Davide Zanin 1035601

# ELABORAZIONE DI DATI TRIDIMENSIONALI – HOMEWORK 1

Sommario

[ELABORAZIONE DI DATI TRIDIMENSIONALI – HOMEWORK 1 1](#_Toc323133312)

[1. Procedura utilizzata 1](#_Toc323133313)

[1.1. Calibrazione stereo 1](#_Toc323133314)

[1.2. Image undistort e rectify 3](#_Toc323133315)

[1.3. Disparity map 3](#_Toc323133316)

[1.4. Range image 3](#_Toc323133317)

[1.5. Generazione della point cloud a partire dalla disparity map 4](#_Toc323133318)

[1.6. Visualizzazione della point cloud 4](#_Toc323133319)

[2. Parametri ottenuti per la calibrazione della telecamera stereo 4](#_Toc323133320)

[2.1. Re-projection error 4](#_Toc323133321)

[2.2. Matrici ottenute 4](#_Toc323133322)

[3. Immagini ottenute 5](#_Toc323133323)

[3.1. Immagini scattate 5](#_Toc323133324)

[3.2. Immagini undistort 5](#_Toc323133325)

[3.3. Immagini rettificate 6](#_Toc323133326)

[3.4. Disparity map 6](#_Toc323133327)

[3.5. Range immagine 7](#_Toc323133328)

[3.6. Point cloud 7](#_Toc323133329)

## Procedura utilizzata

### Calibrazione stereo

Per la calibrazione della fotocamera (sinistra, destra e poi stereo) sono state usate le foto riportate nel file “input\_photo\_sx.txt” e nel file “input\_photo\_dx.txt”. Affinché il programma carichi correttamente le immagini queste devono trovarsi nella directory “calibration\_set”.

Le foto vengono caricate una ad una e tramite la funzione findChessboardCornersvengono individuati i corner. Questa funzione vuole come input l'immagine in cui ricercare i corner, un oggetto di tipo Size che specifica la dimensione delle pattern dei corner da trovare (nel nostro caso 7x5), un vector di Point2f in cui verranno inseriti i corner trovati e un flag. I flag utilizzati sono i seguenti:

* CV\_CALIB\_CB\_ADAPTIVE\_THRESH: fa in modo che venga usata una soglia adatta per la conversione dell’immagine in bianco e nero. Questa soglia è calcolata sulla base della luminosità dell’immagine;
* CV\_CALIB\_CB\_NORMALIZE\_IMAGE, in questo modo la gamma dell’immagine viene normalizzata prima di applicare la soglia per la conversione dell’immagine in bianco e nero.

La funzione findChessboardCornersritorna inoltre un booleano che specifica se i corner sono stati individuati correttamente.

Per migliorare la precisione nel posizionamento dei corner è stata usata la funzione cornerSubPix passando come parametri l'immagine usata nella funzione precedente (convertita in scala di grigi), il vettore di corner trovati e altri parametri impostanti secondo quanto riportato nella documentazione di OpenCV, cioè

* winSize = Size(11, 11), specifica la dimensione della finestra di ricerca;
* zeroZone = Size(-1, -1), specifica un zona all’interno della finestra di ricerca in cui non deve essere effettuata la ricerca. In questo caso, con questo valore si specifica che tale zona non è definita;
* crit = TermCriteria(CV\_TERMCRIT\_ITER+CV\_TERMCRIT\_EPS, 30, 0.01)), specifica quando deve terminare la ricerca del corner. Con questo valore si specifica che la ricerca termini dopo 30 iterazioni oppure dopo aver raggiunto un’accuratezza di 0.01.

Dopo che sono state analizzate tutte le foto è stata utilizzata la funzione calibrateCamerache restituisce la matrice con i parametri della fotocamera e il vettore dei coefficienti di distorsione. Questa funzione è stata chiamata due volte, una volta per la fotocamera sinistra e una volta per la fotocamera destra, quindi alla fine si avranno due matrici con i parametri della fotocamera (M0 per la fotocamera sinistra e M1 per la fotocamera sinistra) e due vettori con i coefficienti di distorsione (D0 per la fotocamera sinistra e D1 per la fotocamera destra). Le altre matrici calcolate da calibrateCamera non vengono usate per le elaborazioni successive e per questo motivo non vengono menzionate.

Con le immagini presi in esame il re-projection error è pari a 0.663745 per la fotocamera sinistra e di 0.606275 per la fotocamera destra. I valori delle matrici delle fotocamera e dei coefficienti di distorsione si possono trovare nella cartella “value” o nel paragrafo 2.

A questo punto è stata eseguita la calibrazione stereo usando la funzione stereoCalibrate a cui vengono passati i seguenti parametri:

* objectPoints, è il vettore con le coordinate nel mondo reale dei corner. Queste coordinate sono uguali per tutti i punti;
* imagePoints0 e imagePoints1, sono i vettori contenenti le coordinate in pixel dei corner, rispettivamente della fotocamera sinistra e destra;
* M0, D0, M1 e D1, sono le matrici calcolate al punto precedente;
* RR è la matrice di rotazione dal sistema di coordinate della fotocamera sinistra a quella destra;
* T è il vettore di traslazione tra i sistemi di coordinate delle due fotocamere;
* TermCriteria(CV\_TERMCRIT\_ITER+CV\_TERMCRIT\_EPS, 100, 1e-5), in questo caso il criterio di terminazione è impostato a 100 iterazioni oppure dopo aver raggiunto un’accuratezza di 0,00001.
* CV\_CALIB\_FIX\_INTRINSIC, specifica che i valori delle matrici delle fotocamere e dei coefficienti di distorsione sono già stati calcolati.
* E, F sono l’essential matrix e la fondamental matrix (non usate per le elaborazioni successive).

Il re-project error ottenuto è pari a 0.637381.

Successivamente con la funzione stereoRectify sono state calcolate le matrici di rotazione per le due fotocamere R0 e R1, le matrici di proiezione nel sistema di coordinate rettificate P0 e P1 e la matrice disparity-to-depth Q. Il flag è stato impostato al valore CV\_CALIB\_ZERO\_DISPARITY, in questo modo la funzione fa in modo che, nella vista rettificata, i punti principali delle immagini delle due fotocamere abbiano le stesse coordinate in pixel. Infine il valore di alpha è stato impostato a 0 in modo che le immagini rettificate vengono zoomate e shiftate in modo da contenere solo pixel validi.

Come ultima operazione è stata chiamata due volte al funzione initUndistortRectifyMap in modo da inizializzare le matrici di undistortion e rectification per entrambe le fotocamere (mx0, my0, mx1, my1). Come tipo per le matrici mx0 e mx1 è stato scelto CV\_16SC2.

Con quest’ultima operazione si conclude la fase di calibrazione.

### Image undistort e rectify

Per rettificare un immagine è sufficiente usare la funzione remap passando come parametri l’immagine da rettificare, l’immagine di output e le matrici mx e my della fotocamera da cui è stata scattata la foto e un flag che specifica il metodo di interpolazione da utilizzare. Questo flag è stato impostato a INTER\_LINEAR in modo da usare l’interpolazione bilineare.

### Disparity map

Il calcolo della disparity map è stato effettuato a partire da una foto scattata dalla fotocamera sinistra e una dalla fotocamera destra. Dopo averle rettificate è stata usata la classe StereoSGBM che usa l’algoritmo semi-global block matching per il calcolo delle corrispondenze stereo nelle due immagini. I parametri della classe sono stati impostati come segue:

* preFilterCap = 4;
* SADWindowSize = 9;
* P1 = 8\*cn\*sgbm.SADWindowSize\*sgbm.SADWindowSize;
* P2 = 32\*cn\*sgbm.SADWindowSize\*sgbm.SADWindowSize;
* minDisparity = 16;
* numberOfDisparities = 16\*8;
* uniquenessRatio = 5;
* speckleWindowSize = 200;
* speckleRange = 32, questo valore e quello precedente permettono di ridurre il rumore dell’immagine di output agendo sui valori di filtraggio.
* disp12MaxDiff = -1;
* fullDP = 1, in questo modo l’elaborazione richiede più tempo ma si ottengono risulatti migliori.

dove cn è il numero di canali dell’immagine, nel nostro caso 1. Partendo dai valori di default si è cercato di individuare i valori per i quali risultasse migliore l’output.

A questo punto è sufficiente passare all’oggetto di tipo StereoSGBM le due immagini rettificate e un immagine di output per ottenere la disparity map.

### Range image

Per il calcolo della range image è stata usata la funzione reprojectImageTo3D che data la disparity map e la matrice Q fornisce una matrice che contiene le coordinate 3D di ogni punto della disparity map. Inoltre è stato specificato alla funzione che gestisca i pixel con disparity minima impostando il valore di z ad un numero molto alto (10000).

In seguito è stata creata un immagine a toni di grigio in cui il nero corrispondesse ai punti con valori di z minori e il bianco a valori di z maggiori.

### Generazione della point cloud a partire dalla disparity map

Per la generazione delle coordinate (x, y, z) dei punti della disparity map è stata usata la funzione reprojectImageTo3D come descritto nel paragrafo precedente.

Per colorare i punti (x, y, z) della cloud è stata anche caricata l’immagine sinistra da cui è stata calcolata la disparity map e, una volta che è stata rettificata, a ciascun punto (x, y, z) veniva assegnato il colore del pixel (x, y) della foto rettificata.

I punti con valore 10000 in z non sono stati inseriti.

### Visualizzazione della point cloud

Per visualizzare la point cloud è stato sufficiente creare un oggetto viewer di tipo PCLVisualizer che visualizzerà la nuvola di punti e un oggetto rgb di tipo PointCloudHandlerRGBField che gestisce la colorazione dei punti della point cloud. A questo punto la point cloud e rgb sono stati aggiunti a viewer e ne è stata avviata la visualizzazione con il metodo spin.

## Parametri ottenuti per la calibrazione della telecamera stereo

### Re-projection error

* Calibrazione fotocamera sinistra: 0.663745
* Calibrazione fotocamera destra: 0.606275
* Calibrazione stereo: 0.637381

### Matrici ottenute

Le matrici sotto riportate si possono trovare nella cartella “value”.

#### Matrici fotocamera sinistra

#### Matrici fotocamera destra

#### Matrici di calibrazione stereo

#### Matrici di rotazione fotocamera sinistra

#### Matrici di rotazione fotocamera destra

#### Matrice disparity-to-depth

## Immagini ottenute

Le immagini sotto riportate si possono trovare in dimensione originale nella cartella “result”.

### Immagini scattate

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Fotocamera sinistra | Fotocamera Destra |  |

### Immagini undistort

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Fotocamera sinistra | Fotocamera Destra |  |

### Immagini rettificate

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Fotocamera sinistra | Fotocamera Destra |  |

### Disparity map



### Range immagine



### Point cloud

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |