

# Detection and identification of a pattern for camera calibration based on Ankur method

Franci Suni Lopez  
Universidad Católica San Pablo-Arequipa Peru  
franci.suni@ucsp.edu.pe  
Joel Osvaldo Gallegos Guillen  
Universidad Católica San Pablo-Arequipa Peru  
joel.gallegos@ucsp.edu.pe

**Abstract**—The calibration process of a camera is necessary to obtain 3D information from 2D images of the scene. There are different techniques based on photogrammetry and self-calibration. As a result, the intrinsic and extrinsic parameters of the camera are obtained. Much work has been done in the calibration of cameras and also in the pre and post processing of the data. In this paper we use a calibration pattern based on rings in the following sections we present our proposal and results.

**Keywords**-camera calibration,

## I. INTRODUCTION

The calibration of a camera is the process that allows, within the field of artificial vision, the obtaining of the parameters that define the conditions of formation of the image, including the internal geometry and the optics of the camera, as well as its position and orientation with respect to a reference object or calibration standard. In short, calibration is a procedure that seeks to know how a camera projects a 3D object in the image plane so that it can extract metric information from the images.

The use of calibrated cameras can be useful to solve a series of problems related to obtaining the 3D position of objects in space from their images or for their three-dimensional reconstruction. This can allow, among other tasks, the realization of maps of the environment of the camera, the tracking of a specific object or the obtaining of the position of the camera with respect to objects that surround it. It can also facilitate navigating around a mobile robot, avoiding obstacles, addressing specific objects or facilitating the definition of the most appropriate path to reach your destination.

The calibration of a camera is a complex problem to solve since many are the parameters to be solved and many are the factors that influence the results. In part, this complexity is limited by the methods of calibration using camera models that are, in fact, ideal or simplified models of their physical equivalents. With these simplifications, quite acceptable results are obtained (although not exact), but it causes many parameters not to be parametrized. The pin-hole model, which is the one used by most calibration methods, does not parametrize optical aspects such as focus distance, depth of field, aperture or possible misalignment between the Flat image and lens. Even an effect as important as the distortion of images is

modelled, in most cases, in a very simplified way, and not in all cases is parametrized.

Other important factors are the possible sources of noise that influence the process of image formation since the image obtained by a camera introduces quantization errors, or the precision in the actual location of the elements (pattern points) used for performing the calibration is not exempt from imprecision. Due to these issues the calibration process of a camera can have multiple solutions, where the errors are compensated between the degrees of freedom (the parameters of the model), being able to obtain as a result of the calibration a parametrization that, although it minimizes an objective function final, it is unreal from the physical point of view.

## II. ANKUR METHOD

Incoming images for camera calibration suffer from non-linear distortion due to camera lenses. [Datta et al.(2009)] proposes an iterative refinement of the control points to recalculate and improve the initial camera parameters and obtain better results; this method is based on the use of a fronto-parallel canonical image without distortion to obtain the new centers of the pattern. The fronto-parallel image can be obtained following the following iterative approach after the refinement of the control points of the calibration pattern. In the first iteration, OpenCV is used to obtain an initial estimation of the parameters of the camera, then the input images are corrected for the distortion and are projected to a fronto-parallel plane where the control points are detected. After locating the centers, these points are reprojected using the calibration parameters initially estimated. Then those reprojected points are used to recalculate the camera parameters using the Levenberg-Marquardt optimization, and the process is repeated again until it converges.

Figure 1 shows the pipeline proposed by [Datta et al.(2009)] to do the fronto-parallel projection, for instance Figure 2 presents an example of frame rectification<sup>1</sup>.

<sup>1</sup> Software Package for Precise Camera Calibration: [http://www.ri.cmu.edu/research\\_project\\_detail.html?project\\_id=617&menu\\_id=261](http://www.ri.cmu.edu/research_project_detail.html?project_id=617&menu_id=261)

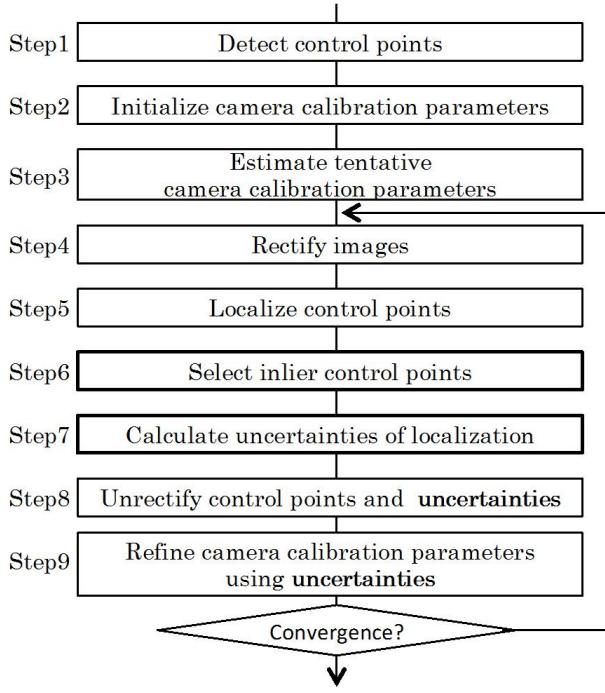


Fig. 1. Pipeline proposed by [Datta et al.(2009)]

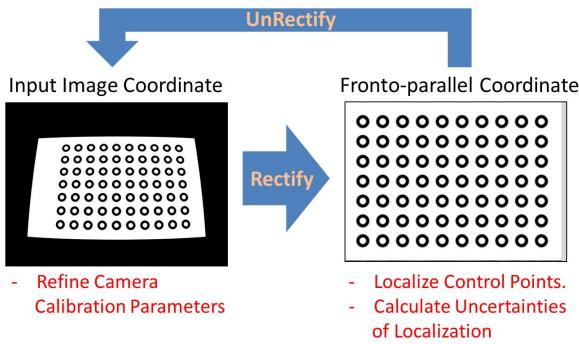


Fig. 2. Ejemplo de cálculo de la proyección fronto-paralela

### III. BACKGROUND

#### A. Homography

Given the point  $x = (u, v, 1)$  of one image and the another point  $x' = (u', v', 1)$  from another image. A homography is the matrix  $H$   $3 \times 3$  [Roth(2007)], where:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

that relates the coordinates of the pixels of both images in the following way:

$$x' = Hx$$

being the new image a deformed version of the original image, this occurs after applying the transformation to all the pixels of the initial image.

Also two images are in relationship by a homography only if:

- Both images are viewed from the same plane, with different angles.
- Both images are taken from the same camera but with different angle.

It should also be noted that the homography relation is independent of the structure of the scene, that is, it does not depend on what the camera is looking at that moment.

In case you need to return from the deformed image to the original image, you would simply perform the calculation with the inverse of the homography to each of the pixels of the distorted image:

$$x = H^{-1}x'$$

#### B. Levenberg-Marquardt Algorithm

Algorithm published by Kenneth Levenberg in 1944 and discovered by Donald Marquardt in 1963. Interpolates the Descent Gradient method together with Gauss Newton's Method. Serves to solve non-linear least-squares problems [Moré(1977)].

The primary application of the Levenberg-Marquardt algorithm lies in the problem of fitting least-squares curves: given a set of  $m$  pairs of empirical data  $(x_i, y_i)$  of independent and dependent variables, find the  $\beta$  parameters of the curve model  $f(x_i, \beta)$  so that the sum of the squares of the deviations is minimized:

$$\hat{\beta} = \text{argmin}_S(\beta) \equiv \text{argmin} \sum_{i=1}^m [y_i - f(x_i, \beta)]^2$$

This algorithm will serve for the optimization of intrinsic parameters and distortion coefficients.

### IV. PROCEDURE

The Ankur method is applied on the set of  $N$  frames selected to calibrate the camera, first calculate the centers of the frames and perform an initial calibration to obtain the intrinsic parameters that will be optimized. Then the distortion of the frame is removed to obtain the fronto-parallel projection and on this new image obtain the centers of the pattern; then these centers must be reprojected to an image without distortion and finally be distorted to obtain the new centers. These new centers are used to optimize the initial camera parameters using the Levenberg-Marquardt method and with the optimized parameters a new camera calibration is performed to start the cycle again. The loop terminates when the error between the previous RMS and the current RMS is less than a given tolerance or the maximum number of allowed iterations is reached. Figure 3 shows the general pipeline of Ankur proposal. In the following subsections we explain each step of the procedure.

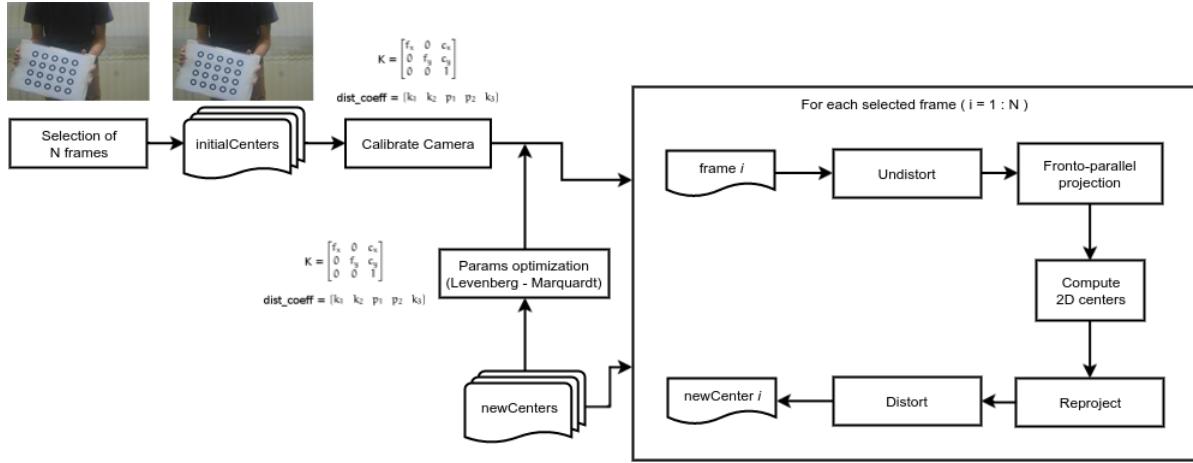


Fig. 3. Pipeline of Ankur method.

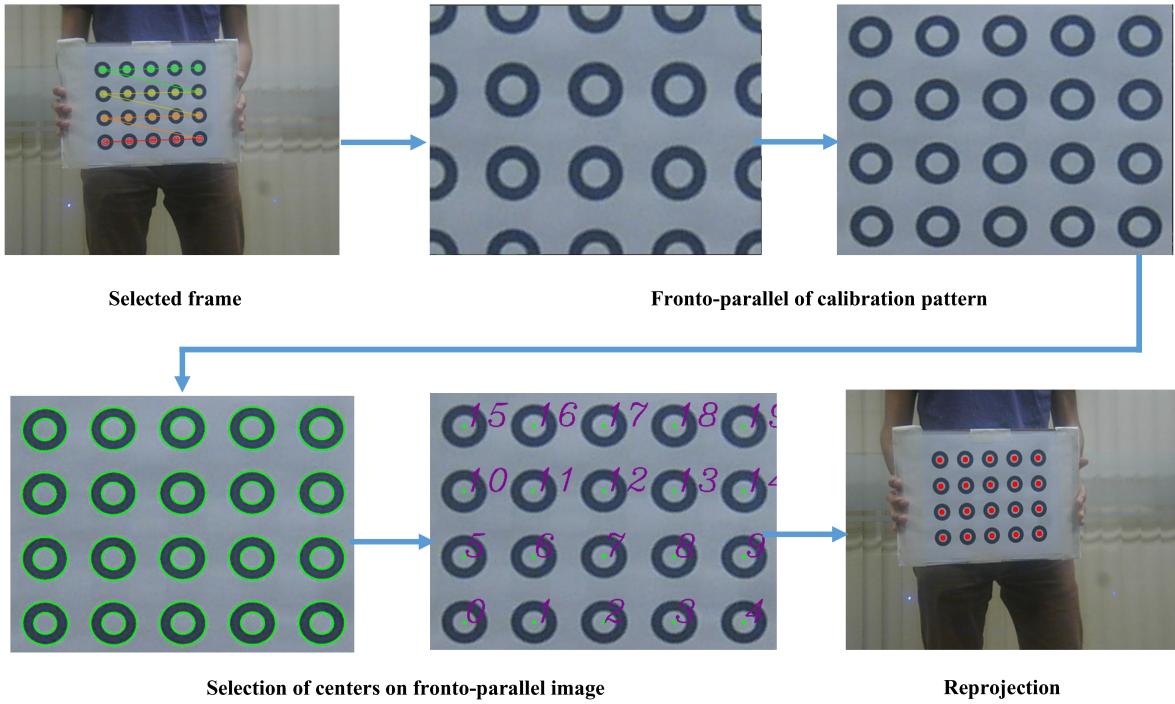


Fig. 4. Pipeline of fronto-parallel with our method and own images.

#### A. Undistort

As the first stage of the Ankur method it is necessary to correct the image with distortion, for this case the *undistort()* function of OpenCV was used [OpenCV(2007c)], that transforms an image to compensate for lens distortion; this function has as main parameters the distorted image, the camera matrix and the distortion coefficients.

#### B. Fronto-parallel projection

After correcting the distortion of the image we must generate the fronto-parallel projection; for this case we generate a template for each pattern, where the positions that the centers

of the pattern will have in the fronto-parallel projection are established.

To perform this transformation we calculate the homography matrix  $H$  using the function *findHomography()* of OpenCV [OpenCV(2007b)], the main parameters being the points to be transformed (centers) and the destination points (template); then we generate the fronto-parallel image with the function: *warpPerspective()* [OpenCV(2007c)], that applies the  $H$  matrix on the input image (image without distortion).

#### C. Computing of centers

Once we have the fronto-parallel image we apply our algorithm implemented to obtain 2D centers on that generated

| Calibration pattern | Life camera |        |        |        |          | PS3 camera |        |        |        |        |
|---------------------|-------------|--------|--------|--------|----------|------------|--------|--------|--------|--------|
|                     | R. error    | fx     | cx     | cy     | R. error | fx         | fy     | cx     | cy     |        |
| Chessboard          | 0.35204     | 624.77 | 624.24 | 324.33 | 244.38   | 0.23562    | 848.48 | 847.97 | 322.21 | 249.81 |
| Asymmetric circles  | 0.2208      | 563.73 | 567.76 | 327.83 | 231.32   | 0.1695     | 927.15 | 931.09 | 333.47 | 238.31 |
| Rings               | 0.19327     | 588.49 | 591.84 | 340.02 | 234.22   | 0.15135    | 820.92 | 823.95 | 337.91 | 241.43 |

TABLE I  
PART 1: COMPARISON OF RESULTS WITH DIFFERENT CALIBRATION PATTERN AND DIFFERENT VIDEO.

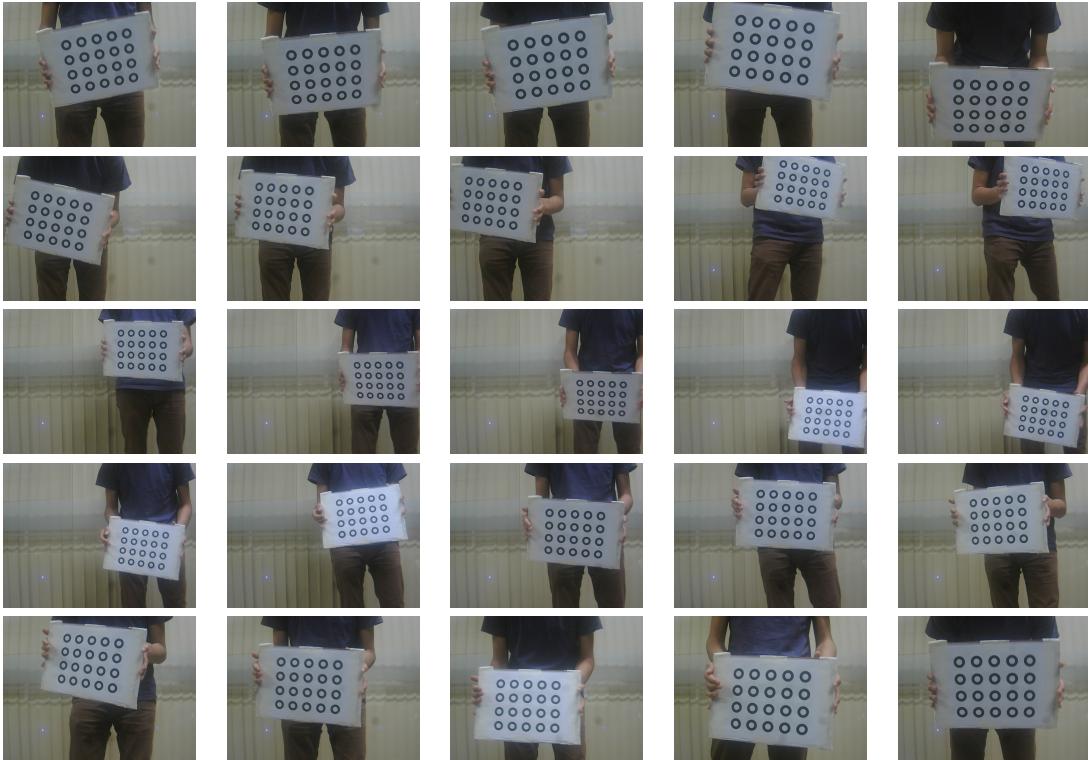


Fig. 5. Part 1: selected video frames for calibration of life camera.

image.

#### D. Reprojection

After obtaining the 2D centers of the fronto-parallel image we must reproject them to an image, without distortion; to do the reprojection we use the *perspectiveTransform()*function of OpenCv [OpenCV(2007a)], that allows to apply a perspective transformation matrix on a set of vectors. The set of input vectors are the 2D centers of the pattern and the transformation matrix is the inverse of the homography matrix  $H$ . Finally, Figure 4 presents the flow of the fronto-parallel process.

#### V. RESULTS FOR CAMERA CALIBRATION

Our results for this document are divided in two parts, the first shows the obtained calibration parameters with the initial videos (Jorge videos) and the second presents our results for the fronto-parallel configurations. On Table IV-A we show the different calibration parameters computed with the two video sources and the different calibration patterns.

We use 25 frames to calibrate the camera also, the selection of the frames is manually, the advantage of this way is that

we can select the best frames to calibrate the camera. Figure 5 shows the selected frames which we get the results on Table IV-A. Finally, we observed that our reprojection error using a calibration pattern based on rings (5x4)is less than others calibration patterns.

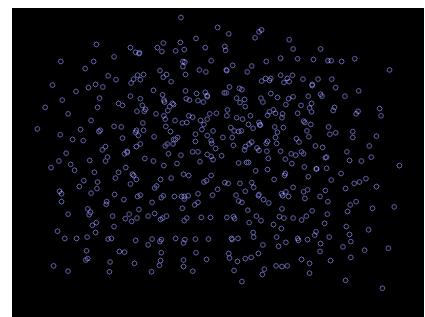


Fig. 6. God distribution of points for camera calibration.

## VI. RESULTS FOR FRONTO-PARALLEL PROJECTION

To get the best parameter for camera calibration we have to select a well distributed points of the scene, opposite case select points of one side of the scene generate a bad calibration. Then Figure 6 shows a good example of a well distribution of the selected points, and Figure 7 presents two examples of a bad distribution of points.

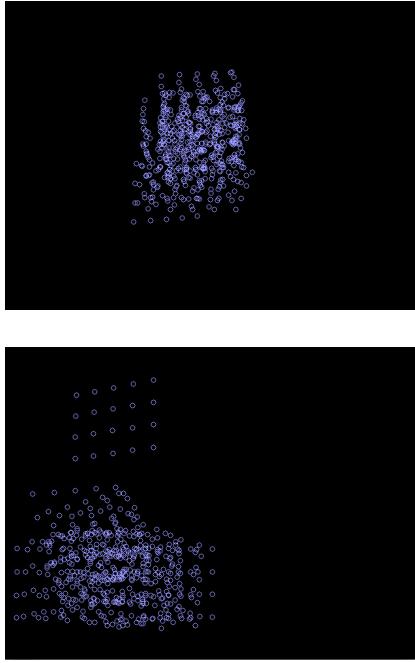


Fig. 7. Bad distributions of points for camera calibration.

The selected frames of the points distribution presented on Figure 6 is showed on Figure 8 and on Figure 9 we present the reprojection of the computed centers on the original selected frames. Finally we can summarize our results as follow:

- Table II presents the calibration parameters using the rings pattern and Life camera, the showed results are of the 50th iteration, and Figures 10, 11, 12 and 13 show the behaviour of each parameter in each iteration.
- Table III presents the calibration parameters using the rings pattern and PS3 camera, the showed results are of the 50th iteration, and Figures 14, 15, 16 and 17 show the behaviour of each parameter in each iteration.
- Table IV presents the calibration parameters using the chessboard pattern and Life camera, the showed results are of the 20th iteration, because we showed that from 20th iteration to more the behaviour is the same. Figures 18, 19, 20 and 21 show the behaviour of each parameter in each iteration.
- Table V presents the calibration parameters using the chessboard pattern and PS3 camera, the showed results are of the 20th iteration, and Figures 22, 23, 24 and 25 show the behaviour of each parameter in each iteration.
- Table VI presents the calibration parameters using the asymmetric pattern and Life camera, the showed results

are of the 20th iteration, because we showed that from 20th iteration to more the behaviour is the same. Figures 26, 27, 28 and 29 show the behaviour of each parameter in each iteration.

- Table VII presents the calibration parameters using the asymmetric pattern and PS3 camera, the showed results are of the 20th iteration, and Figures 30, 31, 32 and 33 show the behaviour of each parameter in each iteration.

## REFERENCES

- [CMiNPack(2007)] CMiNPack. lmdif1.c. <https://github.com/devernay/cminpack/blob/master/lmdif1.c>, 2007.
- [Datta et al.(2009)] Datta, Kim, and Kanade] A. Datta, J. Kim, and T. Kanade. Accurate camera calibration using iterative refinement of control points. In *Workshop on Visual Surveillance (VS), 2009 (held in conjunction with ICCV)*, October 2009.
- [Devernay(2007)] F. Devernay. C/C++ Minpack. <http://devernay.free.fr/hacks/cminpack/>, 2007.
- [Moré(1977)] J. J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, pages 105–116. Springer, Berlin, 1977.
- [OpenCV(2007a)] OpenCV. Operations on Arrays. [http://docs.opencv.org/2.4/modules/core/doc/operations\\_on\\_arrays.html](http://docs.opencv.org/2.4/modules/core/doc/operations_on_arrays.html), 2007a.
- [OpenCV(2007b)] OpenCV. Camera Calibration and 3D Reconstruction. [http://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html](http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html), 2007b.
- [OpenCV(2007c)] OpenCV. Geometric Image Transformations. [http://docs.opencv.org/3.0-beta/modules/imgproc/doc/geometric\\_transformations.html](http://docs.opencv.org/3.0-beta/modules/imgproc/doc/geometric_transformations.html), 2007c.
- [Roth(2007)] G. Roth. Homography. [http://people.scs.carleton.ca/~c\\_shu/Courses/comp4900d/notes/homography.pdf](http://people.scs.carleton.ca/~c_shu/Courses/comp4900d/notes/homography.pdf), 2007.

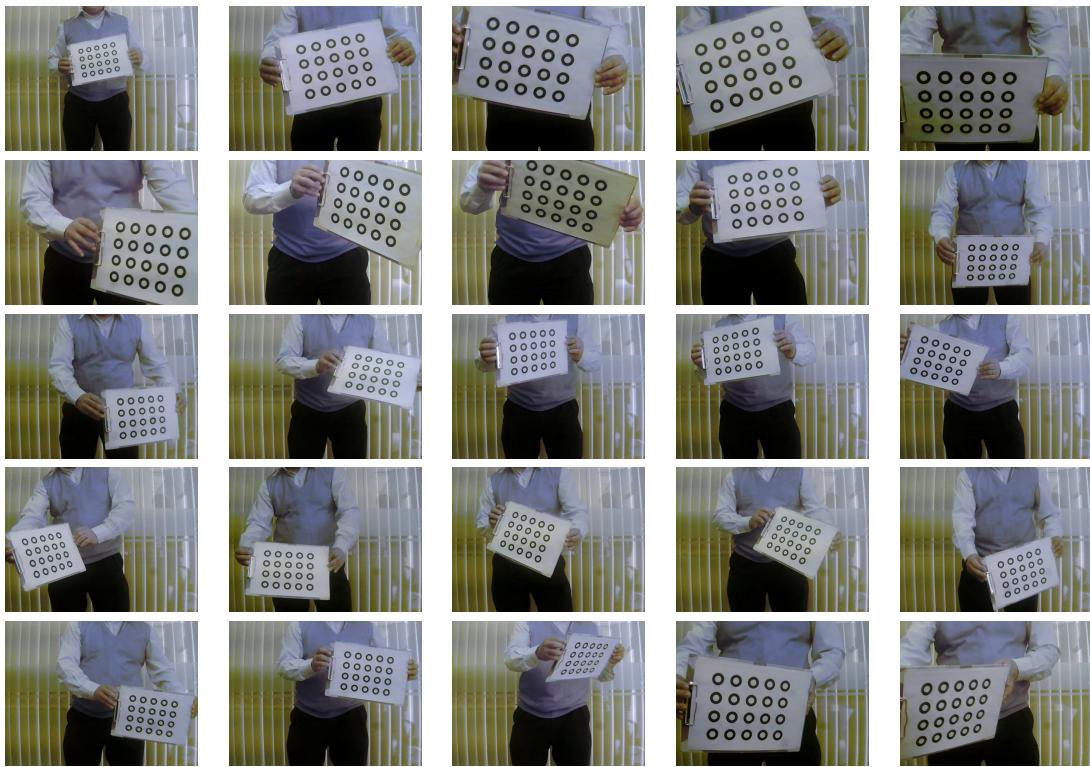


Fig. 8. Selected frames for calibration.

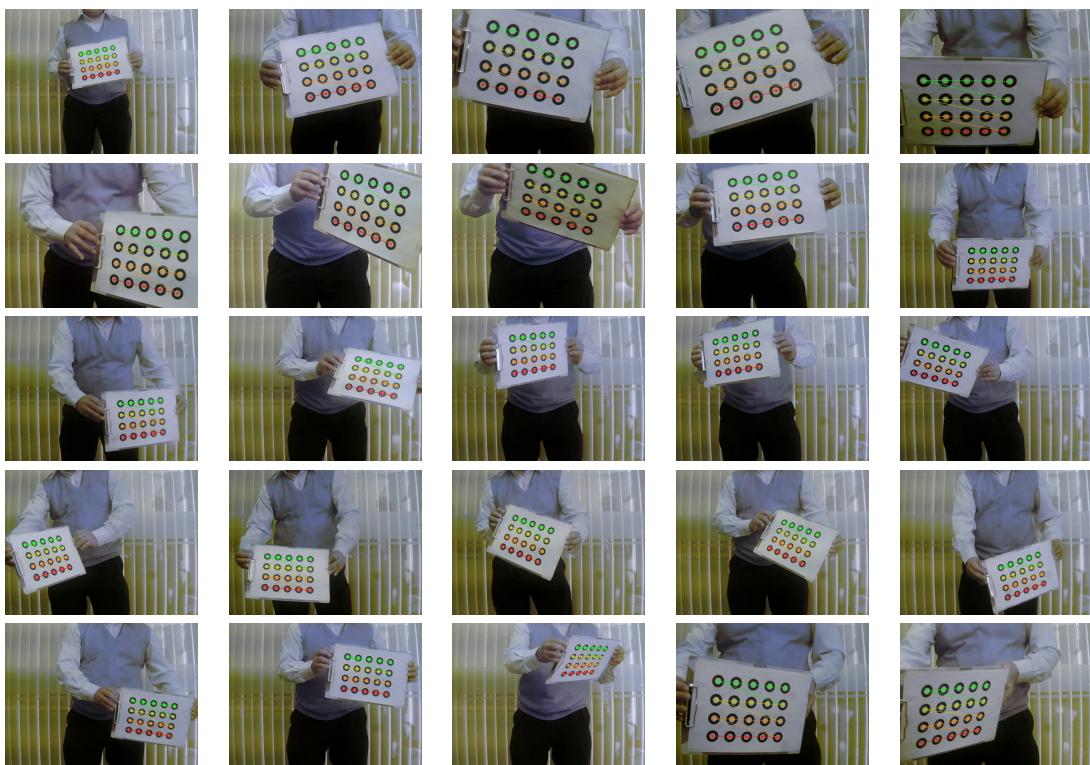


Fig. 9. Reprojection of point on original selected frames presented on Figure 8.

| Numer of frames | Life camera-Rings pattern |         |         |         |          |             |
|-----------------|---------------------------|---------|---------|---------|----------|-------------|
|                 | fx                        | fy      | cx      | cy      | R. error | Col         |
| 25              | 559.794                   | 552.813 | 326.995 | 218.718 | 0.14079  | 0.000089    |
| 30              | 549.744                   | 543.572 | 331.376 | 219.521 | 0.152673 | 0.000101834 |
| 40              | 567.745                   | 560.507 | 334.311 | 227.778 | 0.182651 | 0.000111931 |
| 50              | 577.685                   | 571.781 | 327.861 | 228.734 | 0.207857 | 0.000106074 |

TABLE II

RESULTS OF CALIBRATIONS PARAMETERS OF LIFE CAMERA WITH DIFFERENT NUMBER OF FRAMES FOR RINGS PATTERN, THE SHOWED RESULTS ARE OF THE 50TH ITERATION.



Fig. 10. Results of Life camera calibration with 25 frames and 50 iterations for rings pattern.

| Numer of frames | PS3 camera-Rings pattern |         |         |         |          |             |
|-----------------|--------------------------|---------|---------|---------|----------|-------------|
|                 | fx                       | fy      | cx      | cy      | R. error | Col         |
| 25              | 835.258                  | 828.331 | 311.722 | 245.671 | 0.281844 | 0.000065    |
| 30              | 833.331                  | 826.575 | 300.828 | 248.625 | 0.303913 | 0.00006.9   |
| 40              | 822.559                  | 815.67  | 302.04  | 249.15  | 0.350216 | 0.00008.55  |
| 50              | 827.881                  | 821.391 | 299.945 | 244.493 | 0.411    | 0.000123271 |

TABLE III

RESULTS OF CALIBRATIONS PARAMETERS OF PS3 CAMERA WITH DIFFERENT NUMBER OF FRAMES FOR RINGS PATTERN, THE SHOWED RESULTS ARE OF THE 50TH ITERATION.

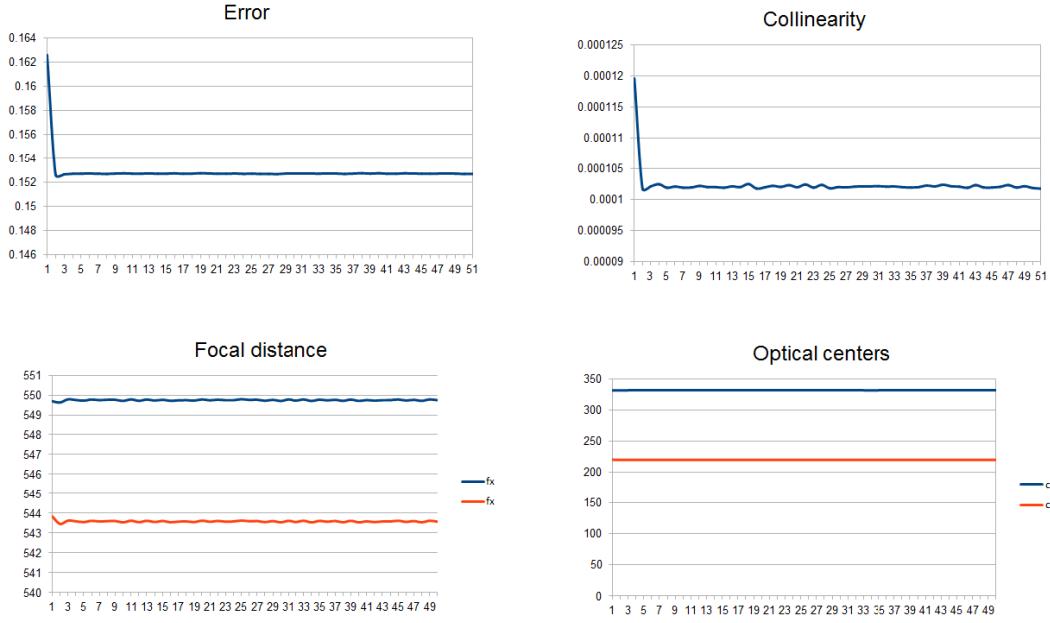


Fig. 11. Results of Life camera calibration with 30 frames and 50 iterations for rings pattern.

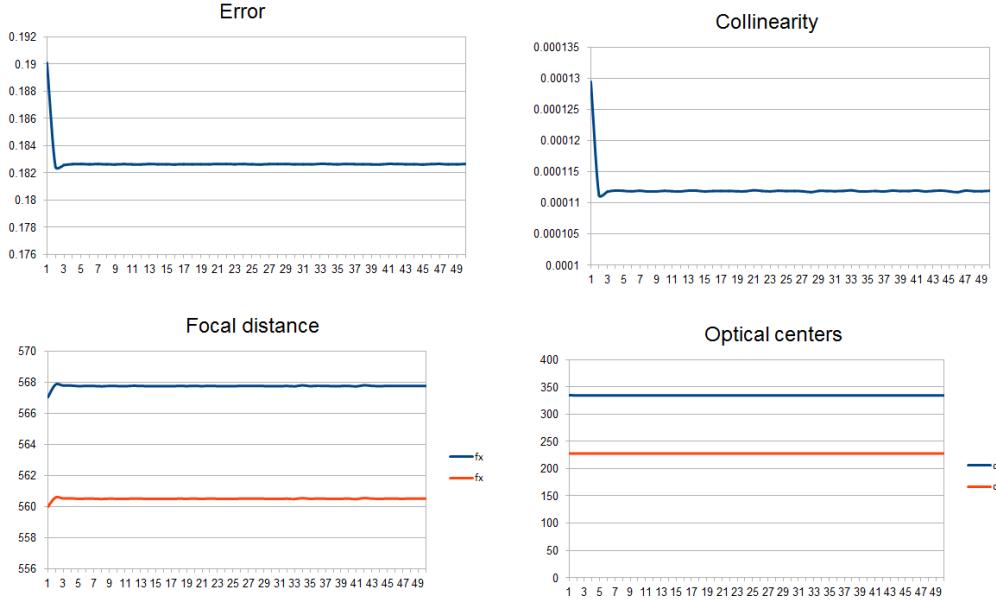


Fig. 12. Results of Life camera calibration with 40 frames and 50 iterations for rings pattern.

| Numer of frames | Life camera-Chessboard pattern |         |         |         |          |            |
|-----------------|--------------------------------|---------|---------|---------|----------|------------|
|                 | $f_x$                          | $f_y$   | $c_x$   | $c_y$   | R. error | Col        |
| 25              | 401.053                        | 407.25  | 373.806 | 192.64  | 4.22128  | 0.00387705 |
| 30              | 509.817                        | 517.819 | 349.112 | 212.217 | 3.76644  | 0.00299109 |
| 40              | 588.951                        | 578.237 | 332.354 | 236.734 | 3.23615  | 0.00187685 |
| 50              | 585.427                        | 579.667 | 334.313 | 239.309 | 2.82979  | 0.00150241 |

TABLE IV

RESULTS OF CALIBRATIONS PARAMETERS OF LIFE CAMERA WITH DIFFERENT NUMBER OF FRAMES FOR CHESSBOARD PATTERN, THE SHOWED RESULTS ARE OF THE 20TH ITERATION.

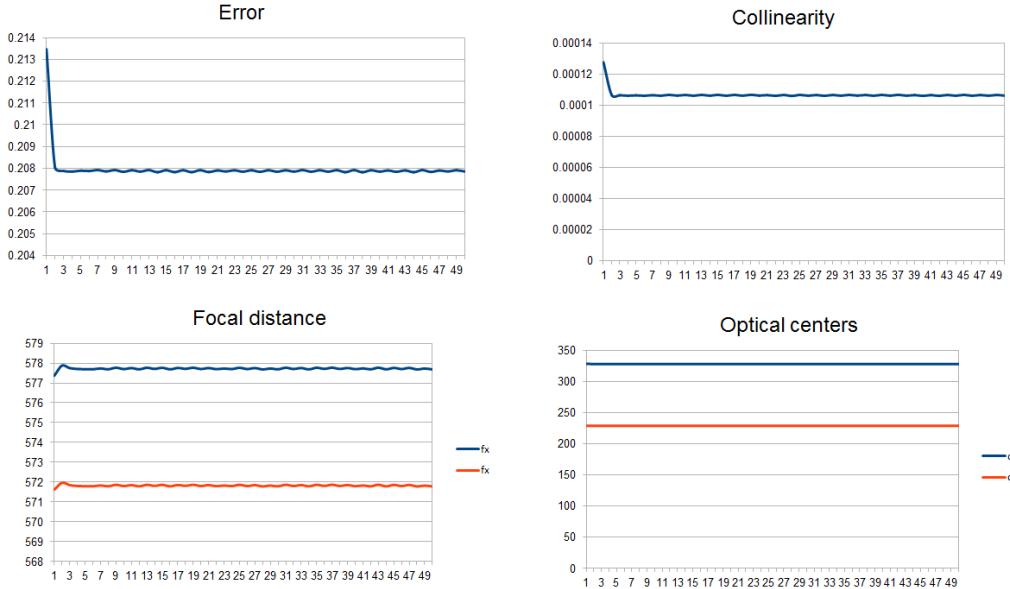


Fig. 13. Results of Life camera calibration with 50 frames and 50 iterations for rings pattern.

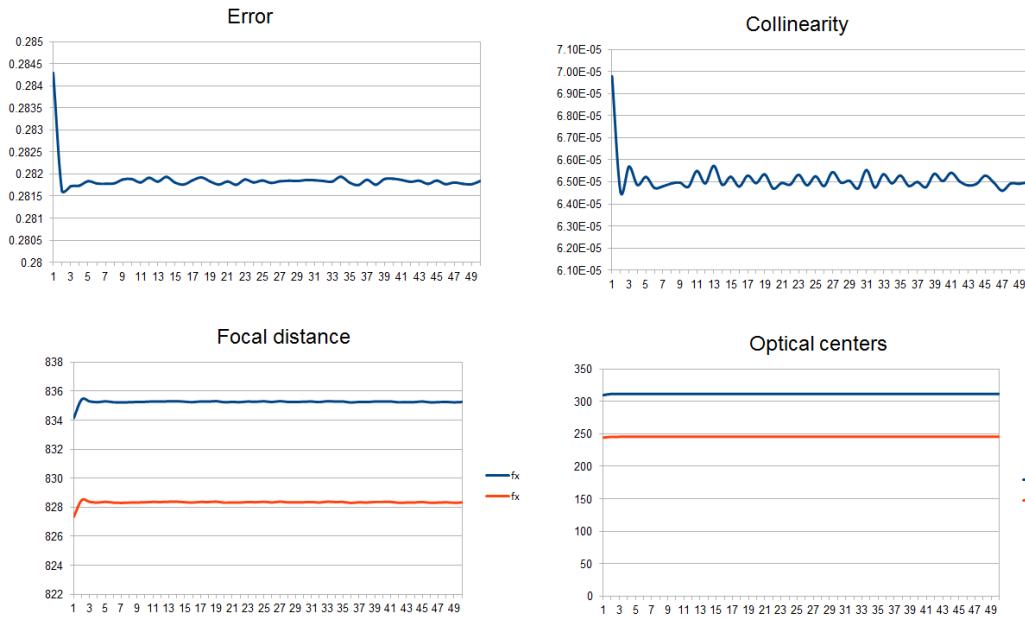


Fig. 14. Results of PS3 camera calibration with 25 frames and 50 iterations for rings pattern.

| Numer of frames | PS3 camera-Chessboard pattern |         |         |         |          |             |
|-----------------|-------------------------------|---------|---------|---------|----------|-------------|
|                 | fx                            | fy      | cx      | cy      | R. error | Col         |
| 25              | 791.046                       | 776.881 | 333.761 | 298.362 | 0.893937 | 0.000151916 |
| 30              | 804.248                       | 791.28  | 325.62  | 292.727 | 0.838611 | 0.00015405  |
| 40              | 817.088                       | 808.095 | 313.358 | 282.489 | 0.797239 | 0.000151916 |
| 50              | 812.07                        | 802.006 | 315.525 | 255.765 | 0.772288 | 0.000134092 |

TABLE V

RESULTS OF CALIBRATIONS PARAMETERS OF PS3 CAMERA WITH DIFFERENT NUMBER OF FRAMES FOR CHESSBOARD PATTERN, THE SHOWED RESULTS ARE OF THE 20TH ITERATION.



Fig. 15. Results of PS3 camera calibration with 30 frames and 50 iterations for rings pattern.

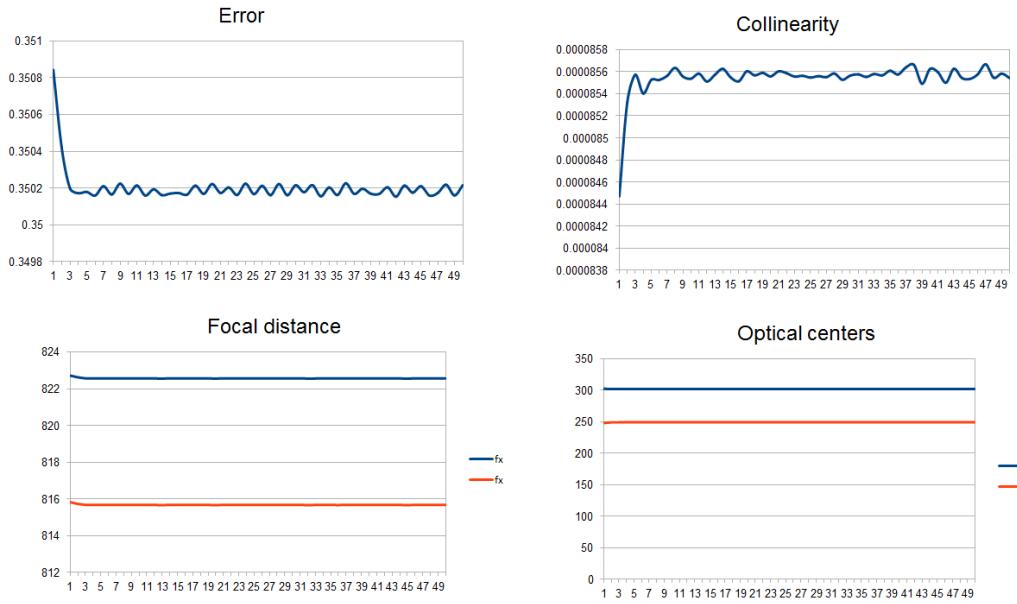


Fig. 16. Results of PS3 camera calibration with 40 frames and 50 iterations for rings pattern.

| Numer of frames | Life camera-Asymmetric pattern |         |         |         |          |             |
|-----------------|--------------------------------|---------|---------|---------|----------|-------------|
|                 | $f_x$                          | $f_y$   | $c_x$   | $c_y$   | R. error | Col         |
| 25              | 614.156                        | 612.825 | 332.037 | 227.527 | 0.197152 | 0.070335225 |
| 30              | 611.596                        | 610.001 | 330.532 | 227.759 | 0.201854 | 0.0756189   |
| 40              | 606.7                          | 605.028 | 327.581 | 227.581 | 0.251959 | 0.0907024   |
| 50              | 611.122                        | 608.233 | 332.812 | 223.02  | 0.284187 | 0.0990664   |

TABLE VI

RESULTS OF CALIBRATIONS PARAMETERS OF LIFE CAMERA WITH DIFFERENT NUMBER OF FRAMES FOR ASYMMETRIC PATTERN, THE SHOWED RESULTS ARE OF THE 20TH ITERATION.

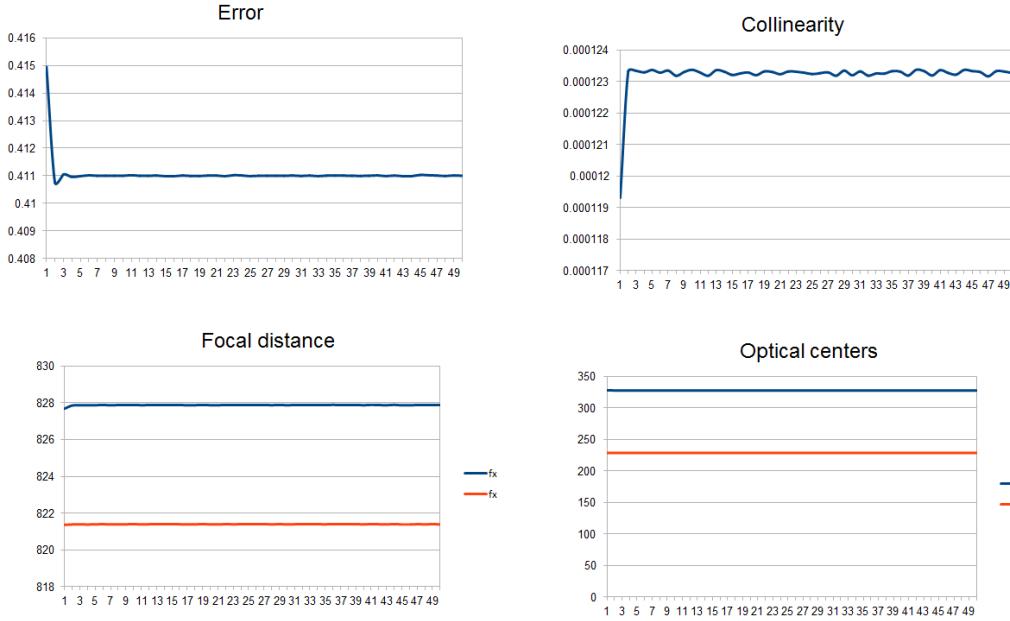


Fig. 17. Results of PS3 camera calibration with 50 frames and 50 iterations for rings pattern.

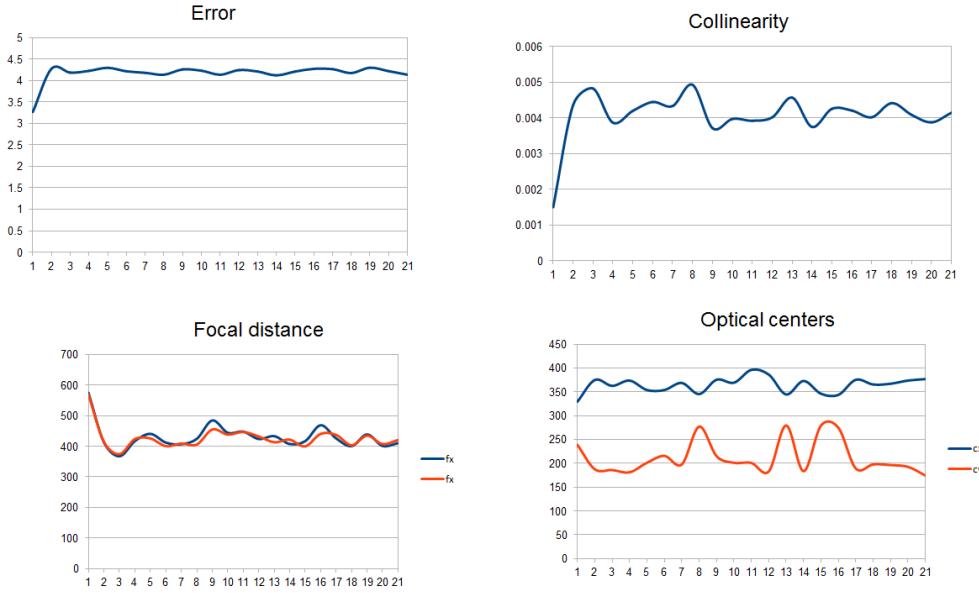


Fig. 18. Results of Life camera calibration with 25 frames and 20 iterations for chessboard pattern.

| Numer of frames | PS3 camera-Asymmetric pattern |         |         |         |          |           |
|-----------------|-------------------------------|---------|---------|---------|----------|-----------|
|                 | fx                            | fy      | cx      | cy      | R. error | Col       |
| 25              | 827.249                       | 824.173 | 331.117 | 261.044 | 0.227696 | 0.062478  |
| 30              | 826.765                       | 823.52  | 323.022 | 266.383 | 0.232171 | 0.0640078 |
| 40              | 837.656                       | 831.34  | 362.501 | 248.74  | 0.267859 | 0.0671237 |
| 50              | 870.909                       | 862.939 | 402.742 | 276.328 | 0.316867 | 0.074168  |

TABLE VII

RESULTS OF CALIBRATIONS PARAMETERS OF PS3 CAMERA WITH DIFFERENT NUMBER OF FRAMES FOR ASYMMETRIC PATTERN, THE SHOWED RESULTS ARE OF THE 20TH ITERATION.

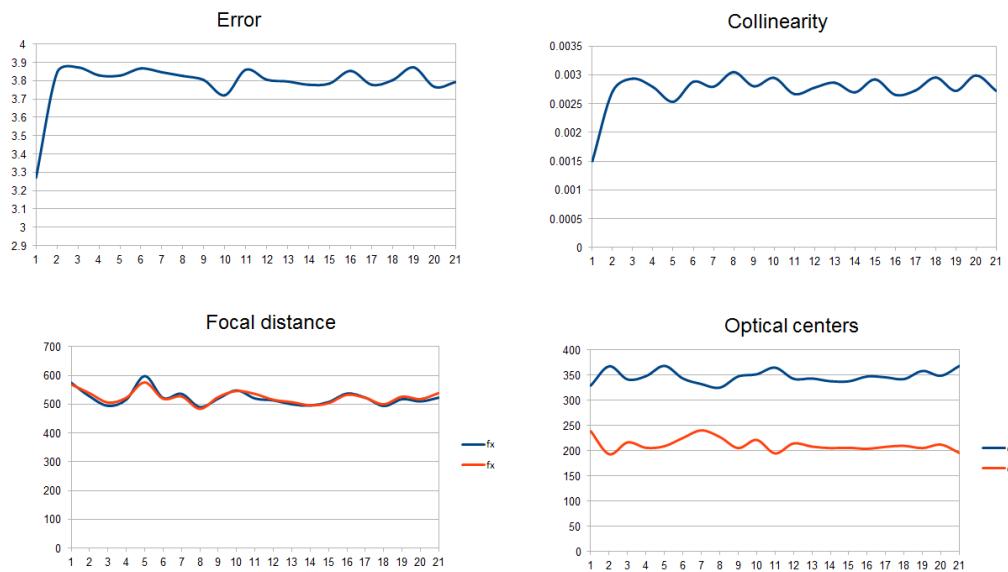


Fig. 19. Results of Life camera calibration with 30 frames and 20 iterations for chessboard pattern.

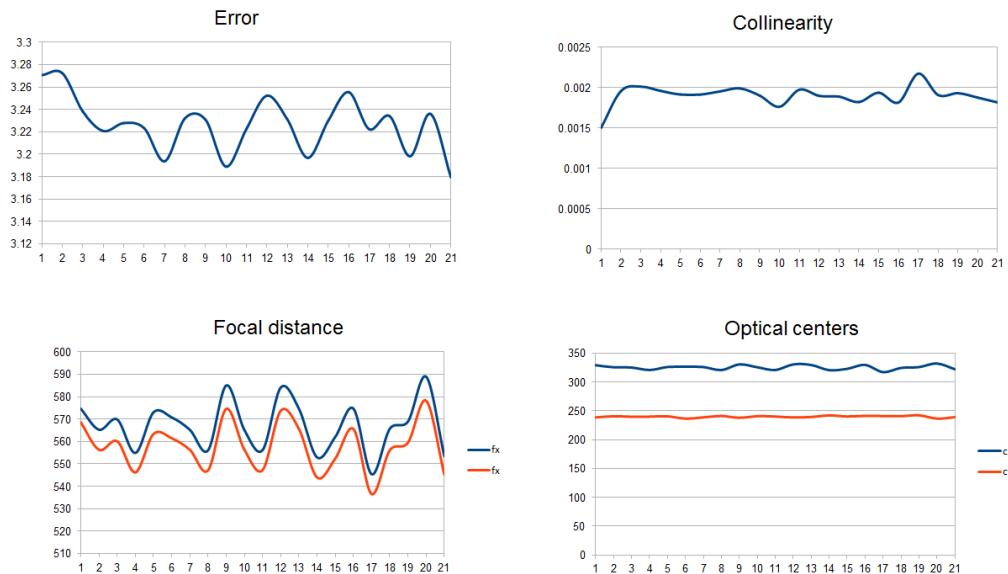


Fig. 20. Results of Life camera calibration with 40 frames and 20 iterations for chessboard pattern.

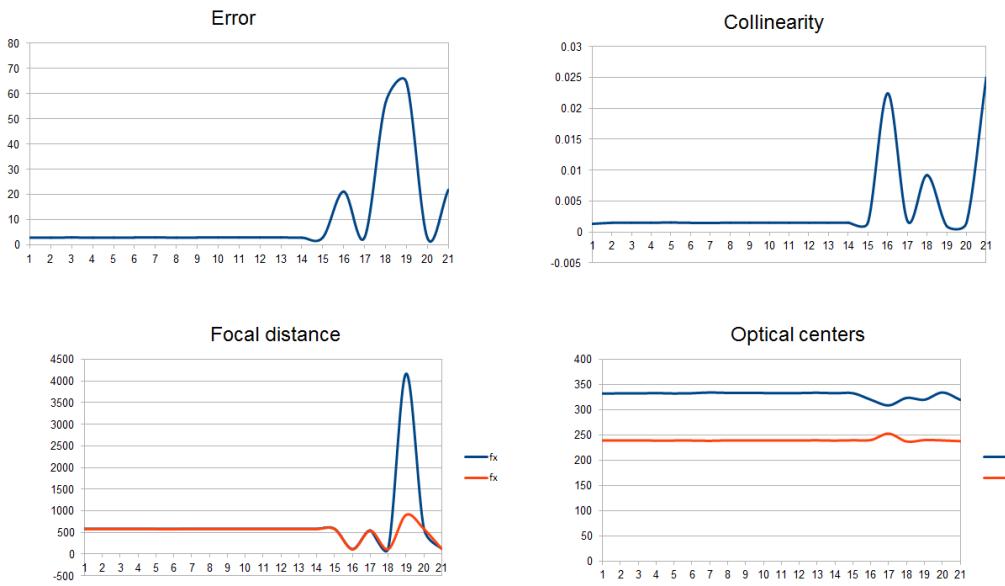


Fig. 21. Results of Life camera calibration with 50 frames and 20 iterations for chessboard pattern.

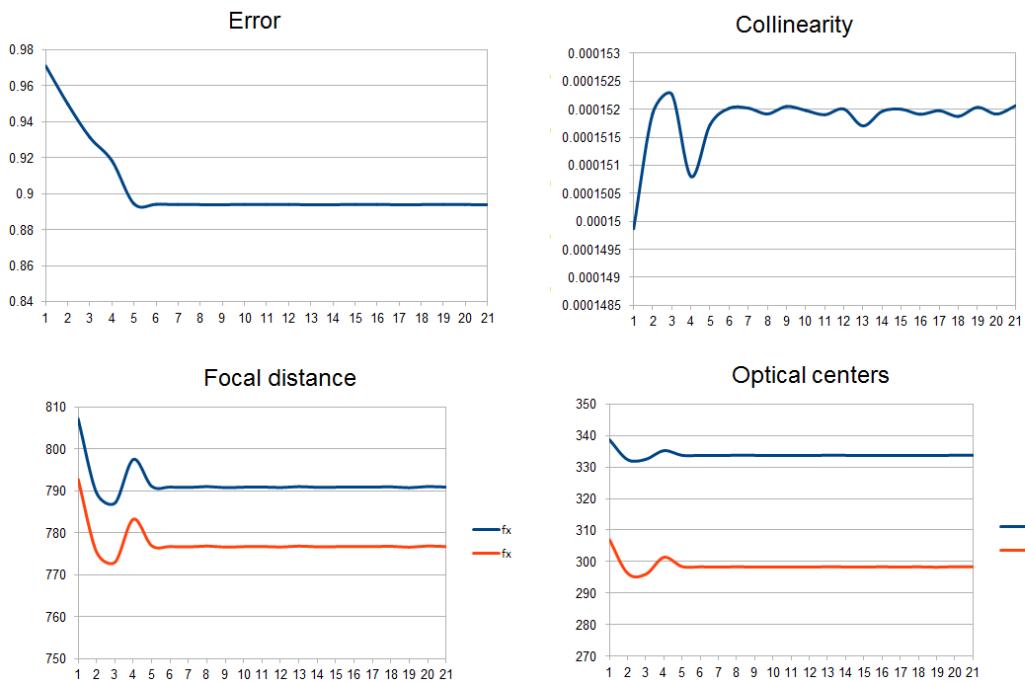


Fig. 22. Results of Life camera calibration with 25 frames and 20 iterations for chessboard pattern.

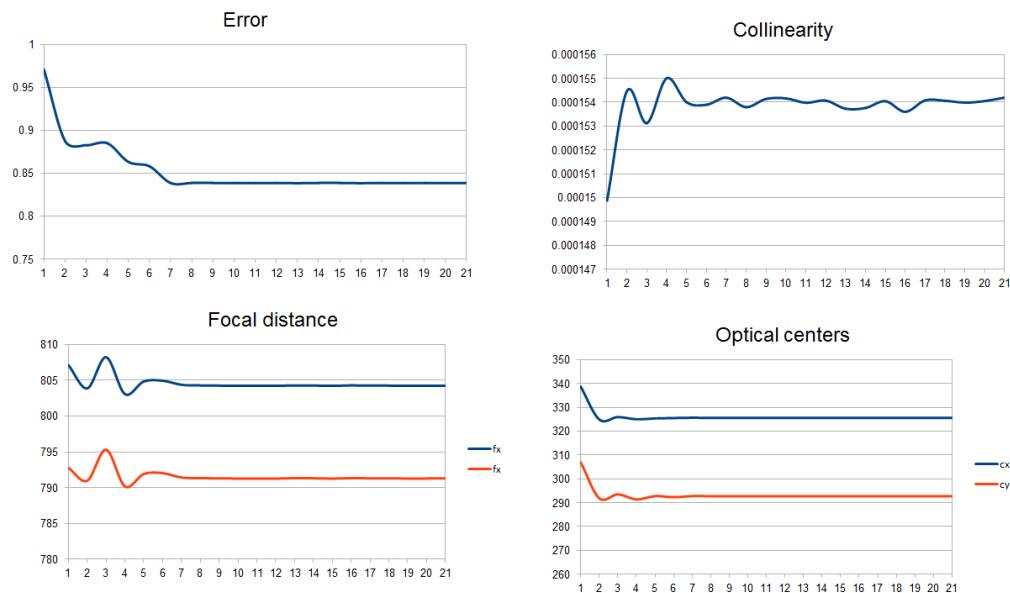


Fig. 23. Results of Life camera calibration with 30 frames and 20 iterations for chessboard pattern.

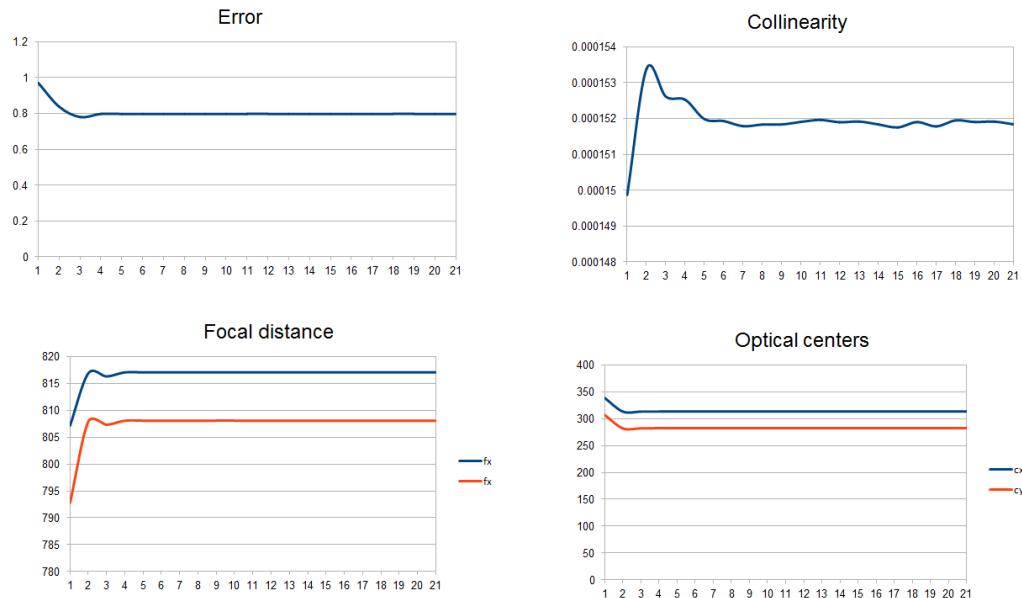


Fig. 24. Results of Life camera calibration with 40 frames and 20 iterations for chessboard pattern.

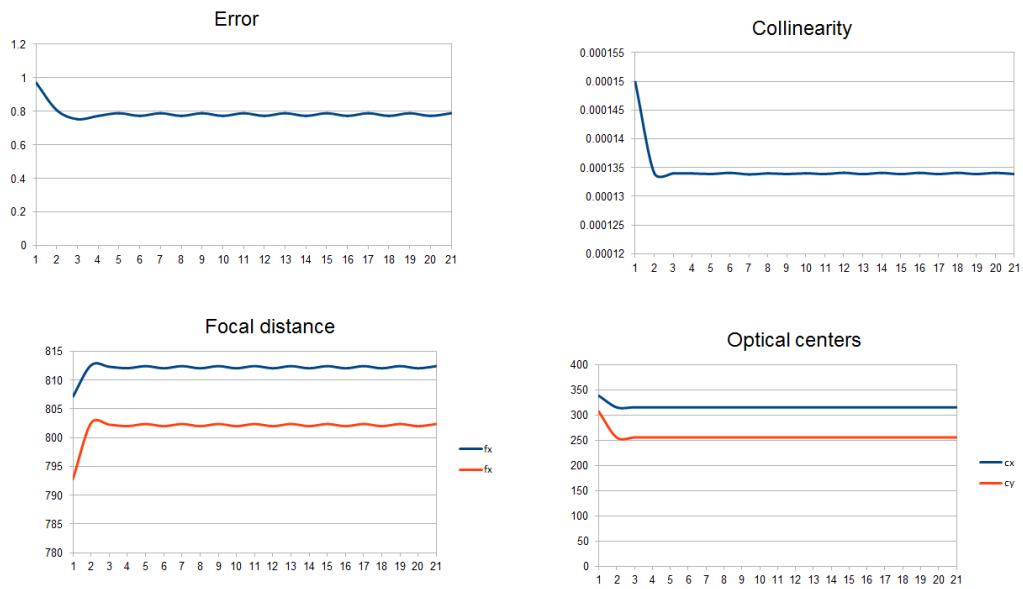


Fig. 25. Results of Life camera calibration with 40 frames and 20 iterations for chessboard pattern.

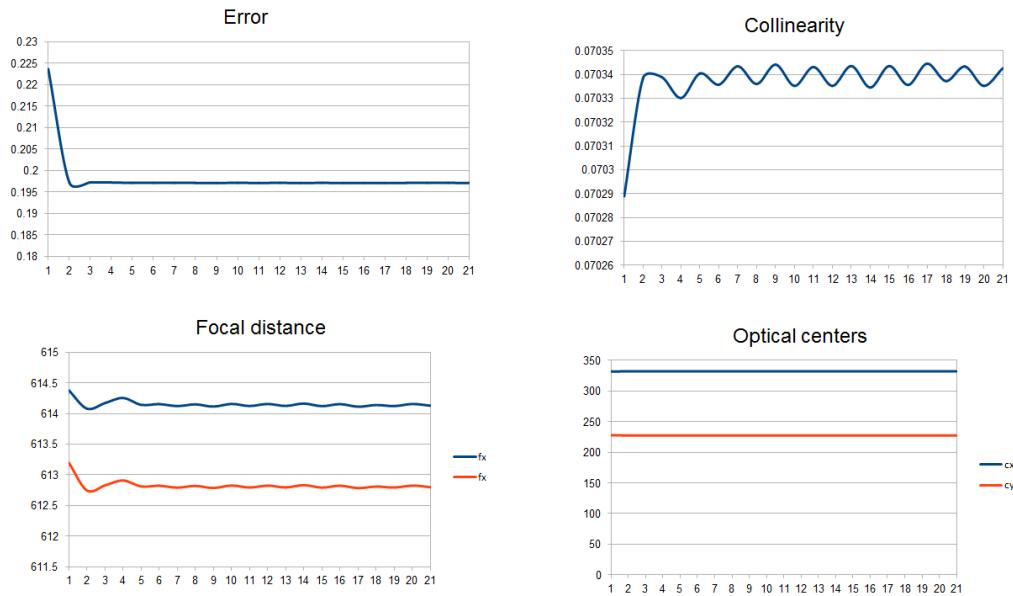


Fig. 26. Results of Life camera calibration with 25 frames and 20 iterations for asymmetric pattern.

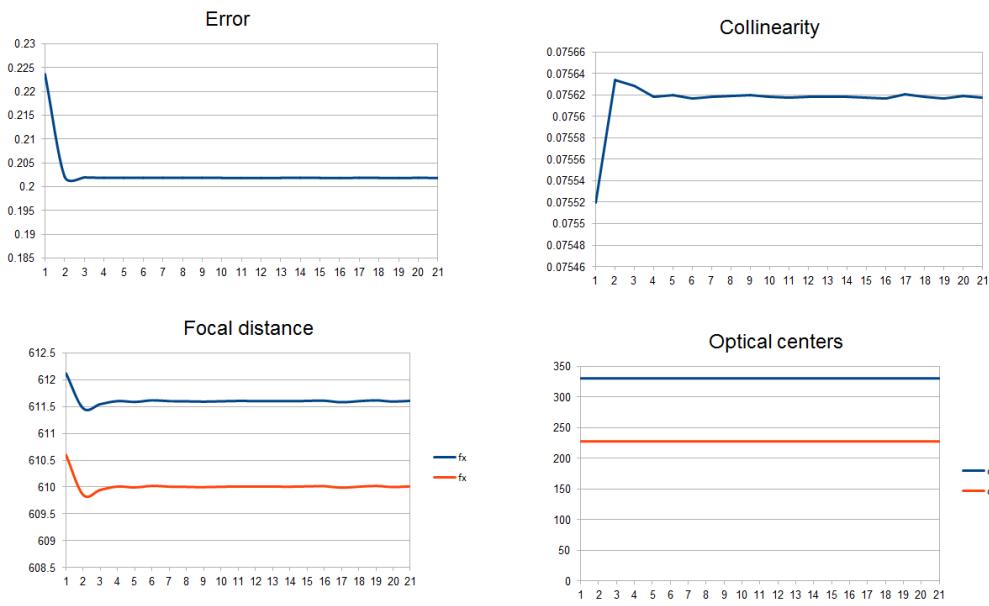


Fig. 27. Results of Life camera calibration with 30 frames and 20 iterations for asymmetric pattern.

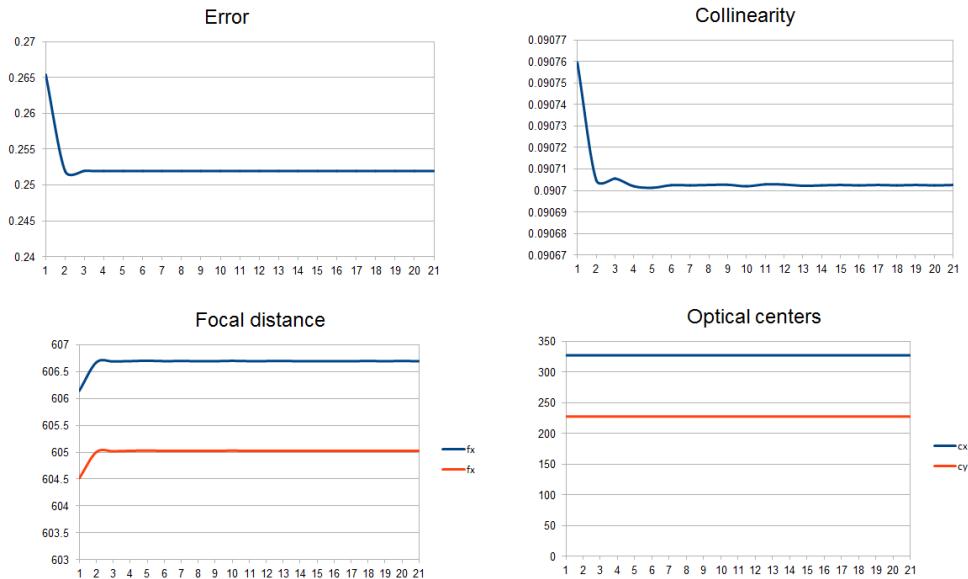


Fig. 28. Results of Life camera calibration with 40 frames and 20 iterations for asymmetric pattern.

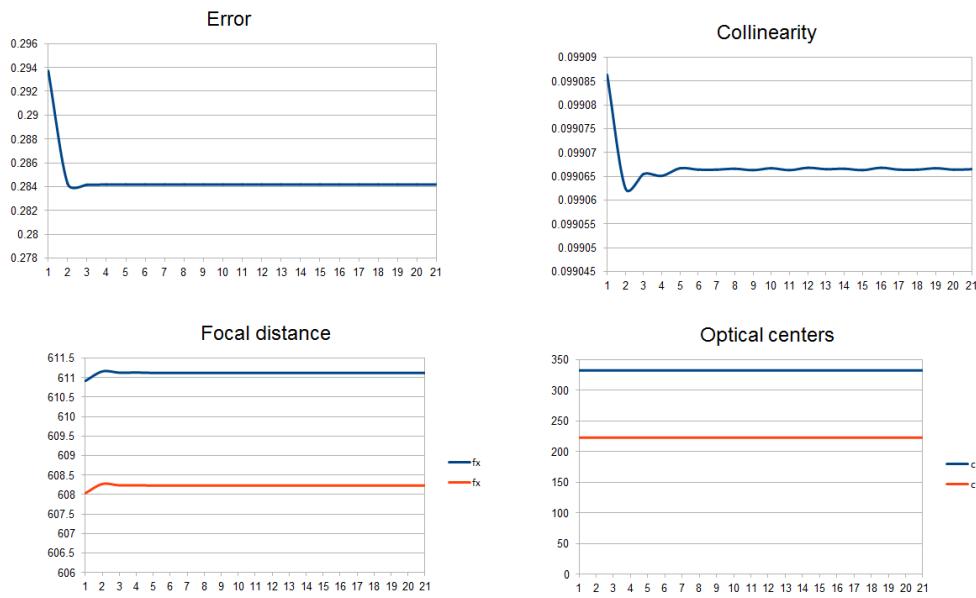


Fig. 29. Results of Life camera calibration with 40 frames and 20 iterations for asymmetric pattern.

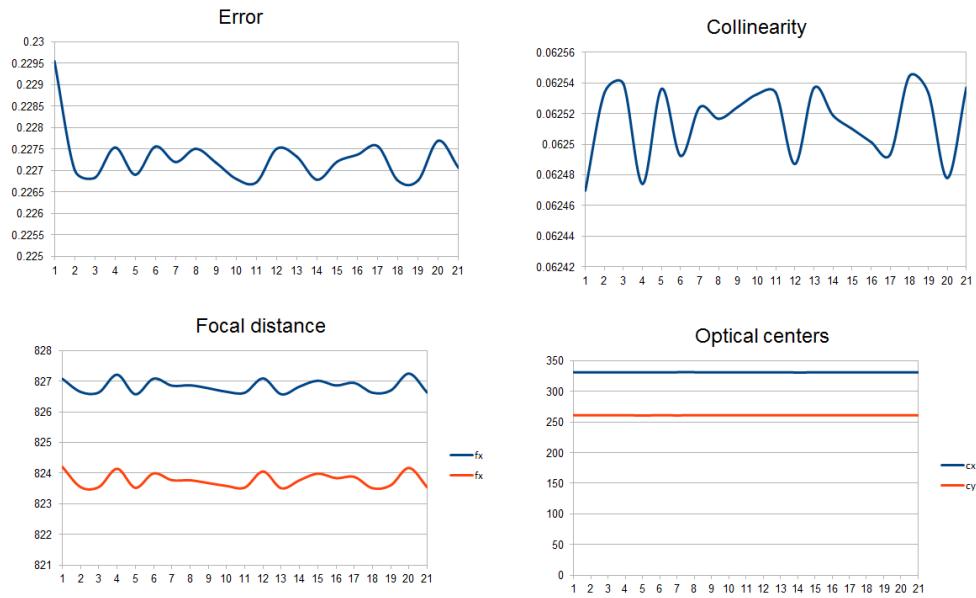


Fig. 30. Results of PS3 camera calibration with 25 frames and 20 iterations for asymmetric pattern.

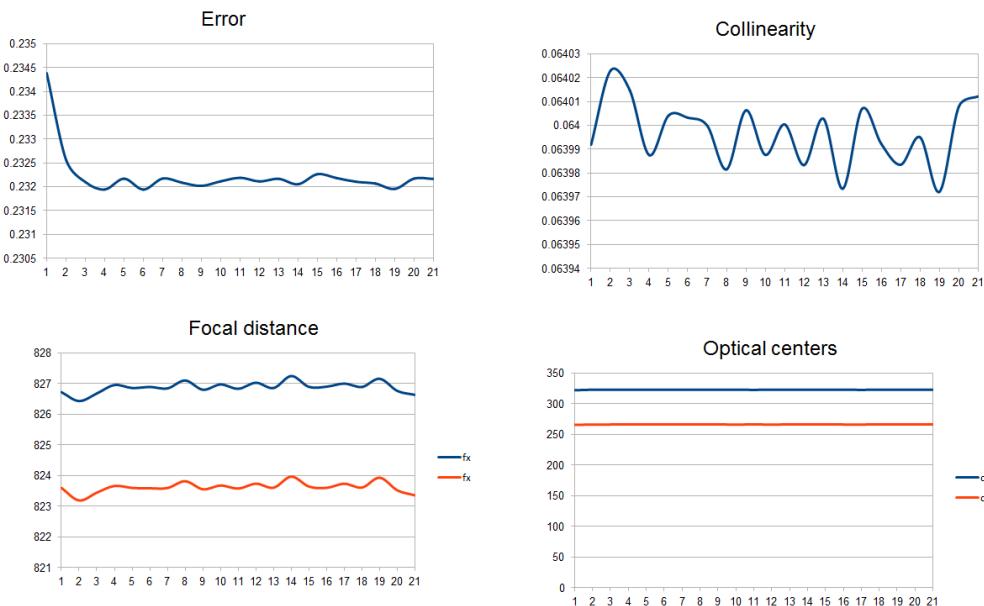


Fig. 31. Results of PS3 camera calibration with 30 frames and 20 iterations for asymmetric pattern.

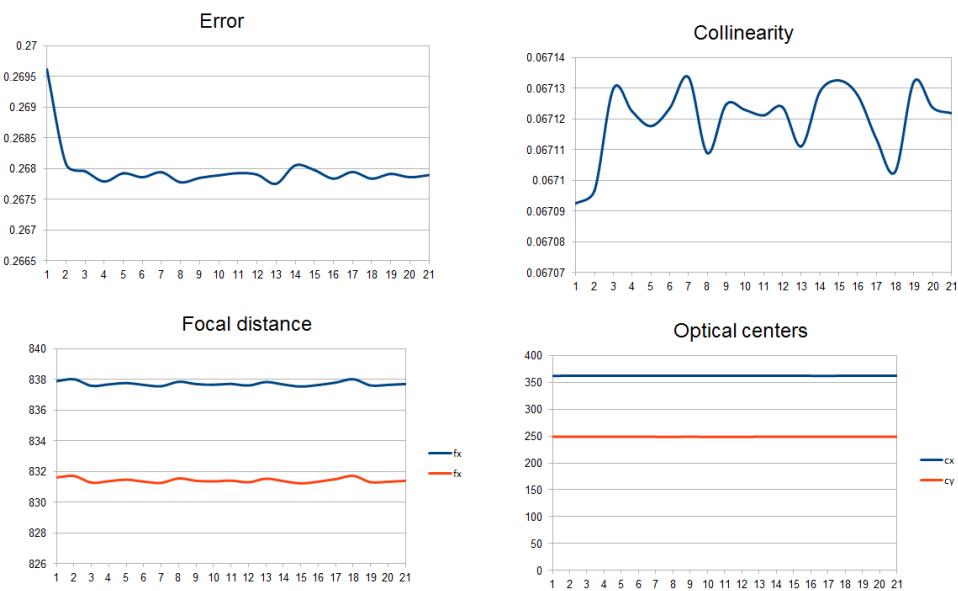


Fig. 32. Results of PS3 camera calibration with 40 frames and 20 iterations for asymmetric pattern.



Fig. 33. Results of PS3 camera calibration with 40 frames and 20 iterations for asymmetric pattern.