Università degli Studi di Padova

Facoltà di Ingegneria

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea in Ingegneria Elettronica

Tesi di Laurea

Ricostruzione 3D tramite spacetime stereo

Relatore

Prof. Guido Maria Cortelazzo

Laureando

Paolo Furlan

Anno Accademico 2009/2010

Sommario

Nell'àmbito del problema della costruzione di modelli tridimensionali di scene reali, è importante disporre di un sistema in grado di produrre risultati di qualità tale da consentire il loro uso come termine di paragone per la valutazione delle prestazioni di nuovi algoritmi. In questo contesto lo space-time stereo attivo costituisce una tecnica precisa e affidabile, oltre che più economica e semplice in termini di messa a punto sperimentale rispetto alle tradizionali tecniche che si avvalgono di un laser scanner.

Dopo l'introduzione, una presentazione sommaria degli aspetti teorici della triangolazione (cap.1) e della forma delle immagini che costituiscono l'input degli algoritmi di calcolo delle disparità (cap.2), si passerà a una ricognizione della classificazione degli algoritmi stessi, finalizzata alla contestualizzazione del metodo fixed windows su cui si basa lo space-time stereo (cap.3). L'implementazione di quest'ultimo costituisce l'obiettivo principale dell'attività svolta (cap.4). In conclusione si presentano un'analisi dei risultati e osservazioni ulteriori (cap.5).

L'algoritmo realizzato permette la costruzione di mappe di disparità che hanno una risoluzione pari o prossima a quella delle immagini da cui hanno origine, e un basso numero di corrispondenze errate o ambigue. Una prova su una scena di geometria nota ha fornito indicazioni sul grado di precisione assoluta del metodo.

Indice

Introduzione

Uno dei problemi più lungamente e ampiamente indagati nel campo della *computer vision* è quello della ricostruzione tridimensionale, ovvero l'acquisizione di informazioni sulla geometria dello spazio a partire da immagini bidimensionali. I primi procedimenti basati sull'uso di immagini fotografiche risalgono al XIX secolo con la nascita (solo un decennio dopo i primi dagherrotipi) della fotogrammetria[1], cioè di una tecnica finalizzata alla costruzione di carte altimetriche a partire dalle informazioni contenute in più vedute aeree. Legati allo stesso campo di applicazione sono i primi algoritmi completamente automatici, risalenti agli anni '70[2]. Più recentemente, la diffusione di processori e sensori di prestazioni elevate e costi ridotti ha reso possibile l'elaborazione delle immagini in tempo reale, ampliando i settori in cui queste tecniche trovano impiego. Un esempi tipico è costituito dalla misura di superfici nel controllo di qualità, in catene di produzione. Altri esempi si trovano in àmbito medico, laddove sia utile una valutazione quantitativa e non soggettiva di forme anatomiche; o commerciale, per creare cataloghi tridimensionali; o ancora nel settore industriale, per guidare robot nell'assemblaggio automatico (figura 1).

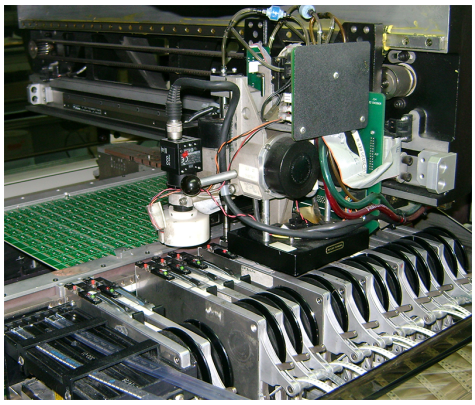


Fig. 1: robot pick & place, dotato di sensori video (da *http://support.eclipse-ems.com/*).

[1]Burtch R. *History of Photogrammetry*, http://www.ferris.edu/faculty/burtchr/sure340/notes/History.pdf

[2]Hannah, M. J. *Computer Matching of Areas in Stereo Images.* 1974 Ph.D. thesis, Stanford University.

Fig. 2: affreschi di Villa Barbaro a Maser, opera di Paolo Veronese.

Precedente allo sviluppo di queste tecniche – e legata allo studio della visone umana – è la nozione di stereopsi, cioè il processo mediante il quale è possibile ricavare informazioni sulla struttura tridimensionale degli oggetti, e dello spazio in cui si trovano, a partire da una visione binoculare. È un dato comune dell'esperienza che di un oggetto osservato i due occhi vedano aspetti diversi. Una differenza pressoché nulla a grande distanza, ma che diventa massima qualora l'oggetto si trovi in prossimità del punto di osservazione. Illusioni prospettiche come quella di figura 2 possono ottenere il loro scopo solo se realizzate in modo da essere osservate da una sufficiente distanza (oltre che da un' angolazione opportuna) in modo tale che i due occhi vedano la parete da prospettive praticamente uguali. Da una posizione troppo ravvicinata la differenza tra gli stimoli che giungono ai due occhi risulterebbe rivelatrice del fatto che si tratta di una superficie piana. D'altra parte sono i metodi stessi della prospettiva scientifica ad avere come presupposto una visione monoculare (figura 3).

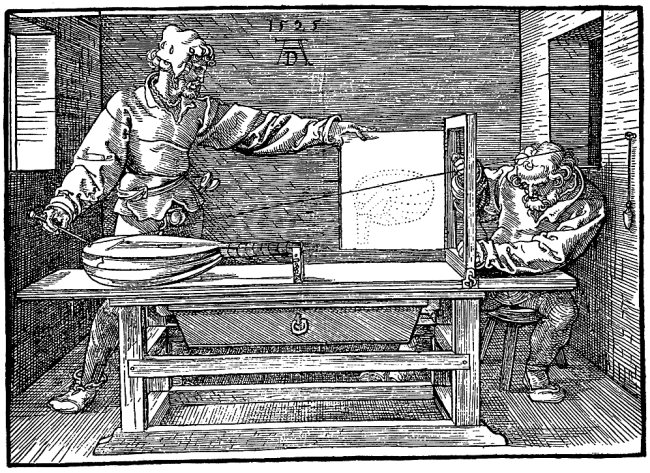


Fig. 3: costruzione di una figura secondo la prospettiva scientifica, in un'incisione di Dürer. La cordicella sostituisce i raggi ottici convergenti verso un unico punto, come nella visione monoculare.

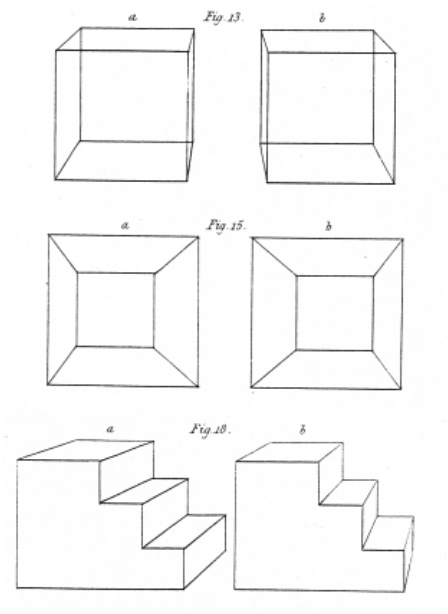


Fig. 4: coppie di immagini utilizzate in uno stereoscopio (da http://www.stereoscopy.com/library/wheatstone-paper1838.html).

L'illusione di trovarsi davanti a una figura solida che si trovi a distanza ridotta può invece essere prodotta solo tenendo conto delle differenti prospettive dei due occhi, e sottoponendo a essi immagini distinte. È il principio su cui si basa lo stereoscopio: due immagini (figura 4), costruite come prospettive diverse di uno stesso oggetto, vengono utilizzate come stimoli separati per i due occhi. Osservate nelle opportune condizioni le due immagini si fondono percettivamente nella sensazione di un'unica forma solida, con caratteristiche che non sono proprie delle immagini bidimensionali da cui ha origine (ad esempio una forma solida simmetrica può essere ottenuta a partire da una coppia di immagini che prese singolarmente sono asimmetriche).

Le sedie di Ames (figura 5) possono esemplificare il tipo di ambiguità che possono presentarsi a una visione monoculare. Si tratta di un esperimento nel campo della percezione visiva, nel quale diverse strutture geometriche sono rese visibili da un unico punto di vista (l'esperimento è illustrato in dettaglio in http://www.archive.org/details/amesdemonstratio00itte, da dove proviene anche l'immagine). Allo stimolo che giunge sulla retina da ciascuno dei segmenti sospesi non è possibile associare alcuna profondità; esso può così essere interpretato come un piccolo bastoncino prossimo all'osservatore o come una lunga asta posta a maggiore distanza. In questo modo l'osservatore, facendo appello a un catalogo di forme familiari, è persuaso di vedere una sedia.

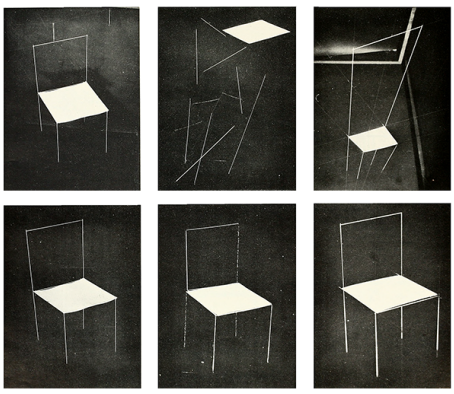


Fig. 5: sedie di Ames. Sopra, viste da una prospettiva generica; sotto, dal punto di vista a cui si è costretti nell esperimento.

In realtà il meccanismo che induce all'errore in queste particolari condizio-

ni sperimentali (il ricorso a un archivio di conoscenze non necessariamente di tipo visivo) è lo stesso che permette a un umano di effettuare buone stime di distanza e di profondità anche servendosi di un solo occhio. In questo senso i metodi di ricostruzione basati su stereopsi computazionale ovviamente differiscono. Infatti nella visione computazionale l'unica informazione considerata è quella presente in una coppia di immagini acquisite da un sistema binoculare, ed è la valutazione delle differenti posizioni delle proiezioni di un punto a fornire una misura indiretta della distanza del punto stesso.

Capitolo 1

Geometria del problema nel caso ideale

Un'immagine bidimensionale è per sua natura priva di informazioni sufficienti a localizzare un punto nello spazio. L'osservazione di uno stesso punto da due posizioni diverse permette invece di impostare il problema di triangolazione, che conduce al calcolo delle coordinate spaziali.

1.1 Descrizione degli aspetti geometrici essenziali del problema

Nella sua formulazione più semplice il sistema adottato per acquisire le due immagini può essere schematizzato come in figura 1.1. In particolare si suppone che la proiezione di un punto dello spazio sul piano del sensore avvenga lungo una linea retta, quindi in assenza di lenti (modello stenopeico, o prospettico), che gli assi ottici siano paralleli e che i piani dei sensori coincidano.

Il sistema cartesiano XZ identifica un piano ortogonale ai due sensori e contenente i due assi ottici (di equazioni $X = 0$ e $X = b$); C e C' sono le posizioni dei centri focali delle due fotocamere; M è la proiezione sullo stesso piano XZ di un generico punto che cade nel campo visivo delle due ottiche, aventi entrambe lunghezza focale f. Con x_{sx} e x_{dx} sono indicate le ascisse delle immagini di M sui due sensori S_{sx} e S_{dx}. Il parametro b identifica la distanza tra i due assi ed è detto *baseline*.

I rapporti geometrici tra le grandezze in gioco sono le proporzioni tra le misure di lati di triangoli simili:

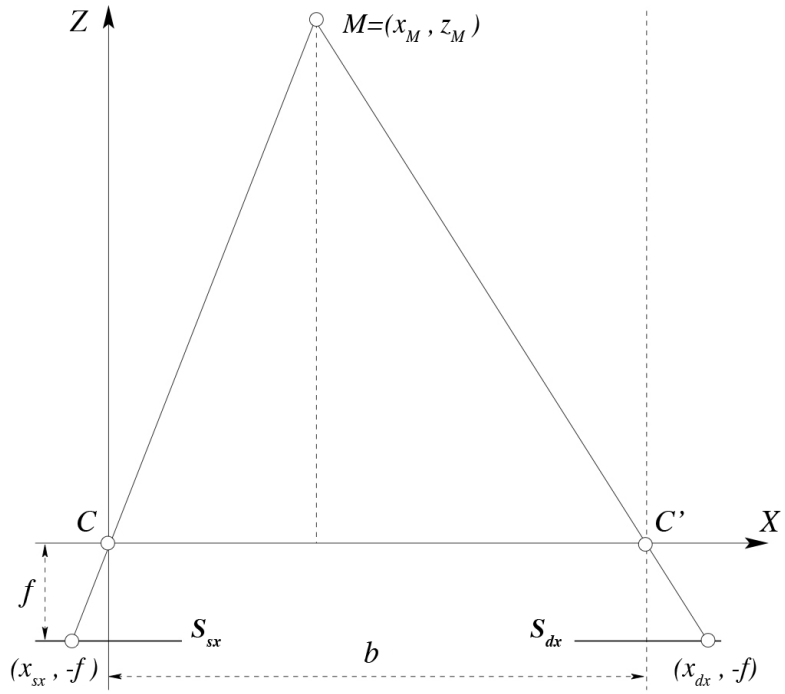


Fig. 1.1: schematizzazione del sistema di acquisizione stereoscopico.

$$\begin{cases} \dfrac{f}{z_M} = \dfrac{-x_{sx}}{x_M} \\ \\ \dfrac{f}{z_M} = \dfrac{x_{dx} - b}{b - x_M} \end{cases} \Rightarrow \begin{cases} x_M = \dfrac{-x_{sx} z_M}{f} \\ \\ x_M = b - \dfrac{(x_{dx} - b) z_M}{f} \end{cases} \Rightarrow z_M = \frac{bf}{(x_{dx} - b - x_{sx})}$$

Ciò che viene effettivamente misurato sono le coordinate delle proiezioni sui sensori, rispetto a riferimenti propri di ciascuna fotocamera (figura 1.2). Rispetto a tali sistemi di coordinate le ascisse sono rispettivamente $u_{sx} = x_{sx} + W/2$ per il sensore di sinistra e $u_{dx} = x_{dx} - b + W/2$ per il sensore di destra. Quindi definendo la disparità d come la differenza tra questi due valori

$$d = u_{dx} - u_{sx} = (x_{dx} - b - x_{sx})$$

la relazione per triangolare la distanza diventa:

$$z_M = \frac{bf}{d} \tag{1.1}$$

con b e f parametri geometrici noti e costanti per una stessa coppia di immagini, e d da valutare per ciascuna coppia di punti corrispondenti (vale a

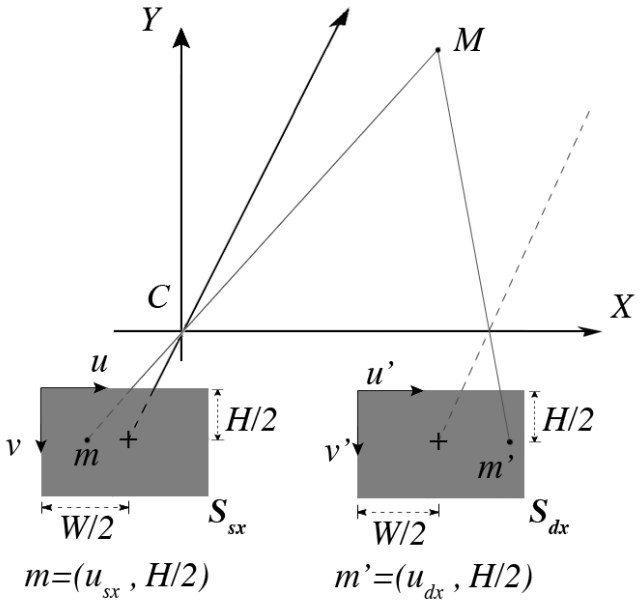


Fig. 1.2: rappresentazione in tre dimensioni del sistema con fotocamere in posizione canonica.

dire: punti che sono le proiezioni di uno stesso punto dello spazio). Il calcolo delle corrispondenze costituisce il problema cruciale della creazione di una mappa che associ una disparità a ciascun pixel, e sarà discusso in seguito.

Si noti che con il posizionamento dei sensori ipotizzato la variazione d è positiva, essendo $u_{dx} > u_{sx}$, ma che per effetto del capovolgimento dell'immagine rispetto ai centri di simmetria C e C' lo spostamento della proiezione di M avviene in direzione opposta, se si osserva l'immagine correttamente orientata (quindi la proiezione di uno stesso punto si trova più sinistra nell' immagine di destra).

Va inoltre osservato che la trattazione bidimensionale del problema (si è supposto che M sia la proiezione su XZ di un generico punto dello spazio) non comporta un'ulteriore riduzione della genericità dell'impostazione. Infatti un punto dello spazio con ordinata $y_M \neq 0$ si proietta in punti dei sensori aventi entrambi ordinata $-fy_M/z_M$, mentre per l'ascissa continuano a valere le relazioni già trovate. In generale, per un sistema con coordinate (u, v) del piano proiettivo centrate sull'asse ottico vale (in coordinate omogenee):

$$z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Rightarrow z\boldsymbol{m} = P\boldsymbol{M} \tag{1.2}$$

1.2 Triangolazione: caso generale

Le relazioni di proporzionalità che descrivono la proiezione di un punto sui due sensori possono essere messe a sistema anche nel caso più generale, quello di un qualsiasi posizionamento e orientamento reciproci tra le fotocamere[1].

Come già detto, le immagini acquisite dalla coppia di fotocamere forni-

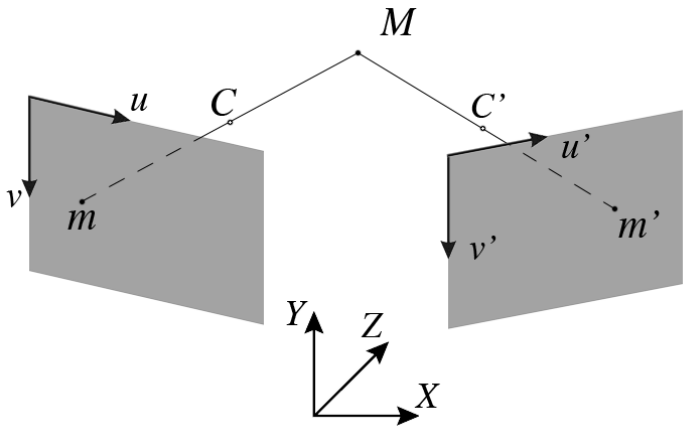


Fig. 1.3: proiezione di un punto su due sensori, in assenza di ipotesi sulla loro posizione reciproca.

scono le coordinate della proiezione di un punto relativamente al riferimento proprio di ciascun dispositivo (rispettivamente (u, v) e (u', v')). Per risolvere il sistema nelle incognite x, y e z è invece necessario scrivere le equazioni rispetto a uno stesso sistema di coordinate, e deve quindi essere nota la posizione assoluta dei due dispositivi rispetto a un riferimento fisso. Ciò è equivalente a conoscere il vettore di traslazione $\boldsymbol{t}$ (e $\boldsymbol{t'}$) e la matrice di rotazione R (e R') che descrivono la trasformazione dalle coordinate del sistema assoluto ($\boldsymbol{M}$) a quelle relative alla fotocamera ($\boldsymbol{M_c}$):

$$\boldsymbol{M_c} = G\boldsymbol{M} = \begin{bmatrix} R & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \boldsymbol{M}$$

I sei parametri (tre per R e tre per $\boldsymbol{t}$) che definiscono la matrice G sono detti *parametri estrinseci*.

Per ottenere le coordinate $\boldsymbol{m}$ (e $\boldsymbol{m'}$) dalle coordinate di $\boldsymbol{M_c}$, si deve conoscere la matrice che descrive la proiezione sul sensore di un punto dello spazio. Tale matrice costituisce una generalizzazione della P presente in

[1] L'impostazione di questa sezione si rifà al capitolo 4 di Fusiello A. *Visione Computazionale*, 2009.

(1.2). Dovendo tenere conto della traslazione rispetto al centro del sensore e della riscalatura degli assi (da una misura in metri si passa a una misura in pixel), si ricava una trasformazione che può essere scritta in forma scalare:

$$\begin{cases} u = k_u \dfrac{-f}{z} x + u_0 \\ \\ v = k_v \dfrac{-f}{z} y + v_0 \end{cases}$$

oppure in forma matriciale, in coordinate omogenee, come:

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -fk_u & 0 & u_0 & 0 \\ 0 & -fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \tag{1.3}$$

e raccogliendo la matrice K dei *parametri intrinseci* (così sono detti i due parametri di traslazione u_0 e v_0 e i due parametri di scala k_u e k_v) costituita dalle prime tre colonne, si ottiene la fattorizzazione $K[I|\mathbf{0}]$ e un'espressione più sintetica della trasformazione:

$$z_c \boldsymbol{m} = K[I|\mathbf{0}]\boldsymbol{M_c}$$

Definendo la matrice di proiezione proiettiva come

$$P = K[I|\mathbf{0}]G$$

la trasformazione complessivamente descritta è:

$$z_c \boldsymbol{m} = K[I|\mathbf{0}]\boldsymbol{M_c} = K[I|\mathbf{0}]G\boldsymbol{M} = P\boldsymbol{M}$$

dove il fattore z_c può essere omesso se all'uguaglianza si sostituisce la relazione, più debole, di uguaglianza a meno di un fattore di scala (indicata con $\simeq$):

$$\boldsymbol{m} \simeq P\boldsymbol{M}$$

Supponendo ora che siano note tali matrici P e P' per entrambe le fotocamere, e mettendone in evidenza i vettori riga, si ha:

$$\boldsymbol{m} \simeq P\boldsymbol{M} = \begin{bmatrix} \boldsymbol{p}_1^T \\ \boldsymbol{p}_2^T \\ \boldsymbol{p}_3^T \end{bmatrix} \boldsymbol{M} \qquad \boldsymbol{m'} \simeq P'\boldsymbol{M} = \begin{bmatrix} \boldsymbol{p'}_1^T \\ \boldsymbol{p'}_2^T \\ \boldsymbol{p'}_3^T \end{bmatrix} \boldsymbol{M}$$

Ciascuna di queste equazioni matriciali fornisce una coppia di equazioni, che assieme permettono di scrivere il sistema che ha per incognite le componenti di $\boldsymbol{M}$:

$$\begin{bmatrix} (\boldsymbol{p}_1 - u\boldsymbol{p}_3)^T \\ (\boldsymbol{p}_2 - u\boldsymbol{p}_3)^T \\ (\boldsymbol{p'}_1 - u'\boldsymbol{p'}_3)^T \\ (\boldsymbol{p'}_2 - u'\boldsymbol{p'}_3)^T \end{bmatrix} \boldsymbol{M} = \boldsymbol{0}$$

La relazione tra le coordinate ottenute e la disparità è data dalla distanza euclidea:

$$d(u, v) = \sqrt{(u - u')^2 + (v - v')^2}$$

La procedura appena illustrata non corrisponde a un concreto algoritmo di calcolo delle disparità, ma è utile a illustrare il significato dei parametri menzionati. L'effettivo metodo di calcolo infatti necessita della conoscenza di questi valori e sarà descritto successivamente.

Sia i parametri intrinseci che quelli estrinseci (oltre a quelli di distorsione) possono essere stimati applicando algoritmi di calibrazione[2], che si basano sull'acquisizione di immagini di oggetti di geometria nota[3].

1.3 Geometria epipolare

Il calcolo della profondità di un punto a partire da immagini rettificate (cioè portate a una forma che verifichi le ipotesi del caso ideale) si basa sulla conoscenza della disparità delle ascisse (o, più in generale, la conoscenza delle coordinate) delle due proiezioni. Tutto ciò presuppone che si sia in grado di mettere in corrispondenza le proiezioni di uno stesso punto. Risulta quindi conveniente conoscere la relazione tra le coordinate della proiezione m di un punto M secondo il centro C sul sensore S_{sx}, e il luogo geometrico dei punti che possono essere l'immagine m' dello stesso punto secondo il centro C' sul sensore S_{dx}.

Facendo riferimento alla figura 1.4, si può osservare che m può essere proiezione su S_{sx} di uno qualsiasi degli $M_1, M_2, ...$ che appartengono alla retta r, raggio ottico passante per m e C. I possibili $m'_1, m'_2, ...$ corrispondenti a m si trovano dunque lungo la retta r' (detta *linea epipolare*), proiezione sul

[2] Jean-Yves Bouguet, *Camera Calibration Toolbox for Matlab* http://www.vision.caltech.edu/bouguetj/calib_doc/.

[3] Janne Heikkilä e Olli Silvén *A Four-step Camera Calibration Procedure with Implicit Image Correction* 1997 Conference on Computer Vision and Pattern Recognition.

piano di S_{dx} secondo il centro C' della retta r. In modo equivalente si può dire che r' è l'intersezione del piano individuato dai punti m, C e C' (detto *piano epipolare*) con il piano del sensore S_{dx}. La retta per C e C' (*baseline*) è comune a tutti i piani epipolari, e le sue intersezioni e ed e' con i piani dei sensori sono dette *epipoli*. Le linee epipolari di uno stesso piano costituiscono dunque un fascio di rette avente per centro l'epipolo.

Un'identica costruzione si può realizzare a partire da un punto sul sensore

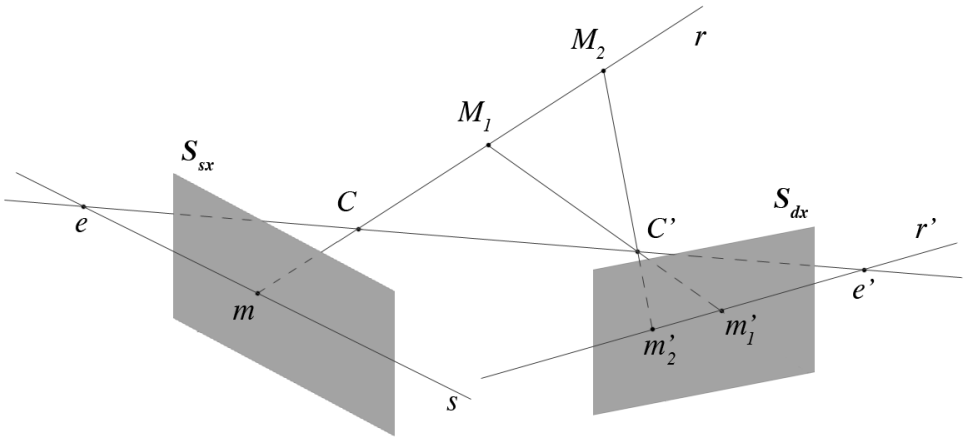


Fig. 1.4: geometria epipolare.

S_{dx}, cercando le possibili corrispondenze su S_{sx}. Per esempio, nella figura 1.4, la retta s è la linea epipolare associata ai punti m'_1 e m'_2. Possibili andamenti delle linee epipolari sono illustrati in figura 1.5. Un caso utile per la ricerca delle corrispondenze è quello in cui il fascio delle linee epipolari degenera, per i piani di entrambi i sensori, in un fascio di rette parallele. Ciò accade quando la *baseline* non interseca i piani delle immagini (caso c di figura 1.5).

Posizionando inoltre i sensori in modo tale che siano coplanari con scanline collineari, e che le linee epipolari siano parallele a una delle due dimensioni, si ottiene una situazione particolarmente vantaggiosa in termini operativi: la ricerca del punto corrispondente a $m(u_m, v_m)$ su un sensore può essere limitata ai soli punti di pari ordinata v_m sull'altro sensore. Si tratta della situazione descritta all'inizio del capitolo (figura 1.1).

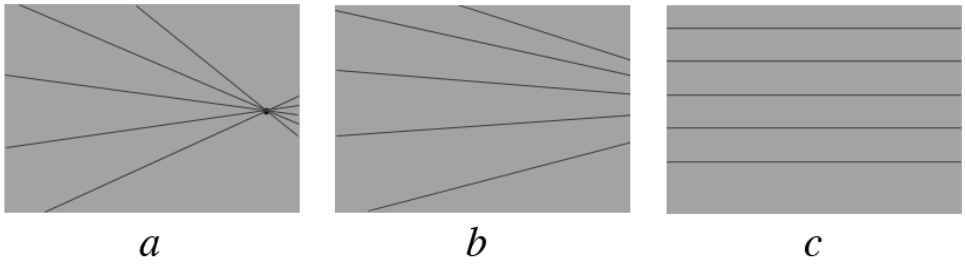


Fig. 1.5: possibili andamenti delle linee epipolari: a) epipolo interno all'area del sensore b) linee convergenti verso un epipolo esterno al sensore c) linee parallele, epipolo collocato all'infinito.

1.4 Risoluzione lungo l'asse Z

La relazione (1.1) mette in evidenza la dipendenza di z_M dal rapporto b/d. Dal punto di vista analitico, nella situazione rappresentata in figura 1.1, questo rapporto è costante, perlomeno fintanto che z_M e f rimangono proporzionali, e a maggior ragione se sono costanti. In pratica però la variazione di b comporta un cambiamento nella risoluzione lungo l'asse Z. Ciò accade perché z_M è inversamente proporzionale a una grandezza discreta – d si misura in pixel – e assume quindi valori distribuiti in modo non lineare in un insieme finito. Posto che le disparità presenti in una coppia di immagini appartengano all' intervallo $[d_{min}, d_{max}]$, allora la risoluzione lungo l'asse Z sarà massima in prossimità di d_{max} e minima vicino a d_{min} (figura 1.6).

Detta Δd la minima variazione di disparità, al valore di disparità immediatamente successivo a d corrisponde una profondità

$$z - \Delta z = \frac{bf}{d + \Delta d}$$

che messa a sistema con la (1.1) porta all'espressione della risoluzione Δz:

$$\Delta z = \frac{z^2 \Delta d}{bf + z\Delta d}$$

la quale per $fb/z >> \Delta d$, ossia per z non eccessivamente grande, può essere così semplificata:

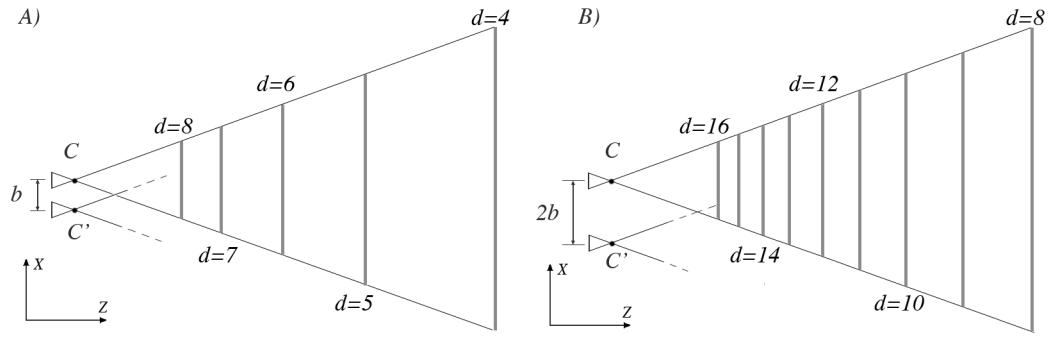


Fig. 1.6: distribuzione delle superfici con Z costante e valori di disparità corrispondenti: A) nel caso di una *baseline* pari a b; B) nel caso di una *baseline* pari a $2b$. La lunghezza focale non è in scala

$$\Delta z = \frac{z^2 \Delta d}{bf} \tag{1.4}$$

La risoluzione può quindi essere migliorata aumentando la *baseline*. Ma è una operazione che presenta controindicazioni, poichè tende a ridurre le coppie di punti in corrispondenza tra loro, i soli per i quali è possibile calcolare la disparità e quindi la profondità. Questo accade per diverse ragioni: perché l'area sovrapponibile delle due immagini diminuisce; perché caratteristiche particolari della scena possono creare occlusioni (figura 1.7); una baseline ampia aumenta la distorsione prospettica di un'immagine relativamente all'altra, con conseguente diminuzione della somiglianza tra aree corrispondenti. Come si vedrà in seguito, un miglioramento della risoluzione è possibile operando un'interpolazione in fase di calcolo delle disparità.

Le considerazioni svolte in questo capitolo sono relative a un sistema di acquisizione ideale, le cui caratteristiche si discostano da quelle dei dispositivi reali. In particolare, con riferimento alle figure 1.1 e 1.2, si è ipotizzato: una proiezione secondo il modello stenopeico; il perfetto parallelismo tra gli assi ottici; la perfetta coplanarità dei sensori; l'allineamento tra le linee orizzontali dei sensori.

Nel seguito verranno descritte le caratteristiche dei sistemi reali, ma va sottolineato che le semplificazioni fin qui adottate non hanno valore puramente teorico o esplicativo. Al contrario: il funzionamento degli algoritmi di calcolo delle corrispondenze vincola sostanzialmente alle stesse ipotesi le coppie di immagini che costituiscono il loro input. La caratterizzazione quantitativa

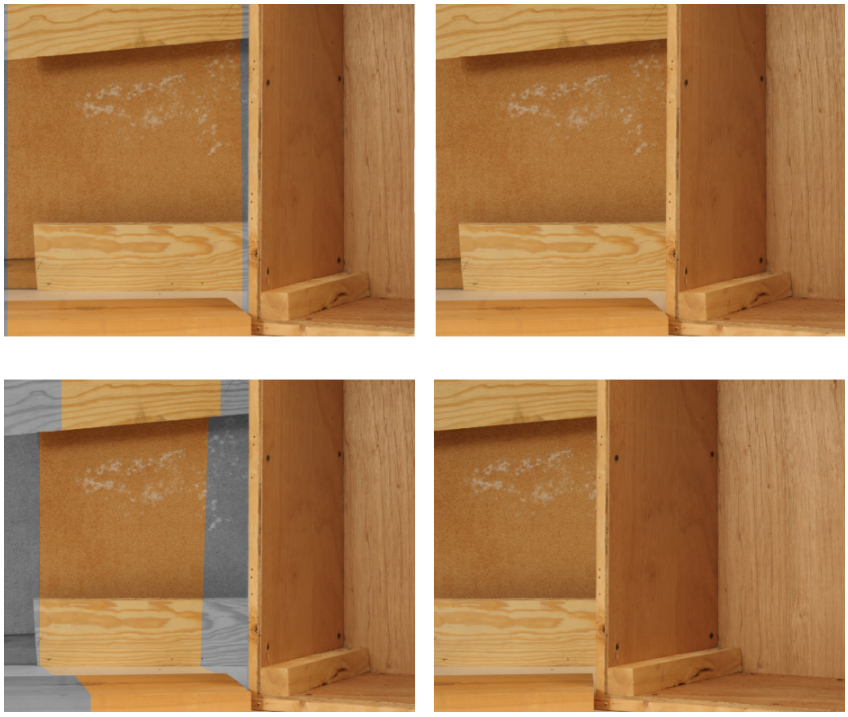


Fig. 1.7: In alto: coppia di immagini ottenute con una baseline corta. Nell'immagine di sinistra sono evidenziate (in bianco e nero) le superfici che non compaiono nell'immagine di destra. L'area grigia più a sinistra è esterna all'inquadratura della fotocamera di destra. L'area grigia al centro invece non è visibile nell'immagine di destra perché occlusa dalla particolare geometria della scena.In basso: coppia di immagini della stessa scena, ottenute però con una *baseline* più lunga rispetto alla figura precedente. Le aree non viste dalla fotocamera di destra si allargano di conseguenza (elaborazioni di immagini tratte da http://vision.middlebury.edu/stereo/data/).

delle apparecchiature realmente utilizzate ha proprio lo scopo di permettere un'elaborazione preventiva delle immagini, che le porti alla forma desiderata.

Capitolo 2

Caratteristiche reali del sistema di acquisizione

Gli scostamenti dall'idealità dipendono principalmente dalle ottiche fotografiche utilizzate, da imperfezioni connaturate alle fotocamere fin dal loro assemblaggio e da imprecisioni nell'allestimento del sistema.

2.1 Distorsioni

A differenza del modello stenopeico le lenti di un obiettivo fotografico introducono una distorsione che comporta uno spostamento dei punti in direzione radiale (distorsione radiale, o *barrel distorsion*), che è funzione della distanza r dall'intersezione tra piano focale e asse ottico (che in prima approssimazione coincide con il centro dell'immagine).

Lo spostamento è descritto dalle relazioni[1]:

$$\begin{cases} x_{corretta} = x_{distorta}(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ \\ y_{corretta} = y_{distorta}(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{cases}$$

ed è esemplificato in figura 2.1. In genere, salvo il caso di focali estremamente corte (tipo *fish-eye*) le distorsioni sono contenute, e il parametro k_3 può essere trascurato.

L'altra principale fonte di distorsione è costituita dall'inclinazione del

[1] Janne Heikkilä e Olli Silvén *A Four-step Camera Calibration Procedure with Implicit Image Correction* 1997 Conference on Computer Vision and Pattern Recognition.

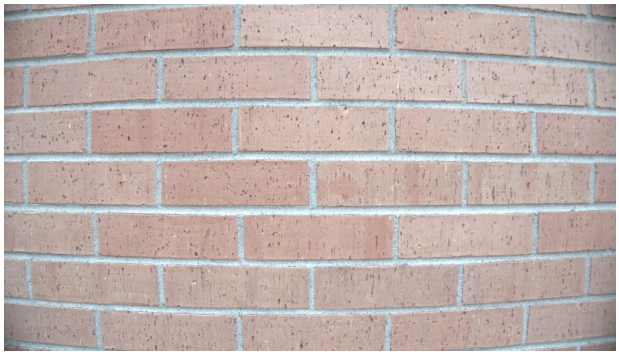


Fig. 2.1: per effetto della distorsione radiale si ottiene una curvatura delle linee rette, tanto più accentuata quanto più ci si allontana dal centro dell'immagine. In particolare le linee orizzontali non sono più parallele ai margini superiore e inferiore.

sensore rispetto al piano ortogonale all'asse ottico (figura 2.2). Ciò è dovuto all'imprecisione dell'assemblaggio del sensore nella fotocamera, soprattutto nei dispositivi economici.

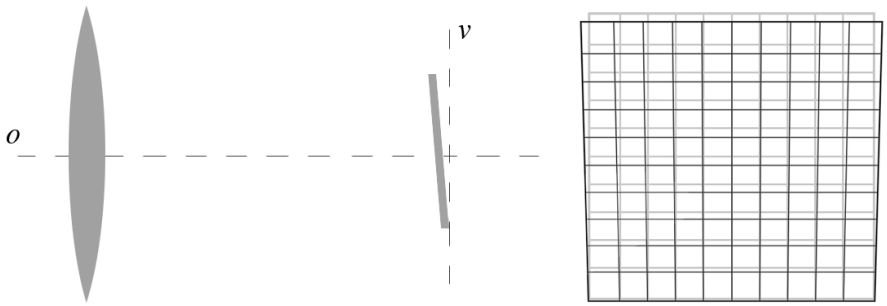


Fig. 2.2: l'inclinazione del sensore è causa della distorsione tangenziale, che corrisponde alla differenza tra la proiezione sul piano focale (griglia grigia) e la proiezione sul piano del sensore (griglia nera).

Le relazioni tra coordinate corrette e distorte sono in questo caso:

$$\begin{cases} x_{corretta} = x_{dist} + [2p_1 y_{dist} + p_2(r^2 + 2x_{dist}^2)] \\ \\ y_{corretta} = y_{dist} + [p_1(r^2 + 2y_{dist}^2) + 2p_2 x_{dist}] \end{cases}$$

che aggiungono i due coefficienti p_1 e p_2 ai tre associati alla distorsione radiale (k_1, k_2 e k_3). Anche questi cinque parametri, come i quattro che compaiono nella (1.3), sono detti intrinseci, poiché caratterizzano il comportamento della fotocamera.

2.2 Calibrazione

Quindi la proiezione di un punto dello spazio (le cui coordinate sono riferite ad un sistema fisso rispetto alla scena) sulla superficie di un sensore di un dato dispositivo fotografico, è descritta dai 15 parametri qui di seguito ricapitolati[2]:

- 6 parametri estrinseci che descrivono il posizionamento e l'orientamento della fotocamera rispetto al sistema di coordinate della scena. A loro volta essi possono essere distinti in:
  - 3 componenti di un vettore di traslazione;
  - 3 rotazioni rispetto agli assi;
- 9 parametri intrinseci che caratterizzano la fotocamera, e dunque la proiezione sul sensore di un punto rappresentato in un sistema di coordinate solidale ad essa. Si possono ulteriormente suddividere in:
  - 4 coefficienti di traslazione e conversione:
    * 2 componenti per specificare la traslazione dell'origine delle coordinate dal centro ottico a un vertice del sensore;
    * 2 fattori di scala (espressi in $pixel/m$) per passare da un'unità di misura in metri a una in pixel;
  - 5 parametri di distorsione:
    * 3 coefficienti di distorsione radiale;
    * 2 coefficienti di distorsione tangenziale;

Tutti i parametri vengono indirettamente misurati tramite la calibrazione, operazione per la quale esistono apposite funzioni sia per Matlab che nella libreria di OpenCV.

In pratica si devono fotografare, con il dispositivo da calibrare, più vedute

[2]L'uso di cinque parametri di distorsione si rifà a Bradski G. e Kaehler A. *Learning OpenCV* O'Reilly p.376. In generale il numero di parametri dipende dal modello adottato per descrivere la distorsione.

differenti di uno stesso oggetto – tipicamente una scacchiera – che presenti caratteristiche tali da agevolare l'identificazione di punti sulla sua superficie (nel caso della scacchiera i punti considerati sono gli angoli dei quadri interni al perimetro).

A meno di un fattore di scala, le informazioni che descrivono la tra-

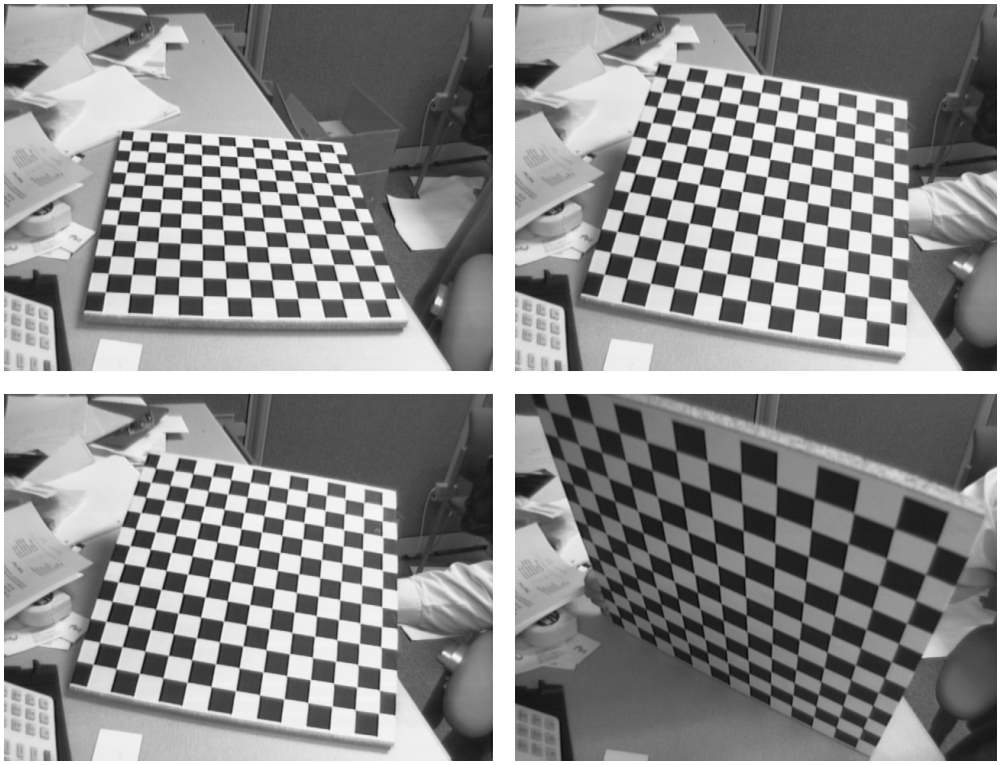


Fig. 2.3: esempi di immagini utilizzate per calibrare una fotocamera (tratte da http://www.vision.caltech.edu/bouguetj/calib_doc/).

sformazione di un punto $\boldsymbol{m_{sc}}$ sul piano della scacchiera nel punto $\boldsymbol{m_{sens}}$ sul piano del sensore, sono deducibili dalle proiezioni di quattro punti. Tale trasformazione tra piani costituisce infatti una omografia, matematicamente rappresentata in coordinate omogenee da una matrice H, di dimensioni 3x3:

$$\boldsymbol{m_{sens}} = H\boldsymbol{m_{sc}} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \boldsymbol{m_{sc}}$$

Scontando il fattore di scala la matrice può essere normalizzata per avere $h_{33} = 1$. In tal modo le otto equazioni relative alle coordinate dei quattro punti costituiscono un sistema compatibile negli otto h_{ij} incogniti. Di conseguenza, un numero maggiore di punti non porta a un aumento del numero di

equazioni linearmente indipendenti (ma è comunque utile in quanto migliora la stabilità numerica e riduce l'effetto del rumore). Quindi ciascuna immagine permette di impostare 8 equazioni, una per ogni coordinata dei quattro punti. C'è da tenere presente che i sei parametri estrinseci variano ad ogni veduta, aumentando così il numero di incognite. Per la stessa ragione – il fatto che essi descrivono la posizione relativa tra fotocamera e scacchiera – il calcolo dei parametri estrinseci è possibile solo se è nota con precisione la posizione della scacchiera rispetto a un riferimento fisso. In caso contrario si ottengono valori significativi solo per i parametri intrinseci.

Considerando i parametri estrinseci e i primi quattro parametri intrinseci (e tralasciando momentaeamente quelli distorsivi, che descrivono una trasformazione bidimensionale), per K immagini si ha un totale di $4 + 6K$ incognite. Per potere risolvere il sistema deve essere $8K \geq 4 + 6K$, ovvero $K \geq 2$. Sono quindi necessarie almeno due vedute di una scacchiera 3x3, ma per le già citate ragioni di robustezza e stabilità è opportuno sovravincolare il problema e ottenere la soluzione che minimizzi l'errore quadratico. Per quel che riguarda i 5 parametri distorsivi, essi possono essere calcolati una volta che siano state rilevate le coordinate distorte di almeno 3 punti, e calcolate le relative coordinate non distorte.

Quanto detto fin qui è riferito ai parametri e alla calibrazione di un singolo dispositivo. Per sistemi di acquisizione stereo la funzione `cvStereoCalibrate()` di OpenCV permette la calibrazione congiunta delle due fotocamere, e restituisce parametri che hanno lo stesso significato di quelli appena visti per il caso di una singola fotocamera.

Una volta compiuta questa operazione è possibile utilizzarne i risultati

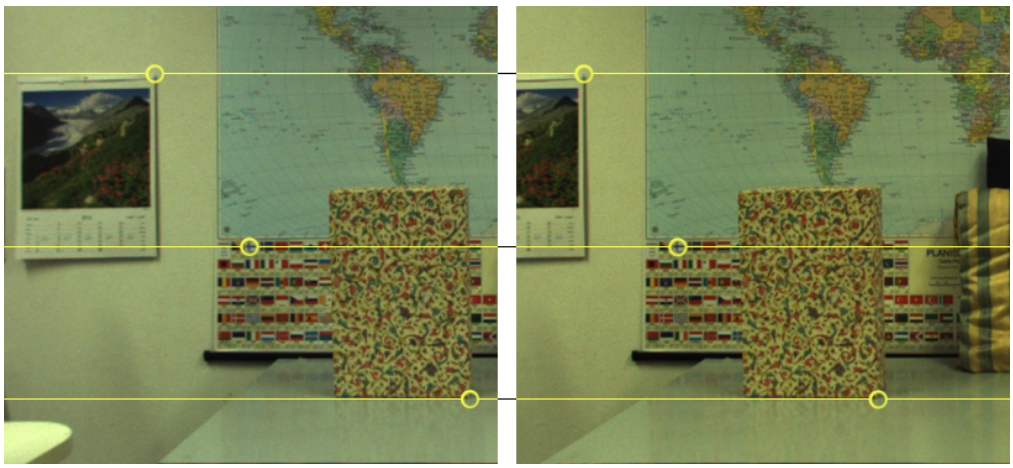


Fig. 2.4: coppia di immagini rettificate: pixel che sono la proiezione di uno stesso punto si trovano lungo la stessa linea orizzontale.

per trasformare le immagini e portarle a verificare le condizioni auspicate in precedenza, discutendo la geometria epipolare. Tutto ciò è realizzabile – anche seguendo procedure diverse – con apposite funzioni di OpenCV: `cvStereoRectifyUncalibrated()`, `cvStereoRectify()` e `cvInitUndistortRectifyMap()`, tra le altre.

Dal punto di vista geometrico questo corrisponde alla rimozione delle distorsioni – garantendo così che le linee epipolari siano linee rette – seguita dalla rettificazione. Quest'ultima operazione consiste nel fare in modo che i piani corrispondenti alle due immagini coincidano, correggendo in tal modo l'orientamento degli assi ottici (la massima accortezza in fase di messa a punto del sistema non basta a garantire il parallelismo tra gli assi) e un'eventuale differenza nelle lunghezze focali. In aggiunta si ottiene un perfetto allineamento delle righe delle due immagini (figura 2.4), in modo che le proiezioni di uno stesso punto abbiano la stessa ordinata (come il parallelismo degli assi, anche questa condizione non può essere garantita da regolazioni manuali).

Da qui in avanti si supporrà che le ipotesi del caso ideale siano sempre verificate. Oltre ai vincoli imposti alle immagini utilizzate per lo stereo matching, sussistono ipotesi implicite anche per la scena. In particolare si supporranno validi i vincoli di unicità e di ordinamento[3]. Il primo stabilisce

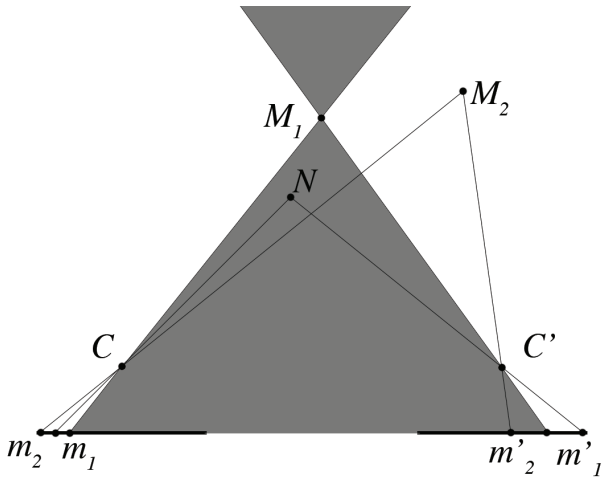


Fig. 2.5: le proiezioni di $\boldsymbol{M_1}$ e $\boldsymbol{M_2}$ sono collocate nello stesso ordine nelle due immagini. $\boldsymbol{N}$ si trova invece nella zona proibita (area grigia) rispetto a $\boldsymbol{M_1}$ e vìola il vincolo di ordinamento.

che una coppia di punti nelle due immagini possano corrispondere al massimo a un punto dello spazio tridimensionale. Questa condizione è sempre

[3]Bogusław Cyganek e J. Paul Siebert *An Introduction to 3D Computer Vision Techniques and Algorithm* p. 66.

valida per superfici opache, ma può essere violata in presenza di superfici trasparenti o riflettenti, che possono moltiplicare le immagini di un punto. La seconda condizione impone che procedendo lungo la scanline i punti si presentino nello stesso ordine nelle due immagini. Questo corrisponde a non collocare oggetti nella zona proibita, delimitata, per ciascun punto $\boldsymbol{M}$, dai raggi ottici passanti per $\boldsymbol{M}$ (figura 2.5).

Capitolo 3

Algoritmi di calcolo della mappa di disparità

Una descrizione dei principali aspetti comuni agli algoritmi di calcolo della mappa di disparità, e di alcune caratterisitche particolari dei metodi attinenti allo *space-time stereo*, risulteranno utili a contestualizzare scopi e metodi di quest'ultimo.

3.1 Mappe di disparità

Data una coppia di immagini $I_{sx}(u, v)$ e $I_{dx}(u, v)$ con $u \in [0, W-1]$ e $v \in [0, H-1]$, la mappa di disparità $d(u, v) \geq 0$ rispetto a I_{sx} è definita in modo tale che sia:

$$I_{sx}(u, v) \equiv I_{dx}(u - d(u, v), v) \tag{3.1}$$

dove $I_{sx}(u, v)$ e $I_{dx}(u, v)$ sono valori di luminosità, e il significato operativo del simbolo $\equiv$ dipende dal criterio di corrispondenza adottato.

Formalmente il ruolo delle due immagini è simmetrico, ed è possibile invertirne i ruoli e definire la mappa $d'(u, v) \geq 0$ rispetto a I_{dx}:

$$I_{sx}(u + d'(u, v), v) \equiv I_{dx}(u, v)$$

Considerando la domanda a cui il calcolo della disparità mira a rispondere, il significato teorico di $\equiv$ si può tradurre con è la proiezione dello stesso punto dello spazio. La situazione è illustrata in figura 14. La convenzione comunemente adottata per la rappresentazione grafica delle mappe di disparità è di convertire i valori di d in livelli di grigio. Perciò le aree più chiare corrispondono a disparità maggiori (z minori) e le aree più scure contraddistinguono disparità minori (z maggiori).

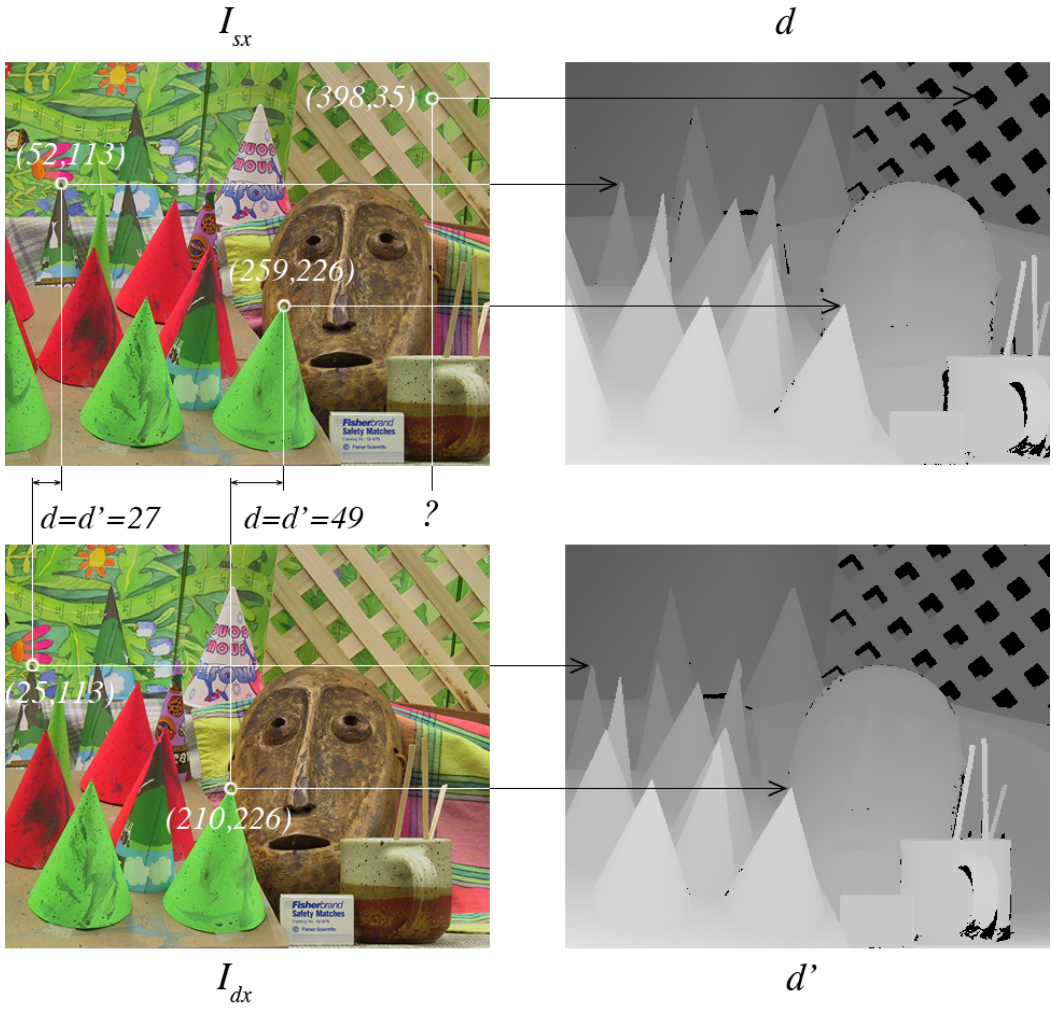


Fig. 3.1: significato della mappa di disparità. A sinistra: il pixel di coordinate $(52, 113)$ nell'immagine I_{sx} corrisponde al pixel di coordinate $(25, 113)$ nell'immagine I_{dx}. Dunque $d(52, 113) = 27$, come risulta dalla differenza delle rispettive ascisse, e si verifica infatti che $I_{sx}(52, 113) \equiv I_{dx}(52 - d(52, 113), 113)$. A destra: la traduzione grafica, in livelli di grigio, delle disparità. Il punto di coordinate $(398, 35)$ nell'immagine I_{sx} non ha controparte nell'immagine I_{dx} e viene classificata come occlusione, e visualizzata in nero nella mappa d (elaborazione di immagini tratte da http://vision.middlebury.edu/stereo/).

Spesso gli algoritmi che calcolano le corrispondenze, procedendo in modo automatico, assegnano una disparità ad ogni pixel. Nell'ipotesi di corretta attribuzione vale:

$$d(u,v) = d'(u - d(u,v), v) \tag{3.2}$$

relazione che, se verificata per ogni (u, v), implicherebbe l'esistenza di una corrispondenza biunivoca tra i pixel delle due immagini. In realtà, come già accennato discutendo gli effetti di un aumento della *baseline*, non esiste tra le immagini tale proprietà, dal momento che alcune parti della scena sono visibili in una sola delle due immagini. Queste aree di un immagine sono dette *occlusioni*. Alla disparità calcolata per punti interni alle occlusioni non è utile attribuire alcun significato: essa è il risultato della ricerca del miglior abbinamento nel caso in cui quello esatto non sia possibile. Il riconoscimento delle occlusioni permette di evitare di attribuire un significato improprio alle relative misure di disparità e di escludere tali zone nella fase di valutazione delle prestazioni di un algoritmo.

Esistono diversi metodi per individuare i pixel occlusi, alcuni dei quali richiedono il calcolo della sola $d(u, v)$ e, ad esempio, il rispetto del vincolo di ordinamento: nella scansione di una riga da sinistra verso destra i pixel devono essere ordinati nello stesso modo dei loro corrispondenti. Dove questo non accade si è in presenza di un'occlusione[1]. Nel seguito del presente scritto si farà sempre riferimento al metodo del *cross-checking*, che consiste nel calcolare sia d che d', verificando poi la validità della proprietà 3.2.

Va notato che in alcuni casi la 3.2 può non essere soddisfatta anche per pixel non geometricamente occlusi. Si tratta in questo caso di corrispondenze errate nella d o nella d' (o in entrambe), e in questo senso il conteggio delle occlusioni è una misura della qualità della mappa.

3.2 Classificazione dei metodi

L'intensa attività di ricerca volta a trovare soluzioni al problema dello *stereo matching* ha prodotto un grande numero di metodi che possono, a un primo esame, essere suddivisi in categorie generali, distinte sulla base di vari criteri, come il tipo di risultato prodotto, o l'approccio adottato nel'uso

[1] B. Cyganek, J. Paul Siebert *An Introduction to 3D Computer Vision Techniques and Algorithms* p. 223.

dell'informazione contenuta nelle immagini[2].

Una possibile classificazione è quella che suddivide tra algoritmi che producono mappe dense e algoritmi che producono mappe sparse. I primi sono quelli implicitamente considerati finora, i quali considerano l'intera immagine e cercano una corrispondenza per ogni pixel: le mappe di disparità generate hanno le stesse dimensioni delle immagini. Una mappa sparsa contiene invece un numero ridotto di disparità e può essere considerata una variante della 3.1:

$$f[I_{sx}(u,v)] \equiv f[I_{dx}(u - d(u,v), v)]$$

dove $f[]$ è un'elaborazione che estrae – dalle immagini che ha per argomento – alcuni tratti salienti (discontinuità, spigoli, vertici). Gli algoritmi sparsi sono più rapidi di quelli densi, ma trovano meno applicazioni, poichè mappe sparse necessitano dell'interpolazione dei valori mancanti.

Nell'ambito dei metodi densi una distinzione fondamentale è quella tra algoritmi locali e algoritmi globali. I nomi si riferiscono all'informazione considerata nel computo del costo delle disparità. I metodi locali valutano solo l'informazione contenuta nel pixel considerato e in quelli nelle vicinanze. Nei metodi globali invece il calcolo, per ogni disparità, tiene conto dell'intera immagine. Questi ultimi danno in genere risultati migliori, ma hanno lo svantaggio di essere di meno semplice implementazione e di comportare un maggiore costo computazionale.

Alternativa alla classificazione in categorie così generiche è la tassonomia proposta – per i metodi che generano mappe dense – da Scharstein e Szeliski[3], basata sull'osservazione che tutti gli algoritmi di questo tipo consistono nell'esecuzione di un sottoinsieme delle seguenti quattro operazioni:

1. calcolo del costo delle corrispondenze;
2. aggregazione del costo sul supporto;
3. calcolo della disparità e ottimizzazione;
4. raffinamento del risultato.

[2]Questa breve panoramica si basa in gran parte sul capitolo 11 di Szeliski R. *Computer Vision: Algorithms and Applications.*

[3]Scharstein D. e Szeliski R. *A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms.*

3.2.1 Calcolo del costo delle corrispondenze

Si tratta di stabilire un criterio quantitativo che costituisca una misura della bontà di una corrispondenza tra pixel. Tale criterio è in pratica una distanza $D()$, da attribuire come costo alla corrispondenza tra il pixel P nell'immagine di sinistra e il pixel Q nell'immagine di destra:

$$C(P,Q) = D[I_{sx}(P), I_{dx}(Q)]$$

Tra le distanze più comunemente usate ci sono la differenza assoluta:

$$C_{AD}(P,Q) = |I_{sx}(P) - I_{dx}(Q)|$$

e la differenza quadratica:

$$C_{SD}(P,Q) = [I_{sx}(P) - I_{dx}(Q)]^2$$

o loro varianti[4]. Per immagini a colori le differenze sono da intendersi tra moduli di vettori nello spazio di colori utilizzato.

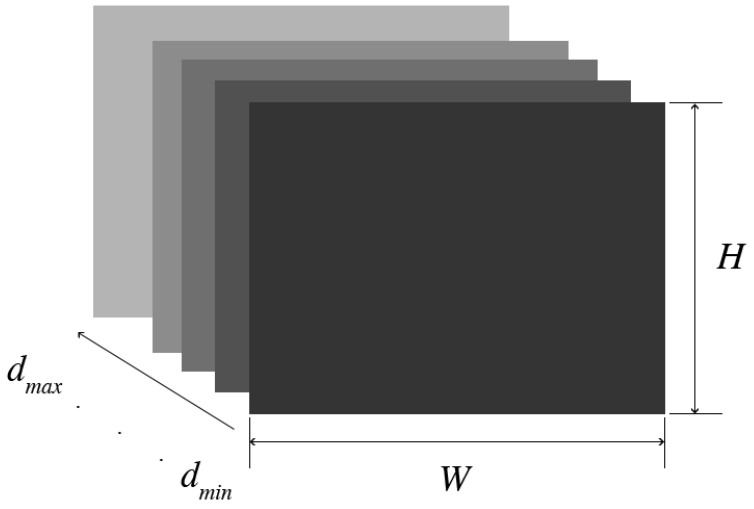


Fig. 3.2: Disparity Space Image. È una matrice tridimensionale contenente i costi $C(u,v,d)$ associati alla corrispondenza tra il pixel (u,v) nell'immagine di sinistra e il pixel $(u-d,v)$ nell'immagine di destra.

[4]Cyganek B. e Siebert J. P. *An Introduction to 3D Computer Vision Techniques and Algorithms.*

Nei termini in cui è stato qui formulato il problema, le precedenti relazioni possono essere riscritte come[5]:

$$C_{AD}(u, v, d) = |I_{sx}(u, v) - I_{dx}(u - d, v)| \tag{3.3}$$

$$C_{SD}(u, v, d) = |I_{sx}(u, v) - I_{dx}(u - d, v)|^2$$

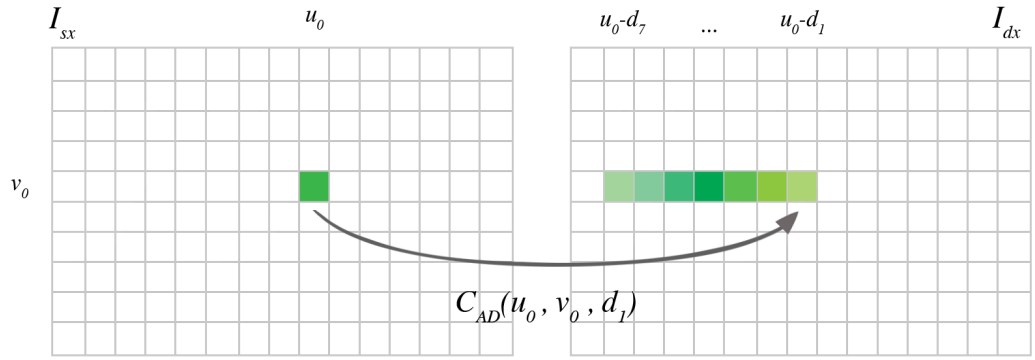


Fig. 3.3: esempio di calcolo dei costi con C_{AD}. Per il pixel di coordinate (u_0, v_0) nell'immagine di sinistra si prendono in esame, nell'immagine di destra, i pixel lungo la stessa scanline spostati orizzontalmente delle quantità interne all'intervallo $[d_{min}, d_{max}]$. Per ciascuno di essi si calcola con la (3.3) la differenza assoluta nello spazio di colori RGB. Ad esempio: $C_{AD}(u_0, v_0, d_1) = |(57, 181, 74) - (172, 212, 115)| = 187$, $C_{AD}(u_0, v_0, d_2) = 113$, $C_{AD}(u_0, v_0, d_3) = 46$ e così via.

Questo tipo di misure di similarità – a differenza di alternative più sofisticate e computazionalmente costose – è sensibile alle differenze di guadagno tra le due immagini, che quindi devono essere preventivamente equalizzate e portate allo stesso valore medio globale, o in alternativa allo stesso valore medio locale, durante l'aggregazione.

[5]Riferendosi alla mappa di disparità, con la lettera d si è inteso un valore di disparità unico per ciascun pixel. Discutendo il calcolo e l'aggregazione dei costi, invece, con d si indicano tutte le possibili disparità prese in esame, interne all'intervallo $[d_{min}, d_{max}]$. Nei contesti in cui sono presenti entrambi i significati d continuerà a indicare un generico valore di disparità nella gamma considerata, e il valore unico che determina la mappa verrà indicato con $\bar{d}$. Laddove non c'è ambiguità si userà semplicemente d.

3.2.2 Aggregazione del costo sul supporto

Questo secondo passo – che caratterizza i metodi locali ed è spesso assente da quelli globali – costituisce un'elaborazione del DSI. Infatti non è in genere sufficiente calcolare il costo di una corrispondenza P-Q tenendo conto dei soli pixel P e Q. Questo perchè l'informazione a essi associata ha bassa capacità discriminatoria, soprattutto in assenza di segnale ad alta frequenza spaziale, consistendo spesso in tre valori quantizzati su 256 livelli, per di più affetti da rumore. Questo rende necessario estendere a un supporto U il calcolo del costo di una corrispondenza.

Ciò può essere messo in atto tramite un filtro passa basso. In particolare un filtraggio del DSI con d costante, che utilizzi le misure di similarità sopra descritte porta alle relazioni (che riuniscono i primi due passi) per la SAD (*Sum of Absolute Differences*):

$$C_{SAD}(u,v,d) = \sum_{(i,j)\in U} |I_{sx}(u+i,v+j) - I_{dx}(u-d+i,v+j)|$$

e la SSD (*Sum of Squared Differences*):

$$C_{SSD}(u,v,d) = \sum_{(i,j)\in U} |I_{sx}(u+i,v+j) - I_{dx}(u-d+i,v+j)|^2$$

entrambe frequentemente usate. Una strategia particolarmente semplice è quella del metodo *fixed windows*, nel quale i supporti U sono quadrati e centrati sul pixel (u,v) correntemente preso in esame. L'uso di supporti bidimensionali, estesi nelle sole dimensioni u e v, privilegia la determinazione delle disparità su superfici frontali (ortogonali agli assi ottici), mentre una generalizzazione che tenga conto delle superfici inclinate richiede un supporto tridimensionale, cioè esteso su più valori di disparità. Per la stessa ragione – l'implicita preferenza per aree a disparità costante – il metodo *fixed windows* non offre buone prestazioni in corrispondenza delle discontinuità, specie all'aumentare delle dimensioni del supporto. In pratica è necessario stabilire un compromesso tra buona risoluzione della mappa (che si ottiene con supporti piccoli) e robustezza nel calcolo dei costi (favorita da supporti grandi).

Un altro inconveniente del metodo *fixed windows* è rappresentato dagli outliers. La causa è il differente spostamento prospettico di pixel che si trovano a profondità diverse (figura 3.4). Una possibile soluzione è quella di limitare il valore massimo del costo a una soglia T (*Truncated Absolute Difference* o TAD):

$$C_{TAD} = \sum_{(i,j)\in U} min\{|I_{sx}(u+i, v+j) - I_{dx}(u-d+i, v+j)|, T\}$$

Fig. 3.4: influenza degli *outliers* sul costo di una corrispondenza. Le due finestre di dimensione 7x7 sono centrate su pixel corrispondenti, ma il cambio di prospettiva ha causato uno spostamento della discontinuità rispetto all'immagine sullo sfondo. La grande differenza tra i pixel delle colonne di destra determina un aumento del costo ed è una potenziale causa di errore (elaborazioni di immagini tratte da http://vision.middlebury.edu/stereo/).

Alternativamente si può ricorrere a filtri di forma diversa (gaussiani, binomiali) che diano maggior peso all'informazione al centro della finestra, ma sopperiscano alla sua eventuale mancanza integrandola con quella presente su un'area più ampia; oppure a filtri che si adattino al contenuto dell'immagine variando la forma e la dimensione del supporto per escludere la discontinuità o posizionandosi in modi diversi rispetto al pixel corrente (*shiftable windows*,

figura 3.5); o ancora si possono sfruttare filtri che pesino il costo di ciascun pixel nel supporto secondo criteri di somiglianza cromatica e distanza euclidea[6].

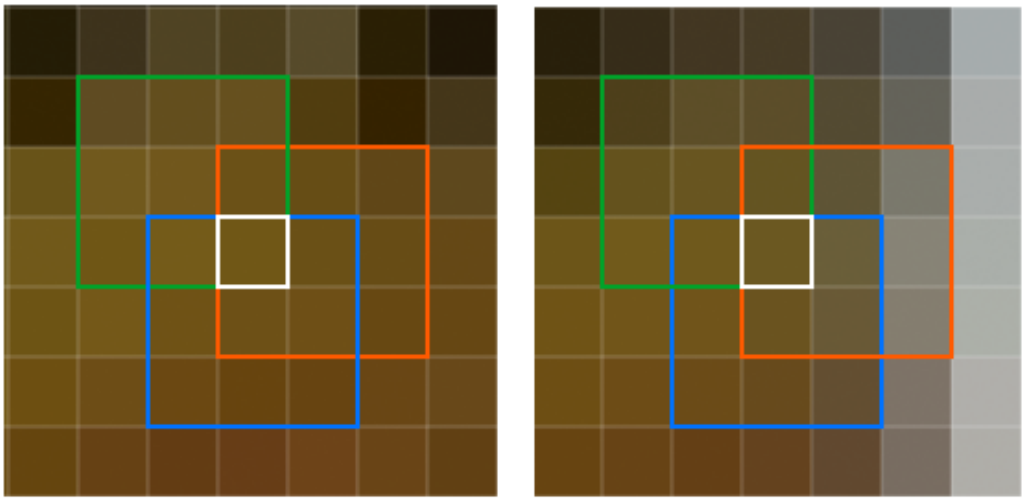


Fig. 3.5: illustrazione del metodo *shiftable windows*, applicato all'esempio di figura 3.4. Consiste nell'aggregare il costo, per ciascuna corrispondenza, su una finestra quadrata di lato N, per tutte le N^2 possibili posizioni relative ai due pixel considerati. Le cornici colorate nella figura rappresentano 3 delle possibili 9 posizioni di una finestra 3x3. La cornice arancione, collocandosi a cavallo della discontinuità nell'immagine di destra, aggrega un costo elevato, poichè include pixel che differiscono molto nelle due immagini. La finestra verde (o le altre finestre spostate a sinistra) aggregano invece un costo basso, e permettono la corretta attribuzione della corrispondenza tra i due pixel nelle cornici bianche.

Filtri più sofisticati conducono a migliori mappe di disparità finali, ma al prezzo di un aumento del costo computazionale. Dette W e H le dimensioni delle immagini, e L l'ampiezza del range delle disparità, algoritmi passa basso come SAD e SSD hanno una complessità $O(WHL)$ e sono molto veloci, mentre una forma diversa del nucleo comporta nel caso più generale una complessità $O(WHLN^2)$, dove N è la dimensione laterale del supporto.

[6]Kuk-Jin Yoon, In So Kweon *Adaptive Support-Weight Approach for Correspondence Search.*

3.2.3 Calcolo della disparità e ottimizzazione

Consiste nel calcolo della mappa di disparità a partire dal DSI. Per i metodi locali un criterio ricorrente è quello *winner takes all*, con scelta della disparità $\bar{d}$ di costo minimo per ciscun pixel (u, v), eventualmente corretta da strategie che permettano di giudicare la bontà di una corrispondenza. Ad esempio[7] stabilendo che non più di due dei tre valori minimi si trovino al di sotto di una certa soglia dipendente dal minimo costo.

Nei metodi globali invece i passi 2 e 3 si fondono in un'unica fase, nella quale si determina il valore di disparità tramite un'ottimizzazione il cui risultato dipende per ciascun pixel dell'intero DSI (che spesso è quella calcolata al passo 1, senza successiva aggregazione).

Alcuni di questi metodi formulano il problema come la minimizzazione di una funzione di energia globale $E(d)$, definita in modo da rispettare sia i dati (utilizzando quindi il DSI) che opportune ipotesi di regolarità delle superfici:

$$E(d) = E_d(d) + \lambda E_s(d)$$

Il primo termine è determinato dal contenuto del DSI:

$$E_d(d) = \sum_{(u,v)} C[u, v, d(u, v)]$$

il secondo riflette l'ipotesi di regolarità, che per ragioni di trattabilità computazionale si limita a tener conto delle variazioni rispetto ai pixel immediatamente adiacenti. Una volta definita l'energia, la minimizzazione può essere calcolata seguendo un qualsiasi metodo di ricerca dei minimi locali.

Spesso l'ipotesi di regolarità è tale da rendere NP-hard il problema di minimizzazione dell'energia. Algoritmi più efficienti si basano sulla programmazione dinamica e comportano il calcolo di una sezione di costo minimo della DSI operando in modo indipendente su scanline diverse. Si distinguono tra loro per la forma della soluzione cercata e il metodo adottato nella ricerca della sezione.

Altri metodi ancora si basano sulla segmentazione delle immagini in aree di disparità uniforme, successivamente interpolate localmente da superfici e riunite in un'unica immagine ricorrendo a strategie di ottimizzazione.

[7] K. Mühlmann, D. Maier, J. Hesser e R. Manner *Calculating dense disparity maps from color stereo images, and efficient implementation*, citato in Bogusław Cyganek e J. Paul Siebert, *An Introduction to 3D Computer Vision Techniques and Algorithms* p. 226.

3.2.4 Raffinamento del risultato

Le disparità sono calcolate come differenze di posizioni, lungo una stessa riga, di pixel in due immagini, e di conseguenza hanno valori interi. Come già osservato la gamma delle disparità è in relazione con la risoluzione lungo l'asse z, in modo tale che quest'ultima peggiora al diminuire della disparità. È possibile ridurre l'impatto della discretizzazione delle disparità sfruttando i costi già calcolati. Utilizzando l'andamento dei costi al variare delle disparità, e interpolandone l'andamento con una funzione continua, si trasforma il problema della ricerca del minimo di una funzione definita in $\mathbb{N}$ in uno analogo per funzione definita in $\mathbb{R}$.

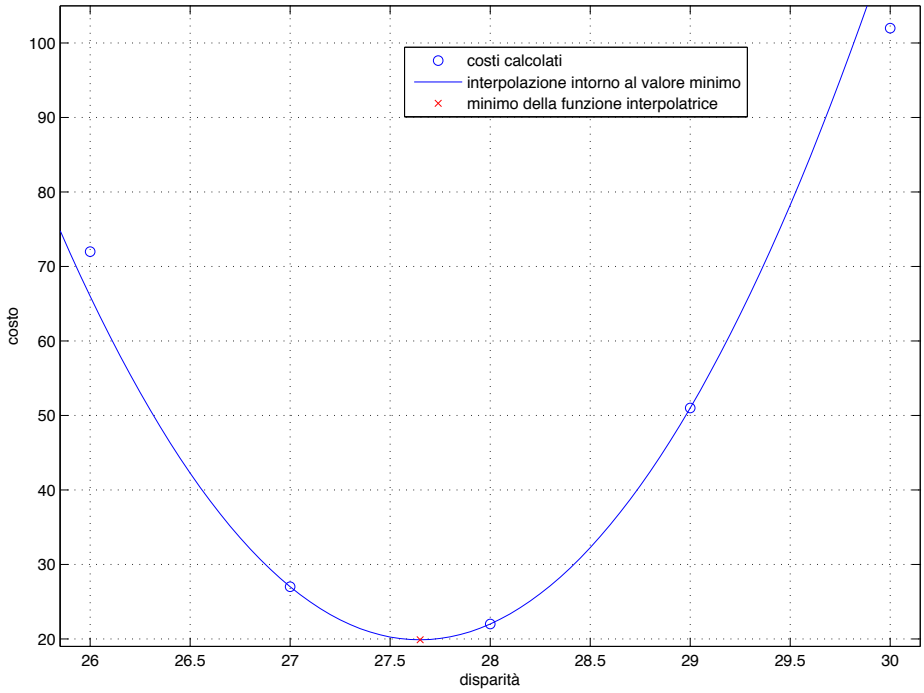


Fig. 3.6: esempio di raffinamento subpixel della disparità, tramite interpolazione parabolica in un intorno del minimo costo.

In teoria è possibile l'interpolazione con qualunque funzione. In pratica un buon compromesso tra l'accuratezza e il costo computazionale si ottiene ricorrendo all'interpolazione con una parabola. Un polinomio di secondo gra-

do ha anche il vantaggio di imporre meno vincoli di regolarità all'andamento dei costi.

Supponendo che per la disparità di costo minimo $\bar{d}$ sia $C(\bar{d}) = c_0$, e che le due disparità adiacenti abbiano costi $C(\bar{d}-1) = c_{-1}$ e $C(\bar{d}+1) = c_{+1}$, allora considerando la parabola $P(x)$ tale che:

$$P(-1) = c_{-1}; \qquad P(0) = c_0; \qquad P(1) = c_{+1};$$

si ottiene il polinomio interpolatore:

$$P(x) = c_{-1}\frac{x(x-1)}{2} - c_0(x+1)(x-1) + c_{+1}\frac{(x+1)x}{2}$$

e imponendo la condizione $P'(x) = 0$ si ricava l'ascissa del minimo:

$$x_m = \frac{c_{-1} - c_{+1}}{2(c_{-1} - 2c_0 + c_{+1})}$$

che è l'addendo correttivo per giungere alla disparità cercata, con precisione subpixel:

$$\tilde{d} = \bar{d} + x_m$$

3.3 Prestazioni del metodo fixed windows

La tecnica dello *space-time stereo* si basa sull'uso di uno dei metodi per lo *stereo matching* binoculare fin qui considerati. In particolare l'algoritmo che sarà descritto nel quarto capitolo sfrutta il metodo *fixed windows*, che è perciò utile osservare più da vicino.
Nel contesto appena descritto questo metodo: rientra nella categoria degli algoritmi locali; può adottatare qualsiasi misura di similarità tra pixel; l'aggregazione avviene su finestre quadrate di dimensione costante e centrate sul pixel correntemente considerato; la disparità viene scelta in base al criterio del minimo costo, con eventuale raffinamento tramite interpolazione con polinomio di secondo grado.

3.3.1 Comportamento con superficie frontale in presenza di segnale alle alte frequenze

Costituisce il caso più favorevole, essendo la superficie piana frontale la situazione geometrica privilegiata dal metodo, oltre che una condizione vantaggiosa in generale. La visione frontale ha infatti il vantaggio di preservare meglio le somiglianze, non introducendo distorsione prospeettica (figura 3.7).

Fig. 3.7: in alto. Coppia di immagini acquisite da un sistema binoculare. In basso. Ingrandimenti delle aree evidenziate. La parte rivolta frontalmente alle fotocamere risulta quasi inalterata, mentre la veduta di scorcio nella parte di sinistra è causa di una sensibile distorsione dovuta al cambio di prospettiva (elaborazione di immagini tratte da http://vision.middlebury.edu/stereo/).

Anche la presenza di componenti ad alta frequenza spaziale è una condizione favorevole, dal momento che una bassa variabilità della luminosità nello spazio ha l'effetto di uniformare i costi delle corrispondenze, rendendo ambigua la scelta di quella ottima. Di seguito i risultati di alcune prove, per l'area evidenziata in figura 3.8.

Fig. 3.8: area considerata nella sezione 3.3.1

Risultati senza aggregazione dei costi

I risultati sono in questo caso scadenti dal punto di vista della qualità della mappa di disparità e del numero di pixel per i quali non vale la (3.2) (e che dunque vengono classificati come occlusi, pur non essendolo di fatto). Tuttavia un'analisi della sezione del DSI (figura 3.10) in corrispondenza della scanline centrale dell'area considerata, mostra un netto avvallamento dei costi in corrispondenza della disparità corretta (tra 25 e 30). Si osserva però una notevole irregolarità dell'andamento dei costi che spiega l'imprecisa attribuzione delle disparità.

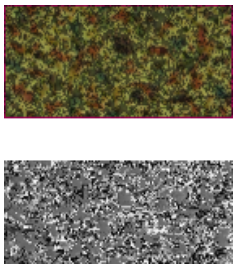


Fig. 3.9: Sopra: area esaminata, con ombreggiatura dei pixel per i quali non vale la (3.2). Sotto: mappa di disparità.

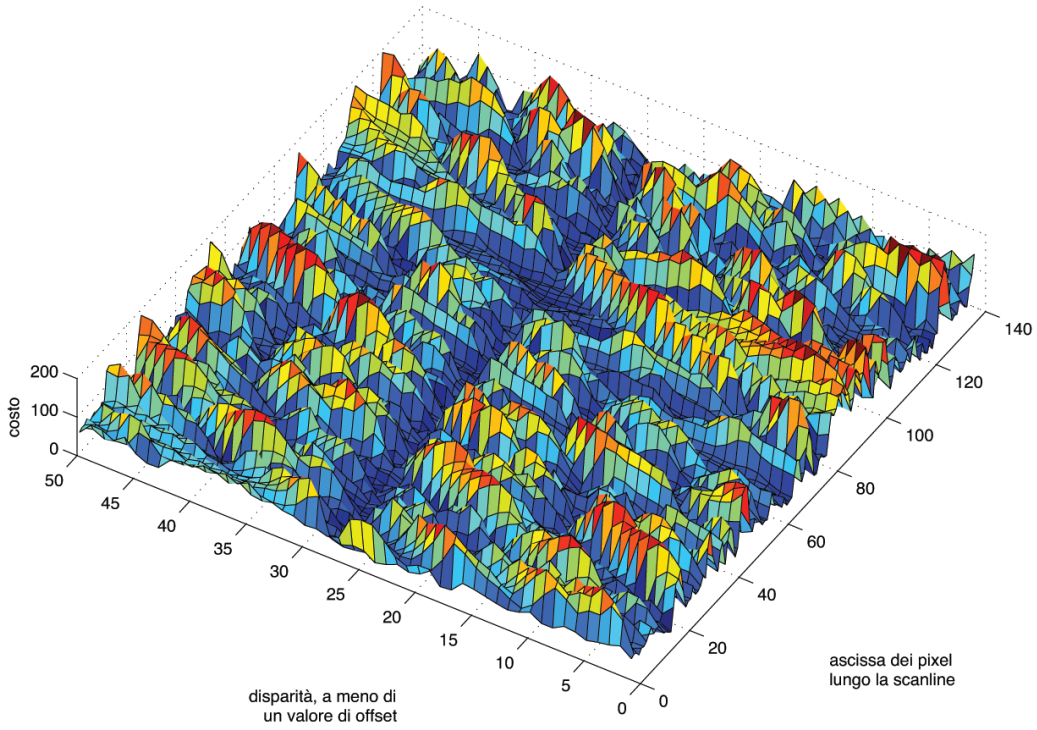


Fig. 3.10: andamento dei costi in una sezione del DSI corrispondente alla scanline centrale dell'area considerata.

Risultati dopo aggregazione su una finestra 7x7

Aggregando i costi si osserva che la qualità della mappa migliora rapidamente all'aumentare della dimensione del supporto, e che per una finestra quadrata di dimensione 7x7 la (3.2) vale per ogni pixel, e la disparità è uniforme. L'andamento dei costi osservati nella sezione del DSI si è fatto più regolare rendendo più robusto il criterio di scelta del minimo.

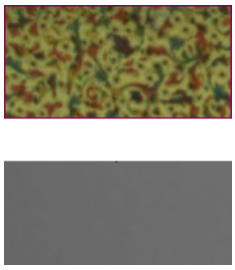


Fig. 3.11: area esaminata e mappa di disparità.

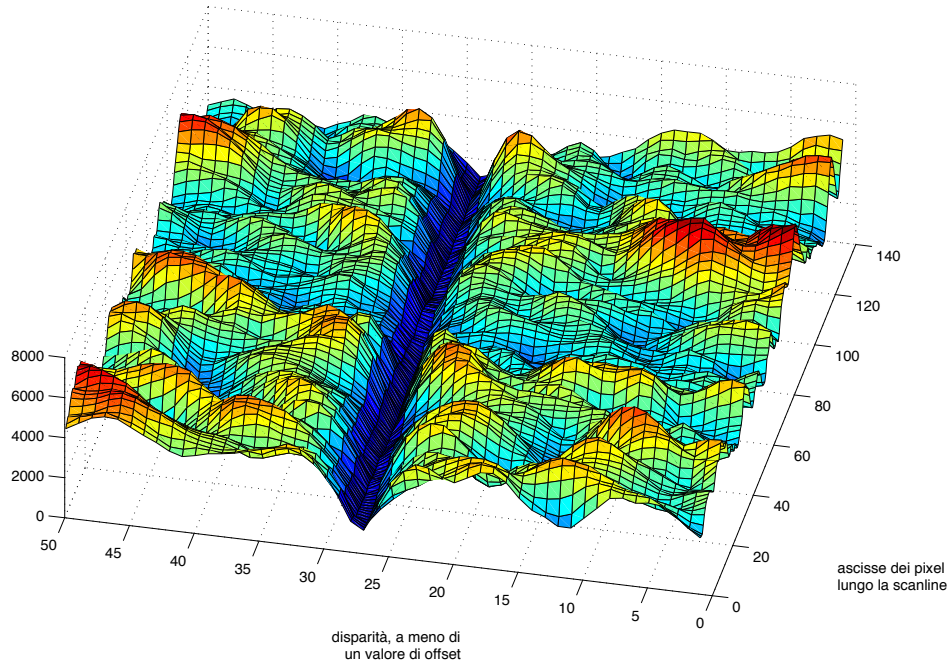


Fig. 3.12: andamento dei costi in una sezione del DSI corrispondente alla scanline centrale dell'area considerata, con aggregazione dei costi su una finestra quadrata di dimensione 7x7, di disparità costante.

3.3.2 Comportamento nelle discontinuità, in presenza di segnale alle alte frequenze

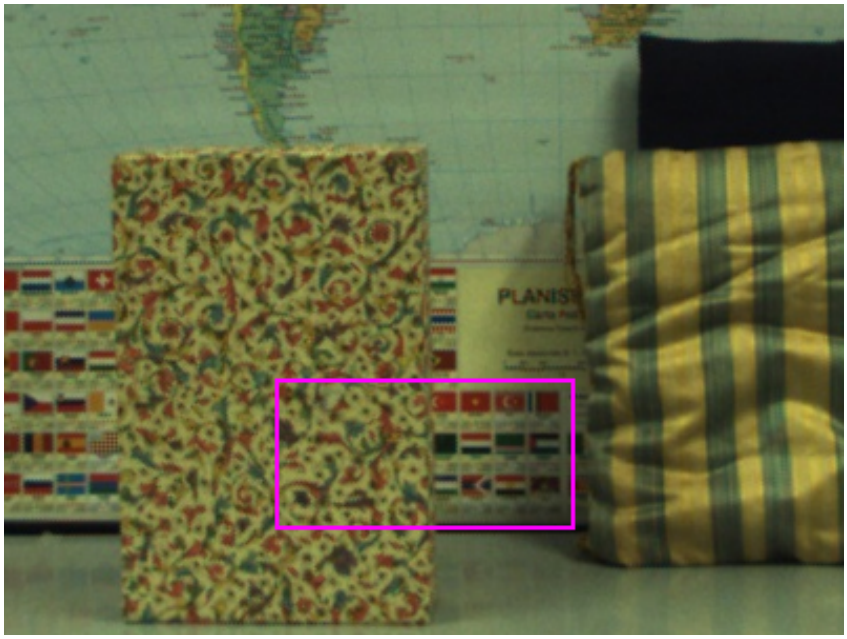


Fig. 3.13: area considerata nella sezione 3.3.2

Risultati senza aggregazione dei costi

I risultati sono analoghi a quelli del caso di superficie piana, trattandosi effettivamente di due superfici piane e non essendoci contaminazione tra i costi di pixel adiacenti (manca la fase di aggregazione). La sezione del DSI (figura 3.15) in corrispondenza della scanline centrale dell'area considerata, evidenzia un andamento irregolare, con righe a basso costo (d=28 sulla sinistra e d=10 sulla destra) non particolarmente marcate, e la presenza di minimi locali anche per altri valori dell'ascissa.

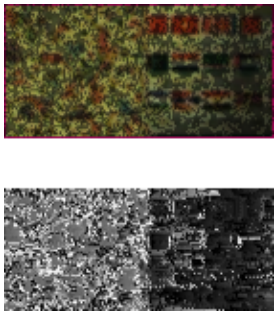


Fig. 3.14: Sopra: area esaminata, con ombreggiatura delle occlusioni. Sotto: mappa di disparità.

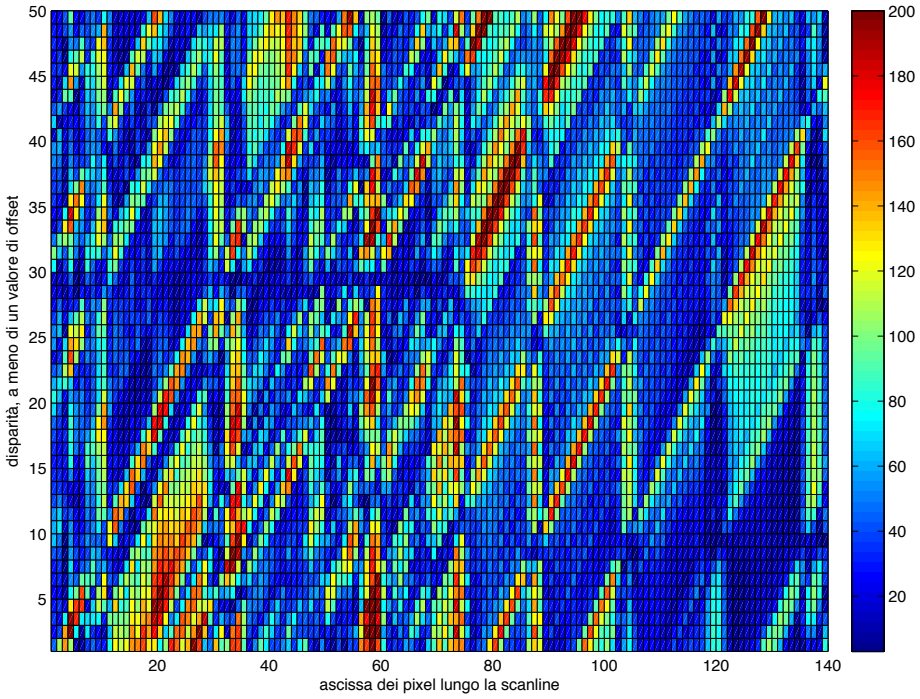


Fig. 3.15: andamento dei costi in una sezione del DSI corrispondente alla scanline centrale dell'area considerata, senza aggregazione dei costi. Si possono osservare tratti orizzontali con costi bassi per d=28 (a sinistra) d=10 (a destra), coerenti con la geometria della scena. L'aggregazione su un supporto a disparità costante rende più marcati questi minimi (figura 3.17), che una somma lungo direzioni diverse avrebbe invece potuto penalizzare con costi maggiori. È questo un esempio del modo in cui il metodo privilegia le superfici frontali.

Risultati dopo aggregazione su una finestra 21x21

L'aggregazione dei costi permette una corretta attribuzione delle disparità sulle superfici, ma con punti ancora privi di corrispondenza biunivoca ($d \neq d'$) nei pressi della discontinuità. Quest'ultima risulta essere stata localizzata correttamente lungo l'asse orizzontale per quel che riguarda la scanline considerata, mentre più in alto e più in basso la discontinuità si è spostata verso destra. Ciò si può giustificare notando nella sezione del DSI che l'estensione longitudinale di un avvallamento dei costi lungo una linea d =costante, è condizionata (in seguito alla fase di aggregazione) dall'entità dei costi lungo la stessa linea. Le estremità di un avvallamento si alzano in base alle caratteristiche delle aree adiacenti dell'immagine, e indipendentemente dalle proprie.

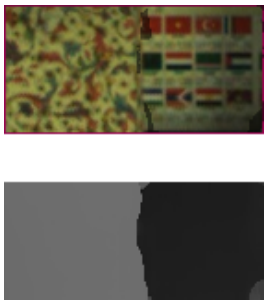


Fig. 3.16: Sopra: area esaminata, con ombreggiatura dei pixel per i quali non vale la (3.2). Sotto: mappa di disparità.

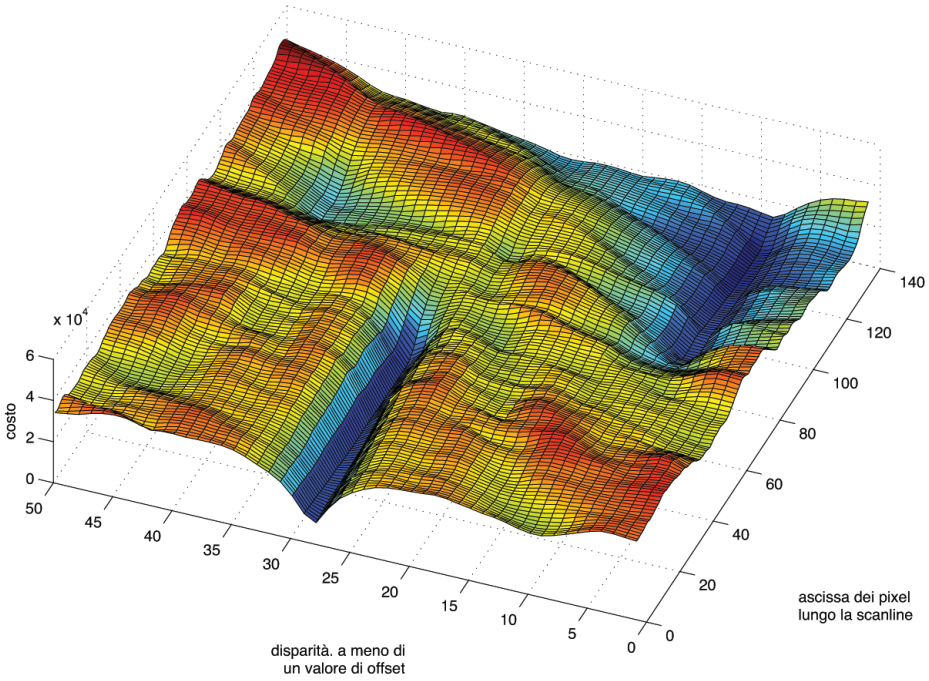


Fig. 3.17: andamento dei costi in una sezione del DSI corrispondente alla scanline centrale dell'area considerata.

3.3.3 Comportamento nelle discontinuità tra regioni disomogenee per caratteristiche del segnale

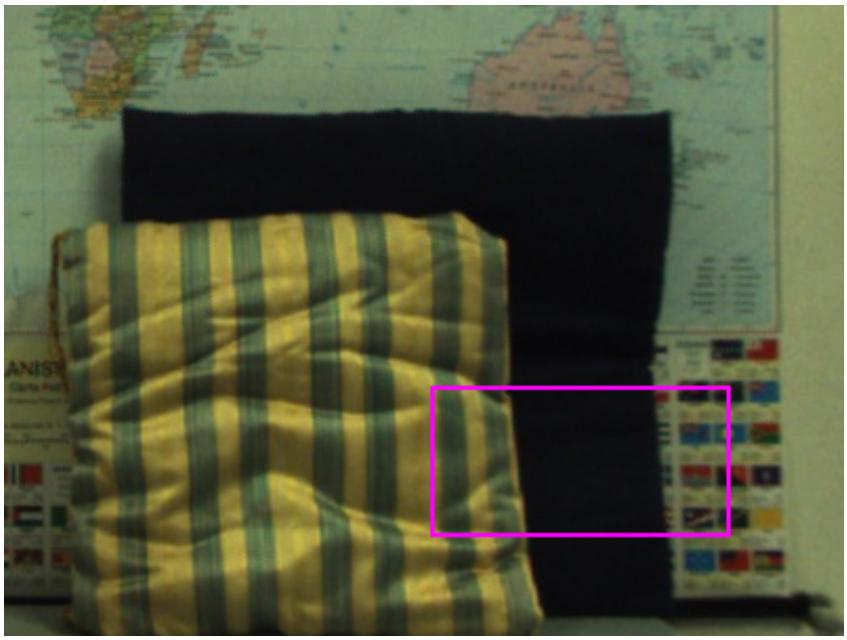


Fig. 3.18: area considerata nella sezione 3.3.3

Risultati senza aggregazione dei costi

Nelle aree di destra e di sinistra l'algoritmo produce una mappa simile a quelle dei casi precedenti, in assenza di aggregazione dei costi. Nell'area centrale nera i pixel risultano per lo più occlusi, e la mappa non attribuisce un valore di disparità intermedio a quelli delle aree laterali (come dovrebbe invece essere, data la geometria della scena). La sezione del DSI mostra che un costo molto basso per qualsiasi disparità, a causa dell'assenza di texture.

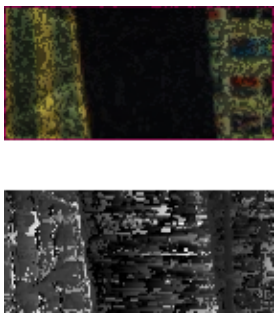


Fig. 3.19: Sopra: area esaminata, con ombreggiatura dei pixel per i quali non vale la (3.2). Sotto: mappa di disparità.

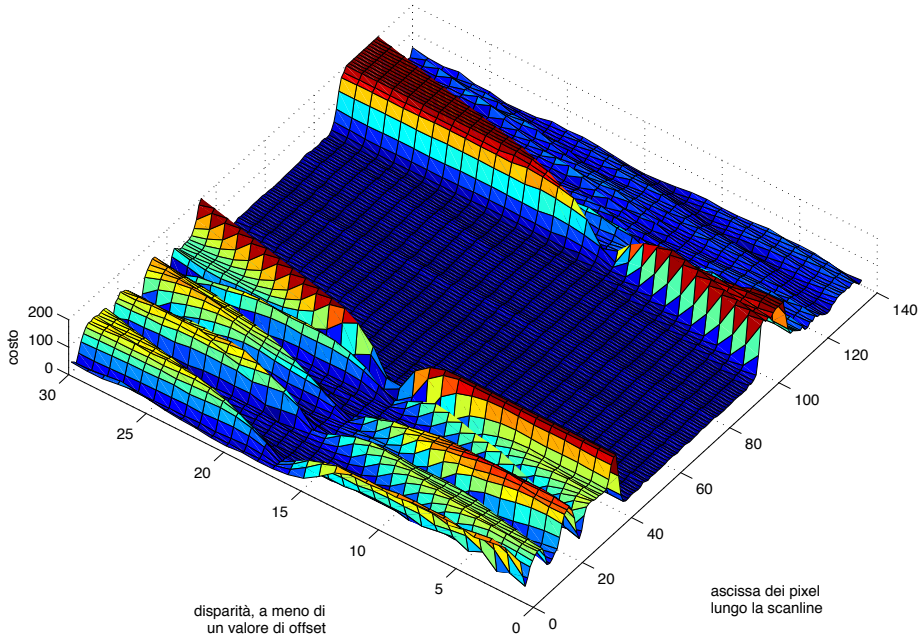


Fig. 3.20: andamento dei costi in una sezione del DSI corrispondente alla scanline centrale dell'area considerata. Le creste trasversali riflettono le discontinuità cromatiche nell'immagine.

Risultati dopo aggregazione su una finestra 21x21

Ancora una volta, nelle aree dove è presente informazione, si osserva l'assegnazione di un valore di disparità coerente con la geometria della scena. Alla parte centrale non viene attribuita una disparità intermedia, si osserva solo una maggiore regolarità di valori incoerenti. Oltre a questo si nota il restringimento dell'area centrale, a vantaggio di quelle laterali. Queste ultime – come si osserva nella sezione del DSI – nella fase di aggregazione allargano l'attribuzione dei costi all'area adiacente, fuorché in corrispondenza degli avvallamenti, che risultano così prolungati.

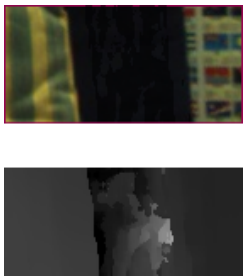


Fig. 3.21: area esaminata e relativa mappa di disparità.

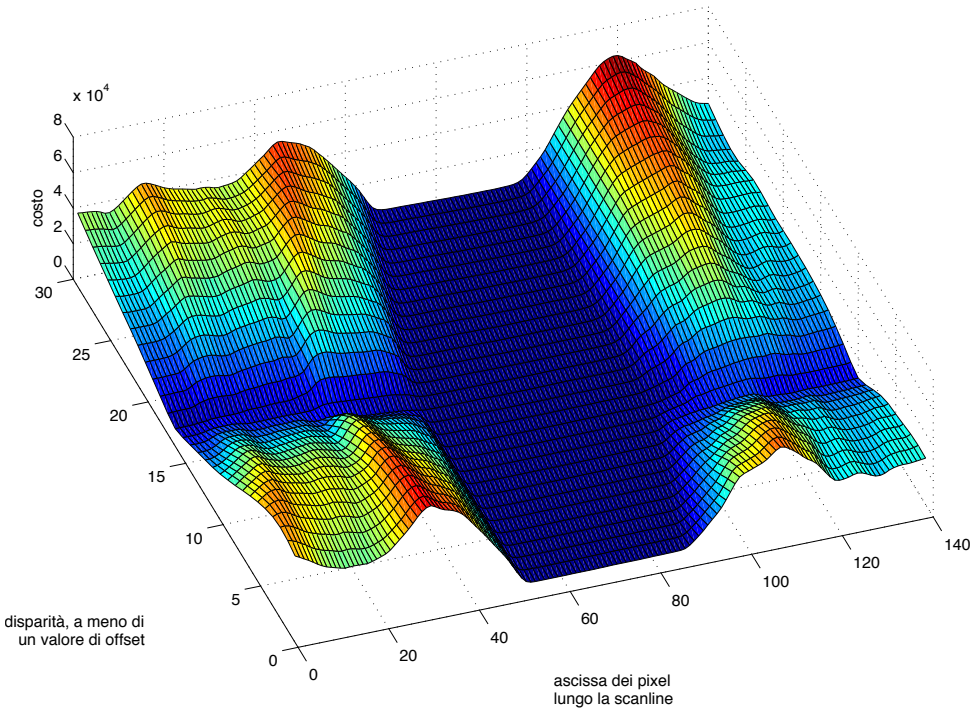


Fig. 3.22: andamento dei costi in una sezione del DSI corrispondente alla scanline centrale dell'area considerata.

3.3.4 Comportamento nelle regioni uniformi

Costituisce una delle situazioni maggiormente critiche per l'attribuzione dei costi.

Risultati senza aggregazione dei costi

La maggior parte della regione esaminata appare occlusa, pur non essendolo in realtà (risulta cioè $d \neq d'$). La mappa di disparità ha un andamento molto irregolare, quando dovrebbe invece essere uniforme, trattandosi di una superficie frontale piana. Inoltre la gamma di disparità presenti è molto estesa (dal bianco al nero), contrariamente a quanto la geometria della scena imporrebbe.

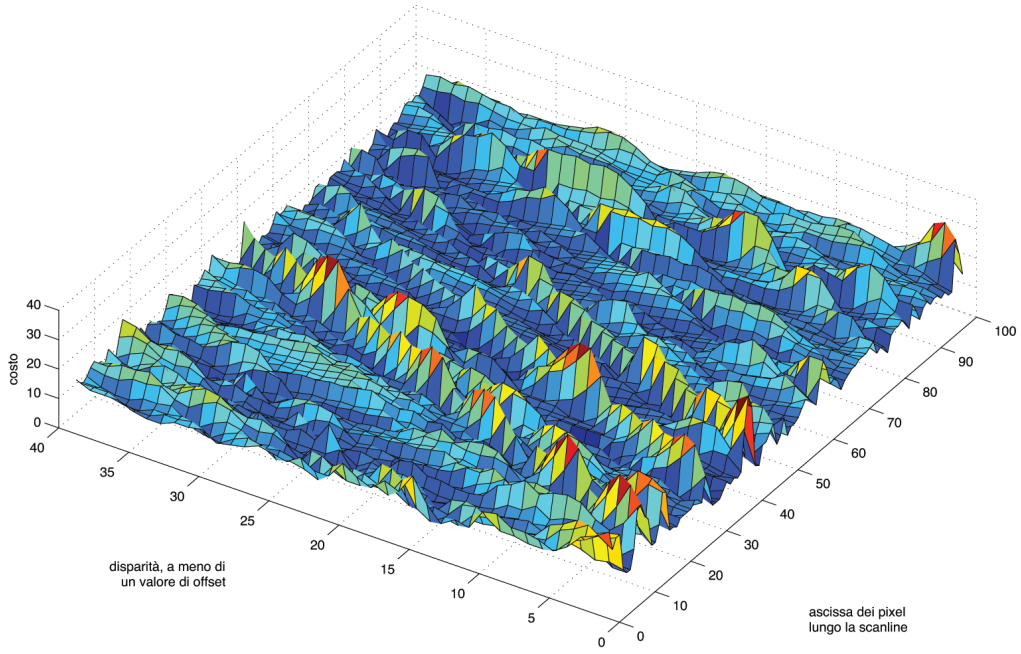
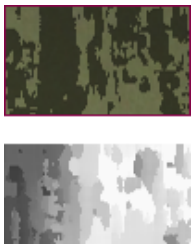


Fig. 3.23: andamento dei costi in una sezione del DSI corrispondente alla scanline centrale dell'area considerata. Malgrado si tratti di una superficie frontale, non appare nessun minimo che si estenda lungo una linea di disparità costante.

Risultati dopo aggregazione su una finestra 31x31

Diminuisce il numero di pixel che risultano occlusi all'esame del cross-checking, ma come risulta dalla mappa di disparità ciò non corrisponde a un significativo miglioramento del risultato dell'elaborazione, che continua a presentare una gamma estesa di disparità.



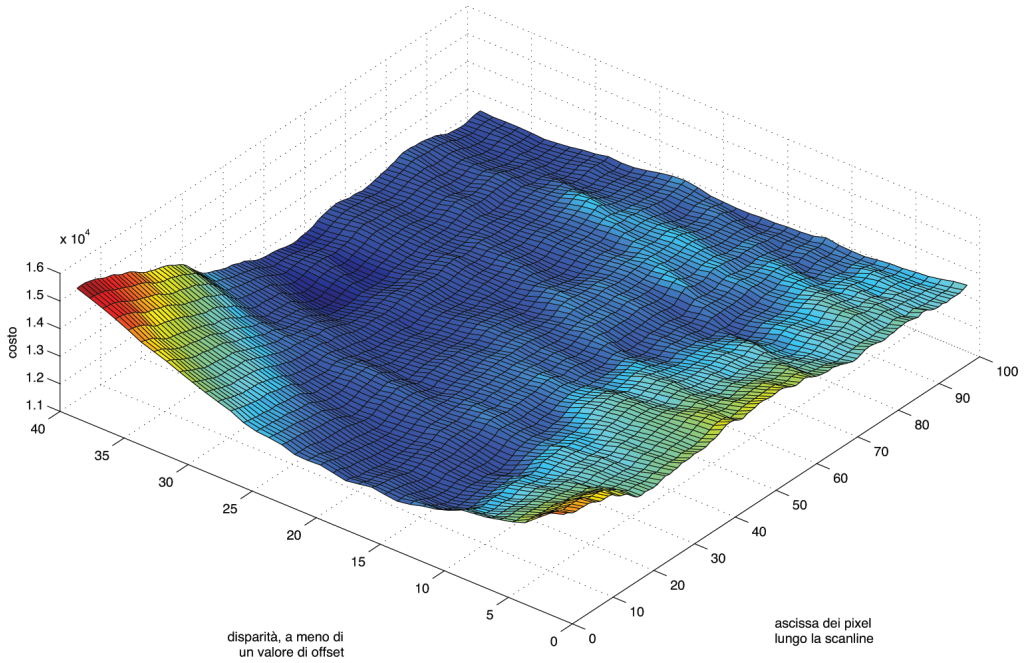


Fig. 3.24: andamento dei costi in una sezione del DSI corrispondente alla scanline centrale dell'area considerata. La disparità corretta vale circa 10, ma la linea $d = 10$ non corrisponde al minimo per tutte le ascisse. I massimi lungo i bordi della sezione non sono giustificabili con la sola immagine visualizzata, dal momento che l'aggregazione su una finestra di lato 31 accumula costi esterni alla zona esaminata.

Capitolo 4

Space-Time Stereo attivo

Includere considerazioni temporali nel problema della ricostruzione di una scena può permettere di raccogliere informazioni più accurate per il calcolo della disparità. Ciò è stato ad esempio realizzato[1] – senza modificare l'hardware impiegato – nell'ambito della triangolazione laser, una tecnica affine allo *stereo matching*, nella quale le informazioni necessarie al calcolo della coordinata z provengono da una fotocamera e da un proiettore laser, anziché da due fotocamere.

Per rendere possibile una simile generalizzazione nel matching binoculare è necessario acquisire più immagini nel tempo. La registrazione delle coppie di immagini in condizioni costanti (scena e illuminazione statiche) non avrebbe un significativo impatto sulla matrice dei costi: nelle ipotesi di rumore casuale a media nulla, e scorrelato dal segnale, permetterebbe solo il miglioramento del rapporto segnale rumore delle immagini di un fattore $\sqrt{N}$, dove N è il numero di coppie di foto. Perché sia lecito aspettarsi un consistente miglioramento della qualità della mappa di disparità è necessario modificare il DSI, alterando l'aspetto della scena (e quindi i costi memorizzati nel DSI) senza modificarne la geometria. Lo stereo attivo[2] mira proprio ad aumentare la variabilità delle singole immagini proiettando sulla scena opportuni pattern di luce. Questa tecnica trova impiego anche nella ricostruzione di scene in movimento, e applicata a scene statiche si è dimostrato essere un metodo adatto alla costruzione di mappe precise e affidabili,[3] utilizzabili come termini di paragone per le prestazioni di altri algoritmi.

[1] B. Curless, M Levoy. *Better Optical Triangulation through Spacetime Analysis.*

[2] J. Davis, R. Ramamoorthi, S. Rusinkiewicz, *Spacetime Stereo: A Unifying Framework for Depth from Triangulation* e Li Zhang, B. Cureless, S. M. Seitz, *Spacetime Stereo: Shape Recovery for Dynamic Scenes*

[3] Daniel Scharstein, Richard Szeliski *High-Accuracy Stereo Depth Maps Using Structured Light.*

4.1 Compromesso tra risoluzione e accuratezza nella aggregazione con fixed window

Le cause degli errori del metodo fixed windows con strategia di ottimizzazione *winner takes all* sono dovute al fatto che[4]:

- è implicita l'ipotesi di una scena composta da superfici frontali;
- vengono ignorate le discontinuità di profondità;
- non cè robustezza rispetto ad aree prive di texture;
- non cè robustezza rispetto ad aree con pattern ripetuti;

Dalle osservazioni che si possono trarre dalle prove che concludono il capitolo precedente i primi due punti sono accomunabili nell'essere conseguenza dell'ipotesi, implicita nell'aggregazione con fixed windows, che ciascun pixel sia interno a una superficie piana della dimensione del supporto U. Il conseguente effetto negativo è attenuato dalla scelta di un supporto più piccolo, ed è annullato quando U è costituito da un solo pixel. In quest'ultimo caso è infatti annullata l'incidenza dei costi delle aree adiacenti (come osservato in 3.3.3).

Tabella 4.1: principali fattori critici nel metodo fixed windows (figura 4.1).

	alta discriminabilità tra pixel	bassa discriminabilità tra pixel
finestra piccola	A) corretta collocazione delle discontinuità; rari errori nel cross-checking;	B) corretta collocazione della discontinuità; numerosi errori nel cross-checking;
finestra grande	C) meno errori nel cross-checking rispetto ad A; imprecisioni nelle discontinuità;	D) diffusi errori nel cross-checking; imprecisoni nelle discontinuità;

[4] schematizzazione tratta da Stefano Mattoccia *http://www.vision.deis.unibo.it/smatt/Seminars/StereoVision.pdf* p. 51

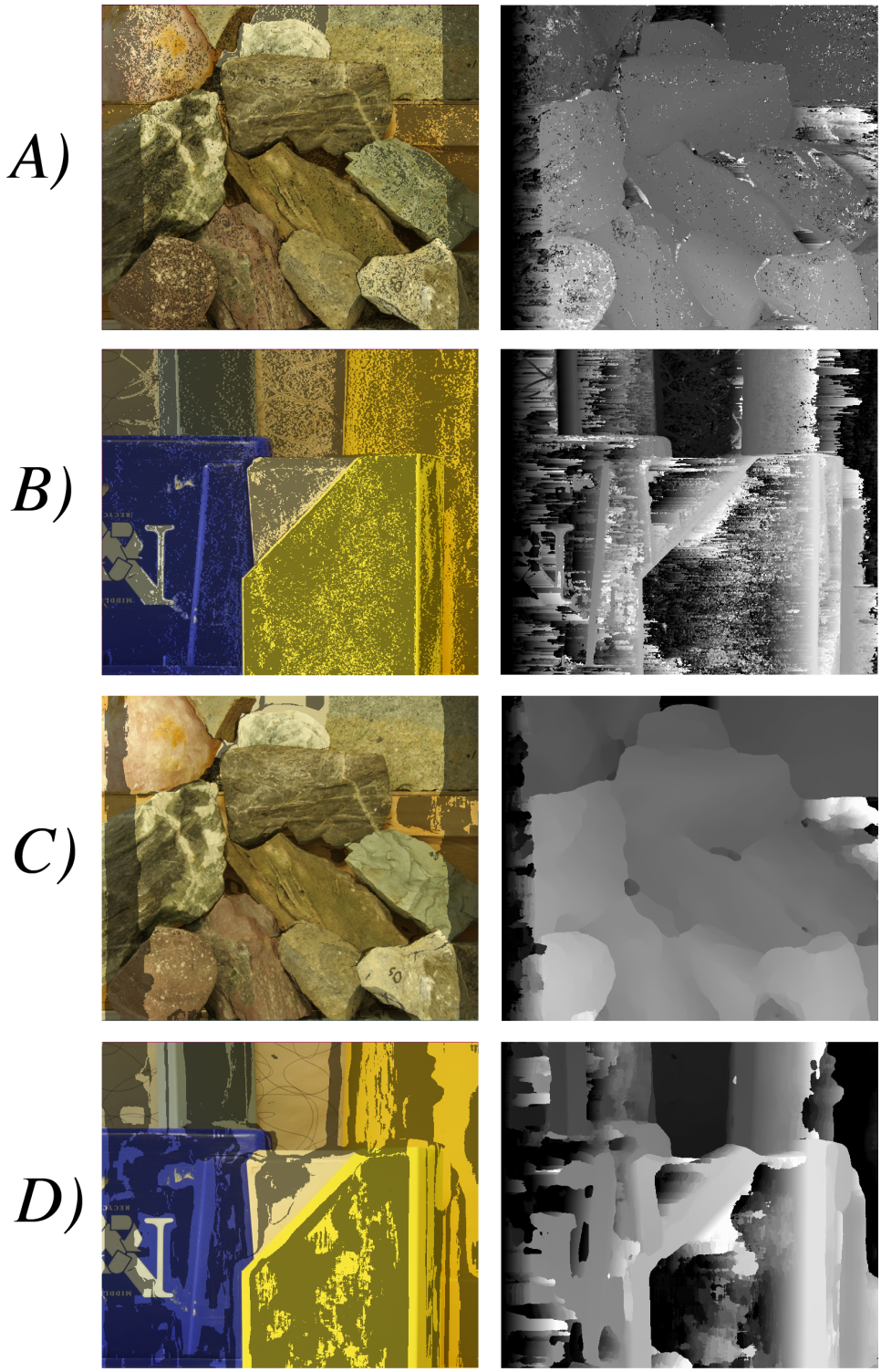


Fig. 4.1: esempi relativi alla tabella 4.1, con supporti 3x3 (A e B) e 31x31 (C e D). A sinistra sono evidenziati i pixel negativi al cross-checking, a destra la mappa di disparità.

I rimanenti due punti possono essere anch'essi riuniti, essendo conseguenza della non discriminabilità tra punti, diffusa in un'area in un caso (aree uniformi), concentrata in specifici punti nell'altro (pattern ripetuti). Con riferimento alla figura 4.1 e alla tabella 4.1 è visibile che i risultati migliori in termini di precisione si ottengono nelle condizioni del caso A. La presenza di aree uniformi aumenta il numero di corrispondenze ambigue ($d \neq d'$) e costringe ad ampliare il supporto, a scapito della precisione.

4.2 Effetto dell'aggregazione nella dimensione del tempo

La tecnica dello space-time stereo ha l'obiettivo di creare un DSI con le caratteristiche del caso A, utilizzando un supporto di dimensioni 1x1 o 3x3, e sopperendo all'eventuale bassa discriminabilità tra pixel con la proiezione di un pattern luminoso (figura 4.2).

Fig. 4.2: esempio di pattern proiettato sulla scena.

Il pattern deve avere caratteristiche tali da aumentare la variabilità lungo le scanline (questo giustifica la scelta di un pattern con righe verticali), facendo crescere il costo associato alle corrispondenze errate. Questo è di per sé sufficiente a migliorare la qualità della mappa ottenuta da una singola coppia di immagini. Ripetendo l'operazone per N coppie di immagini si ottengono altrettante DSI_i con distribuzione dei costi puntualmente diverse ($DSI_i(u_0, v_0, d_0) \neq DSI_j(u_0, v_0, d_0)$, in generale), ma morfologicamente simili nel posizionamento dei minimi relativi che sono coerenti con la geometria della scena. Altri minimi possono essere presenti, conseguenza di uniformità

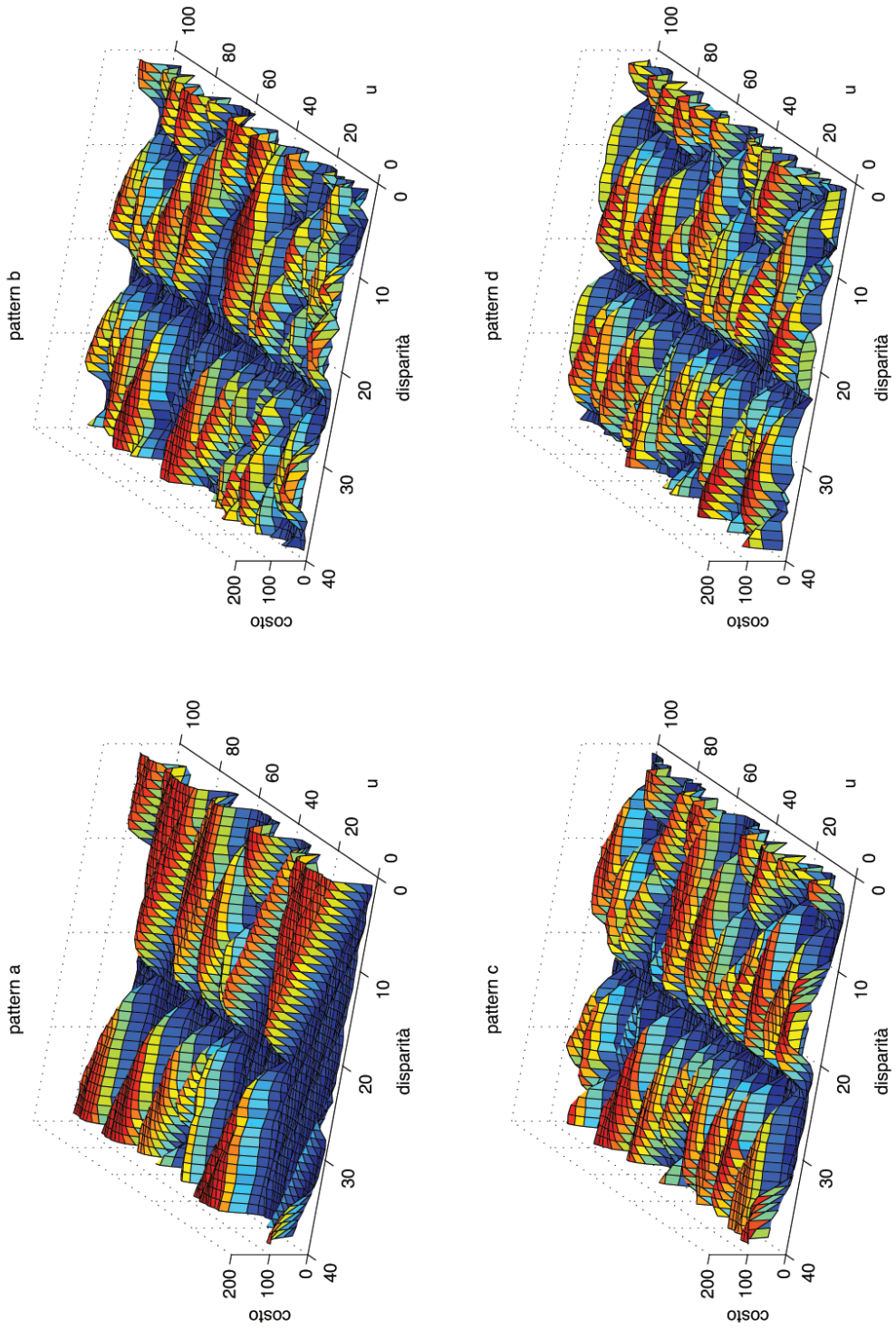


Fig. 4.3: sezioni del DSI relative a una stessa area per i pattern evidenziati in figura 4.4.

Fig. 4.4: quattro pattern proiettati su una parate.

locali dei canali RGB. Variando le righe del pattern questi minimi incoerenti con la geometria cambiano di posizione e vengono sostituiti da creste. Questo è visibile nelle sezioni del DSI di figura 4.3, riferite ai quattro pattern di figura 4.4. Il solco a disparità 20 corrisponde alla distanza dalla parete, ed è presente in tutte le quattro sezioni. Gli altri minimi invece cambiano ad ogni immagine, al variare dello spessore e del posizionamento delle righe proiettate. Integrando nel tempo si ottiene la somma:

$$\overline{DSI}(u, v, d) = \sum_{i=1}^{N} DSI_i(u, v, d);$$

nel caso non ci sia aggregazione dei costi, un pixel non occluso (u_0, v_0) avrà associato un costo basso alla disparità corretta $\bar{d}$ in tutti i DSI_i. Al contrario, salvo situazioni anomale, in tutte le altre disparità il costo varia con il variare del pattern. Il risultato della somma è visibile nella figura 4.5. È evidente che la scelta della disparità di costo minimo non è soggetta ad ambiguità nel caso $N = 100$.

A parte le occlusioni e le superfici riflettenti (già escluse dal vincolo di unicità, e a maggior ragione da evitare nel caso si utilizzi un proiettore), le anomalie sono dovute a aree della scena che non riflettono a sufficienza il pattern. In teoria, per una superficie totalmente opaca la determinazione della disparità non trae vantaggio dall'uso di un proiettore; in termini pratici però il pattern causa sempre un aumento dei costi per le corrispondenze errate, anche se questo incremento può essere molto contenuto. La conseguenza è una convergenza più lenta verso il valore finale. Questi aspetti saranno analizzati quantitativamente nel capitolo successivo.

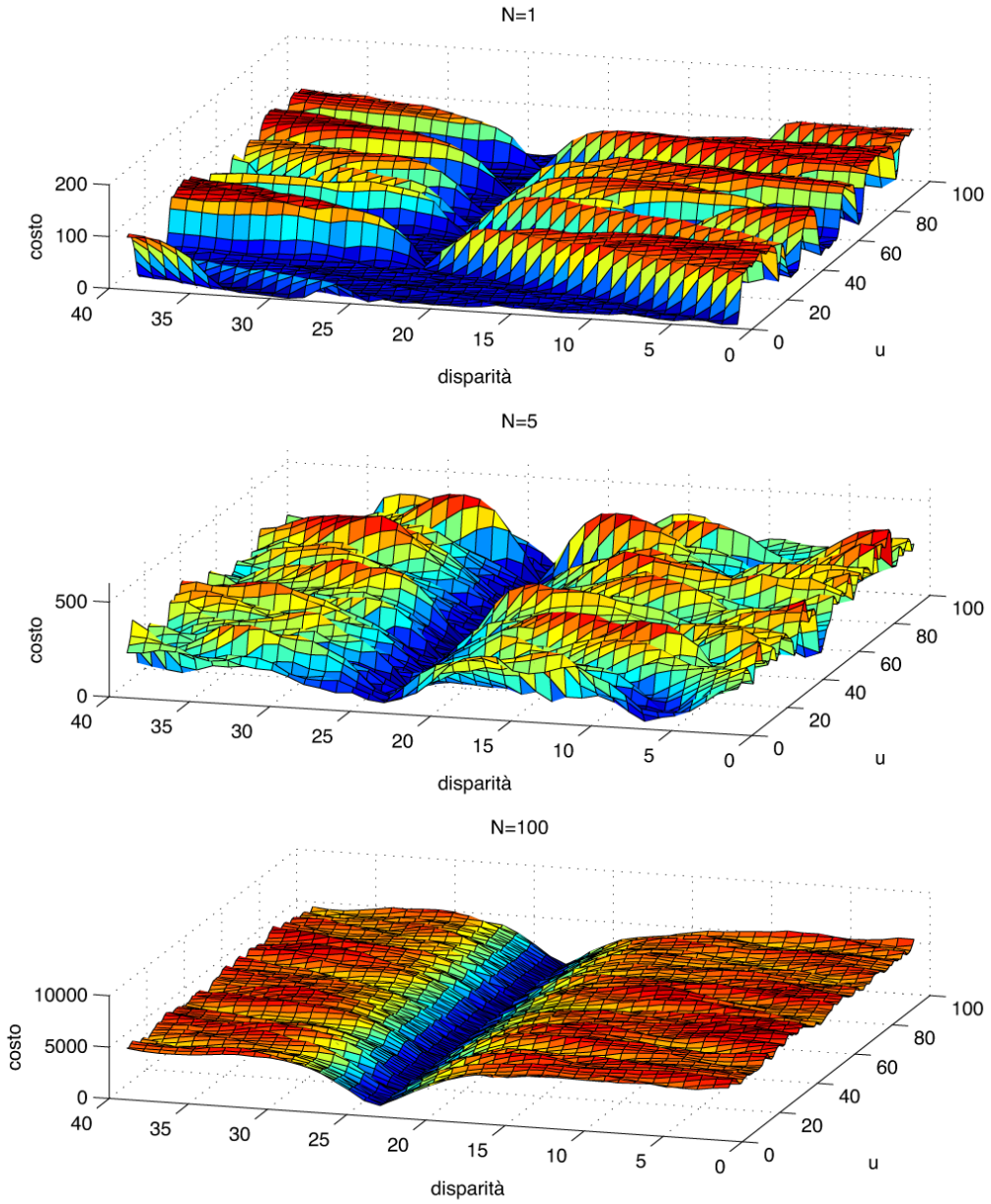


Fig. 4.5: sezioni del $\overline{DSI}$ per diversi valori di N.

4.3 Implementazione dello space-time stereo

L'elaborazione della mappa di disparità a partire dai *dataset* di immagini rettificate è stata implementata in C, utilizzando la libreria OpenCV.

4.3.1 Calcolo dei DSI

Alla base dell'elaborazione della mappa di disparità c'è la funzione che a partire da una coppia di immagini rettificate costruisce il DSI, utilizzando l'algoritmo fixed windows. La velocità di questo metodo è dovuta all'implementazione con algoritmo box-filtering (figura 4.6) che rende la complessità $O(WHL)$, indipendente dalla dimensione del supporto.

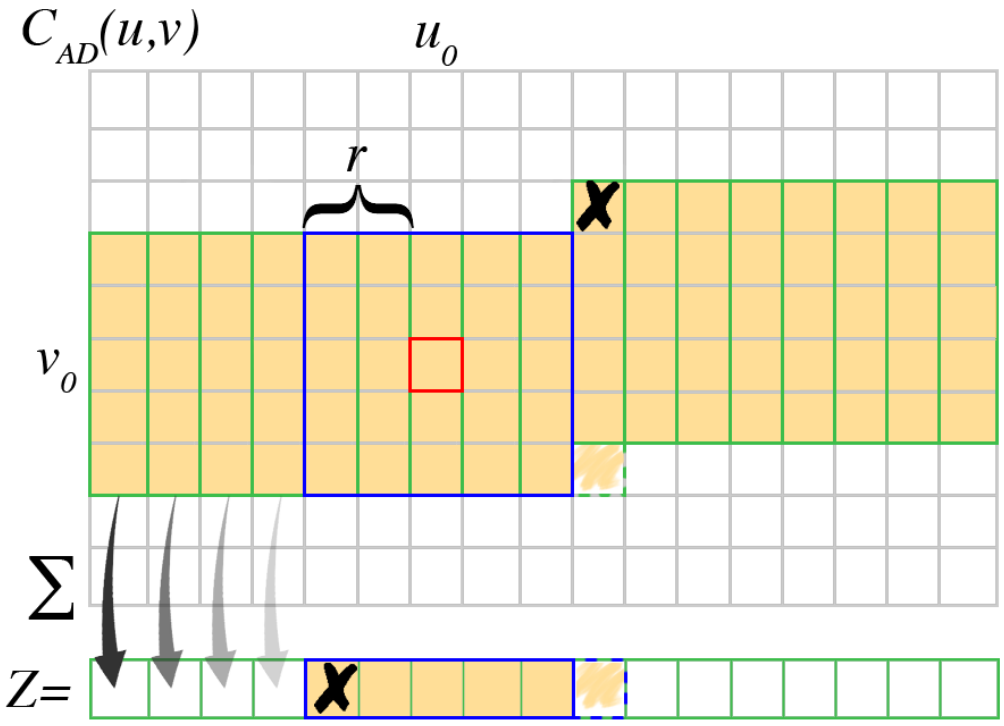


Fig. 4.6: aggregazione dei costi con algoritmo box-filtering. Il calcolo di ciascun costo aggregato (C_{SAD}) richiede solo quattro operazioni, corrispondenti all'aggiornamento dei primi e ultimi termini delle due sommatorie: $Z_i = \sum_{i=v_0-r}^{v_0+r} C_{AD}(u_0, i)$ e $C_{SAD}(u_0, v_0) = \sum_{i=u_0-r}^{u_0+r} Z_i$.

Tale operazione, che combina la fase di calcolo dei costi e quella di aggregazione, è realizzata dalla funzione

```
void box_filtering_TAD_stereo_color()
```

Gli argomenti passati ad essa sono, nell'ordine[5]:

- `int w`: larghezza delle immagini;
- `int h`: altezza delle immagini;
- `int nDisp`: range delle disparità ($disp_{max} - disp_{min} + 1$);
- `int offs_x`: disparità minima $disp_{min}$;
- `IplImage *L`: immagine sinistra;
- `IplImage *R`: immagine destra;
- `int **out`: il risultato dell'elaborazione è accessibile tramite questo puntatore. Esso punta al DSI, calcolato utilizzando l'immagine di sinistra come riferimento. Ovviamente la memoria dev'essere preventivamente allocata;
- `int **out_inv`: come `out`, ma per il DSI calcolato utilizzando l'immagine di destra come riferimento. Se uguale a `NULL` viene ignorato;
- `int r`: raggio del supporto;
- `int threshold`: soglia utilizzata nel calcolo del costo tramite TAD;

Esempio di allocazione del DSI:

```
int** DSI = (int **)calloc(Width*Height, sizeof(int*));
for (int i=0; i<Width*Height; i++)
   DSI[i] = (int *) calloc(nDisp, sizeof(int));
```

Il calcolo di entrambe le DSI non ha un costo computazionale significativamente superiore al calcolo di una singola DSI, dal momento che i costi delle corrispondenze sono gli stessi. A differire è solo l'ordine in cui questi costi vengono salvati nella matrice (figura 4.7).

[5]fatta eccezione per la gestione dell'uscita `out_inv`, il codice di questa funzione è stato messo a disposizione dal professor Stefano Mattoccia dell'università di Bologna.

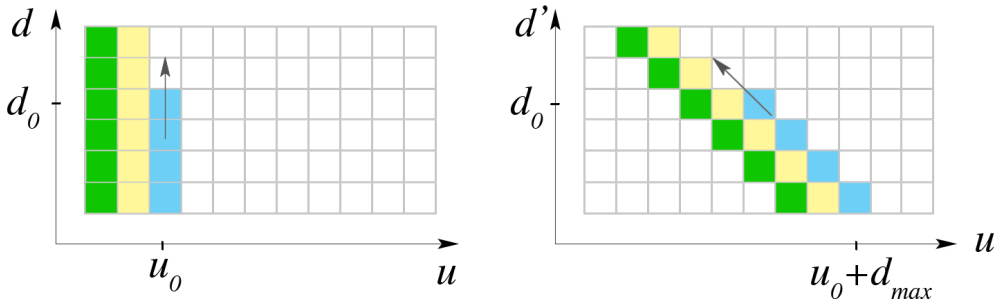


Fig. 4.7: diversa disposizione dei medesimi costi nel DSI che ha per riferimento l'immagine di sinistra e in quello che ha per riferimento l'immagine destra.

4.3.2 Altri metodi

La funzione

```
void shiftable_windows()
```

effettua il calcolo del DSI secondo il metodo omonimo. Essa riceve in ingresso:

- `int Width`: il DSI cumulativo, o anche un DSI frutto di una sola coppia di immagini.
- `int Height`: altezza delle immagini;
- `int nDisp`: range delle disparità ($disp_{max} - disp_{min} + 1$);
- `int offs_x`: disparità minima $disp_{min}$;
- `IplImage *L`: immagine sinistra;
- `IplImage *R`: immagine destra;
- `int **DSI`: come `out` in 4.3.1;
- `int **DSI2`: come `out_inv` in 4.3.1. Viene ignorato se NULL;
- `int n`: dimensione laterale della finestra quadrata;
- `int threshold`: soglia utilizzata nel calcolo del costo tramite TAD;

I costi delle disparità vengono calcolati invocando iterativamente, per n^2 volte (lo stesso fattore che moltiplica anche il costo computazionale), la funzione

```
void box_filtering_TAD_stereo_color_4()
```

che è una versione modificata della funzione che implementa il metodo fixed windows. I parametri sono:

- `int w`: larghezza delle immagini;
- `int h`: altezza delle immagini;
- `int nDisp`: range delle disparità ($disp_{max} - disp_{min} + 1$);
- `int offs_x`: disparità minima $disp_{min}$;
- `IplImage *L`: immagine sinistra;
- `IplImage *R`: immagine destra;
- `int **out`: come in 4.3.1;
- `int **out_inv`: come in 4.3.1. Viene ignorato se NULL;
- `int r_sx`: distanza tra il pixel considerato e il margine sinistro del supporto (in pixel);
- `int r_dx`: distanza tra il pixel considerato e il margine destro del supporto (in pixel);
- `int l_up`: distanza tra il pixel considerato e il margine superiore del supporto (in pixel);
- `int l_down`: distanza tra il pixel considerato e il margine inferiore del supporto (in pixel);
- `int threshold`: soglia utilizzata nel calcolo del costo tramite TAD;

La funzione

`void weighted_DSI()`

elabora il DSI, aggregando i costi e pesandoli secondo la distanza e la somiglianza rispetto al pixel centrale della finestra[6] I parametri della funzione sono:

- `IplImage* L`: immagine sinistra;

[6]si tratta del metodo descritto in Kuk-Jin Yoon, In So Kweon *Adaptive Support-Weight Approach for Correspondence Search.*

- `IplImage* R`: immagine destra;
- `float** DSI`: come `out` in 4.3.1;
- `float** w_sx`: pesi attribuiti ai pixel dell'immagine sinistra;
- `float** w_dx`: pesi attribuiti ai pixel dell'immagine destra;
- `int r`: raggio del supporto;
- `int nDisp`: range delle disparità ($disp_{max} - disp_{min} + 1$);
- `int offs_x`: disparità minima $disp_{min}$;
- `int T`: valore limite del costo;

I pesi `w_sx` e `w_dx` vengono calcolati dalla funzione

`void calcolo_pesi()`

La invocazione deve definire:

- `IplImage* imm`: immagine di cui si calcolano i pesi. Si ottengono risultati migliori se i colori sono rappresentati nello spazio CIELab;
- `float** weight`: è l'output della funzione. E' costituito da un numero di vettori pari al numero di pixel. Ciascun vettore contiene $(2r + 1)^2$ elementi;
- `int r`: raggio del supporto;
- `int w`: larghezza dell'immagine;

4.3.3 Calcolo delle somme parziali dei DSI

L'operazione avviene accumulando i costi dei successivi DSI su matrici delle stesse dimensioni. L' accumulo dei costi richiede $W * H * L$ somme. Alla funzione

`void somma_DSI()`

devono essere passati:

- `int **DSI_sum`: il DSI cumulativo;
- `int** DSI`: è il DSI che si vuole somare al DSI cumulativo;
- `int Width`: larghezza delle immagini;
- `int Height`: altezza delle immagini;
- `int nDisp`: range delle disparità ($disp_{max} - disp_{min} + 1$);

4.3.4 Calcolo della mappa di disparità

Viene calcolata dalla funzione

```
void mappa_disp()
```

che riceve i parametri:

- `int **DSI_sum`: il DSI cumulativo, o anche un DSI frutto di una sola coppia di immagini.
- `int nDisp`: range delle disparità ($disp_{max} - disp_{min} + 1$);
- `int offs_x`: disparità minima $disp_{min}$;
- `int Width`: come in 4.3.1;
- `int Height`: altezza delle immagini;
- `float *data`: matrice della stessa dimensione delle immagini, che costituisce il risultato dell'elaborazione;

La funzione `mappa_disp()` non fa che percorrere `DSI_sum` secondo gli indici relativi ai (W*H) pixel, a cui corrispondono altrettanti vettori, e richiamare per ciascuno la funzione

```
float disp_min_cost()
```

I parametri di quest'ultima sono:

- `int* costi`: sono i vettori che costituiscono il DSI. Ciascuno di essi contiene tutti i costi associati alle corrispondenze di un singolo pixel;
- `int lunghezza`: range delle disparità, come `nDisp` nelle precedenti funzioni. Coincide con la lunghezza del vettore passato come primo argomento;

Questa funzione cerca la disparità di minimo costo. Se essa è interna alla gamma di valori (cioè se non vale d_{min} o d_{max}) allora viene invocata la funzione

```
float get_subpixel_adjustment_min()
```

che a partire dal costo minimo e dai costi delle disparità adiacenti determina – tramite interpolazione con una parabola – l'addendo correttivo per ottenere la precisione subpixel (come descritto nel capitolo 3).
Parametri di `get_subpixel_adjustment_min()`:

- `int left`: costo della disparità precedente a quella di costo minimo;
- `int center`: disparità di costo minimo;
- `int right`: costo della disparità successiva a quella di costo minimo;

4.3.5 Cross-checking

La funzione

`void calcolo_occlusioni()`

valuta per ciascun pixel la validità della condizione:

$$d(u,v) = d'(u - d(u,v), v)$$

Lista dei parametri:

- `int Width`: larghezza dell'immagine;
- `int Height`: altezza delle immagini;
- `int nDisp`: range delle disparità ($disp_{max} - disp_{min} + 1$);
- `int offs_x`: disparità minima $disp_{min}$;
- `float *data`: mappa di disparità ottenuta da `mappa_disp()` utilizzando la DSI che ha l'immagine sinistra come riferimento (`out` della funzione descritta in 4.3.1);
- `float *data2`: come la percedente, ma ottenuta a partire da una DSI calcolata avendo l'immagine destra come riferimento (`out_inv` della funzione descritta in 4.3.1);
- `int T`: è la soglia con cui viene confrontata la differenza delle disparità. Oltre la soglia il pixel risulta occluso;
- `char* occlusioni`: risultato dell'elaborazione, salvata in un vettore di `char` di dimensione $W * H$ e preventivamente allocato. Al valore 1 corrisponde un pixel occluso a 0 un pixel non occluso;

4.3.6 Altre funzioni

Altre funzioni realizzate, non strettamente necessarie per il calcolo della mappa di disparità.

Estrazione aree di interesse

L'elaborazione della DSI di dataset molto grandi può richiedere tempi lunghi (alcune decine di minuti per 200 coppie di immagini da 700K pixel). Il calcolo può essere limitato a una porzione dell'immagine velocizzando l'operazione di un fattore pari alla riduzione dell'immagine. La funzione `IplImage* estraiROI()` estrae una parte dell'immagine. Parametri:

- `IplImage* img`: immagine dalla quale si estrae l'area di interesse;
- `CvRect ROI`: `CvRect` è una struttura della libreria OpenCV che identifica un rettangolo, e che in questo contesto rappresenta l'area d'interesse;
- `int traslazione`: quantifica l'entità di una traslazione verso sinistra della ROI;

Statistiche sulla rapidità di convergenza

Nel caso siano stati salvati i risultati di un'elaborazione precedente, è possibile ripetere la stessa elaborazione, calcolando ad ogni iterazione la distanza media dal valore finale in una determinata area rettangolare. L'operazione è eseguita dalla funzione `float calcolo_norma()`. Parametri:

- `float* d`: mappa di disparità correntemente calcolata;
- `int W`: larghezza delle immagini di cui si calcola la DSI;
- `float* ptr`: disparità precedentemente salvate (rispetto alle quali quelle correnti vengono valutate);
- `CvRect Rett`: area in cui si calcola la distanza tra la mappa corrente e quella finale;
- `CvRect Rett2`: regione di interesse in cui la DSI viene calcolata;
- `char* occl`: vettore delle occlusioni. Se `NULL` non ha alcun effetto. Altrimenti la distanza tra il valore corrente e quello finale non prende in considerazione i punti occlusi;

La funzione somma le differenze assolute tra il valore corrente e quello finale della disparità. La somma viene divisa per il numero totale di pixel considerati (area di `Rett` meno gli eventuali pixel occlusi).

Un'altra funzione, `void file_punto_m()`, scrive in un file di testo, con la sintassi di MATLab, le istruzioni per visualizzare l'andamento della convergenza verso il valore finale. Parametri:

- `int M`: numero di coppie di immagini considerate;
- `int T`: valore di soglia utilizzato dalla TAD ;
- `int R`: raggio del supporto;
- `float* conv`: vettore in cui sono salvate le distanze medie dal valore finale ad ogni passo dell'elaborazione, calcolate tramite `calcolo_norma()`;

`T` e `R` contraddistinguono le diverse curve nella legenda del grafico. Curve diverse vengono rappresentate in uno stesso grafico, dando per scontato che siano relative ad una stessa regione di interesse.

Sezione della DSI

La funzione `void sezione_DSI()` salva, con la sintassi di MATLab, una sezione della DSI su un file sezione.m. È stata utilizzata per realizzare i grafici. Parametri:

- `int** DSI_sum`: DSI;
- `int Width`: larghezza dell'immagine;
- `int nDisp`: range delle disparità;
- `CvRect ROI`: regione d'interesse;
- `CvRect ROI_DSI`: ;

4.3.7 Funzioni ausiliarie

Funzioni per salvare o visualizzare lo stato dell'elaborazione, e per caricare il risultato di elaborazioni precedenti. I dati gestiti sono quelli relativi alle disparità e alle occlusioni. Possono essere salvati sia come file in formato YAML che come immagini.

Capitolo 5

Risultati delle prove

L'algoritmo che implementa lo space-time stereo è stato messo alla prova con diversi dataset di immagini, acquisite presso il Laboratorio di Tecnologia e Telecomunicazioni Multimediali del Dipartimento di Ingegneria dell'Informazione dell'Università di Padova (figura 5.1).

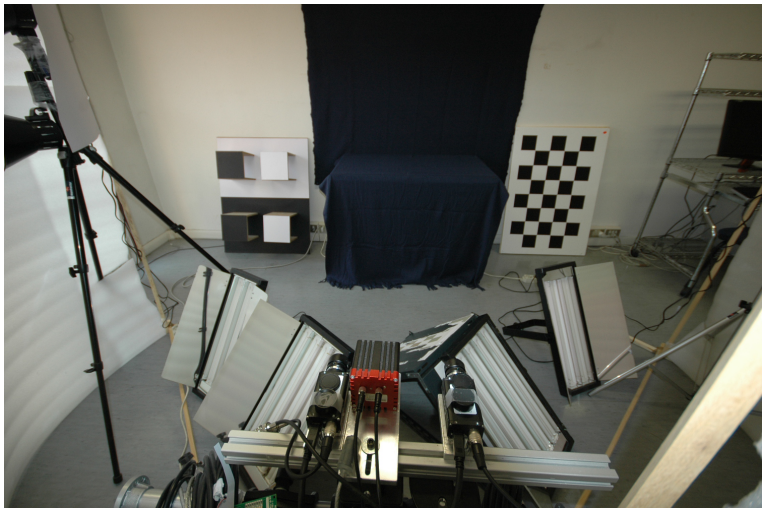


Fig. 5.1: dispositivo di acquisizione delle immagini e apparecchiatura necessaria all'illuminazione della scena.

I criteri di valutazione di una mappa di disparità si basano tipicamente sul confronto rispetto a una misura più accurata e attendibile (*ground truth*). Dal momento che lo space-time stereo si pone come obiettivo quello di produrre mappe di disparità di elevata qualità, non è immediato poter disporre di dati abbastanza precisi da poter permettere un confronto. E il ricorso a dataset reperiti in rete (per lo più rivolti alla valutazione degli algoritmi di stereo

matching per una sola coppia di immagini) non fornirebbe indicazioni utili a giudicare il sistema di acquisizione.

Non disponendo di ground truth, osservazioni possono essere tratte dall'esame della congruenza tra le mappe destra e sinistra.[1] Come già osservato questo tipo di verifica ha come scopo principale quello di individuare i pixel occlusi, visibili cioè solo nell'immagine usata come riferimento. Tuttavia, trattandosi del confronto tra disparità ottenute elaborando in ordine diverso (il vettore dei costi per un pixel nell'immagine di sinistra non ne ha uno uguale nell'immagine di destra) ma secondo uno stesso principio, si possono trarre indicazioni sul comportamento del metodo.

5.1 Considerazioni sulla scelta del dataset

Una coppia di immagini dal dataset utilizzato per i dati presentati in questo capitolo è mostrato nella figura 5.2. Si tratta di una sequenza di 100 coppie di immagini binoculari che sono in realtà un sottoinsieme di una serie di 200 coppie, che ne comprende altre cento del tipo di quelle di figura 4.2, acquisite in presenza di un'altra sorgente luminosa (oltre al proiettore). Le ragioni che giustificano il funzionamento dello space-time stereo suggerirebbero che un maggior numero di immagini dovrebbe condurre a una mappa di qualità migliore.

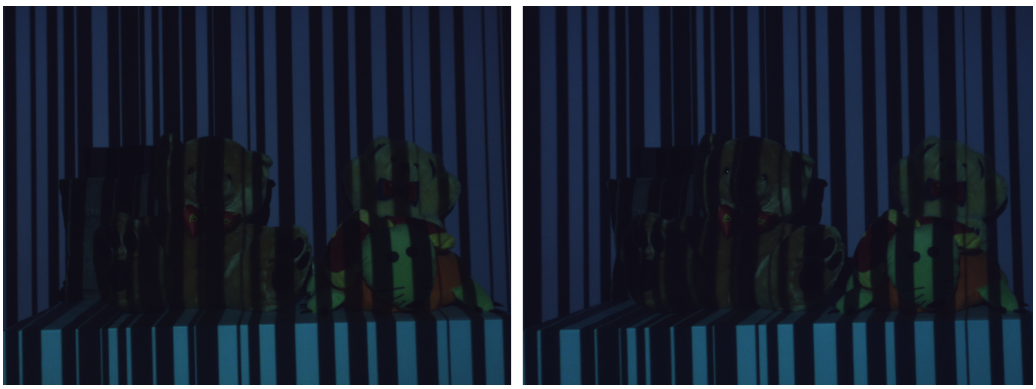


Fig. 5.2: una delle cento coppie di immagini del dataset utilizzato per le prove.

[1] Bogusław Cyganek, J. Paul Siebert *An Introduction to 3D Computer Vision Techniques and Algorithms* p. 228

Nel corso delle prove si è osservato che il dataset costituito dalle cento immagini che si avvalgono della sola luce del proiettore danno un risultato migliore delle altre (4,5% di pixel occlusi contro 8%). Ciò può essere imputato al maggiore contrasto presente nelle immagini, e di conseguenza alla maggiore variabilità lungo la scanline, che ha per conseguenza un aumento dei costi delle corrispondenze sbagliate. Questa differenza tra le mappe andrebbe comunque indagata più in dettaglio, non essendo diffusa, ma concentrata in alcune superfici.

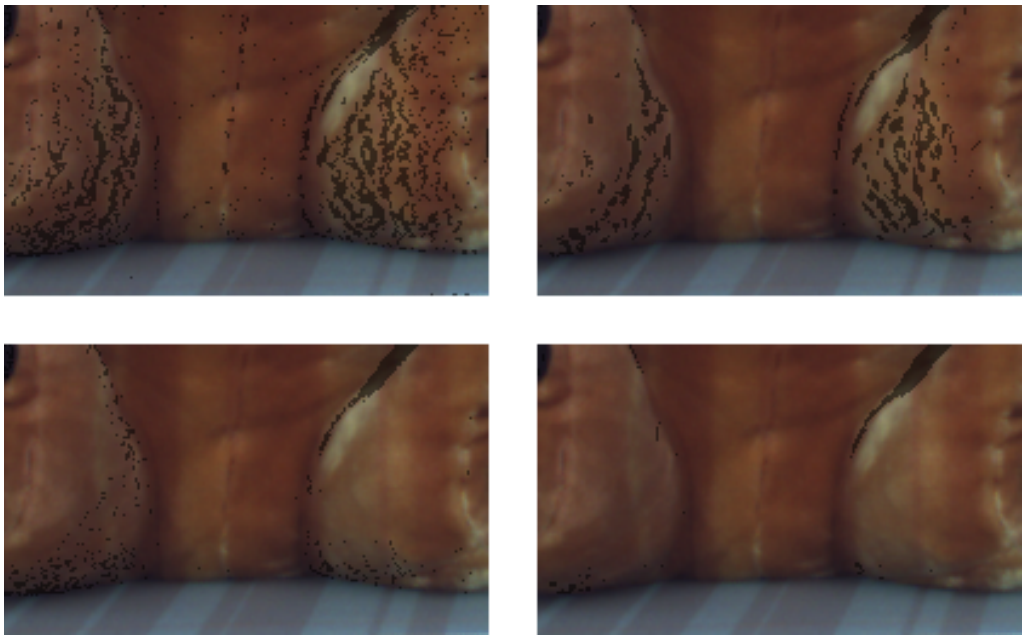


Fig. 5.3: dettaglio della scena, con evidenziazione delle occlusioni al termine delle elaborazioni. In alto a sinistra: dopo elaborazione di 200 immagini (100 con proiettore e luce ambientale e 100 con il solo proiettore), senza aggregazione dei costi. In alto a destra: dopo elaborazione di 200 immagini come nel caso precedente, supporto 3x3. In basso a sinistra: dopo elaborazione di 100 immagini (con il solo proiettore), senza aggregazione dei costi. In basso a destra: dopo elaborazione di 100 immagini come nel caso precedente, supporto 3x3.

In ogni caso la diversa qualità dei dataset potrebbe spiegare anche il fatto che l'elaborazione dell'insieme delle 200 coppie non sia in grado di apportare un significativo miglioramento alla mappa ottenuta con le cento coppie che si avvalgono del solo proiettore(figura 5.3). L'elaborazione in sequenza delle due serie porta a risultati dipendenti dall'ordine dei due dataset: in un caso

peggiori e nell'altro praticamente identici (0,013% pixel occlusi in meno) a quelli ottenuti con il solo proiettore. Per questa ragione si è preferito utilizzare la sola sequenza di cui la figura 5.2 è un esempio. Essendo però l'esposizione ottimizzata per sfruttare appieno la gamma di luminosità nel caso di due sorgenti luminose, quelle con il solo proiettore sono esposte in modo tale da sfruttare al massimo metà dei 256 livelli possibili, perdendo circa un bit per ciascun canale.

5.2 Numero di pixel negativi al cross-checking

Le figure 5.4 e 5.5 mostrano l'andamento del numero di pixel che risultano negativi al controllo incrociato tra le due mappe, e che sono quindi formalmente occlusi. Si osserva una rapida riduzione nel corso dell'elaborazione delle prime 5-6 coppie, seguito da un calo più graduale, che da un certo punto in avanti (circa 20 nel caso di supporto 3x3, 40 nel caso di costi non aggregati) porta a miglioramenti estremamente lenti, che sembrano arrestarsi del tutto entro le 80 coppie elaborate. L'aggregazione su un supporto 5x5 fornisce con buona approssimazione il valore limite delle occlusioni, al di sotto del quale non è possibile scendere, trattandosi di pixel effettivamente occlusi dalla geometria della scena. Il numero di pixel occlusi dopo la prima coppia diminuisce con l'allargarsi del supporto, ma a scapito della qualità del risultato. Per avere una percentuale di pixel occlusi del 5-10% dopo una sola coppia si deve utilizzare un supporto 21x21, con inaccettabili riduzione della risoluzione nelle coordinate (u, v) ed errata localizzazione delle occlusioni.

La figura 5.6 mostra dove sono localizzati i pixel occlusi, e la mappa di disparità ottenuta nei tre casi. Ad un primo esame risulta che l'assenza della fase di aggregazione dei costi determina una maggiore precisione in corrispondenza delle variazioni della disparità (come già si era visto per il metodo fixed windows in generale), mentre un supporto 3x3 riduce il numero di occlusioni quasi fino al limite minimo. Un ulteriore ampliamento del supporto deteriora la mappa senza diminuire significativamente il numero di occlusioni.

Un esame delle immagini di figura 5.6 mostra la presenza di due zone anomale nelle coppie binoculari (figura 5.7). Pur trattandosi di aree relativamente piccole si è ripetuta l'elaborazione su una parte della scena che le escludesse. I risultati (figura 5.8) ricalcano qualitativamente lo stesso andamento già osservato.

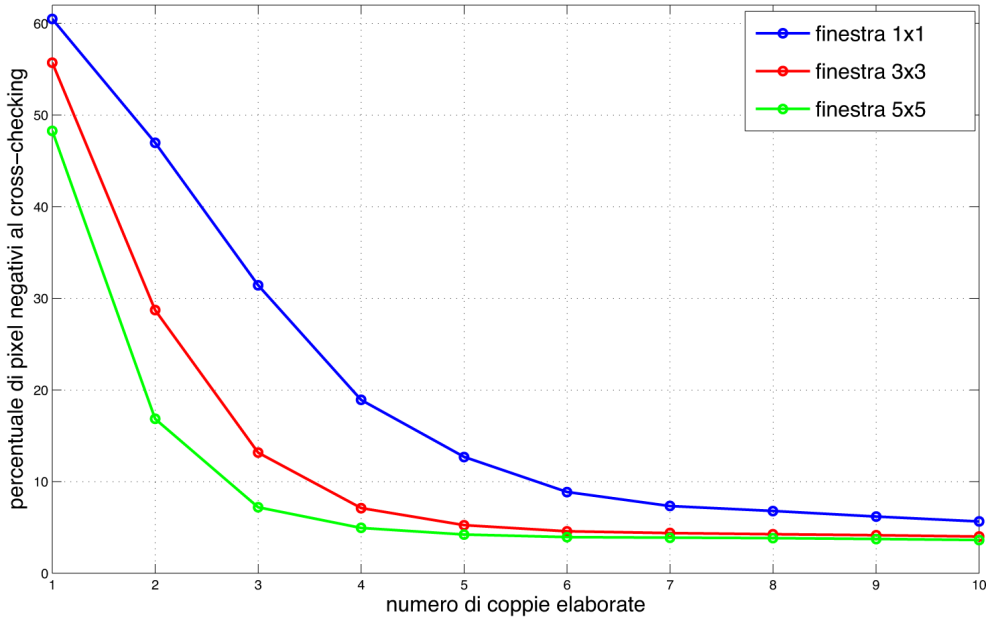


Fig. 5.4: numero di pixel negativi al controllo incrociato tra le due mappe ($d \neq d'$), in funzione del numero di coppie elaborate (da 1 a 10).

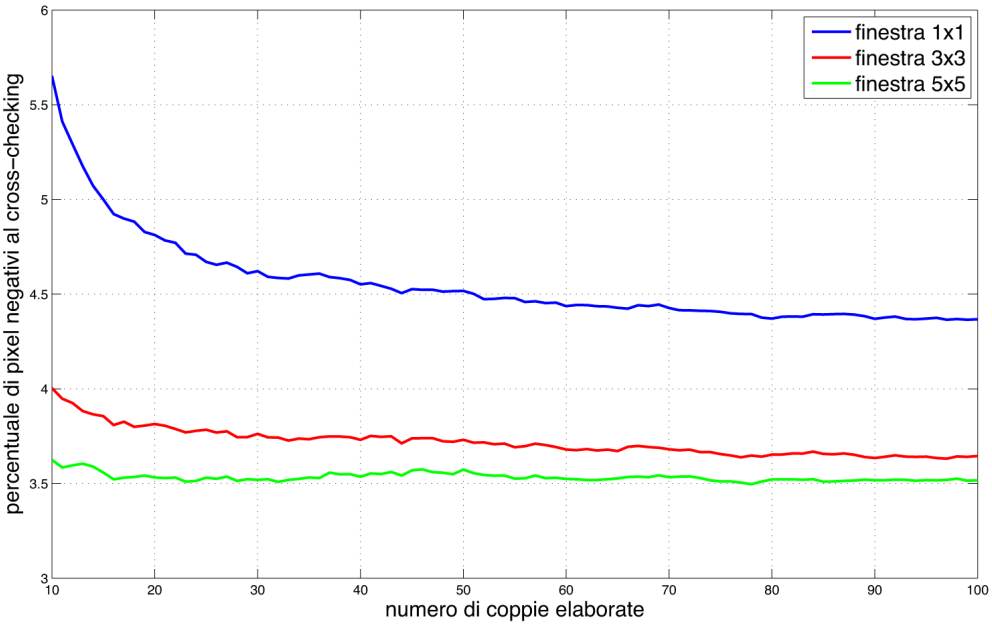


Fig. 5.5: numero di pixel negativi al controllo incrociato tra le due mappe ($d \neq d'$), in funzione del numero di coppie elaborate (da 10 a 100).

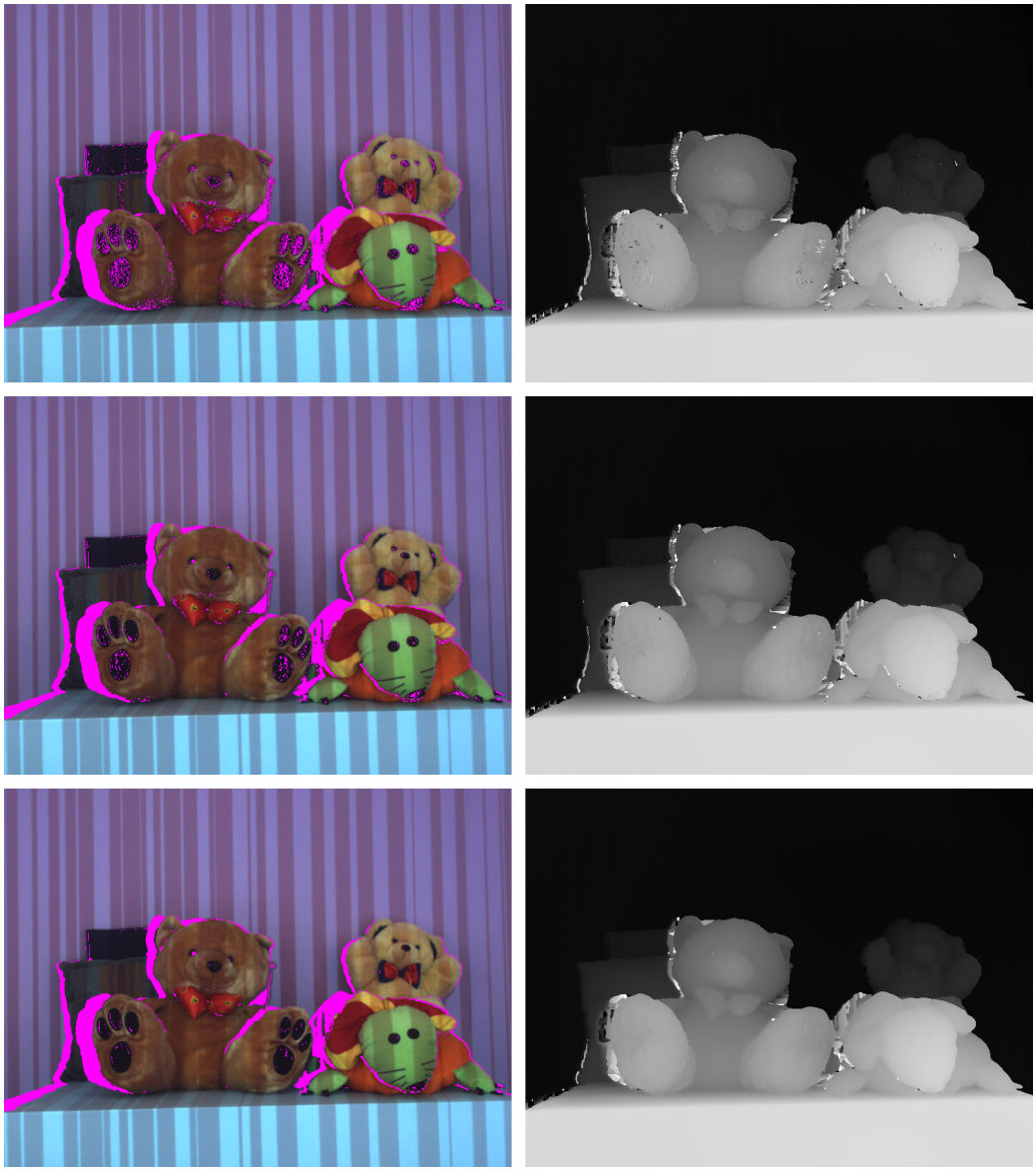


Fig. 5.6: a sinistra sono evidenziati nell'immagine di riferimento (che in questo è quella sinistra della coppia binoculare) i pixel occlusi nell'altra. A destra la mappa di disparità finale. Le immagini corrispondono rispettivamente, dall'alto in basso, a costi non aggregati e ad aggregazione dei costi su supporti di dimensioni 3x3 e 5x5.

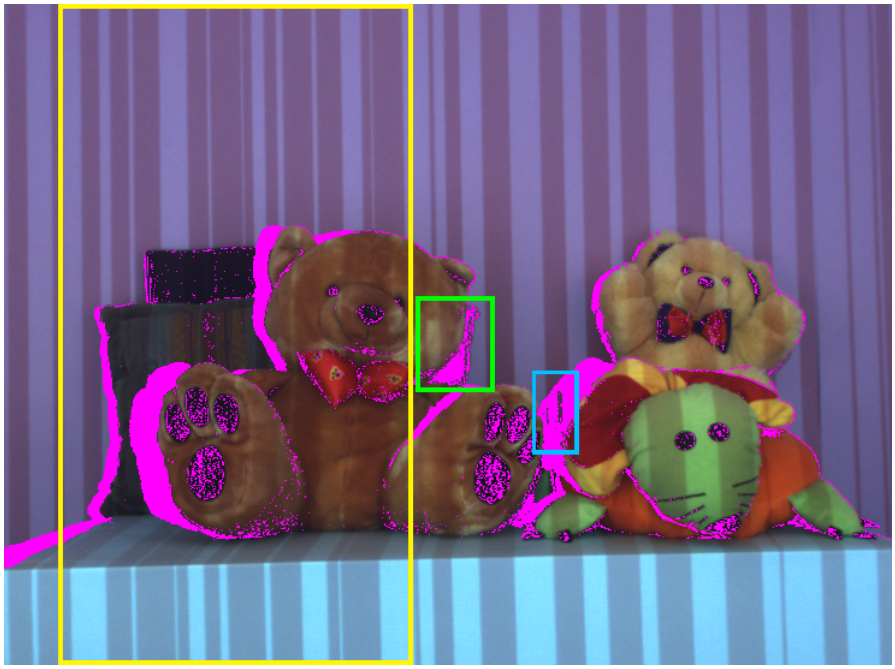


Fig. 5.7: anomalie nel dataset. L'area nella cornice verde non è occlusa, ma è in ombra rispetto al pattern proiettato e quindi praticamente al buio nel dataset che utilizza la sola luce del proiettore. L'area nella cornice celeste è occlusa nell'immagine di destra della coppia, ma in alcuni punti non risulta esserlo perchè l'algoritmo trova corrispondenze a basso costo in altri punti della parete (che sono invece visibili nell'immagine di destra e non in quella di sinistra). L'area nella cornice gialla è stata considerata per ripetere l'elaborazione, in modo da depurare la valutazione dello space-time stereo da queste anomalie.

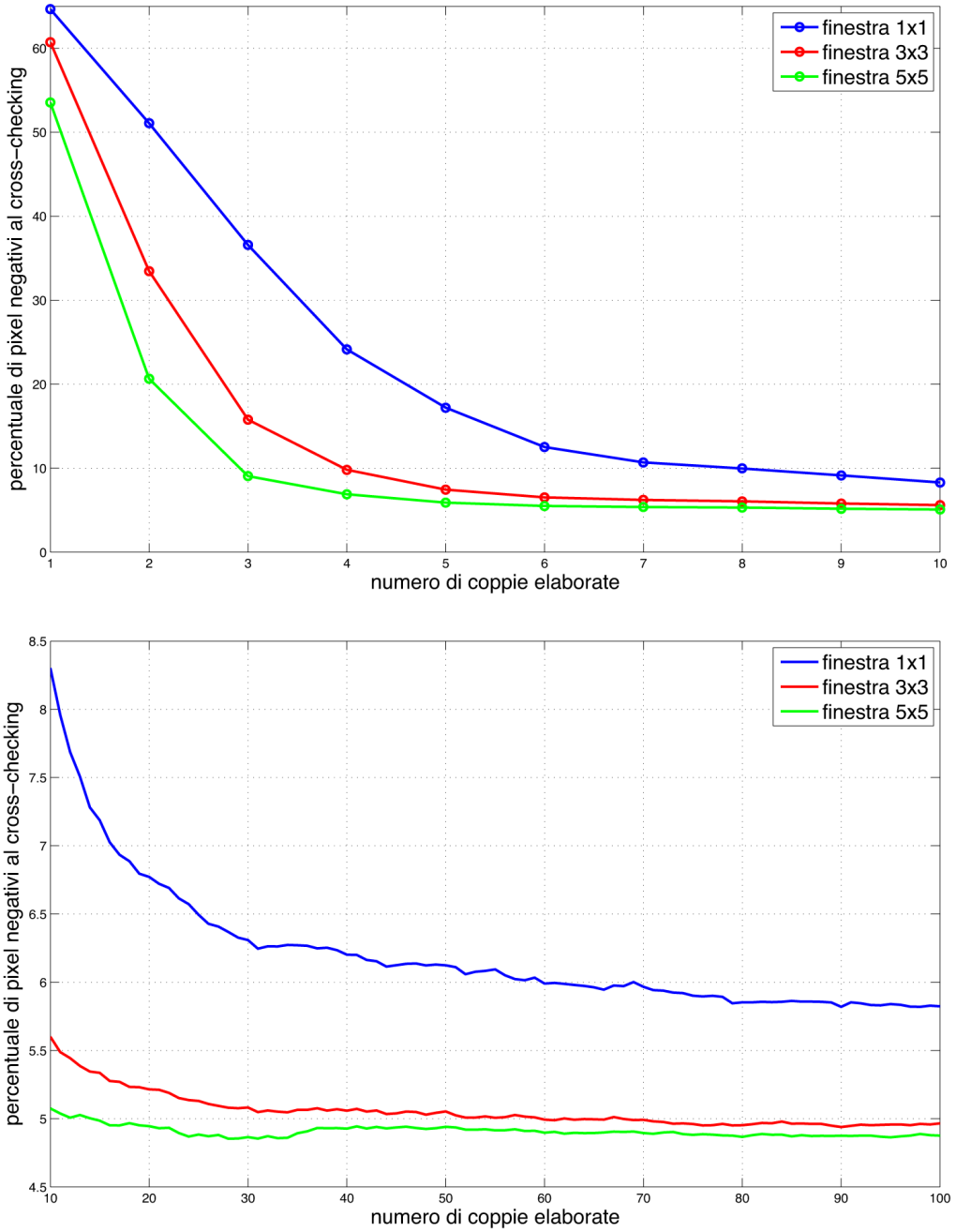


Fig. 5.8: numero di pixel negativi al controllo incrociato tra le due mappe ($d \neq d'$), in funzione del numero di coppie elaborate (da 1 a 10 in alto, da 10 a 100 in basso), per l'area nella cornice gialla di figura 5.7.

5.3 Valutazione della ricostruzione di una geometria nota

L'elaborazione di un dataset relativo a una scena in cui sia presente un piano (figura 5.9) permette di valutare, seppur limitatamente a condizioni particolari, le prestazioni in termini assoluti.

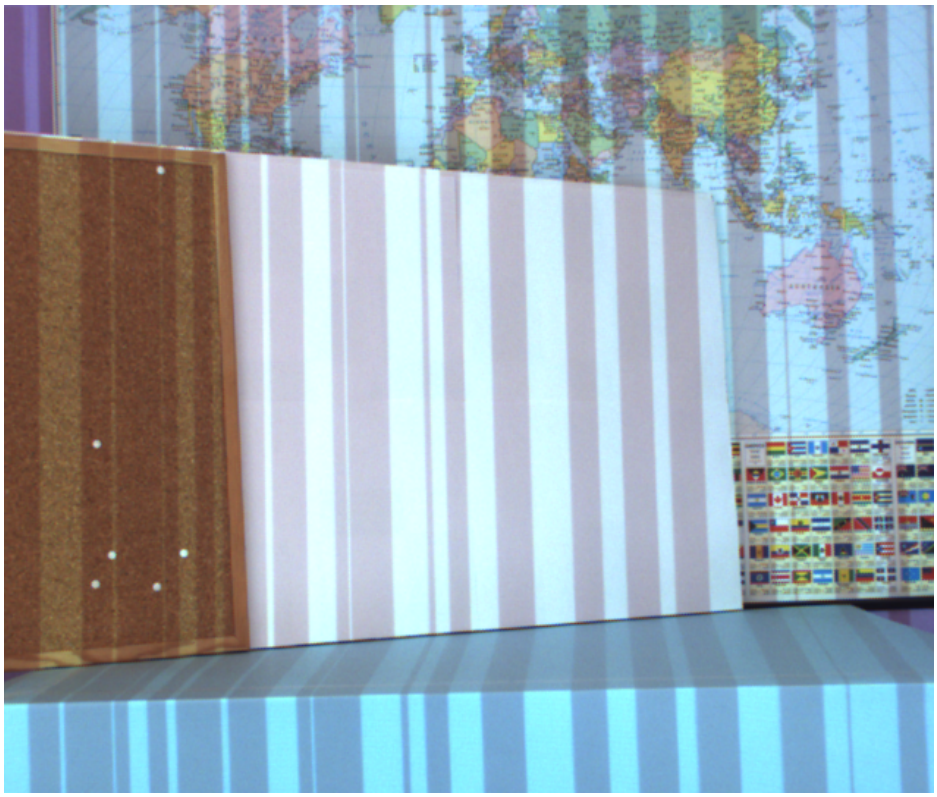


Fig. 5.9: la ricostruzione della geometria del piano al centro dell'immagine permette una valutazione della precisione dell'algoritmo

Campionando con un passo di 5 pixel (in entrambe le direzioni) i valori di disparità nell'area selezionata, e calcolando le coordinate dei punti a partire dai parametri che descrivono il sistema di acquisizione

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = K^{-1} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} z_i$$

si ricava tramite regressione lineare l'equazione del piano che costituisce la miglior approssimazione dell'insieme di punti. La distanza media tra i punti

ricostruiti e quelli del piano calcolato fornisce una misura della precisione della ricostruzione. L'equazione del piano può essere ricavata dalla scomposizione SVD dei dati centrati nell'origine. Con la sintassi di MATLab:

```
m_x=mean(x);
m_y=mean(y);
m_z=mean(z);
[U,S,V]=svd([x-m_x,y-m_y,z-m_z],0);
```

L'ultimo autovettore (l'ultima colonna) di V fornisce i coefficienti dell'equazione del piano che risolve il problema dei minimi quadrati:

$$V_{1,3}(x - m_x) + V_{2,3}(y - m_y) + V_{3,3}(z - m_z) = 0$$

Esplicitando rispetto a z si ricava l'espressione per ottenere i valori stimati, rispetto ai quali calcolare l'errore:

```
N=-V(:,3)/V(3,3);
a=N(1);
b=N(2);
c=-[m_x m_y m_z]*N;
z_p=a*x+b*y+c;
avg_err=mean(abs(z_p-z));
```

Gli errori medi calcolati sono di circa 1 mm utilizzando disparità con precisione subpixel (con un leggero miglioramento nel caso di costi aggregati su un supporto 3x3) e 4,5 mm con disparità intere. In figura 5.10 la visualizzazione dei valori calcolati lungo una scanline, cioè nell'intersezione con un piano orizzontale. Tenendo presente la relazione 1.4, che esprime la risoluzione nel caso si utilizzino disparità intere, l'errore medio commesso in corrispondenza di un tratto di piano vale

$$err = \frac{z^2}{4bf}$$

in accordo con le prove effettuate (nelle quali $z \simeq$1,65 m; b =0,176878 m; f =852,567). A parità di geometria della scena e di lunghezza focale, l'uso di interpolazione subpixel porta dunque – nel caso considerato – a un errore equivalente a quello che si sarebbe ottenuto aumentando la baseline di un fattore pari al rapporto tra gli errori nei due casi (circa 4), senza però aumentare la distorsione prospettica e il numero di pixel occlusi.

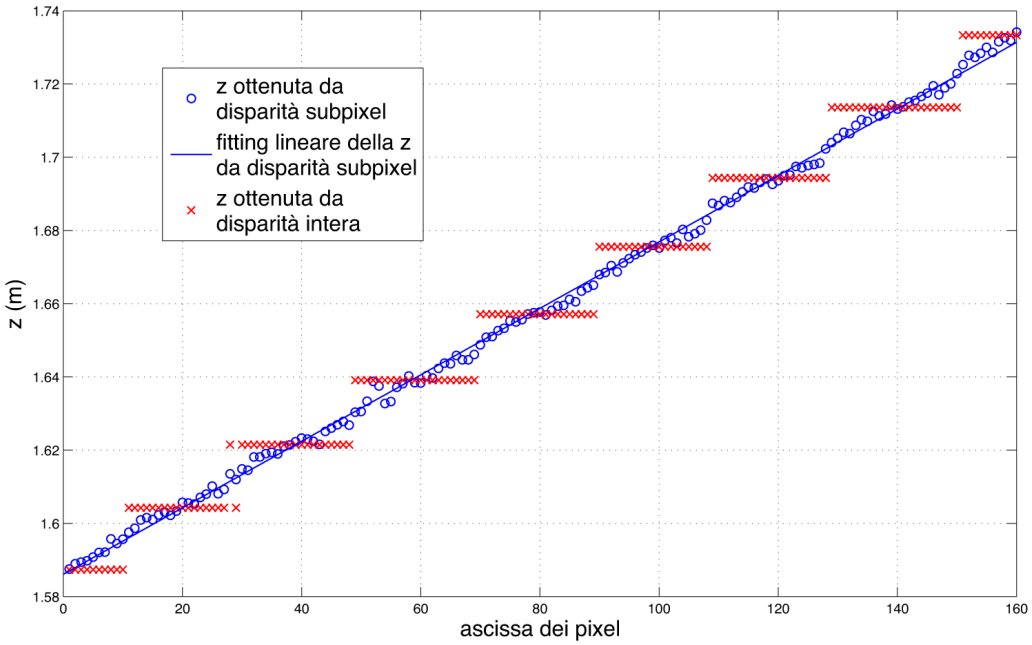


Fig. 5.10: confronto tra i valori di z ricostruiti con disparità subpixel e con disparità intera (senza aggregazione dei costi) per un tratto di scanline interno al piano di figura 5.9.

5.4 Evoluzione della mappa verso il valore finale

L'andamento dei risultati nel corso dell'elaborazione permette di osservare diverse velocità di convergenza, dipendenti dalla superficie considerata.

Considerando ad esempio un'area estesa della parete di fondo, il numero di pixel che risultano occlusi convergono a zero rapidamente (dopo 12 iterazioni in assenza di aggregazione e 4 iterazioni per il supporto 3x3). La distanza media dai valori finali scende sotto il quarto di pixel dopo 5-7 coppie di immagini, come avviene anche per la mappa globalmente considerata. Per altre aree il numero di coppie necessarie è superiore: per la regione nella cornice verde di figura 5.11 sono ad esempio necessarie 20 o 40 coppie (a seconda della dimensione del supporto) con permanenza di pixel occlusi nel caso di supporto coincidente con un pixel (figura 5.6). Dal momento che la convergenza è favorita dall'illuminazione del proiettore, è naturale che essa sia più lenta dove il pattern è meno visibile, come la superficie opaca molto

scura, come quella considerata.

Fig. 5.11: la velocità dell'evoluzione verso il valore finale in aree diverse appare conseguenza dell'attitudine di una superficie a riflettere il pattern: rapida nell'area dentro la cornice gialla, più lenta nell'area dentro la cornice rossa e molto lenta (con permanenza di occlusioni) nell'area nella cornice verde.

Lo stesso fenomeno si può riscontrare sulle superfici la cui normale forma un angolo ϕ prossimo a $\pi/2$, rispetto al fascio proiettato, dal momento che l'intensità risulta ridotta di un fattore $cos\phi$. In questo caso è concomitante l'effetto della distorsione prospettica, che riduce la somiglianza tra le aree (15-20 iterazioni per scendere al di sotto di un quarto di pixel dal valore finale nella regione dentro alla cornice verde).

5.5 Conclusioni

L'algoritmo realizzato risulta in grado di produrre mappe di disparità ad elevata risoluzione nelle coordinate dell'immagine, grazie alla valutazione su supporti piccoli delle somiglianze tra pixel. L'aggregazione dei costi lungo la dimensione temporale garantisce comunque un basso numero di pixel occlusi, riducendo il compromesso tra numero di occlusioni e risoluzione alla scelta tra

i soli supporti di aggregazione di dimensione 1 pixel o 3x3 pixel. In aggiunta l'interpolazione subpixel della disparità determina un miglioramento della risoluzione anche lungo l'asse z.

Bibliografia

[1] Richard Szeliski. Computer Vision: Algorithms and Applications. September 3, 2010 draft, 2010 Springer.

[2] Boguslaw Cyganek e J. Paul Siebert. An Introduction to 3D Computer Vision Techniques and Algorithms. 2009 John Wiley & Sons, Ltd.

[3] Andrea Fusiello. Visione Computazionale. Appunti delle lezioni. 2009.

[4] Carsten Steger, Markus Ulrich e Christian Wiedemann. Machine Vision Algorithms and Applications. 2008, Wiley-VCH.

[5] Gary Bradski e Adrian Kaehler. Learning OpenCV. 2008 O Reilly Media, Inc.

[6] Brian Curless e Marc Levoy. Better Optical Triangulation through Spacetime Analysis. IEEE Int't Conference on Computer Vision '95, pp. 987-994, June, 1995.

[7] Li Zhang, Brian Curless e Steven M. Seitz. Spacetime Stereo: Shape Recovery for Dynamic Scenes. Proc. Computer Vision and Pattern Recognition, 2003.

[8] James Davis, Diego Nehab, Ravi Ramamoorthi e Szymon Rusinkiewicz. Spacetime Stereo: A Unifying Framework for Depth from Triangulation. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 2, February 2005.

[9] Daniel Scharstein e Richard Szeliski. High-Accuracy Stereo Depth Maps Using Structured Light. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), volume 1, pages 195 202, Madison, WI, June 2003.

[10] Daniel Scharstein e Richard Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms.

[11] Federico Tombari, Stefano Mattoccia, Luigi Di Stefano e Elisa Addimanda. Classification and evaluation of cost aggregation methods for stereo correspondence.

[12] Kuk-Jin Yoon e In So Kweon. Adaptive Support-Weight Approach for Correspondence Search. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 4, April 2006.

Ringraziamenti

Si ringraziano il professor Stefano Mattoccia dell'Università di Bologna, per aver messo a disposizione il codice per calcolare il DSI, e Carlo Dal Mutto dell'Università di Padova, per le indicazioni pazientemente fornite.