

Visione Computazionale

Appunti delle lezioni

Andrea Fusiello

<http://profsci.univr.it/~fusiello>

© Copyright 2008,2009 Andrea Fusiello

Quest'opera è pubblicata con licenza  Creative Commons    Attribuzione-NonCommerciale-Condividi allo stesso modo. Per visionare una copia di questa licenza visita <http://creativecommons.org/licenses/by-nc-sa/2.0/deed.it>.

Revisione 3

Indice

<i>Prefazione</i>	<i>pagina ix</i>
1 Introduzione	1
2 Recupero della forma da immagini	4
2.1 Introduzione	4
2.2 Metodi ottici	5
2.3 Immagini 2.5D	7
2.4 Dalla misura al modello 3D	8
2.4.1 Registrazione	9
2.4.2 Fusione geometrica	10
2.5 Altri metodi	12
2.5.1 Frange di interferenza di Moiré	13
2.5.2 Tempo di volo	13
3 Formazione dell'immagine	16
3.1 Geometria della formazione dell'immagine	16
3.2 Lenti sottili	18
3.3 Radiometria della formazione dell'immagine	21
3.4 Immagini digitali	22
4 Calibrazione della fotocamera	24
4.1 Modello della fotocamera	24
4.1.1 Modello semplificato	25
4.1.2 Modello generale	27
4.1.3 Proprietà	31
4.2 Calibrazione	33
4.2.1 Metodo DLT	33
4.2.2 Metodo non lineare	35
4.2.3 Estrazione dei parametri	36
4.2.4 Distorsione radiale	36

5	Chiaroscuro, tessitura, sfocamento	42
5.1	Chiaroscuro	42
5.1.1	Algoritmi di stima della forma dal chiaroscuro	46
5.1.2	Stima della direzione di illuminazione	48
5.1.3	Stereo fotometrico	50
5.2	Tessitura	50
5.2.1	Orientazione del piano da tessitura	51
5.3	Fuoco	53
5.3.1	Fochettatura	53
5.3.2	Sfocamento	54
6	Stereopsi	58
6.1	Introduzione	58
6.2	Triangolazione 3D	59
6.2.1	Metodo linear-eigen	60
6.3	Geometria epipolare	62
6.4	Rettificazione epipolare	65
6.4.1	Rettificazione delle MPP	66
6.4.2	La trasformazione di rettificazione	68
7	Calcolo delle corrispondenze	72
7.1	Introduzione	72
7.1.1	Problemi	72
7.1.2	Vincoli	73
7.1.3	Metodi locali e globali	74
7.2	Metodi di accoppiamento locali	74
7.2.1	Accoppiamento di finestre	74
7.2.2	Compromesso affidabilità - accuratezza	79
7.2.3	Indicatori di affidabilità	81
7.2.4	Occlusioni	82
7.2.5	Altri metodi locali	83
7.3	Metodi di accoppiamento globali	84
7.3.1	Spazio delle corrispondenze	85
7.4	Classificazione dei metodi	86
7.5	Illuminazione strutturata	87
7.5.1	Triangolazione attiva	89
8	Ricostruzione volumetrica	94
8.1	Ricostruzione da sagome	94
8.1.1	Algoritmo di Szeliský	96
8.2	Ricostruzione da foto-coerenza	97
8.2.1	Voxel coloring	99

8.2.2	Space carving	100
8.2.3	Ottimizzazione della ricostruzione	102
9	Moto e struttura	105
9.1	Introduzione	105
9.2	Matrice essenziale	106
9.2.1	Fattorizzazione della matrice essenziale	108
9.2.2	Calcolo della matrice essenziale	109
9.3	Estensione a molte viste	111
9.4	Estrazione di punti salienti	113
9.4.1	Metodo di Harris e Stephens	113
9.5	Corrispondenze di punti salienti	116
10	Flusso ottico	121
10.1	Il campo di moto	121
10.1.1	Analisi del campo di moto	124
10.2	Il flusso ottico	127
10.2.1	Calcolo del flusso ottico	128
10.3	Algoritmo di tracciamento KLT	130
11	Orientazione	136
11.1	Caso 2D-2D: orientazione relativa	137
11.1.1	Metodo iterativo di Horn	137
11.2	Caso 3D-3D: orientazione assoluta	139
11.2.1	Metodo con SVD	139
11.2.2	ICP	140
11.3	Caso 2D-3D: orientazione esterna	143
11.3.1	Metodo lineare di Fiore	143
11.3.2	Metodo non lineare di Lowe	145
11.3.3	Metodo diretto	148
12	Ricostruzione non calibrata	152
12.1	Ricostruzione proiettiva e promozione euclidea	153
12.1.1	Ricostruzione proiettiva	153
12.1.2	Promozione euclidea	155
12.2	Autocalibrazione	157
12.2.1	Matrice fondamentale	157
12.2.2	Metodo di Mendonça e Cipolla	160
12.2.3	Ricostruzione incrementale	161
12.3	Fattorizzazione di Tomasi-Kanade	162
13	Scena planare: omografie	167
13.1	Omografia indotta da un piano	167
13.1.1	Calcolo dell'omografia (DLT)	170

13.1.2	Omografie compatibili	171
13.2	Parallasse	172
13.3	Calibrazione planare	174
13.4	Moto e struttura da omografia calibrata	176
14	Mosaicitura e sintesi di immagini	179
14.1	Mosaici	179
14.1.1	Allineamento	181
14.1.2	Trasformazione geometrica	182
14.1.3	Miscelazione	185
14.2	Altre applicazioni	186
14.2.1	Stabilizzazione dell'immagine	186
14.2.2	Rettificazione ortogonale	186
14.3	Sintesi di immagini	187
14.3.1	Trasferimento con la profondità.	188
14.3.2	Interpolazione con disparità	188
14.3.3	Trasferimento epipolare.	189
14.3.4	Trasferimento con parallasse	190
14.3.5	Trasformazione delle immagini	192
<i>Appendice 1</i>	<i>Nozioni di Algebra lineare</i>	195
<i>Appendice 2</i>	<i>Nozioni di Geometria proiettiva</i>	212
<i>Appendice 3</i>	<i>Miscellanea di nozioni utili</i>	219
<i>Indice Analitico</i>		234

A mio padre

Prefazione

Di solito, ci si convince meglio con le ragioni trovate da sè stessi che non con quelle venute in mente ad altri.

B. Pascal

In questi appunti di Visione computazionale ho raccolto diverso materiale che ho insegnato negli ultimi dieci anni in corsi istituzionali o interventi seminariali. La scelta degli argomenti riflette una visione personale della disciplina e certo sono presenti omissioni alle quali cercherò di porre rimedio nelle future edizioni. Accanto ad argomenti “classici” (come *shape from shading*), vengono presentati anche i risultati più recenti nel campo della ricostruzione geometrica. La trattazione che ho cercato di privilegiare è una visita in profondità dei metodi della Visione computazionale, piuttosto che una visita in ampiezza. Questo vuol dire che questi appunti non hanno la presunzione di fornire una panoramica sui metodi esistenti per risolvere un dato problema ma ne sono stati selezionati alcuni e descritti ad un livello tale da consentirne l’implementazione. A questo proposito, il testo è corredato da una collezione di funzioni MATLAB che implementano i metodi trattati, scaricabile da <http://prof.sci.univr.it/~fusiello/visione/appunti>. Dallo stesso sito è anche possibile scaricare liberamente la versione elettronica di questo testo.

I prerequisiti sono le nozioni di Algebra lineare che vengono richiamate in appendice 1, con la maggior parte delle quali lo studente dovrebbe comunque essere già familiare. La parte di Geometria proiettiva riportata in appendice 2 invece tipicamente non fa parte del bagaglio dello studente di Informatica che si accosta al corso e viene trattata nelle prime lezioni. Altre nozioni di Calcolo numerico e Statistica possono essere brevemente trattate durante il corso quando serve. Il materiale raccolto in questo volume è sovabbondante rispetto ad un tipico corso di un

quadrimestre. Il docente può scegliere un percorso maggiormente focalizzato sulla geometria, omettendo i Capitoli 6, 5 8 e 10, oppure seguire un approccio più classico e generale, omettendo i Capitoli 11, 12 e 13.

Gli appunti nascono, in forma embrionale, nel 1996 e si sono poi evoluti ed ampliati fino alla versione attuale. Ringrazio gli studenti dell'Università di Udine e dell'Università di Verona che, lungo questi anni, hanno segnalato errori, mancanze, e parti poco chiare. Alcuni di essi hanno contribuito con immagini, illustrazioni o porzioni di testo tratte dalle tesi di laurea. Li cito in ordine cronologico: Tiziano Tommasini, Luca da Col, Alberto Censi, Sara Ceglie, Nicola Possato, Massimo Sabbadini, Michele Aprile, Roberto Marzotto, Alessandro Negrente, Alberto Sanson, Andrea Colombari, Michela Farenzena, Riccardo Gherardi, Chiara Zanini. Germana Olivieri ha tradotto in italiano alcune parti che erano originariamente in inglese. La terza revisione ha beneficiato delle puntuali correzioni suggerite da Guido Maria Cortelazzo, Riccardo Gherardi e Samuele Martelli, che ringrazio sentitamente.

Le fonti bibliografiche sono molteplici, e sarebbe impossibile individuare il contributo di ciascuna. Quelle da cui ho tratto maggiore ispirazione sono [Trucco e Verri, 1998] e [Faugeras, 1993].

I crediti per le figure prese dal web o da articoli scientifici sono riconosciuti nella didascalia.

Udine, 30 settembre 2009

Andrea Fusiello

1

Introduzione

Tra tutte le abilità sensoriali, la visione è largamente riconosciuta come quella con le maggiori potenzialità. Le capacità dei sistemi biologici sono formidabili: l'occhio raccoglie una banda di radiazioni elettromagnetiche rimbalzate su diverse superfici e provenienti da fonti luminose diverse ed il cervello elabora questa informazione formando il quadro della scena come noi la percepiamo.

Se volessimo dare una definizione, potremmo dire che la **Visione computazionale** o *Computer Vision*, si occupa della *analisi* di immagini con il calcolatore. L'analisi è finalizzata a scoprire *cosa* è presente nella scena e *dove*.

La Visione computazionale non tenta di replicare la visione umana. Il tentativo è destinato al fallimento per la intrinseca differenza tra i due *hardware*. Si pensi ai tentativi di replicare il volo animale, miseramente falliti, ed a come invece gli aerei abbiano efficacemente risolto il problema.

Seguendo [Ullman, 1996] si usa distinguere tra Visione computazionale di **basso livello** e di alto livello. La prima si occupa di estrarre determinate proprietà fisiche dell'ambiente visibile, come profondità, forma tridimensionale, contorni degli oggetti. I processi di visione di basso livello sono tipicamente paralleli, spazialmente uniformi e relativamente indipendenti dal problema e dalla conoscenza a priori associata a particolari oggetti.

Viceversa, la visione di **alto livello** si occupa della estrazione delle proprietà delle forme, e di relazioni spaziali, di riconoscimento e classificazione di oggetti. I processi di alto livello sono di solito applicati ad una porzione dell'immagine, dipendono dall'obbiettivo della computazione e dalla conoscenza a priori associata agli oggetti.

Anche tralasciando i problemi di alto livello legati alla percezione ed al

riconoscimento di oggetti, il solo compito di ricostruire la pura struttura geometrica della scena (appartenente al basso livello) è difficile.

Tale compito può essere efficacemente descritto come **l'inverso della grafica al calcolatore**, nella quale, dati:

- descrizione geometrica della scena
- descrizione radiometrica della scena (fonti luminose e proprietà delle superfici)
- descrizione completa dell'apparato di acquisizione (fotocamera)

il calcolatore produce l'immagine “sintetica” vista dalla fotocamera.

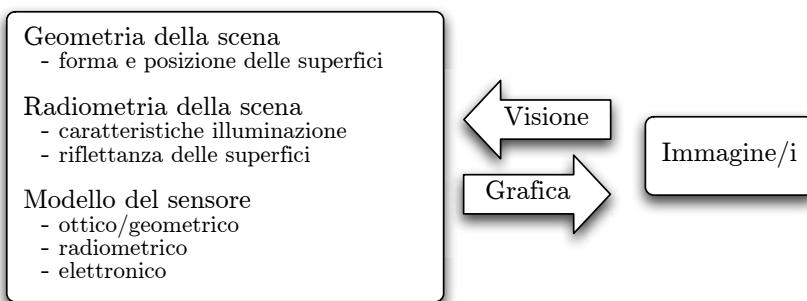


Fig. 1.1. Relazione tra Visione e Grafica.

Evidentemente la riduzione dimensionale operata dalla proiezione (geometria) e la molteplicità delle cause che concorrono a determinare la luminosità ed il colore (radiometria) rendono il problema inverso **sotto-vincolato** (la soluzione non è unica).

In questo breve ciclo di lezioni, vedremo in che modo si è affrontato finora il problema della ricostruzione tridimensionale. L'approccio che prenderemo sarà quello del **ricostruzionismo**, la teoria sviluppata da Marr alla fine degli anni settanta [Marr, 1982]. Essa inquadra il problema della Visione computazionale sulla base della considerazione che un sistema visivo (naturale o artificiale) è un sistema per il trattamento della informazione. Molto succintamente, una volta definite le assunzioni fisiche di base imposte dall'ambiente osservato e assunte le leggi generali del comportamento del sistema osservante, si tratta di individuare un insieme di processi computazionali che consentano di passare da una classe di rappresentazioni pittoriche dell'ambiente ad una sua rappresentazione tridimensionale. In sintesi, lo scopo della Visione computazionale – nel paradigma ricostruzionista — è la produzione

di una **descrizione del mondo completa ed accurata**, impiegante **primitive tridimensionali**.

Questo, naturalmente, non è il solo approccio. Alcuni contestano questo modo di porre il problema [Aloimonos e Shulman, 1989] e sostengono che la descrizione del mondo non debba essere generale, ma dipendente dall'obiettivo (*purposivism*).

Senza entrare nel merito della diatriba, ci limitiamo ad osservare che l'approccio ricostruzionista, per la sua generalità, meglio si presta ad essere oggetto di studio e di insegnamento. L'approccio *purposivist*, proprio per la sua filosofia, è *problem-driven*, e difficilmente si presta a generalizzazioni: che cosa ci insegna di generale, di riusabile, la soluzione ad-hoc (per quanto brillante) di un particolare problema?

Bibliografia

- Aloimonos J.; Shulman D. (1989). Integration of visual modules, an extension to the marr paradigm. *Academic press*.
- Faugeras O. (1993). *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, MA.
- Marr D. (1982). *Vision*. Freeman, San Francisco, CA.
- Trucco E.; Verri A. (1998). *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall.
- Ullman D. (1996). *High-level Vision*. The MIT Press.

2

Recupero della forma da immagini

Come accennato nell'introduzione, questo corso si concentra sullo studio di tecniche computazionali per stimare le proprietà geometriche (la struttura) del mondo tridimensionale (3D) a partire da proiezioni bidimensionali (2D) di esso: le immagini. In questo capitolo introduciamo il **problema della acquisizione della forma** (ovvero *shape/model acquisition, image-based modeling, 3D photography*) che sarà il tema conduttore del corso, e descriviamo sommariamente i passi che è necessario effettuare *a valle* della misura 3D per ottenere veri e propri modelli degli oggetti. In un certo senso partiamo dal fondo, mostrando come impiegare le misure che le tecniche di Visione Computazionale forniscono. La motivazione è duplice. Da un lato c'è la volontà di inquadrare subito i temi che tratteremo all'interno di una cornice di riferimento “comprendibile”, dall'altro l'opportunità pratica di collegare questi argomenti a quelli studiati in un tipico corso di Grafica [Scateni *e al.*, 2005], in particolare il tema della modellazione. Sul tema della acquisizione di modelli si veda [Bernardini e Rushmeier, 2002].

2.1 Introduzione

Esistono molti metodi per l'acquisizione automatica della forma di un oggetto. Una possibile tassonomia dei metodi di acquisizione della forma (*shape acquisition*) è illustrata in figura 2.2.

Noi ci occuperemo di quelli ottici riflessivi[†], visto che ci interessa la Visione. Tra i vantaggi di queste tecniche citiamo il fatto di non richiedere contatto, la velocità e la economicità. Le limitazioni includono il fatto di poter acquisire solo la parte visibile delle superfici e la sensibilità alle proprietà delle superfici come trasparenza, brillantezza, colore.

[†] Esistono anche apparati di imaging *trasmissivo*, di cui non ci occuperemo.



Fig. 2.1. Apparecchiatura per l'acquisizione stereoscopica.

Il problema che affrontiamo, noto come *image-based modeling* o *3D photography* si pone nel modo seguente: gli oggetti irradiano luce visibile; la fotocamera cattura questa luce, le cui caratteristiche dipendono dalla illuminazione della scena, geometria della superficie, riflettanza della superficie, la quale viene analizzata al calcolatore con lo scopo di inferire la struttura 3D degli oggetti

La distinzione fondamentale tra le tecniche ottiche riguarda l'impiego o meno di sorgenti di illuminazione speciali. In particolare, distinguiamo i **metodi attivi**, che irradiano la scena con radiazioni elettromagnetiche opportune (pattern luminosi, luce laser, radiazioni infrarosse etc...), ed i **metodi passivi**, che si basano sull'analisi di immagini (di colore) della scena così com'è. I primi hanno il vantaggio di raggiungere risoluzioni elevate, ma sono più costosi e non sempre applicabili. I secondi sono economici, pongono meno vincoli per l'applicabilità ma hanno risoluzioni più basse.

2.2 Metodi ottici

Il sistema visivo umano sfrutta molteplici “indizi visivi” (*visual cues*) per recuperare la profondità delle superfici della scena, informazione che viene persa nel processo di proiezione sulla retina (bidimensionale). Alcuni di questi sono:

- lo sfocamento
- il parallasse (o disparità)
- il chiaroscuro (*shading*),

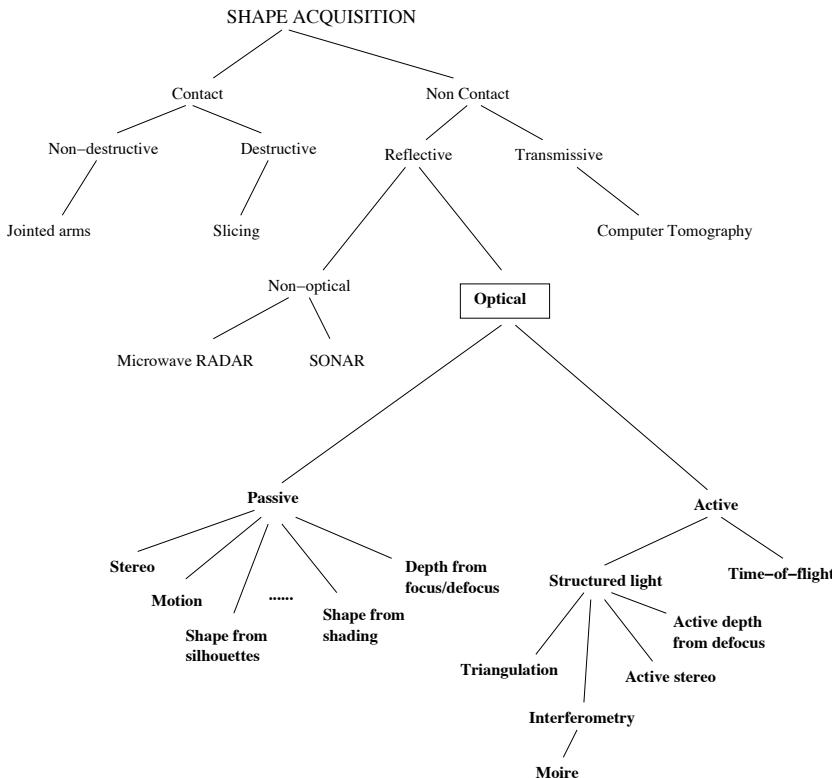


Fig. 2.2. Tassonomia dei sistemi di acquisizione della forma (ripresa da [Curless, 2000]).

- le tessiture.

Le tecniche computazionali che vedremo in questo corso, suddivise tra attive e passive, sfruttano questi indizi ed altri fenomeni ottici per recuperare la forma degli oggetti dalle immagini. Elenchiamo di seguito i metodi che studieremo in maggiore dettaglio nel corso, suddivisi tra attivi e passivi.

Metodi ottici passivi

- *depth from focus/defocus* (capitolo 5)
- *shape from texture* (capitolo 5)
- *shape from shading* (capitolo 5)
- Stereo fotometrico (capitolo 5)
- Stereopsi (capitolo 6)

- *shape from silhouette* (capitolo 8)
- *shape from photo-consistency* (capitolo 8)
- *structure from motion* (capitolo 9, 10)

Metodi ottici attivi

- *active defocus*
- stereo attivo (capitolo 7)
- triangolazione attiva (capitolo 7)
- interferometria
- tempo di volo

Tutti i metodi attivi tranne l'ultimo impiegano una o due fotocamere ed una sorgente di luce speciale, e rientrano nella classe più ampia dei metodi ad **illuminazione strutturata**.

Le tecniche prive di riferimento al capitolo pertinente verranno trattate sommariamente nel § 2.5.

2.3 Immagini 2.5D

Molti dei metodi ottici per l'acquisizione della forma restituiscono una immagine di distanza (o *range*), ovvero una immagine nella quale ciascun pixel contiene la distanza dal sensore ad un punto visibile della scena, invece della sua luminosità (figura 2.3).



Fig. 2.3. Immagine di colore ed immagine *range* dello stesso soggetto. (da <http://www.siggraph.org/publications/newsletter/v33n4/contributions/mcmillan.html>)

Una immagine di distanza è costituita da misure (discrete) di una superficie 3D rispetto ad un piano 2D (il piano immagine del sensore, di solito) e pertanto prende anche il nome di *immagine 2.5D*. La superficie si potrà sempre esprimere nella forma $Z = Z(X, Y)$, se il piano di riferimento è identificato con il piano XY .

Un sensore *range* è un dispositivo che produce una immagine *range*. Nel seguito parleremo di sensori *range* ottici in senso generalizzato, per intendere qualunque sistema ottico di acquisizione della forma, attivo o passivo, composto da apparecchiature e programmi, che restituisce una immagine di distanza. **La bontà di un sensore *range* si valuta prendendo in considerazione (ma non solo):**

risoluzione: il più piccolo cambiamento di profondità che il sensore può rilevare;

accuratezza: differenza tra valore misurato (media di misure ripetute) e valore vero (misura l'errore sistematico);

precisione: variazione statistica (deviazione standard) di misure ripetute di una stessa quantità (misura la dispersione delle misure attorno alla media);

velocità: misure al secondo.

2.4 Dalla misura al modello 3D

Il recupero della informazione 3D non esaurisce però il processo di acquisizione della forma, per quanto ne costituisca il passo fondamentale.

Per ottenere un modello completo di un oggetto, o di una scena, servono molte immagini 2.5D, che devono essere allineate e fuse tra di loro, ottenendo una superficie 3D (in forma di maglia poligonale). La ricostruzione del modello dell'oggetto a partire da immagini 2.5D prevede dunque tre fasi:

- (i) **registrazione:** (o allineamento) per trasformare le misure fornite dalle varie immagini 2.5D in unico sistema di riferimento comune;
- (ii) **fusione geometrica:** per ottenere una singola superficie 3D (tipicamente una maglia poligonale) dalle varie superfici 2.5D;
- (iii) **semplificazione della maglia:** tipicamente i punti restituiti da un sensore *range* sono troppi per avere un modello maneggevole. La maglia finale risultante deve essere semplificata in maniera opportuna.

Nel seguito descriveremo soprattutto la prima fase, tratteremo sommariamente la seconda e tralasceremo la terza. Questo perché questi argomenti sfumano gradualmente nella Grafica Computazionale, ed esulano dunque dagli scopi di questo corso.

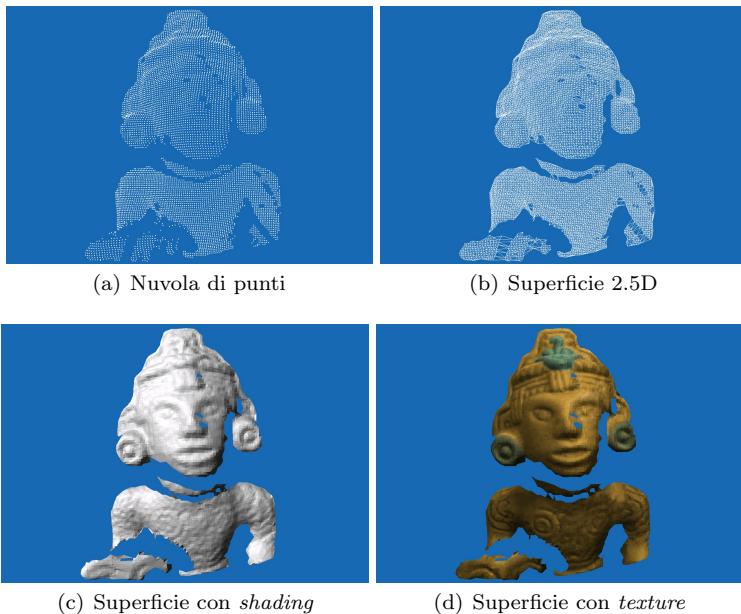


Fig. 2.4. Superficie 2.5D

Superfici 2.5D. Una immagine 2.5D definisce un insieme di punti 3D del tipo $(X, Y, Z(X, Y))$, come in figura 2.4a. Per ottenere una superficie nello spazio 3D (superficie 2.5D) è sufficiente connettere tra di loro i punti più vicini con facce triangolari (figura 2.4b). In pratica, si effettua una triangolazione[†] del piano XY , con l'algoritmo di Delaunay.

In molti casi esistono discontinuità di profondità (buchi, bordi occludenti) che non si vuole vengano coperte dai triangoli che potrebbero crearsi. Per evitare di fare assunzioni non giustificate sulla forma della superficie, dobbiamo evitare di collegare punti che sono troppo distanti (nello spazio). Per questo è opportuno eliminare triangoli con lati eccessivamente lunghi e quelli con angoli eccessivamente acuti.

2.4.1 Registrazione

I sensori *range* tipicamente non catturano la forma di un oggetto con una sola immagine, ne servono molte, ciascuna delle quali cattura una parte della superficie dell'oggetto. Le porzioni della superficie dell'og-

[†] Una triangolazione di un insieme \mathcal{V} di punti nel piano è una suddivisione del piano in triangoli i cui vertici sono i punti di \mathcal{V} .

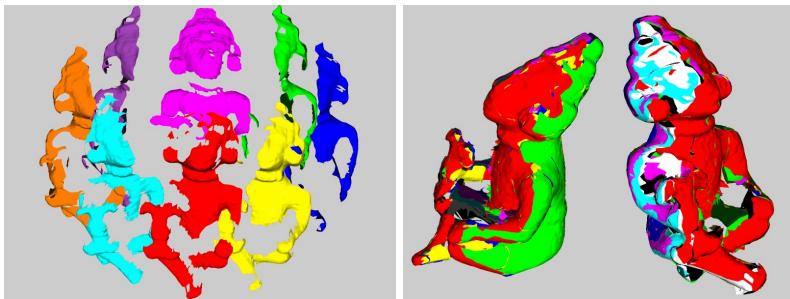


Fig. 2.5. Registrazione. A destra otto immagini *range* di un oggetto ognuna nel proprio sistema di riferimento. A sinistra, tutte le immagini sovrapposte dopo la registrazione.

getto che si ottengono da diverse immagini 2.5D sono espresse ciascuna nel proprio sistema di riferimento (legato alla posizione del sensore). Lo scopo della registrazione o allineamento è di portarle nello stesso sistema di riferimento tramite una opportuna trasformazione rigida (rotazione e traslazione nel 3D).

Se la posizione ed orientazione del sensore sono note (perché vengono rilevate oppure perché esso è controllato meccanicamente), il problema si risolve banalmente. Tuttavia in molti casi, le trasformazioni vanno calcolate usando i soli dati. Studieremo nel capitolo 11 un algoritmo, chiamato ICP (*Iterated Closest Point*) che risolve il problema.

La registrazione di immagini 2.5D presenta alcune analogie con la mosaicatura di immagini, che tratteremo nel capitolo 14 (infatti il processo di registrazione e fusione prende anche il nome di *mosaicatura 3D*). Le proiettività del piano sono sostituite dalle trasformazioni rigide 3D. La fase di miscelazione corrisponde alla fusione geometrica.

2.4.2 Fusione geometrica

Una volta che tutti i dati 2.5D sono stati precisamente registrati in un sistema di coordinate comune, dobbiamo fondere i dati in una singola forma, rappresentata, ad esempio, da una maglia triangolare. Questo è il problema della ricostruzione della superficie, che si può formulare come la stima di una varietà bidimensionale che approssimi la superficie dell'oggetto incognito da un insieme di punti 3D campionati.

Nel problema generale non si ha alcuna informazione di connettività. La ricostruzione di superficie da immagini 2.5D rappresenta un caso più semplice, dato che tale informazione è disponibile passando alle superfici

2.5D. I metodi per la ricostruzione possono sfruttare questa caratteristica o ignorarla.

Possiamo suddividere i metodi di fusione geometrica in due categorie:

- Integrazione di maglie: vengono unite le maglie triangolari corrispondenti alle singole superfici 2.5D.
- Fusione volumetrica: i dati vengono fusi in una rappresentazione volumetrica, dalla quale si estraе poi una maglia triangolare.

2.4.2.1 Integrazione di maglie

Le tecniche di integrazione di maglie mirano ad unire più maglie triangolari 3D sovrapposte in un'unica maglia triangolare (sfruttando la rappresentazione in termini di superficie 2.5D).

Il metodo di [Turk e Levoy, 1994] unisce maglie triangolari sovrapposte impiegando una tecnica detta di *zippering*. Le maglie sovrapposte vengono erose fino ad eliminare la sovrapposizione e poi si impiega una triangolazione nel 2D per ricucire i bordi. Questo richiede che i punti delle due superfici 3D prossimi ai bordi da ricucire vengano proiettati su di un piano 2D. Per una corretta ricostruzione della superficie è necessario che la proiezione da 3D a 2D sia iniettiva. Questa imposizione porta ad una limitazione sulla massima curvatura per avere una ricostruzione corretta.

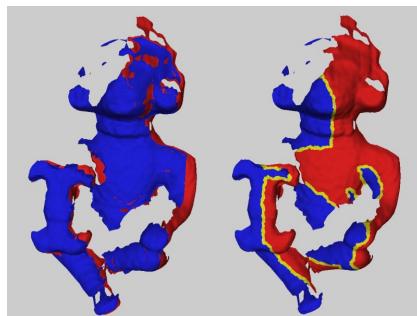


Fig. 2.6. Due superfici registrate (a sinistra) e risultato dello zippering (a destra).

Le tecniche di integrazione di maglie permettono la fusione di più immagini 2.5D senza perdere nell'accuratezza, poiché i vertici della maglia finale coincidono con i punti dei dati misurati. Per lo stesso motivo sono sensibili a misurazioni sbagliate, che possono provocare problemi nella ricostruzione della superficie.

2.4.2.2 Fusione Volumetrica

La fusione volumetrica di misurazioni di superficie costruisce una *superficie implicita* intermedia che unisce le misurazioni sovrapposte in un'unica rappresentazione. La rappresentazione della superficie implicita è una iso-superficie di un campo scalare $f(x, y, z)$. Per esempio, se la funzione di campo è definita come la distanza del punto più vicino sulla superficie dell'oggetto, allora la superficie implicita è rappresentata da $f(x, y, z) = 0$. Questa rappresentazione permette la modellazione di oggetti dalla forma sconosciuta con topologia e geometria arbitrari.

Per passare dalla rappresentazione implicita della superficie ad una maglia triangolare, si usa l'algoritmo Marching Cubes [Lorensen e Cline, 1987] per la triangolazione di iso-superfici a partire dalla rappresentazione discreta di un campo scalare (come le immagini 3D nel campo medico). Lo stesso algoritmo torna utile per ottenere una superficie triangolata a partire da ricostruzioni volumetriche della scena (*shape from silhouette* e *photo consistency*)

Il metodo di [Hoppe e al., 1992] trascura la struttura dei dati (cioè la superficie 2.5D) e calcola una superficie a partire dalla “nuvola” non strutturata dei punti.

Curless e Levoy [1996] invece sfruttano l'informazione contenuta nelle immagini 2.5D per assegnare a “vuoto” i voxel che giacciono lungo la linea di vista che da un punto della superficie 2.5D arriva fino al sensore.

Una limitazione evidente di tutti gli algoritmi di fusione geometrica basati su una struttura intermedia di dati volumetrici discreti è una riduzione nell'accuratezza, che provoca la perdita di dettagli della superficie.
Inoltre lo spazio necessario per la rappresentazione volumetrica cresce rapidamente al crescere della risoluzione.

2.5 Altri metodi

Trattiamo ora sommariamente dei metodi ottici attivi che non saranno studiati in maggior dettaglio. Per quanto riguarda *active defocus* basti dire che funziona come *shape from defocus* ma viene impiegata una illuminazione strutturata per creare tessitura ove non fosse presente. Per i metodi basati sulla interferometria vediamo quelli che sfruttano le frange di interferenza di Moiré poi illustriamo il principio dei sensori *range finders* a tempo di volo.

2.5.1 Frange di interferenza di Moiré

L'idea di questo metodo è quella di proiettare una griglia su un oggetto e prenderne un'immagine attraverso una seconda griglia di riferimento. Questa immagine interferisce con la griglia di riferimento (fenomeno di battimento) e crea le cosiddette frange di interferenza di Moiré, che appaiono come bande di luce ed ombra, come viene mostrato in figura 2.7. L'analisi di questi pattern fornisce informazioni sulla variazione della profondità. In particolare, ogni banda rappresenta un contorno della superficie dell'oggetto di iso-profondità. Un problema è che non è possibile determinare se i contorni adiacenti hanno una profondità più alta o più bassa. La soluzione consiste nel muovere una delle griglie e raccogliere più immagini.

La griglia di riferimento si può anche simulare via software. Questi metodi sono in grado di fornire dati molto accurati sulla profondità (risoluzione fino a circa $10 \mu m$), ma presentano alcuni inconvenienti. Sono tecniche piuttosto costose dal punto di vista computazionale. Le superfici con un'ampia angolazione non sono sempre possibili da misurare, poiché la densità delle bande diventa troppo fitta.

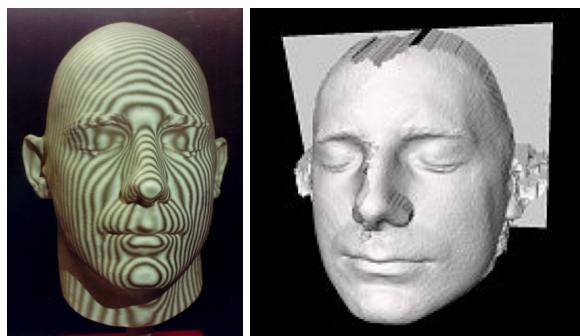


Fig. 2.7. Frange di Moiré. Il modello a destra è stato catturato impiegando l'*OrthoForm clinical prototype facial imaging system*. (da <http://www.faraday.gla.ac.uk/moire.htm>)

2.5.2 Tempo di volo

Un *Range finder* a tempo di volo (TOF) sfrutta lo stesso principio del RADAR, ovvero il fatto che la superficie dell'oggetto riflette la luce laser indietro verso un ricevitore che misura quindi il tempo trascorso tra la trasmissione e la ricezione, in modo da calcolare la distanza. Uno

scanning laser range finder – o *scanner* – usa un fascio di luce laser per scandagliare la scena, in modo da ricavare la distanza di tutti i punti visibili.

La distanza (o *range*) r si ottiene dal tempo di propagazione t_r di un'onda luminosa riflessa dal bersaglio (tempo di volo). Se c indica la velocità della luce, allora t_r è dato da: $t_r = \frac{2r}{c}$.

Questo tempo di volo si può misurare con differenti metodi: direttamente (onda impulsiva) o con la conversione in un ritardo di fase mediante modulazione di ampiezza (AM).

Principio della misurazione del ritardo di fase. La luce emessa dal diodo laser è modulata da un'onda sinusoidale alla frequenza di $F_{AM} = \frac{c}{\lambda_{AM}}$. Il tempo di volo viene convertito in un ritardo di fase:

$$\Delta\phi = 2\pi F_{AM} \frac{2r}{c} \quad (2.1)$$

dato che

$$2r = n\lambda_{AM} + \frac{\Delta\phi}{2\pi}\lambda_{AM} \quad (2.2)$$

A causa del fatto che la fase è misurata modulo 2π , anche le distanze risultano misurate modulo $\Delta r = \frac{\lambda_{AM}}{2}$. Questa ambiguità può essere eliminata attraverso spazzolate con lunghezze d'onda decrescenti.

La scelta della frequenza di modulazione dipende dal tipo di applicazione. Una scelta di $F_{AM}=10\text{MHz}$, che corrisponde ad un intervallo di non ambiguità di 15m, è tipicamente adatta per la visione robotica in ambienti interni. Numerosi *range finder* laser lavorano su distanze più lunghe (più grandi di 15m). Di conseguenza la loro risoluzione in profondità non è adatta per la rilevazione di dettagli (1cm al massimo).

Bibliografia

- Bernardini F.; Rushmeier H. E. (2002). The 3d model acquisition pipeline. *Comput. Graph. Forum*, **21**(2), 149–172.
- Curless B. (2000). Overview of active vision techniques. SIGGRAPH 2000 Course on 3D Photography.
- Curless B.; Levoy M. (1996). A volumetric method for building complex models from range images. In *SIGGRAPH: International Conference on Computer Graphics and Interactive Techniques*, pp. 303–312, New Orleans, Louisiana.

- Hoppe H.; DeRose T.; Duchamp T.; McDonald J.; Stuetzle W. (1992). Surface reconstruction from unorganized points. *Computer Graphics*, **26**(2), 71–78.
- Lorensen W.; Cline H. (1987). Marching cubes: a high resolution 3-D surface construction algorithm In *SIGGRAPH: International Conference on Computer Graphics and Interactive Techniques*. A cura di Stone M., pp. 163–170, Anaheim, CA.
- Scateni R.; Cignoni P.; Montani C.; Scopigno R. (2005). *Fondamenti di Grafica Tridimensionale Interattiva*. Mc Graw Hill, Milano, first edizione. ISBN 88 386 6215-0.
- Turk G.; Levoy M. (1994). Zippered polygon meshes from range images In *SIGGRAPH: International Conference on Computer Graphics and Interactive Techniques*. A cura di Glassner A., pp. 311–318.

3

Formazione dell'immagine

Un apparato di acquisizione di immagini (*imaging*) funziona raccogliendo la luce riflessa dagli oggetti della scena e creando una immagine bidimensionale. Se vogliamo usare l'immagine per ottenere informazione sulla scena, dobbiamo studiare bene la natura di questo processo (che vorremmo poter invertire).

3.1 Geometria della formazione dell'immagine

Il modello geometrico più semplice della formazione dell'immagine è la *fotocamera stenopeica* o *stenoscopio* (o *pinhole camera*), rappresentata in figura 3.1. Si tratta dello stesso principio della *camera oscura* rinascimentale.

Sia M un punto della scena, di coordinate (X, Y, Z) e sia M' la sua

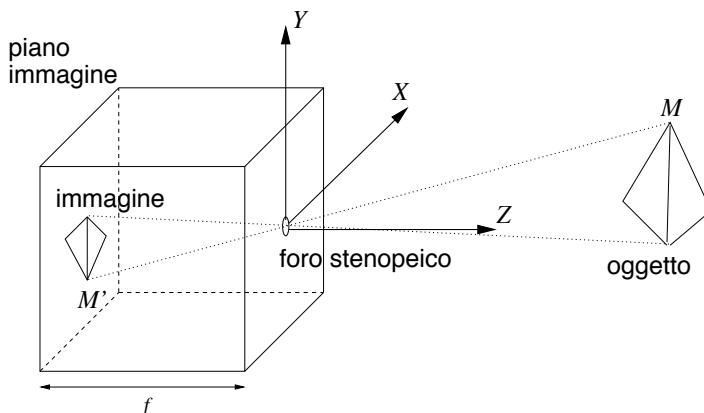


Fig. 3.1. Formazione dell'immagine nello stenoskopio.

proiezione sul piano immagine (o retina), di coordinate (X', Y', Z') . Se f è la distanza del foro (o centro di proiezione) dal piano immagine (distanza focale), allora dalla similitudine dei triangoli si ottiene:

$$\frac{-X'}{f} = \frac{X}{Z} \quad \text{e} \quad \frac{-Y'}{f} = \frac{Y}{Z} \quad (3.1)$$

e quindi

$$X' = \frac{-fX}{Z} \quad Y' = \frac{-fY}{Z} \quad Z' = -f \quad (3.2)$$

Si noti che l'immagine è invertita rispetto alla scena, sia destra-sinistra che sopra-sotto, come indicato dal segno meno. Queste equazioni definiscono il processo di formazione dell'immagine che prende il nome di **proiezione prospettica**. Equivalentemente possiamo modellare la proiezione prospettica ponendo il piano immagine *davanti* al centro di proiezione, eliminando così il segno negativo. Questo accorgimento fu per la prima volta raccomandato nel rinascimento da Leon Battista Alberti.[†]

La divisione per Z è responsabile per l'effetto di *scorcio*[‡] (tecnica risalente anch'essa al rinascimento) per cui la dimensione dell'immagine di un oggetto varia in ragione della sua distanza dall'osservatore.



Fig. 3.2. L'immagine a sinistra è decisamente prospettica – si notino le linee convergenti – mentre l'immagine aerea di destra è decisamente ortografica – la distanza dall'oggetto è sicuramente molto grande rispetto alla sua profondità.

Se l'oggetto inquadrato è relativamente sottile, confrontato con la sua

[†] Alberti è accreditato come scopritore della prospettiva, con il suo trattato “De Pictura” del 1435. Ma già qualche decennio prima Biagio Pelacani aveva pubblicato il libro “Questioni di Prospettiva” nel quale studiava le ombre portate come immagini prospettiche.

[‡] Scorcio: rappresentazione di un oggetto che giace su un piano obliquo rispetto all'osservatore in modo da apparire, secondo le norme di una visione prospettica, accorciato. Dal vocabolario Zingarelli della lingua italiana.

distanza media dalla fotocamera, si può approssimare la proiezione prospettica con la **proiezione ortografica** (scalata) o *weak perspective*. L'idea è la seguente: se la profondità Z dei punti dell'oggetto varia in un intervallo $Z_0 \pm \Delta Z$, con $\frac{\Delta Z}{Z_0} \ll 1$, allora il fattore di scala prospettico f/Z può essere approssimato da una costante f/Z_0 . Anche questo risultato era noto nel rinascimento: Leonardo da Vinci, infatti raccomandava di usare questa approssimazione quando $\frac{\Delta Z}{Z_0} < \frac{1}{10}$. Le equazioni di proiezione diventano allora:

$$X' = \frac{-f}{Z_0} X \quad Y' = \frac{-f}{Z_0} Y \quad (3.3)$$

Si tratta di una proiezione ortografica, composta con una scalatura di un fattore f/Z_0 .

3.2 Lenti sottili

Gli occhi dei vertebrati, le fotocamere e le telecamere usano lenti. Una lente, essendo più grande di un foro di spillo, riesce a raccogliere più luce. Il rovescio della medaglia è che non tutta la scena può essere a fuoco allo stesso tempo. L'approssimazione che facciamo per l'ottica del sistema di acquisizione – che in generale è molto complessa, essendo costituita da più lenti – è quella della **lente sottile**. Le lenti sottili godono delle seguenti proprietà:

- (i) i raggi paralleli all'asse ottico incidenti sulla lente vengono rifratti

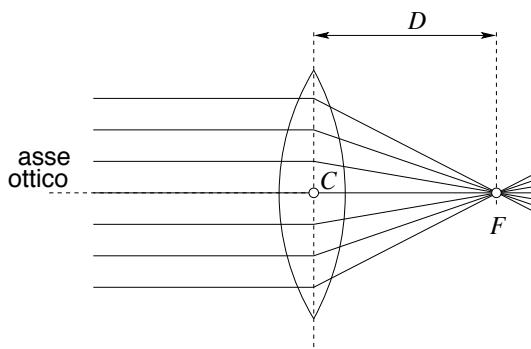


Fig. 3.3. Lente sottile.

in modo da passare per un punto dell'asse ottico chiamato *fuoco* F .

(ii) i raggi che passano per il *centro C* della lente sono inalterati.

La distanza del fuoco F dal centro della lente C prende il nome di *distanza focale D* (figura 3.3). Essa dipende dai raggi di curvatura delle due superfici della lente e dall'indice di rifrazione del materiale costituente.

Dato un punto della scena M è possibile costruirne graficamente l'immagine M' (o *punto coniugato*) servendoci di due raggi particolari che partono da M : il raggio parallelo all'asse ottico, che dopo la rifrazione passa per F ed il raggio che passa inalterato per C (figura 3.4).

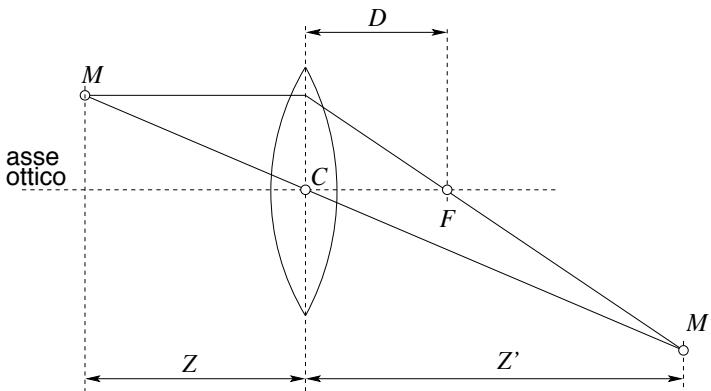


Fig. 3.4. Costruzione dell'immagine di un punto.

Grazie a questa costruzione ed alla similitudine dei triangoli, si ottiene la **formula dei punti coniugati** (o equazione della lente sottile):

$$\frac{1}{Z} + \frac{1}{Z'} = \frac{1}{D} \quad (3.4)$$

L'immagine di un punto della scena distante Z dalla lente, viene prodotta (a fuoco) ad una distanza dalla lente Z' , che dipende dalla profondità Z del punto e dalla distanza focale D della lente. Per mettere a fuoco oggetti a distanze diverse, le lenti dell'occhio cambiano focale deformandosi, mentre le lenti delle fotocamere traslano lungo Z .

In sostanza, l'immagine del punto M , quando è a fuoco, si forma sul piano immagine nello stesso punto previsto dal modello stenopeico con il foro coincidente con il centro della lente C , infatti il raggio passante per C è comune alle due costruzioni geometriche. Gli altri raggi luminosi

che lasciano M e vengono raccolti dalla lente servono ad aumentare la luce che raggiunge M' .

Se la (3.4) non è verificata si ottiene una immagine sfocata del punto, ovvero un cerchio che prende il nome di *cerchio di confusione*. Il piano immagine è coperto da elementi fotosensibili i quali hanno una dimensione piccola ma finita. Finché il cerchio di confusione non supera le dimensioni dell'elemento fotosensibile l'immagine risulta a fuoco. Dunque, esiste un intervallo di profondità per le quali i punti sono a fuoco. Questo intervallo prende il nome di **profondità di campo**. La profondità di campo è inversamente proporzionale al diametro della lente. Infatti la fotocamera stenopeica ha una profondità di campo infinita. La luce che viene raccolta, invece, è direttamente proporzionale al diametro della lente.

Attenzione: la lunghezza focale della lente è una cosa diversa dalla lunghezza focale della fotocamera stenopeica. Abbiamo usato a proposito due notazioni diverse, anche se, purtroppo, le due quantità hanno lo stesso nome.

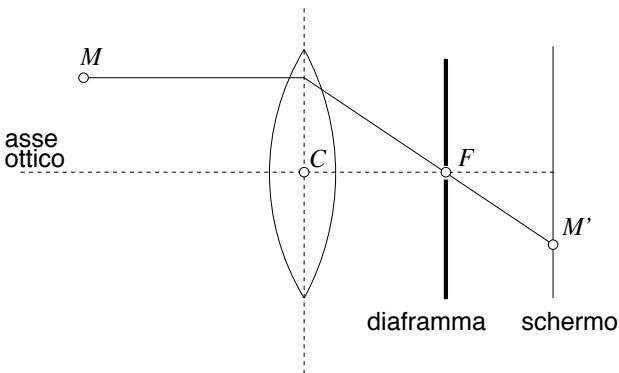


Fig. 3.5. Principio della fotocamera telecentrica.

Ottica telecentrica È interessante notare che posizionando un diaframma con un foro in corrispondenza del fuoco F , come in figura 3.5, vengono selezionati i raggi che lasciano M in direzione parallela all'asse ottico, mentre il raggio passante per C viene bloccato. Si ottiene così un dispositivo che produce una immagine secondo un modello evidentemente diverso dal modello stenopeico/prospettico. L'immagine del punto M , infatti, non dipende dalla sua profondità: abbiamo realizzato una *fotocamera telecentrica*, che realizza il modello di proiezione ortografica.

3.3 Radiometria della formazione dell'immagine

La luminosità (*brightness*) $I(p)$ di un pixel p nell'immagine è proporzionale alla quantità di luce che la superficie S_p – centrata in un punto x – riflette verso la fotocamera (S_p è la superficie che si proietta nel pixel p). Questa a sua volta dipende dal modo in cui la superficie riflette la luce e dalla distribuzione spaziale delle sorgenti luminose†.

La “quantità di luce” emessa o assorbita da un punto si misura formalmente mediante la *radianza*: si definisce radianza $L(x, \omega)$ la potenza della radiazione luminosa per unità di area† per unità di angolo solido emessa (o ricevuta) dal punto x lungo la direzione ω .

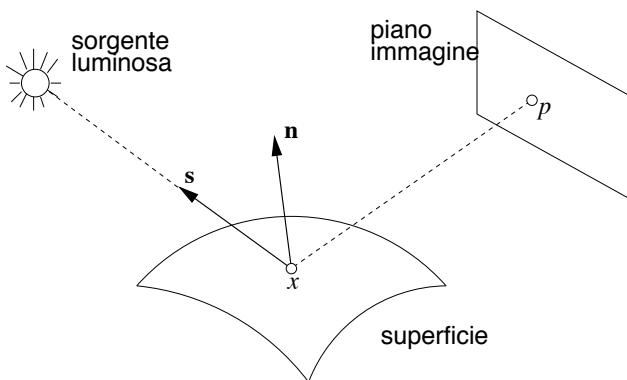


Fig. 3.6. Superficie lambertiana.

Trascurando l'attenuazione dovuta all'atmosfera, la radianza che lascia un punto x della superficie nella direzione del pixel p coincide con la radianza che raggiunge il pixel p dalla medesima direzione. In un modello semplificato, approssimeremo la luminosità del pixel (*brightness*) $I(p)$ con la radianza:

$$I(p) = L(x). \quad (3.5)$$

La riflessione della luce è un fenomeno complesso che dipende dal materiale di cui è composta la superficie. La *riflettanza* è la proprietà di una superficie che descrive il modo in cui essa riflette la luce incidente. I due casi estremi sono la diffusione e la riflessione speculare.

† In prima approssimazione contano solo le sorgenti primarie, ovvero dirette, ma in un modello più accurato anche le sorgenti secondarie, ovvero tutte le altre superfici della scena devono essere considerate.

† Si considera l'area proiettata lungo la direzione di propagazione

Diffusione Nella diffusione la luce incidente viene assorbita e ri-emessa. La superficie appare ugualmente luminosa da ogni direzione. La riflettanza di una superficie lambertiana si può agevolmente caratterizzare, infatti la radianza $L(x)$ emessa da x segue la cosiddetta **legge di Lambert**:

$$L(x) = \rho(x)E(x) \quad (3.6)$$

dove $E(x)$ è la *irradianza* in x , ovvero la potenza della radiazione luminosa per unità di area incidente nel punto x (da tutte le direzioni); $\rho(x)$ è l'albedo di x , che varia da 0 (nero) a 1 (bianco).

Considerando una sorgente luminosa puntiforme, si verifica che $E(x) = L(x, \mathbf{s})(\mathbf{s}^\top \mathbf{n})$, dove \mathbf{s} è la direzione sotto cui x vede la sorgente luminosa e \mathbf{n} è la direzione (il versore) della normale in x (si faccia riferimento alla figura 3.6). In questo caso, la legge di Lambert si scrive:

$$L(x) = \rho(x)L(x, \mathbf{s})(\mathbf{s}^\top \mathbf{n}). \quad (3.7)$$

Riflessione speculare Nella riflessione speculare la radianza riflessa è concentrata lungo una particolare direzione, quella per cui il raggio riflesso e quello incidente giacciono sullo stesso piano e l'angolo di riflessione è uguale all'angolo di incidenza: è il comportamento di uno specchio perfetto. La riflettanza di una superficie speculare, diversamente da quella della superficie lambertiana, tiene conto della direzione di incidenza della luce.

Le superfici lucide presentano un comportamento che (in prima approssimazione) è una combinazione di diffusione e riflessione speculare.

3.4 Immagini digitali

Una fotocamera digitale è composta dall'ottica – che approssimiamo con una lente sottile – e da una matrice di CCD (*Charge-Coupled Device*) o CMOS che costituisce il piano immagine (un CCD misura circa 50 mm² e contiene ordine di $\times 10^6$ elementi). Possiamo vedere quest'ultima come una matrice $n \times m$ di celle rettangolari fotosensibili, ciascuna dei quali converte l'intensità della radiazione luminosa incidente in un potenziale elettrico.

La matrice del CCD (o CMOS) viene convertita in una immagine digitale, ovvero in una matrice $N \times M$ (per esempio 1024×768) di valori interi (per esempio 0...255). Gli elementi della matrice prendono il nome di **pixel** (*picture element*). Indicheremo con $I(u, v)$ il valore dell'immagine (luminosità) nel pixel individuato dalla riga v e colonna u (sistema di coordinate u, v con l'origine nell'angolo in alto a sinistra).

La dimensione $n \times m$ della matrice CCD non è necessariamente la stessa della immagine $N \times M$ (matrice dei pixel); per questo motivo la posizione di un punto del piano immagine è diversa se misurata in elementi CCD piuttosto che in pixel.

$$u_{\text{pix}} = \frac{N}{n} u_{\text{CCD}} \quad (3.8)$$

$$v_{\text{pix}} = \frac{M}{m} v_{\text{CCD}} \quad (3.9)$$

Ad un pixel dunque corrisponde un'area rettangolare sul CCD array (si chiama anche *impronta* del pixel), non necessariamente uguale ad una cella del CCD, le cui dimensioni sono le **dimensioni efficaci del pixel**.

I temi trattati in questo capitolo si trovano approfonditi in [Horn, 1986]. Altre fonti sono [Russel e Norvig, 1995, Cap. 24] [Trucco e Verri, 1998].

Bibliografia

- Horn B. (1986). *Robot Vision*. The MIT Press.
 Russel S.; Norvig P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall.
 Trucco E.; Verri A. (1998). *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall.

4

Calibrazione della fotocamera

Introdurremo ora un modello geometrico per la fotocamera e affronteremo il problema della determinazione dei parametri del modello (calibrazione).

4.1 Modello della fotocamera

In questo paragrafo richiameremo alcuni concetti legati al modello geometrico della formazione dell'immagine, ovvero studieremo come sono collegate la posizione di un punto nella scena e la posizione del punto corrispondente nell'immagine, mediante un modello geometrico del sensore. Il più comune modello geometrico della fotocamera è il cosiddetto modello *stenopecico* o prospettico, il quale consiste (si faccia riferimento alla figura 4.1) di un piano retina (o immagine) \mathcal{R} e di un punto C , *centro ottico* (o centro di proiezione) distante f (*lunghezza focale*) dal piano. La retta passante per C ortogonale a \mathcal{R} è l'*asse ottico* (asse z nella figura 4.1) e la sua intersezione con \mathcal{R} prende il nome di *punto principale*. Il piano \mathcal{F} parallelo ad \mathcal{R} e contenente il centro ottico prende il nome di *piano focale*. I punti di tale piano si proiettano all'infinito sul piano immagine.

Per descrivere analiticamente la proiezione prospettica operata dalla fotocamera, dobbiamo introdurre opportuni sistemi di riferimento cartesiani in cui esprimere le coordinate del punto dello spazio 3D e le coordinate del punto proiettato sul piano immagine.

Consideriamo, inizialmente, un **caso molto speciale**, in cui i sistemi di riferimento sono scelti in modo da ottenere equazioni particolarmente semplici. Inoltre trascuriamo la *pixelizzazione*. Nella §4.1.2 complicheremo il modello studiando il caso più generale.

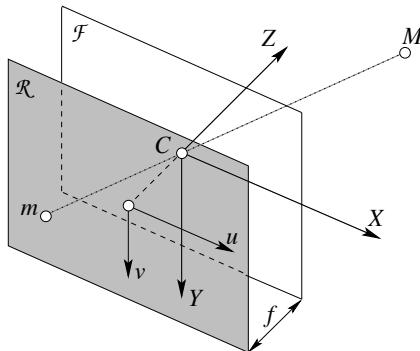


Fig. 4.1. Modello geometrico della fotocamera

4.1.1 Modello semplificato

Introduciamo un sistema di riferimento destrorso (X, Y, Z) per lo spazio tridimensionale (sistema *mondo*) centrato in C e con l'asse Z coincidente con l'asse ottico. Quest'ultimo prende il nome di sistema di riferimento *standard* della fotocamera (in altri termini, stiamo fissando il riferimento mondo coincidente con il riferimento standard della fotocamera).

Introduciamo anche un sistema di riferimento (u, v) per il piano \mathcal{R} centrato nel punto principale e con gli assi u e v orientati come X ed Y rispettivamente, come mostrato in figura 4.1.

Consideriamo ora un punto M di coordinate $\tilde{\mathbf{M}} = [x, y, z]^\top$ nello spazio 3D e sia m di coordinate $\tilde{\mathbf{m}} = [u, v]^\top$ la sua proiezione su \mathcal{R} attraverso C . Nel seguito, salvo diversamente indicato, useremo sempre la \sim per denotare le coordinate cartesiane e distinguere da quelle omogenee[†].

Mediane semplici considerazioni sulla similitudine dei triangoli (figura 4.2), possiamo scrivere la seguente relazione:

$$\frac{f}{z} = \frac{-u}{x} = \frac{-v}{y} \quad (4.1)$$

(il segno $-$ serve a tenere conto della inversione di segno delle coordinate)

[†] In coordinate omogenee un punto 2-D del piano immagine viene denotato con una terna (x_1, x_2, x_3) , dove $(x_1/x_3, x_2/x_3)$ sono le corrispondenti coordinate cartesiane. Sussiste quindi una corrispondenza uno a molti tra coordinate cartesiane ed omogenee. Queste ultime possono rappresentare qualunque punto del piano euclideo ed anche i punti all'infinito, che hanno la terza componente uguale a zero e dunque non possiedono una corrispondente rappresentazione cartesiana. Si veda l'appendice 2.

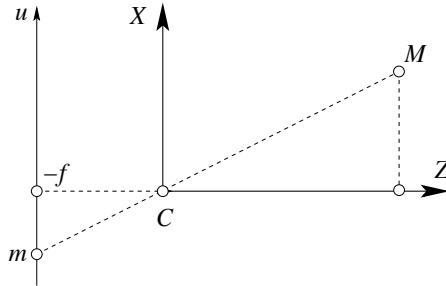


Fig. 4.2. Vista semplificata del modello della fotocamera. I due triangoli che hanno per ipotenusa \overline{CM} e \overline{Cm} sono simili tra loro.

te), ovvero

$$\begin{cases} u = \frac{-f}{z} x \\ v = \frac{-f}{z} y \end{cases}. \quad (4.2)$$

Questa è la *proiezione prospettica*. La trasformazione dalla coordinate 3D a quelle 2-D è chiaramente non lineare (a causa della divisione per Z). Usando invece le coordinate omogenee (e quindi intendendo la trasformazione come tra spazi proiettivi), essa diviene *lineare*.

Siano dunque

$$\mathbf{m} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{M} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (4.3)$$

le coordinate omogenee di m ed M rispettivamente. Si noti che ponendo la terza coordinata ad 1, abbiamo escluso i punti all'infinito (per includerli avremmo dovuto usare una terza componente generica). Dunque l'equazione di proiezione prospettica, in questo caso semplificato, si riscrive:

$$z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -fx \\ -fy \\ z \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (4.4)$$

Dunque, passando alla notazione matriciale:

$$z\mathbf{m} = P\mathbf{M}. \quad (4.5)$$

oppure, anche:

$$\mathbf{m} \simeq P\mathbf{M}. \quad (4.6)$$

dove \simeq significa “uguale a meno di un fattore di scala”.

La matrice P rappresenta il modello geometrico della fotocamera viene chiamata *matrice della fotocamera* o *matrice di proiezione prospettica* (MPP) .

Nel caso molto speciale (ed ideale) in cui il piano immagine è davanti al centro di proiezione, come in figura 4.3, e la focale è unitaria ($f = -1$), si ha

$$P \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = [I|\mathbf{0}]. \quad (4.7)$$

Questa MPP codifica la trasformazione prospettica essenziale, senza alcun parametro.

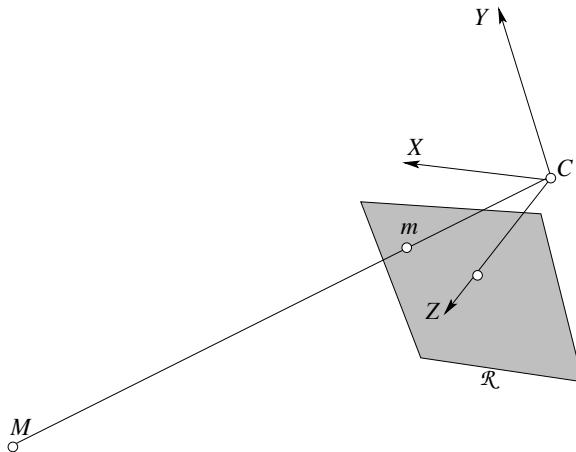


Fig. 4.3. Proiezione prospettica con piano immagine davanti al centro di proiezione.

4.1.2 Modello generale

Un modello realistico di una fotocamera, che descriva la trasformazione da coordinate 3D a coordinate pixel, oltre che della trasformazione prospettica, deve tenere conto di

- trasformazione rigida tra la fotocamera e la scena;

- la pixelizzazione, (forma e dimensione della matrice CCD e sua posizione rispetto al centro ottico).

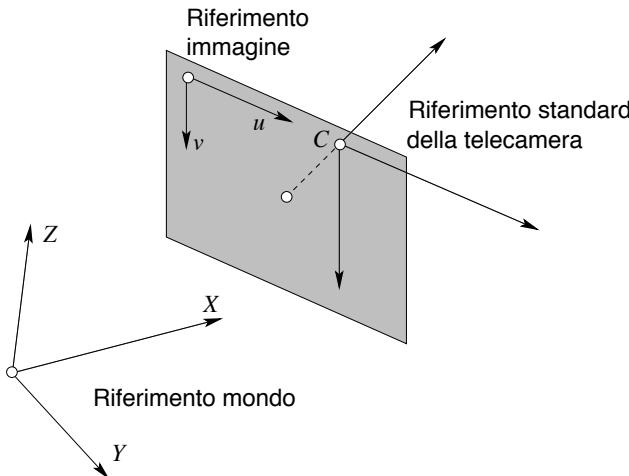


Fig. 4.4. Sistemi di riferimento.

4.1.2.1 Parametri intrinseci

La pixelizzazione viene presa in considerazione mediante una trasformazione affine che tiene conto della traslazione del centro ottico e la riscalatura indipendente degli assi u e v :

$$\begin{cases} u = k_u \frac{-f}{z} x + u_0 \\ v = k_v \frac{-f}{z} y + v_0 \end{cases}, \quad (4.8)$$

dove (u_0, v_0) sono le coordinate del punto principale, k_u (k_v) è l'inverso della dimensione efficace del pixel lungo la direzione u (v); le sue dimensioni fisiche sono $\text{pixel} \cdot \text{m}^{-1}$.

Dopo questo aggiornamento delle equazioni, la MPP diventa:

$$P = \begin{bmatrix} -fk_u & 0 & u_0 & 0 \\ 0 & -fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = K[I|\mathbf{0}] \quad (4.9)$$

dove

$$K = \begin{bmatrix} -fk_u & 0 & u_0 \\ 0 & -fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.10)$$

Poniamo $\alpha_u = -fk_u$ e $\alpha_v = -fk_v$: si tratta della lunghezza focale espressa in pixel orizzontali e verticali rispettivamente. I *parametri intrinseci* (o interni), codificati nella matrice K sono dunque i seguenti quattro: $\alpha_u, \alpha_v, u_0, v_0$.

Il modello più generale prevede anche un ulteriore parametro θ , l'angolo tra gli assi u e v (normalmente però $\theta = \pi/2$):

$$K = \begin{bmatrix} -fk_u & fk_u \cot \theta & u_0 \\ 0 & -fk_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.11)$$

Spesso si trova indicato il parametro di *skew gamma* $= fk_u \cot \theta$.

Coordinate normalizzate. Se si introduce il cambio di coordinate

$$\mathbf{p} = K^{-1}\mathbf{m}. \quad (4.12)$$

la MPP si riduce a $[I|\mathbf{0}]$ (semplice verifica). Queste speciali coordinate prendono il nome di *coordinate normalizzate* o coordinate immagine. Per passare in coordinate normalizzate bisogna conoscere i parametri intrinseci.

4.1.2.2 Parametri estrinseci

Per tenere conto del fatto che – in generale – il sistema di riferimento mondo non coincide con il sistema di riferimento standard della fotocamera, bisogna introdurre la trasformazione rigida che lega i due sistemi di riferimento.

Introduciamo dunque un cambio di coordinate costituito da una rotazione R (si veda § A3.1 sulla rappresentazione delle rotazioni) seguita da una traslazione \mathbf{t} , ed indichiamo con \mathbf{M}_c le coordinate omogenee di un punto nel sistema di riferimento standard della fotocamera e con \mathbf{M} le coordinate omogenee dello stesso punto nel sistema di riferimento mondo. Possiamo dunque scrivere:

$$\mathbf{M}_c = G\mathbf{M} \quad (4.13)$$

dove

$$G = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}. \quad (4.14)$$

Essendo, per la (4.9)

$$\mathbf{m} \simeq K[I|\mathbf{0}]\mathbf{M}_c \quad (4.15)$$

sostituendo la (4.13) si ha:

$$\mathbf{m} \simeq K[I|\mathbf{0}]G\mathbf{M}, \quad (4.16)$$

e dunque

$$P = K[I|\mathbf{0}]G \quad (4.17)$$

Questa è la forma più generale della matrice di proiezione prospettica: la matrice G codifica i *parametri estrinseci* (o esterni) della fotocamera, la matrice K codifica i parametri intrinseci, mentre la matrice $[I|\mathbf{0}]$ rappresenta una trasformazione prospettica in coordinate *normalizzate* nel sistema di riferimento standard. I parametri estrinseci sono dunque sei: tre per specificare la rotazione (secondo la parametrizzazione scelta, vedi § A3.1) e tre per la traslazione.

A seconda della convenienza, potremo anche considerare la seguente fattorizzazione:

$$P = K[R|\mathbf{t}] \quad (4.18)$$

che si ottiene sostituendo nella precedente la forma a blocchi di G e sviluppando il prodotto con la matrice centrale.

Ponendo

$$\mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad e \quad R = \begin{bmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \\ \mathbf{r}_3^\top \end{bmatrix} \quad (4.19)$$

si ottiene la seguente espressione per P in funzione degli elementi delle matrici K e G :

$$P = \begin{bmatrix} \alpha_u \mathbf{r}_1^\top - \frac{\alpha_u}{\tan \theta} \mathbf{r}_2^\top + u_0 \mathbf{r}_3^\top & \alpha_u t_1 - \frac{\alpha_u}{\tan \theta} t_2 + u_0 t_3 \\ \frac{\alpha_v}{\sin \theta} \mathbf{r}_2^\top + v_0 \mathbf{r}_3^\top & \frac{\alpha_v}{\sin \theta} t_2 + v_0 t_3 \\ \mathbf{r}_3^\top & t_3 \end{bmatrix} \quad (4.20)$$

Se assumiamo $\theta = \pi/2$ si ottiene una forma più semplice:

$$P = \begin{bmatrix} \alpha_u \mathbf{r}_1^\top + u_0 \mathbf{r}_3^\top & \alpha_u t_1 + u_0 t_3 \\ \alpha_v \mathbf{r}_2^\top + v_0 \mathbf{r}_3^\top & \alpha_v t_2 + v_0 t_3 \\ \mathbf{r}_3^\top & t_3 \end{bmatrix}. \quad (4.21)$$

Fattore di scala. Una MPP è definita a meno di un fattore di scala arbitrario. Infatti, se si sostituisce P con λP nella (4.6) si ottiene la medesima proiezione, per ogni reale λ non nullo. Quindi, se si vuole scrivere una MPP generica come nella (4.20) è evidentemente necessario riscalarla in modo che il vettore di tre elementi che corrisponde a \mathbf{r}_3 abbia norma uno. Chiameremo **normalizzata** una tale matrice (Atenzione: niente a che vedere con le coordinate normalizzate).

Gradi di libertà. La matrice P è composta da 12 elementi ma possiede 11 gradi di libertà poiché ne perde uno a causa del fattore di scala. D'altra parte dipende da $3+3+5=11$ parametri indipendenti (i conti tornano!).

4.1.3 Proprietà

Studiamo ora alcune interessanti proprietà della MPP. Le formule di questo paragrafo, per la loro importanza, saranno spesso richiamate nel resto del corso.

Forma cartesiana Se scriviamo la MPP secondo le sue righe:

$$P = \begin{bmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \\ \mathbf{p}_3^\top \end{bmatrix}. \quad (4.22)$$

e la inseriamo nell (4.6) otteniamo

$$\mathbf{m} \simeq \begin{bmatrix} \mathbf{p}_1^\top \mathbf{M} \\ \mathbf{p}_2^\top \mathbf{M} \\ \mathbf{p}_3^\top \mathbf{M} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \\ \mathbf{p}_3^\top \end{bmatrix} \mathbf{M} \quad (4.23)$$

e quindi l'equazione di proiezione prospettica, in coordinate cartesiane, diventa:

$$\begin{cases} u = \frac{\mathbf{p}_1^\top \mathbf{M}}{\mathbf{p}_3^\top \mathbf{M}} \\ v = \frac{\mathbf{p}_2^\top \mathbf{M}}{\mathbf{p}_3^\top \mathbf{M}}. \end{cases} \quad (4.24)$$

Si tratta delle generalizzazione della (4.2).

Centro ottico. Si consideri la (4.24). Il piano focale contiene i punti che si proiettano all’infinito e dunque ha equazione $\mathbf{p}_3^\top \mathbf{M} = 0$. I piani di equazione $\mathbf{p}_1^\top \mathbf{M} = 0$ e $\mathbf{p}_2^\top \mathbf{M} = 0$ si proiettano nell’immagine sugli assi $u = 0$ e $v = 0$ rispettivamente.

Con riferimento alla figura 4.1 si intuisce come il centro ottico C sia definito dalla intersezione di questi tre piani:

$$\begin{cases} \mathbf{p}_1^\top \mathbf{C} = 0 \\ \mathbf{p}_2^\top \mathbf{C} = 0 \\ \mathbf{p}_3^\top \mathbf{C} = 0 \end{cases} \quad (4.25)$$

ovvero $P\mathbf{C} = \mathbf{0}$. Dunque, il centro ottico è il nucleo di P , che, per il teorema rango-nullità (teorema 1.24) ha dimensione uno. Poiché $\mathbf{0}$ non rappresenta alcun punto in coordinate omogenee ciò è coerente con il fatto che la proiezione del centro ottico non è definita.

Scriviamo ora la MPP esplicitando la sottomatrice $3 \times 3 Q$:

$$P = [Q|\mathbf{q}]. \quad (4.26)$$

e, ricordando che

$$\mathbf{C} = \begin{bmatrix} \tilde{\mathbf{C}} \\ 1 \end{bmatrix} \quad (4.27)$$

otteniamo quindi

$$Q\tilde{\mathbf{C}} + \mathbf{q} = \mathbf{0} \quad (4.28)$$

da cui

$$\tilde{\mathbf{C}} = -Q^{-1}\mathbf{q} \quad (4.29)$$

Raggio ottico. Il raggio ottico del punto m è la linea retta che passa per il centro ottico C ed m stesso, ovvero il luogo geometrico dei punti $\mathbf{M} : \mathbf{m} \simeq P\mathbf{m}$. Sul raggio ottico giacciono tutti i punti dello spazio dei quali il punto m è proiezione.

Un punto che appartiene a questa retta è il centro ottico C , per definizione. Un altro è il punto (ideale)

$$\begin{bmatrix} Q^{-1}\mathbf{m} \\ 0 \end{bmatrix},$$

infatti basta proiettare tale punto per verificare che:

$$P \begin{bmatrix} Q^{-1}\mathbf{m} \\ 0 \end{bmatrix} = QQ^{-1}\mathbf{m} = \mathbf{m}. \quad (4.30)$$

L'equazione parametrica del raggio ottico è dunque la seguente (vedi appendice 2)

$$\mathbf{M} = \mathbf{C} + \lambda \begin{bmatrix} Q^{-1}\mathbf{m} \\ 0 \end{bmatrix}, \quad \lambda \in \mathbb{R} \cup \{\infty\} \quad (4.31)$$

4.2 Calibrazione

La *calibrazione* consiste nel misurare con accuratezza i parametri intrinseci ed estrinseci del modello della fotocamera. Poichè questi parametri governano il modo in cui punti dello spazio si proiettano sulla retina, l'idea è che conoscendo le proiezioni di punti 3D di coordinate note (punti di calibrazione), sia possibile ottenere i parametri incogniti risolvendo le equazioni della proiezione prospettica. Alcuni metodi diretti di calibrazione, come quello di [Caprile e Torre, 1990] e quello di [Tsai, 1987] formulano il problema della calibrazione identificando come incognite i parametri della fotocamera. Il metodo di calibrazione che illustreremo, tratto da [Faugeras, 1993], risolve invece il problema di stimare la MPP, dalla quale successivamente si possono ricavare i parametri intrinseci ed estrinseci. Nel capitolo 13 vedremo anche un'altro metodo di calibrazione, più pratico.

4.2.1 Metodo DLT

I punti di calibrazione sono gli angoli dei quadrati della scacchiera di figura 4.5 e le loro coordinate sono note per costruzione (il sistema di riferimento viene fissato solidale con l'oggetto, come illustrato in figura 4.5). Le coordinate delle immagini dei punti di calibrazione possono essere ricavate con metodi classici della elaborazione di immagini. La precisione con la quale vengono localizzati tali punti influenza in modo sensibile l'accuratezza della calibrazione.

Dati n punti di calibrazione *non coplanari*, ciascuna corrispondenza tra un punto dell'immagine $\mathbf{m}_i = [u_i, v_i, 1]^\top$, ed il punto della scena \mathbf{M}_i fornisce una coppia di equazioni (dalla (4.24)):

$$\begin{cases} \mathbf{p}_1^\top \mathbf{M}_i - u_i \mathbf{p}_3^\top \mathbf{M}_i = 0 \\ \mathbf{p}_2^\top \mathbf{M}_i - v_i \mathbf{p}_3^\top \mathbf{M}_i = 0 \end{cases} \quad (4.32)$$

In forma matriciale:

$$\underbrace{\begin{bmatrix} \mathbf{M}_i^\top & \mathbf{0} & -u_i \mathbf{M}_i^\top \\ \mathbf{0} & -\mathbf{M}_i^\top & v_i \mathbf{M}_i^\top \end{bmatrix}}_A \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = \mathbf{0}_{2 \times 1} \quad (4.33)$$

Per n punti otteniamo un sistema di $2n$ equazioni lineari omogenee, che possiamo scrivere come

$$A \text{ vec}(P^\top) = \mathbf{0} \quad (4.34)$$

dove A è la matrice $2n \times 12$ dei coefficienti e dipende dalle coordinate dei punti di calibrazione, mentre il vettore delle incognite $\text{vec}(P^\top)$ contiene i 12 elementi di P letti per righe[†]. In teoria, quindi, sei punti (non coplanari) sono sufficienti per il calcolo di P ; nella pratica sono disponibili – ed è consigliabile usare – molti più punti per compensare gli inevitabili errori di misura. Il sistema 4.34 viene dunque risolto ai minimi quadrati. Come ricordato nell'appendice 1, la soluzione è l'autovettore associato al minimo autovalore di $A^\top A$, che si può calcolare sfruttando la SVD di A .

Questo metodo, chiamato *Direct Linear Transform* (DLT) è implementato nella funzione MATLAB `resect`.

La derivazione del metodo DLT è ancora più immediata se sfruttiamo le proprietà del prodotto di Kronecker (§. A1.12). Date le corrispondenze

[†] L'operatore `vec` trasforma una matrice in un vettore per scansione delle colonne, analogamente all'operatore ":" del MATLAB. Quindi `vec(x) = x(:)`

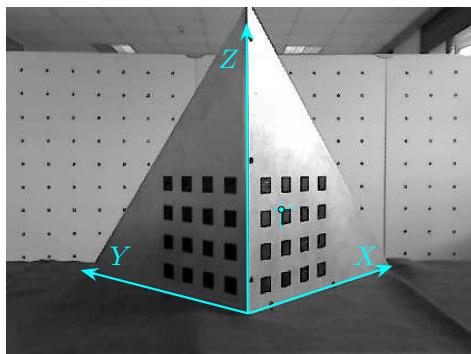


Fig. 4.5. Oggetto di calibrazione con sovrapposto il sistema i riferimento mondo. In questo sistema sono misurate le coordinate dei vertici dei quadratini neri, uno dei quali è evidenziato.

tra punti dell'immagine \mathbf{m}_i e punti della scena \mathbf{M}_i , si richiede di calcolare una matrice P tale che:

$$\mathbf{m}_i \simeq P\mathbf{M}_i \quad i = 1 \dots n \quad (4.35)$$

Per eliminare il fattore di scala si sfrutta il prodotto esterno, riscrivendo l'equazione precedente come:

$$\mathbf{m}_i \times P\mathbf{M}_i = \mathbf{0}. \quad (4.36)$$

Quindi, usando il vec e la (1.51), si deriva:

$$\begin{aligned} \mathbf{m}_i \times P\mathbf{M}_i = \mathbf{0} &\iff [\mathbf{m}_i]_{\times} P\mathbf{M}_i = \mathbf{0} \iff \\ \text{vec}([\mathbf{m}_i]_{\times} P\mathbf{M}_i) = \mathbf{0} &\iff (\mathbf{M}_i^{\top} \otimes [\mathbf{m}_i]_{\times}) \text{vec}(P) = \mathbf{0} \end{aligned} \quad (4.37)$$

Sono tre equazioni in 12 incognite, ma solo due di loro sono indipendenti, infatti il rango di $(\mathbf{M}_i^{\top} \otimes [\mathbf{m}_i]_{\times})$ è due poiché è il prodotto di Kronecker di una matrice di rango uno con una matrice di rango due.

Per ritrovare il risultato precedente trasponiamo l'equazione, ottenendo:

$$([\mathbf{m}_i]_{\times} \otimes \mathbf{M}_i^{\top}) \text{vec}(P^{\top}) = \mathbf{0} \quad (4.38)$$

ed espandendo la matrice dei coefficienti si ottiene:

$$\begin{bmatrix} \mathbf{0}^{\top} & -\mathbf{M}_i^{\top} & v_i \mathbf{M}_i^{\top} \\ \mathbf{M}_i^{\top} & \mathbf{0}^{\top} & -u_i \mathbf{M}_i^{\top} \\ -v_i \mathbf{M}_i^{\top} & u_i \mathbf{M}_i^{\top} & \mathbf{0}^{\top} \end{bmatrix} \text{vec}(P^{\top}) = \mathbf{0}. \quad (4.39)$$

4.2.2 Metodo non lineare

Il metodo lineare appena illustrato è veloce ma minimizza un errore *algebrico* ($\|\mathbf{Ax}\|^2$) che non ha un significato geometrico, dunque è meno stabile e non è invariante per cambio di coordinate. Per ottenere una migliore precisione (dovuta alla minore sensibilità al rumore nelle coordinate dei punti di calibrazione) è necessario minimizzare un *errore geometrico*, ottenendo però così una funzione obiettivo non lineare. Ad esempio è possibile riformulare il problema della calibrazione in maniera non-lineare, minimizzando la seguente quantità:

$$\varepsilon(P) = \sum_{i=1}^n \left(\frac{\mathbf{p}_1^{\top} \mathbf{M}_i}{\mathbf{p}_3^{\top} \mathbf{M}_i} - u_i \right)^2 + \left(\frac{\mathbf{p}_2^{\top} \mathbf{M}_i}{\mathbf{p}_3^{\top} \mathbf{M}_i} - v_i \right)^2. \quad (4.40)$$

Si tratta della somma dei quadrati delle distanze tra i punti m_i e la proiezione dei punti M_i nell'immagine.

Vengono impiegate tecniche di risoluzione di problemi di minimi quadrati non-lineari (come Gauss-Newton), le quali convergono solo localmente al minimo globale, dunque necessitano di una stima iniziale della soluzione, che può essere ottenuta con un metodo lineare.

4.2.3 Estrazione dei parametri

Data una matrice 3×4 di rango pieno, questa si può decomporre come

$$P \simeq K[R|\mathbf{t}]. \quad (4.41)$$

Si tratta dunque di ricavare K, R , e \mathbf{t} data $P = [Q|\mathbf{q}]$. Concentriamoci sulla sottomatrice Q : poiché $P = [KR|K\mathbf{t}]$, per confronto si ha $Q = KR$ con K triangolare superiore ed R ortogonale (rotazione). Sia

$$Q^{-1} = US \quad (4.42)$$

la fattorizzazione QR di Q^{-1} , con U ortogonale e S triangolare superiore. Essendo $Q^{-1} = R^{-1}K^{-1}$, basta porre

$$R = U^{-1} \text{ e } K = S^{-1}. \quad (4.43)$$

Per ricavare \mathbf{t} basta calcolare $\mathbf{t} = K^{-1}\mathbf{q} = S\mathbf{q}$.

Poiché P può contenere un fattore di scala arbitrario che si riflette su K (perché non su R ?) la scala corretta di K si ottiene imponendo $K(3,3) = 1$.

Il metodo è implementato dalla funzione `art`.

4.2.4 Distorsione radiale

Un modello più accurato della fotocamera deve tenere conto della distorsione radiale delle lenti, specialmente per ottiche a focale corta. Il modello standard è una trasformazione dalle coordinate ideali (non distorte) (u, v) alle coordinate reali osservabili (distorte) (\hat{u}, \hat{v}) :

$$\begin{cases} \hat{u} = (u - u_0)(1 + k_1 r_d^2) + u_0 \\ \hat{v} = (v - v_0)(1 + k_1 r_d^2) + v_0 \end{cases}. \quad (4.44)$$

dove $r_d^2 = \left(\frac{(u-u_0)}{\alpha_u}\right)^2 + \left(\frac{(v-v_0)}{\alpha_v}\right)^2$ e (u_0, v_0) sono le coordinate del centro dell'immagine.

Vediamo ora come si procede alla calibrazione della distorsione radiale. Siano $\hat{\mathbf{m}}$ le coordinate di un pixel (distorte) nell'immagine reale,

ed \mathbf{M} le coordinate del corrispondente punto (in metri) nella griglia di calibrazione.

Esiste una MPP (che assumiamo nota, per ora) che trasforma i punti della griglia su punti dell'immagine. Questi costituiscono i punti ideali (non distorti) \mathbf{m} :

$$\mathbf{m} \simeq P\mathbf{M} \quad (4.45)$$

In assenza di distorsione radiale i punti ideali coincidono con quelli $\hat{\mathbf{m}}$ osservati nell'immagine. Tuttavia, a causa di quest'ultima, i punti osservati subiscono una distorsione data dalla (4.44). L'obiettivo è ricavare il coefficiente di distorsione radiale k_1 . Per ogni punto abbiamo dunque due equazioni:

$$\begin{cases} (u - u_0) \left(\left(\frac{(u - u_0)}{\alpha_u} \right)^2 + \left(\frac{(v - v_0)}{\alpha_v} \right)^2 \right) k_1 = \hat{u} - u \\ (v - v_0) \left(\left(\frac{(u - u_0)}{\alpha_u} \right)^2 + \left(\frac{(v - v_0)}{\alpha_v} \right)^2 \right) k_1 = \hat{v} - v \end{cases} \quad (4.46)$$

nella incognita k_1 , assumendo che i parametri intrinseci $u_0, v_0, \alpha_u, \alpha_v$ siano noti. I punti ideali si ottengono dalla (4.45), dati la MPP P e i punti del modello \mathbf{M} .

Un punto solo basterebbe a determinare k_1 . In realtà abbiamo a disposizione molti punti, e dunque bisogna risolvere un sistema sovradeterminato in una incognita. In questo caso la soluzione si ottiene con una semplice formula (che è la specializzazione al caso scalare della soluzione ai minimi quadrati con la pseudoinversa): chiamiamo a_i il coefficiente di

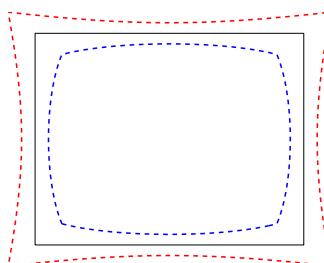


Fig. 4.6. Distorsione radiale a cuscino (linea rossa tratteggiata interna) e a botte (linea blu tratteggiata esterna), ottenute rispettivamente con $k_1 = 0.4$ e $k_1 = -0.4$.

k_1 nella i -esima equazione e b_i il termine noto della i -esima equazione, allora

$$k_1 = \frac{\sum a_i b_i}{\sum a_i^2} \quad (4.47)$$

4.2.4.1 Modello inverso della distorsione radiale

Una volta che k_1 sia noto, possiamo compensare la distorsione radiale nella immagine acquisita dalla fotocamera, invertendo la trasformazione. Usando il cambiamento di coordinate $x = \frac{(u-u_0)}{\alpha_u}$ $y = \frac{(v-v_0)}{\alpha_v}$, la trasformazione diretta – data dalla (4.44) – si riscrive

$$\begin{cases} \hat{x} = x(1 + k_1(x^2 + y^2)) \\ \hat{y} = y(1 + k_1(x^2 + y^2)) \end{cases} \quad (4.48)$$

Per invertirla si deve risolvere il sistema di due equazioni di terzo grado nelle incognite x ed y :

$$\begin{cases} x + k_1x^3 + k_1xy^2 - \hat{x} = 0 \\ y + k_1y^3 + k_1x^2y - \hat{y} = 0 \end{cases} . \quad (4.49)$$

Il sistema si può riscrivere in forma vettoriale come $F(\mathbf{a}) = \mathbf{0}$ dove $\mathbf{a} = (x, y)$, $\mathbf{0} = (0, 0)$ ed F è una funzione vettoriale, il cui valore in \mathbf{a} (un vettore di due elementi) è dato dalle parti sinistre delle due equazioni (4.49) valutate in \mathbf{a} . Usiamo il metodo di Netwon per la soluzione del sistema di equazioni: supponiamo che \mathbf{a} sia la soluzione corrente, e cerchiamo l'incremento $\Delta\mathbf{a}$ che sortisce una soluzione migliore. Usando l'approssimazione di Taylor, vale che

$$F(\mathbf{a} + \Delta\mathbf{a}) \approx F(\mathbf{a}) + J(\mathbf{a})\Delta\mathbf{a} \quad (4.50)$$

dove J è la jacobiana di F , ed è una matrice 2×2 (in questo caso) le cui componenti sono le derivate parziali di F :

$$J(\mathbf{a}) = \begin{bmatrix} \frac{\partial F_1}{\partial x} & \frac{\partial F_1}{\partial y} \\ \frac{\partial F_2}{\partial x} & \frac{\partial F_2}{\partial y} \end{bmatrix} = \begin{bmatrix} 1 + 3k_1x^2 + k_1y^2 & 2k_1xy \\ 2k_1xy & 1 + 3k_1y^2 + k_1x^2 \end{bmatrix} \quad (4.51)$$

Idealmente $F(\mathbf{a} + \Delta\mathbf{a}) = 0$, dunque

$$J(\mathbf{a})\Delta\mathbf{a} = -F(\mathbf{a}) \quad (4.52)$$

quindi l'incremento $\Delta\mathbf{a}$ (un vettore di due elementi) si ottiene risolvendo questo sistema lineare. Si aggiorna la soluzione corrente: $\mathbf{a} = \mathbf{a} + \Delta\mathbf{a}$, e si ripete l'iterazione fino a che l'incremento è sufficientemente piccolo.

Se si desidera avere un tempo di esecuzione fisso, indipendente dai dati, si può stabilire a priori il numero di iterazioni da eseguire, anziché basare la terminazione su $\Delta\mathbf{a}$.

Ritorniamo ora al problema principale del calcolo della calibrazione con calcolo distorsione radiale. Abbiamo assunto che la MPP P nella (4.45) fosse nota, cosa che in realtà non è. Essa infatti dipende dai parametri di calibrazione della fotocamera stenopeica, che vogliamo stimare, ma finché non rimuoviamo la distorsione radiale il modello stenopeico non si applica, quindi i parametri stimati sono affetti da errore (in altri termini, non possiamo conoscere i punti ideali non distorti, quello che misuriamo sono punti affetti da distorsione radiale). Si procede dunque iterativamente, usando i parametri della fotocamera stenopeica imperfetti per stimare la distorsione radiale e quindi calcolare una migliore approssimazione del modello stenopeico e così via, fino alla convergenza.

Esercizi ed approfondimenti

- 4.1 Si consideri un cerchio nello spazio, di raggio r , ortogonale all'asse ottico e centrato sull'asse ottico: $\tilde{\mathbf{M}} = (r\cos\theta, r\sin\theta, Z_0)$. Se il riferimento mondo coincide con il riferimento standard della fotocamera, la sua proiezione è: $\tilde{\mathbf{m}} = (\frac{r\cos\theta}{Z_0}, \frac{r\sin\theta}{Z_0})$, ovvero ancora un cerchio. Se traslo il cerchio nello spazio traslo lungo X , la sua proiezione è ancora un cerchio? Prima di risolvere analiticamente, provate a fare delle congetture.

- 4.2 L'equazione (4.6) con il fattore di scala esplicitato si scrive:

$$\zeta \mathbf{m} = P \mathbf{M}. \quad (\text{E4.1})$$

Dimostrate che se P è normalizzata ζ è pari alla distanza del punto M dal piano focale (o profondità).

Soluzione. Nella (4.24) si ha che

$$\zeta = \mathbf{p}_3^\top \tilde{\mathbf{M}}. \quad (\text{E4.2})$$

Se la MPP è normalizzata, e quindi si scrive come (4.20), si vede facilmente che

$$\zeta = \mathbf{r}_3^\top \tilde{\mathbf{M}} + t_3. \quad (\text{E4.3})$$

Questa è la terza coordinata di M espresso nel riferimento stan-

dard della fotocamera (ricordiamo che $\mathbf{M}_c = G\tilde{\mathbf{M}}$) ovvero la *la distanza del punto M dal piano focale*.

- 4.3 Dimostrare che se P è normalizzata, il parametro λ nella equazione del raggio ottico (4.31) è pari alla profondità ζ del punto, come definita nella (E4.3).
- 4.4 In una fotocamera stenopeica, la traslazione del centro lungo l'asse ottico (movimento in avanti/dietro) e la variazione della lunghezza focale (zoom) producono effetti diversi sulla immagine. Perchè? Suggerimento: derivare le coordinate pixel (u, v) rispetto a f ed a t_3 .
- 4.5 Scrivere la funzione MATLAB `camera` che restituisce una MPP dati il centro ottico (*eye point*), un punto sull'asse ottico (*look point*) ed la direzione verticale (*up vector*).

Soluzione. Ricavando \mathbf{q} dalla (4.29), e ricordando che $P = K[R \mid \mathbf{t}]$ si può riscrivere la MPP come

$$P = K[R \mid -R \tilde{\mathbf{C}}]. \quad (\text{E4.4})$$

dove si è messo in evidenza il centro ottico e la matrice di rotazione che specifica la orientazione della fotocamera. In particolare, le righe di R sono gli assi X, Y , e Z del sistema standard della fotocamera espressi nel sistema di riferimento mondo. Dunque non rimane che fissare:

- (a) L'asse Z che passa per il centro e per il *look point*;
 - (b) L'asse X ortogonale a Z ed alla direzione verticale;
 - (c) L'asse Y ortogonale a XZ (necessariamente).
- 4.6 La proiezione prospettica di un fascio di rette parallele nello spazio è, in generale, un fascio di rette nel piano che passano per un punto comune, chiamato **punto di fuga**. Trovare le coordinate del punto di fuga per una una direzione \mathbf{n} data.
- 4.7 Dimostrare che il punto principale è il baricentro del triangolo individuato dai punti di fuga di tre direzioni ortogonali nello spazio. Questa osservazione è parte del metodo di calibrazione di Caprile e Torre [1990].
- 4.8 Nella compensazione della distorsione radiale il modello inverso serve solo per compensare la posizione di punti isolati. Se invece intendo deformare l'immagine intera per compensare globalmente la distorsione, uso la scansione della immagine destinazione (vedi § 14.1.2), e dunque basta la trasformazione diretta.

Bibliografia

- Caprile B.; Torre V. (1990). Using vanishing points for camera calibration. *International Journal of Computer Vision*, **4**, 127–140.
- Faugeras O. (1993). *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, MA.
- Tsai R. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, **3**(4), 323–344.

5

Chiaroscuro, tessitura, sfocamento

In questo capitolo sono trattate alcune tecniche di recupero della forma da chiaroscuro, tessitura e sfocamento.

5.1 Chiaroscuro

Lo *Shape from Shading* (SFS), è quel processo che, data un'immagine di un oggetto (illuminato) consente di calcolarne la forma, sfruttando le informazioni connesse alla variazione della luminosità della superficie dell'oggetto. Il termine inglese *shading* si traduce con *chiaroscuro* o ombreggiatura o sfumatura[†]. In altri termini, la determinazione della forma dal chiaroscuro può essere pensata come il problema di ricostruire una superficie nello spazio 3D, a partire da un'altra superficie, rappresentante la luminosità sul piano immagine della fotocamera.

Come suggerisce la figura 5.1, la distribuzione dei livelli di grigio nell'immagine (il chiaroscuro) reca con sè informazione utile riguardante la forma della superficie e la direzione di illuminazione.

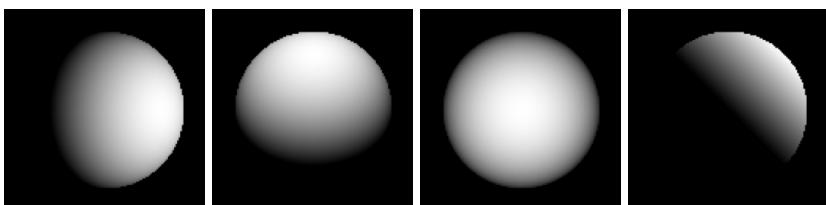


Fig. 5.1. Sfere lambertiane illuminate da diverse direzioni.

Fissiamo, per comodità, il sistema di riferimento 3D XYZ con Z che

[†] Il chiaroscuro è il procedimento pittorico che, usando il bianco, il nero, e le gradazioni intermedie, serve a riprodurre il passaggio graduale dalla luce all'ombra.

punta *dietro* alla fotocamera (Y punta verso l'alto e X a destra dell'osservatore)‡. I parametri intrinseci sono noti, e dunque le coordinate x, y nell'immagine sono normalizzate. Il piano immagine xy è parallelo al piano XY .

La luminosità del punto (x, y) sul piano immagine è uguale alla radianza L del punto (X, Y, Z) della scena che si proietta in (x, y) :

$$I(x, y) = L(X, Y, Z) \quad (5.1)$$

La radianza nel punto (X, Y, Z) , a sua volta, dipende dalla forma (dalla normale) e dalla riflettanza della superficie (e dalle sorgenti luminose).

Assumiamo che la **superficie sia lambertiana**, ovvero

$$L(X, Y, Z) = \rho \mathbf{s}^\top \mathbf{n} \quad (5.2)$$

dove ρ è l'albedo efficace (ingloba la radianza incidente), \mathbf{s} è il versore della direzione di illuminazione (punta nella direzione della sorgente luminosa), \mathbf{n} è il versore della normale; tutte queste quantità dipendono, in principio, dal punto (X, Y, Z) considerato. Se assumiamo **l'albedo costante e l'illuminazione parallela** (sorgente puntiforme a distanza infinita), allora solo la normale dipende dal punto (X, Y, Z) , ovvero

$$L(X, Y, Z) = \rho \mathbf{s}^\top \mathbf{n}(X, Y, Z) = R(\mathbf{n}(X, Y, Z)). \quad (5.3)$$

dove si è introdotta R , la *mappa di riflettanza* $R(\mathbf{n})$ che esprime la luminosità della superficie in un punto in funzione della sua normale. Anche se non è specificato, per non appesantire la notazione, bisogna ricordare che R dipende anche da \mathbf{s} e ρ .

L'equazione fondamentale dello SFS, che lega l'intensità dell'immagine alla superficie nello spazio è la seguente:

$$I(x, y) = R(\mathbf{n}(X, Y, Z)) \quad (5.4)$$

dove (x, y) è il punto immagine in cui si proietta (X, Y, Z) . L'equazione, che prende anche il nome di *vincolo di luminosità*, dice che **tutti i punti con la stessa normale hanno lo stesso livello di grigio**. Il calcolo dello SFS procede cercando di ricavare le normali alla superficie $\mathbf{n}(X, Y, Z)$ da questa equazione. Si assume di conoscere albedo e direzione di illuminazione. Qualora questi non fossero noti a priori, si possono calcolare dalla immagine con una tecnica che verrà illustrata nel seguito.

Osserviamo immediatamente che il problema è localmente sottovincolato. Dal vincolo di luminosità (5.4) possiamo calcolare l'angolo tra \mathbf{n} e

‡ Facciamo in modo che le normali che puntano verso la fotocamera abbiano proiezione positiva sull'asse Z .

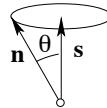


Fig. 5.2. Ambiguità nella determinazione della normale in base alla riflettanza.

\mathbf{s} , ma questo costringe solo \mathbf{n} a giacere su di un cono di direzioni, con asse \mathbf{s} ed angolo al vertice $\theta = \text{acos}(I(x, y)/\rho)$ (vedi figura 5.2).

Esiste comunque un vincolo che le normali devono soddisfare per poter rappresentare una superficie: il *vincolo di integrabilità*. In sostanza, la normale non può variare arbitrariamente da un punto al suo vicino, ma deve farlo in modo regolare (tutto questo può essere reso preciso dal punto di vista matematico).

Espressioni per R . La forma della superficie dell'oggetto può essere specificata tramite le normali \mathbf{n} in ogni punto, il gradiente $[p, q]^\top$ o la profondità $Z = Z(X, Y)$. A seconda di come si descrive la superficie e di come si esprime la direzione di illuminazione R si può esprimere in modi diversi.

Assumiamo che il modello geometrico della formazione dell'immagine sia la **proiezione ortografica**, dunque:

$$x = \frac{-f}{Z_0} X \quad y = \frac{-f}{Z_0} Y \quad (5.5)$$

dove Z_0 è la distanza media della superficie dal piano immagine. Trascurando un fattore di scala si può scrivere:

$$x = X \quad y = Y. \quad (5.6)$$

Dunque si può riferire la superficie alle coordinate immagine: $Z = Z(x, y)$. Naturalmente queste altezze non hanno alcun significato assoluto se non si conosce $\frac{-f}{Z_0}$.

Consideriamo le derivate parziali rispetto x ed y in un punto della superficie $[x, y, Z(x, y)]$, ovvero $[1, 0, \frac{\partial Z}{\partial x}]$ e $[0, 1, \frac{\partial Z}{\partial y}]$. Sappiamo che questi due vettori appartengono al piano tangente alla superficie nel punto, dunque la normale si ottiene come prodotto esterno (normalizzato) tra i due vettori:

$$\mathbf{n} = \frac{[-p, -q, 1]^\top}{\sqrt{1 + p^2 + q^2}}, \quad (5.7)$$

dove $[p, q]^\top$ è il gradiente nel punto, ovvero

$$p = \frac{\partial Z(x, y)}{\partial x} \quad \text{e} \quad q = \frac{\partial Z(x, y)}{\partial y}. \quad (5.8)$$

Si veda [Trucco e Verri, 1998](A.5) per un richiamo sulla geometria differenziale.

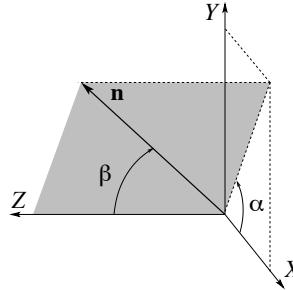


Fig. 5.3. Tilt e slant della normale.

Per esprimere la normale possiamo anche usare due angoli: il *tilt* e lo *slant*. Il *tilt* α (anche chiamato azimut), è l'angolo tra la proiezione di \mathbf{n} sul piano XY e l'asse X , mentre lo *slant* β è l'angolo tra \mathbf{n} e l'asse Z (figura 5.3). Segue che:

$$\mathbf{n} = [\cos \alpha \sin \beta, \sin \alpha \sin \beta, \cos \beta]^\top. \quad (5.9)$$

Anche il versore della direzione di illuminazione \mathbf{s} si può esprimere con i suoi angoli di tilt τ e slant σ . È comune trovare R espresso in funzione di p e q , come nella seguente espressione:

$$R(p, q) = \frac{-p \cos \tau \sin \sigma - q \sin \tau \sin \sigma + \cos \sigma}{\sqrt{1 + p^2 + q^2}} \quad (5.10)$$

dove σ e τ sono rispettivamente lo slant ed il tilt della direzione di illuminazione \mathbf{s} .

La figura 5.4 mostra i grafici di due mappe di riflettanza per due diverse direzioni di illuminazione. Grazie al disegno delle curve di isoluminosità (il valore di R) si nota di nuovo che molti valori di p e q (direzione della normale) forniscono la stessa luminosità.

Un'altra forma, che useremo nel seguito, si ottiene se sia la normale che la direzione di illuminazione sono espressi tramite tilt e slant:

$$R(\alpha, \beta) = \rho(\cos \tau \sin \sigma \cos \alpha \sin \beta + \sin \tau \sin \sigma \sin \alpha \sin \beta + \cos \sigma \cos \beta). \quad (5.11)$$

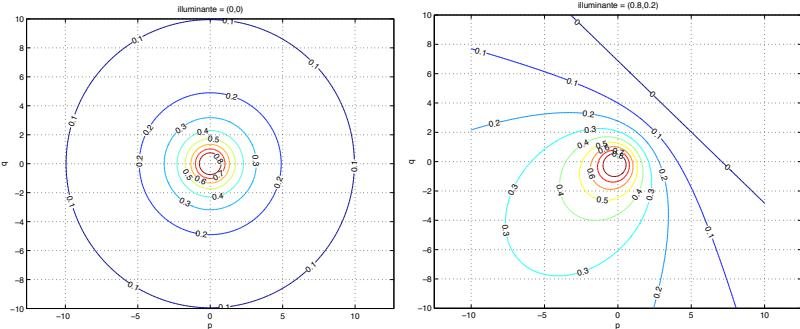


Fig. 5.4. Curve di livello della mappa di riflettanza $R(p, q)$ per due diversi valori della direzione di illuminazione.

5.1.1 Algoritmi di stima della forma dal chiaroscuro

Uno dei primi approcci allo SFS, proposto in [Horn e Ikeuchi, 1981], riconoscendo che il problema è mal posto, applica il metodo della regolarizzazione, che in pratica si riduce nell'aggiungere al vincolo di luminosità un termine di regolarizzazione (*smoothness constraint*) che penalizza soluzioni poco regolari:

$$\min \int \int [I(x, y) - R(p, q)]^2 + \lambda(p_x^2 + p_y^2 + q_x^2 + q_y^2) \ dx \ dy \quad (5.12)$$

Le soluzioni ottenute da questo algoritmo non sono soddisfacenti; inoltre è richiesta la conoscenza a priori del gradiente lungo i bordi occludenti dell'oggetto.

Seguendo [Tsai e Shah, 1994] invece, si parte dall'equazione fondamentale dello SFS:

$$I(x, y) = R(p, q) \quad (5.13)$$

e si approssimano le componenti del gradiente p e q nel seguente modo:

$$\begin{aligned} p &= \frac{\partial Z(x, y)}{\partial x} = Z(x, y) - Z(x - 1, y) \\ q &= \frac{\partial Z(x, y)}{\partial y} = Z(x, y) - Z(x, y - 1) \end{aligned} \quad (5.14)$$

ottenendo:

$$\begin{aligned} 0 &= I(x, y) - R(Z(x, y) - Z(x - 1, y), Z(x, y) - Z(x, y - 1)) \\ &= f(I(x, y), Z(x, y), Z(x - 1, y), Z(x, y - 1)). \end{aligned} \quad (5.15)$$

Questa equazione (si tratta del solito *brightness constraint*) vale per

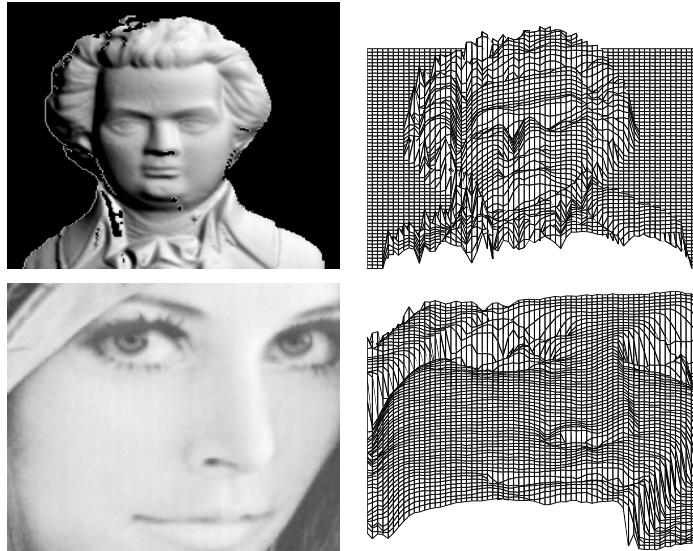


Fig. 5.5. Esempi di risultati ottenuti con l'algoritmo di Tsai-Shah: sono mostrate le immagini di partenza e le superfici ricostruite.

ogni punto (x, y) dell'immagine: se assumiamo di avere già calcolato $Z(x - 1, y)$ e $Z(x, y - 1)$, possiamo riscrivere la (5.15) come:

$$0 = f(Z(x, y)) \quad (5.16)$$

Questa è una equazione non lineare che può essere risolta con il metodo iterativo di Newton. Partendo da un piano, ovvero considerando le altezze iniziali $Z^0(x, y) = 0$, l'altezza del punto (x, y) si calcola iterativamente con:

$$Z^n(x, y) = Z^{n-1}(x, y) + \frac{-f(Z^{n-1}(x, y))}{\partial f(Z^{n-1}(x, y)) / \partial Z(x, y)} \quad (5.17)$$

Nel calcolo di f uso $Z(x - 1, y) = Z^{n-1}(x - 1, y)$ e $Z(x, y - 1) = Z^{n-1}(x, y - 1)$.

L'approssimazione introdotta da questo metodo è duplice: l'approssimazione discreta del gradiente e la linearizzazione della funzione f (ovvero della funzione di riflettanza). L'applicazione del metodo di Newton, infatti, implicitamente contiene l'approssimazione lineare (locale) della funzione.

Si noti che il procedimento non fa assunzioni sulla forma di f : è

possibile impiegare una mappa di riflettanza del tutto generica (non lambertiana).

5.1.2 Stima della direzione di illuminazione

In questo paragrafo ci occupiamo di estrarre dall'immagine un'informazione di cui qualsiasi algoritmo di SFS necessita: la direzione di illuminazione (e l'albedo). Senza di essi, infatti, la mappa di riflettanza non sarebbe nota. Studieremo l'algoritmo di Zheng e Chellappa [1991] (versione semplificata come in [Trucco e Verri, 1998]) che calcola la direzione di illuminazione e l'albedo di una immagine, nell'ipotesi che siano valide le assunzioni fatte precedentemente.

L'algoritmo, inoltre, necessita di fare alcune ipotesi sulla *distribuzione* statistica delle normali della superficie visibile. In sostanza, se consideriamo una scena in cui tutte le normali sono distribuite uniformemente (assunzione forte), le normali delle superfici proiettate sul piano immagine *non* sono equamente distribuite. Infatti è evidente che, a parità di area sulla superficie, una zona la cui normale punta verso l'osservatore ha un'area proiettata maggiore rispetto ad una la cui normale è inclinata rispetto all'asse Z (se è ortogonale all'asse la proiezione è nulla).

L'effetto è illustrato in figura 5.6, dove è disegnata la *sfera gaussiana*: ciascun punto della superficie in esame viene associato al punto della sfera che possiede la stessa normale.

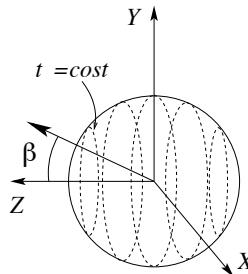


Fig. 5.6. Sfera gaussiana delle normali alla superficie (Z punta verso l'osservatore). L'effetto di riduzione dell'area proiettata dipende solo da β .

Quantitativamente, se α e β sono rispettivamente il tilt e lo slant della normale, la distribuzione rispetto al tilt è uniforme:

$$f_\alpha = \frac{1}{2\pi} \quad (5.18)$$

mentre, per quanto detto, la distribuzione rispetto allo slant è:

$$f_\beta = \cos(\beta). \quad (5.19)$$

Perveniamo dunque alla distribuzione $\mathcal{P}(\alpha, \beta) = \frac{\cos \beta}{2\pi}$ per le normali nella immagine.

Usando la distribuzione delle normali e la mappa di riflettanza lambertiana, calcoliamo la media del livello di grigio dei pixel nell'immagine.

$$\langle I \rangle = \int_0^{2\pi} d\alpha \int_0^{\pi/2} d\beta \mathcal{P}(\alpha, \beta) I(\alpha, \beta) \quad (5.20)$$

dove, usando la (5.11)

$$\begin{aligned} I(\alpha, \beta) &= R(\alpha, \beta) = \\ &\rho(\cos \tau \sin \sigma \cos \alpha \sin \beta + \sin \tau \sin \sigma \sin \alpha \sin \beta + \cos \sigma \cos \beta) \end{aligned} \quad (5.21)$$

Dalla risoluzione dell'integrale (5.20) otteniamo

$$\langle I \rangle = \frac{\pi}{4} \rho \cos \sigma \quad (5.22)$$

Un discorso analogo può essere fatto per il calcolo della media dei quadrati dei livelli di grigio dell'immagine, $\langle I^2 \rangle$, ottenendo

$$\langle I^2 \rangle = \frac{1}{6} \rho^2 (1 + 3 \cos^2 \sigma). \quad (5.23)$$

Abbiamo dunque legato lo slant e l'albedo incogniti a quantità misurabili nell'immagine (la media dei livelli di grigio e la media dei quadrati dei livelli di grigio). È possibile, dunque, sfruttando la (5.22) e (5.23), determinare l'*albedo* ρ e lo *slant* σ

$$\rho = \frac{\gamma}{\pi} \quad (5.24)$$

e

$$\cos \sigma = \frac{4\langle I \rangle}{\gamma} \quad (5.25)$$

con

$$\gamma = \sqrt{6\pi^2 \langle I^2 \rangle - 48\langle I \rangle^2} \quad (5.26)$$

Per calcolare τ si sfruttano le derivate spaziali dell'immagine, I_x e I_y mediante la

$$\tan \tau = \frac{\langle \hat{I}_y \rangle}{\langle \hat{I}_x \rangle}. \quad (5.27)$$

dove

$$[\hat{I}_y \hat{I}_y]^\top = \frac{[I_x \ I_y]^\top}{\sqrt{I_x^2 + I_y^2}}. \quad (5.28)$$

5.1.3 Stereo fotometrico

Lo *stereo fotometrico* si basa sugli stessi principi dello SFS, ma l'ambiguità circa la normale alla superficie in un punto viene risolta cambiando la direzione di illuminazione. Questo equivale ad intersecare le mappe di riflettanza (figura 5.7). Con almeno tre immagini prese con diversa direzione di illuminazione, si risolve il problema.

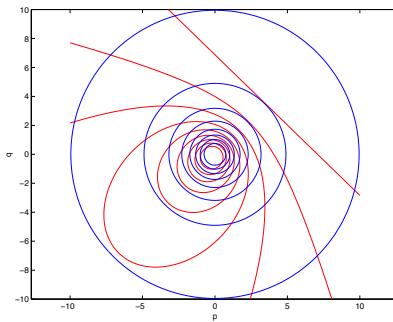


Fig. 5.7. Stereo fotometrico: intersezione delle mappe di riflettanza viste precedentemente.

5.2 Tessitura

La *tessitura* (o *texture*) si riferisce alle rocce ed indica il modo con cui si riuniscono i cristalli che la costituiscono (per esempio diversi tipi di marmo hanno diversa *tessitura*). In visione si riferisce ad un concetto simile, ovvero al modo in cui si dispongono pattern spaziali ripetuti su di una superficie. Si parla anche di *trama*, con riferimento invece ai tessuti. Alcuni esempi sono (figura 5.8): la disposizione dei mattoni sulla facciata di un edificio, le macchie sulla pelle del leopardo, i fili d'erba in un prato, i sassi su una spiaggia, le teste delle persone nella folla, i disegni della tappezzeria. Gli elementi che si ripetono (chiamati anche *texel*) devono essere abbastanza piccoli da non poter essere distinti come oggetti separati (un sasso visto da vicino è un oggetto, quando fa parte della vista di una spiaggia è un *texel*). Alle volte la disposizione è regolare (o

periodica), come nella tappezzeria o nella facciata a mattoni, a volte la regolarità è solo statistica, come nel prato o nella spiaggia. Distinguiamo infatti tra tessiture deterministiche e tessiture statistiche. Spesso le prime derivano dall'opera dell'uomo, mentre le seconde si trovano in natura.



Fig. 5.8. Tessitura deterministica (a sinistra, da <http://www.flickr.com/photos/vsny/172570242/>) e tessitura statistica (a destra, da <http://www.flickr.com/photos/kalyan3/7208334/>).

Anche se i texel sono identici nella scena, in una immagine prospettica la loro dimensione apparente, forma, spaziatura etc... cambia, a causa della proiezione prospettica. Dunque l'immagine prospettica di una tessitura stesa (per esempio) su un piano subirà una distorsione dovuta alla scorciatura, per cui:

- cambia la forma di ciascun texel, in ragione dell'angolo di inclinazione (slant) del piano (i cerchi diventano ellissi, tanto più eccentrici quanto più il piano è inclinato);
- cambia la dimensione apparente dei texel: più lontani sono, più piccoli appaiono (l'area delle ellissi decresce con la distanza).

Questi effetti sono legati alla orientazione del piano e dunque ne costituiscono un indizio. Possiamo risalire alla sua orientazione misurando caratteristiche che quantificano la distorsione (per esempio eccentricità) subita dai texel, oppure la velocità di cambiamento (*gradiente di tessitura*) di certe caratteristiche del texel (per esempio variazione dell'area).

Parliamo di un piano, ma il discorso si può estendere ad una superficie generica che sarà approssimata come planare a tratti (maglia poligonale).

5.2.1 Orientazione del piano da tessitura

Vediamo ora un semplice algoritmo di stima della orientazione di un piano ricoperto da una tessitura statistica. Tratto da [Trucco e Verri,

1998].

Assunzioni:

- I texel sono piccoli segmenti di retta, chiamati *needles* (aghi).
- Gli aghi sono distribuiti uniformemente (nella tessitura originale).
- Consideriamo una sola superficie planare.
- Proiezione ortografica[†] e coordinate normalizzate (intrinseci noti).
- Sistema di riferimento centrato sull'osservatore con asse Z che punta verso il dietro. [‡]

Ricaveremo l'orientazione del piano in termini degli angoli di tilt, τ , e slant, σ . Gli aghi si ottengono passando un estrattore di edge sull'immagine.

Per ciascun ago, consideriamo l'angolo α che esso forma con l'asse X, e definiamo il vettore

$$\mathbf{v} = [\cos(2\alpha), \sin(2\alpha)]. \quad (5.29)$$

L'idea è che si può legare l'orientazione del piano alla distribuzione di questi vettori sul cerchio unitario (vedi figura 5.9). Infatti, il centro di massa (o centroide) dei vettori \mathbf{v} , definito dalle coordinate (C, S) :

$$C = 1/N \sum \cos(2\alpha_i) \quad S = 1/N \sum \sin(2\alpha_i) \quad (5.30)$$

è legato a σ e τ (vale solo per proiezione ortografica):

$$C = \cos(2\tau) \frac{1 - \cos \sigma}{1 + \cos \sigma} \quad S = \sin(2\tau) \frac{1 - \cos \sigma}{1 + \cos \sigma} \quad (5.31)$$

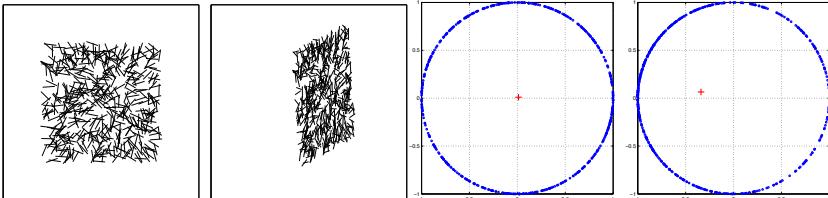


Fig. 5.9. Texture ad aghi originale, ruotata e rispettive distribuzioni dei vettori \mathbf{v} . Si noti che il centroide (croce rossa) risulta traslato nell'ultimo grafico, a causa di un addensamento dei punti nella parte sinistra del cerchio.

[†] Si perde la scorciatura con la distanza ma rimane quella legata all'inclinazione.

[‡] Lo stesso riferimento che nello SFS (paragrafo 5.1). Diverso da [Trucco e Verri, 1998].

5.3 Fuoco

Abbiamo due casi distinti dove si sfruttano le informazioni sulla messa a fuoco per la ricostruzione della profondità: la fochettatura[§] e la misura dello sfocamento.

5.3.1 Fochettatura

Vogliamo determinare la distanza da un punto prendendo diverse immagini con una messa a fuoco sempre migliore, finché il punto che ci interessa è perfettamente a fuoco. Nella letteratura in inglese la tecnica prende il nome di *depth from focus*. Sapere che un punto è a fuoco fornisce informazioni sulla sua profondità tramite la legge delle lenti sottili (3.4). Questo metodo richiede una ricerca in tutte le posizioni del fuoco per ogni punto nell'immagine (lento!).

I problemi chiave nel recupero della distanza tramite fochettatura sono: i) la scelta del criterio di messa a fuoco e ii) la ricerca efficiente della posizione di fuoco ottimo (secondo il criterio fissato).

5.3.1.1 Criterio di fuoco.

Si tratta di operatori che devono avere un massimo in corrispondenza della immagine a fuoco, e decrescere con l'aumentare dello sfocamento. L'idea è di premiare i contributi delle alte frequenze spaziali.

Le tecniche più diffuse sono passate in rassegna in [Subbarao *e al.*, 1993], tra queste citiamo:

L'energia dell'immagine o equivalentemente la varianza dei livelli di grigio:

$$M_1 = \frac{1}{N^2} \sum_{x=1}^N \sum_{y=1}^N (I(x, y) - \langle I \rangle)^2 \quad (5.32)$$

dove $\langle I \rangle$ è la media dei livelli di grigio ed N è la dimensione dell'immagine (quadrata).

L'energia del gradiente dell'immagine

$$M_2 = \sum_{x=1}^N \sum_{y=1}^N |\nabla I(x, y)|^2 \quad (5.33)$$

[§] Il termine fochettatura, benché non proprio corretto, fa parte del gergo dei microscopisti e fotografi, ed indica la ricerca della messa a fuoco ottimale mediante piccoli spostamenti.

dove $|\nabla I(x, y)|$ è il modulo del gradiente (per esempio calcolato con la maschera di Sobel).

L'energia del Laplaciano dell'immagine :

$$M_3 = \sum_{x=1}^N \sum_{y=1}^N (\nabla^2 I(x, y))^2 \quad (5.34)$$

dove $\nabla^2 I(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$ è il Laplaciano dell'immagine (che si calcola con una opportuna maschera, vedi [Gonzales e Wintz, 1987]).

Operatore Tenengrad

$$M_4 = \sum_{x=1}^N \sum_{y=1}^N |\nabla I(x, y)| \quad \text{per} \quad |\nabla I(x, y)| > \Theta \quad (5.35)$$

dove Θ è una soglia che serve a rendere l'indicatore meno sensibile al rumore.

5.3.1.2 Determinazione del massimo.

Per trovare la posizione di fuoco migliore si deve effettuare una ricerca del massimo di una funzione (il criterio di messa a fuoco). Ogni valutazione della funzione significa prendere una immagine e calcolare la messa a fuoco per il punto in esame. Vogliamo dunque limitare il numero di valutazioni. Un metodo adatto in questo caso è la ricerca di Fibonacci, che ottimizza l'incertezza finale fissato a priori il numero di iterazioni [Gill *e al.*, 1981]. Funziona come il noto metodo di bisezione, ma invece di dividere l'intervallo di ricerca a metà, si segue la successione di Fibonacci. A causa del rumore e di altre imperfezioni, l'andamento del criterio di messa a fuoco al variare dei parametri rivela di solito un numero di piccole increspature che possono far cadere la ricerca di Fibonacci in un estremo locale.

5.3.2 Sfocamento

Prendendo un piccolo numero di immagini (al minimo 2) con diversi parametri di fuoco, possiamo determinare la profondità di tutti i punti nella scena. Il metodo si basa sulla relazione diretta fra la profondità, i parametri dell'ottica e lo sfocamento misurabile nell'immagine. Si chiama anche *depth from defocus*.

I problemi nel recupero della distanza dallo sfocamento sono: i) la

misurazione dello sfocamento (di solito stimato confrontando rappresentazioni frequenziali delle immagini) e ii) la calibrazione della relazione tra profondità e sfocamento.

Tradizionalmente lo sfocamento viene modellato come la convoluzione dell'immagine vera I_0 con un nucleo (o *point spread function*) gaussiano (usiamo 1-D per semplicità) con scala spaziale pari a σ :

$$I_1(x) = I_0(x) * g_{\sigma(x)} \quad (5.36)$$

dove $g_{\sigma(x)}$ è la *point spread function* gaussiana, la cui scala spaziale, σ , è funzione della profondità Z e dei parametri della fotocamera (che assumiamo noti) tramite l'equazione (ricavare σ):

$$Z = \frac{Df}{f - D - 2k\sigma D/d} \quad (5.37)$$

dove D la lunghezza focale della lente (da non confondere con la lunghezza focale del modello stenopeico), d è il diametro della lente, f la distanza tra la lente ed il piano immagine, e $k\sigma$ è il raggio del cerchio di confusione (k è una costante).

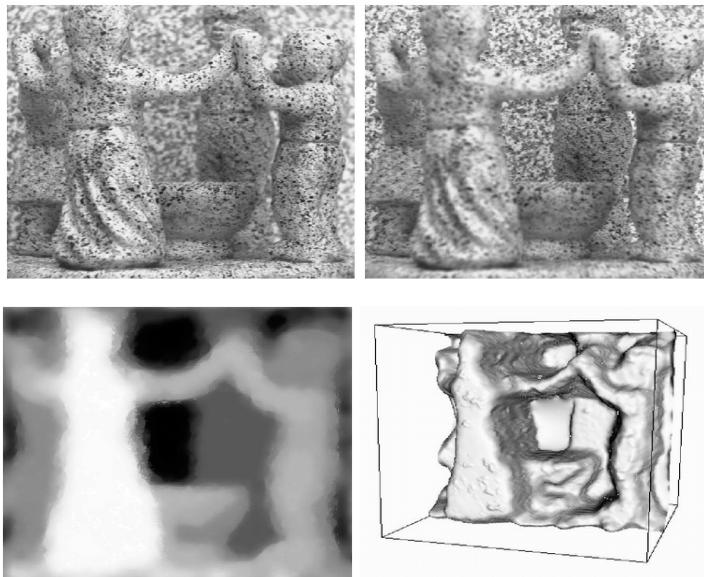


Fig. 5.10. Risultato ottenuti da un recente algoritmo di *depth from defocus* (da http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FAVARO1/dfdtutorial.html). Sopra le due immagini prese con fuoco diverso, sotto l'immagine *range* e la superficie ricostruita.

Nella equazione (5.36) sono incognite sia I_0 che $g_{\sigma(x)}$, dunque essa non basta per ricavare la profondità. Ce ne vogliono almeno due. Intuitivamente, da una immagine sola non posso distinguere una scena da una sua foto. Cambiando il fuoco della fotocamera, ottengo due immagini sfocate e due equazioni

$$\begin{cases} I_1(x) = I_0(x) * g_{\sigma_1(x)} \\ I_2(x) = I_0(x) * g_{\sigma_2(x)} \end{cases} \quad (5.38)$$

Passando alle trasformate di Fourier, dopo alcuni passaggi, si ottiene:

$$\ln \frac{\mathcal{I}_1(s)}{\mathcal{I}_2(s)} = -\frac{1}{2}s^2(\sigma_1^2 - \sigma_2^2) \quad (5.39)$$

dove si è eliminato I_0 , e quindi si può ricavare Z sostituendo la (5.37) per σ_1 e σ_2 (Z è sempre lo stesso).

La figura 5.10 illustra un risultato ottenuto da un recente algoritmo di *depth from defocus*.

Esercizi ed approfondimenti

- 5.1 Avendo a disposizione un operatore che misura lo sfocamento nell'immagine, come quelli visti, è possibile fondere più immagini prese con fuoco diverso per ottenerne una in cui tutti i punti sono a fuoco (aumento artificialmente la profondità di campo).
- 5.2 Una idea simile (ma non si misura lo sfocamento) si applica alla fusione di immagini ottenute con diverse posizioni del diaframma (iris), che regola la quantità di luce che entra (aumento artificialmente la gamma dinamica).

Bibliografia

- Gill P.; Murray W.; Wright M. (1981). *Practical Optimization*. Academic Press.
- Gonzales R. C.; Wintz P. (1987). *Digital image processing*. Addison-Wesley.
- Horn B.; Ikeuchi K. (1981). Numerical Shape from Shading and Occluding Boundaries. *Artificial Intelligence*, **17**, 141–184.
- Subbarao M.; Chio T.; Nikzad A. (1993). Focusing techniques. *Optical Engineering*, pp. 2824–2836.
- Trucco E.; Verri A. (1998). *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall.

- Tsai P. S.; Shah M. (1994). Shape from Shading Using Linear Approximation. *Image and Vision Computing*, **12**(8).
- Zheng Q.; Chellappa R. (1991). Estimation of illuminant direction, albedo, and shape from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**(7), 680–702.

6

Stereopsi

La stereopsi è la capacità percettiva che consente di unire le immagini provenienti dai due occhi, che a causa del loro diverso posizionamento strutturale, presentano uno spostamento laterale. Questa disparità viene sfruttata dal cervello per trarre informazioni sulla profondità e sulla posizione spaziale dell'oggetto mirato. Di conseguenza la stereopsi permette di generare la visione tridimensionale.†

La stereopsi (da stereo=solido e opsis=sguardo) è il processo della percezione visiva che crea la sensazione della profondità dalle due proiezioni lievemente differerenti della scena sui due occhi. [...] Le differenze sono causate dalla diversa posizione degli occhi.‡

6.1 Introduzione

La stereopsi (computazionale) è il processo che consente di ottenere informazioni sulla struttura tridimensionale da una coppia di immagini, provenienti da due fotocamere che inquadrano una scena da differenti posizioni. Possiamo individuare due sottoproblemi: calcolo delle corrispondenze e triangolazione.

Il primo consiste nell'accoppiamento tra punti nelle due immagini che sono proiezione dello stesso punto della scena (figura 6.1). Chiameremo tali punti *coniugati*. Il calcolo dell'accoppiamento è possibile sfruttando il fatto che le due immagini differiscono solo lievemente, sicchè un particolare della scena appare simile nelle due immagini (§ 7.1). Basandosi solo su questo vincolo, però, sono possibili molti *falsi* accoppiamenti. Vedremo che sarà necessario introdurre altri vincoli che rendano il calcolo delle corrispondenze trattabile. Il più importante di questi è il vincolo

† da http://it.wikipedia.org/wiki/Visione_binoculare#Stereopsi

‡ da <http://en.wikipedia.org/wiki/Stereopsis> (tradotto)

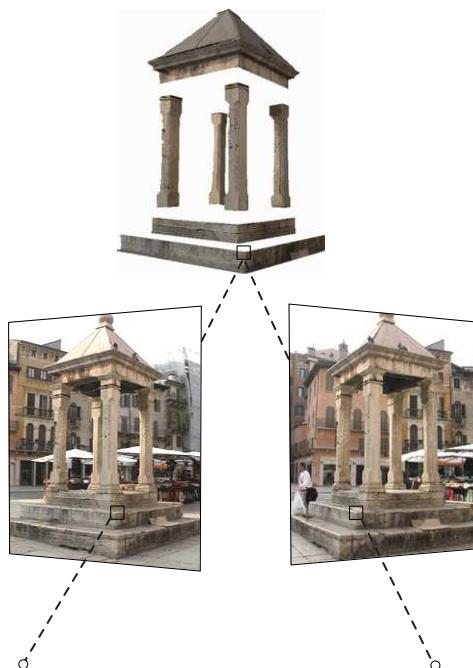


Fig. 6.1. Coppia stereo raffigurante la tribuna di Piazza Erbe (VR). Due punti coniugati nelle immagini sono la proiezione del medesimo punto dello spazio.

epipolare (§ 6.3), il quale afferma che il corrispondente di un punto in una immagine può trovarsi solo su una retta (retta epipolare) nell'altra immagine. Grazie a questo la ricerca delle corrispondenze diventa unidimensionale, invece che bidimensionale.

Noti gli accoppiamenti tra i punti delle due immagini e nota la posizione reciproca delle fotocamere ed i parametri intrinseci del sensore è possibile ricostruire la posizione nella scena dei punti che sono proiettati sulle due immagini (§ 6.2). Questo processo di triangolazione necessita della calibrazione dell'apparato stereo, (§ 4.2), ovvero del calcolo dei parametri intrinseci e della posizione reciproca (parametri *estrinseci*) delle fotocamere.

6.2 Triangolazione 3D

Prima di affrontare la triangolazione nel caso più generale, vediamo che cosa si può imparare da un caso semplificato come il seguente. Si consi-

derino due fotocamere parallele ed allineate (piani retina coincidenti): è facile verificare che si ha una disparità puramente orizzontale e quindi è giustificata la costruzione bidimensionale di figura 6.2.

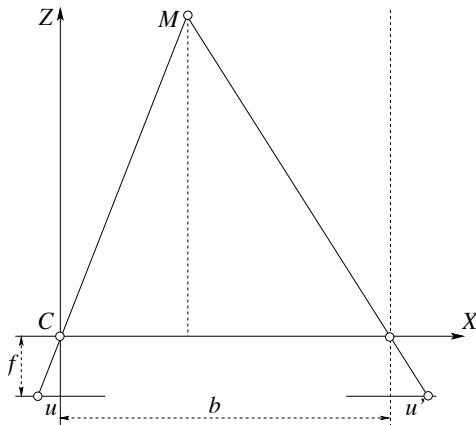


Fig. 6.2. Triangolazione stereoscopica, caso semplificato.

Fissato il riferimento mondo solidale con la fotocamera di sinistra, possiamo scrivere le seguenti equazioni di proiezione prospettica:

$$\begin{cases} \frac{f}{z} = \frac{-u}{x} \\ \frac{f}{z} = \frac{-u'}{x-b} \end{cases} \quad (6.1)$$

da cui, elaborando, si ottiene

$$z = \frac{bf}{u' - u}. \quad (6.2)$$

La (6.2) ci suggerisce che è possibile ricavare la terza coordinata z , noti (i) la geometria del sistema stereo (b ed f in questo semplice caso) e (ii) la disparità ($u - u'$). Si vede anche che la lunghezza della *linea di base* b si comporta come un fattore di scala: la disparità associata ad un punto della scena fissato dipende in modo diretto da b . Si noti inoltre che se il parametro b è incognito è possibile la ricostruzione della struttura tridimensionale *a meno di un fattore di scala*.

6.2.1 Metodo linear-eigen

Vediamo ora come sia possibile la triangolazione nel caso generale (figura 6.3). Dati (i) le coordinate (in pixel) di due punti coniugati e (ii)

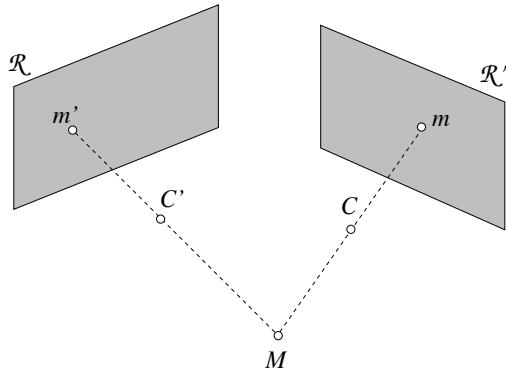


Fig. 6.3. Triangolazione, caso generale.

le due MPP relative alle due fotocamere, si può facilmente ricostruire la posizione in coordinate assolute del punto di cui entrambi sono la proiezione[†].

Consideriamo $\mathbf{m} = [u, v, 1]^\top$, la proiezione del punto M sulla fotocamera che ha MPP P . Dalla equazione di proiezione prospettiva (4.24) si ricava

$$\begin{cases} (\mathbf{p}_1 - u\mathbf{p}_3)^\top \mathbf{M} = 0 \\ (\mathbf{p}_2 - v\mathbf{p}_3)^\top \mathbf{M} = 0 \end{cases} \quad (6.3)$$

e quindi, in forma matriciale

$$\begin{bmatrix} (\mathbf{p}_1 - u\mathbf{p}_3)^\top \\ (\mathbf{p}_2 - v\mathbf{p}_3)^\top \end{bmatrix} \mathbf{M} = \mathbf{0}_{2 \times 1} \quad (6.4)$$

Un punto, dunque, fornisce due equazioni.

Consideriamo ora anche $\mathbf{m}' = [u', v', 1]^\top$, il punto coniugato di m nella seconda immagine, e sia P' la seconda MPP. Essendo m ed m' proiezione del medesimo punto M , le equazioni fornite da entrambi si possono impilare, ottenendo un sistema lineare omogeneo di quattro equazioni in quattro incognite (l'ultima componente di M figura come incognita):

$$\begin{bmatrix} (\mathbf{p}_1 - u\mathbf{p}_3)^\top \\ (\mathbf{p}_2 - v\mathbf{p}_3)^\top \\ (\mathbf{p}'_1 - u'\mathbf{p}'_3)^\top \\ (\mathbf{p}'_2 - v'\mathbf{p}'_3)^\top \end{bmatrix} \mathbf{M} = \mathbf{0}_{4 \times 1} \quad (6.5)$$

[†] Il punto (i) implica la soluzione del problema delle corrispondenze, che sarà affrontato nei paragrafi 6.3 e 7.1. Il punto (ii) assume la calibrazione, studiata nel paragrafo 4.2.

La soluzione è il nucleo della matrice 4×4 dei coefficienti, che dunque deve possedere rango tre, altrimenti si avrebbe la sola soluzione banale $\mathbf{M} = \mathbf{0}$. In presenza di rumore, tuttavia, la condizione sul rango non viene soddisfatta esattamente e dunque si cerca una soluzione ai minimi quadrati con SVD (come per la calibrazione della fotocamera). Hartley e Sturm [1997] chiamano questo metodo “*linear-eigen*”.

Quanto esposto si generalizza al caso di $N > 2$ fotocamere: ogni fotocamera aggiunge due equazioni e si ottiene un sistema omogeneo di $2N$ equazioni in quattro incognite.

Si veda la funzione `intersect` ed in particolare `intersect_1e`.

Come nel caso della calibrazione, una causa di inaccuratezza in questo metodo lineare è che il residuo minimizzato non ha significato geometrico. Una funzione di costo adatta, come l’errore nel piano immagine, potrebbe essere utilizzata per guadagnare una maggior accuratezza:

$$\varepsilon(\mathbf{M}) = \left\| \begin{bmatrix} u \\ v \end{bmatrix} - \begin{bmatrix} \mathbf{p}_1^T \mathbf{M} \\ \mathbf{p}_3^T \mathbf{M} \end{bmatrix} \right\|^2 + \left\| \begin{bmatrix} u' \\ v' \end{bmatrix} - \begin{bmatrix} \mathbf{p}_1'^T \mathbf{M} \\ \mathbf{p}_3'^T \mathbf{M} \end{bmatrix} \right\|^2 \quad (6.6)$$

Anche la funzione costo geometrica si generalizza al caso di più di due fotocamere: l’espressione diventa una sommatoria in cui ciascun termine è l’errore nel piano della i -esima fotocamera.

6.3 Geometria epipolare

Vediamo ora quale relazione lega due immagini di una stessa scena ottenute da due fotocamere diverse, o dalla medesima fotocamera in movimento. In particolare, ci chiediamo, dato un punto m nella prima immagine, quali vincoli esistono sulla posizione del suo coniugato m' nella seconda immagine.

Alcune semplici considerazioni geometriche indicano che il punto coniugato di m deve giacere su di una linea retta nella seconda immagine, chiamata *retta epipolare* di m .

La geometria epipolare è importante anche (e soprattutto) perché descrive la relazione tra due viste di una stessa scena, dunque è fondamentale in qualunque tecnica di visione computazionale basata su più di una immagine. Maggiori dettagli si trovano in [Faugeras, 1993].

Si consideri il caso illustrato in figura 6.4. Dato un punto m nella prima immagine, il suo coniugato m' nella seconda immagine è vincolato a giacere sull’intersezione del piano immagine con il piano determinato da m , C e C' , detto piano epipolare. Questo poichè il punto m' può

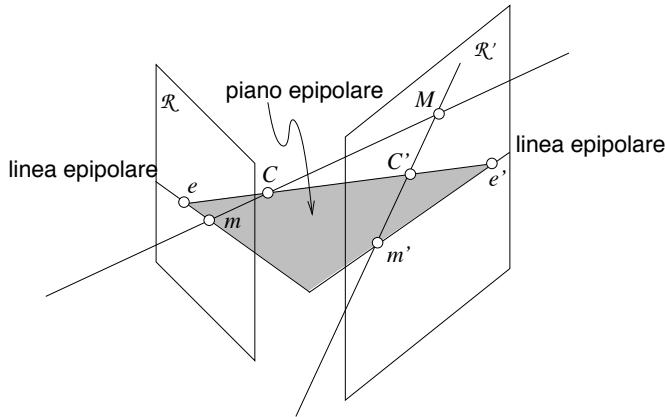


Fig. 6.4. Geometria epipolare.

essere la proiezione di un qualsiasi punto nello spazio giacente sul raggio ottico di m . Inoltre si osserva che tutte le linee epipolari di una immagine passano per uno stesso punto, chiamato *epipolo*, e che i piani epipolari costituiscono un fascio di piani che hanno in comune la retta passante per i centri ottici C e C' . La retta CC' prende il nome di linea di base (o *baseline*). A volte si indica con lo stesso termine il segmento $\overline{CC'}$: la differenza emerge dal contesto. Se si parla di lunghezza della *baseline* è chiaro che ci si riferisce al segmento.

Fissato un sistema di riferimento assoluto, date due MPP $P = [Q|\mathbf{q}]$ e $P' = [Q'|\mathbf{q}']$, sappiamo che:

$$\begin{cases} \mathbf{m} \simeq P\mathbf{M} \\ \mathbf{m}' \simeq P'\mathbf{M} \end{cases} \quad (6.7)$$

La linea epipolare corrispondente ad m è la proiezione secondo P' del raggio ottico di m , che – ricordiamolo – ha equazione

$$\mathbf{M} = \mathbf{C} + \lambda \begin{bmatrix} Q^{-1}\mathbf{m} \\ 0 \end{bmatrix} \quad (6.8)$$

Siccome

$$P'\mathbf{C} = P' \begin{bmatrix} -Q^{-1}\mathbf{q} \\ 1 \end{bmatrix} = \mathbf{q}' - Q'Q^{-1}\mathbf{q} \triangleq \mathbf{e}' \quad (6.9)$$

e

$$P' \begin{bmatrix} Q^{-1}\mathbf{m} \\ 0 \end{bmatrix} = Q'Q^{-1}\mathbf{m} \quad (6.10)$$

la retta epipolare di m ha equazione:

$$\mathbf{m}' \simeq \lambda Q'Q^{-1}\mathbf{m} + \mathbf{e}'. \quad (6.11)$$

Questa è l'equazione (in coordinate omogenee) della retta passante per i punti \mathbf{e}' (l'epipolo) e $Q'Q^{-1}\mathbf{m}$.

Per vedere che esiste una **relazione bilineare** tra i punti coniugati dobbiamo elaborare ulteriormente l'equazione. Moltiplichiamo a destra e sinistra per $[\mathbf{e}']_\times$, la matrice antisimmetrica che agisce come il prodotto esterno con \mathbf{e}' , ottenendo (ricordiamo che $[\mathbf{x}]_\times \mathbf{x} = \mathbf{0}$):

$$[\mathbf{e}']_\times \mathbf{m}' \simeq \lambda [\mathbf{e}']_\times Q'Q^{-1}\mathbf{m} \quad (6.12)$$

La parte sinistra è un vettore ortogonale a \mathbf{m}' , quindi se moltiplichiamo a destra e sinistra per \mathbf{m}'^\top , otteniamo:

$$0 = \mathbf{m}'^\top [\mathbf{e}']_\times Q'Q^{-1}\mathbf{m}. \quad (6.13)$$

Questa equazione, che prende anche il nome di *equazione di Longuet-Higgins*, rappresenta una forma bilineare in \mathbf{m} ed \mathbf{m}' .

Dal punto di vista geometrico essa è coerente con il fatto che, come è spiegato nell'appendice 2, la retta passante per due punti è rappresentata dal prodotto esterno dei due punti, quindi la linea epipolare di \mathbf{m} è rappresentata (in coordinate omogenee) dal vettore:

$$\mathbf{e}' \times (Q'Q^{-1}\mathbf{m}). \quad (6.14)$$



Fig. 6.5. Coppia stereo. A destra sono disegnate le rette epipolari corrispondenti ai punti marcati con un quadrato nell'immagine sinistra.

La matrice

$$F = [\mathbf{e}']_{\times} Q' Q^{-1} \quad (6.15)$$

che contiene i coefficienti della forma bilineare prende il nome di **matrice fondamentale**. L'equazione di Longuet-Higgins quindi si riscrive:

$$\mathbf{m}'^{\top} F \mathbf{m} = 0. \quad (6.16)$$

La matrice fondamentale contiene tutta l'informazione relativa alla geometria epipolare. Conoscendo F possiamo tracciare la retta epipolare di un punto arbitrario. Infatti, preso m , la retta definita† da $F\mathbf{m}$ è la sua retta epipolare.

Per ora la matrice fondamentale ci serve solo per tracciare le rette epipolari, ed è calcolata come funzione delle due MPP. Vedremo nel seguito (§ 12.2.1) come possa venire calcolata direttamente dalle immagini, anche quando la calibrazione (cioè le MPP) non sia disponibile, e ne studieremo meglio le proprietà.

6.4 Rettificazione epipolare

Quando C è nel piano focale della fotocamera coniugata, l'epipolo \mathbf{e}' giace all'infinito e le linee epipolari formano un fascio di linee parallele.

Un caso molto speciale si ha quando entrambi gli epipoli sono all'infinito, che accade quando la linea di base $\overline{CC'}$ è contenuta in entrambi i piani focali, ovvero i piani retina sono paralleli alla linea di base. In questo caso le linee epipolari formano un fascio di linee parallele in entrambe le immagini. Il caso in cui le linee epipolari sono parallele ed orizzontali costituisce una situazione particolarmente favorevole per il calcolo delle corrispondenze: punti corrispondenti giacciono sulla stessa riga (o scan-line) della immagine (vista come una matrice di pixel). Qualunque coppia di immagini può essere trasformata in questo modo tramite il processo di *rettificazione epipolare*. L'idea della rettificazione è definire due nuove MPP che conservano i centri ottici ma con i piani immagine paralleli alla linea di base. Le immagini rettificate possono essere pensate come acquisite da una nuova coppia di fotocamere, ottenuta ruotando le fotocamere originali.

Il metodo che illustriamo, per fotocamere calibrate, è tratto da [Fusiello *et al.*, 2000]. Il caso di fotocamere non calibrate è stato affrontato in [Hartley, 1999, Loop e Zhang, 1999, Isgrò e Trucco, 1999].

† La retta definita da un vettore di tre elementi (a, b, c) ha equazione $au + bv + c = 0$.

6.4.1 Rettificazione delle MPP

Siano P_o e P'_o le due MPP delle due fotocamere. L'idea che sta dietro la rettificazione è quella di definire due nuove MPP P_n e P'_n ottenute ruotando le matrici originali attorno ai loro centri ottici finché i piani focali non diventano copiani (e in tal modo conterranno entrambi la linea di base). Questo ci assicura che gli epipoli sono punti all'infinito, perciò le linee epipolari sono *parallele*. Per avere linee epipolari *orizzontali*, la linea di base deve essere parallela al nuovo asse X di entrambe le fotocamere. Inoltre, si vuole garantire una proprietà di rettificazione più forte, richiedendo che i punti coniugati abbiano *le stesse coordinate verticali* (perché è più comodo). Questo si ottiene forzando le due nuove fotocamere ad avere gli stessi parametri intrinseci (in realtà la coordinata orizzontale del centro dell'immagine è libera di variare, cosa che useremo per centrare le immagini rettificate). Si può osservare che, essendo la lunghezza focale la stessa, i piani retina sono anch'essi copiani, come mostrato in figura 6.6.

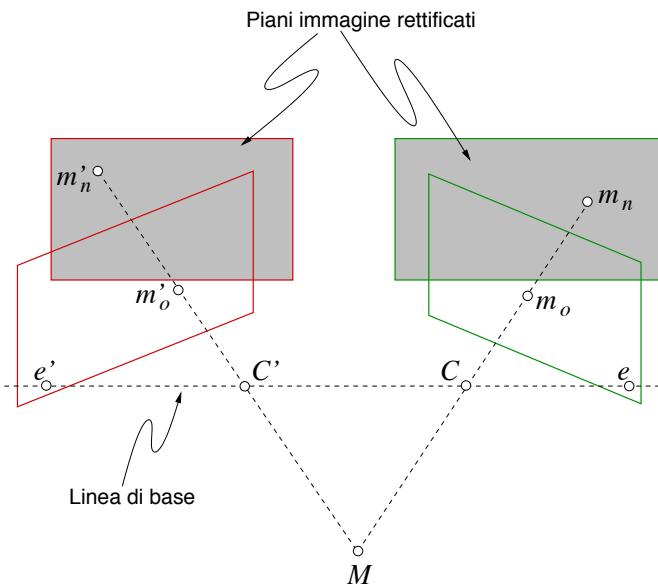


Fig. 6.6. Fotocamere rettificate. I piani retina sono copiani e paralleli alla linea di base.

Riassumendo: le posizioni (ovvero i centri ottici) delle nuove MPP sono uguali a quelle delle vecchie fotocamere, mentre la nuova orientazione (uguale per entrambe le fotocamere) differisce da quella originale per una

rotazione (una diversa per ciascuna fotocamera); i parametri intrinseci sono uguali per entrambe le fotocamere. Quindi, le due matrici MPP risultanti differiranno tra loro solo nei centri ottici, e queste possono essere pensate come una singola fotocamera traslata lungo l'asse X del suo sistema di riferimento.

Scriviamo le nuove MPP nei termini della loro fattorizzazione:

$$P_n = K[R | -R \tilde{\mathbf{C}}], \quad P'_n = K[R | -R \tilde{\mathbf{C}}']. \quad (6.17)$$

(la si ottiene come la (E4.4) nell'esercizio 5 del capitolo 4.)

La matrice dei parametri intrinseci K è la stessa per entrambe le MPP, e può essere fissata arbitrariamente. I centri ottici $\tilde{\mathbf{C}}$ e $\tilde{\mathbf{C}}'$ sono dati dai vecchi centri ottici calcolati per mezzo della (4.29). La matrice \mathbf{R} , che determina l'orientazione della fotocamera, è la stessa per entrambe le MPP. Se scriviamo la rotazione nel seguente modo

$$R = \begin{bmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \\ \mathbf{r}_3^\top \end{bmatrix} \quad (6.18)$$

abbiamo che $\mathbf{r}_1^\top, \mathbf{r}_2^\top, \mathbf{r}_3^\top$ sono, rispettivamente, gli assi X, Y e Z del sistema di riferimento della fotocamera, espressi in coordinate mondo.

In accordo con quanto detto precedentemente, per determinare R , poniamo:

- (i) Il nuovo asse X parallelo alla linea di base:
 $\mathbf{r}_1 = (\tilde{\mathbf{C}}' - \tilde{\mathbf{C}})/\|\tilde{\mathbf{C}}' - \tilde{\mathbf{C}}\|;$
- (ii) Il nuovo asse Y ortogonale a X ed a un versore arbitrario \mathbf{k} :
 $\mathbf{r}_2 = \mathbf{k} \times \mathbf{r}_1;$
- (iii) Il nuovo asse Z ortogonale a XY (per forza):
 $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2.$

Nel punto 2, \mathbf{k} è un vettore unitario arbitrario che fissa la posizione del nuovo asse Y nel piano ortogonale a X (la direzione verticale). Lo prendiamo eguale al vettore unitario Z della vecchia fotocamera, costringendo in tal modo il nuovo asse Y ad essere ortogonale sia al nuovo asse X e sia al vecchio asse Z .

Questo algoritmo fallisce quando l'asse ottico è parallelo alla linea di base, cioè quando c'è un puro movimento in avanti.

Si veda la funzione `rectify`.

6.4.2 La trasformazione di rettificazione

Per rettificare (ad esempio) il piano immagine della fotocamera P_o , dobbiamo calcolare la trasformazione che porta il piano immagine di $P_o = [Q_o|\mathbf{q}_o]$ nel piano immagine di $P_n = [Q_n|\mathbf{q}_n]$. Vedremo ora che la trasformazione cercata è la *collineazione* (trasformazione lineare del piano proiettivo) definita dalla matrice 3×3 : $T = Q_n Q_o^{-1}$. È utile pensare ad un'immagine come l'intersezione del piano retina con il cono dei raggi che hanno origine nel centro ottico e passano per i punti visibili della scena (figura 6.7). Ruotando il piano immagine e tenendo fermo il cono di raggi, otterremo una nuova immagine degli stessi punti visibili.

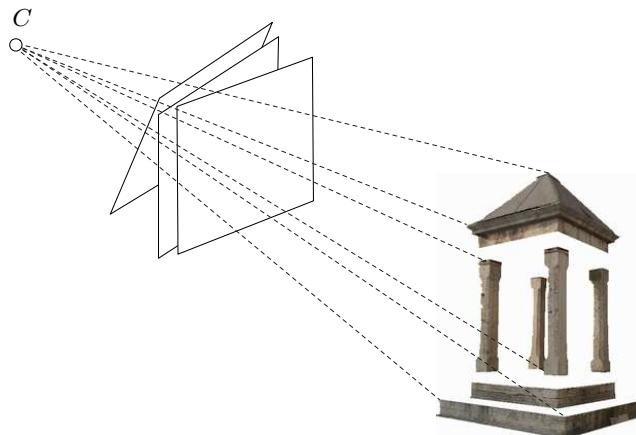


Fig. 6.7. Un'immagine è l'intersezione del piano retina con il cono dei raggi che hanno origine nel centro ottico e passano per i punti visibili della scena. Due immagini con lo stesso centro ottico sono legate da una trasformazione del piano proiettivo.

Per ogni punto 3D \mathbf{M} possiamo scrivere

$$\begin{cases} \mathbf{m}_o \simeq P_o \mathbf{M} \\ \mathbf{m}_n \simeq P_n \mathbf{M}. \end{cases} \quad (6.19)$$

In accordo con (4.31), le equazioni dei raggi ottici sono le seguenti (dato che la rettificazione non sposta il centro ottico):

$$\begin{cases} \tilde{\mathbf{M}} = \tilde{\mathbf{C}} + \lambda_o Q_o^{-1} \mathbf{m}_o & \lambda_o \in \mathbb{R}; \\ \tilde{\mathbf{M}} = \tilde{\mathbf{C}} + \lambda_n Q_n^{-1} \mathbf{m}_n & \lambda_n \in \mathbb{R} \end{cases} \quad (6.20)$$

quindi

$$\mathbf{m}_n \simeq Q_n Q_o^{-1} \mathbf{m}_o. \quad (6.21)$$

La trasformazione T è applicata, poi, all'immagine originale di per produrre l'immagine rettificata, come mostrato in figura 6.8. Per quanto riguarda il modo corretto di applicare una trasformazione ad una immagine, si rimanda alla lettura del capitolo 14.

Le coordinate pixel delle immagini rettificate possono, in principio, variare in regioni del piano anche molto diverse dalla finestra occupata dalla immagine originale (da $(0,0)$ a $(384,512)$ per l'esempio in figura 6.8), specialmente se la trasformazione contiene una forte componente di traslazione. La possibilità di cambiare la coordinata orizzontale del centro dell'immagine senza che la rettificazione ne risenta viene usata per riportare l'immagine rettificata il più possibile all'interno della finestra originale.

Infine, dopo aver calcolato le corrispondenze (o disparità) sulle immagini rettificate, la ricostruzione della struttura 3D attraverso la triangolazione può essere ottenuta direttamente, usando P_n, P'_n .



Fig. 6.8. Coppia stereo rettificata (la coppia originale è mostrata in figura 6.5).

Esercizi ed approfondimenti

- 6.1 Come risente il calcolo della profondità z di un errore Δd sulla misura della disparità?

Soluzione: riprendiamo la formula semplificata della triangolazione stereo (equazione (6.2)):

$$z = \frac{bf}{d} \quad (\text{E6.1})$$

e deriviamo z rispetto a d . Ricaviamo quindi:

$$\Delta z \cong -\frac{bf}{d^2} \Delta d \quad (\text{E6.2})$$

dove Δd è l'errore nella stima della disparità (in pixel). Esplicitiamo d (quantità non nota a priori):

$$d = \frac{bf}{z} \quad (\text{E6.3})$$

e sostituiamolo ottenendo

$$\Delta z = -\frac{z^2}{bf} \Delta d \quad (\text{E6.4})$$

In definitiva (trascurando il segno) ad un errore nella stima della disparità in pixel Δd si ha un conseguente errore nella stima della distanza Δz , che cresce proporzionalmente al quadrato della distanza di lavoro ed inversamente alla lunghezza della linea di base e alla lunghezza focale.

- 6.2 Se le matrici sono normalizzate possiamo riscrivere l'equazione della retta epipolare con le profondità del punto (ζ e ζ') esplicitate:

$$\zeta' \mathbf{m}' = \zeta Q' Q^{-1} \mathbf{m} + \mathbf{e}' \quad (\text{E6.5})$$

- 6.3 Metodo alternativo per la triangolazione. Si può ottenere la profondità di un punto ζ o ζ' risolvendo la (E6.5) (si usa il risultato della proposizione 1.51). Dati ζ o ζ' posso ottenere facilmente le coordinate del punto M (come?).

- 6.4 Raffinamento iterativo del metodo *linear-eigen*. Con riferimento – per esempio – alla prima equazione del sistema lineare 6.5, il residuo che viene minimizzato è $\varepsilon = \mathbf{p}_1^\top \mathbf{M} - u \mathbf{p}_3^\top \mathbf{M}$, mentre invece si vorrebbe minimizzare il residuo geometrico, ovvero la differenza tra la coordinata u misurata e la proiezione di \mathbf{M} : $\varepsilon' = \frac{\mathbf{p}_1^\top \mathbf{M}}{\mathbf{p}_3^\top \mathbf{M}} - u$. Si osservi che $\varepsilon' = \frac{\varepsilon}{\mathbf{p}_3^\top \mathbf{M}}$, ovvero se l'equazione venisse pesata con $\frac{1}{\mathbf{p}_3^\top \mathbf{M}}$ staremmo minimizzando proprio il residuo geometrico. Il problema è che il peso giusto dipende da \mathbf{M} , che è proprio ciò che vogliamo calcolare. La soluzione consiste nell'iterare il metodo *linear-eigen* ponderando le equazioni con

- pesi calcolati sulla base delle coordinate di \mathbf{M} ottenute al passo precedente.
- 6.5 Una coppia rettificata possiede la seguente matrice fondamentale:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Perché?

- 6.6 Alla luce del risultato dell'esercizio precedente, mostrare che il procedimento di rettificazione epipolare illustrato nel §. 6.4 è corretto.

Bibliografia

- Faugeras O. (1993). *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, MA.
- Fusiello A.; Trucco E.; Verri A. (2000). A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, **12**(1), 16–22.
- Hartley R. (1999). Theory and practice of projective rectification. *International Journal of Computer Vision*, **35**(2), 1–16.
- Hartley R. I.; Sturm P. (1997). Triangulation. *Computer Vision and Image Understanding*, **68**(2), 146–157.
- Isgrò F.; Trucco E. (1999). Projective rectification without epipolar geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. I:94–99, Fort Collins, CO.
- Loop C.; Zhang Z. (1999). Computing rectifying homographies for stereo vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. I:125–131, Fort Collins, CO.

7

Calcolo delle corrispondenze

7.1 Introduzione

Affrontiamo ora il problema centrale della visione stereo: il calcolo delle corrispondenze o della *disparità*.

Una coppia coniugata è costituita da due punti in due immagini diverse che sono proiezione dello stesso punto della scena.

La disparità è la differenza (vettore) tra due punti coniugati, immaginando di sovrapporre le due immagini.

Il calcolo delle corrispondenze equivale al calcolo della disparità per i punti (tutti o alcuni) della immagine di riferimento (per esempio quella sinistra). Si ottiene in tal modo una *mappa* o *campo di disparità* (denso o sparso), come in figura 7.1.



Fig. 7.1. Coppia di immagini stereo e mappa di disparità (immagini prese da <http://vision.middlebury.edu/stereo/>).

7.1.1 Problemi

Il calcolo delle corrispondenze si basa sull'assunzione che le due immagini non siano troppo diverse, ovvero che un particolare della scena appaia

simile nelle due immagini. Basandosi sulla similarità, un punto di una immagine può essere messo in corrispondenza con molti punti dell'altra immagine: è il problema delle *false corrispondenze*, che rende difficile l'identificazione delle coppie coniugate. Oltre ai falsi accoppiamenti, vi sono altri problemi che affliggono il calcolo delle corrispondenze, dovuti al fatto che la scena viene inquadrata da due punti di vista differenti.

occlusioni: a causa di discontinuità nelle superfici, vi sono parti della scena che compaiono in una sola delle immagini, ovvero esistono punti in una immagine che non hanno il corrispondente nell'altra immagine. Chiaramente non è possibile definire alcuna disparità per questi punti;

distorsione radiometrica: a causa di superfici non perfettamente lambertiane, l'intensità osservata dalle due fotocamere (la radianza) è diversa per lo stesso punto della scena;

distorsione prospettica: a causa della proiezione prospettica, un oggetto proiettato assume forme diverse nelle due immagini.

Tutti questi problemi si aggravano tanto più quanto più le fotocamere sono distanti. D'altra parte, per avere una disparità significativa, le fotocamere *devono* essere ben separate tra loro.

7.1.2 Vincoli

Alcuni vincoli (introdotti già nei primi lavori di Marr e Poggio [1976]) però possono essere sfruttati nel calcolo delle corrispondenze:

somiglianza: un particolare della scena appare simile nelle due immagini (è implicito);

geometria epipolare: il punto coniugato giace su una retta (epipolare) determinata dai parametri intrinseci e dalla reciproca posizione delle fotocamere (già discusso);

lisciezza: lontano dai bordi, la profondità dei punti di una superficie liscia varia lentamente. Questo pone un limite al gradiente della disparità;

unicità: un punto dell'immagine di sinistra può essere messo in corrispondenza con un solo punto nell'immagine di destra, e viceversa (fallisce se ci sono oggetti trasparenti o in presenza di occlusioni);

ordinamento monotono: se il punto \mathbf{m}_1 in una immagine corrisponde a \mathbf{m}'_1 nell'altra, il corrispondente di un punto \mathbf{m}_2 che giace alla destra (sinistra) di \mathbf{m}_1 deve trovarsi alla destra (sinistra) di \mathbf{m}'_1 . Fallisce per punti che si trovano nella zona proibita di un

dato punto (si veda figura 7.2). Normalmente, per una superficie opaca e coesa, i punti nella zona proibita non sono visibili, dunque il vincolo vale.

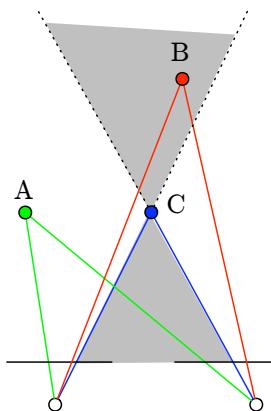


Fig. 7.2. Il punto B viola il vincolo di ordinamento rispetto a C, mentre il punto A lo rispetta. Il cono grigio è la zona proibita di C.

Supporremo, senza perdita di generalità, che le linee epipolari siano parallele e orizzontali nelle due immagini sicché i punti coniugati verranno ricercati lungo le linee orizzontali delle immagini. La mappa di disparità si riduce ad un campo scalare: in ogni pixel viene registrato lo spiazzamento orizzontale che separa il corrispondente punto dell'immagine di riferimento dal suo punto coniugato.

7.1.3 Metodi locali e globali

Tutti i metodi cercano di accoppiare pixel di una immagine con pixel dell'altra immagine sfruttando alcuni dei vincoli elencati sopra. I *metodi locali* impongono il vincolo ad un piccolo numero di pixel che circondano il pixel che si vuole accoppiare. I *metodi globali* impongono vincoli a livello della retta orizzontale che passa per il pixel (*scan-line*) o di tutta l'immagine.

7.2 Metodi di accoppiamento locali

7.2.1 Accoppiamento di finestre

Questi algoritmi – chiamati anche *block matching* – considerano una piccola area (o finestra) rettangolare in una immagine e cercano quella più

somigliante nell'altra immagine, mediante una misura di somiglianza tra i livelli di grigio (o una loro funzione). Questo viene fatto per ogni punto, ottenendo così una mappa densa. In regioni uniformi non è possibile ottenere una misura di disparità. Inoltre, come discusso precedentemente, il livello di grigio è dipendente dal punto di vista.

Più in dettaglio, l'accoppiamento stereo consiste nel calcolare per ogni pixel (u, v) della immagine I_1 il suo corrispondente $(u + d, v)$ in I_2 . Si consideri una finestra centrata in (u, v) di dimensioni $(2n + 1)(2m + 1)$. Questa viene confrontata con una finestra delle stesse dimensioni in I_2 che si muove lungo la linea epipolare corrispondente ad (u, v) ; essendo le immagini rettificate, si considerano le posizioni $(u + d, v)$, $d \in [d_{\min}, d_{\max}]$. La disparità calcolata è lo spostamento che corrisponde alla massima *somiglianza* tra i livelli di grigio delle due finestre.

7.2.1.1 Metriche di accoppiamento

Vi sono vari criteri di somiglianza tra finestre, che si possono classificare in tre categorie:

- basati su correlazione (NCC, ZNCC);
- basati sulle differenze di intensità (SSD, SAD);
- basati su operatori di rango (trasformata *census*)

Uno dei più comuni è la cosiddetta SSD (*Sum of Squared Differences*):

$$SSD(u, v, d) = \sum_{(k,l)} (I_1(u + k, v + l) - I_2(u + k + d, v + l))^2 \quad (7.1)$$

dove $k \in [-n, n]$, $l \in [-m, m]$ e $I(u, v)$ indica il livello di grigio del pixel (u, v) . Più piccolo è il valore della (7.1), più le porzioni delle immagini considerate sono simili. La disparità calcolata è l'argomento del minimo della funzione errore:

$$d_o(u, v) = \arg \min_d SSD(u, v, d) \quad (7.2)$$

In questo modo viene calcolata la disparità con precisione di un pixel. È possibile, interpolando la funzione errore in prossimità del minimo (per esempio con una parabola), ottenere una precisione maggiore (sub-pixel).

L'algoritmo 7.1 in stile C calcola la disparità con SSD (attenzione: usa coordinate riga/colonna invece che (u, v)):

Algoritmo 7.1 BLOCK MATCHING

Input: Immagini I1, I2; dimensioni rows, cols; intervallo dmin, dmax; semidimensioni finestra m, n;

Output: Mappa di disparità D

```

for(i=n; i<rows-n; i++)
    for(j=m-MIN(0,dmin); j< cols-m-MAX(0,dmax); j++)
    {
        min=HUGEVAL;
        d_o = 0;
        for(d=dmin; d<=dmax; d++)
        {
            sum=0.0;
            for(k=-n; k<=n; k++)
                for(l=-m; l<=m; l++)
                    sum+=SQR(I1[i+k] [j+l]-I2[i+k] [j+l+d]);
            if (sum<min)
            {
                d_o=d;
                min=sum;
            }
        }
        D[i] [j]=d_o;
    }
}

```

Per valutare il funzionamento dell'algoritmo di accoppiamento con SSD, sfruttiamo i cosiddetti stereogrammi a punti casuali (RDS), nei quali l'unico indizio visivo presente è la disparità.

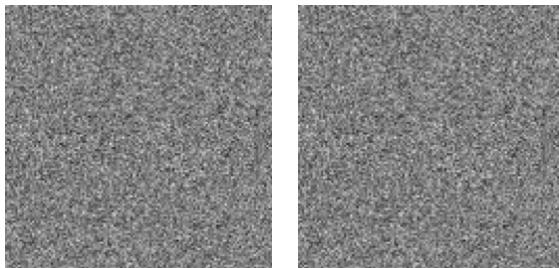


Fig. 7.3. Stereogramma a punti casuali. Nell'immagine di destra un quadrato è traslato a destra di 10 pixel. Lo sfondo è traslato a destra di tre pixel.

Un RDS (Fig 7.3) è una coppia di immagini ottenuta nel modo seguente: si genera una immagine a punti casuali (sinistra), si definisce un quadrato (o una regione di una forma qualsiasi) e si costruisce un'altra immagine (destra) traslando il quadrato di un certo numero di pixel ver-

so destra. La parte che rimane “scoperta” è riempita con punti casuali. La coppia così costruita presenta una disparità tra le due immagini: infatti, osservando l’immagine destra con l’occhio destro e la sinistra con l’occhio sinistro, è possibile percepire il quadrato su un piano più vicino rispetto allo sfondo.



Fig. 7.4. Mappe di disparità ottenute con la correlazione SSD su uno stereogramma a punti casuali con rumore gaussiano $\sigma^2 = 10.0$ aggiunto. Il livello di grigio – normalizzato – rappresenta la disparità. La dimensione della finestra di correlazione vale, da sinistra a destra, 3x3, 7x7 e 11x11

Simile al SSD è il SAD (*Sum of Absolute Differences*), dove il quadrato viene sostituito dal valore assoluto. In questo modo la metrica risulta meno sensibile a rumore di tipo impulsivo: due finestre che sono uguali in tutti tranne un pixel risultano più simili secondo SAD che secondo SSD, poiché il quadrato pesa molto di più le differenze del valore assoluto.

$$SAD(u, v, d) = \sum_{(k,l)} |I_1(u+k, v+l) - I_2(u+k+d, v+l)| \quad (7.3)$$

Seguendo la stessa linea di ragionamento si potrebbe sostituire il valore assoluto con una funzione di penalità più robusta, come la funzione di Cauchy (§A3.2).

La NCC (*Normalized Cross Correlation*) invece è una misura di similarità (dunque è da massimizzare). La si può vedere come il prodotto scalare delle due finestre (vettorizzate) diviso il prodotto delle norme:

$$NCC(u, v, d) = \frac{\sum_{(k,l)} I_1(u+k, v+l) I_2(u+k+d, v+l)}{\sqrt{\sum_{(k,l)} I_1(u+k, v+l)^2} \sqrt{\sum_{(k,l)} I_2(u+k+d, v+l)^2}} \quad (7.4)$$

Per ottenere invarianza a cambi di luminosità (di tipo additivo) tra le due immagini si può sottrarre a ciascun pixel la media della finestra, ottenendo la *Zero-mean NCC* o ZNCC.

7.2.1.2 Trasformata census

Vediamo ora una metrica particolarmente interessante, nella quale prima viene applicata alle immagini una trasformazione (*trasformata census*) basata sull'ordinamento locale dei livelli di grigio e quindi si misura la similarità delle finestre con distanza di Hamming sulle immagini trasformate.

La *trasformata census* [Zabih e Woodfill, 1994] si basa sul seguente operatore di confronto:

$$\xi(I, p, p') = \begin{cases} 1 & \text{se } I(p) < I(p') \\ 0 & \text{altrimenti} \end{cases} \quad (7.5)$$

dove $I(p)$ e $I(p')$ sono, rispettivamente, i valori dell'intensità dei pixel p e p' .

Se denotiamo la concatenazione di bit col simbolo \odot , la *trasformata census* per un pixel p nell'immagine I è:

$$\mathcal{C}[I(p)] = \bigodot_{p' \in S(p, \beta)} \xi(I, p, p') \quad (7.6)$$

dove $S(p, \beta)$ denota una finestra di raggio β centrata in p (finestra di trasformazione).

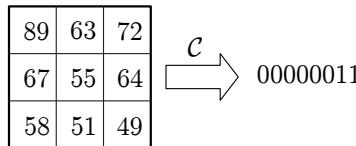


Fig. 7.5. Esempio di trasformata *census* con $\beta = 1$.

La trasformata *census* riassume la struttura spaziale locale. Infatti essa associa ad una finestra che circonda un pixel una stringa di bit che codifica i pixel che hanno un'intensità più bassa (o più alta) rispetto al pixel centrale, come esemplificato in figura 7.5.

L'accoppiamento avviene su finestre, tra immagini trasformate, confrontando stringhe di bit. Indichiamo col simbolo \ominus la distanza di Hamming tra due stringhe di bit, ovvero il numero di bit nei quali esse differiscono. Usando la stessa notazione che abbiamo usato in precedenza, la metrica SCH (*Sum of census hamming distances*) si scrive:

$$SCH(u, v, d) = \sum_{(k,l)} \mathcal{C}[I_1(u + k, v + l)] \ominus \mathcal{C}[I_2(u + k + d, v + l)] \quad (7.7)$$

Ogni termine della sommatoria è il numero di pixel dentro la finestra di trasformazione $S(p, \beta)$, il cui ordine relativo (ovvero l'avere intensità maggiore o minore) rispetto al pixel considerato cambia da I_1 a I_2 .

Questo metodo risulta invariante nei confronti di qualsiasi distorsione che conservi l'ordine delle intensità, per esempio la variazione del guadagno, la somma di una costante (bias) oppure la correzione gamma. In aggiunta a ciò, tale metodo risulta tollerante nei confronti degli errori dovuti a occlusioni (è robusto).

Computazionalmente questo metodo rende tutto estremamente efficiente perché le operazioni fondamentali sono semplici confronti di numeri interi, e si evitano perciò le operazioni in aritmetica in virgola mobile o anche le moltiplicazioni di interi. Il calcolo è puramente locale e la stessa funzione è calcolata in ogni pixel, quindi la si può applicare in parallelo. La trasformata *census* è, dunque, molto adatta per le implementazioni veloci in hardware.

In figura 7.6 mostriamo, a titolo di esempio, le mappe di disparità ottenute con SSD, NCC e SCH sulla coppia stereo di figura 7.1.

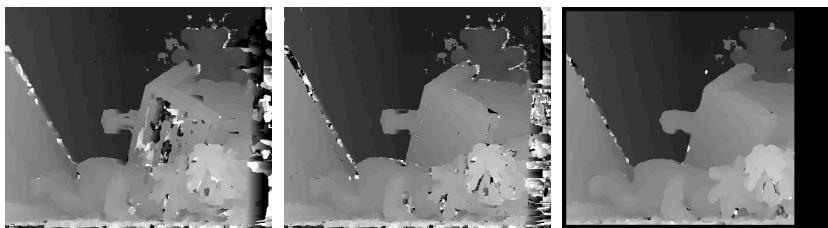


Fig. 7.6. Mappe di disparità prodotte da SSD, NCC e SCH (da sinistra a destra) con finestra 9×9 .

7.2.2 Compromesso affidabilità - accuratezza

Un ben noto problema degli algoritmi basati su finestre è legato alle discontinuità di profondità: se la finestra di correlazione copre una regione in cui la profondità varia, la disparità calcolata sarà inevitabilmente affetta da errore, poiché non esiste *una* disparità che si possa attribuire a *tutta* la finestra (e quindi al pixel centrale). Non si può tuttavia ridurre arbitrariamente la dimensione della finestra per evitare questo problema. Infatti, se la finestra è troppo piccola, il rapporto segnale (variazione di intensità) su rumore è basso e la disparità che si ottiene è poco affidabile. In sostanza, si hanno due richieste contrapposte: per l'accuratezza

si vorrebbero finestre piccole, per l'affidabilità si vorrebbe che ci fosse molta variabilità nella finestra, quindi finestre grandi.

Il fenomeno è ben visibile in figura 7.4: una finestra piccola sortisce disparità con errori casuali ovunque, mentre grosse finestre rimuovono gli errori casuali ma introducono errori sistematici in corrispondenza delle discontinuità della disparità.

7.2.2.1 Finestre adattative/eccentriche

La soluzione proposta da [Kanade e Okutomi, 1994] prevede una finestra le cui dimensioni sono selezionate adattativamente in base al rapporto segnale/rumore locale e alla variazione locale di disparità. L'idea è che la finestra ideale deve comprendere più variazione di intensità e meno variazione di disparità possibile. Poiché la disparità è inizialmente ignota, si parte con una stima ottenuta con finestra fissa e si itera, approssimando ad ogni passo la finestra ottima per ciascun punto, fino alla convergenza (eventuale). Sulla scorta di questo, è stato proposto un algoritmo più semplice basato su finestre di dimensione fissa ma eccentriche [Fusiello *e al.*, 1997]. Si impiegano nove finestre per ciascun punto, ognuna con il centro in una diversa posizione (figura 7.7). La finestra tra le nove che presenta un SSD minore è quella che più probabilmente copre la zona con meno variazione di disparità. Questo metodo affronta solo il problema della accuratezza, mentre per quanto riguarda l'affidabilità, si assume che la dimensione della finestra sia sufficiente.

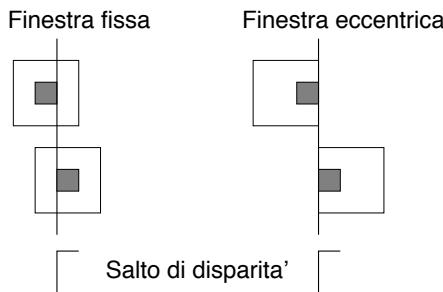


Fig. 7.7. Una finestra eccentrica può coprire una zona a disparità costante anche in prossimità di un salto di disparità.

7.2.2.2 Metodi multirisoluzione

Si tratta di metodi gerarchici che operano a diverse risoluzioni. L'idea è che al livello grossolano finestre larghe forniscono un risultato inaccurato

ma affidabile. Ai livelli più fini, finestre più piccole ed intervalli di ricerca più piccoli migliorano l'accuratezza. Distinguiamo due tecniche:

Coarse-to-fine: L'intervallo di ricerca è il medesimo ma opera su immagini a risoluzioni via via crescenti. La disparità ottenuta ad un livello viene usata come centro per l'intervallo al livello superiore.

Fine-to-fine: Si opera sempre sulla stessa immagine ma con finestre ed intervalli via via più piccoli. Come prima, la disparità ottenuta ad un livello viene usata come centro per l'intervallo al livello superiore.

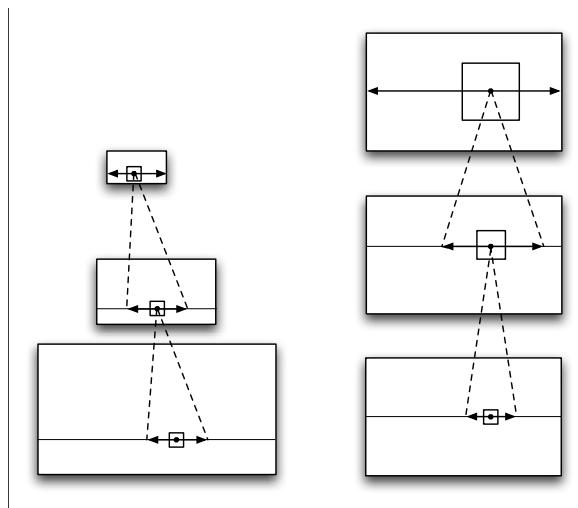


Fig. 7.8. Metodo *coarse-to-fine* (sinistra) e *fine-to-fine* (destra).

7.2.3 Indicatori di affidabilità

L'informazione di profondità che fornisce un algoritmo di stereopsi *area based* non è ovunque ugualmente affidabile. In particolare non vi è informazione per le zone di occlusione e per le aree ad intensità uniforme (o non tessiturate). Questa informazione incompleta può venire integrata con informazione proveniente da altri sensori, ma allora è necessario che essa sia accompagnata da una stima di affidabilità, la quale gioca un ruolo fondamentale nel processo di integrazione. Alcuni indicatori di affidabilità sono:

- Il valore della metrica di accoppiamento.
- La “piccatezza” della metrica nel suo minimo (o massimo per NCC).
- La varianza o l’entropia delle intensità (locale).
- Coerenza con i vicini, ovvero lisciezza della disparità.
- La coerenza destra-sinistra (si veda più avanti).

7.2.4 Occlusioni

Le occlusioni generano punti privi di corrispondente. Il primo passo per la gestione delle occlusioni è rilevarle, evitando che si creino false corrispondenze. Il vincolo di ordinamento, quando applicabile (oggetti coesi) può servire allo scopo.

Più efficace è il vincolo di coerenza destra-sinistra, che si basa sul vincoli di unicità. La coerenza destra-sinistra prescrive che se p viene accoppiato con p' effettuando la ricerca da I_1 a I_2 , allora p' deve essere accoppiato a p effettuando la ricerca da I_2 a I_1 .

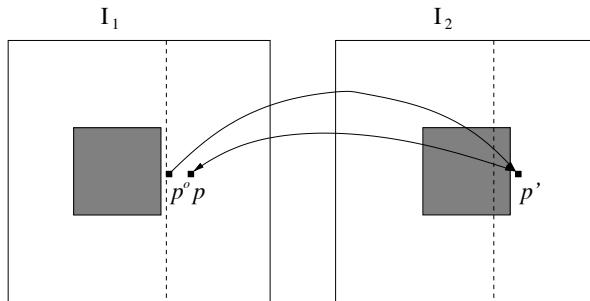


Fig. 7.9. Coerenza destra-sinistra.

Nel procedimento di accoppiamento, per ogni punto di I_1 viene cercato il corrispondente in I_2 . Se per esempio (vedi figura 7.9) una porzione della scena è visibile in I_1 ma non in I_2 , un pixel $p^o \in I_1$, il cui corrispondente è occluso, verrà accoppiato ad un certo pixel $p' \in I_2$ secondo la metrica prescelta. Se il vero corrispondente di p' è $p \in I_1$ – e supponendo che l’accoppiamento operi correttamente – anche p viene accoppiato a p' , violando il vincolo di unicità. Quale dei due è corretto? Lo scopro effettuando la ricerca degli accoppiamenti a partire da I_2 . In questo caso, p' viene accoppiato con il suo vero corrispondente p , dunque il punto p^o si può lasciare senza corrispondente.

Per i punti privi di corrispondente si può stimare una disparità per interpolazione dei valori vicini, oppure lasciarli come sono.

7.2.5 Altri metodi locali

Altri metodi di accoppiamento locali includono quelli basati sul gradiente (si veda il metodo di Kanade Lucas e Tomasi, § 10.3), quelli basati sulla segmentazione (prima segmenta e poi accoppia tra loro le regioni) e quelli basati su caratteristiche salienti (*feature*) dell’immagine.

7.2.5.1 Metodi basati su feature

I metodi *feature-based* estraggono dalle immagini caratteristiche salienti (*feature*) che siano (possibilmente) stabili rispetto al cambio del punto di vista. Il processo di accoppiamento (*matching*) sfrutta una misura di distanza tra i descrittori delle *feature*.

Spigoli (*edge*), angoli (§ 9.4), segmenti rettilinei e curvi, sono alcune delle *feature* che si possono impiegare. Non necessariamente, però, esse devono corrispondere ad una entità geometrica ben definita: possono essere definite dall’applicazione di un certo operatore (per esempio l’operatore di interesse di [Moravec, 1977].) Questi algoritmi sono veloci ma forniscono solo mappe di profondità sparse che vanno poi interpolate. Inoltre dipendono criticamente dalla fase di estrazione delle *feature* ed dal descrittore impiegato.

Classici esempi di *feature* dell’immagine impiegate nella ricerca stereo sono i punti di spigolo (*edge*), linee e angoli o punti salienti. Per esempio, un descrittore di *feature* per un segmento di retta (ripreso da [Trucco e Verri, 1998]) potrebbe contenere valori per:

- la lunghezza, l
- l’orientazione, θ
- le coordinate del punto medio, m
- il contrasto medio lungo la linea, c .

La seguente misura di similarità avrebbe senso in questo caso:

$$S = \frac{1}{w_0(l_l - l_r)^2 + w_1(\theta_l - \theta_r)^2 + w_2(m_l - m_r)^2 + w_3(c_l - c_r)^2} \quad (7.8)$$

dove w_1, w_2, w_3, w_4 sono pesi opportuni che hanno anche una funzione “normalizzante”[†].

Un algoritmo di accoppiamento *feature-based* molto semplice, poiché adotta una strategia *nearest neighbour*, è il seguente:

[†] In principio, variabili incommensurabili sommate tra loro andrebbero divise per la deviazione standard. Questo le rende adimensionali e ne uniforma la variabilità.

Algoritmo 7.2 PRIMO VICINO

Input: Due insiemi di *feature* nelle immagini.

Output: Accoppiamento delle *feature*.

- (i) Sia $R(f_l)$ la regione di ricerca dell'immagine di destra associata ad una *feature* f_l .
 - (ii) Per ogni *feature* f_l nell'immagine di sinistra:
 - calcola la misura di similarità tra f_l e ogni *feature* nell'immagine $R(f_l)$
 - scegli la *feature* dell'immagine di destra f_r che massimizza S
 - salva la corrispondenza
-

7.3 Metodi di accoppiamento globali

I metodi globali sfruttano i vincoli disponibili in modo non locale per ridurre la sensibilità a regioni per le quali l'accoppiamento fallisce (regioni uniformi, occlusioni). Ne risulta un problema di ottimizzazione che tipicamente ha un costo computazionale maggiore rispetto ai metodi locali.

I metodi globali possono essere ben compresi facendo riferimento alla cosiddetta *Disparity Space Image* (DSI). Si tratta di una immagine tridimensionale (un volume) \mathcal{V} , dove $\mathcal{V}(x, y, d)$ è il valore della metrica di accoppiamento tra il pixel (x, y) nella prima immagine ed il pixel $(x, y+d)$ nella seconda immagine. La mappa di disparità che ci aspettiamo che l'algoritmo produca può essere vista come una superficie dentro la DSI (una disparità per ogni (x, y)), la quale deve essere ottima rispetto ad una funzione costo che incorpora la metrica di accoppiamento e le penalità per la violazione dei vincoli di ordinamento, liscezza, ecc.

Il metodo globale che, nei suoi sviluppi più recenti, si è dimostrato essere il migliore, è quello basato sul taglio di un grafo (*graph cuts*) [Roy e Cox, 1998, Kolmogorov e Zabih, 2001].

In estrema sintesi, modelliamo il DSI come una rete di flusso, in cui i nodi corrispondono alle celle e gli archi connettono celle adiacenti, con una capacità associata che è funzione dei costi delle celle incidenti. Il taglio di costo minimo rappresenta la superficie cercata.

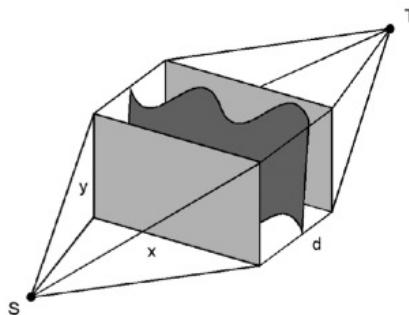


Fig. 7.10. Stima della disparità come taglio minimo in una rete di flusso (figura tratta da [Brown *e al.*, 2003]).

7.3.1 Spazio delle corrispondenze

Anche se il problema, per come l'abbiamo formulato, è inherentemente bidimensionale (si richiede una superficie), un compromesso per ridurre il costo computazionale consiste nel decomporre il problema in molti problemi unidimensionali (più semplici). In pratica si considera ciascuna *scan-line* indipendentemente dalla altre. Si tratta dunque di calcolare il profilo di disparità ottimo per ciascun *scan-line*, considerando la metrica di accoppiamento ed i vincoli di ordinamento, lisciezza, etc. . . .

Vi sono algoritmi che operano in una sezione (x, d) della DSI come [Intille e Bobick, 1994] oppure nel cosiddetto *match space*, come per esempio [Cox *e al.*, 1996] e [Ohta e Kanade, 1985]. In entrambi i casi si tratta di calcolare un percorso di costo minimo attraverso una matrice di costi, come illustrato nella figura 7.11. La programmazione dinamica viene utilmente impiegata a questo scopo.

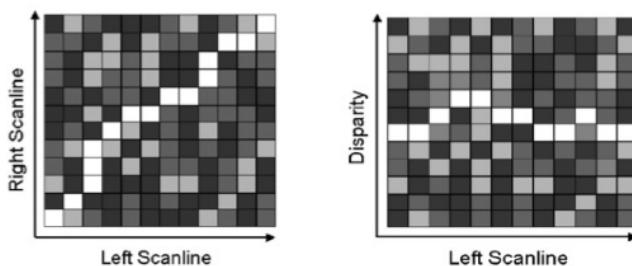


Fig. 7.11. Matrici dei costi. *Match space* (a sinistra) e sezione del DSI (a destra) (figura tratta da [Brown *e al.*, 2003]).

Lo spazio delle corrispondenze (*match space*) è una matrice che contiene i costi di ciascuna coppia di pixel di due linee epipolari corrispondenti. Scegliere un punto della matrice equivale a fissare una corrispondenza. Ragionare nel *match space* ci consente di osservare come le occlusioni sono collegate alle discontinuità della profondità. In molte scene, una discontinuità nella immagine sinistra corrisponde ad una occlusione in quella destra e viceversa. Ad esempio il RDS presenta una discontinuità di profondità ed una occlusione. Se ci riferiamo all'immagine sinistra la discontinuità è sul bordo sinistro del quadrato, mentre l'occlusione è in prossimità del bordo destro (infatti lì si trovano punti che scompaiono nell'immagine destra). Questa dualità occlusioni-discontinuità si comprende bene osservando la figura 7.12, che mostra il *match space* per una linea dello stereogramma a punti casuali. Si noti il salto verticale corrispondente alla discontinuità di disparità ed il buco orizzontale corrispondente alla occlusione. Se si scambia la destra con la sinistra, i salti orizzontali diventano verticali, e viceversa.

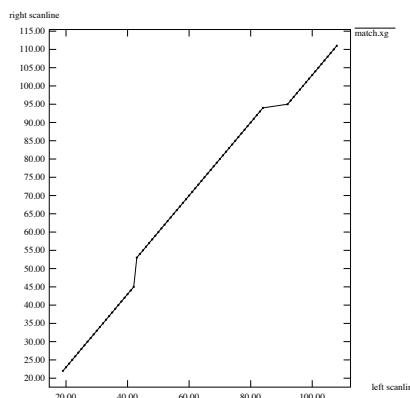


Fig. 7.12. Match space per una linea orizzontale dello stereogramma a punti casuali.

7.4 Classificazione dei metodi

Nella loro rassegna, [Scharstein e Szeliski, 2002] propongono di classificare i metodi di accoppiamento stereo lungo quattro dimensioni, che corrispondono anche ai quattro moduli di base (*building blocks*) per costruire un qualunque algoritmo.

- Metrica di accoppiamento (SSD, SAD, NCC, ...)

- Aggregazione di costi (per esempio somma su una finestra)
- Calcolo della disparità (locale, globale e quale metodo)
- Raffinamento (sub-pixel, rilevamento occlusioni, ...)

Menzioniamo infine che [Scharstein e Szeliski, 2002] propongono anche un protocollo standard per la valutazione degli algoritmi stereo. I risultati si possono consultare sul web a <http://vision.middlebury.edu/stereo/>.

7.5 Illuminazione strutturata

Il processo di stereopsi visto precedentemente prende il nome di “passivo”, contrapposto ai metodi “attivi” che includono nel sistema un dispositivo di illuminazione strutturata (*structured lighting*). Questi sono maggiormente efficaci nel risolvere il problema delle corrispondenze. Vi sono varie tipologie di sistemi di illuminazione strutturata:

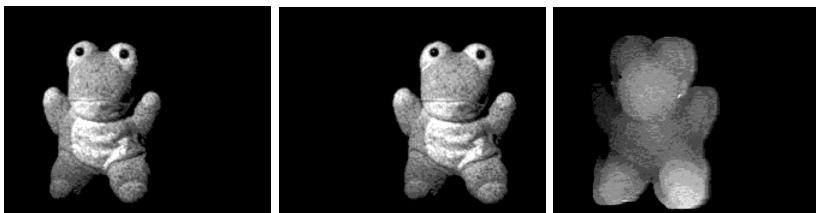


Fig. 7.13. Stereo attivo con tessitura artificiale. Coppia stereo con tessitura artificiale “sale e pepe” proiettata e mappa di disparità risultante.

Stereo attivo: viene proiettata una **tessitura artificiale** sulla scena (per esempio punti casuali “sale e pepe”), agevolando il calcolo delle corrispondenze (come in figura 7.13).

Stereo attivo: un **raggio laser** scandisce la scena, proiettando sulle superfici un punto che viene facilmente rilevato e messo in corrispondenza nelle due immagini. È necessario prendere molte immagini, poiché per ciascuna coppia di immagini solo ad un punto viene assegnata la disparità.

Stereo attivo: una **lama di luce** (laser) viene fatta passare sulla scena[†] (figura 7.14). Le strisce determinate dalla lama nelle due immagini intersecate con le rispettive rette epipolari forniscono i punti corrispondenti. Per ciascuna coppia di immagini solo ai punti della striscia viene assegnata la disparità. È più veloce della soluzione precedente, ma servono sempre molte immagini.

[†] Si può ottenere una lama facendo passare un raggio attraverso una lente cilindrica.

Triangolazione attiva: in realtà quando si proietta una **lama di luce**

è possibile rimuovere una fotocamera, e procedere alla triangolazione intersecando il piano della lama di luce con il raggio ottico. In questo caso è necessario fare un modello geometrico del proiettore laser e calibrarlo, per poter determinare il piano nello spazio.

Triangolazione attiva: invece che proiettare una solo piano, se ne possono proiettare molti simultaneamente, usando un proiettore di **bande di luce**. In questo caso le bande devono essere codificate in qualche modo: questo è il principio della luce codificata (*coded-light*) (figura 7.15). A parità di numero di piani di triangolazione individuati, posso usare meno immagini (tipicamente ne bastano una decina).

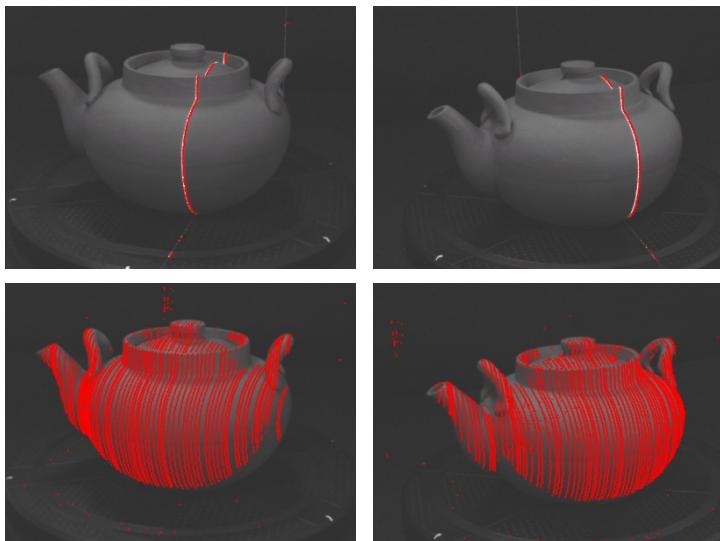


Fig. 7.14. Esempio di acquisizione con stereo attivo e lama laser. Nella riga superiore si vedono le due immagini acquisite dalle fotocamere, nelle quali si nota la striscia formata dalla lama laser. Sotto sono visualizzati, in sovrapposizione, i punti rilevati dopo una passata.

Dunque, tra le tecniche ad illuminazione strutturata consideriamo lo **stereo attivo** e la **triangolazione attiva**. Il primo si ha quando sono presenti due fotocamere e l'illuminazione serve a rendere più agevole il calcolo delle corrispondenze, che viene però effettuato come per lo stereo passivo, come pure la successiva triangolazione. Nella triangolazione

attiva, invece, si ha una fotocamera ed un proiettore (calibrato) ed il calcolo delle corrispondenze e la triangolazione vengono adeguati alla situazione. Ci sono problemi di zone d'ombra se sorgente e rivelatore non sono allineati (analogamente alle occlusioni nello stereo).

7.5.1 Triangolazione attiva

Una fotocamera riprende una scena in cui un dispositivo[†] proietta un piano di luce. La fotocamera ed il proiettore “vedono” la scena da posizioni diverse. Otteniamo la distanza dei punti della scena per triangolazione, intersecando il raggio ottico di un punto dell’immagine con il piano di luce corrispondente emesso dal proiettore.

Consideriamo un punto M dello spazio di coordinate $\tilde{\mathbf{M}}_c = [x_c, y_c, z_c]^T$ nel riferimento della fotocamera. La trasformazione rigida (R, \mathbf{t}) che porta il riferimento fotocamera sul riferimento del proiettore è noto dalla calibrazione, quindi, le coordinate dello stesso punto nel riferimento del proiettore sono:

$$\tilde{\mathbf{M}}_p = \mathbf{R}\tilde{\mathbf{M}}_c + \mathbf{t}. \quad (7.9)$$

La proiezione del punto M sul piano immagine della fotocamera è (in coordinate normalizzate) $\mathbf{p}_c = [u_c, v_c, 1]^T$, e si ottiene dall’equazione di proiezione semplificata:

$$\mathbf{p}_c = \begin{bmatrix} u_c \\ v_c \\ 1 \end{bmatrix} = \begin{bmatrix} x_c/z_c \\ y_c/z_c \\ 1 \end{bmatrix} = \frac{1}{z_c} \tilde{\mathbf{M}}_c. \quad (7.10)$$

Per quanto riguarda il proiettore, lo modelliamo come una fotocamera, in cui però la coordinata verticale del punto proiettato è incognita (assumiamo che le barre di luce siano verticali, nel riferimento interno del proiettore). Sia u_p la coordinata del piano che illumina M , allora, mantenendo il linguaggio della fotocamera, diremo che M si proietta sul punto (in coordinate normalizzate) $\mathbf{p}_p = [u_p, v_p, 1]^T$, nel piano immagine del proiettore (attenzione: v_p è incognito).

$$\mathbf{p}_p = \frac{1}{z_p} \tilde{\mathbf{M}}_p. \quad (7.11)$$

Usando le equazioni (7.9), (7.10) e (7.11) otteniamo l’equazione vettoriale

$$z_p \mathbf{p}_p - z_c \mathbf{R} \mathbf{p}_c = \mathbf{t}. \quad (7.12)$$

[†] Può essere un proiettore LCD o un laser con lente cilindrica o altro.

che si scomponete in un sistema di tre equazioni scalari:

$$\left\{ \begin{array}{l} z_p u_p - z_c \mathbf{r}_1^\top \mathbf{p}_c = t_1 \\ z_p v_p - z_c \mathbf{r}_2^\top \mathbf{p}_c = t_2 \\ z_p - z_c \mathbf{r}_3^\top \mathbf{p}_c = t_3 \end{array} \right. \quad (7.13)$$

delle quali la seconda non può essere usata (fin qui era tutto come per la triangolazione nello stereo passivo). Ricaviamo z_p dalla terza equazione e lo sostituiamo nella prima, ottenendo, dopo qualche passaggio, il risultato cercato, ovvero la profondità z_c del punto M (nel riferimento della fotocamera):

$$z_c = \frac{t_1 - t_3 u_p}{(u_p \mathbf{r}_3^\top - \mathbf{r}_1^\top) \mathbf{p}_c}. \quad (7.14)$$

7.5.1.1 Metodo a luce codificata

Nel metodo a *luce codificata* (*coded-light*) vengono proiettate, nello stesso istante, numerose strisce (o bande) di luce, che devono essere codificate in qualche modo per distinguerle. La codifica più semplice è realizzata assegnando una diversa luminosità ad ogni direzione di proiezione, per esempio proiettando una scala di intensità lineare. Un'ulteriore sviluppo di questa tecnica sfrutta il colore per la codifica delle direzioni di proiezione [Boyer e Kak, 1987]. Una tecnica molto robusta è la cosiddetta codifica spazio-temporale delle direzioni di proiezione. Questo metodo si realizza proiettando, all'interno della scena, una sequenza temporale di n *pattern* di luce a bande, opportunamente generati da un proiettore a cristalli liquidi (LCD) controllato da un calcolatore. A ciascuna direzione di proiezione viene così associato un codice di n bit, in cui il bit i -esimo indica se la corrispondente banda era in luce o in ombra nel *pattern* i -esimo (vedi figura 7.15). Questi metodi permette di distinguere 2^n direzioni di proiezione differenti.

Una fotocamera, da una posizione diversa da quella del proiettore, acquisisce le n immagini a livelli di grigio dei *pattern* a strisce proiettati sulla superficie dell'oggetto. Queste vengono quindi convertite in forma binaria in modo da separare le aree illuminate dal proiettore da quelle non illuminate e per ogni pixel si memorizza il codice di n bit, che, per quanto detto sopra, codifica la direzione di illuminazione della striscia più sottile che ha illuminato il pixel.

Conoscendo la geometria del sistema, la direzione del raggio ottico

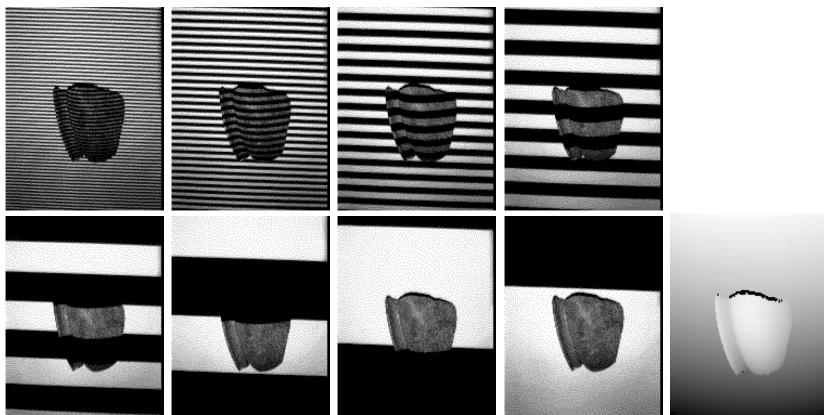


Fig. 7.15. Luce codificata. A destra immagini delle bande proiettate, a destra immagine *range* finale. Le bande sono approssimativamente parallele alle colonne dell'immagine. Da <http://www.prip.tuwien.ac.at/Research/3DVision/struct.html>.

della fotocamera e l'equazione del piano della banda corrispondente, le coordinate 3D del punto della scena osservato possono essere calcolate intersecando il raggio ottico con il piano (come visto prima nella (7.14)).

Per minimizzare l'effetto degli errori si impiega la codifica di Gray per le direzioni dell'illuminazione, in modo che strisce adiacenti differiscano di un solo bit.

Il risultato dipende anche dalla corretta binarizzazione delle strisce nelle immagini. Per rendere questo passo indipendente dal cambio delle condizioni di illuminazione dell'ambiente e dalla variazione delle proprietà di riflessione della superficie, le immagini vengono binarizzate con una soglia che varia attraverso l'immagine. Questa soglia si calcola, per ciascun pixel, come la media tra il livello di grigio nell'immagine della scena completamente illuminata ed il livello di grigio nell'immagine della scena non illuminata (le due immagini vanno acquisite oltre alle n con i *pattern* proiettati).

Il calcolo della immagine di soglia è utile anche per identificare le aree d'ombra presenti nella scena, ovvero quelle zone osservate dalla fotocamera ma non illuminate dal proiettore. Esse sono caratterizzate dal fatto che hanno variazioni minime di livello di grigio tra l'immagine illuminata e quella non illuminata. In queste particolari aree si evita di binarizzare le bande e di conseguenza non si ha una valutazione dei valori di profondità.

Esercizi ed approfondimenti

- 7.1 Implementare in MATLAB la correlazione con SSD senza impiegare cicli `for`. Suggerimento: pensare al DSI.
 Si veda la funzione `imstereo`.
- 7.2 Il proiettore si può modellare, analogamente alla fotocamera stereopeica, con una matrice di proiezione $2 \times 4 P$ che porta punti 3D in rette 2D:

$$P \simeq \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \end{bmatrix} \quad (\text{E7.1})$$

La calibrazione e la triangolazione sono del tutto analoghe al caso dello stereo passivo (vedi [Trobina, 1995]).

Bibliografia

- Boyer K.; Kak A. (1987). Color-encoded structured light for rapid active ranging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **9**(10), 14–28.
- Brown M. Z.; Burschka D.; Hager G. D. (2003). Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**(8), 933–1008.
- Cox I. J.; Hingorani S.; Maggs B. M.; Rao S. B. (1996). A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, **63**(3), 542–567.
- Fusiello A.; Roberto V.; Trucco E. (1997). Efficient stereo with multiple windowing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 858–863, Puerto Rico. IEEE Computer Society Press.
- Intille S. S.; Bobick A. F. (1994). Disparity-space images and large occlusion stereo In *European Conference on Computer Vision*. A cura di Eklundh J.-O., pp. 179–186, Stockholm, Sweden. Springer-Verlag.
- Kanade T.; Okutomi M. (1994). A stereo matching algorithm with an adaptive window: Theory and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**(9), 920–932.
- Kolmogorov V.; Zabih R. (2001). Computing visual correspondence with occlusions using graph cuts. *Proceedings of the International Conference on Computer Vision*, **2**, 508.
- Marr D.; Poggio T. (1976). Cooperative computation of stereo disparity. *Science*, **194**, 283–287.

- Moravec H. P. (1977). Towards automatic visual obstacle avoidance. In *Proceedings of the International Joint Conference on Artificial Intelligence*, p. 584.
- Ohta Y.; Kanade T. (1985). Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **7**(2), 139–154.
- Roy S.; Cox I. J. (1998). A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proceedings of the International Conference on Computer Vision*, p. 492, Washington, DC, USA. IEEE Computer Society.
- Scharstein D.; Szeliski R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, **47**(1), 7–42.
- Trobina M. (1995). Error model of a coded-light range sensor. Relazione Tecnica BIWI-TR-164, ETH-Zentrum.
- Trucco E.; Verri A. (1998). *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall.
- Zabih R.; Woodfill J. (1994). Non-parametric local transform for computing visual correspondence. In *Proceedings of the European Conference on Computer Vision*, volume 2, pp. 151–158, Stockholm.

8

Ricostruzione volumetrica

L’obiettivo della Ricostruzione Volumetrica (si veda [Dyer, 2001] per una rassegna) è creare una rappresentazione che descriva non soltanto la superficie di una regione, ma anche lo spazio che essa racchiude. L’ipotesi fondamentale è che esista un volume noto e limitato all’interno del quale giacciono gli oggetti d’interesse. Tale volume può essere visto come un cubo che circonda la scena e viene rappresentato mediante una griglia discreta di “cubetti” chiamati *voxel*, equivalenti 3D dei pixel. La ricostruzione coincide con l’assegnazione di un’*etichetta di occupazione* (o colore) ad ogni elemento di volume. Il valore di occupazione è solitamente binario (trasparente o opaco).

Rispetto alle tradizionali tecniche di stereopsi, la ricostruzione volumetrica offre alcuni vantaggi: evita il difficile problema della ricerca delle corrispondenze (funziona anche per superfici non tessiture), consente il trattamento esplicito delle occlusioni e consente di ottenere *direttamente* un modello tridimensionale dell’oggetto (non bisogna allineare porzioni del modello) integrando simultaneamente tutte le viste (che sono dell’ordine della decina). Come nella stereopsi, le fotocamere sono *calibrate*.

8.1 Ricostruzione da sagome

Intuitivamente una sagoma (*silhouette*) è il profilo di un oggetto, comprensivo della sua parte interna. Più precisamente si definisce *sagoma* una immagine binaria il cui valore in un certo punto (x, y) indica se il raggio ottico che passa per il pixel (x, y) interseca o meno la superficie di un oggetto nella scena. Un pixel della sagoma può essere un punto dell’oggetto (nero) o un punto dello sfondo (bianco).

Ogni punto nero della sagoma identifica un raggio ottico che inter-

seca l'oggetto ad una distanza ignota. L'unione di tutti i raggi ottici che attraversano pixel neri della sagoma definisce un cono generalizzato all'interno del quale è contenuto l'oggetto. L'intersezione dei coni generalizzati associati a tutte le fotocamere identifica un volume all'interno del quale giace l'oggetto. Questo volume viene preso come una approssimazione dell'oggetto. La tecnica introdotta in [Martin e Aggarwal, 1983], basata sulla intersezione dei coni generati dalle sagome, prende il nome di *Shape from silhouette*.

Al fine di risparmiare memoria si sfrutta la struttura dati di *octree* per memorizzare le etichette dei voxel nello spazio campionario. Gli octree sono una rappresentazione più efficiente in termini di spazio quando la scena contiene ampie zone vuote. Si tratta di alberi a otto vie in cui ciascun nodo rappresenta un certo settore di spazio e i nodi figli sono le otto suddivisioni equispaziate di quel settore (*ottanti*).

Il *visual hull*, introdotto da [Laurentini, 1994], è la miglior approssimazione ottenibile da un numero infinito di sagome. Può essere definita come la forma massimale che restituisce le stesse sagome dell'oggetto reale per tutte le viste esterne al guscio convesso dell'oggetto (figura 8.1). Il *visual hull* è diverso dal *convex hull*: può essere una approssimazione migliore o peggiore dell'oggetto rispetto a quest'ultimo. Nella pratica è disponibile solo un numero finito di sagome e quello che si ottiene è una approssimazione del *visual hull*. Il volume dell'approssimazione decresce all'aumentare del numero di viste.

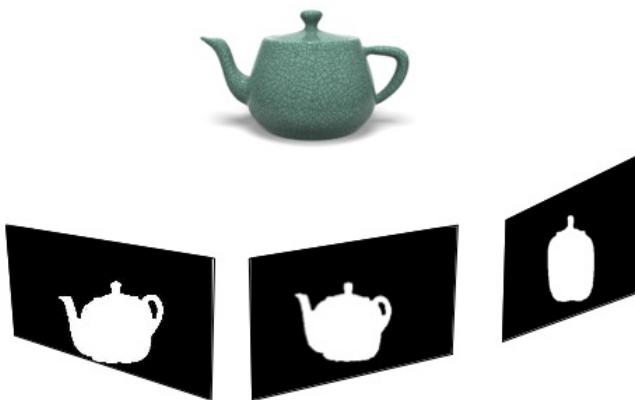


Fig. 8.1. Principio dello *shape from silhouette*.

8.1.1 Algoritmo di Szeliski

Vedremo ora una versione leggermente modificata dell'algoritmo di Szeliski [1993] per calcolare l'octree che rappresenta il *visual hull* di un oggetto a partire da un insieme di sagome.

Si usa una fotocamera calibrata fissa ed un piatto rotante su cui è posto l'oggetto (vedi esercizio 1). Il sistema di riferimento mondo ha un asse coincidente con l'asse di rotazione del piatto. L'orientazione angolare del piatto viene impostata manualmente di volta in volta (mediante tacche di riferimento sul lato del piatto). Per ciascuna posizione si prende una immagine.

Per ricavare la sagoma si esegue una sottrazione adattiva dello sfondo. Si acquisiscono alcune immagini dello sfondo e si calcola il minimo, il massimo, la media e la varianza per ogni pixel. Queste misure vengono usate per calcolare un intervallo di valori normali per ciascun pixel dello sfondo. Ogni pixel la cui intensità cade al di fuori di questo intervallo viene considerato appartenente all'oggetto. In seguito, le sagome così ottenute vengono sottoposte ad un filtraggio morfologico per eliminare il rumore e rendere i contorni più netti.

L'octree viene costruito ricorsivamente suddividendo ogni cubo in otto cubetti (*ottanti*) a partire dal nodo radice che rappresenta il volume di lavoro. Ogni cubo ha associato un colore:

- **nero:** rappresenta un volume occupato;
- **bianco:** rappresenta un volume vuoto;
- **grigio:** nodo interno la cui classificazione è ancora incerta.

Si verifica, per ciascun ottante, se la sua proiezione sull'immagine *i-esima* è interamente contenuta nella regione nera. Se ciò si verifica per tutte le N sagome, l'ottante viene marcato come “nero”. Se invece la proiezione dell'ottante è interamente contenuta nello sfondo (bianco), anche per una sola fotocamera, l'ottante viene marcato come “bianco”. Se si verifica uno di questi due casi, l'ottante diventa una foglia dell'octree e non viene più processato. Altrimenti l'ottante viene classificato come “grigio” e suddiviso a sua volta in otto figli.

Per limitare la dimensione dell'albero, gli ottanti “grigi” di dimensione minima vengono marcati come “neri”. Al termine si ottiene un octree che rappresenta la struttura 3D dell'oggetto.

Un ottante proiettato sull'immagine forma in generale un esagono irregolare. Eseguire un test accurato di intersezione tra una tale figura e la sagoma può essere complesso. Solitamente si adopera un test approssimato basato sulla *bounding box* dell'esagono che, pur essendo approssi-

mativo, classifica con certezza i cubi che stanno totalmente dentro o fuori dalla sagoma. Anche se qualche ottante viene erroneamente classificato come “grigio”, la decisione sul suo colore definitivo viene semplicemente rimandata alla successiva iterazione.



Fig. 8.2. Immagine di una teiera sul piatto rotante e relativa sagoma. Ricostruzione volumetrica ottenuta con 12 sagome (i voxel sono rappresentati da sfere).

8.2 Ricostruzione da foto-coerenza

Quando le immagini di input non sono delle semplici sagome binarie, ma sono foto a colori o a livelli di grigio, è possibile sfruttare l’informazione fotometrica aggiuntiva per migliorare il processo di ricostruzione 3D (*shape from Photo-consistency*).

Si può interpretare l’insieme di immagini in input come un insieme di *vincoli* posti sulla scena 3D da ricostruire. La scena ricostruita, proiettata secondo le MPP note, dovrebbe restituire delle immagini molto simili a quelle di partenza. La definizione di somiglianza dipende dal grado di accuratezza che vogliamo raggiungere.

Può esistere più di una ricostruzione che soddisfa i vincoli dati. Si dice che una riproduzione è coerente se tutti i punti delle superfici visibili nella scena sono *foto-coerenti* rispetto a ciascuna immagine. Un punto di una superficie della scena si definisce *foto-coerente* (*photo-consistent*) con un insieme di immagini se, per ogni immagine I_k in cui esso è visibile, la sua irradianza (vista dalla fotocamera k) è uguale alla intensità del corrispondente pixel nell’immagine. L’uguaglianza può essere definita sulla base della deviazione standard o di una certa norma tra le coppie di colori.

Vi sono principalmente due metodi per determinare la foto-coerenza di un voxel. Nel primo si proietta il centroide del voxel in ciascuna immagine e si applica una sogliatura sulla varianza dei colori dei pixel

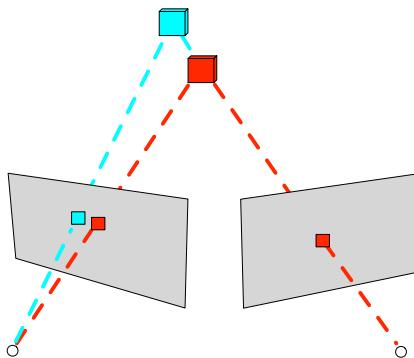


Fig. 8.3. Foto-coerenza. Se ignorassi che il voxel blu non è visibile nella immagine di destra potrei erroneamente considerarlo non foto-coerente.

così ottenuti. Un secondo metodo, meno sensibile al rumore, sfrutta i colori di tutti i pixel intersecati dalla proiezione del voxel (e non soltanto il centroide).

Il vicolo di foto-coerenza offre alcuni vantaggi rispetto alla classica ricerca delle corrispondenze nelle immagini, seguita da un passo di triangolazione per ricostruire la struttura 3D:

- il test di foto-coerenza richiede soltanto delle semplici operazioni di proiezione e confronto tra pixel, dove invece la ricerca delle corrispondenze tra immagini è un compito complesso;
- i metodi basati sulle corrispondenze possono lasciare dei “buchi” (se non usano tutti i punti) oppure essere affetti da errori (se usano tutti i punti).

D'altra parte per verificare correttamente la foto-coerenza sono richieste informazioni sulla geometria degli oggetti, sulle proprietà di riflettanza delle superficie sulla direzione dell'illuminazione. Un caso molto speciale (che supporremo vero) è quello in cui si possono ipotizzare superfici lambertiane.

Anche facendo questa ipotesi, rimangono due problemi. In primo luogo, come determinare la ricostruzione più corretta all'interno della classe di tutte le possibili scene coerenti con i vincoli. In secondo luogo, sono necessari dei metodi efficienti per determinare la visibilità di un voxel (condizione necessaria al test di foto-coerenza).

Infatti, quando si applica il test di foto-coerenza ad un voxel, è necessario sapere su quali immagini quel voxel si proietta, altrimenti potrebbe essere considerato erroneamente non foto-coerente, come accadrebbe al

voxel blu di figura 8.3 se si ignorasse che esso è occluso dal voxel rosso nella vista di destra. Nel caso specifico, la foto-coerenza del voxel rosso deve essere valutata prima di quello blu. In generale, la foto-coerenza di un voxel dovrebbe venire valutata solo dopo che tutti i voxel che ne potrebbero influenzare la visibilità sono stati valutati (ovvero hanno una etichetta opaco/trasparente).

L'implementazione può partire da uno spazio completamente vuoto come in [Seitz e Dyer, 1999] e marcare come opachi i voxel che superano il test di foto-coerenza (si parla allora di *voxel coloring*, equivalente a modellare con la creta), oppure può partire da uno spazio pieno [Kutulakos e Seitz, 2000] e rimuovere selettivamente i voxel che non superano il test (*space carving*, equivalente alla scultura). Osserviamo che il primo approccio risulta in un insieme di voxel che definiscono soltanto le superfici della scena, mentre nel secondo caso si ottiene una ricostruzione del pieno volume degli oggetti.

8.2.1 Voxel coloring

A causa delle complesse dipendenze tra le superfici della scena e la visibilità dei voxel, è di vitale importanza individuare dei metodi efficienti per eseguire il test di visibilità. In particolare, se esiste un ordinamento topologico dei voxel tale che la visibilità di un certo voxel può essere influenzata solamente dai voxel che lo precedono nell'ordinamento, diventa possibile realizzare un algoritmo che verifichi la foto-coerenza con una sola passata.

Ciò è possibile quando abbiamo a che fare con una singola fotocamera oppure quando le fotocamere si trovano tutte dalla stessa parte di un piano. In tal caso i voxel possono essere visitati facendo scorrere detto piano dai voxel più vicini ai voxel più lontani dai centri ottici delle fotocamere. È stato inoltre dimostrato da [Seitz e Dyer, 1999] che un tale ordinamento topologico dei voxel è in generale possibile quando nessun punto della scena cade all'interno del guscio convesso formato dai centri ottici delle fotocamere. Questa generalizzazione consente di realizzare l'algoritmo facendo scorrere non un semplice piano, ma un fronte di superfici a distanza crescente dal guscio convesso.

Quando la configurazione delle fotocamere consente un ordinamento topologico dei voxel come descritto nel paragrafo precedente, una singola passata attraverso la scena è sufficiente per creare una ricostruzione foto-coerente: il *voxel coloring* è un algoritmo ad una passata che, partendo

dallo spazio vuoto, marca come opachi i voxel che passano il test di foto-coerenza.

Le occlusioni vengono memorizzate in una semplice mappa binaria, con un bit per ciascun pixel delle immagini di input. Quando un voxel viene trovato coerente (e quindi opaco), vengono “accesi” i bit di occlusione corrispondenti alle proiezioni del voxel sulle varie immagini. In ogni momento l’insieme di visibilità di un voxel è dato dai pixel su cui il voxel si proietta e i cui bit di occlusione sono nulli.

Il voxel coloring è una tecnica elegante ed efficiente, ma il vincolo di ordinamento dei voxel pone una limitazione significativa: le fotocamere non possono circondare la scena. Ciò significa che alcune superfici saranno inevitabilmente invisibili e non potranno essere ricostruite.

8.2.2 Space carving

Per configurazioni generiche delle fotocamere, non è sufficiente una singola passata perché non esiste un ordinamento topologico dei voxel rispetto a tutti i punti di vista. Ovvero non posso garantire che la visibilità di un voxel non cambi dopo che ne ho valutato la foto-coerenza.

Kutulakos e Seitz [2000] hanno proposto un algoritmo denominato *space carving* (8.1) per la ricostruzione di scene con posizionamento arbitrario delle fotocamere. Poiché ogni modifica dell’etichetta di un voxel ha effetto sulla visibilità di altri voxel, è necessario iterare l’algoritmo finché non si verifica più alcun cambiamento. La convergenza è garantita dal fatto che l’algoritmo è *conservativo*, ovvero modifica l’assegnamento dei voxel soltanto da opaco a trasparente, mai viceversa (non scava mai un voxel che non dovrebbe). Infatti, [Kutulakos e Seitz, 2000] hanno dimostrato che l’algoritmo non rimuove mai voxel che nella scena sono opachi, purché si usi un’opportuna misura di foto-coerenza. In particolare deve trattarsi di una misura *monotona*: se un certo insieme di pixel è incoerente, allora anche ogni sovrainsieme di quei pixel sarà incoerente.

Per la valutazione dei voxel si impiega una procedura detta *multiple plane sweep*: ad ogni iterazione si spazzola (*sweep*) la scena con un piano di diversa orientazione. La visibilità di un voxel viene valutata prendendo in considerazione soltanto le fotocamere che si trovano di fronte al piano corrente.

Per semplificare vengono scelti i tre piani paralleli agli assi coordinati per spazzolare la scena, fatti scorrere nella direzione positiva e negativa degli assi. Inoltre, al termine di ciascuna iterazione viene applicato un

Algoritmo 8.1 SPACE CARVING

Input: Immagini calibrate, voxelizzazione dello spazio V
Output: Assegnamento di V a opaco/trasparente

```

set all voxels opaque
loop
{
    AllVoxelsConsistent := TRUE
    for every opaque voxel V
    {
        find the set S of pixels where V is visible
        if S has consistent color
        {
            V := average color of all pixels in S
        }
        else
        {
            AllVoxelsConsistent := FALSE
            V := transparent
        }
    }
    if AllVoxelsConsistent == TRUE
        quit
}

```

ulteriore test di foto-coerenza rispetto a *tutte* le fotocamere per rimuovere eventuali voxel di bordo[†] in eccesso.

Che relazione esiste tra la scena ricostruita e tutte le altre scene foto-coerenti? Gli autori [Kutulakos e Seitz, 2000] hanno dimostrato che la scena ottenuta con l'algoritmo di *space carving* è il volume *massimale* che rispetta i vincoli di foto-coerenza, denominato **photo hull**. La dimostrazione si basa sul fatto che lo *space carving* rimuove voxel opachi fino ad arrivare ad un voxel di bordo foto-coerente. Ne consegue che il primo voxel foto-coerente che si trova lungo ciascun raggio ottico è un voxel di superficie.

Se la scena è nota contenere un singolo oggetto connesso, si può imporre all'algoritmo un vincolo che impedisca di rimuovere un voxel se la sua rimozione comporterebbe una locale disconnessione dei voxel vicini.

Un problema di questo metodo è che nel test di foto-coerenza non vengono impiegate tutte le fotocamere ma soltanto quelle che si trovano di fronte al piano corrente, anche se il voxel è visibile da altre fotocamere. Ciò comporta un'approssimazione del test e, in genere, una tendenza a conservare più voxel del necessario.

[†] Un voxel di bordo è un voxel opaco adiacente ad un voxel trasparente.



Fig. 8.4. Immagine reale e due viste della ricostruzione volumetrica con *space carving*. Da [Kutulakos e Seitz, 2000] e <http://homepages.inf.ed.ac.uk/cgi/rbf/CVONLINE/>.

8.2.3 Ottimizzazione della ricostruzione

I metodi finora descritti forniscono risultati già sufficientemente fotografistici, ma possono essere ulteriormente migliorati. La relazione tra la scena reale ed il *photo hull* dipende da diversi fattori, che possono condizionare l'accuratezza della ricostruzione:

- la precisione della funzione di riflettanza delle superfici;
- errori nella stima dell'orientazione delle superfici e dell'illuminazione possono provocare errori nel test di foto-coerenza;
- la discretizzazione in voxel causa fenomeni di *aliasing*;
- la dipendenza della foto-coerenza da una soglia causa errori nella classificazione dei voxel.

L'effetto degli errori nel test di foto-coerenza è doppiamente negativo: determina buchi o false concavità nelle zone in cui la soglia era troppo alta (l'algoritmo ha “scavato” troppo in profondità); determina ingrossamenti o false convessità nelle zone in cui la soglia era troppo bassa (l'algoritmo si è fermato troppo presto). Non esiste in generale una soglia che vada bene per tutte le superfici. Solitamente si tende ad abbondare per evitare che vengano rimossi voxel appartenenti alla scena.

Il problema dei “buchi” può essere risolto parzialmente con opportuni algoritmi di riempimento [Curless e Levoy, 1996]. Un'altra strategia è quella di processare il *photo hull* in seguito alla ricostruzione, formulando un problema di ottimizzazione. L'ottimizzatore aggiunge o rimuove iterativamente dei voxel di bordo finché non viene minimizzato l'*errore di riproiezione*, una misura di somiglianza tra le immagini di input ed il modello (solitamente si tratta della somma delle differenze al quadrato).

Il problema degli ingrossamenti è particolarmente pronunciato in regioni con basse variazioni di colore. Quando due punti di superficie hanno radianza simile, il test di foto-coerenza sui voxel antistanti tale superficie può fallire.

L'accuratezza della ricostruzione può essere migliorata anche minimizzando la distanza tra le sagome riproiettate dal modello e le sagome reali.

Il problema dell'aliasing risulta particolarmente fastidioso quando la risoluzione dei voxel è grezza. Una possibile soluzione è assegnare a ciascun voxel un grado di opacità compreso tra 0 (completamente trasparente) e 1 (completamente opaco).

Esercizi ed approfondimenti

- 8.1 Determinare l'asse di rotazione del piatto a partire da due immagini dell'oggetto di calibrazione, prima e dopo una certa rotazione incognita.

Soluzione. Tramite calibrazione si ricavano le due matrici di rotazione R_1 ed R_2 relative alle due posizioni. A questo punto applichiamo una composizione delle due matrici R_1 ed R_2 al fine di ottenere la matrice di rotazione del turntable intorno al proprio asse: $R = R_1^T R_2$. Estrapoliamo quindi da R la sua rappresentazione asse/angolo.

- 8.2 Nel caso di oggetti semitransparenti, bisogna aggiungere un valore di opacità al voxel, accanto al colore. Interessante è il parallelo con i metodi di tomografia al calcolatore (Computerized Tomography). [Gering e Wells, 1999, Dachille *e al.*, 2000] infatti hanno applicato tecniche tomografiche a questo problema.

Bibliografia

- Curless B.; Levoy M. (1996). A volumetric method for building complex models from range images. In *SIGGRAPH: International Conference on Computer Graphics and Interactive Techniques*, pp. 303–312, New Orleans, Louisiana.
- Dachille F.; Mueller K.; Kaufman A. (2000). Volumetric backprojection. In *Proceedings of the 2000 IEEE symposium on Volume visualization*, pp. 109–117. ACM Press.
- Dyer C. (2001). Volumetric scene reconstruction from multiple views

- In *Foundations of Image Understanding*. A cura di Davis L. S., capitolo 16. Kluwer, Boston.
- Gering D. T.; Wells W. M. (1999). Object modeling using tomography and photography. In *Proc. IEEE Workshop on Multi-View Modeling and Analysis of Visual Scenes*, pp. 11–18.
- Kutulakos K. N.; Seitz S. M. (2000). A theory of shape by space carving. *International Journal of Computer Vision*, **38**(3), 199–218.
- Laurentini A. (1994). The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**(2), 150–162.
- Martin W. N.; Aggarwal J. K. (1983). Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **5**(2), 150–158.
- Seitz S. M.; Dyer C. R. (1999). Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, **35**(2), 151–173.
- Szeliski R. (1993). Rapid octree construction from image sequences. *CVGIP: Image Understanding*, **58**(1), 23–32.

9

Moto e struttura

In questo capitolo ci interesseremo del problema dello *structure from motion*: date diverse viste di una scena prese da una fotocamera in movimento con parametri intrinseci noti e dato un insieme di punti corrispondenti, dobbiamo ricostruire il moto della fotocamera e la struttura della scena. In fotogrammetria questo è conosciuto come problema dell'*orientazione relativa*.

Nei capitoli precedenti abbiamo discusso il caso interamente calibrato, nel quale avevamo una coppia di fotocamere calibrate e la ricostruzione era possibile non appena si fosse in grado di stabilire le corrispondenze tra immagini. In questo capitolo consideriamo un'unica fotocamera in movimento; i parametri intrinseci sono noti, ma il moto della fotocamera è incognito (cioè mancano i parametri estrinseci).

9.1 Introduzione

Il problema di ricavare la struttura della scena con una fotocamera in movimento (*structure from motion*) è stato ampiamente studiato in passato [Huang e Netravali, 1994]. Gli approcci al problema si possono partizionare in metodi *differenziali* [Tian *e al.*, 1996, Soatto *e al.*, 1996, Soatto e Brockett, 1998] o *discreti*, a seconda che prendano in ingresso le velocità dei punti nell'immagine (il *campo di moto*) o un insieme di punti corrispondenti. In [Ma *e al.*, 1998] viene esaminata la relazione tra i due tipi di approccio. Tra i metodi discreti sono state impiegate approssimazioni ortografiche [Tomasi e Kanade, 1992] o para-prospettive [Poelman e Kanade, 1993] per la fotocamera. Uno dei metodi più interessanti, che usa il modello prospettico della fotocamera, fu proposto in [Longuet-Higgins, 1981]. Questa tecnica si basa sulla *matrice essenziale*,

che descrive la *geometria epipolare* di due immagini prospettiche (con i parametri intrinseci noti).

L'equazione di Longuet-Higgins, che definisce la matrice essenziale, sarà trattata nel paragrafo 9.2. La matrice essenziale codifica il moto rigido della fotocamera e, infatti, un teorema di [Huang e Faugeras, 1989] ci permette di fattorizzarla in una matrice di rotazione ed in una di traslazione (paragrafo 9.2.1). Poiché i parametri intrinseci sono noti, ciò è equivalente alla completa conoscenza delle MPP delle fotocamere. La struttura (cioè la distanza dei punti dalla fotocamera) si trova facilmente con la triangolazione. Da notare che la componente traslazionale dello spostamento può essere calcolata solo a meno di un fattore di scala, perché è impossibile determinare se il moto misurato nell'immagine è causato da un oggetto vicino che si sposta lentamente o da un oggetto distante che si muove rapidamente. Questo fatto è conosciuto come ambiguità profondità-velocità o *depth-speed ambiguity*; il perché del nome sarà chiarito nel capitolo 10. Nella paragrafo 9.2.2 ci occuperemo del calcolo della matrice essenziale, e illustreremo un semplice metodo lineare, chiamato *algoritmo degli otto punti* [Longuet-Higgins, 1981, Hartley, 1992].

9.2 Matrice essenziale

Supponiamo di avere una fotocamera, con parametri intrinseci noti, che si sta muovendo in un ambiente statico seguendo una traiettoria sconosciuta. Consideriamo due immagini prese dalla fotocamera in due istanti di tempo differenti ed assumiamo che siano dati un certo numero di punti corrispondenti tra le due immagini, in *coordinate normalizzate*. Siano P e P' le MPP delle fotocamere che corrispondono ai due istanti di tempo e $\mathbf{p} = K^{-1}\mathbf{m}$, $\mathbf{p}' = K'^{-1}\mathbf{m}'$ le coordinate normalizzate dei due punti immagine corrispondenti.

Lavorando in coordinate normalizzate e prendendo il sistema di riferimento della prima fotocamera come riferimento mondo, possiamo scrivere le seguenti due MPP:

$$P = [I|\mathbf{0}] \quad \text{e} \quad P' = [I|\mathbf{0}]G = [R|\mathbf{t}] \quad (9.1)$$

Sostituendo queste due particolari MPP nell'equazione di Longuet-Higgins (6.13) si ottiene la forma bilineare che lega punti coniugati in coordinate normalizzate:

$$\mathbf{p}'^\top [\mathbf{t}]_\times R \mathbf{p} = 0. \quad (9.2)$$

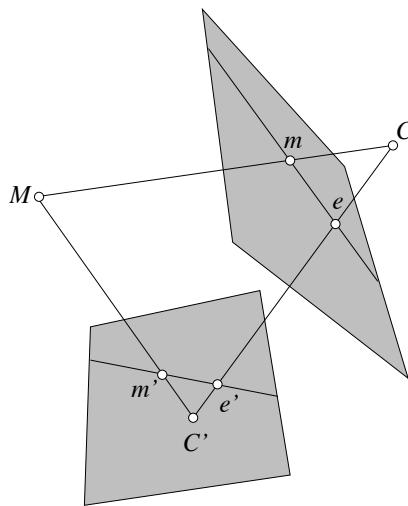


Fig. 9.1. Geometria epipolare.

La matrice

$$E \triangleq [\mathbf{t}]_{\times} R \quad (9.3)$$

che contiene i coefficienti della forma si chiama **matrice essenziale**. La matrice dipende da tre parametri per la rotazione e da *due* parametri per la traslazione. Infatti la (9.2) è omogenea rispetto a \$\mathbf{t}\$, ovvero il modulo del vettore non conta. Questo riflette l'ambiguità profondità-velocità, cioè il fatto che non possiamo ricavare la scala assoluta della scena senza un parametro extra, come la conoscenza della distanza tra due punti. Perciò una matrice essenziale ha solo cinque gradi di libertà, che tengono conto della rotazione (tre parametri) e traslazione a meno di un fattore di scala (due parametri).

In termini di vincoli, possiamo osservare che la matrice essenziale è definita a meno di un fattore di scala ed è singolare, poiché $\det[\mathbf{t}]_{\times} = 0$. Per arrivare ai cinque gradi di libertà ottenuti ragionando sulla parametrizzazione bisogna poter esibire altri *due* vincoli. Il teorema che vedremo nel prossimo paragrafo ci dice che questi due vincoli sono l'eguaglianza dei due valori singolari non nulli di \$E\$ (si ottiene un polinomio negli elementi di \$E\$, che sortisce due vincoli indipendenti).

9.2.1 Fattorizzazione della matrice essenziale

Supponiamo che sia data la matrice essenziale. Il seguente teorema dovuto a Huang e Faugeras [1989] caratterizza la matrice essenziale e ci permette di fattorizzarla in rotazione e traslazione.

Lemma 9.1 *Data una matrice di rotazione R e due matrici ortogonali U e V , allora $\det(UV^\top)URV^\top$ è una matrice di rotazione (ovvero ha determinante positivo).*

Teorema 9.2 *Una matrice reale E 3×3 può essere fattorizzata come prodotto di una matrice non nulla antisimmetrica e di una matrice di rotazione se e soltanto se E ha due valori singolari uguali non nulli ed un valore singolare uguale a zero.*

Dim. Sia $E = SR$ dove R è una matrice di rotazione e S è antisimmetrica. Sia $S = [\mathbf{t}]_\times$ con $\|\mathbf{t}\| = 1$ (con nessuna perdita di generalità, dato che E è definita a meno di un fattore di scala). Sia U la matrice di rotazione t.c. $U\mathbf{t} = [0, 0, 1]^\top \triangleq \mathbf{a}$, quindi $S = [\mathbf{t}]_\times = [U^\top \mathbf{a}]_\times$. In virtù della proprietà 1.48 possiamo scrivere:

$$S = [U^\top \mathbf{a}]_\times = U^\top [\mathbf{a}]_\times U.$$

Consideriamo ora la matrice EE^\top :

$$EE^\top = SRR^\top S^\top = SS^\top = U^\top [\mathbf{a}]_\times U U^\top [\mathbf{a}]_\times^\top U = U^\top \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} U.$$

Gli elementi della matrice diagonale sono gli autovalori di EE^\top , cioè i quadrati dei valori singolari di E . Questo dimostra un'implicazione.

Diamo ora una dimostrazione costruttiva dell'implicazione inversa.

Sia $E = UDV^\top$ la SVD di E , con $D = \text{diag}(1, 1, 0)$ (senza perdita di generalità) e U e V ortogonali. L'osservazione chiave è che:

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \triangleq S'R'$$

dove S' è antisimmetrica e R' una matrice di rotazione. Quindi

$$\begin{aligned} E = UDV^\top &= US'R'V^\top = (US'U^\top)(UR'V^\top) = \\ &= \det(UV^\top)(US'U^\top)\det(UR'V^\top). \end{aligned}$$

Prendendo $S = \det(UV^\top)US'U^\top$ e $R = \det(UV^\top)UR'V^\top$, la fattorizzazione cercata è: $E = SR$. Infatti, $US'U^\top$ è antisimmetrica (si veda l'esercizio 4) e la matrice $\det(UV^\top)UR'V^\top$ è ortogonale con determinante positivo (grazie al termine $\det(UV^\top)$) quindi è di rotazione.

□

Questa fattorizzazione non è unica. Data l'ambiguità del segno di E , possiamo cambiare il segno di D sia cambiando il segno di S' che trasponendo R' (dato che $S'R'^\top = -D$). In totale, dunque, abbiamo quattro possibili fattorizzazioni date da:

$$S = U(\pm S')U^\top \quad (9.4)$$

$$R = \det(UV^\top)UR'V^\top \text{ oppure } R = \det(UV^\top)UR'^\top V^\top \quad (9.5)$$

dove

$$S' \triangleq \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad R' \triangleq \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (9.6)$$

Come osservato da Longuet-Higgins, la scelta tra i quattro spostamenti è determinata dalla richiesta che i punti 3D, la cui posizione può essere calcolata costruendo le MPP e triangolando, debbano giacere davanti ad entrambe le fotocamere, cioè la loro terza coordinata deve essere positiva.

La funzione MATLAB `sr` applica il teorema precedente per la fattorizzazione della matrice essenziale.

9.2.2 Calcolo della matrice essenziale

In questo paragrafo tratteremo il problema della stima di E attraverso le corrispondenze di punti.

Dato un insieme di corrispondenze di punti (sufficientemente grande) $\{(\mathbf{p}_i, \mathbf{p}'_i) \mid i = 1, \dots, n\}$, in coordinate normalizzate, si vuole determinare la matrice essenziale E che collega i punti nella relazione bilineare:

$$\mathbf{p}'_i{}^\top E \mathbf{p}_i = 0. \quad (9.7)$$

La matrice incognita può essere agevolmente ricavata grazie al passaggio al vec ed all'impiego del prodotto di Kronecker (§. A1.12). Infatti si deriva:

$$\mathbf{p}'_i{}^\top E \mathbf{p}_i = 0 \iff \text{vec}(\mathbf{p}'_i{}^\top E \mathbf{p}_i) = 0 \iff (\mathbf{p}_i^\top \otimes \mathbf{p}'_i{}^\top) \text{vec}(E) = 0.$$

Quindi ogni corrispondenza di punti genera un'equazione omogenea lineare nei nove elementi incogniti della matrice E (letta per colonne). Da n punti corrispondenti otteniamo un sistema lineare di n equazioni:

$$\underbrace{\begin{bmatrix} \mathbf{p}_1^\top \otimes \mathbf{p}'_1^\top \\ \mathbf{p}_2^\top \otimes \mathbf{p}'_2^\top \\ \vdots \\ \mathbf{p}_n^\top \otimes \mathbf{p}'_n^\top \end{bmatrix}}_{U_n} \text{vec}(E) = 0 \quad (9.8)$$

La soluzione al sistema lineare omogeneo è il nucleo di U_n . Con $n = 8$ il nucleo della matrice ha dimensione uno, quindi la soluzione è determinata a meno di una costante moltiplicativa (fattore di scala). Perciò questo metodo viene chiamato **algoritmo degli otto punti**[†], anche se si tratta in realtà di una variante del DLT (§ 4.2).

Nella pratica, sono disponibili più di otto corrispondenze di punti e possiamo ottenere gli elementi di E risolvendo un problema lineare di minimi quadrati. La soluzione è l'autovettore unitario che corrisponde al minimo autovalore di $U_n^\top U_n$, che può essere calcolato con la SVD di U_n .

Si noti che la matrice E trovata risolvendo questo insieme di equazioni lineari non soddisferà, in generale, i requisiti del teorema 9.2, ovvero non avrà due valori singolari uguali e un valore singolare pari a zero. Questo si può forzare a posteriori sostituendo E con \hat{E} , la matrice più vicina, in norma di Frobenius, che soddisfa i requisiti. Sia E una matrice 3×3 e $E = UDV^\top$ la sua SVD con $D = \text{diag}(r, s, t)$ e $r \geq s \geq t$. Si può dimostrare che $\hat{E} = U\hat{D}V^\top$ dove $\hat{D} = \text{diag}(\frac{r+s}{2}, \frac{r+s}{2}, 0)$.

Sebbene l'algoritmo lineare che abbiamo descritto necessiti di almeno otto punti per calcolare E , poiché la matrice dipende da solo cinque parametri, è possibile, in linea di principio, calcolarla con cinque corrispondenze più i vincoli polinomiali forniti dal teorema (9.2). Infatti, [Faugeras e Maybank, 1990] hanno dimostrato che la cosa è fattibile e che esistono dieci soluzioni distinte. Di recente algoritmi per il calcolo di E con cinque punti sono stati proposti da [Nister, 2003] e da [Li e Hartley, 2006].

In sommario, la procedura per il calcolo della struttura delineata in questo capitolo è illustrata nell'algoritmo 9.1.

[†] Gli otto punti devono essere in posizione generale, ovvero vanno escluse le configurazioni degeneri evidenziate da [Faugeras e Maybank, 1990]

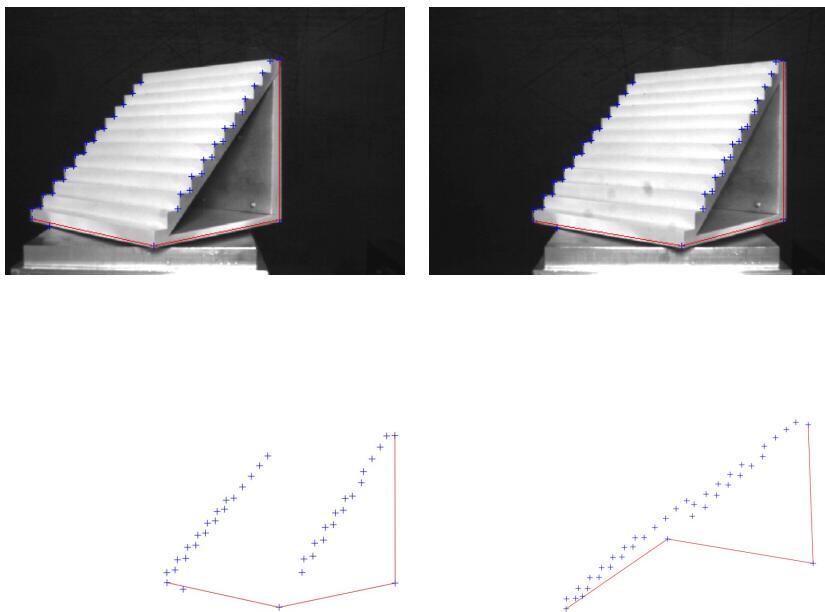


Fig. 9.2. Due immagini con indicati i punti salienti messi in corrispondenza (riga in alto). Due viste dell'oggetto ricostruito (riga in basso). Immagini gentilmente concesse da F. Isgrò.

Algoritmo 9.1 STRUTTURA E MOTO

Input: Punti corrispondenti nelle immagini, parametri intriseci K

Output: Struttura (coordinate 3D dei punti dati, M_i) e moto (R, \mathbf{t})

- (i) Calcolo punti corrispondenti $\mathbf{p}_i, \mathbf{p}'_i$;
 - (ii) Calcolo E con algoritmo otto-punti;
 - (iii) Fattorizza E in $S = [\mathbf{t}]_\times, R$;
 - (iv) Istanzia due MPP $P = [I|\mathbf{0}], P' = [R|\mathbf{t}]$ ed effettua la triangolazione dei punti dati, ottenendo M_i).
-

La figura 9.2 mostra un tipico risultato di ricostruzione di moto e struttura da due viste calibrate.

9.3 Estensione a molte viste

Consideriamo per semplicità il caso di tre viste, che si generalizza facilmente al caso di $N > 3$ fotocamere.

Se si applica l'algoritmo 9.1 alle coppie 1-2, 1-3 e 2-3 si ottengono tre moti rigidi $(R_{12}, \hat{\mathbf{t}}_{12})$, $(R_{13}, \hat{\mathbf{t}}_{13})$ e $(R_{23}, \hat{\mathbf{t}}_{23})$ nei quali ciascuna traslazione è nota solo a meno di un fattore di scala (per questo ne consideriamo solo il versore, denotato da $\hat{\cdot}$).

Per recuperare il corretto rapporto tra le norme delle traslazioni, ricordiamo che i moti rigidi devono soddisfare la seguente regola compozizionale:

$$\mathbf{t}_{13} = R_{23}\mathbf{t}_{12} + \mathbf{t}_{23} \quad (9.9)$$

che si può riscrivere come

$$\hat{\mathbf{t}}_{13} = \mu_1 R_{23}\hat{\mathbf{t}}_{12} + \mu_2 \hat{\mathbf{t}}_{23} \quad (9.10)$$

dove $\mu_1 = \|\mathbf{t}_{12}\|/\|\mathbf{t}_{13}\|$ e $\mu_2 = \|\mathbf{t}_{23}\|/\|\mathbf{t}_{13}\|$. L'equazione si può risolvere rispetto alle incognite μ_1, μ_2 come indicato nella proposizione 1.51, ottenendo:

$$\frac{\|\mathbf{t}_{12}\|}{\|\mathbf{t}_{13}\|} = \mu_1 = \frac{(\hat{\mathbf{t}}_{13} \times \hat{\mathbf{t}}_{23})^\top (R_{23}\hat{\mathbf{t}}_{12} \times \hat{\mathbf{t}}_{23})}{\|R_{23}\hat{\mathbf{t}}_{12} \times \hat{\mathbf{t}}_{23}\|^2}. \quad (9.11)$$

Questo consente – una volta fissata arbitrariamente la norma di \mathbf{t}_{12} – di istanziare tre MPP $P_1 = [I|\mathbf{0}]$, $P_2 = [R_{12}|\mathbf{t}_{12}]$ e $P_3 = [R_{13}|\mathbf{t}_{13}]$ tra loro coerenti e procedere con la triangolazione.

Si noti che poiché si possono ricavare solo rapporti tra le norme, un fattore di scala globale rimane indeterminato, come nel caso di due viste.

L'algoritmo è implementato nella funzione `erec`.

Bundle adjustment. Quando sono presenti molte viste il metodo presentato soffre di una accumulazione degli errori, che porta ad una deriva del risultato. È buona norma, in questi casi, raffinare il risultato con una minimizzazione dell'errore nell'immagine. Poiché la procedura descritta in questo capitolo mira a recuperare sia la struttura sia il moto (ovvero i parametri estrinseci delle fotocamere), la minimizzazione viene effettuata nei confronti di entrambi, e prende il nome di *bundle adjustment*. Si cerca dunque di spostare sia le N fotocamere che gli n punti 3D affinché la somma delle distanze al quadrato tra il punto j -esimo riproiettato tramite la fotocamera i -esima $P_i \mathbf{M}^j$ ed il punto misurato \mathbf{m}_i^j sia più piccola possibile (in ogni immagine dove il punto appare):

$$\min_{R_i, \mathbf{t}_i, \mathbf{M}^j} \sum_{i=1}^N \sum_{j=1}^n d(K_i[R_i|\mathbf{t}_i]\mathbf{M}^j, \mathbf{m}_i^j)^2 \quad (9.12)$$

dove $d(\cdot)$ è la distanza nel piano cartesiano (quindi bisogna convertire in coordinate cartesiane, come si faceva esplicitamente nella (4.40)).

Nell'espressione che viene minimizzata sarà necessario parametrizzare correttamente R_i , in modo che compaiano solo tre incognite (per esempio i tre angoli di Eulero, § A3.1) invece che tutti i nove elementi della matrice.

Tipicamente il problema assume dimensioni ragguardevoli, ed allora per affrontarlo si ricorre ad una strategia in due passi alternati: prima si tengono fermi i punti e si risolve rispetto alle fotocamere, come in un problema di calibrazione, poi si fissano le fotocamere e si calcolano i punti 3D, come in un problema di triangolazione. Si itera fino a convergenza.

Si veda la funzione `bundleadj`.

9.4 Estrazione di punti salienti

Il calcolo della matrice essenziale assume che si stabiliscano le corrispondenze tra un certo numero di punti di due immagini. Alcuni punti vanno meglio di altri per essere messi in corrispondenza. Vedremo in questo paragrafo un operatore per l'estrazione di *punti salienti* o *feature points* basato sull'analisi locale dell'autocorrelazione di una finestra.

9.4.1 Metodo di Harris e Stephens

Spesso questo operatore viene indicato come rilevatore di angoli (*corner detector*), in quanto i punti salienti che estrae spesso coincidono con angoli, ovvero intersezioni di due spigoli o *edge*. Esso in realtà rileva punti caratteristici adatti per essere messi in corrispondenza. Chiameremo comunque “angoli” questi punti salienti, definendo come “angolo” un punto caratterizzato da una discontinuità della intensità lungo due direzioni. Questa può essere rilevata misurando quanta variazione si ottiene traslando una finestra centrata sul punto in un intorno della posizione originale. Se in una direzione è possibile traslare la zona senza apprezzabili variazioni, si è in presenza di un *edge* (lungo la direzione data). Se, viceversa, in ogni direzione la variazione cresce rapidamente si ha un angolo. Calcoliamo allora la variazione, in termini di somma delle differenze al quadrato (SSD), che si ottiene traslando della quantità \mathbf{h} una zona W centrata nel punto $\mathbf{x} = (u, v)$ dell'immagine I :

$$e_{\mathbf{h}}(\mathbf{x}) = \sum_{\mathbf{d} \in W} [I(\mathbf{x} + \mathbf{d}) - I(\mathbf{x} + \mathbf{d} + \mathbf{h})]^2 \quad (9.13)$$

Mediante lo sviluppo in serie di Taylor troncato [Giusti, 1989], si ottiene:

$$\begin{aligned}
 e_h(\mathbf{x}) &= \sum_{\mathbf{d} \in W} [\nabla I(\mathbf{x} + \mathbf{d})^\top \mathbf{h}]^2 \\
 &= \sum_{\mathbf{d} \in W} \mathbf{h}^\top (\nabla I(\mathbf{x} + \mathbf{d})) (\nabla I(\mathbf{x} + \mathbf{d}))^\top \mathbf{h} \\
 &= \sum_{\mathbf{d} \in W} \mathbf{h}^\top \begin{bmatrix} I_u^2 & I_u I_v \\ I_u I_v & I_v^2 \end{bmatrix} \mathbf{h} \\
 &= \mathbf{h}^\top \begin{bmatrix} \sum_{\mathbf{d} \in W} I_u^2 & \sum_{\mathbf{d} \in W} I_u I_v \\ \sum_{\mathbf{d} \in W} I_u I_v & \sum_{\mathbf{d} \in W} I_v^2 \end{bmatrix} \mathbf{h}
 \end{aligned} \tag{9.14}$$

dove si è posto $\nabla I(\mathbf{x} + \mathbf{d}) = [I_u \ I_v]^\top$. Introducendo una funzione peso $w(\cdot)$ che vale 1 sugli elementi di un dominio rettangolare e 0 all'esterno, possiamo scrivere:

$$e_h(\mathbf{x}) = \mathbf{h}^\top \begin{bmatrix} \sum_{\mathbf{d}} I_u^2 w(\mathbf{d}) & \sum_{\mathbf{d}} I_u I_v w(\mathbf{d}) \\ \sum_{\mathbf{d}} I_u I_v w(\mathbf{d}) & \sum_{\mathbf{d}} I_v^2 w(\mathbf{d}) \end{bmatrix} \mathbf{h} \tag{9.15}$$

Invece di prendere una finestra rettangolare, per rendere simmetrico e più regolare l'operatore è conveniente considerare una finestra pesata gaussiana:

$$w(\mathbf{d}) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{\|\mathbf{d}\|^2}{2\sigma^2}} \tag{9.16}$$

Dunque, la variazione (SSD) filtrata in un intorno di \mathbf{x} , per lo spostamento \mathbf{h} è data da

$$\hat{e}_h(\mathbf{x}) = \mathbf{h}^\top \widehat{\mathbf{C}} \mathbf{h} \tag{9.17}$$

dove \mathbf{C} è la matrice simmetrica 2×2

$$\widehat{\mathbf{C}} = \begin{bmatrix} \widehat{I}_u^2 & \widehat{I}_u \widehat{I}_v \\ \widehat{I}_u \widehat{I}_v & \widehat{I}_v^2 \end{bmatrix} \tag{9.18}$$

mentre \widehat{I} indica il risultato della convoluzione di I con la finestra gaus-

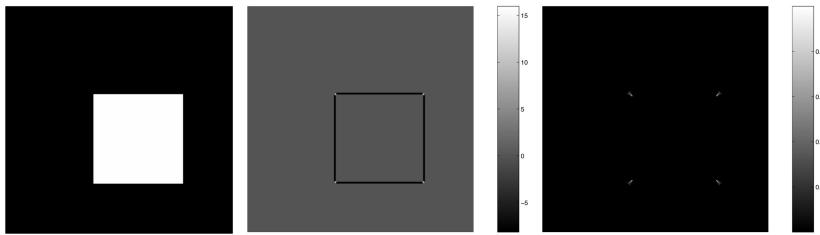


Fig. 9.3. Da sinistra: immagine di prova, risposta dell'operatore di Harris e Stephens (notare che assume valori negativi), risposta dell'operatore di Noble.

siana w . La deviazione standard della gaussiana fissa la scala spaziale alla quale si vogliono determinare gli angoli.

La proposizione 1.33 ci assicura che, preso \mathbf{h} unitario,

$$\lambda_1 < e_{\mathbf{h}}(\mathbf{x}) < \lambda_2 \quad (9.19)$$

dove λ_1 e λ_2 sono rispettivamente il minimo ed il massimo autovalore di $\widehat{\mathcal{C}}$. Quindi, se consideriamo tutte le possibili direzioni \mathbf{h} , il massimo della variazione che otteniamo è λ_2 , mentre il minimo è λ_1 . Possiamo quindi classificare la struttura dell'immagine attorno a ciascun pixel analizzando gli autovalori λ_1 e λ_2 . I casi che possono presentarsi sono i seguenti:

- Nessuna struttura: $\lambda_1 \approx \lambda_2 \approx 0$
- Edge: $\lambda_1 \approx 0$ (direzione dell'*edge*), $\lambda_2 \gg 0$
- Angoli (*corner*): λ_1 e λ_2 entrambi $\gg 0$

L'algoritmo proposto da [Harris e Stephens, 1988] non calcola esplicitamente gli autovalori, ma la quantità:

$$r = \det(\widehat{\mathcal{C}}) - k \operatorname{tr}^2(\widehat{\mathcal{C}}) \quad (9.20)$$

e considera come angoli i punti in cui il valore di r supera una certa soglia. La costante k viene posta a 0.04 (come suggerito da Harris e Stephens). Si noti che

$$\operatorname{tr}(\widehat{\mathcal{C}}) = \lambda_1 + \lambda_2 = \widehat{I}_u^2 + \widehat{I}_v^2 \quad (9.21)$$

e

$$\det(\widehat{\mathcal{C}}) = \lambda_1 \lambda_2 = \widehat{I}_u^2 \widehat{I}_v^2 - \widehat{I}_u \widehat{I}_v^2 \quad (9.22)$$

L'operatore di Harris e Stephens risponde con valori positivi agli angoli, negativi agli *edge* e con valori prossimi allo zero in regioni uniformi (figura 9.3).

Partendo dalle medesime considerazioni sugli autovalori, Noble [1988]

propone un operatore basato anch'esso sulla traccia e sul determinante della matrice \widehat{C} ma privo di parametri. Per ogni punto dell'immagine viene calcolato il seguente rapporto

$$n = \frac{\det(\widehat{C})}{\text{tr}(\widehat{C})} \quad (9.23)$$

il quale in corrispondenza degli angoli fornisce valori elevati (figura 9.3). Se si vuole un operatore adimensionale, bisogna usare:

$$n' = \frac{\det(\widehat{C})}{\text{tr}^2(\widehat{C})}. \quad (9.24)$$

L'algoritmo 9.2 riassume quanto detto sinora e la figura 9.4 ne mostra il risultato in un caso reale.

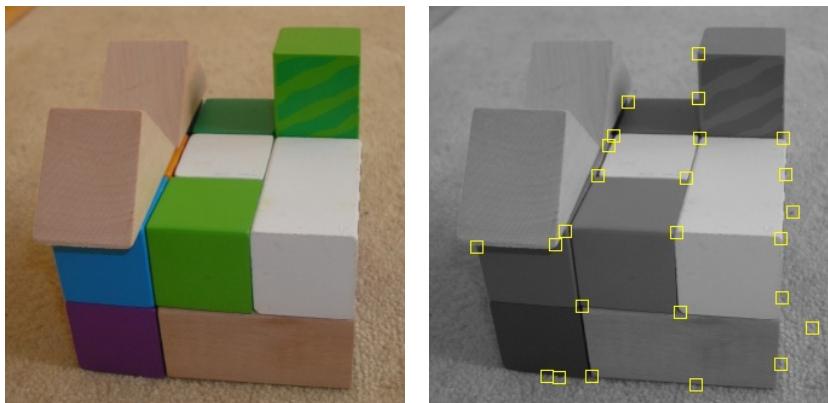


Fig. 9.4. Da sinistra: immagine di prova, punti estratti dall'operatore di Harris e Stephens con l'eliminazione dei non-massimi e la sogliatura.

9.5 Corrispondenze di punti salienti

Il calcolo delle corrispondenze di punti salienti rientra nei metodi di calcolo delle corrispondenze *feature based*, cui abbiamo accennato nel capitolo 7. Una volta estratti i punti salienti come descritto nel paragrafo precedente, si possono accoppiare, tipicamente tenendo conto della prossimità (posizione) e/o della similarità (per esempio con SSD). Nel caso in cui si debba processare una sequenza video l'accoppiamento viene reiterato per ciascuna coppia di fotogrammi: si parla anche di **tracciamento** (*tracking*) dei punti salienti (figura 9.5). L'operazione non è

Algoritmo 9.2 PUNTI SALIENTI**Input:** Immagine**Output:** Coordinate dei punti salienti di Harris-Stephens

- (i) si calcolano le derivate dell'immagine I_u e I_v ;
- (ii) si filtrano con un nucleo gaussiano le immagini I_u^2 , I_v^2 ed $I_u I_v$, ottenendo $\widehat{I_u^2}$, $\widehat{I_v^2}$ e $\widehat{I_u I_v}$;
- (iii) si calcola in ciascun punto la quantità R (o N) con la (9.20);
- (iv) si effettua la soppressione dei non-massimi: si fa scorrere una finestra di dimensioni fissate sull'immagine e si forza il pixel centrale a zero se non è il massimo dell'intorno, altrimenti si lascia inalterato;
- (v) si effettua la sogliatura con un valore prefissato, oppure specificato come frazione del valore massimo (preferibile);
- (vi) i punti risultanti sono angoli (o punti di interesse).

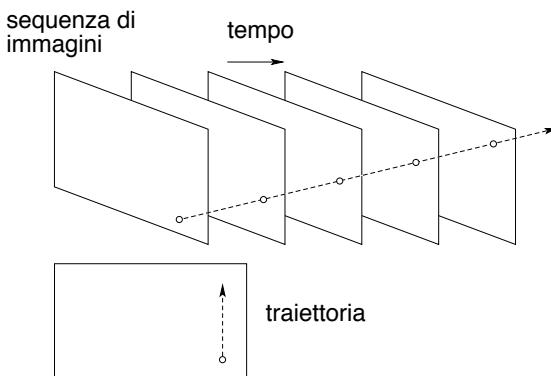


Fig. 9.5. Tracciamento di punti: il percorso di un punto saliente nel volume spazio-temporale prende il nome di traccia.

facile: i punti possono scomparire e riapparire a causa di occlusioni o per il fallimento del rilevatore.

Nel paragrafo 10.3 verrà illustrato il metodo di Kanade-Lucas-Tomasi (KLT): si tratta di un metodo di accoppiamento basato essenzialmente sulla similarità (si assume che lo spostamento sia piccolo). Nel caso in cui invece lo spostamento tra due immagini consecutive sia rilevante, per accoppiare i punti salienti, sfruttiamo la coerenza spazio-temporale per predire dove il punto dovrebbe trovarsi nel fotogramma successivo.

Lo schema classico si basa sulla iterazione di tre fasi: estrazione, pre-

dizione e associazione e tipicamente comprende un **filtro di Kalman**. Il filtro di Kalman (§ A3.2.2) usa un modello del moto per prevedere dove il punto sarà al prossimo istante di tempo, date le sue posizioni precedenti. Inoltre, il filtro mantiene una stima dell'incertezza nella previsione, consentendo così la definizione di una finestra di ricerca. Il rilevamento dei punti salienti viene effettuato all'interno della finestra di ricerca, uno di questi viene associato al punto tracciato ed il filtro aggiorna la stima della posizione usando la posizione del punto trovato e la sua incertezza.

L'operazione di associazione, inoltre, può presentare ambiguità (figura 9.6). Quando si trova più di un punto nella finestra di ricerca, servono filtri più sofisticati. Un filtro adatto, sviluppato dalla comunità del radar ma che trova applicazione anche nel tracciamento ottico è il *Joint Probabilistic Data Association Filter*.

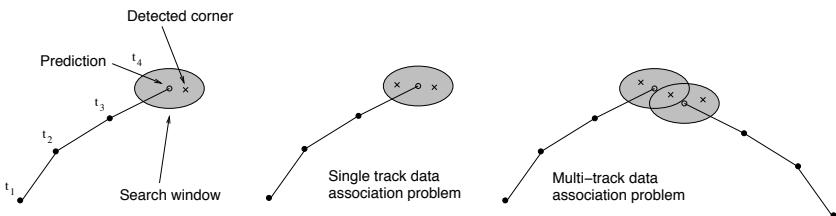


Fig. 9.6. Tracciamento e problema di associazione dei dati.

Si veda [Trucco e Verri, 1998] (pg. 199-203) per una discussione sul tracciamento con Kalman Filter, e [Bar-Shalom e Fortmann, 1988] per il problema di *data association*.

Esercizi ed approfondimenti

9.1 Nel caso studiato in §. 9.2 l'epipolo ha coordinate:

$$\mathbf{e}' = P'\mathbf{C} = [R|\mathbf{t}] \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{t}. \quad (\text{E9.1})$$

Si rifletta sull'analogia con il FOE (definito più avanti, nel capitolo 10).

- 9.2 Implementare in MATLAB un estrattore di angoli seguendo lo schema dell'algoritmo 9.2. Si veda la funzione `imhs`.
- 9.3 Che relazione c'è tra la matrice fondamentale e la matrice essenziale?

- 9.4 Dimostrare che se U è ortogonale ed S' è antisimmetrica, allora anche $US'U^\top$ è antisimmetrica.

Bibliografia

- Bar-Shalom Y.; Fortmann T. E. (1988). *Tracking and data Association*. Academic Press.
- Faugeras O.; Maybank S. (1990). Motion from point matches: multiplicity of solutions. *International Journal of Computer Vision*, **4**(3), 225–246.
- Giusti E. (1989). *Analisi Matematica 2*. Bollati Boringhieri.
- Harris C.; Stephens M. (1988). A combined corner and edge detector. *Proceedings of the 4th Alvey Vision Conference*, pp. 189–192.
- Hartley R. I. (1992). Estimation of relative camera position for uncalibrated cameras. In *Proceedings of the European Conference on Computer Vision*, pp. 579–587, Santa Margherita L.
- Huang T.; Faugeras O. (1989). Some properties of the E matrix in two-view motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**(12), 1310–1312.
- Huang T. S.; Netravali A. N. (1994). Motion and structure from feature correspondences: A review. *Proceedings of IEEE*, **82**(2), 252–267.
- Li H.; Hartley R. (2006). Five-point motion estimation made easy. In *Proceedings of the International Conference on Pattern Recognition*, pp. 630–633, Washington, DC, USA. IEEE Computer Society.
- Longuet-Higgins H. C. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, **293**(10), 133–135.
- Ma Y.; Košecká J.; Sastry S. (1998). Motion recovery from image sequences: Discrete viewpoint vs. differential viewpoint. In *Proceedings of the European Conference on Computer Vision*, pp. 337–353.
- Nister D. (2003). An efficient solution to the five-point relative pose problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 02, p. 195, Los Alamitos, CA, USA. IEEE Computer Society.
- Noble J. (1988). Finding corners. *Image and Vision Computing*, **6**, 121–128.
- Poelman C. J.; Kanade T. (1993). A paraperspective factorization method for shape and motion recovery. Technical Report CMU-CS-93-219, Carnegie Mellon University, Pittsburg, PA.
- Soatto S.; Brockett R. (1998). Optimal structure from motion: Local

- ambiguities and global estimates. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 282–288, Santa Barbara, CA.
- Soatto S.; Frezza R.; Perona P. (1996). Motion estimation via dynamic vision. *IEEE Transactions on Automatic Control*, **41**(3), 393–413.
- Tian T.; Tomasi C.; Heeger D. (1996). Comparison of approaches to ego-motion computation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 315–320, San Francisco, CA.
- Tomasi C.; Kanade T. (1992). Shape and motion from image streams under orthography – a factorization method. *International Journal of Computer Vision*, **9**(2), 137–154.
- Trucco E.; Verri A. (1998). *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall.

10

Flusso ottico

Assumiamo che la fotocamera in movimento (con parametri intrinseci noti) inquadri una scena statica. Dal moto apparente dei punti nelle immagini è possibile ricavare informazioni sulla loro profondità e sul moto della fotocamera stessa. Vi sono essenzialmente due approcci a questo problema: l'approccio discreto, basato sul recupero della geometria epipolare a partire dalla posizione di punti corrispondenti e l'approccio differenziale, basato sull'analisi del campo di moto (velocità dei punti).

Avendo già trattato nel capitolo 9 l'approccio discreto, ci concentriremo in questo capitolo sull'approccio differenziale e sulle tecniche per la misura del flusso ottico.

10.1 Il campo di moto

Ipotesi[†]:

- (i) coordinate normalizzate $\mathbf{p} = (x, y, 1)^\top$ (vuol dire che i parametri intrinseci della fotocamera sono noti).
- (ii) riferimento mondo coincidente con il riferimento standard della fotocamera.
- (iii) il moto relativo fotocamera-scena è rigido, ovvero la fotocamera è in movimento e la scena è stazionaria.

Supponiamo che la fotocamera si muova con velocità (istantanea) traslazionale \mathbf{V} e velocità angolare $\boldsymbol{\Omega}$ rispetto ad un sistema di riferimento *assoluto* e consideriamo un punto $\mathbf{M} = (X, Y, Z)^\top$ stazionario (risp. al sistema assoluto).

[†] Attenzione: in questo capitolo, per non appesantire troppo la notazione, le coordinate cartesiane del punto M sono denotate da \mathbf{M} , senza la $\tilde{\cdot}$. Inoltre è stato omesso l'indice temporale. Per esempio avrei dovuto scrivere $\mathbf{p}(t)$.

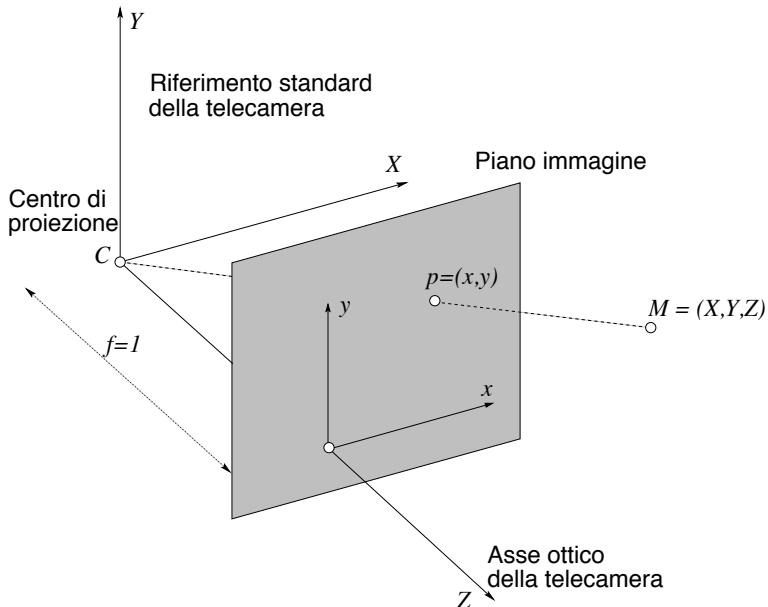


Fig. 10.1. Sistema di riferimento.

La velocità di \mathbf{M} rispetto al riferimento mobile (quello della fotocamera) si calcola con:

$$\dot{\mathbf{M}} = -\mathbf{V} - \boldsymbol{\Omega} \times \mathbf{M}. \quad (10.1)$$

Il segno - compare perchè la scena si muove rispetto alla fotocamera con velocità $-\mathbf{V}$ e $-\boldsymbol{\Omega}$.

Il *motion field* o campo di moto $\dot{\mathbf{p}}$ è la proiezione di $\dot{\mathbf{M}}$ sull'immagine, ovvero è la velocità con cui si muove nell'immagine la proiezione \mathbf{p} di \mathbf{M} :

$$\mathbf{p} = (x, y, 1)^\top = (X/Z, Y/Z, 1)^\top = \frac{1}{Z}\mathbf{M}. \quad (10.2)$$

poichè misuriamo le coordinate 3D di \mathbf{M} nel riferimento standard della fotocamera e lavoriamo in coordinate normalizzate. Dunque, derivando:

$$\dot{\mathbf{p}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{Z\dot{X} - \dot{Z}X}{Z^2} \\ \frac{Z\dot{Y} - \dot{Z}Y}{Z^2} \\ 0 \end{bmatrix} = \frac{Z\dot{\mathbf{M}} - \dot{Z}\mathbf{M}}{Z^2} \quad (10.3)$$

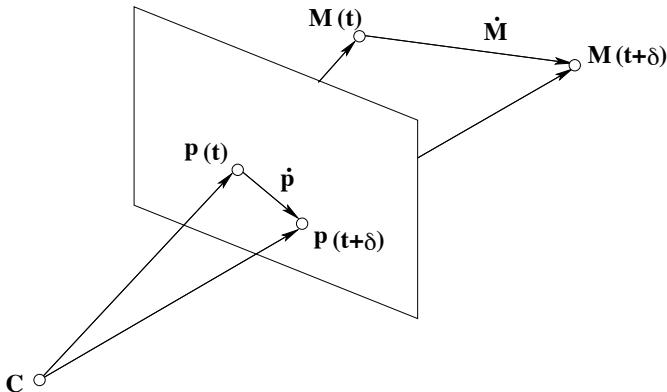


Fig. 10.2. La velocità di un punto sull'immagine è la proiezione della velocità nello spazio del corrispondente punto 3D.

Usando la (10.2) l'equazione precedente diventa:

$$\dot{\mathbf{p}} = \frac{1}{Z} (\dot{\mathbf{M}} - \dot{Z} \mathbf{p}) \quad (10.4)$$

Usando la (10.1) e scrivendo: $\dot{Z} = \dot{\mathbf{M}}^\top \hat{\mathbf{k}}$, dove $\hat{\mathbf{k}}$ indica il versore dell'asse Z (infatti \dot{Z} è la componente di $\dot{\mathbf{M}}$ lungo la direzione $\hat{\mathbf{k}}$), possiamo derivare dalla (10.3) (esercizio) l'equazione fondamentale del *motion field*:

$$\dot{\mathbf{p}} = -\frac{\mathbf{V}}{Z} + \frac{\mathbf{V}^\top \hat{\mathbf{k}}}{Z} \mathbf{p} - \boldsymbol{\Omega} \times \mathbf{p} + ((\boldsymbol{\Omega} \times \mathbf{p})^\top \hat{\mathbf{k}}) \mathbf{p} \quad (10.5)$$

Possiamo scomporre l'equazione in due parti, la prima contiene la coordinata Z e la velocità di traslazione \mathbf{V} , la seconda contiene la rotazione $\boldsymbol{\Omega}$. Notiamo due fatti importanti:

- (i) Z non compare mai assieme alla rotazione. Ovvero, se c'è solo rotazione ($\mathbf{V} = 0$), il campo di moto non dipende dalla struttura della scena.
- (ii) la velocità \mathbf{V} è sempre divisa per Z : è la cosiddetta *ambiguità profondità-velocità*. Ovvero, lo stesso campo di moto può essere causato da un oggetto vicino che si muove piano o da un oggetto lontano che si muove veloce.

Esplicitando le componenti, l'equazione del campo di moto diventa:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix} = \begin{bmatrix} -\frac{V_1}{Z} + \frac{V_3x}{Z} - \Omega_2 + \Omega_3y + (\Omega_1y - \Omega_2x)x \\ -\frac{V_2}{Z} + \frac{V_3y}{Z} - \Omega_3x + \Omega_1 + (\Omega_1y - \Omega_2x)y \\ 0 \end{bmatrix}. \quad (10.6)$$

Talvolta si trova l'equazione del campo di moto nella forma seguente:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} \mathbf{V} + \begin{bmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{bmatrix} \boldsymbol{\Omega}. \quad (10.7)$$

Ogni punto fornisce due vincoli sulle cinque componenti del moto (tre parametri per $\boldsymbol{\Omega}$ e due per \mathbf{V}) incognite, e sulla sua profondità. Dunque servono almeno cinque punti per ottenere una soluzione: essi infatti forniscono 10 equazioni in $5+5=10$ incognite. Si tratta di risolvere il seguente problema di minimizzazione (non lineare):

$$(\check{\mathbf{V}}, \check{\boldsymbol{\Omega}}, \{\check{Z}_i\}) = \arg \min \frac{1}{m} \sum_{i=1}^m \rho(\mathbf{r}) \quad (10.8)$$

dove $m > 5$ è il numero di punti disponibili, \mathbf{r} è il residuo

$$\mathbf{r} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} - \frac{1}{Z_i} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} \mathbf{V} + \begin{bmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{bmatrix} \boldsymbol{\Omega}, \quad (10.9)$$

e $\rho(\cdot)$ è una qualunque funzione convessa a simmetria radiale, chiamata funzione di penalità (*loss function*). Nel caso dei minimi quadrati, $\rho(\cdot) = \|\cdot\|_2$, ma nel caso di stimatori robusti (M-stimatori) si usano funzioni che crescono meno velocemente in periferia, per limitare l'influenza dei campioni periferici (si veda § A3.2).

10.1.1 Analisi del campo di moto

Supponiamo per ora di conoscere il campo di moto (vedremo nel prossimo paragrafo come calcolarne una approssimazione). Possiamo usarlo per calcolare $\mathbf{V}, \boldsymbol{\Omega}$ e la struttura della scena (Z). L'approccio consiste nel calcolare prima i parametri incogniti del moto e quindi ricostruire la struttura (a meno di un fattore di scala) risolvendo l'equazione del campo di moto per Z . Studieremo qui due casi semplificati, in cui una delle due velocità è nulla (Attenzione: grazie all'assunzione di moto rigido, tutti i punti della scena hanno le stesse velocità \mathbf{V} e $\boldsymbol{\Omega}$, mentre ciascuno è caratterizzato da una Z , in principio, diversa).

Solo traslazione ($\Omega = 0$)

$$\dot{\mathbf{p}} = -\frac{\mathbf{V}}{Z} + \frac{\mathbf{V}^\top \hat{\mathbf{k}}}{Z} \mathbf{p} \quad (10.10)$$

Per risolvere per \mathbf{V} devo eliminare Z . Lo faccio introducendo il prodotto esterno con \mathbf{p} e sfruttando il fatto che $\mathbf{p} \times \mathbf{p} = \mathbf{0}$:

$$\dot{\mathbf{p}} \times \mathbf{p} = -\frac{\mathbf{V} \times \mathbf{p}}{Z} \quad (10.11)$$

Questa equazione dice che $\dot{\mathbf{p}}, \mathbf{p}$ e \mathbf{V} sono coplanari, e dunque è equivalente a:

$$(\dot{\mathbf{p}} \times \mathbf{p})^\top \mathbf{V} = 0 \quad (10.12)$$

L'equazione è omogenea, quindi la soluzione è definita a meno di un fattore di scala (a causa della ambiguità profondità-velocità). Con due punti ottengo due quazioni lineari, sufficienti a risolvere per \mathbf{V} (a meno di un fattore di scala).

Solo rotazione ($\mathbf{V} = 0$)

$$\dot{\mathbf{p}} = -\boldsymbol{\Omega} \times \mathbf{p} + ((\boldsymbol{\Omega} \times \mathbf{p})^\top \hat{\mathbf{k}}) \mathbf{p} \quad (10.13)$$

con due punti ottengo quattro equazioni lineari, da cui ricavo $\boldsymbol{\Omega}$.

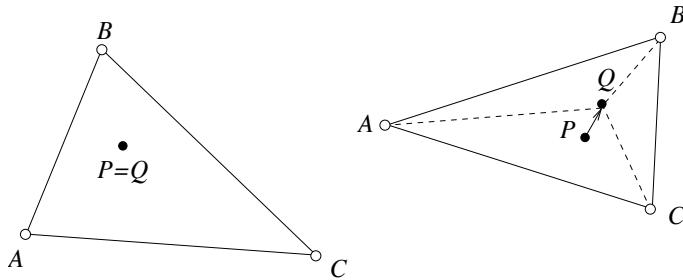
Motion parallax Nel caso di moto generale, le cose si fanno più complicate, devo riuscire a disaccoppiare la componente rotazionale da quella traslazionale. Questo si può fare se due punti in una certa immagine si trovano sovrapposti. Infatti in questo caso la differenza tra le loro velocità nell'immagine (moto relativo) non dipende dalla componente rotazionale (si annulla perché non dipende da Z). Formalmente, se (istantaneamente) $\mathbf{p}_1 = \mathbf{p}_2$, allora

$$\dot{\mathbf{p}}_1 - \dot{\mathbf{p}}_2 = (-\mathbf{V} + (\mathbf{V}^\top \hat{\mathbf{k}}) \mathbf{p}_1) \left(\frac{1}{Z_1} - \frac{1}{Z_2} \right) = \Delta \quad (10.14)$$

Considero Δ (chiamato *motion parallax*) come fosse un campo di moto traslazionale e risolvo per \mathbf{V} e poi per le profondità.

Se non ho due punti coincidenti (è una situazione fortuita) devo escogitare un trucco per “generare” punti coincidenti.

Considero quattro punti A,B,C e P. I primi tre definiscono un piano a cui il quarto, in generale, non appartiene. Posso immaginare un punto Q istantaneamente allineato con P nell'immagine al tempo t e giacente sul piano ABC. Quando passo al tempo $t + \Delta t$, inseguo i punti A,B,C,P (ne misuro la posizione) e calcolo (stimo) invece la nuova posizione di Q

Fig. 10.3. Calcolo del *motion parallax* affine.

a partire da quelle di A,B,C. Poiché Q giace sul piano definito da A,B,C, posso calcolarne la nuova posizione applicando ad esso la deformazione affine definita dallo spostamento di A,B,C (in realtà sappiamo che la trasformazione appropriata sarebbe una proiettività, ma se le profondità dei tre vertici non sono troppo diverse, l'affinità ne è una buona approssimazione). La posizione stimata di Q al tempo $t + \Delta t$, ci fornisce il *motion parallax* Δ cercato.

Fuoco di espansione Riprendiamo il caso di moto puramente traslazionale. Riscrivo la (10.10) per componenti:

$$\dot{x} = -\frac{V_1}{Z} + \frac{V_3 x}{Z} \quad \dot{y} = -\frac{V_2}{Z} + \frac{V_3 y}{Z} \quad (10.15)$$

Esiste un punto dove il campo di moto è istantanemente zero, e si ricava annullando \dot{x} e \dot{y} nella (10.15):

$$\left(x_0 = \frac{V_1}{V_3}, y_0 = \frac{V_2}{V_3} \right) \quad (10.16)$$

Eliminando V_1 e V_2 si ottiene:

$$\dot{x} = \frac{V_3(x - x_0)}{Z} \quad \dot{y} = \frac{V_3(y - y_0)}{Z} \quad (10.17)$$

dunque la direzione del campo di moto è radiale da (o verso) il punto $p_0 = [x_0, y_0]^\top$ che prende il nome di **fuoco di espansione (contrazione)**†. La posizione del fuoco di espansione nel piano immagine è semplicemente uguale al vettore \mathbf{V} interpretato come punto del piano in coordinate omogenee. Geometricamente è l'intersezione del piano immagine con il raggio ottico parallelo a \mathbf{V} . Il fuoco di espansione (FOE)

† Si prende la (10.16) come definizione del FOE, anche nel caso in cui sia presente una componente rotazionale. In questo caso, però, il campo di moto non si annulla nel FOE.

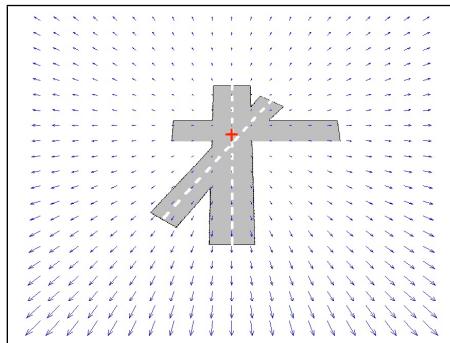


Fig. 10.4. Campo di moto radiale per una fotocamera che si muove verso un piano con $\mathbf{V} = (0, 5000, 100)$. La crocetta rossa rappresenta il FOE, che ha coordinate (0,50). Il moto è puramente traslazionale.

corrisponde al punto dove la fotocamera si sta dirigendo (*heading*), ovvero: i punti 3D che si proiettano nel FOE nell'immagine si trovano (istantaneamente) sul percorso della fotocamera.

Osservazione. Nel caso in cui $V_3 = 0$ il FOE è all'infinito ed il campo di moto è parallelo: è un caso particolare di campo radiale.

10.2 Il flusso ottico

Il flusso ottico è una approssimazione del campo di moto, corrispondente al moto osservabile nell'immagine. Il flusso ottico è legato al cambiamento dei livello di grigio dell'immagine, che vengono interpretati come manifestazione del campo di moto. Il campo di moto ha una definizione precisa, ma non può essere misurato (esempio dell'anfora che gira): non sempre il moto di oggetti nel mondo corrisponde ad un cambiamento dei livelli di grigio nella immagine. Quello che possiamo misurare nelle immagini è il flusso ottico, che indicheremo con \mathbf{v} , per distinguerlo dal campo di moto $\dot{\mathbf{p}}$.

Ipotesi: i punti dell'immagine si muovono mantenendo inalterato il livello di grigio:

$$I(\mathbf{x}, t) = I(\mathbf{x} + \mathbf{v}\Delta t, t + \Delta t) \quad (10.18)$$

Questa equazione non è verificata sempre nella realtà, per

- deviazioni dal modello lambertiano (cambia angolo con sorgente luminosa);

- occlusioni ed in generale effetti prospettici (appaiono e scompaiono punti).

Sviluppando I (funzione scalare di tre variabili) in serie di Taylor troncata al primo ordine attorno a (\mathbf{x}, t) , otteniamo: (trascurando il resto)

$$I(\mathbf{x} + \mathbf{v}\Delta t, t + \Delta t) = I(\mathbf{x}, t) + \text{grad}I(\mathbf{x}, t)^\top \begin{bmatrix} \mathbf{v}\Delta t \\ \Delta t \end{bmatrix} \quad (10.19)$$

dove $\text{grad}I$ indica il gradiente di I , cioè il vettore (colonna) le cui componenti sono le derivate parziali di I . Separando la parte spaziale da quella temporale otteniamo

$$I(\mathbf{x} + \mathbf{v}\Delta t, t + \Delta t) = I(\mathbf{x}, t) + \nabla I(\mathbf{x}, t)^\top \mathbf{v}\Delta t + I_t(\mathbf{x}, t)\Delta t \quad (10.20)$$

Dove ∇I indica il gradiente spaziale $(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})^\top$, e I_t è la derivata temporale di I .

Sostituendo la (10.18), e dividendo per Δt , si ottiene:

$$\nabla I(\mathbf{x}, t)^\top \mathbf{v} + I_t(\mathbf{x}, t) = 0. \quad (10.21)$$

Questa equazione esprime l'ipotesi della costanza della luminosità nella immagine (*image brightness constancy equation*) o equazione del vincolo del gradiente. Osserviamo che vi compaiono due incognite (le due componenti di $\mathbf{v} = (v_x, v_y)$), ma l'equazione lineare è una sola: il problema è sottovincolato. La (10.21) esprime un vincolo sulla quantità $\nabla I(\mathbf{x}, t)^\top \mathbf{v}$ che corrisponde alla proiezione del campo di moto sulla direzione del gradiente. È questa quantità la sola che si possa ricavare dalla (10.21): è l'*effetto dell'apertura*. Le derivate sono operatori locali, e basandosi su informazione locale, non è possibile misurare il moto se non nella direzione del gradiente della luminosità (figura 10.5).

10.2.1 Calcolo del flusso ottico

Si usa la (10.21), ma bisogna aggiungere vincoli perché il problema è sottovincolato.

10.2.1.1 Horn e Schunck.

Il classico metodo di Horn e Schunk [1981] affronta il problema con la tecnica della regolarizzazione: il funzionale da minimizzare è composto da due termini, uno che esprime i vincoli dati dal problema ed uno che

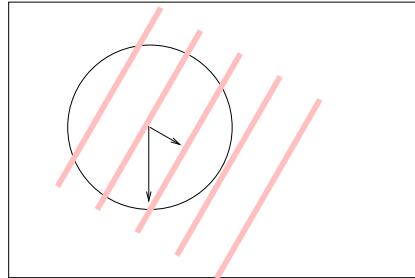


Fig. 10.5. Effetto dell'apertura. Da una visione globale posso dire che il moto delle strisce è dall'alto verso il basso. Da una visione locale (cerchio) sembra che le strisce si muovano in diagonale: la componente del moto lungo la direzione delle strisce non può essere rilevata.

introduce un vincolo di regolarità della soluzione, valutata mediante un operatore derivativo:

$$\min \int_W [\nabla I^\top \mathbf{v} + I_t]^2 + \lambda (||\nabla v_x||^2 + ||\nabla v_y||^2)^2 dW \quad (10.22)$$

L'equazione viene discretizzata e poi risolta iterativamente.

10.2.1.2 Lucas e Kanade.

Il metodo di Lucas e Kanade [1981], invece, assume che il flusso ottico \mathbf{v} sia costante in un intorno W $n \times n$ di ciascun punto, ottenendo così di poter accumulare più equazioni nella stessa incognita \mathbf{v} . Infatti ciascun punto $\mathbf{x}_i \in W$ fornisce una equazione lineare:

$$\nabla I(\mathbf{x}_i, t)^\top \mathbf{v} = -I_t(\mathbf{x}_i, t) \quad (10.23)$$

impilando tutte le $n \times n$ equazioni ottengo un sistema lineare:

$$A\mathbf{v} = \mathbf{b} \quad (10.24)$$

dove

$$A = \begin{bmatrix} \nabla I(\mathbf{x}_1, t)^\top \\ \vdots \\ \nabla I(\mathbf{x}_{n \times n}, t)^\top \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} -I_t(\mathbf{x}_1, t) \\ \vdots \\ -I_t(\mathbf{x}_{n \times n}, t) \end{bmatrix} \quad (10.25)$$

La soluzione ai minimi quadrati di questo sistema lineare sovradeterminato si calcola (metodo della pseudoinversa) con:

$$\mathbf{v} = A^+ \mathbf{b} = (A^\top A)^{-1} A^\top \mathbf{b} \quad (10.26)$$

È usuale e opportuno introdurre una funzione peso $w(\mathbf{x}_i)$ definita sulla finestra W , che dia più peso al centro e meno alla periferia (tipicamente gaussiana). Si ottiene allora la versione pesata della (10.23):

$$w(\mathbf{x}_i) \nabla I(\mathbf{x}_i, t)^\top \mathbf{v} = -w(\mathbf{x}_i) I_t(\mathbf{x}_i, t) \quad (10.27)$$

dunque $A = WA$ e $\mathbf{b} = W\mathbf{b}$, con $W = \text{diag}(w(\mathbf{x}_i))$, da cui

$$\mathbf{v} = (A^\top W^2 A)^{-1} A^\top W^2 \mathbf{b} \quad (10.28)$$

Algoritmo 10.1 FLUSSO OTTICO

Input: Sequenza di immagini

Output: Flusso ottico tra ogni coppia di fotogrammi

- (i) filtrare spazialmente con un nucleo gaussiano 2D;
- (ii) filtrare temporalmente con un nucleo gaussiano 1D;
- (iii) per ogni immagine e per ogni pixel, fissata una finestra W
 - calcolo gradiente e derivata temporale
 - calcolo A e \mathbf{b} con le formule viste
 - calcolo il flusso ottico $\mathbf{v} = (A^\top A)^{-1} A^\top \mathbf{b}$

Relazione con Harris-Stephens. Osserviamo che la matrice $A^\top W^2 A$ è esattamente la matrice che viene usata per l'estrazione dei punti salienti (vedi paragrafo 9.4). Le stesse considerazioni si applicano qui. Ovvvero, siccome $A^\top W^2 A$ viene invertita, il problema del calcolo del flusso ottico è ben condizionato quando la matrice possiede due autovalori non troppo diversi ($c = \lambda_{max}/\lambda_{min} \simeq 1$).

Questa situazione corrisponde ad una finestra molto tessitura, o comunque avente alte frequenze spaziali. Viceversa, ma già lo sapevamo, il calcolo del flusso ottico in regioni molto uniformi è mal condizionato (vuol dire che è estremamente sensibile al rumore nei dati ed alle approssimazioni introdotte dall'algoritmo).

Il lettore interessato troverà una rassegna di algoritmi per il calcolo del flusso ottico in [Barron *e al.*, 1994].

10.3 Algoritmo di tracciamento KLT

Il calcolo del flusso ottico con Lucas e Kanade si può vedere anche come la soluzione ai minimi quadrati del seguente sistema di equazioni (non

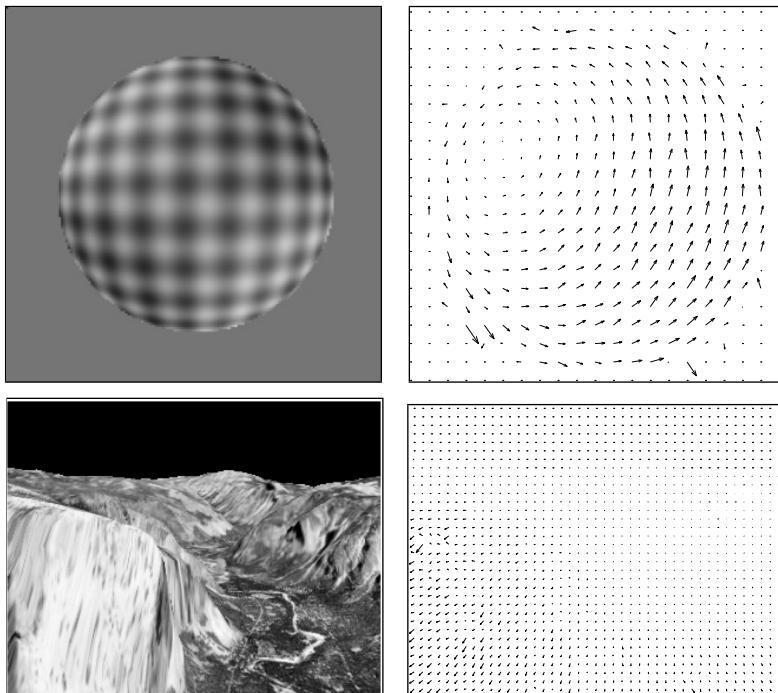


Fig. 10.6. Flusso ottico. Sopra: fotogramma estratto da una sequenza sintetica e relativo flusso ottico. Sotto: fotogramma estratto da una sequenza raffigurante un volo sul parco di Yosemite, e relativo flusso ottico.

lineari) nell'incognita \mathbf{v} :

$$f_i(\mathbf{v}) = I(\mathbf{x} + \mathbf{v}\Delta t, t + \Delta t) - I(\mathbf{x}, t) = 0 \quad i = 1 \dots n \times n \quad (10.29)$$

Un problema di minimi quadrati non lineare si risolve (anche) con il metodo di Gauss-Newton, il quale prevede il calcolo della jacobiana del sistema:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial \mathbf{v}} \\ \vdots \\ \frac{\partial f_{n \times n}}{\partial \mathbf{v}} \end{bmatrix} \quad (10.30)$$

Nel nostro caso

$$\frac{\partial f_i}{\partial \mathbf{v}} = \Delta t \nabla I(\mathbf{x}_i + \mathbf{v}\Delta t, t + \Delta t)^\top. \quad (10.31)$$

Otteniamo dunque una matrice jacobiana J è molto simile alla matrice A definita sopra. La soluzione per \mathbf{v} procede iterativamente calcolando ad ogni passo un incremento $\Delta\mathbf{v}$ come soluzione di

$$J\Delta\mathbf{v} = -\mathbf{f}(\mathbf{v}) \quad (10.32)$$

con

$$\mathbf{f}(\mathbf{v}) = \begin{bmatrix} f_1(\mathbf{v}) \\ \vdots \\ f_{n \times n}(\mathbf{v}) \end{bmatrix} \quad (10.33)$$

Al primo passo, partendo con $\mathbf{v} = \mathbf{0}$, si ha:

$$\frac{\partial f_i}{\partial \mathbf{v}}(\mathbf{0}) = \Delta t \nabla I(\mathbf{x}_i, t + \Delta t)^\top \quad (10.34)$$

e

$$f_i(\mathbf{0}) = I(\mathbf{x}, t + \Delta t) - I(\mathbf{x}, t) \quad i = 1 \dots n \times n \quad (10.35)$$

Si nota immediatamente la somiglianza con l'algoritmo di Lucas e Kanade, visto che $J(\mathbf{0})/\Delta t$ è approssimativamente uguale ad A (cambia solo il tempo in cui sono valutate le derivate spaziali), e $f_i(\mathbf{0})/\Delta t$ è una approssimazione della derivata temporale (e dunque $-\mathbf{f}(\mathbf{0})/\Delta t$ è approssimativamente uguale al vettore \mathbf{b} definito sopra).

Quindi il primo passo della iterazione dell'algoritmo di Gauss-Newton per risolvere il sistema (10.29) è molto simile all'algoritmo di Lucas e Kanade. Si noti che nelle iterazioni successive è necessario calcolare $I(\mathbf{x} + \mathbf{v}\Delta t, t + \Delta t)$, che non si troverà, in generale, sulla griglia dei pixel (serve interpolazione bilineare).

La procedura appena illustrata può essere interpretata come un metodo iterativo per trovare la traslazione che produce il miglior allineamento della finestra W tra il fotogramma al tempo t ed il fotogramma all'istante $t + \Delta t$. Il procedimento prende anche il nome di **registrazione**.

Nel calcolo del flusso ottico ci si ferma al primo passo, ma l'algoritmo proposto da [Tomasi e Kanade, 1991] per accoppiamento di punti salienti lungo una sequenza si basa proprio sulla registrazione appena illustrata. Si tratta dell'algoritmo di Kanade, Lucas e Tomasi (KLT), che si articola in due fasi:

estrazione dei punti salienti nel primo fotogramma della sequenza;
registrazione dei dei punti salienti sul fotogramma successivo.

Per quanto riguarda l'**estrazione**, l'osservazione sul condizionamento di A ci porta a richiedere $\lambda_{max}/\lambda_{min} \simeq 1$. Nella pratica bisogna anche

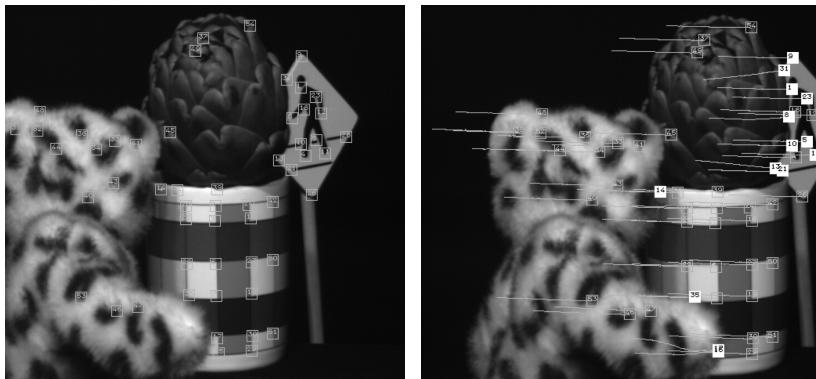


Fig. 10.7. Esempio di tracciamento con KLT su una sequenza di 100 fotogrammi. Il primo fotogramma (a sinistra) con i punti estratti e l'ultimo fotogramma (a destra) con le tracce ed i punti nelle posizioni finali. Tratto da [Tommasini *e al.*, 1998].

aggiungere la condizione che gli autovalori siano “grandi”, per scartare zone con tessiture deboli, che possono essere facilmente corrotte dal rumore. L’estrattore del KLT seleziona punti salienti il cui moto possa essere calcolato in modo massimamente affidabile mediante la condizione

$$\min(\lambda_1, \lambda_2) > \lambda \quad (10.36)$$

dove λ è una soglia prefissata. Se il più piccolo autovalore è sufficiente grande, il fatto che i due autovalori non siano troppo diversi è verificato automaticamente perché il valore del massimo autovalore è superiormente limitato, visto che la variazione di livello di grigio nell’immagine è limitata.

In un contesto di **tracciamento di punti** l’operazione di registrazione viene effettuata tra ogni coppia consecutiva di fotogrammi. Più in dettaglio, si procede come illustrato nell’algoritmo 10.2.

In [Shi e Tomasi, 1994] l’algoritmo viene esteso con un modello *affine* per la deformazione delle finestre (invece che solo traslazionale). In effetti si può complicare a piacimento il modello del moto (10.29), basta essere in grado di calcolarne la jacobiana, che serve dentro Gauss-Newton.

Esercizi ed approfondimenti

- 10.1 Che forma ha il campo di moto quando si osserva una scena planare? (Suggerimento: dall’equazione del piano $\mathbf{n}^\top \mathbf{M} = d$ si

Algoritmo 10.2 TRACCIATORE KLT

Input: Sequenza di immagini**Output:** Tracce di punti salienti

- (i) filtrare spazialmente con un nucleo gaussiano 2D;
 - (ii) filtrare temporalmente con un nucleo gaussiano 1D;
 - (iii) estrarre punti salienti nel primo fotogramma $I(0)$;
 - (iv) per ogni coppia $I(t), I(t+1)$ di fotogrammi della sequenza;
 - registrazione: per ogni punto saliente di $I(t)$, calcola \mathbf{v} su una finestra $n \times n$ centrata sul punto risolvendo (10.29);
 - applica il moto a ciascun punto, ottenendone la posizione in $I(t+1)$.
-

ricava Z e si sostituisce nella (10.6))**Soluzione:**

$$\begin{aligned} \dot{x} = & -\frac{V_1 n_3}{d} - \Omega_2 + \left(-\frac{V_1 n_1}{d} + \frac{V_3 n_3}{d} \right) x + \left(\Omega_3 - \frac{V_1 n_2}{d} \right) y + \\ & + \left(\frac{V_3 n_2}{d} + \Omega_1 \right) yx + \left(\frac{V_3 n_1}{d} - \Omega_2 \right) x^2 \\ \dot{y} = & -\frac{V_2 n_3}{d} + \Omega_1 + \left(-\frac{V_2 n_1}{d} - \Omega_3 \right) x + \left(-\frac{V_2 n_2}{d} + \frac{V_3 n_3}{d} \right) y + \\ & + \left(\frac{V_3 n_1}{d} - \Omega_2 \right) yx + \left(\frac{V_3 n_2}{d} + \Omega_1 \right) y^2 \end{aligned} \tag{E10.1}$$

Si tratta di un polinomio quadratico nelle coordinate (x, y) dei punti.

- 10.2 Assumendo moto puramente traslazionale e velocità costante, stimare il tempo di impatto, ovvero il tempo che manca alla collisione con un determinato punto (visibile) della scena, conoscendo solo il campo di moto, senza sapere velocità e distanza del punto.

Soluzione: Nelle ipotesi date, l'equazione del campo di moto è la (10.17). Il tempo necessario per collidere con un punto di profondità Z è $t = Z/V_3$, ma entrambe queste due quantità sono ignote: possiamo misurare solo il campo di moto. Tuttavia dalla (10.17) si ricava facilmente

$$t = \frac{Z}{V_3} = \frac{x - x_0}{\dot{x}} \tag{E10.2}$$

(analogamente per y).

- 10.3 Con riferimento alla equazione 10.7, il vettore $\begin{bmatrix} \mathbf{V} \\ \boldsymbol{\Omega} \end{bmatrix}$ si chiama *kinematic screw*, e la (10.7) si riscrive come

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = J_{xy} \begin{bmatrix} \mathbf{V} \\ \boldsymbol{\Omega} \end{bmatrix} \quad (\text{E10.3})$$

dove la matrice J_{xy} viene indicata (nell'ambito dell'asservimento visuale o *visual servoing*) come matrice di interazione o jacobiana dell'immagine. Come si vede dalla formula, essa lega le velocità di un punto nello spazio 3D alle velocità della sua proiezione nell'immagine.

Bibliografia

- Barron J. L.; Fleet D. J.; Beauchemin S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision*, **12**(1), 43–77.
- Horn B. K. P.; Schunk B. G. (1981). Determining optical flow. *Artificial Intelligence*, **17**, 185–203.
- Lucas B. D.; Kanade T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*.
- Shi J.; Tomasi C. (1994). Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600.
- Tomasi C.; Kanade T. (1991). Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, Pittsburg, PA.
- Tommasini T.; Fusiello A.; Trucco E.; Roberto V. (1998). Making good features track better. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 178–183, Santa Barbara, CA. IEEE Computer Society Press.

11

Orientazione

Sotto questo titolo studieremo diversi problemi che richiedono il calcolo di una trasformazione rigida tridimensionale (chiamata *orientazione*, con abuso di linguaggio) a partire da corrispondenze tra punti. Si considerano tre categorie di problemi, che dipendono dallo spazio (2D o 3D) dove i punti corrispondenti vengono misurati.

- (2D-2D) sono date le proiezioni dei punti dell'oggetto nei piani immagine[†] di due distinte fotocamere; si vuole determinare la trasformazione fra i due sistemi di riferimento delle fotocamere.
- (3D-3D) sono date le posizioni dei punti dell'oggetto rispetto a due diversi sistemi di riferimento 3D; si vuole determinare la trasformazione fra i due sistemi di riferimento associati all'oggetto.
- (3D-2D) sono date la posizione dei punti nello spazio 3D e la loro proiezione nel piano della fotocamera; si vuole determinare la trasformazione fra il sistema di riferimento della fotocamera ed un sistema di riferimento relativo all'oggetto (localizzazione dell'oggetto rispetto alla fotocamera o calcolo dell'*attitudine* della fotocamera rispetto all'oggetto.)

Nei primi due casi possiamo pensare sia ad un oggetto statico osservato da un sensore in movimento, che ad un oggetto in movimento osservato da un sensore statico: nel caso 2D-2D il sensore è una fotocamera stenopeica, nel caso 3D-3D, invece, si tratta di un sensore *range*. Il caso 3D-2D è simile alla calibrazione della fotocamera, infatti equivale al calcolo dei parametri intrinseci soltanto.

[†] Si assumono noti i parametri intrinseci.

11.1 Caso 2D-2D: orientazione relativa

Consideriamo la situazione in cui alcuni punti di un corpo rigido siano proiettati in due differenti fotocamere, e per ogni punto in una fotocamera sia dato il punto corrispondente nell'altra fotocamera. Il problema della *orientazione relativa* consiste nel determinare la locazione ed orientazione di una fotocamera rispetto all'altra, assumendo i parametri intrinseci noti.

Abbiamo visto (nel capitolo 9) che è possibile risolvere questo problema passando attraverso il calcolo della matrice essenziale. Riassumendo, i passi del metodo lineare sono i seguenti:

- Convertire le coordinate pixel dei punti in coordinate normalizzate;
- Calcolare la matrice essenziale da un minimo di otto corrispondenze di punti;
- Fattorizzare la matrice essenziale per ottenere traslazione e rotazione.

Vedremo ora un metodo introdotto da [Horn, 1990, 1991] che calcola i parametri del moto della fotocamera a partire direttamente dalle corrispondenze di punti in coordinate normalizzate. Essendo basato su una minimizzazione non-lineare, richiede di partire da una soluzione iniziale “vicina” a quella vera.

11.1.1 Metodo iterativo di Horn

Dati n punti corrispondenti, l'equazione di Longuet-Higgins (9.2) prescrive che per ciascuna coppia di punti coniugati $(\mathbf{p}_i, \mathbf{p}'_i)$, in coordinate normalizzate, si abbia:

$$\mathbf{p}'_i^\top \mathbf{t} \times R\mathbf{p}_i = 0 \quad (11.1)$$

Possiamo dunque formulare una soluzione ai minimi quadrati per il problema della orientazione relativa minimizzando la somma dei quadrati degli scarti dalla (11.1):

$$\varepsilon = \sum_{i=1}^n (\mathbf{p}'_i^\top \mathbf{t} \times R\mathbf{p}_i)^2 \quad (11.2)$$

con il vincolo $\mathbf{t}^\top \mathbf{t} = 1$ (decidiamo di rappresentare la traslazione con un vettore unitario, tanto il fattore di scala assoluto non è conoscibile).

Data una stima iniziale per la rotazione e la traslazione, si può attuare un raffinamento iterativo della soluzione che ad ogni passo porti ad una

riduzione dell'errore ε . Siano $\delta\mathbf{t}$ e $\delta\omega$ le correzioni infinitesimali da apportare alla traslazione e rotazione rispettivamente. Dopo la correzione il prodotto triplo diventa (§ A3.1) :

$$\mathbf{p}'^\top (\mathbf{t} + \delta\mathbf{t}) \times (R\mathbf{p}_i + \delta\omega \times R\mathbf{p}_i) \quad (11.3)$$

Sostituendo nella funzione costo e trascurando termini di ordine superiore, le correzioni si ottengono minimizzando

$$\varepsilon = \sum_{i=1}^n (s_i + \mathbf{c}_i^\top \delta\mathbf{t} + \mathbf{d}_i^\top \delta\omega)^2, \quad (11.4)$$

dove $s_i = \mathbf{p}'_i^\top \mathbf{t} \times R\mathbf{p}_i$, $\mathbf{c}_i = R\mathbf{p}_i \times \mathbf{p}'_i$, $\mathbf{d}_i = R\mathbf{p}_i \times (\mathbf{p}'_i \times \mathbf{t})$, con il vincolo $\mathbf{t}^\top \delta\mathbf{t} = 0$. Infatti, poiché la traslazione è rappresentata da un vettore unitario, la correzione non ne deve alterare la lunghezza.

Il problema di minimizzazione vincolata può essere risolto con il metodo dei moltiplicatori di Lagrange. Invece di ε , si minimizza:

$$\varepsilon' = \varepsilon + 2\lambda(\mathbf{t}^\top \delta\mathbf{t}). \quad (11.5)$$

Differenziando ε' rispetto a $\delta\mathbf{t}$, $\delta\omega$, λ ed uguagliando a zero il risultato si ottiene un sistema lineare:

$$\begin{bmatrix} C & B & \mathbf{t} \\ B^\top & D & \mathbf{0} \\ \mathbf{t}^\top & \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \delta\mathbf{t} \\ \delta\omega \\ \lambda \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{c}} \\ \bar{\mathbf{d}} \\ 0 \end{bmatrix} \quad (11.6)$$

dove

$$B = \sum_{i=1}^n \mathbf{c}_i \mathbf{d}_i^\top \quad C = \sum_{i=1}^n \mathbf{c}_i \mathbf{c}_i^\top \quad D = \sum_{i=1}^n \mathbf{d}_i \mathbf{d}_i^\top \quad (11.7)$$

$$\bar{\mathbf{c}} = \sum_{i=1}^n s_i \mathbf{c}_i^\top \quad \bar{\mathbf{d}} = \sum_{i=1}^n s_i \mathbf{d}_i^\top \quad (11.8)$$

Una volta che si siano ottenute le correzioni $\delta\mathbf{t}$ e $\delta\omega$, bisogna applicarle tenendo presente che gli incrementi calcolati sono piccoli ma *finiti* (e dunque le proprietà che valgono per incrementi infinitesimi sono soddisfatte solo approssimativamente). Per quanto riguarda la traslazione basta normalizzare il vettore dopo l'aggiunta dell'incremento. La rotazione, invece, viene aggiornata moltiplicandola per

$$\begin{bmatrix} 0 & -\delta\omega_3 & \delta\omega_2 \\ \delta\omega_3 & 0 & -\delta\omega_1 \\ -\delta\omega_2 & \delta\omega_1 & 0 \end{bmatrix}. \quad (11.9)$$

Il risultato, però, non è una matrice ortogonale (se gli incrementi sono

finiti). Bisogna imporne l'ortogonalità sfruttando la SVD come segue. Sia \hat{R} la matrice quasi ortogonale ottenuta dopo l'aggiornamento e sia $\hat{R} = UDV^\top$ la sua SVD. Si può dimostrare che $R = UV^\top$ è la matrice ortogonale ad essa più vicina, in norma di Frobenius (proposizione 1.44).

Si veda la funzione `horn`.

11.2 Caso 3D-3D: orientazione assoluta

Supponiamo di avere due insiemi di punti 3D, che corrispondono ad un'unica forma, ma che sono espressi in due diversi sistemi di riferimento. Chiameremo uno di questi insiemi \mathcal{X} e l'altro \mathcal{Y} . Assumiamo che per ogni punto di \mathcal{Y} sia noto il punto corrispondente in \mathcal{X} . Il problema dell'*orientazione assoluta* (o problema della registrazione di insiemi di punti) consiste nel trovare la trasformazione rigida 3D (rotazione e traslazione) da applicare ad \mathcal{Y} che lo porti a coincidere con \mathcal{X} , oppure, in presenza di rumore, che renda minima la distanza tra i due insiemi di punti.

Consideriamo il problema più generale della orientazione assoluta con scala, nel quale la relazione tra i due insiemi è la seguente:

$$\mathbf{X}^i = s(R\mathbf{Y}^i + \mathbf{t}) \quad \text{per ogni } i = 1 \dots N \quad (11.10)$$

dove R è una matrice di rotazione 3×3 , \mathbf{t} è un vettore di traslazione 3×1 , s è uno scalare e $\mathbf{X}_i, \mathbf{Y}_i$ sono punti corrispondenti (in coordinate cartesiane) negli insiemi \mathcal{X} ed \mathcal{Y} . L'obiettivo della registrazione è risolvere:

$$\min_{R, \mathbf{t}} \sum_{i=1}^N \|\mathbf{X}_i - s(R\mathbf{Y}_i + \mathbf{t})\|^2, \quad (11.11)$$

Vedremo un metodo basato sulla SVD [Arun *e al.*, 1987].

11.2.1 Metodo con SVD

Facendo la media su i di entrambi i membri della (11.10) si ottiene immediatamente una espressione per la traslazione

$$\mathbf{t} = \frac{1}{s} \left(\frac{1}{N} \sum_{i=1}^N \mathbf{X}_i \right) - R \left(\frac{1}{N} \sum_{i=1}^N \mathbf{Y}_i \right) \quad (11.12)$$

Sostituendo questa nella (11.10) si ottiene di eliminare la traslazione dal problema:

$$\bar{\mathbf{X}}_i = sR\bar{\mathbf{Y}}_i \quad (11.13)$$

dove $\bar{\mathbf{X}}_i = \mathbf{X}_i - \frac{1}{N} \sum_{i=1}^N \mathbf{X}_i$ e $\bar{\mathbf{Y}}_i = \mathbf{Y}_i - \frac{1}{N} \sum_{i=1}^N \mathbf{Y}_i$ sono gli insiemi “centralizzati”, ottenuti sottraendo i rispettivi centroidi.

Poiché le matrici ortogonali applicate ad un vettore non ne alterano il modulo, si può immediatamente ricavare il fattore di scala s prendendo la norma della (11.13):

$$\|\bar{\mathbf{X}}_i\| = s\|\bar{\mathbf{Y}}_i\|. \quad (11.14)$$

Rimane il problema di stimare la rotazione nella (11.13). Siano \bar{X} la matrice $3 \times N$ ottenuta accostando uno accanto all’altro i vettori (colonna) $\bar{\mathbf{X}}_i$ ed \bar{Y} la matrice ottenuta allo stesso modo con i vettori $s\bar{\mathbf{Y}}_i$. Si verifica facilmente che la funzione obiettivo (o residuo) si riscrive:

$$\varepsilon = \sum_{i=1}^N \|\bar{\mathbf{X}}_i - sR\bar{\mathbf{Y}}_i\|^2 = \|\bar{X} - sR\bar{Y}\|_F^2 \quad (11.15)$$

Riconosciamo una istanza del problema procurano ortogonale e dunque la proposizione 1.44 ci garantisce che la soluzione è data da:

$$R = V \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(VU^T) \end{bmatrix} U^\top \quad (11.16)$$

dove $UDV^T = \bar{Y}\bar{X}^T$ è la SVD della matrice $\bar{Y}\bar{X}^T$.

Seguendo [Kanatani, 1993] abbiamo sostituito l’identità tra V e U con una matrice che serve a garantire che il risultato sia una matrice di rotazione, ovvero abbia determinante positivo.

Si veda la funzione `absolute`.

11.2.2 ICP

Nei casi pratici, però, si devono spesso registrare insiemi di punti 3D per i quali non si conoscono le corrispondenze. In tal caso si può applicare un algoritmo chiamato *Iterative Closest Point* [Besl e McKay, 1992, Chen e Medioni, 1992] o ICP, che risolve simultaneamente il problema delle corrispondenze e la stima della trasformazione rigida.

Dato un insieme di punti \mathcal{Y} ed una superficie[†] \mathcal{X} , dove \mathcal{Y} è un sottoinsieme di \mathcal{X} , per ogni punto \mathbf{Y}_i dell’insieme \mathcal{Y} , esiste almeno un punto \mathbf{X}_i sulla superficie \mathcal{X} che è il più vicino a \mathbf{Y}_i rispetto a tutti gli altri punti in \mathcal{X} . In una iterazione, ICP assume che i punti più vicini siano corrispondenti, risolve il problema di orientazione assoluta e applica la

[†] In generale, ICP funziona per registrare sia punti con punti ma anche punti con superfici.

trasformazione rigida ottenuta a \mathcal{Y} . L’idea è che, anche se le corrispondenze non sono quelle vere, la trasformazione che si ottiene avvicina \mathcal{Y} ad \mathcal{X} . Infatti, [Besl e McKay, 1992] dimostrarono che l’algoritmo converge ad un minimo locale dell’errore (11.11). Se si parte abbastanza vicino al minimo globale, si ottiene la soluzione vera. L’algoritmo ICP si riassume così:

Algoritmo 11.1 ICP

Input: Due insiemi di punti 3D, \mathcal{X} e \mathcal{Y}

Output: Moto rigido che allinea \mathcal{Y} su \mathcal{X}

- (i) Per ogni punto in \mathcal{Y} calcola il più vicino in \mathcal{X} ;
 - (ii) Con le corrispondenze trovate al passo 1, calcola la trasformazione incrementale (R, \mathbf{t}) (risolvendo l’orientazione assoluta);
 - (iii) Applica la trasformazione incrementale trovata nel passo 2 agli elementi di \mathcal{Y} ;
 - (iv) Se la media dell’errore quadratico è minore di una certa soglia, termina altrimenti vai al passo 1.
-

Questo metodo convergerà al più vicino minimo locale della somma delle distanze al quadrato tra i punti più vicini. Per assicurare la convergenza alla registrazione corretta, è richiesta una buona stima iniziale della trasformazione tra gli insiemi dei punti. Registrazioni sbagliate possono accadere se l’errore nella trasformazione iniziale è troppo grande (tipicamente è più grande di 20 gradi) o se la superficie non contiene sufficienti informazioni sulla forma per convergere. L’allineamento iniziale può essere ottenuto:

- campionando l’insieme di tutte le trasformazioni possibili;
- considerando i momenti di \mathcal{X} e \mathcal{Y} (media e assi principali).

La fase di maggior costo computazionale è la ricerca del closest-point. Il metodo ingenuo richiede, per ciascun punto, tempo $O(n)$, dove n è la cardinalità di \mathcal{X} . Usando i kD-trees si può fare in $O(\log n)$.

Modifiche sono state introdotte sul ICP originale per migliorare la velocità di convergenza e registrare insiemi solo parzialmente sovrapposti. L’idea è di selezionare le coppie di punti “buone”, in base a certi criteri euristici. I due classici sono:

- sogliatura sulle distanze
- eliminazione degli accoppiamenti sui bordi

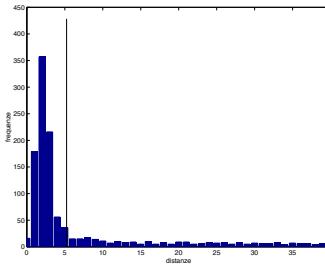


Fig. 11.1. Istogramma delle distanze tra punti accoppiati nell'ultima iterazione di ICP. La linea verticale mostra la soglia di reiezione calcolata con X84, come in [Trucco e al., 1999].

Il primo criterio fissa una soglia sulla distanza massima consentita tra i punti più vicini. Le coppie che hanno distanza maggiore non vengono considerate nel calcolo della orientazione assoluta (figura 11.1). Il secondo si basa sull'osservazione che, nel caso di superfici parzialmente sovrapposte, è verosimile che i punti appartenenti alle parti non comuni vengano accoppiati con punti appartenenti al bordo della superficie. Queste corrispondenze sono evidentemente sbagliate (poiché i punti appartenenti alle zone non in comune non dovrebbero venire accoppiati), e dunque vengono eliminate.

Per rendere più veloce l'accoppiamento dei punti si può anche abbandonare la ricerca dei punti più vicini, accoppiando un punto \mathbf{X}_i con il punto di \mathcal{Y} che si incontra lungo la normale ad alla superficie \mathcal{X} nel punto \mathbf{X}_i (*normal shooting*). Per ulteriori dettagli si rimanda alla lettura di [S. Rusinkiewicz, 2001].

Si veda la funzione MATLAB `icp`.

Per catturare l'intera superficie di un oggetto, spesso sono richieste molte (più di due) immagini *range*. La registrazione di più di due insiemi di punti in un unico sistema di riferimento può essere semplicemente ottenuta dalla concatenazione di registrazioni di coppie di insiemi. Tale procedimento, però non conduce alla soluzione ottimale. Per esempio, se abbiamo tre insiemi di punti sovrapposti e calcoliamo la registrazione tra gli insiemi uno e due seguita dalla registrazione degli insiemi uno e tre, non necessariamente stiamo minimizzando la media delle distanze al quadrato tra gli insiemi due e tre. Serve dunque una procedura di aggiustamento che operi a livello globale, ovvero prendendo in considerazione tutti gli insiemi simultaneamente, come in [Fusiello e al., 2002].

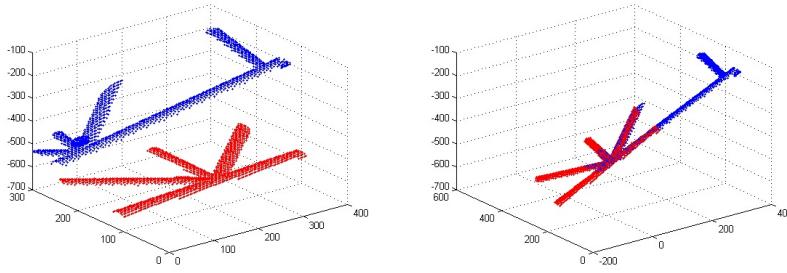


Fig. 11.2. Registrazione di due immagini *range* con ICP. Insiemi di punti nella posizione iniziale (a sinistra) e nelle posizioni finale (a destra).

11.3 Caso 2D-3D: orientazione esterna

Supponiamo di conoscere la posizione 3D di alcuni punti e le loro proiezioni nel piano immagine della fotocamera. Il problema dell'*orientazione esterna* consiste nel determinare la posizione ed orientazione della fotocamera (chiamate anche collettivamente *attitudine* della fotocamera), assumendo noti i parametri intrinseci.

Siano $\mathbf{M}_1 \dots \mathbf{M}_n$ gli n punti di un oggetto, espressi nel sistema di riferimento 3D dell'oggetto stesso e siano $\mathbf{m}_1 \dots \mathbf{m}_n$ i rispettivi punti proiettati sul piano immagine. La relazione fra un punto dell'oggetto ed un punto dell'immagine è data dalla proiezione prospettica (in coordinate normalizzate):

$$\mathbf{p}_i \simeq [R|\mathbf{t}]\mathbf{M}_i. \quad (11.17)$$

dove $\mathbf{p}_i = [x_i, y_i, 1]^\top = K^{-1}\mathbf{m}_i$ sono le *coordinate immagine normalizzate*. Se si hanno a disposizione almeno sei punti si può procedere alla calibrazione con il metodo lineare (§ 4.2), ottenendo una MPP uguale a $[R|\mathbf{t}]$. In realtà, a causa del rumore presente nei dati, la sottomatrice di sinistra non è – in generale – ortogonale, e dunque questa proprietà deve essere forzata a posteriori con SVD, come nel metodo di Horn. L'algoritmo proposto da [Fiore, 2001] invece, essendo basato sulla orientazione assoluta, produce una matrice di rotazione per costruzione.

11.3.1 Metodo lineare di Fiore

Riscriviamo la (11.17) esplicitando i fattori di scala ζ_i :

$$\zeta_i K^{-1}\mathbf{m}_i = [R|\mathbf{t}]\mathbf{M}_i = R\tilde{\mathbf{M}}_i + \mathbf{t} \quad \text{for all } i. \quad (11.18)$$

L'idea centrale dell'algoritmo consiste nel ricavare le ζ_i per ricondurre il problema alla orientazione assoluta. Riscriviamo la (11.18) in forma matriciale:

$$K^{-1} \underbrace{[\zeta_1 \mathbf{m}_1, \zeta_2 \mathbf{m}_2, \dots, \zeta_n \mathbf{m}_n]}_W = [R|\mathbf{t}] \underbrace{[\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n]}_M.$$

Sia $r = \text{r}(M)$. Si calcoli la SVD di M : $M = UDV^\top$ e sia V_r la matrice composta dalle ultime $n - r$ colonne di V , le quali generano il nucleo di M . Quindi: $MV_r = 0_{3 \times (n-r)}$ ed anche:

$$K^{-1} W V_r = 0_{3 \times (n-r)}. \quad (11.19)$$

Prendendo il vec di entrambi i membri e sfruttando il prodotto di Kronecker si ottiene:

$$(V_r^\top \otimes K^{-1}) \text{vec}(W) = \mathbf{0}. \quad (11.20)$$

Si osservi ora che:

$$\text{vec}(W) = \begin{bmatrix} \zeta_1 \mathbf{m}_1 \\ \zeta_2 \mathbf{m}_2 \\ \vdots \\ \zeta_n \mathbf{m}_n \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{m}_1 & 0 & \dots & 0 \\ 0 & 0 & \dots & \mathbf{m}_n \end{bmatrix}}_D \underbrace{\begin{bmatrix} \zeta_1 \\ \vdots \\ \zeta_n \end{bmatrix}}_\zeta$$

E quindi:

$$((V_r^\top \otimes K^{-1}) D) \zeta = \mathbf{0}. \quad (11.21)$$

Impilando abbastanza equazioni come questa si può ricavare ζ – a meno di una costante moltiplicativa – come il nucleo della matrice dei coefficienti. Quante sono “abbastanza”? La dimensione della matrice dei coefficienti in (11.21) è $3(n-r) \times n$, e per determinare una famiglia unidimensionale di soluzioni (dimensione del nucleo pari a uno) deve avere rango $n-1$, quindi deve essere: $3(n-r) \geq n-1$. Ne consegue che servono $n \geq (3r-1)/2$ punti. Per esempio, servono sei punti in posizione generale ($r=4$). Invece se i punti sono coplanari ($r=3$) ne bastano quattro.

Ora che il membro destro della (11.18) è noto a meno di un fattore di scala, non resta che risolvere il problema di orientazione con scala costituito dalla (11.18) con il metodo descritto nel paragrafo 11.2.1. Ne risulta una matrice R ortonormale per costruzione.

Si veda la funzione MATLAB `exterior`.

Questo metodo lineare è veloce, non ha problemi di convergenza ma

i) necessita di più punti del necessario ii) minimizza un errore algebrico.
Per contro, il metodo di Lowe che vediamo ora è iterativo, ma necessita di meno corrispondenze e minimizza un errore di natura geometrica.

11.3.2 Metodo non lineare di Lowe

Espandendo la (11.17) possiamo vedere che ogni corrispondenza di punti genera due equazioni:

$$\begin{cases} x_i = \frac{\mathbf{r}_1^\top \tilde{\mathbf{M}}_i + t_1}{\mathbf{r}_3^\top \tilde{\mathbf{M}}_i + t_3} \\ y_i = \frac{\mathbf{r}_2^\top \tilde{\mathbf{M}}_i + t_2}{\mathbf{r}_3^\top \tilde{\mathbf{M}}_i + t_3} \end{cases} \quad (11.22)$$

dove $R = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]^\top$ e $\mathbf{t} = [t_1, t_2, t_3]^\top$. La matrice di rotazione R viene espressa in funzione di tre parametri (per esempio con i tre angoli di Eulero ϕ_x, ϕ_y, ϕ_z), sicché alla fine avremo un sistema *non lineare* in sei incognite $\mathbf{t}, \phi_x, \phi_y, \phi_z$, le quali possono essere determinate se sono note almeno tre corrispondenze di punti.

Per neutralizzare l'effetto di misurazioni o corrispondenze poco accurate è, comunque, consigliato di usare più corrispondenze possibili, risolvendo in tal modo il sistema col metodo dei minimi quadrati non lineari. Il metodo di Lowe [Lowe, 1991] non è nient'altro che Gauss-Newton applicato alla soluzione di (11.22).

Fissato un punto \mathbf{M}_i l'equazione (11.22) definisce un'applicazione $\ell_i : \mathbb{R}^6 \rightarrow \mathbb{R}^2$ dallo spazio dei 6 parametri delle trasformazioni rigide alle coordinate immagine (x_i, y_i) . Dunque la (11.22) si riscrive, in forma compatta:

$$\mathbf{p}_i - \ell_i(\mathbf{a}) = \mathbf{0}. \quad (11.23)$$

dove $\mathbf{a} = [\mathbf{t}, \phi_x, \phi_y, \phi_z]^\top$. Con $i = 1 \dots n$ si ottiene un sistema (in genere sovradeterminato) di $2n$ equazioni non lineari che può essere risolto col metodo di Gauss-Newton. A partire da una soluzione iniziale, il metodo di Gauss-Newton procede aggiornando il vettore \mathbf{a} delle incognite con $\mathbf{a} \leftarrow \mathbf{a} + \Delta\mathbf{a}$, dove $\Delta\mathbf{a}$ è la soluzione del seguente sistema lineare di $2n$ equazioni:

$$\begin{cases} \mathbf{p}_1 - \ell_1(\mathbf{a}) = \mathbf{J}_{\ell_1} \Delta\mathbf{a} \\ \dots \\ \mathbf{p}_n - \ell_n(\mathbf{a}) = \mathbf{J}_{\ell_n} \Delta\mathbf{a} \end{cases} \quad (11.24)$$

e \mathbf{J}_{ℓ_i} è la jacobiana di $\ell_i(\mathbf{a})$. L'iterazione continua finché il valore della norma dei residui diventa abbastanza piccolo. Come inizializzare? Sperimentalmente si verifica che l'algoritmo non è eccessivamente sensibile alla inizializzazione, grazie al fatto che le non-linearità causate dalla parametrizzazione della rotazione non sono troppo forti.

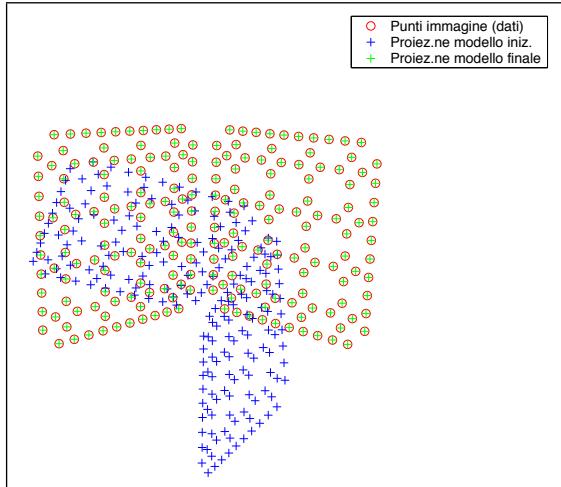


Fig. 11.3. Esempio di applicazione dell'algoritmo di Lowe. I cerchi rossi rappresentano i punti rilevati nell'immagine, e le croci blu sono i corrispondenti punti dell'oggetto proiettati secondo la stima iniziale dell'attitudine. Le croci verdi sono gli stessi punti proiettati secondo la stima finale (risultato), e naturalmente coincidono con i cerchi rossi.

La descrizione dell'algoritmo non sarebbe completa senza la derivazione della jacobiana di ℓ . Per comodità scomponiamo ℓ in due funzioni†:

$$\mathcal{G} : \mathbb{R}^6 \longrightarrow \mathbb{R}^3 \quad \mathcal{G}(\mathbf{a}) = R(\phi_x, \phi_y, \phi_z) \tilde{\mathbf{M}}_i + \mathbf{t} = [X_i, Y_i, Z_i]^\top \quad (11.25)$$

$$\mathcal{F} : \mathbb{R}^3 \longrightarrow \mathbb{R}^2 \quad \mathcal{F}([X_i, Y_i, Z_i]^\top) = \left[\frac{fX_i}{Z_i}, \frac{fY_i}{Z_i} \right]^\top \quad (11.26)$$

tali che

$$\ell(\mathbf{a}) = (\mathcal{F} \circ \mathcal{G})(\mathbf{a}) \quad (11.27)$$

La jacobiana di ℓ si ottiene applicando la regola di derivazione di funzioni composte:

$$J_\ell = J_{\mathcal{F}} J_{\mathcal{G}}. \quad (11.28)$$

† Abbandoniamo da qui in poi il pedice i per alleggerire la notazione.

Il calcolo della jacobiana di \mathcal{F} è immediato:

$$J_{\mathcal{F}} = \begin{bmatrix} \frac{\partial x_i}{\partial X_i} & \frac{\partial x_i}{\partial Y_i} & \frac{\partial x_i}{\partial Z_i} \\ \frac{\partial y_i}{\partial X_i} & \frac{\partial y_i}{\partial Y_i} & \frac{\partial y_i}{\partial Z_i} \end{bmatrix} = \begin{bmatrix} \frac{1}{Z_i} & 0 & -\frac{X_i}{Z_i^2} \\ 0 & \frac{1}{Z_i} & -\frac{Y_i}{Z_i^2} \end{bmatrix} \quad (11.29)$$

La jacobiana della funzione \mathcal{G} è:

$$J_{\mathcal{G}} = \begin{bmatrix} \frac{\partial X_i}{\partial \phi_x} & \frac{\partial X_i}{\partial \phi_y} & \frac{\partial X_i}{\partial \phi_z} & \frac{\partial X_i}{\partial t_x} & \frac{\partial X_i}{\partial t_y} & \frac{\partial X_i}{\partial t_z} \\ \frac{\partial Y_i}{\partial \phi_x} & \frac{\partial Y_i}{\partial \phi_y} & \frac{\partial Y_i}{\partial \phi_z} & \frac{\partial Y_i}{\partial t_x} & \frac{\partial Y_i}{\partial t_y} & \frac{\partial Y_i}{\partial t_z} \\ \frac{\partial Z_i}{\partial \phi_x} & \frac{\partial Z_i}{\partial \phi_y} & \frac{\partial Z_i}{\partial \phi_z} & \frac{\partial Z_i}{\partial t_x} & \frac{\partial Z_i}{\partial t_y} & \frac{\partial Z_i}{\partial t_z} \end{bmatrix} \quad (11.30)$$

per la parte rotazionale si calcola sfruttando una regola che fornisce la derivata di un vettore $\tilde{\mathbf{M}}_i = [X_i, Y_i, Z_i]^\top$ rispetto ad un angolo di rotazione (§ A3.1):

$$\frac{\partial \tilde{\mathbf{M}}_i}{\partial \phi_x} = [1, 0, 0]^\top \times \tilde{\mathbf{M}}_i = [0, -Z_i, Y_i]^\top \quad (11.31)$$

$$\frac{\partial \tilde{\mathbf{M}}_i}{\partial \phi_y} = [0, 1, 0]^\top \times \tilde{\mathbf{M}}_i = [Z_i, 0, -X_i]^\top \quad (11.32)$$

$$\frac{\partial \tilde{\mathbf{M}}_i}{\partial \phi_z} = [0, 0, 1]^\top \times \tilde{\mathbf{M}}_i = [-Y_i, X_i, 0]^\top \quad (11.33)$$

$$J_{\mathcal{G}} = \left[\begin{array}{c|ccc} \frac{\partial \tilde{\mathbf{M}}_i}{\partial \phi_x} & \frac{\partial \tilde{\mathbf{M}}_i}{\partial \phi_y} & \frac{\partial \tilde{\mathbf{M}}_i}{\partial \phi_z} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \right] = \left[\begin{array}{ccc|ccc} 0 & Z_i & -Y_i & 1 & 0 & 0 \\ -Z_i & 0 & X_i & 0 & 1 & 0 \\ Y_i & -X_i & 0 & 0 & 0 & 1 \end{array} \right] \quad (11.34)$$

Quindi la jacobiana di ℓ_i è

$$J_{\ell} = \begin{bmatrix} -\frac{X_i Y_i}{Z_i^2} & \left(1 + \frac{X_i^2}{Z_i^2}\right) & -\frac{Y_i Z_i}{Z_i^2} & \frac{Z_i}{Z_i^2} & 0 & -\frac{X_i}{Z_i^2} \\ -\left(1 + \frac{Y_i^2}{Z_i^2}\right) & \frac{X_i Y_i}{Z_i^2} & \frac{X_i Z_i}{Z_i^2} & 0 & \frac{Z_i}{Z_i^2} & \frac{Y_i}{Z_i^2} \end{bmatrix} \quad (11.35)$$

Si veda la funzione `lowe`.

11.3.3 Metodo diretto

Una debolezza del metodo di Lowe è il fatto di richiedere l'estrazione ed accoppiamento di punti salienti. Un fallimento in questa fase si ripercuote negativamente sulla soluzione finale (a meno di usare tecniche robuste).

Descriptoremo ora una tecnica alternativa [Marchand *e al.*, 1999] che, assumendo di possedere un modello sintetico dell'oggetto (invece che una lista di punti salienti) evita l'estrazione e l'accoppiamento dei punti. L'idea degli autori è semplice: l'attitudine corretta della fotocamera è quella per cui la proiezione del modello coincide con l'immagine dell'oggetto. La "coincidenza" si misura sommando il gradiente dell'immagine in corrispondenza dei punti che appartengono al contorno della proiezione del modello. Quando il contorno del modello proiettato "raccoglie" il massimo del gradiente nell'immagine vuol dire che coincide con il contorno dell'immagine dell'oggetto. Il calcolo della orientazione esterna è riconducibile dunque ad un problema di ottimizzazione di una opportuna funzione costo calcolata direttamente sulla immagine.

Il modello dell'oggetto è costituito dai segmenti che danno origine ad una discontinuità di livello di grigio nell'immagine (*edge*), dunque possono essere sia *edge* geometrici che *edge* di colore.

Data una stima della posizione iniziale dell'oggetto, la posizione esatta si ottiene massimizzando una funzione costo non lineare rispetto ai parametri $\mathbf{a} = [\mathbf{t}, \phi_x, \phi_y, \phi_z]^\top$. Questa funzione costo indica quanto i contorni del modello corrispondano con quelli dell'immagine:

$$E(\mathbf{a}) = \frac{1}{|\Gamma_{\mathbf{a}}|} \sum_{\mathbf{x} \in \Gamma_{\mathbf{a}}} \|\nabla I(\mathbf{x})\| = \frac{1}{|\Gamma_{\mathbf{a}}|} \sum_{\mathbf{x}} \|\nabla I(\mathbf{x}) \chi_{\Gamma_{\mathbf{a}}}(\mathbf{x})\| \quad (11.36)$$

dove $\nabla I(\mathbf{x})$ è il gradiente del livello di grigio dell'immagine, $\Gamma_{\mathbf{a}}$ è la proiezione (visibile) del modello secondo i parametri \mathbf{a} , e $\chi_{\Gamma_{\mathbf{a}}}$ è la funzione caratteristica dell'insieme $\Gamma_{\mathbf{a}}$.

Ogni punto con gradiente diverso da zero e con $\chi_{\Gamma_{\mathbf{a}}} = 1$ dà un contributo alla funzione costo. La funzione costo pesa quanto i punti del modello proiettati sono vicini ad un edge immagine, identificato dal massimo del gradiente.

I parametri di attitudine finali $\hat{\mathbf{a}}$ sono dati da:

$$\hat{\mathbf{a}} = \arg \max_{\mathbf{a}} E(\mathbf{a}) \quad (11.37)$$

Per avere un bacino di convergenza più ampio e liscio, consideriamo non solo il punto appartenente a $\Gamma_{\mathbf{a}}$, ma anche i punti vicini, pesati da una funzione gaussiana decrescente con la distanza dal contorno. Questo

può essere formalizzato considerando una funzione caratteristica “sfumata” $\hat{\chi}_{\Gamma_a}$ che decresce secondo una legge gaussiana tanto più i punti sono lontani dal contorno attuale.

In presenza di rumore o di immagini tessiture, è consigliabile tener conto della direzione del gradiente, proiettandolo sulla normale al contorno del modello $\mathbf{n}(\mathbf{x})$:

$$E(\mathbf{a}) = \frac{1}{|\Gamma_{\mathbf{a}}|} \sum_{\mathbf{x}} |\nabla I(\mathbf{x})^T \mathbf{n}(\mathbf{x}) \hat{\chi}_{\Gamma_{\mathbf{a}}}(\mathbf{x})| \quad (11.38)$$

L’uso della direzione del gradiente rende la funzione di costo selettiva nei dintorni del minimo e ciò neutralizza l’effetto di sfumatura, che può appiattire troppo la funzione costo nelle vicinanze dell’ottimo.

La funzione costo viene quindi minimizzata iterativamente impiegando tecniche dirette (che usano solo il valore della funzione, non le sue derivate) [Hooke e Jeeves, 1961]. Trattandosi di una funzione costo non lineare, serve una soluzione iniziale vicina al minimo globale.



Fig. 11.4. L’oggetto da tracciare è la bottiglietta spray. Il modello riproiettato sulla immagine è mostrato in verde in due fotogrammi campione della sequenza. A sinistra la mappa di gradiente sulla quale si basa il recupero dell’attitudine della fotocamera.

Questo metodo può essere efficacemente impiegato in uno schema di tracciamento dell’oggetto (*object tracking*) in una sequenza video, in cui l’attitudine calcolata per il fotogramma corrente fornisce la soluzione di partenza per il calcolo dell’attitudine per il fotogramma successivo, come illustrato nell’esempio di figura 11.4.

Esercizi ed approfondimenti

- 11.1 Si modifichi il metodo di Lowe per funzionare con accoppiamenti di linee.
- 11.2 La stessa idea del metodo diretto è impiegata nel metodo di

calibrazione introdotto da [Robert, 1996]. L'autore usa il metodo lineare con sei punti per ottenere una stima iniziale della MPP, che poi raffina minimizzando una funzione costo simile alla (11.36). Si implementi in MATLAB il metodo di calibrazione di Robert.

Bibliografia

- Arun K. S.; Huang T. S.; Blostein S. D. (1987). Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **9**(5), 698–700.
- Besl P.; McKay N. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**(2), 239–256.
- Chen Y.; Medioni G. (1992). Object modeling by registration of multiple range images. *Image and Vision Computing*, **10**(3), 145–155.
- Fiore P. D. (2001). Efficient linear solution of exterior orientation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(2), 140–148.
- Fusiello A.; Castellani U.; Ronchetti L.; Murino V. (2002). Model acquisition by registration of multiple acoustic range views. In *Proceedings of the European Conference on Computer Vision*, pp. 805–819.
- Hooke R.; Jeeves T. (1961). Direct search solution of numerical and statistical problems. *Journal of the Association for Computing Machinery (ACM)*, pp. 212–229.
- Horn B. (1990). Relative orientation. *International Journal of Computer Vision*, **4**(1), 59–78.
- Horn B. (1991). Relative orientation revisited. *Journal of the Optical Society of America A*, **8**(10), 1630–1638.
- Kanatani K. (1993). *Geometric Computation for Machine Vision*. Oxford University Press.
- Lowe D. (1991). Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**(5), 441–450.
- Marchand E.; Bouthemy P.; Chaumette F.; Moreau V. (1999). Robust real-time visual tracking using a 2-D-3-D model-based approach. *Proc. of the International Conference on Computer Vision*.
- Robert L. (1996). Camera calibration without feature extraction. *Computer Vision, Graphics, and Image Processing*, **63**(2), 314–325.

- S. Rusinkiewicz M. L. (2001). Efficient variants of the ICP algorithm. In *IEEE Int. Conf. on 3-D Imaging and Modeling, 3DIM '01*, Quebec City (Canada).
- Trucco E.; Fusiello A.; Roberto V. (1999). Robust motion and correspondence of noisy 3-D point sets with missing data. *Pattern Recognition Letters*, **20**(9), 889–898.

12

Ricostruzione non calibrata

Sinora abbiamo assunto che almeno i parametri intrinseci delle fotocamere fossero noti. Cosa è possibile dire della struttura del mondo quando lo osserviamo con fotocamere completamente non calibrate, ovvero delle quali non conosciamo nulla? La risposta, in generale, è che qualcosa si può dire. Infatti la struttura della scena può essere recuperata a meno di una *proiettività* incognita: per questo si chiama ricostruzione proiettiva. Inoltre, se alcune ragionevoli condizioni sono soddisfatte (per esempio almeno tre viste e parametri intrinseci costanti) è possibile passare alla struttura euclidea, ovvero quella che ottenevamo nel caso calibrato. Questo avviene attraverso due strade diverse: possiamo partire dalla ricostruzione proiettiva e quindi promuoverla ad euclidea, oppure possiamo recuperare i parametri intrinseci a partire dalle matrici fondamentali e ricondurci al caso calibrato.

Prima di illustrare le tecniche fissiamo la notazione, chiariamo i termini del problema e l'ambiguità della soluzione. Consideriamo un insieme di punti 3D, visti da m fotocamere con matrici $\{P_i\}_{i=1\dots m}$. Siano \mathbf{m}_i^j le coordinate (omogenee) della proiezione del j -esimo punto nella i -esima fotocamera. Il problema della **ricostruzione** può essere posto nel seguente modo: dato l'insieme delle coordinate pixel $\{\mathbf{m}_i^j\}$, trovare l'insieme delle MPP delle fotocamere $\{P_i\}$ e la struttura della scena $\{\mathbf{M}^j\}$ tale che:

$$\mathbf{m}_i^j \simeq P_i \mathbf{M}^j. \quad (12.1)$$

Senza ulteriori vincoli otterremo, in generale, una ricostruzione definita a meno di una proiettività arbitraria. Infatti, se $\{P_i\}$ e $\{\mathbf{M}^j\}$ sono una ricostruzione, ovvero soddisfano la (12.1), anche $\{P_i T\}$ e $\{T^{-1} \mathbf{M}^j\}$ soddisfano la (12.1) per ogni matrice 4×4 T non singolare.

La matrice T specifica una trasformazione lineare dello spazio proiettivo 3D (una proiettività).

Anche se da una ricostruzione proiettiva si possono trarre alcune utili informazioni [Robert e Faugeras, 1995], quello che vorremmo ottenere è una *ricostruzione euclidea* della struttura, che differisce dalla struttura vera per una similarità. Quest'ultima è composta da una trasformazione rigida (dovuta alla scelta arbitraria del sistema di riferimento mondo) più un cambio uniforme di scala (dovuto alla ben nota ambiguità profondità-velocità).

12.1 Ricostruzione proiettiva e promozione euclidea

In questo paragrafo seguiremo la prima delle due strade delineate all'inizio, ovvero effettuare una ricostruzione proiettiva seguita dalla promozione ad euclidea. Iniziamo con l'algoritmo di ricostruzione proiettiva da molte viste proposto in [Sturm e Triggs, 1996] e basato a sua volta sul metodo di fattorizzazione di Tomasi e Kanade [1992], che vedremo in § 12.3.

12.1.1 Ricostruzione proiettiva

Consideriamo m fotocamere che inquadrono n punti dello spazio tridimensionale, $\mathbf{M}^1 \dots \mathbf{M}^n$. La solita equazione di proiezione prospettica (con il fattore di scala esplicitato) si scrive

$$\zeta_i^j \mathbf{m}_i^j = P_i \mathbf{M}^j \quad i = 1 \dots m, \quad j = 1 \dots n. \quad (12.2)$$

ovvero, in forma matriciale:

$$\underbrace{\begin{bmatrix} \zeta_1^1 \mathbf{m}_1^1 & \zeta_1^2 \mathbf{m}_1^2 & \dots & \zeta_1^n \mathbf{m}_1^n \\ \zeta_2^1 \mathbf{m}_2^1 & \zeta_2^2 \mathbf{m}_2^2 & \dots & \zeta_2^n \mathbf{m}_2^n \\ \vdots & \vdots & \ddots & \vdots \\ \zeta_m^1 \mathbf{m}_m^1 & \zeta_m^2 \mathbf{m}_m^2 & \dots & \zeta_m^n \mathbf{m}_m^n \end{bmatrix}}_{\text{misure } W} = \underbrace{\begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix}}_P \underbrace{[\mathbf{M}^1, \mathbf{M}^2, \dots, \mathbf{M}^n]}_{\text{struttura } M}. \quad (12.3)$$

In questa equazione gli \mathbf{m}_i^j sono noti e tutto il resto è incognito, comprese le ζ_i^j . Essa dice che W si fattorizza nel prodotto di una matrice P $3m \times 4$ e di una matrice M $4 \times n$, e che per questo ha rango quattro.

Se per il momento assumiamo che le ζ_i^j siano note, ricadiamo in un caso analogo a quello del metodo di Tomasi e Kanade [1992], infatti la

matrice W diviene nota e ne possiamo calcolare la SVD:

$$W = UDV^T. \quad (12.4)$$

Nel caso ideale in cui i dati non sono affetti da errore il rango di W è quattro e dunque $D = \text{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4, 0, \dots, 0)$. Quindi solo le prime quattro colonne di U (V) contribuiscono al prodotto tra matrici. Sia dunque $U_{3m \times 4}$ ($V_{n \times 4}$) la matrice formata dalle prime quattro colonne di U (V). Possiamo scrivere la SVD compatta di W :

$$W = U_{3m \times 4} \text{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4) V_{n \times 4}^T. \quad (12.5)$$

Confrontando con la (12.3) possiamo identificare:

$$P = U_{3m \times 4} \text{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4) \quad \text{e} \quad M = V_{n \times 4}^T \quad (12.6)$$

ottenendo così la ricostruzione cercata. Si noti che la scelta di includere $\text{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ in P è arbitraria. Lo si poteva includere in M oppure ripartire tra i due. Questo comunque è coerente con il fatto che la ricostruzione che si ottiene è data a meno di una proiettività, che assorbe dunque anche $\text{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$.

Nel caso reale in cui i dati sono affetti da errori, il rango di W non è 4. Forzando $D = \text{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4, 0, \dots, 0)$ si ottiene la soluzione che minimizza l'errore in norma di Frobenius:

$$\|W - PM\|_F^2 = \sum_{i,j} \|\zeta_i^j \mathbf{m}_i^j - P_i \mathbf{M}^j\|^2 \quad (12.7)$$

Rimaniamo ora con il problema di stimare le ζ_i^j incognite. Abbiamo visto che se fossero note potrei calcolare P ed M . D'altra parte le ζ_i^j si potrebbero calcolare se conoscessi P ed M , infatti, per un dato punto j l'equazione di proiezione si riscrive:

$$\begin{bmatrix} \zeta_1^j \mathbf{m}_1^j \\ \zeta_2^j \mathbf{m}_2^j \\ \vdots \\ \zeta_m^j \mathbf{m}_m^j \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{m}_1^j & 0 & \dots & 0 \\ 0 & \mathbf{m}_2^j & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{m}_m^j \end{bmatrix}}_{Q^j} \underbrace{\begin{bmatrix} \zeta_1^j \\ \zeta_2^j \\ \vdots \\ \zeta_m^j \end{bmatrix}}_{\zeta^j} = PM^j \quad (12.8)$$

Una soluzione iterativa che si adotta spesso in casi come questo consiste nell'alternare la soluzione di due problemi: in un passo si stimano ζ_i^j dati P ed M , nel passo successivo si stimano P ed M dati ζ_i^j . Si itera fino a convergenza.

La procedura è la seguente:

Algoritmo 12.1 RICOSTRUZIONE PROIETTIVA

Input: Punti corrispondenti nelle immagini W , con $\zeta_i^j = 1$

Output: Ricostruzione 3D dei punti dati P, M

- (i) Normalizza W in modo che $\|W\|_F = 1$;
 - (ii) Ottieni P ed M dalla SVD di W ;
 - (iii) Se $\|W - PM\|_F^2$ è abbastanza piccolo allora termina;
 - (iv) Ricava ζ^j da $Q^j \zeta^j = PM^j, \forall j = 1 \dots n$;
 - (v) Aggiorna W ;
 - (vi) Ripeti dal passo (i).
-

Il passo (i) è necessario per impedire la convergenza alla soluzione banale $\zeta_i^j = 0$. L'algoritmo è implementato nella funzione `prec`.

Questa è una tecnica iterativa che tuttavia si dimostra veloce e non richiede una inizializzazione informata. Tuttavia, bisogna sottolineare che non vi è garanzia di convergenza. Un metodo iterativo a convergenza garantita è stato proposto in [Mahamud *e al.*, 2001].

12.1.2 Promozione euclidea

La ricostruzione proiettiva differisce da quella euclidea per una collinearazione sconosciuta dello spazio proiettivo T , che può essere vista come un appropriato cambio di base. Il problema è di calcolare *quella* T che la trasforma la ricostruzione da proiettiva in euclidea, sfruttando tutti i vincoli disponibili. Le assunzioni sono: che siano disponibili abbastanza fotogrammi (o fotocamere) e che i parametri intrinseci, pur sconosciuti, siano *costanti*. Potremo precisare solo nel seguito *quante* fotocamere sono necessarie.

Dalla ricostruzione proiettiva otteniamo un insieme di MPP $\{P_i^P\}$ tali che:

$$P_0^P = [I \mid \mathbf{0}]; \quad P_i^P = [Q_i \mid \mathbf{q}_i]. \quad (12.9)$$

Stiamo cercando la matrice 4×4 non singolare T , che trasforma la ricostruzione proiettiva in quella euclidea. Se $\{\mathbf{M}^j\}$ è la struttura euclidea cercata, \mathbf{T} deve essere tale che: $\mathbf{m}_i^j = \mathbf{P}_i^P T T^{-1} \mathbf{M}^j$, perciò

$$P_i^e \simeq P_i^P T, \quad (12.10)$$

dove il simbolo \simeq significa, come al solito, “uguale a meno di un fattore di scala.”

Possiamo fissare arbitrariamente la trasformazione rigida, scegliendo la prima MPP euclidea: $P_0^e = K[I \mid \mathbf{0}]$, mentre le successive avranno, in generale, la forma

$$P_i^e = K[R_i \mid \mathbf{t}_i] \quad i > 0. \quad (12.11)$$

Con questa scelta, è facile vedere che $\mathbf{P}_0^e = P_0^p T$ e ciò implica che:

$$T = \begin{bmatrix} K & \mathbf{0} \\ \mathbf{r}^\top & s \end{bmatrix} \quad (12.12)$$

dove \mathbf{r}^\top è un vettore arbitrario di tre elementi $[r_1 \ r_2 \ r_3]$. Con questa parametrizzazione T è chiaramente non singolare, ed, essendo definita a meno di un fattore di scala, dipende da otto parametri (fissiamo $s = 1$).

Sostituendo la (12.9) nella (12.10) otteniamo:

$$P_i^e \simeq P_i^p T = [Q_i K + \mathbf{q}_i \mathbf{r}^\top \mid \mathbf{q}_i], \quad (12.13)$$

e dalla (12.11)

$$P_i^e = K[R_i \mid \mathbf{t}_i] = [K R_i \mid K \mathbf{t}_i], \quad (12.14)$$

quindi

$$Q_i K + \mathbf{q}_i \mathbf{r}^\top \simeq K R_i. \quad (12.15)$$

Questa è l'equazione base, che lega le incognite K (cinque parametri) ed \mathbf{r} (tre parametri) ai dati disponibili \mathbf{Q}_i e \mathbf{q}_i . R non si conosce, ma deve essere una matrice di rotazione.

Il metodo proposto in [Heyden e Åström, 1996] si basa sulla (12.15), che può essere riscritta come:

$$P_i^p \begin{bmatrix} K \\ \mathbf{r}^\top \end{bmatrix} \simeq K R_i. \quad (12.16)$$

dato che $R_i R_i^\top = I$ segue che:

$$\begin{aligned} P_i^p \begin{bmatrix} K \\ \mathbf{r}^\top \end{bmatrix} \begin{bmatrix} K \\ \mathbf{r}^\top \end{bmatrix}^\top P_i^{p\top} &= P_i^p \begin{bmatrix} K K^\top & K \mathbf{r} \\ \mathbf{r}^\top K^\top & \mathbf{r}^\top \mathbf{r} \end{bmatrix} P_i^{p\top} \simeq \\ &\simeq K R_i R_i^\top K^\top = K K^\top. \end{aligned} \quad (12.17)$$

I vincoli espressi nella (12.17) sono chiamati i vincoli di Kruppa. Si osserva che la (12.17) contiene cinque equazioni, perché le matrici di entrambi i membri sono simmetriche e il fattore di scala arbitrario riduce di uno il numero di equazioni. Quindi, ogni matrice della fotocamera, esclusa la prima, fornisce cinque equazioni nelle otto incognite

$\alpha_u, \alpha_v, \gamma, u_0, v_0, r_1, r_2, r_3$. Un'unica soluzione si ha non appena si hanno a disposizione tre fotocamere. Se il fattore di scala incognito è introdotto esplicitamente possiamo riscrivere la (12.17) come:

$$0 = f_i(K, \mathbf{r}, \lambda_i) = \lambda_i^2 K K^\top - P_i^p \begin{bmatrix} K K^\top & K \mathbf{r} \\ \mathbf{r}^\top K^\top & \mathbf{r}^\top \mathbf{r} \end{bmatrix} P_i^{p\top}. \quad (12.18)$$

Quindi, tre fotocamere sono sufficienti (e necessarie) poiché forniscono 12 equazioni in 10 incognite (8 sono comuni, due sono i λ che sono diversi per ogni fotocamera). Il sistema di equazioni non lineari si risolve ai minimi quadrati con un metodo numerico (per esempio Gauss-Newton).

12.2 Autocalibrazione

La seconda strada per la ricostruzione euclidea passa attraverso il calcolo esplicito dei parametri intrinseci, che ci riconduce dunque al caso calibrato.

Date le corrispondenze tra punti coniugati, senza conoscere nulla della fotocamera, è possibile ricavare la matrice F , la quale codifica tutta la informazione sulle fotocamere che possiamo ottenere dalle sole corrispondenze. Sfortunatamente si dimostra che non è possibile estrarre da F i parametri intrinseci ed estrinseci delle fotocamere (che pure vi sono contenuti). Questa limitazione si riferisce al caso di due viste. Avendo a disposizione una terna di immagini prese con la stessa fotocamera, di cui non sappiamo nulla tranne che i suoi parametri intrinseci sono costanti, è possibile effettuare la cosiddetta *autocalibrazione*, ovvero calcolarne i parametri intrinseci come risultato di un processo di minimizzazione.

12.2.1 Matrice fondamentale

La matrice fondamentale è definita dalla (6.16) che riportiamo qui per comodità del lettore:

$$\mathbf{m}'^\top F \mathbf{m} = 0. \quad (12.19)$$

Essa dice che il punto corrispondente di \mathbf{m} nella seconda immagine deve giacere sulla sua linea epipolare $F\mathbf{m}$.

Dato che $\det([\mathbf{e}']_\times) = 0$, si ha che $\det(F) = 0$. In particolare si vede che il rango di F è pari a 2. Inoltre, la matrice F è definita a meno di un fattore di scala, in quanto se essa viene moltiplicata per uno scalare arbitrario, l'equazione (6.16) risulta comunque soddisfatta (possiamo pensare di fissare arbitrariamente ad 1 l'elemento in basso a destra della matrice). In principio una matrice 3×3 ha nove gradi di libertà; ogni

vincolo rimuove un grado di libertà, per cui una matrice fondamentale ha solo 7 gradi di libertà. Ciò significa che vi sono solo 7 parametri.

Si osservi che, trasponendo la (6.16) si ottiene la relazione simmetrica dalla seconda immagine alla prima immagine:

$$\mathbf{m}^\top F^\top \mathbf{m}' = 0. \quad (12.20)$$

Poichè l'epipolo appartiene a tutte le rette epipolari, deve valere $\mathbf{e}'^\top F \mathbf{m} = 0$ per ogni \mathbf{m} , e dunque deve essere

$$\mathbf{e}'^\top F = \mathbf{0} \quad (12.21)$$

Si poteva anche dimostrare algebricamente dal fatto che $[\mathbf{e}']_\times \mathbf{e}' = \mathbf{0}$.

Analogamente

$$\mathbf{e}^\top F^\top = \mathbf{0}. \quad (12.22)$$

La matrice essenziale e la matrice fondamentale sono in relazione, dal momento che entrambe codificano il movimento rigido tra due viste. La prima mette in relazione le coordinate *normalizzate* dei punti coniugati, mentre la seconda mette in relazione le coordinate *pixel* dei punti coniugati. È facile verificare che:

$$F = K'^{-\top} E K^{-1}. \quad (12.23)$$

12.2.1.1 Calcolo della matrice fondamentale

La matrice fondamentale si calcola a partire dalle corrispondenze di almeno otto punti, utilizzando il metodo descritto per la matrice essenziale nel § 9.2.2. Si tratta di sostituire le coordinate normalizzate con le coordinate pixel e di cambiare il tipo di vincolo che viene forzato all'uscita. Infatti, a differenza della matrice essenziale, caratterizzata dal teorema 9.2, l'unica proprietà che la F deve avere è di essere singolare.

Lavorando con coordinate pixel, però, sorge un problema di condizionamento del sistema lineare, che in coordinate normalizzate è meno sentito.

Standardizzazione dei dati. L'algoritmo degli otto punti è stato criticato per essere troppo sensibile al rumore, e quindi poco utile nelle applicazioni pratiche. Di conseguenza, sono stati proposti molti algoritmi iterativi non lineari per il calcolo della matrice fondamentale, tutti molto più complicati (vedi [Zhang, 1998] per ulteriori informazioni). Tuttavia [Hartley, 1995] ha dimostrato che l'instabilità del metodo è dovuta principalmente ad un problema di malcondizionamento, piuttosto che alla sua natura lineare. Egli osservò, infatti, che impiegando coordinate

pixel omogenee, molto probabilmente si ottiene un sistema di equazioni lineari mal condizionato, dato che le prime due componenti hanno grandezze molto diverse dalla terza (in un’immagine 256×256 , un tipico punto immagine è $[128, 128, 1]$).

Applicando una semplice *standardizzazione*[†] delle coordinate dei punti, il numero di condizionamento diventa più piccolo ed i risultati possono essere confrontati con quelli degli algoritmi iterativi. La procedura di standardizzazione è la seguente: i punti vengono traslati in modo tale che il loro centroide coincida con l’origine e poi vengono scalati affinché la distanza media dall’origine sia pari a $\sqrt{2}$.

Siano date T e T' le trasformazioni risultanti nelle due immagini e $\bar{\mathbf{m}} = T\mathbf{m}$, $\bar{\mathbf{m}}' = T'\mathbf{m}'$ i punti trasformati. Usando $\bar{\mathbf{m}}$ e $\bar{\mathbf{m}}'$ nell’algoritmo degli otto punti, otteniamo una matrice fondamentale \bar{F} che può essere messa in relazione con quella originale per mezzo di $F = T'^\top \bar{F} T$, come si può facilmente dimostrare.

La standardizzazione non è utile solo nel calcolo di F ma in tutti quei casi in cui si applicano algoritmi lineari in cui la matrice dei coefficienti contiene valori in pixel, quindi, per esempio anche nel caso della calibrazione della fotocamera.

La funzione MATLAB `fm` calcola la matrice fondamentale a partire da corrispondenze di punti, con la standardizzazione. Si noti che questa serve anche per calcolare E all’occorrenza. Infatti, se sappiamo calcolare F , siamo anche in grado di calcolare E – purché siano noti i parametri intrinseci – sia usando (12.23), sia convertendo le coordinate pixel in coordinate normalizzate. La standardizzazione è affidata alla funzione `precond2`.

Residuo geometrico Anche per la matrice fondamentale, la stima ottenuta dall’algoritmo lineare può essere raffinata mediante la minimizzazione di un opportuno residuo geometrico, pari alla somma delle distanze tra punti e rette epipolari coniugate [Luong e Faugeras, 1996].

Siano $\mathbf{m}_i \leftrightarrow \mathbf{m}'_i$ i punti corrispondenti. La matrice fondamentale che li lega si ottiene risolvendo il seguente problema di minimi quadrati non lineare:

$$\min_F \sum_j d(F\mathbf{m}_i, \mathbf{m}'_i)^2 + d(F^\top \mathbf{m}'_i, \mathbf{m}_i)^2 \quad (12.24)$$

[†] Originalmente chiamato “normalizzazione” da Hartley, abbiamo preferito il termine “standardizzazione” per non generare confusione con le coordinate normalizzate.

dove $d()$ denota la distanza punto retta nel piano cartesiano. Si noti che F deve venire opportunamente parametrizzata per garantirne la singolarità.

Risultati analoghi si ottengono con il residuo di Sampson [Luong e Faugeras, 1996].

12.2.2 Metodo di Mendonça e Cipolla

Abbiamo visto che la matrice fondamentale F ha sette gradi di libertà, mentre la matrice essenziale E ne ha cinque, poiché deve soddisfare due vincoli in più, che sono quelli derivanti dall'uguaglianza dei due valori singolari (teorema 9.2). Inoltre sappiamo che E ed F sono legate tramite i parametri intrinseci K dalla (12.23). Questo vuol dire che i due vincoli in più sono disponibili per il calcolo dei parametri intrinseci.

Questa è un'interpretazione algebrica del cosiddetto *vincolo di rigidità* chiamato così poiché per ogni matrice fondamentale F esistono due matrici di parametri intrinseci K e K' ed un moto rigido rappresentato da t e R tali che $F = K'^{-\top}([\mathbf{t}] \times R)K^{-1}$.

Con due soli vincoli a disposizione non è possibile ricavare tutti i parametri intrinseci ([Hartley, 1992] per esempio mostra come ricavare le focali delle due fotocamere). Servono più di due viste, per accumulare vincoli. Se i parametri intrinseci sono costanti, si dimostra che ne bastano tre. Infatti ci sono cinque incognite, ogni coppia di viste fornisce due equazioni e ci sono tre coppie di viste indipendenti: 1-2, 1-3 e 2-3.

Il modo di scrivere i vincoli può variare. Mendonça e Cipolla [1999], che hanno per primi proposto il metodo, usano direttamente il teorema 9.2. Una formulazione alternativa, che evita la SVD si basa sulla seguente:

Proposizione 12.1 ([Huang e Faugeras, 1989]) *La condizione che la matrice E abbia un valore singolare pari a zero e due valori singolari di ugual valore, diversi da zero, è equivalente a:*

$$\det(E) = 0 \quad \text{e} \quad \text{tr}((EE^\top))^2 - 2 \text{tr}((EE^\top)^2) = 0. \quad (12.25)$$

La seconda condizione, equivalente all'uguaglianza dei due valori singolari, può essere scomposta in due relazioni polinomiali indipendenti [Luong e Faugeras, 1996], ecco perché conta per due vincoli.

Il metodo di Mendonça e Cipolla impiega una funzione di costo che prende come argomenti i parametri intrinseci (incogniti), come parametri le matrici fondamentali (calcolate dalle corrispondenze di punti) e

restituisce un valore positivo proporzionale alla violazione del vincolo di rigidità. I parametri intrinseci cercati sono quelli che soddisfano il vincolo, o che lo “violano di meno”, ovvero quelli che minimizzano la funzione costo. In formule, sia F_{ij} la matrice fondamentale relativa alle viste i e j , e sia K la matrice dei parametri intrinseci. La funzione di costo (minimi quadrati) è:

$$\varepsilon(K) = \sum_{i=1}^m \sum_{j=i+1}^m w_{ij} (\text{tr}^2(E_{ij}E_{ij}^\top) - 2 \text{tr}(E_{ij}E_{ij}^\top)^2)^2, \quad (12.26)$$

dove $E_{ij} = K^\top F_{ij} K$ e w_{ij} sono pesi normalizzati che tengono conto della affidabilità con cui ciascuna F_{ij} è stata calcolata (se disponibile). La funzione costo viene minimizzata usando un algoritmo iterativo, dunque la convergenza non è garantita per qualunque valore iniziale. Nella pratica, tuttavia, l’algoritmo converge spesso al valore vero, a partire da una stima ragionevole del valore iniziale.

L’algoritmo è implementato nella funzione MATLAB `autocal`.

Una volta che i parametri intrinseci sono noti, le matrici fondamentali che sono state calcolate vengono promosse a matrici essenziali con la (12.23) e si procede quindi come nel caso calibrato (capitolo 9).

12.2.3 Ricostruzione incrementale

Un metodo alternativo di procedere, più usato in pratica, applica il metodo appena delineato solo ad un sottoinsieme di viste e poi procede incrementalmente aggiungendo fotocamere e punti.

Date le viste disponibili, se ne seleziona un *sottoinsieme* di cardinalità sufficiente a garantire l’autocalibrazione con una certa resilienza al rumore (una decina nel caso di parametri intrinseci costanti). Si calcolano la matrici fondamentali tra tutte le coppie di viste del sottoinsieme (ove le corrispondenze lo consentano) e si procede all’autocalibrazione ed alla ricostruzione come nel capitolo 9. Quindi si procede aggiungendo una alla volta le viste che sono rimaste nel seguente modo: si prende una vista, si effettua la calibrazione della fotocamera rispetto ai punti 3D facenti parte della struttura sin qui ricostruita e quindi si aggiungono alla struttura eventuali nuovi punti per triangolazione con la nuova vista. Si procede fino a quando tutte le viste sono state aggiunte. Un passo di *bundle adjustment* finale è obbligatorio, ma spesso se ne fanno anche nei passi intermedi, se la sequenza è lunga, per contenere la deriva prima che diventi troppo severa.

La figura 12.1 mostra un tipico risultato di ricostruzione non calibrata ottenuto con l'approccio incrementale.

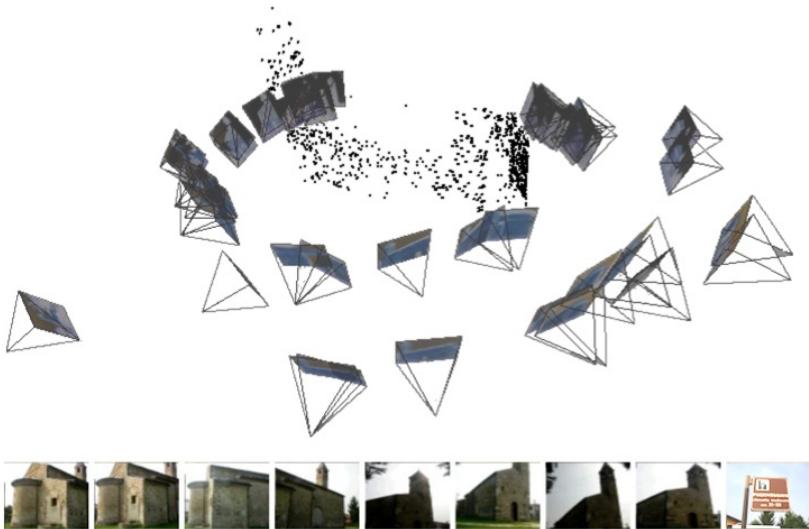


Fig. 12.1. Risultato della ricostruzione incrementale su un insieme di 54 immagini non calibrate della chiesa di Pozzoveggiani (PD). In alto si vede l'insieme di punti 3D ricostruiti e le fotocamere. Sotto, alcune delle immagini impiegate.

12.3 Fattorizzazione di Tomasi-Kanade

Vediamo ora, in appendice al capitolo, la tecnica di fattorizzazione di [Tomasi e Kanade, 1992] che è all'origine di quella presentata nel § 12.1.1.

Per la prima volta consideriamo un modello di fotocamera diverso da quella prospettica, infatti questo metodo assume che valga l'approssimazione di **fotocamera affine** (proiezione ortogonale invece che prospettica). In questo caso si può ottenere direttamente la struttura della scena ed il moto della fotocamera tramite fattorizzazione SVD di una matrice che contiene le coordinate dei punti coniugati (ottenuti da tracciamento dei punti salienti lungo la sequenza). Se i parametri intrinseci della fotocamera sono sconosciuti, la struttura (ed il moto) possono essere calcolati solo a meno di una trasformazione affine incognita. Se invece i parametri intrinseci sono noti, come nel lavoro originale di [Tomasi e Kanade, 1992] (la fotocamera si dice allora **ortografica**), è pos-

sibile recuperare struttura e moto a meno di una trasformazione rigida (corrispondente all'arbitrarietà nel fissare il sistema di riferimento 3D).

La fotocamera affine più generale[†] ha la seguente forma:

$$\begin{aligned} P \triangleq & \left[\begin{array}{ccc} \alpha_u & \gamma & 0 \\ 0 & \alpha_v & 0 \\ 0 & 0 & 1 \end{array} \right] \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right] \left[\begin{array}{cc} R & \mathbf{t} \\ \mathbf{0} & 1 \end{array} \right] = \\ & \left[\begin{array}{ccc} \alpha_u & \gamma & 0 \\ 0 & \alpha_v & 0 \\ 0 & 0 & 1 \end{array} \right] \left[\begin{array}{cc} \mathbf{r}_1 & t_1 \\ \mathbf{r}_2 & t_2 \\ \mathbf{0} & 1 \end{array} \right] \triangleq \left[\begin{array}{cc} A_{2 \times 3} & \mathbf{s}_{2 \times 1} \\ \mathbf{0} & 1 \end{array} \right] \quad (12.27) \end{aligned}$$

Si tratta della generica fotocamera prospettica nella quale si sostituisce la matrice centrale, che rappresenta la proiezione prospettica, con una proiezione ortogonale, ed in cui il punto principale nella matrice dei parametri intrinseci non è definito.

In coordinate cartesiane la proiezione affine si scrive:

$$\tilde{\mathbf{m}} = A\tilde{\mathbf{M}} + \mathbf{s} \quad (12.28)$$

Il punto \mathbf{s} è l'immagine dell'origine delle coordinate mondo.

Consideriamo un insieme di n punti 3D, visti da m fotocamere con matrici $\{P_i\}_{i=1\dots m}$. Siano \mathbf{m}_i^j le coordinate della proiezione del j -esimo punto nella i -esima fotocamera. Lo scopo della ricostruzione è stimare le MPP delle fotocamere $\{A_i, \mathbf{s}_i\}$ ed i punti 3D \mathbf{M}^j t.c. valga:

$$\tilde{\mathbf{m}}_i^j = A_i \tilde{\mathbf{M}}^j + \mathbf{s}_i \quad (12.29)$$

Consideriamo le coordinate “centralizzate” ottenute sottraendo il centroide $\langle \tilde{\mathbf{m}}_i \rangle = \frac{1}{n} \sum_{j=1}^n \tilde{\mathbf{m}}_i^j$ in ciascuna immagine:

$$\bar{\tilde{\mathbf{m}}}_i^j = \tilde{\mathbf{m}}_i^j - \langle \tilde{\mathbf{m}}_i \rangle \quad (12.30)$$

(questo passaggio implica che tutti gli n punti siano visibili in tutte le m immagini).

Sceglieremo il sistema di riferimento 3D nel centroide dei punti, in modo che: $\langle \tilde{\mathbf{M}}^j \rangle = 0$. Tenendo conto di questo, si sostituisca la (12.29) nella (12.30), ottenendo così di eliminare il vettore \mathbf{s}_i nella (12.30), che diventa

$$\bar{\tilde{\mathbf{m}}}_i^j = A_i \bar{\tilde{\mathbf{M}}}^j \quad (12.31)$$

In questo modo si ricava immediatamente che $\mathbf{s}_i = \langle \mathbf{m}_i \rangle$. Riscriviamo

[†] Con $\gamma = 0$ si ha la fotocamera prospettica debole (*weak perspective*).

le equazioni in forma matriciale:

$$\underbrace{\begin{bmatrix} \bar{\mathbf{m}}_1^1, & \bar{\mathbf{m}}_2^1, & \dots & \bar{\mathbf{m}}_n^1 \\ \bar{\mathbf{m}}_1^2, & \bar{\mathbf{m}}_2^2, & \dots & \bar{\mathbf{m}}_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{m}}_1^m, & \bar{\mathbf{m}}_2^m, & \dots & \bar{\mathbf{m}}_n^m \end{bmatrix}}_{\text{misure } W} = \underbrace{\begin{bmatrix} A^1, \\ A^2, \\ \vdots \\ A^m \end{bmatrix}}_{\text{moto } A} \underbrace{\begin{bmatrix} \bar{\mathbf{M}}_1 & \bar{\mathbf{M}}_2 & \dots & \bar{\mathbf{M}}_n \end{bmatrix}}_{\text{struttura } M}. \quad (12.32)$$

La matrice delle misure W rappresenta la posizione di n punti salienti tracciati lungo m fotogrammi. È una matrice $2m \times n$ che ha rango al più 3, essendo il prodotto di una matrice A $2m \times 3$ (moto) ed una M $3 \times n$ (struttura). Questa osservazione è la chiave del metodo di fattorizzazione di Tomasi e Kanade. Infatti, per recuperare A ed M da W si fattorizza quest'ultima nel prodotto di due matrici di rango tre usando la SVD. Sia

$$W_{2m \times n} = U_{2m \times 2m} D_{2m \times n} V_{n \times n}^\top \quad (12.33)$$

la decomposizione ai valori singolari di W . In assenza di rumore solo i primi tre valori singolari sarebbero diversi da zero; nella pratica invece il rango di W è solo approssimativamente tre. Considero quindi i primi tre valori singolari, ottenendo

$$\hat{W}_{2m \times n} = U_{2m \times 3} D_{3 \times 3} V_{n \times 3}^\top \quad (12.34)$$

dove \hat{W} è la miglior (in norma di Frobenius) approssimazione di rango tre di W . La fattorizzazione cercata si ottiene ponendo

$$A = U_{2m \times 3} \quad M = D_{3 \times 3} V_{n \times 3}^\top \quad (12.35)$$

La fattorizzazione non è unica: la matrice $D_{3 \times 3}$ poteva essere ripartita arbitrariamente tra le due matrici, infatti si può inserire nella fattorizzazione una arbitraria matrice T di rango tre senza alterare il risultato:

$$\hat{W} = (AT)(T^{-1}M) \quad (12.36)$$

Si ha una ambiguità affine sulla ricostruzione così ottenuta.

Se invece gli intrinseci sono noti, la fotocamera è ortografica, ovvero, lavorando in coordinate normalizzate,

$$P \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 & t_1 \\ \mathbf{r}_2 & t_2 \\ \mathbf{0} & 1 \end{bmatrix} \quad (12.37)$$

Dunque il centroide $\langle \mathbf{m}_i \rangle$ fornisce la componente di traslazione (in x ed

y), e le righe di A sono due vettori ortogonali, condizione che può essere sfruttata per fissare l'ambiguità.

Si noti comunque che la traslazione di ciascuna fotocamera lungo il proprio asse ottico ottico z non può essere ricavata.

Esercizi ed approfondimenti

- 12.1 Dalla (12.22) sappiamo che: $\mathbf{e}_1^\top F^\top = \mathbf{0}$, o, equivalentemente, $F\mathbf{e}_1 = \mathbf{0}$. Questo significa che l'epipolo \mathbf{e}_1 appartiene al *nucleo* di F , e si può ricavare facilmente, per esempio, dalla fattorizzazione SVD. Analogamente si ricava \mathbf{e}_2 da $F^\top \mathbf{e}_2 = \mathbf{0}$. Si veda la funzione `epipole`.
- 12.2 Come parametrizzare la matrice fondamentale con sette parametri che tengano conto della indeterminazione di scala e della singolarità?

Bibliografia

- Hartley R. I. (1992). Estimation of relative camera position for uncalibrated cameras. In *Proceedings of the European Conference on Computer Vision*, pp. 579–587, Santa Margherita L.
- Hartley R. I. (1995). In defence of the 8-point algorithm. In *Proceedings of the International Conference on Computer Vision*, pp. 1064–1071, Washington, DC, USA. IEEE Computer Society.
- Heyden A.; Åström K. (1996). Euclidean reconstruction from constant intrinsic parameters. In *Proceedings of the International Conference on Pattern Recognition*, pp. 339–343, Vienna.
- Huang T.; Faugeras O. (1989). Some properties of the E matrix in two-view motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**(12), 1310–1312.
- Luong Q.-T.; Faugeras O. D. (1996). The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, **17**, 43–75.
- Mahamud S.; Hebert M.; Omori Y.; Ponce J. (2001). Provably-convergent iterative methods for projective structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. I:1018–1025.
- Mendonça P.; Cipolla R. (1999). A simple technique for self-calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. I:500–505.

- Robert L.; Faugeras O. (1995). Relative 3-D positioning and 3-D convex hull computation from a weakly calibrated stereo pair. *Image and Vision Computing*, **13**(3), 189–197.
- Sturm P.; Triggs B. (1996). A factorization based algorithm for multi-image projective structure and motion. In *Proceedings of the European Conference on Computer Vision*, pp. 709–720, Cambridge, UK.
- Tomasi C.; Kanade T. (1992). Shape and motion from image streams under orthography – a factorization method. *International Journal of Computer Vision*, **9**(2), 137–154.
- Zhang Z. (1998). Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, **27**(2), 161–195.

13

Scena planare: omografie

Mentre nel caso più generale i punti corrispondenti in due viste sono legati dalla matrice fondamentale, ci sono casi di interesse pratico nei quali le corrispondenze di punti tra due viste sono codificate da una proiettività di \mathbb{P}_2 , o omografia o collineazione. Questo accade quando i punti osservati giacciono su di un piano nello spazio.

13.1 Omografia indotta da un piano

Iniziamo stabilendo che l'applicazione tra un piano Π nello spazio e la sua immagine prospettica è una omografia.

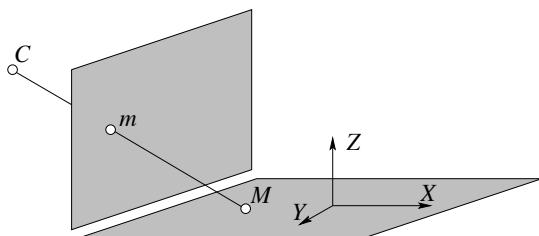


Fig. 13.1. L'applicazione che porta un piano Π di \mathbb{P}_3 sul piano immagine è una omografia.

Lo si vede facilmente se si sceglie il sistema di riferimento mondo in modo che il Π abbia equazione $Z = 0$ (si veda la figura 13.1). Espandendo l'equazione di proiezione prospettica, per un punto appartenente

al piano, si ottiene

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & p_{3,4} \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,4} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (13.1)$$

Quindi, l'applicazione da Π all'immagine è rappresentata da una matrice 3×3 non singolare, ovvero una omografia.

Quindi osserviamo che le proiezioni dei punti di Π in due immagini sono legati tra loro da una omografia. Infatti, abbiamo una omografia tra Π e l'immagine di sinistra, ed analogamente una omografia tra Π e l'immagine di destra. Componendo l'inversa della prima con la seconda otteniamo una omografia dall'immagine sinistra all'immagine destra (si veda la figura 13.2).

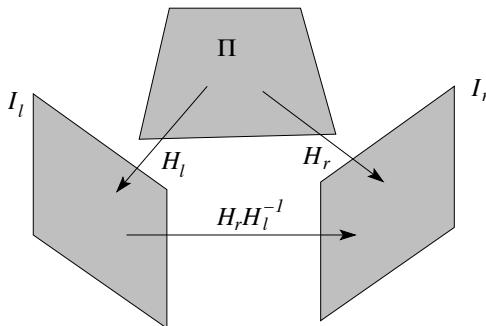


Fig. 13.2. Il piano Π induce una omografia di \mathbb{P}_2 tra i due piani immagine.

Diremo quindi che Π *induce* una omografia tra le due viste, rappresentata dalla matrice H_Π , nel senso che punti corrispondenti nelle due immagini vengono associati tramite H_Π . In formule:

$$\mathbf{m}' \simeq H_\Pi \mathbf{m} \quad \text{se } \mathbf{M} \in \Pi. \quad (13.2)$$

La matrice H_Π è 3×3 non singolare, ed essendo definita a meno di fattore di scala ha 8 gradi di libertà.

Vedremo ora in quali casi due immagini di una scena sono collegate da omografie. Dovremo ripartire dalla geometria epipolare e specializzarla.

Se prendiamo il sistema di riferimento della prima fotocamera come sistema di riferimento mondo, possiamo scrivere le seguenti due MPP:

$$P = K[I|\mathbf{0}] = [K|\mathbf{0}] \quad \text{e} \quad P' = K'[R|\mathbf{t}] \quad (13.3)$$

Sostituendo queste MPP nella equazione della retta epipolare di \mathbf{m} (6.11) otteniamo

$$\mathbf{m}' \simeq \lambda K' R K^{-1} \mathbf{m} + K' \mathbf{t}. \quad (13.4)$$

con

$$\mathbf{e}' = K' [R | \mathbf{t}] \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = K' \mathbf{t}. \quad (13.5)$$

In generale, come già sappiamo, due punti \mathbf{m} e \mathbf{m}' che sono la proiezione del punto 3D \mathbf{M} sulla prima e sulla seconda fotocamera, rispettivamente, sono messi in relazione dalla (13.4). In due casi speciali, tuttavia, i punti nelle due immagini sono legati da una omografia:

- (i) scena generale, la fotocamera ruota attorno al centro ottico,
- (ii) scena planare, moto generale della fotocamera.

Moto rotazionale. Se la fotocamera sta ruotando, allora $\mathbf{t} = \mathbf{0}$ ed abbiamo:

$$\mathbf{m}' \simeq K' R K^{-1} \mathbf{m}. \quad (13.6)$$

$H_\infty = K' R K^{-1}$ è una omografia, e non dipende dalla struttura 3D (si noti l'analogia con il campo di moto rotazionale del §. 10.1.1).

Scena planare. Se il punto 3D \mathbf{M} giace su un piano Π allora l'equazione (13.4) può essere specializzata.

Sia \mathbf{M}' la rappresentazione di \mathbf{M} nel riferimento della seconda fotocamera: $\mathbf{M}' = G\mathbf{M}$. Usando l'equazione del piano $\mathbf{n}^\top \tilde{\mathbf{M}} = d$, dove d è la distanza del piano dall'origine ed \mathbf{n} è la sua normale, e la proiezione prospettica $\mathbf{m} \simeq K\tilde{\mathbf{M}}$ e $\mathbf{m}' \simeq K'\tilde{\mathbf{M}'}$, dopo qualche passaggio si ricava

$$\mathbf{m}' \simeq K' \left(R + \frac{\mathbf{t} \mathbf{n}^\top}{d} \right) K^{-1} \mathbf{m}. \quad (13.7)$$

Quest'ultima equazione afferma che vi è una proiettività indotta dal piano Π tra le due viste, definita dalla matrice

$$H_\Pi = K' \left(R + \frac{\mathbf{t} \mathbf{n}^\top}{d} \right) K^{-1}. \quad (13.8)$$

Se Π si avvicina al piano all'infinito allora $d \rightarrow \infty$ e dunque al limite si ottiene H_∞ , la matrice dell'omografia indotta dal piano all'infinito. Questa omografia ha un doppio ruolo:

- come tutte omografie indotte da un piano associa le proiezioni dei punti di quel piano tra le due immagini. In particolare H_∞ associa le proiezioni dei punti che giacciono sul piano all'infinito, p.es. i punti di fuga;
- inoltre H_∞ associa le proiezioni di *tutti* i punti della scena (non importa il piano a cui appartengono) tra le due immagini quando la fotocamera compie un moto puramente rotazionale.

13.1.1 Calcolo dell'omografia (DLT)

Sono dati n punti corrispondenti $\mathbf{m}_i \leftrightarrow \mathbf{m}'_i$, e si vuole determinare la matrice dell'omografia H tale per cui:

$$\mathbf{m}'_i \simeq H\mathbf{m}_i \quad i = 1 \dots n \quad (13.9)$$

Sfruttando il prodotto esterno per eliminare il fattore moltiplicativo, l'equazione si riscrive:

$$\mathbf{m}'_i \times H\mathbf{m}_i = \mathbf{0} \quad (13.10)$$

Sulla falsa riga della derivazione fatta per la calibrazione (§ 4.2) si ottiene:

$$\begin{aligned} \mathbf{m}'_i \times H\mathbf{m}_i = \mathbf{0} &\iff [\mathbf{m}'_i]_\times H\mathbf{m}_i = \mathbf{0} \iff \\ \text{vec}([\mathbf{m}'_i]_\times H\mathbf{m}_i) = \mathbf{0} &\iff (\mathbf{m}_i^\top \otimes [\mathbf{m}'_i]_\times) \text{vec}(H) = \mathbf{0} \end{aligned}$$

ed analogamente si conclude che la matrice $(\mathbf{m}_i^\top \otimes [\mathbf{m}'_i]_\times)$ ha rango due, quindi solo due equazioni su tre sono linearmente indipendenti. Per n punti otteniamo un sistema di $2n$ equazioni lineari omogenee, che possiamo scrivere come

$$A \text{vec}(H) = \mathbf{0} \quad (13.11)$$

dove A è la matrice $2n \times 9$ dei coefficienti ottenuta impilando due equazioni per ogni corrispondenza, mentre il vettore delle incognite $\text{vec}(H)$ contiene i nove elementi di H letti per colonne. Dunque quattro punti in posizione generale† determinano una matrice dei coefficienti A di rango otto (una equazione delle nove è superflua) il cui nucleo unidimensionale è la soluzione cercata a meno di un fattore di scala. Una omografia, dunque, è determinata dalla sua azione su quattro punti, come illustrato in figura 13.3.

Per $n > 4$ punti, la soluzione che minimizza ai minimi quadrati è l'autovettore associato al minimo autovalore di $A^\top A$ e si calcola tramite la SVD di A .

† Non ve ne devono essere tre allineati.

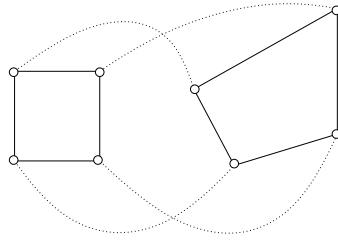


Fig. 13.3. Quattro corrispondenze di punti determinano una omografia.

La funzione MATLAB `h2m` implementa questo metodo.

Come si usa per la matrice fondamentale, anche nel calcolo di H può essere utile riscalare i dati, in modo che il problema si presenti meglio condizionato. Sia $\bar{\mathbf{m}} = T\mathbf{m}$ e $\bar{\mathbf{m}}' = T'\mathbf{m}'$. Partendo dalle corrispondenze tra i punti riscalati $\bar{\mathbf{m}}$ e $\bar{\mathbf{m}}'$ trovo una omografia del piano \bar{H} che è legata a quella cercata da $\bar{H} = T'HT^{-1}$, come si può facilmente vedere.

Anche per il calcolo della omografia possiamo scrivere un residuo geometrico la cui minimizzazione sortisce una stima più accurata e maggiormente significativa. Si tratta dell'errore di trasferimento simmetrico:

$$\min_H \sum_i d(H\mathbf{m}_i, \mathbf{m}'_i)^2 + d(H^{-1}\mathbf{m}'_i, \mathbf{m}_i)^2 \quad (13.12)$$

dove $d()$ è la distanza tra due punti nel piano cartesiano.

13.1.2 Omografie compatibili

Sostituendo la (13.3) nella (6.14) si ottiene:

$$F = [\mathbf{e}']_\times H_\infty. \quad (13.13)$$

con $H_\infty = K'RK^{-1}$ e $\mathbf{e}' = K'\mathbf{t}$.

Si nota una somiglianza con la fattorizzazione $E = [\mathbf{t}]_\times R$, poiché \mathbf{e}' dipende solo dalla traslazione e H_∞ dipende solo dalla rotazione. Sfortunatamente, la fattorizzazione non è unica, e ciò rende impossibile la determinazione diretta di H_∞ da F . Infatti, se una matrice A soddisfa $F = [\mathbf{e}']_\times A$, allora anche $A + \mathbf{e}'\mathbf{v}^\top$, per ogni vettore \mathbf{v} , va bene nella fattorizzazione, dato che

$$[\mathbf{e}']_\times (A + \mathbf{e}'\mathbf{v}^\top) = [\mathbf{e}']_\times A + [\mathbf{e}']_\times \mathbf{e}'\mathbf{v}^\top = [\mathbf{e}']_\times A. \quad (13.14)$$

Una matrice A che soddisfi $F = [\mathbf{e}'] \times A$ si dice **compatibile** con F .

Si osservi che, dato un qualunque piano Π , l'omografia indotta H_Π è sempre compatibile. Infatti, dalla (13.8) si ricava immediatamente che:

$$H_\Pi = H_\infty + \mathbf{e}' \frac{\mathbf{n}^\top}{d} K^{-1}. \quad (13.15)$$

e quindi la compatibilità di H_Π segue da quella di H_∞ .

Vediamo ora come calcolare una ricostruzione proiettiva da due viste. Sia F la matrice fondamentale e sia A una (qualunque) matrice compatibile con F . Si verifica facilmente che la seguente coppia di MPP:

$$P = [I \mid \mathbf{0}] \quad \text{e} \quad P' = [A \mid \mathbf{e}'] \quad (13.16)$$

sortisce la matrice fondamentale data. Una volta che le due MPP sono state definite, la struttura si ottiene mediante triangolazione.

13.2 Parallasse

Un modo di interpretare l'equazione 13.4 è che un punto m viene associato al suo coniugato in due passi: prima viene applicata H_∞ e poi viene aggiunta una correzione, chiamata *parallasse*, lungo la retta epipolare.

Questa osservazione non vale solo per H_∞ , ma si generalizza ad un piano qualunque [Shashua e Navab, 1996]. Infatti, dopo alcuni pasaggi sulla scia del procedimento impiegato per ricavare la (13.7), si ottiene:

$$\mathbf{m}' \simeq H_\Pi \mathbf{m} + \left(\frac{a}{d z} \right) \mathbf{e}' \quad (13.17)$$

dove a è la distanza del punto \mathbf{M} (del quale \mathbf{m} e \mathbf{m}' sono le proiezioni) dal piano Π e z è la sua profondità rispetto alla prima telecamera.

Quando \mathbf{M} appartiene al piano Π , allora $\mathbf{m}' \simeq H_\Pi \mathbf{m}$, altrimenti vi è un termine residuo, detto *parallasse* $\gamma = \frac{a}{d z}$. Come si vede in figura 13.4 il parallasse è la proiezione sul piano della fotocamera di destra del segmento di raggio ottico compreso tra M e l'intersezione con Π .

La (13.17) è una rappresentazione alternativa della geometria epipolare, prendendo un particolare piano come riferimento.

Alcune note:

- γ è indipendente dalla scelta della seconda immagine;
- γ è proporzionale all'inversa della profondità z ;
- quando il piano di riferimento è il piano all'infinito $\gamma = \frac{1}{z}$;
- Il campo di parallasse è radiale con centro nell'epipolo.

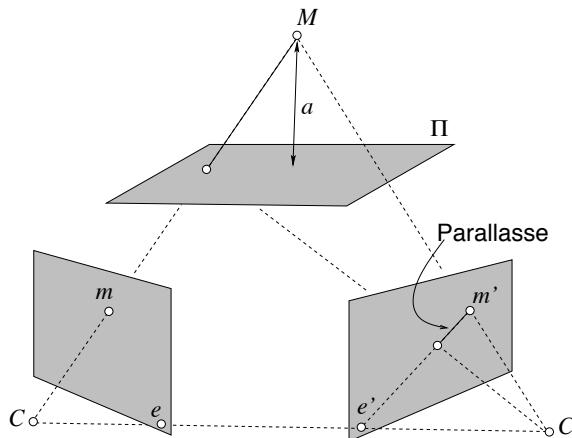


Fig. 13.4. Geometria epipolare con piano + parallasse.

In figura 13.5 viene riportato un semplice esperimento che consente di visualizzare il parallasse. Si prendono due immagini di una scena contenente almeno un piano. Si effettua l'allineamento della prima immagine sulla seconda rispetto al piano, applicando l'omografia indotta da tale piano a tutta la prima immagine. Si osserva che i punti del piano (la facciata del palazzo) coincidono mentre quelli fuori del piano (la statua di Dante) no. La differenza di posizione è il parallasse.

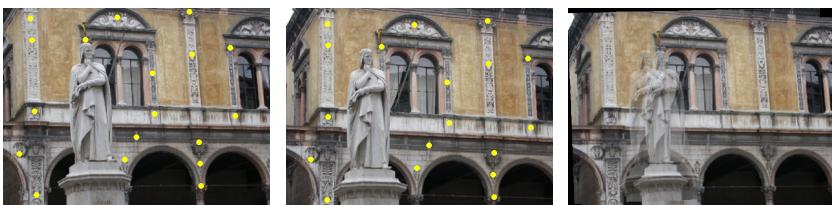


Fig. 13.5. Le prime due immagini da sinistra sono le originali. La terza è la sovrapposizione della seconda con la prima trasformata secondo l'omografia del piano della facciata del palazzo (i punti usati per il calcolo dell'omografia sono evidenziati).

Nei prossimi due paragrafi studieremo due tecniche legate dal fatto che entrambe sfruttano omografie per il recupero di informazioni sulla fotocamera e/o sulla scena. La prima, in analogia alla calibrazione della fotocamera, calcola i parametri intrinseci ed estrinseci di quest'ultima a partire dalla conoscenza di punti 3D su un piano. La seconda è analoga

al recupero di moto e struttura dalla matrice essenziale, ma opera con omografie indotte da un piano e parametri intrinseci noti.

13.3 Calibrazione planare

Nel metodo DLT per la calibrazione che abbiamo descritto nel §.4.2 serve un oggetto che di solito deve essere composto da due o tre piani perfettamente ortogonali (per poter ottenere agevolmente le coordinate dei punti di riferimento). Nella pratica è difficile costruire un tale oggetto senza un’officina meccanica a disposizione; molto più facile è procurarsi un oggetto planare, anche con buona precisione. Il metodo di calibrazione di [Zhang, 2000] si basa proprio su molte (almeno 3) immagini di un piano, invece che su una immagine di molti (almeno 2) piani. Si tratta di una tecnica simile all’autocalibrazione, ma con la differenza che si assume di conoscere ciò che si sta inquadrando. In particolare si assume che vi sia un piano della scena del quale si è in grado di calcolare l’omografia che lo porta sull’immagine. Occorre predisporre all’uopo un oggetto planare di calibrazione, sul quale è disegnata una griglia o una scacchiera, come in figura 13.6. Gli accoppiamenti tra i punti nella immagine **m** ed i corrispondenti punti **M** nella griglia di calibrazione sono assegnati.

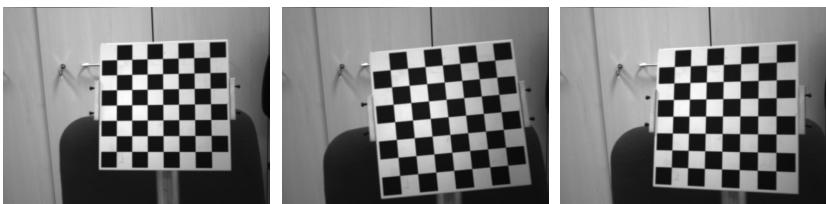


Fig. 13.6. Alcune immagini della scacchiera di calibrazione

Abbiamo già introdotto le proiettività del piano, o omografie, ed abbiamo osservato che quando la fotocamera inquadra un oggetto planare, la trasformazione tra i punti della scena, appartenenti al piano, e l’immagine è proprio una omografia. Se assumiamo per semplicità che il piano abbia equazione $Z = 0$, la matrice dell’omografia contiene la prima seconda e quarta colonna della MPP P , ovvero:

$$H = \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,4} \end{bmatrix} \quad (13.18)$$

In sostanza la matrice H è una semplificazione della matrice P al caso

di punti giacenti su un piano. Con abuso di notazione, visto che d'ora in poi tratteremo solo punti giacenti sul piano $Z = 0$, scriveremo $\mathbf{M} = [X, Y, 1]^\top$. Dunque

$$\mathbf{m} \simeq H\mathbf{M} \quad (13.19)$$

L'omografia H , come nella calibrazione di P , si calcola a partire dalla corrispondenza tra punti modello e punti immagine, ed è nota a meno di un fattore di scala incognito.

Considerato che $P = K[R, \mathbf{t}]$, e posto $R = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$, si ha :

$$H = \lambda K[\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}] \quad (13.20)$$

dove λ è uno scalare incognito. Scrivendo $H = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3]$ otteniamo dunque:

$$\mathbf{r}_1 = \lambda K^{-1} \mathbf{h}_1 \quad (13.21)$$

$$\mathbf{r}_2 = \lambda K^{-1} \mathbf{h}_2. \quad (13.22)$$

Grazi al fatto che le colonne di R sono ortonormali si possono ottenere dei vincoli sui parametri intrinseci. In particolare, l'ortogonalità $\mathbf{r}_1^\top \mathbf{r}_2 = 0$ sortisce $\mathbf{h}_1^\top (KK^\top)^{-1} \mathbf{h}_2 = 0$ o, equivalentemente

$$\mathbf{h}_1^\top B \mathbf{h}_2 = 0 \quad (13.23)$$

dove $B = (KK^\top)^{-1}$. Allo stesso modo, la condizione sulla norma: $\mathbf{r}_1^\top \mathbf{r}_1 = \mathbf{r}_2^\top \mathbf{r}_2$ sin traduce in

$$\mathbf{h}_1^\top B \mathbf{h}_1 = \mathbf{h}_2^\top B \mathbf{h}_2 \quad (13.24)$$

Mediante l'introduzione del prodotto di Kronecker come di consueto, le ultime due equazioni si riscrivono:

$$(\mathbf{h}_2^\top \otimes \mathbf{h}_1^\top) \text{vec}(B) = 0 \quad (13.25)$$

$$((\mathbf{h}_1^\top \otimes \mathbf{h}_1^\top) - (\mathbf{h}_2^\top \otimes \mathbf{h}_2^\top)) \text{vec}(B) = 0 \quad (13.26)$$

Essendo B una matrice 3×3 simmetrica, i suoi elementi non duplicati sono solo sei. Questo si può formalmente tenere in considerazione mediante l'introduzione dell'operatore vech (si veda il paragrafo A1.12):

$$(\mathbf{h}_2^\top \otimes \mathbf{h}_1^\top) D_3 \text{vech}(B) = 0 \quad (13.27)$$

$$((\mathbf{h}_1^\top \otimes \mathbf{h}_1^\top) - (\mathbf{h}_2^\top \otimes \mathbf{h}_2^\top)) D_3 \text{vech}(B) = 0 \quad (13.28)$$

Riassumendo, una immagine fornisce due equazioni in sei incognite. Se si osservano n immagini del piano (con diversa posizione ed orientazione,

come in figura 13.6), si possono impilare le $2n$ equazioni risultanti in un sistema lineare

$$A \operatorname{vech}(B) = \mathbf{0} \quad (13.29)$$

dove A è una matrice $2n \times 6$. Se $n \geq 3$ si ha una soluzione determinata a meno di un fattore di scala. Da $\operatorname{vech}(B)$ si ricava B e quindi K (mediante fattorizzazione di Cholesky), dalla quale si risale poi ad R e \mathbf{t} .

Come noto, la minimizzazione algebrica produce risultati polarizzati, quindi è opportuno raffinare i parametri ottenuti minimizzando l'errore di riproiezione (come nella (4.40)).

L'algoritmo è implementato nella funzione MATLAB `calibz`.

13.4 Moto e struttura da omografia calibrata

La conoscenza della omografia H_Π indotta da un piano tra due immagini calibrate può servire ed estrarre informazioni circa il moto della fotocamera tra le due immagini e la struttura della scena, ovvero i parametri del piano.

Dalla definizione di H_Π (13.8): si ricava immediatamente

$$K'^{-1}H_\Pi K = R + \frac{\mathbf{t}\mathbf{n}^\top}{d}. \quad (13.30)$$

Si vede che H_Π codifica i parametri del moto (R, \mathbf{t}) ed anche i parametri del piano \mathbf{n} e d .

Effettuando la SVD di $\hat{H}_\Pi = K'^{-1}H_\Pi K$ otteniamo

$$\hat{H}_\Pi = UDV^\top \quad (13.31)$$

Quindi

$$D = \frac{d}{s}(sU^\top RV) + (U^\top \mathbf{t})(V^\top \mathbf{n})^\top = d'R' + \mathbf{t}'\mathbf{n}'^\top \quad (13.32)$$

dove $s = \det U \det V$ (ci assicura che R' è una rotazione). Questa equazione può essere risolta (con una duplice ambiguità nella soluzione) per ottenere: (i) la normale al piano \mathbf{n} , (ii) la rotazione R e (iii) la traslazione scalata con la distanza dal piano (\mathbf{t}/d). Si veda [Faugeras e Lustman, 1988] per i dettagli.

Esercizi ed approfondimenti

- 13.1 L'algoritmo implementato nel *calibration toolbox* (http://www.vision.caltech.edu/bouguetj/calib_doc/) è simile a quello di Zhang. Sperimentare la calibrazione di una fotocamera in laboratorio.

- 13.2 Provare che, comunque fissato un piano Π , vale sempre:

$$H_\Pi \mathbf{e} \simeq \mathbf{e}' . \quad (\text{E13.1})$$

Vuol dire che per calcolare l'omografia bastano tre punti appartenenti al piano e gli epipoli, per avere in totale quattro punti coniugati.

- 13.3 Esistono omografie che non sono indotte da alcun piano reale: basta prendere quattro coppie di coniugati generici (non coplanari) e calcolare l'omografia. Le omografie calcolate con tre punti e l'epipolo, invece, sono indotte da un piano, per costruzione.
- 13.4 Esistono matrici compatibili con la matrice fondamentale ma che non sono omografie. Una matrice compatibile (verificare) ma che non è una omografia (è singolare) è la seguente:

$$S = -\frac{1}{\|\mathbf{e}'\|} [\mathbf{e}'] \times F \quad (\text{E13.2})$$

- 13.5 Usando il risultato dell'esercizio 2, mostrare che, date due omografie H_Π e H_Σ indotte da due piani diversi nella stessa coppia di immagini, allora l'epipolo \mathbf{e}' è l'autovettore corrispondente all'autovalore distinto di $H_\Pi H_\Sigma^{-1}$. Che cosa rappresentano gli altri due autovettori?
- 13.6 Mostrare che, date due fotocamere le cui MPP sono $P_1 = (Q_1|\mathbf{q}_1)$ e $P_2 = (Q_2|\mathbf{q}_2)$, l'omografia del piano all'infinito è $H_\infty = Q_2 Q_1^{-1}$. Suggerimento: considerare le proiezioni \mathbf{m}_1 ed \mathbf{m}_2 di un generico punto all'infinito $(X, Y, Z, 0)^\top$.
- 13.7 Mostrare che, se l'omografia H trasferisce punti tra due immagini, allora l'omografia $H^{-\top}$ trasferisce linee rette tra le stesse due immagini.
- 13.8 Consideriamo il caso di due fotocamere con gli stessi parametri intrinseci, ignoti. Se l'omografia del piano all'infinito H_∞ è conosciuta, i parametri intrinseci si possono facilmente ricavare. Infatti, se $K' = K$, da $H_\infty = KRK^{-1}$ otteniamo $R = K^{-1}H_\infty K$, e, poiché $RR^\top = I$, si ha:

$$H_\infty KK^\top H_\infty^\top = KK^\top \quad (\text{E13.3})$$

Dato che la (E13.3) è un'uguaglianza tra matrici simmetriche 3×3 , otteniamo un sistema lineare di sei equazioni nelle cinque incognite del triangolo superiore (o inferiore) di $B = KK^\top$. In effetti, solo quattro equazioni sono indipendenti [Luong e Viéville, 1996], quindi servono almeno tre viste (con gli stessi parametri

intrinseci) per avere un sistema lineare sovradeterminato. Se vale $\gamma = 0$, sono sufficienti due viste.

13.9 Come calcolare H_∞ ? In vari modi:

- conoscendo al minimo tre punti di fuga;
- approssimandola con un piano “abbastanza lontano” dalla fotocamera [Viéville *e al.*, 1996];
- facendo ruotare la fotocamera [Hartley, 1997].

Una qualunque omografia calcolata con DLT è nota a meno di un fattore di scala. Che dire di H_∞ ? Possiamo fissarne la scala in modo non arbitrario? (Suggerimento: matrici simili hanno gli stessi autovalori).

Bibliografia

- Faugeras O.; Lustman F. (1988). Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, **2**, 485–508.
- Hartley R. (1997). Self-calibration of stationary cameras. *International Journal of Computer Vision*, **22**(1), 5–24.
- Luong Q.-T.; Viéville T. (1996). Canonical representations for the geometries of multiple projective views. *Computer Vision and Image Understanding*, **64**(2), 193–229.
- Shashua A.; Navab N. (1996). Relative affine structure: Canonical model for 3D from 2D geometry and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**(9), 873–883.
- Viéville T.; Zeller C.; Robert L. (1996). Using collineations to compute motion and structure in an uncalibrated image sequence. *International Journal of Computer Vision*, **20**(3), 213–242.
- Zhang Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(11), 1330–1334.

14

Mosaicatura e sintesi di immagini

La notevole semplificazione della geometria epipolare che si ottiene nel caso planare rende possibili applicazioni interessanti, la più notevole delle quali è la mosaicatura di immagini. Questa non è strettamente legata al tema della ricostruzione, che abbiamo seguito lungo tutto il corso, tuttavia si è ritenuto di includerla per il suo interesse intrinseco e per la ricchezza di collegamenti con gli argomenti precedenti. Lo stesso vale per la sintesi di immagini, nota anche come *image-based rendering*.

14.1 Mosaici

La *mosaicatura di immagini* è l'allineamento (o registrazione) automatico di più immagini in aggregati più grandi. Ci sono due tipi di mosaici; in entrambi i casi, le immagini sono legate da omografie, come discusso in precedenza.

Mosaici planari: vengono aggregate viste differenti di una scena planare, come la carta geografica di figura 14.1.

Mosaici panoramici: vengono aggregate immagini prese da una fotocamera che ruota sul suo centro ottico, come nel caso di figura 14.2. Si chiamano anche mosaici *panottrici*.

Quando un mosaico panoramico viene proiettato su un piano, man mano che ci si allontana dal fotogramma di riferimento, le immagini vengono “stirate” in orizzontale (ovvio) ed in verticale (meno ovvio), come si vede in figura 14.3. Per far fronte a grandi rotazioni e superare il limite dei 180°, le immagini sono convertite in coordinate cilindriche.

La costruzione di un mosaico è normalmente realizzata allineando le



Fig. 14.1. Mosaico planare in cui sono evidenziate in bianco le cornici delle diverse immagini che lo compongono.

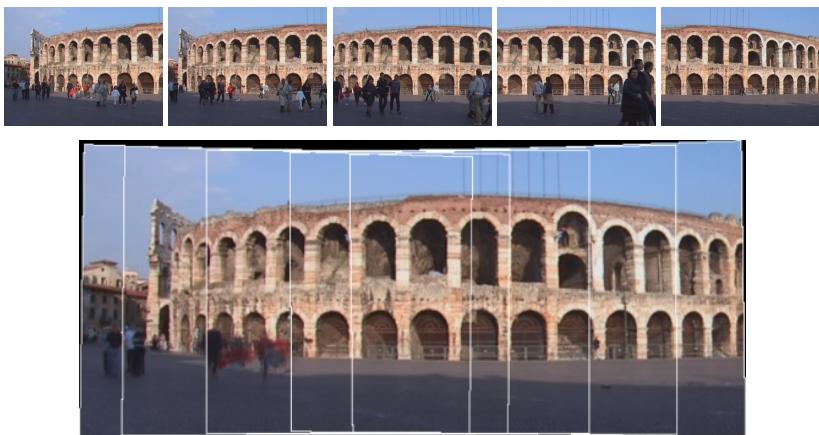


Fig. 14.2. Alcuni fotogrammi della sequenza “Arena” (in alto) ed il mosaico panoramico (in basso). In bianco sono evidenziate le cornici delle diverse immagini che lo compongono. È stata usata la mediana per la miscelazione (si veda più avanti), infatti molti oggetti in movimento risultano rimossi.

immagini della sequenza rispetto ad un comune fotogramma di riferimento e miscelandole, poi, in un'unica immagine mosaico. Si procede attraverso tre fasi:

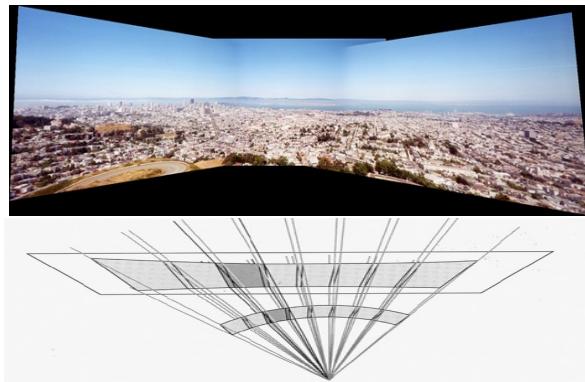


Fig. 14.3. Quando un mosaico panoramico viene proiettato su un piano, man mano che ci si allontana dal fotogramma di riferimento, le immagini vengono “stirate” sia in orizzontale che in verticale.

Allineamento: per ogni fotogramma si calcola l’omografia che lo allinea con il fotogramma di riferimento;

Trasformazione geometrica: si applica l’omografia all’immagine, deforma-

Miscelazione: si assegna un colore a ciascun pixel del mosaico, sulla base dei colori dei pixel (dei vari fotogrammi) che su di esso incidono.

14.1.1 Allineamento

L’allineamento dei fotogrammi della sequenza può essere eseguito nei seguenti modi:

- **Fotogramma con fotogramma:** le omografie vengono prima calcolate tra i fotogrammi consecutivi lungo l’intera sequenza. Esse vengono quindi composte (ovvero moltiplicate) per ottenere le omografie di allineamento tra ogni fotogramma della sequenza e quello di riferimento.
- **Fotogramma con mosaico:** le omografie sono calcolate tra il fotogramma corrente ed il mosaico costruendo. Esse sono usate direttamente per portare il fotogramma sul mosaico.

Nel primo metodo si può costruire un mosaico temporaneo ed aggiornarlo ogni volta che viene esaminata una nuova immagine della sequenza, oppure si possono applicare simultaneamente tutte le omografie calcolate



Fig. 14.4. Mosaico della facciata di S. Zeno (VR), prima (sinistra), e dopo (destra) l'aggiustamento globale (da [Marzotto *e al.*, 2004]). Si osservi in particolare la zona della facciata in basso a sinistra.

per ciascun fotogramma, costruendo così un solo mosaico finale. Nel secondo metodo invece è obbligatorio aggiornare ad ogni passo il mosaico. È quindi leggermente più oneroso, ma garantisce un risultato migliore a fronte di errori di disallineamento che possono essere presenti.

Il metodo migliore per attenuare l'effetto di questi errori è effettuare un *aggiustamento globale* del mosaico (si veda la figura), che consiste in un allineamento che tenga conto (idealmente) di tutti i fotogramma sovrapposti, e non solo di quelli consecutivi.

14.1.2 Trasformazione geometrica

La trasformazione (o deformazione) geometrica dell'immagine è una operazione che altera l'arrangiamento spaziale dei pixel (riposiziona i pixel). Nella letteratura viene indicata come *image warping*. Si tratta di un argomento che interessa sia l'elaborazione delle immagini che la grafica, dove interviene nel *texture mapping*.

Ci sono diversi tipi di trasformazioni geometriche: affinità, omografie (proiettività), affinità a tratti su una triangolazione, la trasformazione di [Beier e Neely, 1992] (impiegata nel *morphing*), trasformazioni non parametriche, definite per punti, come nel caso di un campo di disparità o parallasse. Affronteremo il problema in generale, anche se in questo capitolo siamo interessati solo alle omografie.

Per eseguire una trasformazione geometrica di un'immagine, serve l'applicazione che associa punti corrispondenti tra l'immagine sorgente e l'immagine di destinazione. Se (u, v) sono le coordinate sorgente e (x, y) sono le coordinate di destinazione, può essere data la *trasformazione in avanti*: $x = x(u, v)$ e $y = y(u, v)$ oppure la *trasformazione inversa*: $u = u(x, y)$ e $v = v(x, y)$.

Applicazione della trasformazione geometrica

Scansione della immagine sorgente (*forward mapping*):

```
for v = vmin to vmax
    for u = umin to umax
        x = x(u,v)
        y = y(u,v)
        copy pixel at source[u,v] to dest[x,y]
```

spesso crea buchi (pixel non scritti) anche se tale problema si può evitare disegnando pixel più grandi (*splatting*).

Scansione della immagine destinazione (*backward mapping*):

```
for y = ymin to ymax
    for x = xmin to xmax
        u = u(x,y)
        v = v(x,y)
        copy pixel at source[u,v] to dest[x,y]
```

questo è un metodo migliore, ma deve essere disponibile la trasformazione inversa (vero nel caso delle proiettività).

La trasformazione geometrica di una immagine richiede una fase di ricampionamento, ovvero di conversione di un segnale digitale (2D) da una griglia di campionamento ad un'altra. Se, seguendo letteralmente lo pseudo-codice precedente, si copiano semplicemente i pixel, si ottengono dei pessimi risultati. In particolare:

- Se la trasformazione ingrandisce l'immagine, l'operazione comporta sovraccampionamento o interpolazione. Se non viene eseguita bene, si ottiene l'effetto di *rastering*, ovvero ripetizione di pixel.
- Se la trasformazione rimpicciolisce l'immagine, l'operazione comporta sottocampionamento o decimazione. Se non viene eseguita bene, si ottiene l'effetto di *aliasing*, ovvero la comparsa di artefatti o la perdita di strutture.

Il ricampionamento di buona qualità con fattore di scala arbitrario richiede un attento uso di filtri passa basso con supporto variabile.

Interpolazione bilineare. Una buona tecnica di ricampionamento che funziona per un fattore di scala fino a due (che quindi non aumenta e non riduce di molto) si può ottenere con l'interpolazione bilineare. Si tratta di una funzione continua, poco dispendiosa da calcolare, che interpola i dati su una griglia quadrata. Consideriamo, per semplicità il quadrato di vertici $(0,0)$, $(1,0)$, $(0,1)$, e $(1,1)$ nella immagine sorgente I (figura 14.5). Il valore interpolato nel punto (u,v) , compreso nel quadrato, si calcola con:

$$\begin{aligned} I(u, v) = & (1 - u)(1 - v)I(0, 0) + u(1 - v)I(1, 0) + \\ & + (1 - u)vI(0, 1) + uvI(1, 1). \end{aligned} \quad (14.1)$$

Per un'ottimizzazione della formula si eseguono le seguenti operazioni che necessitano di tre moltiplicazioni invece che di otto:

$$\begin{aligned} Iu0 &= I(0, 0) + u(I(1, 0) - I(0, 0)) \\ Iu1 &= I(0, 1) + u(I(1, 1) - I(0, 1)) \\ I(u, v) &= Iu0 + v(Iu1 - Iu0) \end{aligned} \quad (14.2)$$

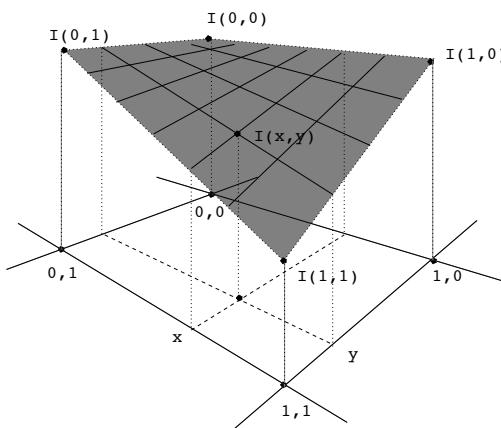


Fig. 14.5. Interpolazione bilineare.

Se si lavora con immagini RGB si fa la stessa operazione su ognuno dei 3 canali, indipendentemente.

Si veda la funzione MATLAB `imwarp`.



Fig. 14.6. Mosaico panoramico dell’Arena di Verona usando la media per la miscelazione. Si notano più artefatti, a causa degli oggetti in movimento, rispetto alla mediana.

14.1.3 Miscelazione

Dopo aver allineato i fotogrammi (ed applicato la corretta trasformazione geometrica), se immaginiamo di intersecare tutti i fotogrammi con una linea temporale essa attraversa tutti i pixel che, idealmente, rappresentano lo stesso punto della scena[†]. Il livello di grigio in ogni pixel del mosaico sarà calcolato applicando un appropriato operatore di miscelazione ai pixel corrispondenti (lungo la linea temporale).

Si possono impiegare diversi operatori di miscelazione, ottenendo effetti diversi. I più comuni sono:

- lo *use-first/last*, che assegna al pixel del mosaico il valore del fotogramma meno/più recente;
- la *media*, efficiente nel rimuovere il rumore, ma provoca artefatti se la sequenza contiene oggetti in movimento (possono apparire sfocati o creare una striscia, a seconda del moto).
- la *mediana* o la *moda*, rimuove il rumore e anche gli oggetti in movimento, a patto che l’oggetto non incida sullo stesso pixel per più della metà dei fotogrammi.

Nelle immagini prese con una comune fotocamera l’illuminazione non è uniforme (lo si può notare specialmente nel cielo). Questo effetto (chiamato *vignetting*) è dovuto all’ottica, che raccoglie più luce in centro e meno alla periferia. Ciò crea degli artefatti nei mosaici, ai quali si può porre rimedio mediante un operatore di miscelazione opportuno, chia-

[†] È come avere annullato il movimento compiuto dalla fotocamera. Il mosaico, infatti, è come una immagine presa da una fotocamera ferma con un maggiore angolo di vista.

mato *feathering*, il quale esegue una media pesata dei colori, con peso crescente in ragione della distanza del pixel dal bordo dell'immagine.

La tecnica che produce migliori risultati nella miscelazione è quella multibanda impiegata anche da [Brown e Lowe, 2005]. Un ottimo riferimento per la mosaicatura è [Szeliski, 2006].

14.2 Altre applicazioni

Descriveremo sommariamente altre tecniche legate alle applicazione di omografie alle immagini, quindi “parenti” della mosaicatura. Tra queste va citata anche la rettificazione epipolare trattata nel § 6.4.

14.2.1 Stabilizzazione dell'immagine

Data una sequenza di immagini, lo scopo della stabilizzazione è compensare il moto della fotocamera, ovvero deformare geometricamente ciascun fotogramma in modo che un dato piano Π della scena appaia immobile (nonostante la fotocamera sia in movimento). Questo si può facilmente fare trasformando ogni fotogramma rispetto ad un fotogramma di riferimento (scelto arbitrariamente) con l'omografia del piano Π . È lo stesso procedimento impiegato nei mosaici, ma senza miscelazione: si visualizza solo l'ultimo fotogramma trasformato, come in figura 14.7. Se la scena scena è planare o il moto è rotatorio (in questo caso il piano Π è quello all'infinito) la sequenza sarà completamente stabilizzata, altrimenti il piano Π apparirà immobile, ed il resto della scena in movimento.

14.2.2 Rettificazione ortogonale

La rettificazione ortogonale serve a “raddrizzare” una immagine prospettica di un piano preso di scorcio (figura 14.8). Questa rettificazione si basa sul fatto che la trasformazione tra un piano della scena e la sua immagine prospettica è una omografia, la quale è completamente definita da quattro punti dei quali si conosca la posizione nel piano della scena. Una volta determinata tale omografia, l'immagine può essere proiettata all'indietro nel piano della scena. Questo è equivalente a sintetizzare un'immagine di una vista fronto-parallela del piano. Tale metodo è conosciuto anche col nome di *rettificazione ortogonale* [Liebowitz e Zisserman, 1998] di un'immagine prospettica.

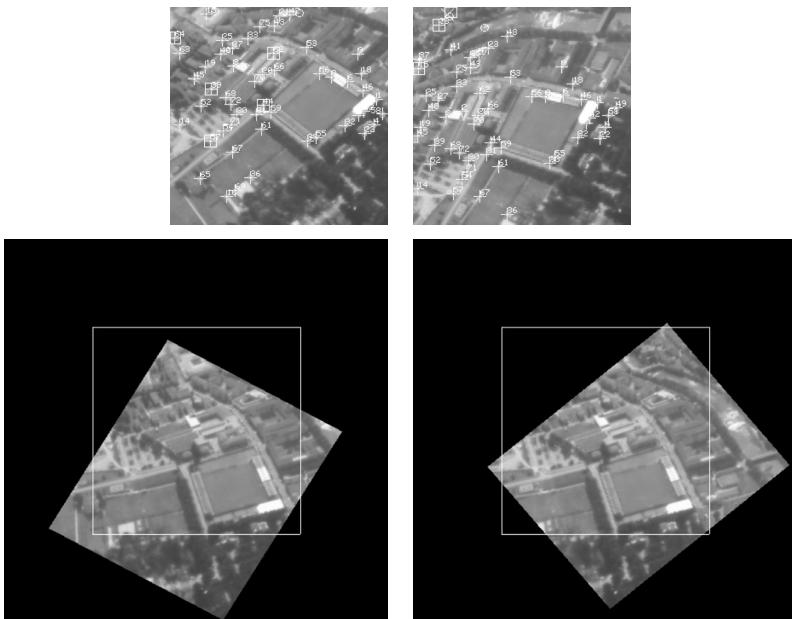


Fig. 14.7. Riga in alto: due fotogrammi di una sequenza aerea. Riga in basso: gli stessi fotogrammi stabilizzati rispetto al piano del terreno. In bianco la cornice del fotogramma di riferimento. Immagini tratte da [Censi *e al.*, 1999].



Fig. 14.8. A destra una fotografia di scorcio di Porta Vescovo (Verona). A sinistra l'immagine ortorettificata rispetto al rettangolo del portone.

14.3 Sintesi di immagini

La sintesi di immagini da altre immagini (*image based rendering*) contrapposta alla sintesi di immagini da modelli tridimensionali (*model based rendering*), è un concetto relativamente recente, che trae origine dalla idea che una scena può essere rappresentata da una collezione di sue im-

magini. Quelle che mancano si possono sintetizzare a partire da quelle esistenti. La geometria entra in gioco nello stabilire la funzione che porta i pixel noti in quelli della costruenda immagine. Si parla di trasferimento di pixel, *warping* o interpolazione di immagini.

14.3.1 Trasferimento con la profondità.

Si considerino le seguenti due MPP:

$$P = K[I|\mathbf{0}] = [K|\mathbf{0}] \quad \text{e} \quad P' = K'[R|\mathbf{t}] \quad (14.3)$$

Sostituendo queste MPP nella equazione della retta epipolare di \mathbf{m} con le profondità esplicitate (E6.5) otteniamo:

$$\zeta' \mathbf{m}' = \zeta K' R K^{-1} \mathbf{m} + K' \mathbf{t}. \quad (14.4)$$

Quindi, se la profondità ζ di un punto \mathbf{m} nell'immagine di riferimento è nota, la sua posizione \mathbf{m}'' nell'immagine sintetica si ottiene usando la (14.4):

$$\mathbf{m}'' \simeq \zeta K'' R K^{-1} \mathbf{m} + K'' \mathbf{t}. \quad (14.5)$$

dove R e \mathbf{t} specificano la posizione e orientazione della fotocamera virtuale rispetto a quella di riferimento (reale). K'' sono i parametri intrinseci della fotocamera virtuale e K sono quelli della fotocamera di riferimento (che dunque è calibrata).

14.3.2 Interpolazione con disparità

Se non si conosce la profondità del punto, ma si possiedono *due* immagini di riferimento I, I' con una mappa densa di corrispondenze, si può sfruttare la geometria epipolare per predire la posizione di un punto nella terza vista (sintetica) I'' . La fotocamera virtuale è vincolata a giacere sulla linea di base tra le due fotocamere di riferimento. Per questo la chiamiamo *interpolazione*. Inoltre si assumono noti i parametri intrinseci.

Assumiamo che le fotocamere siano rettificate. In questo caso la (14.4) si specializza in

$$\mathbf{m}' = \mathbf{m} + \frac{1}{\zeta} K'[t_x, 0, 0]^\top. \quad (14.6)$$

La differenza $\mathbf{m}' - \mathbf{m}$ è la *disparità*. È un vettore, ma siccome solo il primo



Fig. 14.9. Le due immagini agli estremi sono reali, quella centrale è interpolata (da [Seitz e Dyer, 1996]).

componente è diverso da zero, normalmente si identifica la disparità con quest'ultimo. L'equazione di trasferimento nella vista virtuale I'' è:

$$\mathbf{m}'' = \mathbf{m} + [\alpha d, 0, 0]^\top \quad \alpha \in [0, 1]. \quad (14.7)$$

Si verifica facilmente che interpolare la disparità in questo caso è equivalente a spostare la fotocamera in posizioni intermedie lungo la linea di base, da 0 to t_x . L'algoritmo fu introdotto in [Chen e Williams, 1993] ed esteso a viste non rettificate in [Seitz e Dyer, 1996] con il nome di *view morphing*. Questi ultimi semplicemente rettificano le due immagini, effettuano l'interpolazione ed alla fine de-rettificano l'immagine risultante.

14.3.3 Trasferimento epipolare.

Il trasferimento è possibile anche nel caso non calibrato, ovvero senza conoscere i parametri intrinseci. In effetti tutto quello che serve sono corrispondenze e la geometria epipolare. Date due immagini di riferimento I, I' con una mappa densa di corrispondenze, si può sfruttare la geometria epipolare per predire la posizione di un punto nella terza vista (sintetica) I'' . Si assume che siano date le matrici fondamentali $F_{1,2}, F_{1,3}, F_{2,3}$.

Dati due punti coniugati nelle due immagini di riferimento, \mathbf{m}' e \mathbf{m} , la posizione del loro punto coniugato nella terza vista \mathbf{m}'' è determinata dal fatto che egli appartiene simultaneamente alla linea epipolare di \mathbf{m}' ed a quella di \mathbf{m} , come illustrato in figura 14.10. In formule:

$$\mathbf{m}'' \simeq F_{1,3}\mathbf{m} \times F_{2,3}\mathbf{m}'. \quad (14.8)$$

Questo metodo non funziona quando i tre raggi ottici sono coplanari (le due rette nella terza vista diventano coincidenti).

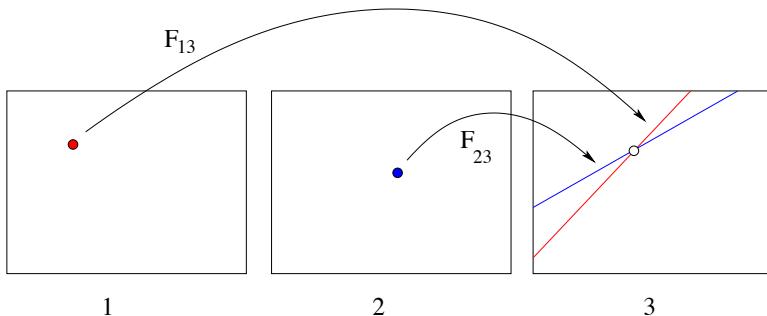


Fig. 14.10. Il punto nella terza vista è determinato dalla intersezione delle rette epipolari degli altri due.



Fig. 14.11. Da sinistra a destra: due immagini reali e l'immagine sintetica, creata con trasferimento epipolare (da [Laveau e Faugeras, 1994]).

14.3.4 Trasferimento con parallasse

Nei due metodi precedenti, le disparità vengono usate come un succedaneo della profondità ignota. Anche il parallasse può servire allo stesso scopo. Ricordiamo che la relazione tra due viste tramite un piano di riferimento è data dalla (13.17), che riscriviamo:

$$\mathbf{m}' \simeq H_{\Pi} \mathbf{m} + \gamma \mathbf{e}'. \quad (14.9)$$

Dati un certo numero (> 6) di punti coniugati, $(\mathbf{m}_1^k; \mathbf{m}_2^k) \quad k = 1, \dots, m$ l'omografia H_{Π} e l'epipolo \mathbf{e}' si possono agevolmente calcolare. Anche il parallasse in ciascun punto della immagine di riferimento si calcola (proposizione 1.51) risolvendo rispetto a γ nella (14.9):

$$\gamma_i = \frac{(\mathbf{H}_{\Pi} \mathbf{m}_i \times \mathbf{m}'_i)^T (\mathbf{m}'_i \times \mathbf{e}')}{\|\mathbf{m}'_i \times \mathbf{e}'\|^2} \quad (14.10)$$

Attenzione che H_{Π} e \mathbf{e}' sono noti solo a meno di un fattore di scala, quindi anche il modulo del parallasse γ contiene un fattore di scala incognito.

Noto il parallasse γ , poiché esso non dipende dalla seconda vista, questa può essere sostituita da una virtuale [Shashua e Navab, 1996]. La (14.9) può dunque essere usata per il trasferimento dei punti nella vista sintetica I'' :

$$\mathbf{m}'' \simeq H_{\Pi}'' \mathbf{m} + \gamma \mathbf{e}'' . \quad (14.11)$$

dove l'omografia H_{Π}'' indotta da Π tra la prima e la terza vista e l'epipolo \mathbf{e}'' nella terza vista devono essere dati, e specificano – indirettamente – posizione ed orientazione della fotocamera virtuale.

Si noti la somiglianza di questa tecnica con il trasferimento con profondità. L'equazione è simile, ma in un caso si impiegano quantità “calibrate” come ζ , R e \mathbf{t} , mentre nell'altro quantità “non calibrate” come γ , H_{Π}'' e \mathbf{e}'' . In altri termini, nel primo caso ci si muove in una cornice euclidea, mentre nel secondo in una cornice proiettiva. La seconda richiede meno informazioni in ingresso (non serve la calibrazione) ma la specifica del punto di vista virtuale è problematica, mentre nel caso euclideo è immediata.

Un esempio di sintesi con parallasse è mostrata in figura 14.12. Il problema del posizionamento della fotocamera virtuale è risolto come descritto in [Fusiello, 2007].

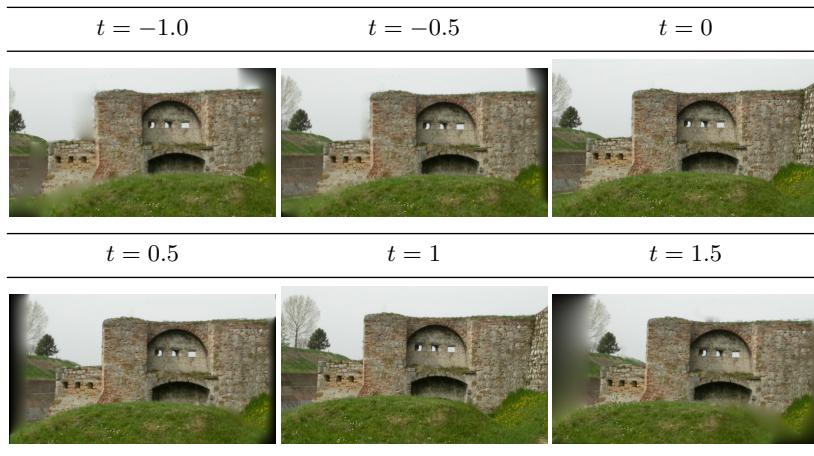


Fig. 14.12. Alcuni fotogrammi di una sequenza (controporta di Palmanova, UD) sintetizzata usando il parallasse. I valori $t = 0$ e $t = 1$ corrispondono alle immagini di riferimento. Per valori compresi tra 0 e 1 si ha interpolazione, per valori esterni all'intervallo si ha estrapolazione.

14.3.5 Trasformazione delle immagini

L'applicazione alle immagini delle formule di trasferimento che abbiamo appena visto solleva alcuni problemi, in parte già trattati nel paragrafo 14.1.2. Rispetto ai mosaici ci troviamo in una situazione peggiore, poiché la trasformazione non è parametrica ma definita per punti, quindi si può solo applicare la scansione della immagine sorgente (*forward mapping*). I problemi che si hanno sono (con riferimento alla figura 14.13):

- Ripiegamento dell'immagine (*folding*): accade quando due o più pixel della immagine sorgente sono associati allo stesso pixel della immagine destinazione.
- Buchi: quando punti non visibili nella immagine sorgente sono invece visibili in quella di destinazione.
- Magnificazione: l'area proiettata di una superficie aumenta considerevolmente nella immagine destinazione (condiviso con i mosaici).

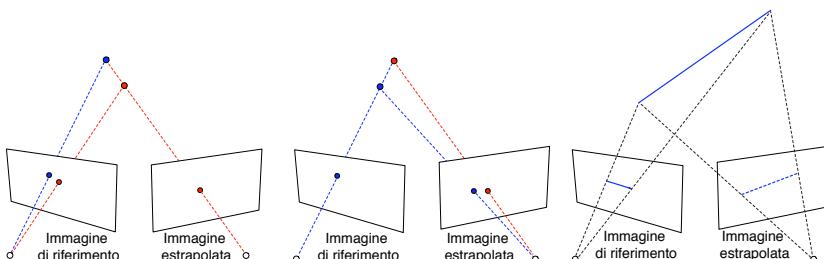


Fig. 14.13. Artefatti nella sintesi di immagini. Da sinistra a destra: *folding*, buchi e magnificazione.

Il *folding* può essere evitato seguendo un opportuno ordine di valutazione dei pixel [McMillan e Bishop, 1995], che garantisce che i punti più vicini alla fotocamera vengano trasferiti per ultimi, in modo da sovrascrivere eventualmente gli altri. Per esempio, se la fotocamera sintetica viene traslata alla sinistra di quella di riferimento, i pixel devono essere processati da destra a sinistra.

Gli artefatti di magnificazione accadono anche nel texture mapping e nell'applicazione di omografie. Si risolvono disegnando pixel più grandi (*splatting*).

I buchi sono difficili da risolvere, perché manca l'informazione necessaria per riempirli. Si possono usare tecniche di interpolazione o di *inpainting* [Criminisi *e al.*, 2004].

Esercizi ed approfondimenti

- 14.1 Mosaici con parallasse. Quando c'è un parallasse residuo significativo oltre il modello planare, servono tecniche che tengono conto della struttura della scena. Il formalismo del piano+parallasse consente di registrare parti di una scena con profondità arbitraria.
- 14.2 Come calcolare l'epipolo dati quattro punti coplanari e due punti non appartenenti al piano in corrispondenza tra due immagini? Suggerimento: con i quattro punti calcolo l'omografia, poi il parallasse degli altri due ...
- 14.3 In una applicazione di navigazione autonoma (per esempio di un autoveicolo), fissiamo come riferimento il piano della strada. Sono immediatamente individuati come ostacoli tutti i punti per cui il parallasse è significativamente diverso da 0. Questo risultato viene ottenuto a partire dalle sole immagini, senza conoscere i parametri della fotocamera.
- 14.4 Nel gioco del calcio, controllare la presenza o meno della palla nella porta usando il parallasse rispetto al piano formato da pali e traversa.
- 14.5 Scrivere una funzione MATLAB per la ortorettificazione.
- 14.6 Ricavare la (14.4) senza fare riferimento alla (E6.5).

Bibliografia

- Beier T.; Neely S. (1992). Feature-based image metamorphosis. In *SIGGRAPH: International Conference on Computer Graphics and Interactive Techniques*, pp. 35–42.
- Brown M.; Lowe D. G. (2005). Unsupervised 3D object recognition and reconstruction in unordered datasets. In *Proceedings of the International Conference on 3D Digital Imaging and Modeling*.
- Censi A.; Fusiello A.; Roberto V. (1999). Image stabilization by features tracking. In *Proceedings of the 10th International Conference on Image Analysis and Processing*, pp. 665–667, Venice, Italy.
- Chen S. E.; Williams L. (1993). View interpolation for image synthesis. In *SIGGRAPH: International Conference on Computer Graphics and Interactive Techniques*. A cura di Kajiya J. T., volume 27, pp. 279–288.
- Criminisi A.; Perez P.; Toyama K. (2004). Region filling and object removal by exemplar-based image inpainting. *Image Processing, IEEE Transactions on*, **13**(9), 1200–1212.

- Fusiello A. (2007). Specifying virtual cameras in uncalibrated view synthesis. *IEEE Transactions on Circuits and Systems for Video Technology*, **17**(5), 604–611.
- Laveau S.; Faugeras O. (1994). 3-D scene representation as a collection of images and fundamental matrices. Technical Report 2205, INRIA, Institut National de Recherche en Informatique et en Automatique.
- Liebowitz D.; Zisserman A. (1998). Metric rectification for perspective images of planes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 482–488.
- Marzotto R.; A. Fusiello; Murino V. (2004). High resolution video mosaicing with global alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pp. 692–698, Whashington, D.C. IEEE Computer Society.
- McMillan L.; Bishop G. (1995). Head-tracked stereo display using image warping. In *Stereoscopic Displays and Virtual Reality Systems II* in SPIE Proceedings, numero 2409, pp. 21–30, San Jose, CA.
- Seitz S. M.; Dyer C. R. (1996). View morphing: Synthesizing 3D metamorphoses using image transforms. In *SIGGRAPH: International Conference on Computer Graphics and Interactive Techniques*, pp. 21–30, New Orleans, Louisiana.
- Shashua A.; Navab N. (1996). Relative affine structure: Canonical model for 3D from 2D geometry and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**(9), 873–883.
- Szeliski R. (2006). Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.*, **2**(1), 1–104.

Appendice 1

Nozioni di Algebra lineare

Questo capitolo riporta una serie di risultati utili per il corso di Visione Computazionale. Per una trattazione sistematica consultare [Strang, 1988].

A1.1 Prodotto scalare

Definizione 1.1 (Prodotto scalare) *Il prodotto scalare di due vettori \mathbf{x} e \mathbf{y} di \mathbb{R}^n si definisce come:*

$$\langle \mathbf{x}, \mathbf{y} \rangle \triangleq \sum_{i=1}^n x_i y_i \quad (1.1)$$

Si denota anche con $\mathbf{x} \cdot \mathbf{y}$. Il prodotto scalare è commutativo, ovvero $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$. Il prodotto scalare induce la norma euclidea:

$$\|\mathbf{x}\|_2 \triangleq \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}. \quad (1.2)$$

Geometricamente, la norma rappresenta la *lunghezza* del vettore. Se \mathbf{y} è *unitario*, ovvero $\|\mathbf{y}\|_2 = 1$, allora $\langle \mathbf{x}, \mathbf{y} \rangle$ rappresenta la lunghezza della proiezione di \mathbf{x} lungo la direzione di \mathbf{y} . L'angolo θ tra due vettori \mathbf{x} e \mathbf{y} si definisce come

$$\theta = \arccos \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2} \quad (1.3)$$

Coerentemente con la definizione di angolo:

Definizione 1.2 (Ortogonalità) *Due vettori \mathbf{x} e \mathbf{y} di \mathbb{R}^n sono ortogonali se*

$$\langle \mathbf{x}, \mathbf{y} \rangle = 0 \quad (1.4)$$

Trattando, cosa che faremo nel seguito, i vettori come matrici formate da una sola colonna, possiamo scrivere il prodotto scalare usando il prodotto matriciale e la trasposizione (apice \top):

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y} = \mathbf{y}^\top \mathbf{x} \quad (1.5)$$

A1.2 Norma matriciale

La norma vettoriale euclidea induce naturalmente una norma matriciale euclidea nel seguente modo (vale in generale per qualunque norma):

$$\|A\|_2 \triangleq \sup_{\|\mathbf{x}\|_2 \leq 1} \frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \quad (1.6)$$

Vedremo in seguito, parlando di *autovalori*, una proprietà che consente di calcolare questa norma.

Esiste un altro modo di generalizzare la norma euclidea alle matrici, che parte da un prodotto scalare sulle matrici che generalizza quello definito per i vettori.

Si definisca il seguente prodotto scalare tra due matrici A e B :

$$\langle A, B \rangle \triangleq \text{tr}(A^\top B) = \sum_{i,j} a_{i,j} b_{i,j} \quad (1.7)$$

il quale si riduce al prodotto scalare tra vettori precedentemente definito nel caso in cui le due matrici siano composte da una sola colonna. La corrispondente norma indotta prende il nome di norma di Frobenius:

Definizione 1.3 (Norma di Frobenius) *Data una matrice A , la sua norma di Frobenius si definisce come:*

$$\|A\|_F = \sqrt{\text{tr}(A^\top A)} = \sqrt{\sum_{i,j} a_{i,j}^2} \quad (1.8)$$

Nel caso particolare in cui la matrice sia un vettore, le due norme coincidono: $\|\mathbf{x}\|_F = \sqrt{\sum_i a_i^2} = \|\mathbf{x}\|_2$

A1.3 Matrice inversa

Definizione 1.4 (Matrice quadrata) *Una matrice A si dice quadrata se ha lo stesso numero di righe e di colonne.*

Una matrice quadrata in cui tutti gli elementi al di sotto (o sopra) della diagonale sono nulli si dice *triangolare*.

Una matrice quadrata in cui tutti gli elementi diversi dalla diagonale sono nulli si dice *diagonale*:

$$\text{diag}(a_1, \dots, a_n) = \begin{bmatrix} a_1 & 0 & 0 & \dots & 0 \\ 0 & a_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_n \end{bmatrix} \quad (1.9)$$

Una particolare matrice quadrata è l'*identità*: $I = \text{diag}(1, \dots, 1)$. La matrice identità è l'elemento neutro per il prodotto matriciale: $IA = AI = A$.

Definizione 1.5 (Matrice inversa) *Sia A una matrice quadrata. Se esiste una matrice B tale che $AB = BA = I$ allora B si chiama inversa di A e si denota con A^{-1} .*

Proprietà:

- (i) L'inversa se esiste è unica.
- (ii) $(A^{-1})^\top = (A^\top)^{-1} \triangleq A^{-\top}$.
- (iii) $(AB)^{-1} = B^{-1}A^{-1}$.

A1.4 Determinante

Definizione 1.6 (Determinante) *Sia A una matrice quadrata $n \times n$. Il determinante di A , si definisce (ricorsivamente) nel modo seguente, per un i fissato:*

$$\begin{cases} \det(A) = \sum_{j=1}^n a_{ij} (-1)^{i+j} \det(A_{ij}) \\ \det(a) = a \end{cases} \quad (1.10)$$

dove $A_{i,j}$ è la sottomatrice di A ottenuta sopprimendo da questa la riga i e la colonna j .

Questa definizione prende anche il nome di *espansione di Laplace* lungo la riga i . Si poteva definire analogamente lungo una colonna. La scelta della riga (o della colonna) è arbitraria.

Per una matrice 2×2 :

$$\det \begin{pmatrix} [a_{1,1} & a_{1,2}] \\ [a_{2,1} & a_{2,2}] \end{pmatrix} = a_{1,1}a_{2,2} - a_{1,2}a_{2,1} \quad (1.11)$$

Osservazione 1.7 Il determinante di una matrice triangolare è il prodotto degli elementi diagonali.

Proprietà:

- (i) $\det(AB) = \det(A)\det(B)$
- (ii) $\det(\alpha A) = \alpha^n \det(A)$ con $\alpha \in \mathbb{R}$
- (iii) $\det(I) = 1$
- (iv) $\det(A^{-1}) = \frac{1}{\det(A)}$
- (v) A è invertibile $\iff \det(A) \neq 0$
- (vi) Lo scambio di due righe (o colonne) di A cambia il segno di $\det(A)$.

Definizione 1.8 (Matrice singolare) Una matrice A tale che $\det(A) = 0$ si dice singolare.

Per la proprietà (v), singolare è sinonimo di non invertibile.

Il determinante è legato alla nozione di lineare indipendenza di un insieme di vettori.

Definizione 1.9 (Lineare indipendenza) I vettori $\mathbf{x}_1, \dots, \mathbf{x}_m$ di \mathbb{R}^n si dicono linearmente indipendenti se

$$\sum_{i=1}^m \alpha_i \mathbf{x}_i = \mathbf{0} \implies \forall i : \alpha_i = 0 \quad (1.12)$$

Se i vettori non sono linearmente indipendenti allora (almeno) uno di essi è combinazione lineare degli altri.

Proposizione 1.10 I vettori $\mathbf{x}_1, \dots, \mathbf{x}_n$ di \mathbb{R}^n sono linearmente indipendenti se e solo se

$$\det(\mathbf{x}_1, \dots, \mathbf{x}_n) \neq 0 \quad (1.13)$$

A1.5 Matrici ortogonali

I vettori $\mathbf{x}_1, \dots, \mathbf{x}_m$ di \mathbb{R}^n si dicono mutuamente ortogonali se ciascuno è ortogonale a ciascun altro. Se i vettori sono tutti unitari, allora si dicono ortonormali.

Proposizione 1.11 Se i vettori $\mathbf{x}_1, \dots, \mathbf{x}_m$ sono mutuamente ortogonali allora sono linearmente indipendenti.

Definizione 1.12 (Matrice ortogonale) Una matrice reale quadrata A si dice ortogonale se

$$AA^\top = A^\top A = I \quad (1.14)$$

Dunque per una matrice ortogonale la trasposta coincide con l'inversa. Inoltre segue dalla definizione che se A è ortogonale, le sue colonne (o righe) sono vettori unitari mutuamente ortogonali (ovvero ortonormali).

Osservazione 1.13 Una matrice rettangolare B può soddisfare $B^\top B = I$ oppure $BB^\top = I$ (ma non entrambe). Vuol dire che le sue colonne o le sue righe sono vettori ortonormali. Si chiama in tal caso semi-ortogonale.

Proprietà. Se A è ortogonale, allora

- (i) $\det(A) = \pm 1$
- (ii) $\|Ax\| = \|x\|$

La seconda proprietà vuol dire che l'applicazione di una matrice ortogonale ad un vettore non ne altera la lunghezza. Infatti, le matrici ortogonali sono il prodotto di una rotazione ed una riflessione speculare. In particolare, quelle con determinante positivo sono una pura rotazione, mentre se il determinante è negativo è presente la riflessione speculare.

A1.6 Forme lineari e quadratiche

Definizione 1.14 (Matrice simmetrica ed antisimmetrica) Una matrice quadrata A si dice simmetrica se $A = A^\top$, ovvero $a_{ij} = a_{ji}$. Si dice antisimmetrica se $A = -A^\top$.

Siano $\mathbf{a} \in \mathbb{R}^n$ ed A quadrata $n \times n$.

- (i) $f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x}$ è una forma lineare in \mathbf{x}
- (ii) $f(\mathbf{x}) = \mathbf{x}^\top A \mathbf{x}$ è una forma quadratica in \mathbf{x}
- (iii) $f(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top A \mathbf{y}$ è una forma bilineare in \mathbf{x} e \mathbf{y}

Nella forma quadratica possiamo rimpiazzare A con $\frac{A+A^\top}{2}$ (simmetrica per costruzione) senza cambiare il risultato (verificare per esercizio).

Definizione 1.15 (Matrice definita positiva) Una matrice simmetrica A si dice definita positiva se $\forall x \neq 0 : \mathbf{x}^\top A \mathbf{x} > 0$. Si dice semidefinita positiva se $\forall x : \mathbf{x}^\top A \mathbf{x} \geq 0$

Proprietà. Siano A quadrata $n \times n$ e B $n \times m$.

- (i) BB^\top e $B^\top B$ sono semidefinite positive per costruzione
- (ii) $\forall x : Bx = \mathbf{0} \iff B = \mathbf{0}$
- (iii) $\forall x : x^\top Ax = \mathbf{0} \iff A$ è antisimmetrica
- (iv) $\forall x : x^\top Ax = \mathbf{0} \wedge A$ è simmetrica $\iff A = \mathbf{0}$.

Il punto (iv) deriva dal fatto che la matrice nulla è l'unica contemporaneamente simmetrica ed antisimmetrica.

Teorema 1.16 (Cholesky) *Una matrice A semidefinita positiva può essere decomposta in modo unico come*

$$A = KK^\top \quad (1.15)$$

dove K è una matrice triangolare superiore con diagonale positiva.

A1.7 Rango

Definizione 1.17 (Determinante di ordine q) *Sia A una matrice $m \times n$. Chiamiamo determinante di ordine q estratto da A il determinante di una sottomatrice quadrata di A ordine q (si chiama anche minore).*

Definizione 1.18 (Rango di una matrice) *Il rango di una matrice A $m \times n$ è il massimo numero r tale che tutti i determinanti di ordine r estratti da A sono non nulli, e si indica con $r(A)$.*

Proprietà.

- (i) $r(A) \leq \min(m, n)$. Se $r(A) = \min(m, n)$ si dice che A possiede *rango pieno*.
- (ii) A quadrata ha rango pieno $\iff \det(A) \neq 0$
- (iii) $r(A) = r(A^\top) = r(AA^\top) = r(A^\top A)$
- (iv) Il rango non cambia se si scambiano due righe (o colonne).
- (v) Se B è quadrata non singolare: $r(AB) = r(A)$
- (vi) In generale $r(AB) \leq \min(r(A), r(B))$

Proposizione 1.19 *Sia A una matrice $m \times n$. Il suo rango è pari al numero di colonne o righe linearmente indipendenti che contiene.*

Teorema 1.20 (Decomposizione QR) *Sia A una matrice $m \times n$ con colonne linearmente indipendenti. Allora può essere fattorizzata come*

$$A = QR \quad (1.16)$$

dove le colonne di Q sono ortonormali (è semi-ortogonale) ed R è triangolare superiore non singolare.

Se A è quadrata, allora Q è ortogonale.

Definizione 1.21 (Nucleo di una matrice) Sia A una matrice $m \times n$. Il suo nucleo, denotato da $\ker(A)$, è il sottospazio di \mathbb{R}^n così definito

$$\ker(A) = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{0}\} \quad (1.17)$$

Perché $\ker(A)$ è un sottospazio? Perché

- contiene sempre il vettore nullo: $\forall A : \mathbf{0} \in \ker(A)$ e
- è chiuso per combinazione lineare: $\mathbf{x}, \mathbf{y} \in \ker(A) \implies \forall \alpha, \beta \in \mathbb{R} : \alpha\mathbf{x} + \beta\mathbf{y} \in \ker(A)$ (dimostrare per esercizio usando la linearità del prodotto matriciale).

Definizione 1.22 (Immagine di una matrice) Sia A una matrice $m \times n$. La sua immagine, denotata da $\text{im}(A)$, è il sottospazio di \mathbb{R}^m così definito

$$\text{im}(A) = \{\mathbf{y} \in \mathbb{R}^m \mid \exists \mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{y}\} \quad (1.18)$$

Osservazione 1.23 L'immagine di A è il sottospazio generato dalle colonne di A , ovvero i vettori di $\text{im}(A)$ sono tutti e soli combinazioni lineare delle colonne di A .

Si vede facilmente grazie al prodotto matriciale a blocchi:

$$A\mathbf{x} = [\mathbf{a}_1, \dots, \mathbf{a}_n] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = x_1\mathbf{a}_1 + \dots + x_n\mathbf{a}_n \quad (1.19)$$

dove con $\mathbf{a}_1, \dots, \mathbf{a}_n$ abbiamo denotato le colonne di A .

Proprietà.

- (i) $\dim(\text{im}(A)) = r(A)$ Deriva dalla osservazione precedente e dalla proposizione 1.19.
- (ii) $\text{im}(A) = \text{im}(AA^\top)$

Il seguente teorema è molto importante:

Teorema 1.24 (Rango-nullità) Sia A una matrice $m \times n$, allora

$$n = \underbrace{\dim(\text{im}(A))}_{r(A)} + \underbrace{\dim(\ker(A))}_{\text{nullità di } A} \quad (1.20)$$

Corollario 1.25 Una matrice A ha rango pieno $\iff \ker(A) = \{\mathbf{0}\}$.

A1.8 Autovalori ed autovettori

Definizione 1.26 (Autovalori) Sia A una matrice $n \times n$. Gli autovalori di A sono le radici del polinomio in λ (polinomio caratteristico):

$$\det(\lambda I - A) = 0. \quad (1.21)$$

Sia λ un autovalore di A , allora $\det(\lambda I - A) = 0 \implies \ker(\lambda I - A) \neq \{\mathbf{0}\} \implies \exists \mathbf{x} \neq \mathbf{0} : (\lambda I - A)\mathbf{x} = \mathbf{0}$ o equivalentemente

$$A\mathbf{x} = \lambda\mathbf{x}. \quad (1.22)$$

Tale vettore \mathbf{x} prende il nome di *autovettore* di A .

Osservazione 1.27 Una matrice singolare possiede almeno un autovalore nullo. Infatti $\det(A) = 0 \implies \det(\lambda I - A) = 0$ con $\lambda = 0$

Proposizione 1.28 Una matrice reale simmetrica ha autovalori reali.

Proposizione 1.29 Una matrice è definita positiva \iff tutti gli autovalori sono positivi.

Proposizione 1.30 Una matrice ortogonale ha autovalori di modulo unitario.

Proprietà. Siano $\lambda_1, \dots, \lambda_n$ gli autovalori della matrice A $n \times n$. Allora:

$$\text{tr}(A) = \sum_{i=1}^n \lambda_i \quad (1.23)$$

$$\det(A) = \prod_{i=1}^n \lambda_i \quad (1.24)$$

Teorema 1.31 (Decomposizione di Schur) [†] Sia A una matrice $n \times n$. Esiste una matrice S $n \times n$ invertibile ed una matrice triangolare superiore M i cui elementi diagonali sono gli autovalori di A , tali che

$$S^{-1}AS = M \quad (1.25)$$

Il seguente teorema è un caso particolare del precedente:

[†] Questa è una versione debole del teorema. La versione forte, dice che S è unitaria.

Teorema 1.32 (Diagonalizzazione di Schur) *Sia A una matrice $n \times n$ reale simmetrica con autovalori $\lambda_1, \dots, \lambda_n$. Allora esiste una matrice diagonale $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ ed una matrice ortogonale S $n \times n$ le cui colonne sono gli autovettori di A , tali che:*

$$S^\top AS = \Lambda \quad (1.26)$$

Proposizione 1.33 *Sia A una matrice $n \times n$ reale simmetrica con autovalori $\lambda_1, \dots, \lambda_n$ ordinati dal più grande al più piccolo. Allora*

$$\lambda_n \leq \frac{\mathbf{x}^\top A \mathbf{x}}{\mathbf{x}^\top \mathbf{x}} \leq \lambda_1 \quad (1.27)$$

Si dimostra facilmente con la Diagonalizzazione di Schur (esercizio).

Da questo segue inoltre che

$$\|A\|_2 = \sqrt{\lambda_1(A^\top A)}. \quad (1.28)$$

Proposizione 1.34 (Matrici simili) *Sia A una matrice $n \times n$ e G una matrice $n \times n$ non singolare, allora $G^{-1}AG$ ha gli stessi autovalori di A . G ed $G^{-1}AG$ si dicono simili.*

A1.9 Decomposizione ai valori singolari

Teorema 1.35 (Decomposizione ai valori singolari) *Sia A una matrice $m \times n$. Esiste una matrice D $m \times n$ con elementi diagonali positivi $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$ ed elementi nulli altrove (se fosse quadrata sarebbe diagonale), una matrice U $m \times m$ ortogonale ed una matrice V $n \times n$ ortogonale tali che:*

$$U^\top AV = D \quad (1.29)$$

Gli elementi diagonali di D prendono il nome di valori singolari.

Come preciseremo meglio nel seguito, la decomposizione ai valori singolari (SVD) è cugina della diagonalizzazione, ed i valori singolari sono cugini degli autovalori.

La seguente proposizione giustifica l'importanza della SVD.

Proposizione 1.36 (Nucleo ed immagine della matrice) *Sia A una matrice $m \times n$ e sia $U^\top AV = D$ la sua decomposizione ai valori singolari. Le colonne di U corrispondenti ai valori singolari non nulli sono una base per $\text{im}(A)$, mentre le colonne di V corrispondenti ai valori singolari nulli sono una base per $\ker(A)$.*

$$\left[\begin{matrix} A \end{matrix} \right] = \left[\begin{matrix} \text{blue bar} & \text{orange square} & \text{blue square} \\ \text{U} \end{matrix} \right] \left[\begin{matrix} \cdot & \cdot \\ \cdot & D \\ \cdot & \cdot \end{matrix} \right] \left[\begin{matrix} \text{blue square} & \text{orange square} \\ V^\top \end{matrix} \right]$$

Confrontare questo risultato con il teorema 1.24.

La proposizione precedente fornisce, tramite la SVD, un algoritmo per determinare $\ker(A)$ ed $\text{im}(A)$. Si potrebbe obiettare che quest'ultimo è generato dalle colonne di A , quindi la decomposizione è inutile. La risposta è che le colonne di A non sono una base.

Osservazione 1.37 *Il numero di valori singolari non nulli è pari al rango della matrice.*

Osservazione 1.38 *Se A è una matrice quadrata, $\det(A) \neq 0 \iff \sigma_1, \dots, \sigma_n \neq 0$.*

Più nello specifico, il valore $\frac{\sigma_n}{\sigma_1}$ indica in che misura la matrice è prossima alla singolarità. Se $c \approx \varepsilon$ (dove ε è l'epsilon macchina) la matrice A è *malcondizionata*, ovvero è singolare agli effetti pratici.

La seguente proposizione precisa la parentela con la diagonalizzazione.

Proposizione 1.39 *Sia A una matrice $m \times n$ e siano $\sigma_1, \dots, \sigma_r$ i suoi valori singolari non nulli. Allora $\sigma_1^2, \dots, \sigma_r^2$ sono gli autovalori non nulli di $A^\top A$ e di AA^\top . Le colonne di U sono gli autovettori di AA^\top , le colonne di V sono gli autovettori di $A^\top A$.*

Dim. $A^\top A = VD^\top U^\top UDV^\top = V(D^\top D)V^\top$. Questa è la diagonalizzazione di Schur di $A^\top A$ (è simmetrica per costruzione, dunque soddisfa le ipotesi del teorema 1.32.) quindi la matrice diagonale $(D^\top D)$ contiene gli autovalori di $A^\top A$ e le colonne di V sono i suoi autovettori. \square

Proposizione 1.40 (SVD compatta) *Sia A una matrice $m \times n$ e sia $U^\top AV = D$ la sua decomposizione ai valori singolari. Siano $\sigma_1, \dots, \sigma_r$ i suoi valori singolari non nulli. Allora*

$$A = U_r D_r V_r^\top \tag{1.30}$$

dove $D_r = \text{diag}(\sigma_1, \dots, \sigma_r)$, U_r è la matrice $m \times r$ composta dalle prime r colonne di U e V_r è la matrice $n \times r$ composta dalle prime r colonne di V .

Possiamo riscrivere: $A = U_r D_r V_r^\top = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ dove \mathbf{u}_i e \mathbf{v}_i sono le colonne di U e V rispettivamente. Ciascuno dei termini della sommatoria è una matrice di rango uno (prodotto di due matrici di rango 1), quindi abbiamo scritto A – che ha rango r – come somma pesata – con pesi pari ai valori singolari corrispondenti – di r matrici di rango 1.

Dunque se volessi ottenere una approssimazione di A rango inferiore sembrerebbe sensato annullare il termine meno pesante, ovvero quello corrispondente a σ_n . Questa intuizione sarà precisata nella prossima proposizione. Prima però osserviamo che:

Osservazione 1.41 $\|A\|_F = \sqrt{\sum_i \sigma_i^2}$.

Segue dalla definizione di norma di Frobenius, dalla (1.23) e dalla proposizione 1.39 (esercizio).

Proposizione 1.42 (Approssimazione di rango inferiore) *Sia A una matrice $m \times n$ di rango r e sia $A = U_r D_r V_r^\top$ la sua decomposizione ai valori singolari compatta. La matrice di rango $k < r$ più prossima ad A in norma di Frobenius è la matrice A_k definita come:*

$$A_k = U_r D_k V_r^\top \quad (1.31)$$

dove $D_k = \text{diag}(\sigma_1, \dots, \sigma_k, \underbrace{0, \dots, 0}_{r-k})$. Inoltre $\|A - A_k\|_F = \sigma_{k+1}$

Si osservi che $A_k = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$.

Un impiego primario per la SVD nel nostro corso sarà la soluzione di sistemi omogenei di equazioni lineari. Consideriamo il sistema $A\mathbf{x} = \mathbf{0}$. Si hanno due casi:

- $\text{r}(A)$ è pieno, quindi $\ker(A) = \{\mathbf{0}\}$ ovvero esiste la sola soluzione banale $\mathbf{x} = \mathbf{0}$
- $\text{r}(A)$ non è pieno, quindi $\dim(\ker(A)) \neq 0$, ovvero il sistema ha soluzioni non banali.

Nella pratica si presenta spesso il caso in cui A possiede rango pieno, quindi a rigore non ci sono soluzioni $\neq \mathbf{0}$, ma questo è dovuto in effetti agli errori di misura o rumore (per esempio punti “quasi” allineati).

In questi casi cerchiamo allora una soluzione *approssimata* non banale, ovvero risolviamo il problema *di minimi quadrati*:

$$\min_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|^2 \quad (1.32)$$

Il vincolo viene posto per evitare la soluzione $\mathbf{x} = \mathbf{0}$ ed il valore 1 è del tutto arbitrario, poiché se \mathbf{x} è una soluzione, anche $\alpha\mathbf{x} \quad \forall \alpha \in \mathbb{R}$ lo è.

Proposizione 1.43 (Soluzione ai minimi quadrati di un sistema di equazioni omogenee) *Sia A una matrice $m \times n$. La soluzione del problema*

$$\min_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|^2 \quad (1.33)$$

è $\mathbf{x} = \mathbf{v}_n$ dove \mathbf{v}_n è l'ultima colonna di V nella decomposizione ai valori singolari di A : $U^\top A V = D$.

Dim. $\min_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|^2 = \min_{\|\mathbf{x}\|=1} \|UDV^\top \mathbf{x}\|^2 = \min_{\|\mathbf{x}\|=1} \|DV^\top \mathbf{x}\|^2 = \min_{\|\mathbf{y}\|=1} \|D\mathbf{y}\|^2$ dopo aver effettuato il cambio di variabile $\mathbf{y} = V^\top \mathbf{x}$ (ricordiamo che V è ortogonale). La funzione obiettivo dunque diventa $\sum (\sigma_i y_i)^2$. Poiché i valori singolari in D sono ordinati, la soluzione di minor costo, non potendo prendere $\mathbf{y} = \mathbf{0}$ è $\mathbf{y} = [0, \dots, 0, 1]^\top$, dunque $\mathbf{x} = \mathbf{v}_n$. \square

Proposizione 1.44 (Problema procustiano ortogonale) *Date due matrici A e B , la soluzione del problema*

$$\min_{W^\top W=I} \|A - WB\|_F^2 \quad (1.34)$$

è $W = VU^\top$ dove $BA^\top = UDV^\top$ la SVD di BA^\top .

Un caso particolare di questo problema si ha quando $B = I$. In tal caso si tratta di trovare la matrice ortogonale più vicina alla matrice A data, ovvero:

$$\min_{W^\top W=I} \|A - W\|_F^2 \quad (1.35)$$

La soluzione equivale a sostituire la matrice D con l'identità nella SVD di A .

Una dimostrazione si trova in [Kanatani, 1993].

A1.10 Pseudoinversa

La pseudoinversa di una matrice $A_{m \times n}$ è la matrice $A_{n \times m}^+$ che soddisfa le seguenti proprietà:

- (i) $AA^+A = A$
- (ii) $A^+AA^+ = A^+$
- (iii) $(AA^+)^\top = AA^+$
- (iv) $(A^+A)^\top = A^+A$

Una tale matrice possiede inoltre le seguenti proprietà:

- (i) $\forall A_{m \times n} \exists! A_{n \times m}^+$
- (ii) se $r(A) = n \quad A^+ = (A^\top A)^{-1}A^\top$
- (iii) se $r(A) = n = m \quad A^+ = A^{-1}$

Nel caso in cui A non abbia rango pieno, la sua pseudoinversa si può calcolare mediante la SVD di A :

$$A = USV^\top \quad (1.36)$$

dove S è una matrice diagonale con elementi non negativi che prendono il nome di *valori singolari*. La sua pseudoinversa è:

$$A^+ = VS^+U^\top \quad (1.37)$$

dove S^+ è una matrice diagonale i cui elementi sono il reciproco di quelli di S non nulli, oppure 0.

La soluzione ai minimi quadrati di un sistema lineare sovradeterminato $A\mathbf{x} = \mathbf{b}$ si ottiene mediante la pseudoinversa di A :

$$\mathbf{x} = A^+\mathbf{b}. \quad (1.38)$$

Se A ha rango pieno (pari al numero di colonne) allora la soluzione è unica. Altrimenti, quella calcolata con la pseudoinversa è quella di norma minima.

Nel caso di un sistema omogeneo $A\mathbf{x} = \mathbf{0}$, la soluzione generale si esprime sempre tramite la pseudoinversa come:

$$\mathbf{x} = (I - A^+A)\mathbf{q}. \quad (1.39)$$

dove \mathbf{q} è un vettore arbitrario di dimensione appropriata. Se A ha rango pieno, allora $A^+A = I$ ed il sistema possiede una sola soluzione (banale) $\mathbf{x} = \mathbf{0}$. Altrimenti vi sono infinite soluzioni al variare di \mathbf{q} .

A1.11 Prodotto esterno

Si tratta di un prodotto tra due vettori che restituisce un vettore, e si definisce solo in \mathbb{R}^3 .

Definizione 1.45 (Prodotto esterno) Il prodotto esterno di due vettori $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$ si definisce come il vettore:

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} \det \begin{pmatrix} a_2 & a_3 \\ b_2 & b_3 \end{pmatrix} \\ -\det \begin{pmatrix} a_1 & a_3 \\ b_1 & b_3 \end{pmatrix} \\ \det \begin{pmatrix} a_1 & a_2 \\ b_1 & b_2 \end{pmatrix} \end{bmatrix} \quad (1.40)$$

Il prodotto esterno serve (tra le altre cose) per controllare se due vettori differiscono solo per una costante moltiplicativa:

Osservazione 1.46 Dati due vettori $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$, $\mathbf{a}, \mathbf{b}, \neq \mathbf{0}$ si ha: $\mathbf{a} \times \mathbf{b} = \mathbf{0} \iff \mathbf{a} = \lambda \mathbf{b}, \lambda \in \mathbb{R}$.

Infatti, l'ipotesi è equivalente ad affermare che tutti i determinanti di ordine due estratti da $[\mathbf{a}, \mathbf{b}]$ sono nulli, quindi essa ha rango uno, da cui segue la tesi, e viceversa.

Il prodotto esterno è associato naturalmente ad una matrice antisimmetrica:

Osservazione 1.47 Dato un vettore $\mathbf{a} \in \mathbb{R}^3$, la matrice

$$[\mathbf{a}]_{\times} \triangleq \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (1.41)$$

agisce come il prodotto esterno per \mathbf{a} , ovvero: $[\mathbf{a}]_{\times} \mathbf{b} = \mathbf{a} \times \mathbf{b}$.

La matrice $[\mathbf{a}]_{\times}$ è antisimmetrica, singolare, e $\ker([\mathbf{a}]_{\times}) = \mathbf{a}$, poiché $[\mathbf{a}]_{\times} \mathbf{a} = \mathbf{a} \times \mathbf{a} = \mathbf{0}$.

Proposizione 1.48 Sia A una matrice 3×3 con $\det(A) = 1$. Si ha che:

$$[A^{-1} \mathbf{u}]_{\times} = A^{\top} [\mathbf{u}]_{\times} A \quad (1.42)$$

Proposizione 1.49 (Prodotto triplo) Dati tre vettori $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^3$, si ha

$$\det([\mathbf{a}, \mathbf{b}, \mathbf{c}]) = \mathbf{a}^{\top} (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \times \mathbf{b})^{\top} \mathbf{c}. \quad (1.43)$$

Il termine destro dell'equazione prende anche il nome di prodotto triplo di $\mathbf{a}, \mathbf{b}, \mathbf{c}$.

La dimostrazione segue immediatamente usando l'espansione di Laplace e la definizione del prodotto esterno (esercizio).

Il prodotto triplo (come il determinante) serve per controllare se tre vettori sono linearmente dipendenti.

Proposizione 1.50 (Formula di Lagrange) *Dati quattro vettori $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{R}^3$, si ha*

$$(\mathbf{a} \times \mathbf{b})^\top (\mathbf{c} \times \mathbf{d}) = (\mathbf{c}^\top \mathbf{a})(\mathbf{b}^\top \mathbf{d}) - (\mathbf{d}^\top \mathbf{a})(\mathbf{b}^\top \mathbf{c}) \quad (1.44)$$

Proposizione 1.51 (Equazione vettoriale) *Dati tre vettori $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^3$ tali che $\mathbf{a}^\top (\mathbf{b} \times \mathbf{c}) = 0$ i due scalari μ, λ tali che*

$$\mathbf{c} = \mu \mathbf{a} - \lambda \mathbf{b} \quad (1.45)$$

si calcolano con:

$$\mu = \frac{(\mathbf{c} \times \mathbf{b})^\top (\mathbf{a} \times \mathbf{b})}{\|\mathbf{a} \times \mathbf{b}\|^2}. \quad (1.46)$$

La formula per λ è analoga.

Dim. L'ipotesi $\mathbf{a}^\top (\mathbf{b} \times \mathbf{c}) = 0$ garantisce l'esistenza della soluzione, poiché uno dei vettori e combinazione lineare degli altri due. Per quanto riguarda la formula (1.46), si ha: $\mathbf{c} \times \mathbf{b} = \mu \mathbf{a} \times \mathbf{b} - \lambda \mathbf{b} \times \mathbf{b} = \mu \mathbf{a} \times \mathbf{b} \Rightarrow (\mathbf{c} \times \mathbf{b})^\top (\mathbf{a} \times \mathbf{b}) = \mu (\mathbf{a} \times \mathbf{b})^\top (\mathbf{a} \times \mathbf{b}) \Rightarrow$ tesi. \square

Si dimostra anche (in [Kanatani, 1993]) che la formula (1.46) risolve anche il problema di minimi quadrati:

$$\min \|\mu \mathbf{a} - \lambda \mathbf{b} - \mathbf{c}\|^2 \quad (1.47)$$

quando $\mathbf{a}^\top (\mathbf{b} \times \mathbf{c}) \neq 0$.

A1.12 Prodotto di Kronecker

Definizione 1.52 (Prodotto di Kronecker) *Siano A una matrice $m \times n$ e B una matrice $p \times q$. Il prodotto di Kronecker[†] di A e B è la matrice $mp \times nq$ definita da*

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix}. \quad (1.48)$$

[†] Implementato dalla funzione `kron` in MATLAB.

Si noti che il prodotto di Kronecker è definito per qualunque coppia di matrici. Proprietà:

- (i) $(A \otimes B) \otimes C = A \otimes (B \otimes C)$;
- (ii) $(A \otimes B)(C \otimes D) = AC \otimes BD$ se le dimensioni sono compatibili.
- (iii) $A \otimes B \neq B \otimes A$
- (iv) $(A \otimes B)^\top = A^\top \otimes B^\top$.
- (v) $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$.

Proposizione 1.53 *Gli autovalori di $A \otimes B$ sono il prodotto tensoriale (ovvero “tutti con tutti”) degli autovalori di A per gli autovalori di B .*

Dim. Siano $A = SLS^{-1}$ e $B = TMT^{-1}$ le decomposizioni di Shur di A e B rispettivamente, dove L ed M sono matrici triangolari superiori i cui elementi diagonali sono gli autovalori di A e B rispettivamente.

$$A \otimes B = (SLS^{-1}) \otimes (TMT^{-1}) = (S \otimes T)(L \otimes M)(S^{-1} \otimes T^{-1}). \quad (1.49)$$

Poiché $A \otimes B$ e $L \otimes M$ sono simili hanno gli stessi autovalori, e poiché $L \otimes M$ è triangolare superiore gli autovalori di $A \otimes B$ sono gli elementi diagonali di $L \otimes M$. \square

Segue immediatamente dalla proposizione il seguente:

Corollario 1.54 (Rango del prodotto di Kronecker)

$$\operatorname{r}(A \otimes B) = \operatorname{r}(A) \operatorname{r}(B). \quad (1.50)$$

La *vettorizzazione* di una matrice è una trasformazione lineare che converte la matrice in un vettore (colonna):

Definizione 1.55 (Vettorizzazione) *La vettorizzazione di una matrice A $m \times n$, denotata da $\operatorname{vec}(A)$, è il vettore $mn \times 1$ che si ottiene impilando le colonne di A una sotto l’altra.*

La connessione tra il prodotto di Kronecker e la vettorizzazione è data dalla seguente (importante) relazione:

$$\operatorname{vec}(AXB) = (B^\top \otimes A) \operatorname{vec}(X) \quad (1.51)$$

per matrici A, B, X di dimensioni compatibili. Questa sarà molto utile per estrarre l’incognita X da una equazione matriciale.

Quando si trattano matrici simmetriche la vettorizzazione può essere compattata, considerando solo gli elementi non ripetuti. A questo scopo si definisce la:

Definizione 1.56 (Semi-vettorizzazione) *La semi-vettorizzazione di una matrice $n \times n$ simmetrica – denotata con $\text{vech}(A)$ – è il vettore $n(n + 1)/2 \times 1$ ottenuto vettorizzando solo la parte triangolare inferiore di A .*

Per passare da $\text{vech}(A)$ a (A) esiste una opportuna matrice $n^2 \times n(n + 1)/2$, chiamata *matrice di duplicazione* D_n tale che $D_n \text{vech}(A) = \text{vec}(A)$.

Bibliografia

- Kanatani K. (1993). *Geometric Computation for Machine Vision*. Oxford University Press.
- Strang G. (1988). *Linear Algebra and its applications*. Harcourt Brace & Co.

Appendice 2

Nozioni di Geometria proiettiva

Per giustificare l'impiego della Geometria proiettiva e dunque delle coordinate omogenee nello studio della visione computazionale partiamo da un approccio storico. Studiamo, come fece Leon Battista Alberti nel “De Pictura”, la proiezione prospettica di un piano. La proiezione prospettica è il tipo di proiezione che nell'occhio dei vertebrati e nelle fotocamere governa la formazione dell'immagine.

A2.1 Proiezione prospettica

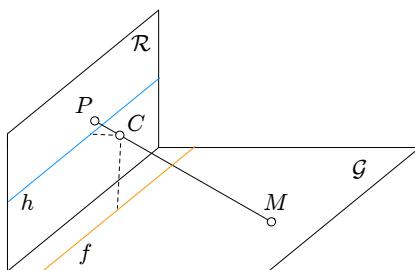


Fig. A2.1. Proiezione prospettica di un piano di terra \mathcal{G} ortogonale al quadro \mathcal{R} . Le linee h ed f sono casi speciali.

Definizione 2.1 (Proiezione prospettica) *La proiezione prospettica porta punti M dello spazio 3D sul piano \mathcal{R} – piano immagine o quadro – intersecando la retta che passa per M e C – centro di proiezione – con \mathcal{R} .*

Gli elementi fondamentali sono il piano \mathcal{R} ed il centro di proiezione C . Le rette dello spazio vengono trasformate in rette in \mathcal{R} . Se consideriamo ora

la proiezione di un piano \mathcal{G} ortogonale ad \mathcal{R} (piano di terra) constatiamo che

- la retta f determinata dalla intersezione del piano \mathcal{G} con il piano parallelo a \mathcal{R} e contenente C non si proietta su \mathcal{R} ;
- la retta h intersezione di \mathcal{R} con il piano parallelo a \mathcal{G} e passante per C non è la proiezione di alcuna retta del piano \mathcal{G} .

Questa situazione non è soddisfacente. Per ovviare si aggiunge al piano euclideo (che modella \mathcal{R} e \mathcal{G}) una retta “ideale”, chiamata retta *all’infinito*.

Si può allora dire che f si proietta sulla retta all’infinito di \mathcal{R} e che h è la proiezione della retta all’infinito di \mathcal{G} . Possiamo pensare alla retta all’infinito di un piano come il luogo dove si incontrano (metaforicamente) le rette parallele di quel piano. Infatti:

Osservazione 2.2 *Con riferimento alla figura A2.1, le immagini di rette parallele in \mathcal{G} si intersecano in un punto di h .*

Vediamo, per esempio, il caso speciale di rette contenute in \mathcal{G} ed ortogonali a \mathcal{R} (chiamate *linee di profondità* nella costruzione Albertiana). La loro immagine prospettica converge al *punto centrico* O , cioè la proiezione ortogonale di C su \mathcal{R} (che $\in h$). Infatti, i piani contenenti C e qualunque linea di profondità passano tutti per la semiretta uscente da C e parallela a \mathcal{G} , chiamata *raggio principale*, e quindi intersecano il quadro \mathcal{R} in una linea che passa per il punto centrico O (figura A2.2)

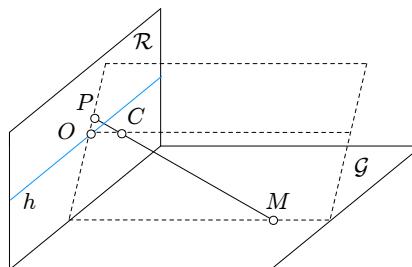


Fig. A2.2. La linea di profondità contenente M si proietta su una retta contenente P e passante per il punto centrico O .

Definizione 2.3 (Piano proiettivo) *Il piano proiettivo \mathbb{P}^2 si definisce come l’unione del piano euclideo \mathbb{R}^2 con la retta all’infinito l_∞ :*

$$\mathbb{P}^2 = \mathbb{R}^2 \cup l_\infty \quad (2.1)$$

Analogamente si può definire lo spazio proiettivo \mathbb{P}^n . In \mathbb{P}^n i punti all'infinito costituiscono un iperpiano.

Nel piano proiettivo due linee hanno un punto di intersezione anche quando sono parallele, ma il punto è *all'infinito*. Ogni retta, quindi, possiede un punto all'infinito a cui ci si avvicina viaggiando lunga la retta in una certa direzione, e tutte le rette ad essa parallele la incontrano in quel punto. Per ora il termine “punto all'infinito” è solo una metafora, ma possiamo dargli un significato matematico preciso. Un modo per farlo è identificare i punti all'infinito con la direzione assoluta di una retta. Consideriamo le seguenti tre proprietà delle rette:

- (i) esiste un'unica retta passante per due punti distinti;
- (ii) esiste un'unica retta passante per un punto ed avente una direzione data;
- (iii) due rette distinte hanno un unico punto in comune oppure hanno la stessa direzione.

Se ci accordiamo di sostituire “direzione” con “punto all'infinito” e poi raggruppiamo tutti i punti all'infinito in un retta all'infinito otteniamo che le precedenti tre proprietà equivalgono a:

- (i) esiste un'unica retta passante per due punti distinti;
- (ii) due rette distinte hanno un unico punto in comune.

Questi sono i due l'assiomi di incidenza nel piano proiettivo. L'aspetto importante di questi due assiomi è che sono duali.

A2.2 Coordinate omogenee

I punti all'infinito sono punti ideali che vengono aggiunti all'usuale piano euclideo. Come i punti complessi, che vengono introdotti per mezzo della loro rappresentazione algebrica, lo stesso si può fare per i punti all'infinito, usando le *coordinate omogenee*.

Fissati due assi di riferimento, tutti i punti di \mathbb{R}^2 ammettono un'unica rappresentazione come coppia di numeri reali (x, y) . Due rette di equazioni $ax + by + c = 0$ e $a'x + b'y + c' = 0$ si incontrano, se non sono parallele, nel punto di \mathbb{R}^2 di coordinate (regola di Cramer)

$$x = -\frac{\det \begin{pmatrix} c & b \\ c' & b' \end{pmatrix}}{\det \begin{pmatrix} a & b \\ a' & b' \end{pmatrix}} \quad y = -\frac{\det \begin{pmatrix} a & c \\ a' & c' \end{pmatrix}}{\det \begin{pmatrix} a & b \\ a' & b' \end{pmatrix}} \quad (2.2)$$

Quando le rette sono parallele il denominatore si annulla, per cui non possiamo dare un significato al rapporto (diventa “infinito”). Notiamo però che, delle tre quantità che compaiono nella (2.2) almeno una è diversa da zero (se le rette sono distinte). Proponiamo allora di tenere queste tre quantità come rappresentazione dei punti di \mathbb{P}^2 .

In sostanza, invece di rappresentare i punti del piano tramite coppie di coordinate (x, y) , ci accordiamo di rappresentarli con terne di *coordinate omogenee* (u, v, w) , collegate alle coordinate di \mathbb{R}^2 dalla relazione $x = u/w$ e $y = v/w$.

Notiamo che terne proporzionali rappresentano lo stesso punto – per questo si chiamano omogenee – e che la terna $(0, 0, 0)$ è esclusa. Quando $w \neq 0$ la terna rappresenta un punto *effettivo*, mentre ogni terna per cui $w = 0$ rappresenta un punto all’infinito.

Definizione 2.4 (Piano proiettivo) *Il piano proiettivo \mathbb{P}^2 consiste delle terne di numeri reali $(u, v, w) \neq (0, 0, 0)$ modulo la seguente relazione di equivalenza: $(u, v, w) \sim \lambda(u, v, w) \quad \forall \lambda \neq 0$. La terna (u, v, w) sono le coordinate omogenee del punto di \mathbb{P}^2 .*

Analogamente si può definire lo spazio proiettivo \mathbb{P}^n .

A2.2.1 Equazione della retta

Nel familiare sistema di riferimento cartesiano nel piano euclideo, ciascuna coppia di numeri identifica un punto, e, viceversa, ogni punto è rappresentato da una sola coppia. Questo non è vero per le rette: una equazione lineare $ax + by + c = 0$ con a e b non contemporaneamente nulli, rappresenta una retta, ma la stessa retta è rappresentata da una classe di equazioni lineari $\lambda ax + \lambda by + \lambda c = 0, \lambda \neq 0$.

Una equazione omogenea lineare $au + bv + cw = 0$ rappresenta una retta di \mathbb{P}^2 . Questa coincide con la retta effettiva $ax + by + c = 0$ se $a \neq 0 \vee b \neq 0$. Altrimenti, l’equazione $w = 0$ rappresenta la retta (ideale) all’infinito λ_∞ .

Una retta nel piano proiettivo è rappresentata da una terna di numeri (a, b, c) , modulo l’equivalenza, come un punto di \mathbb{P}^2 . Nel piano proiettivo punti e rette sono elementi duali.

Confondendo le terne con i vettori di \mathbb{R}^3 , l’equazione della retta $au + bv + cw = 0$ si può scrivere

$$\mathbf{x}^\top \mathbf{y} = 0. \tag{2.3}$$

dove $\mathbf{x} = [u, v, w]^\top$ e $\mathbf{y} = [a, b, c]^\top$. Si noti la simmetria tra il ruolo del vettore che rappresenta la retta e quello che rappresenta il punto.

Proposizione 2.5 *La retta passante per due punti distinti \mathbf{p}_1 e \mathbf{p}_2 è rappresentata dalla terna $\mathbf{p}_1 \times \mathbf{p}_2$.*

Dim. La retta passa per entrambi i punti, infatti: $\mathbf{p}_1^\top(\mathbf{p}_1 \times \mathbf{p}_2) = 0$ e $\mathbf{p}_2^\top(\mathbf{p}_1 \times \mathbf{p}_2) = 0$. \square

Per quanto riguarda il punto determinato da due rette vale la proposizione duale, che si può anche dedurre direttamente dalla (2.2) (esercizio).

Proposizione 2.6 *La retta l passante per due punti distinti \mathbf{p}_1 e \mathbf{p}_2 ha equazione parametrica:*

$$l = \{\mathbf{x} \in \mathbb{P}^2 \mid \mathbf{x} = \alpha\mathbf{p}_1 + \beta\mathbf{p}_2, \quad \alpha, \beta \in \mathbb{R}\} \quad (2.4)$$

Dim. $\mathbf{x} = \alpha\mathbf{p}_1 + \beta\mathbf{p}_2 \iff \det([\mathbf{x}, \mathbf{p}_1, \mathbf{p}_2]) = 0 \iff \mathbf{x}^\top(\mathbf{p}_1 \times \mathbf{p}_2) = 0$. \square

Ponendo $\lambda = \beta/\alpha$ e accettando la convenzione che $\mathbf{p}_1 + \lambda\mathbf{p}_2 = \mathbf{p}_2$ quando $\lambda = \infty$, la retta l ha equazione parametrica

$$l = \{\mathbf{x} \in \mathbb{P}^2 \mid \mathbf{x} = \mathbf{p}_1 + \lambda\mathbf{p}_2, \quad \lambda \in \mathbb{R} \cup \{\infty\}\}. \quad (2.5)$$

A2.3 Trasformazioni

Definizione 2.7 (Proiettività) *Una proiettività o trasformazione proiettiva $f : \mathbb{P}^n \rightarrow \mathbb{P}^n$ è una applicazione lineare in coordinate omogenee:*

$$f : \mathbf{x} \rightarrow H\mathbf{x} \quad (2.6)$$

dove H è una matrice $(n+1) \times (n+1)$ non singolare.

Le proiettività prendono anche il nome di *omografie* o *collineazioni* (in quanto preservano la collinearità dei punti). Le proiettività formano un gruppo, che indichiamo con \mathcal{G}_P .

A causa della rappresentazione omogenea dei punti, le due matrici H e λH con $\lambda \in \mathbb{R}$, $\lambda \neq 0$ rappresentano la stessa proiettività.

Definizione 2.8 (Affinità) *Le proiettività che trasformano punti effettivi in punti effettivi e punti ideali in punti ideali si chiamano affinità o trasformazioni affini.*

Si può dire, alternativamente, che una affinità *preserva* i punti dell'iperpiano all'infinito. Le affinità formano un sottogruppo di \mathcal{G}_P .

La proiezione prospettica del piano \mathcal{G} definita in precedenza è una proiettività di \mathbb{P}^2 (abbiamo introdotto \mathcal{G} proprio per rendere biunivoca l'applicazione!)

Proposizione 2.9 *Le affinità sono tutte e sole le proiettività in cui*

$$H = \begin{bmatrix} A_{n \times n} & \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix} \quad (2.7)$$

Nell'iperpiano all'infinito giace una conica speciale, la *conica assoluta* che ha equazione $x_1^2 + x_2^2 + \dots + x_n^2 = 0 \wedge x_{n+1} = 0$.

Definizione 2.10 (Similarità) *Le proiettività che preservano la conica assoluta si chiamano similarità.*

Le similarità formano un sottogruppo di \mathcal{G}_P .

Proposizione 2.11 *Le similarità sono tutte e sole le proiettività in cui*

$$H = \begin{bmatrix} sR_{n \times n} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad (2.8)$$

dove $R_{n \times n}$ è ortogonale ed s è uno scalare.

L'azione della similarità nello spazio euclideo \mathbb{R}^n si può scrivere come:

$$\tilde{\mathbf{x}} \rightarrow sR_{n \times n}\tilde{\mathbf{x}} + \mathbf{t} \quad (2.9)$$

dove $\mathbf{x} = [\tilde{\mathbf{x}}, 1]^\top$.

Nel caso speciale $s = 1$ si ottiene una *trasformazione rigida* o euclidea.

La seguente tabella riassume la gerarchia di trasformazioni, nel caso di \mathbb{P}^2 .

Trasformazione	G.d.l.	Matrice	Distorsione	Preserva
Proiettività	8	$\begin{bmatrix} H_{1,1} & H_{1,2} & H_{1,3} \\ H_{2,1} & H_{2,2} & H_{2,3} \\ H_{3,1} & H_{3,2} & H_{3,3} \end{bmatrix}$	 	collinearità
Affinità	6	$\begin{bmatrix} H_{1,1} & H_{1,2} & H_{1,3} \\ H_{2,1} & H_{2,2} & H_{2,3} \\ 0 & 0 & 1 \end{bmatrix}$	 	parallelismo
Similarità	4	$\begin{bmatrix} sR & \mathbf{t} \\ 0 & 1 \end{bmatrix}$	 	angoli
euclidea	3	$\begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix}$	 	lunghezza

Per saperne di più, un ottimo testo di geometria proiettiva analitica è il [Semple e Kneebone, 1952]. In [Ayres, 1967] si trovano spunti interessanti.

Bibliografia

- Ayres F. (1967). *Theory and Problems of Projective Geometry*. Schaum's Outline Series in Mathematics. McGraw-Hill.
- Semple J. G.; Kneebone G. T. (1952). *Algebraic projective geometry*. Oxford University Press.

Appendice 3

Miscellanea di nozioni utili

Questo capitolo tratta di rappresentazione delle rotazioni nello spazio, di regressione e di tecniche numeriche per la soluzioni di sistemi di equazioni non lineari.

A3.1 Rotazioni

La rappresentazione mediante angoli di Eulero si basa sul fatto che ogni rotazione in \mathbb{R}^3 attorno ad una asse passante per l'origine può essere considerata come una sequenza di tre rotazioni attorno agli assi principali. Il sistema di Eulero ha il vantaggio di usare solo tre parametri, ma sfortunatamente introduce delle discontinuità quando si passa da π a $-\pi$. Alcune varianti del sistema di Eulero ruotano gli assi principali sia nel sistema di riferimento della fotocamera che in quello del modello mentre la rappresentazione RPY o HPY (*Roll o Heading, Pitch, Yaw*) usa solo un sistema di riferimento. Considerando ora il sistema di riferimento RPY, siano ψ, θ, ϕ gli angoli di rotazione lungo gli assi x, y e z . Sia $R_{asse,angolo}$ la matrice operatore 3x3 che compie una rotazione attorno a un dato asse. Le tre matrici di rotazione lungo gli assi sono

quindi:

$$\begin{aligned} R_{x,\psi} = yaw &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{bmatrix} \\ R_{y,\theta} = pitch &= \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \\ R_{z,\phi} = roll &= \begin{bmatrix} \cos \phi & -\sin \theta & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (3.1)$$

Applicando questi operatori in sequenza da destra a sinistra, è possibile calcolare la matrice di rotazione:

$$R = R_{z,\phi} R_{y,\theta} R_{x,\psi} \quad (3.2)$$

Secondo un noto teorema di Eulero, ogni rotazione 3D si può scrivere come una rotazione di un certo angolo θ attorno ad un asse passante per l'origine, individuato da un versore \mathbf{u} . La corrispondente matrice di rotazione si può ottenere da θ e \mathbf{u} con la formula di Rodriguez:

$$R = I + \sin \theta N + (1 - \cos \theta) N^2 \quad (3.3)$$

dove $N = [\mathbf{u}]_\times$ è la matrice antisimmetrica associata ad \mathbf{u} . Anche in questo caso bastano tre numeri per codificare una rotazione, visto che \mathbf{u} è unitario, si può usare il vettore:

$$\mathbf{r} = \theta \mathbf{u} \quad (3.4)$$

Si vedano le funzioni MATLAB `eul`, `ieul`, `kan` e `ikan`.

A3.1.1 Differenziale della rotazione

Ripartiamo dalla formula di Rodriguez e consideriamo l'applicazione della rotazione di asse \mathbf{u} ed angolo θ al punto \mathbf{x} :

$$\mathbf{x}' = (\mathbf{I} + \sin \theta N + (1 - \cos \theta) N^2) \mathbf{x} \quad (3.5)$$

e quindi la variazione di \mathbf{x} vale:

$$\mathbf{x}' - \mathbf{x} = (\sin \theta N + (1 - \cos \theta) N^2) \mathbf{x} \quad (3.6)$$

Per calcolare la parte lineare dell'incremento $\mathbf{x}' - \mathbf{x}$ considero una rotazione infinitesimale $d\theta$ e le serie delle funzioni trigonometriche troncate al primo termine, da cui: $\sin d\theta = d\theta$ e $\cos d\theta = 1$.

Quindi, detta $d\mathbf{x}$ la parte lineare di $\mathbf{x}' - \mathbf{x}$, si ha

$$d\mathbf{x} = d\theta N\mathbf{x} = d\theta \mathbf{u} \times \mathbf{x} \quad (3.7)$$

L'azione di una rotazione infinitesimale si rappresenta con il prodotto esterno per una matrice antisimmetrica $[d\theta \mathbf{u}]_\times$.

Dalla formula (3.7), si ricava che (cinematica del punto materiale)

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} = \frac{d\theta}{dt} \mathbf{u} \times \mathbf{x} \triangleq \boldsymbol{\omega} \times \mathbf{x} \quad (3.8)$$

dove $\boldsymbol{\omega}$ è un vettore che rappresenta la velocità angolare del punto \mathbf{x} .

Inoltre, dalla formula (3.7) ricaviamo anche:

$$\frac{d\mathbf{x}}{d\theta} = \mathbf{u} \times \mathbf{x} \quad (3.9)$$

che ci dice che la derivata di un vettore rispetto all'angolo di rotazione si ottiene come prodotto esterno dell'asse di rotazione e del vettore cui la rotazione è applicata.

A3.2 Regressione

La regressione, ovvero l'adattamento di un modello ad un insieme di dati rumorosi, è uno strumento statistico importante frequentemente impiegato in visione computazionale per una grande varietà di scopi. In questo paragrafo introdurremo alcuni concetti riguardanti la statistica robusta e descriveremo brevemente alcuni estimatori.

Lo scopo dell'analisi della regressione è di adattare (*fit*) un modello (equazioni) alle osservazioni delle variabili. Nella *regressione lineare* si considera il seguente modello:

$$y_i = x_{i1}\theta_1 + \dots + x_{ip}\theta_p + e_i \quad \text{con } i = 1, \dots, n \quad (3.10)$$

dove n è la dimensione del campione, x_{i1}, \dots, x_{ip} sono le variabili indipendenti, y_i è la variabile dipendente, che viene misurata, e e_i è il termine dell'errore presente nella misurazione. Applicando uno stimatore della regressione a questo insieme di dati otteniamo dei coefficienti di regressione (stime) $\hat{\boldsymbol{\theta}} = [\hat{\theta}_1, \dots, \hat{\theta}_p]^\top$.

A3.2.1 Minimi quadrati

Il metodo dei minimi quadrati, o Least squares (LS), è il più comune stimatore di regressione. Lo scopo di questo metodo è risolvere:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^n \left(\frac{r_i(\boldsymbol{\theta})}{\sigma_i} \right)^2 \quad (3.11)$$

dove $r_i = y_i - x_{i1}\hat{\theta}_1 + \dots + x_{ip}\hat{\theta}_p$ è il residuo e σ_i è la sua varianza. I coefficienti di regressione possono essere ottenuti mediante la pseudoinversa della matrice di osservazione X :

$$\hat{\boldsymbol{\theta}} = X^+ \mathbf{y} \quad (3.12)$$

dove

$$X = \begin{bmatrix} x_{11} & \dots & x_{1p} \\ \vdots & & \vdots \\ x_{n1} & \dots & x_{np} \end{bmatrix} \quad \text{e} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}. \quad (3.13)$$

Si dimostra che la stima ai minimi quadrati è ottima† nell'ipotesi di errore e_i Gaussiano. Quando questa assunzione non è applicabile, la stima ai minimi quadrati può essere completamente fuorviante (vedi paragrafo A3.2.3).

Anche il filtro di Kalman è uno stimatore ai minimi quadrati dello stato di un sistema dinamico lineare.

A3.2.2 Filtro di Kalman

Il filtro di Kalman è uno strumento che produce una stima ottima dello stato di un sistema dinamico, sulla base di osservazioni (misure) affette da rumore (Gaussiano) e di un modello della dinamica del sistema, anch'esso affetto da incertezza. Si veda [Bar-Shalom e Fortmann, 1988] per una trattazione estensiva.

Modello del sistema dinamico. Modello della dinamica (ovvero come evolve lo stato nel tempo):

$$\mathbf{x}(t+1) = \Phi \mathbf{x}(t) + \mathbf{w}(t), \quad \text{Cov}[\mathbf{w}] = Q \quad (3.14)$$

Modello delle misure:

$$\mathbf{z}(t) = H \mathbf{x}(t) + \mathbf{v}(t), \quad \text{Cov}[\mathbf{v}] = R \quad (3.15)$$

† nel senso che lo stimatore che si ottiene è non polarizzato (unbiased) e di minima varianza

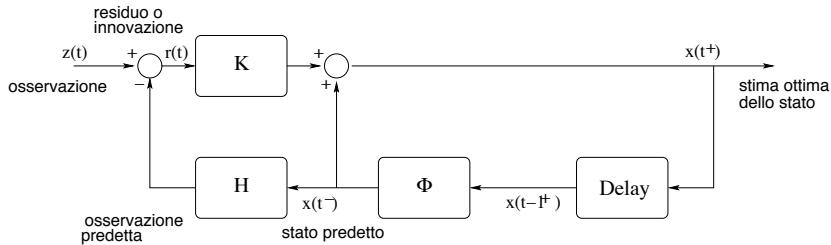


Fig. A3.1. Filtro di Kalman.

Filtro. Predizione o estrapolazione della stima dello stato e della matrice di covarianza dell'errore

$$\hat{\mathbf{x}}(t^-) = \Phi \hat{\mathbf{x}}(t-1^+) \quad (3.16)$$

$$P(t^-) = \Phi P(t-1^+) + Q \quad (3.17)$$

Calcolo del guadagno del filtro:

$$K = P(t^-)H^T[R + HP(t^-)H^T]^{-1} \quad (3.18)$$

oppure anche

$$K = P(t^-)H^T R^{-1} \quad (3.19)$$

Aggiornamento della stima dello stato:

$$\hat{\mathbf{x}}(t^+) = \hat{\mathbf{x}}(t^-) + K[\mathbf{z}(t) - H\hat{\mathbf{x}}(t^-)] \quad (3.20)$$

Il termine $[\mathbf{z}(t) - H\hat{\mathbf{x}}(t^-)]$ prende il nome di residuo o innovazione. Si noti che il termine $[R + HP(t^-)H^T]$ nella formula del guadagno è la covarianza dell'innovazione.

Aggiornamento della covarianza della stima dello stato:

$$P(t^+) = (I - KH)P(t^-). \quad (3.21)$$

Si preferisce la seguente formula, che garantisce che $P(t^+)$ sia simmetrica e definita positiva

$$P(t^+) = (I - KH)P(t^-)(I - KH)^T + KRK^T. \quad (3.22)$$

Sono stati omessi gli indici temporali per le matrici Φ, H, K, Q, R .

A3.2.3 Outlier e robustezza

La tecnica più diffusa di regressione è quella dei minimi quadrati, la quale è ottima solo nel caso in cui gli errori che affliggono i dati siano distribuiti in modo Gaussiano. Nel caso in cui la distribuzione venga contaminata da campioni periferici (*outliers*), ovvero osservazioni lontane dalla tendenza generale dei dati (che si localizzano nella coda della gaussiana), questi possono corrompere in maniera arbitrariamente grande la stima ai minimi quadrati (si veda l'esempio in figura A3.2). Gli *outliers* derivano nella maggior parte dei casi da grossi errori di misura o da rumore impulsivo che affligge i dati.

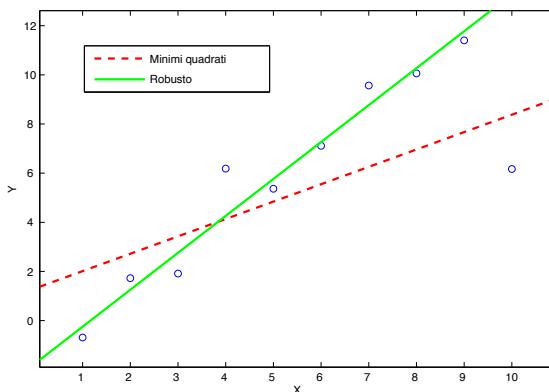


Fig. A3.2. Esempio di effetto di un *outlier*: retta di regressione ai minimi quadrati e retta di regressione robusta (con un M-stimatore).

Al fine di rimediare a questi inconvenienti, sono state sviluppate tecniche statistiche di *regressione robusta* che forniscono risultati affidabili anche in presenza di dati contaminati. Un approccio alternativo è quello di costruire strumenti di *outlier diagnostics*.

Diagnistica e regressione robusta hanno lo stesso obiettivo, ma operano in modo opposto. Usando la diagnostica si cerca di identificare gli *outlier* e in seguito si adatta il modello ai dati “ripuliti” con tecniche classiche di regressione, come ad esempio i minimi quadrati. Diversamente, l'approccio di regressione robusta cerca prima di calcolare un modello di regressione che segue il comportamento della maggioranza dei dati e in seguito scopre gli *outliers* come quei dati che presentano un grande residuo rispetto al modello di regressione robusta. Vi sono situazioni

in cui entrambi gli approcci determinano gli stessi risultati, e quando si verificano differenze la loro interpretazione può risultare soggettiva.

Tre concetti sono solitamente impiegati per valutare un metodo di regressione: l'efficienza relativa, il punto di *breakdown* e la complessità computazionale.

L'*efficienza relativa* di un metodo di regressione è definita come il rapporto tra varianza minima raggiungibile per i parametri stimati e la varianza effettiva fornita dal metodo dato. L'efficienza dipende anche dalla distribuzione del rumore. Per esempio, in presenza di rumore Gaussiano la media campionaria ha un'efficienza asintotica di 1.

Il *punto di breakdown* di un metodo di regressione è la percentuale di *outliers* che lo stimatore riesce a tollerare nei dati, senza perdere precisione nella stima.

Nel caso dei minimi quadrati, ad esempio, il punto di *breakdown* è uguale a

$$\epsilon_n^*(T, Z) = \frac{1}{n}, \quad (3.23)$$

quindi tende a zero all'aumentare della dimensione n del campione, sicché si può affermare che il criterio dei minimi quadrati ha un punto di *breakdown* dello 0%. Anche il punto di *breakdown* della media campionaria è 0%, dato che un singolo grande *outlier* può distorcere la stima. Invece la mediana rimane immutata se meno di metà dei dati è contaminato, quindi possiede valore di *breakdown* 50%.

A3.2.4 Regressione robusta

A3.2.4.1 M-stimatori

Faremo una rapida panoramica sui principali stimatori robusti, la cui teoria fu sviluppata negli anni settanta. Gli stimatori robusti di base sono classificati come M-stimatori e R-stimatori.

Gli *M-stimatori* sono uno dei metodi di regressione robusta più usati. L'idea è quella di sostituire i quadrati dei residui nel metodo dei minimi quadrati tradizionale con una funzione dei residui stessi:

$$\min \sum_i \rho(r_i/\sigma_i), \quad (3.24)$$

dove σ_i è la varianza del residuo r_i e ρ è una funzione simmetrica, sub-quadratiche, avente un unico minimo in zero, chiamata *loss function* o funzione di penalità. Un esempio di funzione di penalità è la funzione di

Cauchy, il cui grafico è visibile in figura A3.3:

$$\rho(x) = \frac{b^2}{2} \log\left(1 + \left(\frac{x}{b}\right)^2\right). \quad (3.25)$$

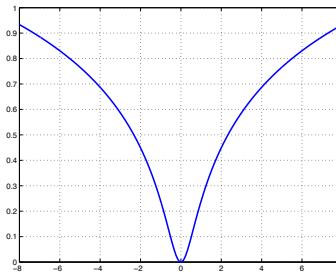


Fig. A3.3. Funzione di penalità di Cauchy.

Notiamo che gli M-estimatori hanno una ridotta sensibilità ai campioni periferici, grazie alla funzione di penalità sub-quadratica, ma comunque mantengono un punto di rottura di 0%.

Gli M-estimatori hanno il punto di *breakdown* pari a $1/(p+1)$, dove p è il numero di parametri nella regressione.

A3.2.4.2 Least Median of Squares

Il metodo della minima mediana dei quadrati (o *Least Median of Squares*) [Rousseeuw e Leroy, 1987] è attualmente una delle tecniche di stima robusta più usate in visione computazionale. Possiede un punto di *breakdown* del 50% e la sua efficienza relativa, intrinsecamente bassa, può essere incrementata se viene combinato con criteri basati sui minimi quadrati. Nel LMedS i parametri sono stimati risolvendo il problema di minimizzazione non lineare:

$$\min_i \{\text{med } r_i^2\}, \quad (3.26)$$

ovvero trova il più piccolo valore della mediana dei quadrati dei residui calcolati per l'intero insieme di dati. Rousseeuw congettura che non sia possibile ottenere una soluzione in forma chiusa per (3.26), la quale non è nemmeno riconducibile ad un problema di minimi quadrati pesati. La minimizzazione numerica diretta della funzione obiettivo diventa subito impraticabile con il crescere del numero di variabili. Un approccio possibile è il *random resampling*, illustrato dall'algoritmo 3.1, dove p è il un

numero minimo di osservazioni necessario per istanziare i parametri del modello.

Algoritmo 3.1 MINIMA MEDIANA DEI QUADRATI

Input: Insieme di osservazioni

Output: Modello che soddisfa la maggioranza delle osservazioni

- (i) prendi un campione casuale di p elementi dall'insieme delle osservazioni.
 - (ii) Per questo sottoinsieme, indicizzato con $J = i_1, \dots, i_p$, effettua la regressione lineare attraverso i p elementi e determina il corrispondente vettore dei coefficienti $\hat{\theta}_J$, chiamato stima di prova.
 - (iii) Calcola la dispersione della stima di prova
- $$\text{med}_i(y_i - \mathbf{x}_i^\top \hat{\theta}_J)^2$$
- (iv) Ripeti i passi 1-3 m volte e mantieni la stima di prova per la quale dispersione è minima.
-

Quanti campioni di dimensione p si devono considerare? Se n è la cardinalità dell'insieme di tutti i dati di partenza, vi sono $\binom{n}{p}$ sottoinsiemi diversi di p elementi. Se non vengono generati tutti, ma se ne considera solo un numero pari ad m , sorge la questione di quanto deve essere grande m affinché il procedimento calcoli la stima corretta con buona probabilità. Poiché il minimo della (3.26) è realizzato dai sottoinsiemi di dati privi di *outlier*, la possibilità di calcolare la stima corretta è legata alla presenza di almeno uno di questi campioni tra tutte le p -tuple. Data una percentuale ϵ di *outlier*, la seguente relazione lega m con la probabilità P di calcolare la stima corretta:

$$P = 1 - (1 - (1 - \epsilon)^p)^m, \quad (3.27)$$

dalla quale, fissando un valore di P prossimo ad 1, possiamo determinare il numero m , dati p ed ϵ :

$$m = \frac{\log(1 - P)}{\log(1 - (1 - \epsilon)^p)}. \quad (3.28)$$

Come viene notato in [Rousseeuw e Leroy, 1987], l'efficienza di LMedS è scarsa in presenza di rumore Gaussiano. Per ovviare a tale problema costruiamo una procedura ai minimi quadrati pesati. L'identificazione

degli *outlier* avviene attraverso la definizione di una funzione peso da applicare ai dati, che vale 0 per gli *outlier* e 1 altrimenti. La stima robusta della deviazione standard è data da

$$\hat{\sigma} = 1.4826[1 + 5/(n - p)] \sqrt{\operatorname{med}_i r_i^2} \quad (3.29)$$

e si definiscono i pesi nel modo seguente:

$$w_i = \begin{cases} 1 & \text{if } r_i^2 \leq (2.5\hat{\sigma})^2 \\ 0 & \text{altrimenti,} \end{cases} \quad (3.30)$$

che verranno usati per risolvere:

$$\min \sum_i w_i r_i^2. \quad (3.31)$$

La complessità computazionale del metodo è molto grande. Infatti vi sono $O(n^p)$ p -tuple e per ognuna di esse il calcolo della mediana comporta un tempo di $O(n)$, portando ad una complessità totale di $O(n^{p+1})$. Nel caso si impieghi una tecnica Monte Carlo per un campionamento dello spazio delle p -tuple, la complessità computazionale di LMedS scende significativamente fino a $O(mn)$.

A3.2.4.3 RANSAC

Un'idea simile a LMedS è implementata dall'algoritmo di *Random Sample Consensus* (RANSAC) [Fischler e Bolles, 1981, Meer *e al.*, 1991], che è particolarmente popolare in Visione Artificiale.

Dato un modello che richiede un numero minimo di p osservazioni per istanziare le proprie variabili indipendenti, l'algoritmo RANSAC si descrive formalmente come segue:

Algoritmo 3.2 RANSAC

Input: Insieme di osservazioni**Output:** Modello che soddisfa la maggioranza delle osservazioni

- (i) Prendi un campione casuale di p elementi dall'insieme delle osservazioni.
- (ii) Per questo sottoinsieme, indicizzato con $J = i_1, \dots, i_p$, determina la regressione attraverso i p elementi (risolvendo un sistema di p equazioni lineari in p incognite) ottenendo una stima di prova $\hat{\boldsymbol{\theta}}_J$
- (iii) Calcola il consenso della stima di prova

$$|\{y_i : (y_i - \mathbf{x}_i^\top \hat{\boldsymbol{\theta}}_J)^2 < \varepsilon\}|$$

ovvero la cardinalità delle osservazioni che concordano con la stima di prova. Queste prendono il nome di insieme di consenso di $\hat{\boldsymbol{\theta}}_J$.

- (iv) Ripeti i passi 1-3 finché il consenso è più grande di una certa soglia T .
 - (v) Effettua una regressione sull'insieme di consenso trovato per calcolare la stima finale.
-

La soglia T dipende dalla stima del numero di errori gravi (*outlier*) nell'insieme delle osservazioni. Se, dopo un certo numero predefinito di prove, l'algoritmo non termina, posso scegliere di effettuare la regressione sui punti del più grande insieme di consenso trovato fino a quel momento, oppure di fallire. Il valore della tolleranza viene stabilito sperimentalmente.

Come osservato da [Stewart, 1999], RANSAC si può vedere come un particolare M-stimatore. Infatti, la funzione obiettivo che RANSAC *minimizza*, ovvero il numero di punti aventi residuo *maggior* di una soglia ε , si può realizzare con una funzione di penalità che vale zero per residui compresi in $[-\varepsilon, \varepsilon]$ e vale uno altrove, come si vede in figura A3.4.

RANSAC è basato sulla votazione: vince la stima dei parametri più votata dai dati. Si noti l'analogia con la trasformata di Hough.

A3.2.5 Diagnostica

Assumiamo che molte misure della stessa quantità siano “molto vicine” e ve ne sia una sola “lontana”. La procedura comune è quella di rigettare l'*outlier*, cioè di scartare il valore più distante e considerare solo i rimanenti. La decisione su cosa significa “lontano” può essere presa “sogget-

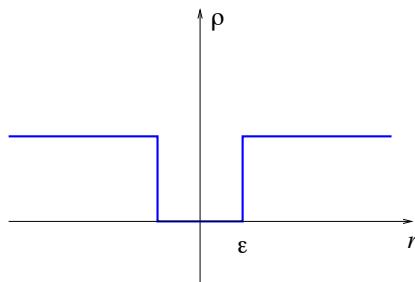


Fig. A3.4. Funzione di penalità per RANSAC.

tivamente”, cioè osservando i dati e prendendo una decisione arbitraria, oppure attraverso una formale regola “oggettiva” per la reiezione degli *outlier*, impiegando qualche test statistico.

Gli scopi della identificazione degli *outlier* sono essenzialmente due. Il primo è basato sull’osservazione che talvolta si possono verificare grandi errori e che anche un solo grosso errore può essere dannoso per la procedura statistica usata: si pensi, ad esempio, alla media, alla varianza o più in generale ai minimi quadrati. Il secondo scopo è quello di identificare valori interessanti per trattamenti particolari. I valori possono essere degli *outlier* che indicano la possibilità di uno sviluppo inaspettato o ricercato che può portare alla formulazione di un nuovo modello, oppure possono essere messi in evidenza dei grandi errori che potrebbero essere successivamente studiati e corretti.

Uno dei modi per eseguire *outlier diagnostic* al fine di individuare *outlier* univariati in ogni singola variabile, sono le cosiddette *regole di reiezione (rejection rules)*. Esse fanno uso della mediana e della deviazione mediana piuttosto che della media e della deviazione standard, e consentono di identificare tutte le osservazioni che hanno un valore superiore ad una certa soglia. Una volta eliminate queste osservazioni, sarà poi possibile eseguire le stime di locazione (media) e di scala (varianza) nei modi classici sui dati “ripuliti”. Tali regole presentano il vantaggio di essere semplici e veloci computazionalmente, inoltre possiedono punto di *breakdown* elevato. Un gruppo di regole di reiezione sono le *Huber-type skipped means*, di cui fa parte anche la X-84.

A3.2.5.1 Regola di reiezione X84

Questa regola [Hampel, 1974] dice che si devono scartare come *outlier* tutti i valori che si discostano dalla mediana (in valore assoluto) più di 5.2 volte la deviazione mediana assoluta (*MAD*). Quest’ultima è definita

da:

$$MAD(y_i) = \operatorname{med}_i \{ |y_i - \operatorname{med}_j y_j| \} \quad (3.32)$$

dove, y_i sono i dati. La regola X84 ha un punto di breakdown del 50%. Sotto le ipotesi di distribuzione normale, il valore di 5.2 deviazioni mediane corrisponde a 3.5 deviazioni standard.

Altri metodi di diagnostica sono basati sui residui provenienti dai minimi quadrati. Può accadere, però, che questo approccio conduca a risultati poco affidabili. Infatti, i minimi quadrati cercano per definizione di evitare di ottenere grandi residui, di conseguenza, un *outlier* può possedere un residuo basso anche se ha molta influenza sulla stima.

Un'altra classe di diagnostiche è basata sul principio di cancellare dai dati un *outlier* alla volta. Per esempio, se denotiamo con $\hat{\theta}(i)$ la stima di θ calcolata dai dati senza la i -esima osservazione, allora la differenza tra $\hat{\theta}$ e $\hat{\theta}(i)$ dà la misura di quanto la presenza dell' i -esimo dato influenzi i coefficienti della regressione. Queste sono chiamate *single-case diagnostics*, in quanto sono computate per ogni caso i . È possibile generalizzarle alle *multiple-case diagnostics* al fine di mettere in evidenza l'influsso simultaneo di più casi, anche se i calcoli possono diventare gravosi a causa del gran numero di sottoinsiemi che si dovrebbero considerare.

A3.3 Soluzione di equazioni non-lineari

A3.3.1 Metodo di Newton

Si tratta di un metodo† numerico iterativo per trovare lo zero di una funzione, ovvero per calcolare la soluzione di

$$f(x) = 0 \quad (3.33)$$

Sviluppiamo f con la serie di Taylor troncata al primo ordine, in un intorno della soluzione ($x + \Delta x$):

$$f(x + \Delta x) = f(x) + f'(x)\Delta x + o(\Delta x) \quad (3.34)$$

poichè $(x + \Delta x)$ è la soluzione, si ha che $f(x + \Delta x) = 0$, dunque $f(x) = -f'(x)\Delta x + o(\Delta x)$. Trascurando il resto, si ha

$$\Delta x = \frac{-f(x)}{f'(x)} \quad (3.35)$$

† detto anche di Newton-Raphson. Esiste anche un metodo di Newton per ottimizzazione di funzioni $\mathbb{R}^n \rightarrow \mathbb{R}$.

A causa di questa approssimazione, $(x + \Delta x)$ non è la soluzione[‡], ma, sotto opportune condizioni su f , una approssimazione della soluzione migliore di x . Iterando il procedimento con $x \leftarrow x + \Delta x$ si ottiene una successione di valori convergente alla soluzione cercata.

Lo stesso ragionamento si può fare nel caso di un sistema di n equazioni non lineari in n incognite: $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, con $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, allora, analogamente

$$\mathbf{f}(\mathbf{x}) = -J(\mathbf{x})\Delta\mathbf{x} \quad (3.36)$$

Dove $J(\mathbf{x})$ è la matrice jacobiana di f calcolata in \mathbf{x} . Il passo $\Delta\mathbf{x}$ si ottiene come soluzione di un sistema lineare $n \times n$. Possiamo scrivere $\Delta\mathbf{x} = -J(\mathbf{x})^{-1}\mathbf{f}(\mathbf{x})$ ma ricordiamo che per risolvere un sistema lineare il metodo migliore *non* è invertire la matrice [Strang, 1988].

A3.3.2 Metodo di Gauss-Newton

Questo invece è un metodo numerico per la soluzione ai minimi quadrati di sistemi di equazioni non lineari sovradeterminati. In forma compatta il sistema si può scrivere $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ con $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, dove $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ sono le m equazioni che compongono il sistema. Posso pensare di applicare il metodo di Newton visto prima, ottenendo

$$\mathbf{f}(\mathbf{x}) = -J(\mathbf{x})\Delta\mathbf{x} \quad (3.37)$$

ma in questo caso la matrice jacobiana di f è rettangolare, quindi il sistema lineare è sovradeterminato. Lo risolvo allora con la pseudoinversa, quindi $\Delta\mathbf{x} = -(J^\top J)^{-1}J^\top f(\mathbf{x})$, o meglio, risolvo il sistema (evitando l'inversione)

$$J^\top f(\mathbf{x}) = -J^\top J\Delta\mathbf{x} \quad (3.38)$$

Si dimostra [Gill *e al.*, 1981] che iterando questo procedimento, si giunge alla soluzione ai minimi quadrati del sistema di equazioni originale, ovvero si calcola il minimo di:

$$F(x) = \frac{1}{2} \sum f_i(\mathbf{x})^2 \quad (3.39)$$

Bibliografia

Bar-Shalom Y.; Fortmann T. E. (1988). *Tracking and data Association*. Academic Press.

[‡] Lo sarebbe esattamente se f fosse una funzione lineare, nel qual caso il resto sarebbe zero.

- Fischler M. A.; Bolles R. C. (1981). Random Sample Consensus: a paradigm model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, **24**(6), 381–395.
- Gill P.; Murray W.; Wright M. (1981). *Practical Optimization*. Academic Press.
- Hampel F. (1974). The influence curve and its role in robust estimation. *Journal of American Statistics Association*, **69**, 383–393.
- Meer P.; Mintz D.; Kim D. Y.; Rosenfeld A. (1991). Robust regression methods in computer vision: a review. *International Journal of Computer Vision*, **6**, 59–70.
- Rousseeuw P. J.; Leroy A. M. (1987). *Robust regression & outlier detection*. John Wiley & sons.
- Stewart C. V. (1999). Robust parameter estimation in computer vision. *SIAM Review*, **41**(3), 513–537.
- Strang G. (1988). *Linear Algebra and its applications*. Harcourt Brace & Co.

Indice Analitico

- absolute**, 140
- art**, 36
- autocal**, 161
- bundleadj**, 113
- calibz**, 176
- camera**, 40
- epipole**, 165
- erec**, 112
- eul**, 220
- exterior**, 144
- fm**, 159
- h2m**, 171
- horn**, 139
- icp**, 142
- ieul**, 220
- ikan**, 220
- imhs**, 118
- imstereo**, 92
- imwarp**, 184
- intersect_le**, 62
- intersect**, 62
- kan**, 220
- lowe**, 147
- precond2**, 159
- prec**, 155
- rectify**, 67
- resect**, 34
- sr**, 109
- 3D photography, 4, 5
- algoritmo degli otto punti, 106, 110
- ambiguità profondità-velocità, 106, 107, 123
- attitudine, 136, 143
- bundle adjustment, 112
- calibrazione, 33, 174
- campo di moto, 105
- centro ottico, 24, 32
- cerchio di confusione, 20
- chiaroscuro, 42
- coarse-to-fine, 81
- coded-light, 88
- collineazione, 68, 155, 167
- compatibile, matrice , 172
- Computer Vision, 1
- coordinate normalizzate, 29, 106
- data association, 118
- depth from defocus, 54
- depth from focus, 53
- distanza focale, 19
- dimensioni efficaci del pixel, 23
- Direct Linear Transform, 34
- Disparity Space Image, 84
- edge, 115
- effetto dell'apertura, 128
- equazione di Longuet-Higgins, 64
- errore geometrico, 35
- feathering, 186
- filtro di Kalman, 118
- fine-to-fine, 81
- flusso ottico, 127
- foto-coerente, 97
- fotocamera affine, 162
- fotocamera stenopeica, 16
- fotocamera telecentrica, 20
- frange di interferenza di Moiré, 12
- funzione di penalità, 225
- fuoco, 19
- fuoco di espansione (contrazione), 126
- fusione geometrica, 8
- geometria epipolare, 106
- gradiente di tessitura, 51
- graph cuts, 84
- illuminazione strutturata, 7, 87
- image warping, 182

- image-based modeling, 4, 5
immagine 2.5D, 7
irradianza, 22
Iterative Closest Point, 140
- Lambert, 22
Least Median of Squares, 226
lente sottile, 18
Longuet-Higgins, 106
luce codificata, 90
lunghezza focale, 24
- M-estimatori, 225
match space, 85, 86
matrice della fotocamera, 27
matrice di proiezione prospettica, 27
matrice essenziale, 105, 107
matrice fondamentale, 65
metodi globali, 74
metodi locali, 74
mosaicatura 3D, 10
mosaicatura di immagini, 179
motion field, 122, 123
motion parallax, 125
MPP, 27
- Normalized Cross Correlation, 77
- object tracking, 149
omografia, 167, 181, 186
orientazione assoluta, 139
orientazione esterna, 143
orientazione relativa, 105, 137
outlier diagnostics, 224
outliers, 224
- parametri estrinseci, 30
parametri intrinseci, 29
piano focale, 24
pixel, 22
point spread function, 55
profondità di campo, 20
proiezione ortografica, 18
proiezione prospettica, 17, 26
punti coniugati, 19, 58
punto di breakdown, 225
punto di fuga, 40
punto principale, 24
- radianza, 21
range, 7, 8
RANSAC, 228
registrazione, 8, 132
regole di reiezione, 230
regressione lineare, 221
regressione robusta, 224
rettificazione epipolare, 65
rettificazione ortogonale, 186
- ricostruzione, 152
ricostruzione euclidea, 153
riflettanza, 21
- sagoma, 94
scanner, 14
scansione della immagine destinazione, 183
scansione della immagine sorgente, 183
scorciatura, 51
scorcio, 17
sfera gaussiana, 48
shading, 5
shape acquisition, 4
slant, 52
space carving, 100
spazio delle corrispondenze, 86
standardizzazione, 159
stenopeico, 24
stenoskopio, 16
stereo fotometrico, 50
structure from motion, 105
Sum of Absolute Differences, 77
- tessitura, 50
texel, 50
tilt, 52
traccia, 117
tracciamento, 116
trasformata *census*, 78
- vincolo di integrabilità, 44
vincolo di rigidità, 160
Visione computazionale, 1
visual hull, 95
voxel coloring, 99
- weak perspective, 18
- zippering, 11