

Algoritmi di calibrazione per telecamere

Padova, 25 novembre 2009

Omar Codemo

Senior Product Engineer

Application Engineering Dept.

e-mail: omar.codemo@mermecgroup.com

Global Footprint




mermec france

- Place: Marseilles, France
- Activity: R&D
- Employees: 10
- Area: 180 sqm




tecnogamma

- Place: Treviso, Italy
- Activity: R&D, Manufacturing
- Employees: 65
- Area: 1,500 sqm

GLOBAL LOCATIONS

mermec

MAIN OFFICE

- Monopoli, Italy

BRANCH OFFICE

- Norway

REP OFFICES

- Roma, Italy
- Switzerland
- China
- Korea

tecnogamma

MAIN OFFICE

- Treviso, Italy

R&D LAB

- Palermo, Italy

REP OFFICE

- China

imagemap

MAIN OFFICE

- Columbia, USA



REP OFFICES

- Chicago
- Brasil
- UK
- Italy
- Spain
- Unghery
- Poland
- China
- Taiwan
- Australia

mermec france

MAIN OFFICE

- Marsiglia, France

imagemap

- Place: Columbia, USA
- Activity: R&D, Manufacturing
- Employees: 36
- Area: 4,325 sqm




mermec

- Place: Monopoli, Italy
- Activity: R&D, Manufacturing
- Employees: 225
- Area: 7,398 sqm

MER MEC Group in Numbers



DIAGNOSTICS

- 72 main clients in 35 countries worldwide
- 41 unique measuring systems in 6 distinct product families
- 8 Integrated diagnostic vehicles manufactured & sold worldwide
- 350,000 km of line measured by MER MEC Group's customers
- 436 measuring systems sold worldwide

SIGNALLING

- 2,100 km of line equipped with SSC SCMT signaling systems
- SSC (1426 km)
 - 1083 LEU
 - 1083 Transponders
 - 212 Transponder Tags
- SCMT (667 km)
 - 100 boxes
 - 508 LEU
 - 3673 Eurobalise
- 2000 entrance in signaling market with a 4 year time to market.

ADVANCED SERVICES

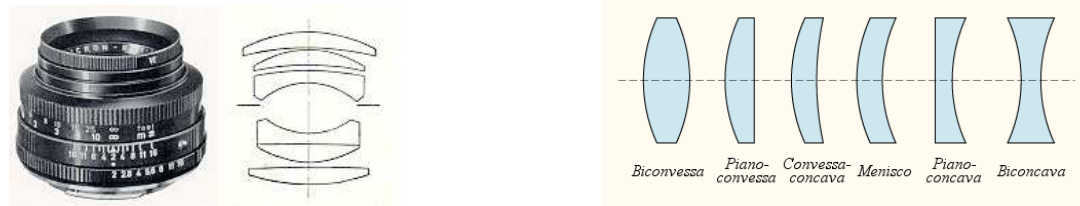
- Measuring Services
 - 2 Hi-Rail diagnostic vehicles with variable gauge capability
 - 1 measuring carriage for use on standard gauge rail lines
 - 5 market segments served (Metro, Light Rail, Ordinary Rail, High-Speed, Heavy-Haul)
 - Integrated services for data acquisition, analysis and maintenance planning

IRON & STEEL INDUSTRY R & D

- 17 main clients in 8 countries worldwide
- 3 industrial sectors served: "Long Products", "Ring Rollings" and "Large Open-die Forgings"
- 35 Systems installed
- 154 people involved in R&D
- MUIR authorized research laboratory since 2001
- 15% of revenues invested annually in R&D
- 14 new products commercially launched between 2004-2007

Definizione del problema

L'uso di telecamere per l'osservazione di una scena richiede l'uso di obiettivi costituiti da un insieme di lenti.



L'uso di tali lenti introduce una distorsione che bisogna compensare prima di poter procedere ad una qualsiasi misurazione.

Inoltre si osserva una scena 3D tramite un sensore (CCD) che produce un'immagine 2D. Questo porta all'introduzione di una mappatura detta *proiezione prospettica*:

$$S \subset 3D \rightarrow I \subset 2D$$

Infine il frame grabber può introdurre distorsioni (line jitter, fattori di scala).

Distorsione delle lenti

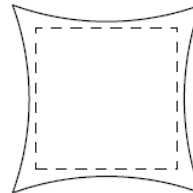
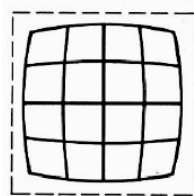
Distorsioni non lineare

Le distorsioni introdotte dalle lenti sono tipicamente non lineari. Si possono suddividere in:

- › radiale;
- › altre distorsioni delle lenti;

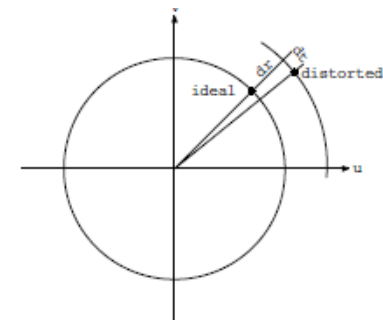


La distorsione radiale sposta i punti fuori dal centro ottico (Barrel o Pincushion) ed è causata principalmente da deviazioni dei raggi ai bordi della lente:



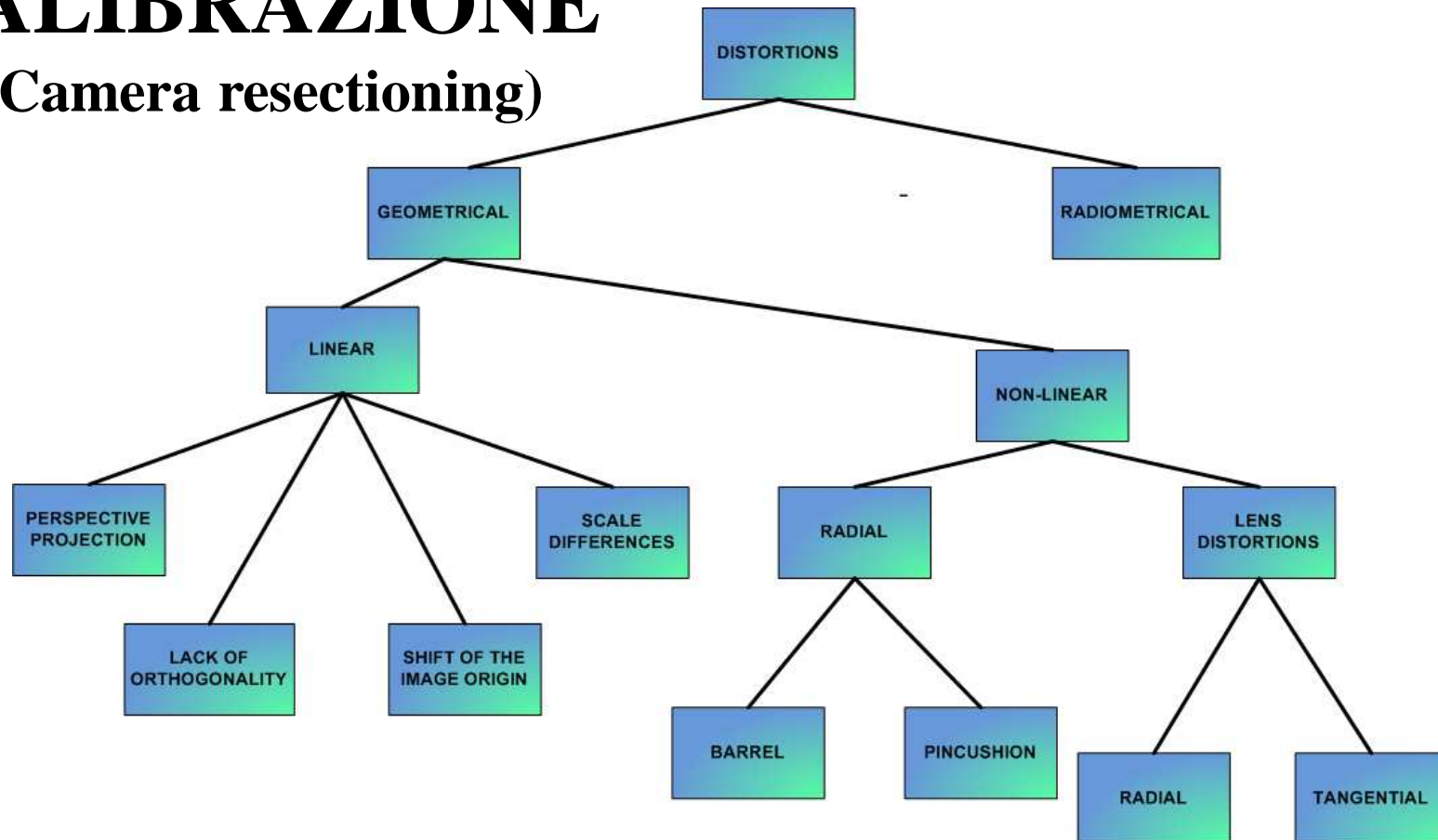
La distorsione delle lenti si può scomporre principalmente in:

- › Radiale (dr);
- › Tangenziale (dt).



Le seguenti distorsioni vanno compensate tramite un processo detto:

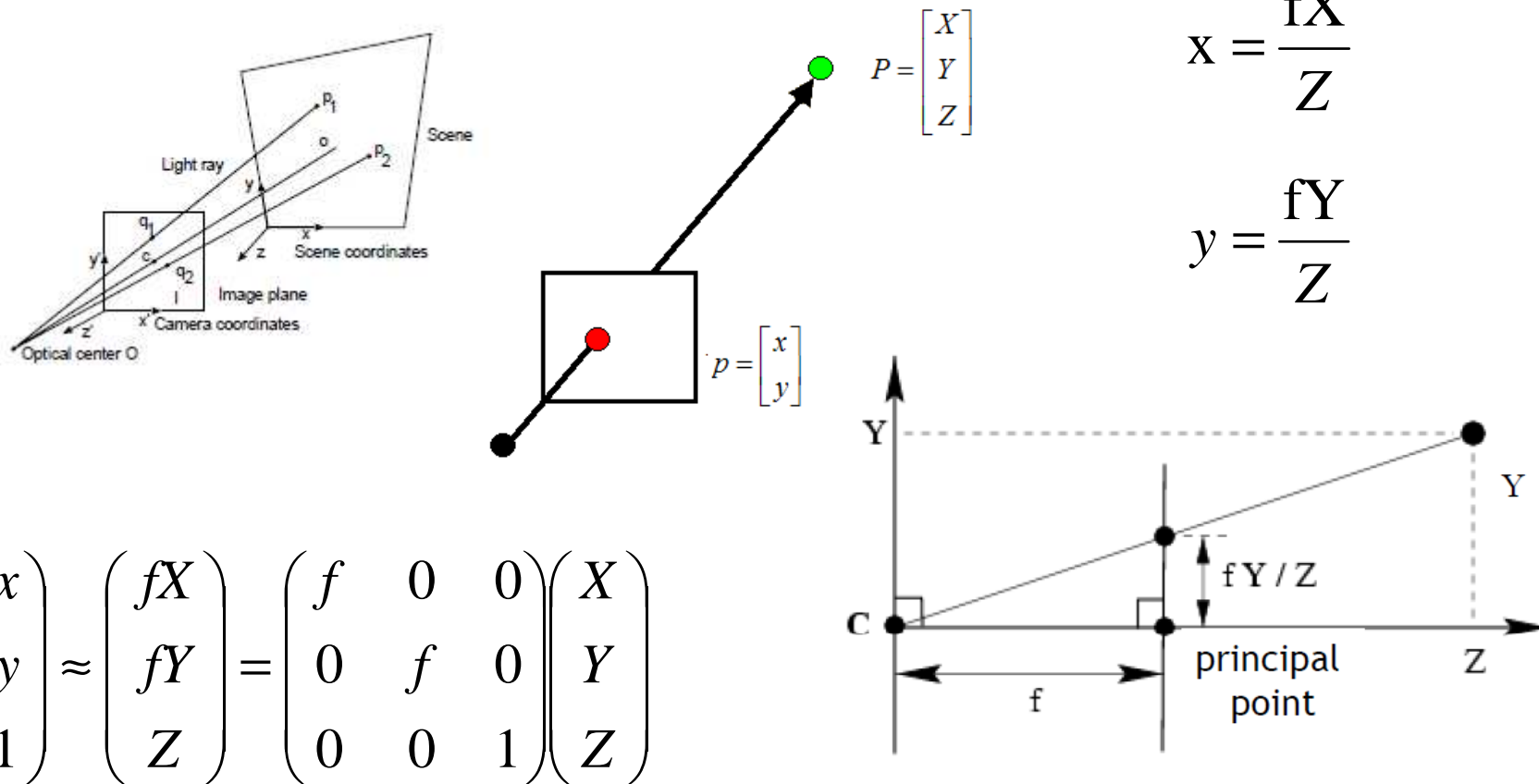
CALIBRAZIONE (o Camera resectioning)



Proiezione prospettica

Distorsione lineare

La proiezione prospettica non rispetta distanze e angoli perciò può essere considerata come una forma di distorsione. Tale distorsione si può esprimere tramite una matrice:



1. La dimensione di un oggetto diminuisce all'allontanarsi dal centro di proiezione;
2. La dimensione della proiezione orizzontale diminuisce al ruotare dell'oggetto;
3. Preserva le rette;
4. Sono possibili più punti di fuga;

Pinhole camera model

Il modello Pinhole è molto semplice ed è costituito da una rototraslazione di un corpo rigido seguita da una proiezione prospettica.

Altre distorsioni a parte la proiezione prospettica (es. quelle delle lenti) non sono modellizzate. Nonostante ciò è una buona approssimazione.

Dette $[x_i, y_i, z_i]^T$ le coordinate di un punto della scena e $[u_i, v_i]^T$ le appropriate coordinate del punto corrispondente sull'immagine si hanno le coordinate intermedie :

$$\begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \end{bmatrix} = \frac{f}{z_i} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

È necessaria una ulteriore trasformazione per ottenere le coordinate immagine:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} D_U & \tilde{u}_i \\ D_V & \tilde{v}_i \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}$$

Dove (u_0, v_0) tengono conto dello scostamento dell'origine e (D_U, D_V) sono fattori di conversione tra le unità di misura del mondo e quelle immagine, e tengono conto dei diversi fattori di scala.

Modificando la scrittura della prima equazione (aggiungendo un vincolo esplicito) la si può rappresentare in forma di sistema :

$$\begin{bmatrix} \lambda u_i \\ \lambda v_i \\ \lambda \end{bmatrix} = A \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$$

Dove la matrice **A** è una matrice triangonale superiore, definita con 5 parametri.

$$A = \begin{bmatrix} \frac{W}{2 \tan \alpha_u} & k_\gamma & u_0 \\ 0 & \frac{H}{2 \tan \alpha_v} & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} k_u & k_\gamma & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Usando le coordinate omogenee:

$$\lambda \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = A\Pi_0 \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

Dove in molti casi si sottointende λ (dopotutto le coordinate omogenee sono definite a meno di un coefficiente di proporzionalità), e Π_0 è una semplice matrice usata per sottointendere la quarta coordinata omogenea (W) non usata in questa trasformazione.

Definiamo la matrice $E = \begin{bmatrix} \vec{R} \\ t_0 \end{bmatrix}$ per tener conto della rototraslazione. Da cui:

$$\begin{bmatrix} \lambda u_i \\ \lambda v_i \\ \lambda \end{bmatrix} = AE \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = P \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

Parametri intrinseci ed estrinseci

Durante la calibrazione un bersaglio fornisce la corrispondenza tra i punti dell'immagine e quelli dello spazio.

Con la calibrazione si identifica la distanza focale f e il punto principale $(X_0, Y_0)^T$. Tali parametri sono legati all'ottica e sono detti *parametri intrinseci*.

Il bersaglio non si può posizionare in modo perfetto allora si devono identificare anche le relazioni tra le coordinate della telecamera e quelle del bersaglio. Tali parametri sono detti *parametri estrinseci*.

Oltre a questi ci sono un *fattore di scala* che tiene conto del ricampionamento a livello di frame grabber e delle *serie di potenze* per modellare le distorsioni

Parametri intrinseci ed estrinseci

Ponendo $Z_i=0$ per ogni i si ottiene:

$$\lambda \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} r_0 & r_1 & T_X \\ r_3 & r_4 & T_Y \\ r_6 & r_7 & T_Z \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} = H \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix}$$

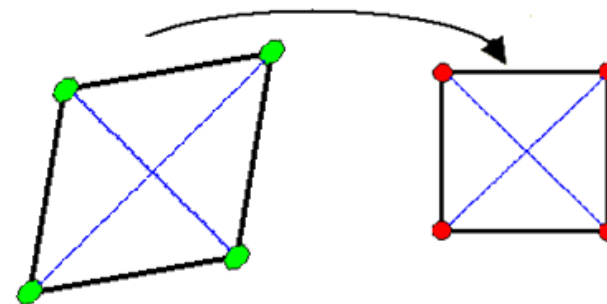
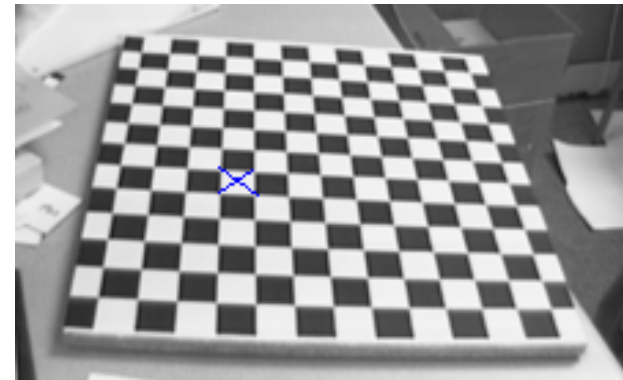
Indicando con $(\tilde{u}_i, \tilde{v}_i)$ le coordinate immagine normalizzate:

$$\lambda \begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \\ 1 \end{bmatrix} = \begin{bmatrix} r_0 & r_1 & T_X \\ r_3 & r_4 & T_Y \\ r_6 & r_7 & T_Z \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} = K \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix}$$

Calibrazione a forza bruta

Banalmente la calibrazione può essere fatta creando relazione tra due matrici di punti (scena ed immagine) ed andando ad interpolare nel caso il punto cada tra due coordinate di riferimento. Tenendo come indice la riga, colonna del CDD si ha una tabella $\mathbf{T}(\text{CCD}_{\text{ROW}}, \text{CCD}_{\text{COL}})$ indicante il punto della scena corrispondente ad un dato punto del CCD.

- › E' il metodo più semplice;
- › Non fa alcun uso di modelli;
- › Molto efficace;
- › Molto sensibile ai disturbi;
- › Può richiedere grosse quantità di memoria;



I sistemi determinati da **K** ed **H** sono lineari in 9 incognite (8 imponendo un elemento costante) più λ . Se si rimuove la dipendenza da λ si ha un sistema che fornisce con 5 punti una prima stima delle 9 incognite:

$$\mathbf{L} = \begin{bmatrix} X_i & Y_i & 0 & 0 & -\tilde{u}_i X_i & -\tilde{u}_i Y_i & 1 & 0 & -\tilde{u}_i \\ 0 & 0 & X_i & Y_i & -\tilde{v}_i X_i & -\tilde{v}_i Y_i & 0 & 1 & -\tilde{v}_i \end{bmatrix}$$
$$\mathbf{a} = (r_0, r_1, r_3, r_4, r_6, r_7, T_X, T_Y, T_Z)^T$$

Si hanno $2 \times N$ equazioni per N punti di controllo per risolvere:

$$\mathbf{L}\mathbf{a} = \mathbf{0}$$

Sotto il vincolo $h_8=1$ si può fare qualcosa di simile per \mathbf{H} :

$$\begin{bmatrix} X_i & Y_i & 1 & 0 & 0 & 0 & -X_i v_i & -Y_i v_i \\ 0 & 0 & 0 & X_i & Y_i & 1 & -X_i v_i & -Y_i v_i \end{bmatrix} \begin{pmatrix} h_0 \\ \vdots \\ h_7 \end{pmatrix} = \begin{pmatrix} u_i \\ v_i \end{pmatrix}$$

Ogni punto da 2 vicoli dunque si possono usare 4 punti.

Una soluzione può essere trovata con l'uso della pseudoinversa.

E' basato sulla proiezione prospettica della pinhole camera e nella sua forma classica è composto da 11 parametri:

1. Lunghezza focale della telecamera (f);
2. Coefficiente di distorsione radiale del primo ordine (k);
3. Coordinate del centro ottico della lente (x_C, y_C);
4. Fattore di scala orizzontale (S_x);
5. Angoli di rotazione tra coordinate scena e coordinate camera ($R_x R_y R_z$)^T;
6. Vettore di traslazione tra coordinate scena e coordinate camera ($T_x T_y T_z$)^T;

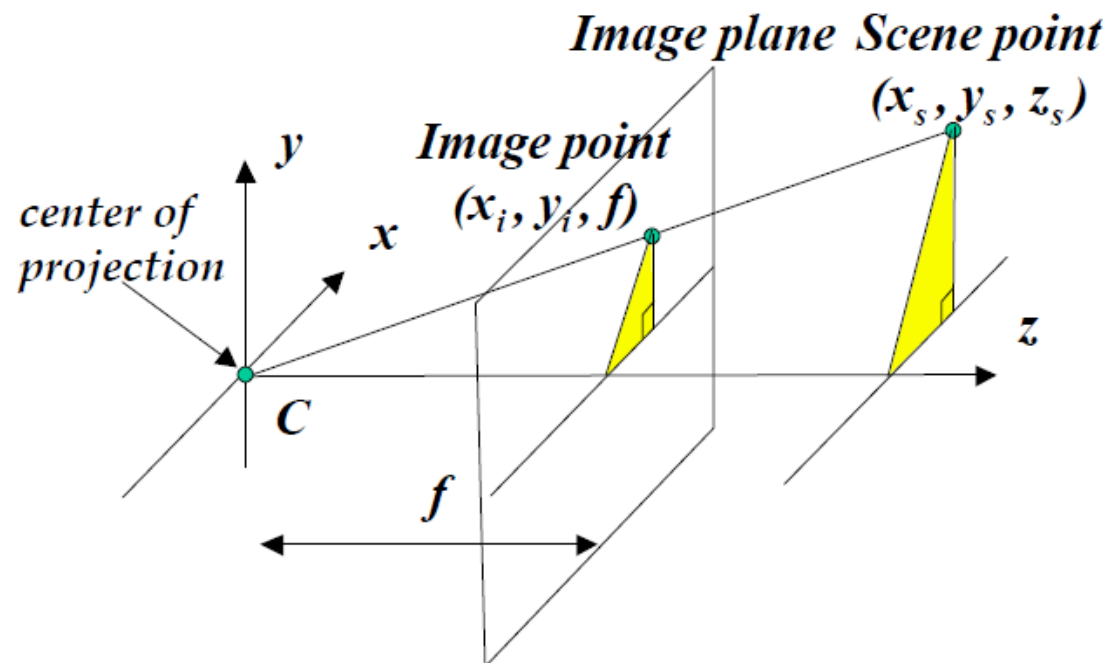
Si riconoscono facilmente i parametri intrinseci ed estrinseci.

Algoritmo di Tsai Parametri intrinseci

Indicando con z l'asse ottico, si identifica un sistema di coordinate con origine nel centro di proiezione e assi x ed y paralleli agli stessi assi dell'immagine.

Usando le relazioni della proiezione prospettica si ha :

$$\frac{x_i - x_0}{f} = \frac{x_C}{z_C} \quad \frac{y_i - y_0}{f} = \frac{y_C}{z_C}$$



Le relazioni tra il sistema di coordinate associate alla scena e quelle della telecamera è data da una matrice di rototraslazione:

$$\begin{pmatrix} x_C \\ y_C \\ z_C \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} x_S \\ y_S \\ z_S \end{pmatrix} + \begin{pmatrix} t_X \\ t_Y \\ t_Z \end{pmatrix}$$

Si hanno 3 angoli + 3 traslazioni più 3 parametri intrinseci = 9 parametri (finora)

Vediamo i due parametri mancanti...

Molte telecamere hanno un filtro passa basso che rende smooth la transizione tra le celle. L'immagine inoltre dopo tale filtraggio viene digitalizzata dal frame grabber. Il campionamento di quest'ultimo tipicamente NON è uguale alla quello spaziale delle celle del sensore.

Al contrario nella direzione verticale il campionamento è controllato della spaziatura delle righe delle celle del sensore.

In questi casi, il rapporto tra dimensioni verticali e orizzontali non è calcolabile a priori basandosi sulle dimensioni delle celle. Dunque si introduce un fattore di scala orizzontale:

$$\frac{x_i - x_0}{f} = S_x \frac{x_C}{z_C}$$

$$\frac{x_i - x_0}{f} = S_x \frac{r_{11}x_S + r_{12}y_S + r_{13}z_S + t_X}{r_{31}x_S + r_{32}y_S + r_{33}z_S + t_Z} \quad \frac{y_i - y_0}{f} = \frac{r_{21}x_S + r_{22}y_S + r_{23}z_S + t_Y}{r_{31}x_S + r_{32}y_S + r_{33}z_S + t_Z}$$

Come abbiamo visto esistono molte distorsioni nelle lenti. Se le superfici sono sferiche e centrate sull'asse ottico le distorsioni sono di tipo radiale (pin-cushion o barrel).

Dato il punto (x_U, y_U) ripreso da una lente ideale viene trasformato nel punto (x_D, y_D) distorto da una funzione dipendente dalla distanza r_d dal centro di distorsione (di solito diverso dal centro prospettico o dal centro geometrico dell'immagine):

$$\frac{r_U}{r_d} = f_d(r_d) = 1 + \kappa_1 r^2 + \kappa_2 r^4 + \dots$$

La funzione f_d è ovviamente pari e da qui le potenze di multiple di 2.

Le distorsioni tangenziali sono di solito trascurabili.

La strategia totale è la seguente:

1. Si stima con un metodo di fitting least-square più parametri possibile (si usa la pseudo-inversa). Durante questo primo passo non si minimizza l'errore nel piano immagine ma un'altra quantità;
2. Si procede con un'ottimizzazione non lineare che cerca il miglior fit tra i punti osservati e quelli predetti con modello.

Algoritmo di Tsai

La conversione

<http://www.cs.cmu.edu/~rgw/TsaiCode.html>

```
void ImageCoord2WorldCoord(double Xfd, double Yfd, double zw, CALIBRATION_CONSTANTS *cc,
                          CAMERA_PARAMETERS *cp, double *xw, double *yw)
{
    // Converte dall'immagine alle coordinate distorte del ccd
    double Xd = cp->dpx * (Xfd - cp->Cx) / cp->sx;
    double Yd = cp->dpy * (Yfd - cp->Cy);

    double Xu, Yu;

    // Converte delle coordinate distorte del ccd alle coordinate non distorte del ccd
    Distorte2NonDistorteCcd(Xd, Yd, &ccTabelle, &Xu, &Yu);

    // Calcola i corrispondenti xw e yw
    double common_denominator = ((cc->r1 * cc->r8 - cc->r2 * cc->r7) * Yu +
                                   (cc->r5 * cc->r7 - cc->r4 * cc->r8) * Xu -
                                   cc->f * cc->r1 * cc->r5 + cc->f * cc->r2 * cc->r4);

    *xw = (((cc->r2 * cc->r9 - cc->r3 * cc->r8) * Yu +
            (cc->r6 * cc->r8 - cc->r5 * cc->r9) * Xu -
            cc->f * cc->r2 * cc->r6 + cc->f * cc->r3 * cc->r5) * zw +
            (cc->r2 * cc->Tz - cc->r8 * cc->Tx) * Yu +
            (cc->r8 * cc->Ty - cc->r5 * cc->Tz) * Xu -
            cc->f * cc->r2 * cc->Ty + cc->f * cc->r5 * cc->Tx) / common_denominator;

    *yw = -(((cc->r1 * cc->r9 - cc->r3 * cc->r7) * Yu +
            (cc->r6 * cc->r7 - cc->r4 * cc->r9) * Xu -
            cc->f * cc->r1 * cc->r6 + cc->f * cc->r3 * cc->r4) * zw +
            (cc->r1 * cc->Tz - cc->r7 * cc->Tx) * Yu +
            (cc->r7 * cc->Ty - cc->r4 * cc->Tz) * Xu -
            cc->f * cc->r1 * cc->Ty + cc->f * cc->r4 * cc->Tx) / common_denominator;
}

void Distorte2NonDistorteCcd(double Xd, double Yd, CALIBRATION_CONSTANTS *cc, double *Xu, double *Yu)
{
    double distortion_factor;

    distortion_factor = 1 + cc->kappa1 * (Xd * Xd + Yd * Yd);
    *Xu = Xd * distortion_factor;
    *Yu = Yd * distortion_factor;
}
```

```
struct CALIBRATION_CONSTANTS
{
    double f; // [mm]
    double kappa1; // [1/mm^2]
    double p1; // [1/mm]
    double p2; // [1/mm]
    double Tx; // [mm]
    double Ty; // [mm]
    double Tz; // [mm]
    double Rx; // [rad]
    double Ry; // [rad]
    double Rz; // [rad]
    double r1; // []
    double r2; // []
    double r3; // []
    double r4; // []
    double r5; // []
    double r6; // []
    double r7; // []
    double r8; // []
    double r9; // []
};
```

Zhang esegue un aggiornamento delle tecniche di calibrazione di Tsai.

Zhang calcola la matrice \mathbf{H} e da questa cerca di ricavare i parametri in maniera esplicita. \mathbf{H} è una matrice omografica e pertanto possiede 8 gradi di libertà. Da questa matrice è possibile porre due vincoli basati sulla ortonormalità della matrice di rotazione forzando almeno 2 dei parametri della matrice dei parametri intrinseci.

$$H = \begin{bmatrix} \tilde{h}_1 & \tilde{h}_2 & \tilde{h}_3 \end{bmatrix} = \lambda A \begin{bmatrix} \tilde{r}_1 & \tilde{r}_2 & \tilde{t} \end{bmatrix}$$

Esprimendo l'ortonormalità tra i vettori colonna:

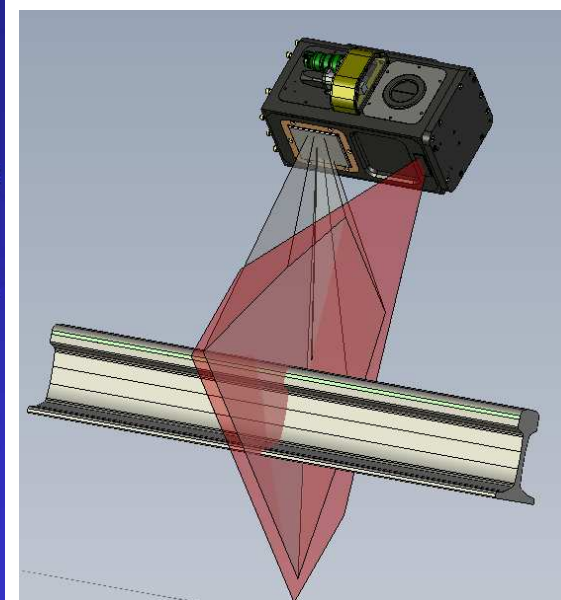
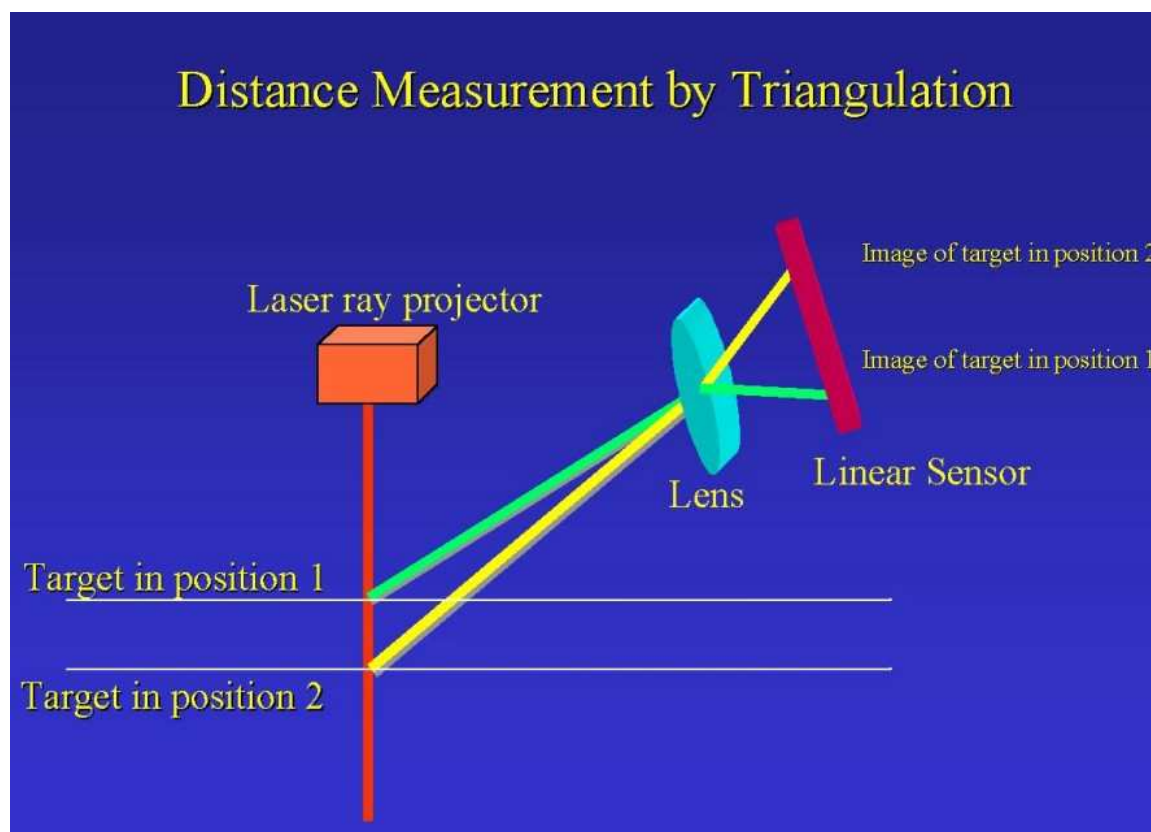
$$\tilde{h}_1^T W \tilde{h}_2 = 0 \quad \tilde{h}_1^T W \tilde{h}_1 = \tilde{h}_2^T W \tilde{h}_2 \quad W = (A^{-1})^T A^{-1}$$

Le incognite della matrice W possono essere risolte usando almeno 2 (o 3) piani diversi, ovvero matrici le cui colonne non siano linearmente dipendenti tra loro. Determinata la matrice con una decomposizione di Choleski si può determinare la matrice originale. Tuttavia Zhang fornisce le equazioni per ottenere i parametri direttamente da W decomponendola.

Cenni di triangolazione laser

Principio 1

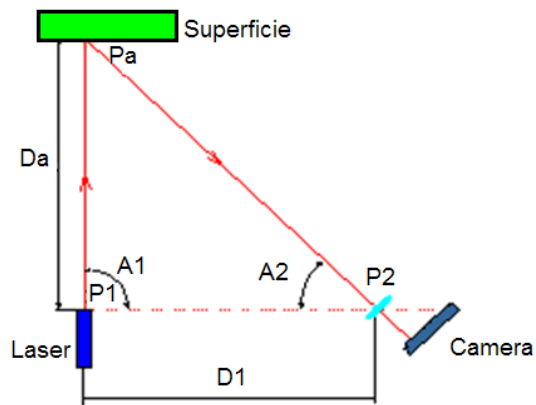
Si lega la posizione del target ad una posizione su un CCD basandosi sulle caratteristiche geometriche che dunque devono restare invariate. Quando queste variano si deve ricalibrare il sistema.



Cenni di triangolazione laser

Principio 2

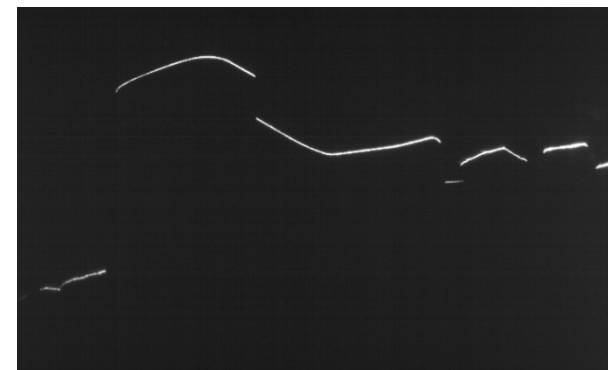
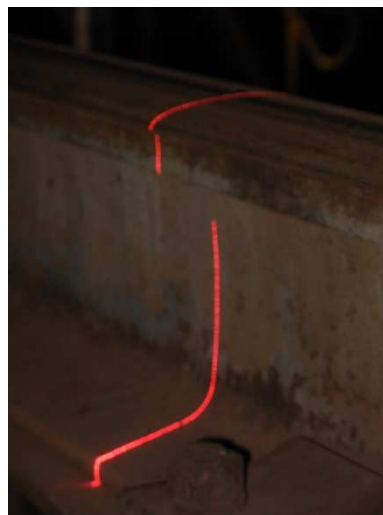
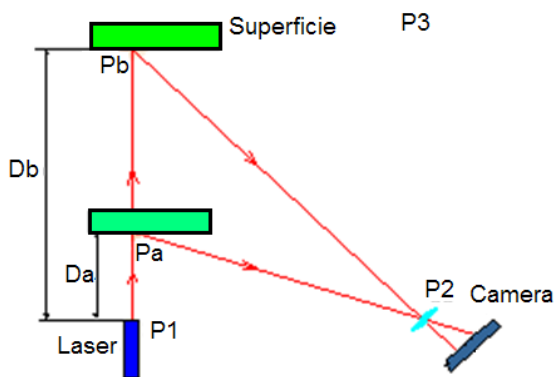
Si lega la posizione del target ad una posizione su un CCD basandosi sulle caratteristiche geometriche. Il laser fornisce il piano su cui si effettueranno le misure.



$D1$ = distanza tra sorgente laser $P1$ e telecamera $P2$ è costante e dettata da considerazioni meccaniche, come $A1$ angolo del fascio laser posto a 90° .
L'inclinazione dell'asse delle telecamera $A2$ è anch'esso calcolato in funzione di considerazioni sulla misura.

In conclusione la distanza D_a tra $P1$ e il punto da misurare P_a è:

$$D_a = D1 * \tan(A2)$$



Cenni di triangolazione laser

Diodi laser 1

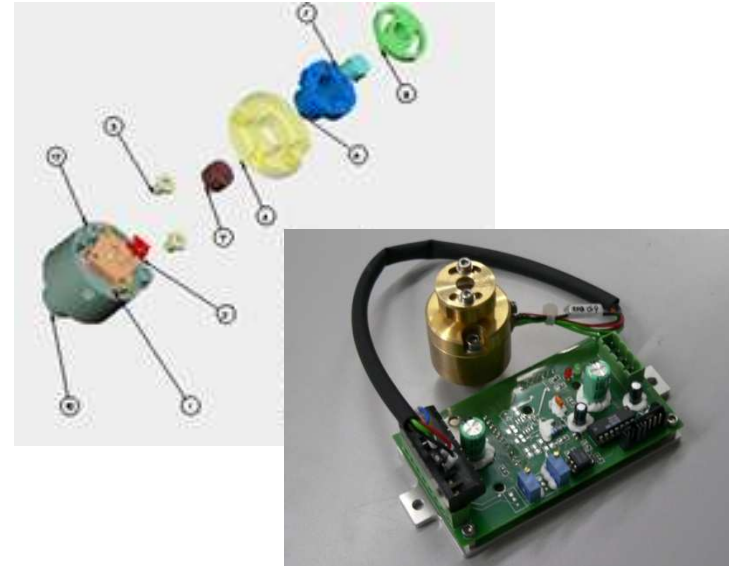
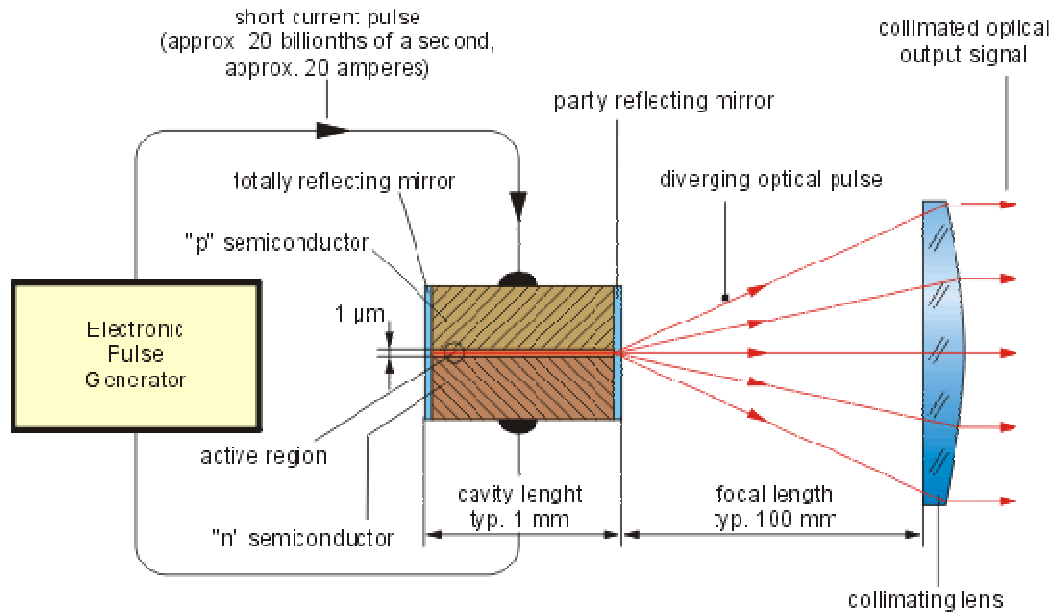
Se su un atomo allo stato eccitato incide un fotone di frequenza opportuna, l'atomo si diseccita cedendo la sua energia sotto forma di fotone avente la stessa frequenza e la stessa fase di quello incidente.

Se si provoca la cosiddetta inversione della popolazione, cioè si fa in modo che gli atomi allo stato eccitato siano più numerosi di quelli allo stato fondamentale, si ha prevalenza dell'emissione stimolata sull'assorbimento. Il processo con cui si attua tale inversione prende il nome di pompaggio.

In un semiconduttore non si può considerare un atomo isolato, ma bisogna considerare tutto il cristallo nel suo insieme, con una certa distribuzione degli elettroni, che si dispongono in "bande" di energia. Operando una semplificazione si può dire che in un semiconduttore avremo una "banda di valenza" che risulterà "piena" di elettroni, ed una "banda di conduzione", ad energia più elevata, a distanza E_g dalla banda di valenza, che conterrà pochi elettroni. Con diversi metodi è possibile ottenere all'interno del cristallo una vera e propria inversione di popolazione. Se non si inserisce il sistema in un risonatore (che può essere costituito dallo stesso cristallo di semiconduttore) si ottiene un LED altrimenti avremo un laser a semiconduttore. Esistono molti di questi laser, che emettono potenze medie di 10 mW in continua e raggiungono i 100 W in regime impulsato.

Cenni di triangolazione laser

Diodi laser 2



LUCE NORMALE

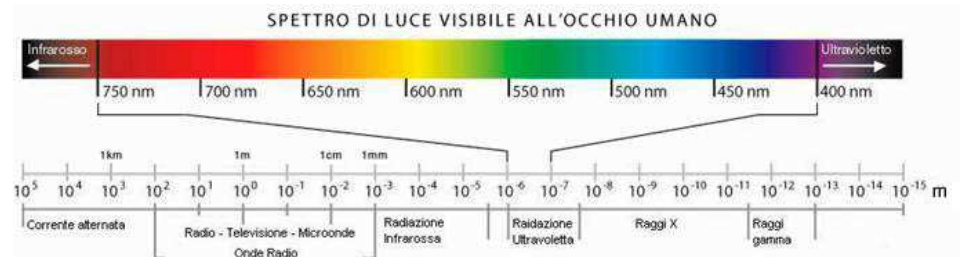


BASSA DIREZIONALITA'
BASSA MONOCROMATICITA'
BASSA COERENZA
BASSA POTENZA

LUCE LASER



ALTA DIREZIONALITA'
ALTA MONOCROMATICITA'
ALTA COERENZA
ALTA POTENZA



CLASSE 1M

Laser che emettono radiazione nell'intervallo di lunghezze d'onda tra 302,5 nm e 4000 nm, sicuri nelle condizioni di funzionamento ragionevolmente prevedibili, ma che possono essere pericolosi se l'utilizzatore impiega ottiche (lenti di ingrandimento, etc.) all'interno del fascio.

CLASSE 2

I laser in questa classe possono emettere radiazione pericolosa, ma la loro potenza è così bassa da risultare in qualche modo dannosa solo in caso di esposizione diretta e prolungata ovvero per un tempo superiore ai 0,25 secondi. Sono compresi in questa classe i laser ad emissione continua e nel visibile (400-700nm), con potenza $\leq 1\text{mW}$.

CLASSE 2M

Laser che emettono radiazione visibile nell'intervallo di lunghezze d'onda tra 400nm e 700nm, in cui la protezione dell'occhio è normalmente assicurata dalle reazioni di difesa compreso il riflesso palpebrale. Tuttavia l'osservazione dell'emissione può risultare pericolosa se, all'interno del fascio, l'utilizzatore impiega ottiche (lenti di ingrandimento, etc.)

CLASSE 3R (CLASSE 3A)

Sono compresi in questa classe i laser con emissione nel visibile e una potenza in uscita fino da 1mW a 5mW.

Possono emettere radiazioni sia nel campo del visibile che in quello del non visibile e i loro fasci non sono pericolosi se osservati direttamente in maniera non continua, mentre lo possono diventare se si utilizzano strumenti che amplificano e concentrano il fascio ottico (quali microscopi, binocoli, ecc.).

CLASSE 3B

I laser di classe 3B hanno potenze medie comprese tra i 5mW e i 500 mW. I laser di classe 3B, sia per radiazione visibile che per quella non visibile, sono pericolosi per gli occhi se non protetti e possono essere pericolosi per la pelle; anche le riflessioni diffuse da questi sistemi possono essere pericolosi. Devono essere prese precauzioni per evitare lo stazionamento nella direzione del fascio o del fascio riflesso da una superficie.

CLASSE 4

Sono i laser più pericolosi in quanto, oltre ad avere una potenza tale da causare seri danni ad occhi e pelle anche se il fascio è diffuso, possono costituire un potenziale rischio di incendio, causare fuoruscita di materiale tossico e spesso il voltaggio e l'ampereaggio di alimentazione sono pericolosamente elevati.

Comprende tutti quei sistemi che superano i livelli imposti alla classe 3B.

Cenni di triangolazione laser

Sicurezza

Un fascio di luce laser sia diretto, che riflesso da superfici speculari può causare danni anche irreversibili alle strutture oculari e alla pelle; la natura di questi danni dipende dalla lunghezza d'onda della radiazione, mentre la gravità è legata alla densità di potenza E (per sorgenti in funzionamento continuo) o alla densità di energia H (per fasci in funzionamento impulsato) e al tempo in cui la struttura oculare è esposta al fascio laser.

Confronto tra il sole e un laser:

SOLE: Intensità massima luce solare a terra = 1 kW/m^2 o 1 mW/mm^2

Assumendo un diametro pupillare di 2 mm l'area è circa 3 mm^2

Quindi la potenza raccolta dall'occhio è = 3 mW

Il sole forma un'immagine $\approx 100 \mu\text{m}$ di raggio sulla retina (area = 0.03 mm^2)

L'intensità sulla retina (Potenza/Area) = $3 \text{ mW}/0.03 \text{ mm}^2 = 100 \text{ mW/mm}^2$.

Tipico laser He Ne da 1 mW (o laser pointer):

Potenza (P) = 1 mW (Classe 3R), raggio del fascio = 1 mm

Forma un'immagine con raggio di $10 \mu\text{m}$ (area dello spot = $3 \cdot 10^{-4} \text{ mm}^2$)

L' **intensità** dell'He-Ne sulla retina è $1 \text{ mW}/(3 \cdot 10^{-4} \text{ mm}^2) = 3100 \text{ mW/mm}^2$

31 volte l'intensità del sole!!

Visione dell'occhio

CONSIGNES D'UTILISATION DES LUNETTES

- Utiliser exclusivement des lunettes :
 - conformes aux exigences des normes EN 207 (utilisation) ou EN 208 (maintenance),
 - adaptées au laser mis en œuvre,
 - en bon état.
- Lire la notice d'utilisation fournie par le fabricant.
- Ne jamais regarder directement le faisceau laser direct ou une de ses réflexions sans un protecteur oculaire.
- Nettoyer régulièrement les verres et les filtres.
- Remettre, après utilisation, les lunettes de protection dans leurs étuis.
- Ranger les étuis hors de la zone laser.
- Éliminer toute paire de lunettes défectueuse (filtres rayés ou endommagés).
- Prévoir des lunettes supplémentaires pour les visiteurs.

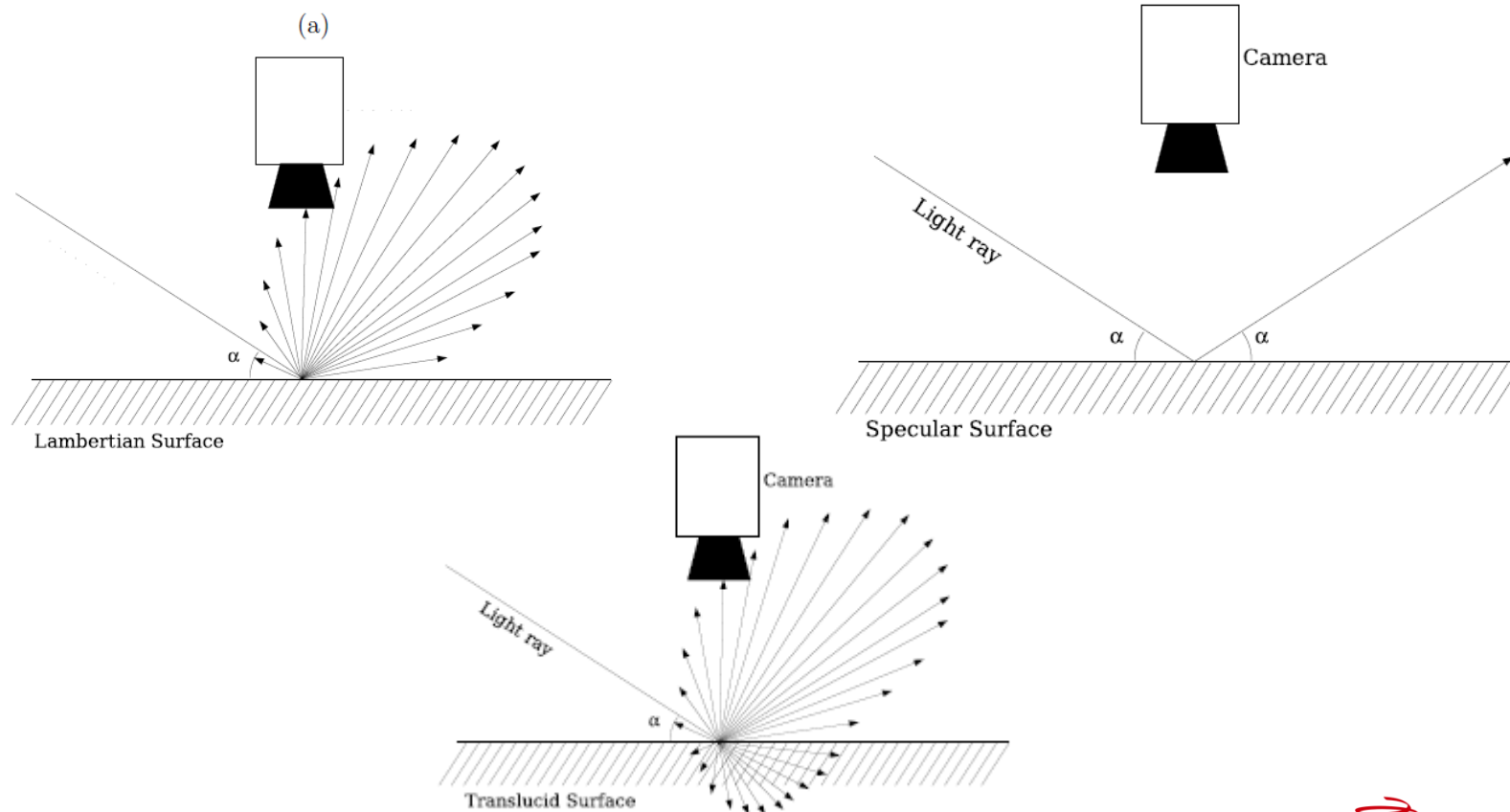


Cenni di triangolazione laser

Tipi di Superficie

Il sistema a triangolazione laser fa uso della luce rifratta (non riflessa) da una superficie lambertiana.

Notare che parte della potenza del laser viene persa nel raggio riflesso. Il caso peggiore si ha nel caso di superfici lucide come gli specchi o superfici levigate.



Trasformata di Hough (HT)

Cenni

Il problema del riconoscimento di oggetti in un'immagine è spesso risolto usando algoritmi di pattern matching o simili minimizzando un indice di errore

Tali algoritmi sono spesso molto onerosi e soffrono in presenza di immagini rumorose o di sovrapposizione tra oggetti.

Permette di individuare forme descritte da equazioni analitiche (es. Rette). Essa trasforma un problema di ricerca di una curva in un più semplice problema di ricerca di massimi.

La HT gode delle seguenti proprietà:

- › Ogni punto dello spazio immagine corrisponde ad una superficie generalizzata nello spazio dei parametri;
- › N punti nello spazio immagine appartenenti alla stessa curva generano N superfici che si intersecano nello stesso punto dello spazio dei parametri;

Esistono poi le proprietà duali.

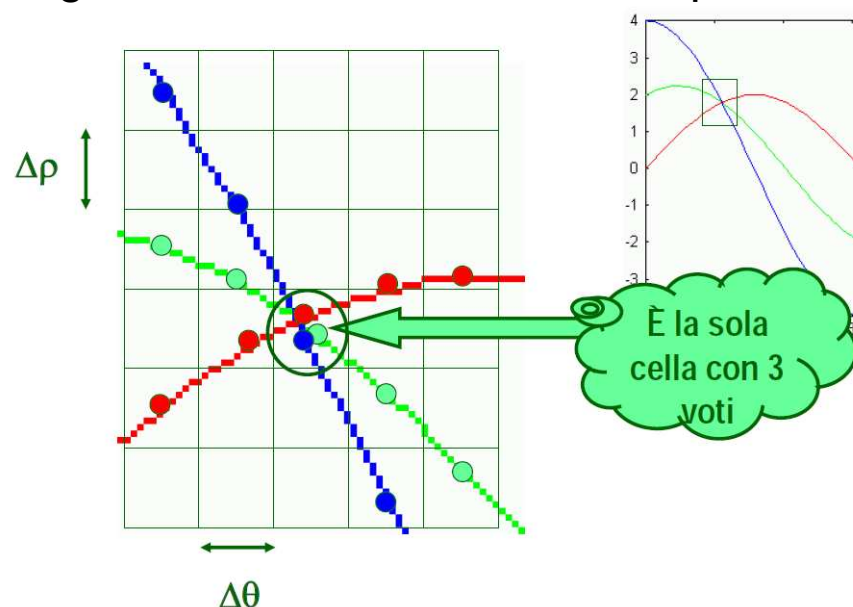
Trasformata di Hough

Celle e voto

Si discretizza lo spazio dei parametri in celle di dimensione dipendente dalle precisioni richieste. Ad ogni cella corrisponde un'istanza quantizzata della curva. Definiamo dunque un accumulatore:

$$Acc(\theta, \rho)$$

Un processo di voto viene poi implementato cosicché ad ogni cella corrispondano tanti voti quante le superfici che la intersecano, ognuna delle quali generata da una curva dello spazio immagine.



Un processo di ricerca del massimo andrà poi a indentificare la curva con più voti, dunque molte intersezioni indicano la presenza della curva analitica cercata.

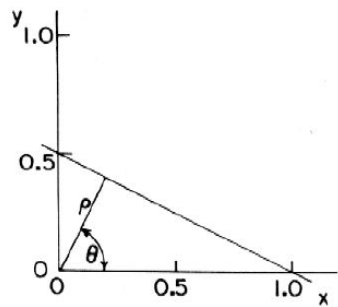
Trasformata di Hough (HT)

Esempio - rette 1

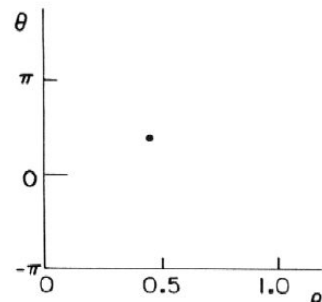
Per prima cosa si passa a coordinate polari perchè hanno variazione limitata:

$$\rho = x \cos \theta + y \sin \theta$$

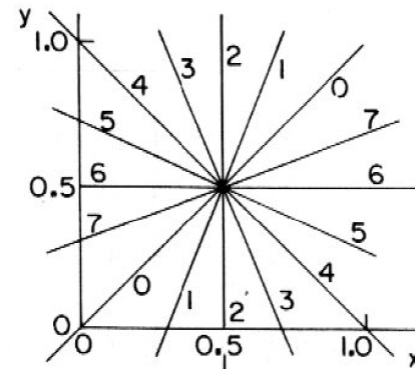
La HT di una linea è un punto. Una stella di rette diviene un insieme di punti connessi nello spazio di Hough:



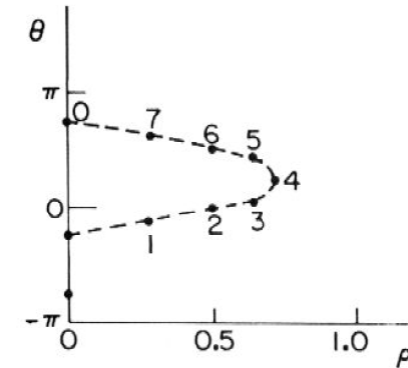
(a) Parametric line



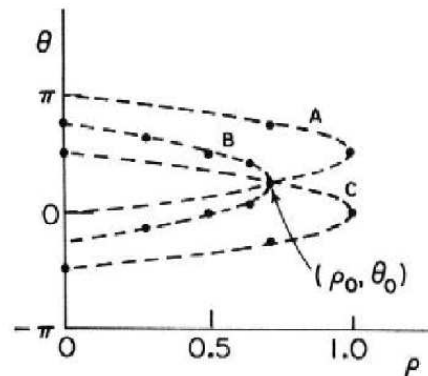
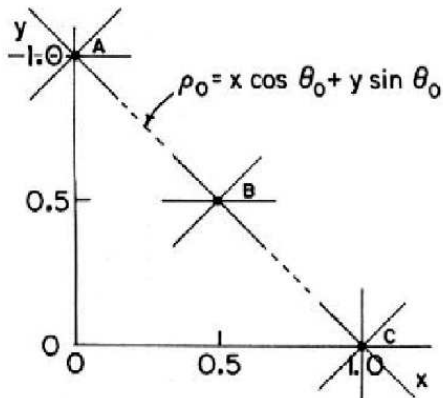
(b) Hough transform of (a)



(c) Family of lines, common point



(d) Hough transform of (c)



Considerando un punto come centro di una stella. L'intersezione dei punti corrispondenti (voto maggiore) nello spazio di Hough indica la retta i cui parametri sono identificati dalla cella.

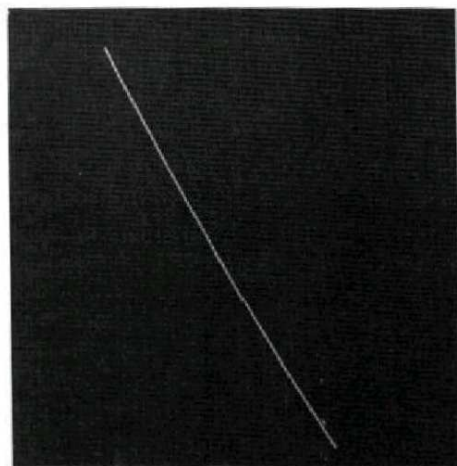
Trasformata di Hough (HT)

Esempio - rette 2

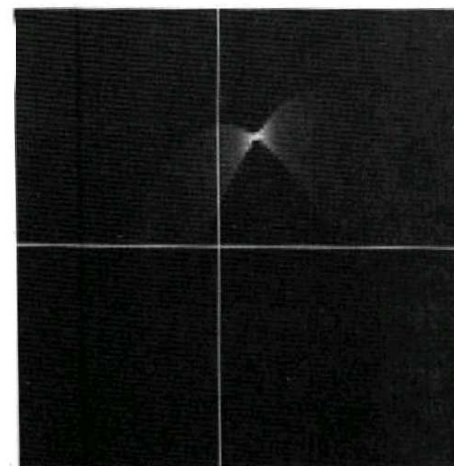
Andando a fare un esempio pratico si vede bene come trasformando tutti i punti dell'immagine una regione ristretta nello spazio di Hough (cella) è interessata da molte intersezioni cioè da molti voti.

Dalla cella cui è possibile ricavare le caratteristiche (pendenza ed intercetta) della retta che unisce i vari punti che l'hanno votata.

In altre parole vi sono N stelle di rette generate da N punti dell'immagine che hanno in comune la retta illustrata e questo si evince dal voto alto ad una cella.



Linea retta.



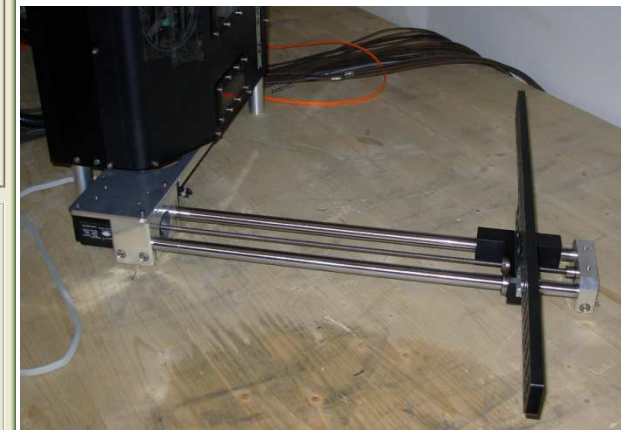
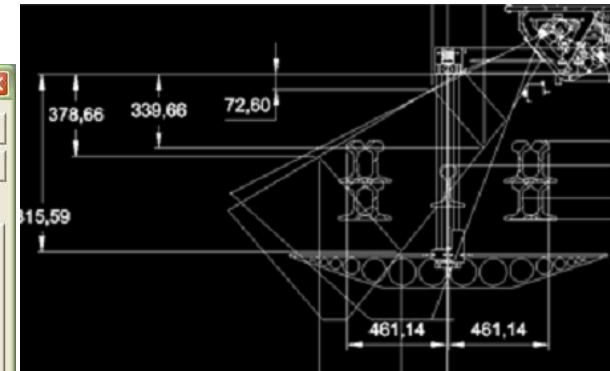
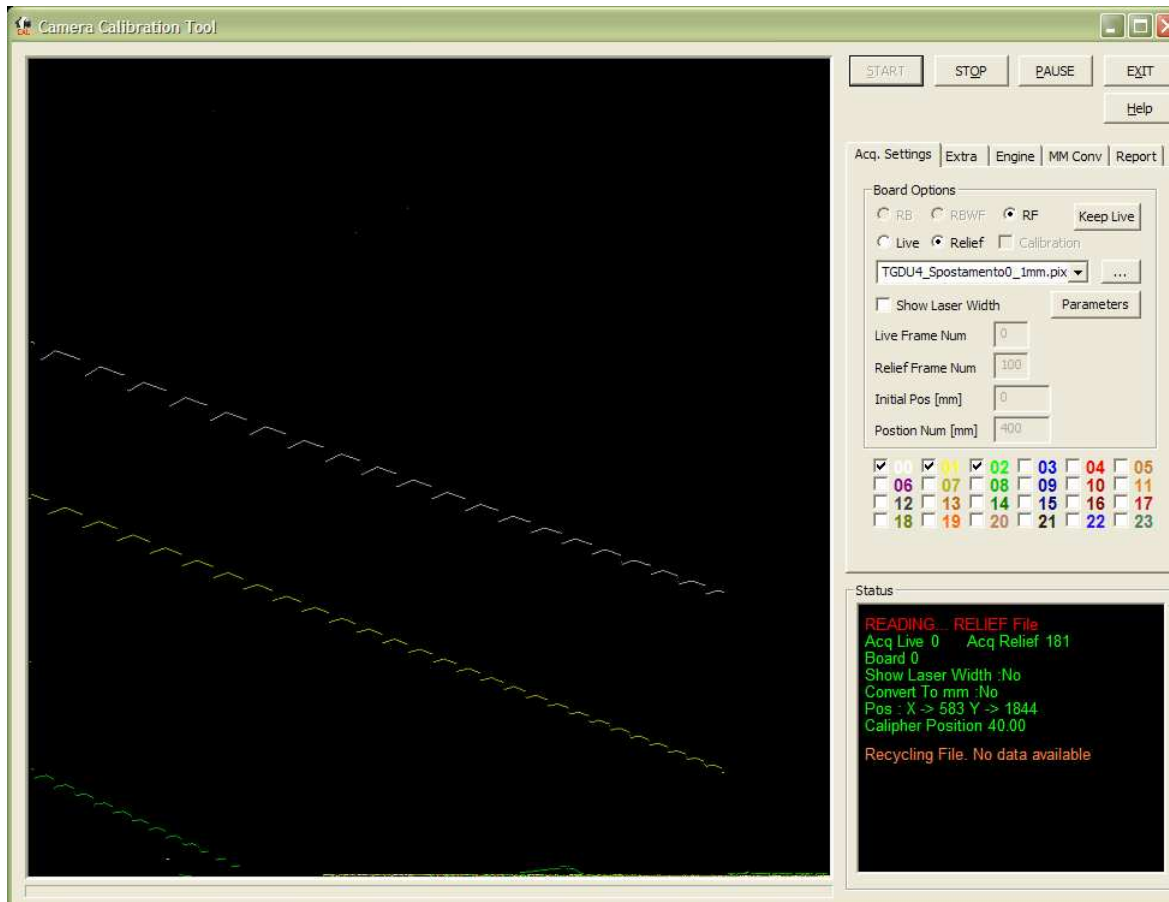
Trasformata della linea.

Processo di calibrazione

Acquisizione del bersaglio in varie posizioni

Acquisizione del bersaglio forato in varie posizioni.

Il bersaglio è movimentato tramite un sistema motorizzato ad alta precisione.
Si possono acquisire più telecamere alla volta:

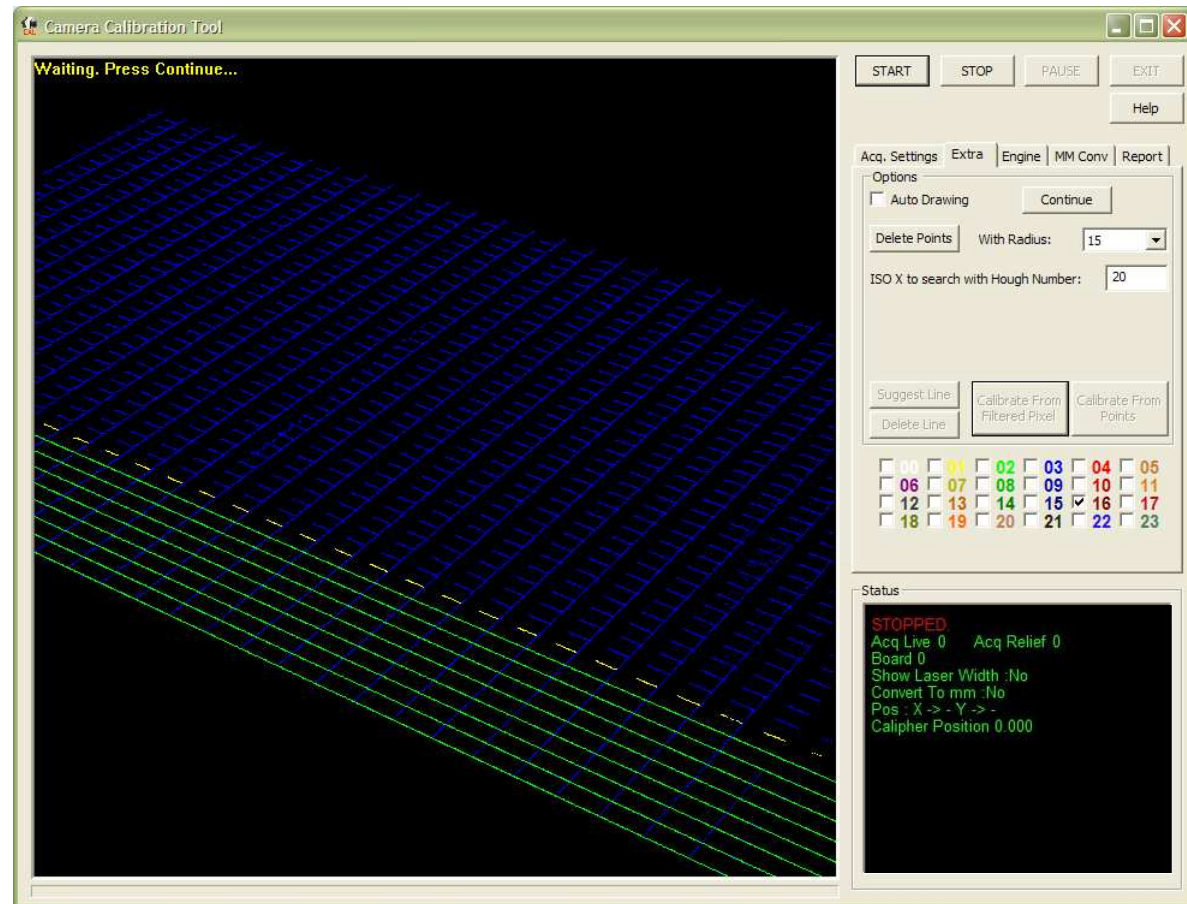


Processo di calibrazione

Identificazione primo set di parabole

Identificazione delle parabole:

1. eliminazione degli spigoli;
2. primo tentativo di fit con una retta;
3. Eliminazione punti troppo ortoganli alla retta trovata;
4. Nuovo fit di una nuova retta (miglioro la stima);
5. Eliminazione punti troppo distanti;
6. fit con una parabola;

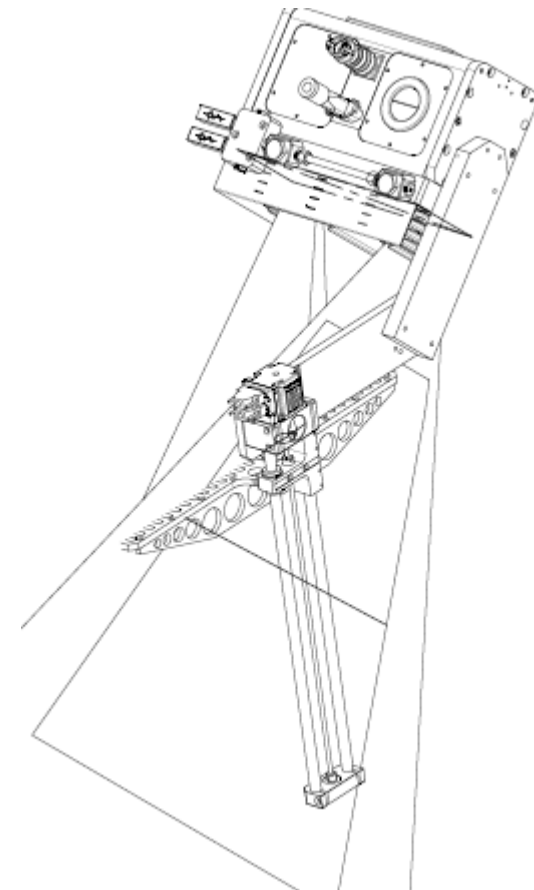
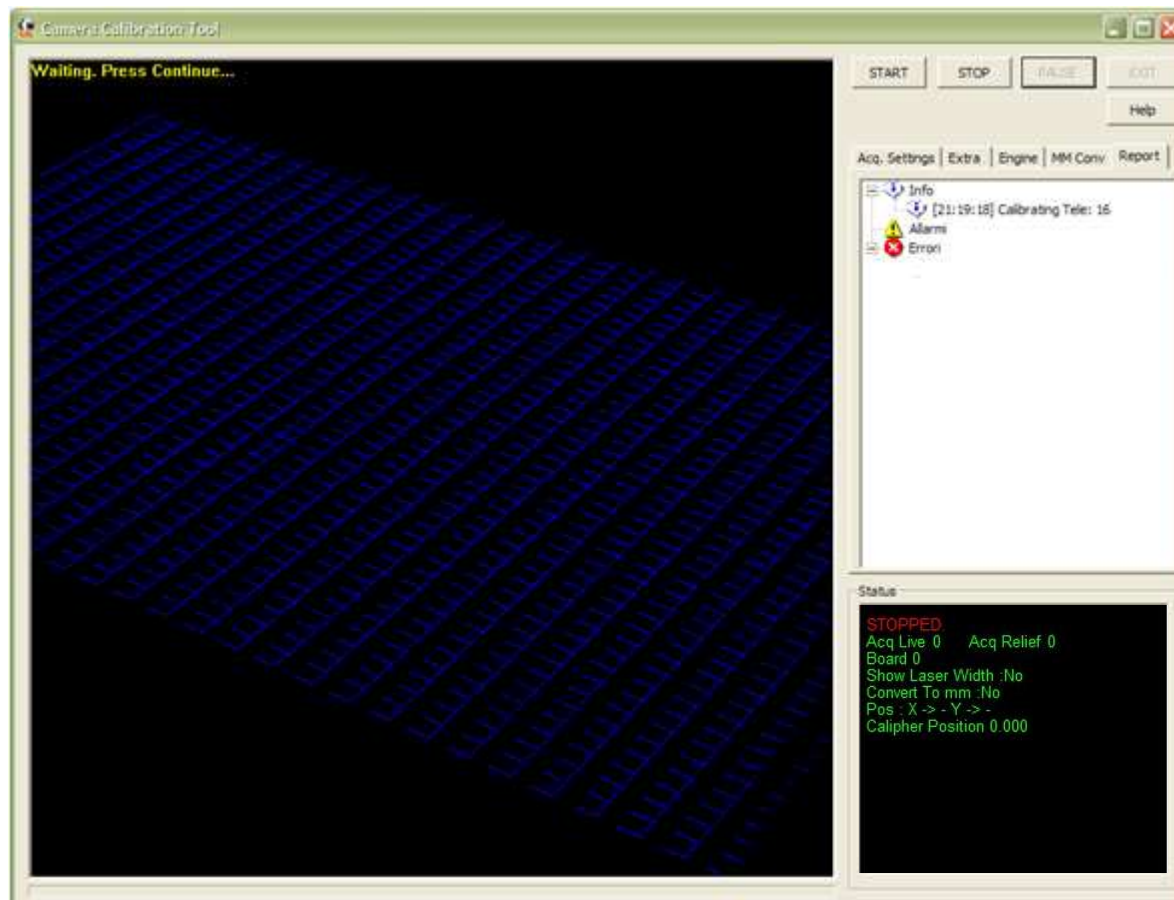


Non si stima una retta perchè vi sono le distorsioni dell'ottica

Processo di calibrazione

Assieme di tutte le acquisizioni del bersaglio

Si ricaricano in memoria tutte le tracce acquisite, una per ogni posizione del bersaglio. Si usano tutti i punti non appartenenti alle parabole già trovate per andare a fare una ricerca con Hough.



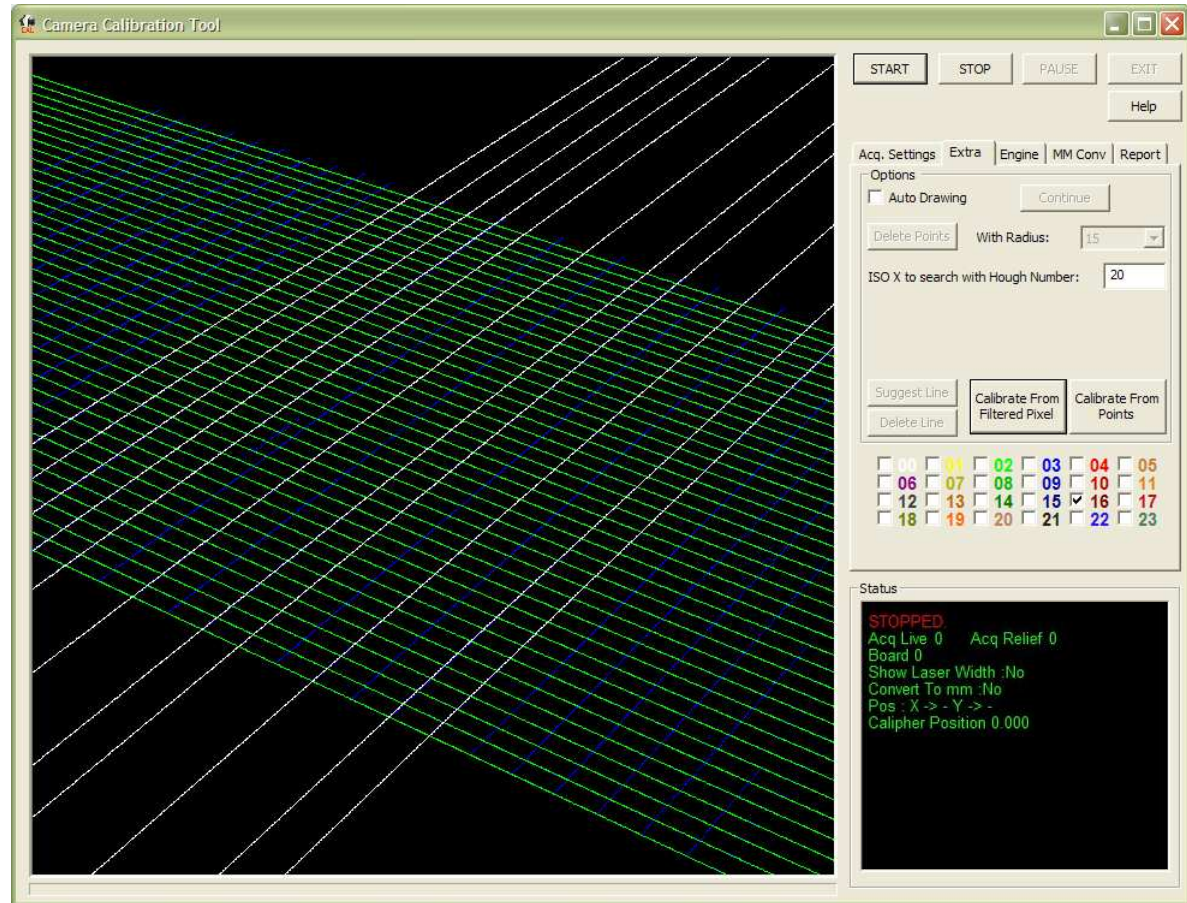
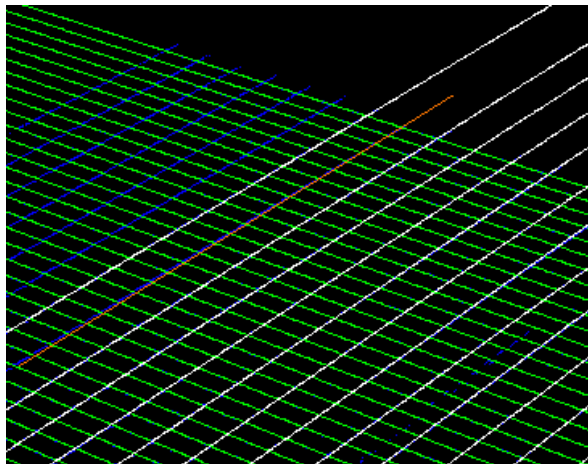
Processo di calibrazione

Identificazione rette mediante trasformata di Hough

Tramite la trasformata di Hough si possono individuare le rette ortogonali alle parabole già trovate imponendo un angolo di 90 gradi (con tolleranza!!!).

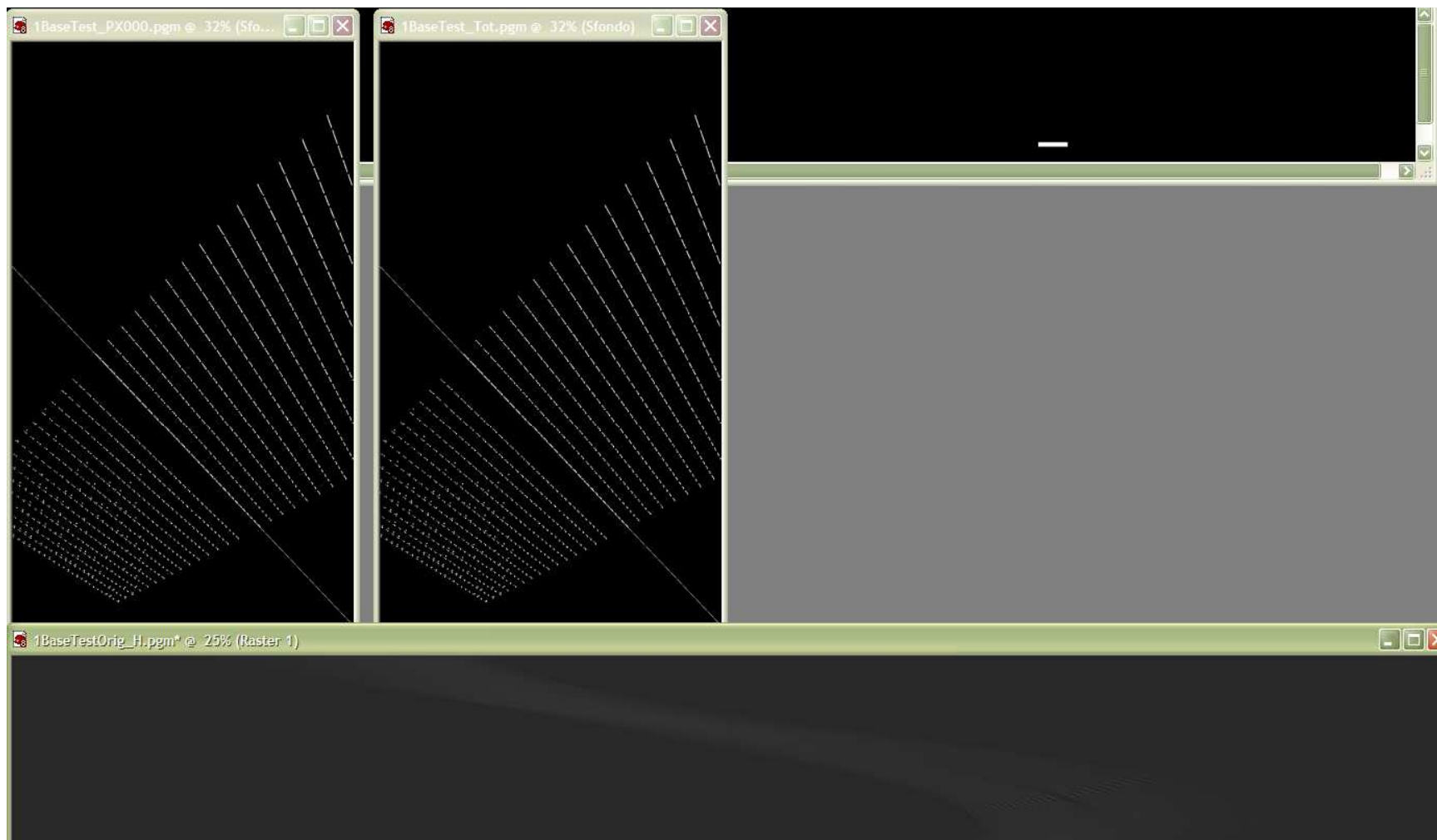
Per far ciò si eliminano i punti già usati considerando la distanza tra punto e parabola e poi si dà tutto in pasto alla HT.

Se non si trovano tutte le rette si possono “suggerire” manualmente.



Processo di calibrazione

Trasformata di Hough

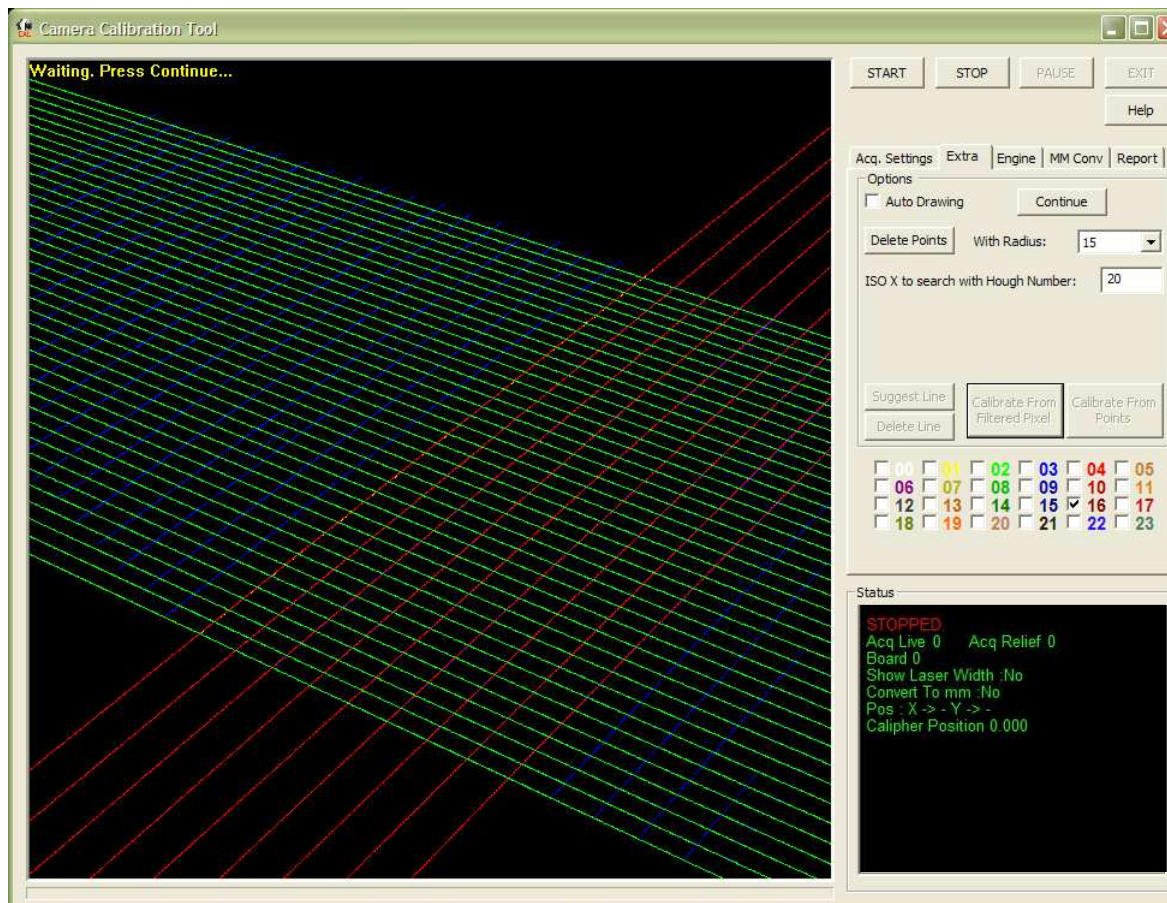


Processo di calibrazione

Identificazione secondo set di parabole

Per prima cosa si ordinano le rette trovate con la HT in base all'intercetta.

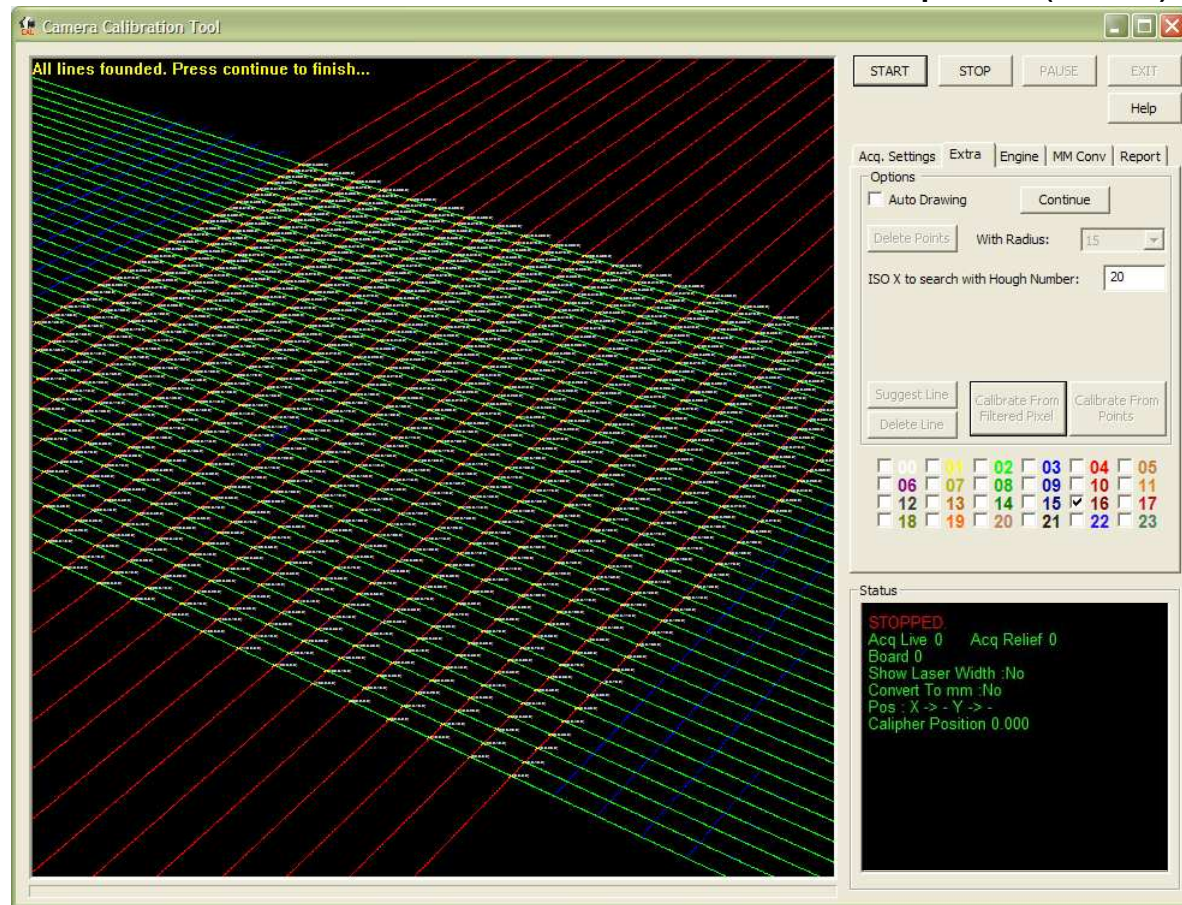
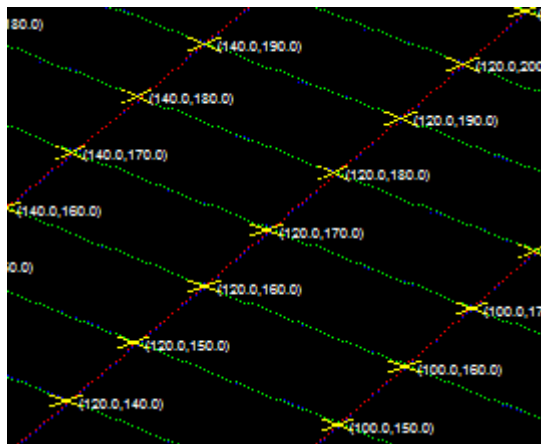
Considerando solo i punti vicini alle rette trovate con la trasformata di Hough si identifica il secondo set di parabole.



Processo di calibrazione

Identificazione degli incroci

A questo punto si hanno due set di parabole tra loro ortogonali. Si procede a determinare tutte le intersezioni che saranno i dati di ingresso all'algoritmo di calibrazione dato che si conoscono le distanze reali in X ed Y tra tali punti (1 cm).



Processo di calibrazione

Algoritmo di Tsai - Verifica

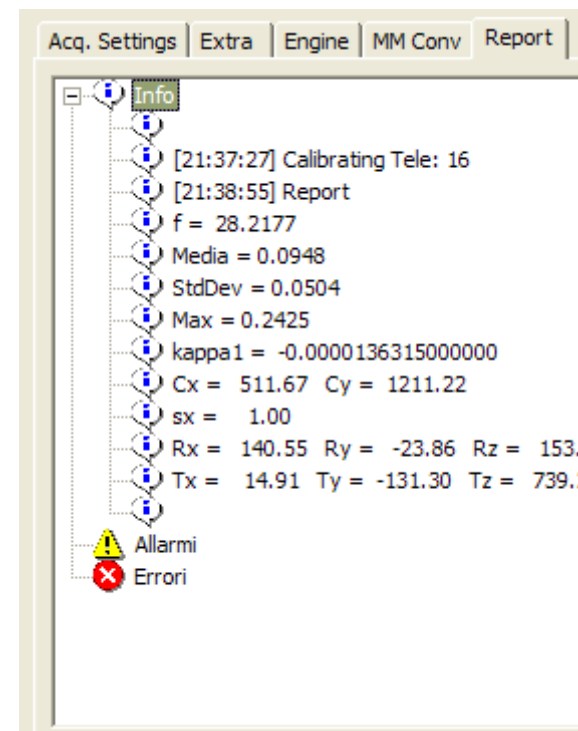
Identificati i punti, si possono dare in ingresso all'algoritmo di calibrazione (qui Tsai) che identifica i parametri del modello.

A questo si possono trasformare i punti di incrocio dei due set di parabole usando la calibrazione appena fatta e confrontare il risultato con reticolo dato in pasto per la calibrazione.

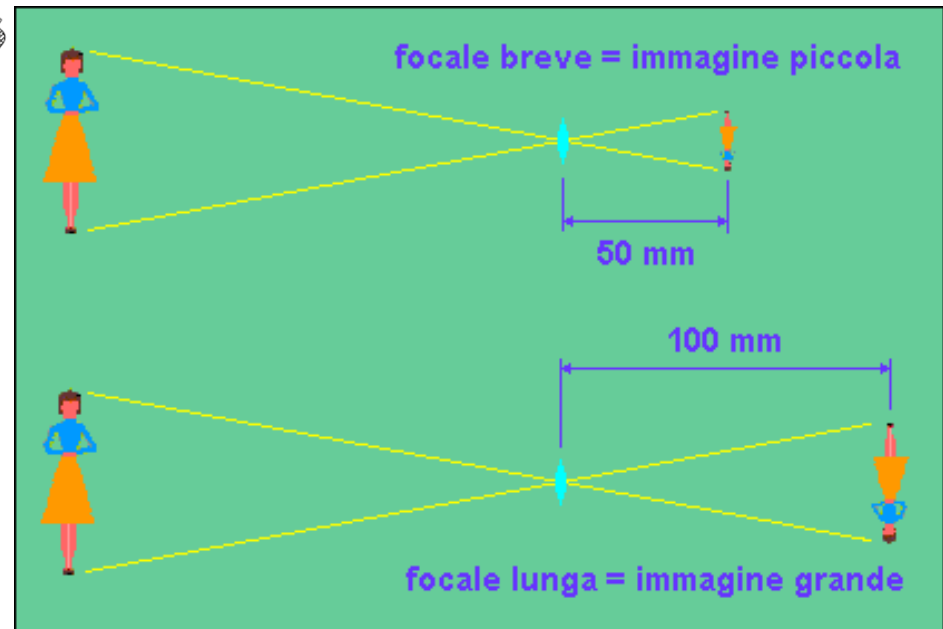
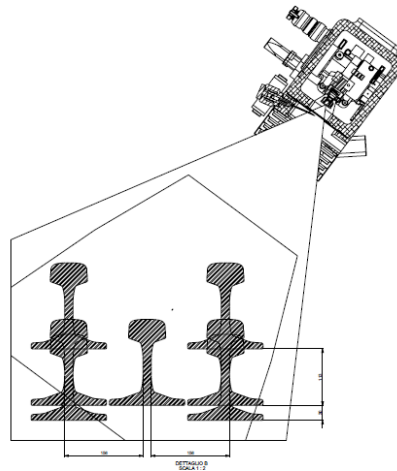
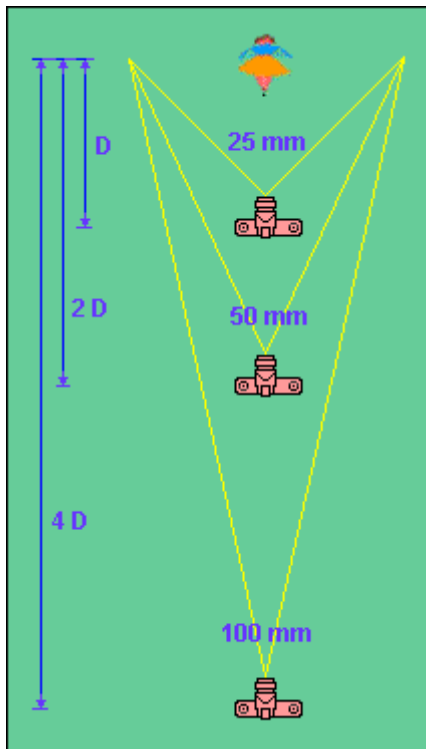
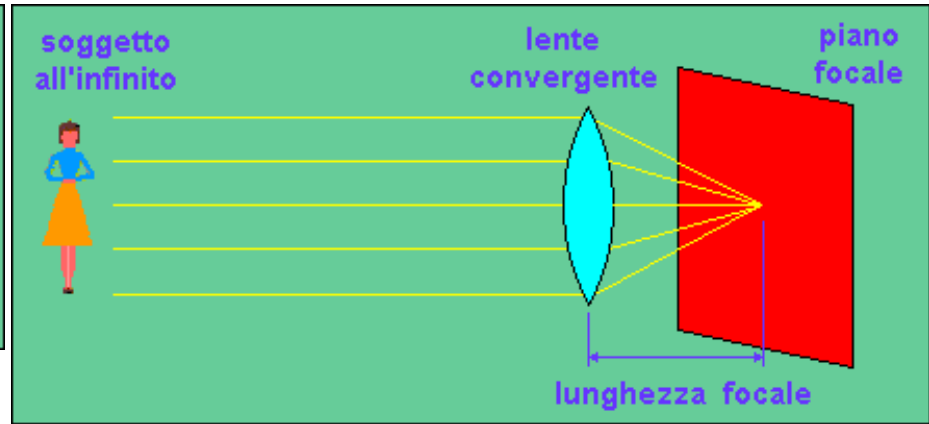
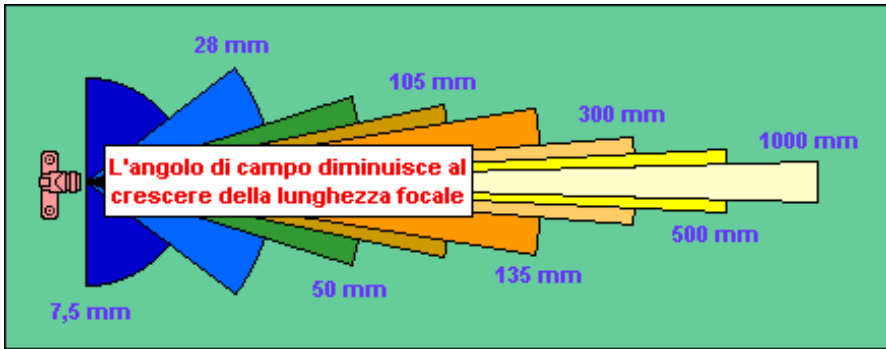
La media dell'errore e la standard deviation sono gli indici della bontà della calibrazione.

La focale è un parametro importante perchè da subito l'idea della bontà della calibrazione.

L'errore massimo dipende dal grado di adeguatezza del modello e spesso riguarda porzioni di campo che poi non verranno utilizzate.

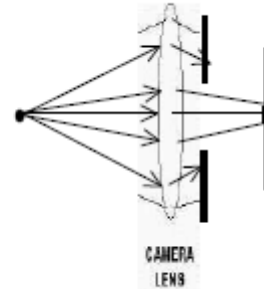
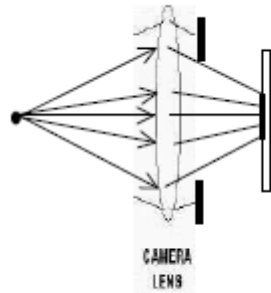


L'obiettivo



Il diaframma

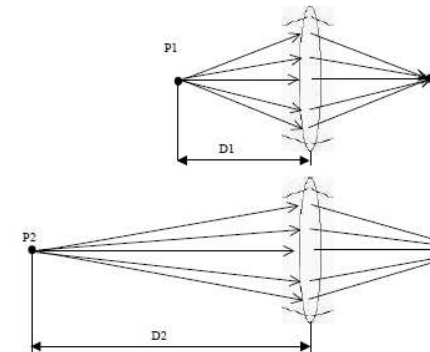
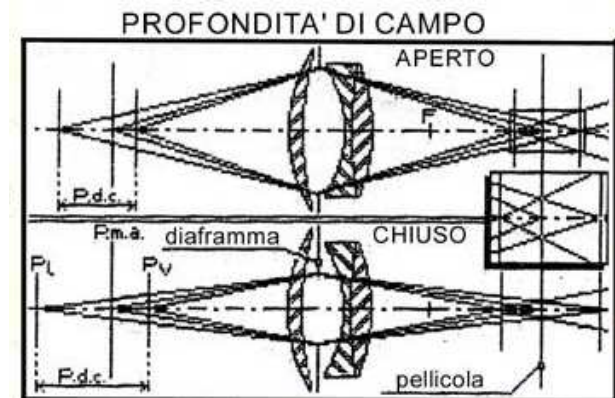
Come si nota dalla figura i raggi che sono raccolti dal bordo della lente subiscono la maggior deflessione. Questo fa sì che in tali zone il modello usato sia meno efficace.



Per attenuare tale fenomeno è conveniente chiudere un po' il diaframma tenendo presente anche le seguenti caratteristiche:

- Chiudendo il diaframma la profondità di campo aumenta (cosa non molto utile se si inclina il CCD);
- Aprendo il diaframma la luminosità aumenta;

Bisogna dunque bilanciare i due fenomeni.



10 Regole d'oro

1. Raccogliere tutte le informazioni di progetto, in particolare:
 - i disegni 3D del montaggio della dima di calibrazione;
 - gli schemi recanti il campo di visione con le quote orizzontali, verticali (e oblique se possibile) rispetto alla base del calibratore e rispetto ad un punto ben identificabile della cassa;
2. Raccogliere i dati sulla focale degli obiettivi;
3. Verificare che il sistema non si debba più manopolare;
4. Test del sistema di movimentazione del bersaglio;
5. Verificare la messa a fuoco del laser e della telecamera in tutto il campo di visione e dell'allineamento dei laser sul bersaglio posto a metà del campo di visione;
6. Regolare i parametri in tutto il campo di visione per eliminare i disturbi e vedere bene il bersaglio;
7. Effettuare la calibrazione in modo che la cassa od il bersaglio non vibrino o peggio si spostino;
8. Verifica risultati di calibrazione (focale, errore medio e massimo);
9. Convertire in millimetri e verificare che si misurino bene le distanze in X ed Y movimentando il calibro;
10. Verificare che un bersaglio rettilineo posto obliquo non venga convertito con una spezzata.

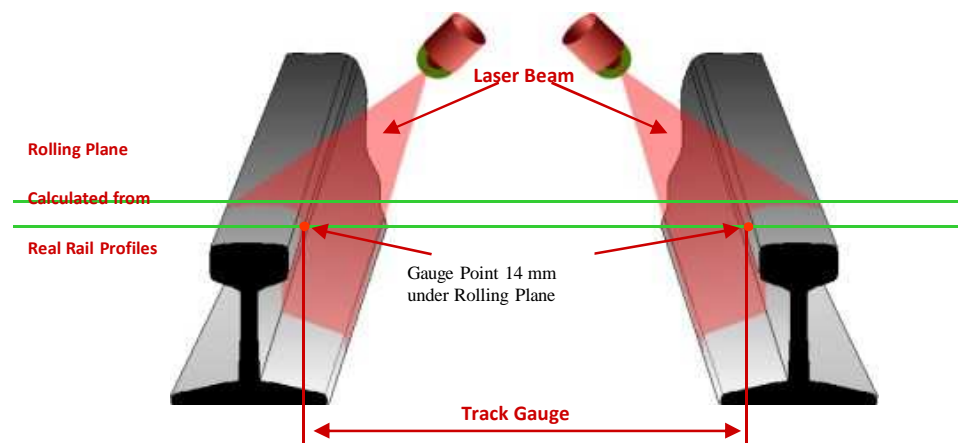
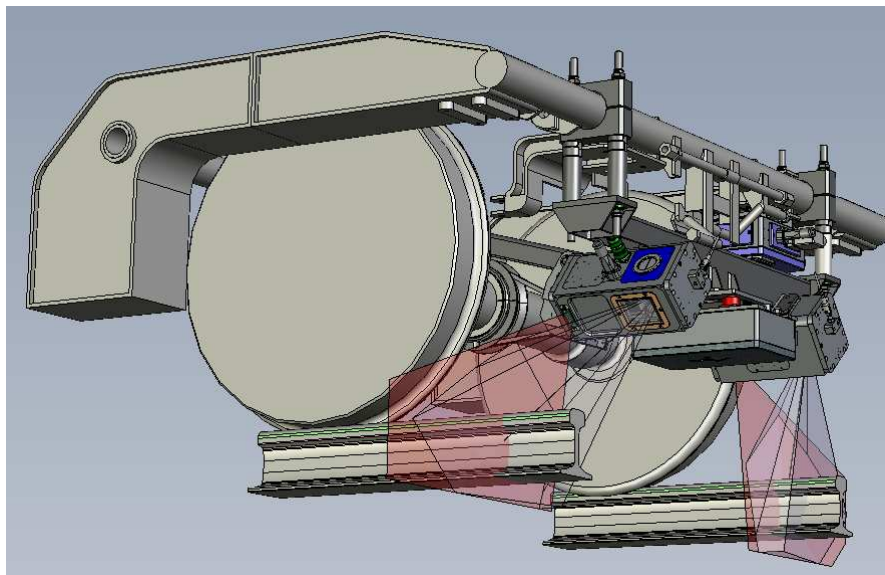
Esempi applicazioni Tecnogamma Mermec Group

Tipi di veicolo



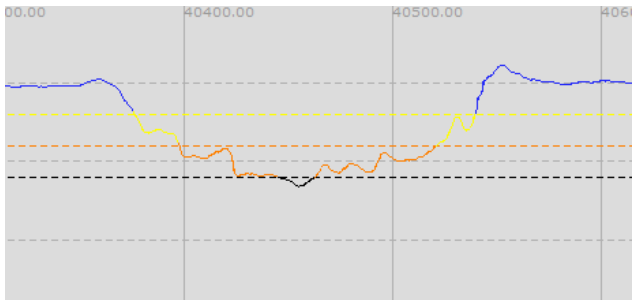
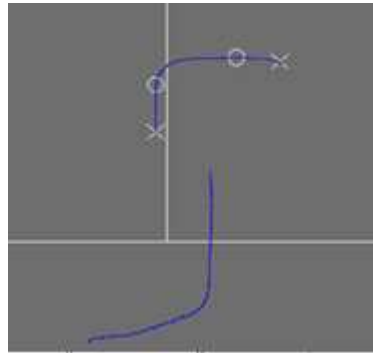
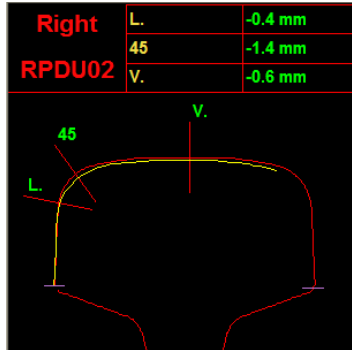
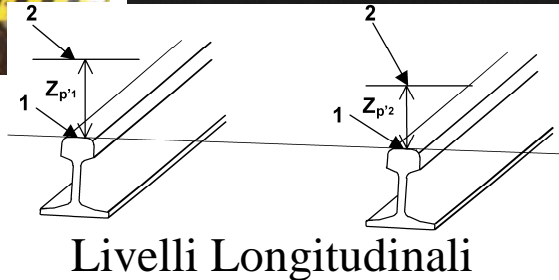
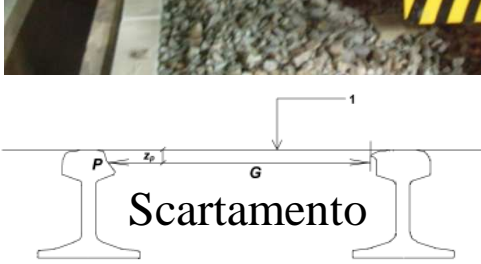
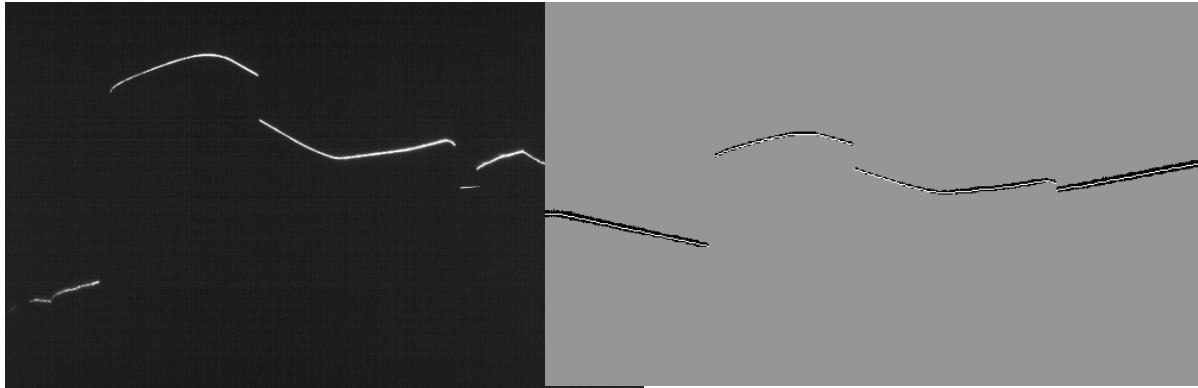
Esempi applicazioni Tecnogamma Mermec Group

Installazione sezione per Track Geometry



Esempi applicazioni Tecnogamma Mermec Group

Track Geometry e Rail Profile



Usura





MERMEC

via Oberdan, 70
70043 Monopoli (BA) Italy
ph. +39 080 8876570
fax +39 080 8874028
www.mermec.com



imagemap

Technical Center
220 Outlet Pointe Blvd.
Columbia, SC 29210, USA
ph. +1 803 213 1200
fax +1 803 798 1909
www.imagemap.com



mermec france

Technopôle de Château-Gombert
Les Baronnies - Bat. A
rue Paul Langevin
13013 Marseille (France)
ph. +33 (0)4 91100190
fax +33 (0)4 91086040
www.inno-tech.fr



tecnogamma

vicolo Ongarie, 13
31050 Morgano (TV), Italy
ph. +39 0422 8391
fax +39 0422 839200
www.tecnogamma.eu



MERMEC group

www.mermecgroup.com

Tsai's camera calibration method revisited

Berthold K.P. Horn

Copyright © 2000

Introduction

Basic camera calibration is the recovery of the principle distance f and the principle point $(x_0, y_0)^T$ in the image plane — or, equivalently, recovery of the position of the center of projection $(x_0, y_0, f)^T$ in the image coordinate system. This is referred to as interior orientation in photogrammetry.

A calibration target can be imaged to provide correspondences between points in the image and points in space. It is, however, generally impractical to position the calibration target accurately with respect to the camera coordinate system using only mechanical means. As a result, the relationship between the target coordinate system and the camera coordinate system typically *also* needs to be recovered from the correspondences. This is referred to as exterior orientation in photogrammetry.

Since cameras often have appreciable geometric distortions, camera calibration is often taken to include the recovery of power series coefficients of these distortions. Furthermore, an unknown scale factor in image sampling may also need to be recovered, because scan lines are typically resampled in the frame grabber, and so picture cells do not correspond discrete sensing elements.

Note that in camera calibration we are trying to recover the transformations, based on measurements of coordinates, where one more often uses known transformation to map coordinates from one coordinate system to another.

Tsai's method for camera calibration recovers the interior orientation, the exterior orientation, the power series coefficients for distortion, and an image scale factor that best fit the measured image coordinates corresponding to known target point coordinates. This is done in stages, starting off with closed form least-squares estimates of some parameters and ending with an iterative non-linear optimization of all parameters simultaneously using these estimates as starting values. Importantly, it is error in the image plane that is minimized.

Details of the method are different for planar targets than for targets occupying some volume in space. Accurate planar targets are easier to make, but lead to some limitations in camera calibration, as pointed out below.

Interior Orientation — Camera to Image

Interior Orientation is the relationship between camera-centric coordinates and image coordinates. The camera coordinate system has its origin at the center of projection, its z axis along the optical axis, and its x and y axes parallel to the x and y axes of the image.

Camera coordinates and image coordinates are related by the perspective projection equations:

$$\frac{x_I - x_0}{f} = \frac{x_C}{z_C} \quad \text{and} \quad \frac{y_I - y_0}{f} = \frac{y_C}{z_C}$$

where f is the principle distance (distance from the center of projection to the image plane), and (x_0, y_0) is the principle point (foot of the perpendicular from the center of projection to the image plane). That is, the center of projection is at $(x_0, y_0, f)^T$, as measured in the image coordinate system.

Interior orientation has three degrees of freedom. The problem of interior orientation is the recovery of x_0 , y_0 , and f . This is the basic task of camera calibration. However, as indicated above, in practice we also need to recover the position and attitude of the calibration target in the camera coordinate system.

Exterior Orientation — Scene to Camera

Exterior Orientation is the relationship between a scene-centered coordinate system and a camera-centered coordinate system. The transformation from scene to camera consists of a rotation and a translation. This transformation has six degrees of freedom, three for rotation and three for translation.

The scene coordinate system can be any system convenient for the particular design of the target. In the case of a planar target, the z axis is chosen perpendicular to the plane, and $z = 0$ in the target plane.

If \mathbf{r}_S are the coordinates of a point measured in the scene coordinate system and \mathbf{r}_C coordinates measured in the camera coordinate system, then

$$\mathbf{r}_C = R(\mathbf{r}_S) + \mathbf{t}$$

where \mathbf{t} is the translation and $R(\dots)$ the rotation.

If we chose for the moment to use an orthonormal matrix to represent rotation, then we can write this in component form:

$$\begin{pmatrix} x_C \\ y_C \\ z_C \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} x_S \\ y_S \\ z_S \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

where $\mathbf{r}_C = (x_C, y_C, z_C)^T$, $\mathbf{r}_S = (x_S, y_S, z_S)^T$, and $\mathbf{t} = (t_x, t_y, t_z)^T$.

The unknowns to be recovered in the problem of exterior orientation are the translation vector \mathbf{t} and the rotation $R(\dots)$.

The Unknown Horizontal Scale Factor

A complicating factor in the calibration of many modern electronic cameras is that the discrete nature of image sampling is not preserved in the signal.

In typical CCD or CMOS cameras, the initially discrete (staircase) sensor signal in analog form is low pass filtered to produce a smooth video output signal in standard form that hides the transitions between cells of the sensor. This waveform is then digitized in the frame grabber. The sampling in the horizontal direction in the frame grabber is typically *not* equal to the spacing of sensor cells, and is not known accurately. The horizontal spacing between pixels in the sampled image do not in general correspond to the horizontal spacing between cells in the image sensor.

This is in contrast with the vertical direction where sampling is controlled by the spacing of rows of sensor cells. Some digital cameras avoid the intermediate analog waveform and the low pass filtering, but many cameras — particularly cheaper ones intended for the consumer market — do not.

In this case the ratio of picture cell size in the horizontal and in the vertical direction is not known a priori from the dimensions of the sensor cells and needs to be determined. This can be done separately using frequency domain methods exploiting limitations of the approximate low pass filter and resulting aliasing effects.

Alternatively, the extra scaling parameter can be recovered as part of the camera calibration process. In this case we use a modified equation for x_I :

$$\frac{x_I - x_0}{f} = s \frac{x_C}{z_C}$$

where s is the unknown ratio of the pixel spacing in the x - and y -directions

It is not possible to recover this extra parameter when using planar targets, as discussed below, and so it has to be estimated separately in that case.

Combining Interior and Exterior Orientation

If we combine the equations for interior and exterior orientation we obtain:

$$\frac{x_I - x_0}{f} = s \frac{r_{11}x_S + r_{12}y_S + r_{13}z_S + t_x}{r_{31}x_S + r_{32}y_S + r_{33}z_S + t_z}$$

$$\frac{y_I - y_0}{f} = \frac{r_{21}x_S + r_{22}y_S + r_{23}z_S + t_y}{r_{31}x_S + r_{32}y_S + r_{33}z_S + t_z}$$

Distortion

Projection in an ideal imaging system is governed by the pin-hole model. Real optical systems suffer from a number of inevitable geometric distortions. In optical systems made of spherical surfaces, with centers along the optical axis, a geometric distortion occurs in the radial direction. A point is imaged at a distance from the principle point that is larger (pin-cushion distortion) or smaller (barrel distortion) than predicted by the perspective projection equations; the displacement increasing with distance from the center. It is small for directions that are near parallel to the optical axis, growing as some power series of the angle. The distortion tends to be more noticeable with wide-angle lenses than with telephoto lenses.

The displacement due to radial distortion can be modelled using the equations:

$$\begin{aligned}\delta x &= x(\kappa_1 r^2 + \kappa_2 r^4 + \dots) \\ \delta y &= y(\kappa_1 r^2 + \kappa_2 r^4 + \dots)\end{aligned}$$

where x and y are measured from the center of distortion, which is typically assumed to be at the principle point. Only even powers of the distance r from the principle point occur, and typically only the first, or perhaps the first and the second term in the power series are retained.

Electro-optical systems typically have larger distortions than optical systems made of glass. They also suffer from tangential distortion, which is at right angle to the vector from the center of the image. Like radial distortion, tangential distortion grows with distance from the center of distortion.

$$\begin{aligned}\delta x &= -y(\epsilon_1 r^2 + \epsilon_2 r^4 + \dots) \\ \delta y &= +x(\epsilon_1 r^2 + \epsilon_2 r^4 + \dots)\end{aligned}$$

In calibration, one attempts to recover the coefficients $(\kappa_1, \dots, \epsilon_1, \dots)$ of these power series. It is also possible to recover distortion parameters *separately* using, for example, the method of plumb lines.

Note that one can express the actual (distorted) image coordinates as a power series using predicted (undistorted) image coordinates as variables, or one can express predicted image coordinates as a power series in the actual image coordinates (that is, the r in the above power series can be either based on actual image coordinates or predicted image coordinates). The power series in the two cases are related by inversion.

If the power series to adjust distorted coordinates to undistorted coordinates is used, then it is more convenient to do the final optimization in undistorted image coordinates rather than distorted (actual) image coordinates.

If instead the power series to adjust undistorted coordinates to distorted coordinates is used, then it is more convenient to do the final optimization in distorted (actual) image coordinates rather than the undistorted image coordinates.

Overall strategy

In calibration, a target of known geometry is imaged. Correspondences between target points and their images are obtained. These form the basic data on which the calibration is based.

Tsai's method first tries to obtain estimates of as many parameters as possible using linear least-squares fitting methods. This is convenient and fast since such problems can be solved using the pseudo-inverse matrix.

In this initial step, constraints between parameters (such as the orthonormality of a rotation matrix) are not enforced, and what is minimized is not the error in the image plane, but a quantity that simplifies the analysis and leads to linear equations. This does not affect the final result, however, since these estimated parameter values are used only as starting values for the final optimization.

In a subsequent step, the rest of the parameters are obtained using a non-linear optimization method that finds the best fit between the observed image points and those predicted from the target model. Parameters estimated in the first step are refined in the process.

Details of the calibration method are different when the target is planar than when it is not. Accurate planar targets are easier to make and maintain than three-dimensional targets, but limit calibration in ways that will become apparent. We start by analysing the case of the non-coplanar target.

Estimating the rotation, and part of the translation

Initially we assume that we have a reasonable estimate of the position of the principle point (x_0, y_0) . This point is usually near the middle of the CCD or CMOS sensor. We refer coordinates to this point using

$$x'_I = x_I - x_0 \quad \text{and} \quad y'_I = y_I - y_0$$

so that

$$\frac{x'_I}{f} = s \frac{x_C}{z_C} \quad \text{and} \quad \frac{y'_I}{f} = \frac{y_C}{z_C}$$

Next, we consider only the *direction* of the point in the image as measured from the principle point. This yields a result that is independent of the unknown principle distance f . It is *also* independent of radial distortion.

$$\frac{x'_I}{y'_I} = s \frac{x_C}{y_C}$$

Using the expansion in terms of components of the rotation matrix R we obtain:

$$\frac{x'_I}{y'_I} = s \frac{r_{11}x_S + r_{12}y_S + r_{13}z_S + t_x}{r_{21}x_S + r_{22}y_S + r_{23}z_S + t_y}$$

which becomes, after cross multiplying:

$$s(r_{11}x_S + r_{12}y_S + r_{13}z_S + t_x)y'_I - (r_{21}x_S + r_{22}y_S + r_{23}z_S + t_y)x'_I = 0$$

or

$$(x_S y'_I)sr_{11} + (y_S y'_I)sr_{12} + (z_S y'_I)sr_{13} + y'_I st_x \\ - (x_S x'_I)r_{21} - (y_S x'_I)r_{22} - (z_S x'_I)r_{23} - x'_I t_y = 0$$

which we can view as a linear homogeneous equation in the eight unknowns

$$sr_{11}, sr_{12}, sr_{13}, r_{21}, r_{22}, r_{23}, st_x, \quad \text{and} \quad t_y$$

The coefficients in the equation are products of components of corresponding scene and image coordinates.

We obtain one such equation for every correspondence between a calibration target point $(x_{S_i}, y_{S_i}, z_{S_i})^T$ and an image point $(x_{I_i}, y_{I_i})^T$.

Note that there is an unknown scale factor here because these equations are homogeneous. That is, if we have a solution for the eight unknowns, then any multiple of that solution is also a solution. In order to obtain a solution, we can convert the homogeneous equation into inhomogeneous equation by arbitrarily setting one unknown — t_y say — to one.

We then have seven unknowns for which we can solve if we have seven correspondences between target coordinates and image coordinates. If we have more than seven correspondences, we can minimize the sum of squares of errors using the pseudo-inverse method.

When we obtain the solution, we do have to remember that the eight unknowns (the seven we solved for plus the one we set to one) can be scaled by an arbitrary factor. Suppose that the best fit solution of the set of equations is

$$sr'_{11}, sr'_{12}, sr'_{13}, st'_x, r'_{21}, r'_{22}, r'_{23}, \quad \text{and} \quad t'_y = 1$$

We can estimate the correct scale factor by noting that the rows of the rotation matrix are supposed to be normal, that is

$$r_{11}^2 + r_{12}^2 + r_{13}^2 = 1 \quad \text{and} \quad r_{21}^2 + r_{22}^2 + r_{23}^2 = 1$$

We need to find the scale factor c to be applied to the solution to satisfy these equalities. It is easy to see that

$$c = 1/\sqrt{r'^2_{21} + r'^2_{22} + r'^2_{23}}$$

and

$$c/s = 1/\sqrt{(sr'_{11})^2 + (sr'_{12})^2 + (sr'_{13})^2}$$

These equations allow us to recover the factor c as well as the ratio s of the horizontal pixel spacing to the vertical pixel spacing.

Note that we did not enforce the orthogonality of the first two rows of the rotation matrix. We can improve matters by adjusting them to make them orthogonal.

Forcing orthonormality of the first two rows

Given the vectors \mathbf{a} and \mathbf{b} , we can find two orthogonal vectors \mathbf{a}' and \mathbf{b}' that are as near as possible to \mathbf{a} and \mathbf{b} as follows:

$$\mathbf{a}' = \mathbf{a} + k\mathbf{b} \quad \text{and} \quad \mathbf{b}' = \mathbf{b} + k\mathbf{a}$$

$$\mathbf{a}' \cdot \mathbf{b}' = \mathbf{a} \cdot \mathbf{b} + k(\mathbf{a} \cdot \mathbf{a} + \mathbf{b} \cdot \mathbf{b}) + k^2\mathbf{a} \cdot \mathbf{b} = 0$$

The solution of this quadratic for k tends to be numerically ill behaved since the first and last coefficients ($\mathbf{a} \cdot \mathbf{b}$) are small when the vectors are already close to being orthogonal. The following approximate solution can be used then

$$k \approx -(1/2)\mathbf{a} \cdot \mathbf{b}$$

(since k is expected to be small and $\mathbf{a} \cdot \mathbf{a}$ and $\mathbf{b} \cdot \mathbf{b}$ near one).

We have to renormalize the first two rows of the rotation matrix after adjusting them to make them orthogonal.

Finally we can obtain the third row of the rotation matrix simply by taking the cross-product of the first two rows. The resulting 3×3 matrix will be orthonormal if we performed the above orthonormalization of the first two rows.

Note that there may be problems with this method of solution if the unknown (t_y) that we decided to set equal to one in order to solve the homogeneous equations happens to be near zero. In this case a solution may be obtained by setting another unknown (t_x perhaps) equal to one. An alternative is to translate the experimental data by some offset in y .

Also note that we did not minimize the error in the image, but some other quantity that led to convenient, linear equations. The resulting rotation matrix and translation components are not especially accurate as a result. This is acceptable *only* because we use the recovered values merely as estimates in the full non-linear optimization described below.

Coplanar Target

The above method cannot be used as is when the target is planar. It turns out that we cannot recover the scale factor s in this case, so we assume that image coordinates have already been adjusted to account for any differences in scaling in the x and y directions.

With a planar target we can always arrange the coordinate system such that $z_S = 0$ for points on the target. This means the products with r_{13} , r_{23} , and r_{33}

drop out of the equations for the image coordinates and we obtain:

$$\frac{x'_I}{y'_I} = \frac{r_{11}x_S + r_{12}y_S + t_x}{r_{21}x_S + r_{22}y_S + t_y}$$

which becomes, after cross multiplying:

$$(r_{11}x_S + r_{12}y_S + t_x)y'_I - (r_{21}x_S + r_{22}y_S + t_y)x'_I = 0$$

or

$$(x_S y'_I) r_{11} + (y_S y'_I) r_{12} + y'_I t_x - (x_S x'_I) r_{21} - (y_S x'_I) r_{22} - x'_I t_y = 0$$

a linear homogeneous equation in the six unknowns

$$r_{11}, r_{12}, r_{21}, r_{22}, t_x, \text{ and } t_y$$

The coefficients in this equation are products of components of corresponding scene and image coordinates.

We obtain one such equation for every correspondence between a calibration target point $(x_{S_i}, y_{S_i}, z_{S_i})^T$ and an image point $(x_{I_i}, y_{I_i})^T$.

As in the non-coplanar case, there is an unknown scale factor because these equations are homogeneous. If we have a solution for the six unknowns, then any multiple of that solution is also a solution. We can convert the homogeneous equations into inhomogeneous equations by arbitrarily setting one unknown — t_y say — to one.

We then have five unknowns for which we can solve if we have five correspondences between target coordinates and image coordinates. If we have more than five correspondences, we can minimize the sum of squares of errors using the pseudo-inverse.

We do have to remember though that the six unknowns (the five we solved for plus the one we set equal to one) can be scaled by an arbitrary factor.

Suppose that the best fit solution of the set of equations is

$$r'_{11}, r'_{12}, r'_{21}, r'_{22}, t'_x, \text{ and } t'_y = 1$$

We now are faced with the task of estimating the full 3×3 rotation matrix based only on its top left 2×2 sub-matrix.

Recovering the full rotation matrix for planar target

We can estimate the correct scale factor by noting that the rotation matrix is supposed to be orthonormal and hence

$$\begin{aligned} r_{11}^2 + r_{12}^2 + r_{13}^2 &= 1 \\ r_{21}^2 + r_{22}^2 + r_{23}^2 &= 1 \\ r_{11}r_{21} + r_{12}r_{22} + r_{13}r_{23} &= 0 \end{aligned}$$

so we have

$$r'_{11}{}^2 + r'_{12}{}^2 + r'_{13}{}^2 = k^2$$

$$r'_{21}{}^2 + r'_{22}{}^2 + r'_{23}{}^2 = k^2$$

$$r'_{11}r'_{21} + r'_{12}r'_{22} + r'_{13}r'_{23} = 0$$

From the first two equations we have

$$(r'_{11}{}^2 + r'_{12}{}^2)(r'_{21}{}^2 + r'_{22}{}^2) = (k^2 - r'_{13}{}^2)(k^2 - r'_{23}{}^2)$$

From the third equation we get

$$(r'_{11}r'_{21} + r'_{12}r'_{22})^2 = (r'_{13}r'_{23})^2$$

Subtracting we obtain

$$(r'_{11}r'_{22} - r'_{12}r'_{21})^2 = k^4 - k^2(r'_{13}{}^2 + r'_{23}{}^2)$$

Since

$$(r'_{13}{}^2 + r'_{23}{}^2) = 2k^2 - (r'_{11}{}^2 + r'_{12}{}^2 + r'_{21}{}^2 + r'_{22}{}^2)$$

we end up with

$$k^4 - k^2(r'_{11}{}^2 + r'_{12}{}^2 + r'_{21}{}^2 + r'_{22}{}^2) + (r'_{11}r'_{22} - r'_{12}r'_{21})^2 = 0$$

a quadratic in k^2 . This then allows us to calculate the missing coefficients r'_{13} and r'_{23} in the first two rows of the rotation matrix using

$$r'_{13}{}^2 = k^2 - (r'_{11}{}^2 + r'_{12}{}^2)$$

$$r'_{23}{}^2 = k^2 - (r'_{21}{}^2 + r'_{22}{}^2)$$

Only the more positive of the two roots for k^2 makes the right hand sides of these equations positive, so we only need to consider that root.

$$k^2 = \frac{1}{2} \left((r'_{11}{}^2 + r'_{12}{}^2 + r'_{21}{}^2 + r'_{22}{}^2) + \sqrt{((r'_{11} - r'_{22})^2 + (r'_{12} + r'_{21})^2)((r'_{11} + r'_{22})^2 + (r'_{12} - r'_{21})^2)} \right)$$

We normalize the first two rows of the rotation matrix by dividing by k . Finally, we make up the third row by taking the cross-product of the first two rows.

There are, however, sign ambiguities in the calculation of r'_{13} and r'_{23} . We can get the sign of the product $r'_{13}r'_{23}$ using the fact that

$$r_{13}r_{23} = -(r_{11}r_{21} + r_{12}r_{22})$$

so there is only a two-way ambiguity — but we do need to pick the proper sign to get the proper rotation matrix.

One way to pick the correct signs for r_{13} and r_{23} is to use the resulting transformation to project the target points back into the image. If the signs are correct, the predicted image positions will be close to the observed image positions. If the signs of these two components of the rotation matrix are picked incorrectly, then the first two components of the third row of the estimated rotation matrix will be wrong also, since that row is the cross-product of the first two. As a result

many predicted image points will lie in the wrong quadrant of the image. We can test for this condition by taking dot-products of vectors in the image plane — measured from the estimated principle point — of corresponding measured and predicted image positions. We try the other signs for r_{13} and r_{23} if

$$\sum_{i=1}^N (x_{Ii}x_{Pi} + y_{Ii}y_{Pi}) < 0.$$

Estimating principle distance and distance to the scene

So far we have estimated the rotation matrix R and the first two components of the translation (t_x and t_y). We do not yet have estimates for the third component (t_z) of the translation, or the principle distance f . We can estimate these two parameters starting from:

$$\frac{x'_I}{f} = s \frac{r_{11}x_S + r_{12}y_S + r_{13}z_S + t_x}{r_{31}x_S + r_{32}y_S + r_{33}z_S + t_z}$$

$$\frac{y'_I}{f} = \frac{r_{21}x_S + r_{22}y_S + r_{23}z_S + t_y}{r_{31}x_S + r_{32}y_S + r_{33}z_S + t_z}$$

Cross multiplying we find

$$s(r_{11}x_S + r_{12}y_S + r_{13}z_S + t_x)f - x'_I t_z = (r_{31}x_S + r_{32}y_S + r_{33}z_S)x'_I$$

$$(r_{21}x_S + r_{22}y_S + r_{23}z_S + t_y)f - y'_I t_z = (r_{31}x_S + r_{32}y_S + r_{33}z_S)y'_I$$

Given that we have estimates for $\{r_{ij}\}$, we can treat these as linear equations in the two unknowns f and t_z . We can solve these equations for f and t_z using one or more correspondence between target and image. If we use many correspondences we can solve the resulting over-determined system using least-squares methods.

If the horizontal scale factor s is not known accurately we may want to only use the equations for y'_I , rather than equations for both x'_I and y'_I .

At this point we have estimates of the rotation matrix R , the translation vector $\mathbf{t} = (t_x, t_y, t_z)^T$, as well as the principle distance f . We still need to find the principle point (x_0, y_0) and the coefficients of the distortion power series. We also need to refine the parameters estimated so far, since these estimates are *not* based on minimization of the image error, but some convenient linear equations.

Note that the target must span a range of depth values (z_C) in order to recover f and t_z . If target points are all at the same depth, then their image positions depend only on the ratio f/t_z . Hence f and t_z cannot be determined separately. Accuracy improves with the depth range of target points. In the case of a planar target this means that the target normal *must* be turned away from the optical axis.

Non-linear optimization

At this point we minimize the image errors, that is, the difference between the observed image positions $(x_I, y_I)^T$ and the positions $(x_P, y_P)^T$ predicted based on the known target coordinates $(x_S, y_S, z_S)^T$. The parameters of interior orientation, exterior orientation and distortion are adjusted to minimize

$$\sum_{i=1}^N (x_{Ii} - x_{Pi})^2 + \sum_{i=1}^N (y_{Ii} - y_{Pi})^2$$

This is best done using iterative numerical optimization such as a modified Levenberg-Marquardt method.

Non-linear optimization methods work best when they have full access to the components of the error. In the case here, it is best to treat $(x_{Ii} - x_{Pi})$ and $(y_{Ii} - y_{Pi})$ as separate error components rather than, for example, lumping them into a combined error term of the form

$$\sqrt{(x_{Ii} - x_{Pi})^2 + (y_{Ii} - y_{Pi})^2}$$

Representation of rotation

To use non-linear optimization methods we need a non-redundant parameterization for rotation. Orthonormal matrices are redundant since they use nine numbers to represent just three degrees of freedom. Maintaining the six constraints of orthonormality in the minimization is very difficult.

Tsai's original method used Euler angles to represent rotations. An alternative is the Gibb's vector

$$\hat{\omega} \tan(\theta/2)$$

The direction of the Gibb's vector is the axis about which the rotation takes place, $\hat{\omega}$, while its magnitude is the tangent of half the angle of rotation, θ . Like all non-redundant representations for rotation, the Gibb's vector has a singularity. It occurs when the rotation is through π radians.

An alternative is to use a redundant representation that has no singularities — and enforce the required non-linear constraint. The axis-and-angle representation of rotation can be related to the unit quaternion notation. The unit quaternion for the rotation is

$$\hat{q} = (\cos(\theta/2), \hat{\omega} \sin(\theta/2))$$

The needed constraint that this be a unit vector is easily incorporated in the non-linear optimization by adding an error term of the form

$$(\hat{q} \cdot \hat{q} - 1)^2$$

Sensitivity of solution

Error in the calibration parameters is directly proportional to error in image measurements. The proportionality factor depends on the imaging geometry, and on the design of the target. Some parameters of the calibration are more sensitive to error than others, and some behave worse when the field of view is narrow or the depth range of the target limited.

It is difficult to investigate the sensitivities to noise analytically because of the non-linearity of the imaging model. However, the sensitivity issue can be studied easily using Monte Carlo simulation. Add random noise to the calibration image data and recompute the calibration parameters. Repeat and collect mean and covariance statistics.

References and Acknowledgements

- Horn, B.K.P. (1986) *Robot Vision*, MIT Press, Cambridge, Massachusetts and McGraw-Hill, New York.
- Horn, B.K.P. (1987) “Closed Form Solution of Absolute Orientation using Unit Quaternions,” *Journal of the Optical Society A*, Vol. 4, No. 4, pp. 629–642, April.
- Horn, B.K.P., H. M. Hilden & S. Negahdaripour, (1988) “Closed Form Solution of Absolute Orientation using Orthonormal Matrices,” *Journal of the Optical Society A*, Vol. 5, No. 7, pp. 1127–1135, July.
- Horn, B.K.P. (1990) “Relative Orientation,” *International Journal of Computer Vision*, Vol. 4, No. 1, pp. 59–78, January.
- Horn, B.K.P. (1991) “Relative Orientation Revisited,” *Journal of the Optical Society of America, A*, Vol. 8, pp. 1630–1638, October.
- Tsai, Roger Y. (1986) “An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, 1986, pp. 364–374.
- Tsai, Roger Y. (1987) “A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses,” *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 4, August 1987, pp. 323–344.

An implementation of Tsai’s method dating to 1995 by Reg Willson of 3M in St. Paul, MN may be found on the web. Willson further acknowledges Piotr Jasiobedzki, Jim Vaughan, Ron Steriti, Torfi Thorhallsson, Frederic Devernay, Volker Rodehorst, and Jon Owen.

The public domain MINPACK “lmdif” package can be used for the non-linear optimization. This uses a modified Levenberg-Marquardt algorithm with a Jacobian calculated by a forward-difference approximation.

A Flexible New Technique for Camera Calibration

Zhengyou Zhang

December 2, 1998
(updated on December 14, 1998)
(updated on March 25, 1999)
(updated on Aug. 10, 2002; a typo in Appendix B)
(last updated on Aug. 13, 2008; a typo in Section 3.3)

Technical Report
MSR-TR-98-71

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

zhang@microsoft.com
<http://research.microsoft.com/~zhang>

A Flexible New Technique for Camera Calibration

Zhengyou Zhang

Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399, USA
zhang@microsoft.com <http://research.microsoft.com/~zhang>

Contents

1	Motivations	2
2	Basic Equations	3
2.1	Notation	3
2.2	Homography between the model plane and its image	4
2.3	Constraints on the intrinsic parameters	4
2.4	Geometric Interpretation [†]	4
3	Solving Camera Calibration	5
3.1	Closed-form solution	5
3.2	Maximum likelihood estimation	6
3.3	Dealing with radial distortion	7
3.4	Summary	8
4	Degenerate Configurations	8
5	Experimental Results	9
5.1	Computer Simulations	9
5.2	Real Data	10
5.3	Sensitivity with Respect to Model Imprecision [‡]	14
5.3.1	Random noise in the model points	14
5.3.2	Systematic non-planarity of the model pattern	15
6	Conclusion	17
A	Estimation of the Homography Between the Model Plane and its Image	17
B	Extraction of the Intrinsic Parameters from Matrix B	18
C	Approximating a 3×3 matrix by a Rotation Matrix	18
D	Camera Calibration Under Known Pure Translation[§]	19

[†]added on December 14, 1998

[‡]added on December 28, 1998; added results on systematic non-planarity on March 25, 1998

[§]added on December 14, 1998, corrected (based on the comments from Andrew Zisserman) on January 7, 1999

A Flexible New Technique for Camera Calibration

Abstract

We propose a flexible new technique to easily calibrate a camera. It is well suited for use without specialized knowledge of 3D geometry or computer vision. The technique only requires the camera to observe a planar pattern shown at a few (at least two) different orientations. Either the camera or the planar pattern can be freely moved. The motion need not be known. Radial lens distortion is modeled. The proposed procedure consists of a closed-form solution, followed by a nonlinear refinement based on the maximum likelihood criterion. Both computer simulation and real data have been used to test the proposed technique, and very good results have been obtained. Compared with classical techniques which use expensive equipment such as two or three orthogonal planes, the proposed technique is easy to use and flexible. It advances 3D computer vision one step from laboratory environments to real world use.

Index Terms— Camera calibration, calibration from planes, 2D pattern, absolute conic, projective mapping, lens distortion, closed-form solution, maximum likelihood estimation, flexible setup.

1 Motivations

Camera calibration is a necessary step in 3D computer vision in order to extract metric information from 2D images. Much work has been done, starting in the photogrammetry community (see [2, 4] to cite a few), and more recently in computer vision ([9, 8, 23, 7, 26, 24, 17, 6] to cite a few). We can classify those techniques roughly into two categories: photogrammetric calibration and self-calibration.

Photogrammetric calibration. Camera calibration is performed by observing a calibration object whose geometry in 3-D space is known with very good precision. Calibration can be done very efficiently [5]. The calibration object usually consists of two or three planes orthogonal to each other. Sometimes, a plane undergoing a precisely known translation is also used [23]. These approaches require an expensive calibration apparatus, and an elaborate setup.

Self-calibration. Techniques in this category do not use any calibration object. Just by moving a camera in a static scene, the rigidity of the scene provides in general two constraints [17, 15] on the cameras' internal parameters from one camera displacement by using image information alone. Therefore, if images are taken by the same camera with fixed internal parameters, correspondences between three images are sufficient to recover both the internal and external parameters which allow us to reconstruct 3-D structure up to a similarity [16, 13]. While this approach is very flexible, it is not yet mature [1]. Because there are many parameters to estimate, we cannot always obtain reliable results.

Other techniques exist: vanishing points for orthogonal directions [3, 14], and calibration from pure rotation [11, 21].

Our current research is focused on a desktop vision system (DVS) since the potential for using DVSs is large. Cameras are becoming cheap and ubiquitous. A DVS aims at the general public, who are not experts in computer vision. A typical computer user will perform vision tasks only from time to time, so will not be willing to invest money for expensive equipment. Therefore, flexibility, robustness and low cost are important. The camera calibration technique described in this paper was developed with these considerations in mind.

The proposed technique only requires the camera to observe a planar pattern shown at a few (at least two) different orientations. The pattern can be printed on a laser printer and attached to a “reasonable” planar surface (e.g., a hard book cover). Either the camera or the planar pattern can be moved by hand. The motion need not be known. The proposed approach lies between the photogrammetric calibration and self-calibration, because we use 2D metric information rather than 3D or purely implicit one. Both computer simulation and real data have been used to test the proposed technique, and very good results have been obtained. Compared with classical techniques, the proposed technique is considerably more flexible. Compared with self-calibration, it gains considerable degree of robustness. We believe the new technique advances 3D computer vision one step from laboratory environments to the real world.

Note that Bill Triggs [22] recently developed a self-calibration technique from at least 5 views of a planar scene. His technique is more flexible than ours, but has difficulty to initialize. Liebowitz and Zisserman [14] described a technique of metric rectification for perspective images of planes using metric information such as a known angle, two equal though unknown angles, and a known length ratio. They also mentioned that calibration of the internal camera parameters is possible provided at least three such rectified planes, although no experimental results were shown.

The paper is organized as follows. Section 2 describes the basic constraints from observing a single plane. Section 3 describes the calibration procedure. We start with a closed-form solution, followed by nonlinear optimization. Radial lens distortion is also modeled. Section 4 studies configurations in which the proposed calibration technique fails. It is very easy to avoid such situations in practice. Section 5 provides the experimental results. Both computer simulation and real data are used to validate the proposed technique. In the Appendix, we provide a number of details, including the techniques for estimating the homography between the model plane and its image.

2 Basic Equations

We examine the constraints on the camera’s intrinsic parameters provided by observing a single plane. We start with the notation used in this paper.

2.1 Notation

A 2D point is denoted by $\mathbf{m} = [u, v]^T$. A 3D point is denoted by $\mathbf{M} = [X, Y, Z]^T$. We use $\tilde{\mathbf{x}}$ to denote the augmented vector by adding 1 as the last element: $\tilde{\mathbf{m}} = [u, v, 1]^T$ and $\tilde{\mathbf{M}} = [X, Y, Z, 1]^T$. A camera is modeled by the usual pinhole: the relationship between a 3D point \mathbf{M} and its image projection \mathbf{m} is given by

$$s\tilde{\mathbf{m}} = \mathbf{A}[\mathbf{R} \quad \mathbf{t}]\tilde{\mathbf{M}}, \quad (1)$$

where s is an arbitrary scale factor, (\mathbf{R}, \mathbf{t}) , called the extrinsic parameters, is the rotation and translation which relates the world coordinate system to the camera coordinate system, and \mathbf{A} , called the camera intrinsic matrix, is given by

$$\mathbf{A} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

with (u_0, v_0) the coordinates of the principal point, α and β the scale factors in image u and v axes, and γ the parameter describing the skewness of the two image axes.

We use the abbreviation \mathbf{A}^{-T} for $(\mathbf{A}^{-1})^T$ or $(\mathbf{A}^T)^{-1}$.

2.2 Homography between the model plane and its image

Without loss of generality, we assume the model plane is on $Z = 0$ of the world coordinate system. Let's denote the i^{th} column of the rotation matrix \mathbf{R} by \mathbf{r}_i . From (1), we have

$$\begin{aligned} s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \mathbf{A} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \\ &= \mathbf{A} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} . \end{aligned}$$

By abuse of notation, we still use \mathbf{M} to denote a point on the model plane, but $\mathbf{M} = [X, Y]^T$ since Z is always equal to 0. In turn, $\tilde{\mathbf{M}} = [X, Y, 1]^T$. Therefore, a model point \mathbf{M} and its image \mathbf{m} is related by a homography \mathbf{H} :

$$s\tilde{\mathbf{m}} = \mathbf{H}\tilde{\mathbf{M}} \quad \text{with} \quad \mathbf{H} = \mathbf{A} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} . \quad (2)$$

As is clear, the 3×3 matrix \mathbf{H} is defined up to a scale factor.

2.3 Constraints on the intrinsic parameters

Given an image of the model plane, an homography can be estimated (see Appendix A). Let's denote it by $\mathbf{H} = [\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3]$. From (2), we have

$$[\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3] = \lambda \mathbf{A} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} ,$$

where λ is an arbitrary scalar. Using the knowledge that \mathbf{r}_1 and \mathbf{r}_2 are orthonormal, we have

$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2 = 0 \quad (3)$$

$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2 . \quad (4)$$

These are the two basic constraints on the intrinsic parameters, given one homography. Because a homography has 8 degrees of freedom and there are 6 extrinsic parameters (3 for rotation and 3 for translation), we can only obtain 2 constraints on the intrinsic parameters. Note that $\mathbf{A}^{-T} \mathbf{A}^{-1}$ actually describes the image of the absolute conic [16]. In the next subsection, we will give an geometric interpretation.

2.4 Geometric Interpretation

We are now relating (3) and (4) to the absolute conic.

It is not difficult to verify that the model plane, under our convention, is described in the camera coordinate system by the following equation:

$$\begin{bmatrix} \mathbf{r}_3 \\ \mathbf{r}_3^T \mathbf{t} \end{bmatrix}^T \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = 0 ,$$

where $w = 0$ for points at infinity and $w = 1$ otherwise. This plane intersects the plane at infinity at a line, and we can easily see that $\begin{bmatrix} \mathbf{r}_1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} \mathbf{r}_2 \\ 0 \end{bmatrix}$ are two particular points on that line. Any point on it

is a linear combination of these two points, i.e.,

$$\mathbf{x}_\infty = a \begin{bmatrix} \mathbf{r}_1 \\ 0 \end{bmatrix} + b \begin{bmatrix} \mathbf{r}_2 \\ 0 \end{bmatrix} = \begin{bmatrix} a\mathbf{r}_1 + b\mathbf{r}_2 \\ 0 \end{bmatrix} .$$

Now, let's compute the intersection of the above line with the absolute conic. By definition, the point \mathbf{x}_∞ , known as the *circular point*, satisfies: $\mathbf{x}_\infty^T \mathbf{x}_\infty = 0$, i.e.,

$$(a\mathbf{r}_1 + b\mathbf{r}_2)^T (a\mathbf{r}_1 + b\mathbf{r}_2) = 0, \quad \text{or} \quad a^2 + b^2 = 0 .$$

The solution is $b = \pm ai$, where $i^2 = -1$. That is, the two intersection points are

$$\mathbf{x}_\infty = a \begin{bmatrix} \mathbf{r}_1 \pm i\mathbf{r}_2 \\ 0 \end{bmatrix} .$$

Their projection in the image plane is then given, up to a scale factor, by

$$\tilde{\mathbf{m}}_\infty = \mathbf{A}(\mathbf{r}_1 \pm i\mathbf{r}_2) = \mathbf{h}_1 \pm i\mathbf{h}_2 .$$

Point $\tilde{\mathbf{m}}_\infty$ is on the image of the absolute conic, described by $\mathbf{A}^{-T} \mathbf{A}^{-1}$ [16]. This gives

$$(\mathbf{h}_1 \pm i\mathbf{h}_2)^T \mathbf{A}^{-T} \mathbf{A}^{-1} (\mathbf{h}_1 \pm i\mathbf{h}_2) = 0 .$$

Requiring that both real and imaginary parts be zero yields (3) and (4).

3 Solving Camera Calibration

This section provides the details how to effectively solve the camera calibration problem. We start with an analytical solution, followed by a nonlinear optimization technique based on the maximum likelihood criterion. Finally, we take into account lens distortion, giving both analytical and nonlinear solutions.

3.1 Closed-form solution

Let

$$\begin{aligned} \mathbf{B} &= \mathbf{A}^{-T} \mathbf{A}^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix} . \end{aligned} \quad (5)$$

Note that \mathbf{B} is symmetric, defined by a 6D vector

$$\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T . \quad (6)$$

Let the i^{th} column vector of \mathbf{H} be $\mathbf{h}_i = [h_{i1}, h_{i2}, h_{i3}]^T$. Then, we have

$$\mathbf{h}_i^T \mathbf{B} \mathbf{h}_j = \mathbf{v}_{ij}^T \mathbf{b} \quad (7)$$

with

$$\mathbf{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, \\ h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T .$$

Therefore, the two fundamental constraints (3) and (4), from a given homography, can be rewritten as 2 homogeneous equations in \mathbf{b} :

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = \mathbf{0} . \quad (8)$$

If n images of the model plane are observed, by stacking n such equations as (8) we have

$$\mathbf{V}\mathbf{b} = \mathbf{0} , \quad (9)$$

where \mathbf{V} is a $2n \times 6$ matrix. If $n \geq 3$, we will have in general a unique solution \mathbf{b} defined up to a scale factor. If $n = 2$, we can impose the skewless constraint $\gamma = 0$, i.e., $[0, 1, 0, 0, 0, 0]\mathbf{b} = 0$, which is added as an additional equation to (9). (If $n = 1$, we can only solve two camera intrinsic parameters, e.g., α and β , assuming u_0 and v_0 are known (e.g., at the image center) and $\gamma = 0$, and that is indeed what we did in [19] for head pose determination based on the fact that eyes and mouth are reasonably coplanar.) The solution to (9) is well known as the eigenvector of $\mathbf{V}^T\mathbf{V}$ associated with the smallest eigenvalue (equivalently, the right singular vector of \mathbf{V} associated with the smallest singular value).

Once \mathbf{b} is estimated, we can compute all camera intrinsic matrix \mathbf{A} . See Appendix B for the details.

Once \mathbf{A} is known, the extrinsic parameters for each image is readily computed. From (2), we have

$$\begin{aligned} \mathbf{r}_1 &= \lambda \mathbf{A}^{-1} \mathbf{h}_1 \\ \mathbf{r}_2 &= \lambda \mathbf{A}^{-1} \mathbf{h}_2 \\ \mathbf{r}_3 &= \mathbf{r}_1 \times \mathbf{r}_2 \\ \mathbf{t} &= \lambda \mathbf{A}^{-1} \mathbf{h}_3 \end{aligned}$$

with $\lambda = 1/\|\mathbf{A}^{-1}\mathbf{h}_1\| = 1/\|\mathbf{A}^{-1}\mathbf{h}_2\|$. Of course, because of noise in data, the so-computed matrix $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ does not in general satisfy the properties of a rotation matrix. Appendix C describes a method to estimate the best rotation matrix from a general 3×3 matrix.

3.2 Maximum likelihood estimation

The above solution is obtained through minimizing an algebraic distance which is not physically meaningful. We can refine it through maximum likelihood inference.

We are given n images of a model plane and there are m points on the model plane. Assume that the image points are corrupted by independent and identically distributed noise. The maximum likelihood estimate can be obtained by minimizing the following functional:

$$\sum_{i=1}^n \sum_{j=1}^m \|\mathbf{m}_{ij} - \hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M}_j)\|^2 , \quad (10)$$

where $\hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M}_j)$ is the projection of point \mathbf{M}_j in image i , according to equation (2). A rotation \mathbf{R} is parameterized by a vector of 3 parameters, denoted by \mathbf{r} , which is parallel to the rotation axis and whose magnitude is equal to the rotation angle. \mathbf{R} and \mathbf{r} are related by the Rodrigues formula [5]. Minimizing (10) is a nonlinear minimization problem, which is solved with the Levenberg-Marquardt Algorithm as implemented in `Minpack` [18]. It requires an initial guess of \mathbf{A} , $\{\mathbf{R}_i, \mathbf{t}_i | i = 1..n\}$ which can be obtained using the technique described in the previous subsection.

3.3 Dealing with radial distortion

Up to now, we have not considered lens distortion of a camera. However, a desktop camera usually exhibits significant lens distortion, especially radial distortion. In this section, we only consider the first two terms of radial distortion. The reader is referred to [20, 2, 4, 26] for more elaborated models. Based on the reports in the literature [2, 23, 25], it is likely that the distortion function is totally dominated by the radial components, and especially dominated by the first term. It has also been found that any more elaborated modeling not only would not help (negligible when compared with sensor quantization), but also would cause numerical instability [23, 25].

Let (u, v) be the ideal (nonobservable distortion-free) pixel image coordinates, and (\check{u}, \check{v}) the corresponding real observed image coordinates. The ideal points are the projection of the model points according to the pinhole model. Similarly, (x, y) and (\check{x}, \check{y}) are the ideal (distortion-free) and real (distorted) normalized image coordinates. We have [2, 25]

$$\begin{aligned}\check{x} &= x + x[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \\ \check{y} &= y + y[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2],\end{aligned}$$

where k_1 and k_2 are the coefficients of the radial distortion. The center of the radial distortion is the same as the principal point. From* $\check{u} = u_0 + \alpha\check{x} + \gamma\check{y}$ and $\check{v} = v_0 + \beta\check{y}$ and assuming $\gamma = 0$, we have

$$\check{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \quad (11)$$

$$\check{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]. \quad (12)$$

Estimating Radial Distortion by Alternation. As the radial distortion is expected to be small, one would expect to estimate the other five intrinsic parameters, using the technique described in Sect. 3.2, reasonable well by simply ignoring distortion. One strategy is then to estimate k_1 and k_2 after having estimated the other parameters, which will give us the ideal pixel coordinates (u, v) . Then, from (11) and (12), we have two equations for each point in each image:

$$\begin{bmatrix} (u-u_0)(x^2+y^2) & (u-u_0)(x^2+y^2)^2 \\ (v-v_0)(x^2+y^2) & (v-v_0)(x^2+y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \check{u}-u \\ \check{v}-v \end{bmatrix}.$$

Given m points in n images, we can stack all equations together to obtain in total $2mn$ equations, or in matrix form as $\mathbf{D}\mathbf{k} = \mathbf{d}$, where $\mathbf{k} = [k_1, k_2]^T$. The linear least-squares solution is given by

$$\mathbf{k} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{d}. \quad (13)$$

Once k_1 and k_2 are estimated, one can refine the estimate of the other parameters by solving (10) with $\hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, M_j)$ replaced by (11) and (12). We can alternate these two procedures until convergence.

Complete Maximum Likelihood Estimation. Experimentally, we found the convergence of the above alternation technique is slow. A natural extension to (10) is then to estimate the complete set of parameters by minimizing the following functional:

$$\sum_{i=1}^n \sum_{j=1}^m \|\mathbf{m}_{ij} - \check{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, M_j)\|^2, \quad (14)$$

*A typo was reported by Johannes Koester [johannes.koester@uni-dortmund.de] via email on Aug. 13, 2008.

where $\check{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, M_j)$ is the projection of point M_j in image i according to equation (2), followed by distortion according to (11) and (12). This is a nonlinear minimization problem, which is solved with the Levenberg-Marquardt Algorithm as implemented in `Minpack` [18]. A rotation is again parameterized by a 3-vector \mathbf{r} , as in Sect. 3.2. An initial guess of \mathbf{A} and $\{\mathbf{R}_i, \mathbf{t}_i | i = 1..n\}$ can be obtained using the technique described in Sect. 3.1 or in Sect. 3.2. An initial guess of k_1 and k_2 can be obtained with the technique described in the last paragraph, or simply by setting them to 0.

3.4 Summary

The recommended calibration procedure is as follows:

1. Print a pattern and attach it to a planar surface;
2. Take a few images of the model plane under different orientations by moving either the plane or the camera;
3. Detect the feature points in the images;
4. Estimate the five intrinsic parameters and all the extrinsic parameters using the closed-form solution as described in Sect. 3.1;
5. Estimate the coefficients of the radial distortion by solving the linear least-squares (13);
6. Refine all parameters by minimizing (14).

4 Degenerate Configurations

We study in this section configurations in which additional images do not provide more constraints on the camera intrinsic parameters. Because (3) and (4) are derived from the properties of the rotation matrix, if \mathbf{R}_2 is not independent of \mathbf{R}_1 , then image 2 does not provide additional constraints. In particular, if a plane undergoes a pure translation, then $\mathbf{R}_2 = \mathbf{R}_1$ and image 2 is not helpful for camera calibration. In the following, we consider a more complex configuration.

Proposition 1. *If the model plane at the second position is parallel to its first position, then the second homography does not provide additional constraints.*

Proof. Under our convention, \mathbf{R}_2 and \mathbf{R}_1 are related by a rotation around z -axis. That is,

$$\mathbf{R}_1 \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{R}_2 ,$$

where θ is the angle of the relative rotation. We will use superscript ⁽¹⁾ and ⁽²⁾ to denote vectors related to image 1 and 2, respectively. It is clear that we have

$$\begin{aligned} \mathbf{h}_1^{(2)} &= \lambda^{(2)}(\mathbf{A}\mathbf{r}^{(1)} \cos \theta + \mathbf{A}\mathbf{r}^{(2)} \sin \theta) = \frac{\lambda^{(2)}}{\lambda^{(1)}}(\mathbf{h}_1^{(1)} \cos \theta + \mathbf{h}_2^{(1)} \sin \theta) \\ \mathbf{h}_2^{(2)} &= \lambda^{(2)}(-\mathbf{A}\mathbf{r}^{(1)} \sin \theta + \mathbf{A}\mathbf{r}^{(2)} \cos \theta) = \frac{\lambda^{(2)}}{\lambda^{(1)}}(-\mathbf{h}_1^{(1)} \sin \theta + \mathbf{h}_2^{(1)} \cos \theta) . \end{aligned}$$

Then, the first constraint (3) from image 2 becomes:

$$\begin{aligned} \mathbf{h}_1^{(2)T} \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2^{(2)} &= \frac{\lambda^{(2)}}{\lambda^{(1)}} [(\cos^2 \theta - \sin^2 \theta)(\mathbf{h}_1^{(1)T} \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2^{(1)}) \\ &\quad - \cos \theta \sin \theta (\mathbf{h}_1^{(1)T} \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_1^{(1)} - \mathbf{h}_2^{(1)T} \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2^{(1)})] , \end{aligned}$$

which is a linear combination of the two constraints provided by \mathbf{H}_1 . Similarly, we can show that the second constraint from image 2 is also a linear combination of the two constraints provided by \mathbf{H}_1 . Therefore, we do not gain any constraint from \mathbf{H}_2 . \square

The result is self-evident because parallel planes intersect with the plane at infinity at the *same circular points*, and thus according to Sect. 2.4 they provide the same constraints.

In practice, it is very easy to avoid the degenerate configuration: we only need to change the orientation of the model plane from one snapshot to another.

Although the proposed technique will not work if the model plane undergoes pure translation, camera calibration is still possible if the translation is known. Please refer to Appendix D.

5 Experimental Results

The proposed algorithm has been tested on both computer simulated data and real data. The closed-form solution involves finding a singular value decomposition of a small $2n \times 6$ matrix, where n is the number of images. The nonlinear refinement within the Levenberg-Marquardt algorithm takes 3 to 5 iterations to converge.

5.1 Computer Simulations

The simulated camera has the following property: $\alpha = 1250$, $\beta = 900$, $\gamma = 1.09083$ (equivalent to 89.95°), $u_0 = 255$, $v_0 = 255$. The image resolution is 512×512 . The model plane is a checker pattern containing $10 \times 14 = 140$ corner points (so we usually have more data in the v direction than in the u direction). The size of pattern is $18\text{cm} \times 25\text{cm}$. The orientation of the plane is represented by a 3D vector \mathbf{r} , which is parallel to the rotation axis and whose magnitude is equal to the rotation angle. Its position is represented by a 3D vector \mathbf{t} (unit in centimeters).

Performance w.r.t. the noise level. In this experiment, we use three planes with $\mathbf{r}_1 = [20^\circ, 0, 0]^T$, $\mathbf{t}_1 = [-9, -12.5, 500]^T$, $\mathbf{r}_2 = [0, 20^\circ, 0]^T$, $\mathbf{t}_2 = [-9, -12.5, 510]^T$, $\mathbf{r}_3 = \frac{1}{\sqrt{5}}[-30^\circ, -30^\circ, -15^\circ]^T$, $\mathbf{t}_3 = [-10.5, -12.5, 525]^T$. Gaussian noise with 0 mean and σ standard deviation is added to the projected image points. The estimated camera parameters are then compared with the ground truth. We measure the relative error for α and β , and absolute error for u_0 and v_0 . We vary the noise level from 0.1 pixels to 1.5 pixels. For each noise level, we perform 100 independent trials, and the results shown are the average. As we can see from Fig. 1, errors increase linearly with the noise level. (The error for γ is not shown, but has the same property.) For $\sigma = 0.5$ (which is larger than the normal noise in practical calibration), the errors in α and β are less than 0.3%, and the errors in u_0 and v_0 are around 1 pixel. The error in u_0 is larger than that in v_0 . The main reason is that there are less data in the u direction than in the v direction, as we said before.

Performance w.r.t. the number of planes. This experiment investigates the performance with respect to the number of planes (more precisely, the number of images of the model plane). The orientation and position of the model plane for the first three images are the same as in the last subsection. From the fourth image, we first randomly choose a rotation axis in a uniform sphere, then apply a rotation angle of 30° . We vary the number of images from 2 to 16. For each number, 100 trials of independent plane orientations (except for the first three) and independent noise with mean 0 and

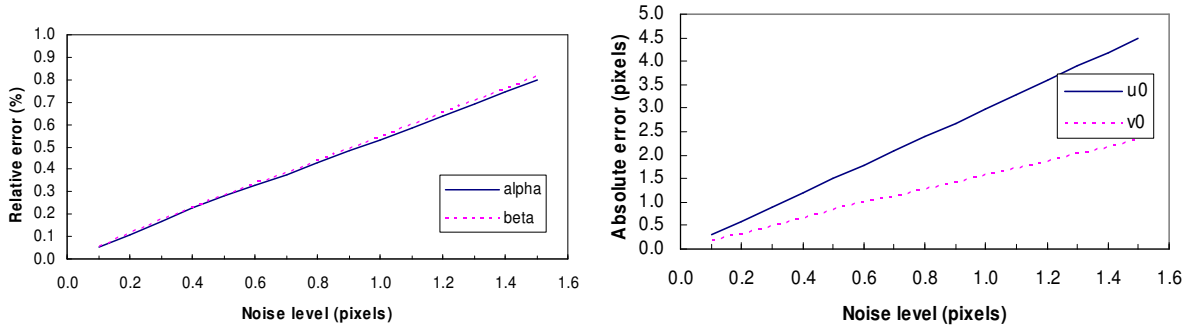


Figure 1: Errors vs. the noise level of the image points

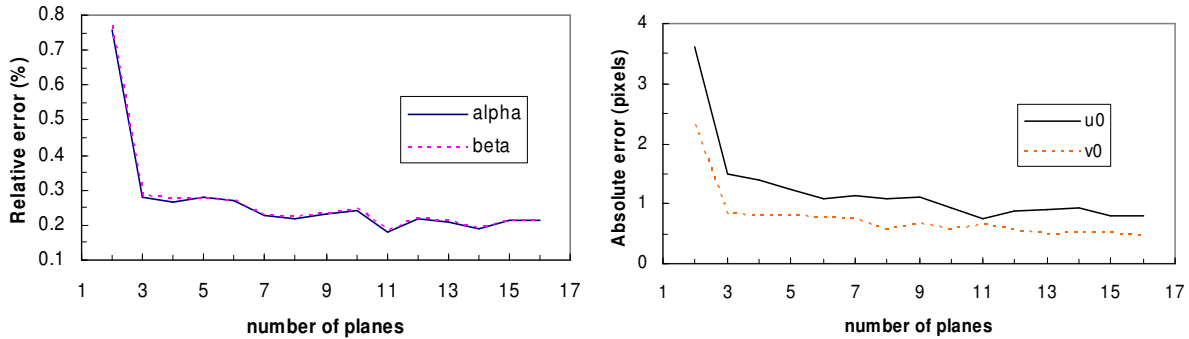


Figure 2: Errors vs. the number of images of the model plane

standard deviation 0.5 pixels are conducted. The average result is shown in Fig. 2. The errors decrease when more images are used. From 2 to 3, the errors decrease significantly.

Performance w.r.t. the orientation of the model plane. This experiment examines the influence of the orientation of the model plane with respect to the image plane. Three images are used. The orientation of the plane is chosen as follows: the plane is initially parallel to the image plane; a rotation axis is randomly chosen from a uniform sphere; the plane is then rotated around that axis with angle θ . Gaussian noise with mean 0 and standard deviation 0.5 pixels is added to the projected image points. We repeat this process 100 times and compute the average errors. The angle θ varies from 5° to 75° , and the result is shown in Fig. 3. When $\theta = 5^\circ$, 40% of the trials failed because the planes are almost parallel to each other (degenerate configuration), and the result shown has excluded those trials. Best performance seems to be achieved with an angle around 45° . Note that in practice, when the angle increases, foreshortening makes the corner detection less precise, but this is not considered in this experiment.

5.2 Real Data

The proposed technique is now routinely used in our vision group and also in the graphics group at Microsoft Research. Here, we provide the result with one example.

The camera to be calibrated is an off-the-shelf PULNiX CCD camera with 6 mm lens. The image resolution is 640×480 . The model plane contains a pattern of 8×8 squares, so there are 256 corners. The size of the pattern is $17\text{cm} \times 17\text{cm}$. It was printed with a high-quality printer and put on a glass.

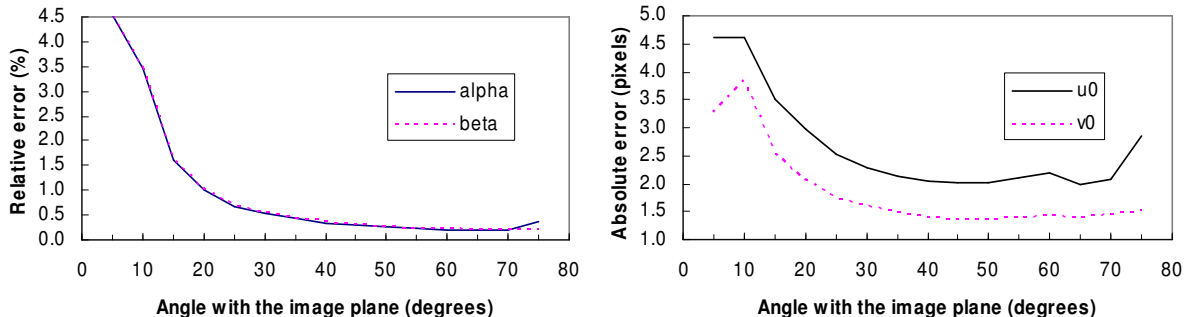


Figure 3: Errors vs. the angle of the model plane w.r.t. the image plane

Table 1: Results with real data of 2 through 5 images

nb	2 images			3 images			4 images			5 images		
	initial	final	σ	initial	final	σ	initial	final	σ	initial	final	σ
α	825.59	830.47	4.74	917.65	830.80	2.06	876.62	831.81	1.56	877.16	832.50	1.41
β	825.26	830.24	4.85	920.53	830.69	2.10	876.22	831.82	1.55	876.80	832.53	1.38
γ	0	0	0	2.2956	0.1676	0.109	0.0658	0.2867	0.095	0.1752	0.2045	0.078
u_0	295.79	307.03	1.37	277.09	305.77	1.45	301.31	304.53	0.86	301.04	303.96	0.71
v_0	217.69	206.55	0.93	223.36	206.42	1.00	220.06	206.79	0.78	220.41	206.59	0.66
k_1	0.161	-0.227	0.006	0.128	-0.229	0.006	0.145	-0.229	0.005	0.136	-0.228	0.003
k_2	-1.955	0.194	0.032	-1.986	0.196	0.034	-2.089	0.195	0.028	-2.042	0.190	0.025
RMS	0.761	0.295		0.987	0.393		0.927	0.361		0.881	0.335	

Five images of the plane under different orientations were taken, as shown in Fig. 4. We can observe a significant lens distortion in the images. The corners were detected as the intersection of straight lines fitted to each square.

We applied our calibration algorithm to the first 2, 3, 4 and all 5 images. The results are shown in Table 1. For each configuration, three columns are given. The first column (*initial*) is the estimation of the closed-form solution. The second column (*final*) is the maximum likelihood estimation (MLE), and the third column (σ) is the estimated standard deviation, representing the uncertainty of the final result. As is clear, the closed-form solution is reasonable, and the final estimates are very consistent with each other whether we use 2, 3, 4 or 5 images. We also note that the uncertainty of the final estimate decreases with the number of images. The last row of Table 1, indicated by *RMS*, displays the root of mean squared distances, in pixels, between detected image points and projected ones. The MLE improves considerably this measure.

The careful reader may remark the inconsistency for k_1 and k_2 between the closed-form solution and the MLE. The reason is that for the closed-form solution, camera intrinsic parameters are estimated assuming no distortion, and the predicted outer points lie closer to the image center than the detected ones. The subsequent distortion estimation tries to spread the outer points and increase the scale in order to reduce the distances, although the distortion shape (with positive k_1 , called pincushion distortion) does not correspond to the real distortion (with negative k_1 , called barrel distortion). The nonlinear refinement (MLE) finally recovers the correct distortion shape. The estimated distortion parameters allow us to correct the distortion in the original images. Figure 5 displays the first two such distortion-corrected images, which should be compared with the first two images shown in Figure 4. We see clearly that the curved pattern in the original images is straightened.

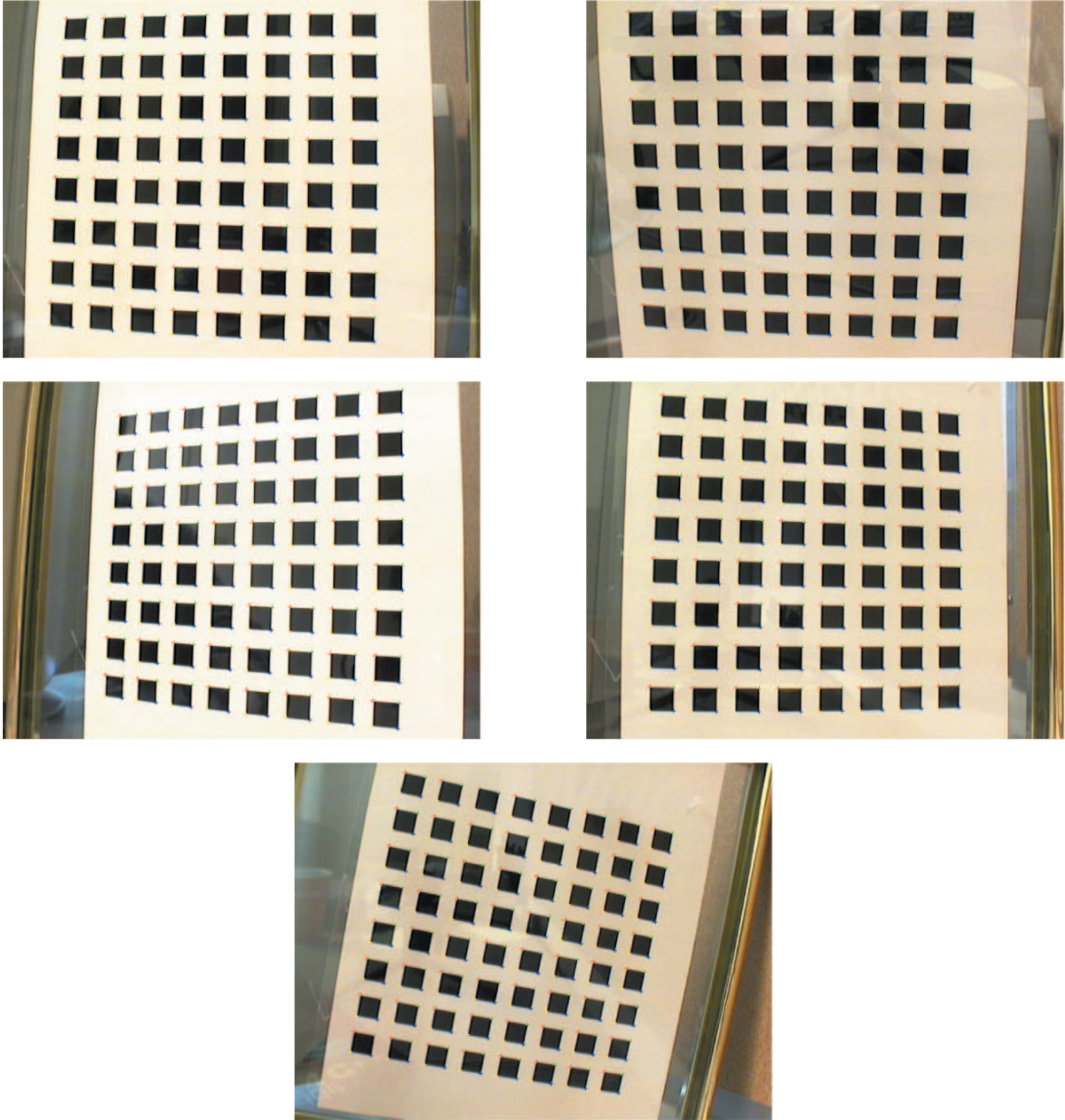


Figure 4: Five images of a model plane, together with the extracted corners (indicated by cross)

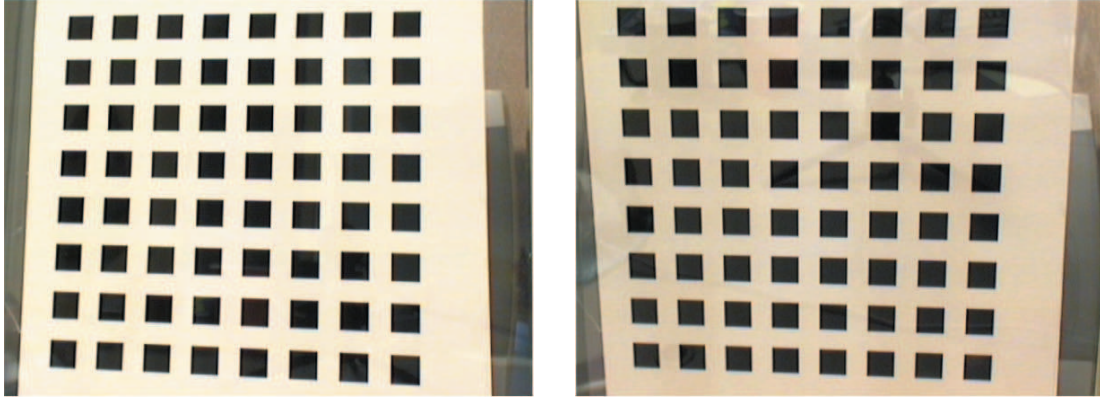


Figure 5: First and second images after having corrected radial distortion

Table 2: Variation of the calibration results among all quadruples of images

quadruple	(1234)	(1235)	(1245)	(1345)	(2345)	mean	deviation
α	831.81	832.09	837.53	829.69	833.14	832.85	2.90
β	831.82	832.10	837.53	829.91	833.11	832.90	2.84
γ	0.2867	0.1069	0.0611	0.1363	0.1096	0.1401	0.086
u_0	304.53	304.32	304.57	303.95	303.53	304.18	0.44
v_0	206.79	206.23	207.30	207.16	206.33	206.76	0.48
k_1	-0.229	-0.228	-0.230	-0.227	-0.229	-0.229	0.001
k_2	0.195	0.191	0.193	0.179	0.190	0.190	0.006
RMS	0.361	0.357	0.262	0.358	0.334	0.334	0.04

Variation of the calibration result. In Table 1, we have shown the calibration results with 2 through 5 images, and we have found that the results are very consistent with each other. In order to further investigate the stability of the proposed algorithm, we have applied it to all combinations of 4 images from the available 5 images. The results are shown in Table 2, where the third column (1235), for example, displays the result with the quadruple of the first, second, third, and fifth image. The last two columns display the mean and sample deviation of the five sets of results. The sample deviations for all parameters are quite small, which implies that the proposed algorithm is quite stable. The value of the skew parameter γ is not significant from 0, since the coefficient of variation, $0.086/0.1401 = 0.6$, is large. Indeed, $\gamma = 0.1401$ with $\alpha = 832.85$ corresponds to 89.99 degrees, very close to 90 degrees, for the angle between the two image axes. We have also computed the aspect ratio α/β for each quadruple. The mean of the aspect ratio is equal to 0.99995 with sample deviation 0.00012. It is therefore very close to 1, i.e., the pixels are square.

Application to image-based modeling. Two images of a tea tin (see Fig. 6) were taken by the same camera as used above for calibration. Mainly two sides are visible. We manually picked 8 point matches on each side, and the structure-from-motion software we developed earlier [27] was run on these 16 point matches to build a partial model of the tea tin. The reconstructed model is in VRML, and three rendered views are shown in Fig. 7. The reconstructed points on each side are indeed coplanar, and we computed the angle between the two reconstructed planes which is 94.7° . Although we do not

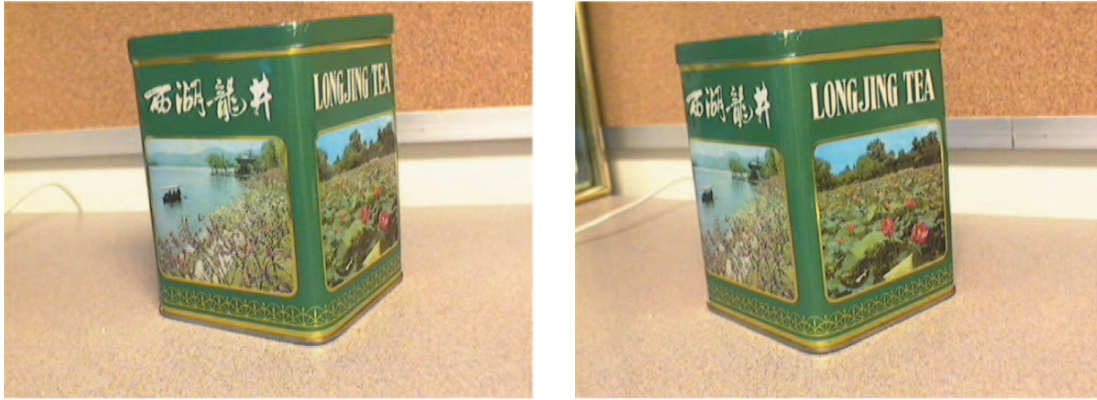


Figure 6: Two images of a tea tin



Figure 7: Three rendered views of the reconstructed tea tin

have the ground truth, but the two sides of the tea tin are indeed almost orthogonal to each other.

All the real data and results are available from the following Web page:

<http://research.microsoft.com/~zhang/Calib/>

5.3 Sensitivity with Respect to Model Imprecision

In the example described above, the 2D model pattern was printed on a paper with a high-quality printer. Although it is significantly cheaper to make such a high-quality 2D pattern than the classical calibration equipment, it is possible that there is some imprecision on the 2D model pattern if we print it on a normal printer, or the pattern is not on a flat surface. This section investigates the sensitivity of the proposed calibration technique with respect to model imprecision.

5.3.1 Random noise in the model points

We conducted this experiment on the same real data as in the last subsection. All five real images were used. To simulate model imprecision, we added Gaussian noise with zero mean to the corners of each square in the model. The standard deviation of the added noise varies from 1% to 15% of the side of each square, which is equal to 1.27cm (more precisely, 0.5inches). 15% corresponds to a standard deviation of 2mm, and people may not want to use such a poor model. For each noise level, 100 trials were conducted, and average errors (deviations from the results obtained with the true model as

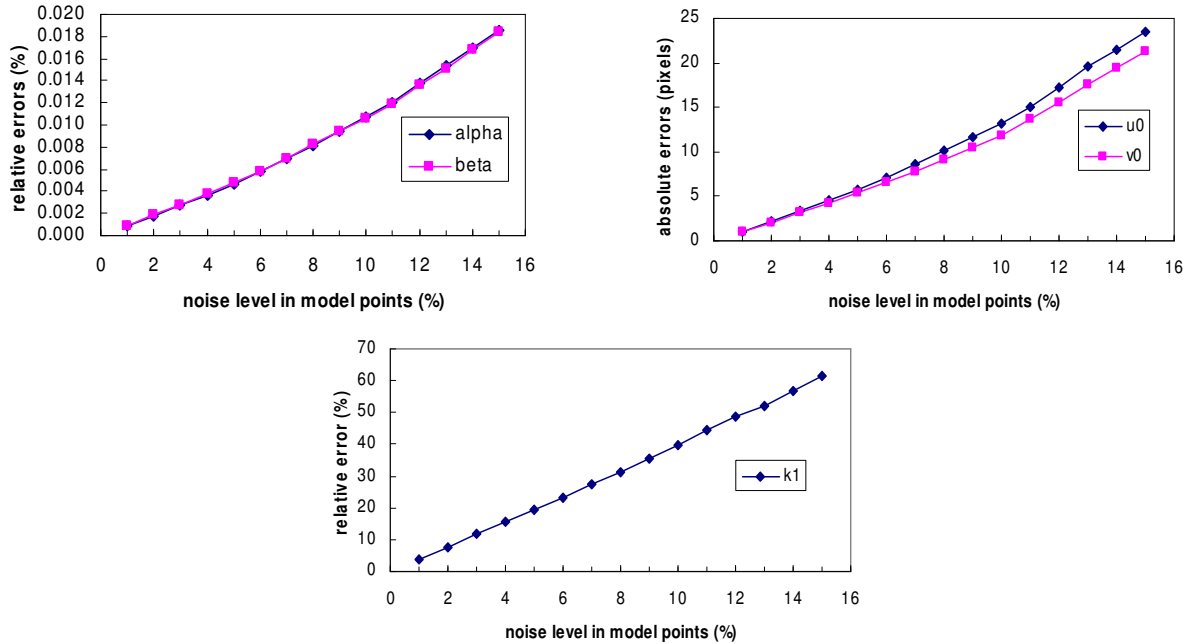


Figure 8: Sensitivity of camera calibration with respect to Gaussian noise in the model points

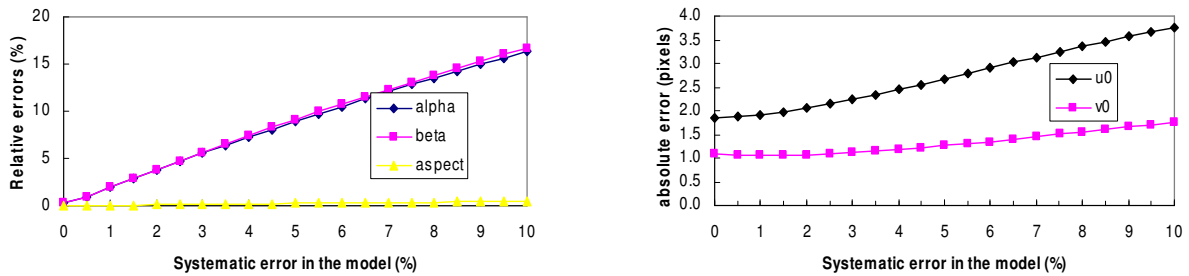


Figure 9: Sensitivity of camera calibration with respect to systematic spherical non-planarity

shown in Table 1) were calculated, and are depicted in Fig. 8. Obviously, all errors increase with the level of noise added to the model points. The pixel scale factors (α and β) remain very stable: the error is less than 0.02%. The coordinates of the principal point are quite stable: the errors are about 20 pixels for the noise level 15%. The estimated radial distortion coefficient k_1 becomes less useful, and the second term k_2 (not shown) is even less than k_1 .

In our current formulation, we assume that the exact position of the points in the model plane is known. If the model points are only known within certain precision, we can reformulate the problem, and we could expect smaller errors than reported here.

5.3.2 Systematic non-planarity of the model pattern

In this section, we consider systematic non-planarity of the model pattern, e.g., when a printed pattern is attached to a soft book cover. We used the same configuration as in Sect. 5.1. The model plane

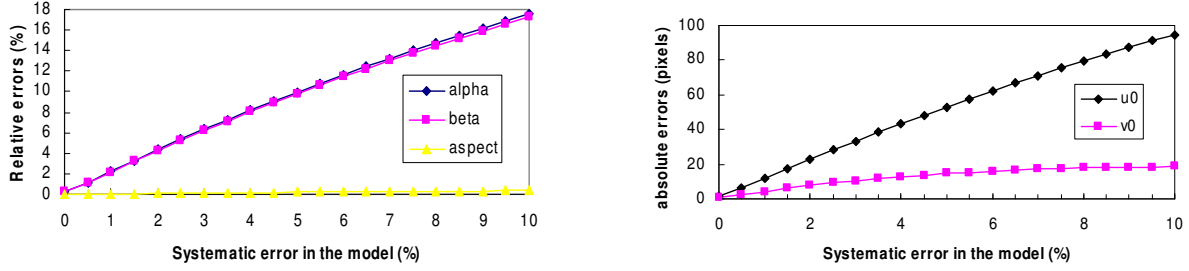


Figure 10: Sensitivity of camera calibration with respect to systematic cylindrical non-planarity

was distorted in two systematic ways to simulate the non-planarity: spherical and cylindrical. With spherical distortion, points away from the center of the pattern are displaced in z according to $z = p\sqrt{x^2 + y^2}$, where p indicates the non-planarity (the model points are coplanar when $p = 0$). The displacement is symmetric around the center. With Cylindrical distortion, points are displaced in z according to $z = px$. Again, p indicates the non-planarity. This simulates bending of the model pattern around the vertical axis. Four images of the model pattern were used: the first is parallel to the image plane; the second is rotated from the first around the horizontal axis by 30 degrees; the third is rotated from the first around the vertical axis by 30 degrees; the fourth is rotated from the first around the diagonal axis by 30 degrees. Although model points are not coplanar, they were treated as coplanar, and the proposed calibration technique was applied. Gaussian noise with standard deviation 0.5 pixels was added to the image points, and 100 independent trials were conducted. The average calibration errors of the 100 trials are shown in Fig. 9 for spherical non-planarity and in Fig. 10 for cylindrical non-planarity. The horizontal axis indicates the increase in the non-planarity, which is measured as the ratio of the maximum z displacement to the size of the pattern. Therefore, 10% of non-planarity is equivalent to maximum 2.5cm of displacement in z , which does not likely happen in practice. Several observations can be made:

- Systematic non-planarity of the model has more effect on the calibration precision than random errors in the positions as described in the last subsection;
- Aspect ratio is very stable (0.4% of error for 10% of non-planarity);
- Systematic cylindrical non-planarity is worse than systematic spherical non-planarity, especially for the coordinates of the principal point (u_0, v_0) . The reason is that cylindrical non-planarity is only symmetric in one axis. That is also why the error in u_0 is much larger than in v_0 in our simulation;
- The result seems still usable in practice if there is only a few percents (say, less than 3%) of systematic non-planarity.

The error in (u_0, v_0) has been found by many researchers to have little effect in 3D reconstruction. As pointed out by Triggs in [22], the absolute error in (u_0, v_0) is not geometrically meaningful. He proposes to measure the relative error with respect to the focal length, i.e., $\Delta u_0/\alpha$ and $\Delta v_0/\alpha$. This is equivalent to measuring the angle between the true optical axis and the estimated one. Then, for 10% of cylindrical non-planarity (see Fig. 10), the relative error for u_0 is 7.6%, comparable with those of α and β .

6 Conclusion

In this paper, we have developed a flexible new technique to easily calibrate a camera. The technique only requires the camera to observe a planar pattern from a few (at least two) different orientations. We can move either the camera or the planar pattern. The motion does not need to be known. Radial lens distortion is modeled. The proposed procedure consists of a closed-form solution, followed by a nonlinear refinement based on maximum likelihood criterion. Both computer simulation and real data have been used to test the proposed technique, and very good results have been obtained. Compared with classical techniques which use expensive equipment such as two or three orthogonal planes, the proposed technique gains considerable flexibility.

Acknowledgment

Thanks go to Brian Guenter for his software of corner extraction and for many discussions, and to Bill Triggs for insightful comments. Thanks go to Andrew Zisserman for bringing his CVPR98 work [14] to my attention, which uses the same constraint but in different form, and for pointing out an error in my discussion on the case of pure translation. Thanks go to Bill Triggs and Gideon Stein for suggesting experiments described in Sect. 5.3. Thanks also go to the members of the Vision Group at MSR for encouragement and discussions. Anandan and Charles Loop have checked the English.

A Estimation of the Homography Between the Model Plane and its Image

There are many ways to estimate the homography between the model plane and its image. Here, we present a technique based on maximum likelihood criterion. Let M_i and \mathbf{m}_i be the model and image points, respectively. Ideally, they should satisfy (2). In practice, they don't because of noise in the extracted image points. Let's assume that \mathbf{m}_i is corrupted by Gaussian noise with mean $\mathbf{0}$ and covariance matrix $\Lambda_{\mathbf{m}_i}$. Then, the maximum likelihood estimation of \mathbf{H} is obtained by minimizing the following functional

$$\sum_i (\mathbf{m}_i - \hat{\mathbf{m}}_i)^T \Lambda_{\mathbf{m}_i}^{-1} (\mathbf{m}_i - \hat{\mathbf{m}}_i),$$

where
$$\hat{\mathbf{m}}_i = \frac{1}{\bar{\mathbf{h}}_3^T M_i} \begin{bmatrix} \bar{\mathbf{h}}_1^T M_i \\ \bar{\mathbf{h}}_2^T M_i \end{bmatrix} \quad \text{with } \bar{\mathbf{h}}_i, \text{ the } i^{\text{th}} \text{ row of } \mathbf{H}.$$

In practice, we simply assume $\Lambda_{\mathbf{m}_i} = \sigma^2 \mathbf{I}$ for all i . This is reasonable if points are extracted independently with the same procedure. In this case, the above problem becomes a nonlinear least-squares one, i.e., $\min_{\mathbf{H}} \sum_i \|\mathbf{m}_i - \hat{\mathbf{m}}_i\|^2$. The nonlinear minimization is conducted with the Levenberg-Marquardt Algorithm as implemented in `Minpack` [18]. This requires an initial guess, which can be obtained as follows.

Let $\mathbf{x} = [\bar{\mathbf{h}}_1^T, \bar{\mathbf{h}}_2^T, \bar{\mathbf{h}}_3^T]^T$. Then equation (2) can be rewritten as

$$\begin{bmatrix} \tilde{\mathbf{M}}^T & \mathbf{0}^T & -u\tilde{\mathbf{M}}^T \\ \mathbf{0}^T & \tilde{\mathbf{M}}^T & -v\tilde{\mathbf{M}}^T \end{bmatrix} \mathbf{x} = \mathbf{0}.$$

When we are given n points, we have n above equations, which can be written in matrix equation as $\mathbf{Lx} = \mathbf{0}$, where \mathbf{L} is a $2n \times 9$ matrix. As \mathbf{x} is defined up to a scale factor, the solution is well known

to be the right singular vector of \mathbf{L} associated with the smallest singular value (or equivalently, the eigenvector of $\mathbf{L}^T\mathbf{L}$ associated with the smallest eigenvalue).

In \mathbf{L} , some elements are constant 1, some are in pixels, some are in world coordinates, and some are multiplication of both. This makes \mathbf{L} poorly conditioned numerically. Much better results can be obtained by performing a simple data normalization, such as the one proposed in [12], prior to running the above procedure.

B Extraction of the Intrinsic Parameters from Matrix \mathbf{B}

Matrix \mathbf{B} , as described in Sect. 3.1, is estimated up to a scale factor, i.e., $\mathbf{B} = \lambda\mathbf{A}^{-T}\mathbf{A}$ with λ an arbitrary scale. Without difficulty[†], we can uniquely extract the intrinsic parameters from matrix \mathbf{B} .

$$\begin{aligned} v_0 &= (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{22} - B_{12}^2) \\ \lambda &= B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})] / B_{11} \\ \alpha &= \sqrt{\lambda / B_{11}} \\ \beta &= \sqrt{\lambda B_{11} / (B_{11}B_{22} - B_{12}^2)} \\ \gamma &= -B_{12}\alpha^2\beta / \lambda \\ u_0 &= \gamma v_0 / \beta - B_{13}\alpha^2 / \lambda. \end{aligned}$$

C Approximating a 3×3 matrix by a Rotation Matrix

The problem considered in this section is to solve the best rotation matrix \mathbf{R} to approximate a given 3×3 matrix \mathbf{Q} . Here, “best” is in the sense of the smallest Frobenius norm of the difference $\mathbf{R} - \mathbf{Q}$. That is, we are solving the following problem:

$$\min_{\mathbf{R}} \|\mathbf{R} - \mathbf{Q}\|_F^2 \quad \text{subject to } \mathbf{R}^T\mathbf{R} = \mathbf{I}. \quad (15)$$

Since

$$\begin{aligned} \|\mathbf{R} - \mathbf{Q}\|_F^2 &= \text{trace}((\mathbf{R} - \mathbf{Q})^T(\mathbf{R} - \mathbf{Q})) \\ &= 3 + \text{trace}(\mathbf{Q}^T\mathbf{Q}) - 2\text{trace}(\mathbf{R}^T\mathbf{Q}), \end{aligned}$$

problem (15) is equivalent to the one of maximizing $\text{trace}(\mathbf{R}^T\mathbf{Q})$.

Let the singular value decomposition of \mathbf{Q} be \mathbf{USV}^T , where $\mathbf{S} = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$. If we define an orthogonal matrix \mathbf{Z} by $\mathbf{Z} = \mathbf{V}^T\mathbf{R}^T\mathbf{U}$, then

$$\begin{aligned} \text{trace}(\mathbf{R}^T\mathbf{Q}) &= \text{trace}(\mathbf{R}^T\mathbf{USV}^T) = \text{trace}(\mathbf{V}^T\mathbf{R}^T\mathbf{US}) \\ &= \text{trace}(\mathbf{ZS}) = \sum_{i=1}^3 z_{ii}\sigma_i \leq \sum_{i=1}^3 \sigma_i. \end{aligned}$$

It is clear that the maximum is achieved by setting $\mathbf{R} = \mathbf{UV}^T$ because then $\mathbf{Z} = \mathbf{I}$. This gives the solution to (15).

An excellent reference on matrix computations is the one by Golub and van Loan [10].

[†]A typo was reported in formula u_0 by Jiyong Ma [mailto:jiyong@cslr.Colorado.EDU] via an email on April 18, 2002.

D Camera Calibration Under Known Pure Translation

As said in Sect. 4, if the model plane undergoes a pure translation, the technique proposed in this paper will not work. However, camera calibration is possible if the translation is known like the setup in Tsai's technique [23]. From (2), we have $\mathbf{t} = \alpha \mathbf{A}^{-1} \mathbf{h}_3$, where $\alpha = 1/\|\mathbf{A}^{-1} \mathbf{h}_1\|$. The translation between two positions i and j is then given by

$$\mathbf{t}^{(ij)} = \mathbf{t}^{(i)} - \mathbf{t}^{(j)} = \mathbf{A}^{-1}(\alpha^{(i)} \mathbf{h}_3^{(i)} - \alpha^{(j)} \mathbf{h}_3^{(j)}) .$$

(Note that although both $\mathbf{H}^{(i)}$ and $\mathbf{H}^{(j)}$ are estimated up to their own scale factors, they can be rescaled up to a single common scale factor using the fact that it is a pure translation.) If only the translation direction is known, we get two constraints on \mathbf{A} . If we know additionally the translation magnitude, then we have another constraint on \mathbf{A} . Full calibration is then possible from two planes.

References

- [1] S. Bougnoux. From projective to euclidean space under any practical situation, a criticism of self-calibration. In *Proceedings of the 6th International Conference on Computer Vision*, pages 790–796, Jan. 1998.
- [2] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.
- [3] B. Caprile and V. Torre. Using Vanishing Points for Camera Calibration. *The International Journal of Computer Vision*, 4(2):127–140, Mar. 1990.
- [4] W. Faig. Calibration of close-range photogrammetry systems: Mathematical formulation. *Photogrammetric Engineering and Remote Sensing*, 41(12):1479–1486, 1975.
- [5] O. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.
- [6] O. Faugeras, T. Luong, and S. Maybank. Camera self-calibration: theory and experiments. In G. Sandini, editor, *Proc 2nd ECCV*, volume 588 of *Lecture Notes in Computer Science*, pages 321–334, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.
- [7] O. Faugeras and G. Toscani. The calibration problem for stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 15–20, Miami Beach, FL, June 1986. IEEE.
- [8] S. Ganapathy. Decomposition of transformation matrices for robot vision. *Pattern Recognition Letters*, 2:401–412, Dec. 1984.
- [9] D. Gennery. Stereo-camera calibration. In *Proceedings of the 10th Image Understanding Workshop*, pages 101–108, 1979.
- [10] G. Golub and C. van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, Maryland, 3 edition, 1996.
- [11] R. Hartley. Self-calibration from multiple views with a rotating camera. In J.-O. Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision*, volume 800-801 of *Lecture Notes in Computer Science*, pages 471–478, Stockholm, Sweden, May 1994. Springer-Verlag.
- [12] R. Hartley. In defence of the 8-point algorithm. In *Proceedings of the 5th International Conference on Computer Vision*, pages 1064–1070, Boston, MA, June 1995. IEEE Computer Society Press.
- [13] R. I. Hartley. An algorithm for self calibration from several views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 908–912, Seattle, WA, June 1994. IEEE.
- [14] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 482–488, Santa Barbara, California, June 1998. IEEE Computer Society.

- [15] Q.-T. Luong. *Matrice Fondamentale et Calibration Visuelle sur l'Environnement-Vers une plus grande autonomie des systèmes robotiques*. PhD thesis, Université de Paris-Sud, Centre d'Orsay, Dec. 1992.
- [16] Q.-T. Luong and O. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *The International Journal of Computer Vision*, 22(3):261–289, 1997.
- [17] S. J. Maybank and O. D. Faugeras. A theory of self-calibration of a moving camera. *The International Journal of Computer Vision*, 8(2):123–152, Aug. 1992.
- [18] J. More. The levenberg-marquardt algorithm, implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, Lecture Notes in Mathematics 630. Springer-Verlag, 1977.
- [19] I. Shimizu, Z. Zhang, S. Akamatsu, and K. Deguchi. Head pose determination from one image using a generic model. In *Proceedings of the IEEE Third International Conference on Automatic Face and Gesture Recognition*, pages 100–105, Nara, Japan, Apr. 1998.
- [20] C. C. Slama, editor. *Manual of Photogrammetry*. American Society of Photogrammetry, fourth edition, 1980.
- [21] G. Stein. Accurate internal camera calibration using rotation, with analysis of sources of error. In *Proc. Fifth International Conference on Computer Vision*, pages 230–236, Cambridge, Massachusetts, June 1995.
- [22] B. Triggs. Autocalibration from planar scenes. In *Proceedings of the 5th European Conference on Computer Vision*, pages 89–105, Freiburg, Germany, June 1998.
- [23] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, Aug. 1987.
- [24] G. Wei and S. Ma. A complete two-plane camera calibration method and experimental comparisons. In *Proc. Fourth International Conference on Computer Vision*, pages 439–446, Berlin, May 1993.
- [25] G. Wei and S. Ma. Implicit and explicit camera calibration: Theory and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):469–480, 1994.
- [26] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, Oct. 1992.
- [27] Z. Zhang. Motion and structure from two perspective views: From essential parameters to euclidean motion via fundamental matrix. *Journal of the Optical Society of America A*, 14(11):2938–2950, 1997.