



UNIVERSITÀ DEGLI STUDI DI CAGLIARI

FACOLTÀ DI INGEGNERIA
Corso di Laurea in Ingegneria Meccanica

**Algoritmi per la calibrazione
della macchina fotografica**

Tesi di Laurea Triennale

Relatore:
Prof. Giuseppe Rodriguez

Candidato:
Riccardo Marcia

Anno Accademico 2014/2015

Ringrazio il professor G. Rodriguez per aver messo a mia disposizione le sue conoscenze, per aver dedicato del tempo alla correzione del mio elaborato, per la disponibilità e la pazienza nel rispondere alle mie domande sui nuovi argomenti trattati nella tesi.

*Dedico questo lavoro
a tutti quelli che mi
sono stati vicini
in questo percorso,
in particolare alla mia famiglia
che mi ha sopportato*

Indice

| | | |
|----------|---|-----------|
| 1 | Concetti di Matematica applicata | 6 |
| 1.1 | Richiami | 6 |
| 1.1.1 | Spazi | 6 |
| 1.1.2 | Matrici | 8 |
| 1.1.3 | Autovalori e autovettori | 12 |
| 1.2 | Ortogonalizzazione di Gram-Schmidt | 13 |
| 1.3 | Risoluzione di sistemi lineari "facili" | 14 |
| 1.4 | Fattorizzazione LDR (o LDU) | 15 |
| 1.5 | Fattorizzazione QR | 16 |
| 1.6 | Problema ai minimi quadrati | 17 |
| 1.7 | Decomposizione dei valori singolari (SVD) | 18 |
| 1.7.1 | La pseudoinversa | 20 |
| 1.8 | La SVD troncata (TSVD) | 21 |
| 1.8.1 | Principal Component Analysis (PCA) | 21 |
| 1.9 | Condizionamento di un sistema lineare | 22 |
| 2 | Camera Calibration | 24 |
| 2.1 | Parametri della camera | 25 |
| 2.1.1 | Parametri estrinseci | 26 |
| 2.1.2 | Parametri intrinseci | 26 |
| 2.2 | Introduzione alla calibrazione della macchina | 28 |
| 2.2.1 | Calibrazione diretta dei parametri | 29 |
| 2.3 | Computer Vision | 35 |
| 2.3.1 | Ricostruzione della forma | 35 |
| 2.3.2 | Metodo di calibrazione di Tsai | 36 |
| 3 | Applicazioni sperimentali | 37 |
| 3.1 | Calcolo coordinate punti nello spazio | 38 |
| 3.2 | Passaggio dallo spazio alla camera | 39 |
| 3.3 | Implementazione su Matlab | 41 |
| 3.3.1 | Dimostrazione numerica | 48 |

Introduzione

Le tematiche affrontate in questa tesi sono, le proprietà di calcolo dello spazio tre-dimensionale da una o più immagini digitali. Le proprietà che ci interessano sono principalmente quelle geometriche (come la forma degli oggetti solidi) e dinamiche (come la velocità degli oggetti). La maggior parte delle considerazioni presentate assumono che abbiano già avuto luogo un considerevole numero di processi di immagine, cioè da una immagine di partenza ne siano state calcolate delle nuove, oppure da una parte di esse si sia resa esplicita l'informazione necessaria per l'obiettivo della calibrazione. La computer vision¹ comprende l'interpretazione computerizzata delle immagini. Gli strumenti dei quali ha bisogno il sistema del computer vision includono l'hardware per l'acquisizione e memorizzazione delle immagini digitali in un pc, elaboratore di immagini e comunicatore di risultati. Ci sono dei sistemi per facilitare il compito del sistema di visione, per esempio controllando l'illuminazione, usando una macchina fotografica ad alta risoluzione oppure vincolando la posizione e la localizzazione degli oggetti. Non sempre però si può avere il pieno controllo della scena, come per esempio succede nel caso in cui si parli di navigazione in luoghi sconosciuti, in questi casi sono necessari degli algoritmi appropriati. Il dominio della computer vision è vasto e interdisciplinare e, in quanto tale, difficile da racchiudere in un'unica trattazione. Vista la complessità dell'argomento è necessario avere chiare alcune nozioni e correlate ad esso. La computer vision è una materia multidisciplinare che si sta evolvendo negli ultimi 30 anni. Grazie alla rapida crescita di questa materia abbiamo delle conseguenze positive, come il fatto che gli obiettivi e alcuni strumenti della computer vision sono accomunati con quelli di parecchie altre discipline. Le principali materie che si avvicinano alla computer vision sono i "processi di immagine", "modelli di riconoscimento" e la "fotogrammetria". Per quanto riguarda il primo, si può dire che alcuni algoritmi per la computer vision richiedono qualche processo d'immagine preliminare. Esempi di processi di immagine sono il miglioramento della qualità dell'immagine rispetto a quella di partenza; lo sviluppo di rappresentazioni compatte per immagini digitali;

¹In italiano si traduce come *visione artificiale*

l'eliminazione di noti effetti di degrado dell'immagine. Molti metodi sviluppati per il riconoscimento del modello lavoravano in modo ottimale con gli oggetti 2D oppure quelli 3D presentati in posizioni vincolate, ma non erano adatte nell'utilizzo all'interno del mondo 3D in generale. La fotogrammetria si occupa dell'ottenimento di misurazioni affidabili e accurate senza il contatto diretto dell'immagine. Questa disciplina è meno attinente alla computer vision rispetto alle altre due. La maggiore differenza sta nel fatto che la fotogrammetria ha un più alto livello di accuratezza rispetto a quello richiesto dalla computer vision. La " *International society of photogrammetry and remote sensing* " è l'organizzazione internazionale per la promozione e l'avanzamento della fotogrammetria.

Capitolo 1

Concetti di Matematica applicata

1.1 Richiami

In questa parte verranno ricordati alcuni concetti fondamentali, definizioni e proprietà base dell'algebra lineare e verrà illustrato il loro uso nell'ambiente di programmazione Matlab.

1.1.1 Spazi

Uno spazio lineare (o vettoriale) reale è un insieme V su cui sono definite due operazioni di somma e prodotto per uno scalare. Invece in uno spazio lineare complesso il campo degli scalari è costituito dall'insieme dei numeri complessi \mathbb{C} . Un sottospazio è un sottoinsieme che sia esso stesso un sistema lineare, $W \subset V$. Per ogni $\alpha, \beta \in \mathbb{R}$ e $x, y, z \in V$ gli spazi lineari godono delle seguenti proprietà:

- $\mathbf{x} + \mathbf{y} \in \mathbf{V}$
- $\alpha \mathbf{x} \in \mathbf{V}$
- $\mathbf{x} + \mathbf{y} = \mathbf{y} + \mathbf{x}$
- $(\mathbf{x} + \mathbf{y}) + \mathbf{z} = \mathbf{x} + (\mathbf{y} + \mathbf{z})$
- esiste $\mathbf{0} \in \mathbf{V}$ tale che $\mathbf{x} + \mathbf{0} = \mathbf{x}$
- esiste $-\mathbf{x} \in \mathbf{V}$ tale che $\mathbf{x} + (-\mathbf{x}) = \mathbf{0}$
- $\alpha(\beta \mathbf{x}) = (\alpha\beta)\mathbf{x}$

- $\alpha(\mathbf{x} + \mathbf{y}) = \alpha\mathbf{x} + \alpha\mathbf{y}$
- $(\alpha + \beta)\mathbf{x} = \alpha\mathbf{x} + \beta\mathbf{x}$
- $\mathbf{1}\mathbf{x} = \mathbf{x}$

I **vettori** sono gli elementi di uno spazio lineare.

Su Matlab

Per introdurre un vettore riga è sufficiente riportare i valori separati da spazi bianchi o virgole.

Ad esempio per introdurre il vettore w di componenti $[1, 2, 3]$

```
w = [1 2 3]
w =
```

```
1    2    3
```

Per introdurre un vettore colonna basta riportare fra parentesi quadre i valori della componenti del vettore stesso separati da un punto e virgola, oppure usare la denotazione precedente con il simbolo " ' " che in linguaggio Matlab vuol dire trasposto. Ad esempio per introdurre $v \in \mathbb{R}$

```
v=[1 2 3]'
```

```
v =
```

```
1
2
3
```

Il comando $V=[1:10]$ genera un vettore riga di dieci componenti dato dai valori $1,2,\dots,10$.

```
V = [1:10]
```

```
V =
```

```
1    2    3    4    5    6    7    8    9   10
```

La **combinazione lineare** dei vettori $\{x_1, \dots, x_k\}$, con $x \in V$ è:

$$x = \sum_{i=1}^k \alpha_i x_i$$

I vettori (x_1, \dots, x_k) sono linearmente indipendenti se

$$\sum_{i=1}^k \alpha_i x_i = 0 \longrightarrow \alpha_i = 0$$

Una **norma** è una funzione che assegna ad ogni vettore di uno spazio vettoriale, tranne lo zero, una lunghezza positiva.

Una **base** è un insieme di vettori linearmente indipendenti tali che ogni vettore dello spazio possa essere espresso come loro combinazione lineare.

La **dimensione** di uno spazio lineare è la cardinalità di una base. Un vettore si dice **normalizzato** rispetto alla sua norma quando questa vale 1.

Una successione si dice **di Cauchy** se

$$\lim_{m, n \rightarrow \infty} \|x_m - x_n\|$$

Questa definizione implica che, per m ed n abbastanza grandi la distanza tra x_m ed x_n tende a zero, indipendentemente dal valore di $|m - n|$. Uno spazio si dice **completo** quando tutte le successioni di Cauchy hanno un limite.

Uno spazio normato completo viene detto **spazio di Banach**

1.1.2 Matrici

Una matrice $m \times n$ è un quadro di numeri reali e complessi di m righe ed n colonne.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Se $m=n$ la matrice è quadrata, altrimenti è rettangolare. Scriveremo con $\mathbb{R}^{m \times n}$ o $\mathbb{C}^{m \times n}$ per riferirci a matrici reali o complesse.

Se $A \in \mathbb{R}$ allora la matrice trasposta sarà $(A^T)_{ij} = a_{ji}$ ottenuta scambiando le righe con le colonne.

Se $A \in \mathbb{R}$ e $B \in \mathbb{R}$ allora la somma e prodotto per uno scalare sono definiti da

$$(A + B)_{ij} = a_{ij} + b_{ij}$$

$$(\alpha A)_{ij} = \alpha a_{ij}$$

Il prodotto fra due matrici si esegue facendo riga per colonna. Per cui una matrice $A \in \mathbb{R}^{m \times n}$ x $B \in \mathbb{R}^{n \times p}$ darà come risultato una matrice $C \in \mathbb{R}^{m \times p}$. Il prodotto matriciale non gode della proprietà commutativa.

L'elemento neutro è la *matrice identità*

$$I = \text{diag}(1, \dots, 1) = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$$

L'elevamento a potenza di una matrice invece è definita come

$$A^p = AA \dots A$$

Su Matlab

Per assegnare la matrice $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

```
A=[1 2 3;4 5 6; 7 8 9]
```

```
A =
```

```

1     2     3
4     5     6
7     8     9
```

Per creare invece rispettivamente una matrice di soli zero o soli uno si usano i comandi "zeros" e "ones" susseguito da un numero tra parentesi, che indica il numero di righe e colonne della matrice.

```
A=zeros(3)
```

```
A =
```

```

0     0     0
0     0     0
0     0     0
```

```
B=ones(3)
```

```
B =
```

```

1   1   1
1   1   1
1   1   1

```

Per creare invece una matrice identità si usa il comando "eye". Oppure per una matrice con coefficienti casuali si usa il comando "rand".

```
C=eye(2)
```

```
C =
```

```

1   0
0   1

```

```
D=rand(3)
```

```
D =
```

```

0.8147    0.9134    0.2785
0.9058    0.6324    0.5469
0.1270    0.0975    0.9575

```

Per tutti questi comandi, se non si desidera una matrice quadrata, successivamente al comando, tra parentesi, si inserisce rispettivamente il numero di righe e di colonne, divise da una virgola.

Una matrice si dice **invertibile** o **non singolare** se esiste la matrice inversa A^{-1} tale che $AA^{-1}=A^{-1}A=I$. A è invertibile se e solo se tutte le righe e colonne sono linearmente indipendenti.

Il **determinante** di una matrice, fissata una riga i , può essere calcolato mediante la formula di Laplace.

$$\det(A) = \sum_{j=1}^n (-1)^{i+j} a_{ij} \det(A_{ij})$$

essendo A_{ij} la sottomatrice di A eliminando la i -esima riga e la j -esima colonna. Da questo è intuibile che A è invertibile se e solo se $\det(A) \neq 0$.

Il rango di una matrice può essere definito il numero massimo di righe o colonne linearmente indipendenti o come l'ordine della più grande sottomatrice con determinante nullo.

Su Matlab

```

A=[1 2 3; 4 5 6; 7 8 9];
rank(A)

ans =

     2
det(A)

ans =

 6.6613e-16

```

Matrici di forma particolare

Matrici Hermitiane Una matrice complessa si dice Hermitiana se coincide con la sua aggiunta ($A = A^*$).

Matrici unitarie Una matrice complessa è **unitaria** se $Q^* Q = Q Q^* = I$
 Una matrice reale è **ortogonale** se $Q Q^T = Q^T Q = I$.

Se Q è unitaria allora $|\det(Q)| = 1$

Matrici triangolari Una matrice A è triangolare **superiore** se i suoi elementi verificano la condizione $a_{ij} = 0$ per $i > j$

$$\begin{bmatrix} * & * & \dots & * \\ & * & \dots & * \\ & & \ddots & \vdots \\ & & & * \end{bmatrix}$$

Una matrice L è triangolare **inferiore** se $l_{ij} = 0$ per $i < j$

$$\begin{bmatrix} * \\ * & * \\ \vdots & \vdots & \ddots \\ * & * & \dots & * \end{bmatrix}$$

Una matrice **diagonale** è contemporaneamente triangolare superiore e inferiore.

Teorema 1.1 *Se T è triangolare il suo determinante è dato dal prodotto degli elementi diagonali.*

Su Matlab

Riportiamo rispettivamente, partendo da una matrice generica A , la corrispondente matrice triangolare superiore, inferiore e diagonale.

```
A=[1 2 3;4 5 6;7 8 9]
```

```
A =
```

```
    1    2    3
    4    5    6
    7    8    9
```

```
>> triu(A)
```

```
ans =
```

```
    1    2    3
    0    5    6
    0    0    9
```

```
>> tril(A)
```

```
ans =
```

```
    1    0    0
    4    5    0
    7    8    9
```

```
>> diag(A)
```

```
ans =
```

```
    1
    5
    9
```

1.1.3 Autovalori e autovettori

Consideriamo solo matrici quadrate.

Definizione: Si definiscono rispettivamente autovalore ed autovettore di una

matrice A , uno scalare λ ed un vettore $x \neq 0$ che verificano la relazione

$$Ax = \lambda x \quad (1.1)$$

La (1.1) possiamo scriverla anche in un'altra forma

$$(A - \lambda I)x = 0 \quad (1.2)$$

Per far sì che questo sistema lineare omogeneo ammetta un λ al quale corrisponde una soluzione non nulla, è necessario che il suo determinante

$$p_A(\lambda) = \det(A - \lambda I)$$

si annulli. Dato che $p_A(\lambda)$ è il **polinomio caratteristico** di A , ovvero un polinomio di grado n in λ , il *Teorema fondamentale dell'algebra* ci dice che esistono n zeri di $p_A(\lambda)$ che, per definizione, sono gli **autovalori** di A .

Usiamo la convenzione di ordinare gli autovalori in ordine crescente in base al loro modulo

$$|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$$

Per ogni autovalore λ_k , con $k = 1, \dots, n$, per ogni soluzione non nulla di (1.2) otteniamo un autovettore corrispondente. Lo **spettro** di una matrice A è l'insieme dei suoi autovalori

$$\sigma(A) = \{\lambda_1, \dots, \lambda_n\}$$

Il **raggio spettrale** è il massimo dei moduli degli autovettori.

$$\rho(A) = \max |\lambda_k|$$

1.2 Ortogonalizzazione di Gram-Schmidt

Il processo di Gram-Schmidt serve per ortonormalizzare n vettori indipendenti di \mathbb{R}^m con $n \leq m$.

I vettori vengono raggruppati in una matrice $A = [a_1, a_2, \dots, a_n]$ e tramite questo procedimento otterremo una matrice $Q = [q_1, q_2, \dots, q_n]$, tale che

$$\text{span}\{q_1, \dots, q_n\} = \text{span}\{a_1, \dots, a_n\} \quad e \quad \langle q_i, q_j \rangle = q_i^T q_j = \delta_{ij} \quad (1.3)$$

Nella 1.3 δ_{ij} è il **delta di Kronecker**, definito come

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Il primo passo dell'algoritmo sarà

$$\tilde{q}_1 = v_1, \quad r_{11} = \|\tilde{q}_1\| \rightarrow q_1 = \frac{\tilde{q}_1}{r_{11}} \quad (1.4)$$

dove la norma usata è quella euclidea. Generalizzando la 1.4

$$\tilde{q}_k = a_k - \sum_{j=1}^{k-1} r_{jk} q_j$$

sapendo che

$$r_{jk} = q_j^T a_k, \quad j = 1, \dots, k-1$$

infine avremo

$$q_k = \frac{\tilde{q}_k}{r_{kk}}, \quad \text{con} \quad r_{kk} = \|\tilde{q}_k\|$$

Il metodo Gram-Schmidt può essere applicato a qualsiasi spazio di Hilbert. Applicare il metodo ad una matrice A, consente di costruire la **fattorizzazione QR** (che vedremo in seguito), della matrice.

Purtroppo questo algoritmo non è perfetto, poichè le colonne di Q tendono a perdere la reciproca ortogonalità al procedere delle iterazioni. Per questo è stato ideato una differente versione dell'algoritmo, detto **MGS** (*modified Gram-Schmidt*).

1.3 Risoluzione di sistemi lineari "facili"

Prima di introdurre algoritmi un po' più complessi, vediamo quelli per determinare classi di sistemi strutturati.

Sistemi diagonali

Un sistema diagonale ha la forma

$$Dx = b$$

dove $D = \text{diag}(d_1, \dots, d_n)$ con qualunque $d_i \neq 0$. La soluzione sarà

$$x_i = \frac{b_i}{d_i}, \quad i = 1, \dots, n$$

Sistemi ortogonali

Hanno la forma

$$Qx = b$$

La peculiarità di questo sistema è che $Q^T Q = Q Q^T = I$. La soluzione di questo è data da

$$x = Q^T b, \quad \text{ovvero} \quad x_i = \sum_{j=1}^n q_{ij} b_j, \quad i = 1, \dots, n$$

Sistemi triangolari

- **Sistema triangolare inferiore** Questo sistema assume la forma

$$Lx = b, \quad \ell_{ij} = 0 \quad i < j$$

e si risolve innanzitutto la prima equazione, essendo in una sola variabile. Il valore calcolato va quindi sostituito nella seconda equazione, e così via. Questo algoritmo viene denominato **forward substitution**.

- **Sistema triangolare superiore** Questo sistema assume la forma

$$Ux = b, \quad u_{ij} = 0 \quad i > j$$

e si procede in modo opposto al precedente. Si svolge per prima l'ultima equazione e si procede a ritroso verso la prima equazione. L'algoritmo si chiama **backward substitution**.

1.4 Fattorizzazione LDR (o LDU)

Sia $A=LU$. Ponendo $D = \text{diag}(u_{11}, \dots, u_{nn})$ e $R = D^{-1}U$, per cui

$$A = LDR$$

L è una matrice triangolare inferiore, ed R triangolare superiore, $\ell_{ii} = r_{ii} = 1$. Se A è simmetrica allora risulta che $R = L^T$ per cui

$$A = LDL^T \tag{1.5}$$

Definizione: Si definisce inerzia di una matrice Hermitiana A la terna dei numeri naturali

$$\text{Inerzia}(A) := \{\pi, \nu, \delta\}$$

nella quale π indica il numero di autovalori positivi, ν quelli negativi, δ il numero degli autovalori nulli di A . La **legge di inerzia di Sylvester** ci dice che due matrici Hermitiane A e B esiste una matrice non singolare C tale che $B=C^*AC$ se e solo se $\text{Inerzia}(A)=\text{Inerzia}(B)$. Se la matrice A è definita positiva, possiamo applicare la legge di Sylvester alla fattorizzazione 1.5 e possiamo dire che tutti gli elementi della matrice D sono positivi per cui

$$D^{\frac{1}{2}} = \text{diag}(\sqrt{d_1}, \dots, \sqrt{d_n})$$

e la matrice $R = D^{1/2}L^T$ con $r_{ii} > 0$ in modo tale da avere

$$A = L \underbrace{D^{1/2}D^{1/2}}_{D^{\frac{1}{2}+\frac{1}{2}=1}} L^T = R^T R$$

La fattorizzazione $A = R^T R$ viene detta anche **fattorizzazione di Cholesky**

1.5 Fattorizzazione QR

Ogni matrice A singolare o non di struttura $m \times n$ si può fattorizzare nella dimensione

$$A = QR$$

essendo Q una matrice ortogonale $m \times m$ e R una matrice triangolare superiore con le stesse dimensioni di A .

A questo punto ci serve un algoritmo per poterla calcolare.

Se A è non singolare quadrata, la fattorizzazione QR può essere usata per risolvere sistemi lineari del tipo $Ax=b$. Così facendo

$$Ax = b \rightarrow Q \underbrace{Rx}_c = b \quad \begin{cases} Qc = b \\ Rx = c \end{cases}$$

La soluzione della prima equazione è data da $c = Q^T b$, mentre la seconda equazione può essere risolta con backward substitution. Un importante vantaggio della fattorizzazione QR riguarda la crescita del condizionamento.

Teorema 1.2 *Se $A=QR$ allora*

$$\|A\|_2 = \|R\|_2$$

Dimostrazione. Sapendo che $A=QR$ ed essendo $\|Q\|_2 = 1$ allora otteniamo che

$$\|A\|_2 \leq \|Q\|_2 \|R\|_2 = \|R\|_2$$

Allo stesso modo è dimostrabile che partendo da una formula inversa come $R = Q^T A$ si può ottenere

$$\|R\|_2 \leq \|A\|_2$$

Entrambi i procedimenti ci portano alla tesi secondo la quale $\|A\|_2 = \|R\|_2$

Inoltre possiamo affermare che la matrice R ha lo stesso numero di condizionamento rispetto alla norma-2 della matrice A .

La fattorizzazione QR serve per trasformare un sistema lineare in un sistema triangolare superiore ad esso equivalente e dotato dello stesso numero di condizionamento. Per questo la fattorizzazione QR si usa per risoluzione di sistemi lineari malcondizionati. I due algoritmi più usati per risolvere questa fattorizzazione sono: il metodo **Householder** e il metodo **Givens**.

1.6 Problema ai minimi quadrati

Il problema dei minimi quadrati è un quesito di primaria importanza, che nasce dal bisogno di "approssimare" un modello matematico lineare a delle osservazioni, in connessione con il problema della *regressione lineare*. In alcune applicazioni pratiche possiamo incontrare dei sistemi lineari con un numero di equazioni differenti dal numero delle incognite. I nostri problemi non hanno soluzione, per cui

$$Ax = b$$

essendo A una matrice a rango pieno $m \times n$, $b \in \mathbb{R}^m$ e $x \in \mathbb{R}^n$. Abbiamo due casi distinti.

- $m > n$ In questo caso abbiamo più equazioni che incognite, per cui il problema potrebbe non avere soluzioni: il sistema è *sovradeterminato*
- $m < n$ Abbiamo a disposizione più incognite che equazioni, il sistema può avere infinite soluzioni: il sistema è *sottodeterminato*

Anche nel caso in cui A è a rango non pieno, oppure quadrata e singolare, ci si può sempre ricondurre a uno dei due casi appena esposti. In queste situazioni i sistemi non hanno una sola soluzione, per questo si dicono *mal posti*. La situazione in cui $m > n$ è molto frequente, dove, come già detto, il numero di equazioni lineari a disposizione per descrivere il fenomeno è superiore a numero delle incognite; questo significa che $m - n$ equazioni sono linearmente dipendenti dalle altre, ma potrebbe non essere noto quali. Se nelle equazioni sono presenti errori, è un fatto positivo avere un grande numero di equazioni, per cercare di ripulire la soluzione dall'influsso degli errori sperimentali.

Per poter risolvere il problema, dobbiamo ricondurci ad un caso di problema *ben posto*.

Se tutte le equazioni non possono essere verificate contemporaneamente, è ragionevole richiedere che lo scarto quadratico medio (varianza) tra il primo e secondo membro del sistema sia minimo. Così abbiamo un **problema ai minimi quadrati**. La condizione di varianza minima si può esprimere come

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2 \quad \min_a$$

Se il minimo va a zero, allora il sistema originale ammette una soluzione in senso classico. Se così non è, allora otteniamo la soluzione nel senso dei minimi quadrati del sistema $Ax=b$.

1.7 Decomposizione dei valori singolari (SVD)

La decomposizione a valori singolari (SVD) consente di fattorizzare una matrice complessa rettangolare. Il teorema spettrale afferma che le matrici normali possono essere unitariamente diagonalizzate usando gli autovettori. La SVD può essere vista come una generalizzazione del teorema spettrale per matrici arbitrarie, non necessariamente quadrate. Detto in altro modo, qualsiasi matrice può essere ridotta in forma diagonale tramite una pre- e post- moltiplicazione per una matrice unitaria. La SVD è una decomposizione matriciale di grande importanza teorica e pratica per la trattazione dei problemi dei minimi quadrati. Per tornare alla nascita di questa scomposizione bisogna andare indietro di oltre un secolo, tuttavia solo recentemente la SVD è stata usata in tutto il suo potenziale. Oggigiorno è un importante strumento per numerose aree di applicazione come il processo di immagine e di segnale, teoria dei controlli ecc.

Teorema 1.3 *Sia $A \in \mathbb{C}^{m \times n}$ una matrice di rango r . Allora esistono le matrici unitarie $U \in \mathbb{C}^{m \times m}$ e $V \in \mathbb{C}^{n \times n}$ tali che*

$$A = U \Sigma V^H, \quad \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \quad (1.6)$$

dove la matrice V^H indicherà la matrice determinata coniugando ogni elemento e prendendo la trasposta, $\Sigma \in \mathbb{C}^{m \times n}$, $\Sigma_1 = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$, e

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0. \quad (1.7)$$

I σ_i vengono chiamati valori singolari di A , e si può scrivere

$$U = [u_1, \dots, u_m], \quad V = [v_1, \dots, v_n] \quad (1.8)$$

dove gli u_i e v_i sono, rispettivamente, i vettori singolari sinistri e destri corrispondenti a σ_i , con $i = 1, \dots, r$.

Il rango di A si può determinare come

$$\text{rank}(A) = r \quad \text{se e solo se } \sigma_r > 0 \quad \sigma_{r+1} = \dots = \sigma_n = 0$$

Il condizionamento invece

$$k_2(A) = \frac{\sigma_1}{\sigma_r}$$

La SVD di A può essere scritta, nella forma compatta

$$A = U_1 \Sigma_1 V_1^H = \sum_{i=1}^r \sigma_i u_i v_i^H \quad (1.9)$$

dove

$$U_1 = [u_1, \dots, u_r], \quad V_1 = [v_1, \dots, v_r]. \quad (1.10)$$

Con questo, una matrice A di rango r viene decomposta nella somma di r matrici di rango 1, con $r = \text{rank}(A)$.

I valori singolari di A sono unici. Il vettore singolare v_j , con $j \leq r$, sarà unico solo quando σ_j^2 è un autovalore semplice di $A^H A$.

Su Matlab

Riportiamo rispettivamente, partendo da una matrice generica A , la corrispondente matrice U, Σ, V

```
A=[1 2 3; 4 5 6; 7 8 9]
```

```
A =
```

```
    1    2    3
    4    5    6
    7    8    9
```

```
>> [U,S,V]=svd(A)
```

```
U =
```

```
-0.2148    0.8872    0.4082
-0.5206    0.2496   -0.8165
-0.8263   -0.3879    0.4082
```

```
S =
```

$$\begin{array}{ccc}
 16.8481 & 0 & 0 \\
 0 & 1.0684 & 0 \\
 0 & 0 & 0.0000 \\
 \\
 V = & & \\
 -0.4797 & -0.7767 & -0.4082 \\
 -0.5724 & -0.0757 & 0.8165 \\
 -0.6651 & 0.6253 & -0.4082
 \end{array}$$

1.7.1 La pseudoinversa

La decomposizione a valori singolari è un potente strumento per risolvere problemi lineari ai minimi quadrati. Questo perché le matrici unitarie che trasformano A nella forma diagonale (1.6) non cambiano la norma euclidea dei vettori. Il seguente risultato può essere applicato sia ai sistemi lineari sovradeterminati sia a quelli sottodeterminati.

Teorema 1.4 *Si consideri il problema generale ai minimi quadrati*

$$\min_{x \in S} \|x\|_2, \quad S = \{x \in \mathbb{R}^n \mid \|b - Ax\|_2 = \min\}, \quad (1.11)$$

dove $A \in \mathbb{C}^{m \times n}$ e il $\text{rank}(A) = r \leq \min(m, n)$. Questo problema ha sempre un'unica soluzione, che può essere scritta in termini di SVD di A come

$$x = V \begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^H b. \quad (1.12)$$

Definizione 1.1 *Scriviamo la (1.12) come $x = A^\dagger b$, dove*

$$A^\dagger = V \begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^H \in \mathbb{C}^{n \times m} \quad (1.13)$$

è chiamata la pseudoinversa di A , e la soluzione (1.12) è chiamata soluzione della pseudoinversa.

1.8 La SVD troncata (TSVD)

Un'importante applicazione della SVD riguarda l'approssimazione della matrice A con una a rango non pieno, ottenuta ponendo a zero alcuni dei valori singolari, a partire dai più piccoli: in tal caso si parla di SVD troncata (TSVD). La TSVD permette un'approssimazione di una matrice tramite un'altra di rango inferiore. Questo può essere utile in diversi ambiti, dalla compressione dei dati alla regolarizzazione delle soluzioni. La TSVD si comporta come un filtro che taglia i vettori e i valori singolari dal $k+1$ -esimo compreso in poi eliminando qualunque rumore fosse in essi contenuto. Poiché i valori singolari decrescono molto rapidamente, saranno gli ultimi che risentiranno di più dell'errore poiché di ordini di grandezza confrontabili o ancora peggio l'errore è più grande del valore singolare stesso. Il problema sorge nella scelta del parametro k a cui tagliare, esistono vari metodi per la stima di questo parametro.

Nel caso di sistemi lineari mal condizionati del tipo $Ax = b$, è possibile ottenere una soluzione approssimata attraverso la TSVD. Si assume che la decomposizione a valori singolari di A sia:

$$A = U\Sigma V^T = \sum_{i=1}^n u_i \sigma_i v_i^T \quad (1.14)$$

e si assegna rango k ad A . Ponendo a zero tutti i valori singolari σ_i , con $i > k$, la soluzione corrispondente può essere scritta nella forma:

$$x = \sum_{i=1}^k \frac{c_i}{\sigma_i} v_i, \quad c = U^T b \quad (1.15)$$

che risulta essere la soluzione della TSVD.

1.8.1 Principal Component Analysis (PCA)

Questo metodo utilizza la SVD troncata. Viene utilizzata specialmente quando si hanno matrici molto grandi e si hanno a disposizione molto dati, come ad esempio per un social network, dove si devono segnare le persone che sono a contatto tra di loro e si hanno perciò matrici di dimensioni esagerate. Con la PCA si considerano solo gli elementi principali della matrice, in questo caso partendo da una matrice A otteniamo delle sottomatrici di U , Σ e V^T il cui prodotto ci dà una sottomatrice di A simile a quella che avremmo senza aver escluso gli elementi di minore importanza.

1.9 Condizionamento di un sistema lineare

Si considerino sistemi lineari con matrici e termini noti reali, anche se tutto ciò è analogo per matrici complesse.

Avendo un sistema lineare $Ax = b$, se perturbiamo i termini noti con un vettore δb anche la soluzione del sistema risulterà perturbata. Detta δx tale perturbazione, il nuovo sistema sarà

$$A(x + \delta x) = (b + \delta b)$$

dove $x + \delta x$ è la soluzione esatta del problema perturbato. Da questo, utilizzando la relazione $Ax = b$, segue il sistema

$$A\delta x = \delta b$$

la cui soluzione, introdotta una norma matriciale consistente, soddisfa la disuguaglianza

$$\|\delta x\| \leq \|A^{-1}\| * \|\delta b\| \quad (1.16)$$

questa relazione dà informazioni sul condizionamento assoluto del problema, ma è possibile esprimere anche il condizionamento relativo.

$$\frac{1}{\|x\|} \leq \|A\| * \frac{1}{\|b\|}$$

Moltiplicando membro a membro quest'ultima disequazione per la (1.16) si ottiene

$$\frac{\|\delta x\|}{\|x\|} \leq \|A\| \|A^{-1}\| * \frac{\|\delta b\|}{\|b\|}$$

Si definisce **numero di condizionamento** di una matrice A la quantità

$$k(A) = \|A\| \|A^{-1}\|$$

questo misura il massimo fattore di amplificazione dell'errore relativo sulla soluzione indotta dall'errore relativo sui dati. Il valore del numero di condizione è ovviamente influenzato dalla norma matriciale adottata. Useremo un pedice, come in $k_2(A)$ o in $k_\infty(A)$, quando vorremo porre in evidenza la norma utilizzata per calcolarlo. L'ipotesi che solo il termine noto sia perturbato è in realtà poco realistica, dato che la matrice A può essere affetta da errori. È possibile ottenere una relazione per il condizionamento relativo anche nel caso più generale, la quale mostra che l'amplificazione dell'errore continua ad essere influenzata in modo essenziale dal numero di condizione.

$$(A + \delta A)(x + \delta x) = (b + \delta b)$$

è il sistema perturbato. Se $\|\delta A\| \leq \|\frac{1}{A^{-1}}\|$ allora l'errore relativo sulla soluzione del sistema soddisfa la seguente disuguaglianza

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{k(A)}{1 - k(A)\frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right) \quad (1.17)$$

Capitolo 2

Camera Calibration

Formazione della geometria dell'immagine

Il nostro obiettivo ora è collegare la posizione dei punti nello spazio che corrispondono ai relativi punti nell'immagine.

Consideriamo un piano π , un piano d'immagine, e un punto nello spazio \mathbf{O} chiamato *centro o fuoco di proiezione*. La distanza tra π e \mathbf{O} è la *lunghezza focale*. La linea attraverso \mathbf{O} e perpendicolare a π è l'*asse ottico*, mentre \mathbf{o} , intersezione tra π e l'asse ottico è chiamato *punto principale o centro d'immagine*.

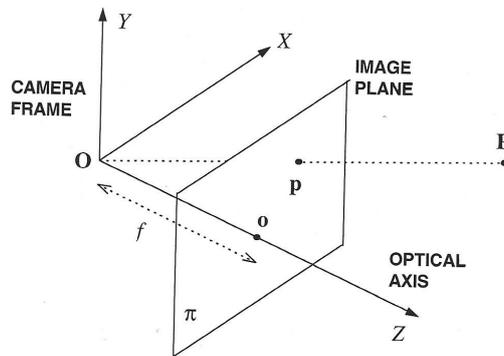


Figura 2.1: La prospettiva della macchina fotografica

Come si vede dalla figura 2.1 il punto \mathbf{p} è l'immagine di \mathbf{P} ed è il punto che sta nella linea tra \mathbf{P} e \mathbf{O} intersecando il piano di immagine π .

Considerando il sistema di riferimento lungo i 3 assi coordinati in cui \mathbf{O} è l'origine e il piano π è ortogonale all'asse Z , consideriamo i punti $\mathbf{P} = [X, Y, Z]^T$ e $\mathbf{p} = [x, y, z]^T$. Questo sistema di riferimento è il sistema di riferimento del-

la macchina fotografica, e ha un'importanza fondamentale nella computer vision. Le equazioni base della proiezione nel sistema di riferimento della macchina sono

$$x = f \frac{X}{Z} \quad y = f \frac{Y}{Z} \quad (2.1)$$

Nel sistema di riferimento della camera la terza componente dei punti dell'immagine è sempre uguale alla lunghezza focale, per questo si omette e si scrive che $\mathbf{p}=[x,y]^T$

2.1 Parametri della camera

In questa sezione analizziamo i parametri che stanno alla base della macchina fotografica.

Definizione

Gli algoritmi della computer vision che ricostruiscono la struttura 3D di una scena o che elaborano la posizione di oggetti nello spazio necessitano di equazioni che legano le coordinate dei punti nello spazio 3D con le coordinate dei corrispondenti punti immagini. Queste equazioni sono scritte nel sistema di riferimento della macchina, ma spesso viene assunto che

- Il sistema di riferimento della macchina possa essere localizzato rispetto a qualche altro sistema, conosciuto come sistema di riconoscimento globale (il *world reference frame*)
- Le coordinate dei punti dell'immagine nel sistema di riferimento della macchina possano essere ottenuti dalle coordinate dei pixels, le uniche ricavabili direttamente dall'immagine.

Questo è equivalente ad assumere la conoscenza di qualche caratteristica della macchina, noti come parametri intrinseci ed estrinseci della macchina. Il nostro successivo compito è quello di capire l'esatta natura dei parametri estrinseci ed intrinseci della macchina.

I parametri estrinseci sono quei parametri che definiscono la posizione e l'orientamento del sistema di riferimento della macchina rispetto al sistema di riferimento nello spazio noto. I parametri intrinseci sono i parametri necessari per collegare le coordinate dei pixel di un punto dell'immagine con le coordinate corrispondenti nel sistema di riferimento della macchina. Il problema di valutare il valore di questi parametri è chiamato calibrazione della macchina.

2.1.1 Parametri estrinseci

Il sistema di riferimento della macchina è stato introdotto con lo scopo di scrivere le equazioni fondamentali della proiezione prospettica (2.1) in una forma semplice. Tuttavia, il sistema di riferimento della macchina non è sempre conosciuto, ed un problema comune è determinare la posizione e l'orientamento della macchina rispetto a qualche riferimento noto, usando solamente l'informazione dell'immagine. I parametri estrinseci sono definiti come un insieme di parametri geometrici che identificano univocamente la trasformazione tra un sistema di riferimento di una macchina incognito e un sistema di riferimento noto, chiamato *world reference frame* (sistema di riferimento nello spazio).

Una scelta tipica per descrivere la trasformazione tra la macchina e il sistema di riferimento nello spazio è di usare:

- Un vettore di traslazione 3D, \mathbf{T} , che descrive le posizioni relative dell'origine dei due sistemi di riferimento
- Una matrice di rotazione 3x3, \mathbf{R} , una matrice ortogonale ($R^T R = R R^T = I$) che porta gli assi corrispondenti dei due sistemi l'uno sull'altro.

Le relazioni di ortogonalità riducono il numero di gradi di libertà di \mathbf{R} a 3. I parametri estrinseci della macchina fotografica sono il vettore di traslazione, \mathbf{T} , e la matrice rotazionale, \mathbf{R} (o, meglio, i suoi parametri liberi), che specificano il passaggio dal sistema di riferimento dello spazio esterno a quello della macchina.

2.1.2 Parametri intrinseci

I parametri intrinseci possono essere definiti come l'insieme dei parametri necessari a caratterizzare le caratteristiche ottiche, geometriche e digitali della "visione" della macchina. Per il foro stenopeico ¹ di una macchina, abbiamo bisogno di tre parametri, che specificano rispettivamente

- La proiezione prospettica, per cui il solo parametro è la lunghezza focale, f ;
- La trasformazione tra le coordinate del sistema di riferimento della macchina e le coordinate del pixel;

¹Il foro stenopeico è un foro, sufficientemente piccolo, che si pratica sulla parete di una camera oscura, per vedere proiettata, sulla parete opposta, l'immagine di ciò che esiste, esternamente, di fronte al foro. Questo nella fotocamera risulta essere un semplice foro posizionato al centro di un lato, come obiettivo.

- La distorsione geometrica introdotta dall'ottica.

Dalla Macchina alle coordinate del Pixel. Per trovare il secondo insieme dei parametri intrinseci, noi dobbiamo collegare le coordinate (x_{im}, y_{im}) di un punto dell'immagine in unità pixel con le coordinate (x, y) dello stesso punto nel sistema di riferimento della macchina. Le coordinate (x_{im}, y_{im}) possono essere pensate come coordinate di un nuovo sistema di riferimento, talvolta chiamato sistema di riferimento dell'immagine.

Dal sistema di coordinate dell'immagine a quello della camera.

Trascurando ogni distorsione geometrica che sia stata introdotta dall'ottica e assumendo che la matrice CCD è costituita da una griglia rettangolare di elementi fotosensibili, abbiamo

$$x = -(x_{im} - o_x)s_x$$

$$y = -(y_{im} - o_y)s_y$$

Il cambiamento del segno è dovuto al fatto che gli assi orizzontali e verticali dell'immagine e il riferimento della macchina hanno orientamento opposto. Indichiamo con (o_x, o_y) le coordinate in pixel del centro dell'immagine (il punto principale) e (s_x, s_y) la dimensione effettiva del pixel (in millimetri) rispettivamente nella direzione orizzontale e verticale.

I parametri intrinseci della macchina sono definiti come la distanza focale, f , la posizione del centro dell'immagine in coordinate pixel, o_x, o_y , la dimensione effettiva del pixel lungo la direzione verticale e orizzontale s_x, s_y , e, se richiesta, il coefficiente della distorsione radiale.

In parecchi casi, l'ottica introduce distorsioni di immagine che diventano evidenti nella periferia dell'immagine, o addirittura altrove quando si usi ottica con grandi campi visivi. Fortunatamente, queste distorsioni possono essere modellate piuttosto accuratamente come semplici distorsioni radiali, secondo le relazioni

$$x = x_d(1 + K_1r^2 + K_2r^4)$$

$$y = y_d(1 + K_1r^2 + K_2r^4)$$

con (x_d, y_d) le coordinate di punti distorti, ed $r^2 = x_d^2 + y_d^2$. Come mostrato dalle equazioni sopra, questa distorsione è una dislocazione radiale dei punti immagine. La dislocazione è nulla nel centro dell'immagine ed aumenta con l'aumentare della distanza del punto dal centro dell'immagine. k_1 e k_2 sono ulteriori parametri intrinseci. Dal momento che generalmente sono molto

piccoli, la distorsione radiale viene ignorata ogniqualvolta l'alta accuratezza non viene richiesta in tutte le regioni dell'immagine, o quando i pixels periferici possono essere scartati. Poichè $k_2 \ll k_1$, k_2 è spesso posto eguale a 0, e k_1 è il solo parametro intrinseco ad essere valutato nel modello di distorsione radiale.

2.2 Introduzione alla calibrazione della macchina

La *camera calibration* stabilisce i valori dei parametri intrinseci ed estrinseci del modello della macchina.

L'idea chiave dietro la calibrazione è di scrivere le equazioni di proiezione che legano le coordinate conosciute di un insieme di alcuni punti 3D e le loro relative proiezioni, e risolverle per i parametri della macchina. Al fine di conoscere le coordinate di qualche punto 3D, i metodi di calibrazione della macchina si affidano ad una o più immagini di un modello di calibrazione, cioè un oggetto 3D geometricamente noto, possibilmente posto in una posizione nello spazio conosciuta e che genera caratteristiche di immagine che possono essere localizzate accuratamente. La figura (2.2) mostra un modello per la

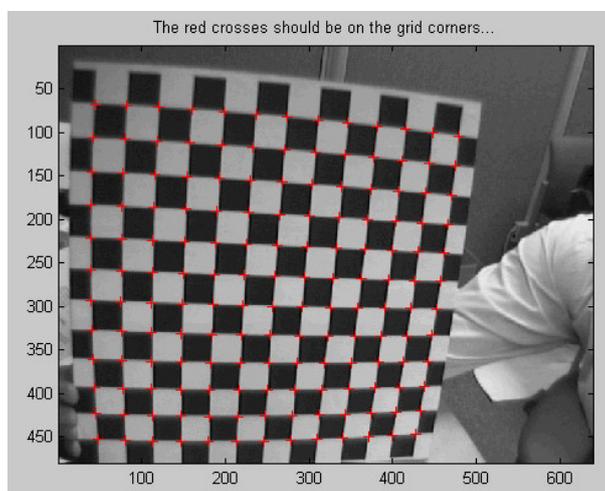


Figura 2.2: Modello per calibrare la camera

calibrazione della macchina, una classica scacchiera. E' facile conoscere la posizione 3D dei vertici di ciascun quadrato, bianco e nero, una volta che la posizione dei due piani è stata misurata, e localizzare i vertici dell'immagine, per esempio come intersezione di linee dell'immagine, grazie al forte contrasto

e alla semplice geometria del modello. Data una o più immagini di un modello di calibrazione dobbiamo stimare

- parametri intrinseci
- parametri estrinseci

L'accuratezza della calibrazione dipende dall'accuratezza delle misure del modello di calibrazione, ovvero dalle tolleranze di costruzione. Per stare sul sicuro, il modello di calibrazione dovrebbe essere costruito con tolleranze di uno o due ordini di grandezza più piccole di quella che si desidera per l'accuratezza della calibrazione. Anche se ci sono tecniche che deducono informazioni 3D dalla scena da macchine non calibrate, efficienti procedimenti di calibrazione aprono la possibilità di usare un ampio raggio di algoritmi esistenti per la ricostruzione e riconoscimento 3D, tutti basandosi sulla conoscenza dei parametri della macchina. Parleremo di due algoritmi per la calibrazione. Il primo metodo ricava i parametri intrinseci ed estrinseci della macchina; il secondo metodo valuta la proiezione della prima matrice, senza risolvere esplicitamente i vari parametri, che sono quindi valutati attraverso funzioni in forma chiusa ² delle variabili di ingresso delle matrici di proiezione. La scelta di ciascun metodo da adottare dipende largamente da quali algoritmi devono essere applicati successivamente.

2.2.1 Calibrazione diretta dei parametri

Iniziamo ad identificare i parametri che devono essere valutati, e poniamo il problema in termini geometrici. Considera il punto 3D, \mathbf{P} , definito dalle sue coordinate $[X^w, Y^w, Z^w]^T$ nel sistema di riferimento nello spazio. Come generalmente accade per la calibrazione, il sistema di riferimento nello spazio è noto. Ciò significa prendere un oggetto di forme semplici definendo tre direzioni mutuamente ortogonali intersecantesi in un punto comune. Questo oggetto è il modello di calibrazione. Siano $[X^c, Y^c, Z^c]^T$ le coordinate di \mathbf{P} nel sistema di riferimento della macchina (con $Z^c > 0$ se \mathbf{P} è visibile). Come di solito accade, l'origine del sistema di riferimento della macchina è il centro della proiezione e Z è l'asse ottico. La posizione e l'orientamento del sistema della macchina non è noto, poiché, a differenza dell'immagine e del sistema di riferimento dello spazio, il sistema di riferimento della macchina non è accessibile direttamente. Questo è equivalente a dire che non conosciamo i parametri estrinseci; cioè la matrice R di rotazione 3x3 e il vettore di

²In matematica, si dice che una funzione è elementare se è scrivibile "in forma chiusa", cioè la trasformata di un arbitrario valore è calcolabile mediante un numero finito di applicazioni delle quattro operazioni elementari dell'aritmetica.

traslazione 3D \mathbf{T} tale che

$$\begin{bmatrix} X^C \\ Y^C \\ Z^C \end{bmatrix} = R \begin{bmatrix} X^W \\ Y^W \\ Z^W \end{bmatrix} + \mathbf{T}$$

Ossia

$$\begin{aligned} X^C &= r_{11}X^W + r_{12}Y^W + r_{13}Z^W + T_x \\ Y^C &= r_{21}X^W + r_{22}Y^W + r_{23}Z^W + T_y \\ Z^C &= r_{31}X^W + r_{32}Y^W + r_{33}Z^W + T_z \end{aligned} \quad (2.2)$$

Assumendo che la distorsione radiale possa essere trascurata, possiamo scrivere l'immagine di $[X^c, Y^c, Z^c]^T$ nel sistema di riferimento dell'immagine come

$$x_{im} = -\frac{f}{s_x} \frac{X^C}{Z^C} + o_x \quad (2.3)$$

$$y_{im} = -\frac{f}{s_y} \frac{Y^C}{Z^C} + o_y \quad (2.4)$$

Per semplicità, e dal momento che non esiste rischio di confusione, possiamo omettere il pedice "im" indicante le coordinate dei pixels dell'immagine e scrivere (x,y) invece di (x_{im}, y_{im}) . Queste coordinate dipendono da 5 parametri intrinseci: f (lunghezza focale), s_x e s_y (dimensione effettiva del pixel orizzontale e verticale), e o_x e o_y (coordinate del centro dell'immagine), e, a causa della particolare forma delle equazioni, i 5 parametri non sono indipendenti. Tuttavia, se supponiamo che $f_x = f/s_x$ e $\alpha = s_y/s_x$ possiamo considerare un nuovo insieme di quattro parametri intrinseci o_x, o_y, f_x e α tutti indipendenti l'uno dall'altro. Il parametro f_x è semplicemente la lunghezza focale espressa tramite l'effettiva dimensione del pixel orizzontale, mentre α , generalmente chiamata rapporto d'aspetto (*aspect ratio*), indica il rapporto matematico tra la larghezza e l'altezza di un'immagine.

Parametri estrinseci

- R, matrice di rotazione 3x3
 - T, vettore di traslazione 3D
-

Parametri intrinseci

- $f_x = f/s_x$, effettiva lunghezza dei pixel lungo l'orizzontale
 - $\alpha = s_y/s_x$, rapporto d'aspetto
 - (o_x, o_y) , coordinate del centro dell'immagine
 - k_1 , coefficiente di distorsione radiale
-

Unendo la (2.2), la (2.3) e la (2.4) si ottiene

$$x - o_x = -f_x \frac{r_{11}X^w + r_{12}Y^w + r_{13}Z^w + T_x}{r_{31}X^w + r_{32}Y^w + r_{33}Z^w + T_z} \quad (2.5)$$

$$y - o_y = -f_y \frac{r_{21}X^w + r_{22}Y^w + r_{23}Z^w + T_y}{r_{31}X^w + r_{32}Y^w + r_{33}Z^w + T_z} \quad (2.6)$$

Queste equazioni aggirano il fatto che il sistema di riferimento della camera sia inaccessibile e collegano direttamente le coordinate dello spazio $[X^w, Y^w, Z^w]^T$ con le coordinate (x, y) del corrispondente punto immagine. Se usiamo un modello di calibrazione noto, entrambi i vettori sono misurabili. Ciò suggerisce che, dato un sufficiente numero di punti sul modello di calibrazione, possiamo provare a risolvere la (2.5) e (2.6) per i parametri incogniti.

Questa è l'idea che sta alla base per il primo metodo di calibrazione, che è articolato in due parti:

1. Assumere che le coordinate del centro dell'immagine siano note, valutare tutti i rimanenti parametri;
2. Trovare le coordinate dell'immagine del centro.

Lunghezza focale, rapporto d'aspetto e parametri estrinseci

Assumiamo che le coordinate del centro dell'immagine siano note. Così possiamo considerare le coordinate traslate $(x, y) = (x - o_x, y - o_y)$. In altre parole, assumiamo che il centro dell'immagine sia l'origine del suo sistema di riferimento. L'idea chiave è di sfruttare le coordinate note di un sufficiente numero di corrispondenti punti immagine e dello spazio. Assumiamo di conoscere la posizione del centro immagine (o_x, o_y) e che la distorsione radiale sia trascurabile, dobbiamo valutare f_x, α, R e \mathbf{T} dai punti immagine (x_i, y_i) $i = 1, \dots, N$, proiezioni di N punti noti dello spazio $[X_i^w, Y_i^w, Z_i^w]^T$ nel sistema di riferimento relativo.

L'osservazione chiave è che (2.5) e (2.6) hanno lo stesso denominatore; perciò, da ciascuna corrispondente coppia di punti $(X^w, Y^w, Z^w), (x_i, y_i)$ possiamo scrivere un'equazione nella forma

$$x_i f_y (r_{21} X_i^w + r_{22} Y_i^w + r_{23} Z_i^w + T_y) = y_i f_x (r_{12} X_i^w + r_{13} Y_i^w + r_{14} Z_i^w + T_x) = 0$$

Poichè $\alpha = f_x / f_y$, dividiamo entrambi i membri per f_y . A questo punto, si può pensare di avere 8 incognite, definite come $\mathbf{v} = (v_1, v_2, \dots, v_8)$

Dove

$$\begin{aligned} v_1 &= r_{21} & v_5 &= \alpha r_{11} \\ v_2 &= r_{22} & v_6 &= \alpha r_{12} \\ v_3 &= r_{23} & v_7 &= \alpha r_{13} \\ v_4 &= T_y & v_8 &= \alpha T_x \end{aligned}$$

Scrivendo l'ultima equazione per N volte, otteniamo un sistema omogeneo lineare di N equazioni.

$$A\mathbf{v} = 0 \tag{2.7}$$

Otteniamo una matrice $N \times 8$ data da

$$A = \begin{bmatrix} x_1 X_1^w & x_1 Y_1^w & x_1 Z_1^w & x_1 & -y_1 X_1^w & -y_1 Y_1^w & -y_1 Z_1^w & -y_1 \\ x_2 X_2^w & x_2 Y_2^w & x_2 Z_2^w & x_2 & -y_2 X_2^w & -y_2 Y_2^w & -y_2 Z_2^w & -y_2 \\ \vdots & \vdots \\ x_N X_N^w & x_N Y_N^w & x_N Z_N^w & x_N & -y_N X_N^w & -y_N Y_N^w & -y_N Z_N^w & -y_N \end{bmatrix}$$

È possibile dimostrare che se $N \geq 7$ e i punti non sono complanari, A ha rango 7, e il sistema ha una soluzione non banale che può essere determinata da una fattorizzazione SVD di A , con $A = UDV^T$, con la colonna di V corrispondente all'unico valore singolo nullo lungo la diagonale di D . Gli effetti del rumore e

dell'inaccurata posizione dell'immagine e dei punti nello spazio fanno sì che il rango di A sia massimo, cioè otto. In questo caso, la soluzione è il vettore singolare destro corrispondente al più piccolo valore singolare.

Una prova rigorosa del fatto che, nel caso ideale (esente dal rumore, perfetta conoscenza delle coordinate) il rango di A sia 7 è troppo complicato da eseguire e si rimanda ad altri testi. Osserviamo che, se il rango effettivo fosse più grande di 7, il sistema avrebbe solo soluzioni banali. Il nostro compito successivo è di determinare il fattore di scala incognito (e da lì i valori dei parametri della macchina) dalla soluzione del vettore $\mathbf{v}=\bar{\mathbf{v}}$. Se chiamiamo γ il fattore di scala, abbiamo

$$\bar{\mathbf{v}} = \gamma(r_{21}, r_{22}, r_{23}, T_y, \alpha r_{11}, \alpha r_{12}, \alpha r_{13}, \alpha T_x) \quad (2.8)$$

Essendo $r_{21}^2 + r_{22}^2 + r_{23}^2 = 1$, dalle prime tre componenti di $\bar{\mathbf{v}}$ abbiamo

$$\sqrt{\bar{v}_1^2 + \bar{v}_2^2 + \bar{v}_3^2} = \sqrt{\gamma^2(r_{21}^2 + r_{22}^2 + r_{23}^2)} = |\gamma|$$

Analogamente $r_{11}^2 + r_{12}^2 + r_{13}^2 = \alpha^2$, ed essendo $\alpha > 0$, la quinta, sesta e settima componente diventano

$$\sqrt{\bar{v}_5^2 + \bar{v}_6^2 + \bar{v}_7^2} = \sqrt{\gamma^2 \alpha^2 (r_{11}^2 + r_{12}^2 + r_{13}^2)} = \alpha |\gamma|$$

Possiamo risolvere queste due equazioni per γ come per il rapporto d'aspetto α . Osserviamo che le prime due righe della matrice di rotazione R e le prime due componenti del vettore di traslazione T adesso possono essere determinate, a partire da un comune segno incognito. Inoltre, la terza riga della matrice R può essere ottenuta come il prodotto vettoriale delle prime due righe ottenute pensate come vettori 3D. Curiosamente, questo implica che il segno della terza riga sia fisso, poiché le variabili di ingresso della terza riga rimangono invariate se i segni di tutte le altre variabili di ingresso delle prime due righe sono invertite. Poiché l'elaborazione della matrice di rotazione stimata, \hat{R}^3 , non tiene conto esplicitamente dei vincoli di ortogonalità, \hat{R} non può essere ortogonale ($\hat{R}\hat{R}^T = 1$). Al fine di rinforzare l'ortogonalità di \hat{R} , si può riesumare la decomposizione dell'onnipresente SVD. Assumi che SVD di \hat{R} sia $R=U\Sigma V^T$. Poiché i tre valori singoli della matrice ortogonale 3x3 sono tutti 1, possiamo semplicemente sostituire D con la matrice identità 3x3, I , in modo che la matrice risultante UIV^T , sia esattamente ortogonale. Alla fine, determiniamo il segno incognito del fattore, γ e otteniamo i valori dei parametri.

A questo scopo, torniamo indietro alla (2.5), per esempio, con x invece di

³Con il cappello " ^ " indicheremo tutti i valori stimati, cioè quelli sensibili di errori.

$x - o_x$, e ricordiamo che per ogni punto $Z^c > 0$ e, perciò, x e $r_{11}X^w + r_{12}Y^w + r_{13}Z^w + T_x$ devono avere segno opposto. Successivamente, è sufficiente controllare il segno di $x(r_{11}X^w + r_{12}Y^w + r_{13}Z^w + T_x)$ per ognuno dei punti. Se è maggiore di zero, i segni delle prime due righe di \hat{R} e delle prime due componenti del vettore di traslazione \mathbf{T} devono essere invertiti. In caso contrario non bisogna fare alcuna ulteriore azione. Un discorso del tutto simile si può fare per la y e $r_{21}X^w + r_{22}Y^w + r_{23}Z^w + T_y$ nella (2.6). A questo punto, abbiamo determinato la matrice di rotazione, \hat{R} , e le prime due componenti del vettore di traslazione, \mathbf{T} , e il rapporto d'aspetto α . Ci siamo lasciati con ancora 2 parametri da determinare: T_z , la terza componente del vettore di traslazione, e f_x , la distanza focale in unità pixels orizzontali. Sia T_z che f_x possono essere ottenuti dalla soluzione dei minimi quadrati grazie ad un sistema di equazioni come la (2.5) o la (2.6), scritte per N punti. Per ottenere ciò, per ciascun punto (x_i, y_i) , possiamo scrivere

$$x_i(r_{31}X_i^w + r_{32}Y_i^w + r_{33}Z_i^w + T_z) = -f_x(r_{11}X_i^w + r_{12}Y_i^w + r_{13}Z_i^w + T_x)$$

Abbiamo un sistema di N equazioni lineari. Costituito da

$$A \begin{bmatrix} T_z \\ f_x \end{bmatrix} = b \quad (2.9)$$

Considerando la (2.5), possiamo trasformarla sotto questa forma

$$x_i T_z + (r_{11}X_i^w + r_{12}Y_i^w + r_{13}Z_i^w + T_x) f_x = -(r_{31}X_i^w + r_{32}Y_i^w + r_{33}Z_i^w) x_i$$

Otteniamo un'equazione del tipo $A\mathbf{v} = b$ dove

$$A = \begin{bmatrix} x_1 & r_{11}X_1^w + r_{12}Y_1^w + r_{13}Z_1^w + T_x \\ \vdots & \vdots \\ x_N & r_{11}X_N^w + r_{12}Y_N^w + r_{13}Z_N^w + T_x \end{bmatrix}$$

$$\mathbf{v} = \begin{bmatrix} T_z \\ f_x \end{bmatrix}$$

$$b = \begin{bmatrix} -x_1(r_{31}X_1^w + r_{32}Y_1^w + r_{33}Z_1^w + T_x) \\ -x_2(r_{31}X_2^w + r_{32}Y_2^w + r_{33}Z_2^w + T_x) \\ \vdots \\ -x_N(r_{31}X_N^w + r_{32}Y_N^w + r_{33}Z_N^w + T_x) \end{bmatrix}$$

La soluzione ai minimi quadrati del problema (\hat{T}_z, \hat{f}_x) è

$$\begin{pmatrix} \hat{T}_z \\ \hat{f}_x \end{pmatrix} = (A^T A)^{-1} A^T b$$

La matrice $A^T A$ è del tipo 2×2 , per cui utilizzare questo metodo o la fattorizzazione QR, avendo solo due colonne, quindi due iterazioni, non dà grandi differenze, a meno che non ci sia un problema di mal condizionamento. Rimane da discutere come possiamo effettivamente acquisire un'immagine di N punti di coordinate spaziali note, e posizionare gli N punti immagine corrispondenti accuratamente. Il modello riportato nella figura (2.2) consiste di alcune griglie ortogonali equamente spaziate da quadrati neri e bianchi disegnati su piani perpendicolari. Imponiamo che il sistema di riferimento spaziale sia il sistema di riferimento 3D centrato nell'angolo sinistro in basso della griglia sinistra e con gli assi paralleli alle tre direzioni identificate dal modello di calibrazione. Se la dimensione orizzontale e verticale delle superfici e l'angolo tra le superfici sono conosciute con grande accuratezza (per costruzione), allora le coordinate 3D dei vertici di ciascuno dei quadrati nel riferimento spaziale può essere facilmente e accuratamente determinato attraverso semplice trigonometria.

2.3 Computer Vision

La ricostruzione della superficie degli oggetti è una speciale disciplina della computer vision.

La disciplina è indirizzata al recupero della superficie di un oggetto, o al calcolo della distanza tra i sensori. Il processo di ricostruzione di forma di un oggetto tridimensionale spesso contiene anche procedure di processo d'immagine o di analisi del modello (rilevamento dei bordi).

2.3.1 Ricostruzione della forma

Nella visione artificiale la forma degli oggetti tridimensionali viene trattata sulla base dei dati visivi. Questi dati sono ottenuti da sensori ottici, che riproducono una scena statica o dinamica. In una scena statica nessuno degli oggetti si muove in un intervallo di tempo in cui si scattano le foto e neppure avvengono cambiamenti di luminosità o nei parametri della macchina. Durante una scena dinamica invece tali cambiamenti o movimenti possono avvenire durante intervallo di tempo. Per esempio, durante l'intervallo di tempo in cui si scattano foto di una scena dinamica la macchina o le macchine possono essere fissate spazialmente e i parametri intrinseci della macchina (esempio lunghezza focale) possono variare. Se un movimento della macchina non può essere escluso allora ciò è anche conosciuto come acquisizione dell'immagine dinamica.

2.3.2 Metodo di calibrazione di Tsai

Un numero di tecniche per la calibrazione geometrica di sensori di macchine fotografiche è nota dalla letteratura della visione artificiale e include il trattamento della distorsione della lente. A questo punto una di queste tecniche è scelta come esempio e spiegata. La valutazione e lo sviluppo ulteriore di tecniche di calibrazione è un argomento molto importante della computer vision. La tecnica di calibrazione di R.Y. Tsai (1986) è tra le più frequenti applicazioni per quanto riguarda la computer vision. Questo metodo include anche la determinazione dei coefficienti k_1 e k_2 della distorsione radiale delle lenti, e del fattore di scala s_x . Il metodo richiede almeno sette punti di calibrazione non-complanari, e accuratamente rilevati che vengono dati in qualsiasi arbitraria, ma non conosciuta, configurazione geometrica. Vengono spiegati sette passi di processo del metodo.

1. *Calcolo delle coordinate del sensore dalle coordinate del buffer immagine.*
2. *Calcolo di sette parametri per la trasformazione delle coordinate del sensore in coordinate dello spazio.*
3. *Calcolo delle componenti Y del vettore di traslazione.*
4. *Determinazione del fattore di scala s_x .*
5. *Calcolo della matrice di rotazione e delle componenti X del vettore di traslazione.*
6. *Approssimazione della lunghezza focale e componenti Z del vettore traslazione senza considerare la distorsione della lente.*
7. *Calcolo di soluzioni esatte per la lunghezza focale, le componenti Z del vettore traslazione, e i coefficienti di distorsione radiale.*

Capitolo 3

Applicazioni sperimentali

Si va ora ad analizzare un modello reale per la calibrazione di una macchina fotografica. Partendo da dei punti nello spazio, questi appartengono ad un oggetto del quale si conoscono le coordinate principali. Per individuare i punti nello spazio si può utilizzare, come supporto per calibrare facilmente la macchina, una scacchiera. Per spostare il sistema di riferimento da quello dell'oggetto nello spazio all'immagine riprodotta nella macchina fotografica si applica un calcolo matriciale, consistente in una rotazione e una traslazione.

Per analizzare ed eseguire il calcolo numerico si utilizza uno specifico software, MATLAB (Matrix Laboratory). Questo programma è un linguaggio di alto livello e l'ambiente interattivo usato da milioni di ingegneri e scienziati in tutto il mondo. Consente di esplorare, visualizzare idee e di mettere in collaborazione diverse discipline, incluse l'elaborazione di segnali e di immagini. Nel capitolo 1 sono riportati alcuni metodi base ed elementari per implementare matrici ed eseguire le più semplici operazioni e fattorizzazioni tra esse, utili per lo svolgimento dell'elaborato di studio.

3.1 Calcolo coordinate punti nello spazio

Per conoscere le coordinate (X,Y,Z) di un oggetto nello spazio, possiamo realizzare una function in Matlab che, ipotizzando una scacchiera di lunghezza unitaria, ci da le coordinate di ogni punto caratteristico della scacchiera, ovvero i vertici della scacchiera e le delimitazioni tra bordo bianco e nero.

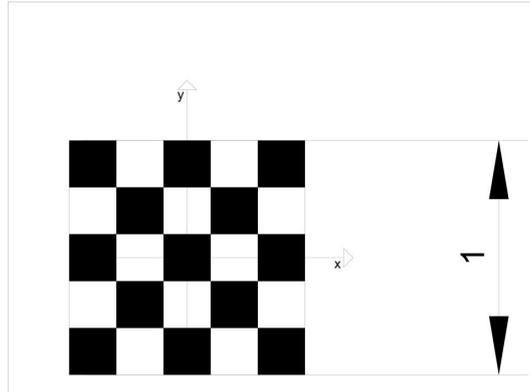


Figura 3.1: Modello usato per calibrare la camera

Prendiamo come sistema di riferimento tre assi coordinati con origine al centro dell'immagine e con l'asse z uscente dal foglio (Figura 3.3). Consideriamo come np il numero di punti considerati, ovvero i vertici principali, di cui vogliamo conoscere le coordinate.

Una possibile function può essere questa:

```
function X = matdati(np)
%MATDATI costruzione del dataset scacchiera
% X=matdati(n) fornisce le coordinate 3D dei punti

if rem(np,2), error('np deve essere pari'), end

v = [-np/2:np/2];
riga1 = repmat(v,1,np+1);
riga2 = kron(v,ones(1,np+1));
riga3 = zeros(1,(np+1)^2);
X = [riga1;riga2;riga3]/np;
```

Dopo aver realizzato ciò, otteniamo una matrice di $(np + 1)^2$ valori, che corrispondono al numero di punti che possiamo considerare. Dato che i punti non possono essere complanari, in realtà si usano due scacchiere sovrapposte. Bisogna anche chiarire che per sperimentare l'accuratezza degli algoritmi

sono stati generati dati sintetici, mentre nella realtà applicativa i dati sono misurati.

3.2 Passaggio dallo spazio alla camera

Ora c'è bisogno di una function che, tramite una rotazione ed una traslazione, sposta il sistema di riferimento dall'oggetto nello spazio alla macchina fotografica.

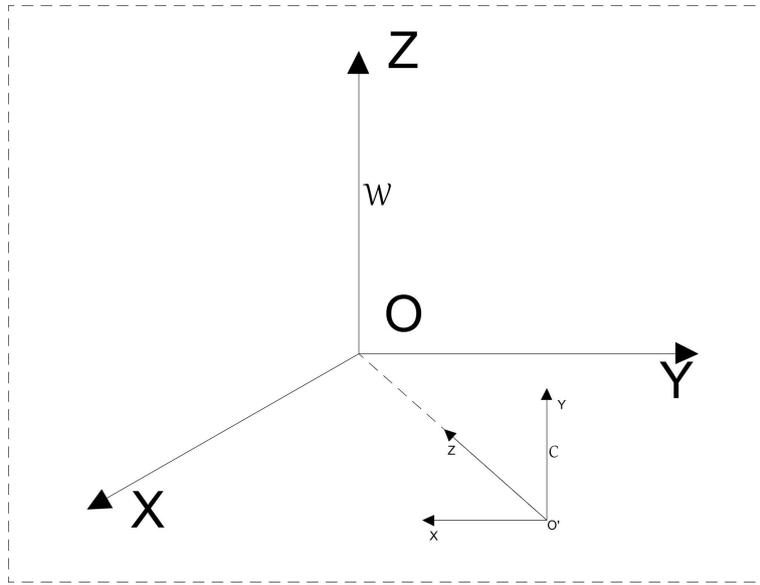


Figura 3.2: Passaggio da il sistema di riferimento dell'oggetto nello spazio (W) a quello della camera (C).

Si indica con W la matrice ottenuta con l'algoritmo "matdati", ovvero i punti specifici dell'oggetto nello spazio. Così facendo si ottiene una matrice del genere

$$W = \begin{bmatrix} X_1^w & \dots & X_N^w \\ Y_1^w & \dots & Y_N^w \\ Z_1^w & \dots & Z_N^w \end{bmatrix} \quad (3.1)$$

Per passare alle coordinate della camera dobbiamo eseguire un calcolo di questo tipo

$$C = RW + Tu^t \quad (3.2)$$

dove u è un vettore a N dimensionale che ha tutte le componenti uguali a 1. Il vettore di traslazione \mathbf{T} ha una forma (considerando la Figura 3.2)

$$T = O - O'$$

Per quanto riguarda la matrice di rotazione R , la si può considerare come costituita da 3 vettori $R = [v_1 \ v_2 \ v_3]$ ortonormali, cioè che $\langle v_i, v_j \rangle = \delta_{ij}$. I vettori devono essere ortonormali perchè la matrice sia ortogonale.

Trovare la direzione di z , essendo anche la direzione di \mathbf{T} , solo in verso opposto, è facile poiché è praticamente un vettore congiungente i centri dei due sistemi di riferimento.

Per cui per trovare il vettore v_3 che rappresenta l'asse z $v_3 = -\frac{T}{\|T\|}$. Per trovare l'asse y , c'è bisogno di richiamare alla mente concetti base di geometria. Consiste semplicemente nel trovare un vettore perpendicolare a z , e verticale.

Un piano si può trovare con un punto e un vettore normale. Per trovarlo

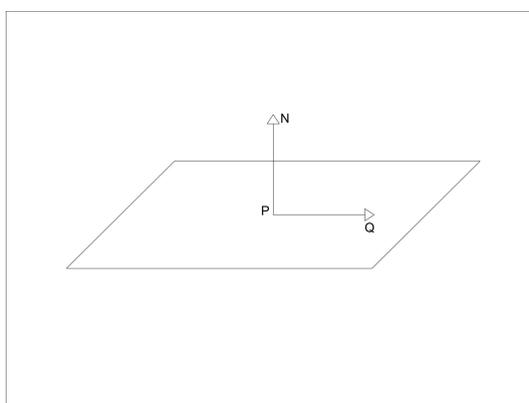


Figura 3.3: Elementi per l'individuazione di un piano

quindi si considera un generico punto P nel piano, e per ottenere un vettore N , si sa che

$$(Q - P) \perp N \text{ cioè } \langle Q - P, n \rangle = 0$$

oppure

$$\langle Q, n \rangle = \langle P, n \rangle$$

Siccome

$$Q - P = \begin{cases} x - P_x \\ y - P_y \\ z - P_z \end{cases}$$

Dove (P_x, P_y, P_z) sono le coordinate di P . Per cui l'equazione del piano sarà

$$(x - P_x)N_x + (y - P_y)N_y + (z - P_z)N_z = 0$$

Di questo piano serve sapere una direzione verticale ed orizzontale. Per semplicità consideriamo una componente orizzontale poichè non ha componente z , per cui $z = P_z$. Per cui l'equazione del piano diventa

$$(x - P_x)N_x + (y - P_y)N_y = 0 \quad (3.3)$$

Se diamo un valore alla x si ottiene

$$\begin{cases} x = 1 \\ y = (P_x - x) \frac{N_x}{N_y} + P_y \\ z = P_z \end{cases}$$

Facendo così si trova un vettore che sta nel piano ortogonale all'asse z , ed è orizzontale. Per cui il vettore che ha questi requisiti è $v_2 = \frac{\tilde{v}_2}{\|\tilde{v}_2\|}$, ottenuto con una sorta di ortogonalizzazione di Gram-Schmidt per via geometrica perché siamo in \mathbb{R}^3 .

Per trovare l'asse x si può semplicemente fare il prodotto vettoriale tra i due vettori appena trovati.

$$v_1 = v_2 \wedge v_3$$

Un problema sorge nel sapere se v_2 ha la direzione corretta. Con questo procedimento si ha la certezza che v_2 sia ortogonale a z , ma non si sa se il verso che ha assunto è quello corretto. Per verificarlo basta notare se il verso di v_1 se punta in alto o in basso. Per cui se $(v_1)_z < 0 \rightarrow v_2 = -v_2$.

Per cui ora conosciamo gli elementi della matrice di rotazione R .

Dei dati noti sono le coordinate dei punti immagine

$$X^{im} = \begin{bmatrix} x_{im} \\ y_{im} \end{bmatrix}$$

Le coordinate x_{im} e y_{im} si calcolano con (2.3) e (2.4).

3.3 Implementazione su Matlab

Come visto in precedenza la matrice che si genera con "matdati" indica tutti i punti caratteristici della scacchiera nello spazio. Ora per completare l'operazione

$$C = RW + Tu^t$$

ed ottenere il passaggio al sistema di coordinate della macchina fotografica possiamo ideare un algoritmo da implementare su Matlab che fornisce la matrice R

```
function R = assi(P)
%assi Fornisce la matrice R di rotazione
T = [0;0;norm(P)];
```

```

v3=-T;
v3=v3/norm(v3);
v1= [1-T(1); (T(1)-1)*v3(1)/v3(2); 0];
v1 = v1/norm(v1);
v2=[v3(2)*v1(3)-v3(3)*v1(2); -(v3(1)*v1(3)-v1(1)*v3(3)); v3(1)*v1(2)
-v3(2)*v1(1)];
R=[v1 v2 v3]';

```

Questa function presuppone di aver inserito le coordinate della nuova origine, quindi il punto \mathbf{P} , e quindi di poter calcolare il vettore di traslazione \mathbf{T} con le sue coordinate $[x \ y \ z]$.

Fatto ciò, la matrice elabora i vettori base v_1, v_2 e v_3 che costruiscono la matrice di rotazione R . Il vettore v_3 è semplicemente opposto al vettore \mathbf{T} , e viene successivamente normalizzato; il vettore v_1 invece viene ottenuto dalla (3.3), normalizzato anch'esso; l'ultimo vettore invece è ottenuto come prodotto scalare. Questo però non è direttamente ottenibile su Matlab, dobbiamo ricorrere ad un artificio: sapendo che $v_2 = v_3 \wedge v_1$ allora sappiamo anche che

$$v_2 = v_3 \wedge v_1 \rightarrow \begin{bmatrix} v_{31} & v_{32} & v_{33} \\ v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \end{bmatrix}$$

L'ultima riga della matrice è ottenuta considerando i minori della matrice 3×3 . Ad esempio v_{21} è ottenuto eliminando la prima colonna, e calcolando il determinante della sottomatrice $\begin{bmatrix} v_{32} & v_{33} \\ v_{12} & v_{13} \end{bmatrix}$, eliminando una colonna per volta procedendo da sinistra verso destra otteniamo rispettivamente tutti i valori del vettore base v_2 .

Infine in uscita otteniamo la matrice di rotazione R . Dopodiché manca solo ottenere la matrice C , di dimensioni uguale alla matrice W .

Per fare ciò, un algoritmo ideale potrebbe essere

```

function C = mondo2camera(W,R,T)
%Calcolo effettivo di C

N = size(Xw,2);
C = R*W+repmat(T,1,N);

end

```

Con il comando "repmat" facciamo in modo che l'operazione $RW + Tu^t$ sia possibile, ovvero riproduce il risultato del prodotto. Due matrici, per poter essere sommate, devono infatti essere simili, ovvero avere lo stesso numero di righe e colonne, ed essendo \mathbf{T} un vettore, sicuramente non ha le stesse dimensioni della matrice ottenuta dal prodotto tra W e R . Per cui ogni colonna della matrice $R*W$ andrà sommata a \mathbf{T} .

Una volta avvenuto il passaggio nel sistema di riferimento della camera, bisogna conoscere le coordinate dei punti caratteristici dell'oggetto nella foto. Per fare questo ci serviamo della (2.3) e (2.4), riteniamo note la lunghezza focale f , la risoluzione dell'immagine (s_x, s_y) o il rapporto d'aspetto $\alpha = \frac{s_y}{s_x}$ e le coordinate del centro immagine (o_x, o_y) .

Implementando ciò

```
function XIM = immagine(f,sx,sy,ox,oy,C )
%Coordinate del punto immagine
% Inserendo da tastiera lunghezza focale, risoluzione immagine, e
% coordinate del centro dell'immagine

xim=-((f/sx)*(C(1,)./C(3,)))+ox;
yim=-((f/sy)*(C(2,)./C(3,)))+oy;

XIM=[xim;yim];

end
```

Fatto ciò si ottiene una matrice XIM che individua le coordinate dei punti nella foto, corrispondenti alle coordinate dei punti nello spazio, ottenuti invece con la matrice W .

Per ottenere la matrice A vista nel paragrafo (2.2.1), si può costruire un algoritmo simile a questo

```
function A = Matrice (XIM,W)
%Costruzione matrice A

a1=XIM(1,).*W(1,);
a2=XIM(1,).*W(2,);
a3=XIM(1,).*W(3,);
a4=XIM(1,);
```

```

a5=-XIM(2,:).*W(1,:);
a6=-XIM(2,:).*W(2,:);
a7=-XIM(2,:).*W(3,:);
a8=-XIM(2,:);

A=[a1;a2;a3;a4;a5;a6;a7;a8]';

end

```

Con il comando `.*` o `./` si intende un prodotto o divisione elemento per elemento tra due matrici.

Calcolando il rango di questa matrice si nota che è pari a 5, mentre invece, secondo quanto esposto nella teoria, dovrebbe risultare 7. Questo accade perché lasciando la Z nulla nella matrice W , due colonne della matrice A sono nulle, quindi è evidente che il rango sia 5. Dalla teoria si evince che per ottenere rango 7, come detto in precedenza, i punti non devono essere complanari. Per cui ipotizziamo di inserire nei calcoli una seconda scacchiera sollevando di una $z=1$

```

Xw1 = matdati(np);
Xw2 = [Xw1(1:2,:); ones(1,size(Xw1,2))];
Xw = [Xw1 Xw2];

```

Per risolvere il sistema (2.7) e trovare quindi \mathbf{v} , si deve usufruire della decomposizione SVD

```

A = MatriceA(Xim,Xw);
b = zeros(N,1);
[U S V] = svd(A);
v = V(:,8);

```

Questo vettore \mathbf{v} corrisponde al vettore (2.8) dove compaiono le 8 incognite che servono per eseguire il processo inverso, ovvero partendo dai soli dati della matrice Xim e Xw , si vuole ottenere \hat{R} , \hat{T} e $\hat{f}x$, confrontare con i valori reali ottenuti in precedenza e verificare l'entità di errore presente. Per estrapolare la matrice \hat{R} si può pensare di realizzare un algoritmo simile

```

gamma = norm(v(1:3));
v=v./gamma;
r2=v(1:3);
Ty=v(4);
alfa=sy/sx;
r1=alfa./v(5:7);
Tx=alfa/v(8);
r3=[r1(2)*r2(3)-r1(3)*r2(2); -(r1(1)*r2(3)-r1(3)*r2(1));
    r1(1)*r2(2)-r1(2)*r2(1)];
Rt=[r1;r2;r3];

```

Dove per prima cosa si calcola γ , come la norma dei primi tre elementi del vettore \bar{v} . Dopodichè essendo γ un fattore di scala che moltiplica tutto il vettore, dividiamo ogni singolo elemento per il fattore.

Così facendo otteniamo la seconda riga di \hat{R} , il quarto elemento del vettore è la seconda componente di \mathbf{T} , la quinta, sesta e settima componente invece corrispondono alla prima riga della matrice di rotazione, moltiplicata per α che va calcolato come $\frac{s_y}{s_x}$. Fatto ciò rimane l'ottavo elemento del vettore, che corrisponde alla componente x del vettore di traslazione, sempre moltiplicato per il rapporto d'aspetto. La terza ed ultima riga di \hat{R} è ottenibile come prodotto vettoriale tra la prima e la seconda riga della matrice.

I termini mancanti per la verifica sono \hat{T}_x e \hat{f}_x . Questi sono calcolabili tramite la (2.9).

```

w = Xim(1,:)';

q= (Rt(1,:)*Xw)'+Tx;

A1=[w q];

b1=-Xim(1,:)'*(Rt(3,:)*Xw)';

Tz=A1\b1;

```

Per avere un'idea del programma ideato, ovvero della posizione delle due scacchiere nello spazio, si può pensare di eseguire un "plot" della matrice X_{im} . Le coordinate dei punti sono indicate ciascuna con un cerchietto. Il comando per eseguire ciò è

```
figure(1)
plot(Xim(1,:),Xim(2,:),'o')
```

Ovviamente i punti cambiano a seconda del punto \mathbf{P} che impostiamo inizialmente, soprattutto dalla componente z . Avendo impostato una distanza di 1 unità tra una scacchiera e l'altra, e posizionando la macchina fotografica con l'origine del relativo sistema di riferimento in un punto $\mathbf{P} = [4; 5; 2]^T$, si può immaginare, già senza eseguire il plot, che la scacchiera più alta si veda in prospettiva da un punto rialzato.

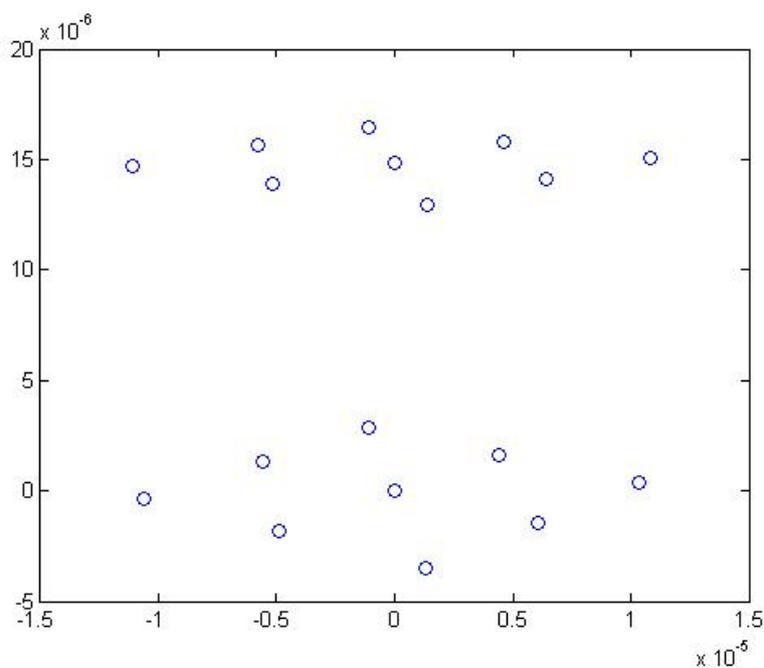


Figura 3.4: Plot in 2D delle due scacchiere

Per avere un plot 3D, ed ottenere una visuale dei due modelli di calibrazione lungo i 3 assi, si imposta il comando

```
figure(2)
plot3(Xc(1,:),Xc(2,:),Xc(3,:), 'o')
xlabel('x')
ylabel('y')
zlabel('z')
```

In questo caso impostiamo le coordinate dei punti dalla matrice Xc , per cui la visuale di cui godremmo sarà quella rispetto alla macchina fotografica.

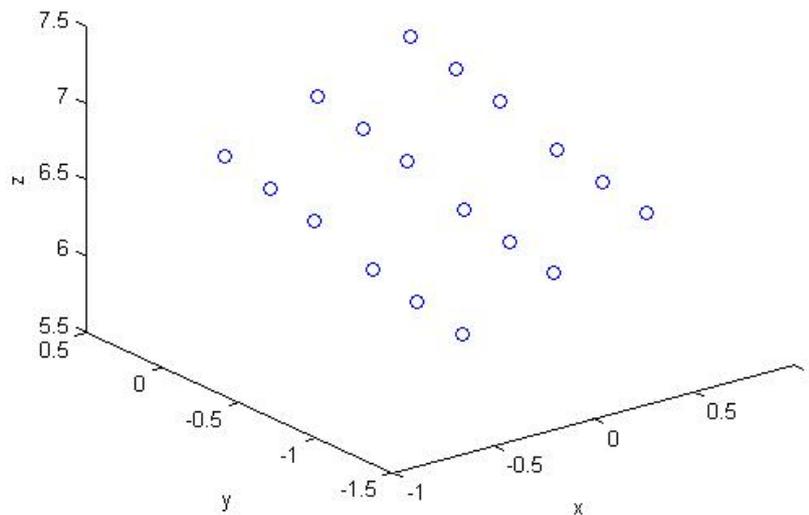


Figura 3.5: Plot in 3D delle due scacchiere

3.3.1 Dimostrazione numerica

Una volta implementato tutto il procedimento si possono fare dei tentativi per vedere, prima di tutto, se il programma appena creato sia corretto, ovvero partendo da un punto \mathbf{P} e da un certo numero di np , calcolare la matrice Xim . Fatto ciò si dovrà effettuare il processo inverso al precedente e constatare se l'errore tra la matrice R e la Rt sia effettivamente molto piccolo. Un'altra modalità di verifica può essere quella consistente nell'impostare il punto \mathbf{P} , eseguire il plot della matrice Xc e Xim , e vedere come vari il punto di vista delle due scacchiere. Un modo semplificato per lavorare è impostare direttamente nell'editor di Matlab tutto il procedimento di calcolo

```

np = 2; % pari
P = [2;3;6];
f = 0.05;
sx = 500;
sy = 500;
ox = 0;
oy = 0;
R = assi(P);
Xw1 = matdati(np);
Xw2 = [Xw1(1:2,:); ones(1,size(Xw1,2))];
Xw = [Xw1 Xw2];
N = size(Xw,2);
T = [0;0;norm(P)];
Xc = mondo2camera(Xw,R,T);
Xim = immagine(f,sx,sy,ox,oy,Xc);
figure(1)
plot(Xim(1,:),Xim(2,:), 'o')
figure(2)
plot3(Xc(1,:),Xc(2,:),Xc(3,:), 'o')
xlabel('x')
ylabel('y')
zlabel('z')
A = MatriceA(Xim,Xw);
b = zeros(N,1);
[U S V] = svd(A);
v = V(:,8);
gamma = norm(v(1:3));
v=v./gamma;

```

```

r2=v(1:3)';
Ty=v(4);
alfa=sy/sx;
alfat = norm(v(5:7));
r1=v(5:7)'/alfa;
Tx=v(8)/alfat;
r3=[r1(2)*r2(3)-r1(3)*r2(2) -(r1(1)*r2(3)-r1(3)*r2(1)) r1(1)*r2(2)-r1(2)*r2(1)];
Rt=[r1;r2;r3];
w = Xim(1,:)';
q= (Rt(1,:)*Xw)'+Tx;
A1=[w q];
b1=-Xim(1,:)'.*(Rt(3,:)*Xw)';
Tz=A1\b1;

```

I plot 2D e 3D relativi a questo punto \mathbf{P} sono rispettivamente

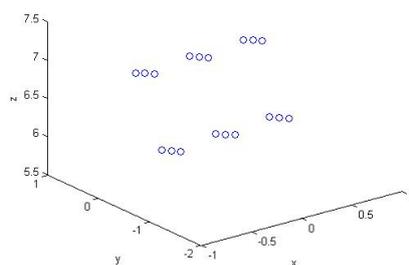


Figura 3.6: Plot in 3D delle due scacchiere

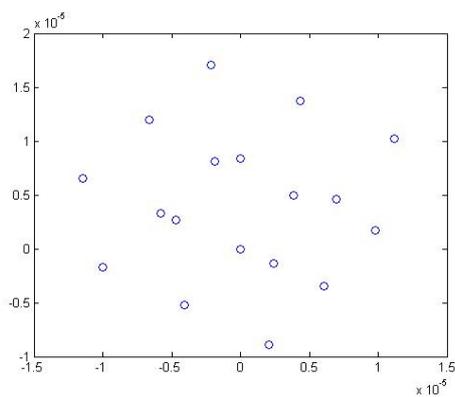


Figura 3.7: Plot in 2D delle due scacchiere

Un metodo utile per capire il rango della matrice A può essere quello di realizzare un grafico semilogaritmico che rappresenti i valori singolari della matrice. Essendo il numero di condizionamento il rapporto tra il primo valore singolare e l'ultimo non nullo, sono da analizzare il primo e il settimo valore.

```
figure(3)
semilogy(svd(A), 'o-')
```

Il comando 'o-' è necessario per unire tutti i punti del grafico con una linea. Si ottiene nel nostro caso

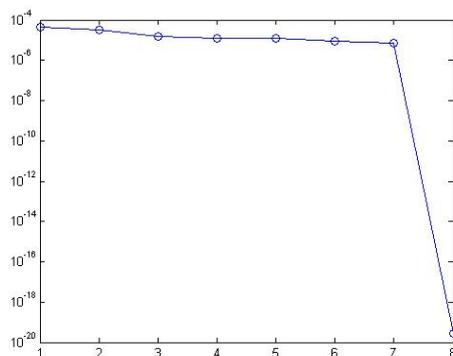


Figura 3.8: Rappresentazione in scala semi-logaritmica dei valori singolari

Dal grafico risulta evidente il gradino tra il settimo e l'ottavo valore; quest'ultimo si aggira nell'ordine di 10^{-19} , per cui non è numericamente nullo, ma ha un valore talmente piccolo che per Matlab può considerarsi come pari a zero. Per questo motivo il rango viene considerato 7, essendo il settimo valore singolare l'ultima cifra non nulla.

Calcolo con errori

Un altro esperimento possibile è quello di inserire dei rumori nei dati, per vedere come cambino i valori, con le matrici perturbate dagli errori.

```
np = 2; % pari
P = [2;3;6];
f = 0.05;
sx = 500;
sy = 500;
```

```

ox = 0;
oy = 0;
R = assi(P);
Xw1 = matdati(np);
Xw2 = [Xw1(1:2,:); ones(1,size(Xw1,2))];
Xw = [Xw1 Xw2];
N = size(Xw,2);
T = [0;0;norm(P)];
Xc = mondo2camera(Xw,R,T);
Ximesatto = immagine(f,sx,sy,ox,oy,Xc);
dsv = 10.^[-16:-2]';
nds = size(dsv,1);
err = zeros(nds,1);
alfa=sy/sx;
vesatto = [R(2,:)' ; T(2); alfa*R(1,:)' ; alfa*T(1)];
for i=1:nds
    ds=dsv(i)
    Xim = Ximesatto + ds*norm(Ximesatto(:))*sqrt(2*N)*randn(2,N);
    figure(1)
    plot(Xim(1,:),Xim(2:,:),'o')
    figure(2)
    plot3(Xc(1,:),Xc(2,:),Xc(3:),'o')
    xlabel('x')
    ylabel('y')
    zlabel('z')
    A = MatriceA(Xim,Xw);
    b = zeros(N,1);
    [U S V] = svd(A);
    v = V(:,8);
    gamma = norm(v(1:3));
    v=v./gamma;
    if min(v(1:3).*vesatto(1:3))<0, v=-v; end
    err(i)= norm(v-vesatto,inf);
    r2=v(1:3)';
    Ty=v(4);
    alfat = norm(v(5:7));
    r1=v(5:7)'/alfa;
    Tx=v(8)/alfat;
    r3=[r1(2)*r2(3)-r1(3)*r2(2) -(r1(1)*r2(3)-r1(3)*r2(1))
        r1(1)*r2(2)-r1(2)*r2(1)];
    Rt=[r1;r2;r3];
    w = Xim(1,:)' ;
    q= (Rt(1,:)*Xw)'+Tx;
    A1=[w q];
    b1=-Xim(1,:)' .*(Rt(3,:)*Xw)';
    Tz=A1\b1;
end
loglog(dsv,err,'o-')
grid

```

Come prima modifica al programma precedente, che non prevedeva perturbazioni, dobbiamo dare degli ordini di errore "dsv" che vanno da un minimo di 10^{-16} ad un massimo di 10^{-2} . Successivamente si registra un valore esatto del vettore soluzione "vesatto" e della matrice "Ximesatto". Dopodichè, tramite un ciclo for, calcoliamo per ogni entità di rumore un vettore v ; l'errore sarà dato dalla norma infinito, necessaria per ottenere il valore massimo, della sottrazione $(v - vesatto)$. A questo punto è realizzabile un grafico nel quale sulle ordinate riportiamo il valore dell'errore e sulle ascisse il valore del rumore.

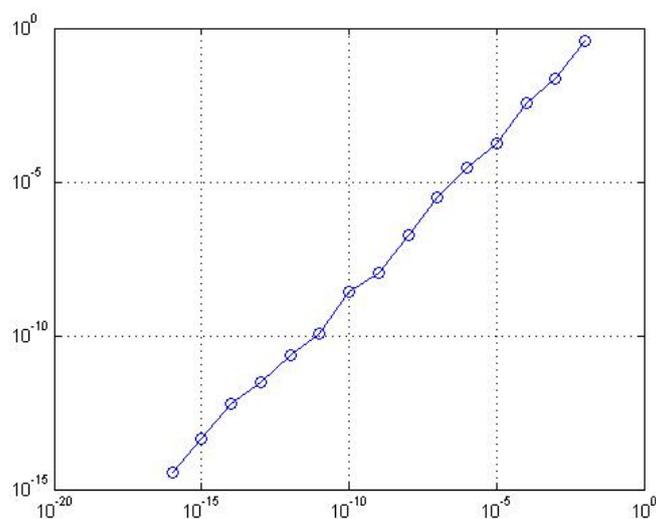


Figura 3.9: Rappresentazione in scala logaritmica degli errori

La formula che dobbiamo sfruttare è (1.17). La usiamo per calcolare la matrice Xim affetta da errori.

Conclusioni

L'obiettivo perseguito da questo elaborato consiste nella realizzazione del passaggio da un sistema di riferimento di spaziale, ad un sistema di riferimento relativo ad una macchina fotografica. Nello specifico si vuole applicare questo procedimento ad un oggetto, con caratteristici punti e relative coordinate spaziali, del quale si vogliono conoscere le rispettive coordinate dei punti nella foto.

A seconda del punto di origine del sistema di riferimento della macchina con il plot del programma, si possono analizzare le coordinate dei punti caratteristici dell'immagine delle due scacchiere, con diversi punti di vista.

Il programma comprende anche una parte relativa alla verifica dei dati ottenuti, dove si calcola una seconda matrice di rotazione Rt ed un secondo vettore di traslazione con coordinate rispettivamente pari a T_x, T_y, T_z . A seconda di quanto differiscano dalla matrice originaria R e dal vettore di traslazione T , si può risalire a quanto siano precisi i calcoli. Per quanto concerne l'algoritmo relativo agli errori, analizzando il grafico, è evidente che con un rumore dell'ordine di 10^{-2} si abbiano errori abbastanza alti. È quindi evidente che con questo metodo di calibrazione si riscontrino errori troppo elevati per le misure che è possibile effettuare su una foto.

Ad esempio, se si vuole considerare un angolo particolare della scacchiera, per ottenere dei risultati sufficientemente precisi, dovremmo avere un rumore molto basso, perchè l'errore sia dell'ordine di 10^{-7} , che non è effettivamente misurabile.

In conclusione si può affermare che il metodo di calibrazione illustrato in questa tesi, non sia sufficientemente preciso, per cui per ottenere dei risultati soddisfacenti si dovrà cercare un metodo, magari più complesso, ma più efficiente.

Bibliografia

- [1] Björck, Å. (1996), *Numerical methods for least squares problems*, SIAM.
- [2] Valeriano Comincioli (1995), *Analisi numerica, metodi modelli applicazioni*, Mc Graw-Hill.
- [3] G. Rodriguez (2008), *Algoritmi numerici*, Pitagora editrice Bologna.
- [4] G. Rodriguez (2010), *Introduzione alla matematica applicata e computazionale*, Pitagora Editrice Bologna
- [5] G. Monegato (1998), *Fondamenti di calcolo numerico*, Elut.
- [6] Steven C.Chapra, Raymond P.Canale, *Numerical methods for engineers*, Mc Graw-Hill
- [7] *The student edition of Matlab*, The math works inc. (1997).
- [8] Thomas Luhmann, Stuart Robson, Stephen Kyle, Ian Harley (2006), *Photogrammetry*, Whittles Publishing.
- [9] Emanuele Trucco, Alessandro Verri (1998), *Introductory techniques for 3D computer vision*, Prentice Hall
- [10] .Klette, K.Schluns, A.Koschan (1998), *Computer Vision*, Springer.
- [11] .Gruen, T.S.Huang (2001), *Calibration and orientation of cameras in computer vision*, Springer