

UNIVERSITÀ DEGLI STUDI DI MILANO
FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
CORSO DI LAUREA IN INFORMATICA



REALIZZAZIONE DI UN SISTEMA
ROBUSTO DI CALIBRAZIONE
DI FOTOCAMERE DIGITALI

Relatore: Prof.ssa Paola CAMPADELLI
Correlatori: Prof. Alberto Nunzio BORGHESE
Dott.ssa Raffaella LANZAROTTI

Tesi di Laurea di:
Fabio Chigiotti
Matr. Nr. 562161

ANNO ACCADEMICO 2002-2003

A mio padre

Indice

Indice	iii
Ringraziamenti	x
Introduzione	1
1 La fotocamera digitale come sistema di misura	4
1.1 Introduzione	4
1.2 Processo di formazione di un'immagine	4
1.2.1 Camera con foro stenopeico	5
1.2.2 Lenti	7
1.2.3 Obiettivo	7
1.2.4 Profondità di campo e diaframma	9
1.2.5 Posizione del piano focale	9
1.2.6 Piano comparatore	12
1.2.7 La tecnologia delle fotocamere	12
1.2.8 La fotocamera come sistema di misura	13
1.3 Le distorsioni	13
1.3.1 Modello matematico di distorsione	14
1.4 Descrizione della fotocamera rispetto al mondo	17
1.4.1 I sistemi di riferimento	17
1.4.2 Il modello geometrico della proiezione	18
1.4.3 Rotazione	20
1.4.4 Modello di proiezione completo	21
1.5 La condizione di collinearità	22
1.5.1 Modello completo	26

2	La calibrazione	28
2.1	Introduzione	28
2.2	Calibrazione e fotogrammetria: cenni storici	29
2.3	Problemi attuali relativi alla calibrazione	32
2.3.1	Il modello Bundle Adjustment	32
2.3.2	Il modello Bundle Adjustment con auto calibrazione	33
2.3.3	Altre tecniche di calibrazione	34
3	Il modello bundle adjustment	37
3.1	Introduzione al modello Bundle Adjustment	37
3.2	Linearizzazione delle equazioni di collinearit�	38
3.3	Linearizzazione con stima iterativa	42
3.3.1	Metodo dei minimi quadrati	43
3.4	Controllo sulla soluzione	45
3.4.1	Analisi di stabilit� del problema numerico	45
3.4.2	Numero di condizionamento	46
3.4.3	Analisi di varianza	48
3.4.4	Analisi di correlazione	51
4	Inizializzazione tramite Direct Linear Transformation	53
4.1	Introduzione alla DLT	53
4.1.1	Il concetto teorico	53
4.1.2	Applicazione della DLT	54
4.2	Calcolo dei parametri di orientamento interni ed esterni	57
4.2.1	Gli undici coefficienti della DLT	58
4.2.2	Calcolo delle coordinate del centro ottico: (X_L, Y_L, Z_L)	60
4.2.3	Calcolo degli elementi di orientamento interno: (x_0, y_0, f)	61
4.2.4	Calcolo degli angoli di rotazione: (ω, ϕ, κ)	62
5	Ricostruzione tridimensionale	65
5.1	Introduzione	65
5.2	Correzione delle distorsioni	65
5.3	La ricostruzione tridimensionale	67
5.3.1	Algoritmo di ray-intersection	68
6	Risultati	79
6.1	Risultati sperimentali	79
6.1.1	Fotocamera FinePix	79
6.2	Calibrazione di fotocamere con dati simulati	81

6.3	Calibrazione di fotocamere con rumore	86
6.4	Calibrazione di fotocamere con dati reali	86
7	Note sul SW	92
7.1	Introduzione	92
7.2	Interfaccia utente	92
7.3	Comandi e funzioni	93
7.4	Creazione della matrice dei punti di coordinate nota	94
7.5	Acquisizione di una coppia di immagini	96
7.6	Creazione della matrice bidimensionale dei punti	96
7.6.1	Coordinate 2D simulate	96
7.6.2	Coordinate 2D prese da foto reali	96
7.7	Inizializzazione dei parametri tramite DLT	97
7.8	Programma bundle adjustment	98
7.8.1	Condizionamento della matrice	101
7.8.2	Rango della matrice	102
7.8.3	Incrementi nelle variabili	102
7.8.4	La media e la varianza sul residuo	103
7.8.5	L'analisi di varianza	103
7.8.6	L'analisi di covarianza	104
7.8.7	L'aggiornamento della soluzione	105
7.9	Minimizzazione delle distorsioni	105
7.10	Ricostruzione tridimensionale	105
7.11	Guida pratica all'utilizzo del software	107
7.12	Sviluppi futuri	108
	Appendice A	109
.1	Introduzione	109
.2	Calcolo delle derivate parziali	109
.2.1	Derivate parziali sui parametri intrinseci ed estrinseci	110
.2.2	Derivate parziali sui parametri ottici	111
	Appendice B	113
.1	Introduzione	113
.2	Problematiche numeriche per la risoluzione tramite bundle adjustment	113
.3	Elementi di analisi delle matrici	114
.3.1	Inversa di una matrice	114
.3.2	Traccia e determinante	115
.3.3	Rango di una matrice	116

.3.4	Norme matriciali	116
.4	Buona posizione e numero di condizionamento di un problema	117
.4.1	Stabilità dei metodi numerici	119
.5	Origine degli errori	120
.5.1	Accuratezza	121
.5.2	Affidabilità	122
.5.3	Efficienza	122
.6	Rappresentazione dei numeri sul calcolatore	122
.6.1	Aritmetica IEC/IEEE	122
.6.2	Arrotondamento di un numero reale nella sua rappresentazione di macchina	123
.6.3	Overflow ed Underflow	124
.6.4	Operazioni di macchina effettuate in virgola mobile	125
.7	Analisi di stabilità del problema numerico	127
.8	Sistemi indeterminati	130
.8.1	Fattorizzazione Singular Value Decomposition (SVD)	131
.8.2	Fattorizzazione SVD e pseudoinversa in forma normale	133
.8.3	Fattorizzazione SVD in MATLAB	133
.8.4	Calcolo della pseudoinversa in MATLAB	135

Bibliografia

137

Elenco delle tabelle

4.1	I parametri stimati tramite DLT	58
4.2	Numero minimo di punti di controllo per DLT	60
5.1	Tipologie di algoritmi di ricostruzione 3D	67
1	Codifiche particolari IEC559	123
2	Risultati per alcune operazioni eccezionali	127

Elenco delle figure

1.1	Modello pinhole camera	5
1.2	Camera oscura di Alphonse Giroux, anno 1847	6
1.3	Il sistema ottico	8
1.4	Esempio di immagine a fuoco e non a fuoco	10
1.5	Il piano comparatore	13
1.6	Una fotocamera a pellicola	14
1.7	Una fotocamera digitale	15
1.8	Effetti di distorsione radiale	16
1.9	Sistema camera, sistema immagine, sistema mondo	18
1.10	Similitudine fra triangoli	19
1.11	Vari cambiamenti dei sistemi di riferimento	21
1.12	La condizione di collinearit�	22
1.13	Sistema di coordinate immagine ruotato in modo da essere parallelo al sistema di coordinate dello spazio oggetto	23
3.1	La funzione MATLAB <i>cond</i>	49
3.2	La funzione MATLAB <i>rcond</i>	49
5.1	Coseni direttori di una retta orientata	69
5.2	Condizione di intersezione	71
5.3	Distanza di due rette sghembe	74
6.1	Fotocamera Fuji FinePix 4900Zoom	80

6.2	Matrice dei punti tridimensionale <i>grid3d</i>	81
6.3	Matrice dei punti bidimensionale <i>grid2d</i> per la prima foto	82
6.4	Valori dati dalla DLT per la prima foto	83
6.5	Calcolo dei valori sulla prima foto	84
6.6	Minimizzazione tramite gradiente	85
6.7	Matrice dei punti tridimensionale iniziale	85
6.8	Matrice dei punti tridimensionale ricostruita	86
6.9	Matrice dei punti tridimensionale ricostruita (con rumore)	87
6.10	Creazione della matrice tridimensionale dei punti <i>grid3d</i>	88
6.11	Valori della DLT sulla seconda foto	89
6.12	Calcolo dei valori sulla seconda foto	90
6.13	Matrice dei punti tridimensionale iniziale	91
6.14	Matrice dei punti tridimensionale ricostruita	91
7.1	Interfaccia MATLAB e directory di lavoro	93
7.2	Programma Reticolo3d.m	94
7.3	Pattern planare	95
7.4	Programma Proietta.m	97
7.5	Programma Reticolo2d.m	98
7.6	Programma Dlt.m	99
7.7	Programma Matlab.m	100
7.8	Numero di parametri di distorsione attivi	101
7.9	Prima approssimazione delle variabili incognite	102
7.10	Analisi di varianza	103
7.11	Analisi di covarianza ed aggiornamento della soluzione	104
7.12	Minimizzazione delle distorsioni	106
13	La variabile di sistema MATLAB <i>eps</i>	125
14	La funzione di decomposizione MATLAB <i>rref</i>	135
15	La funzione di decomposizione MATLAB <i>svd</i>	136
16	Il calcolo della pseudoinversa (<i>pinv</i>) in MATLAB	136

Ringraziamenti

Arrivati alla fine di un viaggio ci si trova a riflettere come sono passati questi anni. Sono stati anni di duro lavoro, con molte rinunce e pochi momenti di svago. Comunque, sono stati anche anni molto belli sotto tanti punti di vista.

Ho iniziato questo viaggio con la consapevolezza di voler arrivare in fondo, di voler arrivare ad un traguardo importante. Ho incontrato molte persone lungo questo cammino. Alcune le ho perse di vista, altre sono risultate determinanti in alcuni momenti di questa carriera universitaria. Le piú importanti sono invece rimaste dall'inizio alla fine a condividere con me le cose belle e le cose brutte di tutti questi anni. Non credo che qualche riga serva e far capire tutto quello che sento o che vorrei dire. Comunque ci provo lo stesso.

Volevo innanzitutto ringraziare il mio correlatore, il prof. Alberto Borghese, per tutto l'aiuto e la comprensione che mi ha dimostrato in questo anno di tesi. Lo ringrazio soprattutto per la pazienza che ha dimostrato nei miei confronti e come non si é mai tirato indietro a correggere ed a migliorare i programmi che ho sviluppato. Se tutto funziona come dovrebbe é soprattutto merito suo. Vorrei ringraziare anche il mio relatore, la Prof.ssa Paola Campadelli, che mi ha sostenuto ed aiutato nei momenti difficili di questa tesi.

Altre persone sono state di grande aiuto nel corso di questo progetto. La prima é Fabio Remondino dell'*Institute of Geodesy and Photogrammetry* di Zurigo. I suoi consigli su come implementare alcune caratteristiche del bundle adjustment sono stati preziosi. La seconda persona che voglio ringraziare é Iuri Frosio. Se i programmi funzionano devo dire grazie anche a lui, che ha dedicato ben piú di qualche ora alla risoluzione dei problemi che di tanto in tanto sorgevano. Un grazie anche alla prof.ssa Anna Maria Zanaboni ed alla dott.ssa Sabrina Gaito, che mi hanno aiutato a capire meglio i problemi statistici che sono stati parte integrante di questo progetto.

Per ringraziare gli amici mi servirebbero pagine e pagine, ma lo spazio é tiranno e speriamo che poche righe bastino.

Ringrazio gli amici di una vita, cioè Matteo ed Elena. Alcune volte é bello evadere dalla vita di tutti i giorni o farsi una risata sui vecchi tempi per scacciare via i brutti pensieri.

Per gli amici che invece ho incontrato in questi ultimi anni l'elenco é lungo. Cominciamo con Alex, fedele compagno in questi anni universitari. Assieme abbiamo fatto il bello ed il brutto tempo, ma alcuni ricordi non potrebbero esistere senza di lui. Ringrazio di cuore anche Daniele, sempre presente soprattutto nei momenti difficili ed amico vero e sincero. Ringrazio anche tutti gli altri amici che, in un modo o nell'altro, mi sono sempre stati vicini. In particolare Simone, Raffaele, Cavallo, Paola, Alessandra, Roberto, Francesco, Barbara e Lisa. Alla fine vi verró a ringraziare ad uno ad uno, ma ricordate che vi ho sempre nel cuore.

Gli ultimi due ringraziamenti sono per le persone piú importanti di tutta la mia vita.

Prima di tutto ringrazio la mia famiglia. Soprattutto in questo ultimo anno non sono stato molto presente in famiglia come invece avrei dovuto, forse perché alcune volte mi sono perso per strada. Spero che dopo questa tesi ritorni il vero Fabio, cioè una persona allegra, sorridente e piena di vita. Una buona parte di questa tesi é dedicata a loro, che mi hanno sempre sostenuto e che non mi hanno mai abbandonato.

L'ultimo ringraziamento é per la persona con cui da sette anni a questa parte condivido gioie e dolori, sofferenze e momenti di felicità. Non so se sarei riuscito a concludere questa tesi senza il costante supporto amorevole di Lorena. La strada da percorrere é stata lunga e forse mi sarei arenato da qualche parte senza di lei, a cui va una parte del mio cuore.

Milano, Italy
3 Febbraio 2004

Fabio Chigiotti

Introduzione

Questa tesi é il risultato di un progetto di ricerca svolto presso il Dipartimento di Scienze dell'Informazione dell'Universitá degli studi di Milano ed ha come obiettivo la realizzazione di un sistema robusto di calibrazione di fotocamere digitali.

La calibrazione é essenzialmente un problema di stima dei parametri che definiscono la fotocamera, a partire dalla misura sul piano immagine di un certo numero di punti di controllo di cui é nota la posizione nello spazio tridimensionale.

I parametri da stimare sono:

- *parametri estrinseci*, che definiscono la posizione e l'orientamento della fotocamera;
- *parametri intrinseci*, che definiscono la distanza focale ed il punto principale;
- *parametri ottici*, che definiscono il campo di distorsione.

Il numero dei parametri ottici non é fissato a priori ma dipende dal tipo di fotocamera e di obiettivo usato. Occorre quindi sviluppare un sistema che sia in grado di identificare il modello migliore per rappresentare le distorsioni.

Per ottenere una stima robusta é stata quindi analizzata a fondo la soluzione denominata *Bundle Adjustment*. Questo modello consente di stimare i parametri minimizzando iterativamente l'errore di retroproiezione dei punti di controllo.

Per valutare l'adeguatezza del modello, per prima cosa é stato valutato il condizionamento della matrice A . Nel caso di sovrapparametrizzazione, il numero di condizionamento della matrice diventa elevato e la soluzione mal condizionata. É stata inoltre valutata la bontá della stima dei parametri mediante l'analisi della varianza della stima e la correlazione tra i parametri mediante l'analisi della correlazione. Questa analisi consente anche di eliminare parametri che risultino stimati in modo non sufficientemente accurato.

La calibrazione *bundle adjustment* é stata poi applicata alla calibrazione di un sistema stereo-scopico costituito da una coppia di fotocamere. Questo sistema puó essere utilizzato per la ricostruzione della posizione di punti nello spazio tridimensionale. L'accuratezza globale della calibrazione per il sistema stereoscopico viene determinata attraverso l'errore tra la posizione tridimensionale ricostruita e la posizione reale di un insieme di punti di controllo.

Il risultato finale é quindi la calibrazione di fotocamere digitali e la successiva ricostruzione dei punti tridimensionali.

Questa tesi é divisa in 7 capitoli e due appendici:

Nel primo capitolo si descrive la fotocamera digitale come sistema di calibrazione e si introducono le equazioni di collinearitá.

Nel secondo capitolo si affrontano i problemi attuali relativi della calibrazione.

Nel terzo capitolo si descrive in modo dettagliato il modello *bundle adjustment*.

Nel quarto capitolo si descrive la *direct linear transformation (DLT)*, con la quale si é inizializzato il modello *bundle adjustment*.

Nel quinto capitolo si presenta l'algoritmo che permette la ricostruzione tridimensionale.

Nel sesto capitolo vengono presentati alcuni risultati ottenuti su alcune prove di laboratorio.

Nel settimo capitolo viene illustrata l'interfaccia utilizzata per la calibrazione e la

ricostruzione.

Nell'Appendice A si elencano le derivate parziali per il modello del bundle adjustment.

Nell'Appendice B si affrontano alcune problematiche numeriche che sono sorte durante la realizzazione dei vari programmi sviluppati.

Capitolo 1

La fotocamera digitale come sistema di misura

1.1 Introduzione

In questi ultimi anni l'elaborazione delle immagini digitali ha acquistato sempre maggiore importanza, soprattutto per la varietà di applicazioni possibili.

Le immagini digitali, come sappiamo, sono rappresentate come *matrici di numeri*.

Prima di tutto occorrerà quindi definire il *processo di formazione di un'immagine*, per poi esaminare le caratteristiche fisiche fondamentali che stanno alla base del funzionamento delle *fotocamere digitali*. In particolare parleremo di:

- lenti;
- obiettivo;
- profondità di campo e diaframma;
- fuoco immagine.

1.2 Processo di formazione di un'immagine

Come per molti sistemi visivi, il processo di *formazione dell'immagine* inizia con

dei raggi di luce che entrano nella fotocamera attraverso un'apertura (detta *pupilla*). Successivamente i raggi luminosi colpiscono il *piano immagine* ed è il dispositivo fotosensibile della fotocamera che registra l'intensità della luce.

Da notare che molti di questi raggi sono il risultato della riflessione dei raggi emessi dalle sorgenti luminose colpendo superfici di oggetti; se una superficie non riflettesse la luce neppure in minima parte non sarebbe visibile (buco nero).

1.2.1 Camera con foro stenopeico

Una *fotocamera* o *telecamera reale* è costituita da un apparato che cattura la luce presente all'interno di una scena tridimensionale che, attraverso processi ottici, viene proiettata su un piano bidimensionale, detto *piano immagine*.

Esistono diversi modelli di fotocamera, ma il più semplice è il cosiddetto *pinhole camera model*.

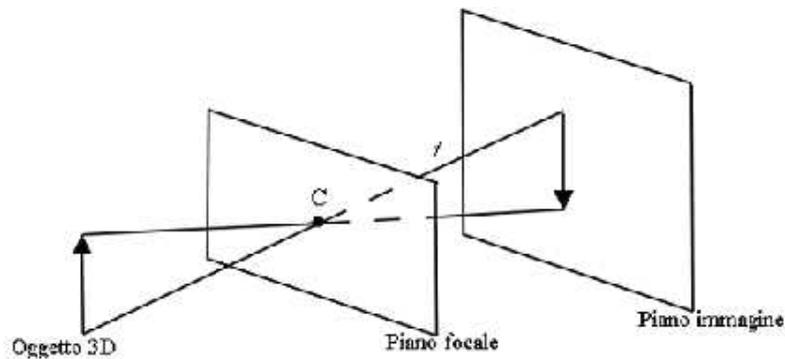


Figura 1.1: Modello pinhole camera

Come possiamo vedere nella figura (1.1) il modello *pinhole* è costituito da due piani denominati *piano focale* e *piano immagine*.

Sul *piano focale* viene praticato un foro di piccole dimensioni denominato *foro*

stenopeico C che permette l'ingresso della luce emessa o riflessa della scena tridimensionale che si desidera riprendere.

Al fine di evitare che l'immagine proiettata sul piano immagine risulti sfocata, il foro deve essere di dimensioni tali in modo da far passare un solo raggio di luce per ogni direzione. Ne consegue che l'energia incidente su ciascun punto del piano immagine é estremamente limitata e quindi il tempo richiesto per formare l'immagine é elevatissimo.

Si definisce *centro ottico* il punto C mentre si definisce *asse ottico* la retta passante per C e perpendicolare al piano immagine. Il punto in cui l'asse ottico interseca il piano immagine viene detto *punto principale*.

Il modello della pinhole camera é ispirato alla *camera oscura* (figura 1.2).



Figura 1.2: Camera oscura di Alphonse Giroux, anno 1847

La luce che entrava attraverso il foro stenopeico formava su uno schermo interno un'immagine capovolta della scena esterna illuminata dalla luce solare.

1.2.2 Lenti

La *lente* é senza dubbio il dispositivo ottico piú usato, anche considerando che molte persone vedono il mondo attraverso un paio di lenti.

Giá nell'antichitá venivano usate lenti artificiali note come *lenti ustorie* che servivano per accendere il fuoco molto tempo prima dell'avvento dei fiammiferi; ne accenna Aristofane in *Le Nuvole*. Dal nostro punto di vista, una lente é un dispositivo rifrangente (o una discontinuitá nei mezzi trasmittenti) che cambia la configurazione di una distribuzione di energia incidente. Questo discorso vale indipendentemente dal tipo di onde dalle quali si propaga l'energia: ultravioletto, onde luminose, infrarosso, microonde, radioonde o persino onde sonore.

La superficie di una lente é determinata dal modo in cui essa deve modificare la forma del fronte d'onda. Ad esempio le sorgenti puntiformi, quali le lampadine di una torcia elettrica o di proiettore, producono fasci di onde sferiche divergenti, che devono essere convertiti in un fascio di onde piane, per impedire al fascio di sparpagliarsi e di indebolirsi mentre si propaga.

Nel nostro caso é necessario raccogliere raggi paralleli incidenti e farli convergere in un punto, concentrando cosí l'energia, come ad esempio nel caso delle lenti ustorie (o nelle lenti di un cannocchiale).

In pratica, quindi, all'interno di una fotocamera reale vengono usate delle *lenti ottiche* in grado di far convergere in un solo punto fasci di raggi paralleli di luce.

La lente consente quindi di risolvere il problema della camera con foro stenopeico.

1.2.3 Obiettivo

L'*obiettivo* é un sistema ottico centrato formato da una o piú lenti in successione. Le proprietá di una lente o di un obiettivo vengono considerati alla stessa stregua e possono riferirsi ad uno schema ottico ideale, rappresentato in figura (1.3). Tratteremo quindi le proprietá di una lente.

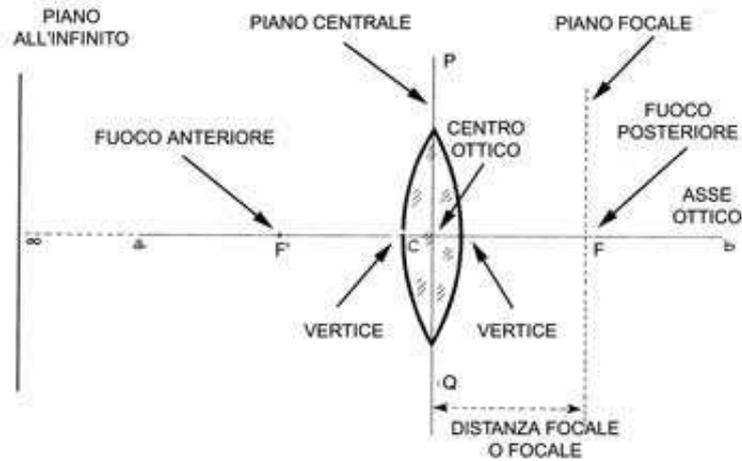


Figura 1.3: Il sistema ottico

L'*asse ottico* è la retta che congiunge i centri di curvatura delle superfici limite della lente. Negli obiettivi di uso più comune, tutti i centri di curvatura delle lenti che li compongono sono allineati sull'asse ottico. La caratteristica dell'asse ottico consiste nel fatto che il raggio che attraversa le lenti secondo questo asse non viene deviato ed interseca il piano immagine formando un angolo retto in un punto chiamato *punto focale* f o *punto principale*.

Per ogni altra direzione, esiste una retta detta *asse principale*, la cui direzione non viene alterata dal gruppo di lenti.

Per ogni lente viene definito un *centro ottico* del sistema. Il centro ottico ha due importanti proprietà:

- ogni raggio luminoso che passa attraverso esso (*raggio principale*) non viene deviato;
- i raggi diversi da quelli principali vengono rifratti e, per ogni direzione, fatti convergere in un unico punto.

Come risultato, per ogni *punto oggetto* (dello spazio tridimensionale) é individuato un unico *punto immagine*.

1.2.4 Profonditá di campo e diaframma

Il discorso appena fatto lo si puó estendere ai piani dicendo che ad ogni *piano oggetto* corrisponde un solo *piano immagine* e viceversa. Secondo questo discorso teorico si dovrebbe dire che con la fotocamera si puó mettere a fuoco un solo piano alla volta perché ad un solo piano immagine (la pellicola) corrisponde un solo piano oggetto. Se questo fosse vero si potrebbero fotografare solo disegni, quadri, francobolli o comunque solo oggetti perfettamente piani.

Sul piano pratico si deve ricordare che l'obiettivo fornisce sempre un'immagine meno precisa dell'oggetto fotografato e che la precisione di tale immagine dipende anche dalle dimensioni della grana della pellicola usata. Per questi motivi si ammette sempre un certo grado d'imprecisione (o meglio di confusione) nell'immagine ottenuta e questo permette di considerare a fuoco anche immagini un poco sfuocate perché il rispettivo piano oggetto é o troppo vicino o troppo lontano.

La distanza fra i due piani oggetto (quello troppo vicino e quello troppo lontano) é definita col nome di *profonditá di campo* e viene indicata fornendo la corretta distanza di messa a fuoco dei due piani oggetto.

Il *diaframma* é, invece, un dispositivo montato davanti all'obiettivo che limita la superficie della lente esposta alla luce.

1.2.5 Posizione del piano focale

Tutti i raggi che provengono da un singolo punto P , dovrebbero convergere dentro un singolo punto del piano immagine, p , l'*immagine* di P . Se questo avviene diciamo che l'immagine di P é a *fuoco*, altrimenti diciamo che l'immagine *non é a fuoco* (figura 1.4). Questo avviene quando il piano immagine si trova ad una distanza opportuna dal

centro di proiezione. Quando l'oggetto è sufficientemente lontano, i raggi luminosi provenienti dai punti dell'oggetto si possono considerare paralleli. In questo caso (detto di *messa a fuoco all'infinito*), il piano di messa a fuoco coincide con il piano focale.

La *distanza focale* è la distanza fra il centro ottico ed il piano focale. Si considerano, perciò, una distanza focale anteriore ed una distanza focale posteriore in riferimento a ciascuno dei due fuochi. Nel caso di lenti simmetriche, le due distanze focali sono uguali.

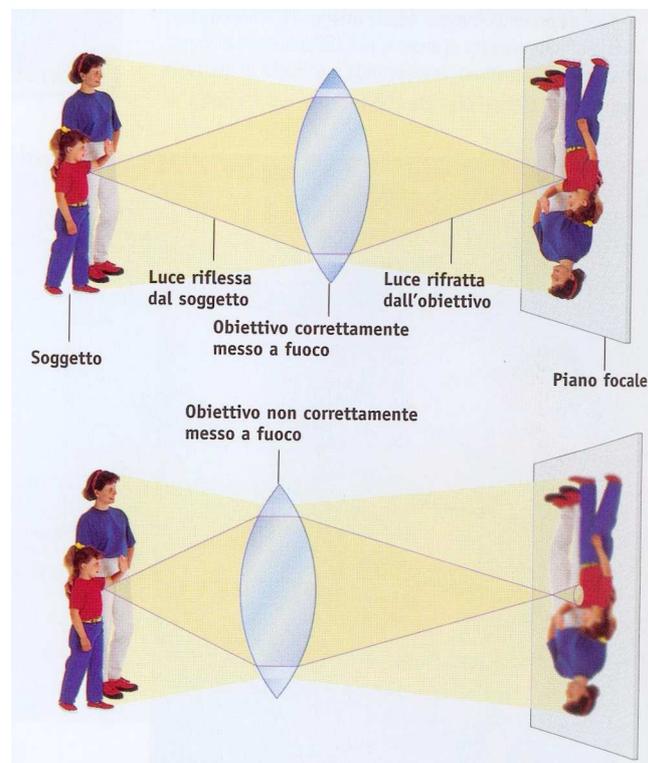


Figura 1.4: Esempio di immagine a fuoco e non a fuoco

Se invece i raggi luminosi non si intersecano, l'immagine di un punto diventa una circonferenza detta *cerchio di confusione*.

Piú il piano di messa a fuoco si allontana dal piano immagine, piú il raggio del cerchio di confusione aumenta fino a far diventare l'immagine gravemente sfuocata.

Diventa quindi fondamentale la regolazione della distanza del *piano di messa a fuoco*. Questa distanza é comunemente chiamata *distanza focale*.

Ad ogni grado di apertura del diaframma corrispondono dei valori numerici direttamente proporzionali alla focale dell'obiettivo ed inversamente proporzionali al diametro del foro stesso.

Autofocus

Esistono principalmente due tipi di autofocus.

Le fotocamere digitali compatte usano un *fascio di raggi infrarossi* (IR) che esaminano la scena quando si preme il pulsante di scatto. I raggi infrarossi riflessi piú vicini e piú forti sono rilevati da un sensore, che calcola la distanza del soggetto ed imposta la fotocamera una frazione di secondo prima che l'immagine sia registrata.

Il secondo sistema é l'*autofocus passivo*. La proprietá basilare di questo sistema é che il sensore autofocus esamina la scena e definisce in automatico se il fuoco deve cadere davanti o dietro il piano focale. Quindi lo stesso analizza la situazione e suggerisce all'obiettivo come muoversi per ottenere la messa a fuoco migliore.

I dispositivi autofocus sono sofisticati, ma non infallibili, soprattutto in particolari situazioni:

- il sensore autofocus si imposta principalmente sul centro dell'immagine, per cui un soggetto decentrato puó non essere messo correttamente a fuoco;
- se si scattano fotografie attraverso un vetro, i suoi riflessi possono confondere il sensore ad infrarossi;
- oggetti molto brillanti nella zona di messa a fuoco, come riflessi metallici, possono sovraccaricare il sensore e compromettere il risultato;

- fotografare al di là di oggetti vicini all'obiettivo, per esempio oltre un cespuglio o tra i paletti di uno steccato, può confondere il sistema autofocus;
- è più facile mantenere a fuoco soggetti vicini in movimento impostando la distanza manualmente e, di conseguenza, regolando la propria posizione avanti o indietro;
- con soggetti che si muovono molto velocemente, è meglio impostare la messa a fuoco ad una distanza predefinita ed aspettare che il soggetto raggiunga il punto giusto per scattare.

Per evitare di avere fotografie non a fuoco sarà dunque buona norma togliere l'autofocus ed impostare il tutto in modo manuale.

1.2.6 Piano comparatore

L'immagine della scena ripresa che si forma sul piano immagine risulta invertita. Per ovviare a questo inconveniente è necessario introdurre il concetto di *piano comparatore* (figura 1.5) che rappresenta un piano posto tra l'osservatore e l'oggetto posto ad una distanza dal centro ottico pari alla distanza del piano immagine.

1.2.7 La tecnologia delle fotocamere

Le fotocamere *digitali* ed a *pellicola* funzionano con il medesimo principio: entrambe registrano la scena sfruttando l'energia luminosa per attuare un mutamento nel materiale fotosensibile.

Tale mutamento è quindi amplificato (od intensificato) con un mezzo, in un caso chimico e nell'altro elettronico, per rendere visibile l'immagine. La differenza principale che distingue le due fotocamere è che in una fotocamera digitale la luce è raccolta da un sensore elettronico, mentre in una fotocamera tradizionale la luce è raccolta da uno spezzone di pellicola.

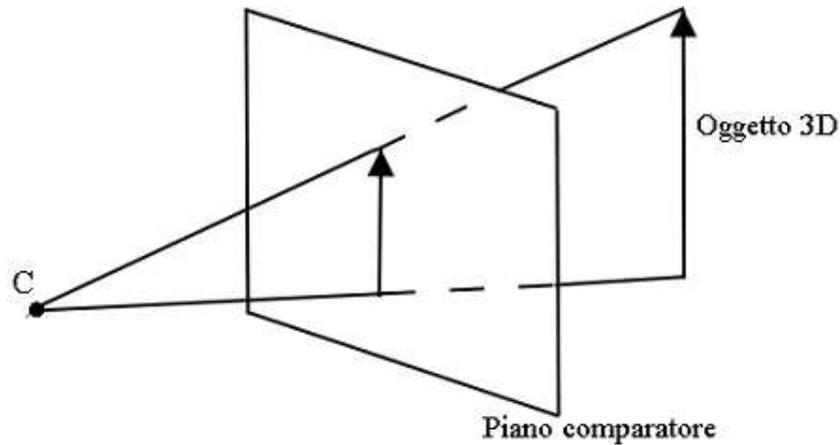


Figura 1.5: Il piano comparatore

In pratica, in un apparecchio digitale tutte le principali fasi di registrazione dell'immagine, acquisizione, elaborazione ed archiviazione hanno luogo all'interno; in un apparecchio a pellicola l'elaborazione e l'archiviazione avvengono all'esterno.

1.2.8 La fotocamera come sistema di misura

I parametri che caratterizzano una fotocamera internamente sono quindi il *punto principale* $p_0(x_0, y_0)$ (l'intersezione dell'asse ottico con il piano immagine) ed f .

1.3 Le distorsioni

Imperfezioni nella costruzione delle singole lenti e nel montaggio degli obiettivi producono *distorsioni* nelle immagini.

Le principali distorsioni sono:

- distorsione radiale;
- distorsione tangenziale.



Figura 1.6: Una fotocamera a pellicola

La *distorsione radiale* causa una traslazione delle coordinate rispetto al centro ottico mentre la *distorsione tangenziale* trasla in modo perpendicolare a quella radiale.

Nelle figura (1.8) é possibile vedere in modo grafico l'effetto della distorsione radiale. Nella prima figura a sinistra (a) abbiamo l'immagine corretta, mentre:

- in b) l'immagine risulta avere una distorsione a *cuscinetto*;
- in c) l'immagine risulta avere una distorsione a *barilotto*.

É dunque necessario riferirsi ad un modello matematico per correggere tali distorsioni.

1.3.1 Modello matematico di distorsione

Come abbiamo visto, le distorsioni sono molto importanti da considerare, in quanto possono infuire in maniera notevole sui dati per il processo di calibrazione.

Il modello matematico per correggere le distorsioni che usiamo nel corso di questa tesi é il seguente:

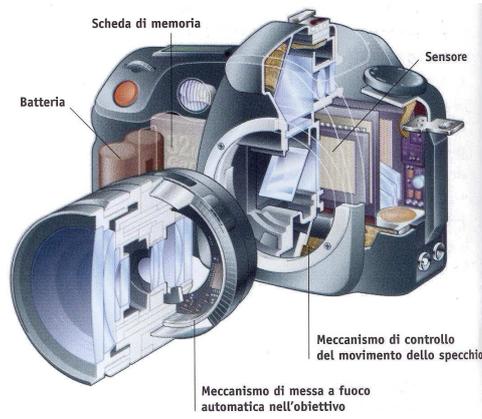


Figura 1.7: Una fotocamera digitale

$$\begin{aligned}\Delta x &= \bar{x}r^2K_1 + \bar{x}r^4K_2 + \bar{x}r^6K_3 + (r^2 + 2\bar{x}^2)P_1 + 2\bar{x}\bar{y}P_2 \\ \Delta y &= \bar{y}r^2K_1 + \bar{y}r^4K_2 + \bar{y}r^6K_3 + 2\bar{x}\bar{y}P_1 + (r^2 + 2\bar{y}^2)P_2\end{aligned}\quad (1.3.1)$$

Questo tipo di modello viene chiamato *modello fisico*, in quanto tutti i suoi componenti possono direttamente essere attribuiti a sorgenti di errori fisici.

I parametri individuali rappresentano:

$$\begin{aligned}\bar{x} &= x - x_0 \\ \bar{y} &= y - y_0 \\ r^2 &= \bar{x}^2 + \bar{y}^2\end{aligned}$$

con

K_1, K_2, K_3 = parametri di distorsione radiale delle lenti

P_1, P_2 = parametri di distorsione tangenziale delle lenti

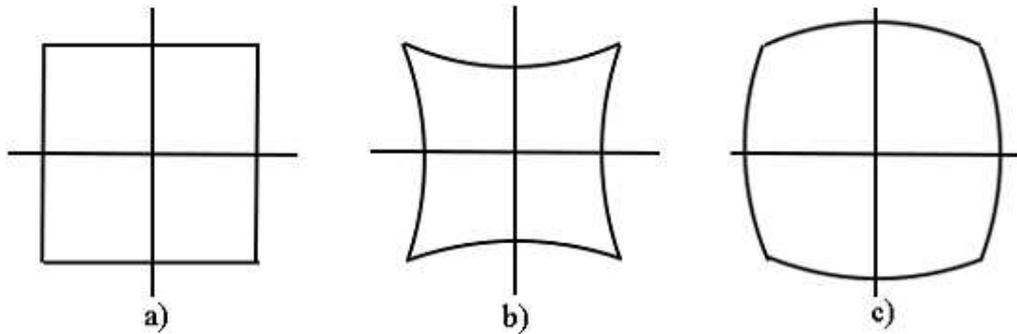


Figura 1.8: Effetti di distorsione radiale

Le equazioni (1.5.15) hanno avuto molto successo nei sistemi con fotocamere digitali in quando risultano molto affidabili. Tengono infatti conto in modo robusto delle distorsioni introdotte dagli obiettivi.

Distorsione radiale delle lenti

La *distorsione radiale* delle lenti é normalmente rappresentata da una serie di polinomi che di solito vengono rappresentati nel seguente modo:

$$\begin{aligned}\Delta x_{RLD} &= K_1(r^2 - 1)x + K_2(r^4 - 1)x + K_3(r^6 - 1)x \\ \Delta y_{RLD} &= K_1(r^2 - 1)y + K_2(r^4 - 1)y + K_3(r^6 - 1)y\end{aligned}\tag{1.3.2}$$

dove

$$r = \sqrt{(x - x_p)^2 + (y - y_p)^2}$$

Il termine K_1 da solo sará usato di solito come suffisso nelle applicazioni con un medio grado di accuratezza.

L'inclusione, invece, dei termini K_2 e K_3 dovrebbe essere richiesto in quelle applicazioni che richiedono un elevato grado di accuratezza. In questo caso la decisione di

includere uno, due o tre termini di distorsione radiale può essere basata su vari test statici di significato.

Distorsione tangenziale delle lenti

La *distorsione tangenziale* delle lenti, invece, è causata dal cattivo allineamento della componentistica della lente. Questi errori possono essere modellati con equazioni di correzione (Brown, 1966) nel seguente modo:

$$\begin{aligned}\Delta x_{DL D} &= P_1(r^2 + 2x^2) + 2P_2xy \\ \Delta y_{DL D} &= P_2(r^2 + 2y^2) + 2P_1xy\end{aligned}\tag{1.3.3}$$

dove P_1 e P_2 rappresentano la distorsione decentrica delle lenti.

1.4 Descrizione della fotocamera rispetto al mondo

1.4.1 I sistemi di riferimento

Allo scopo consideriamo tre sistemi di riferimento usati:

- *Sistema MONDO*: è il sistema di riferimento relativo alla scena che esprime le coordinate mondo dei punti $P_m = [X, Y, Z]^T$ nello spazio;
- *Sistema CAMERA*: è il sistema di riferimento relativo alla fotocamera dove l'asse X_c rappresenta l'asse orizzontale del piano immagine mentre Y_c rappresenta quello verticale. L'asse Z_c coincide con l'asse ottico della fotocamera. Gli assi X_c , Y_c e Z_c formano un sistema di riferimento destrorso e sono disposti in modo che la fotocamera sia orientata verso coordinate negative dell'asse Z_c . Il centro ottico C , situato sull'asse ottico, rappresenta l'origine del sistema di riferimento. Definiamo le coordinate di un punto tridimensionale rispetto al sistema camera con $P_c = [X, Y, Z]^T$;

- *Sistema IMMAGINE*: é il sistema di riferimento che puó essere relativo sia al piano immagine che al piano comparatore dato che questi sono simmetrici. Scegliamo di prendere come piano di riferimento il piano comparatore in modo da evitare che l'immagine proiettata risulti capovolta. L'origine del sistema viene posta nell'angolo in alto a sinistra relativo al piano comparatore.

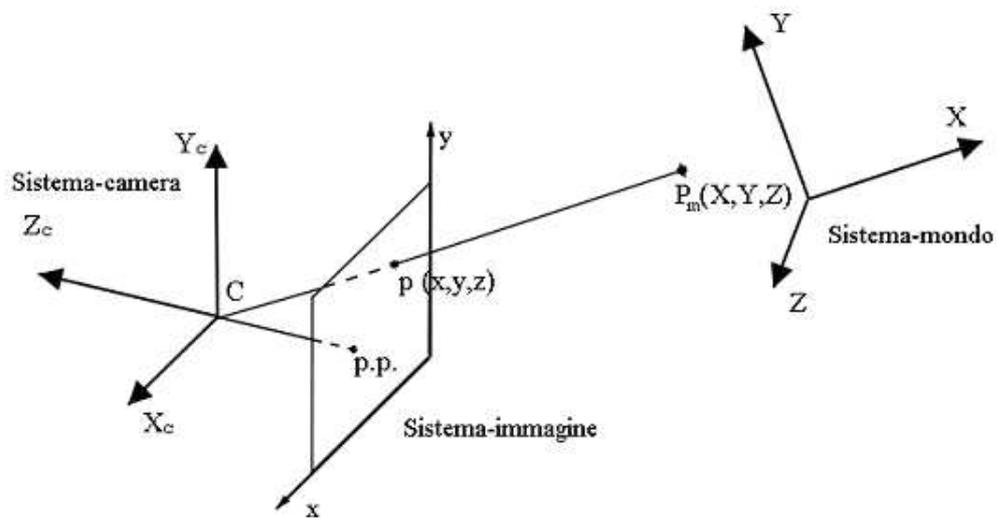


Figura 1.9: Sistema camera, sistema immagine, sistema mondo

1.4.2 Il modello geometrico della proiezione

Le coordinate $p = [x, y]^T$ sono date in pixel mentre lo stesso punto nel sistema camera é espresso mediante le coordinate tridimensionali $p = [x, y, -f]^T$ dove f é la distanza focale.

Indichiamo ora la relazione che sussiste tra un punto nel *sistema mondo* e la sua proiezione sul sistema immagine.

Osservando la figura (1.10) possiamo vedere che i triangoli PCA e pCa sono simili. Questo ci permette di definire la relazione:

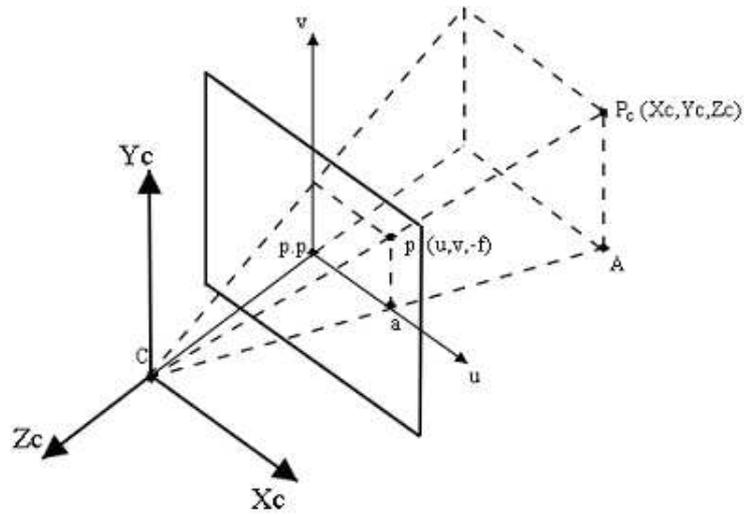


Figura 1.10: Similitudine fra triangoli

$$z : Z_c = u : X_c \quad (1.4.1)$$

In modo analogo possiamo definire:

$$z : Z_c = v : Y_c \quad (1.4.2)$$

Ponendo $z = -f$ otteniamo:

$$u = z \frac{X_c}{Z_c} = -f \frac{X_c}{Z_c} \quad (1.4.3)$$

$$v = z \frac{Y_c}{Z_c} = -f \frac{Y_c}{Z_c} \quad (1.4.4)$$

Essendo presente Z_c al denominatore le equazioni risultano non lineari, dunque non vengono preservate né le distanze tra i punti né gli angoli tra le rette. Tuttavia queste equazioni mantengono la natura delle rette, ovvero la proiezione di una retta nello spazio costituisce sempre una retta.

1.4.3 Rotazione

In realtà é possibile che la telecamera non sia allineata col sistema di riferimento mondo, e quindi é necessario passare da un punto P , con coordinate $P = [X, Y, Z]^T$ riferito al sistema mondo, al punto P_c con coordinate $P_c = [X_c, Y_c, Z_c]^T$ espresso nel sistema di riferimento relativo alla fotocamera, applicando un'operazione di *rototraslazione*.

Le equazioni di trasformazione necessarie per questo scopo possono essere espresse in termini di 6 parametri indipendenti:

- tre angoli di rotazione (ω, ϕ, κ) ;
- tre fattori di traslazione (t_x, t_y, t_z) .

Occorre quindi prima applicare una rotazione R al punto P e successivamente una traslazione t .

$$P_c = RP + t \quad (1.4.5)$$

La matrice di rotazione é data da:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (1.4.6)$$

In particolare

$$R = \begin{bmatrix} \cos\phi \cos\kappa & \sin\omega \sin\phi \cos\kappa + \cos\omega \sin\kappa & -\cos\omega \sin\phi \cos\kappa + \sin\omega \sin\kappa \\ -\cos\phi \sin\kappa & -\sin\omega \sin\phi \sin\kappa + \cos\omega \cos\kappa & \cos\omega \sin\phi \sin\kappa + \sin\omega \cos\kappa \\ \sin\phi & -\sin\omega \cos\phi & \cos\omega \cos\phi \end{bmatrix}$$

Gli elementi individuali della matrice di rotazione sono detti *coseni direttori*, i quali relazioni i due assi del sistema.

Il vettore traslazione vale invece:

$$t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (1.4.7)$$

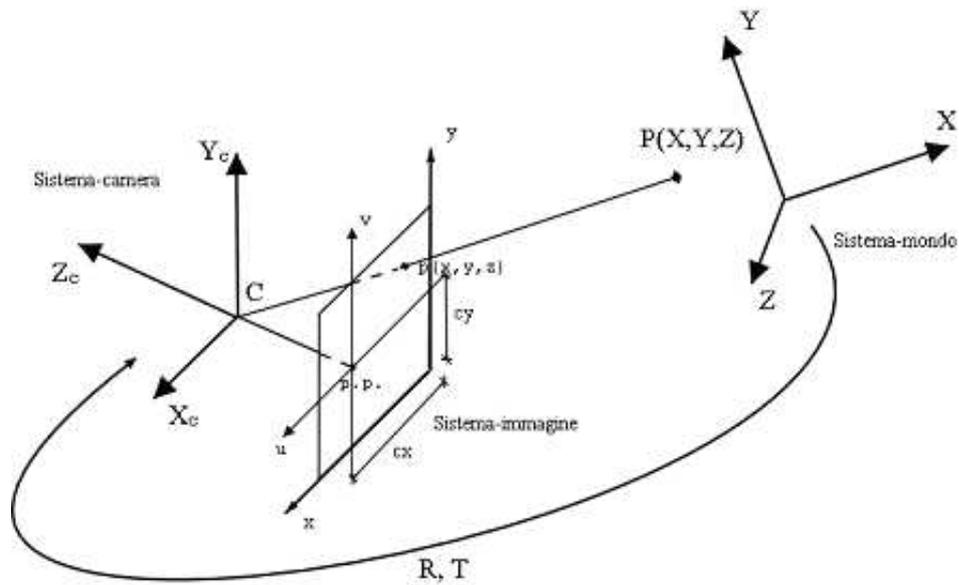


Figura 1.11: Vari cambiamenti dei sistemi di riferimento

1.4.4 Modello di proiezione completo

Nel modello fisico di acquisizione, una volta passati dal sistema di riferimento mondo al sistema camera, si procede a trasformare le coordinate nel sistema immagine (x, y) .

Consideriamo come origine l'angolo in alto a sinistra e come punto principale il punto con coordinate (x_c, y_c) .

Le coordinate del sistema sono espresse in pixel dove dx rappresenta la dimensione orizzontale e dy la dimensione verticale del singolo pixel.

$$x = \frac{u + x_c}{dx} \quad (1.4.8)$$

$$y = \frac{u + y_c}{dy} \quad (1.4.9)$$

1.5 La condizione di collinearit 

Un primo problema da risolvere sar  quello di trovare la posizione degli oggetti nello spazio reale.

Il modello matematico prospettico considerato in questo lavoro   basato sulla *condizione di collinearit * (figura 1.12).

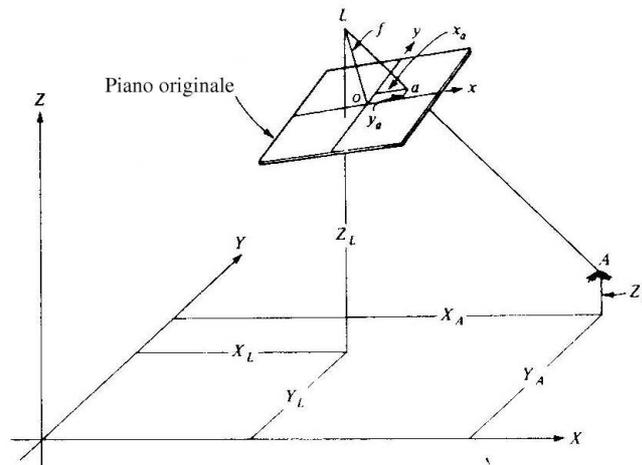


Figura 1.12: La condizione di collinearit 

Le *equazioni di collinearit * descrivono il fatto che il punto dell'oggetto, il corrispondente punto immagine ed il centro prospettico giacciono su di una linea retta.

Queste sono quasi sicuramente le equazioni pi  usate nella fotogrammetria.

Nella figura successiva (1.13), invece, si pu  vedere che la stazione di esposizione L di una foto aerea ha coordinate X_L , Y_L e Z_L rispetto al sistema di coordinate dell'oggetto XYZ .

L'immagine a di un punto oggetto A , mostrato in un piano immagine *ruotato*, ha coordinate spaziali x'_a , y'_a e z'_a dove l'immagine ruotata del sistema di coordinate $x'y'z'$   parallelo allo spazio oggetto del sistema di coordinate $X_A Y_A Z_A$.

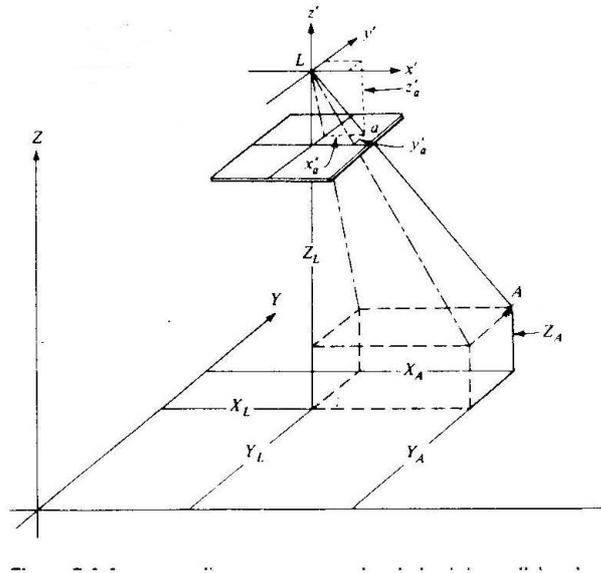


Figura 1.13: Sistema di coordinate immagine ruotato in modo da essere parallelo al sistema di coordinate dello spazio oggetto

In questo modo la condizione di collinearit  assume una geometria dell'immagine ideale ed   espressa come:

$$\begin{bmatrix} x' - x'_a \\ y' - y'_a \\ 0 - f \end{bmatrix} = \lambda M \begin{bmatrix} X_A - X_L \\ Y_A - Y_L \\ Z_A - Z_L \end{bmatrix} \quad (1.5.1)$$

in particolare:

$$\begin{bmatrix} x' - x'_a \\ y' - y'_a \\ 0 - f \end{bmatrix} = \lambda \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_A - X_L \\ Y_A - Y_L \\ Z_A - Z_L \end{bmatrix} \quad (1.5.2)$$

dove:

$x' y'$ = coordinate immagine misurate del punto A

$x'_a y'_a$ = coordinate del punto principale a

f = focale della macchina fotografica

$X_A Y_A Z_A$ = coordinate dello spazio oggetto del punto oggetto A

$X_L Y_L Z_L$ = coordinate dello spazio oggetto del centro prospettico L

λ = fattore di scala tra i vettori immagine ed oggetto

M = matrice di rotazione ortogonale 3X3

r_{ij} = componenti della matrice ortogonale di rotazione

Le equazioni di collinearit  sono sviluppate prendendo come riferimento i triangoli simili della figura (1.13) come segue:

$$\frac{x' - x'_a}{X_A - X_L} = \frac{y' - y'_a}{Y_A - Y_L} = \frac{-z'_a}{Z_A - Z_L}$$

riducendo si trova

$$\begin{aligned} x' - x'_a &= \left(\frac{X_A - X_L}{Z_A - Z_L} \right) z'_a \\ y' - y'_a &= \left(\frac{Y_A - Y_L}{Z_A - Z_L} \right) z'_a \end{aligned} \quad (1.5.3)$$

Per l'identit 

$$z'_a = \left(\frac{Z_A - Z_L}{Z_A - Z_L} \right) z'_a \quad (1.5.4)$$

Sostituendo le equazioni (1.5.3) e (1.5.4) nelle equazioni (1.5.1) troviamo che

$$x' - x'_a = r_{11} \left(\frac{X_A - X_L}{Z_A - Z_L} \right) + r_{12} \left(\frac{Y_A - Y_L}{Z_A - Z_L} \right) + r_{13} \left(\frac{Z_A - Z_L}{Z_A - Z_L} \right) \quad (1.5.5)$$

$$y' - y'_a = r_{21} \left(\frac{X_A - X_L}{Z_A - Z_L} \right) + r_{22} \left(\frac{Y_A - Y_L}{Z_A - Z_L} \right) + r_{23} \left(\frac{Z_A - Z_L}{Z_A - Z_L} \right) \quad (1.5.6)$$

$$0 - f = r_{31} \left(\frac{X_A - X_L}{Z_A - Z_L} \right) + r_{32} \left(\frac{Y_A - Y_L}{Z_A - Z_L} \right) + r_{33} \left(\frac{Z_A - Z_L}{Z_A - Z_L} \right) \quad (1.5.7)$$

Fattorizzando il termine $\left(\frac{z'_a}{Z_a} - Z_L\right)$ dalle equazioni (1.5.5) attraverso (1.5.7) e dividendo per (1.5.5) e (1.5.6) con (1.5.7), sostituendo $(-f)$ in z_a , le *equazioni di collinearit *, introdotte per la prima volta da *Otto von Gruber nel 1930*, risultano essere

$$x' - x'_a = -f \left[\frac{r_{11}(X_A - X_L) + r_{12}(Y_A - Y_L) + r_{13}(Z_A - Z_L)}{r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(Z_A - Z_L)} \right] \quad (1.5.8)$$

$$y' - y'_a = -f \left[\frac{r_{21}(X_A - X_L) + r_{22}(Y_A - Y_L) + r_{23}(Z_A - Z_L)}{r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(Z_A - Z_L)} \right] \quad (1.5.9)$$

dove

$x' y'$ = coordinate immagine misurate del punto A

$x'_a y'_a$ = coordinate del punto principale a

f = lunghezza focale della macchina fotografica

$X_A Y_A Z_A$ = coordinate dello spazio oggetto del punto oggetto A

$X_L Y_L Z_L$ = coordinate dello spazio oggetto del centro prospettico L

r_{ij} = componenti della matrice ortogonale di rotazione

La *forma standard* delle equazioni di collinearit  risulta essere

$$x' - x'_a = -f \left[\frac{U}{W} \right] \quad (1.5.10)$$

$$y' - y'_a = -f \left[\frac{V}{W} \right] \quad (1.5.11)$$

con

$$\begin{aligned} U &= r_{11}(X_A - X_L) + r_{12}(Y_A - Y_L) + r_{13}(Z_A - Z_L) \\ V &= r_{21}(X_A - X_L) + r_{22}(Y_A - Y_L) + r_{23}(Z_A - Z_L) \\ W &= r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(Z_A - Z_L) \end{aligned} \quad (1.5.12)$$

1.5.1 Modello completo

Il modello collineare deve estendersi per venire incontro alla realt  fisica, introducendo alcuni *errori sistematici*. Questi errori sono compensati con correzioni per le coordinate immagine, che sono funzioni di un insieme di *parametri addizionali* (denominati AP).

I parametri addizionali che usiamo nel corso di questa tesi sono:

- parametri K_1 , K_2 e K_3 di *distorsione radiale delle lenti*;
- parametri P_1 e P_2 di *distorsione decentrica delle lenti*;

Queste distorsioni sono molto importanti da considerare, in quanto possono influire in maniera notevole sui dati per il processo di calibrazione.

Le equazioni di collinearit  in modo *esteso* hanno quindi la seguente forma:

$$x' - x'_a + \Delta x = -f \left[\frac{U}{W} \right] \quad (1.5.13)$$

$$y' - y'_a + \Delta y = -f \left[\frac{V}{W} \right] \quad (1.5.14)$$

dove

$$\begin{aligned} \Delta x &= \bar{x}r^2K_1 + \bar{x}r^4K_2 + \bar{x}r^6K_3 + (r^2 + 2\bar{x}^2)P_1 + 2\bar{x}\bar{y}P_2 \\ \Delta y &= \bar{y}r^2K_1 + \bar{y}r^4K_2 + \bar{y}r^6K_3 + 2\bar{x}\bar{y}P_1 + (r^2 + 2\bar{y}^2)P_2 \end{aligned} \quad (1.5.15)$$

Capitolo 2

La calibrazione

2.1 Introduzione

La *calibrazione* costituisce un elemento fondamentale nell'ambito dei problemi di visione tridimensionale computerizzata. Risolvere un problema di calibrazione significa trovare le caratteristiche della fotocamera, anche conosciute come *parametri intrinseci*, *parametri estrinseci* e *parametri ottici*

- i *parametri intrinseci* sono i parametri necessari a collegare le coordinate in pixel dei punti di un'immagine con i punti corrispondenti nel sistema di riferimento *camera*;
- i *parametri estrinseci* sono i parametri che definiscono la localizzazione e l'orientamento della struttura di riferimento della fotocamera rispetto ad una struttura di riferimento conosciuta nel sistema *mondo*;
- i *parametri ottici* sono i parametri che definiscono il campo di distorsione.

La conoscenza dei parametri *intrinseci*, *estrinseci* ed *ottici* della fotocamera é necessaria per permettere una ricostruzione tridimensionale accurata della scena reale.

Ci sono molte tecniche di calibrazione sviluppate in questi ultimi anni che possiamo usare per questo scopo.

Ciascun tipo di queste tecniche può essere classificata in due differenti categorie:

- *Calibrazione con punti noti*: questo tipo di calibrazione é effettuata osservando un oggetto di cui la geometria nello spazio tridimensionale é conosciuta con una precisione molto alta. L'oggetto della calibrazione di solito consiste in due o tre piani ortogonali ad un altro piano. Questo tipo di approccio, comunque, richiede un setup elaborato ma é un metodo di calibrazione efficiente;
- *Auto calibrazione*: questo metodo di calibrazione non usa un oggetto di calibrazione. Muovendo la macchina fotografica in una scena statica, la rigiditá stessa della scena fornisce vincoli sui parametri interni ed esterni della fotocamera usando informazioni prese dalle immagini. Tre immagini prese dalla stessa fotocamera con parametri intrinseci fissi sono sufficienti per determinare tutti gli altri parametri incogniti. Anche se l'approccio risulta molto flessibile, si tratta di una tecnica che non permette di ottenere risultati sempre affidabili.

Nei successivi paragrafi si parlerá di alcune tecniche di calibrazione elaborate dalla Prima guerra mondiale fino agli anni '60, per poi illustrare le problematiche relative alla calibrazione. Emerge l'importanza di calibrare una fotocamera in modo corretto con tecniche robuste in grado di determinare i *parametri intrinseci*, *estrinseci* ed *ottici* della stessa. Una tecnica adeguata per risolvere i problemi della calibrazione é la tecnica del *bundle adjustment*. Vi sono, comunque, diverse tecniche di utilizzo di tale modello.

2.2 Calibrazione e fotogrammetria: cenni storici

Giá negli anni precedenti alla *Prima guerra mondiale* si sono cercate tecniche per calibrare correttamente le fotocamere che erano utilizzate per mappare le carte topografiche a fini bellici. Le procedure utilizzate, comunque, erano piú qualitative

che quantitative. I metodi usati, infatti, introducevano molti errori geometrici causati dall'imprecisione delle fotocamere utilizzate.

Molti ritengono che il pioniere della calibrazione fu il canadese *Deville*, che nel 1910 si costruì un piccolo laboratorio personale per calibrare i componenti ottici della sua fotocamera.

La prima fotocamera montata sugli aerei fu calibrata in Canada nel 1920 ed, anche se ancora imprecisa, da questa tecnica furono tratte alcune distanze principali fra vari paesi e la localizzazione dei principali punti nodali. Per i 30 anni che seguirono, la tecnica sviluppata fu essenzialmente la stessa, applicando delle rifiniture nel corso degli anni. Questa tecnica prese il nome di *calibrazione visuale*.

Molte agenzie governative americane, visto il reale apporto dato da questa tecnica di calibrazione, cominciarono ad ordinare fotocamere già calibrate dal *National Bureau of Standards* ancora prima dello scoppio della Prima guerra mondiale.

La *Seconda guerra mondiale* portò ad un incremento dell'uso della fotografia aerea per ricognizioni e mappature. Per tale motivo, verso la fine degli anni '40, si svolse a Parigi un meeting fra i paesi europei più rappresentativi nelle tecniche di calibratura delle fotocamere che portò alla standardizzazione della stessa. I risultati delle loro delibere furono discusse a Washington nel 1951 dalla *American Society of Photogrammetry*. I risultati del meeting provvidero a risolvere alcuni problemi riguardanti le qualità delle immagini e di come scattare foto all'aperto, dove la luce del sole incide molto sulla qualità della stessa. Vi erano comunque molti problemi legati ad *effetti tangenziali* che furono menzionati solo dopo la Seconda guerra mondiale. Non c'erano metodi per correggere queste distorsioni asimmetriche e tangenziali, equivalenti all'attuale *effetto prisma*. Il problema si rilevò di difficile soluzione, tanto che fu accettata una tolleranza su questo errore, pari a 15μ .

Vi erano comunque altri aspetti della calibrazione che, anche se importanti, rappresentavano un problema di difficile soluzione e vennero, per tal motivo, trascurati.

Un'autorità che favorì la calibrazione delle fotocamere da campo fu la *US Naval Photographic Interpretation Centre*. Merrit nel 1951 discusse molte tecniche di calibrazione ma indicò una forte preferenza per i metodi che usavano goniometri o che usavano l'esposizione delle stelle.

Il *VII Congresso della Fotogrammetria* nel 1952 adottò la risoluzione della *Commissione I* che suggerì che la calibrazione di fotocamere con procedure fotogrammetriche era preferibile ai metodi visuali. I metodi visuali tramite goniometro furono comunque molto applicati in Europa.

Nel mezzo degli anni '50 si riprese a discutere sulle *distorsioni radiali*. Lewis nel 1956 elaborò delle ipotesi sulla natura delle distorsioni radiali e di come la differente magnetudine potesse influenzare il valore scelto per la lunghezza focale della fotocamera.

Considerazioni importanti nei risultati di calibrazione per lo stesso tipo di macchine fotografiche furono notate da Hothmer nel 1958. Egli asserì che gli effetti della variazione della temperatura quando si calibrava in laboratorio o quando si calibrava in aereo erano elementi importanti da tenere in seria considerazione in quanto si poteva ottenere una distorsione pari a $\pm 5\mu$.

Thompson nel 1957 discusse la teoria geometrica della macchina fotografica. Propose un metodo che, per l'apparecchiatura progettata a quei tempi, permetteva difetti di costruzione ed errori d'osservazione nella calibrazione.

Hallert nel 1963 discusse il *metodo dei minimi quadrati* applicato alla calibrazione di macchine fotografiche a collimazione multipla. Egli realizzò un nuovo metodo di calibrazione e ripeté i suoi test con entrambe le pellicole e lastre di vetro. A parte fornire alcuni significati statistici alle distorsioni radiali ed altri parametri della calibrazione da lui computati, Hallert notò una significativa differenza nelle curve della distorsione radiale trovate dalla pellicole e dalle lastre di vetro. Questo indicò che ci sono sorgenti addizionali di questo errore nei negativi delle pellicole, probabilmente

causati dalla planarità difettosa nei supporti.

Nel 1950, gli studiosi impegnati nella fabbricazione e nella calibrazione di macchine fotografiche aeree furono ovviamente interessati allo studio ed alla discussione delle caratteristiche delle distorsioni delle lenti. Altri scienziati e fotogrammetristi furono impegnati in questi lavori, ma i loro studi si rivelarono più di natura confidenziale che scientifica e non furono di pubblico dominio fino a quando Brown pubblicò una serie di relazioni importanti negli anni '60, riferite agli esperimenti segreti dell' *US Air Force Missile Test Base* nel bel mezzo degli anni '50.

La calibrazione di una fotocamera, fin dal passato, è stato un complesso problema di studio.

2.3 Problemi attuali relativi alla calibrazione

Per poter risolvere in modo accurato il problema della calibrazione occorre utilizzare il metodo del *Bundle Adjustment*, elaborato da Brown nel 1965.

Nel prossimi due paragrafi verrà trattato quindi il modello *Bundle Adjustment* e di come questa tecnica è stata ulteriormente migliorata grazie alla tecnica dell' *auto-calibrazione*.

2.3.1 Il modello *Bundle Adjustment*

Nel 1965, Brown presentò il risultato di alcuni suoi esperimenti al Convegno Annuale della *American Society of Photogrammetry*, poi pubblicati col nome *Decentric distortion of lens* nel 1966. In questa pubblicazione, Brown rivisitò il lavoro scientifico dei precedenti 40 anni che in gran parte avevano identificato la distorsione tangenziale come effetto ottenuto disponendo un prisma sottile davanti all'obiettivo.

Secondo Brown era essenziale avere un modello di obiettivo accurato nel limite della misura del suo comparatore (approssimativamente $1 \mu m$), a causa degli

avanzamenti nella triangolazione fotogrammetrica analita.

Brown, grazie ad un suo precedente lavoro, fece vedere come la distorsione tangenziale fosse principalmente attribuibile al decentramento.

Fin dal 1956 aveva pubblicato i rapporti tecnici sulla determinazione simultanea dei parametri dell'obiettivo e dell'orientamento della fotocamera ed aveva sviluppato il *Bundle adjustment* come modello per risolvere simultaneamente coordinate date, localizzazione di fotocamere e parametri per gli obiettivi.

Per Brown la *distorsione radiale* ed il *decentramento* potevano essere modellate efficacemente in quanto non ci sarebbero stati ostacoli nell'uso di buoni obiettivi della lunghezza focale adatta, con varie aperture e campi angolari distinti.

Brown criticó in quegli anni anche la tecnica di calibrazione della *US Coast and Geodetic Survey*, suscitando dichiarazioni provocatorie.

Ci vollero comunque molti anni prima che le sue idee guadagnassero l'accettazione completa nel mondo della cartografia, nelle agenzie di mappatura e nelle autorità governative, molti dei quali non avevano ancora la consapevolezza dell'importante uso che rappresentavano i primi computer.

2.3.2 Il modello Bundle Adjustment con auto calibrazione

Lo sviluppo e l'uso del modello *bundle adjustment* di Brown dal 1965, insieme con le *fotografie terrestri* (Ziemann e El-Hakim) ha significato che era teoricamente e praticamente possibile determinare i parametri incogniti della fotocamera in maniera simultanea avendo a disposizione un certo numero di punti di controllo di cui é nota la posizione nello spazio tridimensionale.

Questa tecnica é diventata nota con il nome di *Bundle Adjustment* ed é un metodo particolarmente robusto in grado di determinare tutti i parametri incogniti. Esso puó essere utilizzato sia con punti di controllo noti come in questo lavoro, oppure con punti non noti (*auto calibrazione*).

Le pubblicazioni che parlano di questo metodo cominciano ad apparire nel 1972 (Kenefich, Gyer e Harp). Il congresso ISPRS nel 1976 fondó un gruppo di lavoro per studiare questa tecnica e, nel 1980, E. Kilpela mostró come i vari membri del gruppo di lavoro avessero sviluppato molte tecniche differenti per determinare i parametri di *calibrazione*.

Sempre Brown nel 1972 aveva riconosciuto l'alta correlazione che esisteva fra determinati parametri e la localizzazione e gli orientamenti delle fotocamere, mentre Fraser nel 1982 aveva dimostrato che l'uso di troppi parametri addizionali (*sovraparametrizzazione*) poteva indebolire la soluzione del modello.

2.3.3 Altre tecniche di calibrazione

Nel corso degli anni, comunque, sono state studiate e sviluppate molte tecniche di calibrazione. Alcune non sono piú state usate, altre sono state migliorate. Vengono presentate due tecniche che erano usate prima dell'uso del Bundle Adjustment.

Il metodo Plumb-Line

However, nel 1971, sviluppó il metodo *plumb-line* apposta per la calibrazione di fotocamere usando linee rette. In una proiezione prospettica, l'immagine di una linea retta avrà come risultato una linea retta se non sono presenti distorsioni nelle lenti.

Le deviazioni nell'immagine da linee rette sono direttamente riferite alla presenza di distorsioni radiali e di decentramento, cosí che Brown descrisse un modello matematico per determinare i parametri delle distorsioni causate dagli obiettivi basati sulle immagini delle linee rette.

Brown, inoltre, con questo modello determinó le formule in grado di modellare esattamente la distorsione radiale in una gamma di lunghezze focali differenti. Se i parametri di distorsione radiale di due distanze separate (regolazione focale) sono

conosciute, i valori di tutte le regolazioni tra queste due distanze dovrebbero essere trovate in modo molto accurato.

I vantaggi di questa tecnica sono molteplici:

1. questo metodo é ragionevolmente semplice da programmare su di un computer ed é soprattutto un metodo pratico che non richiede un laboratorio elaborato e strumenti costosi;
2. é molto semplice raccogliere un grosso numero di punti e quindi verificare subito la soluzione esatta per i parametri.

Principalmente per questi due motivi, l'uso della tecnica *plumb-line* é stata usata nelle fotocamere elettroniche e nelle misurazioni automatiche dei sistemi fino al 1986, quando cioé Fryer e lo stesso Brown modificarono il modello aggiungendo alcune semplici modifiche soprattutto per quanto riguardava le distorsioni decentrate.

Il modello *plumb-line*, comunque, presentava già alcuni problemi. In particolare mostrava difficoltà con le cosiddette *lenti ad occhio di pesce*, dove la distorsione é molto ampia. Per una determinazione degli spostamenti del punto principale, che sono stati indicati da vari ricercatori essere direttamente correlati con i parametri della distorsione decentrica, sono necessarie tecniche di laboratorio oppure l'uso del modello *bundle adjustment con auto calibrazione*.

Calibrazione on-the-job

Lo sviluppo della tecnica *bundle adjustment* ha fatto si che nuove tecniche per la calibrazione delle fotocamere si affermarono. Una di questa é la calibrazione *on-the-job*.

In questo approccio i punti di controllo sono messi intorno all'oggetto. É un approccio facilmente implementabile in laboratorio e dá risultati molto accurati. Questo metodo é anche usato dalle autorità governative per calibrazioni aeree.

Brown nel 1989 ha discusso i criteri che bisogna seguire per ottenere una calibrazione esatta:

1. dovrebbe essere usata solamente una fotocamera per fare almeno tre foto dell'oggetto;
2. sia la geometria interna della macchina fotografica sia i punti misurati sull'oggetto dovrebbero rimanere stabili durante il processo di misurazione;
3. il metodo usato dovrebbe essere in grado di avere un elevato grado di convergenza in pochi passi (*step*);
4. almeno una delle immagini dovrebbe avere un angolo *roll* significativamente diverso dagli altri;
5. dovrebbe essere usato un numero relativamente grande di punti di controllo bene distribuiti.

Capitolo 3

Il modello bundle adjustment

3.1 Introduzione al modello Bundle Adjustment

Il *bundle adjustment* é dunque un metodo di calibrazione che consente di stimare i parametri intrinseci ed estrinseci della fotocamera minimizzando iterativamente l'errore di retroproiezione dei punti di controllo.

Possiamo usare due termini per definire la robustezza del modello: ottimale ed insieme. Il primo termine, *ottimale*, stá ad indicare che la valutazione dei parametri incogniti é trovata minimizzando alcune funzioni che quantificano l'errore del modello. Il termine *insieme*, invece, significa che la soluzione é simultaneamente ottimale riguardo anche agli eventuali cambiamenti del modello della fotocamera o della struttura usata.

Il nome bundle adjustment si riferisce ai *bundles* - letteralmente fasci - raggi luminosi che convergono al centro della fotocamera, la quale é ottimamente *adjusted* - registrata - perché il modello geometrico della fotocamera e la sua posizione tridimensionale sono tali per cui i *bundle* passano per il centro del sistema ottico della fotocamera.

Quasi tutto quanto abbiamo detto fino ad ora e che diremo succesivamente puó

essere applicato a molti problemi simili di valutazione nella visione, nella fotogrammetria, nella metrologia industriale e nella geologia. Una volta che la teoria ed i metodi di base sono capiti, sono facili da adattarsi ad un'ampia variet  di problemi, adattamento che   soprattutto un aspetto di scelta dello schema numerico di ottimizzazione che sfrutta la struttura del problema.

Classicamente, il bundle adjustment   formulato come un *problema di minimi quadrati non lineari*. Il costo computazionale del modello cresce quadraticamente con il numero di punti tridimensionali considerati per il calcolo degli errori di retroproiezione.

Il bundle adjustment ha tre caratteristiche a proprio favore:

1. *flessibilit *: il bundle adjustment funziona anche nel caso in cui alcune caratteristiche tridimensionali o bidimensionali non sono note;
2. *accuratezza*: il bundle adjustment d  risultati precisi e facilmente interpretabili perch  usa accurati modelli sugli errori statistici;
3. *efficienza*: gli algoritmi *bundle*, se comparati con altri metodi, sono efficienti anche su problemi molto complessi.

3.2 Linearizzazione delle equazioni di collinearit 

Il modello bundle adjustment si basa sulle equazioni di collinearit  definite nel primo capitolo.

Nei nostri studi, abbiamo supposto che sia le coordinate tridimensionali X_A , Y_A e Z_A sia le coordinate immagine x ed y di ciascun punto siano note. Quindi le equazioni di collinearit  risultano avere nove fattori incogniti, oltre agli eventuali *parametri addizionali* (AP) di distorsione radiale e tangenziale.

In particolare abbiamo come incognite da stimare:

1. le tre coordinate dello spazio oggetto del centro prospettico X_L, Y_L e Z_L ;
2. i tre angoli di rotazione ω, ϕ, κ inerenti alla matrice di rotazione R ;
3. le due coordinate del punto principale x'_a ed y'_a ;
4. la focale della macchina fotografica f ;
5. i parametri di distorsione radiale e tangenziale K_1, K_2, K_3, P_1 e P_2 .

Ricordiamo che le due equazioni di collinearità

$$x' - x'_a = -f \left[\frac{r_{11}(X_A - X_L) + r_{12}(Y_A - Y_L) + r_{13}(Z_A - Z_L)}{r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(Z_A - Z_L)} \right] + \Delta x$$

$$y' - y'_a = -f \left[\frac{r_{21}(X_A - X_L) + r_{22}(Y_A - Y_L) + r_{23}(Z_A - Z_L)}{r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(Z_A - Z_L)} \right] + \Delta y$$

possono anche essere scritte in forma standard come:

$$x' - x'_a = -f \left[\frac{U}{W} \right] + \Delta x \quad (3.2.1)$$

$$y' - y'_a = -f \left[\frac{V}{W} \right] + \Delta y \quad (3.2.2)$$

con

Δx = distorsione radiale e tangenziale su x

Δy = distorsione radiale e tangenziale su y

$$U = r_{11}(X_A - X_L) + r_{12}(Y_A - Y_L) + r_{13}(Z_A - Z_L) \quad (3.2.3)$$

$$V = r_{21}(X_A - X_L) + r_{22}(Y_A - Y_L) + r_{23}(Z_A - Z_L)$$

$$W = r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(Z_A - Z_L)$$

Le equazioni di collinearitá, comunque, hanno il problema di essere *non lineari*. Ad ogni modo, esse possono essere linearizzate usando il *teorema di Taylor*; in questo modo é possibile ottenere una descrizione locale di (3.2.1) e (3.2.2).

Linearizzando le equazioni di collinearitá, possiamo riscrivere le equazioni (3.2.1) e (3.2.2) nel seguente modo:

$$F = x'W - x'_a W + fU - \Delta x = 0 \quad (3.2.4)$$

$$G = y'W - y'_a W + fV - \Delta y = 0 \quad (3.2.5)$$

In forma linearizzata troviamo

$$AV + B\Delta + F = 0$$

$$AV + B\Delta + G = 0$$

La *matrice di disegno* A é trovata prendendo le derivate parziali delle equazioni di proiezione ripetendo i parametri, mentre Δ rappresenta l'incremento delle variabili indipendenti. Sviluppando le derivate parziali troviamo che

$$\begin{aligned} & \left(\frac{\partial F}{\partial x'}\right)_0 dx' + \left(\frac{\partial F}{\partial x'_a}\right)_0 dx'_a + \left(\frac{\partial F}{\partial y'_a}\right)_0 dy'_a + \left(\frac{\partial F}{\partial f}\right)_0 df + \\ & + \left(\frac{\partial F}{\partial X_L}\right)_0 dX_L + \left(\frac{\partial F}{\partial Y_L}\right)_0 dY_L + \left(\frac{\partial F}{\partial Z_L}\right)_0 dZ_L + \left(\frac{\partial F}{\partial \omega}\right)_0 d\omega + \\ & + \left(\frac{\partial F}{\partial \phi}\right)_0 d\phi + \left(\frac{\partial F}{\partial \kappa}\right)_0 d\kappa + \left(\frac{\partial F}{\partial K_1}\right)_0 dK_1 + \left(\frac{\partial F}{\partial K_2}\right)_0 dK_2 + \\ & + \left(\frac{\partial F}{\partial K_3}\right)_0 dK_3 + \left(\frac{\partial F}{\partial P_1}\right)_0 dP_1 + \left(\frac{\partial F}{\partial P_2}\right)_0 dP_2 + (F)_0 = 0 \end{aligned} \quad (3.2.6)$$

$$\begin{aligned}
& \left(\frac{\partial G}{\partial y'}\right)_0 dy' + \left(\frac{\partial G}{\partial x'_a}\right)_0 dx'_a + \left(\frac{\partial G}{\partial y'_a}\right)_0 dy'_a + \left(\frac{\partial G}{\partial f}\right)_0 df + \\
& + \left(\frac{\partial G}{\partial X_L}\right)_0 dX_L + \left(\frac{\partial G}{\partial Y_L}\right)_0 dY_L + \left(\frac{\partial G}{\partial Z_L}\right)_0 dZ_L + \left(\frac{\partial G}{\partial \omega}\right)_0 d\omega + \\
& + \left(\frac{\partial G}{\partial \phi}\right)_0 d\phi + \left(\frac{\partial G}{\partial \kappa}\right)_0 d\kappa + \left(\frac{\partial G}{\partial K_1}\right)_0 dK_1 + \left(\frac{\partial G}{\partial K_2}\right)_0 dK_2 + \\
& + \left(\frac{\partial G}{\partial K_3}\right)_0 dK_3 + \left(\frac{\partial G}{\partial P_1}\right)_0 dP_1 + \left(\frac{\partial G}{\partial P_2}\right)_0 dP_2 + (G)_0 = 0
\end{aligned} \tag{3.2.7}$$

Nelle equazioni (3.2.6), (3.2.7), $(F)_0$ e $(G)_0$ sono le funzioni F e G nelle equazioni (3.2.4) e (3.2.5) valutate come approssimazioni iniziali dei fattori incogniti; i termini $\left(\frac{\partial F}{\partial \omega}\right)_0$, $\left(\frac{\partial F}{\partial \phi}\right)_0$, $\left(\frac{\partial F}{\partial \kappa}\right)_0$, etc sono le derivate parziali delle funzioni F e G . I valori $d\omega$, $d\phi$, $d\kappa$, etc sono le correzioni incognite che devono essere applicate alle approssimazioni iniziali. I $d\omega$, $d\phi$ e $d\kappa$ sono valori espressi in radianti. Poich  dx' e dy' sono correzioni delle coordinate x' ed y' misurate con la fotocamera, esse dovrebbero essere interpretate come errori residui nelle misurazioni. Questi due termini dovrebbero quindi essere sostituiti con $v_{x'}$ e $v_{y'}$, simboli usati per definire gli errori residui.

Sostituendo quindi dx' e dy' con $v_{x'}$ e $v_{y'}$, possiamo quindi ottenere le seguenti due formule *linearizzate* delle equazioni di collinearit :

$$\begin{aligned}
v_{x_a} = & a_{11}dx'_a + a_{12}dy'_a + a_{13}df + a_{14}dX_L + a_{15}dY_L + a_{16}dZ_L + a_{17}d\omega + a_{18}d\phi + \\
& + a_{19}d\kappa + a_{110}dK_1 + a_{111}dK_2 + a_{112}dK_3 + a_{113}dP_1 + a_{114}dP_2 + J
\end{aligned} \tag{3.2.8}$$

$$\begin{aligned}
v_{y_a} = & a_{21}dx'_a + a_{22}dy'_a + a_{23}df + a_{24}dX_L + a_{25}dY_L + a_{26}dZ_L + a_{27}d\omega + a_{28}d\phi + \\
& + a_{29}d\kappa + a_{210}dK_1 + a_{211}dK_2 + a_{212}dK_3 + a_{213}dP_1 + a_{214}dP_2 + K
\end{aligned} \tag{3.2.9}$$

Nelle equazioni (3.2.8) e (3.2.9), J e K sono uguali a $(F)_{0/q}$ e $(G)_{0/q}$, mentre i termini a_{ij} indicano la posizione di ciascuna derivata parziale all'interno della matrice di disegno A .

Quindi possiamo scrivere le due equazioni di collinearit  linearizzate (3.2.8) (3.2.9) di cui sopra per ogni punto di cui conosciamo sia le coordinate tridimensionali che le coordinate del punto oggetto. La matrice di disegno A sar  dunque:

$$A = \begin{bmatrix} dx'_{ax} & dy'_{ax} & df_x & dX_{Lx}^1 & dY_{Ly}^1 & dZ_{Lz}^1 & d\omega_x^1 & d\phi_x^1 & d\kappa_x^1 & dK_{1x}^1 & dK_{2x}^1 & dK_{3x}^1 & dP_{1x}^1 & dP_{2x}^1 \\ dx'_{ay} & dy'_{ay} & df_y & dX_{Ly}^1 & dY_{Lx}^1 & dZ_{Ly}^1 & d\omega_y^1 & d\phi_y^1 & d\kappa_y^1 & dK_{1y}^1 & dK_{2y}^1 & dK_{3y}^1 & dP_{1y}^1 & dP_{2y}^1 \\ dx'_{ax} & dy'_{ax} & df_x & dX_{Lx}^2 & dY_{Ly}^2 & dZ_{Lz}^2 & d\omega_x^2 & d\phi_x^2 & d\kappa_x^2 & dK_{1x}^2 & dK_{2x}^2 & dK_{3x}^2 & dP_{1x}^2 & dP_{2x}^2 \\ dx'_{ay} & dy'_{ay} & df_y & dX_{Ly}^2 & dY_{Lx}^2 & dZ_{Ly}^2 & d\omega_y^2 & d\phi_y^2 & d\kappa_y^2 & dK_{1y}^2 & dK_{2y}^2 & dK_{3y}^2 & dP_{1y}^2 & dP_{2y}^2 \\ \vdots & \vdots \\ dx'_{ax} & dy'_{ax} & df_x & dX_{Lx}^n & dY_{Ly}^n & dZ_{Lz}^n & d\omega_x^n & d\phi_x^n & d\kappa_x^n & dK_{1x}^n & dK_{2x}^n & dK_{3x}^n & dP_{1x}^n & dP_{2x}^n \\ dx'_{ay} & dy'_{ay} & df_y & dX_{Ly}^n & dY_{Lx}^n & dZ_{Ly}^n & d\omega_y^n & d\phi_y^n & d\kappa_y^n & dK_{1y}^n & dK_{2y}^n & dK_{3y}^n & dP_{1y}^n & dP_{2y}^n \end{bmatrix} \quad (3.2.10)$$

mentre i vettori x ed l saranno:

$$x = \begin{bmatrix} x'_a \\ y'_a \\ f \\ X_L \\ Y_L \\ Z_L \\ \omega \\ \phi \\ \kappa \end{bmatrix} \quad l = \begin{bmatrix} J_x^1 \\ K_y^1 \\ J_x^2 \\ K_y^2 \\ \vdots \\ J_x^n \\ K_y^n \end{bmatrix} \quad (3.2.11)$$

3.3 Linearizzazione con stima iterativa

Il sistema risultante dalla linearizzazione delle equazioni di collinearit  (3.2.8) (3.2.9) pu  essere formulato con il modello di Gauss-Markov come segue:

$$e = Ax - l \quad (3.3.1)$$

dove

- e   l'errore reale;

- A é la matrice di disegno prima definita, contenente le derivate parziali delle equazioni (3.2.8) e (3.2.9), valutata con le relative approssimazioni;
- x é il vettore delle incognite;
- l é il vettore assoluto delle osservazioni.

Il sistema (3.3.1) puó essere risolto con il *metodo dei minimi quadrati*.

3.3.1 Metodo dei minimi quadrati

In molti sistemi che usano equazioni é comodo utilizzare procedure sistematiche per formulare equazioni in forma normale. Consideriamo il seguente sistema di m equazioni lineari di uguale peso contenenti n incognite:

$$\left. \begin{array}{rcccccc} a_1A + b_1B + c_1C + \dots + n_1N - l_1 & = & v_1 \\ a_2A + b_2B + c_2C + \dots + n_2N - l_2 & = & v_2 \\ \vdots & & \vdots \\ a_mA + b_mB + c_mC + \dots + n_mN - l_m & = & v_m \end{array} \right\} \quad (3.3.2)$$

Nelle equazioni (3.3.2) le variabili a, b, c sono i coefficienti noti delle incognite A, B, C ; i valori L sono noti; le v , infine, sono i resti. Facendo il quadrato dei resti e sommandoli fra loro, viene creata la funzione $\sum v^2$, che é una misura dell'errore commesso una volta assegnato il valore alle incognite $A \dots N$.

Le equazioni (3.3.2) devono essere rappresentate come una matrice nel seguente modo:

$$Ax = l \quad (3.3.3)$$

dove

$$A = \begin{bmatrix} a_1 & b_1 & c_1 & \cdots & n_1 \\ a_2 & b_2 & c_2 & \cdots & n_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_m & b_m & c_m & \cdots & n_m \end{bmatrix} \quad x = \begin{bmatrix} A \\ B \\ C \\ \vdots \\ N \end{bmatrix}$$

$$l = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ \vdots \\ l_m \end{bmatrix}$$

Applicando il metodo di Gauss-Markov possiamo quindi trovare il vettore delle incognite x . Prima di tutto moltiplichiamo A^T a destra e a sinistra dell'espressione (3.3.3) nel seguente modo

$$A^T A \hat{x} = A^T l \quad (3.3.4)$$

In questa equazione, $(A^T A)$ é la *matrice dei coefficienti* dell'equazione in *forma normale* dei fattori incogniti. Moltiplicando ciascun fattore dell'equazione precedente con $(A^T A)^{-1}$ e riducendo si ottiene:

$$\begin{aligned} (A^T A)^{-1} (A^T A) \hat{x} &= (A^T A)^{-1} A^T l \\ I \hat{x} &= (A^T A)^{-1} A^T l \\ \hat{x} &= (A^T A)^{-1} A^T l \end{aligned} \quad (3.3.5)$$

In questa riduzione, I é la matrice identità. La (3.3.5) é l'equazione matriciale dei quadrati base per osservazioni *egualmente pesate*. Il vettore \hat{x} contiene i valori incogniti delle equazioni di collinearità.

Per un sistema di equazioni *pesate*, all'equazione (3.3.5) si deve aggiungere un funzione matriciale peso P nel seguente modo

$$\hat{x} = (A^T P A)^{-1} A^T P l \quad (3.3.6)$$

dove P é una matrice diagonale rappresentata nel seguente modo:

$$P = \begin{bmatrix} p_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & p_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & p_m \end{bmatrix}$$

In questa matrice, tutti gli elementi fuori la diagonale principale sono zeri.

Quest'ultimo modello é adatto quando le osservazioni individuali sono indipendenti e non correlate.

Ricalcolando la matrice di disegno A ad ogni passo con i nuovi valori di x e procedendo iterativamente fino a convergenza, si trova una nuova stima robusta dei valori di x .

3.4 Controllo sulla soluzione

3.4.1 Analisi di stabilit  del problema numerico

In questa sezione intendiamo studiare la stabilit  della soluzione del sistema lineare $Ax = l$ rispetto alle perturbazioni sui dati A e l . A tale scopo introduciamo il nuovo sistema reale da risolvere

$$(A + \delta A)(x + \delta x) = l + \delta l \quad (3.4.1)$$

che chiameremo *sistema perturbato* in quanto tutti i dati della matrice A sono

stati arrotondati (per via dell'aritmetica finita della macchina). Naturalmente δA é una matrice e δl é un vettore contenente i coefficienti di perturbazione.

Il problema é quello di stabilire come dipende δx da δA e δl . In questa analisi non si tiene conto degli errori di arrotondamento introdotti dal processo di calcolo della soluzione $x + \delta x$.

Sia $\| v \|$ una norma di vettore e $\| A \|$ la norma matriciale introdotta. Per ogni A invertibile definiamo

$$K(A) = \| A \| \| A^{-1} \| \quad (3.4.2)$$

il suo *numero di condizionamento*. Indicando I come la matrice identità notiamo che:

$$I = \| I \| = \| AA^{-1} \| \leq \| A \| \| A^{-1} \| = K(A) \quad \forall A \quad (3.4.3)$$

3.4.2 Numero di condizionamento

In generale $K(A)$ dipende dalla norma scelta: quando questa dipendenza vorrá essere evidenziata useremo una notazione con un pedice, ad esempio $K_\infty(A) = \| A \|_\infty \| A^{-1} \|_\infty$. Piú in generale, $K_p(A)$ denoterá il condizionamento in norma p di A . Casi particolarmente interessanti sono quelli con $p = 1$, $p = 2$ e $p = \infty$.

Al crescere del numero di condizionamento aumenta la sensibilità della soluzione del sistema lineare $Ax = l$ alle perturbazioni nei dati. Cominciamo con l'osservare che $K(A) \geq 1$ in quanto

$$1 = \| AA^{-1} \| \leq \| A \| \| A^{-1} \| = K(A) \quad (3.4.4)$$

Inoltre $K(A^{-1}) = K(A)$ e $\forall \alpha \in C$ con $\alpha \neq 0$, $K(\alpha A) = K(A)$. Infine, se A é ortogonale, $K_2(A) = 1$, in quanto $\| A \|_2 = \sqrt{\rho(A^T A)} = \sqrt{\rho(I)} = 1$ ed essendo

$A^{-1} = A^T$. Assumeremo per convenzione che il numero di condizionamento di una matrice singolare sia infinito.

Per $p = 2$, $k_2(A)$ può essere meglio caratterizzato. Si dimostra infatti che

$$K_2(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\sigma_1(A)}{\sigma_n(A)} \quad (3.4.5)$$

essendo $\sigma_1(A)$ e $\sigma_n(A)$ il massimo ed il minimo valore singolare di A . Di conseguenza nel caso di matrici simmetriche definite positive si ha

$$K_2(A) = \frac{\lambda_{max}}{\lambda_{min}} = \sigma(A)\sigma(A^{-1}) \quad (3.4.6)$$

essendo λ_{max} e λ_{min} il massimo ed il minimo autovalore di A . Per una verifica diretta si osservi che:

$$\|A\|_2 = \sqrt{\sigma(A^T A)} = \sqrt{\sigma(A^2)} = \sqrt{\lambda_{max}^2} = \lambda_{max} \quad (3.4.7)$$

Inoltre, essendo $\lambda(A^{-1}) = \frac{1}{\lambda(A)}$, si ha che $\|A^{-1}\|_2 = \frac{1}{\lambda_{min}}$.

$K_2(A)$ è detto *numero di condizionamento spettrale*.

In generale, se definiamo la distanza relativa di $A \in C^{m \times n}$ dall'insieme delle matrici singolari rispetto alla norma p come

$$dist_p(A) = \min\left\{\frac{\|\delta A\|_p}{\|A\|_p} : A + \delta A \text{ è singolare}\right\} \quad (3.4.8)$$

si può dimostrare che

$$dist_p(A) = \frac{1}{K_p(A)} \quad (3.4.9)$$

La (3.4.9) esprime il fatto che una matrice A con un numero di condizionamento elevato potrebbe comportarsi come una matrice singolare della forma $A + \delta A$. In

tal caso, a perturbazioni nulle del termine noto potrebbero non corrispondere perturbazioni nulle sulla soluzione in quanto, se $A + \delta A$ é singolare, il sistema omogeneo $(A + \delta A)z = 0$ non ammette come unica soluzione quella nulla.

Dalla (3.4.9) discende inoltre che se $A + \delta A$ é non singolare allora

$$\| \delta A \|_p \| A^{-1} \|_p < 1 \quad (3.4.10)$$

Quindi una prima verifica per valutare l'adeguatezza del modello é quella di stimare il condizionamento della matrice di disegno A . Nel caso di sovrapparametrizzazione, infatti, il numero di condizionamento della matrice diventa elevato e la soluzione mal condizionata. In questo caso la soluzione non é accettabile e si dovrà, dunque, eliminare dei parametri.

Numero di condizionamento in MATLAB

Due funzioni MATLAB ci permettono di stabilire il numero di condizionamento di una matrice; sono *cond* ed *rcond*.

La funzione MATLAB *cond* é una sofisticata funzione ed é basata sul metodo della *Singular value decomposition*. La funzione *cond* dovrebbe tornare un valore poco maggiore di 1 per una matrice ben condizionata, un valore molto maggiore di 1 altrimenti.

La funzione *rcond* é un po' meno utile della funzione *cond* ma é piú veloce. La funzione dá un valore tra 0 ed 1, con valori vicini allo zero che stanno a significare matrici mal poste. Il reciproco della funzione *rcond* é, di solito, dello stesso ordine di grandezza di *cond*.

3.4.3 Analisi di varianza

Un'analisi efficace per verificare la convergenza del nostro sistema di equazioni e quindi l'esistenza di una soluzione accettabile, é quella dell'*analisi di varianza*. La

```
>> help cond

COND    Condition number with respect to inversion.
COND(X) returns the 2-norm condition number (the ratio of the
largest singular value of X to the smallest). Large condition
numbers indicate a nearly singular matrix.

COND(X,P) returns the condition number of X in P-norm:

NORM(X,P) * NORM(INV(X),P).

where P = 1, 2, inf, or 'fro'.
```

Figura 3.1: La funzione MATLAB *cond*

```
>> help rcond

RCOND   LAPACK reciprocal condition estimator.
RCOND(X) is an estimate for the reciprocal of the
condition of X in the 1-norm obtained by the LAPACK
condition estimator. If X is well conditioned, RCOND(X)
is near 1.0. If X is badly conditioned, RCOND(X) is
near EPS.
```

Figura 3.2: La funzione MATLAB *rcond*

varianza di una variabile aleatoria x é la misura della diffusione (o dispersione) della densità di x .

Nel nostro caso possiamo effettuare due differenti analisi di varianza:

- analisi di varianza sul residuo.
- analisi di varianza sulla soluzione x .

Analisi di varianza sul residuo

L'equazione matriciale per calcolare il residuo ad ogni passo é la seguente

$$v = A\hat{x} - l = (A(A^T A)^{-1} A^T - I)l \quad (3.4.11)$$

La varianza sul residuo é data da:

$$\sigma_0^2 = \frac{(v^T v)}{r} \quad (3.4.12)$$

mentre la *deviazione standard* é data da

$$\sigma_0 = \sqrt{\frac{(v^T v)}{r}} \quad (3.4.13)$$

Nelle equazioni (3.4.12) e (3.4.13), r é il numero di gradi di libert a ed é uguale al numero di equazioni osservabili meno il numero dei fattori incogniti ($r = (m - n)$).

La varianza, in questo caso, descrive l'ampiezza del residuo. Una varianza bassa mi dice che le variabili sono stimate in modo corretto, mentre una varianza alta mi dice che le variabili non sono stimate in modo sufficientemente accurato.

Analisi di varianza sulla soluzione x

La seconda verifica di varianza é quella sulla soluzione x . Ad ogni passo di calcolo posso verificare la varianza di ogni parametro incognito della soluzione x . Per farlo, bisogna prima definire una nuova matrice, detta *matrice di covarianza*, nel seguente modo:

$$Q_{XX} = (A^T A)^{-1} \quad (3.4.14)$$

La matrice di disegno A , come abbiamo detto, é funzione della soluzione attuale \hat{x} , visto che contiene le derivate parziali calcolate in x . Nella matrice di covarianza, ciascuna riga e colonna i -esima vengono associate con le incognite i -esime. Moltiplicando dunque ciascun elemento diagonale per la varianza del residuo, ottengo la varianza di ogni singolo elemento di x nel seguente modo:

$$\sigma_{0_{ii}}^2 = Q_{XX_{ii}} \sigma_0^2 \quad (3.4.15)$$

Ad ogni passo di calcolo relazionano la varianza di ogni singolo elemento con le variabili ricalcolate. Se il valore ottenuto é minore di un valore assunto a priori, allora la stima del parametro é affidabile. Altrimenti, se il valore ottenuto é maggiore, la stima potrebbe non essere sufficientemente accurata.

3.4.4 Analisi di correlazione

L'*analisi di correlazione* é molto importante in quanto ci permette di definire se la correlazione fra due parametri é elevata e, quindi, di eliminare uno di questi due parametri in quanto potrebbero risultare stimati in modo non sufficientemente accurato.

L'analisi di correlazione si basa sulla matrice di covarianza introdotta nel precedente paragrafo e che qui riscriviamo:

$$Q_{XX} = (A^T A)^{-1} \quad (3.4.16)$$

Per l'analisi di correlazione fra un parametro i ed un parametro j , si introduce il coefficiente di correlazione ρ_{ij} , definito come:

$$\rho_{ij} = \frac{Q_{XX_{ij}}}{\sqrt{Q_{XX_{ii}} Q_{XX_{jj}}}} \quad (3.4.17)$$

il quale, per le proprietá delle matrici normali, soddisfa la condizione che $-1 \leq \rho \leq +1$.

I coefficienti di correlazione non saranno in generale nulli perché le incognite sono stimate indirettamente attraverso la soluzione del sistema normale e quindi, proprio per questa origine comune, non sono indipendenti fra di loro. Dall'espressione di ρ_{ij} risulta in modo molto chiaro che la dipendenza fra le incognite non proviene da errori di misura che possano riflettersi contemporaneamente su alcune di esse: infatti dalla formula (3.4.17) non compare σ_0^2 , il quale dipende dagli errori delle osservazioni, ma solo i Q_{XX} .

Due parametri saranno a rischio se la correlazione fra loro due supera in valore assoluto lo 0.9. Con valori vicini a 0.99 é opportuno eliminarne uno, per evitare di avere soluzioni non corrette.

Capitolo 4

Inizializzazione tramite Direct Linear Transformation

4.1 Introduzione alla DLT

Il metodo della *Direct Linear Transformation* (DLT) é stato sviluppato nel 1971 da *Abdel-Aziz* e *Karara* per risolvere la *condizione di collinearit *.

Avendo a disposizione solamente le coordinate dell'immagine 2D e le coordinate oggetto 3D del sistema bundle adjustment, con pochi e semplici calcoli la DLT dá in uscita un'approssimazione dei parametri di orientamento interno ed esterno che verranno poi usati come prima inizializzazione per il sistema bundle adjustment.

Il principio della DLT é quello di stabilire la relazione tra la coordinate oggetto 3D e le coordinate dell'immagine 2D usando equazioni lineari.

Per questo motivo, il metodo non richiede la computazione delle derivate parziali e la determinazione delle approssimazioni delle variabili incognite nella soluzione, come invece dovrebbe essere richiesto nella soluzione convenzionale.

4.1.1 Il concetto teorico

Il concetto teorico basilare usato in fotogrammetria é che la fotografia, considerata come un piano perfetto, é una proiezione centrale dello spazio oggetto. Implicitamente

in questo concetto sta la condizione di collinearit  del punto immagine, del centro di proiezione e del punto oggetto. La *condizione di collinearit *, come abbiamo gi  visto,   espressa con la seguente relazione:

$$\begin{bmatrix} x' & - & x'_a \\ y' & - & y'_a \\ 0 & - & f \end{bmatrix} = \lambda \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_A & - & X_L \\ Y_A & - & Y_L \\ Z_A & - & Z_L \end{bmatrix} \quad (4.1.1)$$

dove

$x' \ y'$ = coordinate nello spazio immagine del punto A

$x'_a \ y'_a$ = coordinate nello spazio immagine del punto principale a

f = costante della macchina fotografica (focale)

$X_A \ Y_A \ Z_A$ = coordinate nello spazio oggetto del punto oggetto A

$X_L \ Y_L \ Z_L$ = coordinate nello spazio oggetto del centro prospettico L

λ = fattore di scala tra i vettori immagine ed oggetto

M = matrice di rotazione ortogonale 3X3

r_{ij} = componenti della matrice ortogonale di rotazione

Le coordinate immagine x' ed y' sono il risultato di un processo di affinamento con correzione delle coordinate osservate per la distorsione delle lenti, deformazione lineare ed errori di comparazione.

4.1.2 Applicazione della DLT

Senza perdita di generalit , il sistema delle coordinate delle foto dovrebbe essere assunto parallelo al sistema di coordinate del comparatore.

Notiamo, infine, che x', x'_a, y', y'_a nell'equazione (4.1.1) sono le coordinate nel piano immagine e che quindi dovrebbero essere differenti dalle coordinate digitalizzate che utilizziamo quando scattiamo le fotografie da una fotocamera digitale.

Assumendo che l'unità digitale sia differente dall'unità del piano immagine reale, poniamo

$$x' - x'_a = \lambda_x (x + \Delta x - x_0) \quad (4.1.2)$$

$$y' - y'_a = \lambda_y (y + \Delta y - y_0) \quad (4.1.3)$$

dove

$\lambda_x \lambda_y$ = fattori di scala per la conversione da unità digitale a unità reale

$x \ y$ = coordinate del piano immagine in unità digitale

$x_0 \ y_0$ = coordinate dei punti principali in unità digitale

$\Delta_x \ \Delta_y$ = errori sistematici nelle coordinate

Supponendo le distorsioni nulle, possiamo riscrivere le equazioni di collinearità (4.1.1) nel seguente modo:

$$\begin{aligned} x' - x'_a &= \lambda [r_{11}(X_A - X_L) + r_{12}(Y_A - Y_L) + r_{13}(X_A - X_L)] \\ y' - y'_a &= \lambda [r_{21}(X_A - X_L) + r_{22}(Y_A - Y_L) + r_{23}(X_A - X_L)] \\ -f &= \lambda [r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(X_A - X_L)] \end{aligned} \quad (4.1.4)$$

Dalla (4.1.4) otteniamo

$$\lambda = \frac{-f}{r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(X_A - X_L)} \quad (4.1.5)$$

Sostituendo λ dalla (4.1.5) nella (4.1.4) e tenendo presente le (4.1.2) e (4.1.3) otteniamo

$$x - x_0 = -\frac{f}{\lambda_x} \left[\frac{r_{11}(X_A - X_L) + r_{12}(Y_A - Y_L) + r_{13}(Z_A - Z_L)}{r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(Z_A - Z_L)} \right] \quad (4.1.6)$$

$$y - y_0 = -\frac{f}{\lambda_y} \left[\frac{r_{21}(X_A - X_L) + r_{22}(Y_A - Y_L) + r_{23}(Z_A - Z_L)}{r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(Z_A - Z_L)} \right] \quad (4.1.7)$$

Semplificando le equazioni (4.1.6) e (4.1.7) otteniamo

$$x = \frac{L_1 X_A + L_2 Y_A + L_3 Z_A + L_4}{L_9 X_A + L_{10} Y_A + L_{11} Z_A + 1} \quad (4.1.8)$$

$$y = \frac{L_5 X_A + L_6 Y_A + L_7 Z_A + L_8}{L_9 X_A + L_{10} Y_A + L_{11} Z_A + 1} \quad (4.1.9)$$

dove:

$$\begin{aligned} L_1 &= \frac{x_0 r_{31} - d_u r_{11}}{D} \\ L_2 &= \frac{x_0 r_{32} - d_u r_{12}}{D} \\ L_3 &= \frac{x_0 r_{33} - d_u r_{13}}{D} \\ L_4 &= \frac{(d_u r_{11} - x_0 r_{31})X_L + (d_u r_{12} - x_0 r_{32})Y_L + (d_u r_{13} - x_0 r_{33})Z_L}{D} \\ L_5 &= \frac{y_0 r_{31} - d_v r_{21}}{D} \\ L_6 &= \frac{y_0 r_{32} - d_v r_{22}}{D} \\ L_7 &= \frac{y_0 r_{33} - d_v r_{23}}{D} \\ L_8 &= \frac{(d_v r_{21} - y_0 r_{31})X_L + (d_v r_{22} - y_0 r_{32})Y_L + (d_v r_{23} - y_0 r_{33})Z_L}{D} \\ L_9 &= \frac{r_{31}}{D} \\ L_{10} &= \frac{r_{32}}{D} \\ L_{11} &= \frac{r_{33}}{D} \end{aligned} \quad (4.1.10)$$

con

$$\begin{aligned}
 D &= -(r_{31} X_L + r_{32} Y_L + r_{33} Z_L) \\
 d_u &= \frac{\lambda}{\lambda_x} \quad \text{con } \lambda_x \text{ unità di conversione sull'asse } X \\
 d_v &= \frac{\lambda}{\lambda_y} \quad \text{con } \lambda_y \text{ unità di conversione sull'asse } Y
 \end{aligned} \tag{4.1.11}$$

I coefficienti da L_1 a L_{11} sono chiamati *parametri DLT*; essi riflettono la relazione esistente fra lo spazio oggetto ed il piano immagine. Le equazioni (4.1.8) e (4.1.9) sono le equazioni 3D standard per la DLT, ma gli errori causati dalle distorsioni radiali e tangenziali possono anche essere inclusi, presentando il numero dei coefficienti incogniti.

4.2 Calcolo dei parametri di orientamento interni ed esterni

Le equazioni (4.1.8) e (4.1.9) possono essere usate per la *calibrazione della fotocamera* o per la *computazione delle coordinate 3D*:

1. *calibrazione della fotocamera*: le 11 incognite delle equazioni (4.1.8) e (4.1.9) possono essere determinate con il *metodo dei minimi quadrati* usando almeno 6 punti conosciuti in termini delle coordinate oggetto (X_A, Y_A, Z_A) e le corrispondenti coordinate immagine (x, y). Così i parametri *intrinseci* ed *estrinseci* della fotocamera possono essere determinati con equazioni semplici e lineari.
2. *computazione delle coordinate 3D*: una volta che gli undici parametri sono stati computati, le coordinate 3D dei punti immagine possono essere determinate dalle equazioni (4.1.8) e (4.1.9) con almeno due immagini.

L'approccio tramite DLT dá una grossa accuratezza quando i punti usati sono

misurati con grande precisione. È una soluzione diretta che non richiede approssimazioni iniziali per i parametri incogniti (come richiede il *bundle adjustment*). Quindi i risultati ottenuti possono essere usati come approssimazioni iniziali per la incognite per altre soluzioni convenzionali.

Parametri	
$L_1 - L_{11}$	DLT standard
$L_{12} - L_{14}$	Termini di distorsione del 3°, 5° e 7° ordine
$L_{15} - L_{16}$	Termini di distorsione decentrica

Tabella 4.1: I parametri stimati tramite DLT

Nel nostro studio useremo il metodo della DLT in forma normale, includendo solo gli 11 parametri standard e quindi escludendo le distorsioni radiali e tangenziali. Cominciamo quindi a vedere le equazioni che ci servono per calcolare i fattori che ci interessano.

4.2.1 Gli undici coefficienti della DLT

Riscrivendo le equazioni (4.1.8) e (4.1.9) possiamo ottenere:

$$x = L_1 X_A + L_2 Y_A + L_3 Z_A + L_4 - L_9 x X_A - L_{10} x Y_A - L_{11} x Z_A \quad (4.2.1)$$

$$y = L_5 X_A + L_6 Y_A + L_7 Z_A + L_8 - L_9 y X_A - L_{10} y Y_A - L_{11} y Z_A \quad (4.2.2)$$

Le equazioni (4.2.1) e (4.2.2) sopra trovate sono equivalenti a:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} X_A & Y_A & Z_A & 1 & 0 & 0 & 0 & 0 & -xX_A & -xY_A & -xZ_A \\ 0 & 0 & 0 & 0 & X_A & Y_A & Z_A & 1 & -yX_A & -yY_A & -yZ_A \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \\ L_7 \\ L_8 \\ L_9 \\ L_{10} \\ L_{11} \end{bmatrix} \quad (4.2.3)$$

Espandendo l'equazione (4.2.3) per n punti di controllo otteniamo:

$$\begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{bmatrix} = \begin{bmatrix} X_A^1 & Y_A^1 & Z_A^1 & 1 & 0 & 0 & 0 & 0 & -x^1 X_A^1 & -x^1 Y_A^1 & -x^1 Z_A^1 \\ 0 & 0 & 0 & 0 & X_A^1 & Y_A^1 & Z_A^1 & 1 & -y^1 X_A^1 & -y^1 Y_A^1 & -y^1 Z_A^1 \\ \vdots & \vdots \\ X_A^n & Y_A^n & Z_A^n & 1 & 0 & 0 & 0 & 0 & -x^n X_A^n & -x^n Y_A^n & -x^n Z_A^n \\ 0 & 0 & 0 & 0 & X_A^n & Y_A^n & Z_A^n & 1 & -y^n X_A^n & -y^n Y_A^n & -y^n Z_A^n \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \\ L_7 \\ L_8 \\ L_9 \\ L_{10} \\ L_{11} \end{bmatrix} \quad (4.2.4)$$

Nell'equazione (4.2.4) abbiamo assunto che sia le coordinate immagine misurate (x^n, y^n) sia le coordinate dello spazio oggetto (X_A^n, Y_A^n, Z_A^n) siano tutte conosciute. Un gruppo di *punti di controllo* in cui sia le coordinate x ed y sia le coordinate X_A , Y_A e Z_A sono conosciute dovrebbero essere noti. In altre parole, i punti di controllo dovrebbero formare un volume detto *volume di controllo*.

I punti di controllo sono tipicamente fissati in un frame di calibrazione (od *oggetto di controllo*).

L'equazione (4.2.4) ha come sua forma base la seguente espressione:

$$X L = Y \quad (4.2.5)$$

I parametri della DLT possono essere calcolati usando il metodo dei minimi quadrati:

$$\begin{aligned} X L &= Y \\ (X' X) L &= (X' Y) \\ (X' X)^{-1} (X' X) L &= (X' X)^{-1} (X' Y) \\ L &= (X' X)^{-1} (X' Y) \end{aligned}$$

Per calcolare i parametri della DLT ed i parametri addizionali usando il metodo dei minimi quadrati, l'equazione (4.2.4) dovrebbe essere sovra-determinata (numero delle equazioni > numero delle incognite).

Numero di parametri	Numero minimo di punti di controllo
11 - DLT standard	6
12 - DLT con distorsione del 3° ordine	6
13 - DLT con distorsione del 4° ordine	7
14 - DLT con distorsione del 5° ordine	7
16 - DLT con distorsione decentrica	8

Tabella 4.2: Numero minimo di punti di controllo per DLT

Sapendo che ciascun punto di controllo permette di calcolare due equazioni, il numero minimo dei punti di controllo richiesti é indicato nella tabella (4.2).

4.2.2 Calcolo delle coordinate del centro ottico: (X_L, Y_L, Z_L)

Dalle equazioni (4.1.8) e (4.1.9) possiamo trovare:

$$-L_4 = L_1 X_L + L_2 Y_L + L_3 Z_L$$

$$-L_8 = L_5 X_L + L_6 Y_L + L_7 Z_L$$

$$-1 = L_9 X_L + L_{10} Y_L + L_{11} Z_L$$

in forma matriciale

$$\begin{bmatrix} L_1 & L_2 & L_3 \\ L_5 & L_6 & L_7 \\ L_9 & L_{10} & L_{11} \end{bmatrix} \begin{bmatrix} X_L \\ Y_L \\ Z_L \end{bmatrix} = \begin{bmatrix} -L_4 \\ -L_8 \\ -1 \end{bmatrix} \quad (4.2.6)$$

Invertendo la (4.2.6) si trova che:

$$\begin{bmatrix} X_L \\ Y_L \\ Z_L \end{bmatrix} = \begin{bmatrix} L_1 & L_2 & L_3 \\ L_5 & L_6 & L_7 \\ L_9 & L_{10} & L_{11} \end{bmatrix}^{-1} \begin{bmatrix} -L_4 \\ -L_8 \\ -1 \end{bmatrix} \quad (4.2.7)$$

4.2.3 Calcolo degli elementi di orientamento interno: (x_0, y_0, f)

Dalle equazioni (4.1.8) e (4.1.9) questi elementi possono essere ottenuti nel seguente modo:

$$x_0 = \frac{L_1 L_9 + L_2 L_{10} + L_3 L_{11}}{L_9^2 + L_{10}^2 + L_{11}^2} \quad (4.2.8)$$

$$y_0 = \frac{L_5 L_9 + L_6 L_{10} + L_7 L_{11}}{L_9^2 + L_{10}^2 + L_{11}^2} \quad (4.2.9)$$

$$f = \frac{(f_x + f_y)}{2} \quad (4.2.10)$$

dove

$$\begin{aligned}
f_x &= \sqrt{-x_0^2 + \frac{L_1^2 + L_2^2 + L_3^2}{D}} \\
f_y &= \sqrt{-y_0^2 + \frac{L_5^2 + L_6^2 + L_7^2}{D}} \\
D &= L_9^2 + L_{10}^2 + L_{11}^2
\end{aligned} \tag{4.2.11}$$

4.2.4 Calcolo degli angoli di rotazione: (ω, ϕ, κ)

Calcolo dei coefficienti di rotazione

Dall'equazione (4.1.10) possiamo trovare i *coefficienti* di rotazione nel seguente modo:

$$T_O = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{4.2.12}$$

dove

$$\begin{aligned}
r_{11} &= \frac{D(x_0 L_9 - L_1)}{d_u} \\
r_{12} &= \frac{D(x_0 L_{10} - L_2)}{d_u} \\
r_{13} &= \frac{D(x_0 L_{11} - L_3)}{d_u} \\
r_{21} &= \frac{D(y_0 L_9 - L_5)}{d_v} \\
r_{22} &= \frac{D(y_0 L_{10} - L_6)}{d_v} \\
r_{23} &= \frac{D(y_0 L_{11} - L_7)}{d_v} \\
r_{31} &= D L_9 \\
r_{32} &= D L_{10} \\
r_{33} &= D L_{11}
\end{aligned} \tag{4.2.13}$$

Per calcolare la *matrice di rotazione* (4.2.12) dobbiamo quindi calcolare d_u e d_v .

Dalla (4.2.13) troviamo che

$$\begin{aligned}
d_u &= \sqrt{D^2 ((x_0 L_9 - L_1)^2 + (x_0 L_{10} - L_2)^2 + (x_0 L_{11} - L_3)^2)} \\
d_v &= \sqrt{D^2 ((y_0 L_9 - L_5)^2 + (y_0 L_{10} - L_6)^2 + (y_0 L_{11} - L_7)^2)} \\
D &= \sqrt{\frac{1}{L_9^2 + L_{10}^2 + L_{11}^2}}
\end{aligned}$$

La correttezza dei risultati del calcolo degli r_{ij} può essere verificata controllando che:

$$\begin{aligned}
r_{11}^2 + r_{12}^2 + r_{13}^2 &= 1 = \frac{D^2 ((x_0 L_9 - L_1)^2 + (x_0 L_{10} - L_2)^2 + (x_0 L_{11} - L_3)^2)}{d_u^2} \\
r_{21}^2 + r_{22}^2 + r_{23}^2 &= 1 = \frac{D^2 ((y_0 L_9 - L_5)^2 + (y_0 L_{10} - L_6)^2 + (y_0 L_{11} - L_7)^2)}{d_v^2}
\end{aligned} \tag{4.2.14}$$

Calcolo degli angoli di rotazione

I tre angoli di Eulero (ω , ϕ , κ) possono essere descritti dai nove *coseni direttori*.

Prima di tutto, come abbiamo già visto nella (4.2.12), ricordiamo che la matrice di rotazione R é computata nel seguente modo:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{4.2.15}$$

con

$$R = \begin{bmatrix} \cos\phi \cos\kappa & \sin\omega \sin\phi \cos\kappa + \cos\omega \sin\kappa & -\cos\omega \sin\phi \cos\kappa + \sin\omega \sin\kappa \\ -\cos\phi \sin\kappa & -\sin\omega \sin\phi \sin\kappa + \cos\omega \cos\kappa & \cos\omega \sin\phi \sin\kappa + \sin\omega \cos\kappa \\ \sin\phi & -\sin\omega \cos\phi & \cos\omega \cos\phi \end{bmatrix}$$

Possiamo quindi trovare gli angoli nel seguente modo:

$$\phi = \begin{cases} \sin^{-1}(-r_{31}) & (\sin^{-1}(-r_{31}) > -\frac{\pi}{2}) \text{ e } (\sin^{-1}(-r_{31}) < \frac{\pi}{2}) \\ \pi - \sin^{-1}(-r_{31}) & (\pi - \sin^{-1}(-r_{31}) > -\frac{\pi}{2}) \text{ e } (\pi - \sin^{-1}(-r_{31}) < \frac{\pi}{2}) \end{cases} \quad (4.2.16)$$

e, se $\phi \neq \pm \frac{\pi}{2}$

$$\omega = \begin{cases} \tan^{-1}\left[-\frac{r_{32}}{r_{33}}\right] & (r_{33} \cos \phi < 0) \\ \tan^{-1}\left[-\frac{r_{32}}{r_{33}}\right] + \pi & (r_{33} \cos \phi > 0) \end{cases} \quad (4.2.17)$$

$$\kappa = \begin{cases} \tan^{-1}\left[-\frac{r_{21}}{r_{11}}\right] & (r_{11} \cos \phi < 0) \\ \tan^{-1}\left[-\frac{r_{21}}{r_{11}}\right] + \pi & (r_{11} \cos \phi > 0) \end{cases} \quad (4.2.18)$$

Dalle equazioni (4.2.16), (4.2.17) e (4.2.18) si possono ottenere due differenti insiemi di angoli di orientamento opposti; si deve fare attenzione soprattutto al calcolo dell'angolo ω per scegliere l'insieme giusto degli angoli. In molti casi si potrebbe assumere che $-\frac{\pi}{2} < \omega < \frac{\pi}{2}$.

Un caso speciale potrebbe capitare quando il $\cos(\phi) = 0$. Se capitasse, non si potrebbero computare gli angoli ω e κ in modo separato, ma bensì si potrebbero computare $\omega + \kappa$ e $\omega - \kappa$ nel seguente modo:

$$\omega \pm \kappa = \begin{cases} \tan^{-1}\left[+\frac{r_{23}}{r_{22}}\right] & (r_{22} > 0) \\ \tan^{-1}\left[-\frac{r_{23}}{r_{22}}\right] + \pi & (r_{22} < 0) \end{cases} \quad (4.2.19)$$

Queste condizioni sono chiamate *gimbal locks*.

Capitolo 5

Ricostruzione tridimensionale

5.1 Introduzione

Una volta calibrato un sistema composto da almeno due fotocamere, si può procedere alla ricostruzione tridimensionale dei punti. Prima di fare ciò, comunque, sarà necessario correggere le eventuali distorsioni presenti nella misura delle coordinate oggetto. Successivamente verranno ricostruiti i punti tridimensionali con l'algoritmo di *ray intersection*.

5.2 Correzione delle distorsioni

Le eventuali distorsioni radiali e tangenziali presenti possono influire notevolmente sulla ricostruzione tridimensionale dei punti, alterando in modo sostanziale il suo corretto valore. Se avessimo a che fare con un sistema ottico ideale, i punti misurati sull'immagine $P_{dis} = (x_{dis}, y_{dis})$ potrebbero essere utilizzati direttamente per la ricostruzione tridimensionale; nella realtà, ogni sistema ottico reale introduce delle *distorsioni*, modellizzate da una funzione di distorsione D :

$$P_{dis} = D(P_{non\ dis}) \tag{5.2.1}$$

con

$$P_{non\ dis} = (x_{non\ dis}, y_{non\ dis}) \quad (5.2.2)$$

Per ricondursi alla situazione ideale che permette di utilizzare l'algoritmo di *ray intersection* per il calcolo delle coordinate tridimensionali, dunque, é necessario calcolare $P_{non\ dis}$ a partire da P_{dis} e dai parametri di distorsione calcolati tramite il bundle adjustment. La funzione di distorsione D non é però in genere invertibile; non é perciò possibile calcolare $P_{non\ dis}$ semplicemente come:

$$P_{non\ dis} = D^{-1}(P_{non\ dis}) \quad (5.2.3)$$

Per stimare la posizione del punto non distorto, $P_{non\ dis}$, a partire da quello del punto distorto, P_{dis} , é necessario usare tecniche piú raffinate.

Nel nostro caso, abbiamo definito un errore e pari a:

$$e = \| P_{dis} - D(\hat{P}_{non\ dis}) \|^2 \quad (5.2.4)$$

dove $\hat{P}_{non\ dis}$ é la stima della posizione del punto non distorto. La minimizzazione di e rispetto a $\hat{P}_{non\ dis}$ permette di ottenere la stima delle coordinate del punto non distorto, $\hat{P}_{non\ dis} \sim P_{non\ dis}$.

Per minimizzare e é possibile utilizzare il *metodo del gradiente*, o *steepest descent*.

Sia $\hat{P}_{non\ dis}$ la stima delle coordinate del punto non distorto all'iterazione n -esima. Si calcolino allora $\frac{\partial e}{\partial \hat{x}_{non\ dis}}$ e $\frac{\partial e}{\partial \hat{y}_{non\ dis}}$, utilizzando i valori di $\hat{P}_{non\ dis}$. Il gradiente

$$\vec{\Phi} = \left[\frac{\partial e}{\partial \hat{x}_{non\ dis}}, \frac{\partial e}{\partial \hat{y}_{non\ dis}} \right] \quad (5.2.5)$$

calcolato in $\hat{P}_{non\ dis}$, dá la direzione di massima crescita dell'errore. Al contrario, muovendosi lungo la direzione opposta al gradiente, $-\vec{\Phi}$, l'errore e viene localmente minimizzato.

Il metodo del gradiente dá dunque luogo al seguente algoritmo.

Sia $\hat{P}_{non\ dis}$ la stima delle coordinate di $P_{non\ dis}$ all'iterazione n-esima

1. si calcoli $\vec{\Phi} = \left[\frac{\partial e}{\partial \hat{x}_{non\ dis}}, \frac{\partial e}{\partial \hat{y}_{non\ dis}} \right]$;
2. si aggiorni la stima di $\hat{P}_{non\ dis}$ nel seguente modo:

$$\hat{P}_{non\ dis}^{n+1} = \hat{P}_{non\ dis}^n - \eta \vec{\Phi} \quad (5.2.6)$$

dove η determina l'entità dell'aggiornamento.

Il metodo itera fino a quando $\|\vec{\Phi}\| \rightarrow 0$, cioè quando ci si trova in un minimo locale. Al primo passo si pone $\hat{P}_{non\ dis} = P_{dis}$.

In genere, nel nostro caso sono sufficienti 5 o 6 passi per raggiungere la convergenza.

5.3 La ricostruzione tridimensionale

Esistono diversi tipi di algoritmi di ricostruzione che, a seconda del numero di parametri delle fotocamere note a priori, riescono a ricostruire la scena tridimensionale in modo più o meno accurato.

Si possono distinguere diversi tipi di ricostruzione in base al numero di parametri noti:

Parametri conosciuti	Ricostruzione 3D
Parametri intrinseci ed estrinseci	Coordinate 3D assolute
Parametri intrinseci	Coordinate 3D a meno di un fattore di scala
nessuno	Coordinate 3D a meno di una trasformazione proiettiva

Tabella 5.1: Tipologie di algoritmi di ricostruzione 3D

L'algoritmo di ricostruzione utilizzato nel corso di questa tesi fa parte del primo gruppo tra quelli indicati nella tabella (5.1): é un algoritmo di *ray-intersection*, cioè un algoritmo che ricostruisce le coordinate per triangolazione.

5.3.1 Algoritmo di ray-intersection

L'algoritmo di ricostruzione per triangolazione necessita di essere a conoscenza dei parametri intrinseci ed estrinseci delle fotocamere digitali.

Questo permette di ricostruire le coordinate tridimensionali di un oggetto in un sistema di piú telecamere. La ricostruzione avviene intersecando due o piú rette che passano per i punti sui piani immagine ed i centri ottici delle relative fotocamere.

In realtà le coordinate dei punti sui piani immagine sono sempre affette da errori di misurazione e questo implica che le rette non si intersechino. La presenza di *rette sghembe* ha reso necessario l'utilizzo di un algoritmo per la minimizzazione della distanza del punto 3D dalle due rette.

Coseni direttori per una retta orientata

Una retta passante per due punti può essere descritta in forma parametrica, specificando un punto appartenente alla retta e la sua direzione nello spazio tramite i *coseni direttori*.

I *coseni direttori* sono i coseni dei tre angoli α , β e γ che la retta orientata forma con gli assi x , y e z del sistema di riferimento ortogonale nella quale la retta viene rappresentata. Essi possono essere calcolati a partire di una qualsiasi coppia di punti sulla retta.

In questo caso si prenderanno i punti in modo tale che il primo coincida con il centro ottico C della fotocamera ($C = [C_x, C_y, C_z]$) e come secondo il punto p ($p = [p_x, p_y, p_z]$) che rappresenta la proiezione del punto nello spazio P sul piano immagine della fotocamera.

Quindi i coseni direttori saranno dati da:

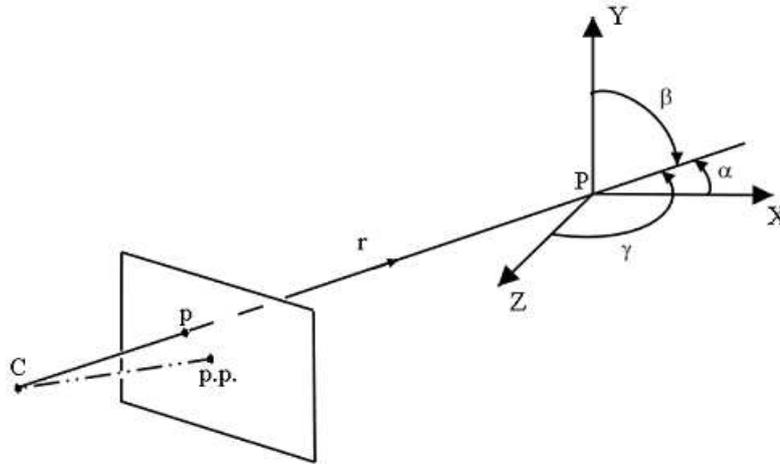


Figura 5.1: Coseni direttori di una retta orientata

$$\begin{aligned}\cos \alpha &= \frac{(p_x - C_x)}{\rho} \\ \cos \beta &= \frac{(p_y - C_y)}{\rho} \\ \cos \gamma &= \frac{(p_z - C_z)}{\rho}\end{aligned}\tag{5.3.1}$$

dove ρ viene definita come la *distanza euclidea* tra i due punti:

$$\rho = \sqrt{(p_x - C_x)^2 + (p_y - C_y)^2 + (p_z - C_z)^2}\tag{5.3.2}$$

Inoltre i coseni direttori sono legati dalla relazione:

$$\cos^2 \alpha + \cos^2 \beta + \cos^2 \gamma = 1\tag{5.3.3}$$

e questo significa che i coseni direttori non possono essere contemporaneamente tutti nulli.

Forma parametrica di una retta

Utilizzando la forma parametrica di una retta é possibile indicare le coordinate di un punto generico sulla stessa retta:

$$\begin{aligned} P_x &= C_x + h \cos \alpha \\ P_y &= C_y + h \cos \beta \\ P_z &= C_z + h \cos \gamma \end{aligned} \quad (5.3.4)$$

dove il modulo del parametro reale h rappresenta la distanza del punto P con le coordinate (P_x, P_y, P_z) dal punto C .

Le formule (5.3.4) in forma matriciale possono essere rappresentate nel seguente modo:

$$P = C + hL \quad (5.3.5)$$

con

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad C = \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} \quad L = \begin{bmatrix} \cos \alpha \\ \cos \beta \\ \cos \gamma \end{bmatrix} \quad (5.3.6)$$

Intersezione tra le rette

Consideriamo

- P_1 e P_2 le proiezioni del punto P sui piani di due fotocamere che osservano la scena;
- C_1 e C_2 i rispettivi centri ottici;
- L_1 e L_2 i coseni direttori delle due rette;

le forme parametriche delle rette sono date da:

$$P_1 = C_1 + h_1 L_1 \quad (5.3.7)$$

$$P_2 = C_2 + h_2 L_2$$

Nello spazio tridimensionale euclideo, se due rette non sono parallele ed esiste un punto di intersezione, questo punto é unico e la condizione di intersezione risulta essere:

$$C_1 + h_1 L_1 = C_2 + h_2 L_2 \quad (5.3.8)$$

Le equazioni (5.3.7) e (5.3.8) formano un sistema di tre equazioni con incognite h_1 e h_2 .

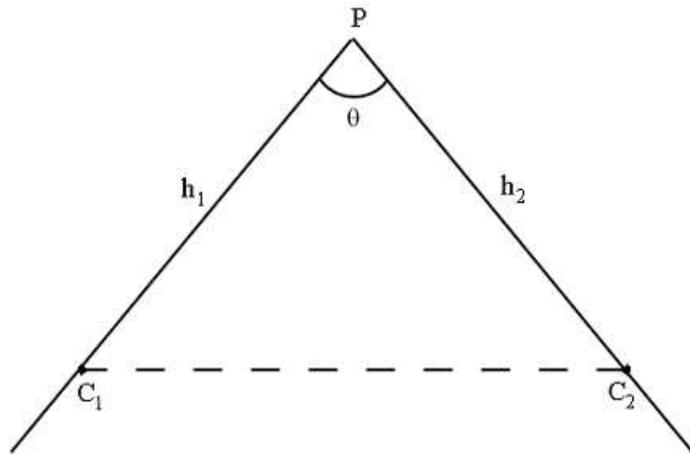


Figura 5.2: Condizione di intersezione

Consideriamo il triangolo C_1PC_2 formato dai segmenti che uniscono i due centri ottici C_1 e C_2 al punto P e dal segmento che unisce i centri ottici stessi.

Applicando il teorema di Carnot troviamo che:

$$\overline{C_1C_2}^2 = \overline{C_1P}^2 + \overline{C_2P}^2 - 2\overline{C_1P}\overline{C_2P}\cos\theta \quad (5.3.9)$$

Riscrivendo l'equazione (5.3.8) nelle sue componenti cartesiane troviamo:

$$\begin{aligned} C_{1x} - C_{2x} &= h_2 \cos \alpha_2 - h_1 \cos \alpha_1 \\ C_{1y} - C_{2y} &= h_2 \cos \beta_2 - h_1 \cos \beta_1 \\ C_{1z} - C_{2z} &= h_2 \cos \gamma_2 - h_1 \cos \gamma_1 \end{aligned} \quad (5.3.10)$$

Elevando al quadrato le equazioni (5.3.10) e sommandole membro a membro si ottiene:

$$(C_{1x} - C_{2x})^2 + (C_{1y} - C_{2y})^2 + (C_{1z} - C_{2z})^2 = \quad (5.3.11)$$

$$h_1^2 + h_2^2 - 2h_1h_2(\cos \alpha_1 \cos \alpha_2 + \cos \beta_1 \cos \beta_2 + \cos \gamma_1 \cos \gamma_2) \quad (5.3.12)$$

Ma:

$$(C_{1x} - C_{2x})^2 + (C_{1y} - C_{2y})^2 + (C_{1z} - C_{2z})^2 = \overline{C_1 C_2}^2 \quad (5.3.13)$$

e

$$\cos \alpha_1 \cos \alpha_2 + \cos \beta_1 \cos \beta_2 + \cos \gamma_1 \cos \gamma_2 = \cos \theta \quad (5.3.14)$$

Quindi la condizione di intersezione é data da:

$$\overline{C_1 C_2}^2 = h_1^2 + h_2^2 - 2h_1h_2 \cos \theta \quad (5.3.15)$$

Distanza minima tra due rette

In realtà le condizioni di intersezione possono non essere soddisfatte a causa sia di distorsioni introdotte dalle lenti ottiche che della risoluzione limitata dei sensori CCD delle fotocamere. Si sceglierá come punto 3D, quindi, il punto che minimizza la distanza tra le due rette nel caso queste non si intersechino.

Prendendo due punti P_1 e P_2 appartenenti alle due rette r ed s indichiamo con D la distanza tra i due punti. Nel caso le due rette non si intersechino, questa distanza non si annullerá per nessuna coppia di punti.

Consideriamo:

$$\Delta_x = \text{distanza tra } P_1 \text{ e } P_2 \text{ lungo } x$$

$$\Delta_y = \text{distanza tra } P_1 \text{ e } P_2 \text{ lungo } y$$

$$\Delta_z = \text{distanza tra } P_1 \text{ e } P_2 \text{ lungo } z$$

La distanza nello spazio si puó calcolare applicando il teorema di Pitagora:

$$D = \sqrt{\Delta_x^2 + \Delta_y^2 + \Delta_z^2} \quad (5.3.16)$$

Riscrivendo le equazioni dei punti P_1 e P_2 situati rispettivamente sulle rette r ed s usando la forma in (5.3.4) troviamo:

$$P_1 = C_1 + \rho_1 L_1 \quad (5.3.17)$$

$$P_2 = C_2 + \rho_2 L_2 \quad (5.3.18)$$

con:

$$L_1 = \begin{bmatrix} \cos \alpha_1 \\ \cos \beta_1 \\ \cos \gamma_1 \end{bmatrix} \quad (5.3.19)$$

$$L_2 = \begin{bmatrix} \cos \alpha_2 \\ \cos \beta_2 \\ \cos \gamma_2 \end{bmatrix} \quad (5.3.20)$$

Sottraendo la (5.3.17) dalla (5.3.18) abbiamo:

$$\Delta = P_1 - P_2 = C_1 + \rho_1 L_1 - C_2 - \rho_2 L_2 \quad (5.3.21)$$

con

$$\Delta = \begin{bmatrix} \Delta_x \\ \Delta_y \\ \Delta_z \end{bmatrix} \quad (5.3.22)$$

che rappresenta un sistema di tre equazioni e cinque incognite ($\Delta_x, \Delta_y, \Delta_z, \rho_1, \rho_2$).

Esiste una sola coppia di punti per cui la distanza D é minima, tranne quando le due rette sono parallele. Il problema consiste nel fatto di ricercare una distanza D minima espressa dalla (5.3.16) e quindi minimizzare una funzione non lineare.

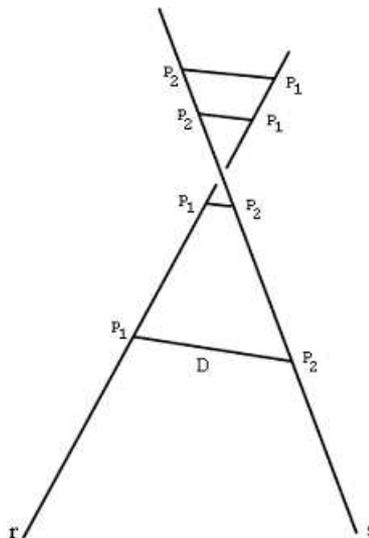


Figura 5.3: Distanza di due rette sghembe

Problema di minimizzazione della distanza

Conviene non considerare la radice quadrata visto che non é una grandezza lineare e non influisce sul risultato finale. Si tratterá quindi di determinare il minimo del quadrato della distanza:

$$D^2 = \Delta x^2 + \Delta y^2 + \Delta z^2 \quad (5.3.23)$$

Ci sono diversi metodi per risolvere questo problema, ma il migliore é risolvere direttamente il sistema lineare di cinque equazioni in cinque incognite:

$$\begin{cases} C_{1x} - C_{2x} + \rho_1 \cos \alpha_1 - \rho_2 \cos \alpha_2 = \Delta x \\ C_{1y} - C_{2y} + \rho_1 \cos \beta_1 - \rho_2 \cos \beta_2 = \Delta y \\ C_{1z} - C_{2z} + \rho_1 \cos \gamma_1 - \rho_2 \cos \gamma_2 = \Delta z \\ \frac{dD^2}{d\rho_1} = 0 \\ \frac{dD^2}{d\rho_2} = 0 \end{cases}$$

Sia le equazioni sui vincoli che le derivate di D^2 rispetto a ρ_1 e ρ_2 sono lineari.

Prima si calcolano le espresioni delle derivate di D^2 esplicitando i termini Δx^2 , Δy^2 , Δz^2 :

$$\begin{aligned} \left[\begin{aligned} \Delta x^2 &= [C_{1x} - C_{2x} + \rho_1 \cos \alpha_1 - \rho_2 \cos \alpha_2]^2 = \\ &= (C_{1x} - C_{2x})^2 + 2(C_{1x} - C_{2x})\rho_1 \cos \alpha_1 + \\ &\quad - 2(C_{1x} - C_{2x})\rho_2 \cos \alpha_2 - 2\rho_1\rho_2 \cos \alpha_1 \cos \alpha_2 \\ &\quad + \rho_1^2 \cos^2 \alpha_1 + \rho_2^2 \cos^2 \alpha_2 \end{aligned} \right. \\ \left[\begin{aligned} \Delta y^2 &= [C_{1y} - C_{2y} + \rho_1 \cos \beta_1 - \rho_2 \cos \beta_2]^2 = \\ &= (C_{1y} - C_{2y})^2 + 2(C_{1y} - C_{2y})\rho_1 \cos \beta_1 + \\ &\quad - 2(C_{1y} - C_{2y})\rho_2 \cos \beta_2 - 2\rho_1\rho_2 \cos \beta_1 \cos \beta_2 + \\ &\quad + \rho_1^2 \cos^2 \beta_1 + \rho_2^2 \cos^2 \beta_2 \end{aligned} \right. \\ \left[\begin{aligned} \Delta z^2 &= [C_{1z} - C_{2z} + \rho_1 \cos \gamma_1 - \rho_2 \cos \gamma_2]^2 = \\ &= (C_{1z} - C_{2z})^2 + 2(C_{1z} - C_{2z})\rho_1 \cos \gamma_1 + \\ &\quad - 2(C_{1z} - C_{2z})\rho_2 \cos \gamma_2 - 2\rho_1\rho_2 \cos \gamma_1 \cos \gamma_2 + \\ &\quad + \rho_1^2 \cos^2 \gamma_1 + \rho_2^2 \cos^2 \gamma_2 \end{aligned} \right. \end{aligned}$$

Ponendo:

$$D_{12}^2 = (C_{1x} - C_{2x})^2 + (C_{1y} - C_{2y})^2 + (C_{1z} - C_{2z})^2 \quad (5.3.24)$$

otteniamo:

$$\begin{aligned}
D^2 = & D_{12}^2 + (\rho_1 \cos \alpha_1 - \rho_2 \cos \alpha_2)^2 + \\
& + (\rho_1 \cos \beta_1 - \rho_2 \cos \beta_2)^2 + (\rho_1 \cos \gamma_1 - \rho_2 \cos \gamma_2)^2 + \\
& + 2(C_{1x} - C_{2x})\rho_1 \cos \alpha_1 - 2(C_{1x} - C_{2x})\rho_2 \cos \alpha_2 + \\
& + 2(C_{1y} - C_{2y})\rho_1 \cos \beta_1 - 2(C_{1y} - C_{2y})\rho_2 \cos \beta_2 + \\
& + 2(C_{1z} - C_{2z})\rho_1 \cos \gamma_1 - 2(C_{1z} - C_{2z})\rho_2 \cos \gamma_2
\end{aligned} \tag{5.3.25}$$

Derivando la (5.3.25) rispetto a ρ_1 e ρ_2 si ottiene:

$$\begin{aligned}
\frac{dD^2}{d\rho_1} = & 2\rho_1 - 2\rho_2(\cos \alpha_1 \cos \alpha_2 + \cos \beta_1 \cos \beta_2 + \cos \gamma_1 \cos \gamma_2) + \\
& + 2(C_{1x} - C_{2x}) \cos \alpha_1 + 2(C_{1y} - C_{2y}) \cos \beta_1 + \\
& + 2(C_{1z} - C_{2z}) \cos \gamma_1 \\
\frac{dD^2}{d\rho_2} = & 2\rho_2 - 2\rho_1(\cos \alpha_1 \cos \alpha_2 + \cos \beta_1 \cos \beta_2 + \cos \gamma_1 \cos \gamma_2) + \\
& + 2(C_{1x} - C_{2x}) \cos \alpha_2 - 2(C_{1y} - C_{2y}) \cos \beta_2 + (4.34) \\
& + 2(C_{1z} - C_{2z}) \cos \gamma_2
\end{aligned}$$

Riscrivendo il sistema lineare in forma matriciale si ha:

$$A\hat{x} + B = 0 \tag{5.3.26}$$

con:

$$x = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \rho_1 \\ \rho_2 \end{bmatrix} \tag{5.3.27}$$

$$A = \begin{bmatrix} 1 & 0 & 0 & -\cos \alpha_1 & \cos \alpha_2 \\ 0 & 1 & 0 & -\cos \beta_1 & \cos \beta_2 \\ 0 & 0 & 1 & -\cos \gamma_1 & \cos \gamma_2 \\ 0 & 0 & 0 & 2 & -2\hat{f} \\ 0 & 0 & 0 & -2\hat{f} & 2 \end{bmatrix} \quad (5.3.28)$$

$$B = \begin{bmatrix} -(C_{1x} - C_{2x}) \\ -(C_{1y} - C_{2y}) \\ 2[(C_{1x} - C_{2x}) \cos \alpha_1 + 2(C_{1y} - C_{2y}) \cos \beta_1 + 2(C_{1z} - C_{2z}) \cos \gamma_1] \\ -2[(C_{1x} - C_{2x}) \cos \alpha_2 + 2(C_{1y} - C_{2y}) \cos \beta_2 + 2(C_{1z} - C_{2z}) \cos \gamma_2] \end{bmatrix} \quad (5.3.29)$$

con

$$\hat{f} = \cos \alpha_1 \cos \alpha_2 + \cos \beta_1 \cos \beta_2 + \cos \gamma_1 \cos \gamma_2 \quad (5.3.30)$$

Affinché il sistema sia risolvibile deve risultare che:

$$\det(A) \neq 0 \quad (5.3.31)$$

Nel nostro caso il $\det(A) = 4(1 - f^2)$, e questo si annulla quando $f = \pm 1$.

La f in (5.3.30) é espressa come prodotto scalare dei coseni direttori delle rette r ed s e $f = \pm 1$ quando l'angolo compreso tra le due rette θ é un multiplo di π (180°), cioè quando le due rette sono parallele.

Calcolo delle coordinate tridimensionali del punto a minima distanza

Una volta trovati Δx , Δy , Δz , ρ_1 , ρ_2 , é possibile trovare il punto che minimizza la distanza tra le due rette \hat{r} ed \hat{s} .

Le coordinate possono essere determinate calcolando:

$$P = C_1 + \rho_1 L_1 + \frac{1}{2} \Delta \quad (5.3.32)$$

oppure

$$P = C_2 + \rho_2 L_2 - \frac{1}{2} \Delta \quad (5.3.33)$$

dove

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (5.3.34)$$

é il punto tridimensionale cosí ricostruito.

Capitolo 6

Risultati

6.1 Risultati sperimentali

Vengono ora presentati alcuni risultati ottenuti con i programmi sviluppati. Nella prima prova viene calibrata una coppia di fotocamere con dati simulati mentre nella seconda prova viene calibrata una coppia di fotocamere con dati presi da immagini reali. Per l'acquisizione delle immagini é stata utilizzata una fotocamera digitale, di cui viene data una descrizione dettagliata.

6.1.1 Fotocamera FinePix

La fotocamera digitale utilizzata é una *FinePix 4900Zoom* prodotta dalle industrie Fujifilm.

Le caratteristiche principali di questa fotocamera sono:

- *Zoom a 6X*: La FinePix 4900Zoom utilizza uno zoom ottico 6X EBC con lenti a largo diametro. La sua lunghezza focale é equivalente ad un 35-210 mm di una fotocamera 35 mm. Le lenti sono professionali ad alta precisione con un'apertura del diaframma massima di f 2.8. Lo zoom é caratterizzato inoltre da un



Figura 6.1: Fotocamera Fuji FinePix 4900Zoom

diaframma incorporato a tredici posizioni che consente di impostare la profondità di campo. Attraverso uno zoom digitale aggiuntivo di 3.75X estremamente graduale, la FinePix 4900 raggiunge un ingrandimento totale di circa 22.5X.

- *Modalità di ripresa:* La fotocamera FinePix4900 zoom é dotata di sei diverse modalità di ripresa. Automatica, semiautomatica a priorità di diaframma (adatta per il controllo della profondità di campo), semiautomatica a priorità di tempi (per azioni estremamente rapide), a scatto programmato (che consente di operare sulle selezioni automatiche effettuate dalla fotocamera), ed a modalità manuale.
- *Display LCD:* La FinePix 4900 é dotata di un display LCD TFT da due pollici, in polisilicone a bassa temperatura, da 130.000 pixel. Questo display é in grado di mostrare anteprime dettagliate che consentono di valutare anticipatamente il risultato della ripresa.
- *Mirino elettronico:* La fotocamera é dotata di un mirino elettronico che é in

grado di mostrare tutte le informazioni necessarie per ogni scatto, quali ad esempio: modalità di scatto corrente, numero di scatti, tempo di otturatore, compensazione dell'esposizione. In questo modo si può avere immediata conferma di tutte le impostazioni semplicemente osservando il visore. È inoltre possibile raddoppiare le dimensioni dell'immagine visualizzata dal mirino, per un'accurata valutazione della messa a fuoco.

6.2 Calibrazione di fotocamere con dati simulati

In questa prima prova viene inizializzata la matrice dei punti 3D *grid3d.txt* con due piani e 30 punti per foto, come si vede dalla figura (6.2).

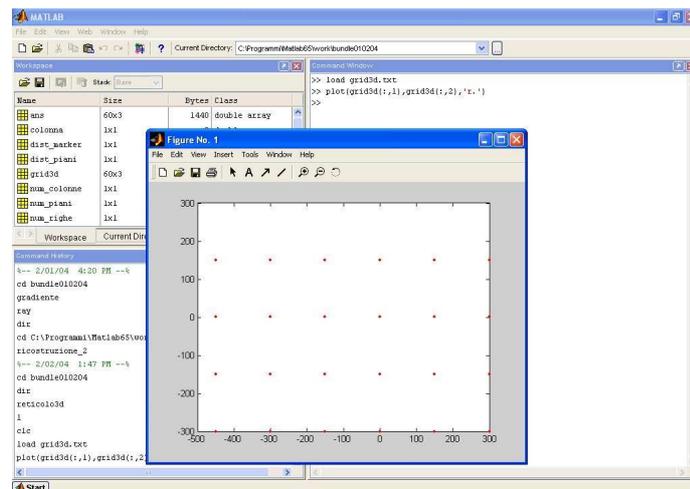


Figura 6.2: Matrice dei punti tridimensionale *grid3d*

Successivamente vengono create due differenti matrici di punti 2D. Nella prima vengono messi nel file *inizia.m* i seguenti valori:

- punti X_L , Y_L e Z_L pari a 50, 50 e 1500;
- punti x_0 e y_0 pari a 0.25 e 0.20;

- angoli ω , ϕ e κ pari a 15, 30 e 10 gradi;
- focale pari a 1.2;
- parametri di distorsione K_1 , K_2 , K_3 , P_1 e P_2 pari a 0.01, 0, 0.0001, 0 e 0.005.

La matrice dei punti bidimensionale creata da *proietta.m* per questa prima foto é quella della figura (6.3).

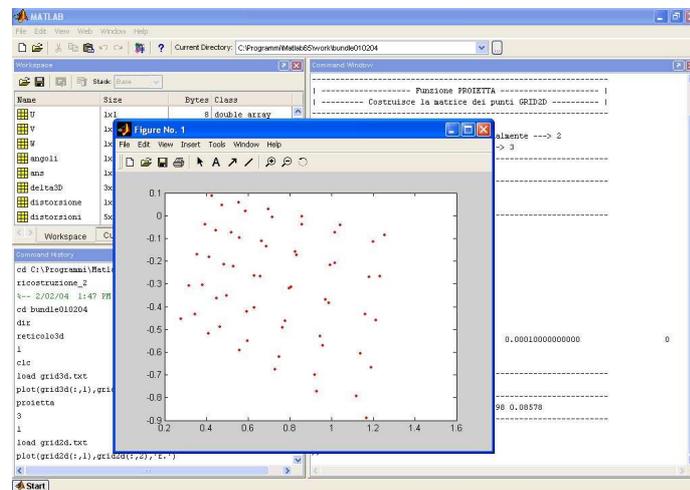


Figura 6.3: Matrice dei punti bidimensionale *grid2d* per la prima foto

Nella seconda matrice dei punti vengono messi nel file *inizia.m* i seguenti valori:

- punti X_L , Y_L e Z_L pari a -30 , -40 e 1400 ;
- punti x_0 e y_0 pari a -0.15 e -0.10 ;
- angoli ω , ϕ e κ pari a -15 , 20 e -10 gradi;
- focale pari a 1.1;
- parametri di distorsione K_1 , K_2 , K_3 , P_1 e P_2 pari a 0.01, 0, -0.0001 , 0 e -0.005 .

La Direct Linear Transformation (DLT) ha calcolati valori molto buoni, notevolmente vicini a quelli simulati. In questo modo si é potuto anche verificare che l'algoritmo funziona e dá valori vicini a quelli reali.

Per la prima foto, come si vede dalla figura (6.4), i risultati sono stati:

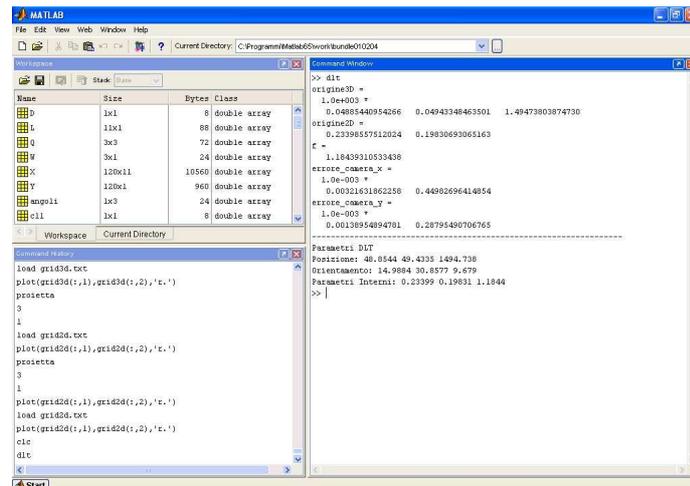


Figura 6.4: Valori dati dalla DLT per la prima foto

- punti X_L , Y_L e Z_L pari a 48.85, 49.43 e 1494.73;
- punti x_0 e y_0 pari a 0.23 e 0.19;
- angoli ω , ϕ e κ pari a 14.9884, 30.8577 e 9.679 gradi.
- focale pari a 1.18;

Per la seconda foto, i risultati sono stati:

- punti X_L , Y_L e Z_L pari a -30.52 , -39.53 e 1396.31;
- punti x_0 e y_0 pari a -0.15 e -0.09 ;

- angoli ω , ϕ e κ pari a -14.81 , 20.43 e -9.87 gradi;
- focale pari a 1.09 ;

Il successivo algoritmo del bundle adjustment é andato a convergenza nel giro di cinque iterazioni per la prima foto, mentre per la seconda foto ha fatto addirittura meglio andando a convergenza in quattro passi, un passo in meno rispetto alla prima foto. Da notare che, ovviamente, non ci sono stati problemi per il calcolo di tutti e cinque i parametri di distorsione in quanto inclusi in *iniziam*, anche se si sono verificate correlazioni alte fra alcuni parametri. Correlazione che comunque non hanno influito sul risultato finale (figura 6.5).

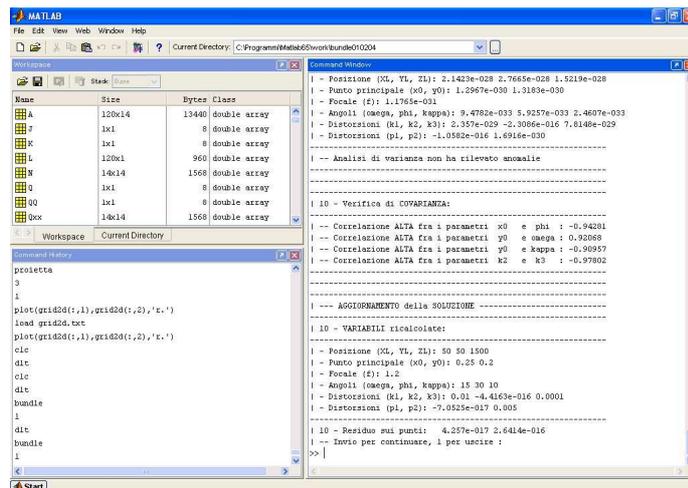


Figura 6.5: Calcolo dei valori sulla prima foto

L'algoritmo di minimizzazione del gradiente ha dato ottimi risultati sia sulla prima che sulla seconda foto, riuscendo ad eliminare completamente le distorsioni iniziali introdotte. Un esempio é dato dalla figura (6.6). In questa figura sono indicati in blu i punti di *grid2d* con distorsioni, mentre sono indicati in rosso i punti reali ricalcolati senza distorsioni.

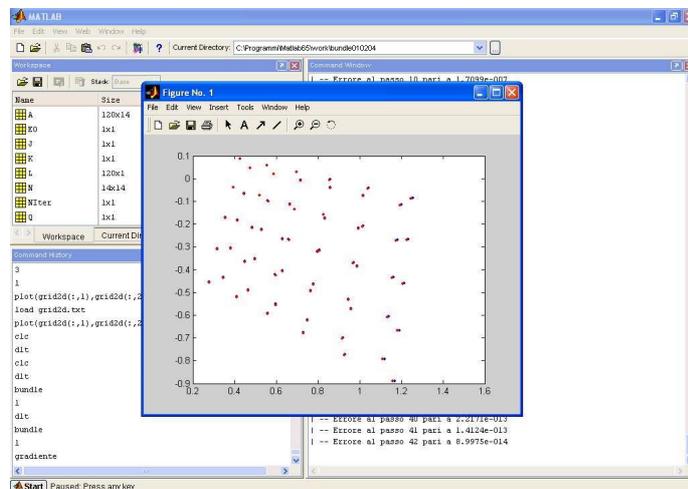


Figura 6.6: Minimizzazione tramite gradiente

La verifica dei buoni risultati raggiunti da questi programmi é data dalla ricostruzione finale dei punti tridimensionali.

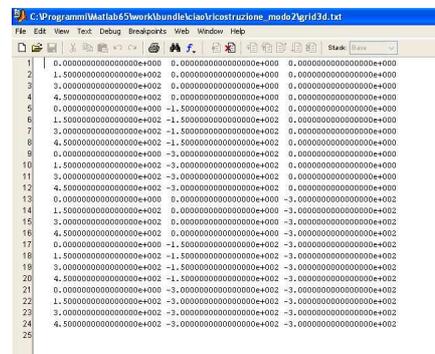


Figura 6.7: Matrice dei punti tridimensionale iniziale

Partendo dalla matrice dei punti tridimensionali come dalla figura (6.7) si é arrivati a ricostruire con una precisione molto inferiore al millimetro la stessa come si vede dalla figura (6.8).

```

1 1 -2.3558800770095736e-005 2.30116015815951121e-005 -9.71110402547490983e-004
2 1 1.500004023410499564e+002 -6.0887502797868365e-005 -4.4017918917234056e-003
3 2 2.9999984225050688e+002 6.266815182932059e-005 1.0071250908367801e-003
4 4 4.899987475110189e+002 5.1571957188301099e-005 6.2520648043573601e-004
5 -2.0861875101385749e-005 -1.5000015360910169e+002 1.506313555856439e-003
6 1 1.5000003679132919e+002 -1.4999996632376633e+002 -8.7446174802607857e-005
7 2 2.9999981952330827e+002 -1.4999990108789978e+002 1.1497167070046999e-003
8 4 4.89999567744700e+002 -1.499999705523251e+002 6.01020154710201e-004
9 2 2.9527256060646323e-005 -2.999999959992236e+002 1.7834186201071134e-004
10 1 4.899994434692621e+002 -2.9999967554388172e+002 1.1039420878660167e-003
11 2 2.9999981042440099e+002 -2.9999955300117392e+002 1.2185153892453779e-003
12 4 4.899996010752056e+002 -2.9999987029565395e+002 3.365637231974804e-004
13 -2.7459942722130037e-005 2.859933227223777e-005 -3.0000143323792361e+002
14 1 1.5000136815746697e+002 -3.0768757233090382e-004 -3.0001912107477256e+002
15 2 2.99997972639946e+002 0.0462574826431365e-005 -2.99999373264441e+002
16 4 4.899974254102108e+002 7.1589509001768192e-005 -2.999987150781585e+002
17 -5.434280307684077e-005 -1.5000026158446812e+002 -3.0000291331187350e+002
18 1 1.500014121220209e+002 -1.500001205777531e+002 -3.0000157945992760e+002
19 2 2.999998289823769e+002 -1.4999979468805389e+002 -2.999985616582327e+002
20 4 4.899986137087990e+002 -1.4999978097437247e+002 -2.999984646656117e+002
21 6 9.357374437013066e-008 -3.000000861687946e+002 -3.000075695099940e+002
22 1 4.8999989064689e+002 -2.999997895316760e+002 -2.99999265100702e+002
23 2 2.9999976035500345e+002 -2.9999957266196179e+002 -2.9999817025227821e+002
24 4 4.899996076805792e+002 -2.9999964664310688e+002 -2.9999849173571783e+002
25

```

Figura 6.8: Matrice dei punti tridimensionale ricostruita

6.3 Calibrazione di fotocamere con rumore

La stessa prova di simulazione fino a qui illustrata é stata poi fatta oggetto di un ulteriore studio introducendo del rumore sui dati. Il rumore introdotto (nell'ordine di 0.0001 pixel) ha fatto si che nelle due matrici di punti 2D le coordinate dei punti non fossero prese in modo corretto. Questo errore si é poi propagato in tutti i programmi e, ovviamente, anche l'algorithmo di minimizzazione non é riuscito ad eliminarlo completamente.

Il risultato lo si puó vedere nella figura (6.9) quando si passa alla ricostruzione tridimensionali dei punti. Aggiungendo del rumore, la precisione di ricostruzione diminuisce notevolmente. In queste prove si é potuto constatare che la stessa si aggira nell'ordine del millimetro.

6.4 Calibrazione di fotocamere con dati reali

Nella seconda prova viene inizializzata la matrice dei punti tridimensionale *grid3d.txt* con due piani e 12 punti per foto, come si vede dalla figura (6.10).

```

C:\Programmi\Matlab65\work\Bundling10204\punto3d_nem.txt
File Edit View Tools Debug Breakpoints Help Window Help
1 1 -4.4811800091614148e+002 2.3935056572331752e+002 2.023266073676425e+002
2 1 -3.002320172849143e+002 3.0017397495592127e+002 1.9921317660447903e+002
3 1 -1.5017024450696357e+002 3.0013123647705714e+002 1.9937951812294523e+002
4 1 2.503138600002303e-002 3.0001324242623349e+002 2.0007010752943961e+002
5 1 1.501320647996090e+002 2.999666189094604e+002 1.998507013111852e+002
6 1 2.9994753856734951e+002 2.9995536280255129e+002 2.0008195034105620e+002
7 1 -4.5029151834008986e+002 1.5024938945488231e+002 1.9942302935336344e+002
8 1 -2.995263965497723e+002 1.498580325022070e+002 2.017385659526304e+002
9 1 -1.499672685515743e+002 1.500382301299472e+002 1.9941903660006051e+002
10 1 -2.2296719868040853e-002 1.4987447311836479e+002 2.0075145337992944e+002
11 1 1.4995122317119892e+002 1.4983958981730701e+002 2.0054372390979029e+002
12 1 2.999232840052098e+002 1.5006496671152497e+002 2.0034812046191184e+002
13 1 -4.5071696911295254e+002 -1.2193818709107518e-003 1.9823273700042501e+002
14 1 -2.9962085736745973e+002 -1.9822520555760106e-002 2.01010807463764028e+002
15 1 -1.5013810202020205e+002 -1.6239414463387732e-002 1.9977155216345548e+002
16 1 -9.9248930254050016e-003 3.743695931932157e-002 2.0050570154280547e+002
17 1 1.5006935470176019e+002 2.2671811961632926e-002 1.9975209326852780e+002
18 1 3.0015994533806735e+002 1.0324609194982211e-002 1.9973297103172370e+002
19 1 -4.4905265949104832e+002 -1.4968016069846607e+002 2.0126839631297307e+002
20 1 -3.0052829037774518e+002 -1.5031041862306580e+002 1.9793602277866512e+002
21 1 -1.499233726809948e+002 -1.4992091508295181e+002 2.0094077748152130e+002
22 1 -4.51549000015596e-002 -1.499330320969604e+002 2.0020097454500094e+002
23 1 1.4981767563413109e+002 -1.4974653367699054e+002 2.0163055897031146e+002
24 1 3.0010480001100370e+002 -1.501576485476901e+002 1.9945801680404037e+002
25 1 -4.4786086274606595e+002 -2.9849259842070214e+002 2.0594537553114539e+002
26 1 -3.0028756697660992e+002 -3.0033454686482628e+002 1.9884834033566970e+002
27 1 -1.5030943475883049e+002 -3.0054728859422373e+002 1.9744344560268664e+002
28 1 -1.6769290996373177e-002 -3.0061425582851211e+002 1.9766914279448020e+002
29 1 1.500417275091370e+002 -3.00025158614801471e+002 1.9988676514913577e+002
30 1 3.0004839304346888e+002 -3.0004004760272727e+002 1.9979759603290254e+002
31 1 -4.5001767166508795e+002 2.99999470651039e+002 5.5186838268986001e-002
32 1 -2.991240497507584e+002 2.9992049355462027e+002 1.7939298400016399e+000
33 1 -1.5037764866925800e+002 3.0058319715017183e+002 -2.642724916671593e+000
34 1 4.3012811255671579e-002 3.001652528320194e+002 -7.103424500803191e-001
35 1 -1.499727942648012e+002 3.0017814925859847e+002 -5.191327710908132e-001

```

Figura 6.9: Matrice dei punti tridimensionale ricostruita (con rumore)

Successivamente vengono create due differenti matrici di punti 2D grazie al programma *reticolo2d.m*. Per le due foto si é utilizzata la fotocamera digitale sopra descritta.

Con la DLT si sono trovati valori di inizializzazione per la variabili incognite. Valori che poi verranno prelevati dal programma del bundle adjustment come prima inizializzazione.

Per la prima foto, che é stata fatta con una risoluzione di 640×480 pixel, i risultati sono stati:

- punti X_L , Y_L e Z_L pari a 5.70, -316.07 e 921.16;
- punti x_0 e y_0 pari a 324.23 e 240.47;
- angoli ω , ϕ e κ pari a 8.35, -11.31 e 4.81 gradi.
- focale pari a 786.04;

Per la seconda foto, anch'essa fatta con una risoluzione di 640×480 pixel, i risultati sono invece stati (figura 6.11):

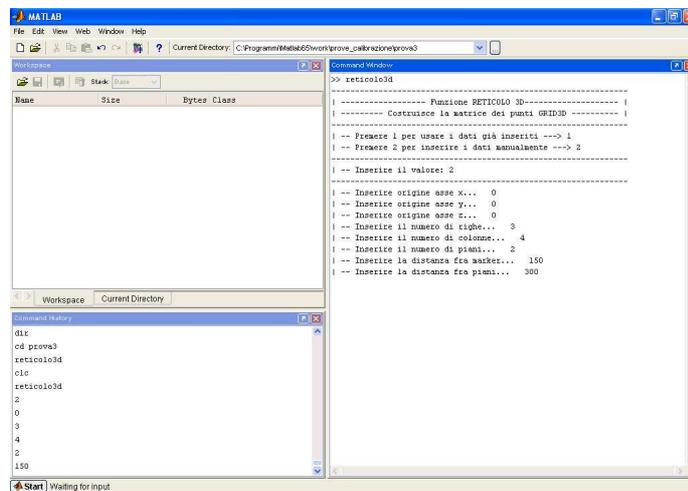


Figura 6.10: Creazione della matrice tridimensionale dei punti *grid3d*

- punti X_L , Y_L e Z_L pari a 534.37, -317.54 e 842.25;
- punti x_0 e y_0 pari a 308.90 e 245.80;
- angoli ω , ϕ e κ pari a 8.85, 16.79 e 0.18 gradi.
- focale pari a 792.61;

Con il successivo algoritmo del bundle adjustment i valori della variabili incognite si sono assestati intorno al passo numero 10 sia per la prima foto che per la seconda foto. Successive iterazioni avrebbero comunque potuto migliorare ulteriormente questi valori anche se in minima parte. Da notare che, sia per la prima che per la seconda foto, si sono dovuti eliminare completamente i 5 parametri di distorsione in quanto, se si fossero lasciati, si sarebbero avuti problemi di condizionamento e di singolarità della matrice di disegno. Se, invece, non fossero stati tolti, i parametri di distorsione risultanti sarebbero stati notevolmente piccoli e, quindi, di poca utilità pratica.

Per la prima foto, dopo 10 iterazioni, i risultati sono stati:

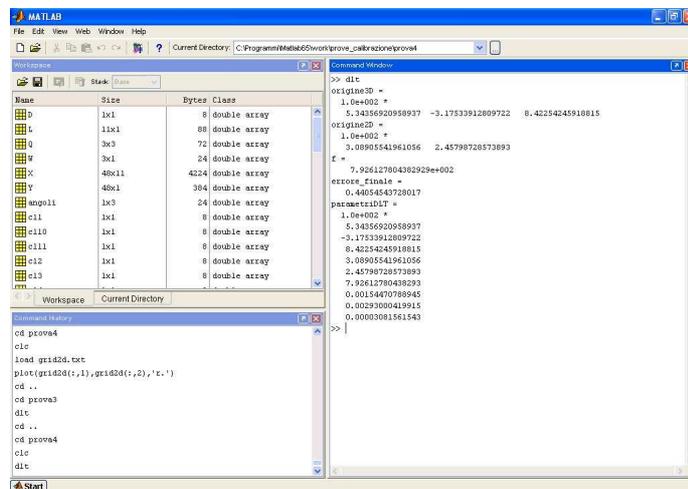


Figura 6.11: Valori della DLT sulla seconda foto

- punti X_L , Y_L e Z_L pari a 5.84, -315.99 e 919.46 ;
- punti x_0 e y_0 pari a 325.80 e 239.83;
- angoli ω , ϕ e κ pari a 8.33, -11.44 e 4.82 gradi.
- focale pari a 784.28;

Per la seconda foto, dopo 10 iterazioni, come si vede dalla figura (6.12), i risultati sono invece stati:

- punti X_L , Y_L e Z_L pari a 533.92, -317.10 e 840.04;
- punti x_0 e y_0 pari a 307.73 e 245.84;
- angoli ω , ϕ e κ pari a 8.85, 16.88 e 0.17 gradi.
- focale pari a 790.66;

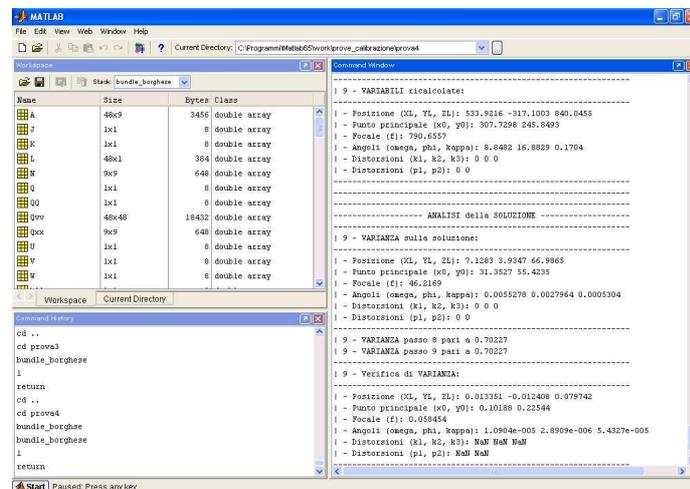


Figura 6.12: Calcolo dei valori sulla seconda foto

L'algoritmo di minimizzazione del gradiente, quindi, non è stato utilizzato. Infatti le distorsioni, essendo minime, non hanno influito sulla misurazione dei punti 2D.

La verifica dei buoni risultati raggiunti, quindi, è da verificarsi con la ricostruzione finale dei punti tridimensionali.

Partendo dalla matrice dei punti tridimensionali inizialmente costruita (figura 6.13) si è arrivati a ricostruire con una precisione molto alta la stessa come si vede dalla figura (6.14). Anche in questo caso, come nella prova dei dati simulati con rumore, l'errore nella precisione si aggira attorno al millimetro.

```

1 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000
2 1.00000000000000e+002 0.00000000000000e+000 0.00000000000000e+000
3 3.00000000000000e+002 0.00000000000000e+000 0.00000000000000e+000
4 4.50000000000000e+002 0.00000000000000e+000 0.00000000000000e+000
5 0.00000000000000e+000 -1.50000000000000e+002 0.00000000000000e+000
6 1.50000000000000e+002 -1.50000000000000e+002 0.00000000000000e+000
7 3.00000000000000e+002 -1.50000000000000e+002 0.00000000000000e+000
8 4.50000000000000e+002 -1.50000000000000e+002 0.00000000000000e+000
9 0.00000000000000e+000 -3.00000000000000e+002 0.00000000000000e+000
10 1.50000000000000e+002 -3.00000000000000e+002 0.00000000000000e+000
11 3.00000000000000e+002 -3.00000000000000e+002 0.00000000000000e+000
12 4.50000000000000e+002 -3.00000000000000e+002 0.00000000000000e+000
13 0.00000000000000e+000 0.00000000000000e+000 -1.00000000000000e+002
14 1.50000000000000e+002 0.00000000000000e+000 -1.00000000000000e+002
15 3.00000000000000e+002 0.00000000000000e+000 -1.00000000000000e+002
16 4.50000000000000e+002 0.00000000000000e+000 -1.00000000000000e+002
17 0.00000000000000e+000 -1.50000000000000e+002 -1.00000000000000e+002
18 1.50000000000000e+002 -1.50000000000000e+002 -1.00000000000000e+002
19 3.00000000000000e+002 -1.50000000000000e+002 -1.00000000000000e+002
20 4.50000000000000e+002 -1.50000000000000e+002 -1.00000000000000e+002
21 0.00000000000000e+000 -3.00000000000000e+002 -1.00000000000000e+002
22 1.50000000000000e+002 -3.00000000000000e+002 -1.00000000000000e+002
23 3.00000000000000e+002 -3.00000000000000e+002 -1.00000000000000e+002
24 4.50000000000000e+002 -3.00000000000000e+002 -1.00000000000000e+002
25

```

Figura 6.13: Matrice dei punti tridimensionale iniziale

```

1 5.31544203293294e+001 -8.95166484650719e+002 -1.120113747121079e+001
2 1.480702023992065e+002 5.993857704365334e+001 8.248953131830079e+001
3 3.0209961776070273e+002 4.014801224312141e+001 3.903899275420203e+001
4 4.497849724934625e+002 5.771246206511717e+002 79.04837747813001e+001
5 -1.1646613102812760e+001 -1.407824820397235e+002 8.674209427115277e+001
6 1.480702023992065e+002 -1.4094624893708106e+002 1.678156487853353e+000
7 3.0209961776070273e+002 -1.496089339851333e+002 7.2423540579293942e+001
8 4.497849724934625e+002 -1.409323587322809e+002 1.028075165548023e+000
9 5.944622444401399e+001 -2.598805205001772e+002 -1.470797389323542e+000
10 1.480702023992065e+002 -3.0881396467833e+002 2.84891720707071e+001
11 3.0209961776070273e+002 -3.020996744610289e+002 2.00782645140966e+001
12 4.497849724934625e+002 -3.001008107277744e+002 -1.731451652033992e+000
13 5.944622444401399e+001 -7.465296739860213e+001 -3.978080632581107e+002
14 1.407824820397235e+002 8.95500652123280e+001 -3.00982727933291e+002
15 3.0209961776070273e+002 5.31038969548295e+001 -1.006500139807001e+002
16 4.4948081315532174e+002 1.285964146364830e+001 -2.998326132183998e+002
17 1.480702023992065e+002 -1.503050303000000e+002 -2.90526716556002e+002
18 1.48071798785231e+002 -1.480718932501026e+002 -3.030189226471480e+002
19 3.0209961776070273e+002 -1.40988144011568e+002 -3.02670024278346e+002
20 4.497849724934625e+002 -1.5048465513078e+002 -2.99138018467054e+002
21 1.11523202793207e+000 -3.0008767406426e+002 -2.976385829742829e+002
22 1.1011670214620134e+002 -3.00646141482074e+002 -3.004762874767978e+002
23 3.0209961776070273e+002 -3.020522187857191e+002 -3.017821394460848e+002
24 4.49729543623025e+002 -2.99972490561466e+002 -2.992123712120084e+002
25

```

Figura 6.14: Matrice dei punti tridimensionale ricostruita

Capitolo 7

Note sul SW

7.1 Introduzione

I programmi sviluppati, in particolare il programma *bundle*, possono essere considerati come i risultati pratici di questa tesi. In questi programmi sono infatti stati sviluppati tutti gli algoritmi e tutta la teoria trattata nei capitoli precedenti.

Tutti i programmi sono stati sviluppati in MATLAB, un programma utilizzato principalmente per la sua struttura matriciale. Quindi sarà necessario dotarsi di un computer con installato un sistema operativo Windows o Linux per poter installare MATLAB.

7.2 Interfaccia utente

Una volta fatto partire MATLAB, il programma andrà di default nella directory di lavoro preimpostata, in questo caso `c:\programmi\matlab65\work`.

Come prima operazione sarà dunque necessario cambiare directory e spostarsi in quella dove sono localizzati i file sorgenti, in questo caso `c:\programmi\matlab65\work\bundle010204`.

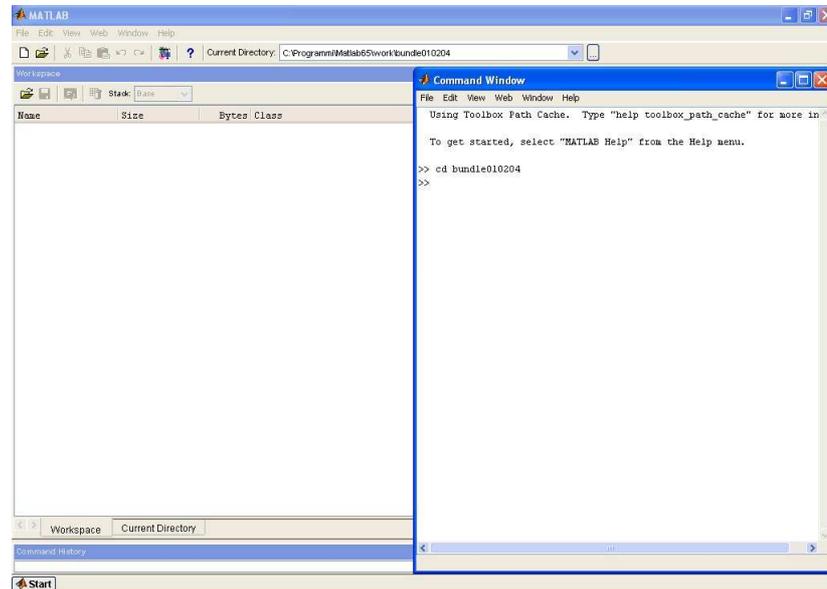


Figura 7.1: Interfaccia MATLAB e directory di lavoro

7.3 Comandi e funzioni

I comandi e le relative funzioni sono illustrati nell'ordine stesso in cui devono essere utilizzati. Sarà necessario fare i seguenti passi:

- creare la matrice contenente le coordinate 3D dell'oggetto di calibrazione;
- acquisire con due fotocamere diverse in posizioni fisse una coppia di immagini del tool di calibrazione (trattiamo qui solo il caso di due fotocamere, ma il sistema é espandibile al caso di N fotocamere);
- creare la matrice dei punti 2D, indicando su ciascuna immagine i punti di coordinate note sull'oggetto di calibrazione;
- inizializzare l'algoritmo del bundle adjustment con la DLT;

- usare l'algoritmo del bundle adjustment per la calibrazione;
- eventuale altra acquisizione con due camere ed individuazione dei punti di matching nelle due immagini;
- eliminare le eventuali distorsioni con l'algoritmo del gradiente;
- ricostruire i punti tridimensionali.

7.4 Creazione della matrice dei punti di coordinate nota

Per prima cosa, é necessario creare la matrice dei punti tridimensionali per la calibrazione. Il programma per crearla si chiama *reticolo3d.m*. Una volta richiamato il file digitando sulla shell di MATLAB il nome omonimo, il programma si presenta con una comoda interfaccia grafica ad opzione multipla.

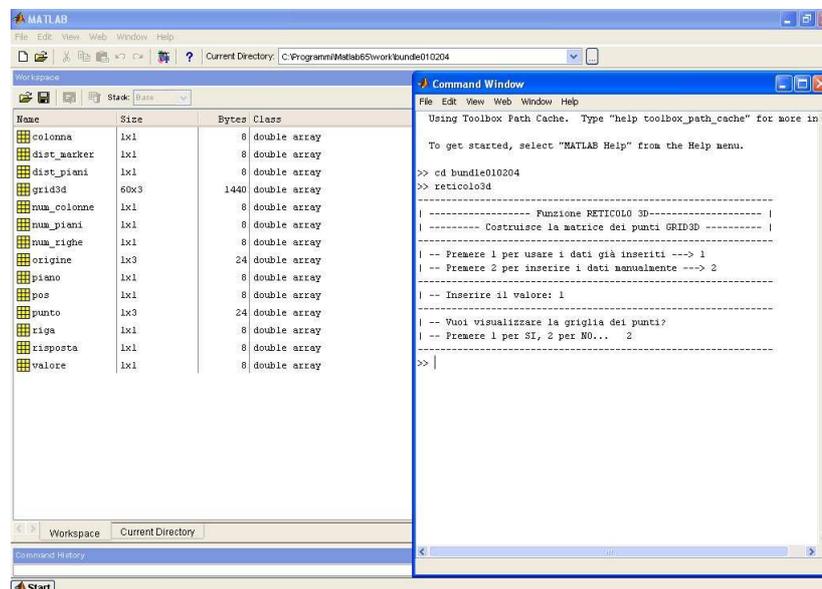


Figura 7.2: Programma Reticolo3d.m

Due sono le possibili scelte da effettuare. Se si digita 1, il programma utilizza i dati già impostati all'interno del suo codice. Se invece si digita 2, il programma richiede varie informazioni sulla matrice dei punti da creare, in particolare:

1. origine asse x , asse y ed asse z ;
2. numero di *righe*, di *colonne* e di *piani*;
3. distanza fra *marker* e fra *piani*.

Questo perché per calibrare la fotocamera digitale si è scelto di utilizzare un pattern planare rappresentato in figura (7.3). Per una buona calibrazione è necessario utilizzare almeno due immagini, riprese con il pattern planare posto a distanze differenti. Il programma dá in uscita il file *grid3d.txt*, contenente i valori X_A , Y_A e Z_A di ciascun punto del pattern.

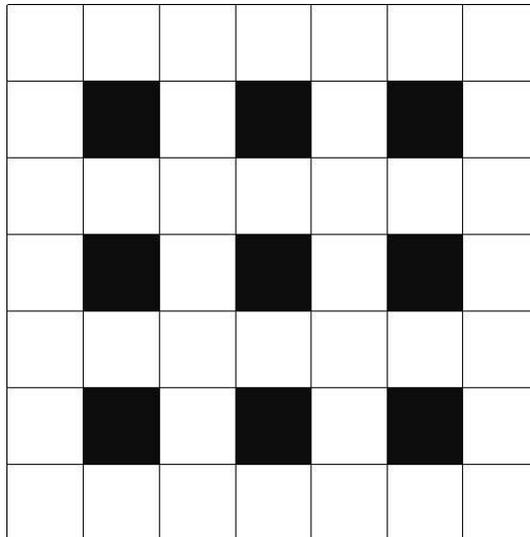


Figura 7.3: Pattern planare

7.5 Acquisizione di una coppia di immagini

Per poter alla fine ricostruire i punti tridimensionali, é necessario acquisire almeno una coppia di immagini con due fotocamere fisse. Per fare questo, le due fotocamere dovrebbero trovarsi a buona distanza fra di loro e comunque fotografare lo stesso pattern planare.

7.6 Creazione della matrice bidimensionale dei punti

Il passaggio successivo é invece quello di generare le coordinate 2D. Possiamo percorrere due strade differenti. La prima é quella di generare le coordinate 2D in maniera simulata, la seconda é quella di prendere a mano le coordinate bidimensionali delle foto.

7.6.1 Coordinate 2D simulate

Nel primo caso si deve richiamare il programma *proietta.m*.

Il programma, avendo note le coordinate tridimensionali precedentemente create e le approssimazioni iniziali dei parametri di orientamento interno ed esterno (inserite manualmente nel file *inizia.m*), crea in automatico la matrice dei punti 2D salvandola con il nome di *grid2d.txt*. Questi punti saranno anche soggetti ad errori di distorsione radiale o tangenziale, sempre se nel file *inizia.m* sono state incluse o no tali distorsioni.

7.6.2 Coordinate 2D prese da foto reali

La seconda strada, invece, é quella di prendere a mano le coordinate bidimensionali delle foto. In questo caso é necessario far partire il programma *reticolo2d.m*.

Per rendere uniforme il tutto, le scelte da effettuare sono uguali a quelle di *reticolo3d.m*. Se si digita 1, il programma utilizza i dati presenti nel suo codice, altrimenti

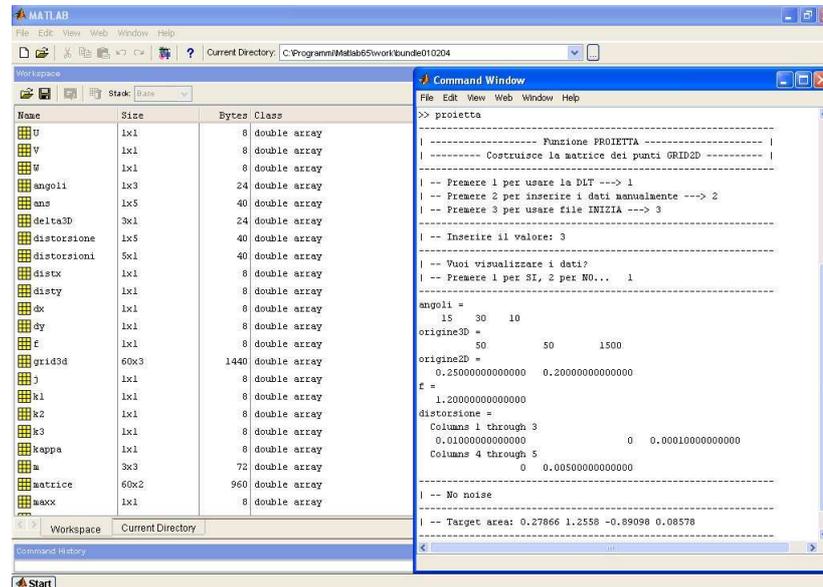


Figura 7.4: Programma Proietta.m

(digitando 2) il programma richiede varie informazioni sulla matrice dei punti da creare, uguali a quelle richieste per *reticolo3d.m*. Inutile dire che il numero di punti inseriti per *reticolo3d.m* deve essere lo stesso anche per *reticolo2d.m*. Dopo la selezione, il programma carica la prima foto, da salvare nella stessa directory del programma con il nome *foto1.jpg*. A questo punto sarà necessario selezionare i punti di controllo nel pattern planare, punti che verranno salvati nel file *grid2d.txt*. Questa operazione viene effettuata per tutte le foto in rapida successione.

A questo punto, sia che si operi con dati simulati sia che si operi con fotografie reali, si avranno a disposizione due file: il file *grid3d.txt* con i dati tridimensionali ed il file *grid2d.txt* con i dati bidimensionali corrispondenti.

7.7 Inizializzazione dei parametri tramite DLT

Sarà ora necessario far partire il programma *dlt.m*.

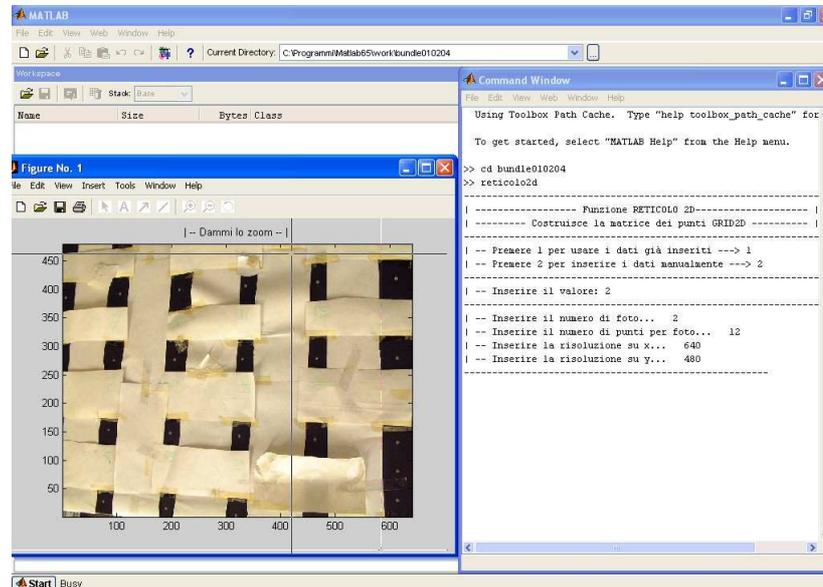


Figura 7.5: Programma Reticolo2d.m

Questo file contiene un piccolo algoritmo capace di dare un'approssimazione iniziale dei parametri intrinseci ed estrinseci della fotocamera essendo note solo le coordinate 2d (caricate dal file *grid2d.txt*) e le coordinate 3d (caricate dal file *grid3d.txt*). Tutti i parametri di orientamento interno ed esterno vengono poi raggruppati in un unico vettore e poi salvate in un file di testo chiamato *dlt_para.txt*. Il file *dlt_para.txt* è di vitale importanza in quanto il successivo algoritmo del bundle adjustment viene inizializzato a partire dai valori contenuti in questo file.

7.8 Programma bundle adjustment

A questo punto bisogna far girare il file *bundle.m*. Il programma del bundle adjustment ha al suo interno un modello per correggere le eventuali distorsioni radiali e tangenziali. Di default è impostato con 5 parametri di distorsione attivi, ma se il modello risulta sovradeterminato è possibile eliminare tale parametri.

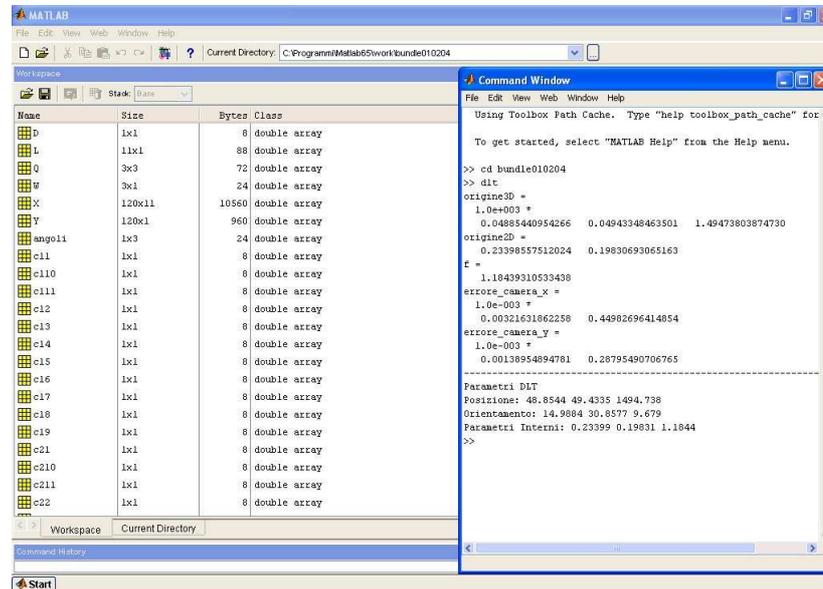


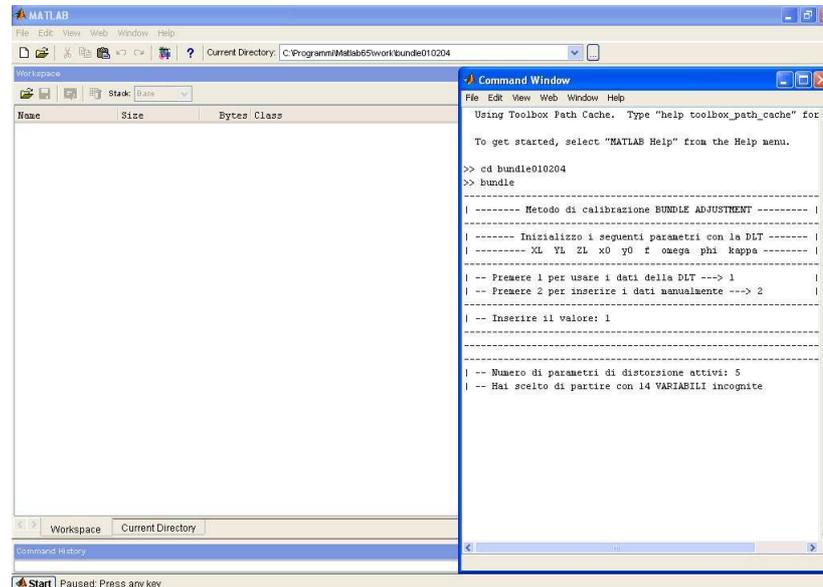
Figura 7.6: Programma Dlt.m

Una volta chiamato, il programma `bundle.m` si presenta come nella figura (7.7).

Anche qui é possibile effettuare una scelta multipla. Se si digita 1 con la tastiera, il programma prende come approssimazioni iniziali i valori contenuti nel file `dlt_para.txt` precedentemente creato. Altrimenti, digitando 2, il programma richiede di indicare le approssimazioni iniziali per i parametri incogniti. In particolare:

- valore di X_L , Y_L e Z_L ;
- valore di x_0 e y_0 ;
- valore del *fuoco*;
- valore degli angoli ω , ϕ e κ (in gradi).

Successivamente il programma fornisce alcune indicazioni utili sul suo funzionamento. Per prima cosa indica con quanti parametri di distorsione si é scelto di partire.



```

MATLAB
File Edit View Web Window Help
Current Directory: C:\Programmi\Matlab65\work\bundle010204

Workspace
Name Size Bytes Class

Command Window
File Edit View Web Window Help
Using Toolbox Path Cache. Type "help toolbox_path_cache" for
To get started, select "MATLAB Help" from the Help menu.
>> cd bundle010204
>> bundle
| ----- Metodo di calibrazione BUNDLE ADJUSTMENT ----- |
| ----- Inizializzo i seguenti parametri con la DLT ----- |
| ----- XL YL ZL x0 y0 r omega phi kappa ----- |
| -- Premere 1 per usare i dati della DLT --> 1 |
| -- Premere 2 per inserire i dati manualmente --> 2 |
| -- Inserire il valore: 1 |
| ----- |
| -- Numero di parametri di distorsione attivi: 5 |
| -- Hai scelto di partire con 14 VARIABILI incognite |
Command History
Start Paused: Press any key

```

Figura 7.8: Numero di parametri di distorsione attivi

- l'analisi di covarianza;
- l'aggiornamento della soluzione.

Vediamo come interpretare ciascuno di questi punti.

7.8.1 Condizionamento della matrice

In questo primo punto l'algoritmo controlla il condizionamento della matrice normale ($A^T A$) in norma 1 ed in norma 2. Se il numero di condizionamento é troppo alto (superiore a $1.2 \cdot 10^{12}$), il programma scrive a video un avviso dove viene evidenziato che la matrice normale potrebbe essere singolare e quindi soggetta ad errori.

Altrimenti, il programma fa vedere il valore del condizionamento della matrice (di solito abbastanza elevato) senza mettere a video ulteriori avvisi. Segno che il problema numerico é ben posto.

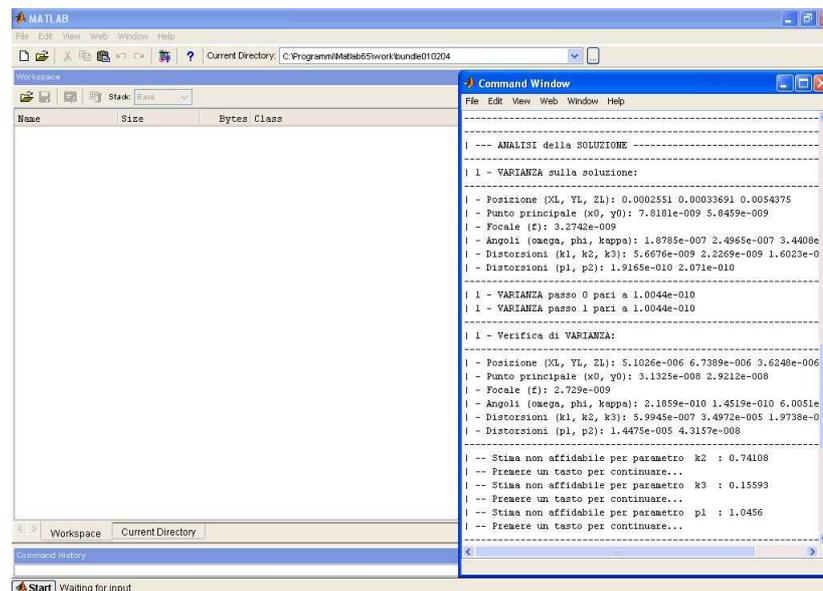
7.8.4 La media e la varianza sul residuo

La media e la varianza sul residuo indicano rispettivamente la media e la varianza sul residuo della soluzione precedentemente calcolata. Ad ogni passo di iterazione questi due valori dovrebbero calare, per poi assestarsi definitivamente quando gli incrementi sulle variabili sono nulli.

7.8.5 L'analisi di varianza

L'analisi di varianza é divisa in due passi:

- varianza sulla soluzione;
- verifica di varianza.



```

MATLAB
File Edit View Web Window Help
Current Directory: C:\Programmi\Matlab65\work\bundle010204

Workspace
Name Size Bytes Class

Command Window
File Edit View Web Window Help

--- ANALISI della SOLUZIONE ---
| |
| | - VARIANZA sulla soluzione:
| |
| | - Posizione (XL, YL, ZL): 0.0002551 0.00033691 0.0054375
| | - Punto principale (x0, y0): 7.8181e-009 5.8459e-009
| | - Focale (f): 3.2742e-009
| | - Angoli (omega, phi, kappa): 1.8785e-007 2.4965e-007 3.4408e
| | - Distorsioni (k1, k2, k3): 5.6676e-009 2.2269e-009 1.6023e-0
| | - Distorsioni (p1, p2): 1.9165e-010 2.071e-010
| |
| | - VARIANZA passo 0 pari a 1.0044e-010
| | - VARIANZA passo 1 pari a 1.0044e-010
| |
| | - Verifica di VARIANZA:
| |
| | - Posizione (XL, YL, ZL): 5.1026e-006 6.7309e-006 3.6248e-006
| | - Punto principale (x0, y0): 3.1325e-008 2.9212e-008
| | - Focale (f): 2.729e-009
| | - Angoli (omega, phi, kappa): 2.1859e-010 1.4519e-010 6.0051e
| | - Distorsioni (k1, k2, k3): 5.9945e-007 3.4972e-005 1.9738e-0
| | - Distorsioni (p1, p2): 1.4475e-005 4.3157e-008
| |
| | -- Stima non affidabile per parametro k2 : 0.74108
| | -- Premere un tasto per continuare...
| | -- Stima non affidabile per parametro k3 : 0.15593
| | -- Premere un tasto per continuare...
| | -- Stima non affidabile per parametro p1 : 1.0456
| | -- Premere un tasto per continuare...

```

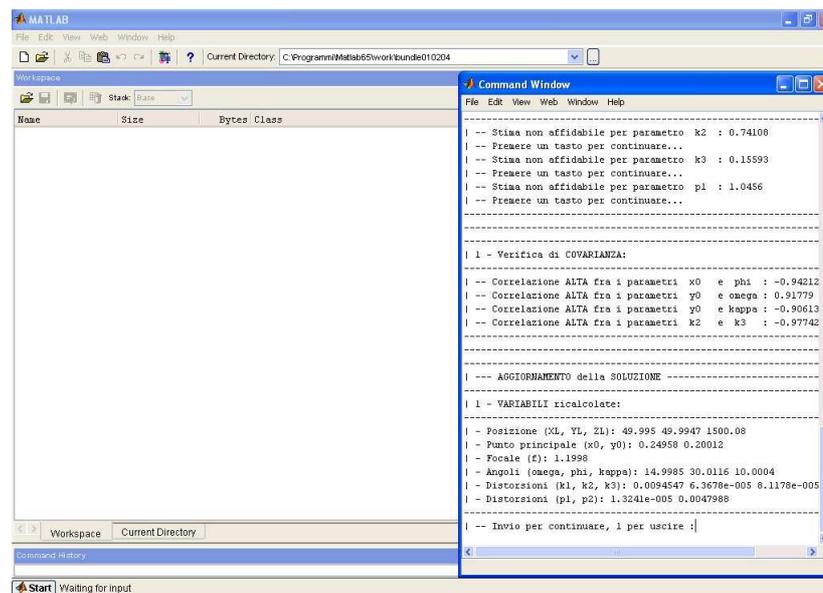
Figura 7.10: Analisi di varianza

La varianza sulla soluzione indica la varianza di ciascun parametro stimato. La verifica di varianza permette di verificare se ci sono stati incrementi sostanziali sul

valore delle variabili ricalcolate. Più il valore diventa basso, più precise saranno le soluzioni finali ricalcolate.

7.8.6 L'analisi di covarianza

Nella analisi di covarianza si utilizza l'analisi di correlazione fra i parametri presenti. Se due o più parametri hanno una correlazione maggiore di 0.9, il programma lo dice a video. Questo significa che i parametri indicati possono non essere stimati in modo corretto. Con una correlazione vicina a 0.99 sarebbe dunque necessario eliminare uno dei due parametri. Per fare questo è sufficiente editare il programma *bundle.m* ed eliminare, nelle prime righe del programma, il parametro (od i parametri) corrispondente.



```
MATLAB
File Edit View Web Window Help
Current Directory: C:\Programmi\Matlab65\work\bundle010204

Workspace
Name Size Bytes Class

Command Window
File Edit View Web Window Help
-----
|-- Stima non affidabile per parametro k2 : 0.74108
|-- Premere un tasto per continuare...
|-- Stima non affidabile per parametro k3 : 0.15593
|-- Premere un tasto per continuare...
|-- Stima non affidabile per parametro p1 : 1.0456
|-- Premere un tasto per continuare...
-----
| | - Verifica di COVARIANZA:
-----
|-- Correlazione ALTA fra i parametri x0 e phi : -0.94212
|-- Correlazione ALTA fra i parametri y0 e omega : 0.91779
|-- Correlazione ALTA fra i parametri y0 e kappa : -0.90613
|-- Correlazione ALTA fra i parametri k2 e k3 : -0.97742
-----
| --- AGGIORNAMENTO della SOLUZIONE ---
| | - VARIABILI ricalcolate:
| - Posizione (XL, YL, ZL): 49.995 49.9947 1500.08
| - Punto principale (x0, y0): 0.24958 0.20012
| - Focale (f): 1.1998
| - Angoli (omega, phi, kappa): 14.9985 30.0116 10.0004
| - Distorsioni (k1, k2, k3): 0.0094547 6.3670e-005 8.1178e-005
| - Distorsioni (p1, p2): 1.3241e-005 0.0047988
-----
|-- Invio per continuare, 1 per uscire :|
```

Figura 7.11: Analisi di covarianza ed aggiornamento della soluzione

7.8.7 L'aggiornamento della soluzione

Alla fine di ogni passo di iterazione, il programma calcola gli aggiornamenti delle variabili e li espone a video. Una volta che i valori delle variabili sono reputati accettabili, é possibile uscire dal programma digitando 1 alla fine del passo di iterazione. Questo comporta sostanzialmente che i valori delle variabili vengono salvati in un vettore di nome *xvett.txt*.

7.9 Minimizzazione delle distorsioni

Visto che con il bundle adjustment sono stati calcolati anche i valori di distorsione radiale e tangenziale sará necessario, prima di procedere alla ricostruzione tridimensionale, minimizzare le distorsioni sulla matrice di punti bidimensionali precedentemente create (*grid2d.txt*). Per questo é necessario richiamare il file *gradiente.m* che carica al suo interno la matrice dei punti *grid2d.txt* ed i valori di distorsione presenti nel file *xvett.txt*.

A video é possibile vedere l'effettiva minimizzazione delle distorsioni sulla matrice 2D.

In uscita il programma dá il nuovo file della matrice dei punti, chiamato *grid2d_non.txt*.

7.10 Ricostruzione tridimensionale

Il passo finale é quello di ricostruire ciascun punto in coordinate tridimensionali. Ovviamente sará prima necessario ripetere dall'inizio il procedimento fin qui descritto per la calibrazione di un'altra fotocamera.

Come sappiamo, la ricostruzione tridimensionale si basa sulle informazioni fornite da una coppia di immagini stereo. Quindi sará necessario, prima di far partire la ricostruzione, distinguere la matrice dei punti 2D di ciascuna delle due foto, oltre ai due vettori contenenti i parametri intrinseci ed estrinseci della fotocamera.

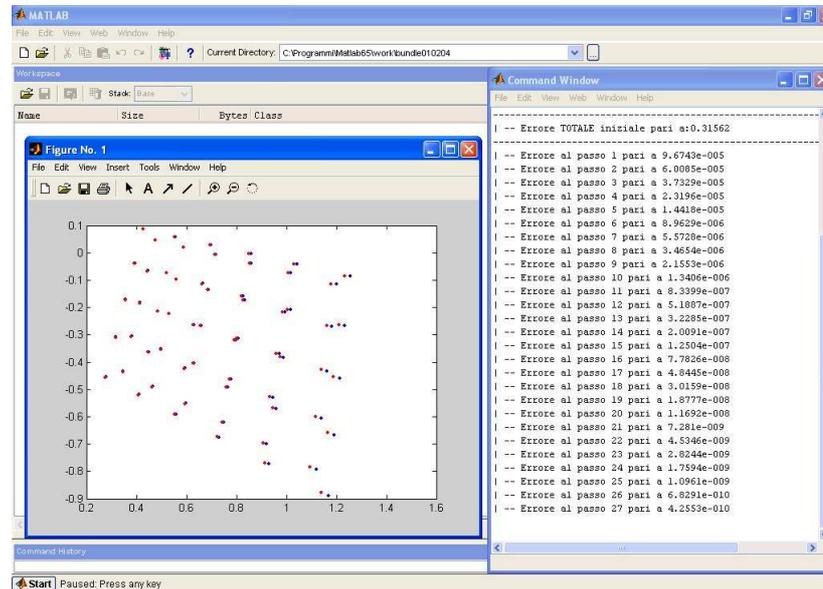


Figura 7.12: Minimizzazione delle distorsioni

I due file della prima foto dovranno cambiare nome nel seguente modo:

1. da *grid2d_non.txt* a *grid2d_foto1.txt*;
2. da *xvett.txt* a *xvett_foto1.txt*.

I due file della seconda foto dovranno cambiare nome nel seguente modo:

1. da *grid2d_non.txt* a *grid2d_foto2.txt*;
2. da *xvett.txt* a *xvett_foto2.txt*.

Una volta che i nomi sono stati cambiati, si può procedere a fare l'ultimo passo. Richiamando il file *ricostruzione.m*, l'algoritmo di *ray intersection* carica le coordinate 2D dei punti delle due fotografie e salva nel file *punto3D.txt* i punti tridimensionali così ricostruiti.

7.11 Guida pratica all'utilizzo del software

Viene ora data una guida veloce per poter usare il software.

Per calibrare una fotocamera digitale bisogna eseguire una serie di passi:

- creare la matrice dei punti tridimensionali con il programma *reticolo3d.m*. A video si potrà scegliere se inserire i dati in modo manuale o caricare quelli già preimpostati nel programma;
- creare la matrice dei punti bidimensionali. In questo caso si potrà scegliere se avere i dati in modo simulato (in questo caso bisogna chiamare il file *proietta.m*) oppure prendere i dati in modo reale da fotografie (in questo caso si chiamerà il file *reticolo2d.m*);
- inizializzare il successivo algoritmo del bundle adjustment con la DLT (chiamando il file *dlt.m*) che prende in input il file *grid3d.txt* ed il file *grid2d.txt* precedentemente creati e da in output il file *dlt_para.txt*;
- chiamare il file *bundle.m* che, prendendo come approssimazioni iniziali quelle contenute nel file *dlt_para.txt*, calibra la fotocamera digitale scrivendo il output il file *xvett.txt* contenente il valore delle variabili;
- per eliminare le distorsioni presenti quando si sono calcolati i punti bidimensionali, é necessario far partire il programma *gradiente.m* che prende in input il file *grid2d.txt* ed in uscita dá il file *grid2d_non.txt*;
- l'ultimo passo é quello di ricostruire i punti. Per farlo occorre fare i passaggi fin qui descritti per un'altra fotografia e cambiare i nomi ai due file per distinguerli. Infine bisogna far partire il file denominato *ricostruzione.m* che in uscita dá il file *punto3d.txt* con i punti così ricostruiti.

7.12 Sviluppi futuri

La qualità del risultato finale ottenuto é sicuramente molto importante. Il modello sviluppato é molto robusto sotto tutti i punti di vista e quindi possiamo essere sicuri che sulla base di questi risultati si possano costruire algoritmi per la ricostruzione completa di oggetti o figure tridimensionali, come ad esempio un volto umano. Sarebbe quindi necessario utilizzare una piú comoda interfaccia grafica, magari sviluppandola in linguaggio C++ tramite le *QT*. Il modello del bundle adjustment potrebbe poi essere perfezionato aggiungendo anche punti di controllo non noti sul pattern tridimensionale, arrivando cosí ad un modello di auto calibrazione.

Appendice A

.1 Introduzione

In questa appendice vengono indicate le derivate parziali dei coefficienti della matrice di disegno A per il calcolo del bundle adjustment. Le derivate sono state calcolate con il programma *Mathematica* della Wolfram Research, estremamente flessibile ed ampio.

.2 Calcolo delle derivate parziali

Ricordiamo che le equazioni di collinearità sono espresse come:

$$(x - x_0) + \Delta_X = -f \left[\frac{r_{11}(X_A - X_L) + r_{12}(Y_A - Y_L) + r_{13}(Z_A - Z_L)}{r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(Z_A - Z_L)} \right] \quad (.2.1)$$

$$(y - y_0) + \Delta_Y = -f \left[\frac{r_{21}(X_A - X_L) + r_{22}(Y_A - Y_L) + r_{23}(Z_A - Z_L)}{r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(Z_A - Z_L)} \right] \quad (.2.2)$$

La *forma standard* delle equazioni di collinearità risulta invece essere

$$F = (x - x_0) + f \left[\frac{U}{W} \right] + \Delta_X \quad (.2.3)$$

$$G = (y - y_0) + f \left[\frac{V}{W} \right] + \Delta_Y \quad (.2.4)$$

con

$$\begin{aligned}
 U &= r_{11}(X_A - X_L) + r_{12}(Y_A - Y_L) + r_{13}(Z_A - Z_L) \\
 V &= r_{21}(X_A - X_L) + r_{22}(Y_A - Y_L) + r_{23}(Z_A - Z_L) \\
 W &= r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(Z_A - Z_L)
 \end{aligned} \tag{.2.5}$$

e

$$\begin{aligned}
 dx &= (x - x_0) \\
 dy &= (y - y_0) \\
 r &= \sqrt{(dx)^2 + (dy)^2} \\
 \Delta_X &= (dx r^2 K_1) + (dx r^4 K_2) + (dx r^6 K_3) + (P_1 ((2 dx)^2 + r^2)) + (2 P_2 dx dy) \\
 \Delta_Y &= (dy r^2 k_1) + (dy r^4 K_2) + (dy r^6 K_3) + (2 P_1 dx dy) + (P_2 ((2 dy)^2 + r^2))
 \end{aligned} \tag{.2.6}$$

.2.1 Derivate parziali sui parametri intrinseci ed estrinseci

Le derivate parziali corrispondenti alle due equazioni di collinearità (.2.3) e (.2.4) sui parametri intrinseci ed estrinseci risultano essere:

$$\begin{aligned}
 \partial_{X_L}(F) &= -\left(\frac{f}{W} (r_{11} - \frac{U}{W} r_{31})\right) \\
 \partial_{Y_L}(F) &= -\left(\frac{f}{W} (r_{12} - \frac{U}{W} r_{32})\right) \\
 \partial_{Z_L}(F) &= -\left(\frac{f}{W} (r_{13} - \frac{U}{W} r_{33})\right) \\
 \partial_f(F) &= \frac{U}{W} \\
 \partial_\omega(F) &= \frac{f}{W} (\partial_\omega(U) - (\partial_\omega(W)) \frac{U}{W}) \\
 \partial_\phi(F) &= \frac{f}{W} (\partial_\phi(U) - (\partial_\phi(W)) \frac{U}{W}) \\
 \partial_\kappa(F) &= f \frac{V}{W}
 \end{aligned}$$

e

$$\begin{aligned}
\partial_{X_L}(G) &= -\left(\frac{f}{W}(r_{21} - \frac{V}{W}r_{31})\right) \\
\partial_{Y_L}(G) &= -\left(\frac{f}{W}(r_{22} - \frac{V}{W}r_{32})\right) \\
\partial_{Z_L}(G) &= -\left(\frac{f}{W}(r_{23} - \frac{V}{W}r_{33})\right) \\
\partial_f(G) &= \frac{V}{W} \\
\partial_\omega(G) &= \frac{f}{W}(\partial_\omega(V) - (\partial_\omega(W))\frac{V}{W}) \\
\partial_\phi(G) &= \frac{f}{W}(\partial_\phi(V) - (\partial_\phi(W))\frac{V}{W}) \\
\partial_\kappa(G) &= -f\frac{U}{W}
\end{aligned} \tag{.2.7}$$

con

$$\begin{aligned}
\partial_\omega(U) &= -r_{13}(Y_A - Y_L) + r_{12}(Z_A - Z_L) \\
\partial_\omega(V) &= -r_{23}(Y_A - Y_L) + r_{22}(Z_A - Z_L) \\
\partial_\omega(W) &= -r_{33}(Y_A - Y_L) + r_{32}(Z_A - Z_L) \\
\partial_\phi(U) &= -(Z_A - Z_L)\cos(\kappa)\cos(\omega)\cos(\phi) + (Y_A - Y_L)\cos(\kappa)\cos(\phi)\sin(\omega) - \\
&\quad - (X_A - X_L)\cos(\kappa)\sin(\phi) \\
\partial_\phi(V) &= -(Z_A - Z_L)\cos(\omega)\cos(\phi)\sin(\kappa) - (Y_A - Y_L)\cos(\phi)\sin(\kappa)\sin(\omega) + \\
&\quad + (X_A - X_L)\sin(\kappa)\sin(\phi) \\
\partial_\phi(W) &= -(Z_A - Z_L)\cos(\kappa)\sin(\phi) + (Y_A - Y_L)\sin(\phi)\sin(\omega) + \\
&\quad + (X_A - X_L)\cos(\phi)
\end{aligned} \tag{.2.8}$$

.2.2 Derivate parziali sui parametri ottici

Le derivate parziali corrispondenti alle due equazioni di collinearit  (2.3) e (2.4) sui parametri ottici risultano invece essere:

$$\begin{aligned}
\partial_{K_1}(\Delta_X) &= dx (dx^2 + dy^2) \\
\partial_{K_2}(\Delta_X) &= dx (dx^2 + dy^2)^2 \\
\partial_{K_3}(\Delta_X) &= dx (dx^2 + dy^2)^3 \\
\partial_{P_1}(\Delta_X) &= 3 (dx)^2 + (dy)^2 \\
\partial_{P_2}(\Delta_X) &= 2 dx dy \\
\partial_{K_1}(\Delta_Y) &= dy (dx^2 + dy^2) \\
\partial_{K_2}(\Delta_Y) &= dy (dx^2 + dy^2)^2 \\
\partial_{K_3}(\Delta_Y) &= dy (dx^2 + dy^2)^3 \\
\partial_{P_1}(\Delta_Y) &= 2 dx dy \\
\partial_{P_2}(\Delta_Y) &= 3 (dy)^2 + (dx)^2
\end{aligned} \tag{.2.9}$$

e

$$\begin{aligned}
\partial_{x_0}(F) &= -1 - 6 P_1 dx - 2 K_1 dx^2 - K_1 (dx^2 + dy^2) - 4 K_2 dx^2 (dx^2 + dy^2) - \\
&\quad - K_2 (dx^2 + dy^2)^2 - 6 K_3 dx^2 (dx^2 + dy^2)^2 - K_3 (dx^2 + dy^2)^3 - 2 P_2 dy \\
\partial_{y_0}(F) &= -2 P_2 dx - 2 P_1 dy - 2 K_1 dx dy - 4 K_2 dx dy (dx^2 + dy^2) - \\
&\quad - 6 K_3 dx dy (dx^2 + dy^2)^2 \\
\partial_{x_0}(G) &= -2 P_2 dx - 2 P_1 dy - 2 K_1 dx dy - 4 K_2 dx dy (dx^2 + dy^2) - \\
&\quad - 6 K_3 dx dy (dx^2 + dy^2)^2 \\
\partial_{y_0}(G) &= -1 - 2 P_1 dx - K_1 (dx^2 + dy^2) - K_2 (dx^2 + dy^2)^2 - K_3 (dx^2 + dy^2)^3 - \\
&\quad - 6 P_2 dy - 2 K_1 dy^2 - 4 K_2 dy^2 (dx^2 + dy^2) - 6 K_3 dy^2 (dx^2 + dy^2)^2
\end{aligned} \tag{.2.10}$$

Appendice B

.1 Introduzione

In questa prima sezione richiamiamo alcuni elementi di algebra lineare propedeutici alla trattazione svolta nel resto della tesi. Queste considerazioni sono molto importanti in quanto i programmi sviluppati sono stati fatti in MATLAB, linguaggio di programmazione molto potente ma dalla trattazione numerica complessa. Verranno quindi prima analizzati i problemi relativi al calcolo matriciale con MATLAB, per poi illustrare le soluzioni ed i rimedi trovati.

.2 Problematiche numeriche per la risoluzione tramite bundle adjustment

Per poter risolvere il problema del *bundle adjustment* con il metodo di risoluzione dei *minimi quadrati*, dobbiamo prendere in considerazione molte problematiche che potrebbero sorgere nei vari calcoli che dobbiamo fare.

In primo luogo dobbiamo prendere atto che il programma che usiamo per fare i conti, in questo caso MATLAB, non ha precisione infinita. Visto che il calcolo delle derivate parziali nel *Bundle Adjustment* porta molte volte a numeri molto piccoli, molto inferiori alla zero e che quindi la moltiplicazione di due numeri così fatti porta ad un risultato ancora più piccolo, dobbiamo stare attenti a non uscire dalla precisione

di macchina che, ricordiamo, risulta essere ($2.220044 * 10^{-16}$). Se, comunque, dovesse succedere, potremmo non avere conti affidabili con conseguente incongruenza dei dati.

Per risolvere questo problema, ci vengono incontro due funzioni MATLAB. La prima é la funzione *rank*, che ci permette di calcolare il rango della matrice A . Se il rango di A fosse pieno, non ci sarebbero problemi a calcolare l'inverso della matrice, evitando dunque che la matrice sia singolare. La seconda é la piú importante funzione *rref* che ci permette di definire se il sistema é sovradeterminato, confrontando il rango della matrice con i suoi valori singolari.

Se i valori equivalgono, il sistema é risolvibile ed ha un'unica soluzione. Se, invece, i valori non coincidessero, in questo caso sarebbe piú opportuno riformulare tutto il sistema togliendo i parametri di troppo.

Infine, sarebbe sempre opportuno risolvere il sistema dei minimi quadrati con la funzione *pinv* di MATLAB, soprattutto nel calcolo di sistemi matriciali molto grandi e, quindi, col alcuni valori singolari molto vicini allo zero, che unisce i vantaggi della fattorizzazione SVD (con i valori singolari nella diagonale principale), con i vantaggi della *pseudoinversa di Moore-Penrose*.

.3 Elementi di analisi delle matrici

Cominciamo a vedere alcune importanti proprietà delle matrici che abbiamo usato nei nostri programmi.

.3.1 Inversa di una matrice

Una matrice A quadrata di ordine n si dice *invertibile* (o *regolare* o *non singolare*) se esiste una matrice B quadrata di ordine n tale che $AB = BA = I$. B viene chiamata *matrice inversa* di A e viene indicata con A^{-1} . Una matrice non invertibile verrà detta *singolare*.

Se A é invertibile anche la sua inversa lo é e $(A^{-1})^{-1} = A$. Inoltre, se A e B sono due matrici invertibili di ordine n anche AB é invertibile e si ha $(AB)^{-1} = B^{-1}A^{-1}$.

Possiamo anche dire che una matrice quadrata é invertibile se e solo se i suoi vettori colonna sono linearmente indipendenti. Chiamiamo *trasposta* di una matrice $A \in R^{m \times n}$ la matrice $n \times m$, denotata con A^T , ottenuta scambiando tra di loro le righe e le colonne di A .

.3.2 Traccia e determinante

Consideriamo una matrice A quadrata di ordine n . La *traccia* di una matrice é la somma degli elementi diagonali di A , ossia $tr(A) = \sum_{i=1}^n a_{ii}$.

Si dice *determinante* di A lo scalare definito dalla seguente formula:

$$\det(A) = \sum_{\pi} \text{sign}(\pi) a_{1\pi_1} a_{2\pi_2} \dots a_{n\pi_n}$$

essendo $P = (\pi = (\pi_1, \dots, \pi_n)^T)$ l'insieme degli $n!$ vettori ottenuti permutando il vettore degli indici $i = (1, \dots, n)^T$ e $\text{sign}(\pi)$ uguale ad 1 (rispettivamente -1) se serve un numero pari (rispettivamente dispari) di scambi per ottenere π da 1.

Valgono inoltre le seguenti proprietá:

$$\det(A) = \det(A^T), \quad \det(AB) = \det(A)\det(B), \quad \det(A^{-1}) = \frac{1}{\det(A)}$$

Se inoltre due righe o due colonne di una matrice sono linearmente indipendenti, il determinante é nullo, mentre lo scambio di due righe (o di due colonne) in una matrice provoca un cambio di segno del determinante della stessa. Ovviamente il determinante di una matrice diagonale é dato semplicemente dal prodotto degli elementi diagonali.

.3.3 Rango di una matrice

Sia A una matrice rettangolare $m \times n$. Chiamiamo determinante di ordine q estratto dalla matrice A , il determinante di ogni matrice quadrata di ordine q ottenuta da A per la soppressione di $m - q$ righe e $n - q$ colonne.

Si dice *rango* di A (e lo si denota con $\text{rank}(A)$) l'ordine massimo dei determinanti non nulli estratti da A . Una matrice si dice di *rango completo* o *pieno* se $\text{rank}(A) = \min(m, n)$.

Osserviamo come il rango di A esprima il massimo numero di vettori colonna di A linearmente indipendenti, ossia la dimensione del *range* o *immagine* di A , definito come

$$\text{range}(A) = \{y \in R^m : y = Ax \text{ per } x \in R^n\}$$

A rigore si dovrebbe parlare di rango di A per colonne, per distinguerlo dal rango di A per righe dato dal massimo numero di vettori riga di A linearmente indipendenti. Si può però dimostrare che il rango per righe ed il rango per colonne sono uguali.

.3.4 Norme matriciali

Una *norma matriciale* è un'applicazione $\| \cdot \| : R^{m \times n} \rightarrow R$ tale che:

$$\| A \| \geq 0 \forall A \in R^{m \times n} \text{ e } \| A \| = 0 \text{ se e solo se } A = 0$$

$$\| \alpha A \| = |\alpha| \| A \| \forall \alpha \in R, \forall A \in R^{m \times n}$$

$$\| A + B \| \leq \| A \| + \| B \| \forall A, B \in R^{m \times n}$$

Diciamo che una norma matriciale $\| \cdot \|$ è sub-moltiplicativa se $\forall A \in R^{m \times n}$ e $\forall B \in R^{m \times q}$

$$\| AB \| \leq \| A \| \| B \| \tag{.3.1}$$

Importanti esempi di norme matriciali indotte sono le cosiddette *norme p* definite come

$$\| A \|_p = \sup_{x \neq 0} \frac{\| Ax \|_p}{\| x \|_p} \quad (.3.2)$$

La norma 1 e la norma ∞ sono dette rispettivamente norma delle *somme per colonna* e norma delle *somme per riga*. Vengono calcolate nel seguente modo:

$$\begin{aligned} \| A \|_1 &= \max_k \sum_{j=1}^n |a_{jk}| \\ \| A \|_\infty &= \max_j \sum_{k=1}^n |a_{jk}| \end{aligned} \quad (.3.3)$$

dove $\| A \|_1$ rappresenta il valore piú alto della *somma delle colonne* della matrice A , mentre $\| A \|_\infty$ rappresenta il valore piú alto della *somma delle righe* della matrice A .

.4 Buona posizione e numero di condizionamento di un problema

Si consideri il seguente problema astratto: trovare x tale che

$$F(x, d) = 0 \quad (.4.1)$$

dove d é l'insieme dei dati da cui dipende la soluzione ed F esprime la relazione funzionale tra x e d . A seconda del tipo del problema rappresentato nella (.4.1), le variabili x e d potranno esprimere numeri reali, vettori o funzioni. Tipicamente (.4.1) viene detto *problema diretto* se F e d sono dati ed x é incognito, *problema inverso* se F e x sono noti e d é incognito, *problema di identificazione* nel caso in cui x e d sono dati, mentre la relazione funzionale F é incognita.

Il problema (.4.1) si dice *ben posto* se ammette un'unica soluzione x per la quale dipende con continuità sui dati. Un problema che non goda della proprietà precedente si dice *mal posto* o *instabile* e prima di affrontarne la soluzione numerica é bene, quando ha senso, *regolarizzarlo* ovvero trasformarlo in modo opportuno in un problema ben posto. Non é infatti appropriato pretendere che sia il metodo numerico a porre rimedio alle patologie di un problema intrinsecamente mal posto.

La *dipendenza continua* dai dati significa che piccole perturbazioni sui dati debbano riflettersi in piccole variazioni sulla soluzione x . Precisamente, indicando con δd una perturbazione ammissibile sui dati e con δx la conseguente variazione nella soluzione in modo che si abbia

$$F(x + \delta x, d + \delta d) = 0 \quad (.4.2)$$

allora

$$\forall \eta > 0, \exists K(\eta, d) : \|\delta d\| < \eta \Rightarrow \|\delta x\| \leq K(\eta, d) \|\delta d\| \quad (.4.3)$$

La norma usata per i dati e quella per la soluzioni possono non coincidere qualora x e d denotino diversi tipi di variabile.

Per il problema (.4.1) definiamo come *numero di condizionamento relativo* il numero

$$K(d) = \sup_{\delta d \in D} \frac{\frac{\|\delta x\|}{\|x\|}}{\frac{\|\delta d\|}{\|d\|}} \quad (.4.4)$$

dove D é un intorno dell'origine ed indica l'insieme delle perturbazioni ammissibili sui dati in corrispondenza dei quali il problema perturbato (.4.2) ha ancora senso. Nel caso in cui $d = 0$ o $x = 0$, sará necessario introdurre il *numero di condizionamento assoluto* dato da

$$K_{abs}(d) = \sup_{\delta d \text{ in } D} \frac{\|\delta x\|}{\|\delta d\|} \quad (.4.5)$$

Il problema (.4.1) si dirá *mal condizionato* se $K(d)$ é grande in corrispondenza di ogni dato d ammissibile. La proprietá di buon condizionamento di un problema é indipendente dal metodo numerico che verrá usato per risolverlo.

Anche nel caso in cui il numero di condizionamento sia infinito, non é necessariamente vero che il problema sia mal posto: é la formulazione del problema ad essere mal posta. In effetti esistono problemi ben posti in corrispondenza dei quali il numero di condizionamento é infinito, ma tali da poter essere riformulati in problemi equivalenti con un numero di condizionamento finito.

.4.1 Stabilitá dei metodi numerici

Un metodo numerico per la risoluzione approssimata di (.4.1) consisterá, in generale, nel costruire una successione di problemi approssimati

$$F_n(x_n, d_n) = 0 \quad n \geq 1 \quad (.4.6)$$

dipendenti da un certo parametro n , con la speranza che $x_n \rightarrow x$ per $n \rightarrow \infty$, ovvero che la soluzione numerica converga alla soluzione esatta. Affinché questo avvenga, é necessario che $d_n \rightarrow d$ e che F_n approssimi F quando $n \rightarrow \infty$. Precisamente, se il dato d del problema (.4.1) é ammissibile per F_n , si dice che (.4.7) é *consistente* se

$$F_n(x, d) = F_n(x, d) - F(x, d) \rightarrow 0 \quad \text{per } n \rightarrow \infty \quad (.4.7)$$

essendo x la soluzione del problema (.4.1) corrispondente al dato d .

Un metodo é detto *fortemente consistente* se $F_n(x, d) = 0$ per ogni valore di n e non solo per $n \rightarrow \infty$.

Piú in generale *non* risulteranno fortemente consistenti tutti i metodi numerici ottenuti dal modello matematico per troncamento di operazioni di passaggio al limite.

.5 Origine degli errori

Se il problema numerico (.4.7) costituisce l'approssimazione del problema matematico (.4.1) e quest'ultimo a sua volta modella un problema fisico (che indichiamo con PF), diremo che (.4.7) é un *modello computazionale per PF*. In questo processo l'errore globale, indicato con e , é espresso dalla differenza fra la soluzione effettivamente calcolata, \bar{x}_n , e la soluzione fisica, x_f , di cui x fornisce il modello. In tale prospettiva, e puó essere visto come la somma dell'errore e_m del modello matematico, espresso da $x - x_f$, con l'errore e_c del modello computazionale, $\bar{x}_n - x$, ovvero $e = e_m + e_c$.

Si generano naturalmente *errori* dovuti alla *modellazione matematica* del problema fisico in esame, oltre a quelli di cui sono affetti i *dati sperimentali*. Ci limiteremo ad osservare che gli *errori nei dati* possono essere ridotti ricorrendo a procedure statistiche od a tecniche di filtraggio volte ad eliminare il rumore di cui sono affetti i dati stessi.

Vi sono poi i cosiddetti *errori di troncamento* allorché un problema matematico (in dimensione infinita) viene trasformato in uno di dimensione finita (approssimando dunque ogni operazione di limite). Infine gli *errori di arrotondamento* sono dovuti al fatto che i numeri reali hanno una rappresentazione sul calcolatore con un numero finito di cifre.

Sia gli *errori di troncamento* che quelli di *arrotondamento* sono oggetto di studio nel calcolo numerico. In particolare, un obiettivo dell'analisi di ogni metodo numerico é la stima dell'errore di approssimazione, ovvero della distanza fra la soluzione numerica calcolata e la soluzione esatta del problema matematico in esame. Tale distanza é dovuta sia agli errori di troncamento sia a quelli di arrotondamento.

In generale, possiamo quindi pensare di individuare i seguenti *tipi di errore*:

1. *errori dovuti al modello*, controllabili curando la costruzione del modello matematico;
2. *errori nei dati*, riconducibili aumentando l'accuratezza nelle misurazioni dei dati stessi;
3. *errori di troncamento*, dovuti al fatto che nel modello numerico le operazioni di passaggio al limite vengono approssimate con operazioni che richiedono un numero finito di passi;
4. *errori di arrotondamento*.

Gli errori nei punti 3 e 4 costituiscono l'*errore computazionale*. Un metodo numerico sar  dunque convergente su questo errore pu  essere arbitrariamente ridotto aumentando lo sforzo computazionale. Ovviamente, seppur primario, la convergenza non   l'unico obiettivo di un metodo numerico, dovendosi coniugare all'*accuratezza*, all'*affidabilit * ed all'*efficienza*.

.5.1 Accuratezza

L'*accuratezza* esprime il fatto che gli errori siano piccoli rispetto ad una tolleranza fissata. Essa in genere si quantifica attraverso l'ordine di infinitesimo dell'errore e_n rispetto al parametro caratteristico della discretizzazione. A questo proposito notiamo che la *precisione* del sistema utilizzato per la rappresentazione dei numeri su un calcolatore nel modello numerico non limita, da un punto di vista teorico, l'accuratezza.

.5.2 Affidabilità

Affidabilità significa che l'errore totale può essere assicurato al di sotto di una certa tolleranza con un margine di probabilità superiore ad una percentuale prestabilita. Evidentemente un modello numerico sarà affidabile soltanto se adeguatamente *verificato*, ossia applicato con successo a numerosi casi test.

.5.3 Efficienza

Efficienza significa infine che la complessità computazionale necessaria per controllare tale errore (ovvero quantità di operazioni impiegate e la dimensione della memoria richiesta) sia la più bassa possibile.

Con a numero reale ed a^* una sua approssimazione, possiamo distinguere due particolari tipi di errore:

- *errore assoluto*, la quantità $|a - a^*|$;
- *errore relativo* la quantità $\frac{|a - a^*|}{|a|}$.

.6 Rappresentazione dei numeri sul calcolatore

Ogni operazione effettuata su un calcolatore risulta affetta da errori di *arrotondamento* (detti errori di *roundoff*). Essi sono dovuti al fatto che su un calcolatore può essere rappresentato solo un sottoinsieme finito dell'insieme dei numeri reali.

.6.1 Aritmetica IEC/IEEE

La possibilità di costruire insiemi di numeri *floating-point* diversi fra loro per base, numero di cifre significative e range dell'esponente, ha dato luogo in passato ad una proliferazione di sistemi numerici. Per porvi rimedio è stato fissato uno standard che oggi viene quasi universalmente riconosciuto. Esso è stato definito e sviluppato

nel 1985 dall'*Institute of Electrical and Electronics Engineers* (in breve, IEEE) come *IEEE standard for binary floating arithmetic* e riferito come standard IEEE754. Specifica il formato, le operazioni, le conversioni fra i diversi formati floating point e quelle tra i diversi sistemi di numerazione ed il trattamento delle eccezioni. É stato successivamente approvato nel 1989 dall'*International Electronical Commission* (IEC) come standard internazionale IEC559 e con tale denominazione é attualmente riconosciuto. L'IEC559 prevede due formati per i numeri *floating-point*: un *formato di base*, costituito dai sistemi $F(2,24,-125,128)$ per la singola precisione, e $F(2,53,-1021,1024)$ per la doppia precisione, entrambi comprendenti anche i numeri denormalizzati, ed un *formato esteso* per il quale vengono solo fissati dei limiti.

Per concludere, notiamo che in IEC559 non tutte le sequenze di *bit* corrispondono ad un numero reale. Nella tabella seguente riportiamo alcune codifiche particolari che corrispondono ai valori ± 0 , $\pm \infty$ ed ai cosiddetti *non numeri* (brevemente *NaN*, dall'inglese *not a number*), corrispondenti ad esempio a $0/0$ o ad altre quantità generate da operazioni eccezionali.

valore	esponente	mantissa
± 0	$L - 1$	0
$\pm \infty$	$U + 1$	0
<i>NaN</i>	$U + 1$	0

Tabella 1: Codifiche particolari IEC559

.6.2 Arrotondamento di un numero reale nella sua rappresentazione di macchina

Su un calcolatore é disponibile soltanto un sottoinsieme $F(\beta, t, L, U)$ di R e ciò pone alcuni problemi pratici, primo fra tutti quello relativo alla rappresentazione in F di un numero reale x qualsiasi. D'altra parte, anche se x ed y fossero due numeri di

F , il risultato di un'operazione su di essi potrebbe non appartenere ad F . Bisognerà dunque definire un'aritmetica anche su F .

L'approccio piú semplice per risolvere il primo problema consiste nell'arrotondare $x \in R$ in modo che il numero arrotondato appartenga ad F . Dato $x \in R$ in notazione posizionale normalizzata, sostituiamo ad x il rappresentante $fl(x)$ in F , definito come

$$fl(x) = (-1)^s(0.a_1a_2\dots\tilde{a}_t)\beta^e \quad (.6.1)$$

$$\tilde{a}_t = \begin{cases} a_t & \text{se } a_{t+1} < \frac{\beta}{2} \\ a_t + 1 & \text{se } a_{t+1} \geq \frac{\beta}{2} \end{cases} \quad (.6.2)$$

L'applicazione $fl : R \rightarrow F$ é la piú comunemente utilizzata ed é detta *rounding*. Evidentemente $fl(x) = x$ se $x \in F$ ed inoltre $fl(x) \leq fl(y)$ se $x \leq y \forall x, y \in R$ (proprietá di monotonia).

.6.3 Overflow ed Underflow

Quanto finora scritto vale soltanto per i numeri che hanno esponente e che appartiene al *range* di F . Se infatti $x \in (-\infty, -x_{max}) \cup (x_{max}, \infty)$ il valore $fl(x)$ non sarebbe definito, mentre se $x \in (-x_{min}, x_{min})$ l'operazione di *rounding* é comunque definita. Nel primo caso, qualora x sia il risultato di un'operazione su numeri di F , si parla di situazione di *overflow*, nel secondo caso di *underflow* (o di *graceful underflow* se sono presenti numeri denormalizzati). L'*overflow* viene gestito dal sistema con un'interruzione delle operazioni in esecuzione.

Precisione macchina

Se $x \in R$ appartiene al range di F , allora

$$fl(x) = x(1 + \delta) \text{ con } |\delta| \leq u \quad (.6.3)$$

essendo

$$u = \frac{1}{2}\beta^{1-t} = \frac{1}{2}\epsilon_M \quad (.6.4)$$

la cosiddetta *unità di roundroff* (o precisione macchina).

Precisione macchina in Matlab

In ambiente Matlab é possibile ottenere il valore di ϵ_M attraverso la variabile di sistema *eps* pari a

```
>> eps
ans =
    2.220446049250313e-016
```

Figura 13: La variabile di sistema MATLAB *eps*

.6.4 Operazioni di macchina effettuate in virgola mobile

Sull'insieme dei numeri macchina é definita un'aritmetica analoga, per quanto possibile, all'aritmetica su R . Data una qualsiasi operazione aritmetica $\circ : RxR \rightarrow R$ a due operandi in R , indicheremo con \odot la corrispondente operazione macchina.

$$\odot : F x F \rightarrow F, \quad x \odot y = fl(fl(x) \circ fl(y)) \quad (.6.5)$$

Dalle proprietà dei numeri *floating-point* ci si può aspettare che per le operazioni aritmetiche a due operandi, quando ben definite, valga la seguente proprietà:

$$\forall x, y \in F, \exists \delta \in R : \quad x \odot y = (x \circ y)(1 + \delta) \quad \text{con } |\delta| \leq u \quad (.6.6)$$

Di fatto, per poter soddisfare la (.6.6) nel caso in cui \circ sia l'operatore di sottrazione, si richiederá un'ipotesi addizionale sulla struttura dei numeri di F , vale a dire la presenza della cosiddetta cifra di guardia. In particolare, se \circ é l'operatore di somma, si trova che $\forall x, y \in R$

$$\frac{|x \oplus y - (x + y)|}{|x + y|} \leq u(1 + u) \frac{|x| + |y|}{|x + y|} + u \quad (.6.7)$$

e quindi l'errore relativo associato alla somma sará piccolo, a meno che $x + y$ non lo sia a sua volta. Un cenno a parte merita perció il caso della somma fra due numeri vicini in modulo, ma opposti di segno: in tal caso infatti $x + y$ puó essere molto piccolo senza che i due numeri lo siano, generando i cosiddetti *errori di cancellazione*.

É importante osservare che, accanto a proprietá dell'aritmetica classica che si conservano nell'aritmetica *floating-point*, altre vanno perse. Un esempio é dato dall'associativitá dell'addizione: si dimostra infatti che in generale

$$x \oplus (y \oplus z) \neq (x \oplus y) \oplus z$$

Indicheremo con *flop* la singola operazione elementare *floating-point*, somma, sottrazione, moltiplicazione o divisione. Con la convenzione precedentemente adottata, un prodotto scalare tra due vettori di lunghezza n richiederá $2n-1$ *flops*, un prodotto matrice-vettore $(2m-1)n$ *flops* se la matrice é $n \times m$ ed infine un prodotto matrice-matrice necessiterá di $(2r-1)mn$ *flops* se le due matrici sono $m \times r$ ed $r \times n$ rispettivamente.

Aritmetica IEC559

Lo standard IEC559 provvede anche a definire un'aritmetica chiusa su F , nel senso che ogni operazione in essa produce un risultato rappresentabile all'interno del sistema stesso, non necessariamente coincidente con quello matematicamente atteso.

eccezione	esempi	risultato
operazione non valida	$\frac{0}{0}$	NaN
overflow		$\pm\infty$
divisione per zero	$\frac{1}{0}$	$\pm\infty$
underflow		numeri sottonormali

Tabella 2: Risultati per alcune operazioni eccezionali

La presenza di un NaN (*Not a Number*) in una sequenza di operazioni comporta automaticamente che il risultato sia un NaN . L'accettazione di questo standard é attualmente in fase di attuazione.

Segnaliamo che non tutti i sistemi *floating-point* soddisfano la (.6.6). Uno dei motivi principali é legato all'assenza della *cifra di guardia* (o *round digit*) nella sottrazione, ovvero di un extra bit che si attivi a livello di mantissa quando si esegue la sottrazione fra due numeri *floating-point*. Un'aritmetica per la quale accade questo é detta *aberrante*. In alcuni calcolatori non esiste la cifra di guardia (si preferisce infatti privilegiare la velocità di calcolo), anche se la tendenza attuale é di adottare addirittura due cifre di guardia.

.7 Analisi di stabilità del problema numerico

Come abbiamo visto prima, abbiamo bisogno che il nostro problema sia ben posto per far sì che ammetta un'unica soluzione e che piccole perturbazioni sui dati influiscano in minima parte sulla soluzione del problema.

In questa sezione intendiamo studiare la stabilità della soluzione del sistema lineare $Ax = b$ rispetto alle perturbazioni sui dati A e b . A tale scopo introduciamo il nuovo sistema reale da risolvere

$$(A + \delta A)(x + \delta x) = b + \delta b \quad (.7.1)$$

che chiameremo *sistema perturbato* in quanto tutti i dati della matrice A sono stati arrotondati (per via dell'aritmetica finita della macchina). Naturalmente δA è una matrice e δb è un vettore contenente i coefficienti di perturbazione.

Il problema è quello di stabilire come dipende δx da δA e δb . In questa analisi non si tiene conto degli errori di arrotondamento introdotti dal processo di calcolo della soluzione $x + \delta x$.

Sia $\|v\|$ una norma di vettore e $\|A\|$ la norma matriciale introdotta. Per ogni A invertibile definiamo

$$K(A) = \|A\| \|A^{-1}\| \quad (.7.2)$$

come suo *numero di condizionamento*.

Se $\|\delta A\| < \frac{1}{\|A^{-1}\|}$, allora abbiamo il seguente risultato:

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{K(A)}{1 - K(A) \frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right) \quad (.7.3)$$

Se $K(A) \cong 1$ si dice che A è *ben condizionato* e, di conseguenza, il sistema $Ax = b$ si dice *ben condizionato*. In tal caso piccole perturbazioni su A e b determinano piccole variazioni sulla soluzione del sistema. Se invece $K(A) \gg 1$ si dice che A è *mal condizionato* e si registra una potenziale *instabilità nella risoluzione del sistema* per effetto della perturbazione nei dati. In altri termini, se A è mal condizionata, piccole perturbazioni di A e/o b possono introdurre un grande errore sulla soluzione x .

Comunque, il buon condizionamento non basta a garantire risultati accurati nella risoluzione di un sistema: sarà infatti importante, come già detto, usare algoritmi stabili. Viceversa, il fatto che una matrice sia mal condizionata non esclude che per particolari scelte del termine noto il sistema risulti complessivamente ben condizionato.

Un caso particolare risulta quando $\delta A = 0$. Allora

$$\frac{1}{K(A)} \frac{\|\delta b\|}{\|b\|} \leq \frac{\|\delta x\|}{\|x\|} \leq K(A) \frac{\|\delta b\|}{\|b\|} \quad (.7.4)$$

Per poter impiegare la disuguaglianza (.7.3) nell'analisi della propagazione degli errori di arrotondamento per i metodi diretti, $\|\delta A\|$ e $\|\delta b\|$ dovranno essere stimati in funzione della dimensione del sistema e delle caratteristiche dell'aritmetica *floating-point* usata.

È infatti ragionevole aspettarsi che le perturbazioni indotte da un metodo per la risoluzione di un sistema lineare siano tali che $\|\delta A\| \leq \gamma \|A\|$ e $\|\delta b\| \leq \gamma \|b\|$, essendo γ un numero positivo che dipende da u , l'unità di *roundoff*. In tal caso, la (.7.3) può essere completata nel seguente modo.

Siano $\delta A \in R^{n \times n}$ e $\delta b \in R^n$ tali che $\|\delta A\| \leq \gamma \|A\|$, $\|\delta b\| \leq \gamma \|b\|$ per un opportuno $\gamma \in R^+$. Allora, se $\gamma K(A) < 1$ si ha

$$\frac{\|x + \delta x\|}{\|x\|} \leq \frac{1 + \gamma K(A)}{1 - \gamma K(A)} \quad (.7.5)$$

e

$$\begin{aligned} \frac{\|x + \delta x\|}{\|x\|} &\leq \frac{\|x\|}{\|x\|} + \frac{\|\delta x\|}{\|x\|} \leq 1 + \frac{k(A)}{1 - K(A) \frac{\|\delta A\|}{\|A\|}} \left(\frac{\gamma \|b\|}{\|b\|} + \frac{\gamma \|A\|}{\|A\|} \right) \\ &= \frac{1 - \gamma K(A) + 2\gamma K(A)}{1 - \gamma K(A)} \quad (.7.6) \end{aligned}$$

quindi

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{2\gamma}{1 - \gamma K(A)} K(A) \quad (.7.7)$$

Se $K(A) \gg 1$ l'analisi del residuo può non essere indicativa dell'errore commesso sulla soluzione. Ad esempio, indichiamo con \bar{x} la soluzione calcolata e con $\bar{r} =$

$b - A\bar{x}$ il residuo ad essa associato. Se $\|x - \bar{x}\|$ é piccolo, allora $\|\bar{r}\|$ é piccolo. Naturalmente l'implicazione opposta sarebbe molto piú utile. A tale riguardo si ha

$$\|x - \bar{x}\| = \|A^{-1}b - A^{-1}(b - \bar{r})\| = \|A^{-1}\bar{r}\| \leq K(A) \frac{\|\bar{r}\|}{\|A\|} \quad (.7.8)$$

quindi

$$\frac{\|x - \bar{x}\|}{\|x\|} \leq K(A) \|\bar{r}\| \frac{1}{\|A\| \|x\|} \leq K(A) \frac{\|\bar{r}\|}{\|b\|} \quad (.7.9)$$

Ad un residuo piccolo potrebbe quindi corrispondere un errore grande sulla soluzione. É dunque opportuno cercare metodi stabili per la risoluzione del problema lineare.

.8 Sistemi indeterminati

In questa sezione diamo un significato alla soluzione di un sistema lineare rettangolare $Ax = b$ con $A \in R^{m \times n}$ sia nel caso *sovradeterminato* nel quale $m > n$, sia nel caso *sottodeterminato* corrispondente a $m < n$.

Questa é la soluzione che maggiormente ci interessa in quanto é quella che usiamo nella risoluzione del *bundle adjustment*.

Facciamo notare che in generale un sistema sovradeterminato non ha soluzione a meno che b non sia un elemento del $range(A)$.

Data $A \in R^{m \times n}$ con $m \geq n$, $b \in R^m$ diciamo che $x^* \in R^n$ é la soluzione del sistema lineare $Ax = b$ nel senso dei *minimi quadrati* se

$$\Phi(x^*) = \|x^* - b\|_2^2 \leq \min_{x \in R^n} \|Ax - b\|_2^2 = \min_{x \in R^n} \Phi(x) \quad (.8.1)$$

Si tratta dunque di minimizzare il residuo della norma euclidea. La soluzione del problema (.8.1) puó essere ottenuta imponendo che il gradiente della funzione Φ definita nella (.8.1) si annulli in x^* . Essendo

$$\Phi(x) = (Ax - b)^T(Ax - b) = x^T A^T A x - 2x^T A^T b + b^T b \quad (.8.2)$$

si trova allora

$$\Delta\Phi(x^*) = 2A^T A x^* - 2A^T b = 0 \quad (.8.3)$$

da cui si deduce che x^* deve essere la soluzione del sistema (quadrato)

$$A^T A x^* = A^T b \quad (.8.4)$$

noto come *sistema delle equazioni normali*. Tale sistema é non singolare se A ha rango pieno. In tal caso, la soluzione nel senso dei minimi quadrati esiste ed é unica.

Per risolvere il sistema (.8.4), essendo $B = A^T A$ simmetrica e definita positiva, si potrebbe usare la *fattorizzazione di Cholesky*. Tuttavia, il sistema (.8.4) é solitamente mal condizionato; inoltre, a causa degli errori di arrotondamento, nel calcolo di $A^T A$ possono andare perdute cifre significative con conseguente perdita della definita positività o addirittura della non singolarità della matrice.

É quindi in generale piú conveniente impiegare per la risoluzione del sistema un altro metodo, la fattorizzazione *SVD*, quella che usiamo per i nostri studi.

.8.1 Fattorizzazione Singular Value Decomposition (SVD)

Una qualunque matrice può essere ridotta in forma diagonale tramite pre e post moltiplicazione per matrici unitarie.

Precisamente sia $A \in C^{m \times n}$. Esistono due matrici unitarie $U \in C^{m \times m}$ e $V \in C^{n \times n}$ tali che

$$U^H A V = \Sigma = \text{diag}(\sigma_1, \dots, \sigma_p) \in C^{m \times n} \quad \text{con } p = \min(m, n) \quad \text{e } \sigma_1 \geq \dots \geq \sigma_p \geq 0 \quad (.8.5)$$

La (.8.5) é detta *decomposizione in valori singolari* (o *Singular Value Decomposition* da cui, in breve, *SVD*) di A ed i $\sigma_i(A)$ sono detti i *valori singolari di A* .

Nel caso in cui A sia reale, anche U e V lo saranno e nella (.8.5) U^H andrà sostituita con U^T . Si ha la seguente caratterizzazione

$$\sigma_i(A) = \sqrt{\lambda_i(A^H A)}, \quad i = 1, \dots, n \quad (.8.6)$$

In effetti dalla (.8.5) si ha che $A = U \sum V^H$, $A^H = V \sum U^H$ e quindi, essendo U e V unitarie, $A^H A = V \sum^2 V^H$ ovvero $\lambda_i(A^H A) = \lambda_i(\sum^2) = (\sigma_i(A))^2$.

Essendo AA^H e $A^H A$ matrici Hermitiane, le colonne di U (rispettivamente di V), dette *valori singolari sinistri* di A (rispettivamente destri) sono gli autovettori di AA^H (rispettivamente $A^H A$) e, di conseguenza, non sono definiti in modo univoco.

Dalla (.8.6) si ricava che se $A \in C^{m \times n}$ é hermitiana con autovalori dati da $\lambda_1, \lambda_2, \dots, \lambda_n$, allora i valori singolari di A sono dati dai moduli degli autovalori di A . Infatti, essendo $AA^H A = A^2$, si ha che $\sigma_i = \sqrt{\lambda_i^2} = |\lambda_i|$ per $i = 1, \dots, n$. Per quanto riguarda il rango, se risulta

$$\sigma_1 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_p = 0 \quad (.8.7)$$

allora il rango di A é pari ad r , il nucleo di A é lo spazio generato dai vettori colonna di V (v_{r+1}, \dots, v_n), ed il range di A é lo spazio generato dai vettori colonna di U (u_1, \dots, u_r). Supponiamo infine che il rango di $A \in C^{m \times n}$ sia pari ad r e che A ammetta una *SVD* del tipo $U^H A V = \sum$.

La matrice $A^\dagger = V \sum^\dagger U^H$ é detta *pseudoinversa di Moore-Penrose*, essendo

$$\sum^\dagger = \left(\frac{1}{\sigma_1}, \dots, \frac{1}{\sigma_r}, 0, \dots, 0 \right) \quad (.8.8)$$

A^\dagger viene anche detta *inversa generalizzata di A*. In effetti se $\text{rank}(A) = n < m$, allora $A^\dagger = (A^T A)^{-1} A^T$, mentre se $n = m = \text{rank}(A)$, $A^\dagger = A^{-1}$.

.8.2 Fattorizzazione SVD e pseudoinversa in forma normale

Sia $A \in R^{m \times n}$. Allora l'unica soluzione di (.8.4) é

$$x^* = A^\dagger b \quad (.8.9)$$

essendo A^\dagger la *pseudoinversa di Moore-Penrose* di A .

Utilizzando la *SVD* di A , $A = UV^T \Sigma$, il problema (.8.4) é equivalente a cercare $w = V^T x$ tale che w abbia norma euclidea minima e

$$\| w \Sigma - U^T b \|_2^2 \leq \| y \Sigma - U^T b \|_2^2 \quad \forall y \in R^n \quad (.8.10)$$

Per quanto riguarda la stabilitá del problema, ci limitiamo a far notare che se il rango di A non é pieno, la soluzione x^* non é necessariamente una funzione continua dei dati e piccoli cambiamenti su di essi possono produrne di grandi su x^* .

Nel caso di sistemi sottodeterminati, in cui $m < n$, se A ha rango massimo si può ancora utilizzare la fattorizzazione *QR*. In particolare, se si opera sulla matrice trasposta si ottiene la soluzione del sistema con norma euclidea minima. Se invece la matrice non ha rango massimo, si ricorre nuovamente alla *SVD*.

Nel caso in cui $m = n$, si può evidentemente ancora usare la *SVD* in alternativa al *MEG* (metodo di risoluzione di Gauss) per la risoluzione del sistema lineare $Ax = b$.

.8.3 Fattorizzazione SVD in MATLAB

La *SVD* di una matrice $A \ m \times n$ é data da

$$A = USV^T \quad (.8.11)$$

dove U é una matrice ortogonale $m \times m$ e V é una matrice $n \times n$. Se la matrice A é complessa, allora U e V sono matrici unitarie. In tutti i casi S é una matrice diagonale $m \times n$.

Gli elementi diagonali della matrice sono detti *valori singolari* di A . Normalmente sono messi in ordine decrescente, cosí che $s_1 > s_2 > s_3 > \dots > s_n$.

$$\begin{bmatrix} s_1 & \dots & \dots & \dots & \dots \\ \vdots & s_2 & \dots & \dots & \dots \\ \vdots & \vdots & s_3 & \dots & \dots \\ \vdots & \vdots & \vdots & \ddots & \dots \\ \vdots & \vdots & \vdots & \vdots & s_n \end{bmatrix} \quad (.8.12)$$

Gli algoritmi per computare gli *SVD* di una matrice sono stati elaborati da *Golub* e *Van Loan* nel 1989.

La *SVD* di una matrice ha molte importanti applicazioni. Prima di tutto possiamo dire che tramite il comando MATLAB *rref* possiamo sapere se il sistema ha un'unica soluzione oppure no. Per fare questo dobbiamo vedere se il rango della matrice equivale al numero dei valori singolari diversi da zero della stessa matrice tramite, appunto, il comando *rref*. Se non fossero uguali dovremmo ridurre i parametri.

Cosí che per una matrice 5×5 di rango 3, s_4 e s_5 dovrebbero essere 0. In pratica, piuttosto che contare i valori singolari diversi da zero, MATLAB determina i ranghi dalla *SVD* contando i numeri piú grandi dei valori singolari rispetto ad alcuni valori di tolleranza. Questo é un approccio piú realistico per determinare il rango.

La *SVD* in MATLAB avviene tramite la funzione omonima.

```

>> help rref

RREF    Reduced row echelon form.
        R = RREF(A) produces the reduced row echelon form of A.

[R,jb] = RREF(A) also returns a vector, jb, so that:
        r = length(jb) is this algorithm's idea of the rank of A,
        x(jb) are the bound variables in a linear system, Ax = b,
        A(:,jb) is a basis for the range of A,
        R(1:r,jb) is the r-by-r identity matrix.

[R,jb] = RREF(A,TOL) uses the given tolerance in the rank tests.

Roundoff errors may cause this algorithm to compute a different
value for the rank than RANK, ORTH and NULL.

```

Figura 14: La funzione di decomposizione MATLAB *rref*

.8.4 Calcolo della pseudoinversa in MATLAB

Un caso problematico che può capitare nel calcolo di sistemi matriciali molto grandi è quando la matrice A non ha rango pieno. In questo caso il sistema dei minimi quadrati

$$A^T A x = A^T b$$

e, di riflesso, il calcolo della pseudoinversa sistema A^\dagger data da:

$$A^\dagger = V(S^T S)^{-1} S^T U^T \quad (.8.13)$$

con V matrice $m \times m$, U matrice $n \times n$ ed S matrice $n \times m$, non è risolvibile in quanto la matrice $S^T S$ non è invertibile a causa dei valori singolari molto vicini ad uguali a 0.

Per risolvere questo problema MATLAB prende solo r valori sigolari diversi da zero della matrice in modo che S sia una matrice $r \times r$ dove r è il rango di A . In questo

```
>> help svd

SVD    Singular value decomposition.
[U,S,V] = SVD(X) produces a diagonal matrix S, of the same
dimension as X and with nonnegative diagonal elements in
decreasing order, and unitary matrices U and V so that
X = U*S*V'.

S = SVD(X) returns a vector containing the singular values.

[U,S,V] = SVD(X,0) produces the "economy size"
decomposition. If X is m-by-n with m > n, then only the
first n columns of U are computed and S is n-by-n.
```

Figura 15: La funzione di decomposizione MATLAB *svd*

modo, con una tolleranza fissata presa a piacere, MATLAB é in grado di risolvere il sistema.

Tutti questi passaggi, compresa la fattorizzazione *SVD* ed il calcolo della *pseudoinversa*, sono fatti in automatico da MATLAB tramite la funzione *pinv*.

```
>> help pinv

PINV   Pseudoinverse.
X = PINV(A) produces a matrix X of the same dimensions
as A' so that A*X*A = A, X*A*X = X and A*X and X*A
are Hermitian. The computation is based on SVD(A) and any
singular values less than a tolerance are treated as zero.
The default tolerance is MAX(SIZE(A)) * NORM(A) * EPS.

PINV(A,TOL) uses the tolerance TOL instead of the default.
```

Figura 16: Il calcolo della pseudoinversa (*pinv*) in MATLAB

Bibliografia

- [1] Exterior orientation parameters, Karsten Jacobsen, Institute for Photogrammetry and GeoInformation, University of Hannover, Germany, 2001.
- [2] Robust camera calibration using 2D to 3D feature correspondences, Fadi Dornai-ka, Christophe Garcia, Insistute for System Design Technology, Sankt Augustin, Germany.
- [3] Image-variant interior orientation and sensor modelling of high-quality digital cameras, H. Hastedt, Th. Luhmann, W. Tecklenburg, Institute for Applied Photogrammetry and GeoInformatics, University of Applied Sciences Oldenburg, Germany.
- [4] A new spatio-temporal matching algorithm for 3D-particle tracking velocimetry, Jochen Willneff, Armin Gruen, Institute of Geodesy and Photogrammetry, Swiss Federal Institute of Technology, Zurich.
- [5] Geometric registration and classification of Hyperspectral airborne pushbroom data, J. S. Bethel, C. Lee, D. A. Landgrebe, Geomatics Area, School of Electrical and Computer Engineering.
- [6] Pose estimation of line cameras using linear features, Young-ran Lee, Ayman Habib, Image and System division, SaTReCi, Korea.

-
- [7] Human body reconstruction from image sequences, Fabio Remondino, Institute of Geodesy and Photogrammetry, ETH Zurich, Switzerland.
- [8] The development of camera calibration methods and models, T. A. Clarke, J. G. Fryer, City University, London.
- [9] An introduction to photogrammetric camera calibration, Horst A. Beyer, Institute of Geodesy and Photogrammetry, ETH Zurich, Switzerland.
- [10] Bundle adjustment - A modern synthesis, B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon, INRIA, Montbonnot, France.
- [11] View frustum optimization to maximize object's image area, Kok-Lim Low, Adrian Ilie, Department of computer science, University of North Carolina at Chapel Hill.
- [12] An introduction to projective geometry, Stan Birchfield, 1998.
- [13] Singular value decomposition and principal component analysis, M. E. Wall, A. Rechtsteiner, L. M. Rocha, System science Ph.D. Program, Portland, Oregon, U.S.A.
- [14] Matematica numerica, A.Quarteroni, R.Sacco, F.Saleri, seconda edizione, Springer, 2000.
- [15] Elementi di calcolo, Alfio Quarteroni.
- [16] Numerical Recipes in C, William H.Press, Saul A.Teukolsky, William T.Vetterling, Brian P,Flannery, Cambridge University Press, 1992.
- [17] Fondamenti di Statistica, Giovanna Togliatti, Edizioni Clup Milano, 1981.

-
- [18] Set theoretic estimation in color scanner characterization, G.Sharma, H.J.Trussell, Electrical and Computer Engineering Dept, North Carolina state university, Raleighm, 1996.
- [19] Solving systems of polynomial equations, D.Manocha, Department of computer science, Univeristy of North Carolina, 1994.
- [20] Fully vision-based Calibration of a Hand-Eye robot, Christophe Garcia, 1999
- [21] A single step calibration procedure for IMU/GPS in aerial photogrammetry, L. Pinto, G. Forlani, 1999
- [22] A Bayesian approach to camera system calibration/spatial intersection, R. J. Valkenburg, 1999
- [23] 3D Modeling of complex environments, Sabry F. El-Hakim, 2001
- [24] Expressive expression mapping with ratio images, Zicheng Liu, Ying Shan, Zhengyou Zhang, 2001
- [25] Robust and rapid generation of Animated Faces from video - Images: A model-based modeling approach, Zhengyou Zhang, Zicheng Liu, Dennis Adler, Michael F.Cohen, Erik Hanson, Ying Shan, 2001
- [26] Model-Based Bundle Adjustment with application to face modeling, Ying Shan, Zicheng Liu, Zhengyou Zhang, Microsoft Research, 2001
- [27] A generalized linear model for principal component analysis of binary data, Andrew I. Schein, Lawrence K. Saul, Lyle H. Ungar, Department of Computer and Information Science

-
- [28] Principal component analysis of remote sensing imagery: effects of additive and multiplicative noise, Brian R. Corner, Ram M. Narayanan, Stephen E. Reichenbach
- [29] Image sequence based automatic multi-camera system calibration techniques, Hans-Gerd Maas.
- [30] Camera Calibration, Zehang Sun, Christian Rayburn, Internship with UNR ee Ford Motor Company, June 2001.
- [31] The design and implementation of PixelFlex: A reconfigurable multi-projector display system, David Gotz.
- [32] Multi-camera calibration, Ismail Oner Sebe, George Qian Chen, AST San Diego Labs.
- [33] Using model-driven Bundle-Adjustment to model heads from raw video sequence, P. Fua, Computer Graphics Lab (LIG), Swiss Federal Institute of Technology.
- [34] A simple technique for self-calibration, Paulo R. S. Mendonca, Roberto Cipolla, Department of Engineering, University of Cambridge.
- [35] Camera Calibration with one-dimensional objects, Zhengyou Zhang, August 2002.
- [36] Camera pose revisited - New linear algorithms, M.Ameller, B.Triggs, L.Quan, Gravis-Inria-CNRS, Montbonnot, France.
- [37] State-of-the-art trends in mapping-past, present and future, K.Jacobsen, Institute of Photogrammetry and Geoinformation, University of Hannover, Germany, 2002.

-
- [38] Comparison of Newton-Gauss with Levenberg-Marquardt algorithm for space resection, Yao Jianchao, Chia Tien Chern, Singapore, 2001.
- [39] Trasformazioni, lucidi tratti dalla pagina Web del prof. Marini.
- [40] Analytical photogrammetry lecture notes, Robert Burtch, Centre for Photogrammetric Training, August 2000.
- [41] Laboratory self-calibration of a multi-band sensor, A. Alharthy, J. Bethel, School of Civil Engineering, Purdue University, West Lafayette.
- [42] Uncertainty of videogrammetric techniques used for aerodynamic testing, AIAA 2002-2794, A. W. Burner, T. Liu, R. DeLoach, NASA Langley Research Center, Hampton, 2002.
- [43] Straight lines have to be straight, F. Devernay, O. Faugeras, INRIA, Sophia Antipolis Cedex, France. 2000.
- [44] An alternative model of radial distortion in wide-angle lenses, J. Pers, S. Kovacic, Faculty of Electrical Engineering, University of Ljubljana, Slovenia.
- [45] Bundle adjustment with self-calibration using straight lines, A. Habib, M. Morgan, Y. Lee, Department of Civil and Environmental Engineering and Geodetic Science, The Ohio State University, 2000.
- [46] Mars rover localization using descent and fido rover imagery, R. Li, F. Ma, F. Xu, Department of Civil and Environmental Engineering and Geodetic Science, The Ohio State University, Columbus, OH.
- [47] Some aspects of zoom lens camera calibration, M. Li, J.M. Lavest, Royal Institute of Technology (KTH), Stockholm, Sweden.

-
- [48] Assessment of two cheap close-range feature extraction systems, A. Elaksher, M. Elghazali, A. Sayed, Y. Elmanadilli, Faculty of Engineering, Cairo University, Giza, Egypt.
- [49] Calibration procedure for short focal length off-the-shelf CCD cameras, J. Heikkila, O. Silven, Machine Vision Group, Department of Electrical Engineering, University of Oulu, Finland.
- [50] Geometric calibration and orientation of digital imaging systems, R. Godding, AICON 3D Systems GmbH, Braunschweig, Germany.
- [51] 3D reconstruction of articulated objects from uncalibrated images, F. Remondino, Institute of Geodesy and Photogrammetry, ETH Zurich, Switzerland.
- [52] On the performance of state estimation for visual servo systems, B.Bishop, S.Hutchinson, M.Spong, Coordinated science lab, University of Illinois, Urbana.
- [53] Nonparametric, model-based radial lens distortion correction using tilted camera assumption, J.Pers, S.Kovacic, Faculty of Electrical Engineering, University of Ljubljana, Slovenia.
- [54] Straight lines have to be straight, F.Devernay, O.Faugeras, Machine vision and applications, 2001.
- [55] Nonparametric correction of distortion, D.Stevenson, M.M.Fleck, Department of computer science, University of Iowa, Iowa City, october 1995.
- [56] A computer program for Direct Linear Transformation solution of the collinearity condition, and some application of it, G. T. Marzan, Dr. H. M. Kamara, Department of Civil Engineering, University of Illinois, Urbana, Illinois.

-
- [57] Direct Linear Transformation from comparator coordinates into object space coordinates in close-range photogrammetry, Y. I. Abdel-Aziz, Dr. H. M. Kamara, Department of Civil Engineering, University of Illinois, Urbana, Illinois.
- [58] Orientation angles, Y. H. Kwon, 1998.
- [59] Last square method, Y. H. Kwon, 1998.
- [60] DLT Method, Y. H. Kwon, 1998.
- [61] Localized DLT, Y. H. Kwon, 1998.
- [62] Double-plane method, Y. H. Kwon, 1998.
- [63] Accuracy, reliability and statistics in close-range photogrammetry, A.Gruen, Chair of Photogrammetry, Technical University of Munich, Muchen.
- [64] Machine vision, A. Gruen, Insistut of Geodasie und Photogrammetrie, ETH Zurich
- [65] Experiences with self-calibration bundle adjustment, A.Gruen, Insitute of Photogrammetry, Technical Univeristy of Munich, Munchen, 1978.
- [66] Photogrammetric networks, A.Gruen, Insitute of Photogrammetry, Technical Univeristy of Munich, Munchen, 1978.
- [67] Bundle solution, A.Gruen, Insitute of Photogrammetry, Technical Univeristy of Munich, Munchen, 1990.
- [68] Indirect georeferencing of airborne multi-line array sensors: a simulated case study, Daniela Poli, Institute of Geodesy and Photogrammetry, Swiss Federal Institute of Technology, Zurich.

-
- [69] La mia nuova macchina fotografica digitale, Cynthia L. Baron, Daniel Peck, Mondadori Informatica, 2002.
- [70] Mathematica - Principi e applicazioni, Giulio Falco, Addison-Wesley, 1993.
- [71] Usare Mathematica, Massimo Banzi, Jackson Libri, 1995.
- [72] Human body modelling from video sequences, Nicola d'Apruzzo, Ralf Planckers, Institute of Geodesy and Photogrammetry and Computer Graphics Lab, Switzerland, 1999.
- [73] Ray tracing, Bruce A. Draper, slide, January 2002.
- [74] Optimal ray intersection for computing 3D points from N-View correspondences, G. Slabaugh, R. Schafer, M. Livingston, october 2001.
- [75] Sistemi di riferimento e coordinate omogenee, Michele Folgheraiter, corso di Robotica, 2004.
- [76] An introduction to projective geometry (for computer vision), Stan Birchfield, march 1998.
- [77] An efficient and robust ray-box intersection algorithm, A.L. Williams, S. Barrus, R.K. Morley, P. Shirley, 2002.
- [78] A Lipschitz method for accelerated volume rendering, B.T. Stander, J.C. Hart, School of EECS, Washington State University.
- [79] Evolving visual routines, M.P. Johnson, P. Maes, T. Darrell, MIT media Laboratory, Cambridge.
- [80] Predicting RF coverage in large environments using ray-beam tracing and partitioning tree represented geometry, A. Rajkumar, B.F. Naylor, F. Feisullin, L. Rogers, AT&T Bell Laboratories, Murray Hill, 1995.