Stima dei parametri

Questo esercizio prevede la calibrazione manuale di un sistema di camere usando un oggetto di geometria nota a priori come riferimento.
Lo script da eseguire è **esercizio2ofass2.m**.
Noi sappiamo che un punto 3D viene proiettato nel piano del sensore della camera mediante una moltiplicazione matriciale la cui forma è $P * X_{3D} = X_{2D}$.
La calibrazione consiste nel ricavarsi appunto la matrice P, conoscendo un insieme di corrispondenze ($X_{3D}^{(i)}$, $X_{2D}^{(i)}$).
Ogni corrispondenza è regolata dalle seguenti due equazioni:

$$\frac{x^{(i)}}{w^{(i)}} = \frac{m_{11}X_W^{(i)} + m_{12}Y_W^{(i)} + m_{13}Z_W^{(i)} + m_{14}}{m_{31}X_W^{(i)} + m_{32}Y_W^{(i)} + m_{33}Z_W^{(i)} + m_{34}}$$

$$\frac{y^{(i)}}{w^{(i)}} = \frac{m_{21}X_W^{(i)} + m_{22}Y_W^{(i)} + m_{23}Z_W^{(i)} + m_{24}}{m_{31}X_W^{(i)} + m_{32}Y_W^{(i)} + m_{33}Z_W^{(i)} + m_{34}}$$

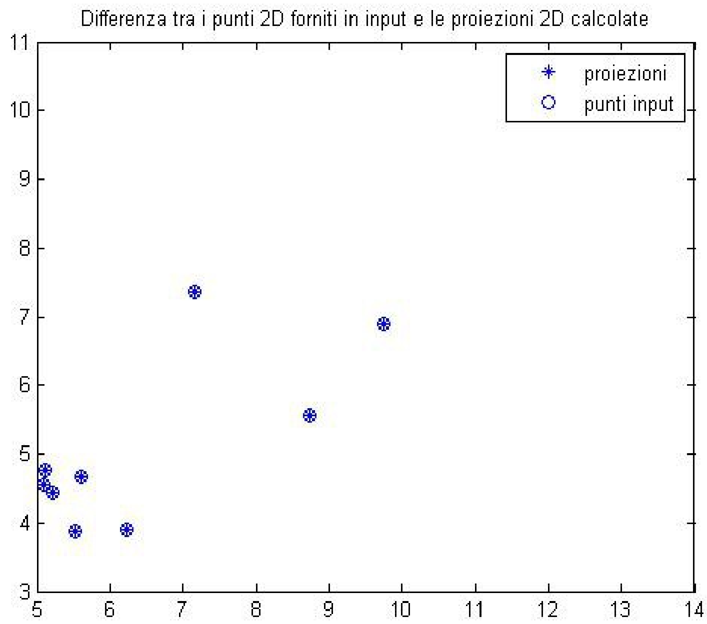
Da esse possiamo ricavare due righe del sistema lineare (le cui 12 incognite sono i coefficienti di P) e possiamo porre w(i) = 1 a causa della conversione diretta da sistema cartesiano a omogeneo.

Le due righe del sistema lineare di incognite P associate ad una corrispondenza i sono:

```
A(j,:) = [0,0,0,0,-world(1,i),-world(2,i),-world(3,i),-1,image(2,i)*world(1,i),image(2,i)*world(2,i),image(2,i)*world(3,i),image(2,i)];
```

```
A(j+1,:) = [-world(1,i),-world(2,i),-world(3,i),-1,0,0,0,0,image(1,i)*world(1,i),image(1,i)*world(2,i),image(1,i)*world(3,i),image(1,i)];
```

Essendo il sistema lineare sovradeterminato, possiamo usare il metodo numerico SVD per ricavare l'autovettore che minimizza A e col successivo "reshaping" dello stesso, ho ricavato la matrice P.
Ho verificato la correttezza del mio operato mostrando con un grafico che i punti 2D dati in input e le proiezioni dei punti 3D di input tramite la matrice P coincidono.



Ora dobbiamo ricavarci C, cioè il vettore delle coordinate del centro di proiezione. Per fare questo, osserviamo che P*C = 0, quindi C sta nel null-space di P che può essere ricavato ancora una volta tramite l'operatore SVD.
L'output di SVD è l'insieme delle 3 matrici U,S,V: se **r** è il rango di U e **n** quello di P, il nostro null-space sarà costituito dalle ultime **n-r** colonne di V. Nel nostro caso, ci basta prendere l'ultima colonna di V.

Per sicurezza, ho confrontato l'output di questa procedura con quello del comando **`null(P)`** che appunto restituisce il null-space di una matrice.

Inoltre ho verificato che **P*C ≈ 0** (non può venire esattamente zero a causa dei metodi numerici usati in Matlab, come del resto non viene esattamente zero neanche facendo **P*null(P)**).

A questo punto, convertiamo di nuovo le coordinate C in coordinate cartesiane, dividendo tutto per w_c e abbiamo concluso il nostro esercizio.

Riporto i dati esattamente computati da Matlab:

Matrice P:

0,127000127000190	0.254000254000382	0,381000381000573	0,508000508000760
0,508000508000763	0,381000381000572	0,254000254000383	0,127000127000188
0,127000127000190	0	0,127000127000191	-2,22044604925031e-16

Coordinate C (convertite nel sistema cartesiano):

[0,999999999999996, -0,999999999999991, -0,999999999999994]