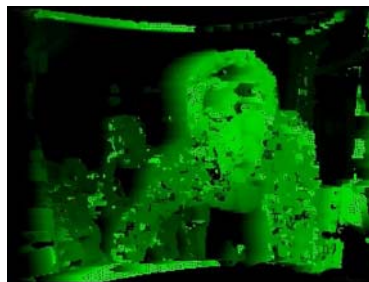
Figura IV.9 – Calcolo della disparità con D_{MAX} variabile

D_{MAX} variabile con soglia interna

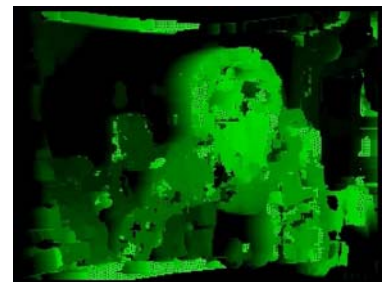
I risultati conseguiti non sono soddisfacenti poiché esiste una grande difficoltà nell'individuare la giusta corrispondenza per zone della scena in cui la profondità varia rapidamente, perché la media di disparità di una regione si modifica troppo lentamente rispetto a grosse variazioni di profondità della scena. Si può provare a migliorare il tutto con l'inserimento di una soglia interna.



4x4



6x4



8x8

$T = 16,75 \text{ s} \quad R(x \ 0.1) = 7,7014 \quad T = 20,32 \text{ s} \quad R(x \ 0.1) = 7,7504 \quad T = 59,97 \text{ s} \quad R(x \ 0.1) = 8,0744$

Esempio IV.11 – Risultati con soglia interna

Risultati per Tipologia II

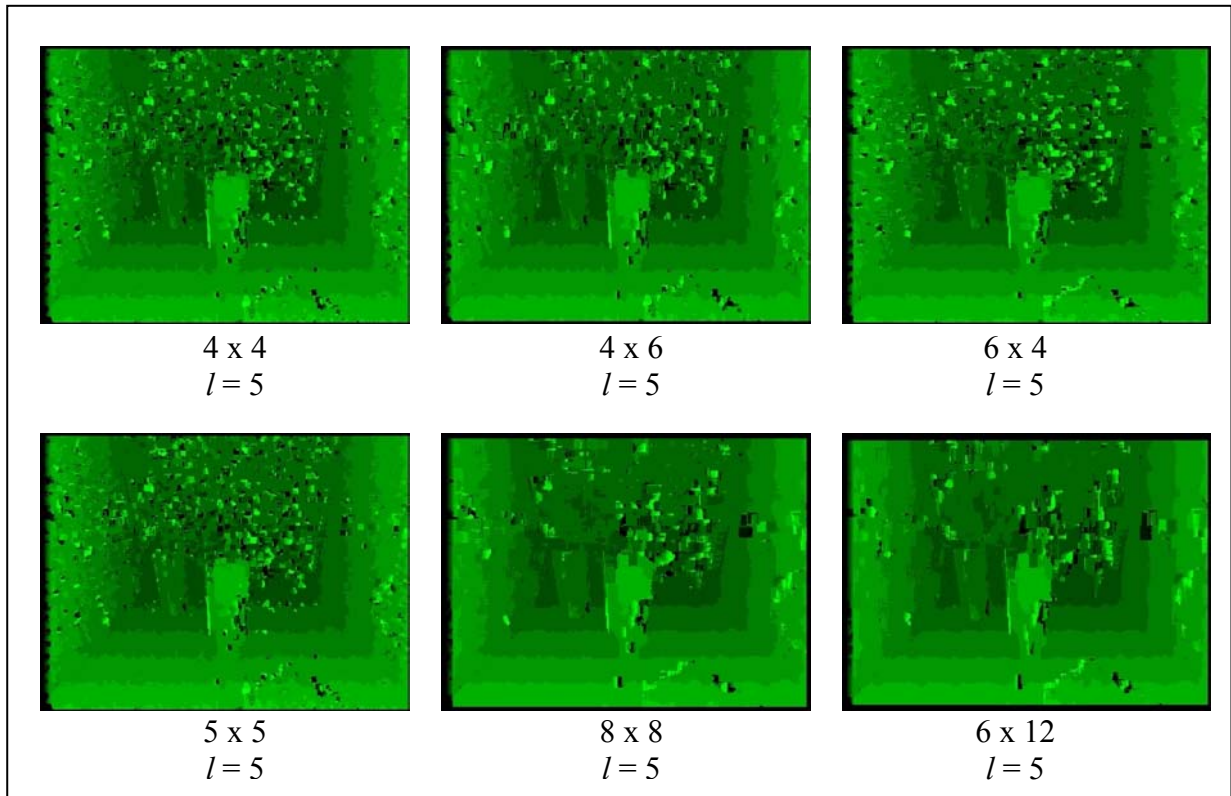


Tabella IV.29 – Risultati per tipologia II

D_{MAX} Variabile Tipo II

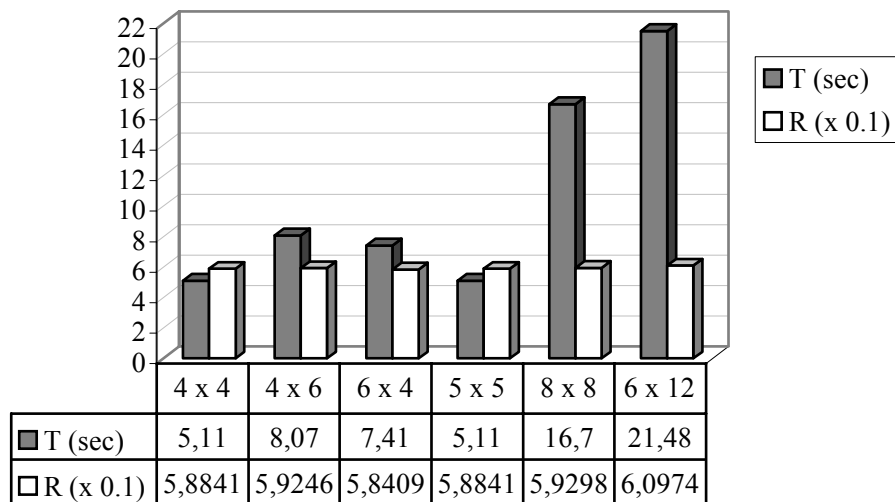


Tabella IV.30 – T ed R per tipologia II

Valutazione per Tipologia II

Parametri

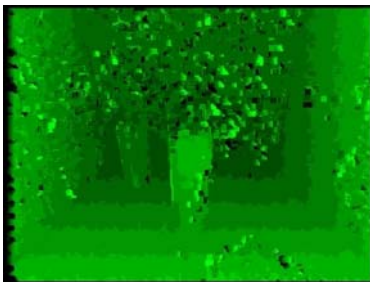
I risultati che vengono presentati sono stati ottenuti con $l = 5$ e senza l'ausilio di alcuna soglia né, tanto meno, di alcuna funzione euristica. Se si osservano i valori per R si trova che per 6x4, 5x5, 8x8, e 6x12 si ottengono le migliori prestazioni rispetto a tutti i precedenti metodi. Questo porta ad una osservazione di notevole importanza.

Osservazione IV.2 *Se la coppia di immagini da correlare non presenta forti discontinuità di disparità, l'algoritmo con D_{MAX} variabile produce risultati migliori rispetto agli altri metodi, perché consente di trattare con efficacia le zone di occlusione ed in più elimina il rumore dovuto a texture ripetitive.*

Tutto questo è possibile grazie alla possibilità di controllare l'escursione di disparità per un pixel, il che localizza e specializza ancor di più la ricerca del miglior match.

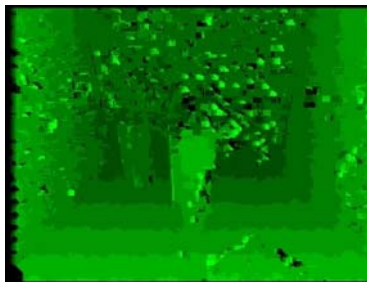
Analisi visiva

L'analisi visiva mostra come il frame 5x5, grazie a D_{MAX} variabile, non soffra più del pesante rumore nell'angolo sinistro, e come tutele immagini ottengano un miglioramento riguardo il rumore sul pavimento. Lo sfondo risulta più uniforme, perché, sebbene il rumore sia sempre presente, i valori errati sono smussati da l , che di fatto limita l'escursione di disparità ad un massimo di 5. Se si riduce l a 3 si ottengono i risultati esposti qui sotto.



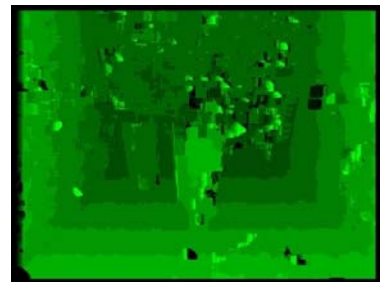
4x4 $l = 3$

$T = 3,68$ s $R(x 0.1) = 5,8177$



6x4 $l = 3$

$T = 5,71$ s $R(x 0.1) = 5,8521$



8x8 $l = 3$

$T = 11,86$ s $R(x 0.1) = 5,9574$

Esempio IV.12 – Risultati con $l = 3$

Per 4x4 si ottiene un miglioramento rispetto a T (circa 1,5 secondi in meno rispetto a $l = 5$) e rispetto a R (58,177 contro 58,841). Per 6x4 e 8x8 il miglioramento (per 8x8 quasi 5 secondi) riguarda solo il tempo di elaborazione a fronte di un piccolo incremento dell'errore R .

Tempo di elaborazione T

I tempi di elaborazione sono molto più elevati di quelli fatti registrare da tutti i precedenti metodi, perché non può essere sfruttata la proprietà di ripetitività propria dell'algoritmo di correlazione se esso è completo, cioè se si ispezionano tutti i frames da 0 a D_{MAX} . Di questo si tratterà con più attenzione nella valutazione di Tipologia III.

Errore R

Il miglior frame è 6x4 seguito da 4x4 e 5x5. E' interessante notare che 4x4 e 5x5 ottengono la medesima mappa di disparità proprio perché il valore di D_{MAX} variabile tende ad appiattire i valori di correlazione e c'è troppa poca differenza tra le aree di 4x4 e 5x5 per produrre risultati diversi.

Risultati per Tipologia III

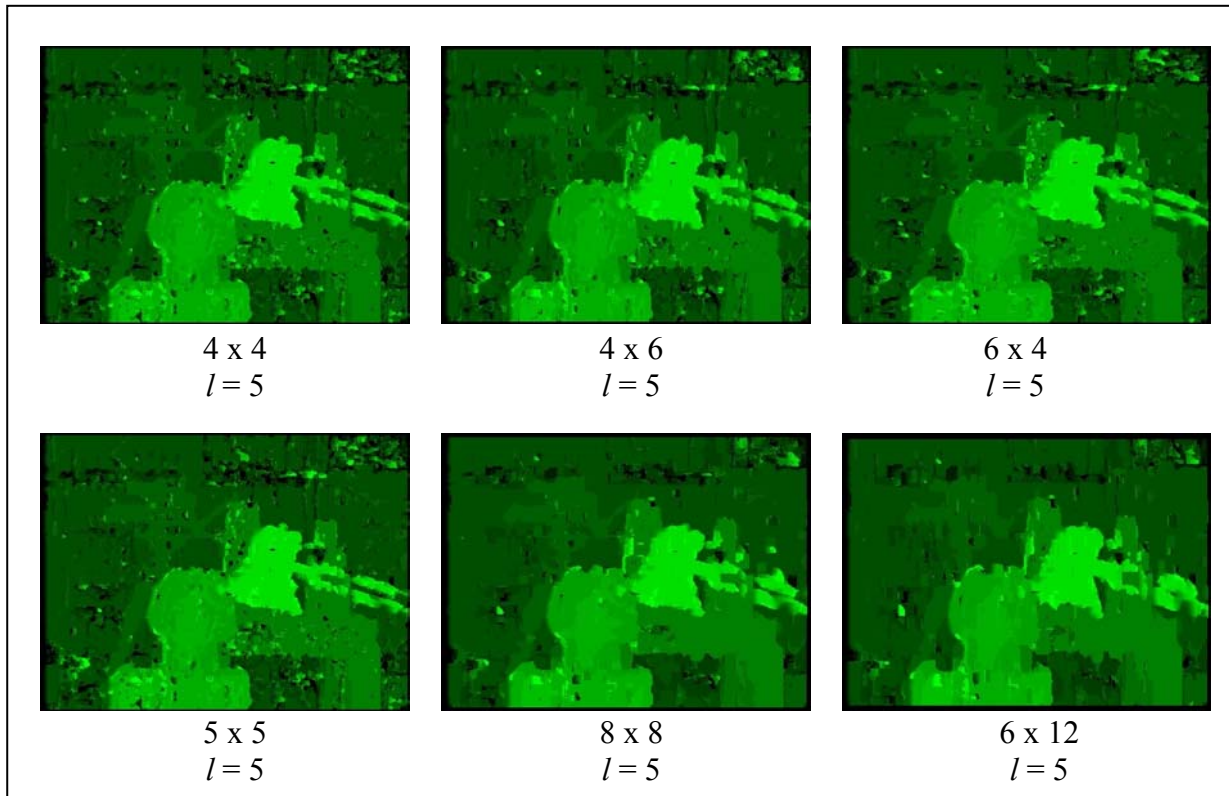


Tabella IV.29 – Risultati per tipologia III

D_{MAX} Variabile Tipo III

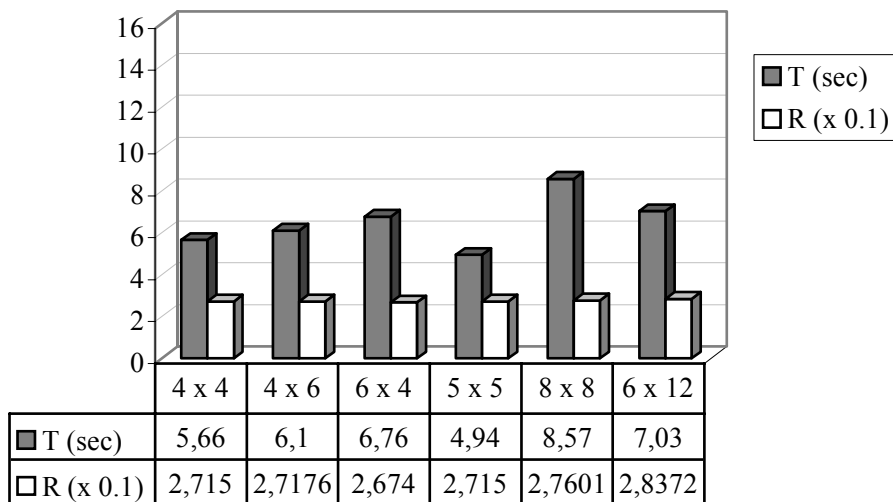


Tabella IV.30 – T ed R per tipologia III

Valutazione per Tipologia III

Parametri

Il valore del limite di ricerca l è stato fissato a 5 e questo permette di risparmiare come minimo 6 iterazioni. I risultati ed i valori di R indicano che tale valore di l è sicuramente adatto alla Tipologia III, perché si tratta di una immagine piuttosto piatta sotto l'aspetto della profondità e che non presenta forti sbalzi di disparità.

Analisi visiva

L'analisi visiva si concentra sull'area attorno all'obiettivo della telecamera dove l'occlusione è completamente scomparsa grazie alla proprietà di località che il metodo con D_{MAX} variabile possiede. Questa proprietà permette di ridurre il rumore sullo sfondo e di gestire al meglio le macchie che si generano sul tavolino e sulla telecamera. L'unico inconveniente è il buco che si crea sulla destra della lampada in primo piano in prossimità della pila più bassa di barattoli poggiata sul tavolino.

Notevole il frame 5x5 che, a differenza dei metodi precedenti, acquista una leggibilità pari alle altre cinque mappe.

Confronto con l'immagine campione

Nei riguardi dell'immagine campione il miglioramento è abbastanza netto per tutte e sei le mappe, come dimostrano i valori di R . La trasformazione positiva è apprezzabile sia rispetto al rumore presente sulla parete di fondo sia riguardo la solita zona di occlusione vicino l'obiettivo della telecamera. Il rumore diminuisce anche nella zona centrale delle immagini, in prossimità del tavolino e del busto.

Tempo di elaborazione T

Il tempo di elaborazione per il metodo D_{MAX} variabile è decisamente maggiore rispetto ad i metodi che sfruttano tutta l'ampiezza di ricerca, poiché questi ultimi possono sfruttare la proprietà di ripetitività dei calcoli per la disparità. Solo calcolando per ogni frame da 0 a D_{MAX} i valori di correlazione è possibile immagazzinare i dati che verranno sfruttati nell'iterazione seguente. La variabilità di D_{MAX} non permette di memorizzare i giusti calcoli, poiché la finestra di ricerca varia da pixel a pixel rendendo molto complicato il riuso. Il mancato riuso produce un'impennata dei costi temporali perché la complessità computazionale sale da $O(ND)$ a $O(NVn)$ con N numero totale dei pixels che compongono l'immagine, D massimo valore di disparità ammesso (D_{MAX}), n area della finestra di ricerca e V limite di ricerca per l'algoritmo variabile. Si ha $D < Vn$ dal momento che n e D sono quantità dell'ordine di 10^2 , V è dell'ordine di 10.

Errore R

L'errore R è nei casi 4x4, 6x4 e 5x5 il migliore ottenuto nella carrellata dei vari metodi, mentre per gli altri frames siamo a ridosso dei miglior risultati conseguiti tramite combinazione di SSD e SAD. Come avvenuto per Tipologia I e per Tipologia II, anche per Tipologia III si verifica lo strano caso dell'identità tra frame 4x4 e 5x5. Questo evento dimostra che D_{MAX} variabile tende ad appiattare i valori di disparità, cosa che migliora la risposta per scene in cui la profondità sia livellata, ovvero per immagini in cui non esistano oggetti a profondità troppo diverse.

IV.10 Conclusioni

La conclusione più evidente che si può estrapolare dalle oltre 120 immagini di disparità proposte durante l'analisi dei vari metodi e dai dati ad esse relativi è la seguente:

Conclusione 1 *Se si adopera un algoritmo di correlazione che permette il riuso dei calcoli già effettuati, ovvero se si ha modo di sfruttare la proprietà di ripetitività, le finestre di correlazione rettangolari con altezza maggiore dell'ampiezza sono, a parità di area, temporalmente più efficienti delle finestre quadrate e di quelle rettangolari in cui l'ampiezza sia maggiore dell'altezza.*

Per quanto riguarda l'errore R emerge un dato chiaro:

Conclusione 2 *Se si adopera una finestra di ricerca rettangolare, più aumenta la differenza di valori tra ampiezza ed altezza più l'errore R tende a crescere. In altre parole, se si vuole ottenere un valore di qualità significativo per la mappa di disparità è necessario scegliere finestre di correlazione che non siano troppo sbilanciate in altezza o in ampiezza, ma che indicativamente abbiano un rapporto tra ampiezza ed altezza maggiore di 0,5 se la finestra è più alta che ampia, oppure un rapporto minore di 1,5 in caso contrario.*

Finestra di correlazione di dimensioni $W \times H$ dove W è l'ampiezza ed H è l'altezza

Se $H > W$ si deve avere $(W / H) > 0.5$

Se $H < W$ si deve avere $(W / H) < 1.5$

L'occlusione può essere gestita nel seguente modo:

Conclusione 3 *Se si ha a che fare con una scena in cui non sono presenti forti differenze di disparità, ma che presenta una zona di occlusione, è possibile gestire l'occlusione adottando un algoritmo di correlazione che presenta una D_{MAX} variabile con un limite di ricerca che sia sufficientemente minore dell'ampiezza massima di ricerca.*

Se si desidera sfruttare la soglia interna e la distanza di picco si deve tener presente che:

Conclusione 4 *Se si intende adoperare valori di soglia interna ed esterna bisogna tener presente che un loro uso combinato permette di raggiungere una efficacia maggiore per il trattamento dell'occlusione e del rumore rispetto all'uso separato di essi.*

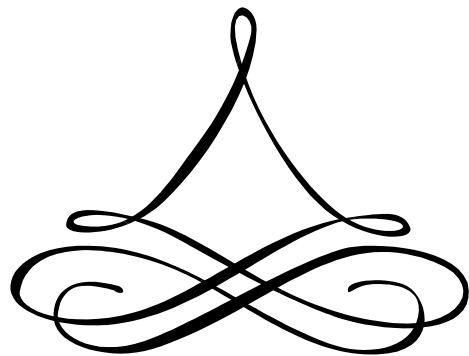
L'unione fa la forza:

Conclusione 5 *Se si uniscono i metodi di correlazione SSD e SAD si ottengono ottimi risultati soprattutto se si utilizza l'interpolazione per i match incoerenti. Il tempo di elaborazione per il metodo SSD + SAD è solo leggermente superiore al metodo SSD puro, mentre i risultati sono quasi sempre migliori.*

Appendice

A

La Stereo Visione



La stereo visione

A.1 Che cos'è la stereo visione

La stereo visione è una tecnica di ottica inversa consistente nell'ottenere informazione di profondità da una coppia di immagini, provenienti da due telecamere che inquadrano una scena da differenti posizioni. Una singola locazione fisica nello spazio tridimensionale della scena osservata corrisponde ad una **unica** coppia di pixels nelle due immagini. Questo è il principio fondante della stereo visione.

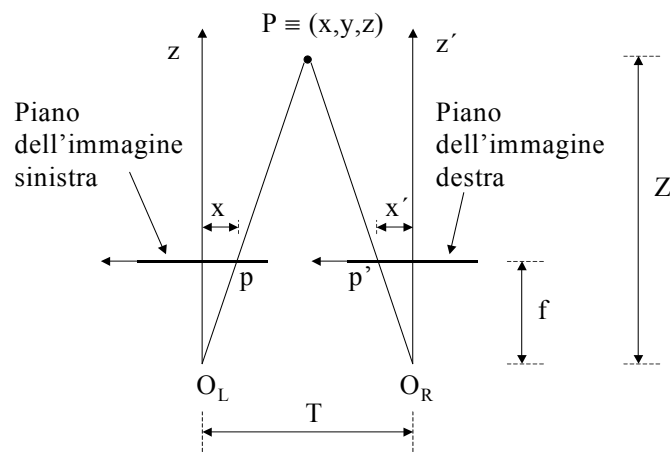


Figura A.1 – Geometria stereo per due telecamere

Avendo a disposizione le due immagini prodotte dalle telecamere, se è possibile localizzare la coppia di pixels $[p, p']$ che corrisponde ad un stesso punto P dello spazio 3D, allora (si veda la figura A.1) è possibile determinare le coordinate (x, y, z) di P . Definiamo p e p' una coppia coniugata, con p coniugato di p' e viceversa.

Definizione A.1 Una coppia coniugata è una coppia di punti in immagini diverse che sono proiezione dello stesso punto della scena.

In generale, noti gli accoppiamenti tra i punti delle due immagini e nota la posizione reciproca delle telecamere ed i parametri interni del sensore è possibile ricostruire la posizione nella scena dei punti che sono proiettati sulle due immagini.

Si considerino le immagini sinistra e destra come matrici di pixels (la seguente figura A.2 ne darà un'idea): p e p' possono essere caratterizzati in base ai rispettivi valori di riga e di colonna. Il pixel p avrà coordinate riga-colonna (i, j) mentre p' si troverà in posizione (i', j') .

Definizione A.2 La disparità è la differenza (vettore) tra punti (pixels) di una coppia coniugata quando le due immagini sono sovrapposte.

La disparità tra p e p' sarà la distanza tra le locazioni (i, j) e (i', j') . Questa distanza può essere quantificata mediante un intero che indichi la differenza in pixels tra le due posizioni della coppia coniugata.

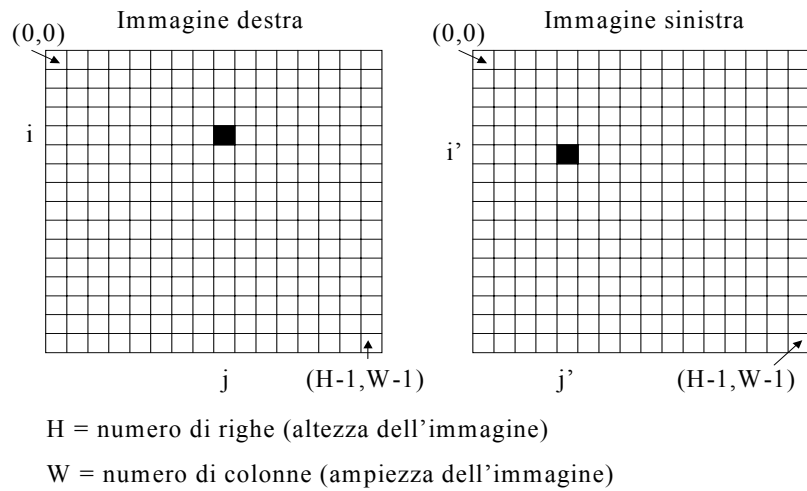


Figura A.2 – Le immagini possono pensarsi come matrici di pixels

Definizione A.3 *La mappa di disparità è l'insieme dei valori di disparità calcolati per ogni punto dell'immagine di riferimento (per esempio, quella sinistra).*

Chiaramente, la disparità può essere calcolata solo per punti della scena che sono visibili in entrambe le immagini; un punto visibile in una immagine ma non all'altra viene detto essere *occluso*.

A.2 Le occlusioni

Il mondo è pieno di occlusioni. In ogni scena, con un po' di attenzione, è possibile trovare decine di zone di occlusione.

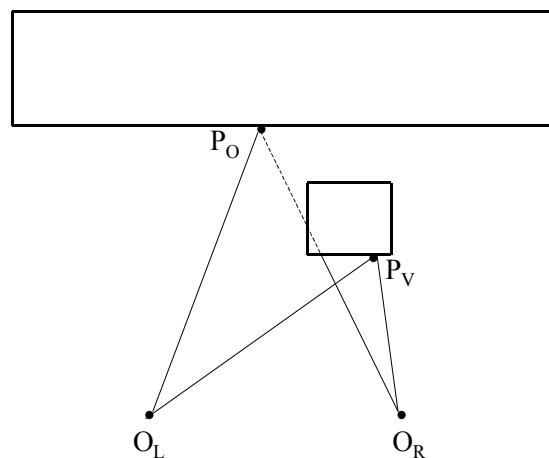


Figura A.3 – Il punto P_O è occluso per O_R

Si osservi la precedente figura A.3. Il punto P_V è visibile da entrambe le telecamere, mentre il punto P_O è visibile solo attraverso l'obiettivo O_L ; di conseguenza, mentre la profondità del punto P_V può essere calcolata, la disparità di P_O non è calcolabile, a meno che non si aggiungano immagini aggiuntive della scena o si facciano assunzioni sulla geometria di quest'ultima.

Il lettore provi, ora, ad effettuare una semplice esperienza: guardando di fronte a sè provi a coprire il proprio occhio destro con una mano. Una parte (quella all'estrema destra) della scena scomparirà alla vista. Ora, provi a coprire l'occhio sinistro: questa volta a scomparire sarà la parte più a sinistra della scena. Le due parti (sull'estrema sinistra e sull'estrema destra) che scompaiono sono alcune delle zone di occlusione di cui soffrirebbe la scena se venisse ripresa da due telecamere i cui centri ottici fossero posizionati al posto degli occhi del lettore. Questo è un primo tipo di occlusione (occlusione laterale), inevitabile poiché dovuto allo sfasamento dei due obiettivi.

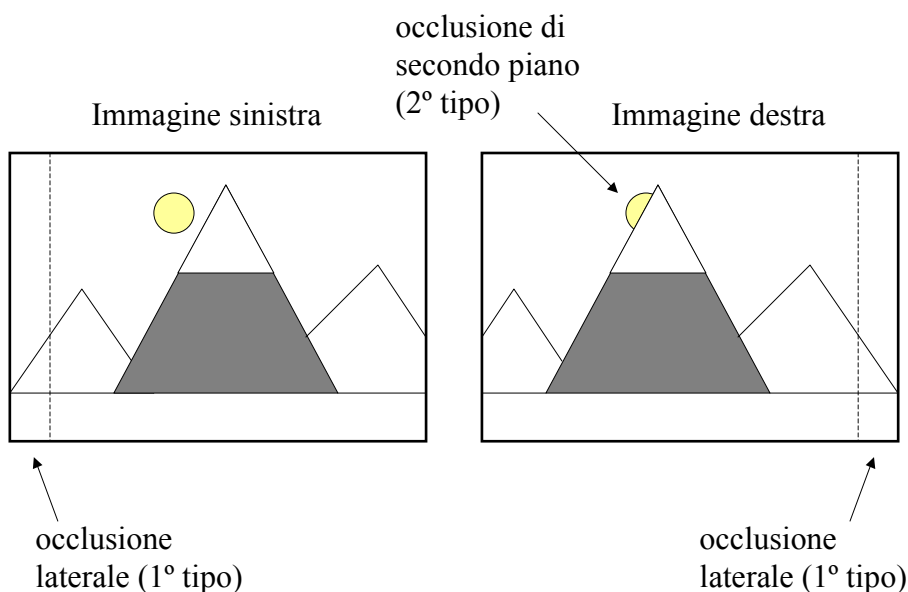


Figura A.4 – I tipi di occlusione

Esiste un secondo tipo di occlusione, dovuto alla posizione degli oggetti nella scena: capita spesso che oggetti molto vicini alle telecamere, nascondano parte degli oggetti in secondo piano ad un obiettivo ma non all'altro (occlusione di secondo piano). Si veda la figura A.4 a tal proposito.

A.3 La ricostruzione della profondità

Il processo di ricostruzione della struttura tridimensionale della scena deve confrontarsi con la soluzione di tre principali problemi: la *calibrazione*, la *corrispondenza* e la *ricostruzione*.

A.3.1 La Calibrazione

Il problema della calibrazione concerne la definizione della geometria interna ed esterna di un sistema di telecamere. La geometria esterna (parametri estrinseci) si riferisce alla posizione reciproca e all'orientazione relativa di ogni camera all'interno del sistema. La geometria interna (parametri intrinseci) riguarda la distanza focale, il centro ottico e la distorsione delle lenti per ogni singola telecamera del sistema. Una stima accurata di questi parametri è necessaria se si vuole mettere in relazione l'informazione che deriva dall'immagine (espressa in pixel) ottenuta dalle camere del sistema con l'effettiva posizione ed orientazione delle telecamere nello spazio. L'idea è

che, conoscendo le corrispondenze tra punti 3D di coordinate assolute note e le loro proiezioni, sia possibile ottenere i parametri incogniti risolvendo le equazioni della proiezione prospettica. Il problema della calibrazione non rientra negli scopi di questo scritto, per questo si assumerà che la calibrazione delle telecamere sia nota e statica.

A.3.2 La Corrispondenza

Il problema della corrispondenza consiste nel trovare per ogni punto della prima immagine il corrispondente punto nella seconda, per poi calcolare la disparità tra questi due punti e produrre la mappa di disparità. Il calcolo dell'accoppiamento è possibile sfruttando il fatto che le due immagini differiscono solo lievemente, sicché un particolare della scena appare simile nelle due immagini. Provare tutte le possibili corrispondenze è fuori dalle capacità degli attuali processori, a causa della complessità esponenziale del problema; per tale motivo, si è tentato di introdurre dei vincoli che lo rendano trattabile. Questi vincoli sono principalmente di tre tipi:

- 1) Vincoli geometrici imposti dal sistema di cattura delle immagini: probabilmente il più importante di questi vincoli è il *vincolo epipolare*, grazie al quale il problema di ricerca delle corrispondenze, in origine bidimensionale, può essere ridotto ad uno monodimensionale.
- 2) Vincoli geometrici derivanti dall'ipotesi di continuità a tratti delle superfici nella scena: il fatto che il mondo solitamente è continuo a tratti implica che punti vicini di una scena abbiano valori di profondità simili, quindi di disparità.
- 3) Vincoli fisici come quelli derivanti dalla interazione tra oggetti e luminosità della scena. Il modello più semplice e maggiormente usato è il modello *Lambertiano* [paragrafo A.5].

E' bene sottolineare che, anche utilizzando questi vincoli, l'impossibilità di ovviare ad alcuni accoppiamenti ambigui generati da occlusioni, errori nella texture e specularità delle immagini impedisce di definire soluzioni generali applicabili al problema.

A.3.3 La Triangolazione

Il problema della triangolazione consiste nel determinare la struttura tridimensionale di una scena a partire dalla mappa di disparità. La profondità di un punto P nello spazio, ripreso da due telecamere con centri ottici O_L ed O_R , è definita dall'intersezione di due rette. La prima di queste rette è quella che congiunge il centro ottico O_L della telecamera di sinistra con il punto p proiezione di P sull'immagine sinistra; la seconda retta unisce il centro ottico O_R della camera di destra con il punto p' proiezione di P sull'immagine destra (si riprenda la figura A.1). Data la distanza T (chiamata *baseline*) tra O_L ed O_R , e nota la distanza focale f delle telecamere, la profondità del punto P può essere calcolata sfruttando le leggi sulla similarità dei triangoli:

$$Z = f \frac{T}{d}$$

dove d è la disparità $d = x - x'$ trovata per P, convertita in unità metriche. Bisogna, però, effettuare una precisazione. Calcolare la disparità attraverso la semplice differenza $x - x'$ dei valori di ascissa dei punti proiezione p e p' è possibile solo se si effettua una assunzione. L'assunzione in questione è che alle immagini sia possibile applicare il già citato vincolo epipolare, il quale afferma che il corrispondente di un punto in una immagine può trovarsi solo sulla linea epipolare propria del punto nell'altra immagine.

A.4 La Geometria Epipolare

Lo sfruttamento della geometria epipolare ci aiuta nel ridurre lo spazio di ricerca delle corrispondenze: dato un punto m_1 nella immagine 1, si vuole cercare il suo coniugato m_2 nella immagine 2. Alcune semplici considerazioni geometriche indicano che il punto coniugato di m_1 deve giacere su di una linea retta nella immagine 2, chiamata *linea epipolare* di m_1 .



Figura A.5 – Geometria epipolare applicata ad immagini stereo

Se le matrici di trasformazione prospettica sono note (tramite la calibrazione) è possibile ottenere l'equazione delle linee epipolari, riducendo così il problema della corrispondenza ad una ricerca monodimensionale.

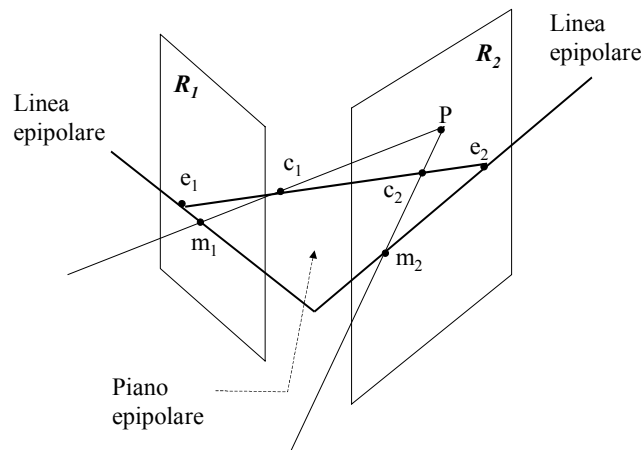


Figura A.6 – Determinazione del piano epipolare

Si consideri il caso illustrato in figura A.6: dato un punto m_1 nell'immagine 1, il suo coniugato m_2 nell'immagine 2 è vincolato a giacere sull'intersezione del piano immagine con il piano determinato da m_1 , c_1 e c_2 , detto piano epipolare. Questo avviene poiché il punto m_2 può essere la proiezione di un qualsiasi punto nello spazio giacente sul raggio ottico di m_1 . Inoltre, si osserva che tutte le linee epipolari di una immagine passano per uno stesso punto, chiamato *epipolo*, e che i piani epipolari costituiscono un fascio di piani che hanno in comune la retta passante per i centri ottici c_1 e c_2 .

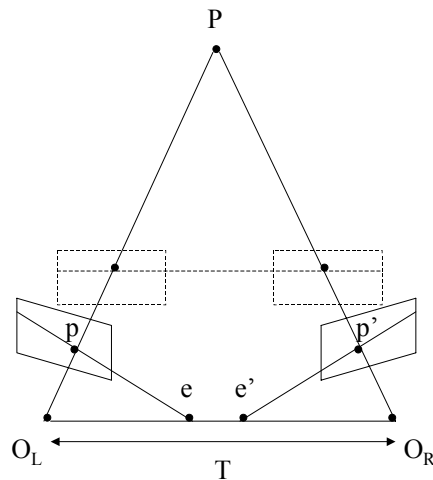


Figura A.7 – Due arbitrarie immagini della stessa scena possono essere rettificate lungo le linee epipolari (in neretto) per produrre scan lines allineate (tratteggiate)

Il punto P in figura A.7 è rappresentato dalle telecamere destra e sinistra rispettivamente come i punti p e p' . La baseline T ed i raggi ottici $O_L P$ e $O_R P$ definiscono il piano di proiezione per il punto P , ovvero il piano epipolare. La linea epipolare che passa attraverso il punto p' è l'immagine del raggio opposto, $O_L P$, che passa attraverso p . Il punto in cui le linee epipolari di una immagine intersecano la baseline è l'epipolo (e è l'epipolo per p , e' per p'). L'epipolo e corrisponde al centro ottico O_R visto dalla telecamera di sinistra.

A.5 Il modello Lambertiano

La luminosità di un pixel p nell'immagine è proporzionale alla quantità di luce diretta verso il sensore visivo dal pezzo di superficie S_p che proietta il pixel p . Questo dipende a sua volta dalle proprietà di riflettanza di S_p , dalla posizione e distribuzione delle sorgenti di luce. Vi è anche una dipendenza dalle proprietà di riflettanza del resto della scena, infatti le altre superfici della scena possono servire da sorgenti di luce indiretta riflettendo la luce ricevuta verso S_p .

La luce riflessa da un oggetto è caratterizzata dall'essere riflessa o diffusamente o specularmente. La luce riflessa diffusamente è luce che è penetrata sotto la superficie dell'oggetto, è stata assorbita e quindi riemessa. La superficie sembra ugualmente illuminata da qualsiasi punto si guardi l'oggetto. La legge del coseno di Lambert viene usata per descrivere la riflessione della luce da una superficie perfettamente diffondente, o superficie lambertiana. L'intensità E della luce riflessa da un diffusore perfetto è data da

$$E = \rho E_o \cos \theta$$

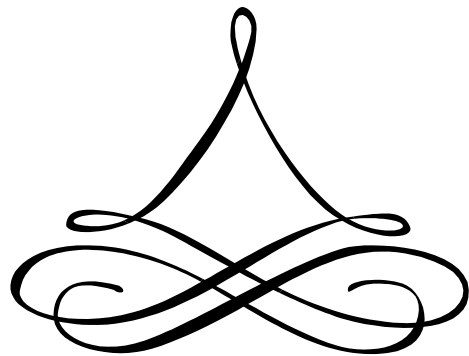
dove E_o è l'intensità della sorgente di luce; ρ è l'albedo che varia da 0 (per superfici perfettamente nere) a 1 (per superfici puramente bianche); θ è l'angolo tra la direzione della luce e la normale alla superficie.

La luce riflessa specularmente è riflessa dalla superficie esterna dell'oggetto. In questo caso l'energia della luce riflessa è concentrata principalmente in una direzione particolare, quella in cui il raggio riflesso è nello stesso piano del raggio incidente e soddisfa la condizione che l'angolo di riflessione è uguale all'angolo di incidenza. Questo è il comportamento di uno specchio perfetto.

Appendice

B

Il Codice



Appendice B

Il Codice

B.1 La classe Correlazione.java

Viene presentata parte del codice dell'applicazione che è stata creata per poter trattare le coppie stereo, al fine di produrre le mappe di disparità. Poiché l'intera applicazione consta di 5 classi Java e di un gran numero di linee di codice, sarebbe inutile riportare tutto il codice, perciò, di seguito, viene trascritta solo la classe Correlazione.java che è il nucleo dell'agente software e la classe dove effettivamente viene gestito l'algoritmo di correlazione. Tale algoritmo è particolare ed è l'unione di SSD e SAD con in più la combinazione di soglia interna e distanza di picco. Si potrebbe definire un compendio di tutti i metodi esaminati nel capitolo IV.

```
/*
 *TECNICHE DI CORRELAZIONE PER LA STEREO VISIONE
 *
 *Autore: Domenico Bloisi
 *
 *mail to: littleriver@tiscali.it
 *
 *CLASSE CORRELAZIONE
 *
 *Le immagini che vengono elaborate nel seguito sono tutte da pensarsi
 *come matrici. La cella (0,0) rappresenta il pixel piu' in alto a sinistra,
 *la cella (i,j) rappresenta il pixel di ascissa i ed ordinata j, con i
 *crescente verso destra e j crescente verso il basso
 *
 */

import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.awt.image.*;
import java.lang.Math;
import java.awt.color.ColorSpace;

public class Correlazione{
    //le due immagini da correlare
    private Image immagine1, immagine2;
    //x e' la larghezza dell'immagine, y e' l'altezza
    //con 1 che indica la prima immagine e 2 la seconda
    private int x1, x2, y1, y2;
    //per comodita' vengono istanziati due array bidimensionali
    //[i][j] dove i e' il numero di colonna (ascissa)
    //e j e' il numero di riga (ordinata)
    private byte[][] matrix1, matrix2;
    //dimensioni del frame di ricerca:
    //altezza
    private int height;
    //ampiezza
    private int width;
    //array che contiene la mappa di disparità
    private int[] disparityMap;
```

```

//valore di DMax, ovvero del massimo range di ricerca per trovare
//il valore di disparita' migliore tra il pixel che si sta processando
//(appartenente alla prima immagine) ed il pixel corrispondente
//(appartenente alla seconda)
private int dMax;
//valore di soglia per l'algoritmo SSD
private int thresholdSSD;
//valore di soglia per l'algoritmo SAD
private int thresholdSAD;
//valore minimo della distanza tra i due maggiori picchi
//per l'algoritmo SSD
private int piccoSSD;
//valore minimo della distanza tra i due maggiori picchi
//per l'algoritmo SAD
private int piccoSAD;
//i pixel che si trovano troppo vicino al margine dell'immagine
//non vengono processati. Il primo pixel ad essere processato
//ha coordinate (initialAsc, initialOrd), l'ultimo ha coordinate
//(finalAsc, finalOrd)
private int initialAsc;
private int initialOrd;
private int finalAsc;
private int finalOrd;
//Per implementare un algoritmo che sia in grado di riutilizzare
//alcuni risultati gia' calcolati, bisogna tenere memoria di tali
//risultati. Viene creato un array per tale motivo
private int[] sommeSSD;
private int[] sommeSAD;
//media dei pixel adiacenti il pixel di interesse
private int media;

//il costruttore di questa classe riceve come argomenti l'immagine di
//destra, l'immagine di sinistra, il valore di ampiezza del frame di
//ricerca, il valore di altezza di questo stesso frame, il valore per
//dMax
public Correlazione(ImageIcon imm1, ImageIcon imm2, int w, int h, int dm){
    //valore dell'ampiezza del frame di ricerca
    width = w;
    //valore dell'altezza del frame di ricerca
    height = h;
    //ampiezza della prima immagine
    x1 = imm1.getIconWidth();
    //altezza della prima immagine
    y1 = imm1.getIconHeight();
    //ampiezza della seconda immagine
    x2 = imm2.getIconWidth();
    //altezza della seconda immagine
    y2 = imm2.getIconHeight();
    //conversione da ImageIcon ad Image
    immagine1 = imm1.getImage();
    immagine2 = imm2.getImage();
    //recupero dell'array rappresentante la prima immagine
    byte[] pixels1 = recuperaPixels(immagine1, 0, 0, x1, y1);
    //trasformazione dell'array monodimensionale in matrice
    matrix1 = new byte[x1][y1];
    //i pixels sono stati immagazzinati nell'array nel seguente modo:
    //il pixel che nell'immagine (pensata come una matrice) si trova
    //nella posizione (i,j) e' memorizzato nella posizione [(j*x)+i]
    //dell'array, dove x rappresenta la larghezza dell'immagine
    //Questa politica di memorizzazione e' propria del costruttore
    //della classe PixelGrabber
    for(int i = 0; i < x1; i++){

```



```

        for(int j = 0; j < y1; j++){
            matrix1[i][j] = pixels1[(j * x1) + i];
        }
    }
    //recupero dell'array rappresentante la seconda immagine
    byte[] pixels2 = recuperaPixels(immagine2, 0, 0, x2, y2);
    //trasformazione dell'array monodimensionale in matrice
    matrix2 = new byte[x2][y2];
    for(int k = 0; k < x2; k++){
        for(int m = 0; m < y2; m++){
            matrix2[k][m] = pixels2[(m * x2) + k];
        }
    }
    //Creazione dell'array che contiene la mappa di disparità.
    //L'array disparityMap e' strutturato nel seguente modo:
    //ogni cella contiene un valore intero che rappresenta
    //la distanza d (in pixel) tra il valore di ascissa i del
    //pixel che si sta processando nella prima immagine ed il
    //valore di ascissa i-d del pixel che meglio corrisponde ad
    //esso nella seconda immagine
    disparityMap = new int[x1 * y1];
    //settaggio del valore di DMax
    dMax = dm;
    //inizializzazione di media
    media = 0;
}

//Il metodo recuperPixels estrae dall'immagine un array di byte che
//rappresenta tutti i pixels
public byte[] recuperaPixels(Image img, int x, int y, int w, int h) {
    //viene creato un array monodimensionale per contenere
    //tutte le informazioni sui pixel dell'immagine
    byte[] pixels = new byte[w * h];
    //per manipolare i pixels adoperiamo la classe PixelGrabber
    PixelGrabber pg = new PixelGrabber(img, x, y, w, h, false);
    //Color Model. Si tratta di immagini in bianco e nero (GRAYSCALE)
    ComponentColorModel ccm = new ComponentColorModel(
        ColorSpace.getInstance(ColorSpace.CS_GRAY),
        false,
        false,
        ColorModel.OPAQUE,
        DataBuffer.TYPE_BYTE);
    //settaggio del ColorModel per il PixelGrabber
    pg.setColorModel(ccm);
    //prendiamo i pixels
    try {
        pg.grabPixels();
    } catch (InterruptedException e) {
        System.err.println("Interruzione nel flusso di pixel");
        return null;
    }
    if ((pg.getStatus() & ImageObserver.ABORT) != 0) {
        System.err.println("Errore nello stato dell'immagine");
        return null;
    }
    //creiamo l'array di byte che rappresenta tutti i pixel
    pixels = (byte[])pg.getPixels();
    return pixels;
}

//L'immagine viene pensata come un reticolo in cui ogni cella
//e' un pixel.

```

```

//I pixels vengono processati a partire da quello piu' in alto a
//sinistra, per poi spostarsi verso destra fino alla fine
//della riga. Il processo riprende dal pixel piu' a sinistra
//della riga successiva
public int[] esecuzione() throws Exception{
    //settaggio dei valori delle coordinate che rimarranno
    //fisse durante lo svolgimento dell'algoritmo
    initialAsc = width / 2;
    initialOrd = height / 2;
    finalAsc = x1 - (width / 2) - 1;
    finalOrd = y1 - (height / 2) - 1;
    //settaggio della dimensione degli arrays che conservano
    //i calcoli gia' effettuati
    sommeSSD = new int[dMax + 1];
    sommeSAD = new int[dMax + 1];
    //cicli che settano la disparita' dei pixel che giacciono
    //sui margini esterni
    //Questa disparita' e' posta a zero. Cio' non implica
    //perdita di informazione poiche' il numero dei pixel
    //sui margini e' molto piccolo
    for(int j1 = 0; j1 < initialOrd; j1++){
        for(int i1 = 0; i1 < x1; i1++){
            disparityMap[(j1 * x1) + i1] = 0;
        }
    }
    for(int j2 = finalOrd + 1; j2 < y1; j2++){
        for(int i2 = 0; i2 < x1; i2++){
            disparityMap[(j2 * x1) + i2] = 0;
        }
    }
    for(int j3 = initialOrd; j3 <= finalOrd; j3++){
        for(int i3 = 0; i3 < initialAsc; i3++){
            disparityMap[(j3 * x1) + i3] = 0;
        }
    }
    for(int j4 = initialOrd; j4 <= finalOrd; j4++){
        for(int i4 = finalAsc + 1; i4 < x1; i4++){
            disparityMap[(j4 * x1) + i4] = 0;
        }
    }
    //scansione dell'immagine. Si parte dal pixel piu' in alto
    //a sinistra, per poi continuare con il pixel sottostante
    //fino alla fine della prima colonna dell'immagine. Si riparte dal
    //pixel piu' in alto immediatamente a destra (colonna accanto).
    //primo for: scansione delle ascisse
    for(int asc = initialAsc; asc <= finalAsc; asc++){
        //secondo for: scansione delle ordinate
        for(int ord = initialOrd; ord <= finalOrd; ord++){
            //riempimento della mappa di disparita'
            //algoritmo che sfrutta la somma della
            //differenza dei quadrati
            int d1 = processaSSD(asc, ord);
            //algoritmo che sfrutta la somma della
            //differenza assoluta
            int d2 = processaSAD(asc, ord);
            //se i risultati combaciano la disparita'
            //e' accettata
            if(d1 == d2){
                disparityMap[(ord * x1) + asc] = d1;
            }
            //altrimenti viene considerata la media delle
            //disparita' dei pixel adiacenti al pixel
            //di coordinate (asc,ord)
        }
    }
}

```

```

        else{
            //settaggio del valore di media
            media = 0;
            for(int or = ord - 1; or <= ord + 1; or++){
                media += disparityMap[(or * x1) + (asc - 1)];
            }
            media += (disparityMap[((ord - 1) * x1) + asc]);
            media = media / 4;
            disparityMap[(ord * x1) + asc] = media;
        }
    }
}
//fine della correlazione. Viene restituita la mappa di disparita'
return disparityMap;
}

//Per trovare la correlazione tra le due immagini bisogna aver
//ben presente che esse hanno subito un processo di wrapping
//il quale le ha modificate in modo che il pixel (i,j) della
//prima immagine (pensata come una matrice di pixels) si trovi nella
//seconda immagine sulla riga j in corrispondenza della colonna i-x ,
//dove x e' proprio il valore che vogliamo trovare tramite il confronto
//della differenza tra il pixel (appartenente alla prima immagine)
//ed il suo corrispondente (appartenente alla seconda immagine)
private int processaSSD(int asc,int ord){

    //per ogni pixel (i,j) deve essere trovata una finestra
    //di dimensioni (width x height) di cui (i,j)
    //ne sia il centro.
    //Il valore di disparita' verra' trovato calcolando la somma delle
    //differenze quadrate tra ogni pixel del frame di ricerca della
    //prima immagine ed il corrispondente pixel nella seconda

    //disp rappresenta il valore di disparita' minimo tra le due
    //finestre di ricerca.
    int disp = -1;
    //dispSeconda serve per implementare la distanza di picco
    int dispSeconda = -1;

    //somma rappresenta la sommatoria delle differenze quadrate
    //dei valori in scala di grigi dei pixel delle due
    //finestre di ricerca. Una volta trovato un
    //valore di disparita' per una finestra, l'interesse si sposta
    //(nella seconda immagine) sulla finestra immediatamente a sinistra
    //rispetto alla finestra corrente.
    //Viene inizializzata a zero
    int somma = 0;

    //sommaMinima conserva il minimo valore di somma attualmente
    //trovato. Questo valore deve essere confrontato con il valore
    //trovato per somma. Assume inizialmente un valore fittizio
    int sommaMinima = -1;
    //sommaSeconda serve per implementare la distanza di picco
    int sommaSeconda = -1;

    //il valore booleano soglia e' un lock che controlla
    //se il valore trovato per somma e' inferiore al valore
    //impostato come soglia
    boolean sogliaTrovata = false;
    //Non sempre e' possibile trovare dei frame in cui il pixel di
    //interesse sia a distanza DMax dal bordo. Per questo va considerato
    //il caso in cui il pixel di coordinate (asc,ord) abbia un valore
    //di asc minore di DMax. Viene settato un valore limite (limit)

```

```

//limit e' il massimo valore che puo' assumere DMax
//infatti, se il pixel di interesse si trova in ascissa i minore
//di DMax il valore massimo di disparita' tra i due pixel e' i
int limit = dMax;
if((asc - initialAsc - limit) < 0){
    limit = asc - initialAsc;
}

//ordMin e' il valore minimo che possono assumere come ordinata
//i pixels piu' in alto del frame di ricerca. Naturalmente non
//si puo' cercare al di fuori dei limiti dell'immagine
int ordMin = ord - initialOrd;

//ordMax e' il valore massimo che possono assumere come ordinata
//i pixels piu' in alto del frame di ricerca. Naturalmente, non
//si puo' cercare al di fuori dei limiti dell'immagine
int ordMax = ord + initialOrd;

//ascMin e' il valore minimo che possono assumere come ascissa
//i pixels piu' a sinistra del frame di ricerca. Naturalmente, non
//si puo' cercare al di fuori dei limiti dell'immagine
int ascMin = asc - (width / 2);

//ascMax e' il valore massimo che possono assumere come ascissa
//i pixels piu' a destra del frame di ricerca. Naturalmente, non
//si puo' cercare al di fuori dei limiti dell'immagine
int ascMax = asc + (width / 2);

//il primo ciclo gestisce la marcia indietro del frame di ricerca
//nell'immagine 2. Infatti un punto che nell'immagine 1 avra'
//ascissa i, nell'immagine 2 avra' ascissa i-x, con x che deve
//essere trovata e che deve essere minore o uguale a DMax
if(ord == initialOrd){
    for(int d = 0; d <= limit; d++){
        //il secondo ed il terzo ciclo insieme implementano la
        //scansione di una matrice di dimensioni (width x height)
        //in cui il centro ha coordinate (i,j)
        for(int o = 0; o <= height; o++){
            for(int a = ascMin; a <= ascMax; a++){
                //somma delle differenze quadrate
                somma +=
                Math.pow(matrix1[a][o] - matrix2[a - d][o], 2);
            }
        }
        //memorizzo i valori di somma trovati, in modo che
        //possano essere riutilizzati al prossimo ciclo
        sommeSSD[d] = somma;
        //si tiene memoria del minimo
        if(sommaMinima == -1){
            sommaSeconda = somma;
            sommaMinima = somma;
            dispSeconda = d;
            disp = d;
            if(somma <= thresholdSSD){
                sogliaTrovata = true;
            }
        }
    }
    //per trovare il minimo valore di somma confronto la
    //somma corrente con il valore sommaMinima che il piu'
    //basso valore per somma finora trovato
    //bisogna anche considerare che puo' essere immesso
    //un valore di soglia per aumentare la risoluzione

```



```

    }
}
//per trovare il minimo valore di somma confronto la
//somma corrente con il valore sommaMinima che il piu'
//basso valore per somma finora trovato
else if((somma < sommaMinima) && (!sogliaTrovata)){
    if(d - disp > 1){
        sommaSeconda = sommaMinima;
        sommaMinima = somma;
        dispSeconda = disp;
        disp = d;
    }
    else{
        sommaMinima = somma;
        disp = d;
    }
    if(somma <= thresholdSSD){
        sogliaTrovata = true;
    }
}
//reset di tutti i valori di somma per la
//realizzazione del prossimo ciclo
somma = 0;
}
if((!sogliaTrovata) &&
((sommaSeconda - sommaMinima) < piccoSSD)){
    return 0;
}
else{
    return disp;
}
}
}

```

```

private int processaSAD(int asc,int ord){

    //Le variabili sono identici a processaSSD e valgono le stesse
    //considerazioni effettuate in precedenza.
    //Cambia solo il calcolo della somma

    int disp = -1;
    int dispSeconda = -1;

    int somma = 0;

    int sommaMinima = -1;
    int sommaSeconda = -1;

    boolean sogliaTrovata = false;

    int limit = dMax;
    if((asc - initialAsc - limit) < 0){
        limit = asc - initialAsc;
    }

    int ordMin = ord - initialOrd;

    int ordMax = ord + initialOrd;

    int ascMin = asc - (width / 2);

    int ascMax = asc + (width / 2);

```

```

if(ord == initialOrd){
    for(int d = 0; d <= limit; d++){
        for(int o = 0; o <= height; o++){
            for(int a = ascMin; a <= ascMax; a++){
                //somma dei valori assoluti
                somma +=
                    Math.abs(matrix1[a][o] - matrix2[a - d][o]);
            }
        }
        sommeSAD[d] = somma;
        if(sommaMinima == -1){
            sommaSeconda = somma;
            sommaMinima = somma;
            dispSeconda = d;
            disp = d;
            if(somma <= thresholdSAD){
                sogliaTrovata = true;
            }
        }
        else if((somma < sommaMinima) && (!sogliaTrovata)){
            if(d - disp > 1){
                sommaSeconda = sommaMinima;
                sommaMinima = somma;
                dispSeconda = disp;
                disp = d;
            }
            else{
                sommaMinima = somma;
                disp = d;
            }
            if(somma <= thresholdSAD){
                sogliaTrovata = true;
            }
        }
        somma = 0;
    }
    if(!sogliaTrovata) &&
        ((sommaSeconda - sommaMinima) < piccoSAD)){
        return 0;
    }
    else{
        return disp;
    }
}
else{
    for(int d = 0; d <= limit; d++){
        int precedente = 0, attuale = 0;
        for(int n = ascMin; n <= ascMax; n++){
            precedente +=
                Math.abs(matrix1[n][ordMin - 1] -
                    matrix2[n-d][ordMin-1]);
            attuale +=
                Math.abs(matrix1[n][ordMax] -
                    matrix2[n - d][ordMax]);
        }
        somma = sommeSAD[d] - precedente + attuale;
        sommeSAD[d] = somma;
        if(sommaMinima == -1){
            sommaSeconda = somma;
            sommaMinima = somma;
            dispSeconda = d;
            disp = d;
            if(somma <= thresholdSAD){

```

```

        sogliaTrovata = true;
    }
}
else if((somma < sommaMinima) && (!sogliaTrovata)){
    if(d - disp > 1){
        sommaSeconda = sommaMinima;
        sommaMinima = somma;
        dispSeconda = disp;
        disp = d;
    }
    else{
        sommaMinima = somma;
        disp = d;
    }
    if(somma <= thresholdSAD){
        sogliaTrovata = true;
    }
}
    somma = 0;
}
if(!sogliaTrovata) &&
((sommaSeconda - sommaMinima) < piccoSAD)){
    return 0;
}
else{
    return disp;
}
}
}

//serie dei getters
public int getImageWidth(){
    return x1;
}
public int getImageHeight(){
    return y1;
}

//serie dei setters
public void setWidth(int w){
    width = w;
}

public void setHeight(int h){
    height = h;
}

public void setDMax(int d){
    dMax = d;
}

public void setThresholdSSD(int tssd){
    thresholdSSD = tssd;
}

public void setThresholdSAD(int tsad){
    thresholdSAD = tsad;
}

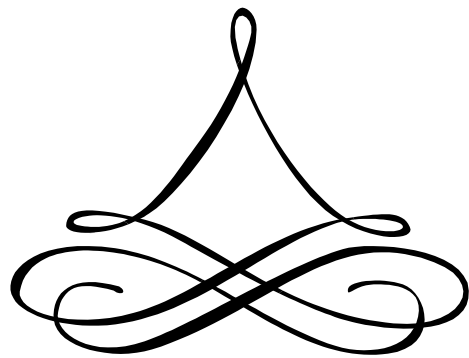
public void setPiccoSSD(int pssd){
    piccoSSD = pssd;
}
}

```



```
public void setPiccoSAD(int psad) {  
    piccoSAD = psad;  
}  
}
```


Bibliografia



Bibliografia

- [1] Myron Z. Brown, Darius Burschka e Gregory D. Hager, *Advances in Computational Stereo*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, num. 8, pag. 993-1008, Agosto 2003
- [2] Kurt Konolige, Artificial Intelligence Center, SRI International, *Small Vision Systems: Hardware and Implementation*
- [3] H.K. Nishihara, *PRISM, a Practical Real-Time Imaging Stereo Matcher*, Technical Report A.I. Memo 780, MIT, 31 pagine, Cambridge, MA, 1984
- [4] J. Banks e P. Corke, *Quantitative Evaluation of Matching Methods and Validity Measures for Stereo Vision*, Int'l J. Robotics Research, vol. 20, num. 7, 2001
- [5] O. Faugeras, B. Hotz, H. Mathieu, T. Viéville, Z. Zhang, P. Fua, E. Théron, L. Moll, G. Berry, J. Vuillemin, P. Bertin e C. Proy, *Real time correlation-based stereo: algorithm, implementations and applications*, INRIA Technical Report 2013, Agosto 1993
- [6] V. Venkateswar e R. Chellappa, *Hierarchical Stereo and Motion Correspondence Using Feature Groupings*, Int'l J. Computer Vision, vol. 15, pag. 265-269, 1995
- [7] R. Szeliski e R. Zabih, *Vision Algorithms: Theory and Practice*, cap. An Experimental Comparison of Stereo Algorithms, pag. 1-19. Corfu, Grecia: Springer-Verlag, Settembre 1999
- [8] Heiko Hirschmüller, Centre for Computational Intelligence, DeMontfort University, Leicester, UK, *Improvements in Real-Time Correlation-Based Stereo Vision*, Proceedings of the IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV'01), 2001
- [9] Patrik Nilsson, *Stereo Vision for Mobile Robots*, Department of Engineering and Technology, DeMontfort University, UK, 2003
- [10] Stuart J. Russell e Peter Norvig, *Intelligenza Artificiale, un approccio moderno*, Edizione italiana a cura di Luigia Carlucci Aiello, Prentice Hall International, UTET 1998
- [11] H. Hirschmüller, P. R. Innocent, J. Garibaldi, *Real-Time Correlation-Based Stereo Vision with Reduced Border Errors*, Draft for the International Journal of Computer Vision, Centre for Computational Intelligence, DeMontfort University, Leicester, UK, 2002
- [12] Andrea Fusiello, *Tutorial on Rectification of Stereo Images*, Dipartimento di Matematica e Informatica, Università di Udine, 2003
- [13] Luca Iocchi, *Multiresolution Stereo Vision System for Mobile Robots*, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", AI Center – SRI International, Aprile 1998
- [14] C. Lawrence Zitnick e Takeo Kanade, *A Cooperative Algorithm for Stereo Matching and Occlusion Detection*, CMU-RI-TR-99-35, Ottobre 1999

- [15] Daniel Scharstein e Richard Szeliski, *A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms*, Int'l J. Computer Vision, vol. 47, num.1, pag. 7-42, 2002
- [16] Aaron F. Bobick e Stephen S. Intille, *Large Occlusion Stereo*, The International Journal of Computer Vision
- [17] Kurt Konolige e David Beymer, Artificial Intelligence Center, SRI International, *SRI Small Vision Systems User's Manual Software version 1.4*, Dicembre 1999
- [18] Sun Microsystems, *The Java Tutorial, a practical guide for programmers*, Settembre 2002