

MODEL KALIBRASI KAMERA UNTUK PENGUKURAN JARAK OBJEK DAN NAVIGASI ROBOT

Budi Rahmani¹, Hugo Aprilianto²

*Program Studi Teknik Informatika, STMIK Banjarbaru, Kalimantan Selatan^{1,2}
budirahmani@gmail.com¹, hugo.aprilianto@gmail.com²*

ABSTRACT

Camera calibration is an important process in computer vision. The calibration process is done, especially if the camera will be used to estimate the distance of the camera to a particular object. This paper delivered one of the processes that can be used to calibrate the camera either mono or stereo. Steps taken to mono cameras include image acquisition, arithmetic and search parameters intrinsic, extrinsic count and search parameters. The experimental results show that the process is carried out have been able to produce calibration parameters can then be applied to the image to be processed later, either in real-time or not.

Keywords: *article, paper, conference (number of keywords not more than 5 keywords)*

ABSTRAK

Kalibrasi kamera merupakan proses yang penting dalam computer vision. Proses kalibrasi dilakukan terutama jika kamera akan digunakan untuk memperkirakan jarak kamera terhadap objek tertentu. Paper ini menyampaikan salah satu proses yang dapat digunakan dalam mengkalibrasi kamera baik mono maupun stereo. Langkah yang dilakukan untuk kamera mono diantaranya adalah akuisisi citra, hitung dan cari parameter intrinsic, hitung dan cari parameter ekstrinsik. Hasil percobaan menunjukkan bahwa proses yang dilakukan telah mampu menghasilkan parameter kalibrasi yang kemudian dapat diterapkan pada citra yang akan diproses kemudian, baik secara real-time maupun tidak.

Keywords: *article, paper, conference (number of keywords not more than 5 keyword)*

PENDAHULUAN

Kalibrasi kamera merupakan proses yang penting dalam *computer vision* (Zhang, 2000). Kalibrasi kamera adalah proses untuk memperkirakan parameter lensa dan sensor citra dari sebuah kamera. Berdasarkan parameter yang didapat, maka dapat dilakukan diantaranya: perbaikan citra akibat *lens distortion*, pengukuran *size of an object* (dimensi dan jarak), dan penentuan posisi objek terhadap kamera. Proses kalibrasi dilakukan terutama jika kamera akan digunakan untuk memperkirakan jarak kamera terhadap objek tertentu. Paper ini menyampaikan salah satu proses yang dapat digunakan dalam mengkalibrasi kamera baik mono maupun stereo.

METODE PENELITIAN

Penulis menggunakan perangkat lunak Matlab 2011b untuk membangun antarmuka kalibrasi kamera ini. Kamera yang dikabrasi adalah kamera stereo. Sedangkan untuk kalibrasinya sendiri digunakan *tools* kalibrasi kamera dari Jean-Yves Bouquet (2015). Langkah yang dilakukan untuk kamera mono diantaranya adalah akuisisi citra, hitung dan cari parameter intrinsik, hitung dan cari parameter ekstrinsik (Weng et al., 1992). Kamera yang diuji untuk dikalibrasi pada kesempatan kali ini adalah kamera stereo yaitu Minoru 3D webcam (Minoru3D, 2009).

Sebelum melakukan kalibrasi, perlu dipersiapkan terlebih dahulu citra pola papan catur 14x10 kotak yang dicetak di kertas ukuran A4 dan ditempelkan pada sebuah papan triplek. Citra pola ini merupakan citra pola standar untuk keperluan kalibrasi. Hanya saja tidak ada ketentuan khusus berapa jumlah kotak pola yang digunakan untuk kalibrasi.

Gambar 1 menunjukkan rancangan antarmuka untuk mengakuisisi citra dari kamera kanan dan kiri secara sekuensial. Cara yang dilakukan pertama adalah dengan menekan tombol 'Start' yang kemudian akan mengaktifkan secara paralel kamera stereo. Citra yang diperoleh dari kedua kamera kemudian ditampilkan pada axes1 dan axes2. Adapun kode program untuk keperluan ini adalah:

```
1. % --- Executes on button press in tombolStart.
2. function tombolStart_Callback(hObject, eventdata, handles)
3. % hObject handle to tombolStart (see GCBO)
4. % eventdata reserved - to be defined in a future version of MATLAB
5. % handles structure with handles and user data (see GUIDATA)
6. proyek = guidata(gcbo); %28 Agt 2016
7. proses = 1;

8. %resolusi citra (piksel)
9. vid1 = videoinput('winvideo', 2, 'RGB24_640x480'); %minoruKanan
10. vid2 = videoinput('winvideo', 3, 'RGB24_640x480'); %minoruKiri

11. %-----
12. src1 = getselectedsource(vid1);
13. src2 = getselectedsource(vid2);
14. % %-----
15. vid1.FramesPerTrigger = 1; %jumlah frame citra yang dicapture
16. vid2.FramesPerTrigger = 1; %jumlah frame citra yang dicapture

17. triggerconfig(vid1, 'manual');
18. triggerconfig(vid2, 'manual');

19. vidObj1=vid1;
20. vidObj2=vid2;
21. %-----
22. vidRes1 = get(vidObj1, 'VideoResolution');
23. vidRes2 = get(vidObj2, 'VideoResolution');

24. imWidth1 = vidRes1(1);
25. imWidth2 = vidRes2(1);

26. imHeight1 = vidRes1(2);
27. imHeight2 = vidRes2(2);
```

```
28. nBands1 = get(vidObj1, 'NumberOfBands');
29. nBands2 = get(vidObj2, 'NumberOfBands');

30. hImage1 = image(zeros(imHeight1, imWidth1, nBands1) );
31. hImage2 = image(zeros(imHeight2, imWidth2, nBands2) );

32. hImage1 = image(zeros(vidRes1(2),vidRes1(1), nBands1),'Parent',handles.axes1);
33. hImage2 = image(zeros(vidRes2(2),vidRes2(1), nBands2),'Parent',handles.axes2);

34. preview(vidObj2, hImage2);
35. preview(vidObj1, hImage1);
```

Berikutnya untuk mengakuisisi citra, penulis menggunakan fungsi ‘get snapshot’ pada Matlab. Fungsi ini dijalankan secara bergantian untuk kamera kanan kemudian kamera kiri. Fungsi dijalankan pada saat tombol ‘Capture Image’ ditekan. Adapun kode program untuk tombol tersebut adalah:

```
1. % --- Executes on button press in captureImage.
2. function captureImage_Callback(hObject, eventdata, handles)
3. % hObject handle to captureImage (see GCBO)
4. % eventdata reserved - to be defined in a future version of MATLAB
5. % handles structure with handles and user data (see GUIDATA)
6. % -----4 Sept 2016 -----
7. proyek = guidata(gcbo); %28 Agt 2016
8. noFile = 1;
9. %-----
10. %resolusi 640x480 piksel
11. vid1 = videoinput('winvideo', 2, 'RGB24_640x480'); %minoruKanan
12. vid2 = videoinput('winvideo', 3, 'RGB24_640x480'); %minoruKiri
13. %-----
14. src1 = getselectedsource(vid1);
15. src2 = getselectedsource(vid2);
16. % % --- mengeset frameRate kamera -----
17. vid1.FramesPerTrigger = 1; %jumlah frame citra yang dicapture
18. vid2.FramesPerTrigger = 1; %jumlah frame citra yang dicapture

19. triggerconfig(vid1, 'manual');
20. triggerconfig(vid2, 'manual');

21. vidObj1=vid1;
22. vidObj2=vid2;
23. %-----
24. vidRes1 = get(vidObj1, 'VideoResolution');
25. vidRes2 = get(vidObj2, 'VideoResolution');

26. imWidth1 = vidRes1(1);
27. imWidth2 = vidRes2(1);

28. imHeight1 = vidRes1(2);
29. imHeight2 = vidRes2(2);

30. nBands1 = get(vidObj1, 'NumberOfBands');
31. nBands2 = get(vidObj2, 'NumberOfBands');

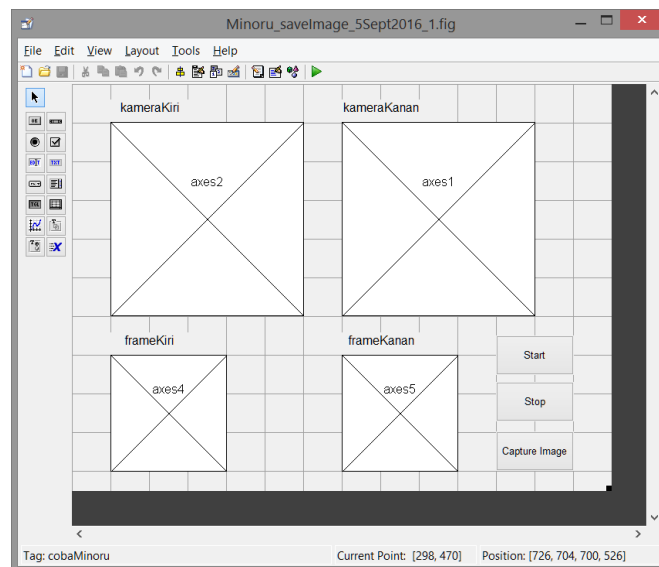
32. hImage1 = image(zeros(imHeight1, imWidth1, nBands1) );
33. hImage2 = image(zeros(imHeight2, imWidth2, nBands2) );

34. hImage1 = image(zeros(vidRes1(2),vidRes1(1), nBands1),'Parent',handles.axes1);
35. hImage2 = image(zeros(vidRes2(2),vidRes2(1), nBands2),'Parent',handles.axes2);

36. % preview(vidObj1, hImage1); %jadi error kalau digunakan
37. % preview(vidObj2, hImage2); %jadi error kalau digunakan
38. I1 = rgb2gray(getsnapshot(vidObj1)); %mencapture frame RGB
39. % Menampilkan hasil frame capture
40. set(proyek.cobaMinoru,'CurrentAxes',proyek.axes5); %tampilkan di axes
41. set(imshow(I1)); %tampilkan di axes
42. imsave; %simpan citra catur kanan
43. %-----
44. preview(vidObj2, hImage2);
45. I2 = rgb2gray(getsnapshot(vidObj2)); %mencapture frame RGB
```

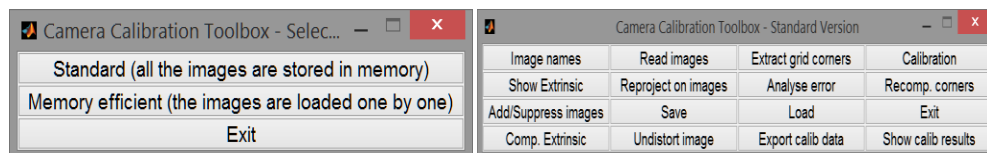
```
46. set(projek.cobaMinoru,'CurrentAxes',projek.axes4); %tampilkan di axes
47. set(imshow(I2)); %tampilkan di axes
48. imsave; %simpan citra catur kiri
```

Akuisisi citra papan catur dilakukan untuk berbagai posisi citra papan catur. Pemosisian ini dilakukan sedemikian rupa agar semua pola papan catur dapat terlihat baik pada kamera kanan dan kiri pada posisi yang sama. File citra berukuran 640x480 piksel kemudian disimpan dengan nama caturKanan_1.tif s.d. caturKanan_20.tif dan caturKiri_1.tif s.d. caturKiri_20.tif. Jumlah citra papan catur yang diakuisisi tidak harus 20 citra, bisa saja kurang dari 20 citra. Hanya saja 20 citra merupakan jumlah yang banyak digunakan oleh banyak penelitian untuk keperluan kalibrasi kamera ini. Penelitian Zhang (2000) hanya menggunakan 5 buah citra pola papan catur untuk kalibrasi.



Gambar 1. Rancangan antarmuka pengakuisisi citra

Kalibrasi kamera kanan dan kiri dimulai dengan menjalankan perintah 'calib' pada *command window* dan berikutnya dipilih menu 'Standar'. Gambar 2 menunjukkan tampilan dari menu 'camera calibration'.



Gambar 2. Menu Camera calibration

Menu 'Image name' yang akan meminta untuk dimasukkan nama file tanpa ekstensi, misal dalam kasus ini 'caturKanan_' atau 'caturKiri_'. Kemudian dijalankan menu 'read image', dan matlab akan membaca dan atau memuat semua citra ke *memory*. Hasil pembacaan 20 citra ditunjukkan pada Gambar 3. Citra ini merupakan citra yang diakuisisi dari kamera kanan. Citra-citra yang diakuisisi dari kamera kiri juga memiliki tampilan serupa, hanya saja posisinya yang berbeda.

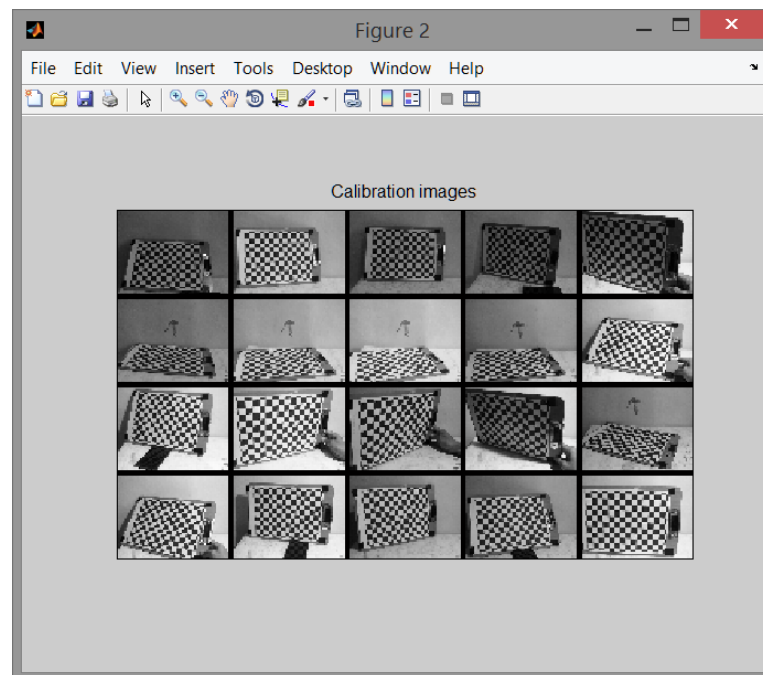
Proses berikutnya 'Extract grid corners' pada semua citra (20 citra kanan dan 20 citra kiri). Proses ini dilakukan dengan menentukan titik paling kiri bagian atas, paling kanan bagian atas, titik paling kanan bagian bawah, dan titik paling kiri bagian bawah dari semua pola papan catur (Gambar 4). Setelah semua selesai barulah proses kalibrasi secara otomatis akan dilakukan oleh sistem. Berikutnya adalah akan diperoleh hasil parameter

kalibrasi yang sudah dilakukan. Adapun parameter yang dihasilkan adalah parameter intrinsik. Parameter tersebut dihasilkan secara terpisah antara kamera kanan dan kamera kiri. Hasil kalibrasi kamera kanan dan kiri kemudian disimpan secara terpisah dengan nama file masing-masing 'Calib_Results_right.mat' dan 'Calib_Results_left.mat'.

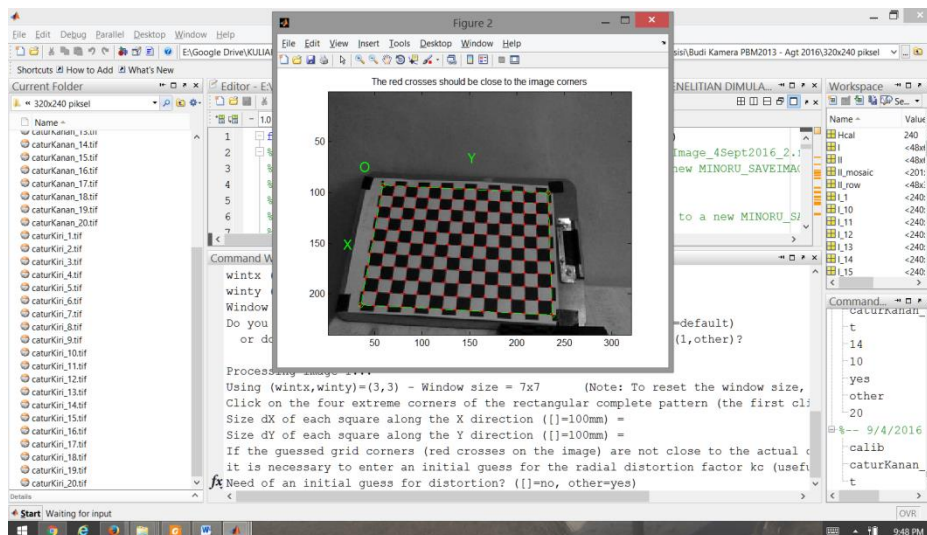
Selanjutnya untuk proses kalibrasi stereo dilakukan dengan menjalankan 'stereo_gui' yang kemudian akan membuka menu untuk kalibrasi stereo. Langkah berikutnya adalah me-Load file hasil kalibrasi kamera kanan dan kiri, dan barulah proses kalibrasi stereo dilakukan guna memperoleh parameter ekstrinsik dari kamera.

HASIL DAN PEMBAHASAN

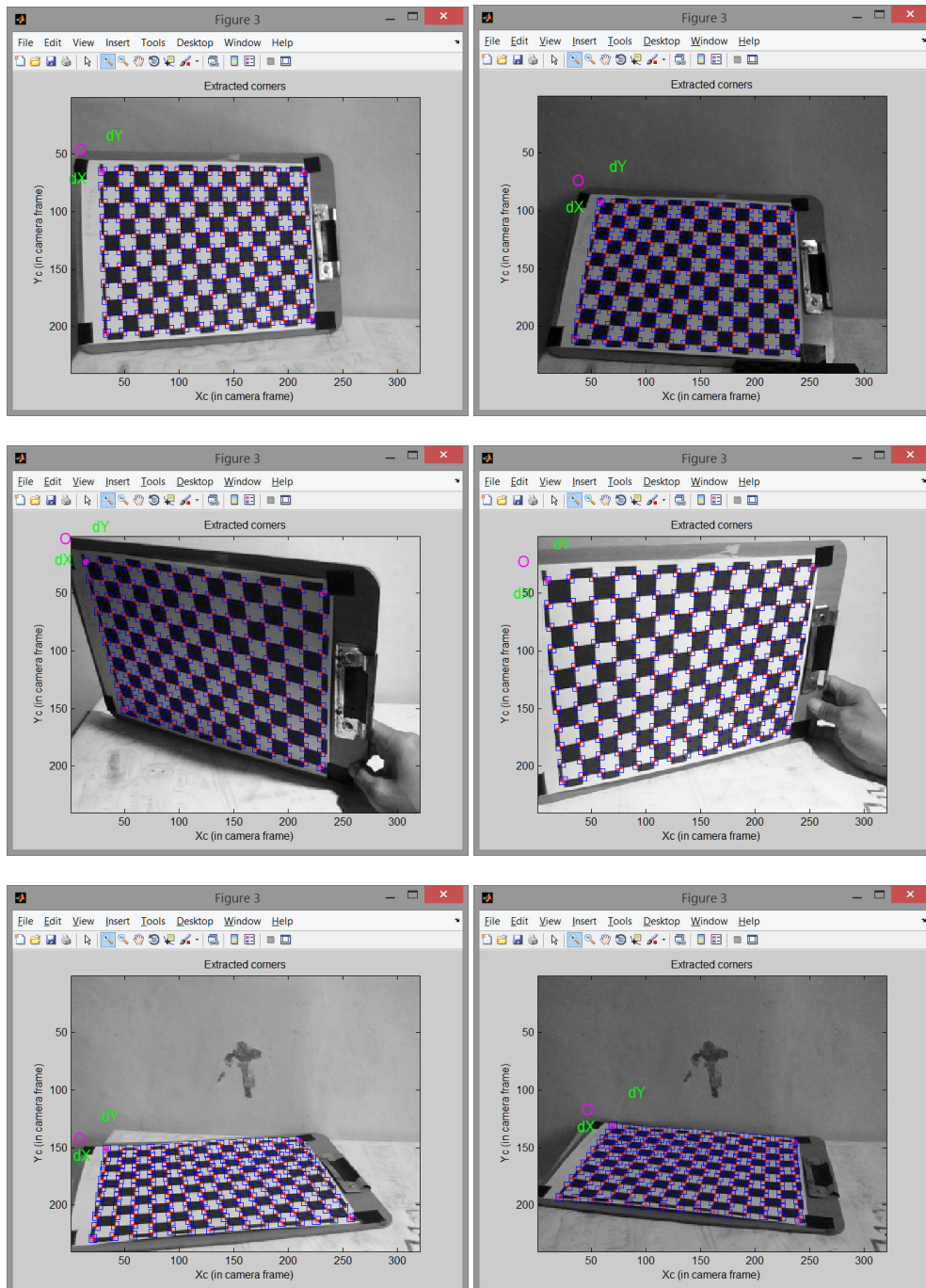
Berdasarkan percobaan yang dilakukan maka diperoleh hasil antara lain: hasil pembacaan 20 citra kanan (dan atau kiri), hasil *extraction grid corner*, parameter intrinsik, dan parameter ekstrinsik dari kamera.



Gambar 3. Citra pola papan catur yang dibaca oleh sistem



Gambar 4. Proses *Extract grid corners* citra papan catur



Gambar 5. Contoh Hasil *Extracted grid corners* 3 pasang citra papan catur

Jika sistem berhasil membaca semua citra yang menjadi bahan untuk dilakukannya kalibrasi untuk kamera kanan maupun kiri, maka sistem akan menampilkan semua citra kanan ataupun citra kiri yang terbaca (**Gambar 3**). Kemudian hasil ‘Extraction grid corner’ dapat dilihat hasilnya seperti diperlihatkan pada **Gambar 5**. Berdasarkan proses ekstraksi *grid corner* yang sudah dilakukan, ada kalanya *grid corner* yang terekstraksi tidaklah tepat pada *corner* atau sudut yang dimaksud. Hal tersebut nantinya akan berpengaruh pada hasil kalibrasi yang diperoleh. Berdasarkan percobaan yang dilakukan oleh penulis, hal ini dapat terjadi karena beberapa hal diantaranya adalah citra yang kurang kontras sehingga pada

saat penentuan titik-titik untuk proses ekstraksi *grid corner* ini menjadi tidak tepat. Artinya titik yang ditentukan tidak berada tepat di titik yang diinginkan.

Parameter intrinsik yang dihasilkan dari proses kalibrasi kamera kanan adalah:

Calibration results after optimization (with uncertainties):
Focal Length: $fc_right = [445.52625 \ 446.48967] \pm [2.25183 \ 2.23616]$
Principal point: $cc_right = [161.97016 \ 120.75739] \pm [4.79820 \ 3.43576]$
Skew: $alpha_c_right = [0.00000] \pm [0.00000] \Rightarrow$ angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion: $kc_right = [-0.12307 \ 0.27081 \ -0.00537 \ -0.00124 \ 0.00000] \pm [0.03096 \ 0.18225 \ 0.00218 \ 0.00288 \ 0.00000]$
Pixel error: $err = [0.36903 \ 0.40799]$

Note: The numerical errors are approximately three times the standard deviations (for reference).

Sedangkan untuk kamera kiri dihasilkan:

Calibration results after optimization (with uncertainties):
Focal Length: $fc = [448.02102 \ 449.55149] \pm [2.25988 \ 2.27298]$
Principal point: $cc = [170.87924 \ 108.23431] \pm [4.64926 \ 3.41166]$
Skew: $alpha_c = [0.00000] \pm [0.00000] \Rightarrow$ angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion: $kc = [-0.15811 \ 0.57712 \ -0.00407 \ -0.00126 \ 0.00000] \pm [0.03867 \ 0.29977 \ 0.00217 \ 0.00277 \ 0.00000]$
Pixel error: $err = [0.75958 \ 0.84768]$

Note: The numerical errors are approximately three times the standard deviations (for reference).

Terakhir yang dihasilkan dari proses kalibrasi stereo adalah parameter ekstrinsik dari kamera (**Gambar 6**). Berikut hasil yang didapatkan yaitu:

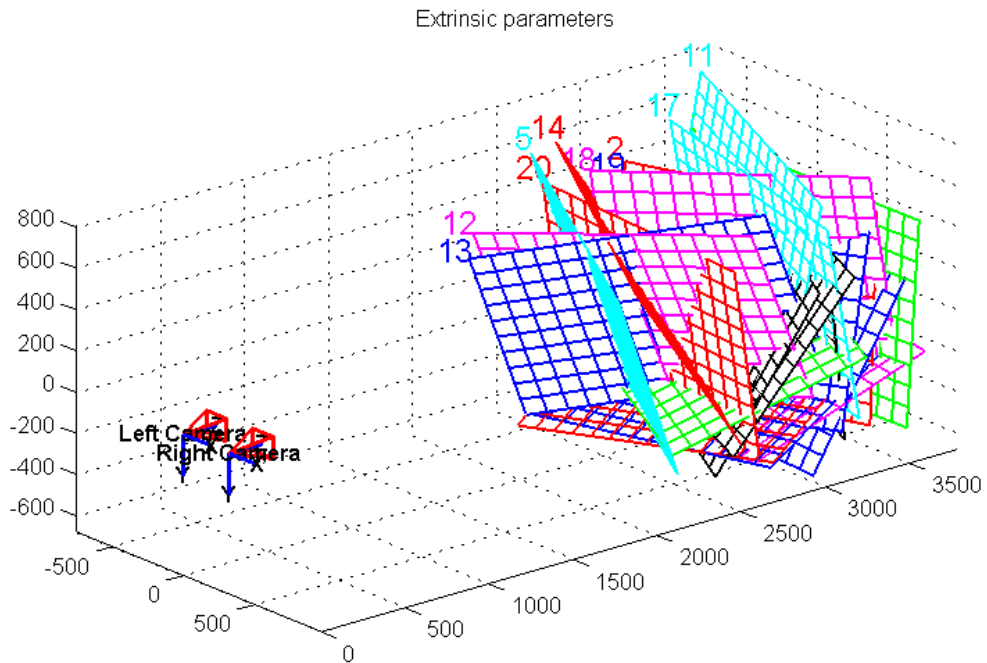
Stereo calibration parameters after loading the individual calibration files:
Intrinsic parameters of left camera:
Focal Length: $fc_left = [448.02102 \ 449.55149] \pm [2.25988 \ 2.27298]$
Principal point: $cc_left = [170.87924 \ 108.23431] \pm [4.64926 \ 3.41166]$
Skew: $alpha_c_left = [0.00000] \pm [0.00000] \Rightarrow$ angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion: $kc_left = [-0.15811 \ 0.57712 \ -0.00407 \ -0.00126 \ 0.00000] \pm [0.03867 \ 0.29977 \ 0.00217 \ 0.00277 \ 0.00000]$

Intrinsic parameters of right camera:
Focal Length: $fc_right = [445.52625 \ 446.48967] \pm [2.25183 \ 2.23616]$
Principal point: $cc_right = [161.97016 \ 120.75739] \pm [4.79820 \ 3.43576]$
Skew: $alpha_c_right = [0.00000] \pm [0.00000] \Rightarrow$ angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion: $kc_right = [-0.12307 \ 0.27081 \ -0.00537 \ -0.00124 \ 0.00000] \pm [0.03096 \ 0.18225 \ 0.00218 \ 0.00288 \ 0.00000]$

Extrinsic parameters (position of right camera wrt left camera):
Rotation vector: $om = [-0.00004 \ 0.00455 \ 0.00249]$
Translation vector: $T = [-328.95729 \ -0.07033 \ 5.96355]$

Saving the stereo calibration results in Calib_Results_stereo.mat

Hasil percobaan menunjukkan bahwa proses yang dilakukan telah mampu menghasilkan parameter kalibrasi yang kemudian dapat diterapkan pada citra yang akan diproses kemudian, baik secara real-time maupun tidak.



Gambar 6. Proyeksi parameter ekstrinsik proses kalibrasi kamera stereo

KESIMPULAN

Berdasarkan percobaan kalibrasi yang telah dilakukan, dapat disimpulkan bahwa proses yang penting adalah penentuan 4 titik dalam rangka ekstraksi *grid corner*. Jika ada kesalahan pada proses ini (misal terhadap salah satu dari 20 citra yang diekstraksi *grid cornernya*, maka akan meningkatkan nilai piksel error dan parameter kalibrasi yang dihasilkan menjadi kurang baik. Cara terbaik mengatasi hal ini adalah dengan mengulangi proses kalibrasi dari awal, baik itu terhadap citra kanan saja, citra kiri saja, ataupun keduanya. Berdasarkan proses kalibrasi yang telah dilakukan didapati pula bahwa kamera buatan pabrik (untuk kasus kamera stereo yang dikalibrasi) tidaklah menjamin bahwa parameter yang didapatkan adalah sama antara kanan dan kiri. Sebagai contoh bahwa titik tengah antara kedua kamera kanan dan kiri ternyata tidaklah sama. Karenanya proses kalibrasi sangatlah penting sebelum kamera digunakan untuk keperluan penentuan jarak objek terhadap kamera maupun untuk keperluan lainnya.

DAFTAR PUSTAKA

- Bouguet, J., 2015. *Camera Calibration Toolbox for Matlab*,
Minoru3D, 2009. Minoru 3D webcam user guide. , pp.1–22. Available at:
<http://www.minoru3d.com>.
Weng, J., Cohen, P. & Herniou, M., 1992. Camera Calibration with Distortion
Models and Accuracy Evaluation. *IEEE Transactions on Pattern Analysis and
Machine Intelligence*, 12(10), pp.965–980.
Zhang, Z., 2000. A Flexible New Technique for Camera Calibration. *IEEE
Transactions on Pattern Analysis and Machine Intelligence*, 22(11), pp.1330–
1334.