Davide Zanin 1035601

ELABORAZIONE DI DATI TRIDIMENSIONALI – HOMEWORK 1

Sommario

1. Procedura utilizzata

1.1. 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 `findChessboardCorners` vengono 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 `findChessboardCorners` ritorna 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 `calibrateCamera` che 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.

1.2. 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.

1.3. 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.

1.4. 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.

1.5. 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.

1.6. 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`.

2. Parametri ottenuti per la calibrazione della telecamera stereo

2.1. Re-projection error

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

2.2. Matrici ottenute

Le matrici sotto riportate si possono trovare nella cartella "value".

2.2.1. Matrici fotocamera sinistra

$$M0 = \begin{bmatrix} 818 & 0 & 500 \\ 0 & 817 & 401 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D0 = [-0.36 \quad 0.12 \quad 2.14 \cdot 10^{-4} \quad 6.68 \cdot 10^{-4} \quad 0.14]$$

2.2.2. Matrici fotocamera destra

$$M1 = \begin{bmatrix} 827 & 0 & 520 \\ 0 & 827 & 392 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D1 = [-0.32 \quad 6.86 \cdot 10^{-2} \quad -4.20 \cdot 10^{-4} \quad 1.02 \cdot 10^{-3} \quad 7.67 \cdot 10^{-2}]$$

2.2.3. Matrici di calibrazione stereo

$$RR = \begin{bmatrix} 1 & 1.37 \cdot 10^{-3} & -6.44 \cdot 10^{-3} \\ -1.37 \cdot 10^{-3} & 1 & 8.05 \cdot 10^{-4} \\ 6.44 \cdot 10^{-3} & -7.97 \cdot 10^{-4} & 1 \end{bmatrix}$$

$$T = [-12 \quad -1.22 \cdot 10^{-2} \quad 0.53]$$

2.2.4. Matrici di rotazione fotocamera sinistra

$$R0 = \begin{bmatrix} 1 & 2.43 \cdot 10^{-3} & -5.06 \cdot 10^{-2} \\ -2.41 \cdot 10^{-3} & 1 & 4.48 \cdot 10^{-4} \\ 5.06 \cdot 10^{-2} & -3.26 \cdot 10^{-4} & 1 \end{bmatrix}$$

$$P0 = \begin{bmatrix} 767 & 0 & 559 & 0 \\ 0 & 767 & 395 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

2.2.5. *Matrici di rotazione fotocamera destra*

$$R1 = \begin{bmatrix} 1 & 1.01 \cdot 10^{-3} & -4.41 \cdot 10^{-2} \\ -1.03 \cdot 10^{-3} & 1 & -3.64 \cdot 10^{-4} \\ 4.41 \cdot 10^{-2} & -4.09 \cdot 10^{-4} & 1 \end{bmatrix}$$

$$P1 = \begin{bmatrix} 767 & 0 & 559 & -9.21 \cdot 10^{3} \\ 0 & 767 & 395 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

2.2.6. *Matrice disparity-to-depth*

$$Q = \begin{bmatrix} 1 & 0 & 0 & -559 \\ 0 & 1 & 0 & -394 \\ 0 & 0 & 0 & 767 \\ 0 & 0 & -8.36 \cdot 10^{-2} & 0 \end{bmatrix}$$

3. Immagini ottenute

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

3.1. Immagini scattate

Fotocamera sinistra

Fotocamera Destra

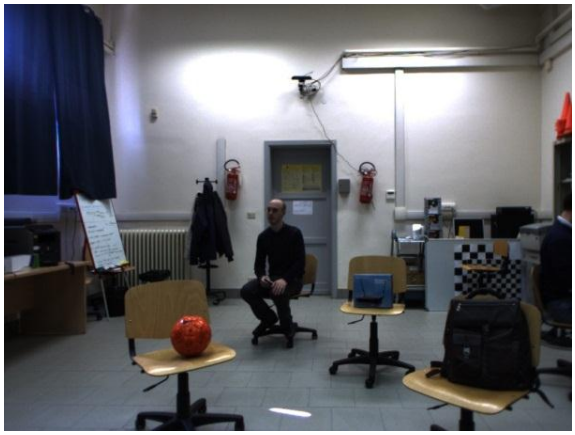
3.2. Immagini undistort

Fotocamera sinistra

Fotocamera Destra

3.3. Immagini rettificate

Fotocamera sinistra



Fotocamera Destra

3.4. Disparity map

3.5. Range immagine

3.6. Point cloud