

Autocalibrazione distribuita di videocamere fisse con vincoli di comunicazione dove una frazione di telecamere è calibrata

Francesca Bristot 607373
Marco Pattarello 605600
Klaus Schmiedhofer 607369

Progettazione di sistemi di controllo -09/10-

16 febbraio 2010

Indice

1	Introduzione	4
2	Teoria di base	5
3	Stato dell'arte	8
4	Algoritmo proposto	9
5	Calcolo matrice F	12
6	Fattorizzazione della matrice essenziale	15
7	Calcolo fattore di scala	17
8	Calcolo dell'altezza	20
9	Calibrazione Stereo	21
10	Simulazioni	24
11	Ricalibrazione	28
12	Considerazioni PTZ	32
13	Conclusioni e sviluppi futuri	34
A	Calcolo della matrice fondamentale	35
B	Funzione di costo	38
C	Calcolo Omografia	40
D	Dimostrazione H	42
E	Trasformata di Hough	44

Sommario

Lo scopo di questo progetto è la calibrazione di tutte le videocamere appartenenti ad una rete di videosorveglianza, sapendo che un numero $n \ll N$ (con N numero totale di agenti) di videocamere è già calibrato.

Il metodo che si è deciso di usare sfrutta la matrice fondamentale tra due videocamere per ricavarne la posizione relativa (passando per la matrice essenziale). Utilizzando le telecamere già calibrate si riesce a trovare la posizione globale di tutti gli agenti all'interno della rete.

Per diminuire l'errore sulla F si sono utilizzati algoritmi per minimizzare una funzione di costo a partire da una soluzione iniziale, questo permette una maggiore precisione.

Siccome la matrice essenziale è definita a meno di un fattore di scala abbiamo studiato dei metodi per il calcolo della baseline in modo da fissare tale fattore. Per diminuire ulteriormente l'errore si utilizza un grafo pesato per rappresentare la rete e un algoritmo per trovare il cammino minimo per ricalibrarla.

1 Introduzione

Il nostro progetto riguarda la calibrazione di una rete di videocamere di sorveglianza, noti i parametri intrinseci lo scopo è calcolare quanto più esattamente possibile la posizione delle camere. Calibrare manualmente ogni singola videocamera è un lavoro estremamente laborioso e richiede molto tempo se la rete è composta da molti agenti, quindi si vuole trovare un metodo che tramite lo studio di una persona in movimento sulla scena permetta la calibrazione della rete. I dati a nostra disposizione sono i parametri intrinseci di tutte le videocamere, la posizione esatta di un numero limitato di videocamere e i frame di ogni videocamera (inizialmente studiamo il caso centralizzato).

Abbiamo adottato un approccio che sfrutta la presenza di più videocamere, quindi siamo passati attraverso la matrice fondamentale e quella essenziale (queste matrici sono delle mappe tra punti su frame in 2 diverse videocamere che corrispondono ad uno stesso oggetto nel mondo reale). Il problema che si presenta nell'uso di questi strumenti è la presenza di errori cospicui derivanti dal calcolo per via analitica e dall'imprecisione (inevitabile) con cui è possibile ricavare i punti da un frame. Per risolverli è opportuno calcolare una soluzione iniziale e poi raffinarla tramite la minimizzazione di una funzione di costo. Tale metodo viene descritto ad esempio in (1).

Un altro problema che si è cercato di risolvere è stato il calcolo del fattore di scala a meno del quale sono definite le matrici F ed E , che si può fissare calcolando la baseline. Abbiamo quindi cercato un algoritmo per calcolarla, considerato il fatto che la letteratura in proposito è decisamente scarna.

Per un ulteriore miglioramento abbiamo scomposto la calibrazione in due fasi, una preliminare per raccogliere i dati che ci servono (baseline e posizione iniziale) e una seconda fase in cui raffiniamo la stima.

2 Teoria di base

Prima di affrontare il problema abbiamo dovuto documentarci su come potessero venire modellate le telecamere, così da avere un modello matematico con il quale lavorare.

Modello telecamera

Le telecamere utilizzate, dalla ditta Videotec, sono PTZ (Pan-Tilt-Zoom) e vengono installate da un operatore che essendo sprovvisto di GPS non può determinarne il loro posizionamento 'globale'.

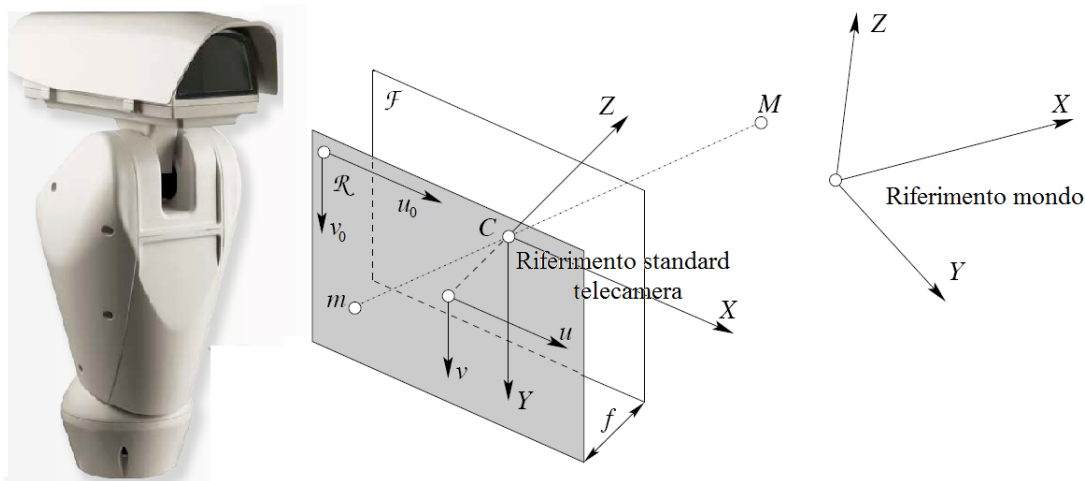


Figura 1: Modello telecamera

Il modello della telecamera utilizzato è quello generale, *stenopeico* (pinhole), che tiene conto di dove un punto 3D nella realtà venga proiettato nel flusso video 2D.

Il modello consiste in un piano retina (o immagine) \mathcal{R} e di un punto C , centro ottico (o centro di proiezione) distante f (lunghezza focale) dal piano. La retta passante per C ortogonale a \mathcal{R} è l'asse ottico (asse Z nella figura) e la sua intersezione con \mathcal{R} prende il nome di punto principale. Il piano \mathcal{F} parallelo ad \mathcal{R} e contenente il centro ottico prende il nome di *piano focale*. I punti di tale piano si proiettano all'infinito sul piano immagine.

Si mettono in evidenza i riferimenti utilizzati: per il piano immagine si utilizza il riferimento u_0, v_0 tipico del linguaggio di programmazione Matlab, mentre il riferimento standard della telecamera è destrorso (X, Y, Z) e ha come origine il centro ottico della telecamera. Infine il sistema di riferimento mondo, che può essere preso a piacere, è necessario per stabilire dove sia posizionata la telecamera rispetto ad esso.

Per descrivere analiticamente la proiezione prospettica operata dalla telecamera dobbiamo introdurre opportune coordinate per esprimere punti dello spazio 3D, in punti dello spazio 2D affinché la proiezione prospettica risulti lineare. Tali coordinate si chiamano coordinate omogenee e mappano un punto M nel sistema di riferimento mondo, in un punto m nel sistema di riferimento immagine (vedi figura 1).

$$\mathbf{M} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad \mathbf{m} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (1)$$

Dove è stata inserita la componente 1 per evitare di rappresentare i punti all'infinito, cosa inevitabile se la componente fosse stata generica. Ipotizziamo ora che il sistema di riferimento mondo coincida con il sistema di riferimento della telecamera, risulta allora evidente che un punto 3D nel sistema di riferimento mondo viene mappato in questo modo nel piano immagine:

$$Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -fk_u & 0 & u_0 & 0 \\ 0 & -fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2)$$

Quindi si può dire che $Z\mathbf{m} = P\mathbf{M}$ dove P viene chiamata *matrice di proiezione prospettica*. Verrà utilizzata anche la notazione $\mathbf{m} \approx P\mathbf{M}$ dove \approx significa uguale a meno di un fattore di scala.

Parametri intrinseci

La matrice P può essere scomposta in questo modo $P = K[I|0]$ dove K è la matrice dei parametri intrinseci.

$$K = \begin{bmatrix} -fk_u & 0 & u_0 \\ 0 & -fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

I parametri intrinseci sono quei parametri interni alla telecamera che non dipendono da come questa sia posizionata, questi sono:

- f è la focale tiene conto dello Zoom della telecamera
- k_u e k_v tengono conto dell'inverso della dimensione efficace del pixel (pixel*metri⁻¹)
- u_0 e v_0 tengono conto dello spostamento del sistema di riferimento immagine rispetto al punto principale

Parametri estrinseci

I parametri estrinseci, invece, variano variando la posizione della telecamera, essi descrivono il cambiamento di coordinate per passare dal sistema di riferimento mondo al sistema di riferimento standard della telecamera.

Il cambiamento di coordinate tra questi due sistemi di riferimento è sempre una rototraslazione. Se M_c è un punto nel sistema di riferimento standard della telecamera mentre M è sempre lo stesso punto rappresentato nel sistema di riferimento mondo allora il loro legame è $M_c = GM$:

$$M_c = \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix} M \quad (4)$$

Dove $R \in \mathbb{R}^{3 \times 3}$ è la matrice di rotazione e $\mathbf{t} = [t_1 \ t_2 \ t_3]^T$ la matrice di traslazione dal riferimento mondo a quello della telecamera. Si nota quindi che i parametri estrinseci sono essenziali al fine di capire la posizione della telecamera rispetto al riferimento mondo.

La matrice di proiezione può essere dunque scomposta in $P = K[R | \mathbf{t}]$ pertanto, una volta trovata la matrice di proiezione con un qualche algoritmo, si possono ricavare le matrici R e \mathbf{t} e quindi la posizione e l'orientazione della telecamera rispetto al sistema di riferimento mondo.

3 Stato dell'arte

Gli algoritmi che si trovano in letteratura per la calibrazione di videocamere riguardano per la maggior parte due tipi di approcci: uno detto *camera calibration* e l'altro *camera autocalibration*.

Il primo approccio richiede generalmente una quantità di informazioni geometriche riguardo all'oggetto sulla scena. Si usano oggetti 3D che consistono in pattern 2D (ad esempio una scacchiera con dimensioni dei quadratini note) disposti su piani ortogonali tra loro. Questo metodo è stato proposto con l'uso per l'appunto di una scacchiera da Zhang nel 2000.

Un altro approccio è il *camera autocalibration*, in cui si sfrutta il movimento della videocamera e il calcolo di punti e linee all'infinito, *vanishing point/line*.

Discutiamo dei mezzi a disposizione quando i parametri interni sono conosciuti. Disponendo di una coppia di videocamere e di una serie di punti corrispondenti nei frame di entrambe è possibile ricavare la matrice fondamentale, da questa se si conoscono i parametri interni si può ricavare la matrice essenziale che una volta fattorizzata dà rotazione e traslazione tra le camere *a meno di un fattore di scala*. Esistono numerose implementazioni per il calcolo della matrice F dati almeno 8 punti di corrispondenza nei frame, questa matrice può essere usata come stima iniziale e necessita di essere migliorata con metodi di minimizzazione di costo. Al contrario raramente si trovano articoli in cui si parla di come va fissato il fattore di scala della matrice essenziale.

Si possono trovare in letteratura anche articoli che trattano la calibrazione (anche distribuita) di una rete e cercano di sfruttare metodi probabilistici (2) per fare una stima della posizione coerente con le osservazioni della scena, ma questo approccio è sconsigliabile in quanto l'implementazione per una rete reale presenta grosse problematiche, bisogna inoltre gestire difficili non linearità che rischiano di portare a grossi errori nel risultato finale.

Il contributo da noi apportato consisterà quindi a grandi linee nell'applicare metodi noti per la calibrazione stereo ad un'intera rete, nella ricerca di un metodo per fissare il fattore di scala a meno del quale viene definita la matrice essenziale, nell'implementazione di algoritmi per il raffinamento della soluzione iniziale trovata per la matrice F.

4 Algoritmo proposto

In questa sezione verrà esposto l'algoritmo proposto per calibrare l'intera rete di videosorveglianza in modo generale, nelle sezioni successive descriveremo in dettaglio le varie parti. L'idea di base è di calcolare la posizione relativa fra le camere a due a due sfruttando il celebre algoritmo degli 8 punti per il calcolo della matrice fondamentale. Fatto questo si sfrutta l'informazione data sulle videocamere già calibrate per tarare l'intera rete sul sistema di riferimento desiderato.

Il problema principale legato all'uso di questo metodo è l'inevitabile imprecisione legata al metodo degli 8 punti. Questo metodo infatti richiede di trovare minimo 8 punti di corrispondenza in 2 diversi frame (su due diverse videocamere) che riprendono una stessa scena. Per trovare i punti è necessario usare algoritmi che individuino una particolare forma nei frame, e se le corrispondenze non sono estremamente precise si generano degli errori. Ulteriori errori derivano dal calcolo analitico della matrice.

Il modo in cui solitamente si procede è quindi il calcolo di una stima iniziale di F con l'algoritmo sopra citato, tale stima viene poi perfezionata con algoritmi che iterativamente cercano di minimizzare una funzione di costo fino a giungere ad un valore ottimo della matrice. Un algoritmo che opera questa minimizzazione è stato ad esempio proposto da Horn, ma anche Hartley e Zisserman ne parlano in (1). Vediamo ora in che modo abbiamo deciso di procedere.

Nel prossimo paragrafo descriveremo i dati iniziali utilizzati per questo progetto, analizzando e dimostrando le scelte adottate. Utilizzeremo anche la trasformata di Hough, un algoritmo di visione computazionale in grado di riconoscere diverse forme geometriche da un frame. Grazie a questo troviamo i centri dei punti che useremo per calcolare le matrici nel piano immagine delle telecamere. Considerando una corrispondenza di punti vista da una coppia di telecamere calcoleremo la matrice fondamentale F . Applicando questo procedimento su tutta la rete, individueremo una prima calibrazione. Questo step consente di calcolare la distanza tra gli agenti, informazione che useremo per applicare un algoritmo che individui un cammino minimo nella rete prima di ricalibrarla.

Dati iniziali

Il sistema è composto da una serie di telecamere, un piano e una persona che si muove su quest'ultimo. L'operatore durante le simulazioni viene modellato da una pallina che rappresenta la testa e una che rappresenta il punto medio tra i

piedi. Questi due marker giacciono sempre su due piani paralleli ad una distanza costante.

La sua traiettoria davanti alla rete di telecamere sarà totalmente arbitraria. Gli unici due vincoli sono:

- Passare davanti a tutte le telecamere
- Passare per ogni coppia di videocamere in una zona vista da entrambe

Consideriamo ora la rete delle telecamere in maniera più dettagliata: sono sincronizzate e posizionate in maniera fissa, la locazione è arbitraria e ogni telecamera singola può avere parametri interni differenti ma noti. L'unica restrizione che consideriamo è quella di avere per ogni coppia di telecamere una zona del campo visivo comune. Nella rete supponiamo di avere un numero n di agenti calibrati, cioè dei quali conosciamo precisamente la posizione nel sistema di riferimento universale.

La traiettoria dell'operatore comincia proprio davanti ad una camera nota. In coincidenza con la sua partenza si avvia l'acquisizione di frame su ogni telecamera, adesso l'operatore segue la traiettoria scelta in precedenza e arrivato alla fine del suo cammino si arrestano tutte le acquisizioni.

Successivamente analizziamo tutti i flussi video partendo dai frame della telecamera nota di partenza. Sappiamo che nel frame al primo istante troviamo l'operatore. Adesso scannerizziamo tutti i frame delle altre telecamere all'istante 1. Se non rintracciamo in nessuno l'operatore, passiamo alla verifica dei frame all'istante successivo. Questa operazione viene ripetuta finché individuiamo una corrispondenza.

A questo punto calcoliamo la matrice di fondamentale e i parametri necessari e la telecamera incognita diventa calibrata. Adesso ripetiamo tutto il procedimento, ma utilizzando l'agente appena calibrato come punto di partenza. In questo modo tarriamo tutta la rete. Valutiamo ora diversi casi speciali.

Se per caso tre o più telecamere vedono allo stesso istante l'operatore procediamo come prima, ma calibriamo le due telecamere incognite una ad una sempre partendo dall'agente noto. Fatta questa prima calibrazione impieghiamo la funzione per determinare la baseline e quindi individuare la distanza tra tutte le telecamere della rete.

Queste nuove informazioni le inseriamo in una matrice di adiacenza e applichiamo un algoritmo per individuare il cammino minimo basato sulla lunghezza della baseline. Seguendo ora questa sequenza di calibrazione (data dal cammino

trovato), diminuirò l'errore sulla posizione di ogni singola telecamera, come spiegato in seguito.

5 Calcolo matrice F

Per il calcolo della matrice F sarà necessario disporre di 2 frame in 2 camere dai quali ricavare minimo 8 punti di corrispondenza. Otterremo tali punti tramite la sovrapposizione di una serie di frame tratti da un video che riprende una persona che percorre una traiettoria casuale (non regolare come una linea retta) sulla scena. Se necessario si può valutare la possibilità di utilizzare un marker per rendere più evidente il punto da prendere per le corrispondenze.

L' algoritmo di calcolo della matrice F viene spiegato nell'appendice A per non appesantire la trattazione in questo capitolo. Qui ci soffermeremo invece sul metodo di raffinamento della soluzione iniziale trovata.

Basandoci sul testo di Hartley e Zisserman (1) pg 284 abbiamo cercato di implementare l'algoritmo Gold Standard per la stima di massima verosimiglianza (MV) della matrice F. Tale algoritmo è più complesso di altri da implementare, ma dà risultati migliori con l'assunto che il rumore che interessa la misura dei punti immagine sia Gaussiano, che è l'ipotesi che faremo noi. In questo caso la stima di MV è quella che minimizza la distanza geometrica (l'errore di proiezione):

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \quad (5)$$

dove $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ sono le corrispondenze misurate e $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ sono le corrispondenze vere (che andranno stimate) che soddisfano esattamente $\mathbf{x}_i \mathbf{F} \mathbf{x}'_i = 0$ per una F di rango 2 (soluzione iniziale).

I passi dell'algoritmo si possono schematizzare nel modo seguente:

- Si calcola una stima iniziale della matrice \hat{F} con un algoritmo lineare.
- Si calcola una stima iniziale delle variabili ausiliarie \hat{x}_i, \hat{x}'_i nel modo seguente:
 - Si prendono $P = [I|0]$ e $P = [[\mathbf{e}'] \times \hat{F} | \mathbf{e}']$ come matrici per le videocamere, con \mathbf{e}' ottenuto da \hat{F} .
 - Da $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ e \hat{F} si calcola con un metodo di triangolazione una stima di $\hat{\mathbf{X}}_i$.
 - Otteniamo infine le corrispondenze $\hat{x}_i = P \hat{\mathbf{X}}_i$ e $\hat{x}'_i = P' \hat{\mathbf{X}}_i$
- Si minimizza il costo

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \quad (6)$$

su \hat{F} e $\hat{\mathbf{X}}_i$, $i = 1, \dots, n$ usando il metodo di Levenberg-Marquardt.

L'idea è di variare P' e $\hat{\mathbf{X}}_i$ in modo da minimizzare l'errore, e infine si calcola F come $F = [t] \times M$.

Notiamo quindi che è necessario determinare i punti $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ che minimizzano la funzione di costo (con relativo vincolo):

$$C(\mathbf{x}, \mathbf{x}') = d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \rightarrow \hat{\mathbf{x}}'^T \mathbf{F} \hat{\mathbf{x}} = 0 \quad (7)$$

Sappiamo che i punti che rispettano il vincolo epipolare devono giacere esattamente sulle rispettive linee epipolari nei due frame, e i punti che minimizzano il costo giacenti sulle linee epipolari sono esattamente quelli che minimizzano la distanza dalle linee dei punti misurati. Possiamo quindi riformulare il problema e dire che ciò che desideriamo minimizzare è

$$d(\mathbf{x}_i, l)^2 + d(\mathbf{x}'_i, l')^2 \quad (8)$$

Vediamo nel dettaglio come effettuare la minimizzazione nell'appendice B, limitandoci a riportare il termine che alla fine andrà minimizzato e i passi dell'algoritmo suggeriti da Hartley e Zisserman.

$$g(t) = t((at + b)^2 + f'^2(ct + d)^2)^2 - (ad - bc)(1 + f'^2t^2)^2(at + b)(ct + d). \quad (9)$$

Algoritmo

- Si parte da due punti corrispondenti $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ e da una matrice F
- Definiamo le matrici di trasformazione

$$T = \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 0 \end{bmatrix} \quad T' = \begin{bmatrix} 1 & 0 & -x' \\ 0 & 1 & -y' \\ 0 & 0 & 1 \end{bmatrix}. \quad (10)$$

che portano i 2 punti nell'origine

- Sostituiamo F con $T'^{-T}FT$ corrispondente alle nuove coordinate

-
- Si calcolano l'epipolo destro e quello sinistro e si normalizzano in modo da avere $e_1^2 + e_2^2 = 1$.
 - Si creano le due matrici

$$R = \begin{bmatrix} e_1 & e_2 & 0 \\ -e_2 & e_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R' = \begin{bmatrix} e'_1 & e'_2 & 0 \\ -e'_2 & e'_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (11)$$

osservando che sono matrici di rotazione.

- Sostituire F con $R'FR^T$ che avrà la forma spiegata nell'appendice
- Porre $f = e_3, f' = e'_3, a = F_{22}, b = F_{23}, c = F_{32}, d = F_{33}$.
- Risolvere in t l'equazione trovando le 6 radici.
- Valutare la funzione di costo $s(t)$ (vedi appendice) nella parte reale delle 6 radici e selezionare il t che dà il valore minimo.
- Calcolare le linee epipolari nel t trovato al punto precedente ed individuare \hat{x} e \hat{x}' come i punti su tali rette più vicini all'origine
- Tornare alle coordinate originali
- Trovare il punto nello spazio 3-D con un metodo omogeneo

6 Fattorizzazione della matrice essenziale

Riportiamo qui di seguito il metodo per fattorizzare la matrice essenziale così come viene esposto dal Professor Fusiello in *Visione computazionale -Appunti delle lezioni-*

Supponiamo che sia data la matrice essenziale $E = K_2^T F K_1$. Il seguente teorema dovuto a Huang e Faugeras [1989] caratterizza la matrice essenziale e ci permette di fattorizzarla la matrice essenziale in rotazione e traslazione.

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

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

Si riporta la dimostrazione così come esposta nel testo del Professor Fusiello.

Sia $E = SR$ dove R è una matrice di rotazione e S è antisimmetrica. Sia $S = [t]_{\times}$ con $||t|| = 1$ (con nessuna perdita di generalità, dato che E è definita a meno di un fattore di scala). Sia U la matrice di rotazione t.c. $Ut = [0, 0, 1]^T := a$, quindi $S = [t]_{\times} = [U^T a]_{\times}$. Possiamo inoltre scrivere:

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

Consideriamo ora la matrice EE^T :

$$EE^T = SR R^T S^T = SS^T = U^T [a]_{\times} U U^T [a]_{\times}^T U = U^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} U. \quad (12)$$

Gli elementi della matrice diagonale sono gli autovalori di E^T cioè, i valori singolari di E . Questo dimostra un'implicazione.

Diamo ora una dimostrazione costruttiva dell'implicazione inversa. Sia $E = UDV^T$ la SVD di E , con $D = \text{diag}(1, 1, 0)$ (senza perdita di generalità) e U e V ortogonali. L'osservazione chiave è che:

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} := S'R' \quad (13)$$

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

$$E = UDV^T = US'R'V^T = (US'U^T)(UR'V^T) = \det(UV')(US'U^T)\det(UV^T)(UR'V^T).$$

Prendendo $S = \det(UV^T)US'U^T$ e $R = \det(UV^T)UR'V^T$, la fattorizzazione cercata è: $E = SR$. Infatti, $US'U^T$ è antisimmetrica e la matrice $\det(UV^T)UR'V^T$ è ortogonale con determinante positivo (grazie al termine $\det(UV^T)$) quindi è di rotazione.

Questa fattorizzazione non è unica. Data l'ambiguità del segno di E, possiamo cambiare il segno di D sia cambiando il segno di S' che trasponendo R' (dato che $S'R'^T = -D$). In totale dunque abbiamo quattro possibili fattorizzazioni date da:

$$\begin{aligned} S &= U(\pm S')U^T \\ R &= \det(UV^T)UR'V^T \text{ oppure } R = \det(UV^T)UR'^T V^T \end{aligned}$$

dove

$$S' := \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad R' := \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

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

7 Calcolo fattore di scala

Dopo aver calcolato la matrice F si è in grado di calcolare la rotazione e la traslazione tra la prima e la seconda telecamera (figura 2) passando per la matrice Essenziale, $E = K_2^T F K_1$, che richiede la conoscenza dei parametri intrinseci delle due telecamere (noti). Purtroppo la traslazione ottenuta risulta nota a meno di un fattore di scala, che evidentemente è necessario calcolare per trovare la posizione relativa tra le due telecamere.

Dopo un'attenta simulazione si è notato che il fattore di scala cercato non è altro che la lunghezza della baseline, ovvero la distanza tra i centri ottici delle telecamere, infatti moltiplicando la matrice di traslazione per questo fattore si ottiene la posizione relativa della seconda telecamera rispetto alla prima (figura 2).

Per risolvere il problema è necessario avere delle informazioni aggiuntive: devono essere note le altezze delle telecamere rispetto al piano e la matrice di rotazione che porta il riferimento della prima telecamera nel riferimento mondo (\tilde{R}).

Per risolvere il problema basta trovare la lunghezza della retta che parte dal centro ottico della prima telecamera e arriva nel centro ottico della seconda telecamera. Il problema pertanto si riduce a trovare l'intersezione tra la retta (baseline) e il piano (verde), dato come sistema di riferimento il sistema di riferimento mondo *riportato alla prima telecamera* (azzurro). Rispetto a quest'ultimo sistema di riferimento il piano (verde) ha la seguente equazione $X = -\text{differenza tra l'altezza della seconda e della prima telecamera} = \text{diff}$, mentre trovare l'equazione della retta con tale riferimento è più complesso.

L'equazione parametrica della retta in \mathfrak{R}^3 è infatti composta da un versore direzionale (V), che esprime appunto la direzione di quest'ultima, e da un punto $C1$, attraverso il quale la retta passa. L'equazione risulta dunque:

$$\text{baseline} : tV + C1 = t \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} + \begin{bmatrix} c1_x \\ c1_y \\ c1_z \end{bmatrix} \quad (14)$$

dove $t \in \mathfrak{R}$ è il parametro della retta. Tale retta può anche essere rappresentata in questo modo:

$$\text{baseline} : \begin{cases} X = v_x t + c1_x \\ Y = v_y t + c1_y \\ Z = v_z t + c1_z \end{cases} \quad (14)$$

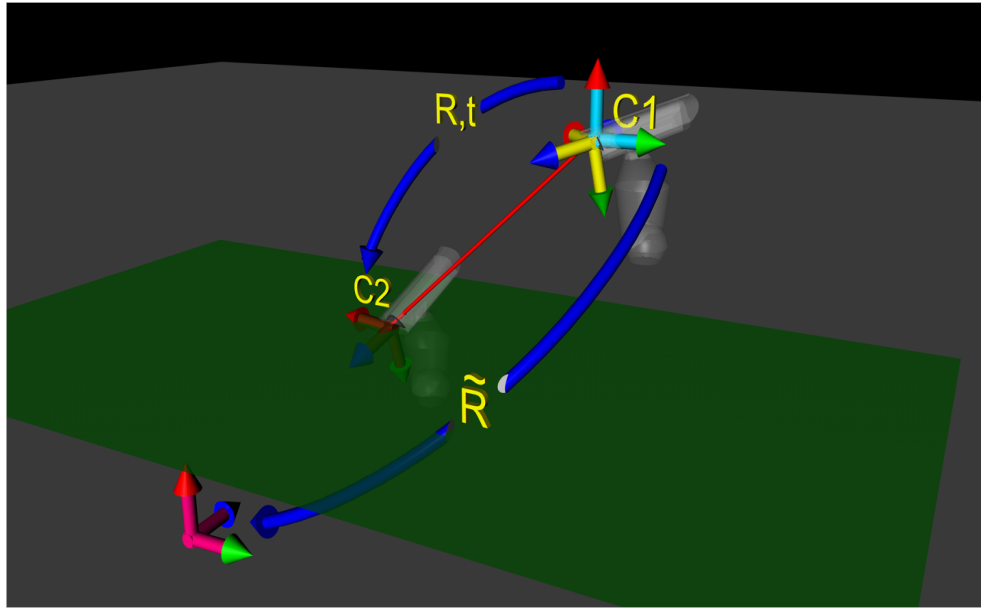


Figura 2: Nella foto si mette in evidenza la baseline (rossa), il riferimento mondo (rosa), il riferimento standard della prima telecamera(giallo), il riferimento standard della seconda telecamera(marrone), il sistema di riferimento mondo riportato alla prima telecamera (azzurro), il piano (verde) dato dalla differenza tra le altezze delle telecamere ed infine i punti C1 e C2 che sono i centri ottici rispettivamente della prima e della seconda telecamera.

Trovando infine t nella prima equazione e sostituendo nelle successive si ha

$$baseline : \begin{cases} -v_y v_x^{-1} X + Y = -c1_x v_y v_x^{-1} + c1_y \\ -v_z v_x^{-1} X + Z = -c1_x v_z v_x^{-1} + c1_z \end{cases}$$

(14)

Queste ultime uguaglianze si chiamano *equazioni cartesiane* della retta. Ora basta portare le equazioni della retta nelle coordinate del sistema di riferimento azzurro, facendo attenzione a come si trasforma il vettore direzionale della baseline.

Inizialmente per trovare C2 nel sistema di riferimento della prima camera (giallo), a meno di un fattore di scala, basta semplicemente utilizzare la matrice inversa dei parametri estrinseci (data da R e t) ricavata con il calcolo della E,

successivamente per riportare tutto al sistema di riferimento azzurro basta moltiplicare per \tilde{R} trovando in questo modo il vettore direzionale V nel sistema di riferimento che volevamo. Si mette inoltre in evidenza che nel nuovo sistema di riferimento $C1 = [0, 0, 0, 1]^T$.

Trovando infine t nella prima equazione e sostituendo nelle successive si ha

$$\begin{cases} X = -diff \\ -v_y v_x^{-1} X + Y = 0 \\ -v_z v_x^{-1} X + Z = 0 \end{cases} \quad (14)$$

Si trovano in questo modo le coordinate del secondo centro ottico $C2$ rispetto al riferimento azzurro, pertanto il fattore di scala iniziale è:

$$fattore\ di\ scala := \sqrt{X^2 + Y^2 + Z^2}$$

Per calcolarlo è dunque necessario conoscere l'altezza delle telecamere rispetto al riferimento mondo (rosa), la matrice \tilde{R} e la scomposizione di E in R e t a meno di un fattore di scala.

Per il calcolo della matrice \tilde{R} si è utilizzato l'algoritmo DLT così da rendere ancora più realistica la simulazione.

8 Calcolo dell'altezza

La base teorica dalla quale partiamo è l'omologia planare H , che è una omografia particolare che ha una linea l di punti fissi e un punto fisso p che non giace su l . Un punto fisso è tale se con H si trasforma in sé stesso.

La scena che consideriamo nella nostra relazione è un sistema con una telecamera, con una superficie piana ed una persona con altezza costante che cammina secondo i vincoli descritti nel capitolo in cui spieghiamo l'algoritmo. La trasformata di Hough fornisce il punto medio (x_1, y_1) della pallina sul piano delle teste, e un punto medio (x_2, y_2) della pallina per terra. Ora avendo le coordinate di questi due nel piano immagine, possiamo calcolarci la retta che interseca i due punti risolvendo il sistema di due equazioni con le solite formule:

$$\begin{aligned}y_1 &= mx_1 + c \\y_2 &= mx_2 + c\end{aligned}$$

Inoltre si trova la distanza di questi due punti, quindi l'altezza del operatore, con la formula della distanza euclidea:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

L'omologia planare è l'immagine di due piani. Prendendo i punti medi delle teste e dei piedi troviamo che giacciono su due piani paralleli distanti tra loro per l'altezza nota della persona. Questi due si incontrano all'infinito in una linea. Riportandola nel piano immagine si trova la vanishing line. Questa è proprio la linea fissa della nostra omologia tra i due piani. Il punto fisso invece è il vanishing point. Con queste informazioni possiamo scrivere la seguente relazione:

$$H = I - \frac{h v^\infty * (l^\infty)^T}{z (v^\infty)^T * l^\infty} \quad (12)$$

ove v^∞ è il vanishing point, l^∞ la vanishing line, h è l'altezza della persona e infine z è l'altezza della telecamera richiesta.

In appendice C è possibile trovare una spiegazione più esaustiva sul calcolo dell'omografia.

9 Calibrazione Stereo

In seguito a tutte le considerazioni fatte finora vediamo come sia possibile calibrare globalmente una telecamera non calibrata che condivide una porzione del campo visivo con una telecamera calibrata.

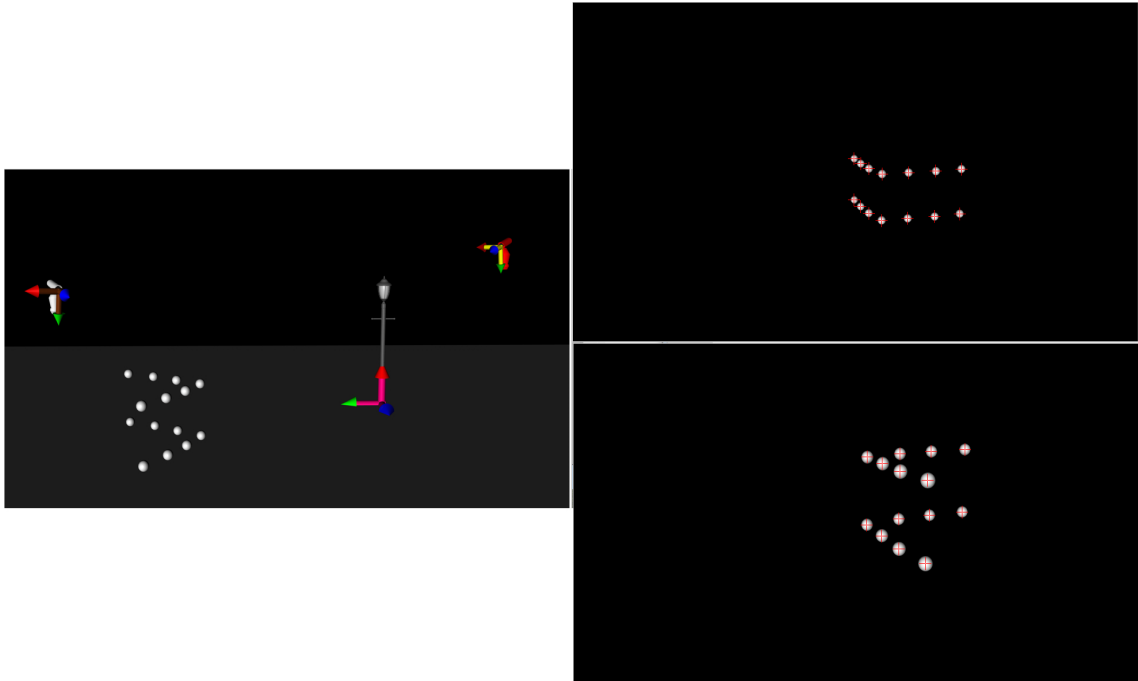


Figura 3: Nell'immagine a sinistra si mette in evidenza il riferimento della telecamera calibrata (giallo), il riferimento della telecamera non calibrata (marrone) ed infine il riferimento mondo (rosa). L'immagine più in alto a destra rappresenta la visuale della camera rossa dopo il post-processing, mentre l'immagine sottostante rappresenta la visuale della telecamera bianca

Della telecamera calibrata saranno ovviamente noti i parametri estrinseci fra il riferimento standard di quest'ultima e il riferimento mondo, che è fisso, all'interno della scena (potrebbe essere un palo, un albero, lo spigolo di una casa, ecc..). Ancora una volta indichiamo con \tilde{R} questa matrice.

La telecamera che si vuole calibrare dovrà avere il campo visivo parzialmente sovrapposto con quella calibrata, in questo modo si riesce a vedere la stessa traiettoria da due punti di vista differenti e una volta fatto il post-processing dei

flussi video si ottengono le immagini di figura 3. Selezionando i punti coniugati nelle due immagini si ottiene così la matrice F e quindi la matrice di rotazione e traslazione (chiamate ancora R e t) che portano la telecamera calibrata in quella da calibrare a meno di un fattore di scala. Adottando il procedimento per determinare il fattore di scala, note le altezze delle telecamere e i loro parametri intrinseci, siamo in grado di calibrare globalmente anche la telecamera inizialmente non calibrata rispetto a quella calibrata. L'idea di partire da una telecamera già calibrata globalmente permettere di non dover calcolare la matrice \tilde{R} in quanto già nota.

Ci si occupa ora di estendere il procedimento ad una rete di telecamere partendo da una telecamera calibrata globalmente.

Calibrazione Rete di telecamere

Una volta capito il procedimento di calibrazione fra una coppia di telecamere non è difficile calibrarne una rete, questa infatti può sempre essere vista come l'insieme di tante coppie di telecamere prese a due a due. Si elencano i dati necessari per la calibrazione stereo:

- L'altezza delle due telecamere
- La matrice \tilde{R} (nota)
- Le matrici R e t (a meno di un fattore di scala)
- I parametri intrinseci delle telecamere (noti)

Supponendo noti i parametri precedenti per ogni coppia di telecamere ci si rende conto che l'unico problema è il calcolo della matrice \tilde{R} rispetto al riferimento della telecamera calibrata di partenza. Per questo motivo si è deciso di utilizzare un approccio incrementale per la ricostruzione della posizione di ogni singola telecamera nella rete. Si parte dunque da due telecamere, una calibrata e una non calibrata, per le quali sono noti i parametri precedenti, come visto si riesce a calibrare la telecamera non calibrata rispetto a quella calibrata (figura 3).

Si aggiunge ora una terza telecamera non calibrata (marrone) nella rete che ha il campo visivo sovrapposto con l'ultima telecamera calibrata, pertanto anche in questo caso si riescono a prendere almeno otto punti coniugati di una traiettoria in modo da calcolare la matrice F e quindi la matrice R_2 e t_2 (sempre a meno di un fattore di scala) tra queste due telecamere. Per terminare la calibrazione è ora necessario conoscere la matrice di rotazione \tilde{R}_2 che porta il riferimento dell'ultima telecamera calibrata nel riferimento mondo, tale matrice è chiaramente:

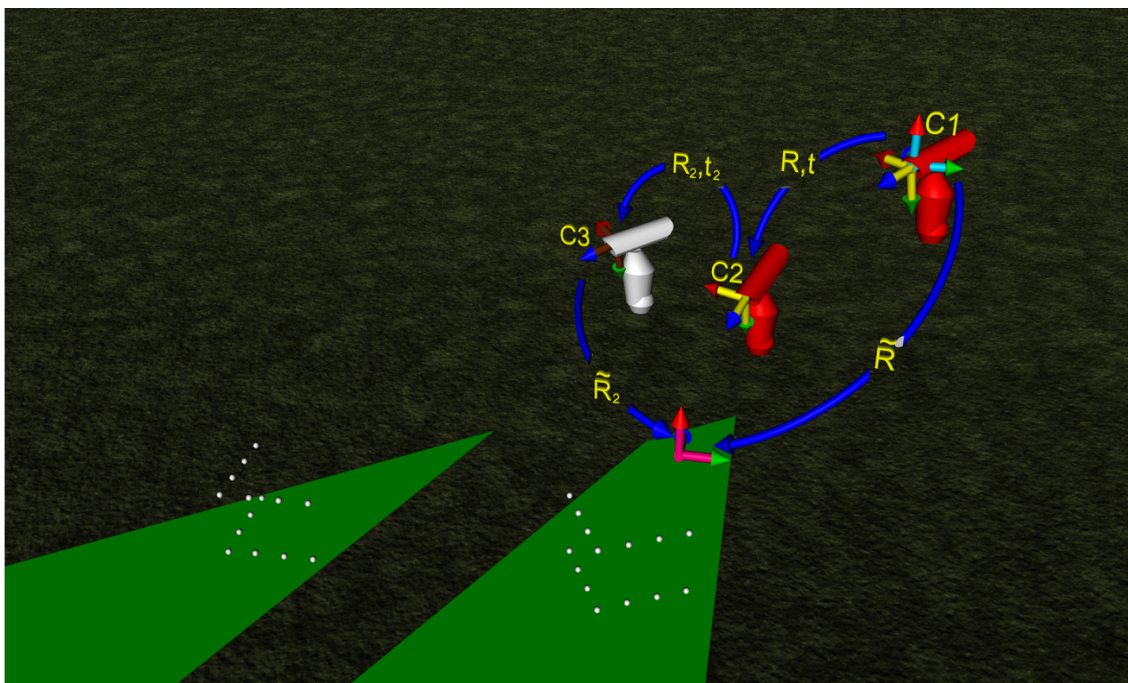


Figura 4: Si mettono in evidenza, in rosso, le prime due telecamere già calibrate (riferimenti gialli), il riferimento di quella non ancora calibrata (marrone), la matrice \tilde{R}_2 e la parte di campo visivo che si sovrappone tra le telecamere(verde)

$$\tilde{R}_2 = \tilde{R}R^{-1}R_2^{-1}$$

Infatti per passare dal riferimento dell'ultima camera calibrata, al riferimento mondo, basta moltiplicare per R^{-1} per passare a quello della prima telecamera calibrata e moltiplicando tutto per \tilde{R} si passa al sistema di riferimento mondo. Mano a mano che si avanza con la calibrazione, all'interno della rete, i parametri che le telecamere devono passarsi sono dunque la matrice \tilde{R} e tutte le matrici di rotazione ricavate dal calcolo della matrice F fino alla telecamera che si sta cercando di calibrare.

10 Simulazioni

Per le simulazioni si è utilizzato il software CINEMA 4D, famoso software per la grafica e l'animazione 3D, che permette l'inserimento di telecamere e altri oggetti all'interno di un ambiente virtuale tridimensionale. Per rendere tale software compatibile con il programma scritto in Matlab si è dovuto implementare un programma aggiuntivo con il programma Xpresso (plug-in di Cinema 4D).

Vista la facilità di utilizzo del programma non è stato difficile creare una scena con due o più telecamere e alcuni punti che simulano la traiettoria dell'operatore. Prendendo successivamente le singole viste delle telecamere e facendole elaborare dal programma in Matlab si riesce a ricostruire la posizione e l'orientazione delle telecamere all'interno della scena:

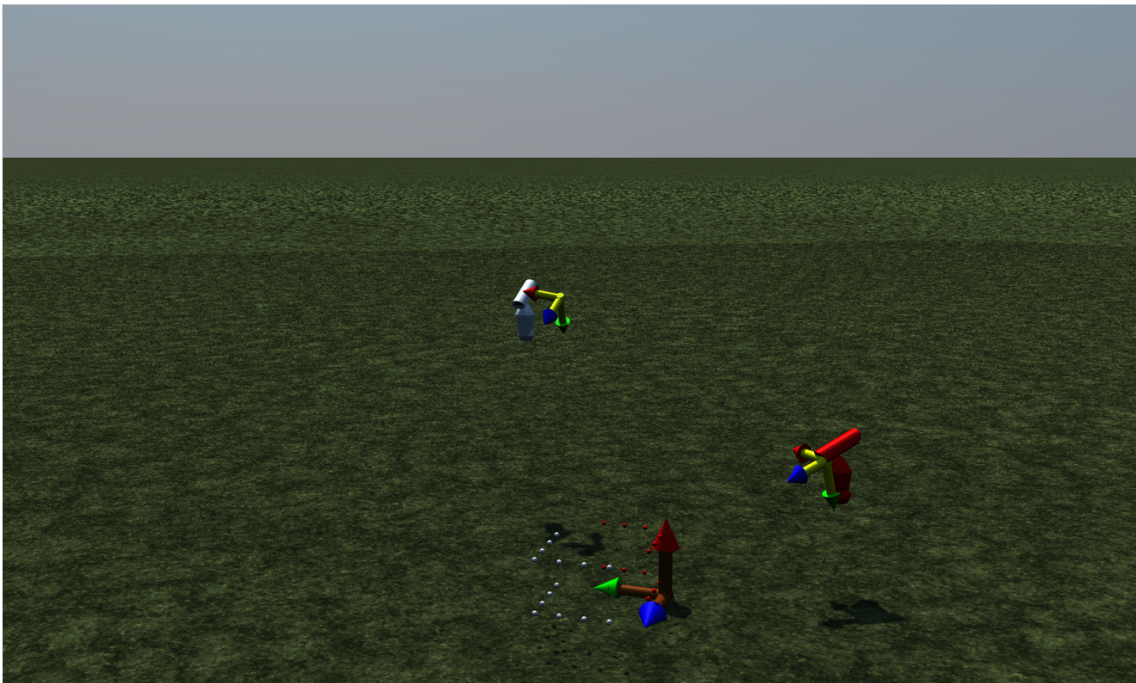


Figura 5: Si mettono in evidenza, in rosso, la telecamera già calibrata rispetto al sistema di riferimento mondo (marrone), il sistema di riferimento giallo da l'indicazione di dove dovrebbe trovarsi la telecamera bianca secondo il nostro algoritmo

Visto che non siamo in possesso dei parametri estrinseci ed intrinseci della

telecamera calibrata li calcoliamo attraverso l'algoritmo DLT e la conoscenza di almeno sei punti (rossi) noti nella scena, rispetto al riferimento mondo adottato. Successivamente si prendono le corrispondenze dei punti della traiettoria (bianchi) tra le telecamere e si calcola la posizione della telecamera non calibrata (bianca) rispetto quella calibrata (rossa) attraverso l'algoritmo degli otto punti.

Si nota che l'algoritmo non riesce a restituire precisamente la posizione della telecamera bianca (vedi riferimento giallo) pertanto, visto che la calibrazione della rete sarà fatta in maniera incrementale, tale errore non potrà che peggiorare ad ogni calibrazione successiva. Si ritorna ad avere nuovamente un errore minimo non appena si incontra nel percorso un'altra telecamera calibrata della quale sappiamo con precisione la matrice di rotazione rispetto al suo riferimento mondo. Per questo motivo effettuiamo una ricalibrazione della rete dopo aver fatto una calibrazione iniziale in sequenza (una videocamera dopo l'altra). La calibrazione iniziale ci permetterà di costruire un grafo della rete che darà utili informazioni su come effettuare la ricalibrazione, come si vedrà in seguito.

Risulta evidente che l'errore viene a propagarsi attraverso le telecamere non calibrate e *in questo caso* peggiora ad ogni passo. A priori non è possibile sapere come l'errore influisca sulla calibrazione, paradossalmente potrebbe anche migliorare, per questo è stata fatta un'analisi sulle cause dell'errore.

Errori

Le cause di errore sono molteplici e influiscono in maniera diversa sul programma implementato. La prima causa di errore è data proprio dal modello della telecamera, la pixelizzazione dell'immagine, infatti, quantizza la scena reale rendendo praticamente impossibile trovare in maniera perfetta i punti coniugati in due o più viste della medesima zona. Se a questo si aggiunge l'errore dovuto all'utente, o ad algoritmi di visione computazionale sofisticati (trasformata di Hough), nel cliccare sui punti coniugati si ha inevitabilmente un errore sulla matrice fondamentale. Tale errore, seppur piccolo, fa in modo che non sussista l'uguaglianza tra i punti coniugati nelle due immagini: $m_2' F m_1 \neq 0$.

Inoltre l'algoritmo utilizzato per calcolare la matrice F (algoritmo degli otto punti) è stato criticato per essere troppo sensibile al rumore e quindi poco utile nelle applicazioni pratiche perché è lineare e mal condizionato [Hartley (3)]. Il problema del mal condizionamento è stato risolto facendo una semplice standardizzazione dei dati, trasladando il centroide dei punti nell'origine dell'immagine e poi scalando i punti affinché la distanza media dall'origine sia pari a $\sqrt{2}$. L'errore che rimane, dovuto anche alla linearità dell'algoritmo, si propaga nel calcolo della

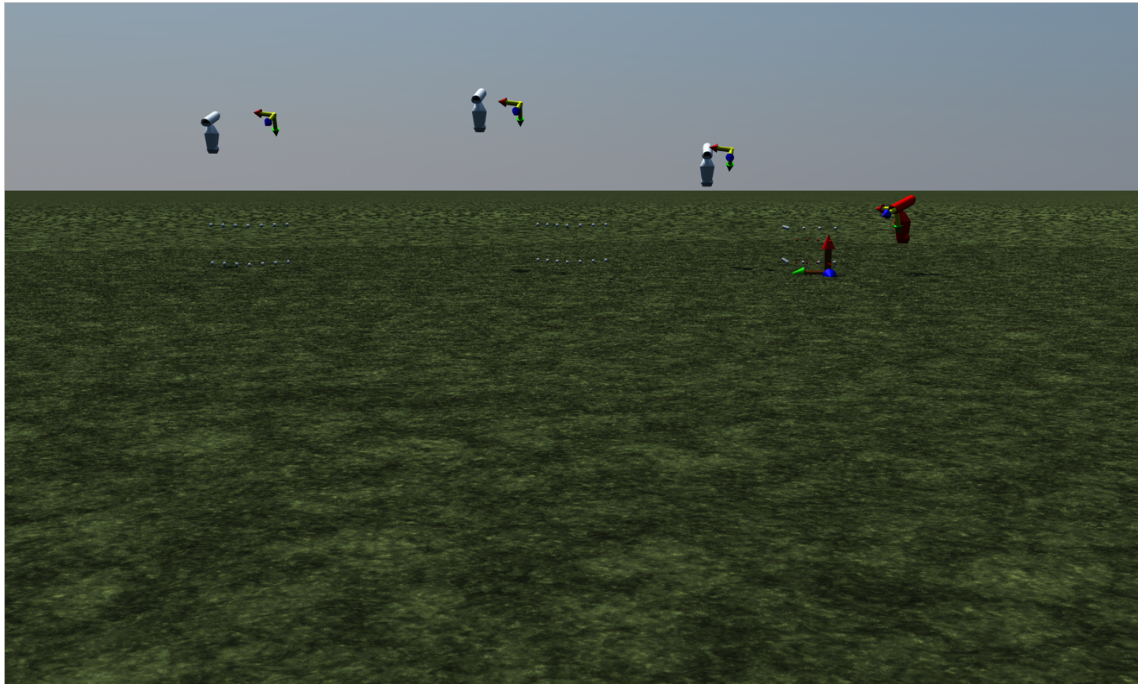


Figura 6: Si mettono in evidenza, in rosso, la telecamera già calibrata rispetto al sistema di riferimento mondo (marrone), i sistemi di riferimento gialli danno l'indicazione di dove dovrebbero trovarsi le telecamere bianche.

matrice essenziale e nella sua scomposizione nelle matrici R e t che portano il riferimento della prima telecamera nella seconda. L'errore fin qui accumulato nelle matrici R e t , nota a meno di un fattore di scala, non è molto grande si sbaglia al più di un grado. Il problema nasce dal fatto che le telecamere sono molto distanti tra loro, quindi per il teorema della corda l'errore aumenta di molto anche se si sbaglia di pochi gradi. Si ha che l'errore di un grado su 30 metri porta ad un errore complessivo di:

$$\bar{AB} = 2R\sin\left(\frac{\beta}{2}\right) = 2 * 30\sin\left(\frac{1^\circ}{2}\right) = 0.52m$$

In seguito a queste considerazioni abbiamo potuto stabilire che l'errore è dovuto alla lontananza tra le telecamere e, a priori, è imprevedibile misurare che effetti porti all'interno del sistema di calibrazione. Infatti non si può sapere in che direzione, e in che verso, è sbagliato il vettore di traslazione (t) che è il vettore-

re direzionale della baseline, portando errori sul calcolo di quest'ultima in una direzione non nota.

L'unico modo per migliorare ulteriormente questo errore è affinare il calcolo della matrice fondamentale e quindi minimizzando un qualche errore geometrico in modo da rendere l'algoritmo degli otto punti non lineare, come evidenziato nel capitolo 5.

11 Ricalibrazione

In questo capitolo si presenta un metodo per diminuire l'errore sulla posizione di ogni singola telecamera della rete inizialmente non calibrata. Dopo la prima calibrazione si conosce la posizione delle telecamere rispetto al relativo sistema di riferimento mondo, noto per costruzione. Si associa alla rete così calibrata un grafo non orientato e pesato nel quale i nodi della rete rappresentano le telecamere e gli archi sono pesati rispetto alla lunghezza della baseline tra i corrispondenti agenti. Successivamente viene applicato un algoritmo per la determinazione dei cammini minimi tra ogni telecamera, inizialmente non calibrata, e le telecamere calibrate fin dall'inizio. Utilizzando queste informazioni si è ora a conoscenza di quale siano le telecamere più vicine a quelle calibrate inizialmente, questo ci fornisce una sequenza più adeguata per ricalibrare gli agenti. Per fare questo si utilizzeranno le seguenti basi teoriche:

- Grafo
- Matrice di adiacenza
- Cammino minimo
- Algoritmo di Dijkstra

Verrà in seguito esposto il modo di operare e si faranno cenni sulla complessità computazionale dell'algoritmo.

Grafo

Un grafo G è un'insieme di elementi detti vertici o nodi v_i , collegati fra loro tramite archi e_i .

$$v_i \in V, e_i \in E \Rightarrow G = (V, E)$$

Nel nostro caso i nodi rappresentano le telecamere mentre la presenza di un arco $\{i, j\}$ indica che le telecamere i -esima e j -esima hanno i campi visivi che si intersecano, cioè vedono passare l'operatore davanti ai loro obiettivi nello stesso istante. L'informazione più importante per questa parte della tesi è però il peso che associamo ad ogni arco, esso rappresenta la norma della distanza tra i centri ottici fra due telecamere cioè la lunghezza della baseline calcolata nella sezione 7.

Matrice di adiacenza

Un grafo con N nodi può essere rappresentato da una matrice $N \times N$. La matrice di adiacenza $A = (a_{ij})$ viene calcolata nel seguente modo, solamente dopo la calibrazione iniziale: nella riga i -esima e nella colonna j -esima viene inserito il peso dell'arco $\{i, j\}$ (baseline). Se una connessione $\{i, j\}$ non esiste si scrive uno zero.

$$a_{ij} = \begin{cases} d_{ij} & , \text{ se le telecamere } i\text{-esima e } j\text{-esima sono adiacenti} \\ 0 & , \text{ se le telecamere } i\text{-esima e } j\text{-esima non sono adiacenti} \end{cases}$$

La matrice di adiacenza ha diverse caratteristiche strutturali. Per prima cosa è simmetrica perché il grafo associato è non orientato, in più, ha elementi nulli sulla diagonale principale perché è impossibile avere cappi in una rete di telecamere. Dalla caratteristica di non avere un grafo completo troviamo una matrice con tanti elementi nulli.

E' importante non perdere l'informazione iniziale su quali siano le telecamere calibrate inizialmente e quali no, perché dopo la prima calibrazione tutte risultano calibrate. Per questo si utilizza un array di dimensione uguale al numero dei vertici del grafo, nel quale ad ogni posizione viene associato un 1 se l'agente è calibrato fin dall'inizio, altrimenti viene assegnato uno 0. L'acquisizione di questi dati può essere svolta insieme alla prima calibrazione della rete, tale struttura dati in seguito sarà chiamata lista calibrata.

Cammino minimo

Dato il nostro grafo $G = (V,E)$ non orientato e pesato. Il costo di un cammino $c = \langle v_0, v_1, v_2, \dots, v_k \rangle$ è dato da:

$$\omega(c) = \sum_{i=1}^k \omega(v_{i-1}, v_i)$$

Un cammino minimo tra una coppia di vertici x e y è un cammino di costo (in questo caso la distanza d_{xy} tra due vertici x e y) minore o uguale a quello di ogni altro cammino tra gli stessi vertici.

Algoritmo Dijkstra

L'algoritmo di Dijkstra deve il suo nome all'informatico Edsger Dijkstra e permette di trovare i cammini minimi in un grafo ciclico con pesi non negativi sugli

archi: in particolare l'algoritmo può essere utilizzato parzialmente per trovare il cammino minimo che unisce due nodi del grafo. Descriviamo ora in breve il pseudocodice:

$D = \{d_{ij}\}$ matrice di adiacenza
 P insieme dei nodi appartenenti al cammino minimo
 d_j distanza del cammino totale
 d_i distanza temporanea

- Inizializzazione:

- $P = \{1\}, D_1 = 0, D_j = d_{1j}$

- Passi dell'algoritmo:

- Trova $i \notin P$ tale che $D_i = \min_{j \notin P} \{D_j\}$ e poni $P \leftarrow P \cup \{i\}$. Se P contiene tutti i nodi l'algoritmo è completato.
 - Per tutti i $j \notin P$, poni $D_j \leftarrow \min\{D_j, d_{ij+D_i}\}$ e torna al passo precedente.

Procedimento

Partiamo dal presupposto di avere la matrice di adiacenza e la lista calibrata a disposizione dopo la prima calibrazione. L'obiettivo è di calcolare il cammino minimo da ogni telecamera evidenziata con lo zero nella lista calibrata, verso tutte le telecamere note, evidenziate con l'1 nella lista calibrata. Quindi diamo in ingresso all'algoritmo di Dijkstra la matrice di adiacenza, un nodo di partenza e uno di arrivo.

Dijkstra fornisce in uscita due parametri, la distanza minima $d_{min,i}$ e un'insieme di vertici T_i che compongono la sequenza del cammino minimo dal nodo di partenza a quello d'arrivo. Calcolati tutti i cammini minimi tra una telecamera assegnata con lo zero sulla lista calibrata e tutte le telecamere evidenziate con 1, si sceglie il più piccolo di questi. Questo procedimento si deve ripetere per ogni telecamera inizialmente non calibrata(0) cioè $N - n$ volte, verso tutte le telecamere inizialmente calibrate(1). Questi $(N - n)$ cammini minimi vengono ordinati in maniera decrescente secondo la loro lunghezza e viene preso il cammino con lunghezza maggiore (il primo elemento). Si calibrano pertanto in maniera ordinata tutte le telecamere che fanno riferimento a questo cammino e una volta terminata questa calibrazione si prende il secondo cammino più lungo (secondo elemento) e

si riprocede nello stesso modo a patto che questo cammino non sia contenuto nel precedente. Alla fine di queste operazioni avremo una rete di telecamere calibrata con un errore minore rispetto alla precedente calibrazione.

Di per sé l'algoritmo di Dijkstra presenta una complessità pari al quadrato dei nodi del grafo per ogni cammino, ma tenendo conto che nel nostro caso per ogni nodo vengono calcolati almeno n cammini la complessità diventa $O(N^2n)$ nel caso peggiore. Si poteva direttamente utilizzare l'algoritmo di Floyd-Warshall, che calcola tutti i cammini minimi, ma ha un costo computazionale $O(N^3)$. A questo costo si somma anche la complessità computazionale dell'algoritmo di ordinamento degli $(N - n)$ cammini minimi che è pari a $O((N - n)^2)$.

Commenti

Il più grande vantaggio di tutta questa ricalibrazione è la diminuzione dell'errore sulla posizione delle telecamere.

Supponiamo il numero di telecamere note fin dall'inizio $n \ll N$, quindi è probabile che tra due telecamere note ci siano molte telecamere da calibrare. Calibriamo inizialmente la rete un agente dopo l'altro partendo da una telecamera inizialmente nota, ovviamente l'errore sulla posizione si propaga da una telecamera all'altra fino alla successiva calibrata. Solo in quel momento si ricomincia la calibrazione da una posizione nota precisamente.

Visto il modo iniziale di operare è ovvio che una telecamera non calibrata che precede una già calibrata avrà un errore molto grande (sulla posizione), la ricalibrazione permette di diminuire drasticamente tale errore calibrando la telecamera non calibrata rispetto alla telecamera calibrata più vicina. Tale algoritmo utilizza gli stessi frame acquisiti inizialmente, senza che l'operatore ripeta nuovamente la sua passeggiata di fronte alle telecamere. Si mette inoltre in evidenza che il numero di archi E nel grafo $E \ll N^2$ allora l'algoritmo di Dijkstra arriva ad avere anche una complessità di $O(E \log(N))$ ottenuta utilizzando una coda di priorità per memorizzare i vertici.

12 Considerazioni PTZ


Arrivati fino a questo punto si è ricalibrato il sistema in modo da avere meno errori possibili e si è a conoscenza dei cammini minimi di ogni telecamera, inizialmente non calibrata, rispetto a quelle calibrate. Sfrutteremo questa informazione per poter calcolare le matrici di rotazione tra il riferimento standard delle telecamere, inizialmente non calibrate, rispetto al riferimento mondo di quelle già calibrate *più vicine*. Per far ciò basta partire da una telecamera non calibrata qualsiasi e seguendo il cammino minimo che la porta alla matrice calibrata più vicina, di volta in volta si dovrà moltiplicare per le matrici di rotazione che si incontrano nel cammino. In riferimento alla figura 4 tale matrice partendo dalla terza telecamera è data da $R_{mondo} = R_2 R \tilde{R}^{-1}$ o equivalentemente si può calcolare come $R_{mondo} = \tilde{R}_2^{-1}$. Applicando lo stesso procedimento per tutte le telecamere, non calibrate inizialmente, si hanno i riferimenti di queste ultime rispetto al sistema di riferimento mondo a loro più vicino, in questo modo, si svincola la calibrazione di queste rispetto alle altre telecamere incontrate nel cammino minimo.

Visto che le telecamere utilizzate sono PTZ, alla prima rotazione che queste compiono, la rete risulta nuovamente scalibrata o meglio non è più nota l'orientazione di queste telecamere rispetto al riferimento mondo a loro più vicino (chiaramente resta invariata la loro posizione rispetto a quest'ultimo). E' dunque necessario fornire un modello per la telecamera PTZ in grado di tener conto di eventuali rotazioni della telecamera successive alla ricalibrazione.


Il modello della telecamera PTZ che abbiamo teorizzato è molto simile al modello della telecamera pinhole già illustrato, a meno di una matrice di rotazione. Si suppone che le telecamere abbiano una posizione di riposo rispetto alla quale un encoder è in grado di restituire l'angolo di Pan e di Tilt. La calibrazione iniziale e la successiva ricalibrazione vengono fatte ovviamente con tutte le telecamere fisse nella loro posizione di riposo, in questo modo si conosce il riferimento di queste rispetto al riferimento mondo a loro più vicine e si può utilizzare il classico modello pinhole.

L'unico problema che si presenta è come rappresentare eventuali rotazioni, successive alla ricalibrazione, visto che il centro ottico delle telecamere finora utilizzate non corrisponde con il centro di rotazione degli angoli Pan e Tilt, tale problema varia in base alla struttura della telecamera e richiede anche l'aggiornamento della matrice di traslazione rispetto al sistema di riferimento mondo più vicino. Per motivi di tempo si assume che il centro ottico della telecamera corrisponda con il suo centro di rotazione, in questo modo non si tiene conto della

matrice di traslazione.

$$R_{pan} = \begin{bmatrix} \cos(-57.134) & 0 & \sin(-57.134) \\ 0 & 1 & 0 \\ -\sin(-57.134) & 0 & \cos(-57.134) \end{bmatrix}$$


-57.134

$$R_{tilt} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(24.33) & -\sin(24.33) \\ 0 & \sin(24.33) & \cos(24.33) \end{bmatrix}$$


24.33

Figura 7: Struttura delle matrici R_{tilt} e R_{pan}

Viste le premesse fatte è chiaro che la matrice(R_{pt}), dovuta alla rotazione della telecamera, sia costruita moltiplicando in questo ordine le matrici trovate $R_{pt} = R_{pan}R_{tilt}$. Tale matrice verrà postmoltiplicata per la matrice di rotazione(R_{mondo}) tra il riferimento standard della telecamera a riposo e il riferimento mondo più vicino per tenere sempre calibrata la rete. Avendo svincolato i riferimenti tra le singole telecamere, la rete resta calibrata anche in seguito alla rottura di una o più telecamere.

13 Conclusioni e sviluppi futuri

In questo progetto abbiamo effettuato la calibrazione di una rete di videocamere allo scopo di ottenerne la posizione quanto più precisa possibile. Il metodo proposto consiste nel calcolo della matrice fondamentale tra coppie di videocamere, da tale matrice si può ottenere la rototraslazione relativa tra 2 camere di ogni coppia e infine sfruttando le informazioni sulle videocamere note abbiamo calibrato globalmente la rete. Per ottenere la stima più accurata possibile abbiamo applicato algoritmi di minimizzazione nel calcolo della matrice fondamentale e abbiamo operato una scelta opportuna dell'ordine con cui calibrare le videocamere della rete. Infine abbiamo proposto un metodo per calcolare la baseline tra due videocamere per fissare il fattore di scala che caratterizza la matrice di rototraslazione.

Possibili sviluppi del metodo riguardano l'estensione al caso distribuito, in cui non si hanno a disposizione tutti i frame raccolti da tutte le videocamere. Ogni videocamera otterrà i frame solo dalle videocamere con cui è in grado di scambiare informazione; è ragionevole supporre che il metodo da noi proposto, basandosi sul calcolo della posizione relativa tra coppie di videocamere, possa essere esteso con relativa semplicità al caso distribuito. Un ulteriore sviluppo potrebbe essere lo studio di videocamere PTZ nel dettaglio. Nel capitolo precedente abbiamo già esposto alcune considerazioni in proposito, ma è mancato il tempo materiale di approfondire e sviluppare l'argomento.

A Calcolo della matrice fondamentale

Quando un punto w in una scena viene ripreso da due videocamere diverse (o dalla stessa in diverse posizioni) viene proiettato nei frame delle due videocamere nei punti m_1 e m_2 . La corrispondenza tra questi due punti è data dalla matrice fondamentale F , e la relazione è data dall'equazione $m_1^T F m_2 = 0$.

Per capire meglio in che modo si trova questa corrispondenza prendiamo a riferimento l'immagine 8:

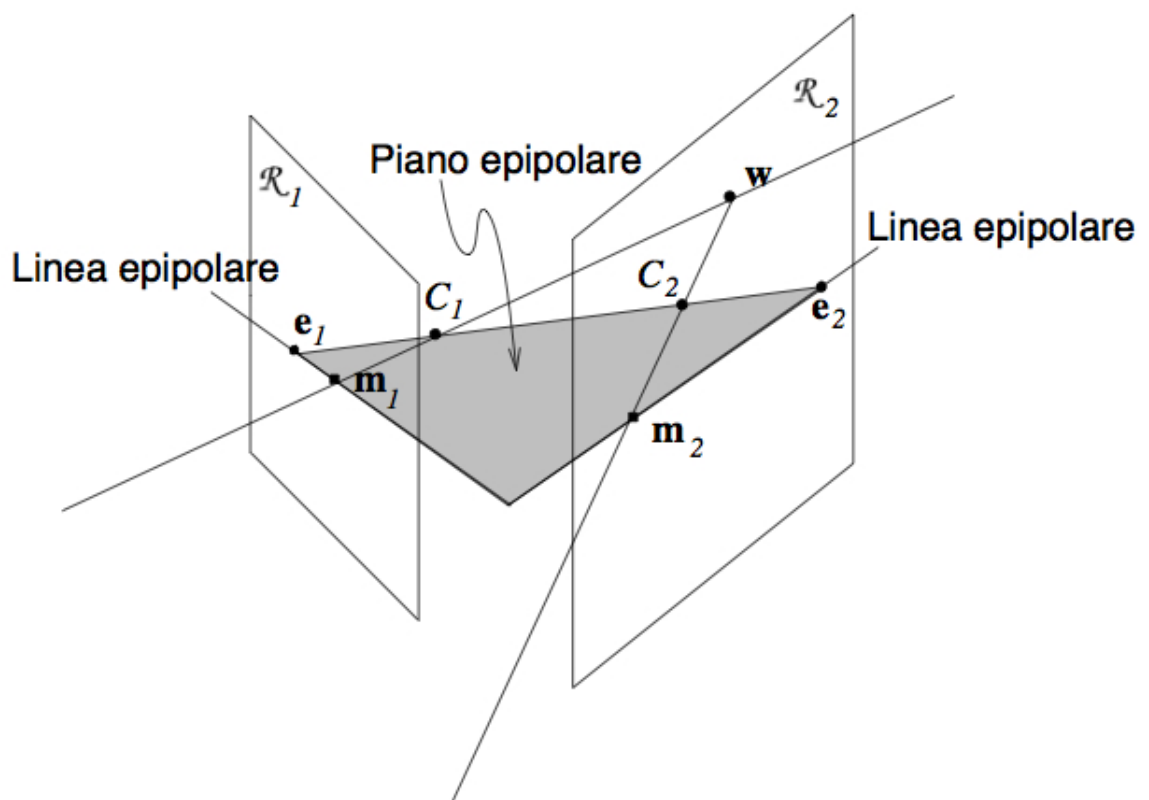


Figura 8: Schema videocamere

Sappiamo che ogni punto m_2 sul secondo frame, corrispondente ad m_1 nel primo dovrà giacere sulla linea epipolare e_2 come si evince dall'immagine. Tale linea è la proiezione sul secondo frame del raggio dal punto m_1 attraverso il centro della videocamera. Il modo in cui m_2 viene mappato sulla linea è appunto

rappresentato dalla matrice F . Le potenzialità dell'uso della matrice fondamentale risiedono nel fatto che si può caratterizzare soltanto usando delle corrispondenze tra punti nei frame, senza ulteriori informazioni, che è esattamente la situazione in cui ci troviamo, dal momento che le ulteriori informazioni derivanti dal conoscere tutti i parametri di alcune telecamere non portano informazione alle videocamere non calibrate, se non dopo aver scoperto qualcosa sulla loro posizione relativa a camere calibrate. Elenchiamo alcune proprietà di F che saranno utili in seguito per capire il metodo di calcolo della matrice:

- F è una matrice di rango 2 con 7 gradi di libertà.
- Valgono le relazioni $F\mathbf{e}_1 = \mathbf{0}$ $F^T\mathbf{e}_2 = \mathbf{0}$ per gli epipoli.
- Relazioni rispetto alle matrici di proiezione:

$$F = [\mathbf{e}_2]^T \times \mathbf{P}_2 \mathbf{P}_1^{-1}$$

Il metodo numerico per il calcolo della matrice F utilizzato è quello descritto da Hartley e Zisserman in [articolo].

Il punto di partenza è l'equazione

$$\mathbf{x}^T \mathbf{F} \mathbf{x}' = 0$$

Scrivendo poi i punti come $\mathbf{x} = (x, y, 1)^T$ e $\mathbf{x}' = (x', y', 1)^T$ si vede che ogni corrispondenza origina un'equazione lineare con incognite le componenti della matrice F , nello specifico:

$$x'x f_{11} + x'y f_{12} + x' f_{13} + y'x f_{21} + y'y f_{22} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0$$

Se ora definiamo $\mathbf{f} = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})$ possiamo da n corrispondenze trovare un sistema di equazioni nella forma:

$$\mathbf{A} \mathbf{f} = \begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = 0$$

Perché esista una soluzione \mathbf{A} deve avere al massimo rango 8 e se il rango è esattamente 8 allora la soluzione è unica (a meno di un fattore di scala). Se il

rango (a causa di errori) è maggiore di 8 si cercano soluzioni con metodi ai minimi quadrati.

Va notato che la matrice F deve avere rango 2 (molte importanti proprietà derivano da questo vincolo), e quindi essere singolare. Le matrici che si trovano risolvendo sistemi di equazioni lineari non sono in generale di rango 2, perciò tale vincolo va imposto a posteriori. Il modo più conveniente è di correggere la matrice F trovata dopo la scomposizione SVD (vedere più avanti per i vari step dell'algoritmo) e sostituirla con la matrice che minimizza la norma di Frobenius e ha rango 2.

Per il buon funzionamento dell'algoritmo degli 8 punti è fondamentale la normalizzazione dei dati di ingresso, in questo caso si tratta di traslare e scalare i punti in modo opportuno. Questa operazione comporta una complessità aggiuntiva irrilevante, ma dà risultati significativamente migliori. Per dettagli sulla normalizzazione si rimanda al testo originale.

Algoritmo

1. **Normalizzazione:** Trasformare le coordinate dei punti tramite una traslazione e un fattore di scala.
2. **Soluzione lineare:** Determinare il vettore f dall'equazione A come il vettore corrispondente al più piccolo valore singolare di A . Da questo determinare la matrice.
3. **Vincolo** Sostituire F con la matrice \hat{F} di rango due più vicina in norma di Frobenius.
4. **Denormalizzare** Invertire il processo di normalizzazione applicato ai punti.

B Funzione di costo

Supponiamo che nessuno dei punti di corrispondenza coincida con un epipolo, in queste ipotesi semplifichiamo l'analisi applicando una trasformazione per portare \mathbf{x} e \mathbf{x}' nell'origine $(0, 0, 1)^T$ in coordinate omogenee, cambiamo inoltre gli epipoli con le coordinate $(1, 0, f)$ e $(1, 0, f')$ rispettivamente. Applicare queste trasformazioni non ha influenza sulla minimizzazione della funzione costo 7 e quindi nemmeno sul problema di minimizzazione. In queste ipotesi la matrice fondamentale assume la forma:

$$F = \begin{bmatrix} ff'd & -f'd & -f'd \\ -fb & a & b \\ -fd & c & d \end{bmatrix}. \quad (11)$$

Consideriamo ora una linea epipolare (che denoteremo $\mathbf{l}(t)$) nella prima immagine che passi per il punto $(0, t, 1)^T$ e per l'epipolo $(1, 0, f)^T$. Il vettore che rappresenta la linea sarà il risultato del prodotto $(0, t, 1) \times (1, 0, f) = (tf, 1, -t)$ e quindi la distanza al quadrato della linea dall'origine è:

$$d(\mathbf{x}, \mathbf{l}(t))^2 = \frac{t^2}{1 + (tf)^2}$$

usiamo poi la matrice fondamentale per trovare l'altra linea epipolare in coordinate omogenee:

$$\mathbf{l}'(t) = F(0, t, 1)^T = (-f'(ct + d), at + b, ct + d)^T.$$

Quindi la distanza di questa linea dall'origine è data dalla formula:

$$d(\mathbf{x}, \mathbf{l}'(t))^2 = \frac{(ct + d)^2}{(at + b)^2 + f'^2(ct + d)^2}$$

Quindi la distanza totale delle 2 rette è

$$s(t) = \frac{t^2}{1 + f^2 t^2} + \frac{(ct + d)^2}{(at + b)^2 + f'^2 (ct + d)^2}$$

funzione che deriveremo per trovare il minimo.

$$s'(t) = \frac{2t}{(1 + f^2 t^2)^2} - \frac{2(ad - bc)(at + b)(ct + d)}{((at + b)^2 + f'^2 (ct + d)^2)^2}$$

ponendo uguale a zero otteniamo infine l'espressione

$$g(t) = t((at + b)^2 + f'^2 (ct + d)^2)^2 - (ad - bc)(1 + f^2 t^2)^2 (at + b)(ct + d)$$

C Calcolo Omografia

La H si trova tramite l'algoritmo DLT utilizzando le corrispondenze tra i punti della testa e dei piedi nei frame. Quindi rimangono da determinare il vanishing point e la vanishing line. Il vanishing point è definito nel seguente modo: data una direzione d e il piano all'infinito π_{∞} , allora è semplicemente l'immagine sul piano della telecamera, del punto di intersezione tra tutte le linee con direzione d e il piano all'infinito.

Nel nostro caso le linee vengono rappresentate dai elementi ortogonali. Ora possiamo utilizzare tutte queste rette nel piano 3D e verificare che nel piano immagine si intersecano nel vanishing point. Bisogna sempre prendere una coppia di rette con la stessa direzione per trovare un'intersezione. Siccome abbiamo a disposizione parecchie coppie di rette troveremo una accumulazione di vanishing point. Per scegliere quello ottimo e fare una preselezione sulle coppie da considerare usiamo l'algoritmo RANSAC. Nell'immagine 10 si vede bene il procedimento descritto.

Adesso passiamo al calcolo della vanishing line. Dato un insieme di piani paralleli P nello spazio 3D, essi si intersecano nel piano all'infinito in una linea, l'immagine di questa linea è la vanishing line. Il piano delle teste e quello dei piedi rappresentano la coppia di piani paralleli necessari. Per determinare il piano delle teste, si considerano tre punti medi della testa in tre posizioni diverse e si connettono queste linee tra di loro. La stessa cosa si fa per il piano dei piedi. Prolungando ora queste sei rette, ogni coppia di linee di ogni retta si interseca in tre punti precisi. Connettendo questi tre si trova la vanishing line. Siccome anche in questo caso abbiamo parecchi punti a disposizione utilizziamo l'algoritmo RANSAC per la miglior stima della vanishing line.

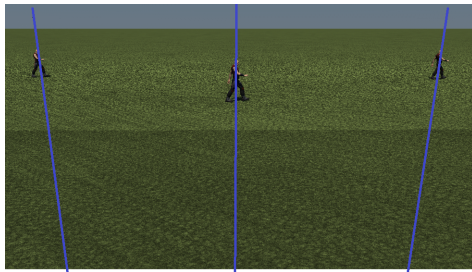


Figura 9: Vanishing Point

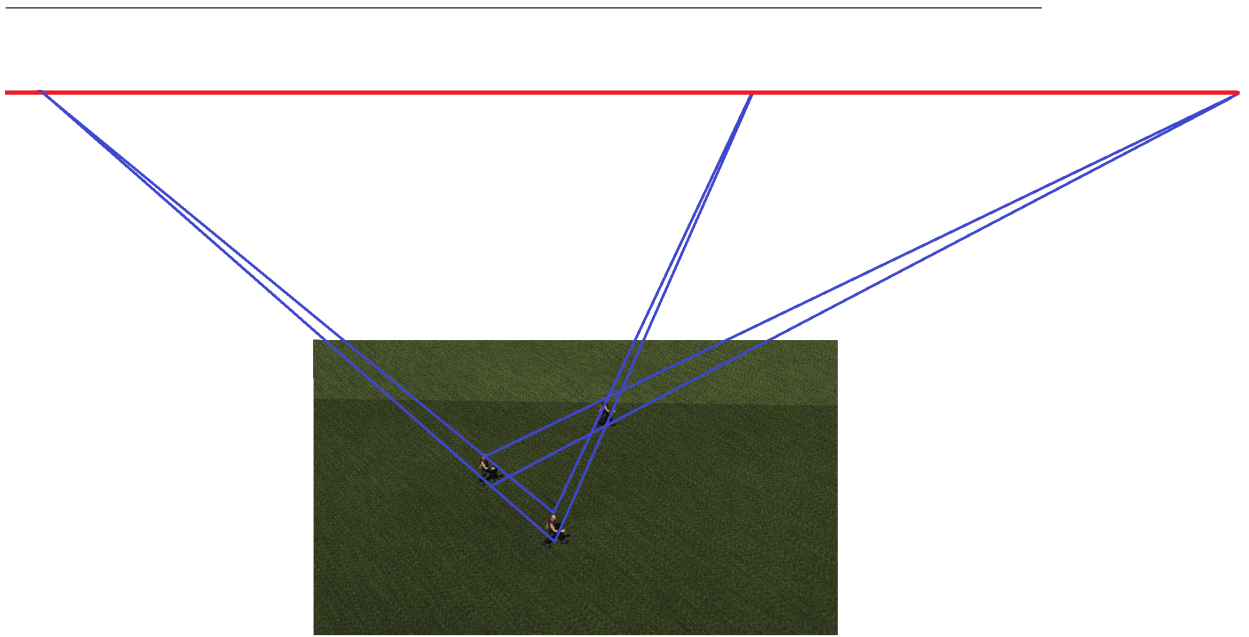


Figura 10: Vanishing Line

D Dimostrazione H

Per la dimostrazione supponiamo di avere a disposizione una seconda telecamera fittizia. Questa avrà una matrice di proiettiva $P = K[I|0]$ invece quella del nostro sistema di prova ha $P' = K'[R|t]$. Inoltre consideriamo un piano $\pi = (n^T, d^T)$ con normale n^T e z la distanza tra la telecamera fittizia e il piano. Quindi per la formula dell'omografia indotta da un piano segue

$$H = K' \left(R - \frac{1}{z} * t * n^T \right) K^{-1}$$

Dallo studio dei parametri intrinseci si ricava

$$v = h/d$$

con v la distanza focale, h altezza del segmento ortogonale ed $d = z$ l'altezza della telecamera. Inoltre ricordando che la seconda telecamera è virtuale, vale $K = K'$, $R = I$ e $h = tn^T$, quindi risulta

$$H = K(I - hz)K^{-1}$$

Infine sapendo che vale che la vanishing line $l_\infty = K^{-T}n$ e il vanishing point $v_\infty = Kn$ troviamo la equazione

$$H = I - \frac{h v_\infty (l_\infty)^T}{z (v_\infty)^T l_\infty}$$

Dimostriamo ora gli ultimi due risultati utilizzati: $l_\infty = K^{-T}n$ e $v_\infty = Kn$. Partiamo dal primo:

Lemma: L'immagine di una linea l e il centro della telecamera definiscono un piano che ha normale uguale a $n = K^T l$. La correttezza di questo enunciato la verifichiamo geometricamente. Un punto x retroproiettato ha direzione $d = K^{-1} * x$ (perché nello spazio una linea è univocamente definita da un punto e una direzione) che è ortogonale alla normale del piano $d^T n = x^T K^{-T} n = 0$. Dato che i punti giacciono sulla linea l soddisfano $x^T l = 0$ segue che $l = K^{-T} n$; invertendo l'equazione si ricava facilmente che $n = K^T l$.

Ora verifichiamo la validità della seconda equazione. Sappiamo che avendo dato una linea con direzione d , il vanishing point corrispondente è $v = Kd$. Notiamo subito che quindi che ha la stessa direzione dei segmenti ortognali al piano . Per questo si può assumere che $d = n$ perché sono paralereli alla normale di . Si trova il risultato cercato: $v_\infty = Kn$.

E Trasformata di Hough

La trasformata di Hough è una tecnica che permette il riconoscimento di configurazioni globali presenti in una immagine (segmenti, curve, forme prestabilite), e si basa sulla trasformazione di tutti i punti costituenti una immagine, in punti di un nuovo spazio detto spazio dei parametri.

Nella sua versione tradizionale, la trasformata di Hough si applica ad immagini binarie, ovvero immagini in cui sono presenti due soli livelli, bianco e nero, ed in cui l'informazione associata ad un punto è rappresentata unicamente dalla sua posizione. Questa tecnica, ideata da Hough nel 1962, è basata sulla validazione delle ipotesi in cui, definita la curva che si intende cercare nella scena, per ogni punto dell'immagine si calcolano i parametri di tutte le curve che potrebbero passare per quel punto e si incrementano le celle di uno spazio n -dimensionale (con n numero dei parametri) che corrispondono alle varie curve. Si ottiene così una funzione di accumulazione definita nello spazio dei parametri. Alla fine saranno i massimi di questa funzione, ovvero i punti nello spazio dei parametri che hanno accumulato il maggior numero di voti a rappresentare le curve che hanno probabilità elevata di essere presenti nell'immagine, come se si trattasse di una ipotesi avvalorata dal maggior numero di conferme sperimentali. Una delle caratteristiche più interessanti della Trasformata di Hough è quella di non risentire del rumore presente in una immagine. Infatti, l'altezza dei picchi nella matrice di accumulazione, dipende in maniera del tutto trascurabile da eventuali lacune presenti nella retta di partenza o dall'esistenza di punti spuri presenti nello spazio immagine.

Riferimenti bibliografici

- [1] Hartley, R. I. and Zisserman, A., Multiple View Geometry in Computer Vision, 2004
- [2] Funiak S., Guestrin C., Paskin M., Sukthankar R. - Distributed Localization of Networked Cameras
- [3] Hartley, R. I. In defence of the 8-point algorithm, 1995
- [4] L. Van Gool, M. Proesmans, A. Zisserman, Planar homologies as a basis for grouping and recognition, 1998.
- [5] C. Springer, Geometry and analysisi of projective spaces, Freeman, 1964.
- [6] J. Meidow, Calibration of stationary cameras by observing objects of equal heights on a ground plane, International Society for Photogrammetry and Remote Sensing, 2004.